A DIFFERENTIAL GEOMETRIC APPROACH TO COMPUTER VISION AND ITS APPLICATIONS IN CONTROL

by

Yi Ma

Memorandum No. UCB/ERL M00/42

8 August 2000

A DIFFERENTIAL GEOMETRIC APPROACH TO COMPUTER VISION AND ITS APPLICATIONS IN CONTROL

by

Yi Ma

Memorandum No. UCB/ERL M00/42

8 August 2000

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Differential Geometric Approach to Computer Vision and its Applications in Control

by

Yi Ma

B.E. (Tsinghua University, P.R.China) 1995
M.S. (University of California at Berkeley) 1997
M.A. (University of California at Berkeley) 2000

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Shankar Sastry, Chair
Professor Jitendra Malik
Professor Alan Weinstein
Professor David Tse

Fall 2000

The dissertation of Yi Ma is approved:

	6/26/00
Chair	Date
	6/21/00
	Date
	6/5/2000
	Date
	6/12/00
	Date

University of California at Berkeley

Fall 2000

A Differential Geometric Approach to Computer Vision and its Applications in Control

by

Yi Ma

Abstract

A Differential Geometric Approach to Computer Vision and its Applications in Control

by

Yi Ma

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor Shankar Sastry, Chair

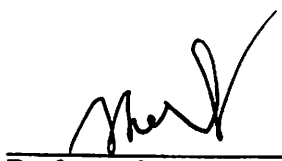
As an important feature of any autonomous mobile agent, such as the human or unmanned (ground and aerial) vehicles, there is usually a vision system embedded within the decision making loop. The role of the vision system, whether biological or artificial, is responsible for retrieving 3D information of the environment from 2D images. Such 3D information contributes to either low-level feedback control so as to safely navigate within and interact with the surroundings, or high-level decision making so as to reliably recognize, evade, pursue or manipulate 3D objects or coordinate with other agents.

Among all the cues available for computing 3D information, the motion cue (also called the stereo, parallax or structure from motion cues) provides the most unequivocal information about the camera motion, calibration and 3D structure. Thus the study of the motion cue has been the subject of intense research in the computer vision community. The majority of the results have been established primarily within a Projective Geometric framework which is not easily exploited by the control and robotics community.

In the first part of this dissertation, we show how to further use a blend of novel techniques in Differential Geometry, Estimation Theory, and Optimization to improve our understanding of the basic geometric laws which govern the visual perception. This new perspective has initiated a series of new developments in and geometric insights to almost every classic problem associated to the motion cue, such as motion estimation, structure recovery and camera self-calibration. In the end, we are able to reach a coherent mathematical theory for multiview geometry. This theory also helps us to discover and analyze certain

singularity, degeneracy and ambiguity inherent in the 2D to 3D reconstruction problem. Further more, the use of differential geometry allows us to extend the existing theory of multiview geometry to non-Euclidean spaces. The second part of this dissertation presents some initial attempts towards such a theory.

The proposed common mathematical framework between computer vision and control/robotics theory enables a better formulation of vision based control. In the third part of this dissertation, we will address two basic approaches to vision based control, namely visual servoing and visual sensing. These two approaches are demonstrated through two vision based control projects: vision based navigation of an unmanned ground vehicle and vision based landing of an unmanned aerial vehicle.



Professor Shankar Sastry
Dissertation Committee Chair

To my dear parents,

for always having faith in me.

Contents

List of Figures

List of Tables

Acknowledgements

The new millennium undoubtly injects a big dose of frenzy into the entire world, and I, at the last days of my PhD study, could not escape. As whether the year 2000 or 2001 is *the* turn of the millennium was still an ongoing debate, modern technology and media had already caused quite a hype at the end of year 2000, started with a sham Y2K bug. Before January 1, 2000, even after I had made three backup CDs of the draft of this dissertation, I still feared that there was not going to be any computer or printer alive after the attack of the Y2K bug. But *nothing* happened! I felt very stupid, even after I managed to convince myself that the emergency kit that I bought for Y2K might still be of some use in case of an earthquake. The celebration of the new year 2000 is therefore ruined – the fireworks at San Francisco's Embarcadero Center did not save it much. Not that I would feel better if something bad had happened, I was simply angry with the misleading media and mad at my poor judgment. So I have decided to celebrate the millennium at January 1, 2001 instead. At least by then I will have something real - my PhD degree and a new job - to celebrate, without any harassment from a "Y2K1" bug. Besides, it is such a unique honor to receive the highest academic degree at such a unique time in history: the first generation PhD of a new millennium. For that, I must sincerely thank all the wonderful people who have made it possible.

My deepest gratitude must go to my MS and PhD advisor Professor Shankar Sastry, who is, by all means, a devoted teacher, caring mentor and role model to all his students. It is his support, advice and even parenting that have helped me through those harsh early days in my graduate life. His knowledge, insight, vision, inspiration and encouragement have guided me through the wonder land of science and have taken me to the frontier of scientific inquiry. His pleasant personality and the respect he has for his students have certainly made such a journey extremely enjoyable. I will always be indebted to him for everything he has taught and given to me.

I would like to thank Professor Jitendra Malik, David Tse, and Alan Weinstein for serving on my Dissertation Committee. Professor Malik's expertise in computer vision is an extremely valuable source that enriches my knowledge and culture in this area. His suggestions have also motivated some of the work in this dissertation. Professor Weinstein is also my MA advisor in mathematics. He and Professor Tse have given unlimited support and encouragement for my PhD research.

There are many other professors who deserve a special thank note. It has been a very pleasant and fruitful research partnership with both Professor Stefano Soatto currently at Washington University at St. Louis (and soon moving to UC Los Angles), and Professor Jana Košecká at George Mason University. My research has also benefited from discussion and interaction with Professor David Forsyth at UC Berkeley, Professor Kostas Daniilidis at University of Pennsylvania, Professor Pietro Perrona at California Institute of Technology, Professor Joao Hespanha at Southern California University and Professor Claire Tomlin at Stanford University. Also Professor Ana Cannas da Silva, who is now at Instituto Superior Técnico, Portugal, will always be remembered for her excellent lectures on Symplectic and Riemannian Geometry during her stay at Berkeley.

It is always a blessing to be surrounded by a big group of intelligent and pleasant people at the Berkeley Robotics Lab for there is never lack of excellent research partners and wonderful friends. Part of this dissertation is joint work with some of the members: Shawn_Wayen Hsu, John Koo, Omid Shakernia and René Vidal. As both my friends and foosball partners, Cenk Cavusoglu, Xinyan Deng, Jianghai Hu, John Lygeros, George Pappas, Cory Sharp, Bruno Sinopoli, Joseph Yan, Jun Zhang, Lizhong Zheng have made my otherwise boring life at the office colorful and enjoyable.

I also like to acknowledge the financial support for my five-year graduate study. I thank UC Berkeley for awarding me the 1995 Regents' Fellowship and thank the Army Research Office for funding most of the remaining four years of my study and research under the Multi-disciplinary University Research Initiative (MURI) grant.

I finally thank my family, especially my parents, for being always supportive and having more faith in me than I do. Although they are certainly more concerned about my health than my career, I suppose that they would be just as happy about me finishing the degree. This dissertation is dedicated to them.

Yi Ma

Chapter 1

Introduction

"The real voyage of discovering consists not in seeking new lands, but in seeing with new eyes."

— Marcel Proust

1.1 Context and Motivation

According to a list recently released by the National Academy of Engineering, "imaging" is ranked as the 14th greatest engineering achievement of the 20th century. This is not surprising. Recording the world the same way as we perceive it through our eyes is by far *the* most effective way to keep and convey information. However, simply recording images is not enough. The tremendous amount of information contained in all the images still need to be processed, sorted, analyzed, extracted and utilized. The fact that the human brain can process visual data with remarkable efficiency and reliability has motivated and inspired the designing of computer vision systems to automate the process of extracting information from images. Unfortunately, the performance of state of the art techniques has not been anywhere near that of the human vision. For many reasons, "vision" seems to be a problem left for the 21st century.

Ever since ancient times, human vision has been a fascinating subject for mathematicians, artists, philosophers, photographers, psycho-physicists and neurobiologists. The colorful history of vision has certainly made it one of the most interdisciplinary endeavor in science. An encyclopedic account of the study of human vision can be found in a recent book by Stephen Palmer [88]. It is, however, not until late 1970s and early 1980s that

vision has been systematically studied from a *computational* viewpoint, *i.e.*, how to develop computational models which may simulate certain functionalities of human vision. Such an effort was initiated by pioneers such as David Marr [75]. In those two decades, much effort was devoted to the problems of recovering three dimensional shape from cues such as texture, shading or contour. While these topics remain to be active research subjects, in 1990s, further advances in computer and imaging technologies have enabled and boosted the study of motion analysis of multiple images or video sequences. The central problem is to recover the scene structure as well as the camera motion from many images taken of the *same* scene. In the computer vision literature, this is referred to as the **structure from motion** (SFM) problem. The geometric theory developed for the study of this problem is referred to as **multiview geometry**.

If we regard imaging roughly as a problem of generating realistic two dimensional images from a given three dimensional scene or structure, vision is then very much the *inverse* problem. This inverse problem by its very nature could be under-determined for different scenes or camera poses may generate the same set of images. This makes computer vision a very challenging subject: A systematic study therefore will not only include the design of general-purpose algorithms, but also consist of a clear understanding of potential singularity, degeneracy and ambiguity in the problem. Regarding the SFM problem in computer vision, its various geometric aspects have been extensively investigated in late 1980s and 1990s [22, 76, 131] whereas there is still need of a unified mathematical framework for reaching a full and satisfying understanding of the geometric nature of this problem. The *main* purpose of this dissertation is then to propose such a framework. However, we do not intend to encompass *all* existing and new results. Rather, we will emphasize on demonstrating how to complete and improve existing results in multiview geometry and how to approach new problems which were not able to be solved in the old paradigm.

In order for the reader to understand better the material covered, subjects studied and mathematics used in this dissertation, it is important that we explain:

1. Why we are interested in computer vision, especially multiview geometry;
2. How we started studying it in the first place;
3. What we are going to use it for.

Five years ago, Berkeley Intelligent Machines and Robotics Laboratory started an ARO

MURI program on "An Integrated Approach to Intelligent Systems". There have been two associated test-beds: intelligent vehicle highway and unmanned helicopter. The purpose of both test-beds is to develop intelligent unmanned (ground or aerial) vehicles. Computer vision has been considered as an option to replace some of the traditional navigation sensors: magnetic lane marks or inertial navigation sensors (INS) such as gyroscopes and accelerometers. Despite many of its advantages, computer vision, unlike most traditional sensors, is the least understood for control purposes. So our study first focused on investigating the role of **computer vision in a feedback loop** and how to design controllers around the vision sensor. Some of the results of this effort have been summarized in Part III of this dissertation. However, while we were studying vision based control, we realized that existing mathematical framework in the computer vision literature for studying SFM was not so compatible with that for control and robotics, and the existing theory for multiview geometry was not complete yet or unified enough for the purpose of designing robust control system based on computer vision. We therefore decided to investigate the the problem of SFM in more depth. The significance of such an investigation is believed to be three-fold:

1. We hope to unify, improve and complete existing results in multiview geometry so that it directly benefits the computer vision community;

2. We try to present a clear picture of this subject within a unified geometric framework which will be more accessible to the control and robotics community.

3. We want to establish a solid geometric theory of SFM which may give useful guidance for the design of better vision based control systems.

Part I and Part II of this dissertation summarize our effort and main results in these directions.

1.2 Research Areas

1.2.1 Vision Based Control System Hierarchy

As we have mentioned above, multiview geometry *per se* is not the only interest of our study. Our ultimate goal is to develop *intelligent* unmanned vehicles with computer vision in the feedback control or decision making loop. Multiview geometry is one of the most important subjects, of which we need to have a very good understanding, in order

to achieve such a goal. Although the architectures of such intelligent systems may be very different depending on applications, conceptually they always can be decomposed into a three-layer hierarchy, shown in Figure 1.1. One must be aware that the three layers in

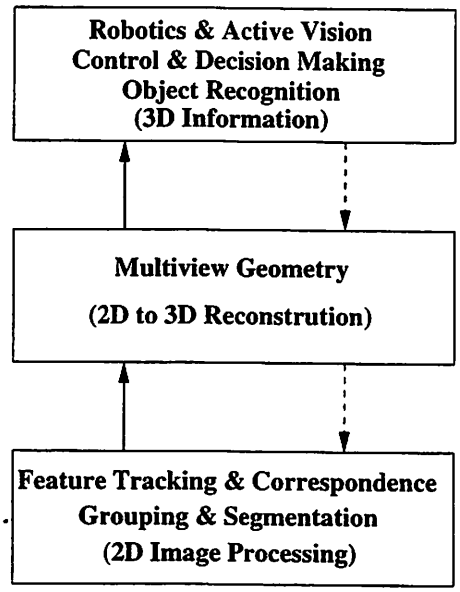


Figure 1.1: A conceptual hierarchy of vision based control (or decision making) systems. Arrows indicate direction of information flow.

this hierarchy are still *coupled* together. For example, the bottom layer provides input information (such as image correspondences and optical flows) to the middle layer, which multiview geometry uses to recover 3D structure and camera motion. In the other direction, knowledge about the structure and motion will certainly improve the accuracy of matching up corresponding image points. A similar coupling also exists between the top two layers. For example, 3D structure recovered from the multiview geometry may be necessary for recognizing certain 3D objects. In return, recognition of a 3D object may dramatically improve the 3D structure recovered from its 2D images.

Because of these couplings, the study of the overall system is extremely complicated and almost intractable. A traditional method to approach such a complicated problem is *divide and conquer*. This dissertation will follow this old tradition. For example, in Part I and Part II, we will focus our study on the second layer, *i.e.*, multiview geometry. We are going to indulge ourselves and assume that there is no coupling with either the top or the bottom layer. That is, we do not assume any knowledge about the object whose 3D structure is to be recovered, nor do we consider the effect of the reconstructed 3D structure

and camera motion on the measurements of image correspondences or optical flows. Due to these assumptions, we will then be able to formulate multiview geometry as a clean mathematical problem and investigate it in depth.

1.2.2 Multiview Geometry

Ever since the landmark paper by Longuet-Higgins in 1981 [60], the study of the geometry of 2D to 3D reconstruction has been revived. This revival has led to a blooming of numerous algorithms on the problem of recovering 3D structure and motion from feature image points. These algorithms differ in many aspects:

1. Linear versus Nonlinear (Suboptimal versus Optimal);
2. Discrete versus Continuous;
3. Two-view versus Multiview;
4. Calibrated Camera versus Uncalibrated Camera;
5. Batch Methods versus Iterative Methods;
6. Orthographic Projection versus Perspective Projection;
7. Euclidean versus Riemannian.

For most of the aspects, a thorough and detailed account of the state of the art techniques can be found in later chapters.

Because the structure from motion problem has been so extensively studied, it is then a very tempting, however extremely challenging task to try to encompass all the existing results in a unified theoretic framework. In the computer vision literature, a well celebrated framework is a three-stage stratification approach proposed by Faugeras in 1995 [23]. The basic concept may be roughly shown as in Figure 1.2. Based on Projective Geometry, this approaches decomposes the original complicated nonlinear SFM problem to a series of subproblems, each of which has an easier, or even linear, solution.

According to this stratification hierarchy, in order to obtain a final reconstruction of the Euclidean structure and motion, one first seeks for a relaxed solution in a projective space, *i.e.*, finding the solution up to an arbitrary projective transformation. Such a

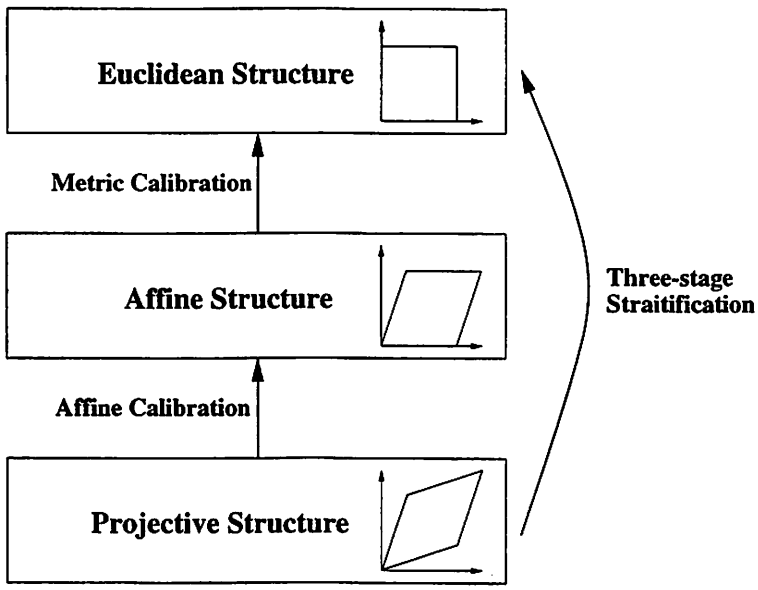


Figure 1.2: The hierarchy of the three-stage stratification approach.

solution is then referred as a **projective structure**. To a large extent, solving for the projective structure is a *linear* problem. Once the projective structure is obtained, extra *metric* information is used to "stratify" this structure back to a Euclidean structure, through an intermediate stage: an affine structure.

A potential gain of such a projective geometric framework is in computation. The computation of the projective structure is very much linear (therefore fast). Moreover, the algorithm for each step is relatively robust and provides already good estimates even in presence of noises, although such estimates may not be unbiased or optimal w.r.t. a given noise model. However, seeking for a solution in a projective space may easily lose track of geometric insight of what is exactly going on in the original Euclidean space. The reduction of the rather geometric SFM problem to an algebraic one makes it harder to fully reveal the geometric intuition behind the results and algorithms, and the focus on the design of general-purpose algorithms may also take a risk with potential singularity, degeneracy and ambiguity hidden in the original problem. Furthermore, we do need a more *delicate* framework which will be able to unify all the results regarding different aspects of SFM, as mentioned in the beginning of this section. Part I and Part II of this dissertation then attempt to give a new perspective to multiview geometry which practically uses no projective geometry but provides a clear resolution to these issues. Conceptually, this approach can

also be interpreted as an alternative stratification of motion and structure separately, as illustrated in Figure 1.3. The gist of this approach is presented in detail in Chapter 6.

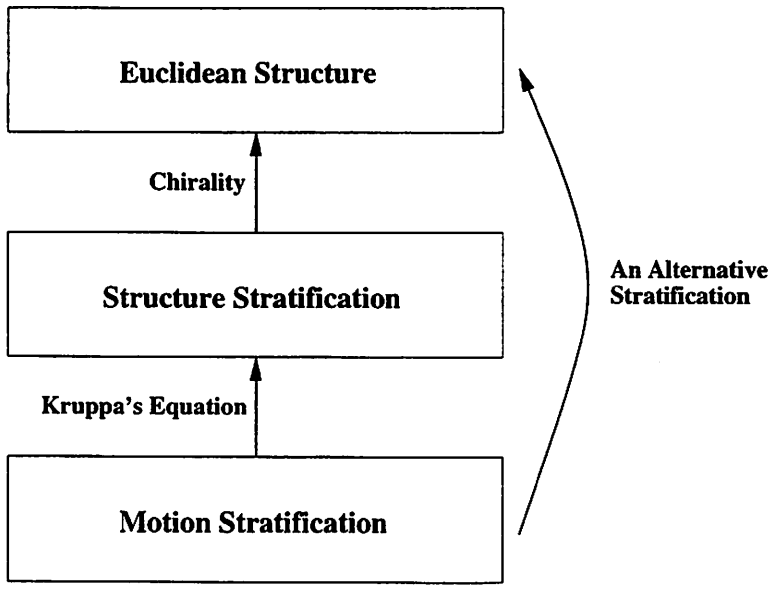


Figure 1.3: The hierarchy of an alternative stratification approach. Details about chirality and Kruppa's equation can be found in Chapter 6.

1.2.3 Vision Based Robotic Control

As one of its main applications, computer vision has been widely used in robotics for many purposes: autonomous navigation, obstacle avoidance, object recognition or manipulation, 3D map building and telepresence. In such a context, an important question that naturally arises is:

> *How should the information from vision sensors be used for robotic control purposes?*

A naive approach would be to first recover all 3D information that vision could possibly provide and then design feedback laws for a given control task based on all the information. However, many 2D to 3D estimation schemes are rather time-consuming and not yet suitable for real-time control tasks. This has been the motivation for the so called **visual servoing** approach, *i.e.*, to design feedback control laws based on measurements which are *directly* available from images, hence certain unnecessary 2D to 3D estimation can be bypassed. In general, the physical robot dynamics are first "lifted" onto the image plane and result

in *induced* dynamics of certain image features or quantities. If a given control task can be expressed in terms of the states of such image dynamics, we may design control laws directly based on these image quantities.

A problem with the visual servoing approach is that it does not apply well to robots with complicated dynamics: The induced dynamics of image features could easily become intractable for consequent control analysis or synthesis. In such a case, it is then more feasible to keep vision and control separate. However, to reduce the amount of 2D to 3D estimation, we must only request vision to provide estimates of states which are essential to achieve the control task. The success of such a vision based control system then relies on a good balance between what control needs and what vision can (efficiently) provide. In this dissertation, we will (informally) refer to this approach as **visual sensing**. Table 1.1 summarizes a conceptual comparison between the visual servoing and visual sensing approaches. In Part III of this dissertation, these two approaches will be compared through two concrete examples: vision guided driving of a ground vehicle and vision guided landing of an aerial vehicle.

Table 1.1: A comparison of visual servoing and visual sensing

	Visual Servoing	Visual Sensing
Space	$\mathbb{RP}^2$	$\mathbb{R}^3$
Dynamics	Lifted	Natural
States	Image Quantities	Physical Quantities
Estimation	2D to 2D	2D to 3D
Vision and Control	Coupled	Separate

1.3 Dissertation Contributions

The nine main chapters (Chapters 3 to 11) in this dissertation are essentially from seven (journal or conference) papers and two (yet to be published) technical reports which I have written during a span of four years on the subjects of computer vision (mostly multi-view geometry) and vision based control. Therefore, each chapter alone consists of a rather self-contained story. At the time each paper was written, there were always very specific reasons and technical contributions to the problem studied. The reader may find a more detailed account in the introduction to each chapter. Here, for the reader to understand

better the gist of the *overall* dissertation, I would like mention a few things which highlight the contributions of this dissertation, at a more conceptual level.

1.3.1 A Differential Geometric Framework for Multiview Geometry

The optimism caused by early success of the projective geometric framework (discussed in Section 1.2.2) has made many people think that structure from motion is already a "solved" problem. If so, it is then natural to ask:

What, if anything, is new in multiview geometry?

As an indirect response to this question, we ask instead a different question, simply out of curiosity: How much is projective geometry really needed for understanding the problem of structure from motion? As the reader will see, this dissertation is going to cover almost every important subject in multiview geometry whereas no projective geometry will be used at all, nor do we assume the reader have any knowledge in projective geometry. Moreover, not only will many existing results be unified, improved and even corrected in the new approach, but also many new problems will be raised and solved which cannot be easily studied in the old paradigm. Many of the new proofs and results have shown how primitive some of our knowledge on this subject yet is. Multiview geometry is still at a young stage where almost everything needs to be organized, clarified or given a better (geometric) interpretation. Is the new perspective or new approach introduced in this dissertation going to lead it to maturity? Maybe or maybe not. But a controversy has certainly been raised:

What if, anything is new in multiview geometry?

We can list a few things in this dissertation to support this point:

1. For computing *discrete* camera motion from image correspondences between two views, there has been a well celebrated three-step singular value decomposition (SVD) based linear algorithm discovered by Huang and Faugeras *el al* in 1980's [24, 119]. However, there has not been much success in finding the continuous counterpart of this algorithm until a new geometric viewpoint is introduced which unifies the discrete and continuous cases (see Chapter 3).

2. The purely algebraic approach to study the constraints among multiple images has been successful, but at a higher price: Heavy machinery from algebraic geometry

must be deployed, and the results lack geometric intuition [43]. Nonetheless, a much easier proof of the same results can be obtained from a new geometric perspective. In addition to the algebraic relationship, both a geometric and statistical relationship can also be revealed in this way (see Chapter 5).

3. The so called Kruppa's equation has been discovered since 1913 and then revived in 1990s for the purpose of camera self-calibration [77]. However, the projective geometric interpretation of this equation has done little in terms of discovering its degeneracy. Such degeneracy is discovered however from a dramatically different geometric interpretation of Kruppa's equation (see Chapter 6).

These new results and the way they are discovered encourage us to think twice about what is an appropriate framework of multiview geometry. At least, they make us no longer so confident with the projective geometric framework.

Mathematically speaking, multiview geometry can be viewed as a geometry which studies the combination of a (motion) group action on a space and a (perspective) projection transformation. In the default case, the motion group is the special Euclidean group $SE(3)$ acting on the space $\mathbb{R}^3$ and the projection is the standard perspective projection $\pi : \mathbb{R}^3 \to \mathbb{RP}^2$. In the projective geometric approach, with an emphasis on the effect of the perspective projection, the motion group $SE(3)$ is artificially extended to the general linear group $GL(4)$.[1] From such a point of view, we can study vision under more general classes of (motion) groups. For example, if we choose the motion group to be the isometry group of a Riemannian manifold, with a proper interpretation of the "projection map", we then can study multiview geometry on such a manifold. For this scheme to work, concepts and techniques from **differential geometry** must be deployed. Chapter 8 presents some of the preliminary results towards this direction. It basically extends the results that we have for multiview geometry in a Euclidean space to spaces of constant curvature.

This is by no means the only reason why we name our approach "a differential geometric approach". Although we emphasize that almost the entire dissertation is very much based on linear algebra and basic knowledge of rigid body motion, differential geometry does serve well as a conceptual framework which provides geometric intuition, interpretation and various techniques to almost every problem that we have studied, for example:

[1] The space $\mathbb{R}^3$ accordingly is extended to $\mathbb{P}^3$.

1. The unification of discrete and continuous linear algorithms relies on a clean geometric characterization of the space of essential and continuous essential matrices (see Chapter 3).

2. The nonlinear algorithms for obtaining optimal (or suboptimal) estimates rely on modern optimization techniques for special classes of Riemannian manifolds (see Chapters 4 and 5).

3. The proof of geometric dependency of constraints among multiple images relies on a clever trick on a quotient space of a Grassmann manifold (see Chapter 5).

4. The discovery of degeneracy of Kruppa's equation relies on a new interpretation of Kruppa's equation as inner product invariants of certain isometry group action (see Chapter 6).

5. A classification of generic ambiguities in the problem of 2D to 3D reconstruction is done with respect to every Lie subgroup of $SE(3)$ (see Chapter 7).

Another reason is that differential geometry has been widely adopted in the study of linear/nonlinear system theory and modern robotics. A theory of multiview geometry based on such a language will be more accessible to people in these communities and provide a more unified framework for the study of vision based robotic control. Because of this, we are able to use the same language throughout the entire dissertation: Part I and Part II (Chapters 1 to 9) on multiview geometry and Part III (Chapters 10 and 11) on vision based robotic control.

The differences between the projective and the differential geometric frameworks can be summarized in Table 1.2. However, it would be unfair to simply claim that either framework is better than the other since each framework is proposed for a different purpose. As we have mentioned in Section 1.2.2, the projective geometric approach has certain computational advantage. On the other hand, the differential geometric framework is proposed for a better geometric insight and stronger connection with control and robotics.

1.3.2 Geometry, Estimation and Optimization

Multiview geometry is a very peculiar subject: The problem itself can be formulated as a pure mathematical one (see Chapter 8); however, traditionally it has been

Table 1.2: A comparison of projective and differential geometric frameworks

	Projective Geometry	Differential Geometry
Spaces: Groups	$\mathbb{P}^3 : GL(4)$	$\mathbb{R}^3 : SE(3)$
Mathematics	Algebraic	Geometric
Metric	None	Euclidean
Invariants	Projective Invariants	Euclidean Invariants
Compatibility with Control	Weak	Strong

studied for mostly practical purposes (in computer vision or robotics community). Many of the existing results have been developed for very specific applications, rather than in a unified theoretical program. This makes multiview geometry both a theoretical and applied subject. We not only need a theory studying its geometric nature, but we also need efficient algorithms which provide robust solutions to the problem. Especially, in a practical situation, the obtained images and measurements are always noisy. It is then crucial to obtain statistically unbiased estimates. If such estimates are given as solutions to certain optimization problems, we then need to know what are the proper optimization techniques to apply.

In this dissertation, besides studying the geometric aspects of multiview geometry, we also focus on an estimation theoretic approach to the structure from motion problem. In many occasions, it helps us to gain a better understanding of the problem from an algorithmic viewpoint. Our study has revealed a close inter-relationship among geometry, estimation and optimization in SFM. As we will show later in this dissertation, SFM in general is an estimation problem with hard geometric constraints and the resulting optimization problem is mostly optimization on some special (and well-structured) geometric spaces (see Chapters 4 and 5).

1.3.3 Singularity, Degeneracy and Ambiguity

As we have mentioned before, the problem of structure from motion by nature is an inverse problem from 2D images to 3D structure and motion, and it may not be well-determined. That is, there is likely inherent ambiguity in the solutions, or singularity and degeneracy in the general-purpose algorithms. For example, the necessary and sufficient conditions for being able to *uniquely* recover camera motion, calibration and 3D scene structure from a sequence of images are very rarely satisfied in practice. We then need to

know:

> *What exactly can be recovered in image sequences of practical importance when such conditions are not satisfied?*

In Chapter 7, we will give a complete answer to this question. For every camera motion subgroup that fails to meet the conditions, we gave explicit formulas for the ambiguities in the reconstructed scene, motion and calibration. Such a characterization is crucial both for designing robust estimation algorithms that do not try to recover parameters that cannot be recovered and for generating novel views of the scene by controlling the vantage point.

As another example, Kruppa's equation [58] has been widely used to solve the problem of camera self-calibration. Although first discovered in 1913 by Kruppa and later revived in 1993 by Maybank and Faugeras [77], the algebraic nature of this equation has never been clearly understood. In fact, Kruppa's equation tends to become degenerate under certain conditions. Hence any general-purpose self-calibration algorithm based on Kruppa's equation may become ill-conditioned when applied to real image sequences. Our analysis in Chapter 6 further shows that under the conditions when degeneracy occurs, Kruppa's equation can however be normalized. Such normalization not only resolves the degeneracy but also makes Kruppa's equation linear. This in fact makes self-calibration relatively easier under the conditions when degeneracy occurs. Moreover, from the new results, one may also achieve a clear understanding of the relationship between Kruppa's equation and all the other methods for self-calibration such as the ones based on absolute quadric constraint, modulus constraint or chirality (see Chapter 6).

1.3.4 Applications in Unmanned Ground and Aerial Vehicles

The emphasis of this dissertation is on the theory of multiview geometry. Although such a theory may have its impact on many conventional applications of multiview geometry, in this dissertation, we are more interested in its usage in vision based robotic control. For that purpose, we have conducted two case studies: a vision guided navigation scheme for **unmanned ground vehicle** (UGV) and a vision based landing system for **unmanned aerial vehicle** (UAV) (in our case, a helicopter). In both studies, estimation issues for the vision sensor and stability issues for the overall closed-loop system are successfully studied together under a unified geometric control framework (see Chapters 10 and 11).

In the unmanned ground vehicle case, instead of using point features as the entire Part I and Part II do, we demonstrate how to analyze curve features in presence of mobile dynamics. In the unmanned aerial vehicle case, we show why and how the general-purpose algorithms given in Part I should be customized to a specific situation. That is, in the case of landing, the standard motion estimation algorithms need to be modified in order to incorporate the extra knowledge that the feature points are all lying a planar surface. Moreover, these two case studies serve for a comparison between the visual servoing and visual sensing approaches of vision based control (discussed in Section 1.2.3).

1.4 Dissertation Outline

1.4.1 Overview

The main body of this dissertation consists of three parts, a total of ten chapters (Chapters 1 to 11) and two appendices (Appendixes A and B):

- **Part I – Multiview Geometry: A Differential Geometric Approach** (Chapters 2 to 7)
- **Part II – Advanced Topics in Multiview Geometry** (Chapters 8 and 9)
- **Part III – Applications: Vision Based Robotic Control** (Chapters 10 and 11)

Part I essential covers the main theory of multiview geometry. Chapter 2 formulates the problem of structure from motion in a Euclidean space. The formulation ensures that the whole dissertation is self-contained. The camera motion, camera imaging model and the two types of image measurements: image correspondences and optical flows are clearly defined in this chapter. It is also the reference chapter of all the notation used throughout the entire dissertation. For the rest of Part I, we partition the structure from motion problem into four interrelated topics or subproblems:

1. Motion and structure from two views.
2. Motion and structure from multiple views.
3. Camera self-calibration.
4. Euclidean reconstruction and reprojection up to subgroups.

The results regarding these topics together form a coherent theory for the multiview geometry. For the first two topics, to simplify the analysis, only calibrated camera model will be considered (see Chapters 3, 4 and 5). We will especially study the geometry of an uncalibrated camera in the third topic (see Chapter 6). In the first three topics, our primary interest is in conditions and algorithms for obtaining a *unique* solution. In the final topic, we will provide a complete characterization of the structure of the set of ambiguous solutions when conditions for a unique solution fail (see Chapter 7).

Part II consists of two independent advanced topics of multiview geometry. A generalization of multiview geometry to non-Euclidean spaces is given in Chapter 8. Chapter 9 provides a (Bayesian) justification of the approach of using feature points for motion estimation based on a simplified stochastic model of imaging.

Part III demonstrates the use of vision in robotic control through two case studies. Chapter 10 studies the problem of a ground mobile robot tracking a given ground curve using on-board camera as the only sensor. The visual servoing approach is applied to this problem. Chapter 11 investigates the problem of landing a helicopter on a ship deck. It serves as an example for the visual sensing approach for vision based control. Since the feature points are now lying on a planar surface, we also study how the motion estimation algorithms given in Chapter 3 need to be modified for the planar case.

1.4.2 Guidelines for Readers

Since this dissertation covers a relatively large amount of material, we have tried our best to reduce cross reference among chapters to the minimum so that readers with different backgrounds and interests do not have to read the dissertation in a linear fashion. Figure 1.4 illustrates the inter-dependency among all the ten chapters (and two appendixes also):

Although "differential geometry" is in the title of the dissertation, the reader should be able to grab the gist of most of Part I with a background in linear algebra, basic robotics and nonlinear programming only. Only basic differential geometric terms are used in Chapters 4 and 5 for optimization on manifolds and in Chapter 6 for a geometric characterization of fundamental matrix and Kruppa's equation. However, readers who are not familiar with these terms can simply skip the related sections without loss of much continuity. However, differential geometry is seriously used in Chapter 8 of Part II for a

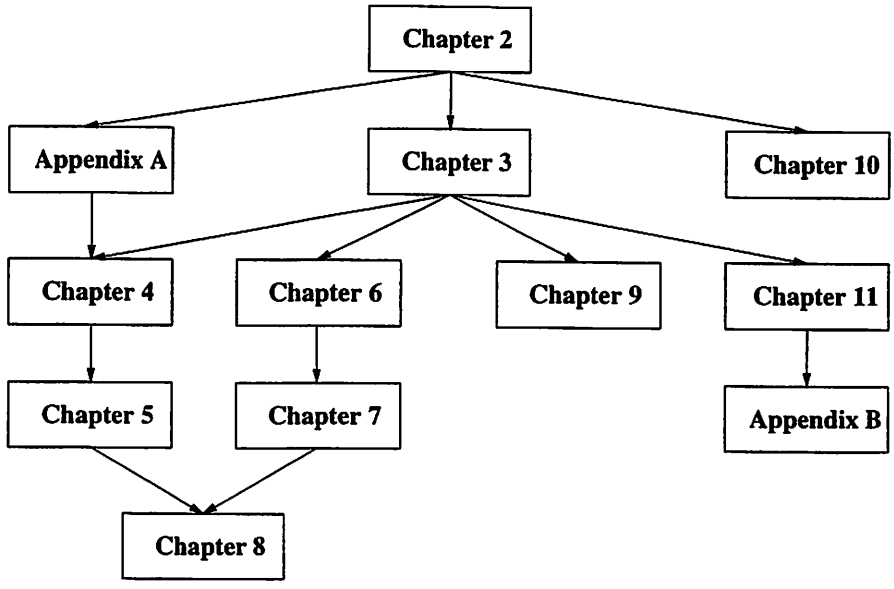


Figure 1.4: Dependency among the chapters and appendixes.

generalization of multiview geometry to non-Euclidean spaces and in Chapters 10 and 11 of Part III for the analysis of nonlinear control systems. For good references on the subject of differential geometry, we recommend the book by Boothby [5] or the one by Kobayashi and Nomizu [55], on the subject of nonlinear control systems, we suggest the book by Sastry [93].

For people with different interests, this dissertation can be read as different packages:

- *Classic Multiview Geometry and Algorithms*
  Chapters 2 to 6, and 9.

- *Theoretical Multiview Geometry (Euclidean and Non-Euclidean)*
  Chapters 2, 3, 5 to 8.

- *Vision Based Robotic Control*
  Chapters 2, 3, some of Chapter 5, and Chapters 10 and 11.

Material from Chapters 2 to 4, 6 and 10, with some supplementary material from the robotics book by Murray, Li and Sastry [84] has been covered as a one semester graduate level course on computer vision and robotics by Professor Košecká at Berkeley in Fall, 1999.

1.5 Notation

The next chapter provides a systematic introduction to the notation that we are going to use in this dissertation. Nevertheless, I would like to have a few words about it before we start. Due to the wide scope of areas covered by this dissertation, we have to make a compromise between the conventional notation used in computer vision and that used in robotics or control theory. For example, we will use $\widehat{p}$ to represent the skew symmetric matrix:

$$\widehat{p} = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}$$

associated to a given vector $p = [p_1, p_2, p_3]^T \in \mathbb{R}^3$. Due to this definition, we then have $p \times q = \widehat{p}q$ for all $q \in \mathbb{R}^3$. This notation is widely used in robotics and matrix Lie group theory. However, traditionally, in the computer vision literature, people prefer to use $p_\times$ instead of $\widehat{p}$. Also, in the computer vision literature, $\omega \in \mathbb{R}^3$ is usually used to represent the absolute conic, we here however have to reserve it for the angular velocity since we are dealing with both the discrete and continuous time cases. We will use $S \in \mathbb{R}^{3\times 3}$ instead to represent the absolute conic, which is however going to be under a different name: metric. The rest of the notation is very consistent to robotics notation used in [84], except that we use $T \in \mathbb{R}^3$ for the translation vector and p for coordinates of a point.[2] We will use bold lower-case symbols to represent image quantities. For example, $\mathbf{x} \in \mathbb{R}^3$ is for coordinates of the image point and $\mathbf{u} \in \mathbb{R}^3$ is for the optical flow. This is very much consistent with notation used in the computer vision literature. Finally, all vectors are *column vectors*!

[2] In [84], $p \in \mathbb{R}^3$ is used for the translation vector and $q \in \mathbb{R}^3$ is used for coordinates of a point.

Part I

Multiview Geometry: A Differential Geometric Approach

Chapter 2

Problem Formulation

"As is well known, geometry presupposes the concept of space, as well as assuming the basic principles for constructions in space."
— G. F. B. Riemann, *On the Hypotheses Which Lie at the Foundations of Geometry*

2.1 Camera Model in a Euclidean Space

We begin by introducing the mathematical model of a camera in a three dimensional Euclidean space. We also introduce the notation which will be consistently used throughout this dissertation.

2.1.1 The Three Dimensional Euclidean Space

Consider that a camera is set in a **three dimensional Euclidean space** $\mathbb{E}^3$. We use p to denote a **generic point** in $\mathbb{E}^3$. $\mathbb{E}^3$ is then isometric to $\mathbb{R}^3$ with its standard metric. For convenience, $\mathbb{E}^3$ is usually considered as a hyper-plane embedded in $\mathbb{R}^4$ and every point p in $\mathbb{E}^3$ can be represented by **homogeneous coordinates** of the form:

$$p = [X_1, X_2, X_3, 1]^T \in \mathbb{R}^4. \tag{2.1}$$

In this expression, the effective part $[X_1, X_2, X_3]^T \in \mathbb{R}^3$ will be referred as the **three dimensional coordinates** of the point $p \in \mathbb{E}^3$. To separate them from the homogeneous ones, we denote them by $\mathbf{X} \in \mathbb{R}^3$:

$$\mathbf{X} = [X_1, X_2, X_3]^T \in \mathbb{R}^3. \tag{2.2}$$

In order to define the camera model properly, we also need the notion of a **vector**. In a Euclidean space, a vector can be simply defined to be the difference between two points with one of them as the starting point (or base point). The set of all vectors in $\mathbb{E}^3$ with the starting point p is denoted as $T_p\mathbb{E}^3$ *i.e.,* $T_p\mathbb{E}^3$ is the **tangent space** of $\mathbb{E}^3$ at p. By this definition, a vector $u \in T_p\mathbb{E}^3$ in homogeneous representation has the form:

$$u = [u_1, u_2, u_3, 0]^T \in \mathbb{R}^4. \tag{2.3}$$

As a vector space, $T_p\mathbb{E}^3$ is isomorphic to $\mathbb{R}^3$. A non-redundant representation of the same vector $u \in T_p\mathbb{E}^3$ is just:

$$u = [u_1, u_2, u_3]^T \in \mathbb{R}^3. \tag{2.4}$$

The **Euclidean metric** $\Phi : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ on $\mathbb{E}^3$ is simply given by the inner product: $\Phi(u, v) = u^T v$ for all $u, v \in T_p\mathbb{E}^3$.

2.1.2 Camera Motion

The isometry (metric preserving diffeomorphism) group of $\mathbb{E}^3$ is the so called **Euclidean group**, denoted by $E(3)$. The motion of the camera is usually modeled as the subgroup of $E(3)$ which preserves the orientation of the space $\mathbb{E}^3$, *i.e.,* the so called **special Euclidean group** $SE(3)$. In the homogeneous representation, $SE(3)$ can be represented as:

$$SE(3) = \left\{ \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \;\middle|\; R \in SO(3), T \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4\times 4} \tag{2.5}$$

where $SO(3)$ is the space of 3×3 rotation matrices (orthogonal matrices with determinant $+1$). We know that the isotropy group of $\mathbb{E}^3$ leaving a point $p \in \mathbb{E}^3$ fixed is the orthogonal group $O(3)$. $SO(3)$ is just the subgroup of $O(3)$ which is the connected component containing the identity $I \in O(3)$. Given an element $g \in SE(3)$ and a point $p \in \mathbb{E}^3$, g maps the point $p \in \mathbb{E}^3$ to a new one $gp \in \mathbb{E}^3$.

Since the motion between a camera and points in the world is *relative*, without loss of generality, we can and will assume throughout the dissertation that:

Assumption 2.1. *It is the camera which is moving and the world is static.*

We use a curve $g(t) \in SE(3), t \in \mathbb{R}$ to represent the rigid body motion of the camera, *i.e.*, the displacement of the camera coordinate frame F_t at time t, relative to its initial coordinate frame F_{t_0} at time t_0. By abuse of notation, the group element $g(t)$ serves both as a specification of the configuration of the camera and as a transformation taking the coordinates of a point $p \in \mathbb{E}^3$ relative to the F_{t_0} frame to those relative to the F_t frame. Clearly, $g(t)$ is uniquely determined by its rotational part $R(t) \in SO(3)$ and translational part $T(t) \in \mathbb{R}^3$. Sometimes we denote $g(t)$ by $(R(t), T(t))$ as a shorthand for its homogeneous representation. Let $p(t) \in \mathbb{R}^4$ be the homogeneous coordinates of a point $p \in \mathbb{E}^3$ relative to the camera frame at time $t \in \mathbb{R}$. Then the **coordinate transformation** of p under the motion $g(t)$ is given by:

$$p(t) = g(t)p(t_0). \tag{2.6}$$

In three dimensional coordinates, the above is simply:

$$\mathbf{X}(t) = R(t)\mathbf{X}(t_0) + T(t). \tag{2.7}$$

This relationship is intuitively shown by Figure 2.1. To obtain a continuous version of the

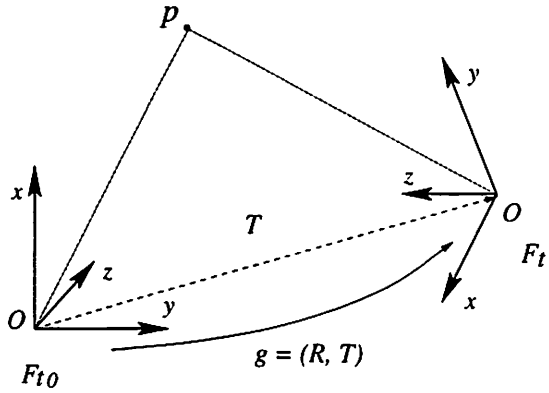


Figure 2.1: Coordinate frames for specifying rigid body motion of a camera.

equation (2.6) we differentiate it with respect to time t:

$$\dot{p}(t) = \dot{g}(t)p(t_0). \tag{2.8}$$

Since $g(t)$ is a curve in the Lie group $SE(3)$, $\dot{g}(t)$ must be of the form:

$$\dot{g}(t) = g(t)\xi(t). \tag{2.9}$$

Then $\xi(t)$ is an element of the Lie algebra $se(3)$ of $SE(3)$:

$$se(3) = \left\{ \left[\begin{array}{cc} \widehat{\omega} & v \\ 0 & 0 \end{array} \right] \Big| \, \widehat{\omega} \in so(3), v \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4\times 4} \tag{2.10}$$

where $so(3)$ is the Lie algebra of the rotation group $SO(3)$, or equivalently, the space of 3×3 skew symmetric matrices. In the above definition we already adopt the convention that, for any vector $\omega \in \mathbb{R}^3$, $\widehat{\omega}$ is the associated skew symmetric matrix such that $\widehat{\omega}u = \omega \times u$ for all $u \in \mathbb{R}^3$. Using the above notation, we immediately obtain the continuous version of the coordinate transformation (2.6):

$$\dot{p}(t) = g(t)\xi(t)g^{-1}(t)p(t). \tag{2.11}$$

It is direct to check that $g(t)\xi(t)g^{-1}(t)$ is still an element in $se(3)$ and we denote:

$$g(t)\xi(t)g^{-1}(t) = \left[\begin{array}{cc} \widehat{\omega}(t) & v(t) \\ 0 & 0 \end{array} \right]. \tag{2.12}$$

In terms of three dimensional coordinates, we then have the continuous version of (2.7):

$$\dot{\mathbf{X}}(t) = \widehat{\omega}(t)\mathbf{X}(t) + v(t). \tag{2.13}$$

ω and v will be referred to as the (body) **angular** and **linear velocities** respectively.

2.1.3 Calibrated Pinhole Camera Model

We assume that the camera coordinate frame is chosen such that the **optical center** of the camera, denoted by o, is the same as the origin of the frame, and the optical axis always coincides with the third coordinate axis (*i.e.*, the X_3-axis, or the Z-axis if the symbol $[X, Y, Z]^T \in \mathbb{R}^3$ for coordinates is used). Define the **image** of a point $p \in \mathbb{E}^3$ to be the vector $\mathbf{x} \in T_o\mathbb{E}^3$ which corresponds to the intersection of the half ray $\{o + \lambda \cdot u \mid u = p - o, \lambda \in \mathbb{R}^+\}$ with a pre-specified (two dimensional) image surface (in $T_o\mathbb{E}^3$).

Both the **spherical projection** and **perspective projection** fall into this type of imaging model. For the spherical projection, the imaging surface is simply a unit sphere $S^2 = \{u \in \mathbb{R}^3 \mid \|u\|^2 = 1\}$ with o as the center. Supposing that the coordinates of $p \in \mathbb{E}^3$ relative to the camera frame is $\mathbf{X} \in \mathbb{R}^3$, then the spherical projection is defined by the map π_s from $\mathbb{R}^3$ to S^2:

$$\begin{array}{rccl} \pi_s : \mathbb{R}^3 & \to & S^2 \\ \mathbf{X} & \mapsto & \mathbf{x} = \dfrac{\mathbf{X}}{\|\mathbf{X}\|}. \end{array}$$

For the perspective projection, we choose the imaging surface to be the plane of unit distance away from the optical center o along the third coordinate axis. The perspective projection onto this plane is then given by the map π_p from $\mathbb{R}^3$ to $\mathbb{RP}^2$:

$$\begin{aligned} \pi_p : \mathbb{R}^3 &\rightarrow \mathbb{RP}^2 \\ \mathbf{X} &\mapsto \mathbf{x} = \frac{\mathbf{X}}{X_3}. \end{aligned}$$

In what follows, we will use the bold upper case symbol $\mathbf{X} = [X_1, X_2, X_3]^T \in \mathbb{R}^3$ or $\mathbf{X} = [X, Y, Z]^T \in \mathbb{R}^3$ for the 3D coordinates of a point p, and use the bold lower case symbol $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^3$ or $\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3$ for the (homogeneous) coordinates of the image of the point p.

In the most general case, for a point $p \in \mathbb{E}^3$ with homogeneous coordinates $p = [X_1, X_2, X_3, 1]^T \in \mathbb{R}^4$, since the optical center o has the coordinates $[0, 0, 0, 1]^T \in \mathbb{R}^4$, the vector $u = p - o \in T_o\mathbb{E}^3$ is then given by $u = [X_1, X_2, X_3]^T \in \mathbb{R}^3$. We can define the **projection matrix** $P \in \mathbb{R}^{3\times 4}$ to be:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.14}$$

Then the projection matrix P can be interpreted as a map from the space $\mathbb{E}^3$ to $T_o\mathbb{E}^3$:

$$\begin{aligned} P : \mathbb{E}^3 &\rightarrow T_o\mathbb{E}^3 \\ p &\mapsto u = Pp. \end{aligned}$$

According to the definition, the image $\mathbf{x}$ of a point p differs from the vector $u = Pp$ by an arbitrary positive scale, which depends on the pre-specified image surface. In general, the relation between the coordinates $\mathbf{X}$ of $p \in \mathbb{E}^3$ and its image $\mathbf{x}$ is given by:

$$\lambda \mathbf{x} = Pp \tag{2.15}$$

for some unknown $\lambda \in \mathbb{R}^+$. The scalar λ encodes the depth information of p and we call λ the **scale** of the point p with respect to the image $\mathbf{x}$. Simply, for perspective projection $\lambda = X_3$; for spherical projection $\lambda = \|\mathbf{X}\|$. The equation (2.15) characterizes the mathematical model of an ideal **calibrated camera**. Figure 2.2 illustrates the images of a point $p \in \mathbb{E}^3$ with the camera at two different locations.

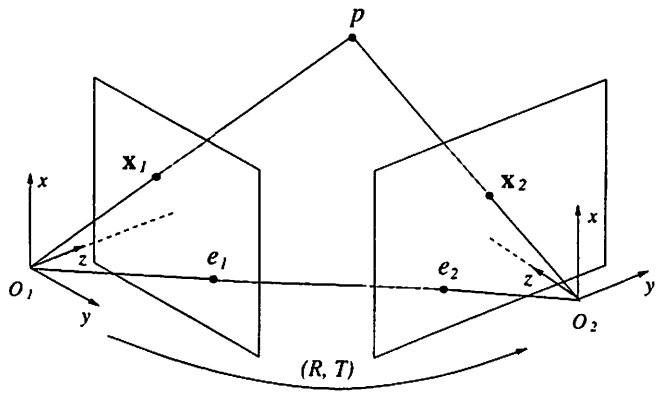


Figure 2.2: Two projections $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ of a 3D point p from two vantage points. The relative Euclidean transformation is given by $(R, T) \in SE(3)$.

2.1.4 Uncalibrated Pinhole Camera Model

Now we are ready to introduce the concept of an **uncalibrated camera**. By an uncalibrated camera, we mean that the image received by the camera is distorted by an unknown linear transformation.[1] This linear transformation is usually assumed to be invertible. Mathematically, this linear transformation can be viewed as an isomorphism ψ of the vector space $T_o\mathbb{E}^3$:

$$\begin{aligned} \psi : T_o\mathbb{E}^3 &\rightarrow T_o\mathbb{E}^3 \\ u &\mapsto Au, \end{aligned}$$

where $A \in \mathbb{R}^{3\times 3}$ is an invertible matrix representing the linear map ψ. We will refer to it as the **calibration matrix**[2] of an uncalibrated camera. The actually received image $\mathbf{x}$ of a point $p \in \mathbb{E}^3$ is then determined by the intersection of the image surface and the ray $\{o + \lambda \cdot u\}$ with $u = \psi(Pp) = APp$. Without loss of generality, we may assume that A has determinant 1, *i.e.*, A is an element in $SL(3)$ (the Lie group consisting of all invertible 3×3 real matrices with determinant 1, *i.e.*, the **special linear group** of $\mathbb{R}^3$). For the (uncalibrated) image $\mathbf{x} \in \mathbb{R}^3$ of p, we then have the following relation:

$$\lambda \mathbf{x} = APp \tag{2.16}$$

[1] Although nonlinear transformations have also been studied in the literature, linear transformations give a very good model of the physical parameters of a camera.

[2] "Calibration matrix", "intrinsic parameter matrix" and "intrinsic parameters" are different names of the same thing in the computer vision literature.

for some scale $\lambda \in \mathbb{R}^+$. The equation (2.16) then characterizes the mathematical model of the uncalibrated camera, as illustrated in Figure 2.3. In practice, the camera calibration A might be *time-varying*. If so, we will denote it as $A(t)$. Nevertheless, in this dissertation, we usually assume the camera calibration is *time-invariant*, unless otherwise stated. From (2.6), the image $\mathbf{x}(t)$ of a point $p \in \mathbb{E}^3$ at time t satisfies the equation:

$$\lambda(t)\mathbf{x}(t) = APg(t)p(t_0). \tag{2.17}$$

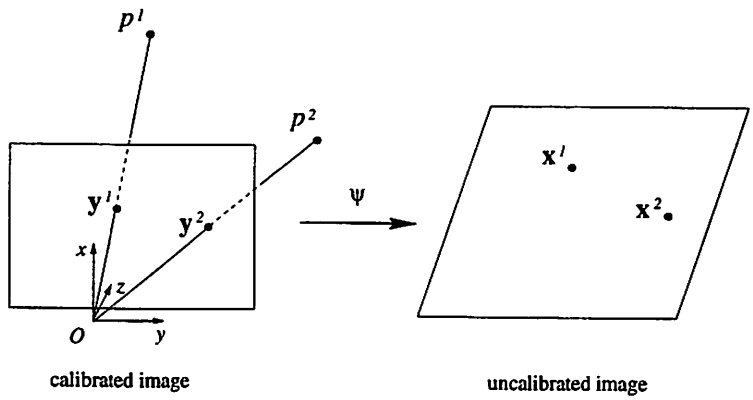


Figure 2.3: The actually received uncalibrated images $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^3$ of two 3D points p^1 and p^2. We here use $\mathbf{y}^1, \mathbf{y}^2 \in \mathbb{R}^3$ to represent the calibrated images (with respect to a normal coordinate system). The linear map ψ transforms the calibrated image to an uncalibrated one.

In the computer vision literature, the calibration matrix A is usually assumed to be of the following form:

$$A = \begin{bmatrix} s_x & s_\theta & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.18}$$

The parameters of the matrix A are called **intrinsic parameters** associated to a camera (as opposed to the **extrinsic parameters**, which usually stand for the displacement of the camera). Note that such an A is not necessarily in $SL(3)$. As we will see in chapter 6 where camera self-calibration is studied, this choice is practically equivalent to ours. Moreover, viewing camera calibration as an (unknown) isomorphism on $T_o\mathbb{E}^3$ makes it quite natural to generalize the vision theory in the Euclidean space to more general Riemannian space. Since Part I of this dissertation focuses on only the Euclidean case, the more advanced

topics about multiview geometry for non-Euclidean spaces can be found in Chapter 8 of Part II.

2.1.5 Image Correspondences and Optical Flows

Image correspondences and optical flows are two fundamental types of measurements one may obtain from image sequences. If m images of n points $p^1, p^2, \ldots, p^n \in \mathbb{E}^3$ at times $t_1, t_2, \ldots, t_m \in \mathbb{R}$ are taken, from (2.17) we have:

$$\lambda^j(t_i)\mathbf{x}^j(t_i) = APg(t_i)p^j(t_0), \quad 1 \leq i \leq m,\ 1 \leq j \leq n. \tag{2.19}$$

Or in three dimensional coordinates, we have:

$$\lambda^j(t_i)\mathbf{x}^j(t_i) = AR(t_i)\mathbf{X}^j(t_0) + AT(t_j), \quad 1 \leq i \leq m,\ 1 \leq j \leq n. \tag{2.20}$$

By **image correspondences** we mean that we have the knowledge that for each j the set of m image points $\{\mathbf{x}^j(t_i)\}_{i=1}^m$ correspond to images of a single 3D point named p^j. When the notion of time is not important, we usually use $\mathbf{x}_i^j$ as a shorthand for $\mathbf{x}^j(t_i)$.

If a sequence of images are taken at times close enough, the displacement of image points on two consecutive images $(\mathbf{x}(t+\Delta t) - \mathbf{x}(t)/\Delta t)$ is approximately the image velocity $\dot{\mathbf{x}}(t)$ which is also called **optical flow** in the literature. From (2.11) and (2.15) we have at any time t:

$$\dot{\lambda}^j\mathbf{x}^j + \lambda^j\dot{\mathbf{x}}^j = APg\xi g^{-1}p^j, \quad 1 \leq j \leq n \tag{2.21}$$

where all the (time-dependent) quantities are evaluated at time t. In three dimensional coordinates, we have:

$$\dot{\lambda}^j\mathbf{x}^j + \lambda^j\dot{\mathbf{x}}^j = A\widehat{\omega}\mathbf{X}^j + Av, \quad 1 \leq j \leq n. \tag{2.22}$$

2.2 Fundamental Problems in Multiview Geometry

According to its mathematical model, we can think of a camera as a moving coordinate frame with a perspective projection associated to it. It is exactly the interplay between the Euclidean motion of the frame and the perspective projection that defines camera as a very special geometric object and a very peculiar sensor. Clearly, the depth information of a point p always gets lost in a single image. However, if two images of p are taken by the

camera at two different vantage points and the camera motion g between the two locations are known, then the 3D coordinates of p relative to the camera can be recovered. This is the so called reconstruction by **stereo**. If the camera motion g and calibration A are not known, the stereo problem becomes more complicated. Nevertheless, it can be shown that the stereo problem is generically solvable if sufficiently many corresponding image points (or optical flows) are available.[3] Generally speaking, Part I of this dissertation is devoted to the geometry of and algorithms for

> *reconstructing 3D scene structure and camera motion from a given set of image correspondences or optical flows. If the camera calibration is not known, the task also includes recovering the unknown camera calibration.*

This is also referred to as the **structure from motion** problem in the computer vision literature and has been extensively studied by numerous researchers for the past decade. However most of the known results are established in a projective geometry framework. One purpose of this paper is to study this problem from a *novel* differential geometric perspective, for the reasons that I have already discussed in the opening introduction. I hope that those reasons will become evident and more convincing when the reader follows through the development of the theory.

In this part, we will partition the structure from motion problem into four inter-related topics or subproblems:

1. **Motion and structure from two views.**
2. **Motion and structure from multiple views.**
3. **Camera self-calibration.**
4. **Euclidean reconstruction and reprojection up to subgroups.**

Results of each topic will be developed under a unified differential geometric framework and a consistent notation. These results together form a coherent theory of multiview geometry in $\mathbb{E}^3$. For the first two topics, to simplify the analysis, only calibrated camera models will be considered (see Chapters 3, 4 and 5). We will especially study the geometry of an uncalibrated camera in the third topic (see Chapter 6). In the first three topics, our

[3] For example, it is known that, for two images, the relative camera motion can be "generically" determined up to ten solutions if five pairs of image correspondences are given.

primary interest is in conditions and algorithms for obtaining a *unique* solution. In the final topic, we will provide a complete characterization of the structure of the set of ambiguous solutions when conditions for a unique solution fail (see Chapter 7).

Chapter 3

Motion Recovery I: Linear Algorithms

"We see because we move; we move because we see."
— J. J. Gibson, *the Perception of the Visual World*

The problem of estimating structure and motion from image sequences has been studied extensively by the computer vision community in the past decade. The various approaches differ in the kinds of assumptions they make about the projection model, the model of the environment, or the type of algorithms they use for estimating the motion and/or structure. Most techniques try to decouple the two problems by estimating the motion first, followed by the structure estimation. Thus the two are usually viewed as separate problems. In spite of the fact that the robustness of existing algorithms has been studied quite extensively, it has been suggested that the fact that the structure and motion estimation are decoupled typically hinders their performance [79]. Some algorithms address the problem of motion and structure recovery simultaneously either in batch [111] or recursive fashion [79].

Approaches to motion estimation alone, can be partitioned into the discrete and continuous methods depending on whether they use as input a set of image correspondences or optical flows. Among the efforts to solve the motion estimation problem, one of the more appealing approaches is the **essential matrix approach**, proposed by Longuet-Higgins, Huang and Faugeras *et al* in 1980's [47, 60]. It shows that the relative 3D displacement of a camera can be recovered from an *intrinsic* geometric constraint between two images

of the same point, the so-called **epipolar constraint** (also called the **Longuet-Higgins constraint, bilinear constraint** or **essential constraint**). Estimating 3D motion can therefore be decoupled from estimation of the structure of the 3D scene. This endows the resulting motion estimation algorithms with some advantageous features: they do not need to assume any *a priori* knowledge about the scene; and are computationally simpler (compared to most non-intrinsic motion estimation algorithms), using mostly linear algebra techniques. Tsai and Huang [119] have proved that, given an essential matrix associated with the epipolar constraint, there are only two possible 3D displacements. The study of the essential matrix then led to a three-step SVD-based algorithm for recovering the 3D displacement from noisy image correspondences, proposed in 1986 by Toscani and Faugeras [112] and later summarized in Maybank [76].

However, the essential matrix approach based on the epipolar constraint recovers only *discrete* 3D displacement. The velocity information can only be obtained approximately from the logarithm map (the inverse of the exponential map), as Soatto *et al* did in [99]. In principle, displacement estimation algorithms obtained by using epipolar constraint work well when the displacement (especially the translation, or the so called base-line) between the two images is relatively large. However, in real-time applications, even if the velocity of the moving camera is not small, the relative displacement between two consecutive images might become small owing to a high frame rate. In turn, the algorithms become singular due to the small translation and the estimation results become less reliable. Further more, in applications such as robotic control, an on-board camera, as a feedback sensor, is required not only to provide relative orientation of the robot but also its relative speed (for control purposes).

A continuous version of the 3D motion estimation problem is to recover the 3D velocity of the camera from optical flows. This problem has also been explored by many researchers: an algorithm was proposed in 1984 by Zhuang *et al* [141] with a simplified version given in 1986 [142]; and a first order algorithm was given by Waxman *et al* [125] in 1987. Most algorithms start from the basic bilinear constraint relating optical flow to the linear and angular velocities and solve for rotation and translation separately using either numerical optimization techniques (Bruss and Horn [10]) or linear subspace methods (Heeger and Jepson [40, 50]). Kanatani [52] proposed a linear algorithm reformulating Zhuang's approach in terms of essential parameters and twisted flow. However, in these algorithms, the similarities between the discrete case and the continuous case are not fully

revealed and exploited.

In this chapter, we develop, in parallel to the discrete essential matrix approach, a **continuous essential matrix approach** for recovering 3D velocity from optical flows. Based on the continuous version of the epipolar constraint, so called **continuous essential matrices** are defined. We then give a complete characterization of the space of these matrices and prove that there exists exactly one 3D velocity corresponding to a given continuous essential matrix. As a continuous counterpart of the three-step SVD-based 3D displacement estimation algorithm, a four-step eigenvector-decomposition-based 3D velocity estimation algorithm is proposed.

One of the big advantages of the continuous approach is easy to exploit the **nonholonomic constraints** of a mobile base where the camera is mounted. In this chapter, we show by example that nonholonomic constraints may reduce the number of dimensions of the motion estimation problem, hence reduce the number of minimum image measurements needed for a unique solution. The proposed motion estimation algorithm can thus be dramatically simplified. The continuous approach developed here can also be generalized to the case of an uncalibrated camera (see [9, 122]), this will be further discussed in Chapter 6. Finally, simulation results will be presented to evaluate the performance of our algorithm in terms of bias and sensitivity of the estimates with respect to the noise in optical flow measurements.

One must note that only linear algorithms will be studied and compared in this chapter. It is well-known that linear algorithms are not optimal and give severely biased estimates when the noise level is high. In order to obtain optimal or less biased estimates, nonlinear schemes have to be used to solve for maximum likelihood estimates. In Chapter 4, we will propose an intrinsic geometric optimization algorithm based on Riemannian optimization techniques on manifolds. However, since nonlinear algorithms are only locally convergent, the linear algorithms studied in this paper can be used to initialize the search process of nonlinear algorithms. Further more, due to their geometric simplicity, clearly understanding the linear algorithms certainly helps in developing and understanding more sophisticated motion estimation schemes. For example, it will be shown in Chapter 4 that under the same conditions when the linear algorithms have a unique solution the nonlinear algorithms have quadratic rate of convergence.

3.1 Continuous Essential Matrix Approach

3.1.1 Review of the Discrete Essential Matrix Approach

Before developing the analysis of the continuous epipolar constraint which is the main focus of this paper, we first provide a brief review of the epipolar geometry in the discrete case, also known as the **essential matrix approach**, originally developed by Huang and Faugeras [47]. Let the 3D displacement of the frame F_t relative to the frame F_{t_0} be given by the rigid body motion $g = (R, T) \in SE(3)$, and let $\mathbf{x}_1, \mathbf{x}_2$ be the images of the same point p taken by the camera at frames F_{t_0} and F_t, respectively.[1] From (2.7), these two images are related through equation:

$$\lambda_2 \mathbf{x}_2 = R\lambda_1 \mathbf{x}_1 + T \tag{3.1}$$

for some positive depth scales $\lambda_1, \lambda_2 > 0$. Multiply $\widehat{T}$ to both sides of this equation and we obtain $\lambda_2 \widehat{T}\mathbf{x}_2 = \widehat{T}R\lambda_1\mathbf{x}_1$. Note that $\widehat{T}\mathbf{x}_2 = T \times \mathbf{x}_2$ hence $\mathbf{x}_2^T\widehat{T}\mathbf{x}_2 = 0$. This implies $\mathbf{x}_2^T\widehat{T}R\lambda_1\mathbf{x}_1 = 0$. Since $\lambda_1 > 0$, the two image points $\mathbf{x}_1, \mathbf{x}_2$ satisfy the so called **epipolar constraint**:

$$\mathbf{x}_2^T\widehat{T}R\mathbf{x}_1 = 0. \tag{3.2}$$

The geometric explanation for this constraint is simply that the two optical centers o_1, o_2 and the point p are coplanar and the two images $\mathbf{x}_1$ $\mathbf{x}_2$ are on the plane spanned by these three points. See the Figure 2.2 in Chapter 2.

In the equation (3.2), we see that the matrix $E = \widehat{T}R$ with $R \in SO(3)$ and $\widehat{T} \in so(3)$ contains the unknown motion parameters. A matrix of this form is called an **essential matrix**; and the set of all essential matrices is called the **essential space**, denoted by $\mathcal{E}$:

$$\mathcal{E} \equiv \left\{ \widehat{T}R \,\middle|\, R \in SO(3), T \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{3\times 3}. \tag{3.3}$$

Huang and Faugeras [47] established that a non-zero matrix E is an essential matrix *if and only if* the singular value decomposition (SVD) of E: $E = U\Sigma V^T$ satisfies:

$$\Sigma = diag\{\sigma, \sigma, 0\} \tag{3.4}$$

[1] To simplify the notation, we here drop the time dependence.

for some $\sigma \in \mathbb{R}_+$. In order to answer the question: given an essential matrix $E \in \mathcal{E}$, how many pairs (R,T) exist such that $\widehat{T}R = E$, we first give the following lemma from linear algebra:

Lemma 3.1. *Consider an arbitrary non-zero skew symmetric matrix $\widehat{T} \in so(3)$ with $T \in \mathbb{R}^3$. If, for a rotation matrix $R \in SO(3)$, $\widehat{T}R$ is also a skew symmetric matrix, then $R = I$ or $e^{\widehat{u}\pi}$ where $u = T/\|T\|$. Further, $\widehat{T}e^{\widehat{u}\pi} = -\widehat{T}$.*

Proof: Without loss of generality, we assume T is of unit length. Since $\widehat{T}R$ is also a skew symmetric matrix, $(\widehat{T}R)^T = -\widehat{T}R$. This equation gives:

$$R\widehat{T}R = \widehat{T}. \tag{3.5}$$

Since R is a rotation matrix, there exists $\omega \in \mathbb{R}^3, \|\omega\| = 1$ and $\theta \in \mathbb{R}$ such that $R = e^{\widehat{\omega}\theta}$. Then, (3.5) is rewritten as: $e^{\widehat{\omega}\theta}\widehat{T}e^{\widehat{\omega}\theta} = \widehat{T}$. Applying this equation to ω, we get: $e^{\widehat{\omega}\theta}\widehat{T}e^{\widehat{\omega}\theta}\omega = \widehat{T}\omega$. Since $e^{\widehat{\omega}\theta}\omega = \omega$, we obtain: $e^{\widehat{\omega}\theta}\widehat{T}\omega = \widehat{T}\omega$. Since ω is the only eigenvector associated to the eigenvalue 1 of the matrix $e^{\widehat{\omega}\theta}$ and $\widehat{T}\omega$ is orthogonal to ω, $\widehat{T}\omega$ has to be zero. Thus, ω is equal to T or $-T$. R then has the form $e^{\widehat{T}\theta}$, which commutes with $\widehat{T}$. Thus from (3.5), we get:

$$e^{2\widehat{T}\theta}\widehat{T} = \widehat{T}. \tag{3.6}$$

According to *Rodrigues' formula* [84], we have:

$$e^{2\widehat{T}\theta} = I + \widehat{T}\sin(2\theta) + \widehat{T}^2(1 - \cos(2\theta)) \tag{3.7}$$

(3.6) yields:

$$\widehat{T}^2\sin(2\theta) + \widehat{T}^3(1 - \cos(2\theta)) = 0. \tag{3.8}$$

Since $\widehat{T}^2$ and $\widehat{T}^3$ are linearly independent (Lemma 2.3 in [84]), we have $\sin(2\theta) = 1 - \cos(2\theta) = 0$. That is, θ is equal to $2k\pi$ or $2k\pi + \pi$, $k \in \mathbb{Z}$. Therefore, R is equal to I or $e^{\widehat{T}\pi}$. It is direct from the geometric meaning of the rotation $e^{\widehat{T}\pi}$ that $e^{\widehat{T}\pi}\widehat{T} = -\widehat{T}$. ∎

Following this lemma, suppose $(R_1,T_1) \in SE(3)$ and $(R_2,p_2) \in SE(3)$ are both solutions for the equation $\widehat{T}R = E$. Then we have $\widehat{T}_1R_1 = \widehat{T}_2R_2$. It yields $\widehat{T}_1 = \widehat{T}_2R_2R_1^T$. Since $\widehat{T}_1, \widehat{T}_2$ are both skew symmetric matrices and $R_2R_1^T$ is a rotation matrix, we then have either $(R_2,T_2) = (R_1,T_1)$ or $(R_2,T_2) = (e^{\widehat{u}_1\pi}R_1, -T_1)$ with $u_1 = T_1/\|T_1\|$. Therefore,

given an essential matrix E there are exactly *two* pairs (R,T) such that $\widehat{T}R = E$. Further, if E has the SVD: $E = U\Sigma V^T$ with $U, V \in SO(3)$,[2] the following formulae give the two solutions:

$$\begin{cases} (\widehat{T}_1, R_1) &= (UR_Z(+\frac{\pi}{2})\Sigma U^T, UR_Z^T(+\frac{\pi}{2})V^T), \\ (\widehat{T}_2, R_2) &= (UR_Z(-\frac{\pi}{2})\Sigma U^T, UR_Z^T(-\frac{\pi}{2})V^T) \end{cases} \tag{3.9}$$

where $R_Z(\theta)$ is defined to be the rotation matrix around the Z-axis by an angle θ, *i.e.*, $R_Z(\theta) = e^{\widehat{e_3}\theta}$ with $e_3 = [0,0,1]^T \in \mathbb{R}^3$.

Since from the epipolar constraint (3.2) one can only recover the essential matrix up to an arbitrary scale (in particular, both E and $-E$ satisfy the same equation), so in general four solutions (R,T) will be obtained from image correspondences. Usually, the **positive depth constraint** can be imposed to discard three of the ambiguous solutions. We here omit these well known details and simply summarize the discrete essential matrix approach for motion estimation as the following algorithm (which is essentially the same as that given in Maybank [76]) and we repeat it here for comparison with the algorithm that we will develop for the continuous case:

Algorithm 3.2 (Three Step SVD Based 3D Motion Estimation).

1. **Estimate the essential matrix:**

   For a given set of image correspondences: $(\mathbf{x}_1^j, \mathbf{x}_2^j)$, $j = 1, \ldots, n$ $(n \geq 8)$, *find the matrix* E *which minimizes the error function:*

   $$V(E) = \sum_{i=1}^{n} (\mathbf{x}_2^{i\,T} E \mathbf{x}_1^i)^2 \tag{3.10}$$

   subject to the condition $\|E\| = 1$;

2. **Singular value decomposition:**

   Recover matrix E *from* $\mathbf{e}$ *and find the singular value decomposition of the matrix* E*:*

   $$E = U diag\{\sigma_1, \sigma_2, \sigma_3\} V^T \tag{3.11}$$

   where $\sigma_1 \geq \sigma_2 \geq \sigma_3$;

3. **Recover displacement from the essential matrix:**

   Define the diagonal matrix Σ *to be:*

   $$\Sigma = diag\{1, 1, 0\}. \tag{3.12}$$

[2] An essential matrix always has a SVD such that $U, V \in SO(3)$.

Then the 3D displacement (p, R) is given by:

$$R = UR_Z^T(\pm\frac{\pi}{2})V^T, \quad \widehat{T} = UR_Z(\pm\frac{\pi}{2})\Sigma U^T. \tag{3.13}$$

The epipolar geometric relationship between projections of the points and their displacements transfers to the continuous case. So, intuitively speaking, the continuous case is an infinitesimal version of the discrete case. However, the continuous case is by no means simply a "first order approximation" of the discrete case. When differentiation takes place, while structure of the geometry of the discrete case is inherited by the continuous case, some degeneracy may occur. Such degeneracy will become clear when we study the continuous version of the epipolar constraint. It is also known that it is exactly due to the degeneracy that camera calibration cannot be fully recovered from continuous epipolar constraint as opposed to the discrete case (see Chapter 6). Generally speaking, the similarity between these two cases is that methods and geometric intuition used in the discrete case can be extended to the continuous case, even though geometric characterization of the objects is different. One of the main goals of this paper is to clarify the geometric *similarity* and *difference* between the discrete and continuous cases. Although the theory will be developed in a calibrated camera framework, the clear geometric nature of this approach has helped us to understand the uncalibrated situation as well, as we will see in Chapter 6.

3.1.2 Continuous Epipolar Constraint

We now develop a **continuous essential matrix approach** for estimating 3D velocity from optical flow in a parallel way to the discrete essential matrix approach for estimating 3D displacement from image correspondences.

The starting point of this approach is a continuous version of the epipolar constraint and associated concept of continuous essential matrix. This constraint is bilinear in nature and has been used extensively in the motion estimation from optical flow measurements [40, 122]. Here we give a characterization of such matrices and show that there exists exactly one 3D velocity corresponding to a non-zero continuous essential matrix; as a continuous version of the three-step SVD-based 3D displacement estimation algorithm, we propose a four-step eigenvector-decomposition-based 3D velocity estimation algorithm; finally, we discuss the reasons why the zero-translation case makes all essential constraint based motion estimation algorithms fail and suggest possible ways to overcome this difficulty.

Assume that camera motion is described by a smooth curve $g(t) = (R(t), T(t)) \in SE(3)$ with body velocities $(\omega(t), v(t)) \in se(3)$. According to (2.13), for a point $p \in \mathbb{R}^3$, its coordinates $\mathbf{X}(t) = \lambda(t)\mathbf{x}(t)$ satisfy:

$$\dot{\mathbf{X}}(t) = \widehat{\omega}(t)\mathbf{X}(t) + v(t). \tag{3.14}$$

From now on, for convenience we will drop the time-dependency from the notation. The image of the point p taken by the camera is $\mathbf{x}$ which satisfies $\lambda\mathbf{x} = \mathbf{X}$. Denote the velocity of the image point $\mathbf{x}$ by $\mathbf{u} = \dot{\mathbf{x}} \in \mathbb{R}^3$. $\mathbf{u}$ is also called **optical flow**.

Theorem 3.3 (Continuous Epipolar Constraint). *Consider a camera moving with body velocities (ω, v). Then the optical flow $\mathbf{u} = \dot{\mathbf{x}}$ of an image point $\mathbf{x}$ satisfies:*

$$\mathbf{u}^T\widehat{v}\mathbf{x} + \mathbf{x}^T\widehat{\omega}\widehat{v}\mathbf{x} \equiv 0 \tag{3.15}$$

or in an equivalent form:

$$[\mathbf{u}^T, \mathbf{x}^T]\begin{bmatrix} \widehat{v} \\ s \end{bmatrix}\mathbf{x} = 0 \tag{3.16}$$

where s is a symmetric matrix defined to be $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \in \mathbb{R}^{3\times 3}$.

Proof: Take the inner product of the vectors in (3.14) with $(v \times \mathbf{x})$:

$$\dot{\mathbf{X}}^T(v \times \mathbf{x}) = (\widehat{\omega}\mathbf{X} + v)^T(v \times \mathbf{x}) = \mathbf{X}^T\widehat{\omega}^T\widehat{v}\mathbf{x}. \tag{3.17}$$

Since $\dot{\mathbf{X}} = \dot{\lambda}\mathbf{x} + \lambda\dot{\mathbf{x}}$ and $\mathbf{x}^T(v \times \mathbf{x}) = 0$, from (3.17) we then have:

$$\lambda\dot{\mathbf{x}}^T\widehat{v}\mathbf{x} - \lambda\mathbf{x}^T\widehat{\omega}^T\widehat{v}\mathbf{x} = 0. \tag{3.18}$$

When $\lambda \neq 0$, we obtain a continuous version of the epipolar constraint:

$$\mathbf{u}^T\widehat{v}\mathbf{x} + \mathbf{x}^T\widehat{\omega}\widehat{v}\mathbf{x} \equiv 0 \tag{3.19}$$

Due to the following fact 3.4, for any skew symmetric matrix $A \in \mathbb{R}^{3\times 3}$, $\mathbf{x}^T A\mathbf{x} = 0$. Since $\frac{1}{2}(\widehat{\omega}\widehat{v} - \widehat{v}\widehat{\omega})$ is a skew symmetric matrix, $\mathbf{x}^T\frac{1}{2}(\widehat{\omega}\widehat{v} - \widehat{v}\widehat{\omega})\mathbf{x} = \mathbf{x}^T s\mathbf{x} - \mathbf{x}^T\widehat{\omega}\widehat{v}\mathbf{x} = 0$. Thus, $\mathbf{x}^T s\mathbf{x} = \mathbf{x}^T\widehat{\omega}\widehat{v}\mathbf{x}$. We then have:

$$\mathbf{u}^T\widehat{v}\mathbf{x} + \mathbf{x}^T s\mathbf{x} \equiv 0. \tag{3.20}$$

■

The proof indicates that there is some redundancy in the expression of the continuous epipolar constraint (3.15). The following fact from linear algebra shows where this redundancy comes from.

Fact 3.4. *Consider matrices $M_1, M_2 \in \mathbb{R}^{3\times 3}$. $\mathbf{x}^T M_1 \mathbf{x} = \mathbf{x}^T M_2 \mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^3$ if and only if $M_1 - M_2$ is a skew symmetric matrix, i.e., $M_1 - M_2 \in so(3)$.*

Let us define an equivalence relation on the space $\mathbb{R}^{3\times 3}$, the space of 3×3 matrices over $\mathbb{R}$: for $x, y \in \mathbb{R}^{3\times 3}$, $x \sim y$ if and only if $x - y \in so(3)$. Denote by $\overline{x} = \{y \in \mathbb{R}^{3\times 3} \mid y \sim x\}$ the equivalence class of x, and denote by $\overline{X}$ the set $\bigcup_{x\in X} \overline{x}$. The quotient space $\mathbb{R}^{3\times 3}/\sim$ can be naturally identified with the space of all 3×3 symmetric matrices. Especially, we have $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \in \overline{\widehat{\omega}\widehat{v}}$, which is the reason why we choose it in the equivalent form (3.16). Using this notation, Theorem 3.3 can then be re-expressed in the following way:

Corollary 3.5. *Consider a camera undergoing a smooth rigid body motion with linear velocity v and angular velocity ω. Then the optical flow $\mathbf{u}$ of a image point $\mathbf{x}$ satisfies:*

$$[\mathbf{u}^T, \mathbf{x}^T] \begin{bmatrix} \widehat{v} \\ \overline{\widehat{\omega}\widehat{v}} \end{bmatrix} \mathbf{x} \equiv 0. \tag{3.21}$$

Because of this redundancy, each equivalence class $\overline{\widehat{\omega}\widehat{v}}$ can only be recovered up to its symmetric component $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \in \overline{\widehat{\omega}\widehat{v}}$. This redundancy is the exact reason why different forms of the continuous epipolar constraint exist in the literature [141, 89, 122, 76, 9], and, accordingly, various approaches have been proposed to recover ω and v (see [109]). It is also the reason why the continuous case cannot be simply viewed as a first order approximation of the discrete case – a first order approximation of the essential matrix $\widehat{T}R$ is $\widehat{v}\widehat{\omega}$, but this is certainly not what one can directly estimate from the continuous epipolar constraint. Instead, one has to deal with its symmetric part $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. This, in fact, makes the study of the continuous case harder than the discrete case (in seek for linear algorithms). Notice that the symmetric matrix s is the same as the matrix K defined in Kanatani [53]. Although the characterization of such matrices has been studied in [53], our constructive proofs given below will lead to a natural algorithm for recovering (ω, v) from s.

3.1.3 Characterization of the Continuous Essential Matrix

We define the space of 6×3 matrices given by:

$$\mathcal{E}' = \left\{ \left[\begin{array}{c} \widehat{v} \\ \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \end{array} \right] \middle| \omega, v \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{6\times 3}. \tag{3.22}$$

to be the **continuous essential space**. A matrix in this space is called a **continuous essential matrix**. Note that the continuous epipolar constraint (3.16) is homogeneous on the linear velocity v. Thus v may be recovered only up to a constant scale. Consequently, in motion recovery, we will concern ourselves with matrices belonging to **normalized continuous essential space**:

$$\mathcal{E}'_1 = \left\{ \left[\begin{array}{c} \widehat{v} \\ \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \end{array} \right] \middle| \omega \in \mathbb{R}^3, v \in \mathbb{S}^2 \right\} \subset \mathbb{R}^{6\times 3}. \tag{3.23}$$

The skew-symmetric part of a continuous essential matrix simply corresponds to the velocity v. The characterization of the (normalized) essential matrix only focuses on the characterization of the symmetric part of the matrix: $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. We call the space of all the matrices of such form the **special symmetric space**:

$$\mathcal{S} = \left\{ \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) \middle| \omega \in \mathbb{R}^3, v \in \mathbb{S}^2 \right\} \subset \mathbb{R}^{3\times 3}. \tag{3.24}$$

A matrix in this space is called a **special symmetric matrix**. The motion estimation problem is now reduced to the one of *recovering the velocity* (ω, v) *with* $\omega \in \mathbb{R}^3$ *and* $v \in \mathbb{S}^2$ *from a given special symmetric matrix* s.

The characterization of special symmetric matrices depends on a characterization of matrices in the form: $\widehat{\omega}\widehat{v} \in \mathbb{R}^{3\times 3}$, which is given in the following lemma. This lemma will also be used in the next section for showing the uniqueness of the velocity recovery from special symmetric matrices. Like the (discrete) essential matrices, matrices with the form $\widehat{\omega}\widehat{v}$ are characterized by their singular value decomposition (SVD): $\widehat{\omega}\widehat{v} = U\Sigma V^T$; moreover, the orthogonal matrices U and V are related. Define the matrix $R_Y(\theta)$ to be the rotation around the Y-axis by an angle $\theta \in \mathbb{R}$, *i.e.*, $R_Y(\theta) = e^{\widehat{e}_2\theta}$ with $e_2 = [0, 1, 0]^T \in \mathbb{R}^3$.

Lemma 3.6. *A matrix* $Q \in \mathbb{R}^{3\times 3}$ *has the form* $Q = \widehat{\omega}\widehat{v}$ *with* $\omega \in \mathbb{R}^3$, $v \in \mathbb{S}^2$ *if and only if* Q *has the form:*

$$Q = -VR_Y(\theta)diag\{\lambda, \lambda\cos(\theta), 0\}V^T \tag{3.25}$$

for some rotation matrix $V \in SO(3)$. *Further,* $\lambda = \|\omega\|$ *and* $\cos(\theta) = \omega^T v/\lambda$.

Proof: We first prove the necessity. The proof follows from the geometric meaning of $\widehat{\omega}\widehat{v}$: for any vector $q \in \mathbb{R}^3$,

$$\widehat{\omega}\widehat{v}q = \omega \times (v \times q). \tag{3.26}$$

Let $b \in \mathbb{S}^2$ be the unit vector perpendicular to both ω and v: $b = \frac{v \times \omega}{\|v \times \omega\|}$ (if $v \times \omega = 0$, b is not uniquely defined. In this case, pick any b orthogonal to v and ω,then the rest of the proof still holds). Then $\omega = \lambda \exp(\widehat{b}\theta)v$ (according this definition, θ is the angle between ω and v, and $0 \leq \theta \leq \pi$). It is direct to check that if the matrix V is defined to be:

$$V = (e^{\widehat{b}\frac{\pi}{2}}v, b, v), \tag{3.27}$$

then Q has the given form (3.25).

We now prove the sufficiency. Given a matrix Q which can be decomposed into the form (3.25), define the orthogonal matrix $U = -VR_Y(\theta) \in O(3)$.[3] Let the two skew symmetric matrices $\widehat{\omega}$ and $\widehat{v}$ given by the formulae:

$$\widehat{\omega} = UR_Z(\pm\frac{\pi}{2})\Sigma_\lambda U^T, \quad \widehat{v} = VR_Z(\pm\frac{\pi}{2})\Sigma_1 V^T \tag{3.28}$$

where $\Sigma_\lambda = diag\{\lambda, \lambda, 0\}$ and $\Sigma_1 = diag\{1, 1, 0\}$. Then:

$$\begin{aligned}
\widehat{\omega}\widehat{v} &= UR_Z(\pm\frac{\pi}{2})\Sigma_\lambda U^T V R_Z(\pm\frac{\pi}{2})\Sigma_1 V^T \\
&= UR_Z(\pm\frac{\pi}{2})\Sigma_\lambda(-R_Y^T(\theta))R_Z(\pm\frac{\pi}{2})\Sigma_1 V^T \\
&= Udiag\{\lambda, \lambda\cos(\theta), 0\}V^T \\
&= Q.
\end{aligned} \tag{3.29}$$

Since ω and v have to be, respectively, the left and the right zero eigenvectors of Q, the reconstruction given in (3.28) is unique. ∎

The following theorem gives a characterization of the special symmetric matrix.

Theorem 3.7 (Characterization of the Special Symmetric Matrix). *A real symmetric matrix* $s \in \mathbb{R}^{3\times3}$ *is a special symmetric matrix if and only if* s *can be diagonalized as* $s = V\Sigma V^T$ *with* $V \in SO(3)$ *and:*

$$\Sigma = diag\{\sigma_1, \sigma_2, \sigma_3\} \tag{3.30}$$

with $\sigma_1 \geq 0, \sigma_3 \leq 0$ *and* $\sigma_2 = \sigma_1 + \sigma_3$.

[3] $O(3)$ represents the space of all orthogonal matrices (of determinant ±1.)

Proof: We first prove the necessity. Suppose s is a special symmetric matrix, there exist $\omega \in \mathbb{R}^3, v \in \mathbb{S}^2$ such that $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. Since s is a symmetric matrix, it is diagonalizable, all its eigenvalues are real and all the eigenvectors are orthogonal to each other. It then suffices to check that its eigenvalues satisfy the given conditions.

Let the unit vector b and the rotation matrix V be the same as in the proof of Lemma 3.6, so are θ and γ. Then according to the lemma, we have:

$$\widehat{\omega}\widehat{v} = -VR_Y(\theta)diag\{\lambda, \lambda\cos(\theta), 0\}V^T. \tag{3.31}$$

Since $(\widehat{\omega}\widehat{v})^T = \widehat{v}\widehat{\omega}$, it yields:

$$s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) = \frac{1}{2}V\left(-R_Y(\theta)diag\{\lambda, \lambda\cos(\theta), 0\} - diag\{\lambda, \lambda\cos(\theta), 0\}R_Y^T(\theta)\right)V^T. \tag{3.32}$$

Define the matrix $D(\lambda, \theta) \in \mathbb{R}^{3\times 3}$ to be:

$$\begin{aligned} D(\lambda,\theta) &= -R_Y(\theta)diag\{\lambda, \lambda\cos(\theta), 0\} - diag\{\lambda, \lambda\cos(\theta), 0\}R_Y^T(\theta) \\ &= \lambda\begin{bmatrix} -2\cos(\theta) & 0 & \sin(\theta) \\ 0 & -2\cos(\theta) & 0 \\ \sin(\theta) & 0 & 0 \end{bmatrix}. \end{aligned} \tag{3.33}$$

Directly calculating its eigenvalues and eigenvectors, we obtain that:

$$D(\lambda,\theta) = R_Y\left(\frac{\theta-\pi}{2}\right)diag\left\{\lambda(1-\cos(\theta)), -2\lambda\cos(\theta), \lambda(-1-\cos(\theta))\right\}R_Y^T\left(\frac{\theta-\pi}{2}\right) \tag{3.34}$$

Thus $s = \frac{1}{2}VD(\lambda,\theta)V^T$ has eigenvalues:

$$\left\{\frac{1}{2}\lambda(1-\cos(\theta)), \quad -\lambda\cos(\theta), \quad \frac{1}{2}\lambda(-1-\cos(\theta))\right\}, \tag{3.35}$$

which satisfy the given conditions.

We now prove the sufficiency. Given $s = V_1 diag\{\sigma_1, \sigma_2, \sigma_3\}V_1^T$ with $\sigma_1 \geq 0, \sigma_3 \leq 0$ and $\sigma_2 = \sigma_1 + \sigma_3$ and $V_1^T \in SO(3)$, these three eigenvalues uniquely determine $\lambda, \theta \in \mathbb{R}$ such that the σ_i's have the form given in (3.35):

$$\begin{cases} \lambda &= \sigma_1 - \sigma_3, & \lambda \geq 0 \\ \theta &= \arccos(-\sigma_2/\lambda), & \theta \in [0, \pi] \end{cases}$$

Define a matrix $V \in SO(3)$ to be $V = V_1 R_Y^T\left(\frac{\theta}{2} - \frac{\pi}{2}\right)$. Then $s = \frac{1}{2}VD(\lambda,\theta)V^T$. According to Lemma 3.6, there exist vectors $v \in \mathbb{S}^2$ and $\omega \in \mathbb{R}^3$ such that:

$$\widehat{\omega}\widehat{v} = -VR_Y(\theta)diag\{\lambda, \lambda\cos(\theta), 0\}V^T. \tag{3.36}$$

Therefore, $\frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega}) = \frac{1}{2}VD(\lambda, \theta)V^T = s$. ∎

Figure 3.1 gives a geometric interpretation of the three eigenvectors of the special symmetric matrix s for the case when both ω, v are of unit length. Theorem 3.7 was given as

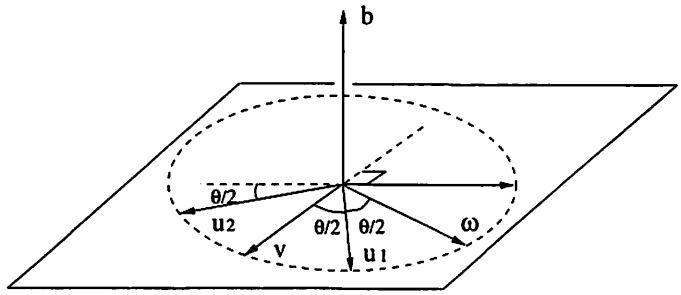


Figure 3.1: Vectors u_1, u_2, b are the three eigenvectors of a special symmetric matrix $\frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. In particular, b is the normal vector to the plane spanned by ω and v, and u_1, u_2 are both in this plane. u_1 is the average of ω and v. u_2 is orthogonal to both b and u_1.

an exercise problem in Kanatani [53] but it has never been really exploited in the literature for designing algorithms. For that purpose, the constructive proof given above is more important since it gives an explicit decomposition of the special symmetric matrix s, which will be studied in more detail next.

According to the proof of the sufficiency of Theorem 3.7, if we already know the eigenvector decomposition of a special symmetric matrix s, we certainly can find at least one solution (ω, v) such that $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. This section discusses the uniqueness of such reconstruction, *i.e.*, how many solutions exist for $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$.

Theorem 3.8 (Velocity Recovery from the Special Symmetric Matrix). *There exist exactly four 3D velocities (ω, v) with $\omega \in \mathbb{R}^3$ and $v \in \mathbb{S}^2$ corresponding to a non-zero special symmetric matrix $s \in \mathcal{S}$.*

Proof: Suppose (ω_1, v_1) and (ω_2, v_2) are both solutions for $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. Then we have:

$$\widehat{v}_1\widehat{\omega}_1 + \widehat{\omega}_1\widehat{v}_1 = \widehat{v}_2\widehat{\omega}_2 + \widehat{\omega}_2\widehat{v}_2. \tag{3.37}$$

From Lemma 3.6, we may write:

$$\begin{cases} \widehat{\omega}_1\widehat{v}_1 &= -V_1 R_Y(\theta_1) diag\{\lambda_1, \lambda_1\cos(\theta_1), 0\} V_1^T \\ \widehat{\omega}_2\widehat{v}_2 &= -V_2 R_Y(\theta_2) diag\{\lambda_2, \lambda_2\cos(\theta_2), 0\} V_2^T. \end{cases} \tag{3.38}$$

Let $W = V_1^T V_2 \in SO(3)$, then from (3.37):

$$D(\lambda_1, \theta_1) = W D(\lambda_2, \theta_2) W^T. \tag{3.39}$$

Since both sides of (3.39) have the same eigenvalues, according to (3.34), we have:

$$\lambda_1 = \lambda_2, \quad \theta_2 = \theta_1. \tag{3.40}$$

We then can denote both θ_1 and θ_2 by θ. It is direct to check that the only possible rotation matrix W which satisfies (3.39) is given by $I_{3\times 3}$ or:

$$\begin{bmatrix} -\cos(\theta) & 0 & \sin(\theta) \\ 0 & -1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad or \quad \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & -1 & 0 \\ -\sin(\theta) & 0 & -\cos(\theta) \end{bmatrix}. \tag{3.41}$$

From the geometric meaning of V_1 and V_2, all the cases give either $\widehat{\omega}_1 \widehat{v}_1 = \widehat{\omega}_2 \widehat{v}_2$ or $\widehat{\omega}_1 \widehat{v}_1 = \widehat{v}_2 \widehat{\omega}_2$. Thus, according to the proof of Lemma 3.6, if (ω, v) is one solution and $\widehat{\omega}\widehat{v} = U diag\{\lambda, \lambda\cos(\theta), 0\} V^T$, then all the solutions are given by:

$$\begin{cases} \widehat{\omega} = U R_Z(\pm\frac{\pi}{2}) \Sigma_\lambda U^T, & \widehat{v} = V R_Z(\pm\frac{\pi}{2}) \Sigma_1 V^T; \\ \widehat{\omega} = V R_Z(\pm\frac{\pi}{2}) \Sigma_\lambda V^T, & \widehat{v} = U R_Z(\pm\frac{\pi}{2}) \Sigma_1 U^T \end{cases} \tag{3.42}$$

where $\Sigma_\lambda = diag\{\lambda, \lambda, 0\}$ and $\Sigma_1 = diag\{1, 1, 0\}$. ∎

Given a non-zero continuous essential matrix $E \in \mathcal{E}'$, according to (3.42) its special symmetric part gives four possible solutions for the 3D velocity (ω, v). However, in general only one of them has the same linear velocity v as the skew symmetric part of E does. We thus have:

Theorem 3.9 (Velocity Recovery from Continuous Essential Matrix). *There is only one solution of 3D velocity (ω, v) corresponding to a non-zero continuous essential matrix $E \in \mathcal{E}'$.*

In the discrete case, there are two 3D displacements corresponding to an essential matrix. However, the velocity corresponding to a continuous essential matrix is unique. This is because, in the continuous case, the twisted-pair ambiguity (see Maybank [76]), which is caused by a 180° rotation of the camera around the translation direction, is avoided.

3.1.4 Algorithm

Based on the preceding study of the continuous essential matrix, we propose an new algorithm which recovers the 3D velocity of the camera from a set of (possibly noisy) optical flows.

Let $E = \begin{bmatrix} \widehat{v} \\ s \end{bmatrix} \in \mathcal{E}_1'$ with $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$ be the essential matrix associated with the continuous epipolar constraint (3.16). Since the sub-matrix $\widehat{v}$ is skew symmetric and s is symmetric, they have the following form:

$$v = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad s = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_4 & s_5 \\ s_3 & s_5 & s_6 \end{bmatrix}. \tag{3.43}$$

Define the (continuous) **essential vector** $\mathbf{e} \in \mathbb{R}^9$ to be:

$$\mathbf{e} = [v_1, v_2, v_3, s_1, s_2, s_3, s_4, s_5, s_6]^T. \tag{3.44}$$

Define a vector $\mathbf{a} \in \mathbb{R}^9$ associated to optical flow $(\mathbf{x}, \mathbf{u})$ with $\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3, \mathbf{u} = [u_1, u_2, u_3]^T \in \mathbb{R}^3$ to be[4]:

$$\mathbf{a} = [u_3 y - u_2 z, u_1 z - u_3 x, u_2 x - u_1 y, x^2, 2xy, 2xz, y^2, 2yz, z^2]^T. \tag{3.45}$$

The continuous epipolar constraint (3.16) can be then rewritten as:

$$\mathbf{a}^T \mathbf{e} = 0. \tag{3.46}$$

Given a set of (possibly noisy) optical flow vectors: $(\mathbf{x}^j, \mathbf{u}^j)$, $j = 1, \ldots, n$ generated by the same motion, define a matrix $A \in \mathbb{R}^{n \times 9}$ associated to these measurements to be:

$$A = [\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^n]^T \tag{3.47}$$

where $\mathbf{a}^j$ are defined for each pair $(\mathbf{x}^j, \mathbf{u}^j)$ using (3.45). In the absence of noise, the essential vector $\mathbf{e}$ has to satisfy:

$$A\mathbf{e} = 0. \tag{3.48}$$

In order for this equation to have a unique solution for $\mathbf{e}$, the rank of the matrix A has to be eight. Thus, *for this algorithm, in general, the optical flow vectors of at least eight points*

[4]For perspective projection, $z = 1$ and $u_3 = 0$ thus the expression for $\mathbf{a}$ can be simplified.

are needed to recover the 3D velocity, i.e., $n \geq 8$, although the minimum number of optical flows needed is 5 (see Maybank [76]). When the measurements are noisy, there might be no solution of $\mathbf{e}$ for $A\mathbf{e} = 0$. As in the discrete case, we choose the solution which minimizes the error function $\|A\mathbf{e}\|^2$.

Since the continuous essential vector $\mathbf{e}$ is recovered from noisy measurements, the symmetric part s of E directly recovered from $\mathbf{e}$ is not necessarily a special symmetric matrix. Thus one can not directly use the previously derived results for special symmetric matrices to recover the 3D velocity. In the algorithms proposed in Zhuang [141, 142], such s, with the linear velocity v obtained from the skew-symmetric part, is directly used to calculate the angular velocity ω. This is an over-determined problem since three variables are to be determined from six independent equations; on the other hand, erroneous v introduces further error in the estimation of the angular velocity ω.

We thus propose a different approach: first extract the special symmetric component from the symmetric matrix s directly estimated from the continuous epipolar constraint; then recover the four possible solutions for the 3D velocity using the results obtained in Theorem 3.8; finally choose the one which has the closest linear velocity to the one given by the skew-symmetric part of E. In order to extract the special symmetric component out of a symmetric matrix, we need a projection from the space of all symmetric matrices to the special symmetric space $\mathcal{S}$, *i.e.,* a continuous version of the projection of a matrix to the essential manifold $\mathcal{E}$ given in Maybank [76].

Theorem 3.10 (Projection to the Special Symmetric Space). *If a real symmetric matrix* $F \in \mathbb{R}^{3\times 3}$ *is diagonalized as* $F = V diag\{\lambda_1, \lambda_2, \lambda_3\} V^T$ *with* $V \in SO(3)$, $\lambda_1 \geq 0, \lambda_3 \leq 0$ *and* $\lambda_1 \geq \lambda_2 \geq \lambda_3$, *then the special symmetric matrix* $E \in \mathcal{S}$ *which minimizes the error* $\|E - F\|_f^2$ *is given by* $E = V diag\{\sigma_1, \sigma_2, \sigma_2\} V^T$ *with:*

$$\sigma_1 = \frac{2\lambda_1 + \lambda_2 - \lambda_3}{3}, \quad \sigma_2 = \frac{\lambda_1 + 2\lambda_2 + \lambda_3}{3}, \quad \sigma_3 = \frac{2\lambda_3 + \lambda_2 - \lambda_1}{3}. \tag{3.49}$$

Proof: Define $\mathcal{S}_\Sigma$ to be the subspace of $\mathcal{S}$ whose elements have the same eigenvalues: $\Sigma = diag\{\sigma_1, \sigma_2, \sigma_3\}$. Thus every matrix $E \in \mathcal{S}_\Sigma$ has the form $E = V_1 \Sigma V_1^T$ for some $V_1 \in SO(3)$. To simplify the notation, define $\Sigma_\lambda = diag\{\lambda_1, \lambda_2, \lambda_3\}$. We now prove this theorem by two steps.

Step 1: We prove that the special symmetric matrix $E \in \mathcal{S}_\Sigma$ which minimizes the error $\|E - F\|_f^2$ is given by $E = V\Sigma V^T$. Since $E \in \mathcal{S}_\Sigma$ has the form $E = V_1 \Sigma V_1^T$, we get:

$$\begin{aligned} \|E - F\|_f^2 &= \|V_1 \Sigma V_1^T - V \Sigma_\lambda V^T\|_f^2 \\ &= \|\Sigma_\lambda - V^T V_1 \Sigma V_1^T V\|_f^2. \end{aligned} \tag{3.50}$$

Define $W = V^T V_1 \in SO(3)$ and W has the form:

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}. \tag{3.51}$$

Then:

$$\begin{aligned} \|E - F\|_f^2 &= \|\Sigma_\lambda - W \Sigma W^T\|_f^2 \\ &= tr(\Sigma_\lambda^2) - 2tr(W \Sigma W^T \Sigma_\lambda) + tr(\Sigma^2). \end{aligned} \tag{3.52}$$

Substituting (3.51) into the second term, and using the fact that $\sigma_2 = \sigma_1 + \sigma_3$ and W is a rotation matrix, we get:

$$\begin{aligned} tr(W \Sigma W^T \Sigma_\lambda) &= \sigma_1(\lambda_1(1 - w_3^2) + \lambda_2(1 - w_6^2) + \lambda_3(1 - w_9^2)) \\ &+ \sigma_3(\lambda_1(1 - w_1^2) + \lambda_2(1 - w_4^2) + \lambda_3(1 - w_7^2)). \end{aligned} \tag{3.53}$$

Minimizing $\|E - F\|_f^2$ is equivalent to maximizing $tr(W \Sigma W^T \Sigma_\lambda)$. From (3.53), $tr(W \Sigma W^T \Sigma_\lambda)$ is maximized if and only if $w_3 = w_6 = 0$, $w_9^2 = 1$, $w_4 = w_7 = 0$ and $w_1^2 = 1$. Since W is a rotation matrix, we also have $w_2 = w_8 = 0$ and $w_5^2 = 1$. All possible W give a unique matrix in $\mathcal{S}_\Sigma$ which minimizes $\|E - F\|_f^2$: $E = V \Sigma V^T$.

Step 2: From step one, we only need to minimize the error function over the matrices which have the form $V \Sigma V^T \in \mathcal{S}$. The optimization problem is then converted to one of minimizing the error function:

$$\|E - F\|_f^2 = (\lambda_1 - \sigma_1)^2 + (\lambda_2 - \sigma_2)^2 + (\lambda_3 - \sigma_3)^2 \tag{3.54}$$

subject to the constraint:

$$\sigma_2 = \sigma_1 + \sigma_3. \tag{3.55}$$

The formula (3.49) for $\sigma_1, \sigma_2, \sigma_3$ are directly obtained from solving this minimization problem. ∎

Remark 3.11. *For symmetric matrices which do not satisfy conditions* $\lambda_1 \geq 0$ *or* $\lambda_3 \leq 0$, *one may simply choose* $\lambda_1' = max(\lambda_1, 0)$ *or* $\lambda_3' = min(\lambda_3, 0)$.

We then have an eigenvalue-decomposition based algorithm for estimating 3D velocity from optical flow.

Algorithm 3.12 (Four Step Eigen-Decomposition Based 3D Velocity Estimation).

1. **Estimate essential vector:**

   For a given set of optical flows: $(\mathbf{x}^j, \mathbf{u}^j)$, $j = 1, \ldots, n$, *find the vector* $\mathbf{e}$ *which minimizes the error function:*

   $$V(\mathbf{e}) = \|A\mathbf{e}\|^2 \tag{3.56}$$

   subject to the condition $\|\mathbf{e}\| = 1$;

2. **Recover the special symmetric matrix:**

   Recover the vector $v_0 \in \mathbb{S}^2$ *from the first three entries of* $\mathbf{e}$ *and the symmetric matrix* $s \in \mathbb{R}^{3\times 3}$ *from the remaining six entries.*[5] *Find the eigenvalue decomposition of the symmetric matrix* s:

   $$s = V_1 diag\{\lambda_1, \lambda_2, \lambda_3\} V_1^T \tag{3.57}$$

   with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. *Project the symmetric matrix* s *onto the special symmetric space* $\mathcal{S}$. *We then have the new* $s = V_1 diag\{\sigma_1, \sigma_2, \sigma_3\} V_1^T$ *with:*

   $$\sigma_1 = \frac{2\lambda_1 + \lambda_2 - \lambda_3}{3}, \quad \sigma_2 = \frac{\lambda_1 + 2\lambda_2 + \lambda_3}{3}, \quad \sigma_3 = \frac{2\lambda_3 + \lambda_2 - \lambda_1}{3}; \tag{3.58}$$

3. **Recover velocity from the special symmetric matrix:**

   Define:

   $$\begin{cases} \lambda &= \sigma_1 - \sigma_3, \quad \lambda \geq 0, \\ \theta &= \arccos(-\sigma_2/\lambda), \quad \theta \in [0, \pi]. \end{cases} \tag{3.59}$$

   Let $V = V_1 R_Y^T \left(\frac{\theta}{2} - \frac{\pi}{2}\right) \in SO(3)$ *and* $U = -V R_Y(\theta) \in O(3)$. *Then the four possible 3D velocities corresponding to the special symmetric matrix* s *are given by:*

   $$\begin{cases} \widehat{\omega} &= U R_Z(\pm\frac{\pi}{2}) \Sigma_\lambda U^T, \quad \widehat{v} = V R_Z(\pm\frac{\pi}{2}) \Sigma_1 V^T \\ \widehat{\omega} &= V R_Z(\pm\frac{\pi}{2}) \Sigma_\lambda V^T, \quad \widehat{v} = U R_Z(\pm\frac{\pi}{2}) \Sigma_1 U^T \end{cases} \tag{3.60}$$

[5] In order to guarantee v_0 to be of unit length, one needs to "re-normalize" $\mathbf{e}$, *i.e.*, multiply $\mathbf{e}$ by a scalar such that the vector determined by the first three entries is of unit length.

where $\Sigma_\lambda = diag\{\lambda, \lambda, 0\}$ *and* $\Sigma_1 = diag\{1, 1, 0\}$;

4. **Recover velocity from the continuous essential matrix:**
From the four velocities recovered from the special symmetric matrix s in step 3, choose the pair (ω^*, v^*) *which satisfies:*

$$v^{*T} v_0 = \max_i v_i^T v_0. \tag{3.61}$$

Then the estimated 3D velocity (ω, v) *with* $\omega \in \mathbb{R}^3$ *and* $v \in \mathbb{S}^2$ *is given by:*

$$\omega = \omega^*, \quad v = v_0. \tag{3.62}$$

Both v_0 and v^* are estimates of the linear velocity. However, experimental results show that, statistically, within the tested noise levels (see next section), v_0 yields a better estimate than v^* . Here, thus, we simply choose v_0 as the estimate. Nonetheless, one can find statistical correlations between v_0 and v^* (experimentally or analytically) and obtain better estimates for v, using both v_0 and v^*. Another potential way to improve this algorithm is to study the systematic bias introduced by the least square method in step 1. A similar problem has been studied by Kanatani [53] and an algorithm was proposed to remove such bias from Zhuang's algorithm [141].

Remark 3.13. *Since both* $E, -E \in \mathcal{E}_1'$ *satisfy the same set of continuous epipolar constraints, both* $(\omega, \pm v)$ *are possible solutions for the given set of optical flows. However, as in the discrete case, one can get rid of the ambiguous solution by adding the "positive depth constraint".*

Remark 3.14. *By the way of comparison to Heeger and Jepson's algorithm [40], note that the equation (3.48) may be rewritten to highlight the dependence on optical flow as:*

$$[A_1(\mathbf{u}) \mid A_2]\,\mathbf{e} = 0$$

where $A_1(\mathbf{u}) \in \mathbb{R}^{n\times 3}$ *is a linear function of the measured optical flow and* $A_2 \in \mathbb{R}^{n\times 6}$ *is a function of the image points alone. Heeger and Jepson compute a left null space to the matrix* A_2 *(*$C \in \mathbb{R}^{(n-6)\times n}$*) and solve the equation:* $CA_1(\mathbf{u})v = 0$ *for* v *alone. Then they use* v *to obtain* ω*. Our method simultaneously estimates* $v \in \mathbb{R}^3, s \in \mathbb{R}^6$*. We make a detailed simulation comparison of these two algorithms in section 4.*

One should note that this linear algorithm is not optimal in the sense that the recovered velocity does not necessarily minimize the originally picked error function $\|A\mathbf{e}(\omega, v)\|^2$ on $\mathcal{E}_1'$ (see next section for a more detailed discussion). However, this algorithm only uses linear algebra techniques and is particularly simpler than a one which tries to optimize on the manifold $\mathcal{E}_1'$ [69].

One potential problem with the (discrete or continuous) essential approaches is that the motion estimation schemes are all based on the assumption that the translation is not zero. In this section, we study what makes the epipolar constraint fail to work in the zero-translation case.

For the discrete case, if two images are obtained from rotation alone *i.e.*, $p = 0$ and $\lambda_2 \mathbf{x}_2 = \lambda_1 R \mathbf{x}_1$, it is straightforward to check that, for all $p \in \mathbb{S}^2$, we have:

$$\mathbf{x}_1^T R^T \widehat{p} \mathbf{x}_2 \equiv 0. \tag{3.63}$$

Thus, theoretically, the estimation schemes working on the normalized essential space $\mathcal{E}_1$ will fail to converge (since there are infinitely many pairs of (R, p) satisfying the same set of epipolar constraints). In the continuous case, we have a similar situation:

Theorem 3.15. *An optical flow field* $(\mathbf{x}, \mathbf{u})$ *is obtained from a pure rotation with the angular velocity* ω *if and only if for all vectors* $v \in \mathbb{S}^2$

$$[\mathbf{u}^T, \mathbf{x}^T] \begin{bmatrix} \widehat{v} \\ \widehat{\omega}\widehat{v} \end{bmatrix} \mathbf{x} = 0. \tag{3.64}$$

Proof: $\mathbf{u} = \widehat{\omega}\mathbf{x}$ since $\mathbf{u}$ is obtained from rotation $\omega \Leftrightarrow \mathbf{u}^T(v \times \mathbf{x}) = -\mathbf{x}^T \widehat{\omega}(v \times \mathbf{x})$ for all $v \in \mathbb{S}^2 \Leftrightarrow [\mathbf{u}^T, \mathbf{x}^T] \begin{bmatrix} \widehat{v} \\ \widehat{\omega}\widehat{v} \end{bmatrix} \mathbf{x} = 0.$ ∎

This theorem implies that the velocity estimation algorithm proposed in the previous section will have trouble when the linear velocity v is zero, since there are infinite many pairs of (ω, v) satisfying the same set of continuous epipolar constraints. However, it is shown by Soatto *et al* [99] that, in the dynamical estimation approach, one can actually make use of the noise in the measurements to obtain correct estimate of the rotational component R regardless of the accuracy of the estimate for the translation vector p. The same should hold also in the continuous case. That is, even in the zero-translation case, the recovery of the angular velocity ω is still possible using dynamic estimation schemes.

Study of such schemes is beyond the scope of this paper and will be addressed in our future research work.

Example 3.16 (Kinematic Model of an Aircraft). *This example shows how to utilize the so called* **nonholonomic constraints** *(see Murray, Li and Sastry [84]) to simplify the proposed linear motion estimation algorithm in the continuous case. Let $g(t) \in SE(3)$ represent the position and orientation of an aircraft relative to the spatial frame, the inputs $\omega_1, \omega_2, \omega_3 \in \mathbb{R}$ stand for the rates of the rotation about the axes of the aircraft and $v_1 \in \mathbb{R}$ the velocity of the aircraft. Using the standard homogeneous representation for g (see Murray, Li and Sastry [84]), the kinematic equations of the aircraft motion are given by:*

$$\dot{g} = g \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & 0 \\ -\omega_2 & \omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.65}$$

where ω_1 stands for pitch rate, ω_2 for roll rate, ω_3 for yaw rate and v_1 the velocity of the aircraft. Then the 3D velocity (ω, v) in the continuous epipolar constraint (3.16) has the form: $\omega = [\omega_1, \omega_2, \omega_3]^T, v = [v_1, 0, 0]^T$. For the algorithm given in section 3.1.4, this adds extra constraints on the symmetric matrix $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$: $s_1 = s_5 = 0$ and $s_4 = s_6$. Then there are only four different essential parameters left to determine and we can re-define the essential parameter vector $\mathbf{e} \in \mathbb{R}^4$ to be: $\mathbf{e} = [v_1, s_2, s_3, s_4]^T$. Then the measurement vector $\mathbf{a} \in \mathbb{R}^4$ is to be: $\mathbf{a} = [u_3 y - u_2 z, 2xy, 2xz, y^2 + z^2]^T$. The continuous epipolar constraint can then be rewritten as:

$$\mathbf{a}^T \mathbf{e} = 0. \tag{3.66}$$

If we define the matrix A from $\mathbf{a}$ as in (3.47), the matrix $A^T A$ is a 4×4 matrix rather than a 9×9 one. For estimating the velocity (ω, v), the dimensions of the problem is then reduced from 9 to 4. In this special case, the minimum number of optical flow measurements needed to guarantee a unique solution of $\mathbf{e}$ is reduced to 3 instead of 8. Further more, the symmetric matrix s recovered from $\mathbf{e}$ is automatically in the special symmetric space S and the remaining steps of the algorithm given in section 3.1.4 can thus be dramatically simplified. From this simplified algorithm, the angular velocity $\omega = [\omega_1, \omega_2, \omega_3]^T$ can be fully recovered from the images. The velocity information can then be used for controlling the aircraft.

3.2 Experimental Results

We have carried out some initial simulations in order to study the performance of our algorithm. We chose to evaluate it in terms of bias and sensitivity of the estimates with respect to the noise in the optical flow measurements. Preliminary simulations were carried out with perfect data which was corrupted by zero-mean Gaussian noise where the standard deviation was specified in terms of pixel size and was independent of velocity. The image size was considered to be 512×512 pixels. Our algorithm has been implemented in Matlab and the simulations have been performed using example sets proposed by [109] in their paper on comparison of the egomotion estimation from optical flow[6]. The motion estimation was performed by observing the motion of a random cloud of points placed in front of the camera. Depth range of the points varied from a to b ($> a$) units of the focal length f, which was considered to be unity. For example, if the focal length is 8mm and $a = 100$ and $b = 400$, the point depth varies from 0.8 m to 3.2 m in front of the camera. This setup makes the simulation depend only on the parameter $c = (b-a)/a$, called **depth variation parameter**. The results presented below are for a fixed field of view (FOV) of 60 degrees unless otherwise stated.

3.2.1 Comparing to Subspace Methods

Each simulation consisted of 500 trials for 50 randomly sampled points in a given depth variation $[a, b] = [100, 400]$ with a fixed noise level and ratio between the optical flow due to translation and rotation for the point in the middle of the random cloud. Figures 3.2 and 3.3 compare our algorithm with Heeger and Jepson's linear subspace algorithm [40]. The presented results demonstrate the performance of the algorithm while rotating around X-axis with rate of 1^o per frame and translating along Y-axis with translation to rotation ratio of 1 and 5 respectively (for the point at the center of the random cloud). The first stage of our analysis was performed using benchmarks proposed by [109]. The bias is expressed as an angle between the average estimate out of all trials (for a given setting of parameters) and the true direction of translation and/or rotation. The sensitivity was computed as a standard deviation of the distribution of angles between each estimated vector and the average vector in case of translation and as a standard deviation of angular

[6]We would like to thank the authors in [109] for making the code for simulations of various algorithms and evaluation of their results available on the web.

differences in case of rotation.

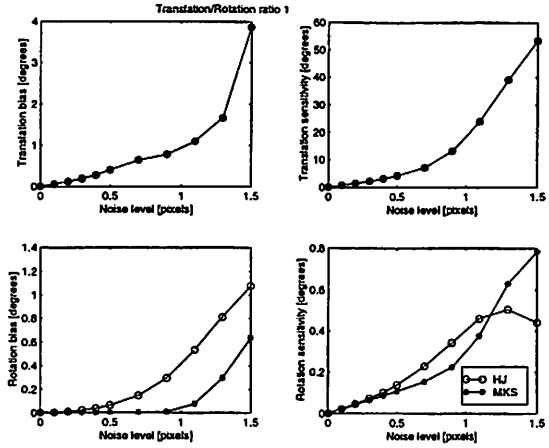


Figure 3.2: Bias for each noise level was estimated by running 500 trials and computing the average translation and rotation. The ratio between the magnitude of linear and angular velocities is 1.

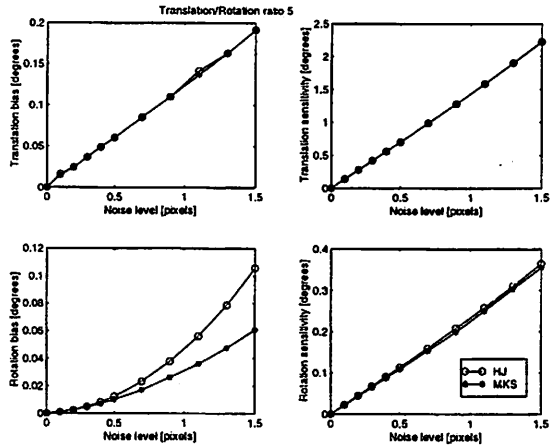


Figure 3.3: Bias for each noise level was estimated by running 500 trials and computing the average translation and rotation. The ratio between the magnitude of linear and angular velocities is 5.

We further evaluated the algorithm by varying the direction of translation and rotation. At the noise level of 0.9 pixel and translation/rotation ratio 1, for different combination of translation and rotation axis, the bias of these two algorithm are shown in Figure 3.4. From the simulation results, we observe that:

1. In terms of translational bias and sensitivity, the subspace method [40] and our algorithm have exactly the same performance at all noise levels.

2. The choice of the rotation axis does not influence the translation estimates at all for both algorithms. It does not generally influence the rotation estimates for the subspace method either but indeed influences our algorithm. This is because the decomposition of the special symmetric matrix $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$ is numerically less accurate when ω and v coincide with each other.

3. Both algorithms give much better estimates when translation along Z-axis is present. This is consistent to the sensitivity analysis done in Daniilidis [15]. In the case of translation in X-Y plane, our algorithm gives better rotation estimates than the subspace method [40], especially when the noise levels are high.

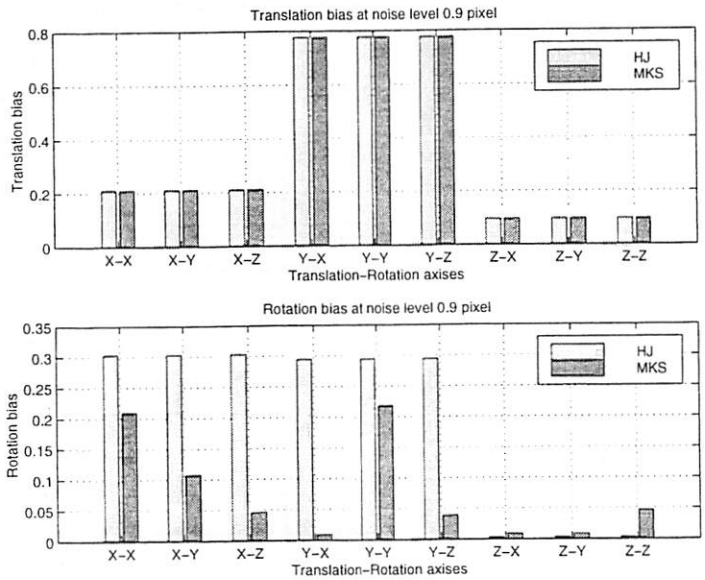


Figure 3.4: Bias dependency on combination of translation and rotation axes. For example, "X-Y" means the translation direction is in X-axis and rotation axis is the Y-axis. Bias for each combination of axes was estimated by running 500 trials at the noise level 0.9 pixel. The ratio between the magnitude of linear and angular velocities is 1.

This is due to the fact that in our algorithm the rotation is estimated simultaneously with the translation, so that its bias is only due to the bias of the initially estimated continuous essential matrix obtained by linear least squares techniques. This is in contrast to the rotation estimate used by the subspace method [40] which uses another least-squares estimation by substituting an already biased translational estimate to compute the rotation. Increasing the ratio between the magnitude of translational and rotational velocities, the performance of both algorithms improves, especially the translation estimates.

3.2.2 Bias Analysis: Relation to Nonlinear Algorithms

A disadvantage of any linear algorithm is that it tries to directly minimize the epipolar constraint, *i.e.*, the objective function:

$$V(\omega, v) = \sum_{j=1}^{n} (\mathbf{u}^{jT} \widehat{v} \mathbf{x}^j + \mathbf{x}^{jT} \widehat{\omega} \widehat{v} \mathbf{x}^j)^2. \tag{3.67}$$

But this is not the likelihood function of ω and v for commonly used noise models of the optical flow. Consequently, estimates given by linear algorithms are usually not close to maximum *a posterior* (MAP) or minimum mean square estimates (MMSE). In general, this

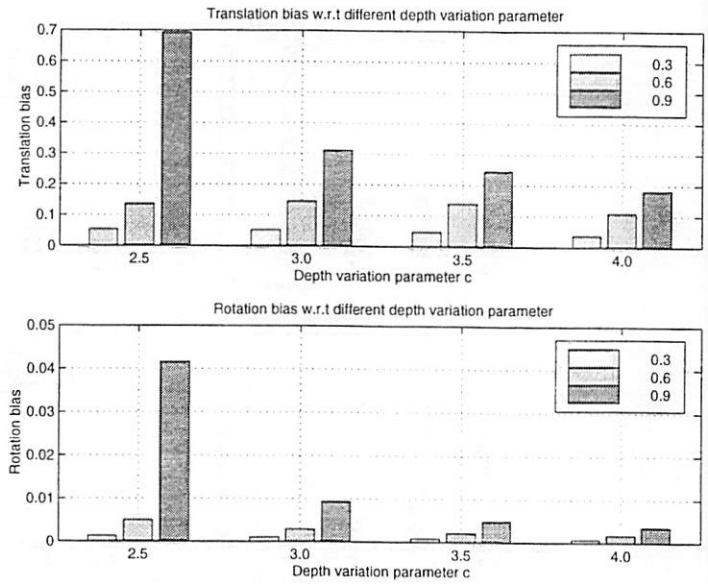


Figure 3.6: Translation bias and rotation bias with respect to different depth variation parameter c. Bias for each noise level and depth variation parameter is estimated by running 500 trials. Translation is along the X-axis and rotation axis is the Z-axis and the ratio between the magnitude of linear and angular velocities is 1.

3.2.4 Translation Estimates

Further evaluation of the results and more extensive simulations are currently underway. We believe that thoroughly understanding the source of translational bias, we can obtain even better performance by utilizing additional information about the linear velocity which is embedded in the special symmetric part of the continuous essential matrix, *i.e.*, v^* (see step 4 of the algorithm in the preceding section). In the above simulations, the linear velocity v was estimated only from the v_0, the skew symmetric part of the continuous essential matrix. Figure 3.7 demonstrates that v_0 is in general a much better estimate than v^*.

3.3 Discussion

In this chapter, we have presented a unified (linear) approach for the problem of egomotion estimation using discrete and continuous epipolar constraints. In either the discrete or continuous setting, a geometric characterization is given for the space of (discrete) essential matrices or continuous essential matrices. Such a characterization gives a natural

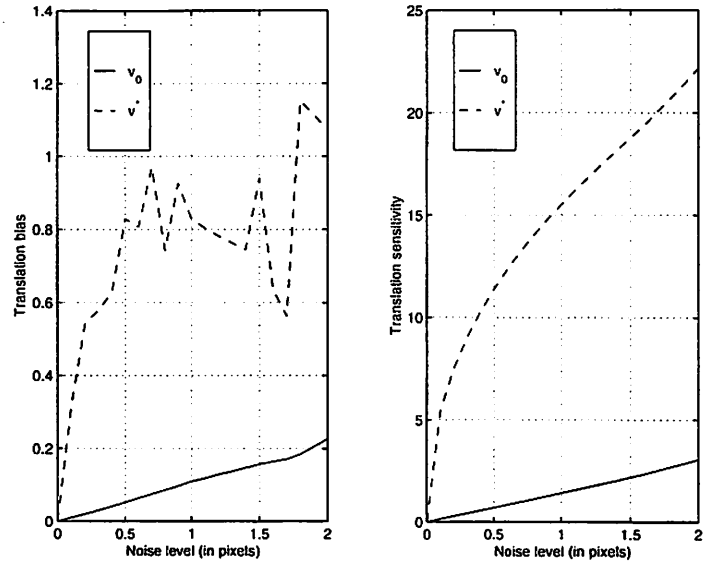


Figure 3.7: Bias and sensitivity of the translation estimates v_0 from the skew symmetric part and v^* from the special symmetric part of the continuous essential matrix. Bias and sensitivity for each noise level are estimated by running 200 trials for a cloud of 50 points. Both translation and rotation are along the X-axis and the ratio between the magnitude of linear and angular velocities is 5.

geometric interpretation for the number of possible solutions to the motion estimation problem. In addition, in the continuous case, understanding of the space of continuous essential matrices leads to a new egomotion estimation algorithm, which is a natural counterpart of the well-known three-step SVD based algorithm developed for the discrete case by [112]. In order to exploit temporal coherence of motion and improve algorithm's robustness, a dynamic (recursive) motion estimation scheme, which uses implicit extended Kalman filter for estimating the essential parameters, has been proposed by Soatto *et al* [99] for the discrete case. The reader should be aware that the same ideas certainly apply to the continuous case.

Chapter 4

Motion Recovery II: Optimal Algorithms

"Since the building of all the universe is perfect and is created by the wisdom creator, nothing arises in the universe in which one cannot see the sense of some maximum or minimum."

— L. Euler

In the previous chapter, we have discussed how to recover camera motion from two views by linear techniques. While the epipolar geometric relationships governing the motion recovery problem have been long understood, the robust or statistically less biased solutions are still sought. New studies of sensitivity of different algorithms, search for intrinsic local minima and new algorithms are still subjects of great interest. Algebraic manipulation of intrinsic geometric relationships typically gives rise to different objective functions, making the comparison of the performance of different techniques often inappropriate and often obstructing issues intrinsic to the problem. In this chapter, we provide new algorithms and insights by giving answers to the following three questions, what we believe are the main aspects of the motion and structure recovery problem (in the simplified two-view, point-feature scenario):

(i) *What is the correct choice of the objective function and its associated statistical and geometric meaning? What are the fundamental relationships among different existing objective functions?*
(ii) *What is the core optimization problem which is common to all objective functions associated with motion and structure estimation?*

(iii) *How does the choice of the objective functions and configurations affect the sensitivity and robustness of the estimates? What is the effect of the bas relief ambiguity and other ambiguities on the sensitivity and robustness of the proposed algorithms?*

The seminal work of Longuet-Higgins [60] on the characterization of the so called **epipolar constraint**, has enabled the decoupling of the structure and motion problems and led to the development of numerous linear and nonlinear algorithms for motion estimation (see [22, 53, 76, 131] for overviews). The epipolar constraint has been formulated both in a discrete and a continuous setting in Chapter 3 and this work has demonstrated the possibility of a parallel development of algorithms for both cases: namely using point feature correspondence and optical flow. A preliminary analysis of linear and nonlinear techniques, exploring the use of different objective functions can be found in [63].

While the (analytic) geometrical aspects of the linear approach have been understood, the proposed solutions to the problem have been shown very sensitive to noise and have often failed in practical applications. These experiences have motivated further studies which focus on the use of a statistical analysis of existing techniques and understanding of various assumptions which affect the performance of existing algorithms. These studies have been done both in an analytical [14, 102] and experimental setting [109]. The appeal of linear algorithms which use the epipolar constraint (in the discrete case [53, 60, 76, 131] and in the continuous case [50, 67, 108]) is the closed form solution to the problem which, in the absence of noise, provides true estimate of the motion. However, a further analysis of linear techniques reveals an inherent bias in the translation estimates [50]. Attempts made to compensate for the bias slightly improve the performance of the linear techniques [53].

Such attempts to remove the bias have led to different choice of nonlinear objective functions. The performance of numerical optimization techniques which minimize nonlinear objective functions has been shown superior to linear ones. The objective functions used are either (normalized) versions of the epipolar constraint or distances between measured and reconstructed image points (the so called reprojection error) [129, 63, 140, 45]. These techniques either require iterative numerical optimization [131, 99] or use Monte-Carlo simulations [50] to sample the space of the unknown parameters. Extensive experiments revealed problems with convergence when initialized far away from the true solution [109]. Since nonlinear objective functions have been obtained from quite different approaches, it is necessary to understand the relationship among all the existing objective functions. Al-

though a preliminary comparison has been made in [140], in this chapter, we provide a more detailed and rigorous account of this relationship and how it affects the complexity of the optimization. In this chapter, we will show, by answering the question **(i)**, "minimizing epipolar constraint", "minimizing (geometrically or statistically[1]) normalized epipolar constraint" [63, 129, 140], "minimizing reprojection error" [129], and "triangulation" [33] can all be unified in a single geometric optimization procedure, the so called "optimal triangulation". As a by-product of this approach, a much simpler triangulation method than [33] is given along with the proposed algorithm. A highlight of our method is an iterative scheme between motion and structure without introducing any 3D scale (or depth).

Different objective functions have been used in different optimization techniques [45, 107, 129]. Horn [45] first proposed an iterative procedure where the update of the estimate takes into account the orthonormal constraint of the unknown rotation. This algorithm and the algorithm proposed in [107] are some of the few which explicitly consider the differential geometric properties of the rotation group $SO(3)$. In most cases, the underlying search space has been parameterized for computational convenience instead of being loyal to its intrinsic geometric structure. Consequently, in these algorithms, solving for optimal updating direction typically involves using Lagrangian multipliers to deal with the constraints on the search space; and "walking" on such a space is done approximately by an **update-then-project** procedure, rather than exploiting geometric properties of the entire space of essential matrices as characterized in Chapter 3 or in [99]. As an answer to the question **(ii)**, we will show that optimizing existing objective functions can all be reduced to optimization problems on the essential manifold. Due to recent developments of optimization techniques on Riemannian manifolds (especially on Lie groups and homogeneous spaces) [97, 19], we are able to explicitly compute all the necessary ingredients, such as **gradient, Hessian and geodesics**, for carrying out intrinsic nonlinear search schemes. In this chapter, we will first give a review of the nonlinear optimization problem associated with the motion and structure recovery. Using a generalized Newton's algorithm as a prototype example, we will apply our methods to solve the optimal motion and structure estimation problem by exploiting the intrinsic Riemannian structure of the essential manifold. The rate of convergence of the algorithm is also studied in some detail. We believe the proposed geometric algorithm will provide us with an analytic framework for design of

[1] In the literature, they are respectively referred to as distance between points and epipolar lines, and gradient-weighted epipolar errors [140] or epipolar improvement [129].

(Kalman) filters on the essential manifold for dynamic motion estimation (see [99]). It also provides us new perspectives for design of algorithms for multiple views.

In this chapter, only the discrete case will be studied, since in the continuous case the search space is essentially Euclidean and good optimization schemes already exist and have been well studied, see [98, 139]. For the continuous case, recent studies [98] have clarified the source of some of the difficulties (for example, rotation and translation confounding) from the point of view of noise and explored the source and presence of local extrema which are intrinsic to the structure from motion problem (*i.e.,* these local extrema are independent of the choice of objective functions). The bas relief ambiguity, in general, can characterized as the most sensitive direction in which the rotation and translation estimates are prone to be confound with each other (for example, see [1, 98, 129] for a more detailed analysis). Here we apply the same line of thought to the discrete case. Since the bas relief effect is evident only when the field of view and the depth variation of the scene are small, we here are more interested in characterizing, besides the bas relief ambiguity, other intrinsic extrema which may show up at a high noise level even for a general configuration, *i.e.,* with large base line, field of view and depth variation. As an answer to the question **(iii)**, we will show both analytically and experimentally that some ambiguities are introduced at a high noise level by certain bifurcation of the objective function and usually result in a sudden 90^o flip in the translation estimate. Understanding such ambiguities is crucial for properly evaluating the performance (especially the robustness) of the algorithms when applied to general configurations. Based on analytical and experimental results, we will give a clear profile of the performance of different algorithms over a large range of signal-to-noise ratio, and under various motion and structure configurations.

Chapter Outline

Section 4.1 relies on some familiarity with Edelman *et al*'s work [19] on geometric optimization and some background of Riemannian geometry (good references for Riemannian geometry are [55, 103]).[2] This section basically outlines how to optimize various objective functions associated to the motion recovery problem using the (Riemannian) Newton's algorithm. Formulae of all the necessary ingredients such as gradient, Hessian and geodesics have been explicitly spelled out. Appendix A provides extra details that fill the

[2]Readers who are not familiar with differential geometry terms may skip technical details in this section without losing much continuity.

gap between Edelman's work and our application. Different objective functions proposed in the literature are unified in Section 4.2 by a single optimization procedure proposed for estimating optimal structure and motion altogether. This procedure gives clear answers to both questions (i) and (ii). Section 4.3 discusses extrema of an objective function on the essential manifold. Among all the possible ambiguities, we characterize those which most likely occur in the motion and structure recovery problem. Sensitivity study and experimental comparison between different objective functions are given in Section 4.4. Section 4.3 and 4.4 together give a clear answer to the question (iii).

4.1 Optimal Motion Recovery

In this section, we apply the Riemannian Newton's algorithm to various objective functions associated with the motion recovery problem in computer vision. Relationship among different objective functions will be studied in detail in the section after.

4.1.1 Minimizing Epipolar Constraints

From Chapter 3, we know that two corresponding image points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ satisfy the so called epipolar constraint:

$$\mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 \;=\; 0 \tag{4.1}$$

where $R \in SO(3)$ and $T \in \mathbb{S}^2$ are relative rotation and translation between the two image frames, respectively. Thus to recover the motion (R,T) from a given set of image correspondences $\mathbf{x}_1^j, \mathbf{x}_2^j \in \mathbb{R}^3, j = 1, \ldots, n$, it is natural to minimize the following objective function:

$$F(R,T) \;=\; \sum_{j=1}^{n} (\mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j)^2, \quad \mathbf{x}_1^j, \mathbf{x}_2^j \in \mathbb{R}^3, (R,T) \in SO(3) \times \mathbb{S}^2. \tag{4.2}$$

In this section, we apply the Newton's algorithm introduced in Chapter A to solve this problem. We will give explicit formulae for calculating all the ingredients needed: *geodesics*, *gradient* G of F, *Hessian* $\mathrm{Hess}(\cdot,\cdot)$ of F and the *optimal updating vector* $\Delta = -\mathrm{Hess}^{-1} G$ (and we will show later how these formulae can be extensively reused for obtaining corresponding formulae for other objective functions). It is well known that an explicit formula for the Hessian is also important for sensitivity analysis of motion estimation [14].

Further, using the formula for the Hessian, we will be able to show that, under certain conditions, the Hessian is guaranteed non-degenerate, whence the Newton's algorithm has quadratic rate of convergence.

Instead of using formulae given in the previous section, the computation of the gradient and Hessian can also be carried out by using explicit formulae for geodesics on these manifolds. On $SO(3)$, the formula for the geodesic at R in the direction $\Delta_1 \in T_R(SO(3))$ is:

$$R(t) = \exp(R, \Delta_1 t) = e^{\widehat{\omega} t} R = (I + \widehat{\omega} \sin t + \widehat{\omega}^2 (1 - \cos t)) R \tag{4.3}$$

where $t \in \mathbb{R}, \widehat{\omega} = \Delta_1 R^T \in so(3)$. The last equation is called the *Rodrigues' formula* (see [84]). $\mathbb{S}^2$ (as a Stiefel manifold) also has very simple expression for geodesics. At the point T along the direction $\Delta_2 \in T_T(\mathbb{S}^2)$ the geodesic is given by:

$$T(t) = \exp(T, \Delta_2 t) = T \cos \sigma t + U \sin \sigma t \tag{4.4}$$

where $\sigma = \|\Delta_2\|$ and $U = \Delta_2 / \sigma$, then $T^T U = 0$ since $T^T \Delta_2 = 0$.

Using the formulae (4.3) and (4.4) for geodesics , we can calculate the first and second derivatives of $F(R, T)$ in the direction $\Delta = (\Delta_1, \Delta_2) \in T_R(SO(3)) \times T_T(\mathbb{S}^2)$:

$$dF(\Delta) = \frac{dF(R(t), T(t))}{dt}\Big|_{t=0} = \sum_{j=1}^{n} \mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j \left(\mathbf{x}_2^{jT} \widehat{T} \Delta_1 \mathbf{x}_1^j + \mathbf{x}_2^{jT} \widehat{\Delta}_2 R \mathbf{x}_1^j \right),$$

$$\begin{aligned} \mathrm{Hess}(\Delta, \Delta) &= \frac{d^2 F(R(t), T(t))}{dt^2}\Big|_{t=0} \\ &= \sum_{j=1}^{n} \left[\mathbf{x}_2^{jT} (\widehat{T} \Delta_1 + \widehat{\Delta}_2 R) \mathbf{x}_1^j \right]^2 + \mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j \left[\mathbf{x}_2^{jT} (-\widehat{T} R \Delta_1^T \Delta_1 - \widehat{T} R \Delta_2^T \Delta_2 + 2 \widehat{\Delta}_2 \Delta_1) \mathbf{x}_1^j \right]. \end{aligned}$$

From the first order derivative, the gradient $G = (G_1, G_2) \in T_R(SO(3)) \times T_S(\mathbb{S}^2)$ of $F(R, T)$ is:

$$G = \sum_{j=1}^{n} \mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j \left(\widehat{T}^T \mathbf{x}_2^j \mathbf{x}_1^{jT} - R \mathbf{x}_1^j \mathbf{x}_2^{jT} \widehat{T} R, \; -\widehat{\mathbf{x}_2^j} R \mathbf{x}_1^j - T \mathbf{x}_1^{jT} R^T \widehat{\mathbf{x}_2^j} T \right) \tag{4.5}$$

It is direct to check that $G_1 R^T \in so(3)$ and $T^T G_2 = 0$, so the G given by the above expression is a vector in $T_R(SO(3)) \times T_T(\mathbb{S}^2)$.

For any pair of vectors $X, Y \in T_R(SO(3)) \times T_T(\mathbb{S}^2)$, polarize $\text{Hess}(\Delta, \Delta)$ to get the expression for $\text{Hess}(X, Y)$:

$$\begin{aligned}
& \text{Hess}(X, Y) \\
= \; & \frac{1}{4}\left[\text{Hess}(X+Y, X+Y) - \text{Hess}(X-Y, X-Y)\right] \\
= \; & \sum_{j=1}^{n} \mathbf{x}_2^{jT}(\widehat{T}X_1 + \widehat{X}_2 R)\mathbf{x}_1^j \mathbf{x}_2^{jT}(\widehat{T}Y_1 + \widehat{Y}_2 R)\mathbf{x}_1^j \\
+ \; & \mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j \left[\mathbf{x}_2^{jT}\left(-\frac{1}{2}\widehat{T}R(X_1^T Y_1 + Y_1^T X_1) - \widehat{T}RX_2^T Y_2 + (\widehat{Y}_2 X_1 + \widehat{X}_2 Y_1)\right)\mathbf{x}_1^j\right] \quad (4.6)
\end{aligned}$$

To make sure this expression is correct, if we let $X = Y = \Delta$, then we get the same expression for $\text{Hess}(\Delta, \Delta)$ as that obtained directly from the second order derivative.

The following theorem shows that this Hessian is non-degenerate in a neighborhood of the optimal solution, therefore the Newton's algorithm will have a quadratic rate of convergence by Theorem 3.4 of Smith [97].

Theorem 4.1 (Nondegeneracy of Hessian). *Consider the objective function $F(R, T)$ as above. Its Hessian is not degenerate in a neighborhood of the optimal solution if there is a unique (up to a scale) solution to the system of linear equations:*

$$\mathbf{x}_2^{jT} E \mathbf{x}_1^j = 0, \quad E \in \mathbb{R}^{3\times 3}, \quad j = 1, \ldots, n.$$

If so, the Riemannian Newton's algorithm has quadratic rate of convergence.

Proof: It suffices to prove for any $\Delta \neq 0$, $\text{Hess}(\Delta, \Delta) > 0$. According to the epipolar constraint, at the optimal solution, we have $\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j \equiv 0$. The Hessian is then simplified to:

$$\text{Hess}(\Delta, \Delta) = \sum_{j=1}^{n}\left[\mathbf{x}_2^{jT}(\widehat{T}\Delta_1 + \widehat{\Delta}_2 R)\mathbf{x}_1^j\right]^2.$$

Thus $\text{Hess}(\Delta, \Delta) = 0$ if and only if

$$\mathbf{x}_2^{jT}(\widehat{T}\Delta_1 + \widehat{\Delta}_2 R)\mathbf{x}_1^j = 0, \quad j = 1, \ldots, n.$$

Since we also have

$$\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j = 0, \quad j = 1, \ldots, n.$$

Then both $\widehat{T}\Delta_1 + \widehat{\Delta}_2 R$ and $\widehat{T}R$ are solutions for the same system of linear equations which by assumption has a unique solution, hence $\text{Hess}(\Delta, \Delta) = 0$ if and only if

$$\begin{aligned}
& \widehat{T}\Delta_1 + \widehat{\Delta}_2 R = \lambda \widehat{T} R, \quad \text{for some } \lambda \in \mathbb{R} \\
\Leftrightarrow \quad & \widehat{T}\widehat{\omega} + \widehat{\Delta}_2 = \lambda \widehat{T}, \quad \text{for } \omega = \Delta_1 R^T \\
\Leftrightarrow \quad & \widehat{T}\widehat{\omega} = \lambda \widehat{T}, \text{ and } \Delta_2 = 0, \quad \text{since } T^T \Delta_2 = 0 \\
\Leftrightarrow \quad & \omega = 0, \text{ and } \Delta_2 = 0, \quad \text{since } T \neq 0 \\
\Leftrightarrow \quad & \Delta = 0.
\end{aligned}$$

■

Remark 4.2. *In the previous theorem, regarding the 3×3 matrix E in the equations $\mathbf{x}_2^{jT} E \mathbf{x}_1^j = 0$ as a vector in $\mathbb{R}^9$, one needs at least eight equations to uniquely solve E up to a scale. This implies that we need at least eight image correspondences $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^n, n \geq 8$ to guarantee the Hessian non-degenerate whence the iterative search algorithm converges in quadratic rate. If we study this problem more carefully, using transversality theory, one may show that five image correspondences in general position is the minimal data to guarantee the Hessian non-degenerate [76]. However, the five point technique usually leads to many (up to twenty) ambiguous solutions, as pointed out by Horn [45]. Moreover, numerical errors usually make the algorithm not work exactly on the essential manifold and the extra solutions for the equations $\mathbf{x}_2^{jT} E \mathbf{x}_1^j = 0$ may cause the algorithm to converge very slowly in these directions. It is not just a coincidence that the conditions for the Hessian to be non-degenerate are exactly the same as that for the eight-point linear algorithm (see [76, 67]) to have a unique solution. A heuristic explanation is that the objective function here is a quadratic form of the epipolar constraint which the linear algorithm is directly based on.*

Returning to the Newton's algorithm, assume that the Hessian is non-degenerate, *i.e.*, invertible. Then, we need to solve for the optimal updating vector Δ such that $\Delta = \text{Hess}^{-1} G$, or equivalently:

$$\text{Hess}(Y, \Delta) = g(-G, Y) = -dF(Y), \quad \text{for all vector fields } Y.$$

Pick five linearly independent vectors, *i.e.*, a basis of $T_R(SO(3)) \times T_S(\mathbb{S}^2)$: $E^k, k = 1, \ldots, 5$. One then obtains five linear equations:

$$\text{Hess}(E^k, \Delta) = -dF(E^k), \quad k = 1, \ldots, 5.$$

Since the Hessian is invertible, these five linear equations uniquely determine Δ. In particular, one can choose the simplest basis such that for $k = 1, 2, 3$: $E^k = (\widehat{e}_k R, 0)$ with e_k, $k = 1, 2, 3$ the standard basis for $\mathbb{R}^3$, and for $k = 4, 5$: $E^k = (0, e_j)$ such that $\{T, e_4, e_5\}$ form an orthonormal basis for $\mathbb{R}^3$. The vectors e_4, e_5 can be obtained using Gram-Schmidt process.

Define a 5×5 matrix $A \in \mathbb{R}^{5\times 5}$ and a 5 dimensional vector $\mathbf{b} \in \mathbb{R}^5$ to be:

$$A_{kl} = \text{Hess}(E^k, E^l), \quad \mathbf{b}_k = -dF(E^k), \quad k, l = 1, \ldots, 5.$$

Then solve for the vector $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5]^T \in \mathbb{R}^5$:

$$\mathbf{a} = A^{-1}\mathbf{b}.$$

Let $u = [a_1, a_2, a_3]^T \in \mathbb{R}^3$ and $v = a_4 e_4 + a_5 e_5 \in \mathbb{R}^3$. Then for the optimal updating vector $\Delta = (\Delta_1, \Delta_2)$, we have $\Delta_1 = \widehat{u}R$ and $\Delta_2 = v$. We now summarize the Riemannian Newton algorithm for the optimal motion recovery, which can be directly implemented.

Algorithm 4.3 (Riemannian Newton's Algorithm for 3D Motion Recovery). Objective Function:

$$F(R, T) = \sum_{j=1}^{n} (\mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j)^2, \quad \mathbf{x}_1^j, \mathbf{x}_2^j \in \mathbb{R}^3, (R, T) \in SO(3) \times \mathbb{S}^2.$$

1. **Compute the optimal updating vector:**

   At the point $(R, T) \in SO(3) \times \mathbb{S}^2$, compute the optimal updating vector $\Delta = -Hess^{-1} G$:

   - *Compute the vectors e_4, e_5 from T using Gram-Schmidt process and obtain the five basis tangent vectors $E^k \in T_R(SO(3)) \times T_T(\mathbb{S}^2), 1 \leq k \leq 5$ as defined in the above,*
   - *Compute the 5×5 matrix $A_{kl} = Hess(E^k, E^l), 1 \leq k, l \leq 5$,*
   - *Compute the 5 dimensional vector $\mathbf{b}_k = -dF(E^k), 1 \leq k \leq 5$,*
   - *Compute the vector $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5]^T \in \mathbb{R}^5$ such that $\mathbf{a} = A^{-1}\mathbf{b}$,*
   - *Define $u = [a_1, a_2, a_3]^T \in \mathbb{R}^3$ and $v = a_4 e_4 + a_5 e_5 \in \mathbb{R}^3$. Then the optimal updating vector:*

   $$\Delta = -Hess^{-1} G = (\widehat{u}R, v).$$

2. **Update the search state:**

   Move (R,T) *in the direction* Δ *along the geodesic to* $(\exp(R,\Delta_1),\exp(T,\Delta_2))$, *using the formula for geodesics on* $SO(3)$ *and* $\mathbb{S}^2$ *respectively:*

$$\begin{aligned}\exp(R,\Delta_1) &= (I+\widehat{\omega}\sin t+\widehat{\omega}^2(1-\cos t))R,\\ \exp(T,\Delta_2) &= T\cos\sigma+U\sin\sigma,\end{aligned}$$

   where $t=\sqrt{\frac{1}{2}tr(\Delta_1^T\Delta_1)},\omega=\Delta_1R^T/t$ *and* $\sigma=\sqrt{\frac{1}{2}\Delta_2^T\Delta_2}$, $U=\Delta_2/\sigma$.

3. **Return** *to Step 1 if* $\|\mathbf{b}\|\geq\epsilon$ *for some pre-determined* $\epsilon>0$.

Remark 4.4. *From calculations above, we note that one can consider a more general objective function with a (positive) weights* $w_j\in\mathbb{R}^+$ *associated with each image correspondence* $(\mathbf{x}_2^j,\mathbf{x}_1^j)$*:*

$$F(R,T)=\sum_{j=1}^{n}w_j(\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2,\quad \mathbf{x}_1^j,\mathbf{x}_2^j\in\mathbb{R}^3,(R,T)\in SO(3)\times\mathbb{S}^2.$$

For example, one may choose $w_j^{-1}=\|\mathbf{x}_1^j\|^2\|\mathbf{x}_2^j\|^2$ *to convert the image points from perspective projection to spherical projection. Then, in the above algorithm, the expressions for the geodesics, the gradient and Hessian only need to be slightly modified.*

4.1.2 Minimizing Normalized Epipolar Constraints

Although the epipolar constraint (3.2) gives the only necessary (depth independent) condition that image pairs have to satisfy, motion estimates obtained from minimizing the objective function (4.2):

$$F(R,T) = \sum_{j=1}^{n}(\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2,\quad \mathbf{x}_1^j,\mathbf{x}_2^j\in\mathbb{R}^3,(R,T)\in SO(3)\times\mathbb{S}^2. \tag{4.7}$$

are not necessarily statistically or geometrically optimal for the commonly used noise model of image correspondences. In general, in order to get less biased estimates, we need to *normalize* (or weight) the epipolar constraints properly, which has been initially observed in [129]. In this section, we will give a brief account of these normalized versions of epipolar constraints. These normalized versions in general are still functions defined on the essential manifold. The reason will become clear in the next section when we see that these normalizations in fact can be unified by a single procedure for getting optimal estimates of motion and structure.

We here discuss this issue for the perspective projection case.[3] In the perspective projection case, coordinates of image points $\mathbf{x}_1$ and $\mathbf{x}_2$ are of the form $[x, y, 1]^T \in \mathbb{R}^3$. Suppose that the actual measured image coordinates of n pairs of image points are:

$$\mathbf{x}_1^j = \tilde{\mathbf{x}}_1^j + \alpha^j, \quad \mathbf{x}_2^j = \tilde{\mathbf{x}}_2^j + \beta^j, \quad j = 1, \ldots, n \tag{4.8}$$

where $\tilde{\mathbf{x}}_1^j$ and $\tilde{\mathbf{x}}_2^j$ are ideal (noise free) image coordinates, $\alpha^j = [\alpha_1^j, \alpha_2^j, 0]^T \in \mathbb{R}^3$ and $\beta^j = [\beta_1^j, \beta_2^j, 0]^T \in \mathbb{R}^3$, and $\alpha_1^j, \alpha_2^j, \beta_1^j$ and β_2^j are independent Gaussian random variables of identical distribution $N(0, \sigma^2)$. Substituting $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ into the epipolar constraint (3.2), we obtain:

$$\mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j \;\; = \;\; \beta^{jT} \widehat{T} R \tilde{\mathbf{x}}_1^j + \tilde{\mathbf{x}}_2^{jT} \widehat{T} R \alpha^j + \beta^{jT} \widehat{T} R \alpha^j.$$

Since the image coordinates $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ usually are magnitude larger than α^j and β^j, one can omit the last term in the equation above. Then $\mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j$ are independent random variables *approximately* of Gaussian distribution $N(0, \sigma^2(\|\widehat{e}_3 \widehat{T} R \mathbf{x}_1^j\|^2 + \|\mathbf{x}_2^{jT} \widehat{T} R \widehat{e}_3\|^2))$ where $e_3 = [0, 0, 1]^T \in \mathbb{R}^3$. If we assume the *a prior* distribution of the motion (R, T) is uniform, the maximum *a posterior* (MAP) estimates of (R, T) is then the global minimum of the objective function:

$$F_s(R, T) = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT} \widehat{T} R \mathbf{x}_1^j)^2}{\|\widehat{e}_3 \widehat{T} R \mathbf{x}_1^j\|^2 + \|\mathbf{x}_2^{jT} \widehat{T} R \widehat{e}_3^T\|^2}, \quad \mathbf{x}_1^j, \mathbf{x}_2^j \in \mathbb{R}^3, (R, T) \in SO(3) \times \mathbb{S}^2. \tag{4.9}$$

We here use F_s to denote the **statistically normalized** objective function associated with the epipolar constraint. This objective function is also referred in the literature under the name **gradient criteria** [63] or **epipolar improvement** [131]. Therefore, we have:

$$(R, T)_{MAP} \;\; \approx \;\; \arg\min F_s(R, T) \tag{4.10}$$

Note that in the noise free case, F_s achieves zero, just like the unnormalized objective function F given by equation (4.2). Asymptotically, MAP estimates approach the unbiased minimum mean square estimates (MMSE). So, in general, the MAP estimates give less biased estimates than the unnormalized objective function F.

Note that F_s is still a function defined on the manifold $SO(3) \times \mathbb{S}^2$. The discussion given in Section A.3 about optimizing a general function defined on the essential manifold

[3] The spherical projection case is similar and is omitted for simplicity.

certainly applies to F_s. Moreover, note that the numerator of each term of F_s is the same as that in F, and the denominator of each term in F_s is simply:

$$\|\hat{e}_3\hat{T}R\mathbf{x}_1^j\|^2 + \|\mathbf{x}_2^{jT}\hat{T}R\hat{e}_3^T\|^2 = (e_1^T\hat{T}R\mathbf{x}_1^j)^2 + (e_2^T\hat{T}R\mathbf{x}_1^j)^2 + (\mathbf{x}_2^{jT}\hat{T}Re_1)^2 + (\mathbf{x}_2^{jT}\hat{T}Re_1)^2 \quad (4.11)$$

where $e_1 = [1,0,0]^T \in \mathbb{R}^3$ and $e_2 = [0,1,0]^T \in \mathbb{R}^3$. That is, components of each term of the normalized objective function F_s are essentially of the same form as that in the unnormalized one F. Therefore, we can exclusively use the formulae of the first and second order derivatives $dF(\Delta)$ and $\text{Hess}F(\Delta,\Delta)$ of the unnormalized objective function F to express those for the normalized objective F_s by simply replacing $\mathbf{x}_1^j$ or $\mathbf{x}_2^j$ with e_1 or e_2 at proper places. This is one of the reasons why the epipolar constraint is so important and studied first. Since for each term of F_s, we now need to evaluate the derivatives of five similar components $(e_1^T\hat{T}R\mathbf{x}_1^j)^2, (e_2^T\hat{T}R\mathbf{x}_1^j)^2, (\mathbf{x}_2^{jT}\hat{T}Re_1)^2, (\mathbf{x}_2^{jT}\hat{T}Re_1)^2$ and $(\mathbf{x}_2^{jT}\hat{T}R\mathbf{x}_1^j)^2$, as opposed to one in the unnormalized case, the Newton's algorithm for the normalized objective function is in general five times slower than that for the unnormalized objective function F. But the normalized one gives statistically much better estimates, as we will demonstrate in Section 4.4.

Another commonly used criterion to recover motion is to minimize the geometric distances between image points and corresponding epipolar lines. This objective function is given as:

$$F_g(R,T) = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\hat{T}R\mathbf{x}_1^j)^2}{\|\hat{e}_3\hat{T}R\mathbf{x}_1^j\|^2} + \frac{(\mathbf{x}_2^{jT}\hat{T}R\mathbf{x}_1^j)^2}{\|\mathbf{x}_2^{jT}\hat{T}R\hat{e}_3^T\|^2}, \quad \mathbf{x}_1^j, \mathbf{x}_2^j \in \mathbb{R}^3, (R,T) \in SO(3) \times \mathbb{S}^2. \quad (4.12)$$

We here use F_g to denote this **geometrically normalized** objective function. For a more detailed derivation and geometric meaning of this objective function see [63, 140]. Notice that, similar to F and F_s, F_g is also a function defined on the essential manifold and can be minimized using the given Newton's algorithm.

The relationship between the statistically normalized objective function F_s and the geometrically normalized objective function F_g will be clearly revealed in the next section when we study the optimal motion and structure recovery as a constrained optimization problem. As we know from [68], in the continuous case, the normalization has no effect when the translational motion is in the image plane, *i.e.*, the unnormalized and normalized objective functions are in fact equivalent. For the discrete case, we have a similar claim. Suppose the camera motion is given by $(R,T) \in SE(3)$ with $T \in \mathbb{S}^2$ and $R = e^{\hat{\omega}\theta}$ for some

$\omega \in \mathbb{S}^2$ and $\theta \in \mathbb{R}$. If $\omega = [0,0,1]^T$ and $T = [T_1, T_2, 0]^T$, *i.e.*, the translation direction is in the image plane, then it is direct to check that $\|\hat{e}_3 \hat{T} R \mathbf{x}_1^j\|^2 = \|\mathbf{x}_2^{jT} \hat{T} R \hat{e}_3^T\|^2 = 1$. Hence, in this case, all the three objective functions F, F_s and F_g are very similar to each other around the actual (R,T).[4] Practically, this implies the normalization will have little effect on the motion estimates, as will be verified by the simulation.[5] Therefore, in certain cases, minimizing the objective function F which is directly related to the epipolar constraint is not necessarily a wrong thing to do.

4.2 Optimal Triangulation

Note that, in the presence of noise, for the motion (R,T) recovered from minimizing the unnormalized or normalized objective functions F, F_s or F_g, the value of the objective functions is not necessarily zero. That is, in general:

$$\mathbf{x}_2^{jT} \hat{T} R \mathbf{x}_1^j \neq 0, \quad j = 1, \ldots, n. \tag{4.13}$$

Consequently, if one directly uses $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ to recover the 3D location of the point to which the two images $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ correspond, the two rays corresponding to $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ may not be coplanar, hence may not intersect at one 3D point. Also, when we derived the normalized epipolar constraint F_s, we ignored the second order terms. Therefore, rigorously speaking, it does not give the exact MAP estimates. Here we want to clarify the effect of such approximation on the estimates both analytically and experimentally. Furthermore, since F_g also gives another reasonable approximation of the MAP estimates, can we relate both F_s and F_g to the MAP estimates in a unified way? This will be studied in this section. Experimental comparison will be given in the next section.

Under the assumption of Gaussian noise model (4.8), in order to obtain the optimal (MAP) estimates of camera motion and a consistent 3D structure reconstruction, in principle we need to solve the following optimization problem:

Optimal Triangulation Problem: *Seek camera motion (R,T) and points $\tilde{\mathbf{x}}_1^j \in \mathbb{R}^3$ and $\tilde{\mathbf{x}}_2^j \in \mathbb{R}^3$ on the image plane such that they minimize the distance from $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$:*

$$F_t(R, T, \tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j) = \sum_{j=1}^{n} \|\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j\|^2 \tag{4.14}$$

[4] Around a small neighborhood of the actual (R,T), they only differ by high order terms.

[5] Strictly speaking, this is the case only when the noise level is low, *i.e.*, corrupted objective functions are not yet so different from the noise-free one.

subject to the conditions:

$$\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j = 0, \quad \tilde{\mathbf{x}}_1^{jT}e_3 = 1, \quad \tilde{\mathbf{x}}_2^{jT}e_3 = 1, \quad j = 1,\ldots,n. \tag{4.15}$$

We here use F_t to denote the objective function for triangulation. This objective function is also referred in literature as the reprojection error. Unlike [33], we do not assume a known essential matrix $\widehat{T}R$. Instead we seek $\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j$ *and* (R,T) which minimize the objective function F_t given by (4.14). The objective function F_t then implicitly depends on the variables (R,T) through the constraints (4.15). Clearly, the optimal solution to this problem is exactly equivalent to the optimal MAP estimates of both motion *and* structure. Using Lagrangian multipliers, we can convert the minimization problem to an unconstrained one:

$$\min_{R,T,\tilde{\mathbf{x}}_1^j,\tilde{\mathbf{x}}_2^j} \sum_{j=1}^{n} \|\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j\|^2 + \lambda^j \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j + \gamma^j(\tilde{\mathbf{x}}_1^{jT}e_3 - 1) + \eta^j(\tilde{\mathbf{x}}_2^{jT}e_3 - 1).$$

The necessary conditions for minima of this objective function are:

$$2(\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j) + \lambda^j R^T\widehat{T}^T\tilde{\mathbf{x}}_2^j + \gamma^j e_3 = 0 \tag{4.16}$$

$$2(\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j) + \lambda^j\widehat{T}R\tilde{\mathbf{x}}_1^j + \eta^j e_3 = 0 \tag{4.17}$$

Under the necessary conditions, we obtain:

$$\begin{cases} \tilde{\mathbf{x}}_1^j &= \mathbf{x}_1^j - \frac{1}{2}\lambda^j\hat{e}_3^T\hat{e}_3R^T\widehat{T}^T\tilde{\mathbf{x}}_2^j \\ \tilde{\mathbf{x}}_2^j &= \mathbf{x}_2^j - \frac{1}{2}\lambda^j\hat{e}_3^T\hat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j \\ \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j &= 0 \end{cases} \tag{4.18}$$

where λ^j is given by:

$$\lambda^j = \frac{2(\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j + \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j)}{\tilde{\mathbf{x}}_1^{jT}R^T\widehat{T}^T\hat{e}_3^T\hat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j + \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\hat{e}_3^T\hat{e}_3R^T\widehat{T}^T\tilde{\mathbf{x}}_2^j} \tag{4.19}$$

or

$$\lambda^j = \frac{2\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j}{\tilde{\mathbf{x}}_1^{jT}R^T\widehat{T}^T\hat{e}_3^T\hat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j} = \frac{2\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j}{\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\hat{e}_3^T\hat{e}_3R^T\widehat{T}^T\tilde{\mathbf{x}}_2^j}. \tag{4.20}$$

Substituting (4.18) and (4.19) into F_t, we obtain:

$$F_t(R,T,\tilde{\mathbf{x}}_1^j,\tilde{\mathbf{x}}_2^j) = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j + \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\hat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\hat{e}_3^T\|^2} \tag{4.21}$$

and using (4.18) and (4.20) instead, we get:

$$F_t(R,T,\tilde{\mathbf{x}}_1^j,\tilde{\mathbf{x}}_2^j) \;=\; \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j)^2}{\|\widehat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j\|^2} + \frac{(\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\widehat{e}_3^T\|^2}. \tag{4.22}$$

Geometrically, both expressions for F_t are the distances from the image points $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ to the epipolar lines specified by $\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j$ and (R,T). Equations (4.21) and (4.22) give explicit formulae of the residue of $\|\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j\|^2$ as $\mathbf{x}_1^j, \mathbf{x}_2^j$ being triangulated by $\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j$. Note that the terms in F_t are normalized **crossed epipolar constraints** between $\mathbf{x}_1^j$ and $\tilde{\mathbf{x}}_2^j$ or between $\tilde{\mathbf{x}}_1^j$ and $\mathbf{x}_2^j$. These expressions for F_t can be further used to solve for (R,T) which minimizes F_t. This leads to the following iterative scheme for obtaining optimal estimates of both motion and structure, without explicitly introducing scale factors (or depths) of the 3D points.

Algorithm 4.5 (Optimal Triangulation). *The procedure for minimizing F_t can be outlined as follows:*

1. **Initialization:**
   Initialize $\tilde{\mathbf{x}}_1^j(R,T)$ and $\tilde{\mathbf{x}}_2^j(R,T)$ as $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ respectively.

2. **Motion estimation:**
   Update (R,T) by minimizing $F_t^*(R,T) = F_t(R,T,\tilde{\mathbf{x}}_1^j(R,T),\tilde{\mathbf{x}}_2^j(R,T))$ given by (4.21) or (4.22) as a function defined on the manifold $SO(3)\times\mathbb{S}^2$.

3. **Structure triangulation:**
   Solve for $\tilde{\mathbf{x}}_1^j(R,T)$ and $\tilde{\mathbf{x}}_2^j(R,T)$ which minimize the objective function F_t defined in (4.14) with respect to a fixed (R,T) computed from the previous step.

4. **Return** *to Step 2 until updates are small enough.*

At step 2, $F_t(R,T)$:

$$F_t^*(R,T) = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j + \tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\widehat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^T\widehat{T}R\widehat{e}_3^T\|^2} = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\widehat{T}R\tilde{\mathbf{x}}_1^j)^2}{\|\widehat{e}_3\widehat{T}R\tilde{\mathbf{x}}_1^j\|^2} + \frac{(\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\tilde{\mathbf{x}}_2^{jT}\widehat{T}R\widehat{e}_3^T\|^2} \tag{4.23}$$

is a sum of normalized crossed epipolar constraints. It is a function defined on the manifold $SO(3)\times\mathbb{S}^2$ again hence can be minimized using the Riemannian Newton's algorithm, which is essentially the same as minimizing the normalized epipolar constraint (4.9) studied in the preceding section. The algorithm ends when (R,T) is already a minimum of F_t^*. It can be

shown that if (R,T) is a critical point of F_t^*, then $(R,T,\tilde{\mathbf{x}}_1^j(R,T),\tilde{\mathbf{x}}_2^j(R,T))$ is necessarily a critical point of the original objective function F_t given by (4.14).

At step 3, for a fixed (R,T), $\tilde{\mathbf{x}}_1^j(R,T)$ and $\tilde{\mathbf{x}}_2^j(R,T)$ can be computed by minimizing the distance $\|\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j\|^2 + \|\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j\|^2$ for each pair of image points. Let $t_2^j \in \mathbb{R}^3$ be the normal vector (of unit length) to the (epipolar) plane spanned by $(\tilde{\mathbf{x}}_2^j, T)$. Given such a t_2^j, $\tilde{\mathbf{x}}_1^j$ and $\tilde{\mathbf{x}}_2^j$ are determined by:

$$\tilde{\mathbf{x}}_1^j(t_1^j) = \frac{\widehat{e}_3 t_1^j t_1^{jT} \widehat{e}_3^T \mathbf{x}_1^j + \widehat{t_1^j}^T \widehat{t_1^j} e_3}{e_3^T \widehat{t_1^j}^T \widehat{t_1^j} e_3}, \quad \tilde{\mathbf{x}}_2^j(t_2^j) = \frac{\widehat{e}_3 t_2^j t_2^{jT} \widehat{e}_3^T \mathbf{x}_2^j + \widehat{t_2^j}^T \widehat{t_2^j} e_3}{e_3^T \widehat{t_2^j}^T \widehat{t_2^j} e_3}, \tag{4.24}$$

where $t_1^j = R^T t_2^j \in \mathbb{R}^3$. Then the distance can be explicitly expressed as:

$$\|\tilde{\mathbf{x}}_2^j - \mathbf{x}_2^j\|^2 + \|\tilde{\mathbf{x}}_1^j - \mathbf{x}_1^j\|^2 = \|\mathbf{x}_2^j\|^2 + \frac{t_2^{jT} A^j t_2^j}{t_2^{jT} B^j t_2^j} + \|\mathbf{x}_1^j\|^2 + \frac{t_1^{jT} C^j t_1^j}{t_1^{jT} D^j t_1^j}, \tag{4.25}$$

where $A^j, B^j, C^j, D^j \in \mathbb{R}^{3\times 3}$ are defined by:

$$\begin{aligned} A^j &= I - (\widehat{e}_3 \mathbf{x}_2^j \mathbf{x}_2^{jT} \widehat{e}_3^T + \widehat{\mathbf{x}_2^j}\widehat{e}_3 + \widehat{e}_3\widehat{\mathbf{x}_2^j}), \quad B^j = \widehat{e}_3^T \widehat{e}_3 \\ C^j &= I - (\widehat{e}_3 \mathbf{x}_1^j \mathbf{x}_1^{jT} \widehat{e}_3^T + \widehat{\mathbf{x}_1^j}\widehat{e}_3 + \widehat{e}_3\widehat{\mathbf{x}_1^j}), \quad D^j = \widehat{e}_3^T \widehat{e}_3 \end{aligned}. \tag{4.26}$$

Then the problem of finding $\tilde{\mathbf{x}}_1^j(R,T)$ and $\tilde{\mathbf{x}}_2^j(R,T)$ becomes one of finding t_2^{j*} which minimizes the function of a sum of two **singular Rayleigh quotients**:

$$\min_{t_2^{jT}T=0, t_2^{jT}t_2^j=1} V(t_2^j) = \frac{t_2^{jT} A^j t_2^j}{t_2^{jT} B^j t_2^j} + \frac{t_2^{jT} R C^j R^T t_2^j}{t_2^{jT} R D^j R^T t_2^j}. \tag{4.27}$$

This is an optimization problem on a unit circle $\mathbb{S}^1$ in the plane orthogonal to the vector T (therefore, geometrically, motion and structure recovery from n pairs of image correspondences is an optimization problem on the space $SO(3) \times \mathbb{S}^2 \times \mathbb{T}^n$ where $\mathbb{T}^n$ is an n-torus, *i.e.*, an n-fold product of $\mathbb{S}^1$). If $N_1, N_2 \in \mathbb{R}^3$ are vectors such that T, N_1, N_2 form an orthonormal basis of $\mathbb{R}^3$, then $t_2^j = \cos(\theta) N_1 + \sin(\theta) N_2$ with $\theta \in \mathbb{R}$. We only need to find θ^* which minimizes the function $V(t_2^j(\theta))$. From the geometric interpretation of the optimal solution, we also know that the global minimum θ^* should lie between two values: θ_1 and θ_2 such that $t_2^j(\theta_1)$ and $t_2^j(\theta_2)$ correspond to normal vectors of the two planes spanned by $(\mathbf{x}_2^j, T)$ and $(R\mathbf{x}_1^j, T)$ respectively (if $\mathbf{x}_1^j, \mathbf{x}_2^j$ are already triangulated, these two planes coincide). Therefore, in our approach the local minima is no longer an issue for triangulation, as oppose to the method proposed in [33]. The problem now becomes a simple bounded

minimization problem for a scalar function and can be efficiently solved using standard optimization routines (such as "fmin" in Matlab or the Newton's algorithm). If one properly parameterizes $t_2^j(\theta)$, t_2^{j*} can also be obtained by solving a 6-degree polynomial equation, as shown in [33] (and an approximate version results in solving a 4-degree polynomial equation [131]). However, the method given in [33] involves coordinate transformation for each image pair and the given parameterization is by no means canonical. For example, if one chooses instead the commonly used parameterization of a circle $\mathbb{S}^1$:

$$\sin(2\theta) = \frac{2\lambda}{1+\lambda^2}, \quad \cos(2\theta) = \frac{1-\lambda^2}{1+\lambda^2}, \quad \lambda \in \mathbb{R}, \tag{4.28}$$

then it is straightforward to show from the Rayleigh quotient sum (4.27) that the necessary condition for minima of $V(t^j)$ is equivalent to a 6-degree polynomial equation in λ.[6] The triangulated pairs $(\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j)$ and the camera motion (R, T) obtained from the minimization automatically give a consistent (optimal) 3D structure reconstruction by two-frame stereo.

The optimal triangulation algorithm successfully resolves some mysteries about the epipolar geometry. First, it clarifies the relationship between previously obtained objective functions based on normalization, including F_s and F_g. In the expressions for F_t, if we simply approximate $\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j$ by $\mathbf{x}_1^j, \mathbf{x}_2^j$ respectively, we may obtain the normalized versions of epipolar constraints for recovering camera motion. From (4.21) we get:

$$F_s(R,T) = \sum_{j=1}^{n} \frac{4(\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\widehat{e}_3\widehat{T}R\mathbf{x}_1^j\|^2 + \|\mathbf{x}_2^{jT}\widehat{T}R\widehat{e}_3^T\|^2} \tag{4.29}$$

or from (4.22) we have:

$$F_g(R,T) = \sum_{j=1}^{n} \frac{(\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\widehat{e}_3\widehat{T}R\mathbf{x}_1^j\|^2} + \frac{(\mathbf{x}_2^{jT}\widehat{T}R\mathbf{x}_1^j)^2}{\|\mathbf{x}_2^{jT}\widehat{T}R\widehat{e}_3^T\|^2} \tag{4.30}$$

The first function (divided by 4) is exactly the same as the statistically normalized objective function F_s introduced in the preceding section; and the second one is exactly the geometrically normalized objective function F_g. From the above derivation, we see that there is essentially no difference between these two objective functions – they only differ by a second order term in terms of $\mathbf{x}_1^j - \tilde{\mathbf{x}}_1^j$ and $\mathbf{x}_2^j - \tilde{\mathbf{x}}_2^j$. Although such subtle differences between F_s, F_g and F_t have previously been pointed out in [140], our approach discovers that all these three objective functions can be unified in the same optimization procedure – they are just

[6] Since there is no closed form solution to 6-degree polynomial equations, directly minimizing the Rayleigh quotient sum (4.27) avoids unnecessary transformations hence can be much more efficient.

slightly different approximations of the same objective function F_t^*. Practically speaking, using either normalized objective function F_s or F_g, one can already get camera motion estimates which are very close to the optimal ones.

Secondly, as we noticed, the epipolar constraint type objective function F_t^* given by (4.23) appears as a key intermediate objective function in an approach which initially intends to minimize the so called reprojection error given by (4.14). The approach of minimizing reprojection error was previously considered in the computer vision literature as an alternative to methods which directly minimize epipolar constraints [33, 129]. We here see that they are in fact profoundly related. Further, the crossed epipolar constraint F_t^* given by (4.23) for motion estimation and the sum of singular Rayleigh quotients $V(t_2^j)$ given by (4.27) for triangulation are simply different expressions for the reprojection error under different conditions. In summary, "minimizing (normalized) epipolar constraints" [63, 140], "triangulation" [33] and "minimizing reprojection errors" [129] are all deeply related to each other. They are in fact different (approximate) versions of the same procedure for obtaining *the* optimal motion and structure estimates from image correspondences.

4.3 Critical Values and Ambiguous Solutions

Note that all objective functions F, F_s, F_g and F_t^* that we have encountered are even functions of $S \in \mathbb{S}^2$.[7] We can then view them as functions on the manifold $SO(3) \times \mathbb{RP}^2$ instead of $SO(3) \times \mathbb{S}^2$, where $\mathbb{RP}^2$ is the two dimensional real projective plane. This objective function could have numerous critical points, such as (local) **minima**, (local) **maxima**, and **saddles**. Since the Euler characteristic of the manifold $SO(3) \times \mathbb{RP}^2$ is 0, any (Morse) function defined on it must have all three kinds of critical values. The nonlinear search algorithms proposed in the above are trying to find the global minimum of given objective functions. The search process, if not properly initialized, may stop at any of the types of critical points listed above, especially the local minima.[8] Moreover, like any nonlinear system, when increasing the noise level, new critical points can be introduced through bifurcation (see [93]). An example of bifurcation is shown in Figure 4.1. Although, in general, many different types of bifurcations may occur when increasing the noise level, the **fold bifurcation** illustrated in Figure 4.1 occurs most frequently in the motion and

[7] A even function $f(S)$ on $\mathbb{S}^2$ satisfies $f(-S) = f(S)$.

[8] Maxima and saddles have a at least one dimensional unstable submanifold hence the Newton's algorithm rarely ends at these points.

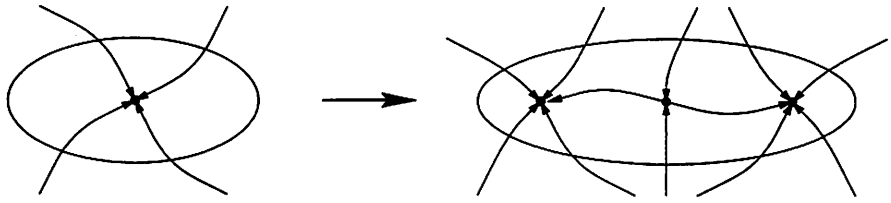


Figure 4.1: Bifurcation which preserves the Euler characteristic by introducing a pair of saddles and a node. The index of the circled regions is 1.

structure estimation problem. We therefore need to understand how such a bifurcation may occur and how it affects the motion estimates.

Since the nonlinear search schemes are usually initialized by the linear algorithm, not all the local minima are equally likely to be reached by the proposed algorithms. From the preceding section, we know all objective functions are more or less equivalent to the epipolar constraints, especially when the translation is parallel to the image plane. If we let $E = \widehat{T}R$ to be the essential matrix, then we can rewrite the epipolar constraint as $\mathbf{x}_2^{jT} E \mathbf{x}_1^j = 0, j = 1, \ldots, n$. Then minimizing the objective function F is (approximately) equivalent to the following least square problem:

$$\min \|Ae\|^2 \tag{4.31}$$

where A is a $n \times 9$ matrix function of entries of $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$, and $e \in \mathbb{R}^9$ is a vector of the nine entries of E. Then e is the (usually one dimensional) null space of the 9×9 symmetric matrix $A^T A$. In the presence of noise, e is simply chosen to be the eigenvector corresponding to the least eigenvalue of $A^T A$. At a low noise level, this eigenvector in general gives a good initial estimate of the essential matrix.[9] However, at a certain high noise level, the smallest two eigenvalues may switch roles, as do the two corresponding eigenvectors – topologically, a bifurcation as shown in Figure 4.1 occurs. Let us denote these two eigenvectors as e and e'. Since they both are eigenvectors of the symmetric matrix $A^T A$, they must be orthogonal to each other, *i.e.*, $e^T e' = 0$. In terms of matrix notation, we have $tr(E^T E') = 0$. For the motions recovered from E and E' respectively, we have $tr(R^T \widehat{T}^T \widehat{T}' R') = 0$. It is well known that the rotation estimate R is usually much less sensitive to noise than the translation estimates S. Therefore, approximately, we have $R \approx R'$ hence $tr(\widehat{T}^T \widehat{T}') \approx 0$. That is, T and T' are almost orthogonal to each other. This phenomenon is very common in the motion estimation problem: at a high noise level, the translation estimate may suddenly

[9]Such estimate might be biased towards the bas relief ambiguity.

change direction by roughly 90^o, especially in the case when translation is parallel to the image plane. We will refer to such estimates as the **second eigenmotion**. Similar to detecting local minima in the continuous case (see [98]), the second eigenmotion ambiguity can usually be detected by checking the positive depth constraints. A similar situation of the 90^o flip in the motion estimates for the continuous case has previously been reported in [15].

Figure 4.2 and 4.3 demonstrate such a sudden appearance of the second eigenmotion. They are the simulation results of the proposed nonlinear algorithm of minimizing the function F_s for a cloud of 40 randomly generated pairs of image correspondences (in a field of view 90^o, depth varying from 100 to 400 units of focal length.). Gaussian noise of standard deviation of 6.4 or 6.5 pixels is added on each image point (image size 512×512 pixels). To make the results comparable, we used the same random seeds for both runs. The actual rotation is 10^o about the Y-axis and the actual translation is along the X-axis.[10] The ratio between translation and rotation is 2.[11] In the figures, "+" marks the actual translation, "$*$" marks the translation estimate from linear algorithm (see [76] for detail) and "o" marks the estimate from nonlinear optimization. Up to the noise level of 6.4 pixels, both rotation and translation estimates are very close to the actual motion. Increasing the noise level further by 0.1 pixel, the translation estimate suddenly switches to one which is roughly 90^o away from the actual translation. Geometrically, this estimate corresponds to the second smallest eigenvector of the matrix A^TA as we discussed before. Topologically, this estimate corresponds to the local minimum introduced by a bifurcation as shown by Figure 4.1. Clearly, in Figure 4.2, there are 2 maxima, 2 saddles and 1 minima on $\mathbb{RP}^2$; in Figure 4.3, there are 2 maxima, 3 saddles and 2 minima. Both patterns give the Euler characteristic of $\mathbb{RP}^2$ as 1.

From the Figure 4.3, we can see that the the second eigenmotion ambiguity is even more likely to occur (at certain high noise level) than the other local minimum marked by "$\diamond$" in the figure which is a legitimate estimate of the actual one. These two estimates always occur in pair and exist for general configuration even when both the FOV and depth variation are sufficiently large. We propose a way for resolving the second eigenmotion ambiguity already by linear algorithm which is used for initialization. An indicator of the

[10] We here use the convention that Y-axis is the vertical direction of the image and X-axis is the horizontal direction and the Z-axis coincides with the optical axis of the camera.

[11] Rotation and translation magnitudes are compared with respect to the center of the cloud of 3D points generated.

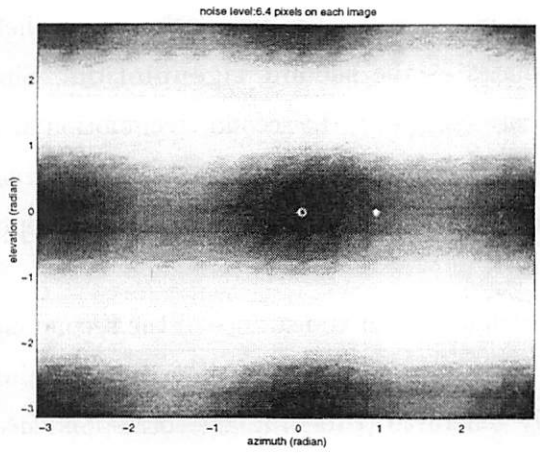


Figure 4.2: Value of objective function F_s for all T at noise level 6.4 pixels (rotation fixed at the estimate from the nonlinear optimization). Estimation errors: 0.014 in rotation estimate (in terms of the canonical metric on $SO(3)$) and 2.39^o in translation estimate (in terms of angle).

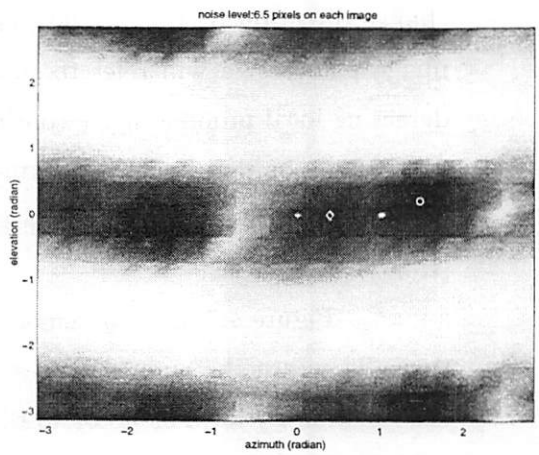


Figure 4.3: Value of objective function F_s for all T at noise level 6.5 pixels (rotation fixed at the estimate from the nonlinear optimization). Estimation errors: 0.227 in rotation estimate (in terms of the canonical metric on $SO(3)$) and 84.66^o in translation estimate (in terms of angle).

configuration being close to critical is the ratio of the two smallest eigenvalues of $A^T A$ σ_9 and σ_8 . By using both eigenvectors v_9 and v_8 for computing the linear motion estimates and choosing the one which satisfies the positive depth constraint by larger margin (i.e. larger number of points satisfies the positive depth constraint) leads to the motion estimates closer to the true one. The motion estimate (R, T) which satisfies the positive depth constraint should make the following inner product greater then 0 for all the corresponding points.

$$(\widehat{T}\mathbf{x}_1^j)^T(\widehat{\mathbf{x}_1^j}R^T\mathbf{x}_2^j) > 0 \tag{4.32}$$

While for low noise level all the points satisfy the positive depth constraint, with increasing noise level some of the points fail to satisfy it. We therefore chose the solution where majority of points satisfies the positive depth constraint. Simple re-initialization then guarantees convergence of the nonlinear techniques to the true solution. Figures 4.4 and 4.5 depict a slice of the objective function for varying translation and for the rotation estimate obtained by linear algorithm using v_9 and v_8 as two different estimates of the essential matrix.

This second eigenmotion effect has a quite different interpretation as the one which was previously attributed to the bas relief ambiguity. The bas relief effect is only evident when FOV and depth variation is small, but the second eigenmotion ambiguity may show

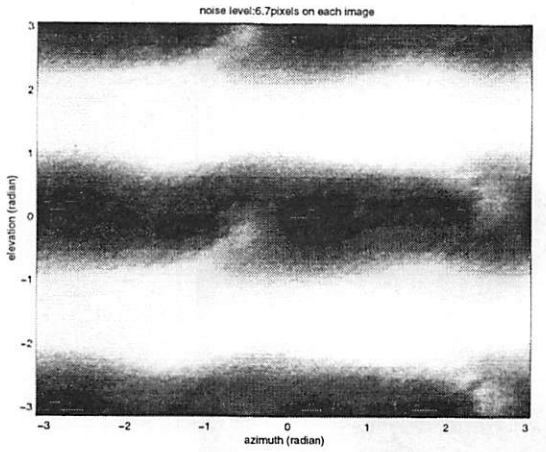


Figure 4.4: Value of objective function F_s for all T at noise level 6.7 pixels. Rotation is fixed at the estimate from the linear algorithm from the eigenvector v_9 associated with the smallest eigenvalue. Note the verge of the bifurcation of the objective function.

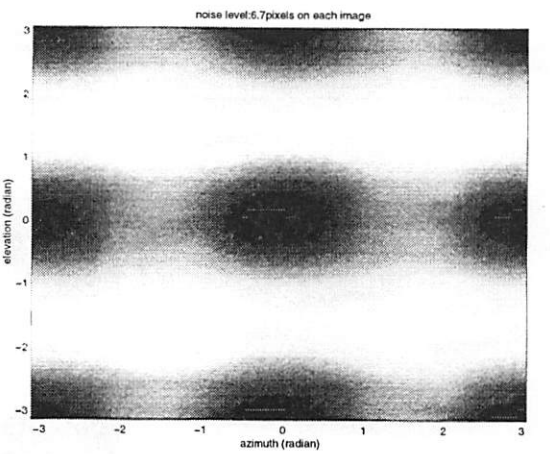


Figure 4.5: Value of objective function F_s for all T at noise level 6.7 pixels. Rotation is fixed at the estimate from the linear algorithm from the eigenvector v_8 associated with the second smallest eigenvalue. The objective function is well shaped and the nonlinear algorithm refined the linear estimate closer to the true solution.

up for general configurations. Bas relief estimates are statistically meaningful since they characterize a sensitive direction in which translation and rotation are the most likely to be confound. The second eigenmotion, however, is not statistically meaningful: it is an artifact introduced by a bifurcation of the objective function; it occurs only at a high noise level and this critical noise level gives a measure of the **robustness** of the given algorithm. For comparison, Figure 4.6 demonstrates the effect of the bas relief ambiguity: the long narrow valley of the objective function corresponds to the direction that is the most sensitive to noise.[12] The (translation) estimates of 20 runs, marked as "o", give a distribution roughly resembling the shape of this valley – the actual translation is marked as "+"in the center of the valley which is covered by circles.

4.4 Experiments and Sensitivity Analysis

In this section, we clearly demonstrate by experiments the relationship among the linear algorithm (as in [76]), nonlinear algorithm (minimizing F), normalized nonlinear

[12]This direction is given by the eigenvector of the Hessian associated with the smallest eigenvalue.

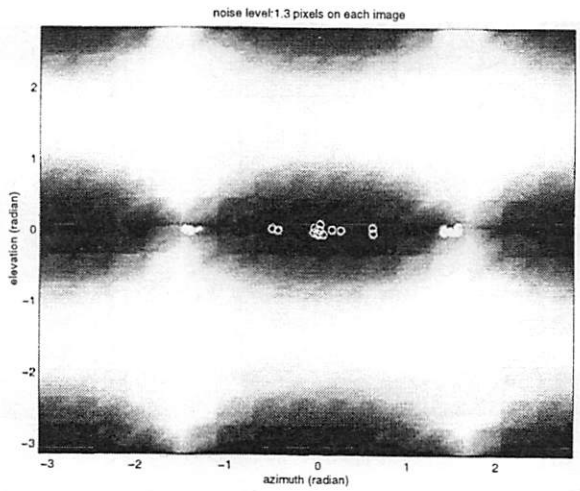


Figure 4.6: Bas relief ambiguity. FOV is 20^o and the random cloud depth varies from 100 to 150 units of focal length. Translation is along the X-axis and rotation around the Y-axis. Rotation magnitude is 2^o. T/R ratio is 2. 20 runs at the noise level 1.3 pixels.

algorithm (minimizing F_s) and optimal triangulation (minimizing F_t). Due to the nature of the second eigenmotion ambiguity, it gives statistically meaningless estimates. Such estimates should be treated as "outliers" if one wants to properly evaluate a given algorithm and compare simulation results. In order for all the simulation results to be statistically meaningful and comparable to each other, in following simulations, we usually keep the noise level below the critical level at which the second eigenmotion ambiguity occurs unless we need to comment on its effect on the evaluation of algorithms' performance.

We follow the same line of thought as the analysis of the continuous case in [98]. We will demonstrate by simulations that seemingly conflicting statements in the literature about the performance of existing algorithms can in fact be given a *unified* explanation if we systematically compare the simulation results with respect to a *large range* of noise levels (as long as the results are statistically meaningful). Some existing evaluations of the algorithms turn out to be valid only for a certain small range of signal-to-noise ratio. In particular, algorithms' behaviors at very high noise levels have not yet been well understood or explained. Since, for a fixed noise level, changing base line is equivalent to changing the signal-to-noise ratio, we hence perform the simulations at a fixed base line but the noise level varies from very low (< 1 pixels) to very high (tens of pixels for a typical image size of 512×512 pixels). The conclusions therefore hold for a large range of base line. In particular, we emphasize that some of the statements given below are valid for the continuous case as

well.

In following simulations, for each trial, a random cloud of 40 3D points is generated in a region of truncated pyramid with a field of view (FOV) 90^o, and a depth variation from 100 to 400 units of the focal length. Noises added to the image points are i.i.d. 2D Gaussian with standard deviation of the given noise level (in pixels). Magnitudes of translation and rotation are compared at the center of random cloud. This will be denoted as the translation-to-rotation ratio, or simply the T/R ratio. The algorithms will be evaluated for different combinations of translation and rotation directions. We here use the convention that Y-axis is the vertical direction of the image and X-axis is the horizontal direction and the Z-axis coincides with the optical axis of the camera. All nonlinear algorithms are initialized by the estimates from the standard 8-point linear algorithm (see [76]), instead of from the ground truth.[13] The criteria for all nonlinear algorithms to stop are: 1. The norm of gradient is less than a given error tolerance, which usually we pick as 10^{-8} unless otherwise stated;[14] and 2. The smallest eigenvalue of the Hessian matrix is positive.[15]

4.4.1 Axis Dependency Profile

It has been well known that the sensitivity of the motion estimation depends on the camera motion. However, in order to give a clear account of such a dependency, one has to be careful about two things: 1. The signal-to-noise ratio and 2. Whether the simulation results are still statistically meaningful while varying the noise level.

Figure 4.7, 4.8, 4.9 and 4.10 give simulation results of 100 trials for each combination of translation and rotation ("T-R") axes, for example, "X-Y" means translation is along the X-axis and the rotation axis is the Y-axis. Rotation is always 10^o about the axis and the T/R ratio is 2. In the figures, "linear" stands for the standard 8-point linear algorithm; "nonlin" is the Riemannian Newton's algorithm minimizing the epipolar constraints F, "normal" is the Riemannian Newton's algorithm minimizing the normalized epipolar constraints F_s.

By carefully comparing the simulation results in Figure 4.7, 4.8, 4.9 and 4.10, we

[13]We like to point out that evaluation based on initializing from the ground truth is misleading for using these algorithms in real applications since it usually does not reveal correctly the relationship between the linear algorithm and nonlinear algorithms.

[14]Our current implementation of the algorithms in Matlab has a numerical accuracy at 10^{-8}.

[15]Since we have the explicit formulae for Hessian, this condition would keep the algorithms from stopping at saddle points.

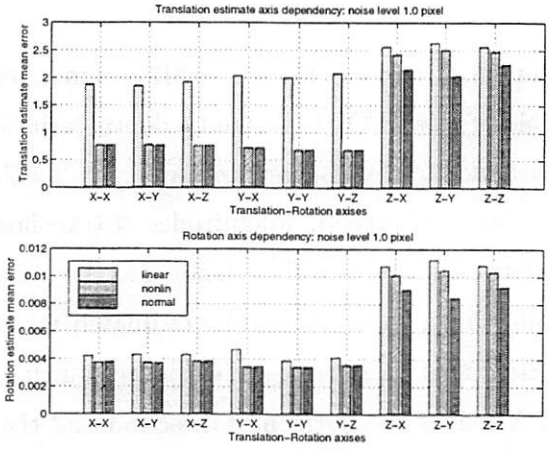


Figure 4.7: Axis dependency: estimation errors in rotation and translation at noise level 1.0 pixel. T/R ratio = 2 and rotation = 10^o.

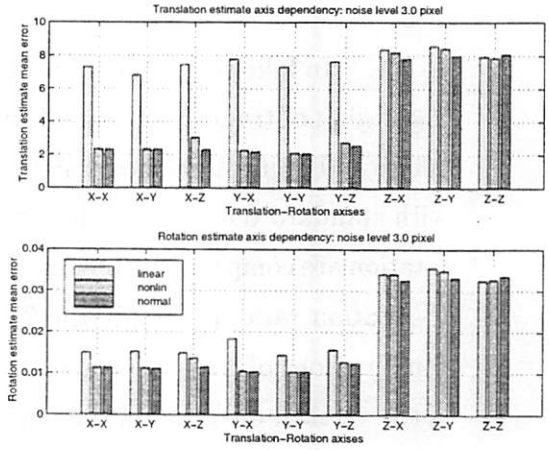


Figure 4.8: Axis dependency: estimation errors in rotation and translation at noise level 3.0 pixels. T/R ratio = 2 and rotation = 10^o.

can draw the following conclusions:

- **Optimization Techniques (linear vs. nonlinear)**

  1. Minimizing F in general gives better estimates than the linear algorithm at low noise levels (Figure 4.7 and 4.8). At higher noise levels, this is no longer true (Figure 4.9 and 4.10), due to the more global nature of the linear technique.
  2. Minimizing the normalized F_s in general gives better estimates than the linear algorithm at moderate noise levels (all figures). Very high noise level case will be studied in the next section.

- **Optimization Criteria (F vs. F_s)**

  1. At relatively low noise levels (Figure 4.7), normalization has little effect when translation is parallel to the image plane; and estimates are indeed improved when translation is along the Z-axis.
  2. However, at moderate noise levels (Figure 4.8, 4.9 and 4.10), things are quite the opposite: when translation is along the Z-axis, little improvement can be gained by minimizing F_s instead of F since estimates are less sensitive to noise in this case (in fact all three algorithms perform very close); however, when translation

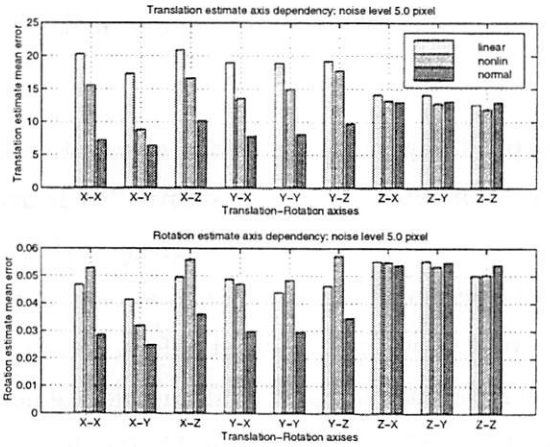


Figure 4.9: Axis dependency: estimation errors in rotation and translation at noise level 5.0 pixel. T/R ratio = 2 and rotation = 10^o.

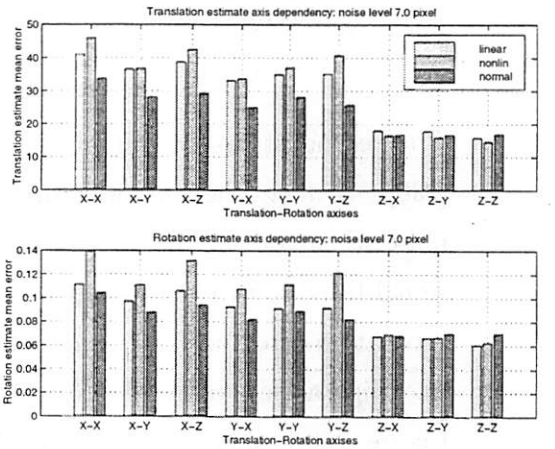


Figure 4.10: Axis dependency: estimation errors in rotation and translation at noise level 7.0 pixels. T/R ratio = 2 and rotation = 10^o.

is parallel to the image plane, F is more sensitive to noise and minimizing the statistically less biased F_s consistently improves the estimates.

- **Axis Dependency (translation parallel to image plane vs. along Z-axis)**

  1. All three algorithms are the most robust to the increasing of noise when the translation is along Z. At moderate noise levels (all figures), their performances are quite close to each other.

  2. Although, at relatively low noise levels (Figure 4.7, 4.8 and 4.9), estimation errors seem to be larger when the translation is along the Z-axis, estimates are in fact much less sensitive to noise and more robust to increasing of noise in this case. The larger estimation error in case of translation along Z-axis is because the displacements of image points are smaller than those when translation is parallel to the image plane. Thus, with respect to the same noise level, the signal-to-noise ratio is in fact smaller in the case of translating along the Z-axis.

  3. At a noise level of 7 pixels (Figure 4.10), estimation errors seem to become smaller when the translation is along Z-axis. This is not only because, estimates are less sensitive to noise for this case, but also due to the fact that, at a noise level of 7 pixels, the second eigenmotion ambiguity already occurs in some of the trials when the translation is parallel to the image plane. Outliers given by the

second eigenmotion are averaged in the estimation errors and make them look even worse.

The second statement about the axis dependency supplements the observation given in [130]. In fact, the motion estimates are both robust and less sensitive to increasing of noise when translation is along the Z-axis. Due to the exact reason given in [130], smaller signal-to-noise ratio in this case makes the effect of robustness not to appear in the mean estimation error until at a higher noise level. As we have claimed before, for a fixed base line, high noise level results resemble those for a smaller base line at a moderate noise level. Figure 4.10 is therefore a generic picture of the axis dependency profile for the continuous or small base-line case (for more details see [68]).

4.4.2 Non-iterative vs. Iterative

In general, the motion estimates obtained from directly minimizing the normalized epipolar constraints F_s or F_g are already very close to the solution of the optimal triangulation obtained by minimizing F_t iteratively between motion and structure. It is already known that, at low noise levels, the estimates from the non-iterative and iterative schemes usually differ by less than a couple of percent [140]. This is demonstrated in Figure 4.11 and 4.12 – "linear" stands for the linear algorithm; "norm nonlin" for the Riemannian Newton's algorithm minimizing normalized epipolar constraint F_s; "triangulate" for the iterative optimal triangulation algorithm. For the noise level from 0.5 to 5 pixels, at the error tolerance 10^{-6}, the iterative scheme has little improvement over the non-iterative scheme – the two simulation curves overlap with each other. Simulation results given in Figure 4.13 and 4.14 further show that the improvements of the iterative scheme become a little bit more evident when noise levels are very high, but still very slim. Due to the second eigenmotion ambiguity, we can only perform high noise level simulation properly for the case when the translation direction is along the Z-axis.

By comparing the simulation results in Figures 4.11, 4.12, 4.13 and 4.14, we can therefore draw the following conclusions:

- Although the iterative optimal triangulation algorithm usually gives better estimates (as it should), the non-iterative minimization of the normalized epipolar constraints F_s or F_g gives motion estimates with only a few percent larger errors for all range

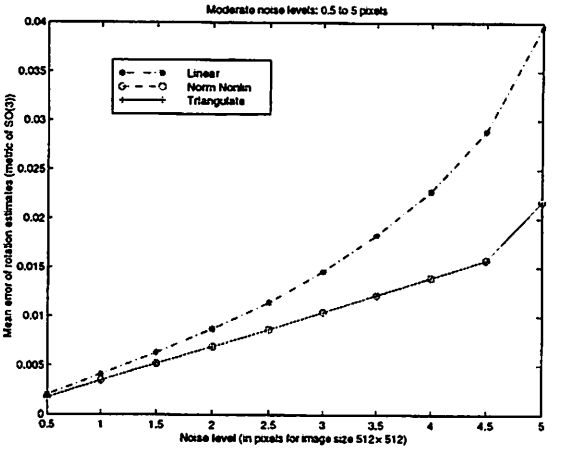


Figure 4.11: Estimation errors of rotation (in canonical metric on $SO(3)$). 50 trials, rotation 10 degree around Y-axis and translation along X-axis, T/R ratio is 2. Noises range from 0.5 to 5 pixels.

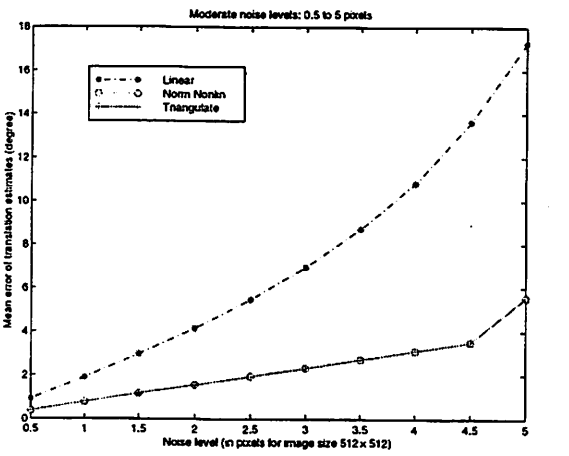


Figure 4.12: Estimation errors of translation (in degree). 50 trials, rotation 10 degree around Y-axis and translation along X-axis, T/R ratio is 2. Noises range from 0.5 to 5 pixels.

of noise levels. The higher the noise level, the more evident the improvement of the iterative scheme is.

- Within moderate noise levels, normalized nonlinear algorithms consistently give significantly better estimates than the standard linear algorithm, especially when the translation is parallel to the image plane. At very high noise levels, the performance of the standard linear algorithm, out performs nonlinear algorithms. This is due to the more global nature of the linear algorithm. However, such high noise levels are barely realistic in real applications.

For low level Gaussian noises, the iterative optimal triangulation algorithm gives the MAP estimates of the camera motion and scene structure, the estimation error can be shown close to the theoretical error bounds, such as the Cramer-Rao bound. This has been shown experimentally in [131]. Consequently, minimizing the normalized epipolar constraints F_s or F_g gives motion estimates close to the error bound as well. At very high noise levels, linear algorithm is certainly more robust and gives better estimates. Due to numerous local minima, running nonlinear algorithms to update the estimate of the linear algorithm does not necessarily reduce the estimation error further.

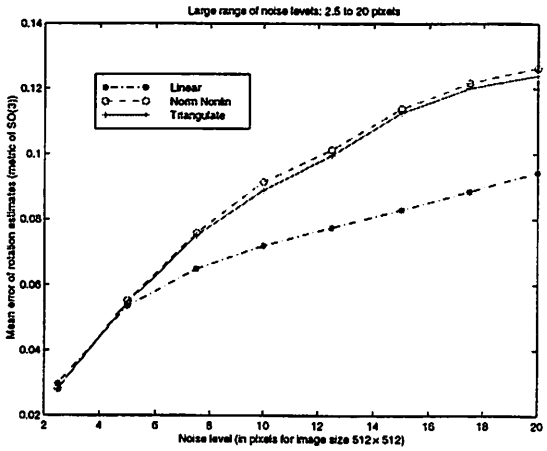


Figure 4.13: Estimation errors of rotation (in canonical metric on $SO(3)$). 40 points, 50 trials, rotation 10 degree around Y-axis and translation along Z-axis, T/R ratio is 2. Noises range from 2.5 to 20 pixels.

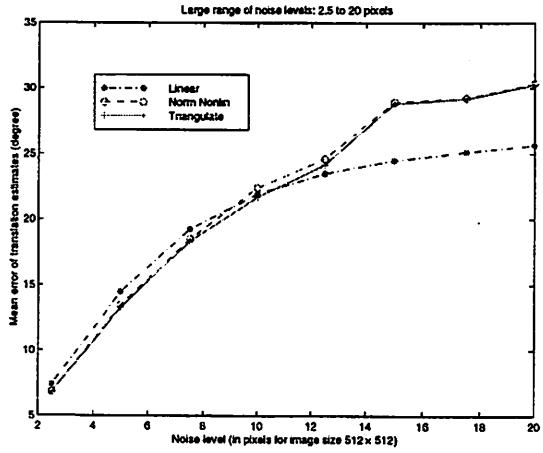


Figure 4.14: Estimation errors of translation (in degree). 40 points, 50 trials, rotation 10 degree around Y-axis and translation along Z-axis, T/R ratio is 2. Noises range from 2.5 to 20 pixels.

4.4.3 Mutual Information Between Structure Estimates and Noises

So far, we have understood some of the difficulties in motion and structure estimation caused by various ambiguities, such as the bas relief ambiguity which is related to the sensitivity issue, or the second eigenmotion ambiguity which is related to the robustness issue. We here like to address, from an information theoretic viewpoint, another difficulty caused by noise in motion and structure estimation. More specifically, we like to ask the following questions:

> Is the (2-frame) motion and structure recovery problem well-defined from an estimation theoretic viewpoint?[16] If not, how much information can still be preserved in the presence of noise? Consequently, is there any simple criteria that a "good" estimation algorithm should achieve?

The answer to the first question is unfortunately negative due to following reasons. Let us assume the same noise model as given by (4.8).[17] As shown in Figure 4.15, given the noisy $\mathbf{x} = \mathbf{x}_0 + \alpha$ where α is any isotropic noise on the image plane. Then the valid estimate of $\mathbf{x}_0$ is given by $\tilde{\mathbf{x}}$, the projection of $\mathbf{x}$ onto the epipolar line. Therefore, the component of α which is parallel to the epipolar line is absorbed into the estimates. Without loss of

[16] It is certainly well defined geometrically: in the noise free case, the linear algorithm gives closed-from solutions.

[17] The Gaussian assumption is not necessary here. The following arguments hold for all isotropic noises.

generality, we assume the variance of the noise α is 1.[18] Then the variance left in the residue $\Delta\mathbf{x} = \mathbf{x} - \tilde{\mathbf{x}}$ is about 0.5. In other words, regardless of algorithms, at least half of the noise will always become part of the estimated 3D structure. Consequently, any good (2-frame) motion and structure estimation algorithm should have a residue variance (relative to the noise variance) close to 0.5. This is a very simple and important statistic for evaluating any structure and motion estimation algorithm. For the proposed optimal triangulation algorithm, we computed the average residue variance for all the runs which are presented in Figure 4.11 and 4.12. It gives 0.4988, very close to the theoretical value.

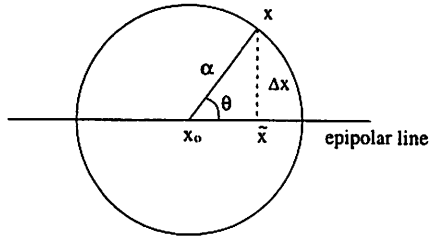


Figure 4.15: Estimated $\tilde{\mathbf{x}}$ from noisy $\mathbf{x}$.

4.5 Discussion

The motion and structure recovery problem has been studied extensively and many researchers have proposed efficient nonlinear optimization algorithms. One may find historical reviews of these algorithms in [53, 76]. Although these algorithms already have good performance in practice, the geometric concepts behind them have not yet been completely revealed. The non-degeneracy conditions and convergence speed of those algorithms are usually not explicitly addressed. Due to the recent development of optimization methods on Riemannian manifolds, we now can have a better mathematical understanding of these algorithms, and propose new geometric algorithms or filters (for example, following [99]), which exploit the intrinsic geometric structure of the motion and structure recovery problem. As shown in this chapter, regardless of the choice of different objectives, the problem of optimization on the essential manifold is common and essential to the optimal motion and structure recovery problem. Furthermore, from a pure optimization theoretic viewpoint, most of the objective functions previously used in the literature can be unified in a sin-

[18] Note $\mathbf{x}$ is a vector, so here we mean the expectation $E(\|\alpha\|^2) = 1$ where $\|\cdot\|$ is the 2-norm.

gle optimization procedure. Consequently, "minimizing (normalized) epipolar constraints", "triangulation", "minimizing reprojection errors" are all different (approximate) versions of the same simple optimal triangulation algorithm.

We have applied only Newton's algorithm to the motion and structure recovery problem since it has the fastest convergence rate (among algorithms using second order information, see [19] for the comparison). In fact, the application of other conjugate gradient algorithms would be easier since they usually only involve calculation of the first order information (the gradient, not Hessian), at the cost of a slower convergence rate. Like most iterative search algorithms, Newton's and conjugate gradient algorithms are local methods, *i.e.*, they do not guarantee convergence to the global minimum. Due to the fundamental relationship between the motion recovery objective functions and the epipolar constraints discovered in this chapter, at high noise levels all the algorithms unavoidably will suffer from the second eigenmotion (except the case when translation is along the Z-axis). Such an ambiguity is intrinsic to the problem of motion and structure recovery and independent of the choice of objective functions.

In this chapter, we have studied in detail the problem of recovering a discrete motion (displacement) from image correspondences. Similar ideas certainly apply to the continuous case where the rotation and translation are replaced by angular and linear velocities respectively (as the linear case in Chapter 3). Optimization schemes for the continuous case have also been studied by many researchers, including the most recent Bilinear Projection Algorithm (BPA) proposed in [98] and a robust algorithm proposed in [139]. Similarly, one can show that they all in fact minimize certain normalized versions of the continuous epipolar constraint. We hope the Riemannian optimization theoretic viewpoint proposed in this chapter have provided people a different perspective.

Although the study of the proposed algorithms is carried out in a calibrated camera framework, the same approach and optimization schemes can be generalized with little effort to the uncalibrated case as well. As we pointed out in this chapter, Riemannian optimization algorithms can be easily generalized to products of manifolds. Thus, although the proposed Newton's algorithm is for two views and a single rigid body motion, it can be easily generalized to multiview and multi-body cases. This is to be shown in the next chapter where motion (and structure) recovery from multiple images is studied.

Chapter 5

Motion and Structure from Multiple Images

"Algebra is but written geometry; geometry is but drawn algebra."
— Sophie Germain

In this chapter, we study the classic problem in structure from motion: *How to recover camera motion and (Euclidean) scene structure from correspondences of a cloud of points seen in* **multiple** *(perspective) images?* With such a vast body of literature studying almost every aspect of this problem (see, for example, reviews of batch methods [106], recursive methods [79, 99], orthographic case [111] and projective reconstruction [114]), it is quite reasonable to ask what, if anything, can still be new in this topic.

First of all, we want to have a clear picture about the relationship among multiple images. While constraints involving two images at a time (epipolar constraints) have been well understood from previous chapters and involve clean notation and geometric interpretation, constraints among multiple images are more difficult to work with and to interpret. On our way to develop algorithms, we then first pause to reflect on the nature of these constraints. It seems therefore natural to ask the following question:

> (i) *Do constraints among multiple images carry information that is not contained in the epipolar ones?*

The nature of the constraints among images of the same point in different images has been studied extensively, and is known to be multilinear (see for instance [25, 44, 117]). These

multilinear constraints turn out to be reducible to three fundamental types: bilinear, trilinear and quadrilinear constraints, named after the number of images that they respectively involve. The bilinear constraint is exactly the epipolar constraints between two images. Further **algebraic dependency** among these three types of constraints has been established by means of elimination [132] or algebraic geometry tools [43]. However, an explicit characterization of how the information is encoded in different constraints - which is crucial in the design of robust estimation algorithms - is hard to derive by such means. In this chapter, we will provide an more geometric way to study multilinear constraints. Especially the **geometric dependency** among multilinear constraints will be introduced and clearly studied.

After we have understood well the algebraic and geometric relationship among multilinear constraints [43, 74, 95, 113] (which will be briefly reviewed in Section 5.2.1), when it comes to using them for designing motion or structure recovery algorithms, they are usually used as *objectives* rather than, *constraints*. Many researchers believe that multilinear tensors should be directly computed in their natural linear form [34]. Algebraically, this is true. Nevertheless, when a noise model is considered and the direct objective is to minimize certain statistics, such as the **reprojection error** (also called **nonlinear least squares error** as in [106]), it becomes quite unclear how to incorporate these multilinear constraints into the objective, or how to obtain less biased estimates of these tensors. More specifically, we want to answer the questions:

> (ii) *Can we convert such a constrained estimation (or optimization) problem to an unconstrained one? If so, what weight should be assigned to each constraint?*

As we will soon see, proper weighting usually ends up with nonlinear constraints, instead of linear.

Secondly, we have every reason to believe that, for such a constrained estimation problem, its *a posteriori* likelihood function (or some variation of it) still needs to be found. From an estimation theoretic viewpoint, such a function should indeed capture some peculiar statistical nature of the multiview structure from motion problem. Other than the well known algebraic and geometric relationship between bilinear and trilinear constraints, we may ask:

> (iii) *What is the* **statistical relationship** *between bilinear and trilinear constraints? Is trilinear constraint really needed for motion (or structure) estimation in the degenerate rectilinear motion case?*

On the other hand, from an optimization theoretic viewpoint, with such a function we can further understand:

> **(iv)** *What is the exact nature of the* **optimization** *associated to the original problem? What geometric space does the optimization take place on? Is there any generic optimization technique available for minimizing such a function?*

Finally, in applications which require high accuracy, noise sensitivity becomes the primary issue [14, 73, 139]. Although a specific sensitivity study is needed for every algorithm, it is still possible to study the *intrinsic* sensitivity inherent in the initial problem. From statistics, we know that the Hessian of the *a posteriori* likelihood function at the maximum closely approximates the covariance matrix of the estimates. Hence an *explicit* expression for the likelihood function is absolutely necessary for a systematic study of the intrinsic sensitivity issue. As we will soon see, the normalized epipolar constraint to be derived is such a function and we will show how to compute its Hessian, even though the sensitivity issue is not a main subject of this chapter (see Section 5.2.3).

Chapter Outline

In this chapter, we will give clear answers to the above questions through the development of a solution to the constrained nonlinear least squares optimization problem which minimizes the reprojection error subject to constraints among multiple images. Question **(i)** is answered in Section 5.1 where a clean expression for all the multilinear constraints is given. Also the concept of geometric dependency is introduced and compared with the algebraic one. Question set **(ii)** will be answered in Section 4.1.2. The answers will become evident from the derivation and the form of the normalized epipolar constraint. For Question set **(iii)**, the statistical relationship between bilinear and trilinear constraints will be revealed by Simulation 3 in Section 5.2.4 and some further explanation will be given in Comment 5.10. Question set **(iv)** are to be answered in Section 5.2.3 where a generic optimization algorithm is explicitly laid out for minimizing the normalized epipolar constraint. Although our results, including the algorithm, can be easily generalized to trilinear constraints or even to an uncalibrated framework, we choose to present the calibrated case using bilinear (epipolar) constraints so as to clearly convey the main ideas. Nevertheless, we will comment on the trilinear case and uncalibrated case in due time. In Section 5.3, an extension to continuous or hybrid settings is briefly discussed.

Relation to Previous Work: The algorithm to be proposed belongs to the so called **batch methods** for motion and structure recovery from multiple views, like that of [106, 111, 114], and is a necessary extension to the unconstrained nonlinear least squares method [106]. We here emphasize again that, our focus is *not* on an algorithm for computing motion or structure faster than the ones in [86, 139], although we will mention briefly how to speed up our algorithm. Instead, we are using our algorithm as a means of revealing the interesting *geometry* in multiview structure from motion, by way of identifying it with the *optimality* of each step of the algorithm. In doing so, one will be able to see what roles multilinear constraints essentially play in the design of optimal algorithms. Especially, the revelation of the statistical relationship between bilinear and trilinear constraints is an important complement to the well known algebraic or geometric results [43, 74, 95, 113]. Our results, especially the normalized epipolar constraint, may also help improve existing *recursive* methods such as in [79, 99] if the filter objective function is modified to the one given by us. Moreover, studying the Hessian of such an objective will allow an extension of existing sensitivity study [14, 73] to the multiview case.

5.1 Dependency of Multilinear Constraints

As before, we model the world as a collection of points in a three-dimensional Euclidean space. We denote the homogeneous coordinates of a point $p \in \mathbb{E}^3$ with respect to some inertial coordinate frame (as if the time is t_0) as $p = p(t_0) = [X_1, X_2, X_3, 1]^T \in \mathbb{R}^4$. The perspective projection of p onto a two-dimensional image plane is represented by homogeneous coordinate $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^3$. According to (2.17), it satisfies:

$$\lambda(t)\mathbf{x}(t) = A(t)Pg(t)p, \quad t \in \mathbb{R} \tag{5.1}$$

where $\lambda(t) \in \mathbb{R}$ is a scalar parameter related to the distance of the point p from the center of projection and the non-singular matrix $A(t)$ - called "calibration matrix" - describes the intrinsic parameters of the camera. We for now assume the most general case that the camera calibration may be time-varying. Without loss of generality we will re-scale the above equation so that the determinant of $A(t)$ is 1. The set of 3×3 matrices with determinant one is called special linear group denoted by $SL(3)$. The rigid motion of the camera $g(t)$ is represented by a translation vector $T(t) \in \mathbb{R}^3$ and a rotation matrix $R(t) \in SO(3)$; $g(t) = (R(t), T(t))$ then belongs to $SE(3)$, the special Euclidean group of

rigid motion in $\mathbb{R}^3$. In equation (5.1) we know that only $\mathbf{x}(t)$ is measured, while everything else is unknown.

When we consider measurements at m different times, we organize the above equations by defining:

$$M_i = A(t_i)Pg(t_i) \in \mathbb{R}^{3\times 4} \tag{5.2}$$

which we will assume to be full-rank, that is $rank(M_i) = 3$ for $i = 1, \ldots, m$. So we have

$$\begin{bmatrix} \mathbf{x}(t_1) & 0 & \cdots & 0 \\ 0 & \mathbf{x}(t_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}(t_m) \end{bmatrix} \begin{bmatrix} \lambda(t_1) \\ \lambda(t_2) \\ \vdots \\ \lambda(t_m) \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_m \end{bmatrix} p$$

which we re-write in a more compact notation as

$$X^d \vec{\lambda}^d = M^d p \tag{5.3}$$

where $M^d \in \mathbb{R}^{3m\times 4}$ will be called the **motion matrix**, $X^d \in \mathbb{R}^{3m\times m}$ the **image matrix**, and $\vec{\lambda}^d \in \mathbb{R}^m$ the **scale vector**. We here use the superscript d to indicate the *discrete* multiview case, in order to differentiate from the *continuous* or *hybrid* cases which will be discussed later on in this chapter.

5.1.1 Multilinear Constraints on Multiple Images

Let $\vec{m}_k \in \mathbb{R}^{3m}, k = 1, \ldots, 4$ denote the four column vectors of the matrix M^d and $\vec{x}_i \in \mathbb{R}^{3m}, i = 1, \ldots, m$ be the m column vectors of the matrix X^d. From the equation (5.3), we know that these column vectors must be linearly dependent. This relationship is concisely captured by the following statement:

Proposition 5.1 (Discrete Multilinear Constraints). *The coordinates $\{\mathbf{x}(t_i)\}_{i=1}^m$ represent images of the same point $p \in \mathbb{E}^3$ seen from m different views if and only if the column vectors of the matrices M^d and X^d defined in the equation (5.3) satisfy the following wedge product equation:*

$$\vec{m}_1 \wedge \vec{m}_2 \wedge \vec{m}_3 \wedge \vec{m}_4 \wedge \vec{x}_1 \wedge \cdots \wedge \vec{x}_m = 0. \tag{5.4}$$

This constraint is obviously multilinear in the measurements $\mathbf{x}(t_i)$. Constraints involving four different images are call **quadrilinear**, constraints involving three images are called **trilinear**, and those involving two images are called **bilinear.**[1] It is then straightforward to check that the bilinear type constraints are exactly the **epipolar** constraints that we have studied extensively in previous chapters for the two-view case. In general, the coefficients of all the multilinear constraints are minors of the motion matrix M^d. As it has been directly shown (see, for instance, Triggs in [117]), constraints involving more than four frames are necessarily dependent on quadrilinear, trilinear and bilinear ones. In this section we go one step further to discuss how trilinear and quadrilinear constraints are dependent on bilinear ones.

When studying the dependency among constraints, one must distinguish between **algebraic** and **geometric dependency**. Roughly speaking, algebraic dependency concerns the conditions that a point in an image must satisfy in order to be the correspondent of a point in another image. Vice versa, geometric dependency is concerned with the information that corresponding points give on the operator that maps one to the other. The two notions are related but not equivalent, and the latter bears important consequences when one is to use the constraints in optimization algorithms to recover structure and calibration. While the geometric dependency of multilinear constraints has been established before under the assumption of constant calibration [44], we give a novel, simple and rigorous proof that is valid under the more general assumption of time-varying calibration.

5.1.2 Algebraic vs. Geometric Dependency

To clarify the relation between algebraic and geometric dependency, note that in general we can express a multilinear constraint in the form: $\sum_l \alpha_l(M^d)\beta_l(X^d) = 0$ where α_l are some polynomials of entries of M^d and β_l polynomials of entries of the image coordinates X^d, with M^d and X^d defined as before. α_l's are called the **coefficients of multilinear constraints**. Studying the **algebraic dependency** between constraints then corresponds to fixing the coefficients α_l and asking whether there are some additional constraints among the joint image coordinates X^d generated by three and four views[2]. This problem has been studied many researchers and an elegant answer can be found in [43] by explicitly

[1] In the literature, these constraints may also be referred to as **quadrifocal, trifocal and bifocal tensors.**

[2] In other words, it addresses the dependency among algebraic ideals associated with the three types of multilinear constraints.

characterizing the **primary decomposition** of the ideal (in the polynomial ring of image coordinates $\mathbf{x}_i$'s) generated by the bilinear constraints in terms of that generated by trilinear ones or quadrilinear ones.

Geometric dependency, on the other hand, investigates whether, given the image coordinates X^d, the coefficients α_l corresponding to motion parameters in additional views can give additional information about M^d. These two different types of dependencies were previously pointed out (see for instance the work of Heyden [44]). For both types of dependencies, the answer is negative, *i.e.,* trilinear and quadrilinear constraints in general are dependent of bilinear ones. We here give a simple but rigorous study of the geometric dependency. The results will also validate the ambiguity analysis given in following sections.

Consider the case $m = 3$ and, for the moment, disregard the internal structure of the motion matrix $M^d \in \mathbb{R}^{9\times 4}$. Its columns can be interpreted as a basis of a four-dimensional subspace of the nine-dimensional space. The set of k-dimensional subspaces of an n-dimensional space is called a Grassman manifold and denoted by $G(n,k)$. Therefore, M^d is an element of $G(9,4)$. By just re-arranging the three blocks $M_i,\ i = 1,\ldots,3$ into three pairs, (M_1, M_2), (M_1, M_3) and (M_2, M_3), we define a map ϕ between $G(9,4)$ and three copies of $G(6,4)$

$$\begin{array}{rcl} \phi : G(9,4) & \rightarrow & G(6,4) \times G(6,4) \times G(6,4) \\ \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} & \mapsto & \left[\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}, \begin{bmatrix} M_2 \\ M_3 \end{bmatrix}, \begin{bmatrix} M_1 \\ M_3 \end{bmatrix} \right]. \end{array}$$

The question of whether trilinear constraints are independent of bilinear ones is tightly related to whether these two representations of the motion matrix M^d are equivalent. Since the coefficients in the multilinear constraints are homogeneous in the entries of each block M_i, the motion matrix M^d is only determined up to the equivalence relation:

$$M^d \sim M'^d \; if \; \exists \lambda_i \in \mathbb{R}^*, M_i = \lambda_i M_i', \quad i = 1,\ldots,m \tag{5.5}$$

where $\mathbb{R}^* = \mathbb{R} \setminus \{0\}$. Thus for multilinear constraints the motion matrix is only well-defined as an element of the quotient space $G(3m,4)/\sim$ which is of dimension $(11m-15)$, [3] as was already noted by Triggs [117].

[3] The Grassman manifold $G(3m,4)$ has dimension $(3m-4)4 = 12m-16$. The dimension of the quotient space is $m-1$ smaller since the equivalence relation has $m-1$ independent scales.

We are now ready to prove that coefficients α_l's in trilinear and quadrilinear constraints depend on those in bilinear ones.

Theorem 5.2 (Geometric Dependency). *Given three (or four) views, the coefficients of all bilinear constraints or equivalently the corresponding fundamental matrices uniquely determine the motion matrix M^d as an element in $G(9,4)/\sim$ (or $G(12,4)/\sim$) given that $Ker(M_i)$'s are linearly independent.*

Proof: It is known that between any pair of images (i,j) the motion matrix: $\begin{bmatrix} M_i \\ M_j \end{bmatrix} \in G(6,4)$, is determined by the corresponding fundamental matrix F_{ij} up to two scalars λ_i, λ_j: $\begin{bmatrix} \lambda_i M_i \\ \lambda_j M_j \end{bmatrix} \in G(6,4), \quad \lambda_j \in \mathbb{R}^*$. Hence for the three view case all we need to prove is that the map:

$$\tilde{\phi} : (G(9,4)/\sim) \;\rightarrow\; (G(6,4)/\sim)^3$$

is injective. To this end, assume $\tilde{\phi}(M^d) = \tilde{\phi}(M'^d)$; then we have that, after re-scaling, $\begin{bmatrix} M_1' \\ M_2' \end{bmatrix} = \begin{bmatrix} \lambda_1 M_1 \\ M_2 \end{bmatrix} G_1$, $\begin{bmatrix} M_2' \\ M_3' \end{bmatrix} = \begin{bmatrix} \lambda_2 M_2 \\ M_3 \end{bmatrix} G_2$, $\begin{bmatrix} M_1' \\ M_3' \end{bmatrix} = \begin{bmatrix} M_1 \\ \lambda_3 M_3 \end{bmatrix} G_3$ for some $\lambda_i \in \mathbb{R}^*$ and $G_i \in GL(4)$,[4] $i = 1,2,3$. This yields $M_1(\lambda_1 G_1 - G_3) = 0, M_2(\lambda_2 G_2 - G_1) = 0, M_3(\lambda_3 G_3 - G_2) = 0$. Therefore there exist $U_i \in \mathbb{R}^{4\times 4}, i = 1,2,3$ with each column of U_i is in $Ker(M_i)$ such that:

$$G_3 - \lambda_1 G_1 = U_1, \quad G_1 - \lambda_2 G_2 = U_2, \quad G_2 - \lambda_3 G_3 = U_3.$$

Combining these three equations, we obtain:

$$(1 - \lambda_1\lambda_2\lambda_3) G_1 = \lambda_2\lambda_3 U_1 + \lambda_2 U_3 + U_2.$$

The matrix on the right hand side of the equation has a non-trivial null space since its column vectors are in the space $span\{Ker(M_1), Ker(M_2), Ker(M_3)\}$ which has dimension three. However, G_1 is non-singular, and therefore it must be $\lambda_1\lambda_2\lambda_3 = 1$. This gives $\lambda_1 G_1 - G_3 = -\lambda_1(\lambda_2 G_2 - G_1) - \lambda_1\lambda_2(\lambda_3 G_3 - G_2)$. That is, the columns of $\lambda_1 G_1 - G_3$ are linear combinations of columns of $\lambda_2 G_2 - G_1$ and $\lambda_3 G_3 - G_2$. But $Ker(M_i), i = 1,2,3$ are

[4] $GL(4)$ is the general linear group of all non-degenerate 4×4 real matrices.

linearly independent. Thus we have $\lambda_1 G_1 = G_3, \lambda_2 G_2 = G_1, \lambda_3 G_3 = G_2$. This implies

$$\begin{bmatrix} M_1' \\ M_2' \\ M_3' \end{bmatrix} = \begin{bmatrix} \lambda_1 M_1 \\ M_2 \\ \lambda_1 \lambda_3 M_3 \end{bmatrix} G_1.$$

which means that M'^d and M^d are the same, up to the equivalence relation defined in equation (5.5). Therefore, they represent the same element in $G(9,4)/\sim$, which means that the map $\tilde{\phi}$ is injective.

In the case of four views, in order to show that coefficients in quadrilinear constraints also depend on bilinear ones, one only needs to check that the obvious map from $G(12,4)/\sim$ to $(G(9,4)/\sim)^4$ is injective. This directly follows from the above proof of the three frame case. ∎

Comment 5.3. *As a consequence of the theorem, coefficients α_l's in trilinear and quadrilinear constraints are functions of those in bilinear ones. While the above proof shows that the map $\tilde{\phi}$ can be inverted, it does not provide an explicit characterization of the inverse. Such an inverse can in principle be highly non-linear and conditioning issues need to be taken into account in the design of estimation algorithms. We emphasize that the geometric dependency does* not *imply that two views are sufficient for reconstruction! It claims that given n views, their geometry is characterized by considering only combinations of pairs of them through bilinear constraints, while trilinear constraints are of help* only *in the case of singular configurations of points and camera (see comment 5.4). For four views, the condition that $Ker(M_i), i = 1, \ldots, 4$ are linearly independent is not necessary. A less conservative condition is that there exist two groups of three frames which satisfy the condition for the three view case.*

Theorem 5.2 requires that the one-dimensional kernels of the matrices $M_i, i = 1, \ldots, m$ ($m = 3$ or 4) are linearly independent. Note that the kernels of M_i for $i = 1, 2, 3, 4$ are given by $(-T_i^T R_i, 1)^T$, where the vector $-R_i^T T_i \in \mathbb{R}^3$ is exactly the position of the i^{th} camera center. Hence the condition of the theorem is satisfied if and only if the centers of projection of the cameras generate a hyper-plane of dimension $m - 1$. In particular, when $m = 3$, the three camera centers form a triangle, and when $m = 4$, the four camera centers form a tetrahedron.

Comment 5.4 (Critical Surfaces and Motions). *Although we have shown that the coefficients of multilinear constraints depend on those of bilinear ones, we have assumed that the latter (or the corresponding fundamental matrices) are uniquely determined by the epipolar geometry. However, this is not true when all the points lie on critical surfaces. In this case, as argued by Maybank in [76], we may obtain up to three ambiguous solutions from the bilinear constraints. This is one of the cases when trilinear and quadrilinear constraints provide useful information. On this topic, see also [78]. Also, when the camera is undergoing a rectilinear motion (i.e., all optical centers are aligned), trilinear constraints provide independent information in addition to bilinear ones. This fact has been pointed out before; see for instance Heyden in [42].*

5.2 Motion Recovery from Normalized Epipolar Constraints

5.2.1 Geometric Interpretation of Multilinear Constraints

Theorem 5.2 states a very important fact: information about the camera motion is already fully contained in the bilinear constraints unless the camera center moves in a straight line – such a motion is also called **rectilinear motion.** Geometrically, this degenerate case is illustrated in Figures 5.1. In fact, a set of points $\{\mathbf{x}_j\}_{j=1}^m$ on m image planes satisfy all multilinear constraints **if and only if** "rays" extending from camera centers along these image points intersect at a *unique* point in 3D. As a consequence of this

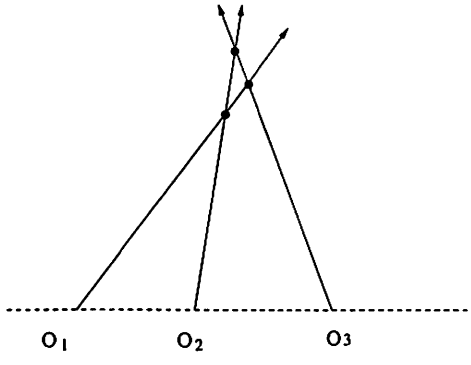


Figure 5.1: Degeneracy: Centers of camera lie on a straight line. Coplanar constraints are not sufficient to uniquely determine the intersection hence trilinear constraints are needed.

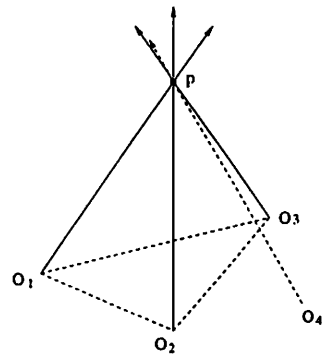


Figure 5.2: Sufficiency: Centers of camera and the point are not coplanar. Three (bilinear) coplanar constraints are sufficient to uniquely determine the intersection.

geometric interpretation of multilinear constraints, in order for an extra image to satisfy all multilinear constraints, it only needs to satisfy two (bilinear) coplanar constraints given that the new camera center is not collinear with the previous ones. For example, in Figure 5.2, in order for the fourth image to satisfy all multilinear constraints, it is sufficient for the ray (o_4, p) to be coplanar with the ray (o_2, p) and the ray (o_3, p). The coplanar condition between the ray (o_4, p) and the ray (o_1, p) is redundant.

5.2.2 Normalized Epipolar Constraints of Multiple Images

Multilinear constraints have conventionally been used to formulate various objective functions for motion recovery. However, if we do use them as constraints, we only need to pick a minimal set of independent ones. The minimal requirement is needed for Lagrangian multipliers to have a unique solution. The dependency among multilinear constraints suggests that if the centers of the camera do not lie on a straight line, pairwise epipolar constraints already provide a sufficient set of constraints. In this chapter we will assume this condition is satisfied unless otherwise stated – Comments 5.6 and 5.10 will discuss about the degenerate case. Furthermore, the (pairwise) epipolar constraints among consecutive three images naturally give a minimal set of independent constraints. In this section, we show how to use these constraints to derive a clean form of an optimal objective function for motion (and structure) recovery. In the next section, we will show how to use geometric optimization techniques to find the *optimal* solution which minimizes the objective function derived here.

Let us assume that we have m images $\{\mathbf{x}_i^j\}_{i=1}^m$ of n 3D points $p^j, 1 \leq j \leq n$ with respect to m camera frames. The rigid body motion between the k^{th} and i^{th} camera frames is $g_{ki} = (R_{ki}, T_{ki}) \in SE(3)$, $1 \leq i, k \leq m$. Thus the coordinates of each 3D point $p^j \in \mathbb{R}^3$ with respect to frames i and k are related by:

$$\mathbf{X}_k^j = R_{ki}\mathbf{X}_i^j + T_{ki}. \tag{5.6}$$

Recall the definition of essential matrix. Let us denote by $E_{ki} = \widehat{T}_{ki}R_{ki} \in \mathbb{R}^{3\times 3}$ the essential matrix associated with the camera motion between the k^{th} and i^{th} frames, then in absence of noise, image points $\mathbf{x}_i^j$ satisfy the epipolar constraints:

$$\mathbf{x}_k^{jT} E_{ki} \mathbf{x}_i^j = 0. \tag{5.7}$$

Optimal Triangulation Problem (Multiview Case): *In presence of* isotropic *noises, we seek for points* $\tilde{\mathbf{x}} = \{\tilde{\mathbf{x}}_i^j\}$ *on the image plane and a configuration of* m *camera frames* $\mathcal{G} = \{g_{ki}\}$ *such that they minimize the total* **reprojection error**. *That is, we are to minimize the objective:*

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{j=1}^{n} \sum_{i=1}^{m} \|\tilde{\mathbf{x}}_i^j - \mathbf{x}_i^j\|^2 \tag{5.8}$$

subject to the constraints:

$$\tilde{\mathbf{x}}_{i+1}^{jT} E_{i+1,i} \tilde{\mathbf{x}}_i^j = 0, \quad \tilde{\mathbf{x}}_{k+2}^{jT} E_{k+2,k} \tilde{\mathbf{x}}_k^j = 0, \quad \tilde{\mathbf{x}}_l^{jT} e_3 = 1 \tag{5.9}$$

where $e_3 = [0, 0, 1]^T \in \mathbb{R}^3, 1 \leq i \leq m-1, 1 \leq k \leq m-2, 1 \leq l \leq m$ *and* $1 \leq j \leq n$.

The first two constraints are epipolar constraints among three consecutive images. From the previous section, we know that they form a minimal (but sufficient) set of constraints among multiview images under a generic configuration. We will discuss the degeneracy case in Comments 5.6 and 5.10. The last constraint is for the imaging model of perspective projection.[5] Using **Lagrangian multipliers**, the above constrained optimization problem is equivalent to minimizing:

$$\sum_{j=1}^{n} \sum_{i=1}^{m} \Big(\|\tilde{\mathbf{x}}_i^j - \mathbf{x}_i^j\|^2 + \sum_{k=i+1}^{i+2} \alpha_{ki}^j \tilde{\mathbf{x}}_k^{jT} E_{ki} \tilde{\mathbf{x}}_i^j 1_{k \leq m} + \beta_i^j (\tilde{\mathbf{x}}_i^{jT} e_3 - 1) \Big)$$

for some $\alpha_{ki}^j, \beta_i^j \in \mathbb{R}$. From the necessary condition $\nabla F = 0$ at local minima,

$$2(\tilde{\mathbf{x}}_i^j - \mathbf{x}_i^j) + \sum_{k=i+1}^{i+2} \alpha_{ki}^j E_{ki}^T \tilde{\mathbf{x}}_k^j 1_{k \leq m} + \sum_{k=i-2}^{i-1} \alpha_{ik}^j E_{ik} \tilde{\mathbf{x}}_k^j 1_{k \geq 1} + \beta_i^j e_3 \;\; = \;\; 0$$

for all $i = 1, \ldots, m,\ j = 1, \ldots, n$. Multiplying the above equation by $\widehat{e}_3^T \widehat{e}_3$ to eliminate β_i^j, we obtain:

$$2(\mathbf{x}_i^j - \tilde{\mathbf{x}}_i^j) = \widehat{e}_3^T \widehat{e}_3 \Big(\sum_{k=i+1}^{i+2} \alpha_{ki}^j E_{ki}^T \tilde{\mathbf{x}}_k^j 1_{k \leq m} + \sum_{k=i-2}^{i-1} \alpha_{ik}^j E_{ik} \tilde{\mathbf{x}}_k^j 1_{k \geq 1} \Big) \tag{5.10}$$

for all $i = 1, \ldots, m,\ j = 1, \ldots, n$. It is readily seen that, in order to convert the above constrained optimization to an unconstrained one, we need to solve for α_{ki}^j and α_{ik}^j's. For this purpose, we define vectors $\tilde{\mathbf{x}}^j, \mathbf{x}^j, \Delta\mathbf{x}^j \in \mathbb{R}^{3m}$ associated to the j^{th} point: $\tilde{\mathbf{x}}^j =$

[5] Without loss of generality, we here will only discuss the perspective projection. The spherical projection is similar and hence omitted for simplicity.

$\left[\tilde{\mathbf{x}}_1^{jT}, \ldots, \tilde{\mathbf{x}}_m^{jT}\right]^T, \mathbf{x}^j = \left[\mathbf{x}_1^{jT}, \ldots, \mathbf{x}_m^{jT}\right]^T, \Delta\mathbf{x}^j = \mathbf{x}^j - \tilde{\mathbf{x}}^j$, and the vector of all the Lagrangian multipliers $\alpha^j = [\alpha_{21}^j, \alpha_{31}^j, \alpha_{32}^j, \alpha_{42}^j, \alpha_{43}^j, \ldots, \alpha_{m,m-2}^j, \alpha_{m,m-1}^j]^T \in \mathbb{R}^{2m-3}$, and matrix $D \in \mathbb{R}^{3m\times 3m}$ with $\widehat{e}_3^T\widehat{e}_3$ as diagonal blocks:

$$D = \begin{bmatrix} \widehat{e}_3^T\widehat{e}_3 & \cdots & 0_{3\times 3} \\ \vdots & \ddots & \vdots \\ 0_{3\times 3} & \cdots & \widehat{e}_3^T\widehat{e}_3 \end{bmatrix}.$$

We define, for $m \geq 3$, matrices $E = E(m) \in \mathbb{R}^{3m\times 3(2m-3)}$ and $\tilde{X}^j = \tilde{X}^j(m) \in \mathbb{R}^{3m\times(2m-3)}$ recursively as:

$$\begin{aligned} E(m) &= \left[\begin{array}{c|c} E(m-1) & 0_{(3m-9)\times 6} \\ 0_{3\times 3(2m-5)} & E_m \end{array}\right], \\ \tilde{X}^j(m) &= \left[\begin{array}{c|c} \tilde{X}^j(m-1) & 0_{(3m-9)\times 2} \\ 0_{3\times(2m-5)} & \tilde{X}_m^j \end{array}\right] \end{aligned}$$

with

$$E(2) = \begin{bmatrix} E_{21}^T \\ E_{21} \end{bmatrix}, \qquad E_m = \begin{bmatrix} E_{m,m-2}^T & 0_{3\times 3} \\ 0_{3\times 3} & E_{m,m-1}^T \\ E_{m,m-2} & E_{m,m-1} \end{bmatrix},$$

$$\tilde{X}^j(2) = \begin{bmatrix} \tilde{\mathbf{x}}_2^j \\ \tilde{\mathbf{x}}_1^j \end{bmatrix}, \qquad \tilde{X}_m^j = \begin{bmatrix} \tilde{\mathbf{x}}_m^j & 0_{3\times 1} \\ 0_{3\times 1} & \tilde{\mathbf{x}}_m^j \\ \tilde{\mathbf{x}}_{m-2}^j & \tilde{\mathbf{x}}_{m-1}^j \end{bmatrix}.$$

We define the **pseudo-array multiplication** $E \cdot \tilde{X}^j$ recursively as:

$$E(m) \cdot \tilde{X}^j(m) = \left[\begin{array}{c|c} E(m-1) \cdot \tilde{X}^j(m-1) & 0_{(3m-9)\times 2} \\ 0_{3\times(2m-5)} & E_m \cdot \tilde{X}_m^j \end{array}\right]$$

with

$$E(2) \cdot \tilde{X}^j(2) = \begin{bmatrix} E_{21}^T\tilde{\mathbf{x}}_2^j \\ E_{21}\tilde{\mathbf{x}}_1^j \end{bmatrix},$$

$$E_m \cdot \tilde{X}_m^j = \begin{bmatrix} E_{m,m-2}^T\tilde{\mathbf{x}}_m^j & 0_{3\times 1} \\ 0_{3\times 1} & E_{m,m-1}^T\tilde{\mathbf{x}}_m^j \\ E_{m,m-2}\tilde{\mathbf{x}}_{m-2}^j & E_{m,m-1}\tilde{\mathbf{x}}_{m-1}^j \end{bmatrix}.$$

Using this notation, the equation (5.10) can be rewritten as:

$$2\Delta\mathbf{x}^j = DE \cdot \tilde{X}^j\alpha^j. \tag{5.11}$$

Note that D is a projection matrix, *i.e.*, $D^2 = D$. All the constraints in (5.9) then can be rewritten compactly as two matrix equations:

$$\tilde{\mathbf{x}}^{jT} E \cdot \tilde{X}^j = 0, \quad D\Delta\mathbf{x}^j = \Delta\mathbf{x}^j. \tag{5.12}$$

The first equation is simply a matrix expression for all the epipolar constraints. Thus we can solve from equation (5.11) for α^j:

$$\alpha^j = 2\left(\tilde{X}^{jT} \cdot E^T D E \cdot \tilde{X}^j\right)^{-1} \tilde{X}^{jT} \cdot E^T \mathbf{x}^j \tag{5.13}$$

given that the matrix $G = \tilde{X}^{jT} \cdot E^T D E \cdot \tilde{X}^j$ is invertible. We call matrix G the **observability Grammian.**

Comment 5.5 (Observability Grammian). *In general, the observability Grammian is invertible even in cases that the algorithm is not designed for, i.e., the camera motions are such that optical centers lie on a straight line, except for points on the line. In fact, 3D points which make the Grammian degenerate, i.e.,* $\det(G) = 0$ *are very rare. Geometrically, it means that, given a sequence of camera motions, the 3D location of a point whose images make the Grammian degenerate is not* **observable.** *For example, for camera translating in a straight line, points on the line itself then satisfy* $\det(G) = 0$ *hence their images contain no information about neither their 3D location nor the camera motion on the line. In this sense, G can be thought of as the* **observability matrix** *in control theory.*

Substituting the expression for α^j (5.13) into (5.11), we then obtain the expression for $\Delta\mathbf{x}^j$ and we have:

$$\|\Delta\mathbf{x}^j\|^2 = \mathbf{x}^{jT} E \cdot \tilde{X}^j \left(\tilde{X}^{jT} \cdot E^T D E \cdot \tilde{X}^j\right)^{-1} \tilde{X}^{jT} \cdot E^T \mathbf{x}^j. \tag{5.14}$$

Substituting this expression into the objective function $F(\mathcal{G}, \tilde{\mathbf{x}})$ we obtain:

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{j=1}^{n} \mathbf{x}^{jT} E \cdot \tilde{X}^j \left(\tilde{X}^{jT} \cdot E^T D E \cdot \tilde{X}^j\right)^{-1} \tilde{X}^{jT} \cdot E^T \mathbf{x}^j. \tag{5.15}$$

Notice that the terms on the right hand side of the equation are exactly multiview versions of the **crossed normalized epipolar constraints**, but it is *by no means* a trivial sum of the pairwise crossed normalized epipolar constraints [73]. In order to minimize $F(\mathcal{G}, \tilde{\mathbf{x}})$, we need to iterate between the camera motion $\mathcal{G}$ and triangulated structure $\tilde{\mathbf{x}}$, which would be essentially a multiview version of the **optimal triangulation** procedure proposed in [73].

In this chapter, however, we will only demonstrate how to obtain optimal motion estimates. Note that, in the expression for $F(\mathcal{G}, \tilde{\mathbf{x}})$, the matrix $\tilde{X}^j$ is a function of $\tilde{\mathbf{x}}^j$ instead of the measured $\mathbf{x}^j$. In general, the difference between $\tilde{\mathbf{x}}^j$ and $\mathbf{x}^j$ is small. Therefore, we may approximate $\tilde{X}^j$ by replacing $\tilde{\mathbf{x}}_i^j$ in $\tilde{X}^j$ by the known $\mathbf{x}_i^j$. We call the resulting matrix as X^j. We then obtain a new function (in camera motion only) $F_n(\mathcal{G}) = F(\mathcal{G}, \mathbf{x})$:

$$F_n(\mathcal{G}) = \sum_{j=1}^{n} \mathbf{x}^{jT} E \cdot X^j \left(X^{jT} \cdot E^T D E \cdot X^j \right)^{-1} X^{jT} \cdot E^T \mathbf{x}^j. \tag{5.16}$$

In absence of noise, each term of $F_n(\mathcal{G})$ should be:

$$\mathbf{x}^{jT} E \cdot X^j \left(X^{jT} \cdot E^T D E \cdot X^j \right)^{-1} X^{jT} \cdot E^T \mathbf{x}^j = 0. \tag{5.17}$$

We call this the **normalized epipolar constraint** of multiple images. This is a natural generalization of the normalized epipolar constraint in the two view case [73]. Thus, as in the two view case, $F_n(\mathcal{G})$ can be regarded as a statistically adjusted objective function for directly estimating the camera motions.

Comment 5.6 (Bilinear vs. Trilinear Constraints). *It is true that one can also use a set of independent trilinear constraints to replace those in (5.9) and, with a similar exercise, derive its normalized version for motion (and structure) estimation. However, trilinear tensors (as functions of camera motions) do not have as good geometric structure as the bilinear ones. This makes the associated optimization problem harder to describe, even though it is essentially an equivalent optimization problem. One must also be aware that, in the rectilinear motion case, the normalized epipolar constraint objective F_n is not supposed to have a unique minimum (as we will soon see in Simulation 3, in presence of noise, this is not completely true. We will discuss further the new meaning of the minimum in Comment 5.10) while the corresponding normalized trilinear one always gives a unique solution.*

Comment 5.7 (Calibrated vs. Uncalibrated Camera). *In the case of an uncalibrated camera, nothing substantial will change in the above derivation except that the essential matrices need to be replaced by fundamental matrices and that the camera intrinsic parameters will introduce 5 new unknowns.*

5.2.3 Geometric Optimization Techniques

F_n in the previous section is a function defined on the space of configurations of m camera frames, which is not a regular Euclidean space. Thus conventional optimization techniques cannot be directly applied to minimizing F_n. In this section, we show how to apply newly developed geometric optimization techniques [19, 97] to solve this problem. We here will adopt the Newton's method, although it may not be the fastest, because it allows us to compute the Hessian of the objective function which is potentially useful for sensitivity analysis.

The configuration $\mathcal{G}$ of m camera frames are determined by relative rotations and translations:

$$\begin{aligned} \mathcal{R} &= [R_{21}, R_{32}, \ldots, R_{m,m-1}] \quad \in SO(3)^{m-1}, \\ \mathcal{T} &= [T_{21}^T, T_{32}^T, \ldots, T_{m,m-1}^T]^T \quad \in \mathbb{R}^{3m-3}. \end{aligned}$$

Then $F_n(\mathcal{G})$ can be denoted as $F_n(\mathcal{R}, \mathcal{T})$. It is direct to check that $F_n(\mathcal{R}, \lambda\mathcal{T}) = F_n(\mathcal{R}, \mathcal{T})$ for all $\lambda \neq 0$. Thus $F_n(\mathcal{R}, \mathcal{T})$ is a function defined on the manifold $M = SO(3)^{m-1} \times \mathbb{S}^{3m-4}$ where $\mathbb{S}^{3m-4}$ is a $3m-4$ dimensional spheroid. M is simply a product of Stiefel manifolds and it has total dimension $6m-7$. Furthermore, the (induced) Euclidean metrics on $SO(3)$ and $\mathbb{S}^{3m-4}$ are the same as their canonical metrics as Stiefel manifolds. This gives a natural Riemannian metric $\Phi(\cdot,\cdot)$ on the total manifold M. Note that any tangent vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{T})}M$ can be represented as $\mathcal{X} = (\mathcal{X}_\mathcal{R}, \mathcal{X}_\mathcal{T})$, with $\mathcal{X}_\mathcal{R} \in T_\mathcal{R}(SO(3)^{m-1})$ and $\mathcal{X}_\mathcal{T} \in T_\mathcal{T}(\mathbb{S}^{3m-4})$ defined by the expressions:

$$\mathcal{X}_\mathcal{R} = [\widehat{\omega}_{21}R_{21}, \ldots, \widehat{\omega}_{m,m-1}R_{m,m-1}], \tag{5.18}$$

$$\mathcal{X}_\mathcal{T} = [\mathcal{X}_{21}^T, \ldots, \mathcal{X}_{m,m-1}^T]^T \tag{5.19}$$

where $\omega_{i+1,i} \in \mathbb{R}^3, \mathcal{X}_{i+1,i} \in \mathbb{R}^3, i = 1, \ldots, m-1$ and $\mathcal{X}_\mathcal{T}{}^T\mathcal{T} = 0$. Then the Riemannian metric $\Phi(\cdot,\cdot)$ on the manifold M is explicitly given by:

$$\Phi(\mathcal{X}, \mathcal{X}) = \sum_{i=1}^{m-1} \omega_{i+1,i}^T \omega_{i+1,i} + \mathcal{X}_\mathcal{T}^T \mathcal{X}_\mathcal{T}. \tag{5.20}$$

Similar to the two view case in Chapter 4, we can directly apply the Riemannian optimization schemes developed in [19, 97] for minimizing the function $F_n(\mathcal{R}, \mathcal{T})$.

Algorithm 5.8 (Riemannian Newton's Algorithm Minimizing $F_n(\mathcal{R}, \mathcal{T})$).

1. *Pick an orthonormal basis* $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ *on* $T_{(\mathcal{R},\mathcal{T})}M$. *Compute the vector* $\mathbf{g} \in \mathbb{R}^{6m-7}$ *with its* i^{th} *entry given by* $(\mathbf{g})_i = dF_n(\mathcal{B}^i)$. *Compute the matrix* $\mathbf{H} \in \mathbb{R}^{(6m-7)\times(6m-7)}$ *with its* $(i,j)^{th}$ *entry given by* $(\mathbf{H})_{i,j} = HessF_n(\mathcal{B}^i, \mathcal{B}^j)$. *Compute the vector* $\delta = -\mathbf{H}^{-1}\mathbf{g} \in \mathbb{R}^{6m-7}$.

2. *Recover the vector* $\Delta \in T_{(\mathcal{R},\mathcal{T})}M$ *whose coordinates with respect to the orthonormal basis* $\mathcal{B}^i$*'s are exactly* δ. *Update the point* $(\mathcal{R},\mathcal{T})$ *along the geodesic to* $\exp(\Delta)$.

3. *Repeat step 1 if* $\|\mathbf{g}\| \geq \epsilon$ *for some pre-specified tolerance* $\epsilon > 0$.

In the above algorithm, we still need to know: how to pick an orthonormal basis on TM, how to compute geodesics on the manifold M, and how to compute the gradient and Hessian of F_n.

Using the Gram-Schmidt process, we can find vectors $V_{\mathcal{T}}^1, \ldots, V_{\mathcal{T}}^{3m-4} \in \mathbb{R}^{3m-3}$ such that, together with $\mathcal{T}$, they form an orthonormal basis of $\mathbb{R}^{3m-3}$. Let $e_1, e_2, e_3 \in \mathbb{R}^3$ be the standard orthonormal basis of $\mathbb{R}^3$. Then a natural orthonormal basis $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ on $T_{(\mathcal{R},\mathcal{T})}M$ is given by:

$$\mathcal{B}^{3i-3+j} = \left([0,\ldots,0,\widehat{e}_j R_{i+1,i},0,\ldots,0],\mathbf{0}\right)$$

for $1 \leq i \leq m-1$, $1 \leq j \leq 3$ and

$$\mathcal{B}^{3m-3+i} = \left(\mathbf{0}, V_{\mathcal{T}}^i\right), \quad \text{for } 1 \leq i \leq 3m-4.$$

Given a vector $\mathcal{X} = (\mathcal{X}_{\mathcal{R}}, \mathcal{X}_{\mathcal{T}}) \in T_{(\mathcal{R},\mathcal{T})}M$ with $\mathcal{X}_{\mathcal{R}}$ and $\mathcal{X}_{\mathcal{T}}$ given by (5.18) and (5.19) respectively, the geodesic $(\mathcal{R}(t),\mathcal{T}(t)) = \exp(\mathcal{X}t), t \in \mathbb{R}$ is given by:

$$\mathcal{R}(t) = (e^{t\widehat{\omega}_{21}}R_{21}, e^{t\widehat{\omega}_{32}}R_{32}, \ldots, e^{t\widehat{\omega}_{m,m-1}}R_{m,m-1}), \tag{5.21}$$

$$\mathcal{T}(t) = \mathcal{T}\cos(\sigma t) + U\sin(\sigma t), \ \sigma = \|\mathcal{X}_{\mathcal{T}}\|, U = \mathcal{X}_{\mathcal{T}}/\sigma. \tag{5.22}$$

The tangent of this geodesic at $t = 0$ is exactly $\mathcal{X}$.

With an orthonormal basis, the computation of gradient and Hessian can be reduced to directional derivatives along geodesics on M. Given a vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{T})}M$, let $(\mathcal{R}(t),\mathcal{T}(t)) = \exp(\mathcal{X}t)$. Then we have:

$$dF_n(\mathcal{X}) = \frac{dF_n(\mathcal{R}(t),\mathcal{T}(t))}{dt},$$

$$\text{Hess}F_n(\mathcal{X},\mathcal{X}) = \frac{d^2F_n(\mathcal{R}(t),\mathcal{T}(t))}{dt^2}.$$

Polarizing $\mathrm{Hess}F_n(\mathcal{X},\mathcal{X})$ we can obtain the expression for $\mathrm{Hess}F_n(\mathcal{X},\mathcal{Y})$ for arbitrary $\mathcal{X},\mathcal{Y} \in T_{(\mathcal{R},\mathcal{T})}M$:

$$\mathrm{Hess}F_n(\mathcal{X},\mathcal{Y}) = \frac{1}{4}\big(\mathrm{Hess}F_n(\mathcal{X}+\mathcal{Y},\mathcal{X}+\mathcal{Y}) - \mathrm{Hess}F_n(\mathcal{X}-\mathcal{Y},\mathcal{X}-\mathcal{Y})\big).$$

According to the definition of gradient, $\mathrm{grad}F_n \in T_{(\mathcal{R},\mathcal{T})}M$, which is given by:

$$dF_n(\mathcal{X}) = \Phi(\mathrm{grad}F_n,\mathcal{X}), \quad \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{T})}M, \tag{5.23}$$

is exactly equal to the 1-form dF_n with respect to an orthonormal frame. Therefore, at each point $(\mathcal{R},\mathcal{T})$, we pick the orthonormal basis $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ on $T_{(\mathcal{R},\mathcal{T})}M$ as above and compute the first and second order derivatives of F_n with respect to corresponding geodesics of the base vectors. The gradient and Hessian of F_n are then explicitly expressed by the vector $\mathbf{g}$ and the matrix $\mathbf{H}$ as described in the above algorithm. The updating vector Δ computed in the algorithm is in fact intrinsically defined[6] and satisfies:

$$\mathrm{Hess}F_n(\Delta,\mathcal{X}) = \Phi(-\mathrm{grad}F_n,\mathcal{X}), \quad \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{T})}M. \tag{5.24}$$

Note that F_n has a very good structure – only matrix E depends on $(\mathcal{R},\mathcal{T})$ and it consists of blocks of essential matrices $E_{i+1,i}$ and $E_{i+2,i}$. The computation of the Hessian can then be reduced to computing derivatives of these matrices with respect to the chosen base vectors. From the definition of the essential matrix E_{ki}, we have:

$$\begin{aligned} E_{i+1,i} &= \widehat{T}_{i+1,j}R_{i+1,i}, \\ E_{i+2,i} &= E_{i+2,i+1}R_{i+1,i} + R_{i+2,i+1}E_{i+1,i}. \end{aligned}$$

Hence the computation can be further reduced to derivatives of essential matrix $E_{i+1,i}$ only. For a vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{T})}M$ of the form given by (5.18) and (5.19), by direct computation, we have:

$$\begin{aligned} dE_{i+1,i}(\mathcal{X}) &= \widehat{T}_{i+1,i}\widehat{\omega}_{i+1,i}R_{i+1,i} + \widehat{\mathcal{X}}_{i+1,i}R_{i+1,i}, \\ \mathrm{Hess}E_{i+1,i}(\mathcal{X},\mathcal{X}) &= \widehat{T}_{i+1,i}\widehat{\omega}^2_{i+1,i}R_{i+1,i} + 2\widehat{\mathcal{X}}_{i+1,i}\widehat{\omega}_{i+1,i}R_{i+1,i} - \mathcal{X}^T_{i+1,i}\mathcal{X}_{i+1,i}\widehat{T}_{i+1,i}R_{i+1,i} \end{aligned}$$

for $i = 1,\ldots,m-1$. Note that these formulae are consistent to the corresponding ones in the two view case. Thus we now have all the necessary ingredients for implementing the proposed optimization scheme. For any given number of camera frames, we get an optimal estimate of the camera relative configuration by minimizing the normalized epipolar objective F_n.

[6]That is, the definition of Δ is independent of the choice of coordinate frame.

Comment 5.9 (Newton vs. Levenberg-Marquardt). *The difference between Newton and Levenberg-Marquardt (LM) methods is that in LM the Hessian is approximated by some form of the objective function's gradient. Since the gradient only involves first order derivatives, LM in general is much less costly in each step. From our implementation of the Newton's algorithm, the Hessian indeed takes more than 95% of the computing time. Nevertheless, we computed the Hessian anyway since the formula would be useful for future sensitivity analysis of motion estimation in the multiview case.*

5.2.4 Simulations and Experiments

In this section, we show by simulations and experiments the performance of the normalized epipolar constraint. We will apply it to cases *with* or *without* the sufficiency of the epipolar constraint satisfied.

Setup: Table 5.1 shows the simulation parameters used. In the table, u.f.l. stands for *unit of focal length.* The ratio of the magnitude of translation and rotation, or simply the T/R *ratio*, is compared at the center of the random cloud (scattered in the truncated pyramid specified by the given field of view and depth variation). For all simulations, independent Gaussian noise with std given in unit of pixel is added to each image point. In general, the amount of rotation between consecutive frames is about 20^o and the amount of translation is then automatically given by the T/R ratio. In the following, camera motions will be specified by their translation and rotation axes. For example, between a pair of frames, the symbol XY means that the translation is along the X-axis and rotation is along the Y-axis. If n such symbols are connected by hyphens, it specifies a sequence of consecutive motions. Error measure for rotation is $\arccos\left(\frac{tr(R_{ki}\tilde{R}_{ki}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true R. Error measure for translation is the angle between T_{ki} and $\tilde{T}_{ki}$ in degrees where $\tilde{T}$ is

Table 5.1: Simulation parameters

Parameter	Unit	Value
Number of trials		100 - 500
Number of points		20
Number of frames		3-4
Field of view	degrees	90
Depth variation	u.f.l.	100 - 400
Image size	pixels	500 × 500

an estimate of the true T. All nonlinear (two view or multiview) algorithms are initialized by estimates from the conventional two view linear algorithm.[7]

Simulation 1 (Comparison with Two Frame Bilinear and Normalized Epipolar Constraints) Figure 5.3 plots the errors of rotation estimates and translation estimates compared with results from the standard 8-point linear algorithm and nonlinear algorithm for pairwise views [73]. As we see, normalization among multiple images indeed performs better than normalization among only pairwise images.

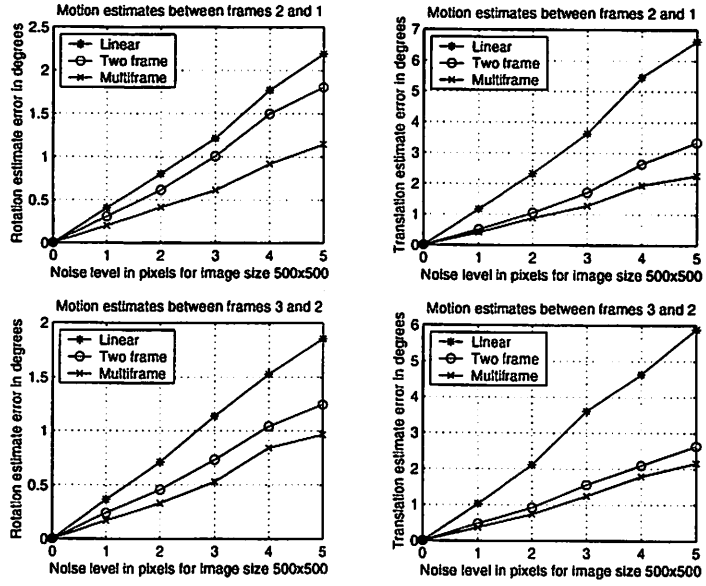


Figure 5.3: Motion estimate error comparison between normalized epipolar constraint of three frames, normalized epipolar constraint of two frames and (bilinear) epipolar constraint. The number of trials is 500, camera motions are XX-YY and T/R ratio is 1.

Simulation 2 (Axis Dependency Profile) We run the multiview algorithm with consecutive motions along the *same* rotation and translation axes for all nine possible combinations. See Figure 5.4. Note that our multiview algorithm is not designed to work in rectilinear motion case, such as XX-XX, YY-YY and ZZ-ZZ. Nevertheless, the simulation results in the figure show that the translation estimates still converge to the correct translational direction and the error angles between estimates and the true ones are comparable to other generic cases. As we see, the estimate error is larger when translation along the Z-axis is present. This is because of a smaller signal to noise ratio in this case.

[7] In the multiview case, the relative scales between translations are initialized by triangulation since the directions of translations are known from estimates given by the linear algorithm.

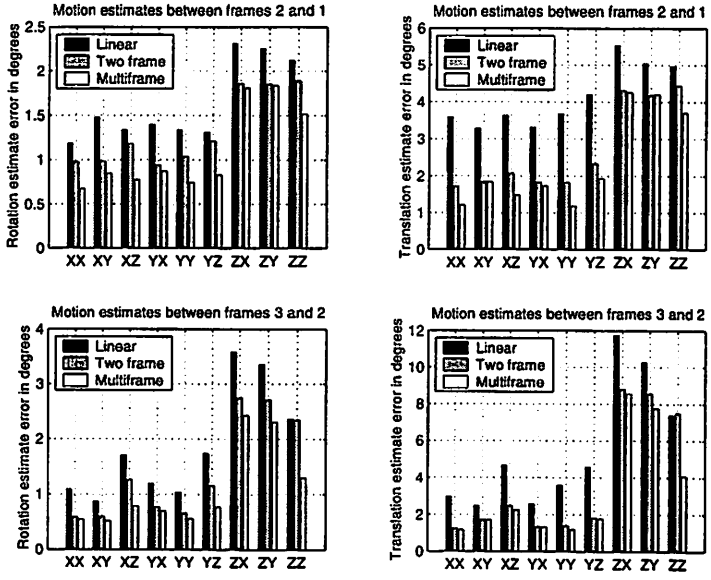


Figure 5.4: Axis dependency profile: The algorithms are run for all nine combinations of camera rotation and translation w.r.t. the X, Y and Y axes. The number of trials is 100, noise level is 3 pixel std and T/R ratio is 1.

Simulation 3 (A Statistically Stable Solution for Rectilinear Motion from Normalized Epipolar Constraint) From the previous simulation, we notice that the algorithm indeed converges to the correct translational direction in the rectilinear motion case. Then how about the relative scales between consecutive translations? They are usually believed to be captured only by trilinear constraints but not by bilinear ones. This is *not completely true*: The rectilinear motion is indeed a degenerate case for the bilinear constraints, from which there is no unique solution for the relative scales – (for example see Figure 5.1). However, statistically, the *true* relative scales must be a *stable* solution among all the possible ones. That is, if we properly normalize the epipolar constraint w.r.t. the noise model, the true relative scale should be captured by the epipolar constraints alone as a statistically stable solution. Here, noise essentially plays a positive role of "singling out" the stable solution which otherwise would be lost when degeneracy occurs. Figure 5.5 plots two histograms of relative scale estimates given by minimizing our normalized epipolar constraint: One is for a rectilinear motion and the other one for a generic motion. Clearly, in both cases, the histogram resembles a Gaussian distribution with the mean centered at the true scale, as a result of the proper normalization. Moreover, the two histograms are comparable to each other, which suggests that, using (normalized) epipolar constraint

alone, scale estimates in a degenerate case are not necessarily worse than in a generic case.

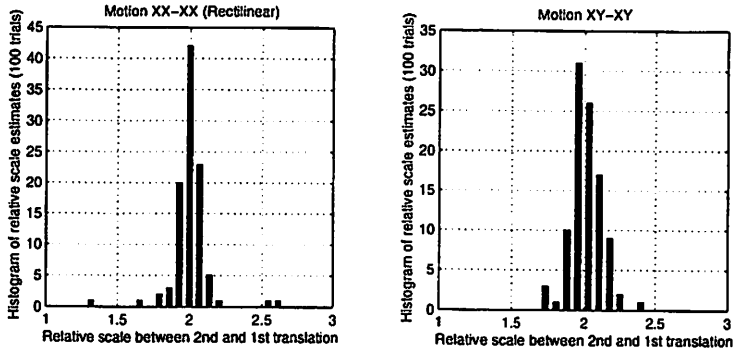


Figure 5.5: Histogram of relative scale estimates by normalized epipolar constraint in a rectilinear motion case and a generic motion case. The number of trial is 100, noise level is 3 pixel std and the true relative scale between consecutive translation is 2.

Comment 5.10. (Bilinear vs. Trilinear Constraints Continued) *Simulation 3 reveals a remarkable statistical relationship between bilinear and trilinear constraints: If an optimal estimate is obtained for generic cases, it can still be retrieved as the stable estimate in a degenerate case - the (noise-free) deterministic constraint may be degenerate, but there is no reason for the a posteriori distribution of the estimate to be degenerate as well. Geometrically, the estimate obtained in a degenerate configuration can be interpreted as a "limit" of a sequence of estimates of generic configurations. Such an estimate may also be viewed as the so called "viscous solution" of the normalized epipolar constraint if the Gaussian noise added on images is regarded as some kind of "diffusion". Therefore,* in principle, *we do not really need trilinear constraints in order to estimate motion (including relative scales) correctly even in the rectilinear motion case, although such an estimate may be more sensitive or less robust (if the noise model changes).*

Experiment (Motion Recovery from Real Images) We simply tested our algorithm on a set of real images taken by a commercial pan-tilt camera. Figure 5.6 shows four images of a cubic corner with feature points, Figure 5.7 plots the estimated and hand measured actual camera location, and Table 5.2 gives the errors between the estimated and measured motions. The camera is self-calibrated by Hartley's method for a pure rotating camera. Since our camera calibration and motion measurements are still crude, errors of this size are expected. We are currently fine-tuning our hardware setup to get better results.

Table 5.2: Motion estimate errors in degrees

Motions	Rotation Errors	Translation Errors
Frames 2-1	8.1^o	4.6^o
Frames 3-2	6.3^o	5.8^o
Frames 4-3	4.4^o	4.5^o

5.3 Continuous and Hybrid Cases

The continuous case is a limiting case of the discrete case. In this section, we study the continuous version of some of the constraints from previous sections. Some of these continuous constraints have already been used in computer vision to recover motion or structure.

5.3.1 Continuous Multilinear Constraints

Suppose that the camera calibration matrix $A(t)$ varies very slowly so that we may treat it as constant A for a short period of time around time t, then the image $\mathbf{x}(t)$ of a point $p \in \mathbb{E}^3$ satisfies:

$$\lambda(t)\mathbf{x}(t) = APg(t)p. \tag{5.25}$$

At time t, differentiating this equation $(m-1)$ times, we obtain the equation that higher order derivatives of the optical flow should satisfy:

$$\begin{bmatrix} \mathbf{x} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \dot{\mathbf{x}} & \mathbf{x} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \mathbf{x}^{(i)} & \vdots & c_k^i \mathbf{x}^{(i-k)} & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \ddots & \ddots & \vdots \\ \mathbf{x}^{(m-2)} & \cdots & \cdots & \cdots & \cdots & \mathbf{x} & 0 \\ \mathbf{x}^{(m-1)} & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{x} \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ \vdots \\ \lambda^{(i)} \\ \vdots \\ \lambda^{(m-2)} \\ \lambda^{(m-1)} \end{bmatrix} = \begin{bmatrix} APg \\ AP\dot{g} \\ \vdots \\ APg^{(i)} \\ \vdots \\ APg^{(m-2)} \\ APg^{(m-1)} \end{bmatrix} p.$$

where $c_k^i = \begin{pmatrix} i \\ k \end{pmatrix} \in \mathbb{Z}^+$ for $0 \leq k \leq i \leq (m-1)$. The quantities $\mathbf{x}^{(i)}, 0 \leq i \leq (m-1)$ are the i^{th} order derivatives of the image point, similar for $\lambda^{(i)}$ and $g^{(i)}$. If we define $c_k^i = 0$ for $i < k$, the $(i,k)^{th}$ entry (in fact a tuple) of the first matrix in the above equation has the

unified form $c_k^i \mathbf{x}^{(i-k)}$, $0 \leq i, k \leq (m-1)$. We may define matrices $X^c \in \mathbb{R}^{3m \times m}$, $M^c \in \mathbb{R}^{3m \times 4}$ and $\vec{\lambda}^c \in \mathbb{R}^m$ such that the above matrix equation can be rewritten as:

$$X^c \vec{\lambda}^c = M^c p \tag{5.26}$$

We here use the superscript c to indicate the *continuous* case. We then have the continuous version of the Proposition 5.1.

Proposition 5.11 (Continuous Multilinear Constraints). *Consider the image $\mathbf{x}(t) \in \mathbb{R}^3$ of a point p under the camera motion $g(t) \in SE(3)$. Then for the matrices $X^c \in \mathbb{R}^{3m \times m}$ and $M^c \in \mathbb{R}^{3m \times 4}$ defined in equation (5.26), the column vectors $\{\vec{x}_i\}_{i=1}^m \in \mathbb{R}^{3m}$ of the matrix X^c and column vectors $\vec{m}_1, \vec{m}_2, \vec{m}_3, \vec{m}_4 \in \mathbb{R}^{3m}$ of the matrix M^c satisfy the following wedge product equation:*

$$\vec{m}_1 \wedge \vec{m}_2 \wedge \vec{m}_3 \wedge \vec{m}_4 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_m = 0. \tag{5.27}$$

This wedge equation contains all the projective invariants associated with the motion of the image of a single point. One would see that most of the constraints given by the wedge product involve high order derivatives of the optical flows or the structural scales. Due to numerical accuracy, they are not very useful for reconstruction purpose. However, constraints involving the first derivative have been widely used. These are simply the bilinear constraints on optical flows, which are a continuous version of the bilinear epipolar constraints in the discrete case.

Without loss of generality, we may assume $g(t) = I$. Then $\dot{g}$ has the **twist form**:

$$\dot{g} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}$$

where $\omega \in \mathbb{R}^3$ is the angular velocity and $v \in \mathbb{R}^3$ the linear velocity. Then, in the special

case that $m = 2$ and $A = I$, the wedge product equation gives:

$$\vec{m}_1 \wedge \ldots \wedge \vec{m}_4 \wedge \vec{x}_1 \wedge \vec{x}_2 = \det \begin{bmatrix} Pg & \mathbf{x} & 0 \\ P\dot{g} & \dot{\mathbf{x}} & \mathbf{x} \end{bmatrix} e_1 \wedge \ldots \wedge e_6 = 0$$

$$\Leftrightarrow \quad \det \begin{bmatrix} Pg & \mathbf{x} & 0 \\ P\dot{g} & \dot{\mathbf{x}} & \mathbf{x} \end{bmatrix} = 0 \quad \Leftrightarrow \quad \det \begin{bmatrix} I & 0 & \mathbf{x} & 0 \\ \widehat{\omega} & v & \dot{\mathbf{x}} & \mathbf{x} \end{bmatrix} = 0$$

$$\Leftrightarrow \quad \det \begin{bmatrix} I & 0 & \mathbf{x} & 0 \\ 0 & v & \dot{\mathbf{x}} - \widehat{\omega}\mathbf{x} & \mathbf{x} \end{bmatrix} = 0 \quad \Leftrightarrow \quad \det[v, \dot{\mathbf{x}} - \widehat{\omega}\mathbf{x}, \mathbf{x}] = 0$$

$$\Leftrightarrow \quad \dot{\mathbf{x}}^T \widehat{v}\mathbf{x} + \mathbf{x}^T \widehat{v}\widehat{\omega}\mathbf{x} = 0.$$

This is exactly the continuous version of the epipolar constraint as we have discussed in Chapter 3. Suppose that there are n image points observed. Then such a constraint holds for all the n image points:

$$\dot{\mathbf{x}}^{jT} \widehat{v}\mathbf{x}^j + \mathbf{x}^{jT} \widehat{\omega}\widehat{v}\mathbf{x}^j = 0, \quad 1 \leq j \leq n.$$

5.3.2 Recovery of Relative Scale in the Continuous Case

As we have seen in the discrete case, the purpose of exploiting Euclidean constraints is to reconstruct the scales of the motion and structure. In the continuous case, the scale information is encoded in $\lambda^j, \dot{\lambda}^j, 1 \leq j \leq n$ for the structure of the n points and $\eta \in \mathbb{R}^+$ for the linear velocity v as in the following equation:

$$\dot{\lambda}^j \mathbf{x}^j + \lambda^j \dot{\mathbf{x}}^j = \widehat{\omega}(\lambda^j \mathbf{x}^j) + \eta v \quad \Leftrightarrow \quad \dot{\lambda}^j \mathbf{x}^j + \lambda^j (\dot{\mathbf{x}}^j - \widehat{\omega}\mathbf{x}^j) - \eta v = 0, \quad 1 \leq j \leq n \quad (5.28)$$

Known $\mathbf{x}, \dot{\mathbf{x}}, \omega$ and v, these constraints are all linear in $\lambda^j, \dot{\lambda}^j, 1 \leq j \leq n$ and η. Also, if $\mathbf{x}^j, 1 \leq j \leq n$ are linearly independent of v, *i.e.,* the feature points do not line up with the direction of translation, these linear constraints are not degenerate hence the unknown scales are determined up to a universal scale. As in the discrete case, we call a configuration **critical** if there is any $\mathbf{x}^j, 1 \leq j \leq n$ which lines up with the translational direction v. In fact, this is the limiting case of the critical configuration defined in the discrete case.

We can arrange all the scale quantities into a single vector $\vec{\lambda}$:

$$\vec{\lambda} = [\lambda^1, \ldots, \lambda^n, \dot{\lambda}^1, \ldots, \dot{\lambda}^n, \eta]^T \in \mathbb{R}^{2n+1}.$$

For n optical flows, $\vec{\lambda}$ is a $2n+1$ dimensional vector. (5.28) gives $3n$ (scalar) linear equations. The problem of solving $\vec{\lambda}$ from (5.28) is usually over-determined. As in the discrete case, it

is easy to check that in the absence of noise the set of equations given by (5.28) uniquely determine $\vec{\lambda}$ if the configuration is non-critical. As in the discrete case, we can write all the equations in the matrix form:

$$M\vec{\lambda} = 0$$

with $M \in \mathbb{R}^{3n\times(2n+1)}$ being a matrix depending on ω, v and $\{\mathbf{x}^j, \dot{\mathbf{x}}^j\}_{j=1}^n$. Then in the presence of noise, the LLSE estimate of $\vec{\lambda}$ is just the eigenvector of M^TM corresponding to the smallest eigenvalue.

Notice that the rate of scales $\{\dot{\lambda}^j\}_{j=1}^n$ are also estimated. This piece of information has been ignored in most of previous structure from motion algorithms. However, it turns out to be a very important piece of information. If we do the above estimation for a time interval, say (t_0, t_f), then we obtain the estimation $\vec{\lambda}(t)$ as a function of time t. But the estimation of $\vec{\lambda}(t)$ at each time t is only determined up to an arbitrary scale. Hence $\rho(t)\vec{\lambda}(t)$ is also a valid estimation for any positive function $\rho(t)$. However, since $\rho(t)$ is multiplied to both $\lambda(t)$ and $\dot{\lambda}(t)$. Their ratio:

$$r(t) = \dot{\lambda}(t)/\lambda(t)$$

is independent of the choice of $\rho(t)$ at each time t. Notice $\frac{d}{dt}(\ln\lambda) = \dot{\lambda}/\lambda$. Let the logarithm of the structural scale λ to be $y = \ln\lambda$. Then a time-consistent estimation $\lambda(t)$ needs to satisfy the following ordinary differential equation, we call it the **dynamic scale ODE**:

$$\dot{y}(t) = r(t).$$

Given $y(t_0) = y_0 = \dot{\lambda}(t_0)/\lambda(t_0)$, solve this ODE and and obtain $y(t)$ for $t \in [t_0, t_f]$. Then the time-consistent scale $\lambda(t)$ is simply given by:

$$\lambda(t) = \exp(y(t)).$$

Thus, all the scales estimated at different times are with respect to the scales at time t_0. One can also normalize all the scales with respect to those at time t_f by setting the final value $y(t_f)$ and then integrating the ODE backwards. Therefore, in the continuous case, we are able to recover all the scales as a function of time up to a universal scale. Notice that in particular the (relative) scales of the translational motion v are fully recovered, which is very important to many applications in mobile robot navigation.

In the continuous case, the notion of **triangulation** is essentially the same: try to find a consistent reconstruction of the Euclidean structure from all the structure estimated over time. However, it is much harder to implement in a practical algorithm since it involves integration of the motion $(\omega(t), v(t))$ unless we have an estimation of the transformation $g(t) = (R(t), T(t))$ from other sources. The issue of estimating the velocity and the transformation together will be addressed in section 5.3.3 which deals with hybrid settings. In practice, the ratio function $r(t)$ may not be available for all the times $t \in [t_0, t_f]$. One can use some simple interpolation schemes to recover $r(t)$, hence the time-consistent scale $\lambda(t)$. It is up to the user to adjust the algorithm appropriately for the specific applications.

Comment 5.12. *In both the discrete and continuous cases, the proposed algorithms reconstruct both the Euclidean structure and motion up to a single universal scale. These algorithms provide any vision-based autonomous agent, for example an autonomous mobile robot, with* **complete information** *about its surrounding environment and its ego-motion relative to the environment. The universal scale is not important since it only scales up or down the overall configuration space. All the intrinsic geometric (including metric) properties of the space are preserved. In this sense, no information is really lost through a vision system.*

5.3.3 Hybrid Multilinear Constraints

We now study the cases where both point correspondences and optical flow measurements are available. Such cases are referred to as **hybrid**. In practical systems the quality of the motion/structure estimates naturally depend on the quality of the measurements. Large motions, occlusions, reflectance variations, aliasing etc. affect negatively the quality of the flow estimates as well as the point correspondences. Therefore it is of interest to study the case when both types of measurements are used for motion and structure estimation.

Like the continuous case, we assume that the calibration matrix $A(t)$ varies slowly so that we can treat it as constant nearby each time instant t_i, for $i = 1, \ldots, m$. Suppose one point p is projected on all m image frames (in discrete positions) and its optical flows on these frames are also measured. This is a natural combination (a "direct sum") of the discrete case and the continuous case we studied in the preceding sections. For this case,

we have:

$$\begin{bmatrix} \mathbf{x}(t_1) & 0 & 0 & \cdots & 0 \\ \dot{\mathbf{x}}(t_1) & \mathbf{x}(t_1) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{x}(t_m) & 0 \\ 0 & \cdots & 0 & \dot{\mathbf{x}}(t_m) & \mathbf{x}(t_m) \end{bmatrix} \begin{bmatrix} \lambda(t_1) \\ \dot{\lambda}(t_1) \\ \vdots \\ \lambda(t_m) \\ \dot{\lambda}(t_m) \end{bmatrix} = \begin{bmatrix} A(t_1)Pg(t_1) \\ A(t_1)P\dot{g}(t_1) \\ \vdots \\ A(t_m)Pg(t_m) \\ A(t_m)P\dot{g}(t_m) \end{bmatrix} p.$$

In general, $g(t_i), \dot{g}(t_i), 1 \leq i \leq m$ have the form:

$$g(t_i) = \begin{bmatrix} R_i & T_i \\ 0 & 1 \end{bmatrix},$$

$$\dot{g}(t_i) = g(t_i) \begin{bmatrix} \widehat{\omega}_i & v_i \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_i\widehat{\omega}_i & R_iv_i \\ 0 & 0 \end{bmatrix}.$$

Similar to the discrete and continuous cases, we may define matrices $X^h \in \mathbb{R}^{6m\times 2m}$, $M^h \in \mathbb{R}^{6m\times 4}$ and $\vec{\lambda}^h \in \mathbb{R}^{2m}$ such that the above matrix equation can be rewritten as:

$$X^h\vec{\lambda}^h = M^h p. \tag{5.29}$$

We here use superscript h to indicate the *hybrid* case. We then have a hybrid version of the Propositions 5.1 and 5.11.

Proposition 5.13 (Hybrid Multilinear Constraints). *Consider n images and optical flows $\mathbf{x}^j, \dot{\mathbf{x}}^j \in \mathbb{R}^3$ for $j = 1, \ldots, n$ of a point p. Then for the matrices $X^h \in \mathbb{R}^{6m\times 2m}$ and $M^h \in \mathbb{R}^{6m\times 4}$ defined in equation (5.29), the column vectors $\{\vec{x}_i\}_{i=1}^m \in \mathbb{R}^{6m}$ of the matrix X^h and column vectors $\vec{m}_1, \vec{m}_2, \vec{m}_3, \vec{m}_4 \in \mathbb{R}^{6m}$ of the matrix M^h satisfy the following wedge product equation:*

$$\vec{m}_1 \wedge \vec{m}_2 \wedge \vec{m}_3 \wedge \vec{m}_4 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_m = 0. \tag{5.30}$$

Obviously, this wedge product equation gives all the discrete multilinear constraints (bilinear, trilinear and quadrilinear ones); it also gives all the continuous (bilinear) epipolar constraints. Further, some new constraints are given by this wedge product. These constraints involving both velocity $\{(\omega_i, v_i)\}_{i=1}^m$ and transformation $\{(R_i, T_i)\}_{i=1}^m$ are called **hybrid constraints**. In fact all the constraints given by the wedge product equation are

the same as that all the $(2m+4) \times (2m+4)$ minors of the $6m \times (2m+4)$ matrix (X^h, M^h) are degenerate (*i.e.*, the determinant is zero). All the non-trivial constraints given by these minors will be homogeneous equations in terms of the entries of $\{(\mathbf{x}_i, \dot{\mathbf{x}}_i)\}_{i=1}^m$. According to the structure of the matrix X^h, *the degree of these homogeneous (hybrid) constraints is from degree 2 to degree 8.*

Without loss of generality, we will assume that consecutive frames are non-critical. Then the homogeneous constraints above determine the velocities $\{(\omega_i, v_i)\}_{i=1}^m$ and motions $\{(R_i, T_i)\}_{i=1}^m$ with translational motion $\{v_i\}_{i=1}^m$ and $\{T_i\}_{i=1}^m$ up to unknown scales. In order to reconstruct the structural scales and the scales of motions, one needs to use the following set of Euclidean constraints from both the discrete case and the continuous case:

$$\lambda_i^j \mathbf{x}_i^j - \lambda_{i-1}^j R_i \mathbf{x}_{i-1}^j - \gamma_i T_i = 0, \quad 2 \leq i \leq m, 1 \leq j \leq n$$
$$\dot{\lambda}_i^j \mathbf{x}_i^j + \lambda_i^j (\dot{\mathbf{x}}_i^j - \widehat{\omega}_i \mathbf{x}_i^j) - \eta_i v_i = 0, \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

As long as the discrete case and continuous case respectively have unique solutions, the overall hybrid case has a unique solution (up to a universal scale). The estimation is simply an LLSE problem.

In particular, the scales of velocities at a particular time can be uniquely recovered with respect to the transformation between the current image frame and a reference image frame. This is very important for applications such as mobile robot navigation since a consistent estimation of the displacements and velocities can be obtained. Notice that, in the i^{th} image frame, we certainly can measure optical flows for points which do not have projections in the other image frames at all. Their structural scales can also be determined with respect to the same universal scale. Then *the occlusion is usually not a problem at all in the hybrid case for the recovery of depth.*

Notice that in the hybrid case, the quantities $\{\dot{\lambda}_i^j\}_{i=1,j=1}^{m,n}$ are not quite useful since we are not measuring the optical flows in a continuous fashion. So one can get rid of them by applying cross product with $\{\mathbf{x}_i^j\}_{i=1,j=1}^{m,n}$ to the continuous Euclidean constraints:

$$\dot{\lambda}_i^j \mathbf{x}_i^j + \lambda_i^j (\dot{\mathbf{x}}_i^j - \widehat{\omega}_i \mathbf{x}_i^j) - \eta_i v_i = 0 \quad \Leftrightarrow \quad \lambda_i^j (\dot{\mathbf{x}}_i^j - \widehat{\omega}_i \mathbf{x}_i^j) \times \mathbf{x}_i^j - \eta_i v_i \times \mathbf{x}_i^j = 0.$$

Then the number of states in the associated LLSE estimation problem can be reduced. This is essentially the bilinear constraint used by some researchers in the structure from motion algorithms using optical flow, see for example [98].

5.4 Discussion

In this chapter, we have introduced and clearly studied the geometric relationship of constraints among multiple images. It has been shown that epipolar constraints alone, except in the degenerate rectilinear motion case, have provided sufficient constraints for multiple images. We further contend by using (bilinear) epipolar constraint that multilinear constraints need to be properly normalized in order to get less biased estimates (of the multifocal tensors). There are several consequences of such a normalization. First, the so obtained objective function is no longer linear hence it does not preserve the tensor structure of multilinear constraints. Second, such a normalization is a natural generalization of the well known normalized epipolar constraint between two images but by no means a trivial sum of them. Third, the normalization not only provides near optimal motion estimates but, more importantly, reveals certain statistical relationship between epipolar and trilinear constraints – as a necessary complement to the well known algebraic or geometric relationship. We now know that in principle normalized epipolar constraint alone suffices for estimating correct motion as a statistically stable solution even in the rectilinear motion case. However, more extensive simulation, experiments and analysis are still needed to evaluate how really practical it is when applied to degenerate cases because it may be less robust to model change. For example, in the case when the noise on the images is no longer isotropic or identically independently distributed, we do not know whether the rectilinear motion can still be well estimated. In a practical implementation, the reader is recommended to extend the idea of normalization in this paper to trilinear constraints or even to an uncalibrated camera.

In both this chapter and the previous one, we use the generic Newton's algorithm to minimize the normalized epipolar constraint. One disadvantage is that it is slower than most gradient based algorithms, such as the commonly used Levenberg-Marquardt algorithm. For this reason, we recommend the reader to use those algorithms instead for practical implementations. We here outlined the Newton's algorithm to demonstrate how to compute all the necessary geometric entities associated to the optimization.

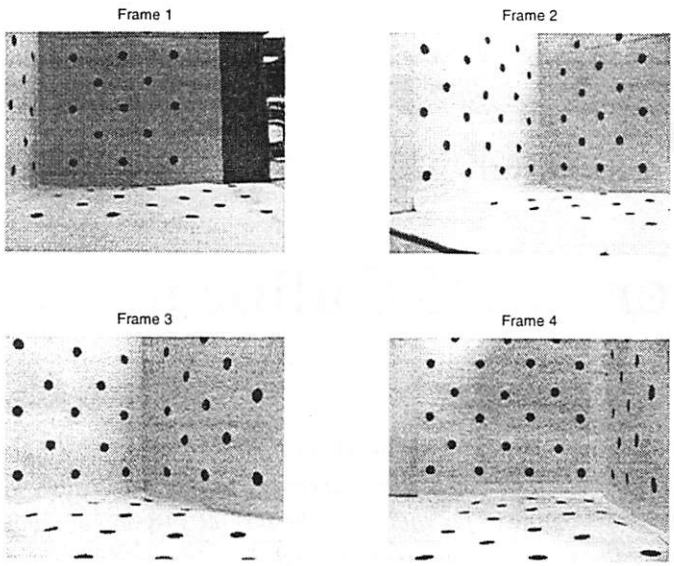


Figure 5.6: Four images of a cubic corner taken by the camera.

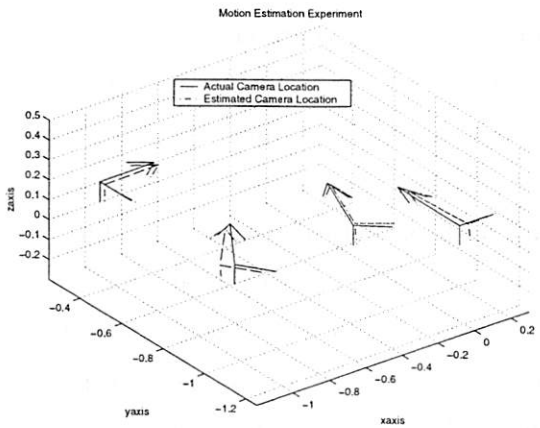


Figure 5.7: Comparison of estimated and measured camera configuration for the four images.

Chapter 6

Camera Self-Calibration

"Thus arises the problem of seeking out the simplest data from which the metric relations of Space can be determined, a problem which by its very nature is not completely determined, for there may be several systems of simple data which suffice to determine the metric relations of Space;..."

— G. F. B. Riemann, *On the Hypotheses Which Lie at the Foundations of Geometry*

The problem of camera self-calibration refers to the problem of obtaining intrinsic parameters of a camera using only information from image measurements, without any *a priori* knowledge about the motion between frames and the structure of the observed scene. The general calibration problem is motivated by a variety of applications in mobile robot navigation and control using on-board computer vision system as a motion sensor. Many navigation or control tasks, such as target tracking, obstacle avoidance or map building, require the knowledge of both the camera (or the object) motion and a full Euclidean structure of the environment, which is possible only when the intrinsic parameters of the camera are known. Both theoretical studies as well as practical algorithms of camera self-calibration have recently received an increased interest in the computer vision and robotics community. The appeal of a successful solution to the camera self-calibration problem lies in the elimination of the need for an external calibration object [118] as well as the possibility of on-line calibration of time-varying internal parameters of the camera. The latter feature is of great importance for active vision systems. The majority of the camera self-calibration in the computer vision literature have been derived in a projective geometry framework. Here, we redevelop the theory in a differential geometric framework which enables not only new perspectives and algorithms but also a resolution of some mistreated

problems in self-calibration.

The original problem of determining whether the image measurements "only" are sufficient for obtaining the information about intrinsic parameters of the camera has been answered in the computer vision context by [77]. The proposed approach and solution utilize invariant properties of the image of the so called absolute conic. Since the absolute conic is invariant under Euclidean transformations (*i.e.*, its representation is independent of the position of the camera) and depends only on the camera intrinsic parameters, the recovery of the image of the absolute conic is then equivalent to the recovery of the camera intrinsic parameter matrix. The constraints on the absolute conic are captured by the so called Kruppa's equations derived by Kruppa in 1913 [58].

The derivation of the Kruppa's equations was mainly developed in a projective geometry framework and its understanding required good intuition of the projective geometric entities (with the exception of [35]). This derivation is quite involved and the development appears to be rather unnatural since, both the constraints captured by Kruppa's equations and the image of (dual) absolute conic are in fact directly linked to the invariants of the group of Euclidean transformation (rather than projective transformation). We here provide an alternative derivation of Kruppa's equations, which in addition to being concise and elegant, also provides an intrinsic geometric interpretation of the so called fundamental matrices and its associated Kruppa's equations. Such an interpretation is crucial for designing intrinsic optimization schemes for solving the problem (for example, see [72]).

In spite of the fact that the basic formulation of appropriate constraints, such as the Kruppa's equations, is in place and there are many successful applications [136], to our knowledge, there is not yet a clear understanding of the geometry of an uncalibrated camera, and there is no complete analysis of the necessary and sufficient condition for a unique solution of the self-calibration problem. This often leads to situations where the algorithms are applied in ill-conditioned settings or where a unique solution is not even obtainable. The differential geometric approach we take in this chapter will allow us to fully understand the intrinsic geometric characterization of an uncalibrated camera and it will easily lead to a clear answer to the questions:

> **(i)** *What is the necessary and sufficient condition for a unique solution of camera self-calibration? Do Kruppa's equations provide sufficient conditions on the camera intrinsic parameters?*

The first question has been previously studied by [104]. However the analysis is incorrect

since it makes a wrong assumption that one can at best recover the structure up to an arbitrary projective transformation from uncalibrated images [38]. Therefore, the results given in [104] are incorrect and have led to a misleading characterization of the necessary and sufficient condition for a unique solution of self-calibration (see Section 6.5.4 and 6.5.3 for a more detailed account). In this chapter, we will give the necessary and sufficient condition in a very clear and compact form. Our results imply that, in principle, one can recover 3D Euclidean motion and structure up to a one parameter family from two uncalibrated images, as opposed to an arbitrary projective transformation [38]. Answer to the second question is unfortunately no, as counter examples have been given in the literature (e.g. [116]). Here we will give a complete account of exactly what is missing in the Kruppa's equations. As we will see, there exist solutions of the Kruppa's equations which do not allow any Euclidean reconstruction of the camera motion and scene structure. After excluding these solutions, solving Kruppa's equations is then equivalent to the necessary and sufficient condition for a unique self-calibration.

One class of approaches to the design of self-calibration algorithms instead of directly using the Kruppa's equations, solves for the entire projection matrices which are compatible with the camera motion and structure of the scene [36]. Such methods suffer severely from numerous local minima. Another class of approaches, as we have mentioned, directly utilizes the Kruppa's equations which provide quadratic constraints in the camera intrinsic parameters. The so called epipolar constraint between a pair of images provides 2 such constraints, hence it usually requires the total of 3 pairs of images for a unique solution of all the 5 unknown parameters. The solution proposed to solve the Kruppa's equations in the literature using homotopy continuation is quite computationally expensive and requires a good accuracy of the measurements [77]. Some alternative schemes have been explored in [62, 138]. It has been well-known that, in presence of noise, these Kruppa's equation based approaches do not usually provide good recovery of the camera calibration [7]. Thus, it is important to answer:

> (ii) *Under what conditions can the Kruppa's equations become degenerate or ill-conditioned? When such conditions are satisfied, how do the self-calibration algorithms need to be modified?*

The answer to the former question is rather unfortunate: for camera motions such that the rotation axis is parallel or perpendicular to the translation, the Kruppa's equations are degenerate (in the sense that constraints provided are dependent); most practical image

sequences are in fact taken through motions close to these two types. This explains why conventional approaches to self-calibration based on the (nonlinear) Kruppa's equations usually fail when being applied to real image sequences. However, we further show in this chapter that when such motions occur, the corresponding Kruppa's equations can be "renormalized" and become linear. This gives us opportunities to design linear self-calibration algorithms besides the pure rotation case [36]. Our study also clarifies some incorrect analysis and results that exist in the literature regarding the solutions of the Kruppa's equations [138]. This is discussed in Section 6.5.2.

From previous chapters, we know that it is possible to develop a parallel set of theory and algorithms for recovering camera motion and scene structure for the discrete and continuous cases. We therefore ask:

> **(iii)** *Whether there is a parallel theory and a set of algorithms of self-calibration for the discrete and continuous cases?*

The answer is unfortunately no, as was previously pointed out by [9]. Due to certain degeneracy of the continuous epipolar constraint, it is in general impossible to obtain a full calibration from it while, for the discrete case, full information of camera calibration is already available from the epipolar constraint only. In this chapter, similarities and differences between the discrete and continuous cases are unified in the same geometric framework.

Chapter Outline

Section 6.1 studies the geometry of an uncalibrated camera system. It gives an intrinsic geometric interpretation of the camera self-calibration problem. As a theoretical foundation for the design of self-calibration algorithms, geometric invariants associated to an uncalibrated camera are studied in detail in Section 6.2. In particular, we show that the (dual) absolute conics are generated by these basic invariants. Section 6.3 reviews the epipolar geometry in the uncalibrated case. Based on invariant theory, Section 6.4 provides a geometric characterization of the space of fundamental matrices. This characterization naturally associates the Kruppa's equations with basic invariants of the uncalibrated camera. In Section 6.5, we then study the solvability of Kruppa's equations. Several important cases which allow for linear self-calibration algorithms are presented. These cases also reveal difficulties in the conventional Kruppa's equation based approaches. Section 6.6 provides a

brief study of the continuous case, as a comparison to the theory of the discrete case. Some preliminary experiments of proposed algorithms are presented in Section 6.7.

6.1 Geometry of an Uncalibrated Camera

Before trying to solve the camera self-calibration problem, we first need to know some geometric properties of an uncalibrated camera: we will see that the study of an uncalibrated camera is equivalent to that of a calibrated camera in a (Euclidean) space with an unknown metric. Further, the problem of recovering the calibration matrix A is mathematically equivalent to that of recovering this unknown metric. Consequently, the camera intrinsic parameters given in (2.18) can be geometrically characterized as the space $SL(3)/SO(3)$. Some elementary Riemannian geometry notation will be used here. For good references on Riemannian geometry, we refer the reader to [5, 55, 103].

Let $\mathbb{E}^3$ be the three dimensional Euclidean space (isometric to $\mathbb{R}^3$). Consider a map ψ from $\mathbb{E}^3$ to itself:

$$\begin{aligned} \psi : \mathbb{E}^3 &\rightarrow \mathbb{E}^3 \\ \mathbf{X} &\mapsto \mathbf{X}' = A\mathbf{X} \end{aligned}$$

where $\mathbf{X}$ and $\mathbf{X}'$ are 3 dimensional coordinates of the points $p \in \mathbb{E}^3$ and $p' = \psi(p) \in \mathbb{E}^3$ respectively. Then ψ is the transformation from the *calibrated* space to the *uncalibrated* space. To differentiate these two spaces, we will use a prime on the entities associated to the uncalibrated space, unless it is clear from the context. Let $\Phi(\cdot,\cdot)$ to be the standard Euclidean metric on $\mathbb{E}^3$. Then the map ψ induces a new metric $\Phi'(\cdot,\cdot)$ on $\mathbb{E}^3$ as following:

$$\Phi'(u,v) = \Phi(\psi^{-1}(u), \psi^{-1}(v)) = u^T A^{-T} A^{-1} v, \quad \forall u, v \in T_{p'}\mathbb{E}^3, \quad \forall p' \in \mathbb{E}^3. \tag{6.1}$$

We define the symmetric matrix $S \in \mathbb{R}^{3\times 3}$ associated to the matrix A as:

$$S = A^{-T} A^{-1}. \tag{6.2}$$

Then the metric $\Phi'(\cdot,\cdot)$ is determined by the matrix S. Let $\mathbb{K} \subset SL(3)$ be the subgroup of $SL(3)$ which consists of all upper-triangular matrices. That is, any matrix $A \in \mathbb{K}$ has the form:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}. \tag{6.3}$$

Note that if A is upper-triangular, so is A^{-1}. Clearly, there is a one-to-one correspondence between $\mathbb{K}$ and the set of all upper-triangular matrices of the form given in (2.18); also the equation (6.2) gives a finite-to-one correspondence between $\mathbb{K}$ and the set of all 3×3 symmetric matrices with determinant 1 (by the *Cholesky factorization*). Usually, only one of the upper-triangular matrices corresponding to the same symmetric matrix has a physical interpretation as the calibration of a camera. Thus, if the calibration matrix A does have the form given by (2.18), the self-calibration problem is equivalent to the problem of recovering the matrix S, *i.e.*, the new metric $\Phi'(\cdot,\cdot)$ of the uncalibrated space.

Now let us consider the general case that the uncalibrated camera is characterized by an arbitrary matrix $A \in SL(3)$. A has the QR-decomposition:

$$A = QR, \quad Q \in \mathbb{K}, R \in SO(3). \tag{6.4}$$

Then $A^{-1} = R^T Q^{-1}$ and the associated symmetric matrix $S = A^{-T}A^{-1} = Q^{-T}Q^{-1}$. In general, if $A = BR$ with $A, B \in SL(3), R \in SO(3)$, the $A^{-T}A^{-1} = B^{-T}B^{-1}$. That is A and B induces the same metric on the uncalibrated space. In this case, we say that matrices A and B are **equivalent**. The quotient space $SL(3)/SO(3)$ will be called the **intrinsic parameter space**. It gives an "intrinsic-indeed" interpretation for the camera intrinsic parameters given in (2.18). This will be explained in more detail in the rest of this section.

We contend that, without knowing camera motion and scene structure, the matrix $A \in SL(3)$ can only be recovered up to an equivalence class $\bar{A} \in SL(3)/SO(3)$. To see this, suppose $B \in SL(3)$ is another matrix in the same equivalence class as A. Then $A = BR_0$ for some $R_0 \in SO(3)$. The coordinate transformation (2.7) yields:

$$A\mathbf{X}(t) = AR(t)\mathbf{X}(t_0) + AT(t) \quad \Leftrightarrow \quad BR_0\mathbf{X}(t) = BR_0R(t)R_0^T R_0\mathbf{X}(t_0) + BR_0T(t). \tag{6.5}$$

Notice that the conjugation:

$$\begin{aligned} \mathrm{Ad}_r : SE(3) &\rightarrow SE(3) \\ h &\mapsto rhr^{-1} \end{aligned}$$

is a group homomorphism where $r = \begin{bmatrix} R_0 & 0 \\ 0 & 1 \end{bmatrix}$. Then from the equation (6.5), there is no way that one can tell an uncalibrated camera with calibration matrix A undergoing the motion $(R(t), T(t))$ and observing the point $p \in \mathbb{E}^3$ from another uncalibrated camera with calibration matrix B undergoing the motion $(R_0R(t)R_0^T, R_0T(t))$ and observing the point

$R_0 p \in \mathbb{E}^3$. The effect of R_0 is nothing but a *rotation* of the overall configuration space. We will soon see that this property naturally shows up in the fundamental matrix (to be introduced) when we study epipolar constraint for the uncalibrated case.

Therefore, without knowing camera motion and scene structure, the matrix A associated with an uncalibrated camera can only be recovered up to an equivalence class $\bar{A}$ in the space $SL(3)/SO(3)$. The subgroup $\mathbb{K}$ of all upper-triangular matrices in $SL(3)$ is one representation of such a space, as is the space of 3×3 symmetric matrices with determinant 1. Thus, $SL(3)/SO(3)$ does provide an intrinsic geometric interpretation for the unknown camera parameters. In general, the problem of camera self-calibration is then equivalent to the problem of recovering the symmetric matrix $S = A^{-T}A^{-1}$, *i.e.*, the new metric $\Phi'(\cdot,\cdot)$, from which the upper-triangular representation of the intrinsic parameters can be easily obtained from the *Cholesky factorization.*

The space $\mathbb{E}^3$ with the new metric $\Phi'(\cdot,\cdot)$ is still a Euclidean space. Nevertheless, without knowing this metric, we do not know how to transform the chosen coordinate charts of the uncalibrated camera back to an orthonormal one. That is, the space $\mathbb{E}^3$ is now uncalibrated. From (2.7), the coordinate transformation in the uncalibrated space is given by:

$$A\mathbf{X}(t) = AR(t)\mathbf{X}(t_0) + AT(t) \quad \Leftrightarrow \quad \mathbf{X}'(t) = AR(t)A^{-1}\mathbf{X}'(t_0) + T'(t) \tag{6.6}$$

where $\mathbf{X}' = A\mathbf{X}$ and $T' = AT$. In homogeneous coordinates, the transformation group on the uncalibrated space is given by:

$$G' = \left\{ \begin{bmatrix} ARA^{-1} & T' \\ 0 & 1 \end{bmatrix} \,\middle|\, T' \in \mathbb{R}^3, R \in SO(3) \right\} \subset \mathbb{R}^{4\times 4} \tag{6.7}$$

It is direct to check that the metric $\Phi'(\cdot,\cdot)$ is invariant under the action of G'. Thus G' is a subgroup of the isometry group[1] of the uncalibrated space. If the motion of a calibrated camera in the uncalibrated space is given by $g'(t) \in G', t \in \mathbb{R}$, the homogeneous coordinates of a point $p' \in \mathbb{E}^3$ satisfy:

$$p'(t) = g'(t)p'(t_0). \tag{6.8}$$

From the calibrated camera model, the image of the point p' with respect to a calibrated

[1]The isometry group of a manifold M is the set of all transformations which preserve its Riemannian metric.

camera is given by:

$$\lambda \mathbf{x} = Pp'. \tag{6.9}$$

It is then direct to check that this image is the same as the image of $p = \psi^{-1}(p') \in \mathbb{E}^3$ with respect to the uncalibrated camera, *i.e.*, we have:

$$\lambda \mathbf{x} = APp. \tag{6.10}$$

From this property, the problem of camera self-calibration is indeed equivalent to the problem of recovering the unknown metric $\Phi'(\cdot, \cdot)$ of the uncalibrated space assuming a calibrated camera.

6.2 Geometric Invariants of an Uncalibrated Camera

Since isometric transformation (group) G' preserves the metric $\Phi'(\cdot, \cdot)$, invariants preserved by such transformation are therefore keys to recover such a metric. This section will give a complete account of these invariants. Although the explicit form of the metric $\Phi'(\cdot, \cdot)$ is unknown, we know the uncalibrated space is isomorphic to the standard Euclidean space through an isomorphism ψ. Thus the invariants of the uncalibrated space under its isometry group G' are in one-to-one correspondence to the invariants of the Euclidean group. The complete list of Euclidean invariants is given by the following proposition:

Proposition 6.1 (Euclidean Invariants). *For a n dimensional vector space V, a complete list of basic invariants of the group $SO(n)$ consists of (1) the inner product $\Phi(u, v) = u^T v$ of two vectors $u, v \in V$ and (2) the determinant* $\det[u^1, \ldots, u^n]$ *of n vectors $u^1, \ldots, u^n \in V$.*

See [134] for a proof of this proposition. Then the set of all Euclidean invariants is the algebra generated by these two types of basic invariants. In the uncalibrated camera case, we have:

Corollary 6.2 (Invariants of an Uncalibrated Camera). *For the space $\mathbb{E}^3$ with the metric $\Phi'(\cdot, \cdot)$, a complete list of basic invariants of the isometry group G' consists of (1) the inner product $\Phi'(u, v) = u^T A^{-T} A^{-1} v$ of two vectors $u, v \in T\mathbb{E}^3$ and (2) the determinant* $\det[A^{-1}u^1, A^{-1}u^2, A^{-1}u^3]$ *of three vectors $u^1, u^2, u^3 \in T\mathbb{E}^3$.*

Then the set of invariants associated to an uncalibrated camera is the algebra generated by these two types of basic invariants. Since

$$\det[A^{-1}u^1, A^{-1}u^2, A^{-1}u^3] \quad = \quad \det(A^{-1}) \cdot \det[u^1, u^2, u^3],$$

it follows that the invariant $\det[A^{-1}u^1, A^{-1}u^2, A^{-1}u^3]$ is independent of the matrix A. Therefore the determinant type invariant is useless for recovering the unknown matrix A and only the inner product type invariant can be helpful.

For any n-dimensional vector space V, its **dual space** V^* is defined to be the vector space of all linear functions on V. An element in V^* is called a **covector**. If $e^i, i = 1, \ldots, n$ are a basis for V, then the set of linear functions $e_j, j = 1, \ldots, n$ defined as:

$$e_j(e^i) = \delta_{ij} \tag{6.11}$$

form a (dual) basis for the dual space V^*. The **pairing** between V and its dual V^* is defined to be the bilinear map:

$$< \cdot, \cdot >: V^* \times V \quad \rightarrow \quad \mathbb{R} \tag{6.12}$$

$$(\xi, u) \quad \mapsto \quad \xi(u). \tag{6.13}$$

If we use the coordinate vector $\xi = [\alpha_1, \ldots, \alpha_n]^T \in \mathbb{R}^n$ to represent a covector $\xi = \sum_{j=1}^n \alpha_j e_j \in V^*, \alpha_j \in \mathbb{R}$, and similarly, $u = [\beta_1, \ldots, \beta_n]^T \in \mathbb{R}^n$ to represent $u = \sum_{i=1}^n \beta_i e^i \in V, \beta_i \in \mathbb{R}$ (note that we use column vector convention for both vectors and covectors), then with respect to the chosen bases the pairing is given by:

$$< \xi, u >= \xi^T u.$$

For a linear transformation $f : V \rightarrow V$, its dual is defined to be the linear transformation $f^* : V^* \rightarrow V^*$ which preserves the pairing:

$$< u, f(v) >=< f^*(u), v >, \quad \forall u \in V^*, v \in V. \tag{6.14}$$

Let $A \in \mathbb{R}^{n \times n}$ be the matrix representing f with respect to the basis $e^i, i = 1, \ldots, n$. Since:

$$< u, f(v) >= u^T A v = (A^T u)^T v, \tag{6.15}$$

it follows that the dual f^* is represented by A^T with respect to the (dual) basis $e_j, j = 1, \ldots, n$.

The invariants given in Corollary 6.2 are invariants of the vector space $T\mathbb{E}^3 \cong \mathbb{R}^3$ under the action of the isotropy subgroup $ASO(3)A^{-1}$ (here we identify an element in $ASO(3)A^{-1}$ with its differential map since everything is linear). As we know from above, this group action induces a dual action on the dual space of $T\mathbb{E}^3$, denoted by $T^*\mathbb{E}^3$. This dual action can then be represented by the transpose group $A^{-T}SO(3)A^T$ since

$$(ARA^{-1})^T = A^{-T}R^TA^T \quad \in A^{-T}SO(3)A^T$$

for all $R \in SO(3)$. We call invariants associated with this dual group action on the covectors as **coinvariants**. Consequently we have:

Corollary 6.3 (Coinvariants of an Uncalibrated Camera). *For the space $\mathbb{E}^3$ with the metric $\Phi'(\cdot,\cdot)$, a complete list of basic coinvariants of the isometry group G' consists of (1) the induced inner product $\xi^TAA^T\eta$ of two covectors $\xi,\eta \in T^*\mathbb{E}^3$ and (2) the determinant* $\det[\xi_1,\xi_2,\xi_3]$ *of three covectors $\xi_1,\xi_2,\xi_3 \in T^*\mathbb{E}^3$.*

Note that in the above we use the convention that vectors are enumerated by superscript and covectors by subscript. One may also refer to Weyl [134] or Goodman and Wallach [31] for a detailed study of polynomial invariants of classical groups – Corollary 6.2 and 6.3 can then be deduced from the First Fundamental Theorem of groups $G \subset GL(V)$ preserving a non-degenerate (symmetric) form (see [31]). Note that the induced inner product on $T^*\mathbb{E}^3$ is given by the symmetric matrix $S^{-1} = AA^T$, the inverse of $S = A^{-T}A^{-1}$. In terms of projective geometry, S and S^{-1} define two conics *dual* to each other.

We next want to show that the so called **absolute conic** (or the dual absolute conic) is actually a special invariant generated by inner product type invariants (or coinvariants). In the projective geometry approach to camera self-calibration, the absolute conic plays an important role. In order to give a rigorous definition of the absolute conic, we need to introduce the space $\mathbb{CP}^3$, the three dimensional complex projective space[2]. Let $p = [p_1,p_2,p_3,p_4]^T \in \mathbb{C}^4$ be the homogeneous representation of a point p in $\mathbb{CP}^3$. Then the absolute conic, denoted by Ω, is defined to be the set of points in $\mathbb{CP}^3$ satisfying:

$$p_1^2 + p_2^2 + p_3^2 = 0, \quad p_4 = 0 \tag{6.16}$$

[2] $\mathbb{CP}^3$ is the space of all one dimensional (complex) subspaces in $\mathbb{C}^4$, *i.e.*, the quotient space $\mathbb{C}^4 / \sim$ where the equivalence relation $\sim$ is: $[z_1,z_2,z_3,z_4]^T \sim [z\cdot z_1, z\cdot z_2, z\cdot z_3, z\cdot z_4]^T$ for all $z \neq 0$.

Note that this set is invariant under the complex Euclidean group:

$$E(3,\mathbb{C}) = \left\{ \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \Big| \, T \in \mathbb{C}^3, R \in U(3) \right\} \subset \mathbb{C}^{4\times 4} \tag{6.17}$$

where $U(3)$ is the space of all (complex) 3×3 unitary matrices. The special Euclidean group $SE(3)$ is just a subgroup of $E(3,\mathbb{C})$ hence the absolute conic is invariant under $SE(3)$ as well.

For any $p = [p_1, p_2, p_3, p_4]^T \in \Omega$, suppose

$$p_j = u_j + iv_j, \quad u_j, v_j \in \mathbb{R}, \quad j = 1, \ldots, 4 \tag{6.18}$$

where $i = \sqrt{-1}$. Since $u_4 = v_4 = 0$, we obtain a pair of vectors $u = [u_1, u_2, u_3, 0]^T$ and $u = [v_1, v_2, v_3, 0]^T$ of the 3 dimensional (real) Euclidean space $\mathbb{E}^3$ (in homogeneous representation). From (6.16), these two vectors satisfy:

$$u^T u = v^T v, \quad u^T v = 0 \tag{6.19}$$

On the other hand, any pair of vectors $u, v \in T\mathbb{E}^3$ which satisfy the above conditions (*i.e.*, u and v are orthogonal to each other and have the same length) define a point on the absolute conic Ω. Therefore, the absolute conic Ω is the same as the set of all pairs of such vectors. Since all the inner product type quantities in (6.19) are invariant under the Euclidean group $SE(3)$, the absolute conic Ω is simply generated by these basic invariants.

In the uncalibrated camera case, if we let $S = A^{-T}A^{-1}$ and $p' = [p'_1, p'_2, p'_3, p'_4]^T \in \mathbb{C}^4$, the corresponding absolute conic (6.16) is then given by:

$$[p'_1, p'_2, p'_3] S [p'_1, p'_2, p'_3]^T = 0, \quad p'_4 = 0. \tag{6.20}$$

Therefore, the camera self-calibration problem is also equivalent to the problem of recovering this absolute conic (for example see Maybank [76]). It is direct to check that this absolute conic is generated by basic invariants given in Corollary 6.2. Define the dual absolute conic Ω^* to be the set of points in $\mathbb{CP}^3$ satisfying:

$$[p'_1, p'_2, p'_3] S^{-1} [p'_1, p'_2, p'_3]^T = 0, \quad p'_4 = 0. \tag{6.21}$$

Similarly, one can show that it is generated by the inner product type coinvariants given in Corollary 6.3.

6.3 Epipolar Constraint in the Uncalibrated Case

Before we can apply the invariant theory given in the previous section to the problem of camera self-calibration, we first need to know what quantities we can directly obtain from images and what type of geometric entities they are.

The epipolar (or Longuet-Higgins) constraint plays an important role in the study of the geometry of calibrated cameras. In this section, we study its uncalibrated version. From (6.6), for a point $p' \in \mathbb{E}^3$ we have:

$$\begin{aligned} & \mathbf{X}'(t) = AR(t)A^{-1}\mathbf{X}'(t_0) + T'(t) \quad \Rightarrow \quad T'(t) \times \mathbf{X}'(t) = T'(t) \times AR(t)A^{-1}\mathbf{X}'(t_0) \\ \Rightarrow \quad & \mathbf{X}'(t)^T \widehat{T'(t)} AR(t)A^{-1}\mathbf{X}'(t_0) = 0. \end{aligned} \tag{6.22}$$

Let $\mathbf{x}_1 \in \mathbb{R}^3$ and $\mathbf{x}_2 \in \mathbb{R}^3$ be images of p' at time t_0 and t respectively, *i.e.*, there exist $\lambda_1, \lambda_2 > 0$ such that $\lambda_1 \mathbf{x}_1 = \mathbf{X}'(t_0)$ and $\lambda_2 \mathbf{x}_2 = \mathbf{X}'(t)$. To simplify the notation, we will drop the time dependence from the motion $(AR(t)A^{-1}, T'(t))$ and simply denote it by (ARA^{-1}, T'). Then (6.22) yields:

$$\mathbf{x}_2^T \widehat{T'} ARA^{-1}\mathbf{x}_1 = 0. \tag{6.23}$$

Note that in the above equation the matrix:

$$F_1 = \widehat{T'} ARA^{-1} \quad \in \mathbb{R}^{3\times 3} \tag{6.24}$$

is of particular interest – it contains useful information about camera intrinsic parameters as well as the motion of camera.

Recall that the motion (ARA^{-1}, T') in the uncalibrated space is equivalent to the motion (R, T) in the calibrated space, with $T = A^{-1}T'$. Also from (6.6), we have:

$$\begin{aligned} & A^{-1}\mathbf{X}'(t) = R(t)A^{-1}\mathbf{X}'(t_0) + T(t) \quad \Rightarrow \quad T(t) \times A^{-1}\mathbf{X}'(t) = T(t) \times R(t)A^{-1}\mathbf{X}'(t_0) \\ \Rightarrow \quad & \mathbf{X}'(t)^T A^{-T} \widehat{T(t)} R(t) A^{-1}\mathbf{X}'(t_0) = 0 \end{aligned} \tag{6.25}$$

We then have a second form for the constraint given in (6.23):

$$\mathbf{x}_2^T A^{-T} \widehat{T} RA^{-1}\mathbf{x}_1 = 0. \tag{6.26}$$

The matrix

$$F_2 = A^{-T} \widehat{T} RA^{-1} \quad \in \mathbb{R}^{3\times 3} \tag{6.27}$$

is called the **fundamental matrix** in the Computer Vision literature. When $A = I$, the fundamental matrix simply becomes $\widehat{T}R$ which is exactly the **essential matrix** E that we have studied extensively in previous chapters. In fact, the two constraints (6.23) and (6.26) are equivalent and they are both called the **epipolar constraint** for the uncalibrated case. We prove this by showing that the two matrices F_1 and F_2 are actually equal.

Lemma 6.4 (The Hat Operator). *If $T \in \mathbb{R}^3$, $A \in SL(3)$ and $T' = AT$, then $\widehat{T} = A^T \widehat{T'} A$.*

Proof: Since both $A^T\widehat{(\cdot)}A$ and $\widehat{A^{-1}(\cdot)}$ are linear maps from $\mathbb{R}^3$ to $\mathbb{R}^{3\times 3}$, using the fact that $\det(A) = 1$, one may directly verify that these two linear maps are equal on the bases: $[1,0,0]^T, [0,1,0]^T$ or $[0,0,1]^T$. ∎

This simple lemma will be frequently used throughout the paper. By this lemma, we have:

$$F_2 = A^{-T}\widehat{T}RA^{-1} = A^{-T}\widehat{T}A^{-1}ARA^{-1} = \widehat{T'}ARA^{-1} = F_1. \tag{6.28}$$

We then can denote F_1 and F_2 by the same notation F. Define the space of fundamental matrices associated to $A \in SL(3)$ as:

$$\mathcal{F} = \{A^{-T}\widehat{T}RA^{-1} \mid R \in SO(3), T \in \mathbb{R}^3\}. \tag{6.29}$$

The space $\mathcal{F}$ is also called **fundamental space**. Note that $\mathcal{F} = A^{-T}\mathcal{E}A^{-1}$.

In the preceding section, we have shown that if two matrices A and B are in the same equivalence class of $SL(3)/SO(3)$, we are not able to tell them apart only from images. We may assume $B = AR_0$ for some $R_0 \in SO(3)$. Then with the same camera motion (R, T), the fundamental matrix associated with B is:

$$B^{-T}\widehat{T}RB^{-1} = A^{-T}R_0\widehat{T}RR_0^TA^{-1} = A^{-T}\widehat{R_0T}(R_0RR_0^T)A^{-1}. \tag{6.30}$$

As we noticed, the essential matrix $\widehat{T}R$ is simply replaced by another essential matrix $\widehat{R_0T}(R_0RR_0^T)$. Therefore, without knowing actual camera motion, only from the fundamental matrix, one cannot tell camera B from camera A.

6.4 Geometric Characterization of the Space of Fundamental Matrices

In this section, we give a geometric characterization of the space of fundamental matrices. It will be shown that this space can be naturally identified with the cotangent bundle of the matrix Lie group $A^{-T}SO(3)A^T$, therefore, fundamental matrices by their nature can be viewed as covectors. This characterization is quite different from the conventional way of characterizing fundamental matrices as a degenerate matrix which represents the epipolar map between two image planes (for example see [62]), but it directly connects a fundamental matrix with its related Kruppa's equation, as we will soon see in Section 6.5.

We define a metric $\Psi(\cdot,\cdot)$ on the space $\mathbb{R}^{3\times3}$ as:

$$\Psi(B,C) \;=\; \mathrm{tr}(BSC^T), \quad \forall B,C \in \mathbb{R}^{3\times3} \tag{6.31}$$

where $S = A^{-T}A^{-1}$. It is direct to check that so defined g is indeed a metric. This metric may be used to identify the space $\mathbb{R}^{3\times3}$ with its dual $(\mathbb{R}^{3\times3})^*$ (the space of linear functions on $\mathbb{R}^{3\times3}$). In other words, under this identification, given a matrix $B \in \mathbb{R}^{3\times3}$, we may identify it as a member in the dual space $(\mathbb{R}^{3\times3})^*$ through:

$$\begin{aligned} f : \mathbb{R}^{3\times3} &\;\to\; (\mathbb{R}^{3\times3})^* \\ B &\;\mapsto\; B^* = \Psi(B,\cdot). \end{aligned}$$

From the metric definition (6.31), B^* can be represented in the matrix form as $B^* = BS$ (with respect to the standard Euclidean metric on $\mathbb{R}^{3\times3}$). Since S is non-degenerate, the map f is an isomorphism and it induces a metric on the dual space as follows:

$$\Psi^*(B^*,C^*) \;=\; \Psi(B,C) \;=\; \mathrm{tr}(B^*S^{-1}(C^*)^T). \tag{6.32}$$

A tangent vector of the Lie group $A^{-T}SO(3)A^T$ has the form $A^{-T}\widehat{T}RA^T \in \mathbb{R}^{3\times3}$ where $R \in SO(3)$ and $T \in \mathbb{R}^3$. By restricting this metric to the tangent space of $A^{-T}SO(3)A^T$, *i.e.*, $T(A^{-T}SO(3)A^T)$, the metric Ψ induces a metric on the Lie group $A^{-T}SO(3)A^T$:

$$\Psi(A^{-T}\widehat{T}_1RA^T, A^{-T}\widehat{T}_2RA^T) = \Psi(A^{-T}\widehat{T}_1A^T, A^{-T}\widehat{T}_2A^T). \tag{6.33}$$

The equality shows that this induced metric on the Lie group $A^{-T}SO(3)A^T$ is **right invariant**.

The cotangent vector corresponding to the tangent vector $A^{-T}\widehat{T}A^T$ is given by:

$$(A^{-T}\widehat{T}RA^T)^* = A^{-T}\widehat{T}RA^TS = A^{-T}\widehat{T}RA^{-1}. \tag{6.34}$$

Note that the matrix $A^{-T}\widehat{T}RA^{-1}$ is the exact form of a fundamental matrix. Therefore, the space of all fundamental matrices can be identified with the cotangent space of the Lie group $A^{-T}SO(3)A^T$, *i.e.*, $T^*(A^{-T}SO(3)A^T)$. There is an induced metric on the cotangent space:

$$\Psi^*(A^{-T}\widehat{T}_1RA^{-1}, A^{-T}\widehat{T}_2RA^{-1}) = \Psi^*(\widehat{T_1'}, \widehat{T_2'}) \tag{6.35}$$

where $T_1' = AT_1$ and $T_2' = AT_2$. Since a fundamental matrix can only be determined up to scale, we may consider the unit cotangent bundle $T_1^*(A^{-T}SO(3)A^T)$. Define the space of unit fundamental matrices to be:

$$\mathcal{F}_1 = \{A^{-T}\widehat{T}RA^{-1} \mid R \in SO(3), T \in \mathbb{R}^3, \Psi^*(\widehat{AT}, \widehat{AT}) = 1\}. \tag{6.36}$$

The space $\mathcal{F}_1$ is also called **unit fundamental space**. The relation between the unit fundamental space $\mathcal{F}_1$ and the unit cotangent space $T_1^*(A^{-T}SO(3)A^T)$ is given by:

Theorem 6.5 (Geometric Characterization of Fundamental Space). *The unit cotangent space $T_1^*(A^{-T}SO(3)A^T)$ is a double covering of the unit fundamental space $\mathcal{F}_1$.*

The proof essentially follows from the fact that the unit tangent bundle $T_1(SO(3))$ is a double covering of the normalized essential space $\mathcal{E}_1$, see Appendix A. For a fixed matrix $A \in SL(3)$, the normalized fundamental space $\mathcal{F}_1$ is, same as $\mathcal{E}_1$, a five dimensional connected compact manifold embedded in $\mathbb{R}^{3\times 3}$.

Comment 6.6. *The identification of the fundamental space as the cotangent space of the Lie group $A^{-T}SO(3)A^T$ is only artificial. That is, these two spaces happen to have the same matrix representation. Such an identification by no means implies that the translation T* is by nature *a tangent vector of the rotation R.*

After all the preparation in geometry, we are now ready to investigate possible schemes for recovering the unknown calibration matrix A, or equivalently, the symmetric matrix $S = A^{-T}A^{-1}$.

6.5 Kruppa's Equations

Without loss of generality, we may assume that both the rotation R and translation T are non-trivial, *i.e.*, $R \neq I$ and $T \neq 0$ hence the epipolar constraint (6.23) is not degenerate and the fundamental matrix can be estimated. The camera self-calibration problem is then reduced to recovering the symmetric matrix $S = A^{-T}A^{-1}$ or $S^{-1} = AA^T$ from fundamental matrices. In previous sections, we have shown that, even if we here have chosen A to be an arbitrary element in $SL(3)$, A can only be recovered up to a rotation, *i.e.*, as an element in the quotient space $SL(3)/SO(3)$. Note that $SL(3)/SO(3)$ is only a 5-dimensional space. From the fundamental matrix, the epipole vector p' can be directly computed (up to an arbitrary scale) as the null space of F. Given a fundamental matrix $F = \widehat{T}'ARA^{-1}$, its **scale** (usually denoted as λ) is defined as the norm of T'. If $\lambda = \|T'\| = 1$, such a F is called a **normalized fundamental matrix.**[3] For now, we assume that the fundamental matrix F happens to be normalized.

Suppose the standard basis of $\mathbb{R}^3$ is $e_1 = [1,0,0]^T, e_2 = [0,1,0]^T, e_3 = [0,0,1]^T \in \mathbb{R}^3$. Now pick any rotation matrix $R_0 \in SO(3)$ such that $R_0T' = e_3$. Using Lemma 6.4, we have $\widehat{T}' = R_0^T\widehat{e}_3R_0$. Define matrix $D \in \mathbb{R}^{3\times 3}$ to be:

$$D = R_0F = \widehat{e}_3R_0ARA^{-1} = [-e_2, e_1, 0]^TR_0ARA^{-1}. \tag{6.37}$$

Then D has the form $D = [\xi_1, \xi_2, 0]^T$ with $\xi_1, \xi_2 \in \mathbb{R}^3$ being the first and second row vectors of D. Hence we have $\xi_1 = A^{-T}R^TA^T(-R_0^Te_2)$, $\xi_2 = A^{-T}R^TA^TR_0^Te_1$. Define vectors $\eta_1, \eta_2 \in \mathbb{R}^3$ as $\eta_1 = R_0^Te_1, \eta_2 = -R_0^Te_2$, then it is direct to check that S^{-1} satisfies:

$$\xi_1^TS^{-1}\xi_1 = \eta_2^TS^{-1}\eta_2, \quad \xi_2^TS^{-1}\xi_2 = \eta_1^TS^{-1}\eta_1, \quad \xi_1^TS^{-1}\xi_2 = \eta_1^TS^{-1}\eta_2. \tag{6.38}$$

We thus obtain three homogeneous constraints on the matrix S^{-1}, the inverse of the matrix S. These constraints can be used to compute S^{-1} hence S.

The above derivation is based on the assumption that the fundamental matrix F is normalized, *i.e.*, $\|T'\| = 1$. However, since the epipolar constraint is homogeneous in the fundamental matrix F, it can only be determined up to an arbitrary scale. Suppose λ is the length of the vector $T' \in \mathbb{R}^3$ in $F = \widehat{T}'ARA^{-1}$. Consequently, the vectors ξ_1 and ξ_2 are also scaled by the same λ. Then the ratio between the left and right hand side quantities in each equation of (6.38) is equal to λ^2. This gives two independent constraints on S^{-1},

[3] Here $\|\cdot\|$ represents the standard 2-norm.

the so called **Kruppa's equations** (after its initial discovery by Kruppa in 1913):

$$\lambda^2 = \frac{\xi_1^T S^{-1} \xi_1}{\eta_2^T S^{-1} \eta_2} = \frac{\xi_2^T S^{-1} \xi_2}{\eta_1^T S^{-1} \eta_1} = \frac{\xi_1^T S^{-1} \xi_2}{\eta_1^T S^{-1} \eta_2}. \tag{6.39}$$

Alternative means of obtaining the Kruppa's equations are by utilizing algebraic relationships between projective geometric quantities [77] or via SVD characterization of F [35]. Here we obtain the same equations from a quite different approach. Equation (6.39) further reveals the geometric meaning of the Kruppa ratio: it is the square of the length of the vector T' in the fundamental matrix F. This discovery turns out to be quite useful when we later discuss the renormalization of Kruppa's equations. In general, each fundamental matrix provides *at most two* algebraic constraints on S^{-1}, *if* the two equations in (6.39) are independent. Since the symmetric matrix S has five degrees of freedom, in general *at least three* fundamental matrices are needed to uniquely determine S. But, as we will soon see, this is not the case for many special camera motions.

Comment 6.7. *One must be aware that solving Kruppa's equations for camera calibration is not equivalent to the camera self-calibration problem in the sense that there may exist solutions of Kruppa's equations which are not solutions of a "valid" self-calibration. Given a non-critical set of camera motions, the associated Kruppa's equations do not necessarily give enough constraints to solve for the calibration matrix A. See Section 6.5.3 for a complete account.*

The above derivation of Kruppa's equations is straightforward, but the expression (6.39) depends on a particular rotation matrix R_0 that one chooses – note that the choice of R_0 is not unique. However, there is an even simpler way to get an equivalent expression for the Kruppa's equations in a matrix form. Given a normalized fundamental matrix $F = \widehat{T}'ARA^{-1}$, it is then straightforward to check that $S^{-1} = AA^T$ must satisfy the following equation:

$$FS^{-1}F^T = \widehat{T}'S^{-1}\widehat{T}'^T. \tag{6.40}$$

We call this equation the **normalized matrix Kruppa's equation.** It is readily seen that this equation is equivalent to (6.38). If F is not normalized (since usually we can only estimate it up to a scale), we may always assume it is of the form $F = \lambda\widehat{T}'ARA^{-1}$ with $\|T'\| = 1$ and $\lambda \in \mathbb{R}$ unknown. We then have the **matrix Kruppa's equation:**

$$FS^{-1}F^T = \lambda^2\widehat{T}'S^{-1}\widehat{T}'^T. \tag{6.41}$$

This equation is equivalent to the scalar version given by (6.39) and is independent of the choice of the rotation matrix R_0. In fact, the matrix form reveals that the nature of Kruppa's equations is nothing but the **inner product (co)invariants** that we have studied in Section 6.2.

6.5.1 Solving the Kruppa's Equations

Algebraic properties of Kruppa's equations have been extensively studied (see e.g. [77, 138]). However, conditions on dependency among Kruppa's equations obtained from the fundamental matrix have not been fully discovered. Therefore it is hard to tell in practice whether a given set of Kruppa's equations suffice to guarantee a unique solution for calibration. As we will soon see in this section, for very rich classes of camera motions which commonly occur in many practical applications, the Kruppa's equations will become degenerate. Moreover, since the Kruppa's equations (6.39) or (6.41) are highly nonlinear in S^{-1}, most self-calibration algorithms based on directly solving these equations suffer from being computationally expensive or having multiple local minima [7, 64]. These reasons have motivated us to study the geometric nature of Kruppa's equations in order to gain a better understanding of the difficulties commonly encountered in camera self-calibration. Our attempt to resolve these difficulties will lead to simplified algorithms for self-calibration. These algorithms are linear and better conditioned for these special classes of camera motions.

Given a fundamental matrix $F = \widehat{T}'ARA^{-1}$ with p' of unit length, the normalized matrix Kruppa's equation (6.40) can be rewritten in the following way:

$$\widehat{T}'(S^{-1} - ARA^{-1}S^{-1}A^{-T}R^TA^T)\widehat{T}'^T = 0. \tag{6.42}$$

According to this form, if we define $C = ARA^{-1}$, a linear (Lyapunov) map $\sigma : \mathbb{R}^{3\times3} \to \mathbb{R}^{3\times3}$ as $\sigma : X \mapsto X - CXC^T$, and a linear map $\tau : \mathbb{R}^{3\times3} \to \mathbb{R}^{3\times3}$ as $\tau : Y \mapsto \widehat{T}'Y\widehat{T}'^T$, then the solution S^{-1} of equation (6.42) is exactly the (symmetric real) kernel of the composition map:

$$\tau \circ \sigma : \quad \mathbb{R}^{3\times3} \xrightarrow{\sigma} \mathbb{R}^{3\times3} \xrightarrow{\tau} \mathbb{R}^{3\times3}. \tag{6.43}$$

This interpretation of Kruppa's equations clearly decomposes effects of the rotational and translational parts of the motion: if there is no translation *i.e.*, $p = 0$, then there is no map

τ; if the translation is non-zero, the kernel is enlarged due to the composition with map τ. In general, the symmetric real kernel of the composition map $\tau \circ \sigma$ is 3 dimensional – while the kernel of σ is only 2 dimensional as we will prove below. The solutions for the unnormalized Kruppa's equations are even more complicated due to the unknown scale λ. However, we have the following lemma to simplify things a little bit.

Lemma 6.8. *Given a fundamental matrix $F = \widehat{T}'ARA^{-1}$ with $T' = AT$, a real symmetric matrix $X \in \mathbb{R}^{3\times 3}$ is a solution of $FXF^T = \lambda^2\widehat{T}'X\widehat{T}'^T$ if and only if $Y = A^{-1}XA^{-T}$ is a solution of $EYE^T = \lambda^2\widehat{T}Y\widehat{T}^T$ with $E = \widehat{T}R$.*

Using Lemma 6.4, the proof of this lemma is simply algebraic. This simple lemma, however, states a very important fact: given a set of fundamental matrices $F_i = \widehat{T}_i'AR_iA^{-1}$ with $T_i' = AT_i, i = 1,\ldots,m$, there is a one-to-one correspondence between the set of solutions of the equations:

$$F_iXF_i^T = \lambda_i^2\widehat{T}_i'X\widehat{T}_i'^T, \quad i = 1,\ldots,m. \tag{6.44}$$

and the set of solutions of the equations:

$$E_iYE_i^T = \lambda_i^2\widehat{T}_iY\widehat{T}_i^T, \quad i = 1,\ldots,m \tag{6.45}$$

where $E_i = \widehat{T}_iR_i$ are essential matrices associated to the given fundamental matrices. Note that these essential matrices are determined only by the camera motion. Therefore, the conditions of uniqueness of the solution of Kruppa's equations only depend on the camera motion. Our next task is then to study *how* the solutions of Kruppa's equations depend on the camera motion.

6.5.2 Renormalization and Degeneracy of Kruppa's Equations

From the derivation of the Kruppa's equations (6.39) or (6.41), we observe that the reason why they are nonlinear is that we do not usually know the scale λ. It is then helpful to know under what conditions the matrix Kruppa's equation will have the same solutions as the normalized one, *i.e.*, with λ set to 1. Here we will study two special cases for which we are able to know directly what the missing λ is. The fundamental matrix can then be **renormalized** and we can therefore solve the camera calibration from the normalized matrix Kruppa's equations, which are linear! These two cases are when the rotation axis is parallel or perpendicular to the translation. That is, if the motion is

represented by $(R,T) \in SE(3)$ and the unit vector $u \in \mathbb{R}^3$ is the axis of R,[4] then the two cases are when u is parallel or perpendicular to T. As we will soon see, these two cases are of great theoretical importance: Not only does the calibration algorithm become linear, but it also reveals certain subtleties of the Kruppa's equations and explains when the nonlinear Kruppa's equations are most likely to become ill-conditioned.

Lemma 6.9. *Consider a camera motion $(R,T) \in SE(3)$ where $R = e^{\widehat{u}\theta}$, $\theta \in (0,\pi)$ and the axis $u \in \mathbb{R}^3$ is parallel or perpendicular to T. If $\gamma \in \mathbb{R}$ and positive definite matrix Y are a solution to the matrix Kruppa's equation: $\widehat{T}RYR^T\widehat{T}^T = \gamma^2\widehat{T}Y\widehat{T}^T$ associated to the essential matrix $\widehat{T}R$, then we must have $\gamma^2 = 1$. Consequently, Y is a solution of the normalized matrix Kruppa's equation: $\widehat{T}RYR^T\widehat{T}^T = \widehat{T}Y\widehat{T}^T$.*

Proof: Without loss of generality we assume $\|T\| = 1$. For the parallel case, let $x \in \mathbb{R}^3$ be a vector of unit length in the plane spanned by the row vectors of $\widehat{T}$. All such x lie on a unit circle. There exists $x_0 \in \mathbb{R}^3$ on the circle such that $x_0^T Y x_0$ is maximum. We then have $x_0^T RYR^T x_0 = \gamma^2 x_0^T Y x_0$, hence $\gamma^2 \leq 1$. Similarly, if we pick x_0 such that $x_0^T Y x_0$ is minimum, we have $\gamma^2 \geq 1$. Therefore, $\gamma^2 = 1$. For the perpendicular case, since the rows of $\widehat{T}$ span the subspace which is perpendicular to the vector T, the eigenvector u of R is in this subspace. Thus we have: $u^T RYR^T u = \gamma^2 u^T Y u \Rightarrow u^T Y u = \gamma^2 u^T Y u$. Hence $\gamma^2 = 1$ if Y is positive definite. ∎

Combining Lemma 6.9 and Lemma 6.8, we immediately have:

Theorem 6.10 (Kruppa's Equation Renormalization). *Consider an unnormalized fundamental matrix $F = \widehat{T}'ARA^{-1}$ where $R = e^{\widehat{u}\theta}$, $\theta \in (0,\pi)$ and the axis $u \in \mathbb{R}^3$ is parallel or perpendicular to $T = A^{-1}T'$. Let $e = T'/\|T'\|$. Then if $\lambda \in \mathbb{R}$ and a positive definite matrix S are a solution to the matrix Kruppa's equation: $FS^{-1}F^T = \lambda^2 \widehat{e}S^{-1}\widehat{e}^T$, we must have $\lambda^2 = \|T'\|^2$.*

This theorem claims that, for the two types of special motions considered here, there is no solution for λ in the Kruppa's equation (6.41) besides the true scale of the fundamental matrix. Hence we can decompose the problem into finding λ first and then solving for S or S^{-1}. The following theorem allows to directly compute the scale λ for a given fundamental matrix:

[4] R can always be written of the form $R = e^{\widehat{u}\theta}$ for some $\theta \in [0,\pi]$ and $u \in S^2$.

Theorem 6.11 (Fundamental Matrix Renormalization). *Given an unnormalized fundamental matrix $F = \lambda \widehat{T}' A R A^{-1}$ with $\|T'\| = 1$, if $T = A^{-1}T'$ is parallel to the axis of R, then λ^2 is $\|F^T \widehat{T}' F\|$, and if T is perpendicular to the axis of R, then λ is one of the two non-zero eigenvalues of $F^T \widehat{T}'$.*

Proof: First we prove the parallel case. It is straightforward to check that, in general, $F^T \widehat{T}' F = \lambda^2 \widehat{A R^T T}$. Since the axis of R is parallel to T, we have $R^T T = T$ so that $F^T \widehat{T}' F = \lambda^2 \widehat{T}'$. For the perpendicular case, let $u \in \mathbb{R}^3$ be the axis of R. By assumption $T = A^{-1}T'$ is perpendicular to u. Then there exists $v \in \mathbb{R}^3$ such that $u = \widehat{T} A^{-1} v$. Then it is direct to check that $\widehat{T}' v$ is the eigenvector of $F^T \widehat{T}'$ corresponding to the eigenvalue λ. ∎

Then for these two types of special motions, the associated fundamental matrix can be immediately normalized by being divided by the scale λ. Once the fundamental matrices are normalized, the problem of finding the calibration matrix S^{-1} from normalized matrix Kruppa's equations (6.40) becomes a simple *linear* one! A normalized matrix Kruppa's equation in general imposes *three* linearly independent constraints given by (6.38) on the unknown calibration. However, this is *no longer the case* for the special motions that we are considering here.

Theorem 6.12 (Degeneracy of Kruppa's Equations). *Let us consider the camera motion $(R, T) \in SE(3)$ where $R = e^{\widehat{u}\theta}$ has the angle $\theta \in (0, \pi)$. If the axis $u \in \mathbb{R}^3$ is parallel or perpendicular to T, then the normalized matrix Kruppa's equation: $\widehat{T} R Y R^T \widehat{T}^T = \widehat{T} Y \widehat{T}^T$ imposes only 2 linearly independent constraints on the symmetric matrix Y.*

Proof: For the parallel case, by restricting Y to the plane spanned by the row vectors of $\widehat{T}$, it is a symmetric matrix $\tilde{Y}$ in $\mathbb{R}^{2\times 2}$. The rotation matrix $R \in SO(3)$ restricted to this plane is a rotation $\tilde{R} \in SO(2)$. The normalized matrix Kruppa's equation is then equivalent to $\tilde{R} \tilde{Y} \tilde{R}^T = \tilde{Y}$. Since $0 < \theta < \pi$, this equation imposes exactly 2 constraints on the 3 dimensional space of 2×2 real symmetric matrices. The identity $I_{2\times 2}$ is the only solution. Hence the normalized Kruppa's equation imposes exactly 2 linearly independent constraints on Y.

For the perpendicular case, since u in the plane spanned by the row vectors of $\widehat{T}$, there exist $v \in \mathbb{R}^3$ such that (u, v) form an orthonormal basis of the plane. Then the normalized matrix Kruppa's equation is equivalent to:

$$\widehat{T} R Y R^T \widehat{T}^T = \widehat{T} Y \widehat{T}^T \quad \Leftrightarrow \quad (u, v)^T R Y R^T (u, v) = (u, v)^T Y (u, v). \tag{6.46}$$

Since $R^T u = u$, the above matrix equation is equivalent to two equations $v^T RYu = v^T Yu, v^T RYR^T v = v^T Yv$. These are the only two constraints given by the normalized matrix Kruppa's equation. ∎

According to this theorem, although we can renormalize the fundamental matrix when rotation axis and translation are parallel or perpendicular, we only get two independent constraints from the resulting (normalized) Kruppa's equation corresponding to a single fundamental matrix. Hence for these motions, in general, we still need 3 such fundamental matrices to uniquely determine the unknown calibration. On the other hand, if we do not renormalize the fundamental matrix in these cases and directly use the unnormalized Kruppa's equations (6.39) to solve for calibration, the two nonlinear equations in (6.39) are in fact algebraically dependent! Therefore, one can only get one constraint, as opposed to the expected two, on the unknown calibration S^{-1}. This is summarized in Table 6.1.

Table 6.1: Dependency of Kruppa's equation on angle $\phi \in [0, \pi)$ between the rotation and translation.

Cases	Type of Constraints	# of Constraints on S^{-1}
$(\phi \neq 0)$ and $(\phi \neq \frac{\pi}{2})$	Unnormalized Kruppa's Equation	2
	Normalized Kruppa's Equation	3
$(\phi = 0)$ or $(\phi = \frac{\pi}{2})$	Unnormalized Kruppa's Equation	1
	Normalized Kruppa's Equation	2

Although, mathematically, motion involving translation either parallel or perpendicular to the rotation is only a zero-measure subset of $SE(3)$, they are very commonly encountered in applications: most images sequences are in fact taken by moving the camera around an object in a planar or orbital trajectory, in which case the rotation axis and translation direction are likely perpendicular to each other. Another example is a so called **screw motion**, whose rotation axis and translation are parallel. Such a motion shows up frequently in aerial mobile motion. This observation may explain why self-calibration based on directly solving the Kruppa's equations (6.39) is likely to be ill-conditioned when being applied to real image sequences taken under such motions [7]. To intuitively demonstrate the practical significance of our results, we give an example in Figure 6.1. Our analysis reveals that in these cases, it is crucial to renormalize the Kruppa's equation using Theorem 6.12: once the fundamental matrix or Kruppa's equations are renormalized, not only is one more constraint recovered, but we also obtain linear (normalized) Kruppa's equations.

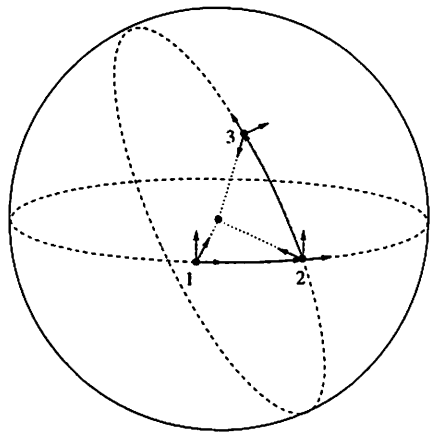


Figure 6.1: Two consecutive orbital motions: even if pairwise fundamental matrices among the three views are considered, one only gets at most $1+1+2=4$ effective constraints on the camera intrinsic matrix if the three matrix Kruppa's equations are *not* renormalized. After renormalization, however, we may get back to $2+2+2 \geq 5$ constraints.

Comment 6.13 (Special Motion Sequences). *Interestingly, for a walking human looking forward, the main rotation of the eyes and the head is yaw and pitch whose axes are perpendicular to the direction of walking. As the theorem suggests, self-calibration in this situation is linear hence more robust to noise. Similar cases can also often be found in vision-guided navigation systems, on-board planar mobile robots. The screw motion, on the other hand, shows up very frequently in motion of aerial mobile robots such as an autonomous helicopter.*

Comment 6.14 (Solutions of the Normalized Kruppa's Equations). *Claims of Theorem 6.12 run contrary to the claims of Propositions B.5 hence B.9 in [138]: In Proposition B.5 of [138], it is claimed that the solutions of the normalized Kruppa's equations when the translation is parallel or perpendicular to the rotation axis are 2 or 3 dimensional. In Theorem 6.12, we claim that the solutions are always 4 dimensional. Theorem 6.12 does not cover the case when the rotation angle θ is π. However, if one allows the rotation to be π, the solutions of normalized Kruppa's equations are even more complicated. For example, we know $e^{\widehat{u}\pi}\widehat{T} = -explanation\widehat{T}$ if u is of unit length and parallel to T (see Lemma 3.1). Therefore, if $R = e^{\widehat{u}\pi}$, the corresponding normalized Kruppa's equation is completely degenerate and imposes no constraints at all on the calibration matrix.*

Comment 6.15 (Number of Solutions). *Although Theorem 6.11 claims that for the perpendicular case λ is one of the two non-zero eigenvalues of $F^T\widehat{T}'$, unfortunately, there*

is no way to tell which one is the right one - simulations show that it could be either the larger or smaller one. Therefore, in a numerical algorithm, for given $n \geq 3$ fundamental matrices, one needs to consider all possible 2^n combinations. According to Theorem 6.10, in the noise-free case, only one of the solutions can be positive definite, which corresponds to the the true calibration.

6.5.3 Kruppa's Equations and Chirality

It is well known that if the scene is *rich enough* (with to come), then the necessary and sufficient condition for a unique camera calibration (see [66]) says that two general motions with rotation along different axes already determine a unique Euclidean solution for camera motion, calibration and scene structure. However, the two Kruppa's equations obtained from these two motions will only give us *at most four* constraints on S, which is not enough to determine S which is of *five* degrees of freedom. We hence need to know what information is missing from the Kruppa's equation. State alternatively, can we get other independent constraints on S from the fundamental matrix?

The proof of Theorem 6.11 suggests another equation can be derived from the fundamental matrix $F = \lambda \widehat{T'} A R A^{-1}$ with $\|T'\| = 1$. Since $F^T \widehat{T'} F = \lambda^2 \widehat{A R^T T}$, we can obtain the vector $\alpha = \lambda^2 A R^T T = \lambda^2 A R^T A^{-1} T'$. Then it is obvious that the following equation for $S = A^{-T} A^{-1}$ holds:

$$\alpha^T S \alpha = \lambda^4 T'^T S T'. \tag{6.47}$$

Notice that this is a constraint on S, not like the Kruppa's equations which are constraints on S^{-1}. Combining the Kruppa's equations given in (6.39) with (6.47) we have:

$$\lambda^2 = \frac{\xi_1^T S^{-1} \xi_1}{\eta_2^T S^{-1} \eta_2} = \frac{\xi_2^T S^{-1} \xi_2}{\eta_1^T S^{-1} \eta_1} = \frac{\xi_1^T S^{-1} \xi_2}{\eta_1^T S^{-1} \eta_2} = \sqrt{\frac{\alpha^T S \alpha}{T'^T S T'}}. \tag{6.48}$$

Is the last equation algebraically independent of the two Kruppa's equations? Although it seems to be quite different from the Kruppa's equations, it is in fact dependent on them, which can be shown either numerically or using simple algebraic tools such as Maple. Thus, it appears that our effort to look for more independent constraints from the fundamental matrix has failed. In the following, we will give an explanation to this by showing that not all S which satisfy the Kruppa's equations may give valid Euclidean reconstructions of both the camera motion and scene structure. The extra constraints which are missing in Kruppa's

equations are in fact captured by the so called *chirality* criteria, which was previously studied in [37]. We now give a clear and concise description between the relationship of the Kruppa's equations and chirality.

Theorem 6.16 (Kruppa's Equations and Chirality). *Consider a camera with calibration matrix I and motion (R,T). If $T \neq 0$, among all the solutions $Y = A^{-1}A^{-T}$ of the Kruppa's equation $EYE^T = \lambda^2\widehat{T}Y\widehat{T}^T$ associated to $E = \widehat{T}R$, only those which guarantee $ARA^{-1} \in SO(3)$ may provide a* **valid Euclidean reconstruction** *of both camera motion and scene structure in the sense that any other solution pushes some plane $N \subset \mathbb{R}^3$ to the plane at infinity, and feature points on different sides of the plane N have different signs in recovered depth.*

Proof: The images $\mathbf{x}_2, \mathbf{x}_1$ of any point $p \in \mathbb{R}^3$ satisfy the coordinates transformation:

$$\lambda_2\mathbf{x}_2 = \lambda_1 R\mathbf{x}_1 + T.$$

If there exists $Y = A^{-1}A^{-T}$ such that $EYE^T = \lambda^2\widehat{T}Y\widehat{T}^T$ for some $\lambda \in \mathbb{R}$, then the matrix $F = A^{-T}EA^{-1} = \widehat{T}'ARA^{-1}$ is also an essential matrix where $T' = AT$, that is, there exists $\tilde{R} \in SO(3)$ such that $F = \widehat{T}'\tilde{R}$ (see [76] for an account of properties of essential matrices). Under the new calibration A, the coordinate transformation is in fact:

$$\lambda_2 A\mathbf{x}_2 = \lambda_1 ARA^{-1}(A\mathbf{x}_1) + T'.$$

Since $F = \widehat{T}'\tilde{R} = \widehat{T}'ARA^{-1}$, we have $ARA^{-1} = \tilde{R} + T'v^T$ for some $v \in \mathbb{R}^3$. Then the above equation becomes: $\lambda_2 A\mathbf{x}_2 = \lambda_1\tilde{R}(A\mathbf{x}_1) + \lambda_1 T'v^T(A\mathbf{x}_1) + T'$. Let $\beta = \lambda_1 v^T(A\mathbf{x}_1) \in \mathbb{R}$, we can further rewrite the equation as:

$$\lambda_2 A\mathbf{x}_2 = \lambda_1\tilde{R}A\mathbf{x}_1 + (\beta + 1)T'. \tag{6.49}$$

Nonetheless, with respect to the solution A, the reconstructed images $A\mathbf{x}_1, A\mathbf{x}_2$ and $(\tilde{R}, T')$ must also satisfy:

$$\gamma_2 A\mathbf{x}_2 = \gamma_1\tilde{R}A\mathbf{x}_1 + T' \tag{6.50}$$

for some scale factors $\gamma_1, \gamma_2 \in \mathbb{R}$. Now we prove by contradiction that $v \neq 0$ is impossible for a valid Euclidean reconstruction. Suppose that $v \neq 0$ and we define the plane $N = \{p \in \mathbb{R}^3 | v^T p = -1\}$. Then for any point $p = \lambda_1 A\mathbf{x}_1 \in N$, we have $\beta = -1$. Hence, from (6.49),

Ax_1, Ax_2 satisfy $\lambda_2 Ax_2 = \lambda_1 \tilde{R} Ax_1$. Since Ax_1, Ax_2 also satisfy (6.50) and $T' \neq 0$, both γ_1 and γ_2 in (6.50) must be ∞. That is, the plane N is "pushed" to the plane at infinity by the solution A. For points not on the plane N, we have $\beta + 1 \neq 0$. Comparing the two equations (6.49) and (6.50), we get $\gamma_i = \lambda_i/(\beta + 1), i = 1, 2$. Then for a point in the far side of the plane N, *i.e.*, $\beta + 1 < 0$, the recovered depth scale γ is negative; for a point in the near side of N, *i.e.*, $\beta + 1 > 0$, the recovered depth scale γ is positive. Thus, we must have that $v = 0$. ∎

Comment 6.17. *Theorem 6.16 essentially implies the chirality constraints studied in [37]. According to the above theorem, if only finite many feature points are measured, a solution of the calibration matrix A which may allow a valid Euclidean reconstruction should induce a plane N not cutting through the convex hull spanned by all the feature points and camera centers.*

As we will soon show in next section that, in general, all A's which make ARA^{-1} a rotation matrix form a one parameter family. Consequently, there is only one A such that both AR_1A^{-1} and AR_2A^{-1} are rotation matrices if R_1 and R_2 are two rotation matrices with independent rotation axes. Theorem 6.16 then implies that the calibration matrix A can be *uniquely* determined with two independent rotations *regardless of translation* if enough feature points are available. An intuitive example is provided in Figure 6.2.

The significance of Theorem 6.16 is that it explains why we get only two constraints from one fundamental matrix even in the two special cases when the Kruppa's equations can be renormalized – extra ones are imposed by the structure, not the motion. The theorem also resolves the discrepancy between the Kruppa's equations and the necessary and sufficient condition for a unique calibration: the Kruppa's equations, although convenient to use, do not provide sufficient conditions for a valid calibration which allows a valid Euclidean reconstruction of both the camera motion and scene structure. However, the fact given in Theorem 6.16 is somewhat difficult to harness in algorithms. For example, in order to exclude invalid solutions, one needs feature points on or beyond the plane N.[5] Alternatively, if such feature points are not available, one may first obtain a **projective reconstruction** and use the so called **absolute quadric constraints** [115] to calibrate the camera. However, in such a method, the camera motions cannot be critical in the

[5]Some possible ways of harnessing the constraints provided by chirality have been discussed in [37]. Basically they give *inequality* constraints on the possible solutions of the calibration.

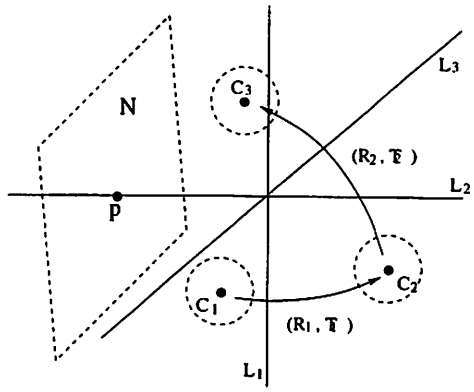


Figure 6.2: A camera undergoes two motions (R_1, T_1) and (R_2, T_2) observing a rig given by the three lines L_1, L_2, L_3. Then the camera calibration is uniquely determined as long as R_1 and R_2 have independent rotation axes and rotation angles in $(0, \pi)$, regardless of T_1, T_2. This is because, for any invalid solution A, the associated plane N (see the proof of Theorem 6.16) must intersect the three lines at some point, say p. Then the reconstructed depth of point p with respect to the solution A would be infinite (points beyond the plane N would have negative recovered depth). This gives us a criteria to exclude all such invalid solutions.

sense specified in [104], which is obviously a more strict condition than requiring only two independent rotations.

6.5.4 Necessary and Sufficient Condition for Unique Calibration

In this section, we establish in detail the conditions of A under which the matrix ARA^{-1} is also a rotation matrix given that R is a rotation matrix. Let us suppose ARA^{-1} is a rotation matrix. We then have:

$$ARA^{-1}(A^{-T}R^TA^T) = I \quad \Leftrightarrow \quad RXR^T = X \tag{6.51}$$

where $X = A^{-1}A^{-T}$ is a positive definite matrix. Thus X has to be in the **symmetric real kernel** of the Lyapunov map:

$$\begin{aligned} L : \mathbb{C}^{3\times 3} &\rightarrow \mathbb{C}^{3\times 3} \\ X &\mapsto X - RXR^T. \end{aligned} \tag{6.52}$$

We will denote this kernel as SRKer(L). According to Callier and Desoer [11], the map L has eigenvalues $1 - \lambda_i\lambda_j^*, 1 \leq i, j \leq 3$ where $\lambda_i, i = 1, 2, 3$ are eigenvalues of the matrix

R. Without loss of generality, the rotation matrix R has eigenvalues $1, \alpha, \bar{\alpha} \in \mathbb{C}$ and corresponding right eigenvectors $u, v, \bar{v} \in \mathbb{C}^3$. Then the (complex) kernel of L is given by:

$$\text{Ker}(L) \;=\; \text{span}\{X_1 = uu^*, X_2 = vv^*, X_3 = \bar{v}\bar{v}^*\} \quad \subset \mathbb{C}^{3\times 3} \tag{6.53}$$

where, for a vector $v \in \mathbb{C}^3$, $\bar{v}$ is its conjugate and v^* is its conjugate transpose. We assume here R is neither the identity matrix I or a 180^o rotation, *i.e.*, R is not of the form $e^{\widehat{u}k\pi}$ for some $k \in \mathbb{Z}$ and some $u \in \mathbb{R}^3$ of unit length. Then only X_1 is real and $X_2 = \bar{X}_3$ are complex, and L has a three dimensional real kernel but one dimension is spanned by $i(X_2 - X_3)$ which is skew-symmetric (here $i = \sqrt{-1}$). Therefore, the solution space for a symmetric real X is 2 dimensional and must have the form $X = \beta X_1 + \gamma(X_2 + X_3)$ with $\beta, \gamma \in \mathbb{R}$. Summarizing the above we obtain:

Lemma 6.18. *Given a rotation matrix R not of the form $e^{\widehat{u}k\pi}$ for some $k \in \mathbb{Z}$ and some $u \in \mathbb{R}^3$ of unit length, the symmetric real kernel associated with the Lyapunov map $L : X \mapsto X - RXR^T$ is 2 dimensional. If R is of the form $e^{\widehat{u}k\pi}$, then SRKer(L) is 4 dimensional if k is odd and 6 dimensional if k is even.*

Note that the case when the rotation is 180^o has little practical significance in real situations, since no image correspondences are available in this case. Thus, from now on we may assume that all rotations that we consider for the camera self-calibration problem are strictly less than 180^o unless otherwise stated.

Suppose now we have m rotation matrices $R_i, i = 1, \ldots, m$. For AR_iA^{-1} to be rotation matrices, $X = A^{-1}A^{-T}$ has to be in the intersection of symmetric real kernels of all the linear maps:

$$\begin{aligned} L_i : \mathbb{C}^{3\times 3} &\rightarrow \mathbb{C}^{3\times 3}, \qquad i = 1, \ldots, m \\ X &\mapsto X - R_iXR_i^T. \end{aligned} \tag{6.54}$$

That is $X \in \cap_{i=1}^m \text{SRKer}(L_i)$.

Theorem 6.19 (Necessary & Sufficient Condition for Unique Calibration). *Suppose the camera motion is given by a subset $\{(R_i, T_i)\}_{i=1}^m \subset SE(3)$ with R_i are not of the form $e^{\widehat{u}k\pi}$ for some $k \in \mathbb{Z}$ and some $u \in \mathbb{R}^3$ of unit length. Then the camera calibration matrix A can be uniquely determined if and only if there are at least* **two** *rotation components R_i and R_j whose axes are linearly independent.*

Proof: Theorem 6.16 allows us to check that, if A is a valid camera calibration, then AR_iA^{-1} has to be a rotation matrix for each R_i. The necessity is obvious: if two rotation matrices R_1 and R_2 have the same axis, they have the same eigenvectors hence $\text{SRKer}(L_1) = \text{SRKer}(L_2)$ where $L_i : X \mapsto X - R_iXR_i^T, i = 1, 2$. We now only need to prove the sufficiency. We may assume u_1 and u_2 are the two rotation axes of R_1 and R_2 respectively and are linearly independent. Since, by assumption, both R_1 and R_2 considered are not 180^o rotation, both $\text{SRKer}(L_1)$ and $\text{SRKer}(L_2)$ are 2 dimensional. Since u_1 and u_2 are linearly independent, the matrices $u_1u_1^T$ and $u_2u_2^T$ are linearly independent and are in $\text{SRKer}(L_1)$ and $\text{SRKer}(L_2)$ respectively. Thus $\text{SRKer}(L_1)$ is not fully contained in $\text{SRKer}(L_2)$ hence their intersection $\text{SRKer}(L_1) \cap \text{SRKer}(L_2)$ has at most 1 dimension. Thus $X = I$ for $X \in SL(3)$. ∎

It is well know many motion subgroups of $SE(3)$, though of practical importance, do not have rotation along two independent axes. For example, the planar motion and screw motion. According to the theorem, if the motion of the camera falls into such a motion group, unique self-calibration is impossible. A more detailed analysis will be given in Chapter 7 about to what extend we can still recover camera calibration, motion and scene structure with respect to each Lie subgroup of $SE(3)$.

Although it has little practical importance, in order to make the theory complete, we also give the results of self-calibration in presence of rotation of 180^o (for simplicity, we here do not give the proof). Combined with Theorem 6.19, they give conditions for a unique calibration in more general cases.

Remark 6.20. *Suppose $R_i = e^{\widehat{u}_i\theta_i}, i = 1, 2$ are elements in $SO(3)$. u_i are vectors of unit length. Let L_i be the Lyapunov map associated to R_i. Then we have the following cases:*

$$
\begin{aligned}
u_1^Tu_2 = 0,\ |\theta_1| = |\theta_2| = \pi &\Rightarrow SRKer(L_1) \cap SRKer(L_2) = span\{I,\ u_1u_1^T,\ u_2u_2^T\},\\
0 < |u_1^Tu_2| < 1, |\theta_1| = |\theta_2| = \pi &\Rightarrow SRKer(L_1) \cap SRKer(L_2) = span\{I, \widehat{u}_2u_1u_1^T\widehat{u}_2\},\\
u_1^Tu_2 = 0,\ |\theta_1| = \pi,\ 0 < |\theta_2| < \pi &\Rightarrow SRKer(L_1) \cap SRKer(L_2) = span\{I, u_2u_2^T\},\\
0 < |u_1^Tu_2| < 1, |\theta_1| = \pi, 0 < |\theta_2| < \pi &\Rightarrow SRKer(L_1) \cap SRKer(L_2) = span\{I\}.
\end{aligned}
$$

6.6 Continuous Case

So far, we have understood camera self-calibration when the motion of the camera is discrete – positions of the camera are specified as discrete points in $SE(3)$. In this section,

we study its continuous version. Suppose the coordinates of a point $p \in \mathbb{E}^3$ under the camera velocities $(\omega(t), v(t))$ is $\mathbf{X}(t)$. Let $\mathbf{X}'(t) = A\mathbf{X}(t)$ be the coordinates in the uncalibrated space. From (2.13), we directly have:

$$\dot{\mathbf{X}}'(t) = A\widehat{\omega(t)}A^{-1}\mathbf{X}(t) + Av(t). \tag{6.55}$$

For simplicity, we will drop the time dependence and define two new vectors $v' = Av \in \mathbb{R}^3$ and $\omega' = A\omega \in \mathbb{R}^3$.

6.6.1 General Motion Case

By the general case we mean that both the angular and linear velocities ω and v are non-zero. Note that $\mathbf{X} = \lambda\mathbf{x}$ yields $\dot{\mathbf{X}} = \dot{\lambda}\mathbf{x} + \lambda\dot{\mathbf{x}}$. Then (6.55) gives:

$$\begin{aligned} & \dot{\mathbf{X}} = A\widehat{\omega}A^{-1}\mathbf{X} + v' \quad \Rightarrow \quad (v' + \mathbf{x}) \times \dot{\mathbf{X}} = (v' + \mathbf{x}) \times A\widehat{\omega}A^{-1}\mathbf{X} \\ \Rightarrow \quad & \dot{\mathbf{x}}^T A^{-T}\widehat{v}A^{-1}\mathbf{x} + \mathbf{x}^T A^{-T}\widehat{\omega}\widehat{v}A^{-1}\mathbf{x} = 0. \end{aligned} \tag{6.56}$$

The last equation is called the **uncalibrated continuous epipolar constraint**, which is apparently the uncalibrated version of (3.15). As the calibrated case, define the optical flow $\mathbf{u} = \dot{\mathbf{x}}$ and the special symmetric matrix $s = \frac{1}{2}(\widehat{\omega}\widehat{v} + \widehat{v}\widehat{\omega})$. Define the **continuous fundamental matrix** $F' \in \mathbb{R}^{6\times 3}$ to be:

$$F' = \begin{bmatrix} A^{-T}\widehat{v}A^{-1} \\ A^{-T}sA^{-1} \end{bmatrix}. \tag{6.57}$$

Then from (6.56) we have an equivalent expression of the uncalibrated continuous epipolar constraint:

$$[\mathbf{u}^T, \mathbf{x}^T]F'\mathbf{x} = 0 \tag{6.58}$$

F' can therefore be estimated from as few as eight optical flows $(\mathbf{x}, \dot{\mathbf{x}})$ from (6.56).

Note that $\widehat{v'} = A^{-T}\widehat{v}A^{-1}$ and $\widehat{\omega'} = A^{-T}\widehat{\omega}A^{-1}$. Applying Lemma 6.4 repeatedly, we obtain

$$A^{-T}sA^{-1} = \frac{1}{2}(A^{-T}\widehat{\omega}A^T\widehat{v'} + \widehat{v'}A\widehat{\omega}A^{-1}) = \frac{1}{2}(\widehat{\omega'}S^{-1}\widehat{v'} + \widehat{v'}S^{-1}\widehat{\omega'}). \tag{6.59}$$

Then the uncalibrated continuous epipolar constraint (6.56) is equivalent to:

$$\dot{\mathbf{x}}^T\widehat{v'}\mathbf{x} + \mathbf{x}^T\frac{1}{2}(\widehat{\omega'}S^{-1}\widehat{v'} + \widehat{v'}S^{-1}\widehat{\omega'})\mathbf{x} = 0. \tag{6.60}$$

Suppose $S^{-1} = BB^T$ for another $B \in SL(3)$, then $A = BR_0$ for some $R_0 \in SO(3)$. We have:

$$\begin{aligned} & \dot{\mathbf{x}}^T \widehat{v'}\mathbf{x} + \mathbf{x}^T \frac{1}{2}(\widehat{\omega'} S^{-1} \widehat{v'} + \widehat{v'} S^{-1} \widehat{\omega'})\mathbf{x} = 0 \\ \Leftrightarrow \quad & \dot{\mathbf{x}}^T \widehat{v'}\mathbf{x} + \mathbf{x}^T \frac{1}{2}(\widehat{\omega'} BB^T \widehat{v'} + \widehat{v'} BB^T \widehat{\omega'})\mathbf{x} = 0 \\ \Leftrightarrow \quad & \dot{\mathbf{x}}^T B^{-T} \widehat{R_0 v} B^{-1}\mathbf{x} + \mathbf{x}^T B^{-T} \widehat{R_0 \omega} \widehat{R_0 v} B^{-1}\mathbf{x} = 0. \end{aligned} \tag{6.61}$$

Comparing to (6.56), one cannot tell the camera A with motion (ω, v) from the camera B with motion $(R_0\omega, R_0 v)$. Thus, like the discrete case, without knowing the camera motion the calibration can only be recovered in the space $SL(3)/SO(3)$, *i.e.*, only the symmetric matrix S^{-1} hence S can be recovered.

However, unlike the discrete case, the matrix S cannot be fully recovered in the continuous case. Since $S^{-1} = AA^T$ is a symmetric matrix, it can be diagonalized as:

$$S^{-1} = R_1^T \Sigma R_1, \quad R_1 \in SO(3) \tag{6.62}$$

where $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \sigma_3\}$. Then let $\omega'' = R_1\omega'$ and $v'' = R_1 v'$. Applying Lemma 6.4, we have:

$$\begin{aligned} \widehat{v'} &= R_1^T \widehat{v''} R_1 \\ \frac{1}{2}(\widehat{\omega'} S^{-1} \widehat{v'} + \widehat{v'} S^{-1} \widehat{\omega'}) &= R_1^T \frac{1}{2}(\widehat{\omega''} \Sigma \widehat{v''} + \widehat{v''} \Sigma \widehat{\omega''}) R_1. \end{aligned} \tag{6.63}$$

Thus the continuous epipolar constraint (6.56) is also equivalent to:

$$(R_1\dot{\mathbf{x}})^T \widehat{v''}(R_1\mathbf{x}) + (R_1\mathbf{x})^T \frac{1}{2}(\widehat{\omega''} \Sigma \widehat{v''} + \widehat{v''} \Sigma \widehat{\omega''})(R_1\mathbf{x}) = 0. \tag{6.64}$$

From this equation, one can see that there is no way to tell a camera A with $AA^T = R_1^T \Sigma R_1$ from a camera $B = R_1 A$. Therefore, only the diagonal matrix Σ can be recovered as camera parameters since both the scene structure and camera motion are unknown.

Note that Σ is in $SL(3)$ hence $\sigma_1\sigma_2\sigma_3 = 1$. The singular values only have two degrees of freedom. Hence we have:

Theorem 6.21 (Self-Calibration in Continuous Case). *Consider an uncalibrated camera with an unknown calibration matrix $A \in SL(3)$. Then only the eigenvalues of AA^T can be recovered from the uncalibrated continuous epipolar constraint.*

If we define that two matrices in $SL(3)$ are equivalent if and only if they have the same singular values. The intrinsic parameter space is then reduced to the space $SL(3)/\sim$ where $\sim$ represents this equivalence relation. The fact that only two camera parameters can be recovered was known to Brooks *et al.* [9]. They have also shown how to do calibration for certain matrices A with only two unknown parameters. But our proof has been much more simpler due to the use of Lemma 6.4.

Comment 6.22. *It is a little surprising to see that the discrete and continuous cases are different for the first time, especially knowing that in the calibrated case these two cases have almost exactly parallel sets of theory and algorithms. We believe that this has to do with the map:*

$$\begin{aligned} \gamma_A : \mathbb{R}^{3\times 3} &\rightarrow \mathbb{R}^{3\times 3} \\ X &\mapsto AXA^T \end{aligned}$$

where A is an arbitrary matrix in $\mathbb{R}^{3\times 3}$. Let $so(3)$ be the Lie algebra of $SO(3)$. The restricted map $\gamma_A \mid_{so(3)}$ is an endomorphism while $\gamma_A \mid_{SO(3)}$ is not. Consider $\gamma_A \mid_{so(3)}$ to be the first order approximation of $\gamma_A \mid_{SO(3)}$. Then the information about the calibration matrix A does not fully show up until the second order term of the map γ_A. This also somehow explains why in the discrete case the (Kruppa) constraints that we can get for A are in general nonlinear.

Comment 6.23. *From the above discussion, if one only uses the (bilinear) continuous epipolar constraint, at most two intrinsic parameters of the calibration matrix A can be recovered. However, it is still possible that the full information about A can be recovered from multilinear constraints on the higher order derivatives of optical flow. A complete list of such constraints has been given in Chapter 5.*

6.6.2 Pure Rotation Case

Since full calibration is not possible in the general case when translation is present, we need to know if it is possible in some special case. The only case left is when there is only rotational motion, *i.e.*, the linear velocity v is always zero. In this case the continuous fundamental matrix is no longer well defined. However from the equation (6.55) we have:

$$\begin{aligned} \dot{\mathbf{X}} = A\widehat{\omega}A^{-1}\mathbf{X} \quad &\Rightarrow \quad \dot{\lambda}\mathbf{x} + \lambda\dot{\mathbf{x}} = A\widehat{\omega}\lambda A^{-1}\mathbf{x} \\ \Rightarrow \quad \widehat{\mathbf{x}}\dot{\mathbf{x}} = \widehat{\mathbf{x}}A\widehat{\omega}A^{-1}\mathbf{x}. & \end{aligned} \tag{6.65}$$

This is a degenerate version of the continuous epipolar constraint and it gives two independent constraints on the matrix $A\widehat{\omega}A^{-1}$ for each $(\mathbf{x}, \dot{\mathbf{x}})$. Given $n \geq 4$ optical flow measurements $\{(\mathbf{x}_i, \dot{\mathbf{x}}_i)\}_{i=1}^n$, one may uniquely determine the matrix $A\widehat{\omega}A^{-1}$ by solving a linear equation:

$$Mc = b \tag{6.66}$$

where $M \in \mathbb{R}^{2n\times 9}$ is a matrix function of $\{(\mathbf{x}_i, \dot{\mathbf{x}}_i)\}_{i=1}^n$, $b \in \mathbb{R}^9$ is a vector function of $\widehat{\mathbf{x}}_i\dot{\mathbf{x}}_i$'s and $c \in \mathbb{R}^9$ is the 9 entries of $A\widehat{\omega}A^{-1}$. The solution is given by the following lemma:

Lemma 6.24. *If $\omega \neq 0$, then $A\widehat{\omega}A^{-1} = C - \gamma I$ where $C \in \mathbb{R}^{3\times 3}$ is the matrix corresponding to the least square solution c of the equation $Mc = b$ and γ is the unique real eigenvalue of C.*

The proof is straightforward. Then the self-calibration problem becomes how to recover $S = A^{-T}A^{-1}$ or $S^{-1} = AA^T$ from matrices of the form $A\widehat{\omega}A^{-1}$. Without loss of generality, we may assume ω is of unit length.

Let $C' = A\widehat{\omega}A^{-1} \in \mathbb{R}^{3\times 3}$. Then we have:

$$SC' = A^{-T}\widehat{\omega}A^{-1} = \widehat{\omega'} \tag{6.67}$$

where $\omega' = A\omega$. Thus $SC' = -(SC')^T$, *i.e.*, $SC' + (C')^T S = 0$. That is, S has to be in the kernel of the Lyapunov map:

$$\begin{aligned} L' : \mathbb{C}^{3\times 3} &\to \mathbb{C}^{3\times 3} \\ X &\mapsto (C')^T X + XC' \end{aligned} \tag{6.68}$$

If $\omega \neq 0$, the eigenvalues of $\widehat{\omega}$ have the form $0, i\alpha, -i\alpha$ with $\alpha \in \mathbb{R}$. Let the corresponding eigenvectors are $\omega, u, \bar{u} \in \mathbb{C}^3$. According to Callier and Desoer [11], the null space of the map L' has three dimensions and is given by:

$$\mathrm{Ker}(L') = \mathrm{span}\{S_1 = A^{-T}\omega\omega^*A^{-1}, S_2 = A^{-T}uu^*A^{-1}, S_3 = A^{-T}\bar{u}\bar{u}^*A^{-1}\}. \tag{6.69}$$

As in the discrete case, the symmetric real S is of the form $S = \beta S_1 + \gamma(S_2 + S_3)$, *i.e.*, the symmetric real kernel of L' is only two dimensional. We denote this space as $\mathrm{SRKer}(L')$. We thus have:

Lemma 6.25. *Given a matrix $C' = A\widehat{\omega}A^{-1}$ with $\omega \in S^2$, the symmetric real kernel associated with the Lyapunov map $L' : (C')^T X - XC'$ is 2 dimensional.*

Similar to the discrete case we have:

Theorem 6.26. *Given matrices* $C_j' = A\widehat{\omega}_j A^{-1} \in \mathbb{R}^{3\times 3}, j = 1,\ldots,n$ *with* $\|\omega_j\| = 1$. *The real symmetric matrix* $S = A^{-T}A^{-1} \in SL(3)$ *is uniquely determined if and only if at least* **two** *of the* n *vectors* $\omega_j, j = 1,\ldots,n$ *are linearly independent.*

6.7 Simulation Results

In this section, we test the performance of the proposed algorithms through different experiments. The error measure between the actual calibration matrix A and the estimated calibration matrix $\tilde{A}$ was chosen to be:

$$error = \frac{\|A - \tilde{A}\|}{\|A\|} \times 100\%$$

Table 6.2 shows the simulation parameters used in the experiments.[6] The calibration

Table 6.2: Simulation parameters

Parameter	Unit	Value
Number of trials		100
Number of points		20
Number of frames		3-4
Field of view	degrees	90
Depth variation	u.f.l.	100 - 400
Image size	pixels	500 × 500

matrix A is simply the transformation from the original 2×2 (in unit of focal length) image to the 500 × 500 pixel image. For these parameters, the true A should be:

$$A = \begin{pmatrix} 250 & 0 & 250 \\ 0 & 250 & 250 \\ 0 & 0 & 1 \end{pmatrix}.$$

The ratio of the magnitude of translation and rotation, or simply the T/R ratio, is compared at the center of the random cloud (scattered in the truncated pyramid specified by the given field of view and depth variation). For all simulations, the number of trials is 100.

In the following, we only simulate the three cases which have linear calibration algorithms: pure rotation case, and the two cases when the translation is perpendicular or

[6] u.f.l. stands for unit of focal length.

parallel to the rotation axe. Although I do not outline the algorithms here, they are evident from corresponding sections.

Pure rotation case: Figures 6.3, 6.4 and 6.5 show the experiments performed in the pure rotation case. The axes of rotation are X and Y for Figures 6.3 and 6.5, and X and Z for Figure 6.4. The amount of rotation is 20^o. The perfect data was corrupted with zero-mean Gaussian noise with standard deviation σ varying from 0 to 5 pixels. In Figures 6.3 and 6.4 it can be observed that the algorithm performs very well in the presence of noise, reaching errors of less than 6% for a noise level of 5 pixels. Figure 6.5 shows the effect of the amount of translation. This experiment is aimed to test the robustness of the pure rotation algorithm with respect to translation. The T/R ratio was varied from 0 to 0.5 and the noise level was set to 2 pixels. It can be observed that the algorithm is not robust with respect to the amount of translation.

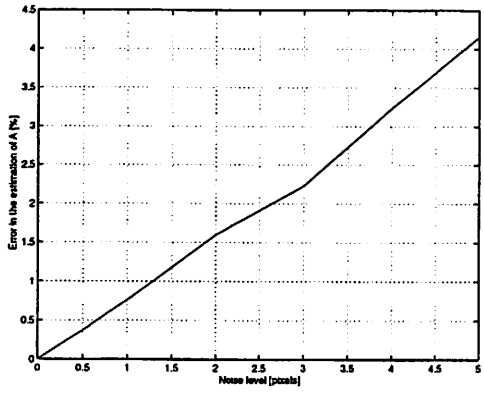


Figure 6.3: Pure rotation algorithm. Rotation axes X-Y.

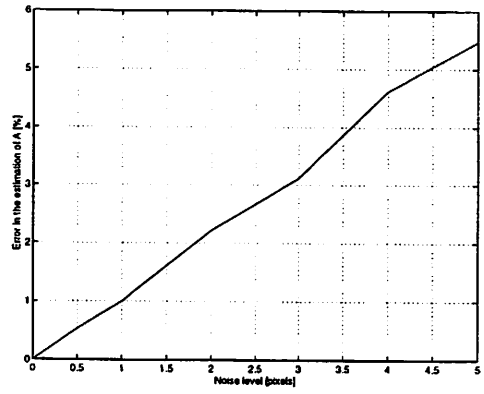


Figure 6.4: Pure rotation algorithm. Rotation axes X-Z.

Translation parallel to rotation axis: Figures 6.6 and 6.7 show the experiments performed when translation is parallel to the axis of rotation.[7] The non-isotropic normalization procedure proposed by Hartley [35] and statistically justified by Mühlich and Mester [82] was used to estimate the fundamental matrix. Figure 6.6 shows the effect of noise in the estimation of the calibration matrix for $T/R = 1$ and a rotation of $\theta = 20^o$ between consecutive frames. It can be seen that the normalization procedure improves the estimation of the calibration matrix, but the improvement is not significant. This result is consistent with

[7]For specifying the Rotation/Translation axes, we simply use symbols such as "XY-YY-ZZ" which means: for the first pair of images the relative motion is rotation along X and translation along Y; for the second pair both rotation and translation are along Y; and for the third pair both rotation and translation are along Z.

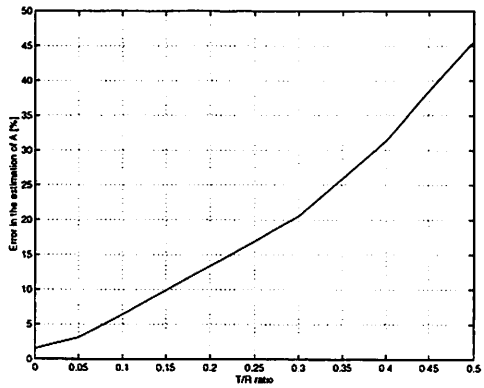


Figure 6.5: Rotation axes X-Y, $\sigma = 2$.

that of [82], since the effect of normalization is more important for large noise levels. On the other hand, the performance of the algorithm is not as good as that of the pure rotation case, but still an error of 5% is reached for a noise level of 2 pixels. Figure 6.7 shows the effect of the angle of rotation in the estimation of the calibration matrix for a noise level of 2 pixels. It can be concluded that a minimum angle of rotation between consecutive frames is required for the algorithm to succeed.

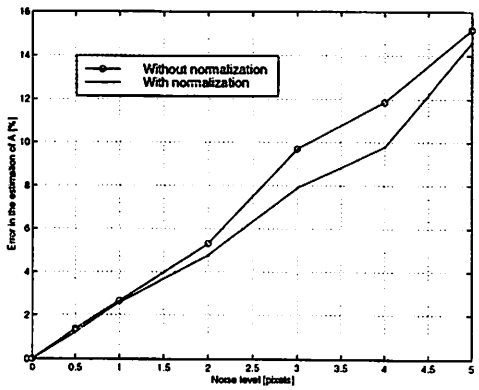


Figure 6.6: Rotation parallel to translation case. $\theta = 20^o$. Rotation/Translation axes: XX-YY-ZZ, T/R ratio = 1.

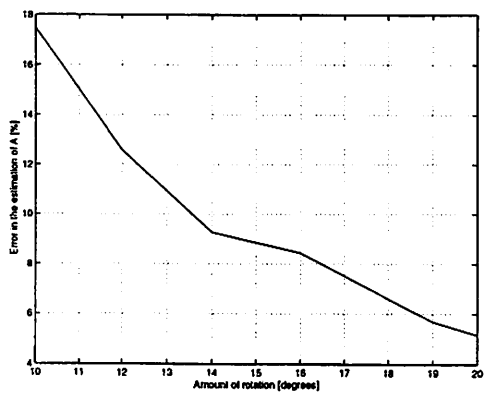


Figure 6.7: Rotation parallel to translation case. $\sigma = 2$. Rotation/Translation axes: XX-YY-ZZ, T/R ratio = 1.

Translation perpendicular to rotation axis: Figures 6.8 and 6.9 show the experiments performed when translation is perpendicular to the axis of rotation. It can be observed that this algorithm is much more sensitive to noise. The noise has to be less than 0.5 pixels in order to get an error of 5%. Experimentally it was found that Kruppa's equations are very

sensitive to the normalization of the fundamental matrix F and that the eigenvalues λ_1 and λ_2 of $F^T\widehat{T}'$ are close to each other. Therefore in the presence of noise, the estimation of those eigenvalues might be ill conditioned (even complex eigenvalues are obtained) and so is the solution of Kruppa's equations. Another experimental problem is that more than one non-degenerate solution to Kruppa's equations can be found. This is because, when taking all possible combinations of eigenvalues of $F^T\widehat{T}'$ in order to normalize F, the smallest eigenvalue of the linear map associated to "incorrect" Kruppa's equations can be very small. Besides, the eigenvector associated to this eigenvalue can eventually give a non-degenerate matrix. Thus in the presence of noise, you can not distinguish between the correct and one of these incorrect solutions. The results presented here correspond to the best match (to the ground truth) when more than one solution is found. Finally it is important to note that large motions can significantly improve the performance of the algorithm. Figure 6.9 shows the error in the estimation of the calibration matrix for a rotation of 30^o. It can be observed that the results are comparable to that of the parallel case with a rotation of 20^o.

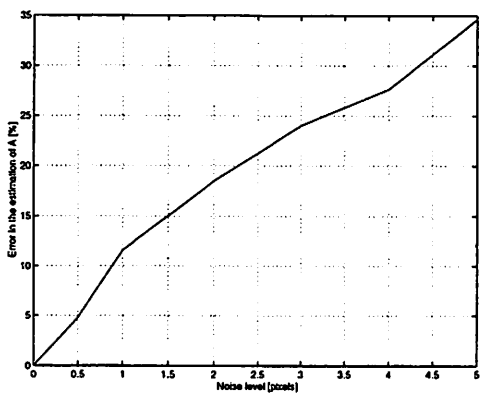


Figure 6.8: Rotation orthogonal to translation case. $\theta = 20^o$. Rotation/Translation axes: XY-YZ-ZX, T/R ratio = 1.

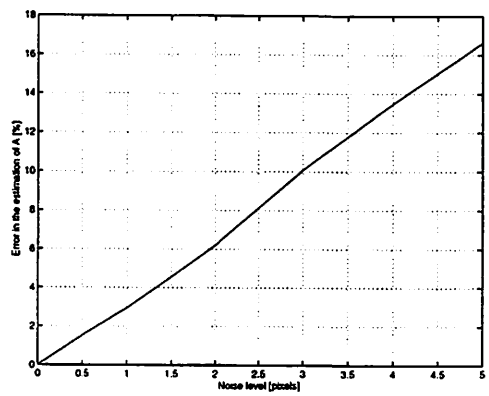


Figure 6.9: Rotation orthogonal to translation case. $\theta = 30^o$. Rotation/Translation axes: XY-YZ-ZX, T/R ratio = 1.

Robustness: In order to check how robust the algorithms are with respect to the angle ϕ between the rotation axis and translation, we run them with ϕ varying from 0^o to 90^o. The noise level is 2 pixels, amount of rotation is always 20^o and the T/R ratio is 1. Translation and rotation axes are given by Figure 6.10. Surprisingly, as we can see from the results given in Figure 6.11, for the range $0^o \leq \phi \leq 50^o$, both algorithms give pretty close estimates. This is because, for this range of angle, numerically the eigenvalues of the matrix $F^T\widehat{T}'$ are

complex and their norm is very close to the norm of the matrix $F^T\widehat{T}'F$. Therefore, the computed renormalization scale λ from both algorithms is very close, as is the calibration estimate. For $\phi > 50^o$, the eigenvalues of $F^T\widehat{T}'$ become real and the performance of the two algorithms is no longer the same.

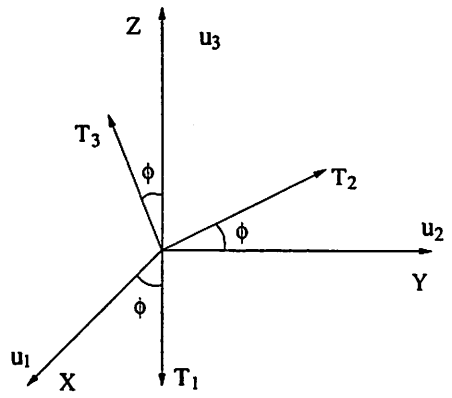


Figure 6.10: The relation of the three rotation axes u_1, u_2, u_3 and three translations T_1, T_2, T_3.

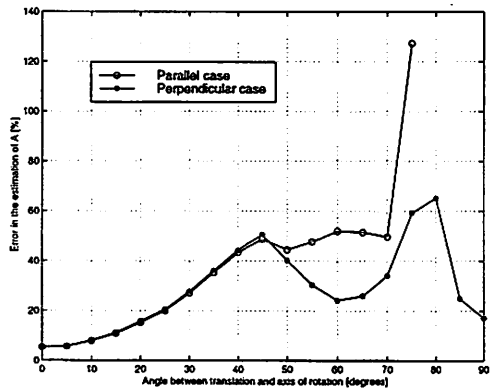


Figure 6.11: Estimation error in calibration w.r.t. different angle ϕ. Noise level $\sigma = 2$. Rotation and translation axes are shown by the figure to the left. Rotation amount is always 20^o and T/R ratio is 1.

6.8 Discussion

In this chapter, we have revisited the Kruppa's equations based approach for camera self-calibration. Through a detailed study of the cases when the camera rotation axis is parallel or perpendicular to the translation, we have discovered generic difficulties in the conventional self-calibration schemes based on directly solving the nonlinear Kruppa's equations. Our results not only complete existing results in the literature regarding the solutions of Kruppa's equations but also provide brand new linear algorithms for self-calibration other than the well-known one for a pure rotating camera. Simulation results show that, under the given conditions, these linear algorithms provide good estimates of the camera calibration despite the degeneracy of the Kruppa's equations. The performance is close to that of the pure rotation case.

The relationship between Kruppa's equations and chirality given in Theorem 6.16 has revealed an intrinsic condition for a unique calibration given in Theorem 6.19. This

condition is extremely important for us to clearly characterize the generic ambiguities in the problem of reconstruction from multiple images. This is the subject of next chapter.

Chapter 7

Reconstruction and Reprojection up to Subgroups

> *"The rise of projective geometry made such an overwhelming impression on the geometers of the first half of the nineteenth century that they tried to fit all geometric considerations into the projective scheme. ... The dictatorial regime of the projective idea in geometry was first successfully broken by the German astronomer and geometer Möbius, but the classical document of the democratic platform in geometry establishing the group of transformations as the ruling principle in any kind of geometry and yielding equal rights to independent consideration to each and any such group, is F. Klein's Erlangen program."*
>
> — Herman Weyl, *Classical Groups*

Reconstructing spatial properties of a scene from a number of images taken by an uncalibrated camera is a classical problem in computer vision. It is particularly important when the camera used to acquire the images is not available for calibration, as for instance in video post-processing, or when the calibration changes in time, as in vision-based navigation. If we represent the scene by a number of isolated points in three-dimensional space and the imaging process by an ideal perspective projection, the problem can be reduced to a purely geometric one, which has been subject to the intense scrutiny of a number of researchers during the past ten years. Their efforts have led to several important and useful results. The problem is that *conditions for a unique Euclidean reconstruction are almost never satisfied in sequence of images of practical interest.* In fact, they require as a necessary condition that the camera undergoes rotation about at least two independent axes, which is rarely the case both in video processing and in autonomous navigation.

In this chapter we address the question of *what exactly can be done when the necessary and sufficient conditions for unique reconstruction are* **not** *satisfied.* In particular:

> (i) For all the motions that do not satisfy the conditions, to what extent can we reconstruct structure, motion and calibration?
> (ii) If the goal of the reconstruction is to produce a new view of the scene from a different vantage point, how can we make sure that the image generated portrays a "valid" Euclidean scene?

Relation to Previous Work: The study of ambiguities in Euclidean reconstruction (i) arises naturally in the problem of motion and structure recovery and self-calibration from multiple cameras. There is a vast body of literature on this topic, which cannot be reviewed in the limited space allowed. Here we only comment on some of the work that is most closely related to this chapter, while we refer the reader to the literature for more details, references and appropriate credits (see for instance [12, 34, 65, 104, 117, 132] and references therein).

It has long been known that in the absence of any *a priori* information about motion, calibration and scene structure, reconstruction can be performed at least up to a projective transformation [21]. Utilizing additional knowledge about the relationship between geometric entities in the image (e.g., parallelism) one can stratify the different levels of reconstructions from projective all the way to Euclidean [6, 13, 21, 81]. At such a level of generality, the conditions on the uniqueness and existence of solutions are restrictive and the algorithms are computationally costly, often exhibiting local minima [64].

Recently, Sturm [104] has proposed a taxonomy of critical motions, that is motions which do not allow a unique reconstruction. However, not only the given taxonomy is by no means intrinsic to Euclidean reconstruction (see [66]), but also no explicit characterization of the ambiguities in the reconstructed shape, motion and calibration has been given. A natural continuation of these efforts involved the analysis of cases where the motion and/or calibration were restricted either to planar or linear motion [4, 81] and techniques were proposed for affine reconstruction or up to a one-parameter family.

Several techniques have been proposed to synthesize novel views of a reconstructed scene (ii): in [2], trilinear constraints have been exploited to help generate reprojected images for a calibrated camera. In the case of a partially uncalibrated camera, such a method has to face the issues of whether the reprojected image portrays a valid Euclidean scene.

Chapter Outline

The well-known - but conservative - answer to question (i) is that structure can at least be recovered up to a global projective transformation of the three-dimensional space. However, there is more to be said, as we do in Section 7.1 for the case of constant calibration.[1] There, we give explicit formulae for exact ambiguities in the reconstruction of scene structure, camera motion and calibration with respect to all subgroups of the Euclidean motion. In principle, one should study ambiguities corresponding to all critical configurations as given in [66]. However, it is only the ambiguities that exhibit a *group structure* that are of practical importance in the design of estimation algorithms. In such a case, not only can the analysis be considerably simplified but also clean formulae for all generic ambiguities can be derived. Such formulae are important for 3D reconstruction as well as for synthesizing novel 2D views. Question (ii) is then answered in Section 7.2, where we characterize the complete set of vantage points that generate "valid" images of the scene regardless of generic ambiguities in 3D reconstruction.

These results have great practical significance, because they quantify precisely to what extent scene structure, camera motion and calibration can be estimated in sequences for which many of the techniques available to date do not apply. Furthermore, the analysis clarifies the process of 2D view synthesis from novel viewpoints. In addition to that, we give a novel account of known results on the role of multilinear constraints and their relationship to bilinear ones.

Granted the potential impact on applications, this chapter is mainly concerned with theory. We address neither algorithmic issues, nor do we perform experiments of any sort: the validation of our statements is in the proofs. We have tried to keep our notation as terse as possible. Our tools are borrowed from linear algebra and some differential geometry, although all the results should be accessible without background in the latter. We use the language of (Lie) groups because that allows us to give an explicit characterization of all the ambiguities in a concise and intuitive fashion. Traditional tools involved in the analysis of self-calibration involved complex loci in projective spaces (e.g., the "absolute conic"), which can be hard to grasp for someone not proficient in algebraic geometry.

[1] In fact, even in the case of time-varying calibration, in principle, the best one can do is an affine reconstruction, not just a projective one!

7.1 Reconstruction under Motion Subgroups

The goal of this section is to study all "critical" motion groups that do *not* allow unique reconstruction of structure, motion and calibration. While a *classification* of such critical motions has been presented before (see [66]), we here go well beyond by giving an *explicit characterization* of the ambiguity in the reconstruction for each critical motion. Such an explicit characterization is crucial in deriving the ambiguity in the generation of novel views of a scene, which we study in section 7.2.

In this section, we characterize the generic ambiguity in the recovery of (a) structure, (b) motion and (c) calibration corresponding to each possible critical motion. A subgroup of $SE(3)$ is called *critical* if the reconstruction is not unique when the motion of the camera is restricted to it. For the purpose of this section, we assume that the calibration matrix A is constant.

7.1.1 Some Preliminaries

So far the only restriction we have imposed on the constant calibration matrix A is that it is non-singular and is normalized as to have $\det(A) = 1$. However, according to Section 6.1, A can only be determined up to an equivalence class of rotations, that is $A \in SL(3)/SO(3)$.[2] The unrecoverable rotation in our choice of A simply corresponds to a rotation of the entire camera system.

In Section 6.5.4, Theorem 6.19 states a very important and useful fact: the condition for a unique calibration has nothing to do with translation (as opposed to the results given in [104])! [3] Due to this theorem, many proper continuous subgroups of $SE(3)$ are critical for self-calibration. So the first step in our analysis consists in classifying all continuous Lie subgroups of $SE(3)$ which are critical. It is a well known fact that a complete list of subgroups of $SE(3)$ can be classified by all Lie subalgebras of the Lie algebra $se(3)$ of $SE(3)$ and then exponentiate them. It is then straightforward to show that each of these subgroups must have the same ambiguity in reconstruction as one in the following list (as

[2]Here take left coset as elements in the quotient space. A representation of this quotient space is given, for instance, by upper-triangular matrices; such a representation is commonly used in modeling calibration matrices by means of physical parameters of cameras such as focal length, principal point and pixel skew.

[3]This is because we here only consider the *generic ambiguity* in reconstruction, *i.e.*, such ambiguity exists no matter what the camera sees and no matter what the algorithms do.

we will explain in the comments):

Translational Motion: $(\mathbb{R}^3, +)$ and its subgroups

Rotational Motion: $(SO(3), \cdot)$ and its subgroups

Planar Motion: $SE(2)$

Screw Motion: $(SO(2), \cdot) \times (\mathbb{R}, +)$

Planar + Elevation: $SE(2) \times (\mathbb{R}, +)$

Rigid Body Motion: $SE(3)$

Comment 7.1 (Special Lie Subalgebras of $se(3)$). *The above list is by no means a complete list of ALL subgroups of $SE(3)$. For example, the "planar orbital motion", i.e., camera moving on a circle with the optical axis always facing the center, is none of the motion in the above list. However, it is can be treated as a special case of the planar motion since, as far as reconstruction is concerned, they obviously have the same generic ambiguities. In order to show that all subgroups have the same ambiguity in reconstruction as one of the above motions, we must go through all the possible Lie subalgebras of $se(3)$. It can be shown that, if a Lie subalgebra has at least 4 dimension and has two independent rotation components, then it must be $se(3)$ itself. Now the only interesting case is some three dimensional Lie subalgebras which, without loss of generality, are generated by elements:*

$$X = \begin{bmatrix} \widehat{e}_1 & u \\ 0 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} \widehat{e}_2 & v \\ 0 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} \widehat{e}_3 & w \\ 0 & 0 \end{bmatrix} \tag{7.1}$$

where e_1, e_2, e_3 are standard basis of $\mathbb{R}^3$ and u, v and w are three vectors in $\mathbb{R}^3$. In order for the Lie algebra generated by X, Y, Z is three dimensional. We must have the vector $\alpha = [u^T, v^T, w^T]^T \in \mathbb{R}^9$ in the null space of the matrix:

$$Q = \begin{bmatrix} \widehat{e}_2 & -\widehat{e}_1 & I \\ I & \widehat{e}_3 & -\widehat{e}_2 \\ -\widehat{e}_3 & I & \widehat{e}_1 \end{bmatrix}. \tag{7.2}$$

That is $Q\alpha = 0$. If $\alpha = 0$, then the subgroup generated by the algebra is just the pure rotation group $SO(3)$. If $\alpha \neq 0$, then the subgroup generated contains three independent rotation axes and translation (parallax). For such subgroups, a unique reconstruction is available. That is, they are not critical for reconstruction or have the same ambiguity as

the full rigid body motion $SE(3)$. A generic example for such a three dimensional subgroup is the isometry group of $\mathbb{S}^2$.

We are now ready to explore to what extent scene structure, camera motion and calibration can be reconstructed when motion is constrained onto one of the above subgroups. In other words, we will study the *generic* ambiguities of the reconstruction problem. In what follows, we use $p(t) = [p_1(t), p_2(t), p_3(t)]^T \in \mathbb{R}^3$ to denote the 3D coordinates of the point $p = [p_1, p_2, p_3, 1]^T \in \mathbb{R}^4$ with respect to the camera frame at time t: $p(t) = (R(t), T(t))p$. To simplify notation, recall that, for any $u \in \mathbb{R}^3$, we have defined $\widehat{u}$ to be a 3 skew-symmetric matrix such that $\forall v \in \mathbb{R}^3$ the cross product $u \times v = \widehat{u}v$.

7.1.2 Generic Ambiguities in Structure, Motion and Calibration

Translational motion ($\mathbb{R}^3$ and its subgroups). Pure translational motion is generated by elements of $se(3)$ of the form:

$$\xi = \begin{bmatrix} 0_{3\times 3} & u \\ 0 & 0 \end{bmatrix}, \quad u \in \mathbb{R}. \tag{7.3}$$

In this special transformation subgroup, the coordinate transformation between different views is given by

$$Ap(t) = Ap(t_0) + AT(t), \tag{7.4}$$

where $T(t) \in \mathbb{R}^3$ is the translation vector. According to Theorem 6.19, the calibration $A \in SL(3)$ cannot be recovered from pure translational motion, and therefore the corresponding structure p and translational motion T can be recovered only up to the unknown transformation A. We therefore have the following

Theorem 7.2 (Ambiguity under $\mathbb{R}^3$). *Consider an uncalibrated camera described by the calibration matrix $A \in SL(3)$, undergoing purely translational motion $\mathbb{R}^3$ (or any of its nontrivial subgroups) and let B be an arbitrary matrix in $SL(3)$. If the camera motion $T \in \mathbb{R}^3$ and the scene structure $p \in \mathbb{R}^4$ are unknown, then B, $B^{-1}AT$ and $B^{-1}Ap$ are the only generic ambiguous solutions for the camera calibration, camera motion and the scene structure respectively.*

Note that this ambiguity corresponds exactly to an affine reconstruction [81].

Comment 7.3. *Thus the group $SL(3)$ can be viewed as characterizing the generic ambiguity of reconstruction under pure translation, and will therefore be called the "ambiguity subgroup".*

In section 5.1 we have argued that multilinear constraints do not provide additional information. We verify here that, indeed, multilinear constraints do not reduce the generic ambiguity. Without loss of generality, we can assume the camera frame at time t_1 coincides with that at t_0, *i.e.*, $T(t_1) = 0$. Suppose $T(t_2), T(t_3) \in \mathbb{R}^3$ are translations from the second and third frames to the original one respectively. We then have:

$$\begin{bmatrix} A & 0 & \mathbf{x}(t_1) & 0 & 0 \\ A & AT(t_2) & 0 & \mathbf{x}(t_2) & 0 \\ A & AT(t_3) & 0 & 0 & \mathbf{x}(t_3) \end{bmatrix} \begin{bmatrix} A^{-1}B & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} B & 0 & \mathbf{x}(t_1) & 0 & 0 \\ B & AT(t_2) & 0 & \mathbf{x}(t_2) & 0 \\ B & AT(t_3) & 0 & 0 & \mathbf{x}(t_3) \end{bmatrix}.$$

Therefore the two sides of the equation span the same subspace. Hence trilinear constraints are identical for all the ambiguous solutions. One can easily check that the same is true for quadrilinear constraints.

Rotational motion ($SO(3)$). Pure rotation is generated by elements of $se(3)$ of the form:

$$\xi = \begin{bmatrix} \widehat{\omega} & 0 \\ 0 & 0 \end{bmatrix}, \quad \omega \in \mathbb{R}^3. \tag{7.5}$$

If any two entries of ω are zero, the subgroup $SO(2)$ is generated instead. The action of $SO(3)$ transforms the coordinates in different cameras by

$$Ap(t) = AR(t)p(t_0), \tag{7.6}$$

where $R(t) \in SO(3)$ is the rotation. According to Theorem 6.26, the calibration A can be recovered uniquely, and so can the rotational motion $R(t) \in SO(3)$. However, it is well known that the depth information of the structure cannot be recovered at all due to lack of parallax. We summarize these facts into the following:

Theorem 7.4 (Ambiguity under $SO(3)$). *Consider an uncalibrated camera with calibration matrix $A \in SL(3)$ undergoing purely rotational motion $SO(3)$ and let λ be an arbitrary (positive) scalar. If both the camera motion $R \in SO(3)$ and the scene structure $p \in \mathbb{R}^3$ are unknown, then A, R and $\lambda \cdot p$ are the only generic ambiguous solutions for the camera calibration, camera motion and the scene structure respectively.*

Comment 7.5. *The multiplicative group* $(\mathbb{R}^+, \cdot)$ *can be viewed as characterizing the ambiguity of the reconstruction under pure rotation. Note that such a group* $(\mathbb{R}^+, \cdot)$ *acts independently on each point. More specifically, the group consists of all smooth functions* $\phi : \mathbb{RP}^2 \to \mathbb{R}^+$.

As for the case of pure translation, there is no independent constraint among three or more images.

Planar motion $(SE(2))$**.** While the previous two cases were of somewhat academic interest and the theorems portray well-known facts, planar motion arises very often in applications. We will therefore study this case in some more detail.

Let $e_1 = [1,0,0]^T, e_2 = [0,1,0]^T, e_3 = [0,0,1]^T \in \mathbb{R}^3$ be the standard basis of $\mathbb{R}^3$. Without loss of generality, we may assume the camera motion is on the plane normal to e_3 and is represented by the subgroup $SE(2)$.

$$SE(2) = \left\{ \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \middle| R = e^{\widehat{e}_3 \theta}, \theta \in \mathbb{R}, T = (T_1, T_2, 0)^T \in \mathbb{R}^3 \right\}. \tag{7.7}$$

Let A be the unknown calibration matrix of the camera. As described in section 7.1.1 we consider A as an element of the quotient space $SL(3)/SO(3)$. According to Section 6.5.4, any possible calibration matrix $A_0 \in SL(3)/SO(3)$ is such that the matrix $X = (A_0^{-1}A)^{-1}(A_0^{-1}A)^{-T}$ is in the *symmetric real kernel* $(SRKer)$ of the Lyapunov map for all $R \in SO(2)$:

$$L : \mathbb{C}^{3\times 3} \to \mathbb{C}^{3\times 3}; \qquad X \mapsto X - RXR^T. \tag{7.8}$$

By the choice of e_1, e_2, e_3, the real eigenvector of R is e_3. Imposing $X \in SL(3)$, we obtain $X = D(s)$, where $D(s) \in \mathbb{R}^{3\times 3}$ is a matrix function of s:

$$D(s) = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1/s^2 \end{pmatrix}, \quad s \in \mathbb{R} \setminus \{0\}. \tag{7.9}$$

Geometrically, this reveals that only metric information within the plane can be recovered while the relative scale between the plane and its normal direction cannot be determined. That is, if we choose an erroneous matrix A_0 from the set of possible solutions for calibration other than the true A, then $A_0 B = A$ for some matrix $B \in SL(3)$. We then have that, for some $s \in \mathbb{R}$,

$$(A_0^{-1}A)^{-1}(A_0^{-1}A)^{-T} = D(s) \;\Rightarrow\; B^{-1}B^{-T} = D(s). \tag{7.10}$$

A solution of (7.10) is of the form $B = HD(t)$ with $H \in SO(3)$ and some $t \in \mathbb{R}$. Let us define a one-parameter Lie group $G_{SE(2)}$ as:

$$G_{SE(2)} = \{D(s) \mid s \in \mathbb{R} \setminus \{0\}\}. \tag{7.11}$$

Then the solution space of (7.10) is given by $SO(3)G_{SE(2)}$. The group $G_{SE(2)}$ can be viewed as a natural representation of ambiguous solutions in the space $SL(3)/SO(3)$.

Once we have a calibration matrix, say A_0, we can extract motion from the fundamental matrix $F = A^{-T}\widehat{T}RA^{-1}$ as follows: we know that $A = A_0B$ for some $B = HD(s) \in SO(3)G_{SE(2)}$. Then we define $E = A_0^TFA_0$ and note that, for $R = \exp(\widehat{e}_3\theta)$, we have that $D(s)$ commutes with R *i.e.*, $D(s)RD(s)^{-1} = R$ and also $H^T = H^{-1}$. Then E is an essential matrix since

$$E = H^{-T}D^{-T}(s)\widehat{T}RD^{-1}(s)H^{-1} = \widehat{HD(s)T}HRH^T.$$

The motion recovered from E is therefore $(HRH^T, HD(s)T) \in SE(3)$, where $(R,T) \in SE(2)$ is the true motion. Note that $(HRH^T, HD(s)T)$ is actually a *planar motion* (in a plane rotated by H from the original one). The coordinate transformation in the uncalibrated camera frame is given by $AT(t) = ARp(t_0) + AT(t)$. If, instead, the matrix A_0 is chosen to justify the camera calibration, the coordinate transformation becomes:

$$\begin{aligned} &A_0Bp(t) = A_0BRp(t_0) + A_0BT(t) \;\Rightarrow \\ &HD(s)p(t) = HRH^T(HD(s)p(t_0)) + HD(s)T(t). \end{aligned}$$

Therefore, any point p viewed with an uncalibrated camera A undergoing a motion $(R,T) \in SE(2)$ is not distinguishable from the point $HD(s)p$ viewed with an uncalibrated camera $A_0 = AD^{-1}(s)H^T$ undergoing a motion $(HRH^T, HD(s)T) \in SE(2)$. We have therefore proven the following

Theorem 7.6 (Ambiguity under $SE(2)$). *Consider a camera with unknown calibration matrix $A \in SL(3)$ undergoing planar motion $SE(2)$ and let $B(s) = HD(s)$ with $H \in SO(3)$ and $D(s) \in G_{SE(2)}$. If both the camera motion $(R,T) \in SE(2)$ and the scene structure $p \in \mathbb{R}^3$ are unknown, then $AB^{-1}(s) \in SL(3)$, $(HRH^T, B(s)T) \in SE(2)$ and $B(s)p \in \mathbb{R}^3$ are the only generic ambiguous solutions for the camera calibration, camera motion and scene structure respectively.*

Comment 7.7. *Note that the role of the matrix $H \in SO(3)$ is just to rotate the overall configuration. Therefore, the only generic ambiguity of the reconstruction is characterized by the one parameter Lie group $G_{SE(2)}$.*

Further note that the above ambiguities are obtained only from bilinear constraints between pairs of images. We now verify, as we expect from section 5.1, that multilinear constraints do not reduce the ambiguity. In fact, the matrix $D(s)$ commutes with the rotation matrix, so that

$$\begin{bmatrix} A & 0 & \mathbf{x}(t_1) & 0 & 0 \\ AR(t_2) & AT(t_2) & 0 & \mathbf{x}(t_2) & 0 \\ AR(t_3) & AT(t_3) & 0 & 0 & \mathbf{x}(t_3) \end{bmatrix} \begin{bmatrix} B^{-1}(s) & 0 \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} A_0 & 0 & \mathbf{x}(t_1) & 0 & 0 \\ A_0 H R(t_2) H^T & AT(t_2) & 0 & \mathbf{x}(t_2) & 0 \\ A_0 H R(t_3) H^T & AT(t_3) & 0 & 0 & \mathbf{x}(t_3) \end{bmatrix}$$

Subgroups $SO(2)$, $SO(2) \times \mathbb{R}$ and $SE(2) \times \mathbb{R}$. We conclude our discussion on subgroups of $SE(3)$ by studying $SO(2)$, $SO(2) \times \mathbb{R}$ and $SE(2) \times \mathbb{R}$ together. This is because their generic ambiguities are similar to the case of $SE(2)$, which we have just studied. Notice that in the discussion of the ambiguity $G_{SE(2)}$, we did not use the fact that the translation p has to satisfy $p_3 = 0$. Therefore, we have:

Corollary 7.8 (Ambiguity under $SO(2)\times\mathbb{R}$ and $SE(2)\times\mathbb{R}$). *The generic reconstruction ambiguities of $SO(2) \times \mathbb{R}$ and $SE(2) \times \mathbb{R}$ are exactly the same as that of $SE(2)$.*

The only different case is $SO(2)$. It is readily seen that the ambiguity of $SO(2)$ is the "product" of that of $SE(2)$ and that of $SO(3)$ due to the fact $SO(2) = SE(2) \cap SO(3)$. As a consequence of Theorem 7.4 and Theorem 7.6 we have:

Corollary 7.9 (Ambiguity under $SO(2)$). *Consider an uncalibrated camera with calibration matrix $A \in SL(3)$ undergoing a motion in $SO(2)$ and let $B(s) = HD(s)$ with $H \in SO(3)$, $D(s) \in G_{SE(2)}$ and $\lambda \in (R^+, \cdot)$. If both the camera motion $R \in SO(3)$ and the scene structure $p \in \mathbb{R}^3$ are unknown, then $AB^{-1}(s) \in SL(3)$, $HRH^T \in SO(3)$ and $\lambda \cdot B(s)p \in \mathbb{R}^3$ are the only generic ambiguous solutions for the camera calibration, camera motion and scene structure respectively.*

From the above discussion of subgroups of $SE(3)$ we have seen that generic ambiguities exist for many proper subgroup of $SE(3)$. Furthermore, such ambiguities - which

have been derived above based only on bilinear constraints, are not resolved by multilinear constraints according to Theorem 5.2.

7.2 Reprojection under Partial Reconstruction

In the previous section we have seen that, in general, it is possible to reconstruct the calibration matrix A and the scene's structure p only *up to a subgroup* - which we call K, the ambiguity subgroup. For instance, in the case of planar motion, an element in K has the form $D(s)$ given by equation (7.9). Therefore, after reconstruction we have

$$\tilde{p}(K) = Kp, \quad \tilde{A}(K) = AK^{-1}. \tag{7.12}$$

Now, suppose one wants to generate a novel view of the scene, $\tilde{\mathbf{x}}$ from a new vantage point, which is specified by a motion $\tilde{g} \in SE(3)$ and must satisfy $\tilde{\lambda}\tilde{\mathbf{x}}(K) = \tilde{A}(K)\tilde{g}\tilde{p}(K)$. In general, the reprojection $\tilde{\mathbf{x}}(K)$ depends both on the ambiguity subgroup K and on the vantage point $\tilde{g}$ and there is no guarantee that it is an image of the original Euclidean scene.

It is only natural, then, to ask what is the set of vantage points that generate a **valid reprojection**, that is an image of the original scene p taken as if the camera A was placed at some vantage point $g(K)$. We discuss this issue in section 7.2.1. A stronger condition to require is that the reprojection be independent (**invariant**) of the ambiguity K, so that we have $g(K) = \tilde{g}$ regardless of K; we discuss this issue in section 7.2.2.

7.2.1 Valid Euclidean Reprojection

In order to characterize the vantage points - specified by motions $\tilde{g}$ - that produce a valid reprojection we must find $\tilde{g}$ such that: $\tilde{A}(K)\tilde{g}\tilde{p}(K) = Ag(K)p$ for some $g(K) \in SE(3)$. Since the reprojected image $\tilde{\mathbf{x}}$ is $\tilde{\lambda}\tilde{\mathbf{x}}(K) = \tilde{A}(K)\tilde{g}\tilde{p}(K) = Ag(K)p$, the characterization of all such motions $\tilde{g}$ is given by the following Lie group:

$$R(K) = \{\tilde{g} \in SE(3) \mid K^{-1}\tilde{g}K \subset SE(3)\}. \tag{7.13}$$

We call $R(K)$ the **reprojection group** for a given ambiguity group K. For each of the generic ambiguities we studied in section 7.1, the corresponding reprojection group is given by the following

Theorem 7.10 (Reprojection Groups). *The reprojection groups corresponding to each of the ambiguity groups K studied in section 7.1 are given by:*

1. $R(K) = (\mathbb{R}^3, +)$ *for* $K = SL(3)$ *(ambiguity of* $(\mathbb{R}^3, +)$*).*
2. $R(K) = SO(2)$ *for* $K = G_{SE(2)} \times (\mathbb{R}^+, \cdot)$ *(ambiguity of* $SO(2)$*).*
3. $R(K) = SE(2) \times \mathbb{R}$ *for* $K = G_{SE(2)}$ *(ambiguity of* $SE(2), SO(2) \times \mathbb{R}, SE(2) \times \mathbb{R}$*).*
4. $R(K) = SE(3)$ *for* $K = I$ *(ambiguity of* $SE(3)$*).*

Even though the reprojected image is, in general, not unique, the family of all such images are still parameterized by the same ambiguity group K. For a motion outside of the group $R(K)$, *i.e.*, for a $\tilde{g} \in SE(3) \setminus R(K)$, the action of the ambiguity group K on a reprojected image cannot simply be represented as moving the camera: it will have to be a more general non-Euclidean transformation of the shape of the scene. However, the family of all such non-Euclidean shapes are minimally parameterized by the quotient space $SE(3)/R(K)$.

Comment 7.11 (Choice of a "Basis" for Reprojection). *Note that in order to specify the viewpoint it is not just sufficient to choose the motion* $\tilde{g}$ *for, in general,* $g(K) \neq \tilde{g}$*. Therefore, an imaginary "visual-effect operator" will have to adjust the viewpoint* $g(K)$ *acting on the parameters in* K*. The ambiguity subgroups derived in section 7.1 are one-parameter groups (for the most important cases) and therefore* the choice is restricted to one parameter. *In a projective framework (such as [21]), the user has to specify a projective basis of three-dimensional space, that is 15 parameters. This is usually done by specifying the three-dimensional position of 5 points in space.*

7.2.2 Invariant Reprojection

In order for the view taken from $\tilde{g}$ to be unique, we must have

$$\tilde{\lambda}\tilde{\mathbf{x}} = \tilde{A}(K)\tilde{g}\tilde{p}(K) = AK^{-1}\tilde{g}Kp \tag{7.14}$$

independent of K. Equivalently we must have $K^{-1}\tilde{g}K = \tilde{g}$ where K is the ambiguity generated by the motion on a subgroup G of $SE(3)$. The set of $\tilde{g}$ that satisfy this condition is a group $N(K)$, the so called **normalizer** of K in $SE(3)$. Therefore, all we have to do is to characterize the normalizers for the ambiguity subgroups studied in section 7.1.

Theorem 7.12 (Normalizers). *The set of viewpoints that are invariant to reprojection is given by the normalizer of the ambiguity subgroup. For each of the motion subgroups analyzed in section 7.1 the corresponding normalizer of the ambiguity group is given by:*

1. $N(K) = I$ *for* $K = SL(3)$ *(ambiguity of* $(\mathbb{R}^3, +)$*).*
2. $N(K) = SO(2)$ *for* $K = G_{SE(2)} \times (\mathbb{R}^+, \cdot)$ *(ambiguity of* $SO(2)$*).*
3. $N(K) = SO(2)$ *for* $K = G_{SE(2)}$ *(ambiguity of* $SE(2), SO(2) \times \mathbb{R}, SE(2) \times \mathbb{R}$*).*
4. $N(K) = SE(3)$ *for* $K = I$ *(ambiguity of* $SE(3)$*).*

For motions in every subgroup, the reprojection performed under any viewpoint determined by the groups above is unique.

7.3 Discussion

When the necessary and sufficient conditions for a unique reconstruction of scene structure, camera motion and calibration are not satisfied, it is still possible to retrieve a reconstruction up to a global subgroup action (on the entire configuration of the camera system). We characterize such subgroups explicitly for all possible motion groups of the camera. The reconstructed structure can then be re-projected to generate novel views of the scene. We characterize the "basis" of the reprojection corresponding to each subgroup, and also the motions that generate a unique reprojection. We achieve the goal by using results from two view analysis established through previous chapters. This is possible because the coefficients of multilinear constraints are geometrically dependent of those of bilinear constraints. Therefore, the only advantage in considering multilinear constraints is in the presence of singular surfaces and rectilinear motions. Our future research agenda involves the design of optimal algorithms to recover all (and only!) the parameters that can be estimated from the data based upon their generic ambiguities. The reconstruction and reprojection problem studied in this chapter is for a constant calibration matrix. Generalization to the time-varying case is yet a largely open problem.

Part II

Advanced Topics in Multiview Geometry

Chapter 8

Absolute Vision in Spaces of Constant Curvature

"In order to investigate a subfield of a science, one bases it on the smallest possible number of principles, which are to be as simple, intuitive, and comprehensible as possible, and which one collects together and sets up as axioms."
— David Hilbert, *The New Grounding of Mathematics: First Report*

In Part I, following the formulation given in Chapter 2, we have studied almost every aspect of the classical structure from motion problem in multiview geometry. Nevertheless, all the results are developed under a default assumption: the underlying space is a Euclidean space $\mathbb{E}^3$. Mathematically, it is then natural to ask: If the Euclidean assumption on the underlying space is violated, can we still study vision, and how? In order to answer this question, we need clearly understand what are all the hidden assumptions which have essentially enabled the development in Part I, and how these assumptions can be re-stated in a more abstract mathematical form so as to also work for non-Euclidean spaces. In this chapter, we attempt to provide an answer to these questions. Basically, we want to show that, under certain assumptions, it is possible to generalize multiview geometry to non-Euclidean spaces. As we will see, many results that we have obtained in Part I have their natural extensions in the non-Euclidean case and the Euclidean case in many ways can be interpreted as a special case of a non-Euclidean multiview geometry. We hope that such a generalization not only captures essential geometric characteristics of any imaging system but also provides a meaningful mathematical model in which we may gain a deeper

understanding of underlying principles of multiview geometry in general.

8.1 An Axiomatic Formulation of Multiview Geometry

Imagine an intelligent creature living *in* a sphere – a typical example of non-Euclidean space. Then what kind of multiview geometry it could have developed? Let us put ourselves in the shoes of the creature and try to understand what are the basic elements of which a vision system in such a space must consist. In this section, we give an axiomatic formulation of a mathematical model of an abstract vision system (in a Riemannian manifold). Although this model seems to be given in a rather abstract manner, it is a natural generalization of the conventional camera model in a Euclidean space. Such a generalization allows us to fully discover the geometric nature of a computer vision system, in a very concise and precise way.

Let us consider a (connected) Riemannian manifold (M, Φ), *i.e.*, a differentiable manifold equipped with a positive definite symmetric 2-form Φ as its metric. If the reader is not familiar with differential geometry, he or she may simply view (M, Φ) as the Euclidean space $\mathbb{R}^3$ with its standard inner product metric. In this paper, we will be mostly interested in three dimensional spaces although the model given below is for the most general case.

Assumption 8.1 (Camera). *A camera is modeled as a point $o \in M$, which usually stands for the optical center of the camera, and an orthonormal coordinate chart is chosen on T_oM, the tangent space of M at the point o.*

Assumption 8.2 (Motion). *M is a complete and orientable Riemannian manifold. G is the orientation-preserving subgroup of the isometry group of M. This group then models valid motions of the camera. Its representation might depend on the position of the optical center o.*

Assumption 8.3 (Light). *In the manifold M, light always travels along geodesics with constant speed. For simplicity, we may assume this speed to be infinite.*

Assumption 8.4 (Image). *The image of a point $p \in M$ is a ray in T_oM which corresponds to the direction of the geodesic connecting p and the optical center o.*

Assumption 8.5 (Calibration). *The effect of camera calibration can be modeled as an unknown isomorphism $\psi : T_oM \to T_oM$ (as a vector space). In the calibrated case, one may assume this isomorphism is known or simply the identity map.*

The Lie group G which models the motion of the camera is obtained in the model as being the isometry group of M. In fact the relation between G and M is symmetric at least in the case that the motion group G acting transitively on M: letting H be the isotropy subgroup[1] of G, then the manifold M is simply the quotient space G/H. The Riemannian metric Φ on M can be derived from the canonical metrics of G and H by this quotient. In practice, this viewpoint is far more useful than the above axiomatic definition since, as we will soon see, interesting manifolds are usually given as submanifolds of an Euclidean space which are invariant under the action of certain Lie groups G. Therefore, geometric properties of a vision system in such manifolds are uniquely determined by the structure of G.

As pointed out by Weinstein [127], different requirements on the properties of the motion group G in fact determine the types of manifolds that M must be. For example, if we require G act transitively on the frame bundle of M, it can be shown that M must be **spaces of constant curvature** [55]. A less restrictive requirement on G is to allow that the optical axis of the camera can point to any direction at any point of M. In this case, M is the so call **symmetric spaces of rank 1**. One can further relax the Assumption 8.2 so that G does not have to be a subgroup of the isometry group of M. Then M can be any Riemannian manifold. A study of vision theory in general Riemannian manifolds is out of the scope of this dissertation. For the remaining of this Chapter, we will focus only on the spaces of constant curvature and demonstrate how to generalize the vision theory that we have developed for Euclidean space in previous chapters.

Assumptions 8.1 to 8.5 formally define a camera model in a class of Riemannian manifolds. When the manifold M is the Euclidean space $\mathbb{E}^3$, the so obtained model is exactly equivalent to the conventional model that we have been using in Part I. Even in the most general case, the above model is based on direct geometric intuition. The only difference is that the world space (represented by M) is explicitly distinguished from the image space (represented by T_oM). In the Euclidean case, these two spaces happen to coincide. Intuitively, this can be illustrated in the Figure 8.1.

[1] A subgroup of G which fixes a point of M.

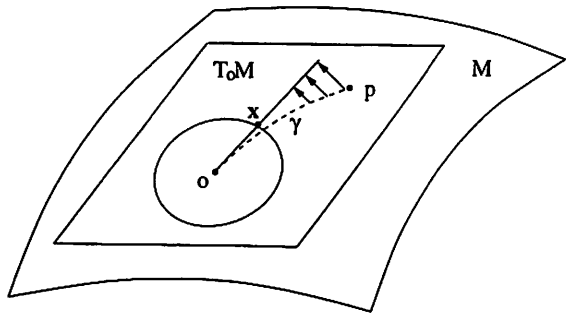


Figure 8.1: The curve γ is the geodesic connecting o and p; arrows mean the inverse of the exponential map $\exp : T_oM \to M$; $\mathbf{x}$ then represents the image of the point p with respect to a camera centered at the point o.

8.2 Non-Euclidean Multiview Geometry in Spaces of Constant Curvature

Can the abstract model introduced in the preceding section of any use? In this section we will demonstrate that, using this model, one can actually extend most of the results that we have developed in Part I for Euclidean space to a much larger class of spaces: the spaces of constant curvature. For example, the epipolar geometry has no peculiar meaning to Euclidean space. It is also true in more general spaces. For simplicity, in this chapter we will only investigate the calibrated case although extension to uncalibrated case is straightforward.

8.2.1 Spaces of Constant Curvature

Spaces of constant curvature are Riemannian manifolds with constant sectional curvature. In differential geometry, they are also referred to as **space forms**. A Riemannian manifold of constant curvature is said to be **spherical**, **hyperbolic** or **flat** (or **locally Euclidean**) according as the sectional curvature is positive, negative or zero. Geometry about spaces of constant curvature is also called **absolute geometry**, coined by one of the co-founders non-Euclidean geometry: Janos Bolyai [46].

Not until Einstein's general relativity theory, non-Euclidean geometry, or Riemannian geometry in general, is just a pure mathematical creation rather than geometry of physical spaces. In general relativity theory, the physical space is typically described as a (3 dimensional) Riemannian manifold (with possibly non-zero curvature). In such a space,

light travels the **geodesics** of the manifold (corresponding to straight lines in the Euclidean case). Locally, the curvature of a Riemannian manifold is approximately constant. Thus the study of vision theory in spaces of constant curvature will help understand vision problems in general Riemannian manifolds.

In this paper, we study vision theory in 3 dimensional spaces of constant curvature, as a natural generalization of the vision theory we have developed so far for 3 dimensional Euclidean space. In particular, we will focus on vision in spherical and hyperbolic spaces since the Euclidean case has been well understood. On the other hand, the Euclidean case will always show up as a special limit case of generic cases.

Geometric properties of n dimensional space of constant curvatures have been well studied in differential geometry [55, 135] (as an important case of **symmetric spaces**). In the rest of this section, we briefly *review* some of the main results which serve for our purposes.

8.2.2 Characteristics of Spaces of Constant Curvature

In this section, we characterize 3 dimensional spaces of constant curvature. In fact, most of the results directly follow from general results about n dimensional spaces of constant curvature, in Kobayashi [55] and Wolf [135].

The next theorem which follows directly from Kobayashi [55] (Theorem 3.1 Chapter V) characterizes the 3 dimensional space of constant curvatures:

Proposition 8.6 (3D Spaces of Constant Curvature). *Let $[x_1, x_2, x_3, x_4]^T$ be the coordinate system of $\mathbb{R}^4$ and M be the hyper-surface of $\mathbb{R}^4$ defined by:*

$$x_1^2 + x_2^2 + x_3^2 + r x_4^2 = r \quad (r\text{: a nonzero constant}). \tag{8.1}$$

Let g be the Riemannian metric of M obtained by restricting the following form to M:

$$dx_1^2 + dx_2^2 + dx_3^2 + r\, dx_4^2.$$

Then

1. *M is a 3 dimensional space of constant curvature with sectional curvature $1/r$.*
2. *The group G of linear transformations of $\mathbb{R}^4$ leaving the quadratic form $x_1^2 + x_2^2 + x_3^2 + r x_4^2$ invariant acts transitively on M as the group of isometry of M.*

3. *If $r > 0$, then M is isometric to a sphere of a radius $\sqrt{r}$. If $r < 0$, then M consists of two mutually isometric connected manifolds each of which is diffeomorphic with $\mathbb{R}^3$.*

Now let Q be the 4×4 matrix associated to the quadratic form defining M:

$$Q = \begin{bmatrix} I_3 & 0 \\ 0 & r \end{bmatrix}.$$

The **isometry group** G of M is then given as a subgroup of $GL(4, \mathbb{R})$:

$$G = \left\{ g \in \mathbb{R}^{4\times 4} \,\middle|\, g^T Q g = Q \right\}. \tag{8.2}$$

For an element $g \in G$, it has the form:

$$g = \begin{bmatrix} W & y \\ z^T & w \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

with $W \in \mathbb{R}^{3\times 3}, y \in \mathbb{R}^3, z \in \mathbb{R}^3, w \in \mathbb{R}$ and the conditions:

$$W^T W + r \cdot z z^T = I_3, \quad W^T y + r \cdot w z = 0, \quad y^T y + r \cdot w^2 = r. \tag{8.3}$$

It follows that the **Lie algebra** $\mathfrak{g}$ of the group G (as a Lie group) is the set of the matrices of the form:

$$\xi = \begin{bmatrix} A & b \\ c^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{8.4}$$

where $A \in \mathbb{R}^{3\times 3}, b \in \mathbb{R}^3$ and $c \in \mathbb{R}^3$ satisfy the conditions:

$$A^T + A = 0, \quad b + r \cdot c = 0. \tag{8.5}$$

The **isotropy group** H of G which leaves the point $o = [0, 0, 0, 1]^T \in M$ fixed is isomorphic to $O(3)$:

$$H = \begin{bmatrix} O(3) & 0 \\ 0 & 1 \end{bmatrix}. \tag{8.6}$$

As a result, the manifold M is identified with the **homogeneous space** G/H. In fact, the orthonormal frame bundle of M is isomorphic to G as a principle H bundle, Kobayashi [55].

Let $\mathfrak{m}$ be the linear subspace of the Lie algebra $\mathfrak{g}$ of G consisting of matrices of the form:

$$\begin{bmatrix} 0 & b \\ c^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{8.7}$$

with $b, c \in \mathbb{R}^3$ and $b + rc = 0$. Let $\mathfrak{h}$ be the Lie algebra of H as a subspace of $\mathfrak{g}$ consisting of matrices of the form:

$$\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{8.8}$$

with $A \in \mathbb{R}^{3\times 3}$ and $A^T + A = 0$. Then we have a **canonical decomposition**:

$$\mathfrak{g} = \mathfrak{h} + \mathfrak{m}. \tag{8.9}$$

It is direct to check the following relations between the subspaces hold:

$$[\mathfrak{h}, \mathfrak{h}] \subset \mathfrak{h}, \quad [\mathfrak{h}, \mathfrak{m}] \subset \mathfrak{m}, \quad [\mathfrak{m}, \mathfrak{m}] \subset \mathfrak{h} \tag{8.10}$$

where $[\cdot, \cdot]$ stands for **Lie bracket**. Let $\mathfrak{h}$ be the **vertical tangent subspace** of G and $\mathfrak{m}$ be the **horizontal tangent subspace**. Then this decomposition gives a **canonical connection** on the principle bundle $G(G/H, H)$ (Theorem 11.1 of Chapter II, Kobayashi [55]) which induces constant sectional curvature $1/r$ on $G/H = M$.

The space M is a symmetric space with the symmetry s_o of M at the point $o = [0, 0, 0, 1]^T$ given by:

$$\begin{aligned} s_o : M &\to M \\ [x_1, x_2, x_3, x_4]^T &\mapsto [-x_1, -x_2, -x_3, x_4]^T. \end{aligned}$$

Obviously, $s_o^2 = Id(M)$. Due to Kobayashi [55] (Theorem 1.5 of Chapter XI), this induces a (symmetric) automorphism σ on G such that H lies between G_σ (subgroup of G fixed under σ) and the identity component of G_σ.

Denote the projection from G to G/H as π and Let $\exp(\cdot)$ be the exponential map from $\mathfrak{g}$ to G. Then according to Kobayashi [55] (Theorem 3.2 of Chapter XI), we have:

Proposition 8.7 (Geodesics in 3D Spaces of Constant Curvature). *Consider the 3 dimensional space of constant curvature $M = G/H$ as above. For each $X \in \mathfrak{m}$, $\pi(\exp(tX)) = \exp(tX) \cdot o$ is a geodesic starting from o and, conversely, every geodesic from o is of this form.*

As we will soon see, this theorem is very important for modeling and studying vision in the spaces of constant curvature.

Let T be the subset of G consisting of all the matrices of the form $\exp(X)$ with $X \in \mathfrak{m}$. Then T corresponds to **transvection** on M (see Kobayashi [55]), an analogy to the translation in the Euclidean space. Notice that in general T is not a subgroup of G (although it is in the Euclidean case) and its representation depends on the base point. Naturally, the subgroup H of G corresponds to **rotation** on M. As in the Euclidean case, for a "rigid body motion" on M, it is natural to consider the rotation is in the special orthogonal group $SO(3)$ instead of the full group $O(3)$. One of the reasons for only considering $SO(3)$ is that it preserves the orientation of the space.

8.2.3 Euclidean Space as a Space of Constant Curvature

Proposition 8.6 requires the curvature parameter $r \in \mathbb{R} \setminus \{0\}$ hence only the spherical and hyperbolic spaces were considered. However, the Euclidean case can be regarded as the limit case when r goes to infinite, *i.e.*, the curvature $1/r$ goes to zero.

When $r = \infty$, a point in $\mathbb{R}^4$ which satisfies the quadratic form (8.1) always has the form $[x_1, x_2, x_3, 1]^T \in \mathbb{R}^4$. This is just the homogeneous representation of the 3 dimensional Euclidean space $\mathbb{R}^3$, see Murray [84]. From (8.3), we have $w^2 = 1$, $z = 0$, $W^T W = I_3$ and $y \in \mathbb{R}^3$. Thus the group G is just the Euclidean group $E(3)$. In particular, the special Euclidean group $SE(3)$ with elements:

$$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{8.11}$$

with $R \in SO(3)$ and $T \in \mathbb{R}^3$ is a subgroup of $G = E(3)$. $SE(3)$ then represents the **rigid body motion** in $M = \mathbb{R}^3$.

When $r = \infty$, the Lie algebra $se(3)$ of $SE(3)$ or $e(3)$ of $E(3)$ then has the form given in (8.4) with the condition $c = 0$. In robotics literature [84], an element this Lie algebra is usually represented as:

$$\xi = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{8.12}$$

where $\omega, v \in \mathbb{R}^3$ and $\widehat{\omega}$ is the skew-symmetric matrix associated with $\omega = [\omega_1, \omega_2, \omega_3]^T$:

$$\widehat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}. \tag{8.13}$$

According to Proposition 8.7, the geodesics in $\mathbb{R}^3$ are given in the form:

$$\exp\left(t \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} I_3 & vt \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{8.14}$$

This is exactly the straight line in $\mathbb{R}^3$ in the direction of v.

From the above discussion, the Euclidean space can be treated as a limit case of general spaces of constant curvature given in Proposition 8.6. Because of this, the vision theory for Euclidean space should also be a limit case of vision theory for general spaces of constant curvature.

8.2.4 Camera Motion and Projection Model

Based upon the mathematical facts given in the preceding section, we are ready to study vision in the spaces of constant curvature. Similar to the Euclidean case, we first need to specify the (valid) motion of the camera and the projection model of the camera, *i.e.*, how the 2 dimensional image is formulated in spaces of constant curvature.

First notice that, as in the Euclidean case, the transvection set T of the isometry group G acts transitively on a space M of constant curvature. Then for any $g \in G$, there exists $g_t \in T$ such that $g_t^{-1}(g(o)) = o$, *i.e.*, $g_t^{-1}g$ fixes the origin o. So $g_t^{-1}g = g_h \in H$, the isotropy group of o. It then follows that the group G is equal to $G = TH$. This is the so-called **Cartan decomposition**. By **rigid body motion** in spaces of constant curvature, we mean the connected subgroup of G which preserve the orientation of the space M. That is, the rotation group H is just $SO(3)$ (the subgroup of $O(3)$ which is connected to the identity element). We still use G to denote the group of rigid body motion:

$$G = TH \quad \text{with } H \in SO(3).$$

A point p, in the space M of constant curvature, can be represented in **homogeneous coordinates** as $p = [p_1, p_2, p_3, p_4]^T \in \mathbb{R}^4$ which satisfies the quadratic form:

$$p_1^2 + p_2^2 + p_3^2 + rp_4^2 = r$$

with $1/r$ the sectional curvature of M. Then under the motion $g(t) \in G, t \in [t_0, t_f] \subset \mathbb{R}$ of the camera, the homogeneous coordinates of the point p (with respect to the camera frame) satisfy the transformation:

$$p(t) = g(t)p(t_0). \tag{8.15}$$

Notice that, with this representation, the point $o = [0,0,0,1]^T \in \mathbb{R}^4$ is always in M. We then call the point o the **origin** in the homogeneous representation of M. Without loss of generality, the origin is identified with the center of the camera.

According to Proposition 8.7, any geodesic connecting a point $p = [p_1, p_2, p_3, p_4]^T \in M$ to the origin o has the form: $p = \exp(tX) \cdot o$ for some $t \in \mathbb{R}, X \in \mathfrak{m}$. Without loss of generality, we may assume X has the form:

$$X = \begin{bmatrix} 0 & b \\ -b^T/r & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

for some unit vector $b \in \mathbb{R}^3, \|b\| = 1$. It is then direct to check that:

$$p = \exp(tX) \cdot o = \begin{bmatrix} f(r,t)bb^T & h_1(r,t)b \\ h_2(r,t)b^T & g(r,t) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1(r,t)b \\ g(r,t) \end{bmatrix} \in \mathbb{R}^4$$

for some real scalar functions $f(r,t), g(r,t), h_1(r,t)$ and $h_2(r,t)$ of r and t (the explicit expressions of these functions are given in the next section). Thus the unit vector b is equal to:

$$b = \frac{[p_1, p_2, p_3]^T}{\sqrt{p_1^2 + p_2^2 + p_3^2}}. \tag{8.16}$$

This is exactly the unit tangent vector of M at the origin o. In this way, we may identify the tangent space $T_o(M)$ of M at o to the subspace $\mathfrak{m}$ by:

$$\begin{aligned} \phi : T_o(M) &\to \mathfrak{m} \\ b \in T_o(M) &\mapsto \begin{bmatrix} 0 & b \\ -b^T/r & 0 \end{bmatrix} \in \mathfrak{m}. \end{aligned}$$

Under this identification, the exponential map $\exp : T_o(M) \to M$ is given by:

$$\exp(b) = \exp(\phi(b)) \cdot o, \quad b \in T_o(M).$$

Then from previous discussion, the light from $p = [p_1, p_2, p_3, p_4] \in M$ to the origin o has the direction $b \in T_o(M)$ given by (8.16). In homogeneous coordinate, the vector b can be represented as

$$b = [p_1, p_2, p_3]^T \in \mathbb{R}^3$$

which only keeps the information of the direction of the light from p.

Then in the case of the space M of constant curvature, if the space M is represented by the homogeneous coordinates as above, the image of a point $p = [p_1, p_2, p_3, p_4]^T \in M$ is simply given by $\mathbf{x} = \lambda^{-1}[p_1, p_2, p_3]^T \in \mathbb{R}^3$ where $\lambda \in \mathbb{R}^+$ and $\mathbf{x} \in \mathbb{R}^3$. Define the **projection matrix** to be:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 4}.$$

We then have the relation:

$$\lambda \mathbf{x} = Pp. \tag{8.17}$$

We call the scalar λ the **scale** of the point p with respect to the image $\mathbf{x}$. The scale λ then encodes the depth information of the point p in the scene.

8.2.5 Epipolar Geometry and Multilinear Constraints

In this section, we study the relation between the images of a point $p \in M$ before and after a rigid body motion of the camera. We know that the motion of the camera can be expressed in the form:

$$g = g_t \cdot g_h, \quad g_t \in T, g_h \in H.$$

The transvection part g_t and rotation part g_h respectively have the forms:

$$g_t = \exp(X) = \begin{bmatrix} W & y \\ z^T & w \end{bmatrix}, \quad g_h = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}, \quad X \in \mathfrak{m}, R \in SO(3). \tag{8.18}$$

We will later give the expressions of $W \in \mathbb{R}^{3\times 3}, y \in \mathbb{R}^3, z \in \mathbb{R}^3$ and $w \in \mathbb{R}$ in terms of X.

Denote the images of $p = [p_1, p_2, p_3, p_4]^T$ before and after the transformation g are $\mathbf{x}_1 \in \mathbb{R}^3$ and $\mathbf{x}_2 \in \mathbb{R}^3$, respectively. Then according to (8.15) and (8.17) we have:

$$\lambda_1 \mathbf{x}_1 = Pp, \quad \lambda_2 \mathbf{x}_2 = Pgp.$$

It yields:

$$\lambda_2 \mathbf{x}_2 = WR \cdot \lambda_1 \mathbf{x}_1 + p_4 y \;\Rightarrow\; y \times \lambda_2 \mathbf{x}_2 = y \times (WR \cdot \lambda_1 \mathbf{x}_1) \;\Rightarrow\; \mathbf{x}_2^T \widehat{y} WR \mathbf{x}_1 = 0. \quad (8.19)$$

In the Euclidean case, (8.19) would exactly give the well-known bilinear epipolar constraint. In the case of spaces of constant curvature, the role of essential matrix is replaced by $\widehat{y}WR$. We need to study the structure of such matrices.

Any matrix $X \in \mathfrak{m}$ has the form:

$$\begin{bmatrix} 0 & b \\ -b^T/r & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

with vector $b \in \mathbb{R}^3$. To simply the notation, denote $\gamma = \|b\|$ and $T = b/\gamma \in \mathbb{R}^3$. Here by abuse of language, we use the same notation T as the transvection group to represent a translational vector. We consider $\sin(\cdot)$ and $\cos(\cdot)$ as the complex functions:

$$\begin{aligned} \sin(u) &= \frac{1}{2i}(e^{iu} - e^{-iu}), \quad u \in \mathbb{C} \\ \cos(u) &= \frac{1}{2}(e^{iu} + e^{-iu}), \quad u \in \mathbb{C}. \end{aligned}$$

Also define $\rho = \sqrt{1/r} \in \mathbb{C}$. Then through direct calculation we get:

$$\exp(X) = \begin{bmatrix} W & y \\ z^T & w \end{bmatrix} = \begin{bmatrix} I_3 + (\cos(\gamma\rho) - 1)TT^T & \rho^{-1}\sin(\gamma\rho)T \\ \rho\sin(\gamma\rho)T^T & \cos(\gamma\rho) \end{bmatrix}. \quad (8.20)$$

Notice that we always have $\widehat{T}TT^T = 0$. Then suppose $\sin(\gamma\rho) \neq 0$, (8.19) yields:

$$\mathbf{x}_2^T \widehat{T} WR \mathbf{x}_1 = 0 \quad \Leftrightarrow \quad \mathbf{x}_2^T \widehat{T}(I_3 + (\cos(\gamma\rho) - 1)TT^T)R\mathbf{x}_1 \quad \Leftrightarrow \quad \mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 = 0. \quad (8.21)$$

This is exactly the well-known bilinear **epipolar constraint**. Here we see that this constraint holds for all spaces of constant curvature. As in the Euclidean case, we call $E = \widehat{T}R$ the **essential matrix**.

Comment 8.8. *The condition* $\sin(\gamma\rho) \neq 0$ *is equivalent to the condition that the translation* $T \neq 0$ *in the Euclidean case. The reason is when* $\sin(\gamma\rho) = 0$, *we have* $\exp(X) = I_4$, *i.e., the motion is equivalent to the identity transformation on* M. *In spaces of constant curvature, we may have* $\sin(\gamma\rho) = 0$ *without* $T = 0$. *This occurs only when the curvature* r *is positive, i.e., the space is spherical. If so, let* $\gamma = 2k\pi\sqrt{r} \in \mathbb{R}, k = 1, 2, \ldots$, *we then have* $\sin(\gamma\rho) = \sin(2k\pi) = 0$. *This implies that translation with distance* $2\pi\sqrt{r}$ *along the geodesics*

(big circles) in a spherical space of radius $\sqrt{r}$ is equivalent to the identity transformation (back to the initial position). One can simply check this phenomenon on the 2 dimensional sphere S^2.

As a summary of the above discussion, we have the following theorem:

Theorem 8.9 (Epipolar Constraint). *Consider a rigid body motion of a camera in a space M of constant curvature. If $T \in \mathbb{R}^3$ is the vector associated to the direction of the translation and $R \in SO(3)$ the rotation, then the images $\mathbf{x}_1 \in \mathbb{R}^3$ and $\mathbf{x}_2 \in \mathbb{R}^3$ of a point $p \in M$ before and after the motion satisfy the epipolar constraint:*

$$\mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 = 0. \tag{8.22}$$

As in the Euclidean case, the normalized essential matrix $E = \widehat{T}R$ can be estimated from more than eight image correspondences $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^n, n \geq 8$ in general positions using linear or nonlinear estimation schemes given in Part I. The rotation matrix R and the translation vector T can further be recovered from the essential matrix E.

Comment 8.10. *Notice that the epipolar constraint is independent of the scale λ of the point p, the scale γ of the translational motion b and the curvature $1/r$ of the space M. The motion recovery is then decoupled from the 3D structure, as in the Euclidean case.*

It is already known that in the Euclidean case, m images of a point satisfy more general multilinear constraints besides the bilinear epipolar constraint. Similar constraints exist in the case of spaces of constant curvature. Suppose $\mathbf{x}_i \in \mathbb{R}^3, i = 1, 2, \ldots, m$ are m images of the same point p with respect to the camera at m different position. Suppose the relative motion between the i^{th} and 1^{th} positions is $g_i \in G, i = 1, 2, \ldots, m$. Without loss of generality, we may always assume $g_1 = I$. Let $\lambda_i \in \mathbb{R}^+, i = 1, 2, \ldots, m$ be the scales of $\mathbf{x}_i, i = 1, 2, \ldots, m$ with respect to p. Then we have the following equation (for a calibrated camera):

$$\begin{bmatrix} \mathbf{x}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_m \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix} = \begin{bmatrix} Pg_1 \\ Pg_2 \\ \vdots \\ Pg_m \end{bmatrix} p.$$

Now define the **motion matrix** $M^a \in \mathbb{R}^{3m\times 4}$ to be:

$$M^a = \begin{bmatrix} Pg_1 \\ Pg_2 \\ \vdots \\ Pg_m \end{bmatrix} \in \mathbb{R}^{3m\times 4}$$

and the four columns of M^a are denoted by $\vec{m}_1, \vec{m}_2, \vec{m}_3, \vec{m}_4$ respectively. Here we use superscript a to indicate the case of *absolute* vision. Define the vector $\vec{x}_i \in \mathbb{R}^{3m}$ associated to the i^{th} image $\mathbf{x}_i$ as:

$$\vec{x}_i = [0, \ldots, 0, \mathbf{x}_i^T, 0, \ldots, 0]^T \in \mathbb{R}^{3m}, \quad 1 \leq i \leq m.$$

Similar to the Euclidean case [71], in spaces of constant curvature, we also have:

Theorem 8.11 (Multilinear Constraints). *Consider m images $\{\mathbf{x}_i\}_{i=1}^m \in \mathbb{R}^3$ of a point p in a space M of constant curvature, and the motion matrix is $M^a = [\vec{m}_1, \vec{m}_2, \vec{m}_3, \vec{m}_4] \in \mathbb{R}^{3m\times 4}$ as defined above. Then the associated vectors $\{\vec{x}_i\}_{i=1}^m \in \mathbb{R}^{3m}$ satisfy the following wedge product equation:*

$$\vec{m}_1 \wedge \vec{m}_2 \wedge \vec{m}_3 \wedge \vec{m}_4 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_m = 0. \tag{8.23}$$

The proof is essentially the same as the Euclidean case in Chapter 5. The reason that this wedge product constraint is called **projective constraint** is because it is invariant under projective transformation (see [71]). For the same reasons as in Euclidean case, the non-trivial constraints given by the wedge product equation are either **bilinear, trilinear** or **quadrilinear**. One may use these constraints to design more delicate motion estimation schemes.

8.2.6 Non-Euclidean Structure from Motion

Knowing motion, the next problem is how to reconstruct the scale information from images, which includes the depth λ of the point p with respect to its image $\mathbf{x}$, the scale of the translational motion p and, if possible, the constant curvature $1/r$ of the space M (but we will soon see, the curvature cannot be recovered from vision). Although our

formulation allows to study reconstruction from multiple image frames, we here simply demonstrate the case of two image frames so as to convey the main ideas.

To simplify the notation, in this section, we assume the image $\mathbf{x}$ of a point p is always normalized, *i.e.*, $\|\mathbf{x}\| = 1$ (in the Euclidean case, this corresponds to the spherical projection). Suppose the distance from p to the optical center o is $\eta \in \mathbb{R}^+$. Recall that $\phi(\cdot)$ is the map from $T_o(M)$ to $\mathfrak{m}$. Then the homogeneous coordinate of p is given in terms of $\mathbf{x}$ and η by:

$$p = \exp(\eta\phi(\mathbf{x})) \cdot o = \begin{bmatrix} \rho^{-1}\sin(\eta\rho)\mathbf{x} \\ \cos(\eta\rho) \end{bmatrix} \in \mathbb{R}^4.$$

Consequently, the scale λ of p with respect to $\mathbf{x}$ is given by $\lambda = \rho^{-1}\sin(\eta\rho)$. To differentiate from the scale λ, the distance quantity η will be called the **depth** of the point p with respect to the image $\mathbf{x}$.

Let η_1 and η_2 be the depths of the point p with respect its two images $\mathbf{x}_1$ and $\mathbf{x}_2$ taken by the camera at two positions, respectively. Suppose the camera motion $g \in G$ is specified by the rotation $R \in SO(3)$, the translation direction $T \in S^2$ and the scale of translation γ (as in the preceding section). Then the first equation in (8.19) yields:

$$\rho^{-1}\sin(\eta_2\rho)\mathbf{x}_2 = \left[I_3 + (\cos(\gamma\rho) - 1)TT^T\right] R\rho^{-1}\sin(\eta_1\rho)\mathbf{x}_1 + \cos(\eta_1\rho)\rho^{-1}\sin(\gamma\rho)T. \quad (8.24)$$

This is the **coordinate transformation formula in spaces of constant curvature.** It looks kind of complicated. However, it is no more than a natural generalization of the Euclidean coordinate transformation formula which people are familiar with. Notice when the curvature $1/r$ goes to zero, so does ρ. Since

$$\lim_{\rho\to 0}\cos(x\rho) = 1, \quad \lim_{\rho\to 0}\rho^{-1}\sin(x\rho) = x, \quad x \in \mathbb{R},$$

when the curvature of the space goes to zero, we have:

$$\lambda_i = \lim_{\rho\to 0}\rho^{-1}\sin(\eta_i\rho) = \eta_i, \quad i = 1, 2,$$

and (8.24) simply becomes:

$$\lambda_2\mathbf{x}_2 = R\lambda_1\mathbf{x}_1 + \gamma T. \quad (8.25)$$

That it, in the limit case, the scale λ and the depth η are the same; and the equation (8.24) gives the Euclidean coordinate transformation formula. The Euclidean transformation (8.25) is extensively used for reconstructing Euclidean structure in Part I. Naturally,

to reconstruct structure in spaces of constant curvature, the equation (8.24) has to be exploited.

Notice that equation (8.24) is homogeneous in the scale of ρ. Since the quantities η_1, η_2 and γ are all multiplied with ρ, they can only be determined with respect to an arbitrary scale of ρ. In Euclidean case, this corresponds to the fact that the Euclidean structure can only be reconstructed up to a universal scale [71]. Thus in the case of spaces of constant curvature, we may normalize everything with respect to the scale of the curvature: if $r > 0$, let $\rho = 1$; if $r < 0$, let $\rho = i = \sqrt{-1}$. That is, now the space M has constant sectional curvature of either $+1$ or -1. Then (8.24) becomes:

$$\begin{aligned} \sin(\eta_2)\mathbf{x}_2 &= \left[I_3 + (\cos(\gamma) - 1)TT^T\right] R \cdot \sin(\eta_1)\mathbf{x}_1 + \cos(\eta_1)\sin(\gamma)T, \quad \rho = 1; \\ \sinh(\eta_2)\mathbf{x}_2 &= \left[I_3 + (\cosh(\gamma) - 1)TT^T\right] R \cdot \sinh(\eta_1)\mathbf{x}_1 + \cosh(\eta_1)\sinh(\gamma)T, \quad \rho = i. \end{aligned}$$

These two equations correspond to coordinate transformations in (normalized) spherical and hyperbolic spaces, respectively.

From the preceding section, we know R and T can be estimated from epipolar constraints. The problem left is to reconstruct η_1, η_2 and γ. In computer vision, this problem is usually referred to as **structure from motion** (this name is used by some authors for the problem of reconstructing both motion and structure from images, but we shall maintain the distinction here). One may directly use the above coordinate transformation formula to formulate objective function for estimating scales η_1, η_2 and γ. In the Euclidean case, such objective functions are linear in the scales [71]. However, in the Non-Euclidean case, such objective functions are usually nonlinear.

In stead of directly using the coordinate transformation formula, one may use some well-known constraints in spaces of constant curvature, *i.e.*, **Bolyai's law of sine** and **law of cosine** (for absolute geometry), which have been well summarized by Hsiang in [46]. Define functions:

$$\alpha(x) = \begin{cases} \sin(x), & \rho = 1, \\ \sinh(x), & \rho = i, \end{cases} \qquad \beta(x) = \begin{cases} \cos(x), & \rho = 1, \\ \cosh(x), & \rho = i. \end{cases}$$

The next theorem follows from Hsiang [46] as a special case:

Proposition 8.12 (Laws of Absolute Trigonometry). *Consider a geodesic triangle $\triangle ABC$ in a space M of constant curvature ± 1, and let a, b, c be the lengths of the opposite*

sides of angles A, B, C respectively. Then we have:

$$\frac{\sin(A)}{\alpha(a)} = \frac{\sin(B)}{\alpha(b)} = \frac{\sin(C)}{\alpha(c)}, \qquad \textit{Bolyai's sine law.} \tag{8.26}$$

and

$$\begin{aligned} \alpha(a)\alpha(b)\cos(C) &= \beta(c) - \beta(a)\beta(b), \\ \alpha(b)\alpha(c)\cos(A) &= \beta(a) - \beta(b)\beta(c), \qquad \textit{law of cosine} \\ \alpha(c)\alpha(a)\cos(B) &= \beta(b) - \beta(c)\beta(a). \end{aligned} \tag{8.27}$$

Suppose the two optical centers of the camera are o_1 and o_2. A geodesic triangle is formed by the three points (o_1, o_2, p), see Figure 8.2. The angle A is given by the angle be-

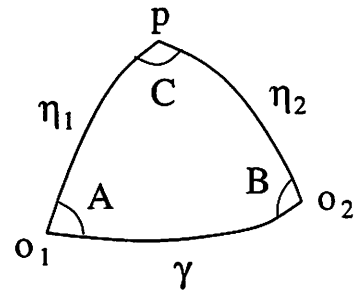


Figure 8.2: Geodesic triangle formed by two optical centers o_1, o_2 and a point p in the scene.

tween the two vectors $R\mathbf{x}_1$ and $-T$; B is given by the angle between $\mathbf{x}_2$ and T; C is given by the angle between $R\mathbf{x}_1$ and $\mathbf{x}_2$. The quantities $\sin(A), \sin(B), \sin(C), \cos(A), \cos(B), \cos(C)$ can be directly calculated from those vectors.

Applying Bolyai's sine law (8.26) to the geodesic triangle, $\alpha(\eta_1), \alpha(\eta_2)$ and $\alpha(\gamma)$ are determined up to a unknown scalar $k \in \mathbb{R}$ by linear equations:

$$\sin(A)\alpha(\eta_1) = \sin(B)\alpha(\eta_2), \quad \sin(C)\alpha(\eta_2) = \sin(A)\alpha(\gamma). \tag{8.28}$$

The scalar k can be then determined by using one of the cosine law (8.27). Suppose

$$(s_1, s_2, s_3)^T = (k\alpha(\eta_1), k\alpha(\eta_2), k\alpha(\gamma))^T \in \mathbb{R}^3$$

is a solution of (8.28). In the hyperbolic case, from the first equation of (8.27), the scalar k satisfies:

$$s_1 s_2 \cos(C) = k\sqrt{s_3^2 - k^2} - \sqrt{(s_1^2 - k^2) \cdot (s_2^2 - k^2)}. \tag{8.29}$$

In the spherical case, we may assume $0 \leq \eta_1, \eta_2, \gamma \leq \pi/2$ (*i.e.*, comparing to the size of the whole space, the structure we consider is relatively small). Then the first equation of (8.27) yields:

$$s_1 s_2 \cos(C) = k\sqrt{k^2 - s_3^2} - \sqrt{(k^2 - s_1^2) \cdot (k^2 - s_2^2)}. \tag{8.30}$$

In order to calculate k, the above equations can be easily reduced to algebraic equations in k^2 of degree 4. Since there is a general formula for roots of algebraic equations of degree 4, **k has a closed-form solution.** Knowing k, $\alpha(\eta_1), \alpha(\eta_2)$ and $\alpha(\gamma)$ can be calculated hence η_1, η_2 and γ. The above approach clearly outlines the geometry of stereo in any space of constant curvature.

8.3 Discussion

In this chapter, we have generalized basic vision theorems in Euclidean space to spaces of constant curvature. A uniform treatment is possible because a unified homogeneous representation of these spaces exists and the isometry groups of these spaces have similar structures. As we have seen, the Euclidean vision theory can always be viewed as a limit case of the general one.

One may have noticed that the epipolar geometry in spaces of constant curvature is remarkably similar to that of Euclidean space. Especially, the bilinear epipolar constraint is exactly the same. As in the Euclidean case, the motion is nicely decoupled from structure by the epipolar constraint. This allows us to use most of the motion recovery algorithms which were previously developed only for Euclidean space to spherical and hyperbolic spaces, without any modification. In the continuous case, the epipolar geometry also remains to be the same as in Euclidean case.

As for the structure from motion problem, the three dimensional structure can only be reconstructed up to a universal scale, the same as the Euclidean case. In a space of non-zero curvature, the curvature of the space cannot be recovered from vision. However, the three dimensional structure of objects can be determined with respect to the curvature. In this paper, we normalize the curvature with absolute value 1. Although the (noise-free) structure from motion can be solved as a linear problem in the Euclidean case, it is no longer linear for spherical and hyperbolic spaces. We have shown that using sine and cosine laws for Absolute Geometry there is a closed-form solution for the structure from motion

problem.

Although any Riemannian manifold locally can be approximated by spaces of constant curvature (when the sectional curvature in all directions is close each other), it is still interesting to know if the results of epipolar geometry hold for more general classes of Riemannian manifolds (for example, symmetric spaces); and how the structure from motion problem needs to be changed in general. These will be interesting research topics for the future.

Chapter 9

Bayesian Motion Estimation: Likelihood and Geometry

In Part I and the previous chapter, we have established motion (and structure) recovery schemes using image point features: image correspondences in the discrete case and optical flows in the continuous case. However, point features are *not* a type of measurements that images directly provide. A raw image is better modeled to be a real function defined on the image plane $I : \mathbb{RP}^2 \to \mathbb{R}$ which indicates the gray level of the image intensity. The purpose of this chapter is trying to study the motion estimation problem from this level of raw inputs. The concept of point feature hence must be derived. Based on a very simplified noise model, we are going to establish an argument for why the use of point features (in the Part I) is *the* correct thing to do for motion estimation. More strictly speaking, under certain conditions and assumptions, point features are sufficient statistics for motion estimation. Very much like the previous chapter, the study in this chapter is only conceptual and suggestive. The emphasis is on analysis but not on algorithm. In the end, we will discuss how the feature point based approach may fail when certain assumptions of the model are violated. This discussion – although does not really undermine all the study given in Part I and Chapter 8 – will indeed reveal a more complicated picture of motion estimation in general.

9.1 Image Noise Models

For a perspective image point, we still use $\mathbf{x} = [x_1, x_2, 1]^T$ to denote its homogeneous coordinates. In this chapter, we will use the vector $x = [x_1, x_2]^T \in \mathbb{R}^2$ to denote its 2D coordinates. Let $t \in \mathbb{R}$ denote the time. Points on the image plane evolve according to some vector field – also called **image velocity** in computer vision literature:

$$\dot{x} = \phi(x, t). \tag{9.1}$$

Let $\Phi_x(t)$ denote the solution of this ODE with x as the initial state. That is:

$$\dot{\Phi}_x(t) = \phi(\Phi_x(t), t). \tag{9.2}$$

For a sequence of gray-level images of the same 3D scene, we may assume that the **image intensity function** $I(x, t)$ is invariant under the flow of the vector field $\phi(x, t)$. Therefore, we have:

$$I(\Phi_x(t), t) = I(x, t_0). \tag{9.3}$$

We choose the noise model to be:

$$I(\Phi_x(t) + N_1(x, t), t) = I(x, t_0) + N_2(x, t) \tag{9.4}$$

where the random process $N_1 \in \mathbb{R}^2$ models the **spatial noise** – from quantization error of the location of the image points, and $N_2 \in \mathbb{R}$ models the **temporal noise** – from quantization error or variation of the image intensity function. The stochastic processes $N_1(x, t)$ and $N_2(x, t)$ are assumed to be independent Brownian motions (in time t) with initial states $N_1(x, t_0) = 0$ and $N_2(x, t_0) = 0$.

Let $\nabla I = (\frac{\partial I}{\partial x_1}, \frac{\partial I}{\partial x_2})^T \in \mathbb{R}^2$, $I_t = \frac{\partial I}{\partial t}$, $n_1 = \frac{dN_1}{dt}$ and $n_2 = \frac{dN_2}{dt}$. Differentiating equation (9.4) with respect to time t, we obtain:

$$\nabla I^T(\phi + n_1) + I_t = n_2. \tag{9.5}$$

Note that in this equation ∇I and I_t are evaluated at $(\Phi_x(t) + N_1, t)$. In order to obtain the equation in which all quantities are evaluated at (x, t_0), note that according to assumption we have $N_2(x, t_0) = 0, \nabla N_2(x, t_0) = 0$ and this gives the relations:

$$\lim_{t \to t_0} \nabla I(\Phi_x(t) + N_1, t) = \nabla I(x, t_0) \tag{9.6}$$

$$\lim_{t \to t_0} I_t(\Phi_x(t) + N_1, t) = I_t(x, t_0) \tag{9.7}$$

Then at (x, t_0), the equation (9.5) yields:

$$\nabla I^T(\phi + n_1) + I_t = n_2 \tag{9.8}$$

where all quantities are evaluated at (x, t_0). We call the random vector:

$$u = \phi + n_1 \quad \in \mathbb{R}^2 \tag{9.9}$$

the **optical flow**, and the previous equation can then be rewritten as:

$$\nabla I^T u + I_t = n_2. \tag{9.10}$$

Note that this optical flow model is similar to that used in [96] but the assumptions on noises are slightly different. Since both N_1 and N_2 are independent Brownian motions, their temporal derivatives are independent Gaussian. Without loss of generality we may assume that $n_1 \sim N(0, \sigma_1^2 I)$, $n_2 \sim N(0, \sigma_2^2)$.

Given u, the random variable $y = \nabla I^T u + I_t$ is of the distribution:

$$y \sim N(0, \sigma_2^2). \tag{9.11}$$

Then the conditional distribution $P(\nabla I, I_t \mid u)$ has the density function:

$$p(\nabla I, I_t \mid u) \;\propto\; e^{-\frac{(\nabla I^T u + I_t)^2}{2\sigma_2^2}}. \tag{9.12}$$

Comment 9.1. *Although in the rest of the paper we will only use the noise model (9.5) to illustrate how Bayesian method is carried on for motion estimation, we here discuss one possible variation of this model. Note that, in the model (9.5), we implicitly assumed that we can apply the differential operators ∇ and $\frac{\partial}{\partial t}$ to the image intensity function $I(x,t)$ precisely. However, in practice, this is questionable – numerical approximation usually introduces noises to computation. If we simply assume that numerical errors introduced by these operators do not depend on which function they apply to, we have:*

$$\nabla(\cdot) = \tilde{\nabla}(\cdot) + n_3 \tag{9.13}$$

$$\frac{\partial}{\partial t}(\cdot) = \frac{\tilde{\partial}}{\partial t}(\cdot) + n_4 \tag{9.14}$$

where $n_3 \sim N(0, \sigma_3^2)$ and $n_4 \sim N(0, \sigma_4^2)$ are Gaussian random noises independent of everything else, and $\tilde{\nabla}$ and $\frac{\tilde{\partial}}{\partial t}$ stand for ideal differential operators. Then (9.10) is modified to:

$$\nabla I^T u + I_t = n_2 + n_3^T u + n_4. \tag{9.15}$$

Consequently, the conditional density function (9.12) becomes:

$$p(\nabla I, I_t \mid u) \propto e^{-\frac{(\nabla I^T u + I_t)^2}{2(\sigma_3^2 u^T u + \sigma_2^2 + \sigma_4^2)}}. \tag{9.16}$$

Comment 9.2. *An implicit assumption we made in order to get the expression (9.12) is that, given u, the random vector $(\nabla I^T, I_t)^T \in \mathbb{R}^3$ is of uniform distribution on the plane orthogonal to the vector $b_1 = (u^T, 1)^T \in \mathbb{R}^3$. If we view uniform distribution as degenerate Gaussian, in general, we may assume that conditional distribution of $(\nabla I^T, I_t)^T$ given u is the joint Gaussian:*

$$p(\nabla I, I_t \mid u) \propto e^{-(\nabla I^T, I_t)^T \frac{1}{2}\left(\frac{b_1 b_1^T}{\sigma_2^2} + \frac{b_2 b_2^T}{\sigma^2} + \frac{b_3 b_3^T}{\sigma^2}\right)(\nabla I^T, I_t)^T} \tag{9.17}$$

where $\sigma \in \mathbb{R}$ is usually a large variance, and $b_2, b_3 \in \mathbb{R}^3$ are unit vectors and form an orthogonal basis with b_1. Note that if $\sigma = \infty$, this gives the same model as (9.12).

9.2 A Bayesian Motion Estimation Model

The question we are interested now is, given ∇I and I_t at time t, what is the optical flow estimate $u^*(x)$, and what is the estimate of camera velocity ω^*, v^* assuming that the scene is static and optical flows are generated by the motion of the camera only. From a Bayesian viewpoint, we need to derive the *a posterior* distribution:

$$p(u(x), \omega, v \mid \nabla I, I_t). \tag{9.18}$$

In this paper, we choose $u^*(x), \omega^*, v^*$ to be the maximum *a posterior* (MAP) estimate:

$$\arg\max\ p(u(x), \omega, v \mid \nabla I, I_t). \tag{9.19}$$

By the Bayesian estimation method, the *a posterior* distribution can be computed from the following relation of probability density functions:

$$p(u(x), \omega, v \mid \nabla I, I_t) \propto p(\nabla I, I_t \mid u(x)) \cdot p(u(x) \mid \omega, v) \cdot p(\omega, v) \tag{9.20}$$

where we in fact assume that $\nabla I, I_t$ are conditionally independent of ω, v given $u(x)$. The conditional distributions $p(\nabla I, I_t \mid u(x))$ and $p(u(x) \mid \omega, v)$ are also called likelihood functions, and $p(\omega, v)$ is the *a priori* distribution of ω, v.

9.3 Likelihood Functions and *a priori* Distribution

In this section, we study how to determine the likelihood functions $p(\nabla I, I_t \mid u(x))$ and $p(u(x) \mid \omega, v)$ from geometric properties of the image.

9.3.1 Local Likelihood Function of Optical Flow

Let $u_1 \in \mathbb{R}^2$ be the minimum-norm solution of the equation:

$$\nabla I^T u + I_t = 0. \tag{9.21}$$

The the density function $p(\nabla I, I_t \mid u)$ in (9.12) can be rewritten as:

$$p(\nabla I, I_t \mid u) \;\propto\; e^{-\frac{(u-u_1)^T \nabla I \nabla I^T (u-u_1)}{2\sigma_2^2}}. \tag{9.22}$$

Then the likelihood of $(u - u_1)$ has a Gaussian-like form, but the inverse of the covariance matrix is degenerate because $\nabla I \nabla I^T$ is of rank 1. It therefore does not impose any penalty on $u - u_1$ in the direction perpendicular to ∇I. In order to obtain a non-degenerate local likelihood at a image location x_0, the matrix $\nabla I \nabla I^T / 2\sigma_2^2$ is usually replaced by a local *integration* (average):

$$Q_1 \;=\; \frac{1}{2\sigma_2^2} \int_{U(x_0)} \nabla I(x) \nabla I(x)^T dx \quad \in \mathbb{R}^{2\times 2} \tag{9.23}$$

where $U(x_0)$ is a neighborhood of x_0. Then Q_1 will be non-degenerate if x_0 is on a curve or is near the intersection of several curves or straight lines. In the later case, x_0 is usually called a **corner** or **point feature**. The local likelihood $p(\nabla I, I_t \mid u)$ now becomes:

$$p(\nabla I, I_t \mid u) \;\propto\; e^{-(u-u_1)^T Q_1 (u-u_1)} \tag{9.24}$$

where u_1 is simply replaced by a local linear least square estimate (LLSE) of u for all flow equations (9.21) in the neighborhood $U(x_0)$:

$$u_1 = \left(\int_{U(x_0)} \nabla I(x) \nabla I^T(x)\, dx \right)^{-1} \int_{U(x_0)} \nabla I(x) I_t(x)\, dx. \tag{9.25}$$

It can be shown that the covariance matrix of this LLSE estimate is exactly given by Q_1^{-1} (see [32] chapter 16 of volume II). Connection between Q_1 and image local geometry is exemplified by the fact that, for a point on a single curve, the ratio of the eigenvalues of the matrix Q_1 is (approximately) proportional to square of the curvature at the point of interest.

9.3.2 Likelihood Function of Camera Motion

Let $\mathbf{x} = (x_1, x_2, 1)^T \in \mathbb{R}^3$ and consequently $\dot{\mathbf{x}} = (\phi^T, 0)^T \in \mathbb{R}^3$. It is well-known that if the optical flow ϕ is generated from a rigid body motion, it must satisfy the epipolar constraint (which has been used as a hard constraint on optical flows, see Chapter 3):

$$\dot{\mathbf{x}}^T \hat{v} \mathbf{x} + \mathbf{x}^T \hat{\omega} \hat{v} \mathbf{x} = 0. \tag{9.26}$$

Thus, for some functions $f(v) \in \mathbb{R}^2$ and $g(\omega, v) \in \mathbb{R}$, we have:

$$f(v)^T \phi + g(\omega, v) = 0. \tag{9.27}$$

Substitute $u = \phi + n_1$ into this equation and we have:

$$f(v)^T u + g(\omega, v) = f(v)^T n_1. \tag{9.28}$$

The right hand side is simply a random variable of a Gaussian distribution $N(0, \sigma_1^2 f(v)^T f(v))$. Then given ω and v, we have:

$$f(v)^T u + g(\omega, v) \quad \sim \quad N(0, \sigma_1^2 f(v)^T f(v)). \tag{9.29}$$

Let $u_2 \in \mathbb{R}^2$ be the minimum norm solution of the equation:

$$f(v)^T u + g(\omega, v) = 0 \tag{9.30}$$

That is: $u_2 = -g(\omega, v) f(v) / f(v)^T f(v)$, and let

$$Q_2 \quad = \quad \frac{f(v) f(v)^T}{2\sigma_1^2 f(v)^T f(v)} \quad \in \mathbb{R}^{2\times 2}. \tag{9.31}$$

Then the conditional density function $p(u \mid \omega, v)$ is given as:

$$p(u \mid \omega, v) \quad \propto \quad e^{-(u-u_2)^T Q_2 (u-u_2)}. \tag{9.32}$$

Comment 9.3. *Note that the expression:*

$$(u - u_2)^T Q_2 (u - u_2) \quad = \quad \frac{(f(v)^T u + g(\omega, v))^2}{2\sigma_1^2 f(v)^T f(v)} \tag{9.33}$$

in the absence of noise should be zero. This immediately gives us a probabilistically (v.s. geometrically) "canonical" normalized version of the epipolar constraint.

Comment 9.4. *Similar to the comments we gave in Comment 9.2, in order to obtain (9.32), we implicitly assumed that, given ω, v, the random vector $(u - u_2)$ is joint Gaussian and of uniform distribution on the line orthogonal to the vector $f(v)$. A more general model may be obtained by modifying the matrix Q_2 to:*

$$Q_2 = \frac{f(v)f(v)^T}{2\sigma_1^2 f(v)^T f(v)} + \frac{bb^T}{2\sigma^2} \tag{9.34}$$

where $b \in \mathbb{R}^2$ is a unit vector orthogonal to $f(v)$ and $\sigma \in \mathbb{R}$ as before is a large variance. If $\sigma = \infty$, this gives the same model as (9.32). This explains the geometric meaning of the covariance matrix. What is then the geometric meaning of the mean u_2? It is more clear from the epipolar constraint which yields the following:

$$(\dot{\mathbf{x}}^T + \mathbf{x}^T\hat{\omega})\hat{v}\mathbf{x} = 0 \tag{9.35}$$

that the optical flow in homogeneous coordinates has a mean given by $\hat{\omega}\mathbf{x}$ – velocity generated by rotation. Then u_2 is simply the 2D version of it.

9.3.3 The *a priori* Distribution of Camera Motion

The *a priori* distributions of ω and v can be assumed to be independent. $\omega \in \mathbb{R}^3$ has the Gaussian distribution:

$$\omega \sim N(0, \sigma_\omega^2 I). \tag{9.36}$$

Note that the likelihood function (9.32) takes the same value on v and λv for all $\lambda \in \mathbb{R}\backslash\{0\}$. $v \in \mathbb{R}^3$ then should a distribution on the 2D sphere S^2. We here simply use the uniform distribution. If ω or v has an initial estimate, say ω_0 or v_0, it can be assumed to be the mean of the distribution. In that case, v may be assumed to be a "Gaussian" distribution on the 2D sphere S^2 (induced from the stereo-graphic projection).

9.4 Sufficient Statistics for Rigid Body Motion Estimation

In order to compute the MAP estimate:

$$\arg\max_{u,\omega,v} p(u, \omega, v \mid \nabla I, I_t), \tag{9.37}$$

we first compute the optical flow estimate u^* at each location x of interest as a function of ω and v. Note that this computation only involves the likelihoods $p(\nabla I, I_t \mid u)$ and $p(u \mid \omega, v)$:

$$u^*(x, \omega, v) = \arg\max_u \; p(\nabla I, I_t \mid u) \cdot p(u \mid \omega, v). \tag{9.38}$$

This is equivalent to minimize the function:

$$V_1(u) \;\; = \;\; (u - u_1)^T Q_1 (u - u_1) + (u - u_2)^T Q_2 (u - u_2). \tag{9.39}$$

It yields:

$$u^*(x, \omega, v) = (Q_1 + Q_2)^{-1} (Q_1 u_1 + Q_2 u_2) \tag{9.40}$$

Now notice that we are in fact estimating ω, v from a field of measurements $\nabla I(x), I_t(x)$, $x \in \mathbb{R}^2$, instead of from a single point. We therefore need the *a posterior* distribution $p(u, \omega, v \mid \nabla I, I_t)$ where $u, \nabla I, I_t$ are viewed as random fields – random functions on (an open subset of) $\mathbb{R}^2$. We may assume the two Brownian process $N_1(x,t)$, $N_2(x,t)$ are spatially independent, *i.e.*, $N_i(x_1, t)$ and $N_i(x_2, t)$ are independent at different points x_1, x_2 for $i = 1, 2$. Then we have:

$$p(\nabla I, I_t, u \mid \omega, v) \;\; \propto \;\; e^{\int \ln(p(\nabla I(x), I_t(x), u(x) \mid \omega, v)) dx}. \tag{9.41}$$

Then the MAP estimates of the camera motion ω^* and v^* are given by:

$$\arg\max_{\omega, v} \; p(u^*, \omega, v \mid \nabla I, I_t). \tag{9.42}$$

Substituting the estimate u^* into the Bayesian formula (9.20) and use (9.41) for $p(\nabla I, I_t, u \mid \omega, v)$, we get:

$$p(u^*, \omega, v \mid \nabla I, I_t) \;\; \propto \;\; e^{-\int (u_1 - u_2)^T Q_1 (Q_1 + Q_2)^{-1} Q_2 (u_1 - u_2) dx - \frac{\omega^T \omega}{2\sigma_\omega^2}}. \tag{9.43}$$

Let matrix $W \in \mathbb{R}^{2\times 2}$ be $Q_1(Q_1 + Q_2)^{-1} Q_2$. Note that W is in fact a (non-negative definite) symmetric matrix and its entries are functions of v only. The MAP estimates of ω and v are therefore given by global minima of the objective function:

$$V_2(\omega, v) \;\; = \;\; \int (u_1 - u_2)^T W (u_1 - u_2) \; dx + \frac{\omega^T \omega}{2\sigma_\omega^2}. \tag{9.44}$$

Note that u_2 is a linear function in ω since $g(\omega, v)$ is. Hence, the objective function $V_2(\omega, v)$ is quadratic in ω. One can first solve ω^* as an explicit function of v and then convert the optimization problem to one for v on the 2D sphere S^2 only.

From the definition of the matrix W, note that if Q_1 is a singular matrix, it can be shown that W is exactly *zero*! This means that at points near a straight line, because of the aperture problem, the gradient measurements $\nabla I, I_t$ will have absolutely no contribution to the MAP estimate (or even the MMSE estimate) – one of the reasons why we favor using corners or line intersections in motion estimation. From a statistical viewpoint, we conclude:

Theorem 9.5. *For the given image noise model, gradient measurements $\nabla I, I_t$ at locations where Q_1 is non-singular are sufficient statistics for estimating motion ω and v.*

Comment 9.6. *In the expression of $V_2(\omega, v)$, the term $(u_2 - u_1)^T W (u_2 - u_1)$ gives a probabilistically "canonical" distance between the LLSE estimate u_2 from the flow equation (9.10) and the estimate u_1 from the epipolar constraint (9.26). The MAP estimate intends to minimize this distance.*

9.5 Discussion

We here investigate representative cases when some of the assumptions of the motion estimation model proposed above are violated and therefore the proposed optimization scheme no longer provides valid estimates. We also discuss possible ways to resolve such problems.

1. **Imaginary corners and intersections.** The motion estimation model proposed above explicitly relies on the assumption that the intensity of a image point changes is due to the (rigid body) motion of the corresponding 3D point. If there is no one-to-one correspondence between a image point and a 3D point, the model is violated. For example, as illustrated in Figure 9.1, intersections of those lines are "imaginary" – the image of such a intersection in fact corresponds to (at least) two spatial points. Figure 9.1 is an conceptual example. In real images, such imaginary corners or intersections usually occur along contours of solid objects, as the so called "T"-junctions. One way to resolve such a problem is to build an estimation model based directly on "motion" of (parameterized) lines instead of that of points. However, such a scheme still does not apply to cases when imaginary intersections are caused by curves instead of straight lines, for example Figure 9.2 shows an image of a trefoil curve in $\mathbb{R}^3$ – the imaginary intersections have to be discarded.

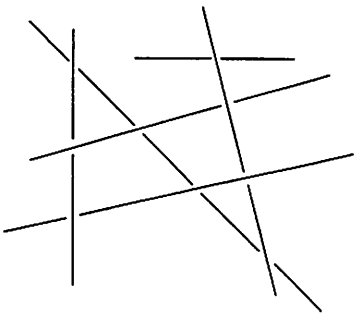


Figure 9.1: Imaginary intersections.

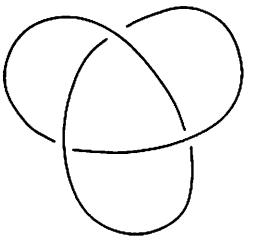


Figure 9.2: Imaginary intersections of curves.

2. **Multiple rigid body motion.** The proposed motion explicitly assumes that there is only a single rigid body motion of the whole scene. If there are multiple rigid body moving, as shown in Figure 9.3, the proposed estimation scheme no longer applies. However, this problem can be resolved using the **Expectation-Maximization** (EM)

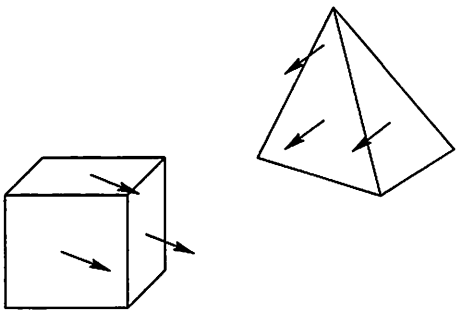


Figure 9.3: Multibody motion.

scheme with little change of previously given likelihoods, for example see [128]. In such a case, motion estimation and segmentation are solved together in a single Bayesian estimation framework. However, such segmentation will be based on different rigid

body motions, not simply on multiple smooth motion layers (as in [128]) which may encounter difficulties in segmenting a rigid object such as the trefoil (see Figure 9.2).

3. **Non-rigid body motion.** Roughly speaking, non-rigid body motion can be viewed as an extreme case of multiple rigid body motion – there are infinitely many small rigid bodies linked together. Figure 9.4 gives an example of non-rigid body motion. In such a case, the proposed motion estimation scheme will fail since it relies on the assumption that optical flows are generated by a rigid body motion such that the epipolar constraint (9.26) can be used to determine the motion likelihood function. In a case that an object indeed exhibits non-rigidity property, the proposed motion estimation scheme has to be fundamentally changed since the motion space is no longer the pair ω, v. For computational convenience, an efficient parameterization scheme is usually needed (and used) for a particular non-rigid body motion. The space of a non-rigid motion is not necessarily always infinitely dimensional – as the example shown in Figure 9.4, the motion can be simply parameterized by the principle radius of the ellipse. However, the likelihood $p(u \mid \theta \in \Theta)$ between the new motion parameter space, say Θ, and the optical flow field u has to be carefully re-determined.

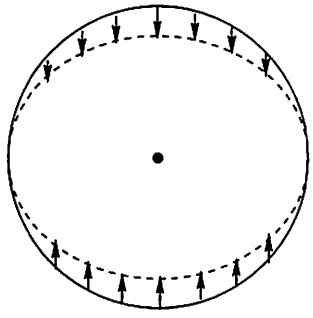


Figure 9.4: Non-rigid body motion.

4. **Non-Lambertian surfaces.** Clearly, our model is primarily based on the equation (9.3) which assumes that the intensity (or colors) of a point does not change even if the viewing angle varies, *i.e.,* the surfaces of objects have to be **Lambertian**. However, for metallic or plastic surfaces, this is usually not the true: they do not only have Lambertian reflection but also have **specular reflection** which gives these surfaces local shiny effects. In such a case, the estimates given by the given algorithm may be erroneous. Fortunately, since places where specular occurs usually have a much higher

local intensity (than the average of the image), we can simply exclude measurements at these places from the algorithm.

Besides the cases discussed above, any change of assumptions on the noises, such as the Markovness, Gaussianness, temporal or spatial dependencies, will also change the difficulty of analysis, resulting objective functions and eventually the estimates. Since these changes are more technical than conceptual, we do not discuss them in detail here.

Part III

Applications: Vision Based Robotic Control

Chapter 10

Vision Guided Navigation of an Unmanned Ground Vehicle (UGV)

Sensing of the environment and subsequent control are important features of the navigation of an autonomous mobile agent. In spite of the fact that there has been an increased interest in the use of visual servoing in the control loop, sensing and control problems have usually been studied separately. The literature in computer vision has mainly concentrated on the process of estimating necessary information about the state of the agent in the environment and the structure of the environment, e.g., [30, 40, 99, 111]. Control issues are often addressed separately. On the other hand, control approaches typically assume the full specification of the environment and task as well as the availability of the state estimate of the agent.

The dynamic vision approach proposed by Dickmanns, Mysliwetz and Graefe [16, 17, 18] makes the connection between the estimation and control tighter by setting up a dynamic model of the evolution of the curvature of the road in a driving application. Curvature estimates are used only for the estimation of the state of the vehicle with respect to the road frame in which the control objective is formulated or for the feed-forward component of the control law. Control for steering along a curved road directly using the measurement of the projection of the road tangent and its optical flow has been previously considered by Raviv and Herman [91]. However, stability and robustness issues have not yet been addressed, and no statements have been made as to what extent these cues are sufficient for general road scenarios. A visual servoing framework proposed in [20, 92] by

Espiau, Rives and Samson *et al* addresses control issues directly in the image plane and outlines the dynamics of certain simple geometric primitives. Further extensions of this approach for nonholonomic mobile platforms has been made by Pissard-Gibollet and Rives [87]. Generalization of the curve tracking and estimation problem outlined in Dickmanns to arbitrarily shaped curves addressing both the estimation of the shape parameters as well as control has been explored in [29] by Frezza and Picci. They used an approximation of an arbitrary curve by a spline, and proposed a scheme for recursive estimation of the shape parameters of the curve, and designed control laws for tracking the curve.

For a theoretical treatment of the problem, a general understanding of the dynamics of the image of an arbitrary ground curve is crucial. Therefore, before we specify particular control objectives (such as point-stabilization or trajectory tracking), we first study general properties of dynamic systems associated with image curves. In a talk given at Berkeley in October 1996, Soatto [100] formulated the problem of tracking as that of controlling the shape of the ground curve in the image plane. In spite of the fact that the system characterizing the image curve is in general infinite-dimensional, we show that for linear curvature curves the system is finite dimensional. When the control problem is formulated as one of controlling the image curve dynamics, we prove that the controllability distribution has dimension 3 and show that the system characterizing the image curve dynamics is fully controllable only up to the linear curvature term regardless of the kinematics of the mobile robot base. The controllability results indicate that the parameters characterizing the images of linear curvature curves (to be defined in Section 10.1.2) can be controlled using the driving and steering inputs. We show that the dynamics of the images of linear curvature curves can be transformed to a canonical chained-form, which already has existing point-to-point steering control scheme in Murray and Sastry [84, 85].

We then formulate the task of tracking ground curves as a problem of controlling the image curves in the image plane. We design stabilizing feedback control laws for tracking general piecewise analytic curves (for general treatments of stabilizing trajectory tracking control of nonlinear systems, one could refer to, e.g., [39, 123]). We also propose to approximate general curves by piecewise linear curvature curves. We present how to compute the image parameters for such approximating virtual curves so as to obtain the appropriate controls to track them. Simulation results are given for these control schemes.

We also study the observability of curve dynamics from the direct measurements of the vision sensor. Based on sensor models, an extended Kalman filter is proposed to

dynamically estimate the image quantities needed for the feedback control. We thus obtain a complete closed-loop vision-guided navigation system for non-holonomic mobile robots.

Chapter Outline

Section 10.1 introduces the dynamics of image curves, *i.e.*, how the shape of the image of a ground curve evolves in the image plane. Section 10.2 studies controllability issues for the dynamic systems derived in Section 10.1. Section 10.3 shows how to formulate specific control tasks for the mobile robot in the image plane. Corresponding control designs and their simulation results are also presented in the same section. Section 10.4 develops an extended Kalman filter to estimate on-line the image quantities needed for feedback control. Observability issues of the sensor model are also presented. Simulations for the entire closed-loop vision-guided navigation system are presented in Section 10.5.

10.1 Curve Dynamics

We derive equations of motion for the image curve under motions of a ground-based mobile robot. We begin with a unicycle model for the mobile robot and consider generalizations later.

10.1.1 Mobile Robot Kinematics

Consider the case where $g_{fm}(t) \in SE(2)$ is a one parameter curve in the Euclidean Group $SE(2)$ (parameterized by time) representing a trajectory of a unicycle: more specifically, the rigid body motion of the **mobile frame** F_m attached to the unicycle, relative to a fixed **spatial frame** F_f, as shown in the Figure 10.1.

Let $T_{fm}(t) = [x, y, z]^T \in \mathbb{R}^3$ be the position vector of the origin of frame F_m from the origin of frame F_f and the rotation angle θ is defined in the counter-clockwise sense about the y-axis, as shown in Figure 10.1. For the unicycle kinematics, $\theta(t)$ and $T_{fm}(t)$ satisfy:

$$\begin{cases} \dot{x} &= v \sin\theta \\ \dot{z} &= v \cos\theta \\ \dot{\theta} &= \omega \end{cases} \tag{10.1}$$

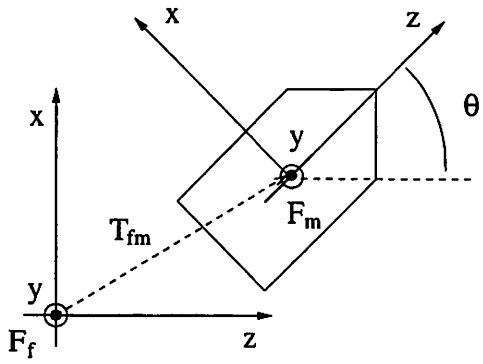


Figure 10.1: Model of the unicycle mobile robot.

where the steering input ω controls the angular velocity $\dot{\theta}$; the driving input v controls the linear velocity along the direction of the wheel.

Now, suppose a monocular camera mounted on the mobile robot which is facing downward with a tilt angle $\phi > 0$ and the camera is elevated above the ground plane by distance d, as shown in Figure 10.2. The **camera coordinate frame** F_c chosen for the camera is such that the z-axis of F_c is the optical axis of the camera, the x-axis of F_c and x-axis of F_m coincide, and the optical center of the camera coincides with the origins of both F_m and F_c.[1]

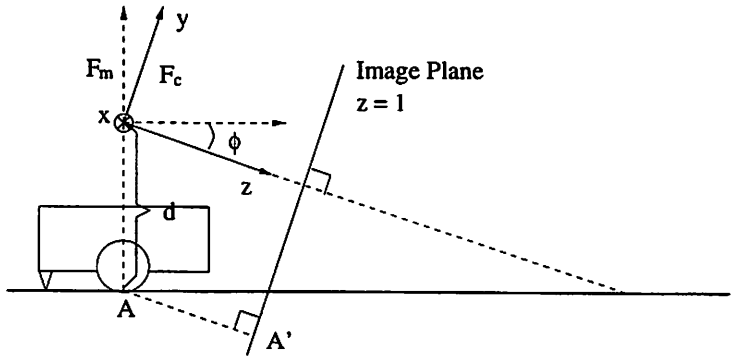


Figure 10.2: The side-view of the unicycle mobile robot with a camera facing downward with a tilt angle $\phi > 0$.

Then the kinematics of a point $p_c = [x, y, z]^T$ attached to the camera frame F_c is

[1]Without loss of generality, we assume the camera is in such a position that such a choice of coordinate frame is possible.

given in the (instantaneous) camera frame by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ \sin\phi \\ \cos\phi \end{bmatrix} v + \begin{bmatrix} y\sin\phi + z\cos\phi \\ -x\sin\phi \\ -x\cos\phi \end{bmatrix} \omega. \tag{10.2}$$

For a unit focal length camera, the image plane is $z = 1$ in the camera coordinate frame, as shown in Figure 10.2.

The use of dynamic models for the task of steering the vehicle along the roadway has been explored by Košecká *et al* [57]. In applications such as high speed highway driving the dynamic considerations play an important role. The full nonlinear dynamic model of a car has 6 degrees of freedom of motion and 4 additional degrees of freedom for tires. A simplified version of this nonlinear dynamic model which captures lateral and yaw dynamics is used for controller design. The additional parameters of the dynamic model such as load, inertia, speed and cornering stiffness may vary depending on the driving situation and/or road conditions, and affect the design of the control laws. The modeled dynamics also allows incorporation of the ride comfort criteria expressed in terms of limits on lateral acceleration into the performance specification of the system.

For steering tasks at low speed and normal driving conditions dynamic effects are not very prevalent so that the use of kinematic models may be well justified. Consequently, for simplicity of analysis, we stick to kinematic models in this paper. Extensions of our results to dynamic models is possible as well. We first establish our results for the kinematics of the unicycle model and then extend it to the bicycle model capturing the kinematics of the car.

10.1.2 Image Curve Dynamics Analysis

In this section, we consider a planar curve Γ on the ground, and study how the shape of the image of the curve Γ evolves under the motion of the mobile robot. For the rest of this paper, we make the following assumptions:

Assumption 10.1. *The ground curve Γ is an analytic curve, i.e., Γ can be locally represented by its convergent Taylor series expansion.*

Assumption 10.2. *The ground curve Γ is such that it can be parameterized by y in the camera coordinate frame F_c.*

Assumption 10.2 guarantees that the task of tracking the curve Γ can be solved using a smooth control law. For example, if the curve is orthogonal to the direction of the heading of the mobile robot, such as the curve Γ_2 shown in Figure 10.3, it can not be

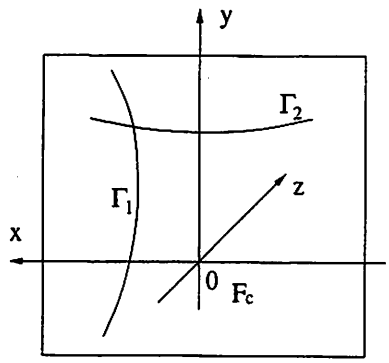


Figure 10.3: An example showing that a ground curve Γ_2 cannot be parameterized by y, while the curve Γ_1 can be.

parameterized by y. Obviously, in this case, if the mobile robot needs to track the curve Γ_2, it has to make a decision as to the direction for tracking the curve: turning right or turning left. This decision cannot be made using smooth control laws [8].

Relations between Orthographic and Perspective Projections

According to Assumption 10.2, at any time t, the curve Γ can be expressed in the camera coordinate frame as $[\gamma_1(y,t), \gamma_2(y,t), \gamma_3(y,t)]^T \in \mathbb{R}^3$. Since Γ is a planar curve on the ground, $\gamma_2(y,t)$ and $\gamma_3(y,t)$ is given by:

$$\gamma_2(y,t) = y, \quad \gamma_3(y,t) \;=\; \frac{d + y\cos\phi}{\sin\phi}. \tag{10.3}$$

which is a function of only y. Thus only $\gamma_1(y,t)$ changes with time and determines the dynamics of the ground curve. In order to determine the dynamics of the image curve we consider both **orthographic** and **perspective** projections and show that under certain conditions they are equivalent.

The orthographic projection image curve of Γ in the image plane $z = 1$ given by $[\gamma_1(y,t), y, 1]^T \in \mathbb{R}^2$ is denoted by $\tilde{\Gamma}$, as shown in Figure 10.4.

On the other hand, the perspective projection image curve, denoted by Λ =

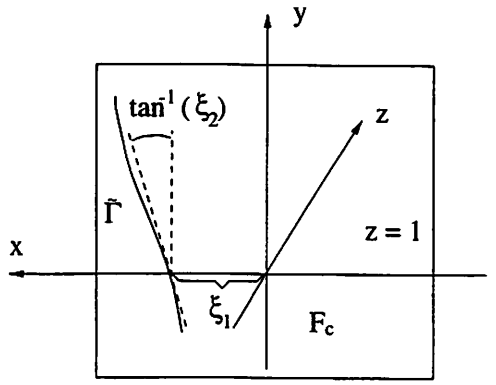


Figure 10.4: The orthographic projection of a ground curve on the $z = 1$ plane. Here $\xi_1 = \gamma_1$ and $\xi_2 = \frac{\partial \gamma_1}{\partial y}$.

$[\lambda_1(y,t), \lambda_2(y,t), 1]^T$, is given in the image plane coordinates by:

$$\begin{cases} \lambda_1(y,t) &= \frac{\gamma_1}{\gamma_3} = \frac{\gamma_1(y,t)\sin\phi}{d+y\cos\phi} \\ \lambda_2(y,t) &= \frac{\gamma_2}{\gamma_3} = \frac{y\sin\phi}{d+y\cos\phi} \end{cases} \tag{10.4}$$

Note in equation (10.4) that $\lambda_2(y,t)$ is a function of y alone and that the derivative of $\lambda_2(y,t)$ with respect to y is given by:

$$\frac{\partial \lambda_2(y,t)}{\partial y} = \frac{d\sin\phi}{(d+y\cos\phi)^2} > 0 \tag{10.5}$$

so long as $\phi > 0$ and $y \neq -d/\cos\phi$. Using the inverse function theorem, locally, the image curve Λ can be re-parameterized by $Y = \lambda_2(y,t)$ when $\frac{\partial \lambda_2(y,t)}{\partial y} \neq 0$. Λ can then be represented by $[\lambda_1(Y,t), Y]^T \in \mathbb{R}^2$ in the image plane coordinates, where the function $\lambda_1(Y,t)$ can be directly measured. However, since, as we will soon see, for the given ground curve Γ, it is easier to get an explicit expression for the dynamics of its orthographic image $\tilde{\Gamma}$ than the perspective projection image Λ. Thus, it will be helpful to find the relation between these two image curves $\tilde{\Gamma}$ and Λ, *i.e.*, the relations between the two functions γ_1 and λ_1.

First, let us simplify the notation. Define:

$$\begin{cases} \xi_{i+1} &\equiv \frac{\partial^i \gamma_1(y,t)}{\partial y^i}, \quad i = 0,1,2,\ldots \\ \zeta_{i+1} &\equiv \frac{\partial^i \lambda_1(Y,t)}{\partial Y^i}, \quad i = 0,1,2,\ldots \end{cases} \tag{10.6}$$

and:

$$\begin{cases} \xi^i &\equiv [\xi_1, \xi_2, \ldots, \xi_i]^T \in \mathbb{R}^i, \quad \xi \equiv \xi^\infty \\ \zeta^i &\equiv [\zeta_1, \zeta_2, \ldots, \zeta_i]^T \in \mathbb{R}^i, \quad \zeta \equiv \zeta^\infty \end{cases} \tag{10.7}$$

If $\gamma_1(y,t)$ is an analytic function of y, $\gamma_1(y,t)$ is completely determined by the vector ξ evaluated at any y; similarly for $\lambda_1(Y,t)$. Thus, the relations between $\tilde{\Gamma}$ and Λ are given by the relations between ξ and ζ for the case of analytic curves.

Lemma 10.3. (Equivalence of ξ, ζ Coordinates) *Consider the orthographic projection image curve $\tilde{\Gamma} = [\gamma_1(y,t), y, 1]^T$ and the perspective projection image curve $\Lambda = [\lambda_1(Y,t), Y, 1]^T$, with ξ and ζ defined in (10.6) and (10.7). Assume that the tilt angle $\phi > 0$ and $y \neq -d/\cos\phi$. Then for any fixed y,*

$$\zeta^n = A_n(y)\xi^n, \quad \forall n \in \mathbb{N} \tag{10.8}$$

where $A_n(y) \in \mathbb{R}^{n\times n}$ is a nonsingular lower triangular matrix.

Proof: We prove this lemma by using mathematical induction. For $n = 1$, from (10.4), $\zeta^1 = \frac{\sin\phi}{d+y\cos\phi}\xi^1$, so that the lemma is true for $n = 1$. Now suppose that the lemma is true for all $n \leq k$, *i.e.*,

$$\zeta^n = A_n(y)\xi^n, \quad n = 1, 2, \ldots, k \tag{10.9}$$

where all $A_n(y)$ are nonsingular lower triangular matrices. Clearly, in order to prove that for $n = k+1$ the lemma is still true, it suffices to prove that ζ_{k+1} is a linear combination of ξ^{k+1}, *i.e.*,

$$\zeta_{k+1} = \sum_{i=1}^{k+1} \beta_i(y)\xi_i. \tag{10.10}$$

Since $A_{k+1}(y)$ is nonsingular, $\beta_{k+1}(y)$ needs to be non-zero. Differentiating (10.9) with respect to y, we have:

$$\frac{\partial\zeta^k}{\partial Y}\frac{\partial Y(y,t)}{\partial y} = A_k'(y)\xi^k + A_k(y)\frac{\partial\xi^k}{\partial y} \Rightarrow \frac{\partial\zeta^k}{\partial Y} = \frac{A_k'(y)}{\frac{\partial Y(y,t)}{\partial y}}\xi^k + \frac{A_k(y)}{\frac{\partial Y(y,t)}{\partial y}}\frac{\partial\xi^k}{\partial y} \tag{10.11}$$

where the last entry of the column vector $\frac{\partial\zeta^k}{\partial Y}$ is ζ_{k+1} and:

$$\frac{\partial\xi^k}{\partial y} = [\xi_2, \xi_3, \ldots, \xi_{k+1}]^T. \tag{10.12}$$

Therefore, according to (10.11), ζ_{k+1} is a linear combination of ξ^{k+1} and, since $A_k(y)$ is a $k \times k$ nonsingular lower triangular matrix, $A_k(y)_{kk} \neq 0$,[2] the coefficient $\beta_{k+1}(y) = \frac{A_k(y)_{kk}}{\frac{\partial Y(y,t)}{\partial y}}$ is non-zero. ∎

[2] $A_k(y)_{kk}$ is the (k,k) entry of the matrix $A_k(y)$.

Example 10.4. *We calculate the matrix* $A_4(y) \in \mathbb{R}^{4\times 4}$ *to be:*

$$\zeta^4 = \begin{bmatrix} \frac{\sin\phi}{d+y\cos\phi} & 0 & 0 & 0 \\ -\frac{\cos\phi}{d} & \frac{d+y\cos\phi}{d} & 0 & 0 \\ 0 & 0 & \frac{(d+y\cos\phi)^3}{d^2\sin\phi} & 0 \\ 0 & 0 & 3\frac{(d+y\cos\phi)^4\cos\phi}{d^3\sin^2\phi} & \frac{(d+y\cos\phi)^5}{d^3\sin^2\phi} \end{bmatrix} \xi^4. \tag{10.13}$$

Lemma 10.3 tells us that under certain conditions, the dynamics of the system ξ for the orthographic projection image curve and that of ζ for the perspective projection image curve are algebraically equivalent. We may obtain either one of them from the other. ζ are quantities that we can directly measure from the perspective projection image Λ. Our ultimate goal is to design feedback control laws exclusively using these image quantities. However, as we will soon see, it is much easier to analyze the curve's dynamics in terms of ξ, the quantities in the orthographic projection image. It also turns out to be easier to design feedback control laws in terms of ξ. For these reasons, in the following sections, we choose system ξ (*i.e.,* the orthographic projection image) for studying our problem and design control laws since it simplifies the notation.

Dynamics of General Analytic Curves

While the mobile robot moves, a point attached to the spatial frame F_f moves in the opposite direction relative to the camera frame F_c. Thus, from (10.2), for points on the ground curve $\Gamma = [\gamma_1(y,t), y, \gamma_3(y)]^T$, we have:

$$\dot{\gamma}_1(y,t) = -(y\sin\phi + \gamma_3\cos\phi)\omega. \tag{10.14}$$

Also, by chain rule:

$$\dot{\gamma}_1(y,t) = \frac{\partial\gamma_1}{\partial t} + \frac{\partial\gamma_1}{\partial y}\dot{y} = \frac{\partial\gamma_1}{\partial t} + \frac{\partial\gamma_1}{\partial y}(-(v\sin\phi - \gamma_1\omega\sin\phi)). \tag{10.15}$$

The shape of the orthographic projection of the ground curve $\tilde{\Gamma} = [\gamma_1(y,t), y, 1]^T$ then evolves in the image plane $z = 1$ according to the following *Riccati-type* partial differential equation [3]:

$$\frac{\partial\gamma_1}{\partial t} = -(y\sin\phi + \gamma_3\cos\phi)\omega + \frac{\partial\gamma_1}{\partial y}(v\sin\phi - \gamma_1\omega\sin\phi). \tag{10.16}$$

[3]This equation is called a Riccati-type PDE since it generalizes the classical well-known Riccati equation for the motion of a homogeneous straight line under rotation around the origin [29, 30].

Using the notation ξ from (10.6) and the expression (10.3) for γ_3, this partial differential equation can be transformed to an infinite-dimensional dynamic system ξ through differentiating equation (10.16) with respect to y repeatedly:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \vdots \\ \dot{\xi}_i \\ \vdots \end{bmatrix} = - \begin{bmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_2, \ldots, \xi_i) \\ \vdots \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{bmatrix} v \qquad (10.17)$$

where $g_i(\xi_2, \ldots, \xi_i)$ are appropriate functions (polynomials) of only $\xi_2, \ldots, \xi_i$. In the general case, the system (10.17) is an infinite-dimensional system.

Comment 10.5. *It may be argued that the projective or orthographic projections induce a diffeomorphism (so-called homography, in the vision literature (see for example Weber et al [126])) between the ground plane and the image plane. Thus, we could write an equation of the form (10.17) for the dynamics of the mobile robot following a curve in the coordinate frame of the ground plane. These could be equivalent to the curve dynamics (10.17) described in the image plane through the push forward of the homography. We have not taken this point of view for reasons that we explain in Section 10.2.*

Dynamics of Linear Curvature Curves

In this section, we consider a special case: the ground planar curve Γ is a **linear curvature curve** (defined below). Its image dynamics in ξ can then be reduced to a three-dimensional system, which will be shown to be controllable in the following sections.

Definition 10.6. *We say that a planar curve has* **linear curvature** *if the derivative of its curvature $k(s)$ with respect to its arc-length parameter s is a non-zero constant, i.e., $k'(s) \equiv c \neq 0$. These curves are also referred to as clothoids. If $k'(s) \equiv 0$, the curve is a* **constant curvature** *curve.*

Note that, according to this definition, both straight lines and circles are constant curvature curves, but not linear curvature curves. Constant curvature curves may be regarded as degenerate cases of linear curvature curves. For linear curvature curves, we have

Lemma 10.7. *For a ground curve Γ of linear curvature, i.e., $k'(s) \equiv c \neq 0$, for any $i \geq 4$, ξ_i can be expressed as a function of ξ_1, ξ_2, and ξ_3 alone.*

Proof: Consider the ground curve $\Gamma = [\gamma_1(y,t), y, \gamma_3(y,t)]^T$ where $\gamma_3(y,t)$ is given in (10.3). For the arc-length parameter s and the curvature k, the following relationships hold:

$$s'(y) = \sqrt{\left(\frac{\partial \gamma_1}{\partial y}\right)^2 + 1 + \left(\frac{\partial \gamma_3}{\partial y}\right)^2} \tag{10.18}$$

$$k(y) = \frac{\|\Gamma'(y) \times \Gamma''(y)\|_2}{s'(y)^3} = \frac{a\frac{\partial^2 \gamma_1}{\partial y^2}}{\left(\sqrt{a^2 + (\frac{\partial \gamma_1}{\partial y})^2}\right)^3} \tag{10.19}$$

where a is defined as $a \equiv \sqrt{1 + \cot^2 \phi} = (\sin \phi)^{-1}$. Thus the derivative of the curvature k with respect to the arc-length parameter s is given by:

$$k'(s) = \frac{k'(y)}{s'(y)} = a\frac{\frac{\partial^3 \gamma_1}{\partial y^3}(a^2 + (\frac{\partial \gamma_1}{\partial y})^2) - 3\frac{\partial \gamma_1}{\partial y}\left(\frac{\partial^2 \gamma_1}{\partial y^2}\right)^2}{\left(a^2 + (\frac{\partial \gamma_1}{\partial y})^2\right)^3} \equiv c. \tag{10.20}$$

Using the definition of ξ_i, from (10.20) ξ_4 can be expressed by:

$$\xi_4 = \frac{c(a^2 + \xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2 + \xi_2^2}. \tag{10.21}$$

Therefore, ξ_4 is a function of ξ_1, ξ_2, and ξ_3 alone. According to the definition of ξ_i, it follows that, for all $i > 4$, ξ_i are functions of ξ_1, ξ_2,and ξ_3 alone. ∎

Using Lemma 10.7, for a ground linear curvature curve Γ, the dynamics of its orthographic projection image $\tilde{\Gamma}$, *i.e.*, system (10.17) for ξ, can then be simplified to be the following three-dimensional system $\xi^3 = [\xi_1, \xi_2, \xi_3]^T$:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix} = -\begin{bmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \end{bmatrix} v \tag{10.22}$$

where ξ_4 is given by (10.21).

Combining Lemma 10.3 and Lemma 10.7, we have the following remark

Remark 10.8. *For a ground curve of linear curvature, the dynamics of ζ for the perspective projection image of the curve are completely determined by three independent states $\zeta_1, \zeta_2, \zeta_3$, or equivalently, for $i \geq 4$, ζ_i is a function of only ζ_1, ζ_2, and ζ_3. The two systems $\zeta^3 = [\zeta_1, \zeta_2, \zeta_3]^T$ and $\xi^3 = [\xi_1, \xi_2, \xi_3]^T$ are equivalent and related by equation (10.13). This implies, for instance, that these two systems have the same controllability properties.*

Comment 10.9. *In the case that Γ is a constant curvature curve, i.e., $k'(s) \equiv 0$, one can show that ξ_3 is actually a function of only ξ_1, ξ_2, so for all $\xi_i, i > 3$ are functions of only ξ_1, ξ_2. There are then only two independent states ξ_1, ξ_2 for the dynamics of system ξ.*

Linear curvature is an *intrinsic* property (which is preserved under Euclidean motions *i.e.*, $SE(2)$) of planar curves. Thus, the expression (10.21) always holds under all planar motions of the robot. However, some other seemingly natural and simple assumptions that the literature has taken for the ground curve (so as to simplify the problem) might fail to be preserved under the robot's motions. For example, if, in order to simplify (10.17), one assumes $\xi_i = 0$ for $i \geq 4$, *i.e.*, $\gamma_1(y,t)$ is of the form:

$$\gamma_1(y,t) = \xi_1(y_0,t) + \xi_2(y_0,t)(y - y_0) + \frac{1}{2}\xi_3(y_0,t)(y - y_0)^2 \tag{10.23}$$

This property is not preserved under rotations. More generally, it is not an intrinsic property for a planar curve that its Taylor series expansion has a finite number of terms. Therefore, one cannot simplify system (10.17) to a finite-dimensional system by assuming that the curve's Taylor series expansion is finite (which might be the case only at special positions).[4]

10.2 Controllability Issues

We are interested in being able to control the shape of the image curves. From the above discussion, this problem is equivalent to the problem of controlling system ξ (10.17) in the unicycle case. For linear curvature curves, the infinite-dimensional system ξ is reduced to the three-dimensional system ξ^3 (10.22). In this section, we study the controllability of such systems. If the systems characterizing the curve Γ are controllable, that essentially means that given our control inputs we can steer the mobile base in order to achieve desired position and shape of the curve in the image plane. Controllability of system (10.22) is

[4]Essentially, it only "simplifies" the initial conditions of the system (10.17), not the system dimension.

directly checked in Section 10.2.1. Controllability of system (10.17) can be obtained through studying the controllability for a general ground-based mobile robots (for details on this subject, see [70]).

Note that ξ and ζ are still functions of y (or Y). They need to be evaluated at a fixed y (or Y). Since the ground curve Γ is analytic, it does not matter at which specific y they are evaluated (as long as the relation between ξ and ζ is well-defined according to Lemma 10.3)[5]. However, evaluating ξ or ζ at some special y might simplify the formulation of some control tasks.

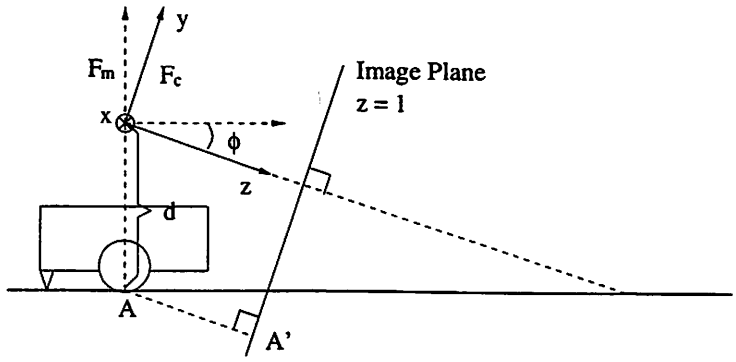


Figure 10.5: A' is the orthographic projection image of the point A where the wheel touches the ground.

For example, suppose a mobile robot is to track the given ground curve Γ. According to Figure 10.5, let A' be the orthographic projection image of the point A where the wheel of the mobile robot touches the ground. Obviously, the coordinates of A' are given by $[0, -d\cos\phi, 1]^T$. When the mobile robot is perfectly tracking the given curve Γ, *i.e.,* the wheel keeps touching the curve, the orthographic projection image $\tilde{\Gamma} = [\gamma_1(y,t), y, 1]^T$ of the curve Γ should satisfy:

$$\gamma_1(y,t)|_{y=-d\cos\phi} \equiv 0. \tag{10.24}$$

Furthermore, the tangent to the curve Γ at $y = -d\cos\phi$ should be in the same direction as the mobile robot. This requires:

$$\frac{\partial\gamma_1(y,t)}{\partial y}|_{y=-d\cos\phi} \equiv 0. \tag{10.25}$$

[5]For analytic curves, there is a one-to-one correspondence between the two sets of coefficients of the Taylor series expanded at two different points.

Thus, if ξ is evaluated at $y = -d\cos\phi$, the task of tracking Γ becomes a control problem of steering both ξ_1 and ξ_2 to 0 for the system (10.17). *For these reasons, from now on, we always evaluate ξ (or ζ) at $y = -d\cos\phi$ unless explicitly stated.*

10.2.1 Controllability in the Linear Curvature Curve Case

If the given ground curve Γ is a linear curvature curve, the dynamics of its image is given by (10.22).

Theorem 10.10 (Dimension of Controllability Lie Algebra). *Consider the system of (10.22):*

$$\dot{\xi}^3 = f_1\omega + f_2 v \tag{10.26}$$

where the vector fields (f_1, f_2) are:

$$f_1 = -\begin{bmatrix} \xi_1\xi_2\sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3\sin\phi + \xi_2^2\sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4\sin\phi + 3\xi_2\xi_3\sin\phi \end{bmatrix}, \quad f_2 = \begin{bmatrix} \xi_2\sin\phi \\ \xi_3\sin\phi \\ \xi_4\sin\phi \end{bmatrix} \tag{10.27}$$

and $\xi_4 = \frac{c(a^2+\xi_2^2)^3/a+3\xi_2\xi_3^2}{a^2+\xi_2^2}$. If $\phi \neq 0$, and $y = -d\cos\phi$, then the distribution $\Delta_{\mathcal{L}}$ spanned by the Lie algebra $\mathcal{L}(f_1, f_2)$ generated by (f_1, f_2) is of rank 3 when $c \neq 0$, and is of rank 2 when $c = 0$.

Proof: Directly calculate the Lie bracket $[f_1, f_2]$:

$$[f_1, f_2] = [-1, 0, 0]^T. \tag{10.28}$$

The determinant of matrix $(f_1, f_2, [f_1, f_2])$ is:

$$\det(f_1, f_2, [f_1, f_2]) = -c(a^2 + \xi_2^2)^3/a^3. \tag{10.29}$$

Therefore, the distribution $\Delta_{\mathcal{L}}$ spanned by $\mathcal{L}(f_1, f_2)$ is of rank 3 if $c \neq 0$, and of rank 2 if $c = 0$. ∎

Comment 10.11. *Since $\Delta_{\mathcal{L}}$ is of full rank at all points, it is involutive as a distribution. Chow's Theorem [84] states that the reachable space of system (10.22) for ξ^3 is of dimension 3 when $c \neq 0$, and 2 when $c = 0$. This makes sense since, when $c = 0$, i.e., the case of constant curvature curves, there are only two independent parameters, ξ_1 and ξ_2, needed to describe the image curves, the reachable space of such system can be at most dimension 2.*

10.2.2 Front Wheel Drive Car

In this section, we show how to extend the study of unicycle model to the kinematic model of a front wheel drive car as shown in Figure 10.6.

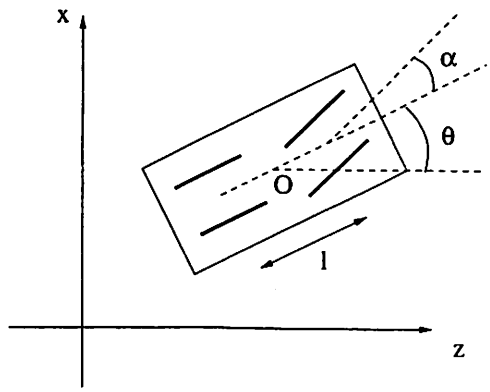


Figure 10.6: Front wheel drive car with a steering angle α and a camera mounted above the center O.

The kinematics of the front wheel drive car (relative to the spatial frame) is given by:

$$\begin{cases} \dot{x} &= \sin\theta u_1 \\ \dot{z} &= \cos\theta u_1 \\ \dot{\theta} &= l^{-1}\tan\alpha u_1 \\ \dot{\alpha} &= u_2 \end{cases} \tag{10.30}$$

where u_1 is the forward velocity of the rear wheels of the car and u_2 is the velocity of the steering rate angle.

Comment 10.12. *The dynamic model of the front wheel drive car, the so called "bicycle model" [57] has the same inputs and the same kinematics as this kinematic model of the car. In the dynamic setting the lateral and longitudinal dynamics are typically decoupled in order to obtain two simpler models. The lateral dynamics model used for the design of the steering control laws captures the system dynamics in terms of lateral velocity (or alternatively slip angle) and yaw rate. The control laws derived using this kinematic model are applicable to the highway driving scenarios providing that the 3D effects of the road curvature are negligible and the variations in the pitch angle can be compensated for. Under normal operating conditions and lower speeds the dynamical effects are not so dominant.*

Comparing (10.30) to the kinematics of the unicycle, we have:

$$\omega = l^{-1}\tan\alpha u_1, \quad v = u_1. \tag{10.31}$$

If we rewrite the system (10.17) as:

$$\dot{\xi} = f_1\omega + f_2 v \tag{10.32}$$

the dynamics of the image of a ground curve under the motion of the front wheel drive car is given by:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 \\ l^{-1}\tan\alpha f_1 + f_2 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_2 = \tilde{f}_1 u_1 + \tilde{f}_2 u_2. \tag{10.33}$$

Calculating the controllability Lie algebra for this system, we get:

$$\tilde{f}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \tilde{f}_1 = \begin{bmatrix} 0 \\ l^{-1}\tan\alpha f_1 + f_2 \end{bmatrix},$$

$$\left[\tilde{f}_1, \tilde{f}_2\right] = \begin{bmatrix} 0 \\ -l^{-1}\sec^2\alpha f_1 \end{bmatrix}, \qquad \left[\tilde{f}_1, [\tilde{f}_1, \tilde{f}_2]\right] = \begin{bmatrix} 0 \\ l^{-1}\sec^2\alpha [f_1, f_2] \end{bmatrix}.$$

Clearly, as long as $\sec^2\alpha \neq 0$, *i.e.*, α is away from $\pm\pi/2$, we have:

$$rank(\tilde{f}_1, [\tilde{f}_2, \tilde{f}_1], [\tilde{f}_1, [\tilde{f}_2, \tilde{f}_1]]) = rank(f_1, f_2, [f_1, f_2]) \tag{10.34}$$

Thus, the controllability for the front wheel drive car is the same as the unicycle. As a corollary to Theorem 10.10, we have

Corollary 10.13. *For a linear curvature curve, the rank of the distribution spanned by the Lie algebra generated by the vector fields associated with the system (10.33) is exactly* 4. *For constant curvature curves, i.e., straight lines or circles, the rank is exactly* 3.

Therefore, under the motion of the front wheel drive car, the shape of a image curve is controllable only up to its linear curvature terms, as is the unicycle case. The reader may refer to [70] for a discussion on the controllability of an *arbitrary* analytic curve under the motion of an *arbitrary* ground mobile vehicle. The conclusion can be roughly summarized as in the following remark:

Remark 10.14. *The shape of the image curve is only controllable up to its linear curvature terms, i.e.,* ξ_1, ξ_2, ξ_3 *at most under the motion of any ground mobile vehicle.*

10.3 Control Design in the Image Plane

In this section, we study the design of control laws for controlling the shape of the image curve in the image plane so as to facilitate successful navigation of the ground-based mobile robot. We consider two basic control tasks: 1. Controlling the apparent shape of the curve on the image; 2. Tracking a given ground curve.

10.3.1 Controlling the Shape of Image Curve

According to the controllability results presented in the previous section, one can only control up to three parameters $[\xi_1, \xi_2, \xi_3]^T$ of the image of a given ground curve. This means the shape of the image curve can only be controlled up to the linear curvature features of a given curve. In this section, we study how to obtain control laws for controlling the image of a linear curvature curve, as well as propose how to control the image of a general curve.

Unicycle

For a unicycle mobile robot, the dynamics of the image of a linear curvature ground curve is given by system (10.22). According to Theorem 10.10, this two-input three-dimensional system is controllable (*i.e.*, has one degree of nonholonomy) for $c \neq 0$. Thus, using the algorithm given in Murray and Sastry [84, 85], system (10.22) can be transformed to the canonical **chained-form**.

The resulting change of coordinates is:

$$\begin{cases} x_1 &= \xi_2 \\ x_2 &= -\frac{a^3\xi_3}{c(a^2+\xi_2^2)^3} \\ x_3 &= (\xi_1 - \frac{a\xi_2\xi_3}{c(a^2+\xi_2^2)^2}) \\ \omega &= \frac{-ca(a^2+\xi_2^2)^3+3a^2\xi_2\xi_3^2}{c(a^2+\xi_2^2)^4}u_1 - \frac{\xi_3}{a}u_2 \\ v &= \frac{-ca(a^2+\xi_2^2)^3+3a^2\xi_2\xi_3(a^2+\xi_2^2+\xi_3)}{c(a^2+\xi_2^2)^4}u_1 - \frac{a^2+\xi_2^2+\xi_3}{a}u_2 \end{cases} \tag{10.35}$$

where $a = (\sin\phi)^{-1}$. Then, the transformed system has the chained-form:

$$\begin{cases} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \end{cases} \tag{10.36}$$

For the chained-form system (10.36), using **piecewise smooth sinusoidal inputs** [85], one can arbitrarily steer the system from one point to another in $\mathbb{R}^3$. More robust closed loop control schemes based on time varying feedback techniques can also be found in [120]. In principle, one can therefore control the shape of the image of a linear curvature curve.

As for controlling the image of an arbitrary ground (analytic) curve, the best we can do is to approximate this curve locally by a linear curvature curve (if $k''(s) \approx 0$) and then, the controls for controlling the image of this approximating linear curvature curve can approximately control the image of the original curve freely up to its first three parameters $[\xi_1, \xi_2, \xi_3]^T$ in a local range.

Note that when $c = 0$, *i.e.,* the curve is of constant curvature, the above transformation is not well-defined. This is because the system ξ now only has two independent states ξ_1 and ξ_2. It is much easier to steer such a two-input two-state system than the above chained-form system.

Remark 10.15. *Using Lemma 10.3, the dynamic system ζ^3 of the perspective projection image of a linear curvature curve can be also transformed to chained-form.*

Front Wheel Drive Car

In this section we show that the image curve dynamical system (10.33) for the front wheel drive car model is also convertible to chained-form. According to Tilbury [110], the necessary and sufficient conditions for a system to be convertible to the chained-form are given by the following theorem:

Proposition 10.16 (Murray [83]). *Consider a n-dimensional system with two inputs u_1, u_2:*

$$\dot{x} = g_1 u_1 + g_2 u_2, \quad x \in \mathbb{R}^n. \tag{10.37}$$

Let the distribution $\Delta = span\{g_1, g_2\}$ and define two nested sets of distributions:

$$\begin{aligned}
E_0 &= \Delta, & F_0 &= \Delta \\
E_1 &= E_0 + [E_0, E_0], & F_1 &= F_0 + [F_0, F_0] \\
E_2 &= E_1 + [E_1, E_1], & F_2 &= F_1 + [F_1, F_0] \\
&\vdots & &\vdots \\
E_{i+1} &= E_i + [E_i, E_i], & F_{i+1} &= F_i + [F_i, F_0].
\end{aligned} \tag{10.38}$$

The system is convertible to chained-form if and only if:

$$dim(E_i) = dim(F_i) = i + 2, \quad i = 0, \ldots, n-2. \tag{10.39}$$

Then we can directly check the two sets of distributions for the dynamical system (10.33) of the image curve for the front wheel drive car:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 \\ l^{-1}\tan\alpha f_1 + f_2 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_2 = \tilde{f}_1 u_1 + \tilde{f}_2 u_2. \tag{10.40}$$

$$E_0 = F_0 = span\left\{\tilde{f}_1, \tilde{f}_2\right\} \tag{10.41}$$

$$E_1 = F_1 = span\left\{\tilde{f}_1, \tilde{f}_2, [\tilde{f}_1, \tilde{f}_2]\right\} \tag{10.42}$$

Clearly, $[\tilde{f}_1, [\tilde{f}_1, \tilde{f}_2]] \in [F_1, F_0] \subset F_2$. For a linear curvature curve, (10.33) is a 4-dimensional system. According to Corollary (10.13), $dim(F_2) = dim(F_1 + [F_1, F_0]) = 4$. Since $F_2 \subset E_2$, we have $dim(E_2) = dim(F_2) = 4$. Thus, according to Theorem 10.16, the system (10.33) is convertible to chained-form. The coordinate transformation may be obtained using the method given by Tilbury , Murray and Sastry in [110].

Everything we discussed in the previous section for the unicycle also applies to the front wheel drive car model. In the rest of the paper, only the unicycle case will be studied in detail but it is easy to generalize all the results to the car model as well.

10.3.2 Tracking Ground Curves

Tracking Analytic Curves

In this section, we formulate the problem of mobile robot tracking a ground curve as a problem of controlling the shape of its image with the dynamics described by (10.17). We design a *state feedback* control law for this system such that the mobile robot (unicycle) asymptotically tracks the given curve.

First, let us study the **necessary and sufficient conditions** for perfect tracking of a given curve. As already explained at the beginning of Section 10.2, when the mobile robot is perfectly tracking the given curve:

$$\xi_1 = \gamma_1(y,t)|_{y=-d\cos\phi} \equiv 0 \tag{10.43}$$

$$\xi_2 = \frac{\partial \gamma_1(y,t)}{\partial y}|_{y=-d\cos\phi} \equiv 0. \tag{10.44}$$

From (10.22) when $\xi_1 = \xi_2 \equiv 0$, we have:

$$\dot{\xi}_2 = -\xi_3 v \sin\phi + \omega / \sin\phi \equiv 0. \tag{10.45}$$

This gives the **perfect tracking angular velocity**:

$$\omega = \xi_3 \sin^2\phi \; v. \tag{10.46}$$

It is already known that system (10.17) is a nonholonomic system. According to Brockett [8], there do not exist smooth state feedback control laws which asymptotically stabilize a *point* of a nonholonomic system. However, it is still possible that smooth control laws exist for the mobile robot to asymptotically track a given curve, *i.e.*, to stabilize the system ξ around the subset $M = \{\xi \in \mathbb{R}^\infty : \xi_1 = \xi_2 = 0\}$.

A global tracking scheme has been proposed by Hespanha and Morse [41] based on the idea of "partial" feedback linearization.

Proposition 10.17 (Hespanha and Morse). *For the system ξ (10.17), set:*

$$\begin{cases} v &= v_0 + \xi_1 \omega, \quad v_0 > 0 \\ \omega &= \frac{\sin\phi}{1+\sin^2\phi\xi_2^2}(v_0 \sin\phi\xi_3 + a\xi_1 + b\xi_2), \quad a, b > 0 \end{cases} \tag{10.47}$$

Then the partial closed loop system of ξ_1, ξ_2 is linearized and given by:

$$\begin{cases} \dot{\xi}_1 &= v_0 \sin\phi\xi_2 \\ \dot{\xi}_2 &= -a\xi_1 - b\xi_2 \end{cases} . \tag{10.48}$$

This control law guarantees the partial system that we are interested is globally exponentially stable regardless of the boundness on the curvature. Thus the closed loop mobile robot globally asymptotically tracks an arbitrarily given analytic curve.

In the above, we have assumed that the set point for the linear velocity v_0 is always nonzero. In the case that $v_0 = 0$, the control (10.47) is still stabilizing but no longer asymptotically. From the partially linearized system (10.48), ξ_1 remains constant but ξ_2 can still be steered to zero. This makes sense because, without linear velocity, one can only rotate the unicyle and line up its heading with the curve but the distance to the curve remains the same.

Although Proposition 10.17 only deals with analytic (C^ω) curves, it actually can be generalized to **C^1-smooth piecewise analytic curves**[6].

[6] "C^1-smooth" means that the tangent vector along the whole curve is continuous.

Corollary 10.18. *Consider an arbitrary C^1-smooth piecewise analytic (ground) curve. If the maximum curvature $|k|_{max}$ exists for the whole curve, the linearization feedback control law given by (10.47) guarantees that the mobile robot locally asymptotically tracks the given curve.*

Remark 10.19. *Using Lemma 10.3, the control law (10.47) can be converted to a stabilizing tracking control law for ζ of the perspective projection image.*

Tracking Arbitrary Curves

Corollary 10.18 suggests that, for tracking an arbitrary continuous (C^0-**smooth**) ground curve (not necessarily analytic), one may approximate it by a C^1-smooth piecewise analytic curve, a **virtual curve**, and then track this approximating virtual curve by using the tracking control law. However, since the virtual curve cannot be "seen" in the image, how could one get the estimates of ξ for the "image" of the virtual curve so as to get the feedback controls v and ω subsequently? It turns out that, the virtual ξ is exactly the solution of the differential equation of the closed-loop system (10.17) with v and w given by the tracking control law. The initial conditions for solving such differential equation can be obtained from when designing the virtual curve.

Now, the control becomes an open-loop scheme, and in order to track this virtual curve, one has to solve the differential equation (10.17) in advance and then get the desired controls v and ω. It is computationally expensive to approximate a given curve by an arbitrary analytic curve in which case, in principle, we have to solve the infinite-dimensional differential equation (10.17).

However, as argued in Section 10.1.2, a special class of analytic curves, the linear curvature curves, can reduce the infinite-dimensional system (10.17) to a three-dimensional system (10.22), and the three states ξ^3 of the system (10.22) also have captured all the controllable features of the system ξ, according to [70]. Therefore, it is much more computationally economical to approximate the given curve by a C^1-smooth piecewise linear curvature curve and then solve the three-dimensional differential equation (10.22) to get the appropriate controls v and ω.

Few applications do require tracking of arbitrary (analytic) curves. The target curves usually can be modeled as piecewise linear curvature curves. For instance, in the case of vehicle control, in the United States, most highways are designed to be of piecewise

constant curvature, and in Europe, as clothoids. Therefore, piecewise linear curvature curves are simple as well as good models for most tracking tasks.

Comment 10.20. *Strictly speaking, when approximating a given curve by a piecewise polynomial curve, for example by using splines [29], in order to get the estimate of ξ for the evolution of the approximating virtual curve, one has to solve the infinite-dimensional differential equation (10.17). What the "polynomial" property really simplifies is just the initial conditions of the differential equation but not the dimension of the problem, as already argued in Section 10.1.2.*

Example 10.21 (Mobile Robot Tracking Corridors). *Consider a simple example: the mobile robot is supposed to track a piecewise linear curve consisting of intersection of l_1 and l_2 (as a reasonable model for corridors inside a building), as shown in Figure 10.7. A natural and simple way to smoothly connect them together is to use a piece of arc AB which is tangential to both of the straight lines (at points A and B respectively). From point A, the mobile robot switches to track the virtual curve, arc AB until it smoothly steers into the next piece, i.e., the line l_2. The $\xi^3(t)$ for tracking this virtual arc AB is then given by the solution of the closed-loop system of (10.22) with $c = k'(s) \equiv 0$ and the initial conditions at point A: $\xi^3(0) = [0, 0, -a^2/r]^T$.*

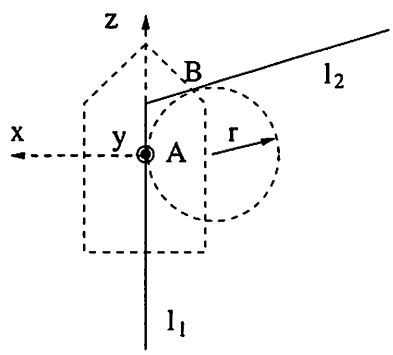


Figure 10.7: Using arcs to connect curves which are piecewise straight lines.

In the above example, since the approximating virtual curve is to be as close to the original curve as possible, the radius r of the arc AB should be as small as possible. But, in real applications, the radius r is limited by the maximal curvature that the mobile robot can track ($r = 1/|k|$). Thus, one needs to consider this extra constraint when designing the

virtual curves. The following result tells us a way to decide the maximal curvature $|k|_{max}$ that the mobile robot can track:

Remark 10.22. *Consider the unicycle mobile robot. If its linear velocity v and angular velocity ω satisfy $|v| \geq c_1$ and $\omega^2 + v^2 \leq c_2^2$, then the maximal curvature that it can track is:*

$$|k| \leq \sqrt{\left(\frac{c_2}{c_1}\right)^2 - 1}. \tag{10.49}$$

Consider now that the image curve obtained is not even continuous, *i.e.*, the robot "sees" several chunks of the image of the real curve that it is supposed to track. Basically, there are two different approaches that one might take in order to track such a curve: first, one may use some estimation schemes and based on the estimated features of the real curve to apply the feedback control law (as studied by Frezza and Picci [29]); second, one may just smoothly connect these chunks of the image curve by straight lines, arcs or linear curvature curves and then apply the virtual tracking scheme as given above to track the approximating virtual curves.

10.3.3 Simulation Results of Tracking Ground Curves

In this section, we show simulation results of the mobile robot tracking some specific ground curves using the control schemes designed in previous sections. We assume that all the image features ξ are already available. In next section, we discuss how to actually estimate ξ from the real (probably noisy) images. For all the following simulations, we choose the camera tilt angle $\phi = \pi/3$, and $v_0 = 1$. The reference coordinate frame F_f is chosen such that the initial position of the mobile robot is $z_{f0} = 0, x_{f0} = 0$ and $\theta_0 = 0$.

Tracking a Linear Curvature Curve

For the simulation results given in Figure 10.8, the nominal trajectory is chosen to be a linear curvature curve with the constant curvature varying rate $c = k'(s) \equiv -0.05$. Its initial position given in the image plane is $\xi_{10} = 0.1$, $\xi_{20} = 0.1$, and $\xi_{30} = 2$.

Tracking Piecewise Straight-Line Curves

Consider now the example discussed in Section 10.3.2: the mobile robot is to track a piecewise linear curve consisting of intersection of l_1 and l_2 as shown in the Figure 10.9. We

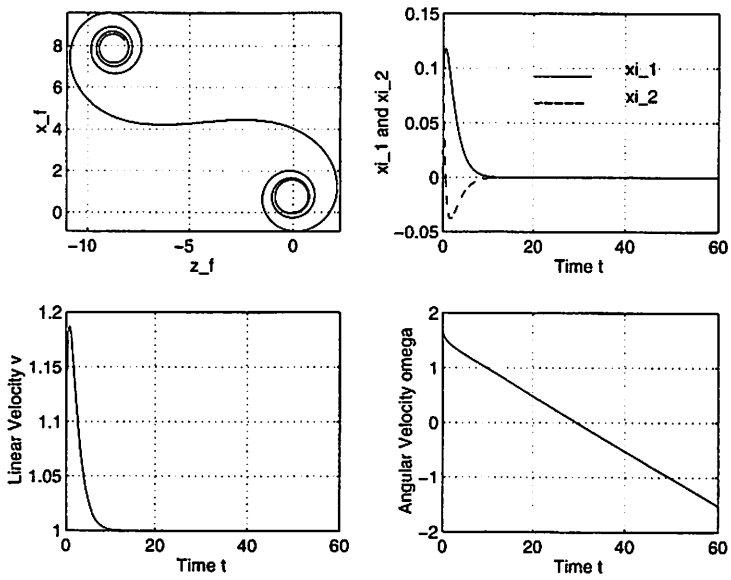


Figure 10.8: Simulation results for tracking a linear curvature curve ($c = k'(s) = -0.05$). Subplot 1: the trajectory of the mobile robot in the reference coordinate frame; subplot 2: the image curve parameters ξ_1 and ξ_2; subplot 3 and 4: the control inputs v and ω.

compare the simulation results of two schemes: 1. Using only the feedback tracking control law; 2. Using a pre-designed approximating virtual curve (an arc in this case) around the intersection point. From Figure 10.9, it is obvious that, by using the pre-designed virtual curve, the over-shoot can be avoided. But the computation is more intensive: one needs to design the virtual curve and calculate the desired control inputs for tracking it.

10.4 Observability Issues and Estimation of Image Quantities

As we have discussed in Section 10.1.2, ξ are the features of the orthographic projection image $\hat{\Gamma}$ of the ground curve Γ, and are not yet the real image (which, by convention, means the perspective projection image Λ) quantities ζ. However, ξ and ζ are algebraically related by Lemma 10.3. In principle, one can obtain ξ from the directly-measurable ζ.

In order to apply the tracking control laws given before, one need to know the values of ξ_1, ξ_2, and ξ_3, *i.e.*, ζ_1, ζ_2 and ζ_3. Suppose, at each instant t, the camera provides N measurements of the image curve Λ:

$$\left\{[\lambda_1(Y_k, t), Y_k, 1]^T\right\}, \quad k = 1, \ldots, N \tag{10.50}$$

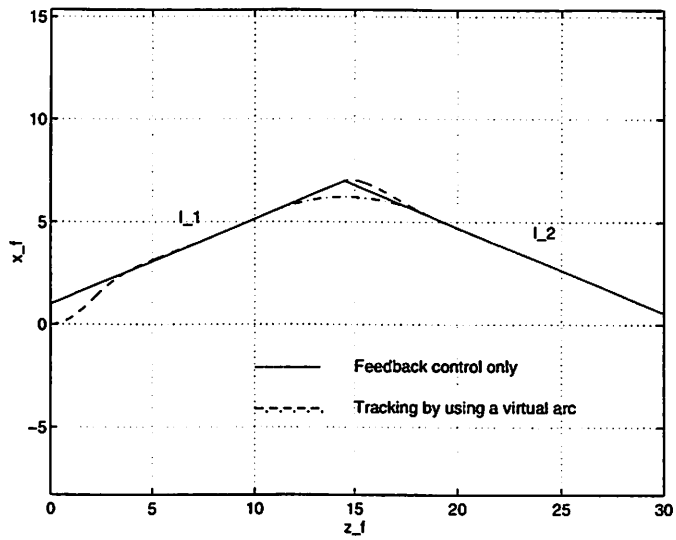


Figure 10.9: Comparison between two schemes for tracking a piecewise straight-line curve.

where $\{Y_1, Y_2, \ldots, Y_N\}$ are fixed distances from the origin. If the distances between Y_k are small enough, one can estimate the values of $\zeta_1(Y_k), \zeta_2(Y_k)$, and $\zeta_3(Y_k)$ simply by:

$$\begin{cases} \hat{\zeta}_1(Y_k) &= \lambda_1(Y_k, t) \\ \hat{\zeta}_2(Y_k) &= \frac{\lambda_1(Y_{k+1},t)-\lambda_1(Y_k,t)}{Y_{k+1}-Y_k} \\ \hat{\zeta}_3(Y_k) &= \left(\frac{\lambda_1(Y_{k+2},t)-\lambda_1(Y_{k+1},t)}{Y_{k+2}-Y_{k+1}} - \frac{\lambda_1(Y_{k+1},t)-\lambda_1(Y_k,t)}{Y_{k+1}-Y_k}\right)/(Y_{k+1}-Y_k) \end{cases} \tag{10.51}$$

for $k = 1, \ldots, N-2$. However, in practice, the measurements $\{[\lambda_1(Y_k, t), Y_k, 1]^T\}_{k=1}^N$ are noisy and the estimates (10.51) for ζ^3 become very inaccurate, especially for the higher order terms ζ_2 and ζ_3. It is thus appealing to estimate ζ^3 or ξ^3 by only using the measurements $\{[\lambda_1(Y_k, t), Y_k, 1]^T\}_{k=1}^N$ but not their differences.

10.4.1 Sensor Models and Observability Issues

General Analytic Curves

The curve dynamics are already given by (10.17). If we only use the measurement $\zeta_1 = \lambda_1(Y, t)$ as the output of the vision sensor, then we have the following sensor model:

$$\begin{aligned}
\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \vdots \\ \dot{\xi}_i \\ \vdots \end{bmatrix} &= - \begin{bmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_2, \ldots, \xi_i) \\ \vdots \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{bmatrix} v \\
h(\xi) &= \zeta_1 = \tfrac{\sin\phi}{d + y\cos\phi}\xi_1
\end{aligned} \tag{10.52}$$

where $h(\xi)$ is the measurable output.

Theorem 10.23 (Observability of the Camera System). *Consider the system given by (10.52). Let:*

$$f_1 = - \begin{bmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_2, \ldots, \xi_i) \\ \vdots \end{bmatrix}, \quad f_2 = \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{bmatrix}. \tag{10.53}$$

If $\phi \neq 0$, then the annihilator Q of the smallest codistribution Ω invariant under f_1, f_2 and which contains $dh(\xi)$ is empty.

Proof: Through direct calculations, the k-th order Lie derivative of the covector field $dh(\xi)$ along the vector field f_2 is:

$$L_{f_2}^k dh(\xi) = \frac{\sin^{k+1}\phi}{d + y\cos\phi} d\xi_{k+1}, \quad k = 0, 1, 2, \ldots, \infty. \tag{10.54}$$

Thus, Ω contains all $d\xi_i$, $i \in \mathbb{N}$ and therefore Q is an empty distribution. ∎

Comment 10.24. *According to the Theorem 1.9.8 in Isidori [48], Theorem 10.23 guarantees that the system (10.52) is observable. In other words, the (locally)* **maximal output zeroing manifold** *of the system (10.52) does not exist, according to the Proposition 10.16 in Sastry [93]. Since this system is observable, ideally, one then can estimate the $\hat{\xi}$ from the output $h(\xi)$. However, the observer construction may be difficult.*

Linear Curvature Curves

The sensor model (10.52) is an infinite-dimensional system. In order to build an applicable estimator for ξ^3 (so as to apply the tracking control laws), one has to assume some regularity on the given curve Γ so that the sensor model becomes a finite-dimensional system. In other words, one has to approximate Γ by simpler curve models which have finite-dimensional dynamics.

In Frezza and Picci [29], the models chosen are **third-order B-splines**. However, as we have pointed out in Section 10.1.2, the polynomial form is not an intrinsic property of a curve and it cannot be preserved under the motion of the mobile robot. Furthermore, simple curves like a circle cannot be expressed by third-order B-splines. We thus propose to use (piecewise) linear curvature curves as the models. The reasons for this are obvious from the discussions in previous sections: the dynamics of a linear curvature curve is a three-dimensional system (10.22); such a system has very nice control properties; and piecewise linear curvature curves are also natural models for highways. However, a most important reason for using linear curvature curves is that, according to Proposition 10.17, one actually only needs the estimation of three image quantities, *i.e.*, ξ_1, ξ_2 and ξ_3 to be able to track any analytic curve. All the "higher order terms" ξ_i, $i \geq 4$ are not necessary.

For a linear curvature curve, since we do not have a priori knowledge about the constant curvature varying rate $c = k'(s)$, we also need to estimate it. Let $\eta = c$ and we have the following sensor model for linear curvature curves:

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\eta} \end{bmatrix} &= - \begin{bmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{bmatrix} v \\ h(\xi^3, \eta) &= \zeta_1 = \frac{\sin\phi}{d + y\cos\phi}\xi_1 \end{aligned} \tag{10.55}$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2+\xi_2^2}$ and $h(\xi^3, \eta)$ is the measurable output.

Theorem 10.25 (Observability of the Simplified Sensor Model). *Consider the sys-*

tem (10.55). Let:

$$f_1 = -\begin{bmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{bmatrix}, \quad f_2 = \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{bmatrix}. \tag{10.56}$$

If $\phi \neq 0$, then the smallest codistribution Ω invariant under f_1, f_2 and which contains $dh(\xi^3, \eta)$ is of constant rank 4.

Proof: Through direct calculations, we have:

$$L_{f_2}^k dh(\xi^3, \eta) = \frac{\sin^{k+1}\phi}{d + y\cos\phi} d\xi_{k+1}, \quad k = 0, 1, 2 \tag{10.57}$$

and:

$$L_{f_2}^3 dh(\xi^3, \eta) = \frac{(a^2 + \xi_2^2)^2 \sin^5\phi}{d + y\cos\phi} d\eta. \tag{10.58}$$

Thus, Ω contains all $d\xi_1, d\xi_2, d\xi_3$, and $d\eta$ and it has constant rank 4. ∎

Therefore, the system (10.55) is observable according to the Theorem 1.9.8 in Isidori [48] or the Proposition 10.16 in Sastry [93].

10.4.2 Estimation of Image Quantities by Extended Kalman Filter

The sensor model (10.55) is a nonlinear observable system. **Extended Kalman filter** (EKF) is a widely used scheme to estimate the states of such systems. In the computer vision community, estimation schemes based on Kalman filter have been commonly used for dynamical estimation of motion [99, 101] or road curvature [17, 18], etc. Here, we use the EKF algorithm to estimate on-line the $\hat{\xi}_1, \hat{\xi}_2, \hat{\xi}_3$, and $\hat{\eta}$. Alternatives to the EKF, which are based on nonlinear filtering, are quite complicated and are rarely used.

Multiple-Measurement Sensor Model

In order to make the EKF converge faster, we need to use more than one measurement (in the sensor models (10.52) and (10.55)). From the N measurements:

$$\left\{[\lambda_1(Y_k, t), Y_k, 1]^T\right\}, \quad k = 1, \ldots, N \tag{10.59}$$

we have N outputs:

$$h_k(\xi) = \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k \cos\phi}\xi_1(y_k), \quad k = 1, \ldots, N \tag{10.60}$$

where Y_k and y_k are related by (10.4) $Y_k = \frac{y_k \sin\phi}{d+y_k\cos\phi}$.

For linear curvature curves, all the measurements $\xi_1(y_k)$ are functions of only ξ^3 and the linear curvature η since all the Taylor series expansion coefficients ξ_i, $i \in \mathbb{N}$ are functions of only ξ^3 and η according to Lemma 10.7. Let:

$$h(\xi^3, \eta, y) = \sum_{i=1}^{\infty} \frac{\xi_i}{(i-1)!}(y + d\cos\phi)^{i-1}. \tag{10.61}$$

$\xi_1(y_k)$ are then given by $\xi_1(y_k) = h(\xi^3, \eta, y_k)$.

The sensor model (10.55) can be modified as:

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\eta} \end{bmatrix} &= - \begin{bmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{bmatrix} v \\ h_k(\xi^3, \eta) &= \zeta_1(Y_k) = \frac{\sin\phi}{d+y_k\cos\phi} h(\xi^3, \eta, y_k), \quad k = 1, \ldots, N \end{aligned} \tag{10.62}$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2+\xi_2^2}$, and h_k are the measurable outputs.

Noise Model

In order to track the variations in the rate of change of the curvature of a curve, we choose:

$$\dot{\eta} = \mu_\eta \tag{10.63}$$

where μ_η is white noise of appropriate variance.[7]

The output measurements are inevitably noisy, and the actual ones are given by:

$$h_k(\xi^3, \eta) = \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k\cos\phi} h(\xi^3, \eta, y_k) + \mu_{h_k}, \quad k = 1, \ldots, N \tag{10.64}$$

where μ_{h_k} are appropriate noise models for the N outputs. Strictly speaking, μ_{h_k} are color noise processes since image quantization errors[8] are main sources for μ_{h_k} which generically

[7] One may also model η as a second order random walk.

[8] Including the errors introduced by the image-processing algorithms used to process the original images.

produce color noises. The explicit forms for the output h_k are given by the Taylor series expansion (10.61). Truncating the higher order terms of the expansion can be regarded as another color noise source for the output noises μ_{h_k}. However, in order to approximately estimate the states ξ^3 and η, we may simplify μ_{h_k} to white noise processes and then actually build an extended Kalman filter (Jazwinski [49], Mendel [80]) to get the estimates $\hat{\xi}^3$ and $\hat{\eta}$ for the states of the nonlinear stochastic model:

$$\begin{aligned}
\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\eta} \end{bmatrix} &= - \begin{bmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{bmatrix} \omega + \begin{bmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mu_\eta \qquad (10.65) \\
h_k(\xi^3, \eta) &= \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k \cos\phi} h(\xi^3, \eta, y_k) + \mu_{h_k}, \quad k = 1, \ldots, N
\end{aligned}$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2+\xi_2^2}$, and μ_η and μ_{h_k} are white noises with appropriate variances. For a detailed implementation of this extended Kalman filter, one may refer to the technical report [70].

The computational complexity of Kalman filter is $O(n^3)$ where n is the system dimension [80]. Although, in some sense, both linear curvature curves and third-order B-splines (Frezza and Picci [29]) are third-order approximations for general curves, the dimension of the Kalman filter for estimating the B-spline parameters is $N + 2$ where N is the number of measurements. However, the EKF we propose here is only 4-dimensional. Since the number of measurements N is usually larger than 4, the scheme proposed above is less computationally expensive.

Simulation Results of the Extended Kalman Filter

For illustration, we here give some simulation results of using the EKF to estimate the image quantities ξ^3 and η (*i.e.*, the states of the system (10.65)). We first show a simple example for which the EKF converges. The curve is simply chosen to be a constant curvature curve (a circle) *i.e.*, $c = k'(s) \equiv 0$. The initial values chosen for the estimates are $\hat{\xi}^3(0) = [0, 0, 0]^T$ and $\hat{\eta}(0) = 0.1$, and for the nominal states $\xi^3(0) = [0.1, 1, 1]^T$. The number of output measurements N is 5. The feedback tracking control laws now use the estimates $\hat{\xi}^3$ for v and ω. Since we use synthetic images here, we do not add noise here. The simulation results are shown in Figure 10.10.

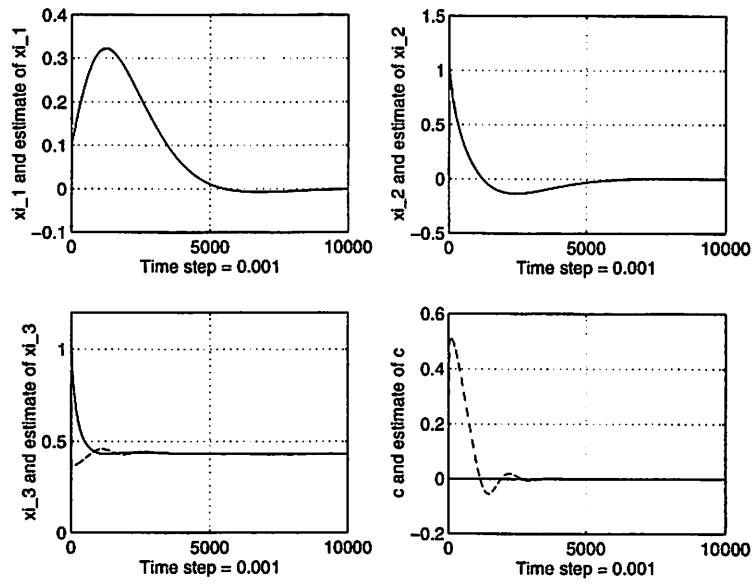


Figure 10.10: The simulation results of using the Extended Kalman Filter to estimate the image quantities ξ^3 and $\eta(=c=k'(s))$ with the number of output measurements $N=5$: Solid curves are for true states; dashed curves are for estimates.

These results show that the estimates $\hat{\xi}^3$ and $\hat{\eta}$ converge to the nominal values ξ^3 and $\eta(=c)$. $\hat{\xi}_1$ and $\hat{\xi}_2$ converge especially quickly to ξ_1 and ξ_2 and their curves almost coincide. The results also show that the mobile robot eventually tracks the circle by using the estimates $\hat{\xi}^3$ for the tracking control laws since both ξ_1 and ξ_2 eventually converge to zero.

10.5 Simulation of the Vision Based Closed-loop System

In the previous sections, we have developed control and estimation schemes for mobile robot navigation (tracking given curves) using vision sensors. The image parameters needed for the tracking control schemes can be efficiently estimated from direct, probably noisy, image measurements. Combining the control and estimation schemes together, we thus obtain a complete **closed-loop vision-guided navigation system** which is outlined in Figure 10.11.

In order to know how this system works, we simulate it by using synthetic images of the ground curve. A **synthetic image** of a ground curve $\Gamma=[\gamma_1(y,t),y,\gamma_3(y,t)]^T$ is a

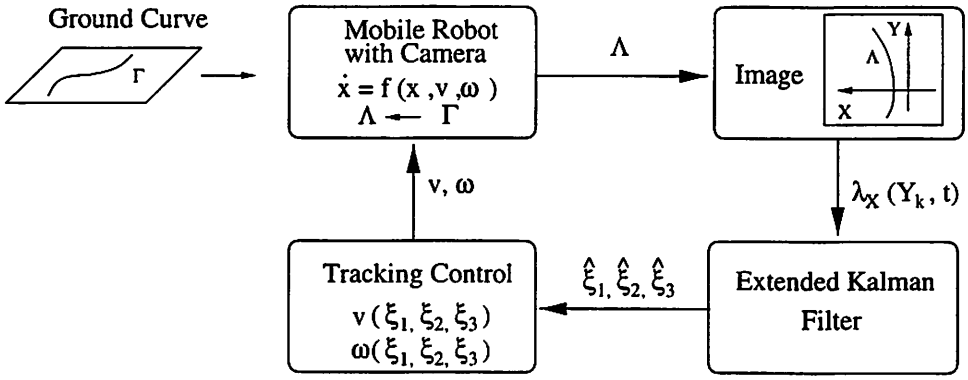


Figure 10.11: The closed-loop vision-guided navigation system for a ground-based mobile robot.

set of image points:

$$I = \{[\lambda_1(Y_i,t), Y_i, 1]^T = \pi_p \circ [\gamma_1(y_i,t), y_i, \gamma_3(y_i,t)]^T\}_{i=1}^{M} \tag{10.66}$$

where π_p denotes the perspective projection map and the number of image points M maybe different for different time t. The output measurements from this synthetic image I are taken at N pre-fixed distances: $Y_1, \ldots, Y_N$. **Linear interpolation** is used to obtain an approximate value of $\lambda_1(Y_k, t)$ if there is no point in I whose Y coordinate is Y_k.

Simulation results show that the control and the estimation schemes work well with each other in the closed-loop system. For illustration, Figure 10.12 presents the simulation results for the simple case when Γ is a circle. We have also developed animations for synthetic images and simulation data. Figure 10.13 shows a synthetic image of a circular road viewed from the camera.

10.6 Discussion

In order to use the vision sensors inside the control servo loop, one first need to study the dynamics of the image. The dynamics of certain simple geometric primitives, like points, planes and circles, have been studied and exploited by Espiau [20], Pissard-Gibollet and Rives [87] *et al.* In this paper, we show that, for ground-based mobile robot, it is possible to study the dynamics of the image of a more general class of objects: analytic curves. Based on the understanding of image curve dynamics, we design control laws for tasks like controlling the shape of a image curve or tracking a given curve. Our study indicates that the shape of the image curve is controllable only up to its linear curvature

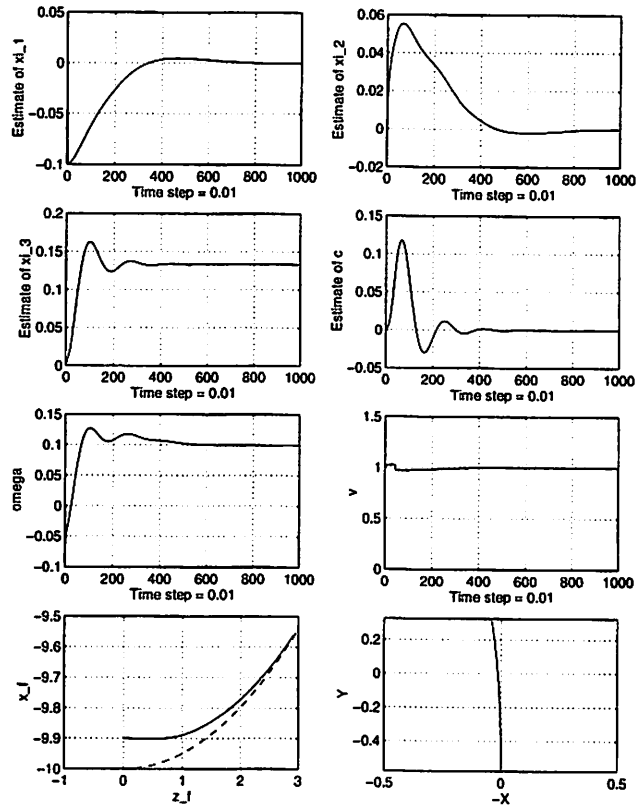


Figure 10.12: Simulation results for the closed-loop vision-guided navigation system for the case when the ground curve is a circle: In subplot 7, the solid curve is the actual mobile robot trajectory (in the space frame F_f) and the dashed one is the nominal circle; subplot 8 is the image of the circle viewed from the camera at the last simulation step, when the mobile robot is perfectly aligned with the circle.

terms (in the 2-dimensional case). However, there exist state feedback control laws (using only "up to curvature" terms) enabling the mobile robot to track arbitrary analytic curves. Such control laws are not necessarily the only ones. In applications, other control laws may be designed and used to obtain better control performances.

In the cases that one has to approximate a general curve (which has infinite-dimensional dynamics) by simpler models, it is crucial to use models with properties which are invariant under the Euclidean motion (so-called intrinsic properties). We propose that linear curvature curves are very good candidates for such models. In some sense, linear curvature curves are a third-order approximation for general curves, so are third-order B-splines used by Frezza and Picci [29]. However, the Extended Kalman Filters needed to

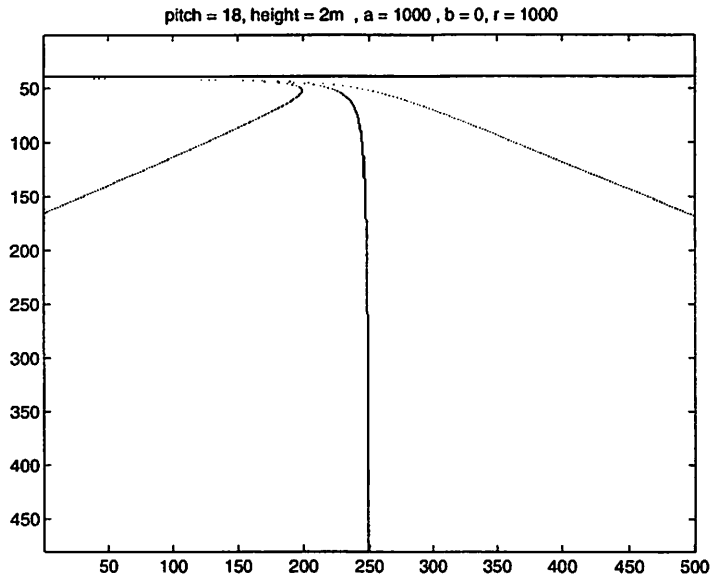


Figure 10.13: A synthetic image of a piece of circular road viewed from the camera.

estimate their parameters are 4-dimensional and $(N + 2)$-dimensional respectively (where N is the number of output measurements). The computation intensities of the two schemes therefore are different.

We are aware of the extensive literature on vision based control in driving applications. The models and the control laws that we propose are more appropriate for mobile robot applications, where typically in typically indoor environments that the ground plane assumption is satisfied, and kinematic models are appropriate. We are currently working on generalizing some of the ideas presented in this Chapter to the context of dynamic models. Some of the work in this direction can be found in Košecká [57].

Although visual servoing for ground-based mobile robot navigation has been extensively studied, its applications in aerial robot navigation have not received much attention. In the aerial robot case, the motions are 3-dimensional rigid body motions $SE(3)$ instead of $SE(2)$ for ground-based mobile robots. Generally speaking, due to the complexity of aerial robot – such as a helicopter – dynamics, the analysis based on lifting mobile kinematics (or dynamics) up to the image plane likely becomes intractable. Therefore, in this case, vision is usually used as a pure sensor for estimating the states of the robot dynamics and control analysis is done separately. Then the central issue becomes for a given task, how to design good controllers based on states which can be reliably and effectively estimated through

vision, and how to choose or customize vision algorithms for the specific task. This issue will be investigated further in the next chapter.

Chapter 11

Vision Guided Landing of an Unmanned Aerial Vehicle (UAV)

Unmanned air vehicles (UAVs) are being used more and more in a number of civilian and military applications, for example remote monitoring of traffic, search and rescue operations, and surveillance. This has generated considerable interest in the control community, mainly due to the fact that the design of UAVs brings to light research questions falling in some of the most exciting new directions for control. One of these directions is the use of computer vision as a sensor in the feedback control loop. The task of autonomous aircraft landing is particularly well suited to vision based control, especially in cases where the landing pad is in an unknown location and is moving, such as the deck of a ship.

Typically, a vision system on board a UAV augments a sensor suite including a Global Positioning System (GPS) which provides position information relative to the inertial frame, and Inertial Navigation Sensors (INS) which provide acceleration information [133]. As a cheap, passive and information-abundant sensor, computer vision is gaining more and more importance in the sensor suite of mobile robots. There has been a growing interest in control design around a vision sensor. In [94], stereo vision systems are proposed to augment a multi-sensor suite including laser range-finders in the landing maneuver of a UAV. In [137], the use of projections of parallel lines is proposed for the purpose of estimating the location and orientation of the helicopter landing pad. Using this approach, the vision sensor provides position and orientation estimates of the camera relative to the landing pad, but can not estimate the camera velocity, which is important for controlling

a UAV. In this chapter, we present computer vision algorithms to estimate UAV motion (position and orientation, linear and angular velocity) relative to a landing pad using a calibrated monocular camera. The given algorithms are linear, computationally inexpensive, numerically robust, and amenable to real-time implementation. We also present a thorough performance evaluation of the vision based motion estimation under varying levels of image measurement noise, altitudes, and camera motions relative to the landing pad.

Further more, the use of computer vision in the control of UAVs is more challenging than in the classical "visual servoing" approach discussed in the preceding chapter because UAVs are under-actuated nonlinear dynamical systems. In order for a guaranteed performance such as stability for the overall closed-loop system, a thorough characterization of the UAV dynamics are absolutely necessary. We hereby present a full dynamic model of the UAV. Based on geometric control theory, we decompose the dynamics into two subsystems: inner and outer systems. A nonlinear controller is proposed based on differential flatness of the outer system. In addition to the work in [121], we also give a detailed stability analysis of the closed-loop system, and clear conditions are derived for system stability. The proposed controller is tightly coupled with the vision based state estimation and the only auxiliary sensor needed to implement the controller is an INS for measurement of acceleration. The INS is used since second order derivatives of image features are highly sensitive to noise. Finally, we show through simulation that the designed vision-in-the-loop controller is stable even for large levels of image measurement noise. Implementation on real helicopters will be reported in future work.

Chapter Outline

In Section 11.1, we review a little the camera imaging models. In Section 11.2 we formulate the problem of motion estimation from image measurements of a planar scene. We present a new geometrical scheme for the recovery of the camera linear and angular velocities from the velocities of feature image points. In Section 11.2.4 we provide simulation results of the planar ego-motion estimation algorithms and evaluate their performance under the presence of noise, and different types of motion relative to the plane. In Section 11.3 we give the dynamic model of the UAV and the design of a controller based on differential flatness. Conditions for closed-loop stability are also studied in detail. In Section 11.4 we describe how the obtained vision algorithms can be placed in the feedback loop as a state estimator

for the controller, and provide simulation results of the vision based landing maneuver.

11.1 Camera Model

We assume that a monocular camera is fixed to the UAV and the optical axis of the camera coincides with the vertical axis of the UAV body frame. As for notation, we will adhere to the convention specified in Chapter 2: We denote the three dimensional coordinates of a point p with respect to the camera frame as $\mathbf{X} = [X_1, X_2, X_3]^T \in \mathbb{R}^3$.

The imaging of the camera is given by the **perspective projection** of points in the 3D world onto the image plane. We assume a calibrated camera, and without loss of generality we take the image plane to be at a unit distance from optical center of the camera. Then the perspective projection of the camera is then given by:

$$\begin{aligned} \pi : \mathbb{R}^3 &\rightarrow \mathbb{RP}^2 \\ \mathbf{X} &\mapsto \frac{\mathbf{X}}{X_3}. \end{aligned} \tag{11.1}$$

If $\mathbf{x}$ is the image of the point p, *i.e.*, $\mathbf{x} = \pi(\mathbf{X})$, then we can write:

$$\lambda \mathbf{x} = \mathbf{X} \tag{11.2}$$

where $\lambda = X_3 \in \mathbb{R}$ encodes the depth of p from the optical center of the camera. Denoting the optical axis by $e_3 = [0, 0, 1]^T$, we have $\lambda = e_3^T \mathbf{X}$. Rewriting equation (11.2), we get the following identity:

$$(I - \mathbf{x} e_3^T)\mathbf{X} = 0 \tag{11.3}$$

which will be useful in the later development.

11.2 Motion Estimation from Planar Scene

In this section, we first study an **ego-motion estimation problem** associated to landing a UAV. Ego-motion estimation in general settings has been extensively studied in Part I. The goal is to recover the motion of the camera using image measurements of fixed points in the scene. The ego-motion estimation problem for the purpose of landing a UAV is a special case of the general one: All the image features correspond to coplanar points on the landing pad. It is well known that the case where all features points in the scene are

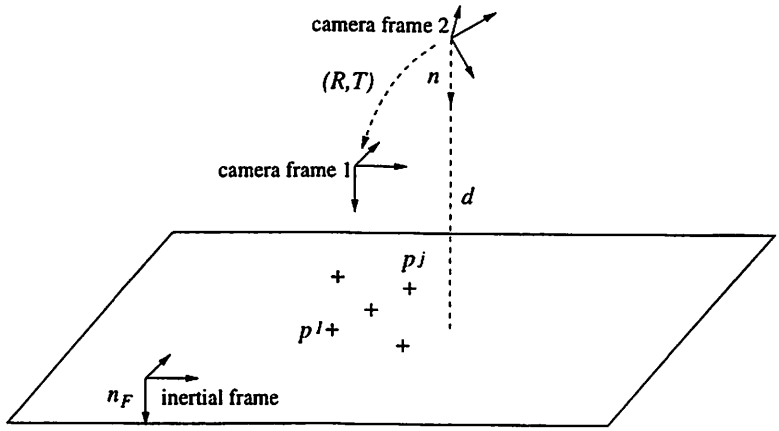


Figure 11.1: Geometry of camera frames relative to the landing pad.

coplanar is a degenerate case for the general-purpose 8-point algorithm and it gives rather poor estimation results [24]. Hence we need algorithms customized to the planar case. The discrete version of the planar ego-motion problem has been studied extensively [26, 61, 131]. Here we only formulate the problem and briefly revisit well-known results that can be found in [131]. Our contribution is to the continuous version of the problem. The continuous version is important when the task is to control a dynamic mobile robot such as a UAV, since velocity estimates are needed for the computation of control inputs. The continuous planar ego-motion estimation problem has also been studied by many researchers [51, 61, 105, 124], however each using a different approach. In the same spirit of the general purpose 8-point algorithms studied in Part I, We here propose a new geometric approach which unifies both the discrete and continuous scenario: First a planar discrete (or continuous) epipolar constraint is derived for image correspondences (or optical flows); secondly, such planar epipolar constraints are used to estimate a planar discrete (or continuous) essential matrix; finally use SVD (or eigenvalue-decomposition) of the essential matrix to recover the unknown motion or structure parameters.

11.2.1 Review of the Discrete Case

Suppose we have a set of n fixed coplanar points $\{p^j\}_{j=1}^n \subset \mathcal{P}$, where $\mathcal{P}$ denotes the landing plane. Without loss of generality, we take the origin of the inertial frame to be in $\mathcal{P}$. Figure 11.1 depicts the geometry of the camera frame relative to the landing pad. We will assume throughout the chapter that the optical center of the camera never

passes through the plane. We have the following proposition, which gives a constraint on the coordinates of the coplanar points.

Proposition 11.1. *Suppose that the* $(R,T) \in SE(3)$ *is the rigid body transformation from frame 2 to 1. Then the coordinates* $\{\mathbf{X}_1^j\}_{j=1}^n, \{\mathbf{X}_2^j\}_{j=1}^n$ *of the fixed coplanar points* $\{p^j\}_{j=1}^n \subset \mathcal{P}$ *in the two camera frames are related by:*

$$\mathbf{X}_1^j = \left(R + \frac{1}{d}TN^T\right)\mathbf{X}_2^j, \quad j = 1,\ldots,n \tag{11.4}$$

where d *is the perpendicular distance of camera frame 2 to the plane* $\mathcal{P}$ *and* $N \in S^2$ *is the unit surface normal of* $\mathcal{P}$ *relative to camera frame 2.*

Proof: Let $(R_1,T_1),(R_2,T_2) \in SE(3)$ be the configurations of camera frames 1 and 2, respectively. Without loss of generality, we take $R_1 = I$, and hence the rigid body transformation from frame 2 to frame 1 is $(R,T) = (R_2^T, T_1 - R_2^T T_2)$. For each $j = 1,\ldots,n$ we have:

$$\mathbf{X}_1^j = R\mathbf{X}_2^j + T \tag{11.5}$$

where $\mathbf{X}_1^j, \mathbf{X}_2^j$ are the coordinates of p^j in camera frames 1 and 2 respectively. Let $N_F \in S^2$ be the unit normal vector of the plane $\mathcal{P}$ in terms of the inertial frame. Then the surface normal in the coordinate frame of camera 2 is given by $N = R^T N_F$. If $d > 0$ denotes the distance from the plane $\mathcal{P}$ to the optical center of camera frame 2, then we have:

$$\frac{1}{d}N^T\mathbf{X}_2^j = 1, \qquad j = 1,\ldots,n. \tag{11.6}$$

Substituting equation (11.6) into equation (11.5) gives the result. ∎

We call the matrix:

$$A = \left(R + \frac{1}{d}TN^T\right) \in \mathbb{R}^{3\times 3} \tag{11.7}$$

the **planar essential matrix**, since it contains all the motion parameters $\{R,T\}$ and the structure parameters $\{N,d\}$ that we need to recover about the relative configuration between frames 1 and 2. Notice that due to the inherent scale ambiguity in the term $\frac{1}{d}T$ in equation (11.7), the vision sensor can in general only recover the ratio of the camera translation scaled by the inverse distance to the plane. In section 11.4.1 we show how to resolve this ambiguity when the vision sensor is used for landing.

Proposition 11.2 (Planar Discrete Epipolar Constraint). *The matrix* $A = (R + \frac{1}{d}TN^T)$ *satisfies the constraint:*

$$(I - \mathbf{x}_1^j e_3^T) A \mathbf{x}_2^j = 0, \qquad j = 1, \ldots, n \tag{11.8}$$

where $\{\mathbf{x}_1^j\}_{j=1}^n$, $\{\mathbf{x}_2^j\}_{j=1}^n$, *are the images of* $\{p^j\}_{j=1}^n$ *with respect to camera frames 1 and 2 respectively.*

Proof: Simply apply equation (11.3) to equation (11.4) ∎

Equation (11.8) is the **planar discrete epipolar constraint**. Since the constraint given by Lemma 11.2 is **linear** in A, by stacking the entries of A as a vector:

$$a = (a_{11}, a_{12}, a_{13}, a_{21}, \ldots, a_{33})^T \in \mathbb{R}^9$$

, we may re-write equation (11.8) as $\mathbf{f}_i a = 0$, where $\mathbf{f}_i \in \mathbb{R}^{3\times 9}$ is a function of $\mathbf{x}_1^j, \mathbf{x}_2^j$. Since the third row in equation (11.8) is all zeroes, the third row of $\mathbf{f}_i$ contains all zeroes, so we simply drop it and take $\mathbf{f}_i \in \mathbb{R}^{2\times 9}$. With this notation, given n image points correspondences, by defining $\mathbf{F} = (\mathbf{f}_1^T, \ldots, \mathbf{f}_m^T)^T \in \mathbb{R}^{2n\times 9}$ we can combine the equations (11.8) and rewrite them as:

$$\mathbf{F}a = 0. \tag{11.9}$$

In order to solve uniquely (up to a scale) for a, we must have $rank(\mathbf{F}) = 8$. Each pair of image point correspondences gives two constraints, hence we would expect that at least four point correspondences would be necessary for the estimation of A. We say a set of coplanar points are in **general configuration** if there is a set of four points such that no three are collinear.

Proposition 11.3 (Weng [131]). $rank(\mathbf{F}) = 8$ *if and only if the points* $\{p^j\}_{j=1}^n$ *are in general configuration in the plane.*

Proposition 11.3 says that if there are at least four point correspondences of which no three are collinear, then we may apply standard linear least squares estimation to recover A up its scale. That is, we can recover $A_L = \xi A$ for some unknown $\xi \in \mathbb{R}$. Due to the nature of least squares estimation, as the number of feature points increases, the estimation of the A matrix, and hence the motion estimates, improves.

It turns out that the middle singular value of any matrix of the form $A = R + \frac{1}{d}TN^T$ is identically equal to 1 [26, 131]. Then, if $(\sigma_1, \sigma_2, \sigma_3)$ are the singular values of A_L, we set $A = \frac{1}{\sigma_2}A_L$, which determines A up to a sign. To get the correct sign, we use $\lambda_1^j \mathbf{x}_1^j = \lambda_2^j A \mathbf{x}_2^j$ and the fact that $\lambda_1^j, \lambda_2^j > 0$ to impose the constraint $(\mathbf{x}_1^j)^T A \mathbf{x}_2^j > 0$ for $j = 1, \ldots, n$. Thus, we have that if the points $\{p\}_{j=1}^n$ are arranged in general configuration then the matrix A can be uniquely estimated from the image measurements. Once we have recovered A, we need some more SVD analysis in order to decompose it into its motion and structure parameters. For the details on the decomposition please refer to [131]. In general, for a matrix $A = (R + \frac{1}{d}TN^T)$, there are two physically possible solutions for its decomposition into parameters $\{R, \frac{T}{d}, N\}$. In section 11.4.1 we give a method of disambiguating the solutions when the task is landing a UAV on a landing pad whose geometry is known *a priori*.

11.2.2 Continuous Case

Here, in addition to measuring image points, we measure **optical flows** $\mathbf{u} = \dot{\mathbf{x}}$.

Proposition 11.4. *Suppose the camera undergoes a rigid motion with body linear and angular velocities $\omega(t), v(t)$. Then the coordinates of coplanar points $\{p\}_{j=1}^n$ in the camera frame satisfy:*

$$\dot{\mathbf{X}}^j(t) = \left(\widehat{\omega} + \frac{1}{d}vN^T\right)\mathbf{X}^j(t), \quad j = 1, \ldots, n. \tag{11.10}$$

Proof: Each of the points $\mathbf{X}^j$ satisfies:

$$\dot{\mathbf{X}}^j = \widehat{\omega}\mathbf{X}^j + v. \tag{11.11}$$

Let $N(t) = R(t)N_F$, be the surface normal to $\mathcal{P}$ in the camera frame at time t, where $R(t)$ is the orientation of the camera frame. Then, if $d(t) > 0$ is the distance from the optical center of the camera to the plane $\mathcal{P}$ at time t, then:

$$\frac{1}{d(t)}N(t)^T\mathbf{X}^j(t) \equiv 1, \quad j = 1, \ldots, n. \tag{11.12}$$

Substituting equation (11.12) into equation (11.11) gives the result. ∎

We call the matrix:

$$B = \left(\widehat{\omega} + \frac{1}{d}vN^T\right) \in \mathbb{R}^{3\times 3} \tag{11.13}$$

the **planar continuous essential matrix**, since it contains all the continuous motion parameters $\{\omega, v\}$ and structure parameters $\{N, d\}$ that we need to recover. B is exactly a continuous version of the planar discrete essential matrix A. As in the discrete case, there is an inherent scale ambiguity in the term $\frac{1}{d}v$ in equation (11.13). Thus the vision sensor can in general only recover the ratio of the camera translational velocity scaled by the inverse distance to the plane. In section 11.4.1 we show how to resolve this ambiguity when the vision sensor is used for landing.

Estimating Matrix B

We first give a proposition which will be used to prove the main result of this section: Given image velocities of at least four points in general configuration in the plane, we can **uniquely** estimate the planar continuous essential matrix.

Proposition 11.5 (Planar Continuous Epipolar Constraint). *The matrix* $B = (\widehat{\omega} + \frac{1}{d}vN^T)$ *satisfies the constraint:*

$$\mathbf{u}^j = (I - \mathbf{x}_i e_3^T)B\mathbf{x}^j, \quad j = 1, \ldots, n \tag{11.14}$$

where $\{\mathbf{x}^j(t), \mathbf{u}^j(t)\}_{j=1}^n$ *are image points and optical flow of points* $\{p^j\}_{j=1}^n$ *in the landing plane.*

Proof: We will drop the superscript j for ease of notation. Differentiating $\lambda\mathbf{x} = \mathbf{X}$ and substituting $\dot{\mathbf{X}} = B\mathbf{X}$ gives $\lambda\dot{\mathbf{x}} + \dot{\lambda}\mathbf{x} = \lambda B\mathbf{x}$. Differentiating $\lambda = e_3^T\mathbf{X}$ gives $\dot{\lambda} = \lambda e_3^T B\mathbf{x}$. Using these relations and eliminating λ gives the result. ∎

Equation (11.14) is the **planar continuous epipolar constraint**. Since the constraint is linear in B, by stacking the entries of B as $b = (b_{11}, b_{12}, b_{13}, b_{21}, \ldots, b_{33})^T \in \mathbb{R}^9$, we may re-write (11.14) as $u^j = \mathbf{g}^{jT}b$, where $\mathbf{g}^j \in \mathbb{R}^{9\times 3}$ is a matrix function of $\mathbf{x}^j$. However, since the third row of equation (11.14) contains only zeros, each image point velocity only imposes two constraints on the matrix B. Given a set of n image point and velocity pairs $\{\mathbf{x}^j, \mathbf{u}^j\}_{j=1}^n$ of fixed points in the plane, we may stack each equation $\mathbf{u}^j = \mathbf{g}^{jT}b$ into a single equation:

$$\mathbf{U} = \mathbf{G}b \tag{11.15}$$

where $\mathbf{U} = (\mathbf{u}^{jT}, \ldots, \mathbf{u}^{jT})^T \in \mathbb{R}^{3n}$ and $\mathbf{G} = (\mathbf{g}^1, \ldots, \mathbf{g}^n)^T \in \mathbb{R}^{3n\times 9}$.

Proposition 11.6. *$rank(\mathbf{G}) = 8$ if and only if the points $\{p^j\}_{j=1}^n$ are in general configuration in the plane.*

Proof: We will use the fact that a set of points in the plane are collinear if and only if the images of the points are collinear in the image plane [131]. This allows us to work with the images of features points on the plane.

For sufficiency, suppose there exists a set of four points in the plane such that no three are collinear. By contradiction, we will prove that the corresponding eight rows of $\mathbf{G}$ are linearly independent. In the following we use the notation $\mathbf{x}^j = [x^j, y^j, z^j]^T$, $j = 1, \ldots, 4$. Suppose that the matrix:

$$\mathbf{G}^T = - \begin{bmatrix} \mathbf{x}^1 & 0 & \mathbf{x}^2 & 0 & \mathbf{x}^3 & 0 & \mathbf{x}^4 & 0 \\ 0 & \mathbf{x}^1 & 0 & \mathbf{x}^2 & 0 & \mathbf{x}^3 & 0 & \mathbf{x}^4 \\ -x^1\mathbf{x}^1 & -y^1\mathbf{x}^1 & -x^2\mathbf{x}^2 & -y^2\mathbf{x}^2 & -x^3\mathbf{x}^3 & -y^3\mathbf{x}^3 & -x^4\mathbf{x}^4 & -y^4\mathbf{x}^4 \end{bmatrix} \in \mathbb{R}^{9\times 8}$$

has $rank(\mathbf{G}) < 8$. Then there exists $\xi = (a_1, c_1, a_2, c_2, a_3, c_3, a_4, c_4)^T \in \mathbb{R}^8$ such than $\xi \neq 0$ and $\mathbf{G}^T\xi = 0$. Define $d_j \equiv a_j x^j + c_j y^j$. Now let $\mathbf{a} = (a_1, a_2, a_3, a_4)^T$, $\mathbf{c} = (c_1, c_2, c_3, c_4)^T$, $\mathbf{d} = (d_1, d_2, d_3, d_4)^T$ and define $X = (\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4) \in \mathbb{R}^{3\times 4}$. With this notation, the condition $\mathbf{G}^T\xi = 0$, $\xi \neq 0$ implies $\mathbf{a} \neq 0$ or $\mathbf{c} \neq 0$ and $X\mathbf{a} = X\mathbf{c} = X\mathbf{d} = 0$.

Without loss of generality, take $\mathbf{a} \neq 0$. Since by the hypothesis, no three of $\mathbf{x}^j$ are collinear, each set of 3 columns of X are linearly independent. Since $X\mathbf{a} = 0$, then if one component of $\mathbf{a}$ is zero, then we must have $\mathbf{a} = 0$. Hence $\mathbf{a} \neq 0$ implies $a_j \neq 0$ for $j = 1, \ldots, 4$. Since each set of 3 columns of X are linearly independent, we have $\dim(\ker(X)) \leq 1$ and $\exists \gamma, \delta \in \mathbb{R}$ such that $\mathbf{c} = \gamma\mathbf{a}$ and $\mathbf{d} = \delta\mathbf{a}$. This implies:

$$d_j = a_j x^j + c_j y^j = a_j x^j + \gamma a_j y^j = \delta a_j. \tag{11.16}$$

But since $a_j \neq 0$ for each j, we have $x^j + \gamma y^j = \delta$ which implies that all four image points $\mathbf{x}^j$ are collinear in the image plane, resulting in a contradiction.

For necessity, we first show that if all points are collinear, then $rank(\mathbf{G}) \leq 5$. Let $u = (\alpha, \beta, 0) \in \mathbb{R}^3$ be the unit normal to the line in the image plane containing the image points $\mathbf{x}^j, j = 1, \ldots, n$. That is $\mathbf{x}^{jT}u = 0$ for $j = 1, \ldots, n$. Define four vectors in $\mathbb{R}^9$ by:

$$h_1 = \begin{bmatrix} u \\ 0 \\ 0 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 \\ u \\ 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}, \quad h_4 = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \in \mathbb{R}^9 \tag{11.17}$$

where $(e_1, e_2, e_3) = I_{3\times 3}$. Using $u^T u = 1$ and $e_3^T u = 0$, it is direct to check that for $H = (h_1, h_2, h_3, h_4) \in \mathbb{R}^{9\times 4}$, $\det(H^T H) = 2$ and hence $rank(H) = 4$. From the structure of $\mathbf{G}$ in equation (11.16) is is clear that $\mathbf{G}h_i = 0$ for $i = 1, \ldots, 4$. Then $\dim(\ker(\mathbf{G})) \geq 4$ and hence $rank(\mathbf{G}) \leq 9 - 4 = 5$.

Now suppose condition of the proposition is not satisfied. The claim is trivially proved if the number of image points is less than 3. Suppose there are more than 4 image points, not all collinear, and for each set of four points at least 3 are collinear. Without loss of generality, suppose $\mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$ lie on a line (call this the common line), and $\mathbf{x}^1$ does not lie on the common line. By induction, we prove that all $\mathbf{x}^j$'s for $j \geq 4$ lie on the common line. Suppose $\mathbf{x}^j$ lies on the common line for some $j \geq 4$ and $\mathbf{x}^{j+1}$ does not. Choose two points out of $\mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^j$ such that they do not lie at the intersection of the common line and the line connecting $\mathbf{x}^1, \mathbf{x}^{j+1}$. Call these points $\mathbf{x}^k, \mathbf{x}^l$. Then the four points $\mathbf{x}^1, \mathbf{x}^k, \mathbf{x}^l, \mathbf{x}^{j+1}$ are in a general configuration. This is a contradiction, and hence $\mathbf{x}^{j+1}$ lies on the common line. Since all image points lie on a single line expect for $\mathbf{x}^1$, then $rank(\mathbf{G}) \leq 5 + 2 = 7$. ∎

If the points are in general configuration in the plane then using linear least squares techniques equation (11.15) can be used to recover b up to one dimension, since $\mathbf{G}$ has a one dimensional null space. That is, we can recover $B = B_L + \xi B_K$ where B_L corresponds to the minimum norm linear least squares estimate of B, B_K corresponds to a vector in $\ker(\mathbf{G})$ and $\xi \in \mathbb{R}$ is an unknown scale. By inspection of equation (11.14) one can see that $B_K = I$. Then we have:

$$B = B_L + \xi I. \tag{11.18}$$

Thus, in order uniquely estimate B, we only need to recover the unknown ξ. So far, we have not considered the special structure of the B matrix. Next we give constraints imposed by the structure of B which can be used to recover ξ, and thus uniquely estimate B.

Lemma 11.7. *Suppose $u, v \in \mathbb{R}^3$, and $\|u\|^2 = \|v\|^2 = \alpha$. If $u \neq v$, the matrix $D = uv^T + vu^T \in \mathbb{R}^{3\times 3}$ has eigenvalues $\{\lambda_1, 0, \lambda_3\}$, where $\lambda_1 > 0$, and $\lambda_3 < 0$. If $u = \pm v$, the matrix D has eigenvalues $\{\pm 2\alpha, 0, 0\}$.*

Proof: Let $\beta = u^T v$. If $u \neq \pm v$, we have $-\alpha < \beta < \alpha$. We can solve the

eigenvalues and eigenvectors of D by inspection:

$$\begin{aligned} D(u+v) &= (\beta+\alpha)(u+v) \\ D(u\times v) &= 0 \\ D(u-v) &= (\beta-\alpha)(u-v). \end{aligned}$$

Clearly $\lambda_1 = (\beta+\alpha) > 0$ and $\lambda_3 = \beta - \alpha < 0$. It is direct to check the conditions on D when $u = \pm v$. ∎

Theorem 11.8. *The matrix B can be uniquely estimated from the image measurements if and only if there are four points of $\{p^j\}_{j=1}^n$ in the plane such that no three are collinear.*

Proof: In this proof, we will work with sorted eigenvalues, that is if $\{\lambda_1, \lambda_2, \lambda_3\}$ are eigenvalues of some matrix, then $\lambda_1 \geq \lambda_2 \geq \lambda_3$. If the points are not in general configuration, then by Proposition 11.6, $rank(\mathbf{G}) < 7$, and the problem is under-constrained. Now suppose the points are in general configuration. Then by least squares estimation we may recover $B = B_L + \xi I$ for some unknown $\xi \in \mathbb{R}$. By Lemma 11.7, we have that $B + B^T = \frac{1}{d}vN^T + \frac{1}{d}Nv^T$ has eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ where $\lambda_1 \geq 0$, $\lambda_2 \equiv 0$, and $\lambda_3 \leq 0$. Compute the eigenvalues of $B_L + B_L^T$ and denote them as $\{\gamma_1, \gamma_2, \gamma_3\}$. Since we have $B = B_L + \xi I$, then $\lambda_i = \gamma_i + 2\xi$, for $i = 1, 2, 3$. Since we must have $\lambda_2 = 0$, we have $\xi = -\frac{1}{2}\gamma_2$, and set $B = B_L - \frac{1}{2}\gamma_2 I$. ∎

Decomposing Matrix B

We now address the task of decomposing B into its motion and structure parameters. The following constructive proof gives a new technique for the recovery of motion and structure parameters.

Theorem 11.9. *Given a matrix $B \in \mathbb{R}^{3\times 3}$ in the form $B = \widehat{\omega} + \frac{1}{d}vN^T$, one can recover the motion and structure parameters $\{\widehat{\omega}, \frac{v}{d}, N\}$ up to at most 2 physically possible solutions. There is a unique solution if $v = 0$, $v \times N = 0$ or $e_3^T v = 0$, where $e_3 = [0, 0, 1]^T$ is the optical axis.*

Proof: Compute the eigenvalue/eigenvector pairs of $B + B^T$ and denote them as $\{\lambda_i, u_i\}$, $i = 1, 2, 3$. If $\lambda_i = 0$ for $i = 1, 2, 3$, then we have $v = 0$ and $\widehat{\omega} = B$. In this case we can not recover the normal of the plane N. Otherwise, if $\lambda_1 > 0$, and $\lambda_3 < 0$, then

we have $v \times N \neq 0$. Let $\alpha = \|v/d\| > 0$, let $\tilde{v} = v/\sqrt{\alpha}$ and $\tilde{N} = \sqrt{\alpha}N$, and let $\beta = \tilde{v}^T\tilde{N}$. According to Lemma 11.7, the eigenvalue/eigenvector pairs of $B + B^T$ are given by:

$$\begin{cases} \lambda_1 = \beta + \alpha > 0, & u_1 = \frac{1}{\|\tilde{v}+\tilde{N}\|}(\tilde{v} + \tilde{N}) \\ \lambda_3 = \beta - \alpha < 0, & u_3 = \frac{1}{\|\tilde{v}-\tilde{N}\|}(\tilde{v} - \tilde{N}). \end{cases} \tag{11.19}$$

Then $\alpha = \frac{1}{2}(\lambda_1 - \lambda_3)$. It is direct to check that $\|\tilde{v} + \tilde{N}\|^2 = 2\lambda_1$, $\|\tilde{v} - \tilde{N}\|^2 = -2\lambda_3$. Then together with (11.19), we have a solution:

$$\begin{cases} \tilde{v}_1 &= \frac{1}{2}(\sqrt{2\lambda_1}\, u_1 + \sqrt{-2\lambda_3}\, u_3) \\ \tilde{N}_1 &= \frac{1}{2}(\sqrt{2\lambda_1}\, u_1 - \sqrt{-2\lambda_3}\, u_3) \\ \widehat{\omega}_1 &= \frac{1}{2}((B - \tilde{v}_1\tilde{N}_1^T) - (B - \tilde{v}_1\tilde{N}_1^T)^T). \end{cases} \tag{11.20}$$

The estimate of $\widehat{\omega}_1$ is computed as above because, in the presence of noise, in general $B - \tilde{v}_1\tilde{N}_1^T$ is not necessarily an element in $so(3)$. We here take the projection of $B - \tilde{v}_1\tilde{N}_1^T$ onto $so(3)$.

However, the eigenvalue-decomposition $\{\lambda_i, u_i\}$ is not unique – there is a sign ambiguity in the eigenvectors u_1 and u_3. This sign ambiguity leads to a total of 4 possible solutions for $\tilde{v}$ and $\tilde{N}$ computed according to (11.20). It is direct to check that that if $\{\widehat{\omega}, \frac{v}{d}, N\}$ are the true motion and structure parameters, then the 4 possible solutions obtained by (11.20) are:

Solution 1 (true)	$v_1 = v/d$ $N_1 = N$ $\widehat{\omega}_1 = \widehat{\omega}$		Solution 3	$v_3 = -v_1$ $N_3 = -N_1$ $\widehat{\omega}_3 = \widehat{\omega}_1$	
Solution 2	$v_2 = \|v/d\|\, N$ $N_2 = \frac{1}{\|v/d\|}v/d$ $\widehat{\omega}_2 = \widehat{\omega} - Nv^T/d + vN^T/d$		Solution 4	$v_4 = -v_2$ $v_4 = -N_2$ $\widehat{\omega}_3 = \widehat{\omega}_2$	

In order to reduce the number of physically possible solutions, we impose the so-called "positive depth constraint" – since the camera can only see points that are in front of it, we must have $N^T e_3 > 0$. This constraint eliminates solution 3 as being physically impossible. If $v^T e_3 \neq 0$, one of solutions 2 or 4 will be eliminated, whereas if $v^T e_3 = 0$ both solutions 2 and 4 are eliminated. For the case that $v \times N = 0$, it is easy to see that solutions 1 and 2 are equivalent, and that imposing the positive depth constraint leads to a unique solution for all motion and structure parameters. ∎

The results for the ambiguities of solutions were also reported in [51, 105, 124]. In section 11.4.1 we give a method of disambiguating the solutions when the task is landing a UAV on a landing pad whose geometry is known *a priori*.

11.2.3 Implementation Issues

For both the discrete and continuous algorithms, the most computationally intensive task is the linear least squares estimation of the A and B matrices, which involves the singular value decomposition (SVD) of the matrices $\mathbf{F}, \mathbf{G} \in \mathbb{R}^{2n \times 9}$ where n is the number of tracked feature points. The cost of the SVD of a matrix $M \in \mathbb{R}^{m \times n}$ for $m \leq n$ is $O(m^2 n)$ flops. Then, as the number of tracked feature points n increases, the cost of the vision algorithms grows as $O(n)$.

We have implemented the above algorithms using the MATHLIB C++ library in Matlab, and have found that on a 450 MHz Pentium II running Linux, the vision algorithms can perform motion estimation based on 25 tracked feature points at a rate of over 150 Hz, a rate far beyond that of most current real-time feature tracking hardware.

11.2.4 Simulation of Motion Estimation Algorithms

Since our goal is to use the estimated motion and structure from the vision as a sensor in a control loop, of utmost consideration is the performance of this sensor in the presence of noise in the measurements of point correspondences and image velocity. Another important criteria to analyze is how the estimation errors depend on different camera motions with respect to the observed plane. To this end, we have implemented both the discrete and continuous algorithms and performed various simulations in order to evaluate their performance. In order to assess the performance of the planar algorithms, for all simulations we compare the results with the traditional 8-point algorithm as described in Chapter 3.

For all simulations, we generated 50 random points uniformly distributed within the field of view of the camera, FOV = 60°. The image correspondences and the image velocity measurements were corrupted by additive white Gaussian noise. For evaluating the 8-point algorithm, we randomly scattered the depths of these points uniformly between distance of `zmin` and `zmax` focal lengths, where for all simulations, we set `zmax` = 400 and `zmin` = 100 unless otherwise noted. For evaluating the planar algorithm, we placed the

points on the fronto-parallel plane at a distance of (zmax+zmin)/2. Each data point on each plot is the mean result of 50 trials for a given motion, noise level, and distance. We studied the performance of the algorithms as a function of depth variation, noise in the image measurements, and motion about different translation/rotation axes.

Depth Sensitivity

In planar case, our depth variation analysis attempts to see how the errors in the estimates depend on the depth of the plane being viewed. Notice that in the matrices $A = R + \frac{1}{d}TN^T$ and $B = \widehat{\omega} + \frac{1}{d}vN^T$, the translation term is scaled by the inverse distance of the plane. Thus, for a fixed translation and a fixed noise level, as the distance of the plane increases, the "signal" from the translation term decreases while the noise level stays constant. Thus, one would expect that as the signal to noise ratio decreases, the performance of the algorithms also decrease. Also, from the structure of the A and B matrices, we see that the errors in the rotational components should not depend on the depths of the points. This expectation was validated as shown in Figure 11.2.

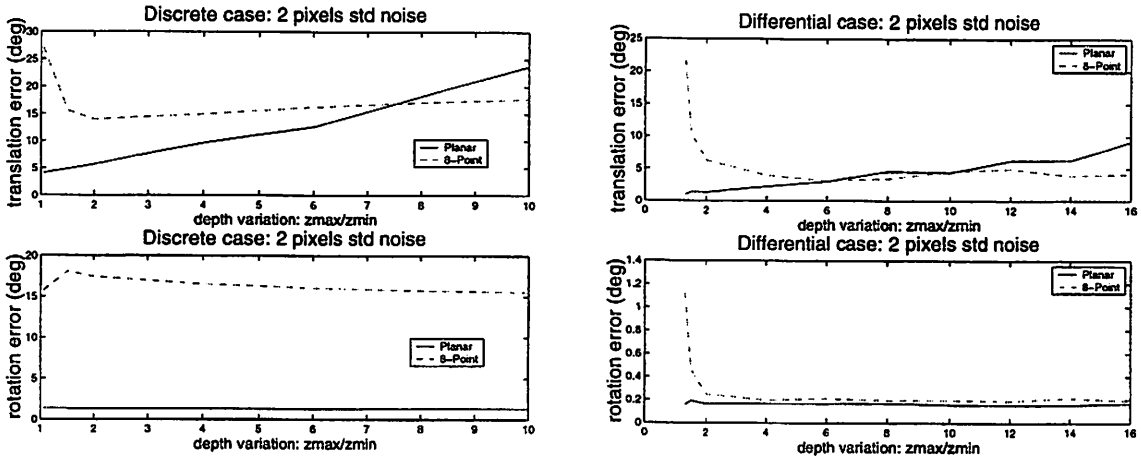


Figure 11.2: Depth sensitivity.

Notice that for very low depth variation, the 8-point algorithm for both discrete and continuous case performs poorly. This is a result of singularities that occur in the algorithm when the feature points are coplanar. Also, notice that for the planar case, as expected, the errors increase as the distance of the plane increases. One interesting observation is that for the discrete case, the rotation estimate is always better in the planar

case than in the general case.

Noise Sensitivity

In the simulations presented in Figure 11.3, for a given motion we corrupted the correspondences and image velocities with increasing levels additive white Gaussian noise. Notice from the simulation results that for both discrete and continuous cases, the planar algorithm performs better that the 8-point algorithm.

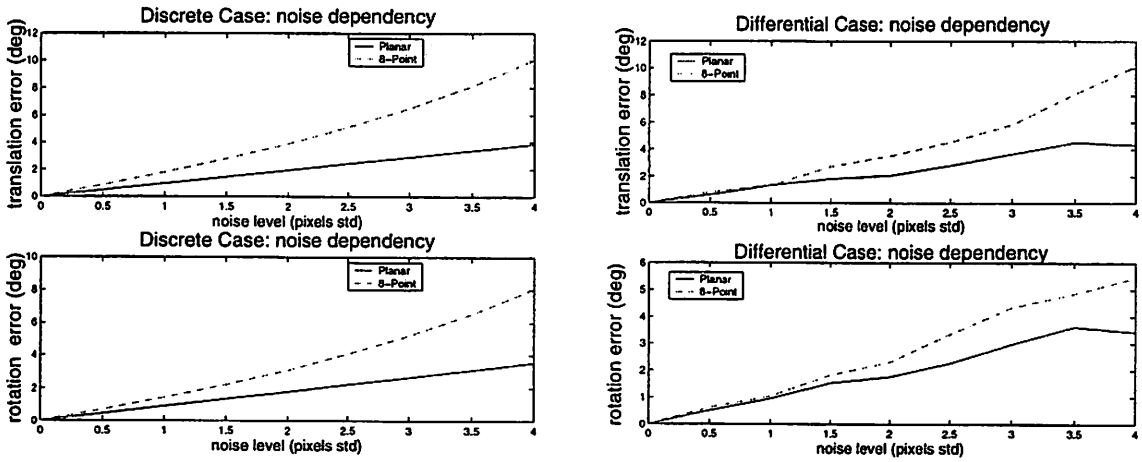


Figure 11.3: Noise Sensitivity.

Motion Sensitivity

Next we study the sensitivity of the algorithm with respect to different motions relative to the plane. We ran the algorithms for a motion about each translation-rotation axis pair for two different noise levels (low and high). In general, the planar algorithms perform better than the 8-point algorithms except when the translation axis is parallel optical axis (and hence the surface normal of the plane). The higher sensitivity in that case can seen as an overall numerical sensitivity to perturbations in the algebraic eigenvalue/eigenvector problem when there are repeated eigenvalues. For example, if a matrix has a pair of repeated eigenvalues then any vector in certain two dimensional subspace can be considered an eigenvector corresponding to the repeated eigenvalue. Because the eigenvectors corresponding to repeated eigenvalues are defined up to subspace, it is intuitive to see that for

two different perturbations of the matrix, the corresponding eigenvectors could be quite different. A similar phenomenon occurs in the case of repeated singular vectors. Thus, an algorithm that uses the computation of eigenvectors (singular vectors) is inherently sensitive to noise in the case of repeated eigenvalues (singular values).

The situation of having repeated eigenvalues (singular values) occurs in the planar continuous (discrete) algorithm in the case that the translational motion is parallel to the surface normal of the plane. In the 8-point algorithm, the situation of repeated eigenvalues occurs in the case that the translation and rotation axes are parallel. The simulation results for both the discrete (in Figure 11.4) and the continuous case (in Figure 11.5) validate our expectation of higher noise sensitivity in the case of repeated singular values and eigenvalues.

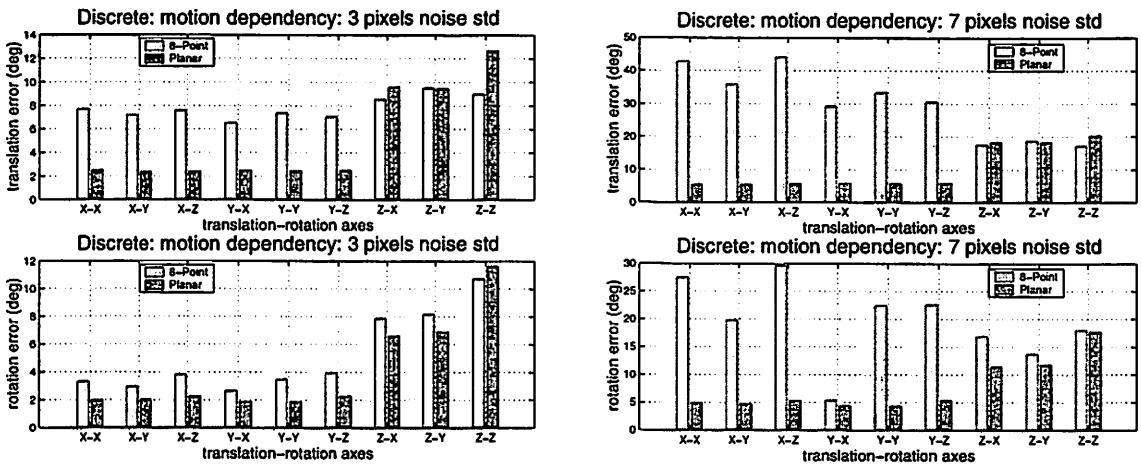


Figure 11.4: Discrete Case: sensitivity to translation-rotation axes.

11.3 Nonlinear Control for a UAV Dynamic Model

In this section, we present the dynamical model of the UAV, a control design based on differential flatness, and a stability analysis of the closed-loop system. The proposed controller is general in the sense that it can be applied towards trajectory tracking. For the purpose of landing, the UAV is asked to track a fixed point at the desired configuration above the landing pad.

We parameterize the orientation $R \in SO(3)$ of the UAV relative to the inertial frame by the ZYX (or "roll, pitch, yaw") Euler angles denoted by $\Theta = [\phi, \theta, \psi]^T$. Thus

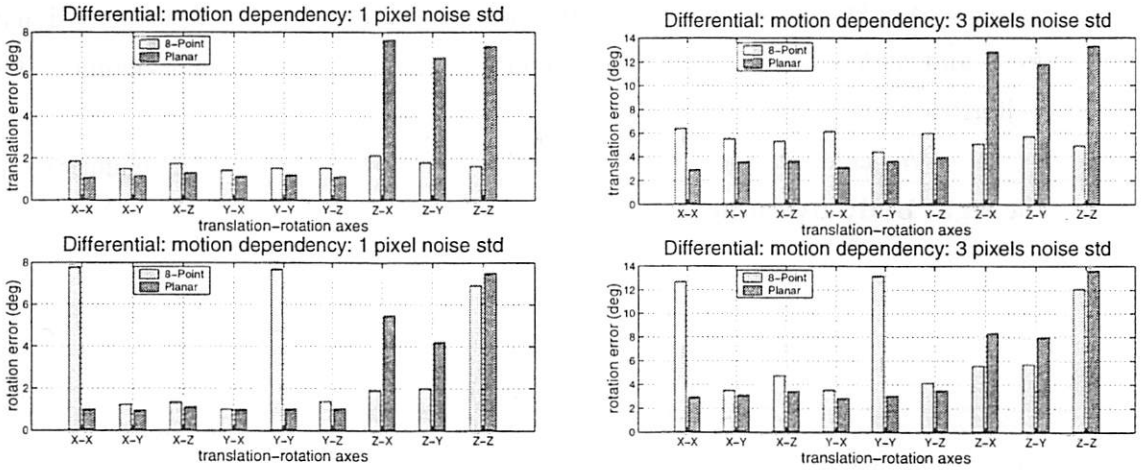


Figure 11.5: Continuous Case: sensitivity to translation-rotation axes.

we have $R = \exp(\widehat{e}_3\psi)\exp(\widehat{e}_2\theta)\exp(\widehat{e}_1\phi)$ with $e_1 = [1,0,0]^T$, $e_2 = [0,1,0]^T$, $e_3 = [0,0,1]^T$. Under this parameterization, there is a mapping $\Psi(\Theta) \in \mathbb{R}^{3\times 3}$ given by:

$$\Psi(\Theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{11.21}$$

which maps the body rotational velocity to Euler angle velocity, that is: $\dot{\Theta} = \Psi\omega$.

11.3.1 System Dynamics

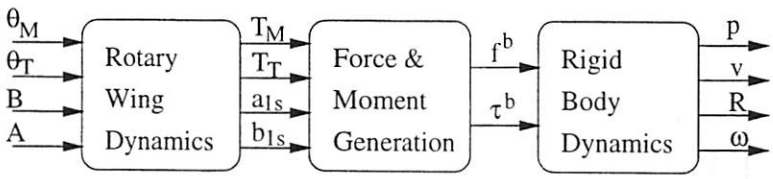


Figure 11.6: Block diagram of UAV dynamics.

A complete model of a helicopter can be divided into four different subsystems: **actuator dynamics, rotary wing dynamics, force and moment generation processes**, and **rigid body dynamics**. The dynamics of the engine and actuators (which depend on the flexibility of the rotors and fuselage) are quite complex and intractable for

analysis. We here consider a helicopter model including only the rigid body dynamics, the force and moment generation process and a simplified rotary wing dynamics. This model is illustrated in Figure 11.6.

We now articulate each of the three subsystems. First, the equations describing the **rigid body dynamics** are given by:

$$\begin{cases} \ddot{p} &= \frac{1}{m} R f^b \\ \dot{\Theta} &= \Psi \omega \\ \dot{\omega} &= \mathcal{I}^{-1}(\tau^b - \omega \times \mathcal{I}\omega) \end{cases} \tag{11.22}$$

where $m > 0$ is the body mass, $\mathcal{I} \in \mathbb{R}^{3\times 3}$ is the inertial matrix and $f^b, \tau^b \in \mathbb{R}^3$ are the **body force** and **torque** given by:

$$\begin{cases} f^b &= \begin{bmatrix} X_M \\ Y_M + Y_T \\ Z_M \end{bmatrix} + R^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \\ \tau^b &= \begin{bmatrix} R_M \\ M_M + M_T \\ N_M \end{bmatrix} + \begin{bmatrix} Y_M h_M + Z_M y_M + Y_T h_T \\ -X_M h_M + Z_M l_M \\ -Y_M l_M - Y_T l_T \end{bmatrix}. \end{cases} \tag{11.23}$$

The body forces and torques generated by the main rotor are controlled by T_M, a_{1s} and b_{1s}, in which a_{1s} and b_{1s} are the longitudinal and lateral tilt of the tip path plane of the main rotor with respect to the shaft, respectively. The tail rotor is considered as a source of pure lateral force Y_T and anti-torque Q_T, which are controlled by T_T. The forces and torques can be expressed as:

$$\begin{cases} X_M &= -T_M \sin a_{1s} & R_M &\simeq \frac{\partial R_M}{\partial b_{1s}} b_{1s} - Q_M \sin a_{1s} \\ Y_M &= T_M \sin b_{1s} & M_M &\simeq \frac{\partial M_M}{\partial a_{1s}} a_{1s} + Q_M \sin b_{1s} \\ Z_M &= -T_M \cos a_{1s} \cos b_{1s} & N_M &\simeq -Q_M \cos a_{1s} \cos b_{1s} \\ Y_T &= -T_T & M_T &= -Q_T. \end{cases} \tag{11.24}$$

The moments generated by the main and tail rotor can be calculated from the constants $\{l_M, y_M, h_M, h_T, l_T\}$, where h_i, l_i and y_i denote the vertical, longitudinal, and lateral distance between the center of gravity and the center of the rotor specified by $i = M$ or T. These system parameters are given in Appendix B. In the simulation, we will approximate

the rotor torque equations by $Q_i \simeq C_i^Q T_i^{1.5} + D_i^Q$ for $i = M, T$, with details described in [59]. The values of C_i^Q, D_i^Q are also given in Appendix B.

Finally, the **rotary wing dynamics** are in general harder to express explicitly. In an operating region near hovering, the rotary wing dynamics can be approximated by the following equations (for details see [90]):

$$T_M = c_{M1}\theta_M + c_{M3}\theta_M^3, \quad T_T = c_{T1}\theta_T + c_{T3}\theta_T^3, \quad a_{1s} = -B, \quad b_{1s} = A$$

where θ_M, θ_T are the main and tail rotor collective pitch, and B, A are the longitudinal and lateral cyclic pitch.

11.3.2 Inner and Outer System Partitioning

A system $\dot{x} = f(x, t, u)$ is called **differentially flat** if there exist output functions, called **flat outputs**, such that all states and inputs can be expressed in terms of the flat outputs and their derivatives [28]. Differential flatness has been applied to approximate models of aircraft [27] and helicopter [56] for trajectory generation. The full helicopter dynamics are not flat in general, however it can be shown that the dynamics can be partitioned into an "inner system" (*e.g.* the attitude dynamics) and an "outer system" (*e.g.* the position dynamics) where the outer system is flat. This scheme has been successfully used for generating a two stage control synthesis for many systems which are not completely flat [121]. Such a scheme which utilizes the flatness of the outer system is roughly illustrated in Figure 11.7. In the figure, P_O is the outer system which is flat, and P_I is the inner system which is not necessarily flat. Given a desired output trajectory, say $y_d^O(\cdot)$, the mapping F in Figure 11.7 utilizes the flatness property of the outer system to generate an desired output trajectory $y_d^I(\cdot)$ for the inner system. The control synthesis for the overall system then reduces to the design of an inner system controller, C, which drives the inner system output $y^I(t) \to y_d^I(t)$ (exponentially) as $t \to \infty$. As the inner system output converges, one can show that the outer system output converges to the desired one, $y^O(t) \to y_d^O(t)$ as $t \to \infty$. That is, the overall system asymptotically tracks the desired trajectory.

It has been shown in [56] that the helicopter dynamics are *approximately* differentially flat with the position and heading $\{p, \psi\}$ as the flat outputs. The approximation is based on the assumption that the coupling terms a_{1s}, b_{1s}, T_T are small and can be neglected

in the model. So if $a_{1s}, b_{1s}, T_T \approx 0$, the outer system dynamics (11.22) can be rewritten as:

$$\ddot{p} = \frac{1}{m} R(\Theta) \begin{bmatrix} 0 \\ 0 \\ -T_M \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + h \tag{11.25}$$

with

$$h = \frac{1}{m} R(\Theta) \begin{bmatrix} -T_M \sin a_{1s} \\ T_M \sin b_{1s} - T_T \\ -T_M (\cos a_{1s} \cos b_{1s} - 1) \end{bmatrix}.$$

where the inputs are $u^O = y^I = [\Theta^T, T_M]^T$, and the outputs are $y^O = [p, \dot{p}, \ddot{p}, \psi]^T$. One must notice that this approximation introduces a small non-vanishing modeling error h which depends on $\Theta, T_M, a_{1s}, b_{1s}, T_T$. We will soon show its effect on the stability of the closed-loop system.

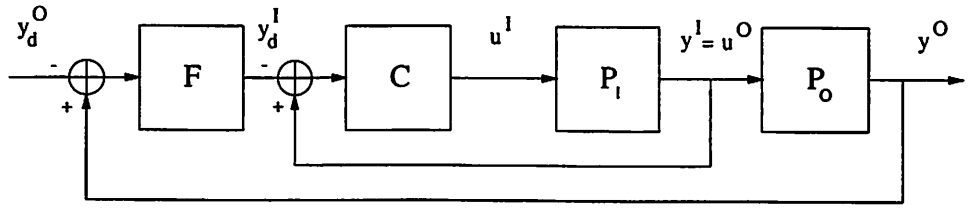


Figure 11.7: Partitioned inner and outer systems.

11.3.3 Control Design

The control design for the overall system is be based on an assumption that there exists a controller C such that $e^I = 0$ is an exponentially stable equilibrium point for the inner error system:

$$\dot{e}^I = f(e^I, e^O, t)|_{e^O=0}, \quad f(0,0,t) = 0$$

where $e^O = y^O - y_d^O$ and $e^I = y^I - y_d^I$. There have been various design methodologies proposed for the controller of the inner system, *e.g.* [59]. We here are only interested in the performance of the overall system assuming such a controller C is already available. As shown in [56], for the approximated outer system (11.25), there exists a smooth mapping

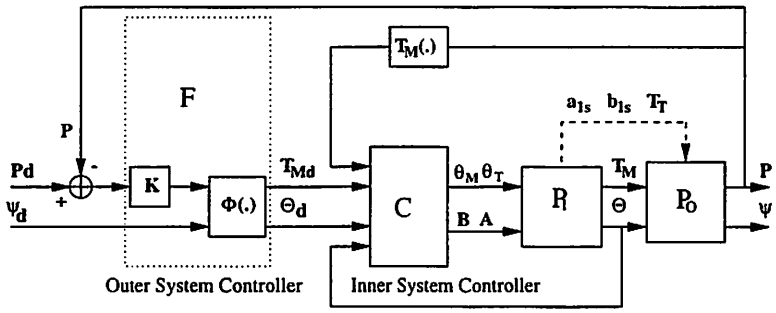


Figure 11.8: Block diagram of control scheme.

from the outer system output to the inner system output:

$$\begin{aligned} \Phi: \quad \mathbb{R}^4 &\rightarrow \mathbb{R}^4 \\ (\ddot{p}, \psi) &\mapsto (\Theta, T_M) \end{aligned}$$

which is defined by the equations:

$$\begin{cases} T_M &= m\sqrt{(\ddot{p}_x)^2 + (\ddot{p}_y)^2 + (\ddot{p}_z - g)^2} \\ \phi &= \sin^{-1}\left(\frac{-\ddot{p}_x \sin\psi + \ddot{p}_y \cos\psi}{T_M/m}\right) \\ \theta &= \text{atan2}\left(\frac{\ddot{p}_x \cos\psi + \ddot{p}_y \sin\psi}{-T_M \cos\phi/m}, \frac{\ddot{p}_z - 1}{-T_M \cos\phi/m}\right) \\ \psi &= \psi \end{cases}$$

where $\phi, \theta \neq \pm\pi/2$. Suppose that the desired output trajectory of the outer system is $y_d^O = [p_d, \dot{p}_d, \ddot{p}_d, \psi_d]^T$. To obtain the desired trajectory of the inner system, we define a pseudo-input vector:

$$v_p = \ddot{p}_d + K_v(\dot{p} - \dot{p}_d) + K_p(p - p_d) \tag{11.26}$$

where $K_p,\ K_v \in \mathbb{R}^{3\times 3}$ are control parameters. With the above pseudo-input, the desired output of the inner system y_d^I is given by:

$$(\Theta_d, T_{Md}) = \Phi(v_p, \psi_d). \tag{11.27}$$

A more detailed schematic of the controller for this system is illustrated in Figure 11.8. Clearly, if the inner system *exactly* tracks the desired trajectory (Θ_d, T_{Md}), that is, $y_d^I = y^I$ in Figure 11.7, then the behavior of the overall closed-loop system is specified by the outer system only, which, due to chosen the control law (11.26), is *approximately* a linear system with poles assigned by the parameters K_v, K_p.

Now if we summarize all conditions so far and rewrite the dynamics of the overall closed-loop system in terms of the tracking errors e^I and e^O of the inner and outer systems respectively, they have the form:

$$\begin{cases} \dot{e}^I &= f(e^I, e^O, t) \\ \dot{e}^O &= Ae^O + g(e^I, t) + h(e^I, e^O, t) \end{cases} \tag{11.28}$$

where

$$g(e^I, t) = \frac{1}{m} R(\Theta) \begin{bmatrix} 0 \\ 0 \\ -T_M \end{bmatrix} - \frac{1}{m} R(\Theta_d) \begin{bmatrix} 0 \\ 0 \\ -T_{Md} \end{bmatrix}.$$

In the above equations, $f(e^I, e^O, t)$ is in general a function of both e^I and e^O since the input of the inner system is a function of e^O. The function $h(e^I, e^O, t)$ from (11.25) is a small non-vanishing approximation error, and $g(e^I, t)$ vanishes when the inner system exactly tracks the desired trajectory, *i.e.*, $g(0, t) = 0$. Since the helicopter model is smooth and many of the parameters are physically bounded, $g(e^I, t)$ is in fact (globally) bounded as $\|g(e^I, t)\| \leq L\|e^I\|$ for some constant $L > 0$.[1]

11.3.4 Stability Analysis

We now analyze the performance of the overall closed-loop system. As we have argued before, the function f in (11.28) is in general a function of both e^I and e^O. However, in practice, the inner system is usually designed to have a much faster convergence rate than the outer system. To simplify the analysis, for now we assume that the inputs $T_{Md}(\cdot)$ and $\Theta_d(\cdot)$ of the inner system are approximately constant, and thus f is only a function of e^I (the more general case will be presented afterwards).

Recall that given an general system $\dot{x} = f(x, t)$, by the Lyapunov theorem and its converse [93], the system is exponentially stable if and only if there exists a *Lyapunov function* $V(x, t)$ satisfying:

$$\alpha_1\|x\|^2 \leq V(x, t) \leq \alpha_2\|x\|^2 \tag{11.29}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x, t) \leq -\alpha_3\|x\|^2 \tag{11.30}$$

$$\left\| \frac{\partial V}{\partial x} \right\| \leq \alpha_4\|x\| \tag{11.31}$$

[1] Such a L can be estimated from the system equation (11.22).

for some positive **Lyapunov constants** $\alpha_1, \alpha_2, \alpha_3, \alpha_4 > 0$. We can apply this theorem to both the nominal outer system $\dot{e}^O = Ae^O$ and the inner system $\dot{e}^I = f(e^I, t)$ and denote the corresponding Lyapunov functions as V^O and V^I and the Lyapunov constants as $\alpha_1, \alpha_2, \alpha_3, \alpha_4 > 0$ and $\beta_1, \beta_2, \beta_3, \beta_4 > 0$ respectively.

Theorem 11.10. *Consider the following system:*

$$\begin{cases} \dot{e}^I &= f(e^I, t) \\ \dot{e}^O &= Ae^O + g(e^I, t) \end{cases} \tag{11.32}$$

where $g(e^I, t)$ is a perturbation term that satisfies $\|g(e^I, t)\| \leq L\|e^I\|$. If both the nominal outer system $\dot{e}^O = Ae^O$ and inner system $\dot{e}^I = f(e^I, t)$ are exponentially stable, then the overall system is exponentially stable for any $L > 0$.

Proof: Apply the converse Lyapunov theorem to both the outer and inner systems, and denote the corresponding Lyapunov functions as V^O, V^I and the constants as $\{\alpha_i\}_{i=1}^4, \{\beta_i\}_{i=1}^4$ respectively. We consider the candidate Lyapunov function $V = V^I + \mu V^O$ for the overall system. Then we have:

$$\begin{aligned} \dot{V} = \dot{V}^I + \mu \dot{V}^O &\leq -\beta_3 \|e^I\|^2 - \mu\alpha_3 \|e^O\|^2 + \mu\alpha_4 L \|e^O\| \|e^I\| \\ &= -(\|e^I\|, \|e^O\|) Q (\|e^I\|, \|e^O\|)^T \end{aligned}$$

where the matrix $Q \in \mathbb{R}^{2\times 2}$ is:

$$Q = \begin{bmatrix} \beta_3 & -\frac{1}{2}\mu\alpha_4 L \\ -\frac{1}{2}\mu\alpha_4 L & \mu\alpha_3 \end{bmatrix}.$$

The matrix Q can be positive definite if and only if there exists a small enough $\mu > 0$ such that $\det(Q) > 0$. It is easy to check that it suffices to have $0 < \mu < \frac{4\beta_3\alpha_3}{\alpha_4^2 L^2}$. Such a μ always exists. Therefore, the overall system is always exponentially stable regardless of L. ∎

This theorem states a very interesting fact for the system (11.32): as long as the inner system and outer system are exponentially stable, the system is extremely robust (in terms of exponential stability) to any (vanishing) perturbation of the outer system which only depends on the tracking error of the inner system.

In the above theorem we assumed that the inner system does not depend on the tracking error e^O of the outer system. For the more general case, we may write:

$$f(e^I, e^O, t) = f(e^I, 0, t) + d(e^I, e^O, t)$$

where $d(e^I, e^O, t) = f(e^I, e^O, t) - f(e^I, 0, t)$. The nominal system $\dot{e}^I = f(e^I, 0, t)$ is exponentially stable as designed and we still denote its Lyapunov function as V^I and Lyapunov constants as $\{\beta_i\}_{i=1}^4$. Then for the overall system, following the spirit of Theorem 11.10, we have the result:

Theorem 11.11. *Consider the following perturbed system:*

$$\begin{cases} \dot{e}^I &= f(e^I, e^O, t) &= f(e^I, 0, t) + d(e^I, e^O, t) \\ \dot{e}^O &= Ae^O + g(e^I, t) \end{cases} \tag{11.33}$$

where $g(e^I, t)$ is a perturbation term that satisfies $\|g(e^I, t)\| \leq L_1\|e^I\|$ for some $L_1 > 0$. If, for $d(e^I, e^O, t)$, there exists $L_2 > 0$ such that $\|d(e^I, e^O, t)\| \leq L_2\|e^O\|$, then the overall system is exponentially stable if the product of the two Lipschitz constants satisfies the inequality:

$$L_1 \cdot L_2 < \frac{\alpha_3}{\alpha_4} \cdot \frac{\beta_3}{\beta_4}. \tag{11.34}$$

Proof: The proof is very similar to that of Theorem 11.10. We consider the candidate Lyapunov function $V = V^I + \mu V^O$ for the overall system. Then we have:

$$\begin{aligned} \dot{V} = \dot{V}^I + \mu\dot{V}^O &\leq -\beta_3\|e^I\|^2 + \beta_4 L_2\|e^I\|\|e^O\| - \mu\alpha_3\|e^O\|^2 + \mu\alpha_4 L_1\|e^O\|\|e^I\| \\ &= -(\|e^I\|, \|e^O\|)Q(\|e^I\|, \|e^O\|)^T \end{aligned}$$

where the matrix $Q \in \mathbb{R}^{2\times 2}$ is:

$$Q = \begin{bmatrix} \beta_3 & -\frac{1}{2}(\beta_4 L_2 + \mu\alpha_4 L_1) \\ -\frac{1}{2}(\beta_4 L_2 + \mu\alpha_4 L_1) & \mu\alpha_3 \end{bmatrix}.$$

Q is positive definite if and only if $\det(Q) > 0$. That is, there exists $\mu > 0$ such that:

$$-\alpha_4^2 L_1^2\mu^2 + (4\beta_3\alpha_3 - 2\beta_4 L_2\alpha_4 L_1)\mu - \beta_4^2 L_2^2 > 0.$$

This is true if and only if the discriminant of the quadratic function of μ on the left hand side is positive which yields: $L_1 \cdot L_2 < \frac{\alpha_3}{\alpha_4} \cdot \frac{\beta_3}{\beta_4}$. ∎

This theorem states a very interesting fact about the system (11.33): heuristically, α_3 and β_3 are proportional to the convergence rates of the outer and inner systems respectively,[2] hence the stability of the perturbed systems requires *only* that the *product* of the Lipschitz constants of the perturbation terms is less than the *product* of the two convergence rates, regardless of the rate of each individual system.

[2] A more precise estimates of the convergences rates are given by $\frac{\alpha_3}{2\alpha_2}$ and $\frac{\beta_3}{2\beta_2}$.

Comment 11.12. *The stability of a similar model of the overall closed-loop system has been studied before in [121], however, no explicit conditions are provided under which a μ exists such that the overall system is stable. Here, Theorems 11.10 and 11.11 give more detailed and useful results in characterizing the properties of the closed-loop system.*

Although we have established the conditions for the system (11.33) to be exponentially stable, estimates of its Lyapunov constants indeed depend on L_1, L_2 and all the Lyapunov constants of the inner and outer systems. These constants can be optimized by maximizing the smaller eigenvalue of Q with respect to μ. We here omit the detail and carry on the analysis by assuming that the system (11.33) is exponentially stable and its Lyapunov constants are denoted by $\gamma_1, \gamma_2, \gamma_3, \gamma_4 > 0$. We now want to estimate the effect of the non-vanishing error term h on the performance of the closed-loop system (11.28). In general, we can no longer expect asymptotic stability when a non-vanishing perturbation is introduced. However, according to [54], we can still have good estimates of a bound on the tracking error and the rate of convergence outside this bound.

Proposition 11.13. *Assume that the system (11.33) has the Lyapunov constants $\{\gamma_i\}_{i=1}^4$. Then, for the closed-loop system (11.28), if $\|h(e^I, e^O, t)\| \leq \delta < \frac{\gamma_3}{2\gamma_4}\sqrt{\frac{\gamma_1}{\gamma_2}}$, then the tracking error of the overall system is bounded by $b = \frac{2\gamma_4}{\gamma_3}\sqrt{\frac{\gamma_2}{\gamma_1}}\delta$, and, outside this bound, the error exponentially decreases with a rate larger than $\lambda = \frac{\gamma_3}{4\gamma_2}$.*

The control parameters K_v and K_p can be adjusted so as to minimize the error bound b. For the helicopter model, the error term $h(e^I, e^O, t)$ is usually extremely small, as is δ. We can also choose the control parameters such that the inner and outer systems have very fast rates of convergence, hence a large γ_3. Consequently, the error bound b is very small, and usually barely noticeable in simulations and experiments, as we will soon see.

11.4 Vision in the Control Loop

In this section, we discuss how the discrete and continuous motion estimation algorithms described in section 11.2 are used in the control loop for landing a UAV onto a landing pad with a known geometry.

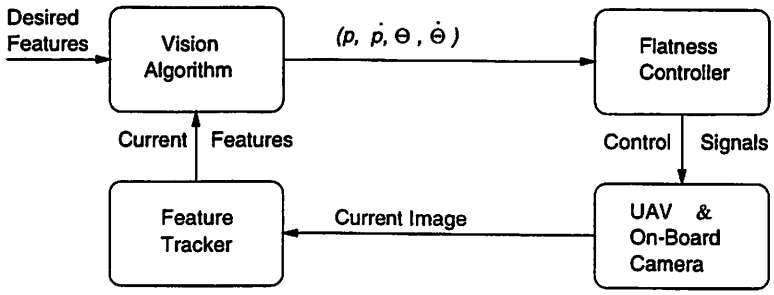


Figure 11.9: Block diagram of vision in control loop.

11.4.1 Disambiguation of Motion Estimates

We assume that the image of the landing pad taken from the *desired* landing configuration are given. The feature points on the landing pad are assumed to be in general configuration (they could for example be corners on the typical "H" pattern). Without loss of generality, suppose $(I, T_1) \in SE(3)$ is the configuration of the desired camera frame, and $d_1 = -N_F^T T_1 > 0$ is the desired distance of the camera to the landing plane with known surface normal $N_F \in \mathbb{R}^3$.

Proposition 11.14. *Suppose* $A = (R + \frac{1}{d} T N^T) \in \mathbb{R}^{3\times 3}$ *is the planar essential matrix associated with two camera frames relative to a plane. If* $d_1 > 0$ *is the distance from the first camera to the plane, then the distance of the second camera to the plane is given by* $d = d_1 / \det(A)$.

Proof: Suppose the configuration of the second camera frame is $(R_2, T_2) \in SE(3)$. Then $d_1 = -N_F^T T_1$, $d = -N_F^T R_2^T T_2$ are the distances from the first and second cameras to the plane. Since $N_F = RN$, we have $AR^T = (I + \frac{1}{d} T N_F^T)$, hence the eigenvalues of AR^T are $\{\lambda, 1, 1\}$ where $\lambda = 1 + \frac{1}{d} N_F^T T$. But $N_F^T T = N_F^T (T_1 - R_2^T T_2) = -d + d_1$. Using $\det(A) = \det(AR^T) = \lambda$, it is direct to check that $\det(A) = d_1/d$. ∎

The knowledge of N_F allows us to disambiguate the pair of solutions discussed in Theorem 11.9 by taking the one that minimizes $\|N_{\text{est}} - R_{\text{est}}^T N_F\|$, where N_{est} is the vision estimated surface normal, and R_{est} is the estimated rotation matrix according to the discrete algorithm. Also, the knowledge of d_1 allows to find d according to Proposition 11.14, which solves the scale ambiguity in T/d in the discrete case and v/d in the continuous case.

The vision algorithms described above generate estimates of $\{R, T, v, \omega\}$. However,

to compute the control signals such as (11.26) we need estimates of $\{p, \dot{p}, \Theta, \dot{\Theta}\}$. Note that given $R \in SO(3)$, the ZYX Euler angles (away from the singularity) can be recovered by:

$$\begin{cases} \theta &= \text{atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\ \phi &= \text{atan2}(r_{32}/\cos\theta, r_{33}/\cos\theta) \\ \psi &= \text{atan2}(r_{21}/\cos\theta, r_{11}/\cos\theta) \end{cases} \tag{11.35}$$

where r_{ij} is the entry of the $i-$th row and $j-$th column of R. Thus, we can recover $\{\Theta, \dot{\Theta}\}$ from $\{R, \omega\}$ by applying equations (11.35), (11.21) and $\dot{\Theta} = -\Psi\omega$. We can recover $\dot{p}$ using the estimates $\{R, v\}$ through $\dot{p} = -Rv$. The closed-loop system configuration is depicted in Figure 11.9. For the estimate of T_M one needs $\ddot{p}$ as in equation (11.26), which can be measured by accelerometers that give $a = R^T\ddot{p}$

11.4.2 Simulation Results for the Closed-Loop System

We present the simulation results of the proposed vision based landing scheme. In these simulations, we apply the proposed controller for the full dynamic model of the UAV. We add Gaussian noise of standard deviation (in pixel units) to the correspondences and image velocities, and perform the discrete and continuous motion estimation algorithms based on the noisy data. In Figures 11.10 and 11.11, we present the simulation results for image measurement noise levels of 1 and 4 pixels standard deviation in both the image correspondences and the image velocities.

In these simulations, the initial position is $p = [2, 1, 5]^T$ meters away from the desired landing configuration above the landing pad (the origin), the initial orientation is $[\theta, \phi, \psi]^T = [0, 0, 0.4]^T$ radians. The dominant poles of the outer loop controller are placed at $-2, -.45$. The inner loop attitude controller is designed based on feedback linearization [56], and it has the form $\Theta^{(3)} = V_\Theta$, where V_Θ is designed as three decoupled pole-placement controllers with poles located at -10 and $-7 \pm 7.1414i$ for each controller. The main rotor thrust is controlled based on dynamic inversion and the pole is placed at -5.

Since the origin of the closed-loop system is exponentially stable, it is robust to relatively large levels of noise. As we see, the controller performs very well at a noise level of 1 pixel standard deviation, which is the accuracy of most state-of-the-art feature-tracking techniques [3], and remains stable at a large noise level of 4 pixel standard deviation. Due to the gain margin in the controller, the closed loop system is also robust to possible modeling errors which are omitted, such as the camera calibration.

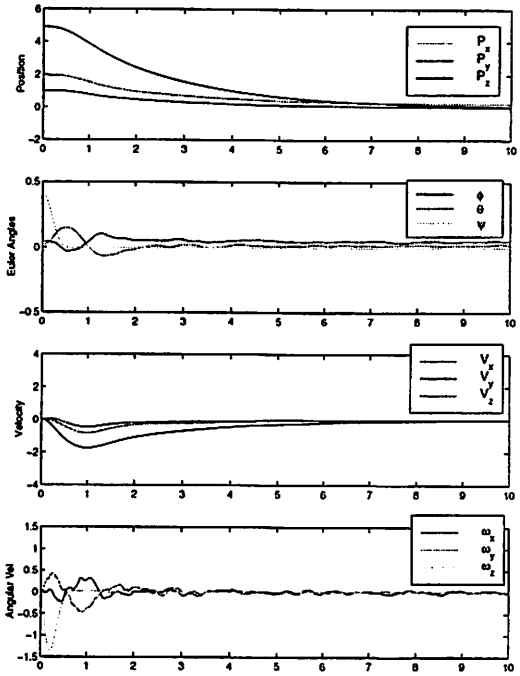


Figure 11.10: Closed-loop system simulation with 1 pixel noise.

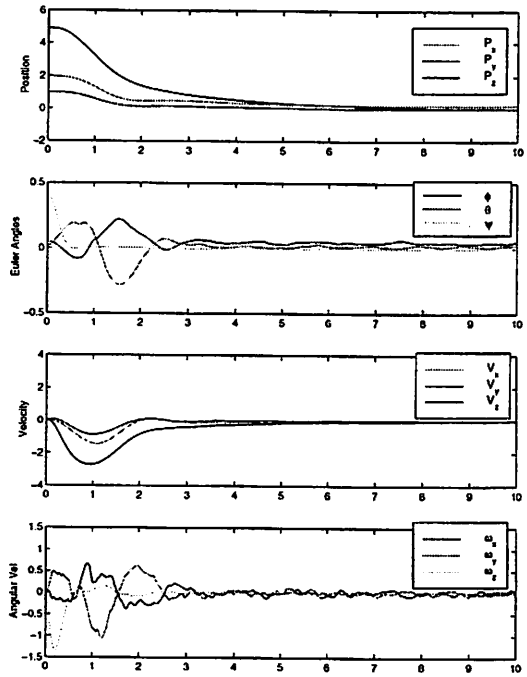


Figure 11.11: Closed-loop system simulation with 4 pixel noise.

11.5 Discussion

In this chapter have we studied the problem of using computer vision as a feedback sensor to control the landing of a dynamic Unmanned Aerial Vehicle. We derive a novel geometric algorithm for estimating the camera angular and linear velocity relative to a planar scene, and give a thorough performance evaluation. We propose a nonlinear controller based on differential flatness for a full UAV dynamic model, and give detailed conditions for stability of the overall closed loop system. Through extensive simulation, the vision based controller is shown to result in stable landing maneuvers for large noise levels.

We are currently implementing the above vision algorithms and controller on a model helicopter as part of the UC BErkeley Aerial Robot (BEAR) project. One of our UAVs is a Yamaha R-50 model helicopter, on which we have mounted computers, INS, GPS, and a vision system, consisting of a camera, a real-time feature tracker board, and a Pentium II running Linux. Figure 11.12 shows one of the UC Berkeley UAVs on which we will implement the proposed landing scheme.

Figure 11.12: A member of UC Berkeley UAV fleet: a Yamaha R-50 helicopter.

Chapter 12

Conclusions

"Supposing truth is a woman – what then?"
— Friedrich Nietzsche, *Beyond Good and Evil*

"The essential political problem for the intellectual is not to criticize the ideological contents supposedly linked to science, or to ensure that his own scientific practice is accompanied by a correct ideology, but that of ascertaining the possibility of constituting a new politics of truth... Hence the importance of Nietzsche."
— Michel Foucault, *Truth and Power*

As its title suggests, this dissertation attempts to make a connection among three relatively independent research disciplines: Computer Vision, Differential Geometry, and (Robotic) Control. Such an interdisciplinary study is probably just as promising as it is risky. It certainly produces tremendous opportunities with new perspectives, new methods and new problems; however, the effort might be easily under appreciated by either of the above disciplines. Maybe because of this, ever since I decided to explore this rocky road as my PhD program, every once a while, there have been warm-hearted people warning me of the hardship I would expect down the road. At those occasions, I just have to take the warnings as encouragement for me to try extra harder. Due to the ever growing practice in vision based robotic control, a unified study of both computer vision and control is simply inevitable, neither can it wait any longer. Had there not been us, someone would have done the same work already.

This dissertation summarizes the main work that I have been doing for the past four years on the subjects of computer vision and vision based control. When the time comes for me to put all the related papers together (to make this dissertation), to my

surprise, all the pieces of the jigsaw puzzle seem to fit with each other very well. The six chapters of Part I consist of a rather coherent theory of the classic structure from motion problem from an entirely new perspective. In addition to presenting new results, we take a lot of effort to clarify some misunderstandings among the literature. Undoubtly, most of the results will directly benefit the computer vision community. At the mean time, this new perspective opens the door to a unified study of multiview geometry in both Euclidean and non-Euclidean spaces. We pursue this quest in Chapter 8 of Part II, where we have laid out basic ingredients for the study of multiview geometry in more general classes of spaces or Riemannian manifolds. Many new and interesting problems are therefore raised regarding how to study geometric properties of certain spaces from a vision point of view. While these questions mostly attract mathematicians, especially differential geometers, it is the improved understanding in multiview geometry that benefits control theorists the most. Therefore in Part III of this dissertation, we shift the focus from vision to control and demonstrate how to design vision based control systems. The two examples presented are both representative applications of vision in robotic control: vision guided driving of ground mobile vehicles and vision guided landing of aerial mobile vehicles.

It would be very hard to picture any next generation intelligent robots without any on-board visual sensors. In fact, the level of intelligence and automation of the future robots will be very much determined by how well the on-board computer processes information collected from the visual sensors. However, despite that we seem to know quite a lot about vision already, especially it as an information processing entity, state of the art computer vision systems still have no match for the human vision, not even close. This can only mean one thing: A large part about the nature of vision is yet unknown to us. While mathematics allow us to study fundamental geometric principles underlying visual perception as this dissertation has shown, a full understanding of the phenomena of vision must however rely on a more interdisciplinary effort from many other disciplines such as neurobiology, psychophysics, computer science, and cognitive science.

Appendix A

Geometric Optimization on Manifolds

A.1 Optimization on Riemannian Manifold Preliminaries

Newton's and conjugate gradient methods are classical nonlinear optimization techniques to minimize a function $f(x)$, where x belongs to an open subset of Euclidean space $\mathbb{R}^n$. Recent developments in optimization algorithms on Riemannian manifolds have provided geometric insights for generalizing Newton's and conjugate gradient methods to certain classes of Riemannian manifolds. Smith [97] gave a detailed treatment of a theory of optimization on general Riemannian manifolds; Edelman, Arias and Smith [19] further studied the case of Stiefel and Grassmann manifolds,[1] and presented a unified geometric framework for applying Newton and conjugate gradient algorithms on these manifolds. These new mathematical schemes solve the more general optimization problem of minimizing a function $f(x)$, where x belongs to some Riemannian manifold (M, Φ), where $\Phi : TM \times TM \to C^\infty(M)$ is the Riemannian metric on M (and TM denotes the tangent space of M). An intuitive comparison between the Euclidean and Riemannian nonlinear optimization schemes is illustrated in Figure A.1.

Conventional approaches for solving such an optimization problem are usually application dependent. The manifold M is first embedded as a submanifold into a higher

[1]Stiefel manifold $V(n,k)$ is the set of all orthonormal k-frames in $\mathbb{R}^n$; Grassmann manifold $G(n,k)$ is the set of all k dimensional subspaces in $\mathbb{R}^n$. Then canonically, $V(n,k) = O(n)/O(n-k)$ and $G(n,k) = O(n)/O(k) \times O(n-k)$ where $O(n)$ is the orthogonal group of $\mathbb{R}^n$.

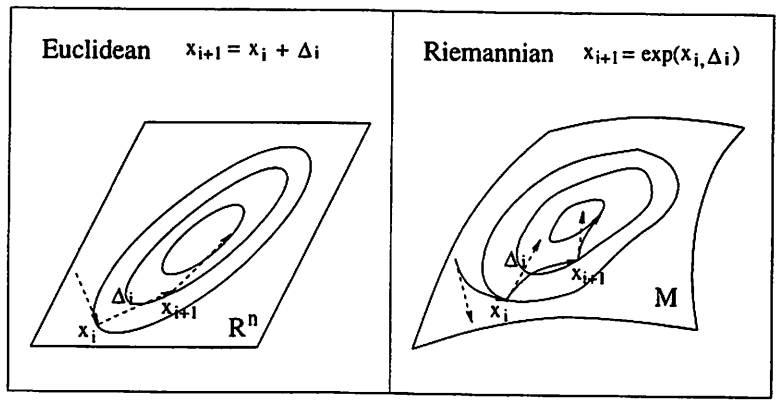


Figure A.1: Comparison between the Euclidean and Riemannian nonlinear optimization schemes. At each step, an (optimal) updating vector $\Delta_i \in T_{x_i}M$ is computed using the Riemannian metric at x_i. Then the state variable is updated by following the geodesic from x_i in the direction Δ_i by a distance of $\sqrt{g(\Delta_i, \Delta_i)}$ (the Riemannian norm of Δ_i). This geodesic is usually denoted in Riemannian geometry by the exponential map $\exp(x_i, \Delta_i)$.

dimensional Euclidean space $\mathbb{R}^N$ by choosing certain (global or local) *parameterization* of M. *Lagrangian multipliers* are often used to incorporate additional constraints that these parameters should satisfy. In order for x to always stay on the manifold, after each update, it needs to be *projected* back onto the manifold M. However, the new analysis of [19] shows that, for "nice" manifolds, *i.e.*, for example Lie groups or homogeneous spaces such as Stiefel and Grassmann manifolds, one can make use of the *canonical* Riemannian structure of these manifolds and systematically develop a Riemannian version of the Newton's algorithm or conjugate gradient methods for optimizing a function defined on them. Since the parameterization and metrics are canonical and the state is updated using geodesics (therefore always staying on the manifold), the performance of so obtained algorithms is no longer parameterization dependent, and in addition they typically have polynomial complexity and super-linear (quadratic) rate of convergence [97]. An intuitive comparison between the conventional update-then-project approach and the Riemannian method is demonstrated in Figure A.2 (where M is illustrated as the standard 2D sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 \mid \|x\|^2 = 1\}$).

One of the purposes of this paper is to apply these new Riemannian optimization schemes to solve the nonlinear optimization problem of recovering 3D motion from image correspondences. As we will soon see the underlying Riemannian manifold for this problem (the so called essential manifold) is a product of Stiefel manifolds instead of a single one. We

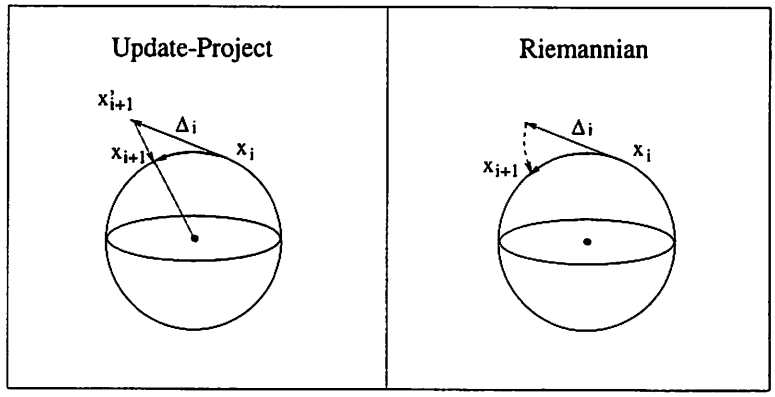


Figure A.2: Comparison between the conventional update-then-project approach and the Riemannian scheme. For the conventional method, the state x_i is first updated to x'_{i+1} according to the updating vector Δ_i and then x'_{i+1} is projected back to the manifold at x_{i+1}. For the Riemannian scheme, the new state x_{i+1} is obtained by following the geodesic, *i.e.*, $x_{i+1} = \exp(x_i, \Delta_i)$.

first need to generalize Edelman *et al*'s methods [19] to the product of Stiefel (or Grassmann) manifolds. Suppose (M_1, Φ_1) and (M_2, Φ_2) are two Riemannian manifolds with Riemannian metrics:

$$\Phi_1(\cdot,\cdot) : TM_1 \times TM_1 \to C^\infty(M_1),$$
$$\Phi_2(\cdot,\cdot) : TM_2 \times TM_2 \to C^\infty(M_2)$$

where TM_1 is the tangent bundle of M_1, similarly for TM_2. The corresponding Levi-Civita connections (*i.e.*, the unique metric preserving and torsion-free connection) of these manifolds are denoted as:

$$\nabla_1 : \mathcal{X}(M_1) \times \mathcal{X}(M_1) \to \mathcal{X}(M_1),$$
$$\nabla_2 : \mathcal{X}(M_2) \times \mathcal{X}(M_2) \to \mathcal{X}(M_2)$$

where $\mathcal{X}(M_1)$ stands for the space of smooth vector fields on M_1, similarly for $\mathcal{X}(M_2)$.

Now let M be the product space of M_1 and M_2, *i.e.*, $M = M_1 \times M_2$. Let $i_1 : M_1 \to M$ and $i_2 : M_2 \to M$ be the natural inclusions and $\pi_1 : M \to M_1$ and $\pi_2 : M \to M_2$ be the projections. To simplify the notation, we identify TM_1 and TM_2 with $i_{1*}(TM_1)$ and $i_{2*}(TM_2)$ respectively. Then $TM = TM_1 \times TM_2$ and $\mathcal{X}(M) = \mathcal{X}(M_1) \times \mathcal{X}(M_2)$. For any vector field $X \in \mathcal{X}(M)$ we can write X as the composition of its components in the two subspaces TM_1 and TM_2: $X = (X_1, X_2) \in TM_1 \times TM_2$. The canonical Riemannian metric

$\Phi(\cdot,\cdot)$ on M is determined as:

$$\Phi(X,Y) = \Phi_1(X_1,Y_1) + \Phi_2(X_2,Y_2), \quad X,Y \in \mathcal{X}(M).$$

Define a connection ∇ on M as:

$$\nabla_X Y = (\nabla_{1X_1} Y_1, \nabla_{2X_2} Y_2) \in \mathcal{X}(M_1) \times \mathcal{X}(M_2), \quad X,Y \in \mathcal{X}(M).$$

One can directly check that this connection is torsion free and compatible with the canonical Riemannian metric Φ on M (*i.e.*, preserving the metric) hence it is the Levi-Civita connection for the product Riemannian manifold (M,Φ). From the construction of ∇, it is also canonical.

According to Edelman *et al* [19], in order to apply Newton's or conjugate gradient methods on a Riemannian manifold, one needs to know how to explicitly calculate parallel transport of vectors on the manifolds and an explicit expression for geodesics. The reason that Edelman *et al*'s methods can be easily generalized to any product of Stiefel (or Grassmann) manifolds is because there are simple relations between the parallel transports on the product manifold and its factor manifolds. The following theorem follows directly from the above discussion of the Levi-Civita connection on the product manifold.

Theorem A.1. *Consider $M = M_1 \times M_2$ the product Riemannian manifold of M_1 and M_2. Then for two vector fields $X,Y \in \mathcal{X}(M)$, Y is parallel along X if and only if Y_1 is parallel along X_1 and Y_2 is parallel along X_2.*

As a corollary to this theorem, the geodesics in the product manifold are just the products of geodesics in the two factor manifolds. Consequently, the calculation of parallel transport and geodesics in the product space can be reduced to those in each factor manifold.

A.2 Riemannian Structure of the Essential Manifold

In this section we study the Riemannian structure of the essential manifold, which plays an important role in motion recovery from image correspondences (for details see [67]). Recall that, for any vector $u = (u_1,u_2,u_3)^T \in \mathbb{R}^3$, the notation $\widehat{u}$ means the associated

skew-symmetric matrix:

$$\widehat{u} = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \in \mathbb{R}^{3\times 3}.$$

Then for any two vectors $u, v \in \mathbb{R}^3$, the cross product $u \times v$ is equal to $\widehat{u}v$.

Camera motion is modeled as rigid body motion in $\mathbb{R}^3$. The displacement of the camera belongs to the special Euclidean group $SE(3)$:

$$SE(3) = \{(R,T) : R \in SO(3), T \in \mathbb{R}^3\} \tag{A.1}$$

where $SO(3) \in \mathbb{R}^{3\times 3}$ is the space of rotation matrices (orthogonal matrices with determinant $+1$). An element $g = (R,T)$ in this group is used to represent the coordinate transformation of a point in $\mathbb{E}^3$. We already know that two corresponding images $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same point $p \in \mathbb{E}^3$ satisfy the so called *epipolar constraint*:

$$\mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 = 0. \tag{A.2}$$

A good property of this constraint is that it decouples the problem of motion recovery from that of structure recovery. The matrix $\widehat{T}R$ in the epipolar constraint is the so called *essential matrix*, and the *essential manifold* is defined to be the space of all such matrices, denoted by:

$$\mathcal{E} = \{\widehat{T}R \mid R \in SO(3), \widehat{T} \in so(3)\}.$$

$SO(3)$ is a Lie group of 3×3 rotation matrices, and $so(3)$ is the Lie algebra of $SO(3)$, *i.e.*, the tangent plane of $SO(3)$ at the identity. $so(3)$ then consists of all 3×3 skew-symmetric matrices. As we have seen in Chapter 4, for the problem of recovering camera motion (R, S) from image correspondences, the associated objective functions are usually functions of the epipolar constraint. Hence they are of the form $f(E) \in \mathbb{R}$ with $E \in \mathcal{E}$. Moreover such functions in general are homogeneous in E. Thus the problem of motion recovery is equivalent to optimize functions defined on the so called *normalized essential manifold*:

$$\mathcal{E}_1 = \{\widehat{T}R \mid R \in SO(3), \widehat{T} \in so(3), \frac{1}{2} tr(\widehat{T}\widehat{T}^T) = 1\}.$$

Note that $\frac{1}{2} tr(\widehat{T}\widehat{T}^T) = T^T T$. Strictly speaking, the essential manifold $\mathcal{E}$ is not a differential manifold because of the singularity at $T = 0$.[2] On the other hand, the normalized essential

[2]It is, however, shown to be an algebraic variety [76].

manifold $\mathcal{E}_1$ is indeed a differential manifold which has a natural Riemannian structure, as we will soon see.

In order to study the optimization problem on $\mathcal{E}_1$, it is crucial to understand its Riemannian structure. We start with the Riemannian structure on the tangent bundle of the Lie group $SO(3)$, *i.e.*, $T(SO(3))$. The tangent space of $SO(3)$ at the identity e is simply its Lie algebra $so(3)$:

$$T_e(SO(3)) = so(3).$$

Since $SO(3)$ is a compact Lie group, it has an intrinsic bi-invariant metric [5] (such metric is unique up to a constant scale). In matrix form, this metric is given explicitly by:

$$\Phi_0(\widehat{T}_1, \widehat{T}_2) = \frac{1}{2} tr(\widehat{T}_1 \widehat{T}_2^T), \quad \widehat{T}_1, \widehat{T}_2 \in so(3).$$

Notice that this metric is induced from the Euclidean metric on $SO(3)$ as a Stiefel submanifold embedded in $\mathbb{R}^{3\times 3}$. For any $R \in SO(3)$ we define $\theta_R : SO(3) \times R \to SO(3)$ to be the *right action* of R on $SO(3)$, *i.e.*, $\theta_R(R_1) = R_1 R$ for all $R_1 \in SO(3)$. The tangent space at any other point $R \in SO(3)$ is then given by the push-forward map θ_{R*}:

$$T_R(SO(3)) = \theta_{R*}(so(3)) = \{\widehat{T}R \mid \widehat{T} \in so(3)\}.$$

Thus the tangent bundle of $SO(3)$ is:

$$T(SO(3)) = \bigcup_{R \in SO(3)} T_R(SO(3))$$

Since the tangent bundle of a Lie group is trivial [103], $T(SO(3))$ is then equivalent to the product $SO(3) \times so(3)$. $T(SO(3))$ can then be expressed as:

$$T(SO(3)) = \{(R, \widehat{T}R) \mid R \in SO(3), \widehat{T} \in so(3)\} \cong SO(3) \times so(3).$$

If we identify the tangent space of $so(3)$ with itself, then the metric Φ_0 of $SO(3)$ induces a canonical metric on the tangent bundle $T(SO(3))$:

$$\tilde{\Phi}(X, Y) = \Phi_0(X_1, X_2) + \Phi_0(Y_1, Y_2), \quad X, Y \in so(3) \times so(3).$$

Note that this metric restricted to the fiber $so(3)$ of $T(SO(3))$ is the same as the Euclidean metric if we identify $so(3)$ with $\mathbb{R}^3$. Such an induced metric on $T(SO(3))$ is invariant under the right action of $SO(3)$.

Then the metric $\tilde{\Phi}$ on the whole tangent bundle $T(SO(3))$ induces by restriction a canonical metric Φ on the unit tangent bundle of $T(SO(3))$:

$$T_1(SO(3)) \cong \{(R, \widehat{T}R) \mid R \in SO(3), \widehat{T} \in so(3), \frac{1}{2}tr(\widehat{T}\widehat{T}^T) = 1\}.$$

It is direct to check that, with the identification of $so(3)$ with $\mathbb{R}^3$, the unit tangent bundle is simply the product $SO(3) \times \mathbb{S}^2$ where $\mathbb{S}^2$ is the standard 2-sphere embedded in $\mathbb{R}^3$. According to Edelman *et al* [19], $SO(3)$ and $\mathbb{S}^2$ both are Stiefel manifolds $V(n,k)$ of the type $n = k = 3$ and $n = 3, k = 1$, respectively. As Stiefel manifolds, they both possess canonical metrics by viewing them as quotients between orthogonal groups. Here $SO(3) = O(3)/O(0)$ and $\mathbb{S}^2 = O(3)/O(2)$. Fortunately, for Stiefel manifolds of the special type $k = n$ or $k = 1$, the canonical metrics are the same as the Euclidean metrics induced as submanifold embedded in $\mathbb{R}^{n \times k}$. From the above discussion, we have

Theorem A.2. *The unit tangent bundle $T_1(SO(3))$ is equivalent to $SO(3) \times \mathbb{S}^2$. Its Riemannian metric Φ induced from the bi-invariant metric on $SO(3)$ as above is the same as that induced from the Euclidean metric with $T_1(SO(3))$ naturally embedded in $\mathbb{R}^{3\times 4}$ by the map $i : (R, \widehat{T}R) \mapsto (R, T)$. Further, $(T_1(SO(3)), \Phi)$ is the product Riemannian manifold of $(SO(3), \Phi_1)$ and $(\mathbb{S}^2, \Phi_2)$ with Φ_1 and Φ_2 canonical metrics for $SO(3)$ and $\mathbb{S}^2$ as Stiefel manifolds.*

However, the unit tangent bundle $T_1(SO(3))$ is not exactly the normalized essential manifold $\mathcal{E}_1$. Due to the equation (3.9), it is a double covering of the normalized essential manifold $\mathcal{E}_1$, *i.e.*, $\mathcal{E}_1 = T_1(SO(3))/\mathbb{Z}^2$. The natural covering map from $T_1(SO(3))$ to $\mathcal{E}_1$ is:

$$\begin{aligned} h : T_1(SO(3)) &\rightarrow \mathcal{E}_1 \\ (R, \widehat{T}R) \in T_1(SO(3)) &\mapsto \widehat{T}R \in \mathcal{E}_1. \end{aligned}$$

The inverse of this map is given by:

$$h^{-1}(\widehat{T}R) = \left\{(R, \widehat{T}R),\ (e^{\widehat{T}\pi}R, -\widehat{T}e^{\widehat{T}\pi}R)\right\}.$$

Comment A.3. *As we know from Lemma 3.1, the two pairs of rotation and translation corresponding to the same normalized essential matrix $\widehat{T}R$ are (R, T) and $(e^{\widehat{T}\pi}R, -T)$. As pointed out by Weinstein, this double covering h is equivalent to identifying a left-invariant vector field on $SO(3)$ with the one obtained by flowing it along the corresponding geodesic by distance π, the so-called time-π map of the geodesic flow on $SO(3)$.*

If we take for $\mathcal{E}_1$ the Riemannian structure induced from the covering map h, the original optimization problem of optimizing $f(E)$ on $\mathcal{E}_1$ can be converted to optimizing $f(R,S)$ on $T_1(SO(3))$.[3] Generalizing Edelman *et al*'s methods to the product Riemannian manifolds, we may obtain intrinsic geometric Newton's or conjugate gradient algorithms for solving such an optimization problem. Due to Theorem A.2, we can simply choose the induced Euclidean metric on $T_1(SO(3))$ and explicitly give these intrinsic algorithms in terms of the matrix representation of $T_1(SO(3))$. Since this Euclidean metric is the same as the intrinsic metrics, the apparently extrinsic representation preserves all intrinsic geometric properties of the given optimization problem. In this sense, the algorithms we are about to develop for the motion recovery are different from other existing algorithms which make use of particular parameterizations of the underlying search manifold $T_1(SO(3))$.

A.3 Optimization on the Essential Manifold

Let $f(R,T)$ be a function defined on $T_1(SO(3)) \cong SO(3) \times \mathbb{S}^2$ with $R \in SO(3)$ represented by a 3×3 rotation matrix and $T \in \mathbb{S}^2$ a vector of unit length in $\mathbb{R}^3$. This section gives Newton's algorithm for optimizing a function defined on this manifold (please refer to [19] for the details of the Newton's or other conjugate gradient algorithms for general Stiefel or Grassmann manifolds).

In order to apply Newton's algorithm to a Riemannian manifold, we need to know how to compute three things: the *gradient*, the *Hessian* of a given function and the *geodesics* of the manifold. Since the metric of the manifold is no longer the standard Euclidean metric, the computation for these three needs to incorporate the new metric. In the following, we will give general formulae for the gradient and Hessian of a function defined on $SO(3) \times \mathbb{S}^2$ using results from [19]. In the next section, we will however give an alternative approach for directly computing these ingredients by using the explicit expression of geodesics on this manifold.

Let Φ_1 and Φ_2 be the canonical metrics for $SO(3)$ and $\mathbb{S}^2$ respectively and ∇_1 and ∇_2 be the corresponding Levi-Civita connections. Let Φ and ∇ be the induced Riemannian metric and connection on the product manifold $SO(3) \times \mathbb{S}^2$. The gradient of the function

[3] Although the topological structures of $\mathcal{E}_1$ and $T_1(SO(3))$ are different, the nonlinear optimization only relies on local Riemannian metric and this identification will not affect effectiveness of the search schemes.

$f(R,T)$ on $SO(3) \times \mathbb{S}^2$ is a vector field $G = \text{grad}(f)$ on $SO(3) \times \mathbb{S}^2$ such that:

$$df(Y) = \Phi(G,Y), \quad \text{for all vector fields } Y \text{ on } SO(3) \times \mathbb{S}^2.$$

Geometrically, so defined gradient G has the same meaning as in the standard Euclidean case, *i.e.*, G is the direction in which the function f increases the fastest. On $SO(3) \times \mathbb{S}^2$, it can be shown that the gradient is explicitly given as:

$$G = (f_R - Rf_R^T R,\ f_T - Tf_T^T T) \quad \in T_R(SO(3)) \times T_T(\mathbb{S}^2)$$

where $f_R \in \mathbb{R}^{3\times 3}$ is the matrix of partial derivatives of f with respect to elements of R and $f_T \in \mathbb{R}^3$ is the vector of partial derivatives of f with respect to the elements of T, *i.e.*,

$$(f_R)_{ij} = \frac{\partial f}{\partial R_{ij}}, \quad (f_T)_k = \frac{\partial f}{\partial T_k}, \quad 1 \le i,j,k \le 3.$$

Geometrically, the Hessian of a function is the second order approximation of the function at a given point. However, when computing the second order derivative, unlike the Euclidean case, one should take the *covariant derivative* with respect to the Riemannian metric Φ on the given manifold.[4] On $SO(3) \times \mathbb{S}^2$, for any $X = (X_1, X_2), Y = (Y_1, Y_2) \in T(SO(3)) \times T(\mathbb{S}^2)$, the Hessian of $f(R,S)$ is explicitly given by:

$$\begin{aligned} \text{Hess } f(X,Y) &= f_{RR}(X_1,Y_1) - tr f_R^T \Gamma_R(X_1,Y_1) \\ &+ f_{TT}(X_2,Y_2) - tr f_T^T \Gamma_T(X_2,Y_2) \\ &+ f_{RT}(X_1,Y_2) + f_{TR}(Y_1,X_2). \end{aligned}$$

where the Christoffel functions Γ_R for $SO(3)$ and Γ_T for $\mathbb{S}^2$ are:

$$\begin{aligned} \Gamma_R(X_1,Y_1) &= \frac{1}{2} R(X_1^T Y_1 + Y_1^T X_1), \\ \Gamma_T(X_2,Y_2) &= \frac{1}{2} T(X_2^T Y_2 + Y_2^T X_2) \end{aligned}$$

and the other terms are:

$$\begin{aligned} f_{RR}(X_1,Y_1) &= \sum_{ij,kl} \frac{\partial^2 f}{\partial R_{ij} \partial R_{kl}} (X_1)_{ij} (Y_1)_{kl}, \quad f_{TT}(X_2,Y_2) = \sum_{i,j} \frac{\partial^2 f}{\partial T_i \partial T_j} (X_2)_i (Y_2)_j, \\ f_{RT}(X_1,Y_2) &= \sum_{ij,k} \frac{\partial^2 f}{\partial R_{ij} \partial T_k} (X_1)_{ij} (Y_2)_k, \quad f_{TR}(Y_1,X_2) = \sum_{i,jk} \frac{\partial^2 f}{\partial T_i \partial R_{jk}} (Y_1)_i (X_2)_{jk} \end{aligned}$$

[4] It is a fact in Riemannian geometry that there is a unique metric preserving and torsion-free covariant derivative.

For Newton's algorithm, we need to find the *optimal updating* tangent vector Δ such that:

$$\text{Hess } f(\Delta, Y) = \Phi(-G, Y) \quad \text{for all tangent vectors } Y.$$

Δ is then well-defined and independent of the choice of local coordinate chart. In order to solve for Δ, first find the tangent vector $Z(\Delta) = (Z_1, Z_2) \in T_R(SO(3)) \times T_T(\mathbb{S}^2)$ (in terms of Δ) satisfying the linear equations:

$$\begin{aligned} f_{RR}(\Delta_1, Y_1) + f_{TR}(Y_1, \Delta_2) &= \Phi_1(Z_1, Y_1) \quad \text{for all tangent vectors } Y_1 \in T_R(SO(3)) \\ f_{TT}(\Delta_2, Y_2) + f_{RT}(\Delta_1, Y_2) &= \Phi_2(Z_2, Y_2) \quad \text{for all tangent vectors } Y_2 \in T_T(\mathbb{S}^2) \end{aligned}$$

From the expression of the gradient G, the vector $\Delta = (\Delta_1, \Delta_2)$ then satisfies the linear equations:

$$\begin{aligned} Z_1 - R\,\text{skew}(f_R^T \Delta_1) - \text{skew}(\Delta_1 f_R^T) R &= -(f_R - R f_R^T R) \\ Z_2 - f_T^T T \Delta_2 &= -(f_T - T f_T^T T) \end{aligned}$$

with $\Delta_1 R^T$ being skew-symmetric and $\Delta_2^T T = 0$. In the above expression, the notation skew(A) means the skew-symmetric part of the matrix A: skew$(A) = (A - A^T)/2$. For this system of linear equations to be solvable, the Hessian has to be non-degenerate, in other words the corresponding Hessian matrix in local coordinates is invertible. This non-degeneracy depends on the chosen objective function f.

According to Newton's algorithm, knowing Δ, the search state is then updated from (R, T) in direction Δ along geodesics to $(\exp(R, \Delta_1), \exp(T, \Delta_2))$, where $\exp(R, \cdot)$ stands for the exponential map from $T_R(SO(3))$ to $SO(3)$ at point R, similarly for $\exp(T, \cdot)$. Explicit expressions for the geodesics $\exp(R, \Delta_1 t)$ on $SO(3)$ and $\exp(T, \Delta_2 t)$ on $\mathbb{S}^2$ are given in Chapter 4. The overall algorithm can be summarized in the following:

Riemannian Newton's algorithm for minimizing $f(R, T)$ on the normalized essential manifold:

- *At the point* (R, T),
  - *Compute the gradient* $G = (f_R - R f_R^T R,\ f_T - T f_T^T T)$,
  - *Compute the updating vector* $\Delta = -\ Hess^{-1} G$.

- *Move* (R, T) *in the direction* Δ *along geodesic to* $(\exp(R, \Delta_1), \exp(T, \Delta_2))$.
- *Repeat if* $\|G\| \geq \epsilon$ *for pre-determined* $\epsilon > 0$.

Since the manifold $SO(3) \times \mathbb{S}^2$ is compact, this algorithm is guaranteed to converge to a (local) extremum of the objective function $f(R, T)$. Note that this algorithm works for any objective function defined on $SO(3) \times \mathbb{S}^2$. For an objective function with non-degenerate Hessian, the Riemannian Newton's algorithm has quadratic (super-linear) rate of convergence [97].

Appendix B

UAV System Parameters

All variables except for the state variables and inputs are numeric constants, which can be obtained by measurements and experiments. The followings are the values of the constants:

I_x	=	0.142413	I_y	=	0.271256	I_z	=	0.271492
l_M	=	-0.015	y_M	=	0	h_M	=	0.2943
h_T	=	0.1154	l_T	=	0.8715	m	=	4.9
C_M^Q	=	0.004452	D_M^Q	=	0.6304	$\frac{\partial R_M}{\partial b_{1s}}$	=	25.23
C_T^Q	=	0.005066	D_T^Q	=	0.008488	$\frac{\partial M_M}{\partial a_{1s}}$	=	25.23
c_{M1}	=	6.4578	c_{M3}	=	100.3752	c_{T1}	=	0.1837
c_{T3}	=	0.1545						

The operation regions in radian for a_{1s}, b_{1s} and newton for T_M, T_T are: $|a_{1s}| \leq 0.4363$, $|b_{1s}| \leq 0.3491$, $-20.86 \leq T_M \leq 69.48$, $-5.26 \leq T_T \leq 5.26$.

Bibliography

[1] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–89, 1989.

[2] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1034 – 1040, 1997.

[3] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. Journal on Computer Vision*, 12(1):43–77, 1994.

[4] P. Beardsley and A. Zisserman. Affine calibration of mobile vehicles. In *Europe-China Workshop on Geometric Modeling and Invariants for Computer Vision*, 1995.

[5] W. M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry.* Academic Press, second edition, 1986.

[6] B. Boufama, R. Mohr, and F. Veillon. Euclidean constraints for uncalibrated reconstruction. In *ICCV*, pages 466–470, Berlin, Germany, 1993.

[7] S. Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 790–796, 1998.

[8] R. W. Brockett, R. S. Millman, and H. J. Sussmann. *Differential Geometric Control Theory.* Boston: Birkhauser, 1983.

[9] M. J. Brooks, W. Chojnacki, and L. Baumela. Determining the ego-motion of an uncalibrated camera from instantaneous optical flow. *in press*, 1997.

[10] A. R. Bruss and B. K. Horn. Passive navigation. *Computer Graphics and Image Processing*, 21:3–20, 1983.

[11] F. M. Callier and C. A. Desoer. *Linear System Theory.* Springer Texts in Electrical Engineering. Springer-Verlag, 1991.

[12] S. Carlsson. Multiple image invariance using the double algebra. In *Applications of invariance in computer vision*, 1994.

[13] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on PAMI*, 18(11):1098–1104, 1996.

[14] K. Danillidis. *Visual Navigation.* Lawrence Erlbaum Associates, 1997.

[15] K. Danillidis and H.-H. Nagel. Analytical results on error sensitivity of motion estimation from two views. *Image and Vision Computing*, 8:297–303, 1990.

[16] E. D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):241–261, 1988.

[17] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):223–240, 1988.

[18] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-D road and relative ego-state estimation. *IEEE Transactions on PAMI*, 14(2):199–213, February 1992.

[19] A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applications*, to appear.

[20] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[21] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV*, pages 563–578. Springer-Verlag, 1992.

[22] O. Faugeras. *There-Dimensional Computer Vision.* The MIT Press, 1993.

[23] O. Faugeras. Stratification of three-dimensional vision: projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3):465–84, 1995.

[24] O. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *Proceeding of IEEE First International Conference on Computer Vision*, pages 25–34, London, England, 1987. IEEE Comput. Soc. Press.

[25] O. Faugeras and T. Papadopoulo. Grassmann-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. In *Proceeding of the IEEE workshop of representation of visual scenes*, 1995.

[26] O. D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[27] M. Fliess. Aircraft control using flatness. In *Proceedings of Symposium on Control, Optimization and Supervision*, pages 194–9, Lille, France, July 1996.

[28] M. Fliess, J. Lévine, Ph. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and applications. *Int. Journal of Control*, 61(6):1327–1361, 1995.

[29] R. Frezza and G. Picci. On line path following by recursive spline updating. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 4, pages 4047–4052, 1995.

[30] B. K. Ghosh and E. P. Loucks. A perspective theory for motion and shape estimation in machine vision. *SIAM Journal on Control and Optimization*, 33(5):1530–1559, Sept. 1995.

[31] R. Goodman and N. R. Wallach. *Representations and Invariants of the Classical Groups*. Cambridge University Press, 1998.

[32] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume I and II. Addison-Wesley Publishing Company, 1993.

[33] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–57, 1997.

[34] R. I. Hartley. Lines and points in three views - a unified approach. In *Proceeding of 1994 Image Understanding Workshop*, pages 1006–1016, Monterey, CA USA, 1994. OMNIPRESS.

[35] R. I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):133–135, February 1997.

[36] R. I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.

[37] R. I. Hartley. Chirality. *International Journal of Computer Vision*, 26(1):41–61, 1998.

[38] R. I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proceeding of Conference on Computer Vision and Pattern Recognition*, pages 761–4, Urbana-Champaign, IL, USA, 1992. IEEE Comput. Soc. Press.

[39] J. Hauser and R. Hindman. Maneuver regulation from trajectory tracking: Feedback linearization systems. *A Post-print Volume from the 3rd IFAC Symposium Proceedings of IFAC Symposium on Nonlinear Control Systems Design*, 2:595–600, 1996.

[40] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[41] J. P. Hespanha and A. S. Morse. Personal communication, May 1998.

[42] A. Heyden. Reduced multilinear constraints – theory and experiments. *International Journal of Computer Vision*, 30(2):5–26, 1998.

[43] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in Applied Sciences*, 20(13):1135–62, 1997.

[44] A. Heyden, G. Sparr, and K. Åström. Perception and action using multilinear forms. *Algebraic Frames for the Perception-Action Cycle*, pages 54–65, 1997.

[45] B. Horn. Relative orientation. *International Journal of Computer Vision*, 4:59–78, 1990.

[46] W.-Y. Hsiang. Absolute geometry revisited, Center for Pure and Applied Mathematics, University of California at Berkeley. *PAM-628*, 1995.

[47] T. Huang and O. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE PAMI*, 11(12):1310–12, 1989.

[48] A. Isidori. *Nonlinear Control Systems.* Communications and Control Engineering Series. Springer-Verlag, second edition, 1989.

[49] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* NY: Academic Press, 1970.

[50] A. D. Jepson and D. J. Heeger. Linear subspace methods for recovering translation direction. *Spatial Vision in Humans and Robots, Cambridge Univ. Press*, pages 39–62, 1993.

[51] K. Kanatani. Detecting the motion of a planar surface by line & surface integrals. In *Computer Vision, Graphics, and Image Processing*, volume 29, pages 13–22, 1985.

[52] K. Kanatani. 3d interpretation of optical flow by renormalization. *International Journal of Computer Vision*, 11(3):267–282, 1993.

[53] K. Kanatani. *Geometric Computation for Machine Vision.* Oxford Science Publications, 1993.

[54] H. K. Khalil. *Nonlinear Systems.* Prentice-Hall, 2nd edition, 1996.

[55] S. Kobayashi and T. Nomizu. *Foundations of Differential Geometry: Volume I and Volume II.* John Wiley & Sons, Inc., 1996.

[56] T. J. Koo and S. Sastry. Output tracking control design of a helicopter model based on approximate linearization. In *Proceedings of the 37th Conference on Decision and Control*, pages 3635–40, Tampa, Florida, December 1998.

[57] J. Košecká, R. Blasi, C. J. Taylor, and J. Malik. Vision-based lateral control of vehicles. In *Proc. Intelligent Transportation Systems Conference*, Boston, 1997.

[58] E. Kruppa. Zur ermittlung eines objecktes aus zwei perspektiven mit innerer orientierung. *Sitz.-Ber.Akad.Wiss., Math.Naturw., Kl.Abt.IIa, 122:1939-1948*, 1913.

[59] E. H. Lee, H. Shim, H. Park, and K. I. Lee. Design of hovering attitude controller for a model helicopter. In *Proceedings of Society of Instrument and Control Engineers*, pages 1385–1389, Tokyo, Japan, August 1993.

[60] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[61] H. C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. In *Proceedings of Royal Society of London*, volume 227 of *B*, pages 399–410, 1986.

[62] Q.-T. Luong and O. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *IJCV*, 1994. toappear.

[63] Q.-T. Luong and O. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996.

[64] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *IJCV*, 22(3):261–89, 1997.

[65] Q.-T. Luong and T. Vieville. Canonical representations for the geometries of multiple projective views. *ECCV*, pages 589–599, 1994.

[66] Y. Ma, J. Košecká, and S. Sastry. A mathematical theory of camera self-calibration. *Electronic Research Laboratory Memorandum, UC Berkeley*, UCB/ERL(M98/64), June 1998.

[67] Y. Ma, J. Košecká, and S. Sastry. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. In *Proceeding of European Conference on Computer Vision, Volume II*, pages 337–53, 1998.

[68] Y. Ma, J. Košecká, and S. Sastry. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. *Electronic Research Laboratory Memorandum, UC Berkeley*, UCB/ERL M98/11, June 1998.

[69] Y. Ma, J. Košecká, and S. Sastry. Optimal motion from image sequences: A Riemannian viewpoint. *Electronic Research Laboratory Memorandum, UC Berkeley*, UCB/ERL(M98/37), June 1998.

[70] Y. Ma, J. Košecká, and S. Sastry. Vision guided navigation of a nonholonomic mobile robot. *UC Berkeley Memorandum*, UCB/ERL(M97/34), 1997.

[71] Y. Ma, J. Košecká, and S. Sastry. Euclidean structure and motion from image sequences. *UC Berkeley Memorandum No. UCB/ERL M98/38*, 1998.

[72] Y. Ma, J. Košecká, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *submitted to International Journal of Computer Vision*, 1999.

[73] Y. Ma, J. Košecká, and S. Sastry. Optimization criteria, sensitivity and robustness of motion and structure estimation. In *Proceedings of ICCV workshop on Vision Theory and Algorithm, to appear*, Corfu, Greece, 1999.

[74] Y. Ma, S. Soatto, J. Košecká, and S. Sastry. Euclidean reconstruction and reprojection up to subgroups. In *Proceedings of 7th ICCV*, pages 773–80, Corfu, Greece, 1999.

[75] D. Marr. *Vision: a computational investigation into the human representation and processing of visual information.* W.H. Freeman and Company, 1982.

[76] S. Maybank. *Theory of Reconstruction from Image Motion.* Springer Series in Information Sciences. Springer-Verlag, 1993.

[77] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[78] S. Maybank and A. Shashua. Ambiguity in reconstruction from images of six points. In *ICCV*, pages 703–8, Bombay, India, 1988.

[79] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of IEEE fifth International Conference on Computer Vision*, pages 314–20, Cambridge, MA USA, 1995. IEEE Com. Soc. Press.

[80] J. M. Mendel. *Lessons in Digital Estimation Theory.* Prentice-Hall Signal Processing Series. Prentice-Hall, first edition, 1987.

[81] T. Moon, L. Van Gool, M. Van Dients, and E. Pauwels. Affine reconstruction from perspective pairs obtained by a translating camera. In J. L. Mundy and A. Zisserman, editors, *Applications of Invariance in Computer Vision*, pages 297–316, 1993.

[82] M. Mühlich and R. Mester. The role of total least squares in motion analysis. In *Proceedings of European Conference on Computer Vision*, pages 305–321, 1998.

[83] R. M. Murray. Nilpotent bases for a class of non-integrable distributions with applications to trajectory generation for nonholonomic systems. *Technical Report CIT/CDS 92-002, California Institute of Technology*, October 1992.

[84] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC press Inc., 1994.

[85] R. M. Murray and S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, May 1993.

[86] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *In press*, 1999.

[87] R. Pissard-Gibollet and P. Rives. Applying visual servoing techniques to control a mobile hand-eye system. In *Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan*, volume 1, pages 166–171, May 1995.

[88] S. E. Plamer. *Vision Science: Photons to Phenomenology.* The MIT Press, 1999.

[89] J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: a new approach to weak calibration. *International Journal of Computer Vision*, 28(3):223–43, 1998.

[90] R. W. Prouty. *Helicopter Performance, Stability, and Control.* Krieger Publishing Co., Inc., Boston, USA, 1995.

[91] D. Raviv and M. Herman. A "non-reconstruction" approach for road following. In Proceedings of the SPIE, editor, *Intelligent Robots and Computer Vision*, volume 1608, pages 2–12, 1992.

[92] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: The Task Function Approach.* Oxford Engineering Science Series. Claderon Press, 1991.

[93] S. S. Sastry. *Nonlinear Systems: Analysis, Stability and Control.* Springer-Verlag, 1999.

[94] F. R. Schell and E. D. Dickmanns. Autonomous landing of airplanes by dynamic machine vision. *Machine Vision and Applications*, 7:127–134, 1994.

[95] A. Shashua. Trilinearity in visual recognition by alignment. In *the Proceedings of ECCV, Volume I*, pages 479–484. Springer-Verlag, 1994.

[96] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 310–15. IEEE Computer Society, June 1991.

[97] S. T. Smith. Geometric optimization methods for adaptive filtering. *PhD thesis, Division of Applied Sciences, Harvard University*, May 1993.

[98] S. Soatto and R. Brockett. Optimal and suboptimal structure from motion. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[99] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996.

[100] S. Soatto, R. Frezza, and P. Perona. Visual navigation by controlling apparent shape. *UC Berkeley AI/Robotics/Vision Seminar Notes*, October 1996.

[101] S. Soatto and P. Perona. Recursive estimation of camera motion from uncalibrated image sequences. In *Proceedings ICIP-94*, volume 3, pages 58–62, Nov. 1994.

[102] M. Spetsakis. Models of statistical visual motion estimation. *CVIPG: Image Understanding*, 60(3):300–312, November 1994.

[103] M. Spivak. *A Comprehensive Introduction to Differential Geometry: Volume II.* Publish or Perish, Inc., second edition, 1979.

[104] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, pages 1100–1105. IEEE Comput. Soc. Press, 1997.

[105] M. Subbarao and A. M. Waxman. On the uniqueness of image flow solutions for planar surfaces in motion. *Third IEEE workshop on computer vision: representation and control*, pages 129–140, 1985.

[106] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least square. *Carnegie Mellon Research Report Series*, 1993.

[107] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on PAMI*, 17(11):1021–32, 1995.

[108] I. Thomas and E. Simoncelli. Linear structure from motion. Ms-cis-94-61, Grasp Laboratory, University of Pennsylvania, 1995.

[109] T. Y. Tian, C. Tomasi, and D. J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 315–20, Los Alamitos, CA, USA, 1996. IEEE Comput. Soc. Press.

[110] D. Tilbury, R. Murray, and S. Sastry. Trajectory generation for the n-trailer problem using goursat normal form. *IEEE Transactions on Automatic Control*, 40(5):802–819, May 1995.

[111] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *Intl. Journal of Computer Vision*, 9(2):137–154, 1992.

[112] G. Toscani and O. D. Faugeras. Structure and motion from two noisy perspective images. *Proceedings of IEEE Conference on Robotics and Automation*, pages 221–227, 1986.

[113] B. Triggs. Matching constraints and the joint image. In *Proceeding of Fifth International Conference on Computer Vision*, pages 338–43, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.

[114] B. Triggs. Factorization methods for projective structure and motion. In *Proceeding of 1996 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 845–51, San Francisco, CA, USA, 1996. IEEE Comput. Soc. Press.

[115] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 1997.

[116] B. Triggs. Autocalibration from planar scenes. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 1998.

[117] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. *International Journal of Computer Vision*, to appear.

[118] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proceedings, CVPR '86 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, June 22–26, 1986)*, IEEE Publ.86CH2290-5, pages 364–374. IEEE, 1986.

[119] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[120] D. Tsakiris, C. Samson, and P. Rives. Vision-based time-varying mobile robot control. In *Proceedings of the Research Workshop of ERNET*, pages 163–72, 1996.

[121] M. J. van Nieuwstadt and R. M. Murray. Outer flatness: Trajectory generation for a model helicopter. In *Proceedings of European Control Conference*, Brussels, Belgium, 1997.

[122] T. Vieville and O. D. Faugeras. Motion analysis with a camera with unknown, and possibly varying intrinsic parameters. *Proceedings of Fifth International Conference on Computer Vision*, pages 750–756, June 1995.

[123] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216–222, January 1994.

[124] A. Waxman and S. Ullman. Surface structure and three-dimensional motion from image flow kinematics. *Int. J. Robotics Research*, 4(3):72–94, 1985.

[125] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao. Closed form solutions to image flow equations for 3D structure and motion. *International Journal of Computer Vision 1*, pages 239–258, 1987.

[126] J. Weber, D. Koller, Q.-T. Luong, and J. Malik. An integrated stereo-based approach to automatic vehicle guidance. In *Proceedings of IEEE International Conference on Computer Vision*, pages 52–57, June 1995.

[127] A. Weinstein. Mathematics Department, UC Berkeley. *Personal communications*, 2000.

[128] Y. Weiss. Bayesian motion estimation and segmentation. *MIT PhD thesis*, 1998.

[129] J. Weng, N. Ahuja, and T. Huang. Optimal motion and structure estimation. *IEEE Transactions PAMI*, 9(2):137–154, 1993.

[130] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–475, 1989.

[131] J. Weng, T. S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences.* Springer Verlag, 1993.

[132] M. Werman and A. Shashua. The study of 3D-from-2D using elimination. In *Proc. of Europe-China Workshop on Geometrical Modeling and Invariants for Computer Vision*, pages 94–101, Xi'an, China, 1995.

[133] S. Werner, S. Furst, D. Dickmanns, and E. D. Dickmanns. A vision-based multi-sensor machine perception system for autonomous aircraft landing approach. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2736, pages 54–63, Orlando, FL, USA, 1996.

[134] H. Weyl. *The Classical Groups: Their Invariants and Representations.* Princeton University Press, 1946.

[135] J. A. Wolf. *Spaces of Constant Curvature.* Publish or Perish, Inc., 5th edition, 1984.

[136] ECCV Workshop. *3D structure from Multiple Images of Large-Scale Environments.* in connection with ECCV'98, 1998.

[137] Z. F. Yang and W. H. Tsai. Using parallel line information for vision-based landmark location estimation and an application to automatic helicopter landing. *Robotics and Computer-Integrated Manufacturing*, 14(4):297–306, 1998.

[138] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. *Research Report 2793, INRIA, France*, 1996.

[139] T. Zhang and C. Tomasi. Fast, robust and consistent camera motion estimation. In *to appear in Proceeding of CVPR*, 1999.

[140] Z. Zhang. Understanding the relationship between the optimization criteria in two-view motion analysis. In *Proceeding of International Conference on Computer Vision*, pages 772–77, Bombay, India, 1998.

[141] X. Zhuang and R. M. Haralick. Rigid body motion and optical flow image. *Proceedings of the First International Conference on Artificial Intelligence Applications*, pages 366–375, 1984.

[142] X. Zhuang, T. S. Huang, and N. Ahuja. A simplified linear optic flow-motion algorithm. *Computer Vision, Graphics and Image Processing*, 42:334–344, 1988.