THE UNIVERSITY
OF ADELAIDE
AUSTRALIA

Fundamental Numerical Schemes for Parameter Estimation in Computer Vision

by

Tony Scoleri

Thesis submitted for the degree of

Doctor of Philosophy

Faculty of Engineering, Computer and Mathematical Sciences

The University of Adelaide, Australia

2008

Abstract

An important research area in computer vision is parameter estimation. Given a mathematical model and a sample of image measurement data, key parameters are sought to encapsulate geometric properties of a relevant entity. An optimisation problem is often formulated in order to find these parameters. This thesis presents an elaboration of fundamental numerical algorithms for estimating parameters of multi-objective models of importance in computer vision applications. The work examines ways to solve unconstrained and constrained minimisation problems from the view points of theory, computational methods, and numerical performance.

The research starts by considering a particular form of multi-equation constraint function that characterises a wide class of unconstrained optimisation tasks. Increasingly sophisticated cost functions are developed within a consistent framework, ultimately resulting in the creation of a new iterative estimation method. The scheme operates in a maximum likelihood setting and yields near-optimal estimate of the parameters. Salient features of the method are that it has simple update rules and exhibits fast convergence. Then, to accommodate models with functional dependencies, two variant of this initial algorithm are proposed. These methods are improved again by reshaping the objective function in a way that presents the original estimation problem in a reduced form. This procedure leads to a novel algorithm with enhanced stability and convergence properties.

To extend the capacity of these schemes to deal with constrained optimisation problems, several a posteriori correction techniques are proposed to impose the so-called ancillary constraints. This work culminates by giving two methods which can tackle ill-conditioned constrained functions. The combination of the previous unconstrained methods with these post-hoc correction schemes provides an array of powerful constrained algorithms.

The practicality and performance of the methods are evaluated on two specific applications. One is planar homography matrix computation and the other trifocal tensor estimation. In the case of fitting a homography to image data, only the unconstrained algorithms are necessary. For the problem of estimating a trifocal tensor, significant work is done first on expressing sets of usable constraints, especially the ancillary constraints which are critical to ensure that the computed object conforms to the underlying geometry. Evidently here, the post-correction schemes must be incorporated in the computational mechanism. For both of these example problems, the performance of the unconstrained and constrained algorithms is compared to existing methods. Experiments reveal that the new methods perform with high accuracy to match a state-of-the-art technique but surpass it in execution speed.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution. To the best of my knowledge and belief, it contains no material previously published or written by any other person, except where due reference is made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being made available in all forms of media, now or hereafter known.

Tony Scoleri
July 2008

Acknowledgments

The work in this thesis has been very challenging until the end and several people come to my mind for what they have contributed, directly or indirectly.

Firstly, I would like to offer my deeply felt thanks to my supervisor Professor Wojciech Chojnacki. I am grateful for his encouragement, diligence, and support throughout the course of my PhD. He has shared with me his experience and passion for mathematics and computer vision, and always displayed much enthusiasm and positive attitude.

Secondly, I would like to express my gratitude to my other supervisors, Professor Michael Brooks and Professor Michael Murray, for their assistance in various situations.

My thanks also extend to the current and former members of the Computer Vision group at the University – Dr. Anton van den Hengel, Dr. Darren Gawley, Dr. Thorsten Thormählen, Dr. John Bastian, Dr. Daniel Pooley, and Rhys Hill – for many interesting discussions and shared experiences.

I would like to acknowledge my friends, in particular Dr. Julian Sorensen, Dr. Luke Rosenberg, and Voon Siong Wong for their help and perspectives.

Finally, my greatest thanks go to my family for their understanding, love, and support over the years, especially after a near-death cycling accident which occurred beyond my control only a few months before this thesis was due. I would have had many reasons to give up this work but they gave me the courage necessary to carry it out to completion.

Publications

[1] T. Scoleri, W. Chojnacki and M. J. Brooks. A multi-objective parameter estimator for image mosaicing. In A. Bouzerdoum and A. Beghdadi, editors, Proceedings of the *Eighth IEEE International Symposium on Signal Processing and its Applications (ISSPA-05)*, Sydney, Australia, 28-31 August, 2005, vol 2, pp. 551-554. ISBN 0-7803-9244-2.

[2] T. Scoleri, W. Chojnacki and M. J. Brooks. A decoupled algorithm for vision parameter estimation with application to the trifocal tensor. Proceedings of the *Fifth Digital Image Computing Techniques and Applications (DICTA-07)* conference, Adelaide, Australia, 3-5 December, 2007, vol 2, pp. 138-143. ISBN 0-7695-3067-2.

[3] T. Scoleri, W. Chojnacki and M. J. Brooks. Dimensionality Reduction for More Stable Vision Parameter Estimation. *IET Computer Vision journal*, vol 2, issue 4, December 2008. ISSN 1751-9632.

[4] T. Scoleri. Post-hoc Correction Techniques for Constrained Parameter Estimation in Computer Vision. Proceedings of the *Sixth Digital Image Computing Techniques and Applications (DICTA-08)* conference, Canberra, Australia, 1-3 December, 2008.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

$\mathbf{a} \in \mathbb{R}^p$	column vector of length p
$\mathtt{A} \in \mathbb{R}^{n \times m}$	$n \times m$ dimensional matrix
$\mathtt{I}_{n \times n}$	$n \times n$ identity matrix
$\mathtt{K}_n$	$n \times n$ commutation matrix
$\mathtt{A}^\mathsf{T}$	transpose of $\mathtt{A}$
$\mathtt{A}^{-1}$	inverse of $\mathtt{A}$
$\mathtt{A}^+$	Moore-Penrose inverse of $\mathtt{A}$
$\mathtt{A}_r^-$	r-truncated pseudo-inverse of $\mathtt{A}$
$\dim(\mathbf{a})$	dimensionality of $\mathbf{a}$
$\mathrm{tr}(\mathtt{A})$	trace of matrix $\mathtt{A}$
$\det(\mathtt{A})$	determinant of $\mathtt{A}$
$\mathrm{vec}(\mathtt{A})$	vectorisation of $\mathtt{A}$
$\mathrm{rank}(\mathtt{A})$	rank of $\mathtt{A}$
$\mathrm{diag}(\sigma_1, \ldots, \sigma_n)$	$n \times n$ matrix with $\sigma_1, \ldots, \sigma_n$ along the diagonal and zeros elsewhere
$\lVert\mathtt{A}\rVert_\mathrm{F}$	Frobenius norm of $\mathtt{A}$
$\mathbf{a} \times \mathbf{b}$	cross product of $\mathbf{a}$ and $\mathbf{b}$
$\mathtt{A} \otimes \mathtt{B}$	Kronecker product of $\mathtt{A}$ and $\mathtt{B}$
$\partial_{\boldsymbol{\theta}} J$	Jacobian (row) vector of J, $\left[\frac{\partial J}{\partial \theta_1}, \ldots, \frac{\partial J}{\partial \theta_n}\right]$
$\partial^2_{\boldsymbol{\theta}\boldsymbol{\theta}} J$	Hessian matrix of J, $\left[\frac{\partial^2 J}{\partial \theta_i \partial \theta_j}\right]_{1 \leq i,j \leq n}$
$\mathtt{H}_{\boldsymbol{\theta}}$	alternative notation for the Hessian matrix of J
$\mathbb{P}^2$	projective plane
$\mathbb{P}^{2*}$	dual projective plane

Vector and Matrix Sizes

$\mathbf{f}$	multi-objective constraint vector of length m
$\mathbf{f}'$	multi-objective sub-constraint vector of length r
$\mathbf{\Phi}$	multi-objective ancillary constraint vector of length q
$\boldsymbol{\theta}$	parameter vector of length l
$\boldsymbol{\mu}$	reduced parameter vector of length $l - m$
$\boldsymbol{\alpha}$	complementary parameter vector of length m such that $\boldsymbol{\theta} = [\boldsymbol{\mu}^\mathsf{T}, \boldsymbol{\alpha}^\mathsf{T}]^\mathsf{T}$
$\boldsymbol{\beta}$	parameter vector of length s
$\mathbf{x}_i$	$k \times 1$ element of data; $i = 1, \ldots, n$
$\mathtt{U}_i$	$l \times m$ measurement matrix made from $\mathbf{x}_i$
$\mathtt{U}'_i$	$(l - m) \times m$ reduced measurement matrix made from $\mathbf{x}_i$
$\mathrm{vec}(\mathtt{U}_i)$	vector of length ml
$\mathrm{vec}(\mathtt{U}'_i)$	vector of length $m(l - m)$
$\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{U}_i)$	$ml \times k$ derivative matrix of $\mathrm{vec}(\mathtt{U}_i)$ with respect to $\mathbf{x}_i$
$\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{U}'_i)$	$m(l - m) \times k$ derivative matrix of $\mathrm{vec}(\mathtt{U}'_i)$ with respect to $\mathbf{x}_i$
$\mathtt{\Lambda}_{\mathbf{x}_i}$	$k \times k$ covariance matrix of $\mathbf{x}_i$
$\mathtt{B}_i$	$ml \times ml$ propagated covariance matrix of $\mathbf{x}_i$
$\mathtt{\Sigma}_i$	$m \times m$ matrix
$\boldsymbol{\eta}_i$	Lagrange multiplier vector of length m
$\tilde{\mathtt{Z}}$	$(l - m) \times m$ "centroid" of $\mathtt{U}'_i$
$\mathtt{C}_{\boldsymbol{\theta}}$	$l \times l$ covariance matrix of $\boldsymbol{\theta}$
$\mathtt{X}_{\boldsymbol{\theta}}$	$l \times l$ matrix involved in the derivative of J_{AML} with respect to $\boldsymbol{\theta}$
$\mathbf{r}_k$	residual vector of length l
$\mathtt{J}_k$	$l \times s$ Jacobian matrix of $\mathbf{r}_k$
$\mathbf{p}_k$	search direction vector of length s

Chapter 1

Introduction

Modern computer vision systems need to process considerable amounts of data to perform several tasks almost simultaneously. Many of the applications are concerned with scene geometry estimation, object detection, trajectory estimation, online 3-D localisation, tracking, or some related sub-problems. Typically any such application is modelled mathematically by some algebraic constraints which establish a relationship between an underlying geometric problem and some image measurement data. Often the geometric problem can be summarised by key parameters. So, central to many vision applications is the problem of estimating parameters from given image features. This task puts a heavy toll on algorithms and numerical methods which are required to be more precise, more robust, and execute faster in order to serve the higher level purpose of the application.

The mathematical model underpinning a parameter estimation problem is generally multi-objective in character in its dealing with a system of equations and combines both the sought parameters and the image data. In some applications, the parameters are subject to ancillary constraints not involving image tokens. In this case, a parameter vector solution must satisfy both the principal multi-equation system and the ancillary constraints.

When prior knowledge about the measurement errors is available, covariance matrices can be incorporated in the estimation process to weight the observed data according to their degree of reliability. A statistical noise model suitable in this situation is the so-called errors-in-variables model. An estimator capable of such informed weighting can be formally derived from the principle of maximum likelihood (ML). This procedure allows for an optimal estimation of the parameters. Since the constraints are geometric in nature, the problem is regarded as an instance of geometric fitting.

One of several drawbacks with ML estimation is that the dimensionality of the parametric model increases with each new measurement. The associated optimisation problem can quickly become intractable. This thesis presents fundamental algorithms based on an approximated maximum likelihood formulation of the underlying estimation task. The parameter estimates generated exhibit nearly optimal statistical behaviour and—unlike the maximum likelihood estimate—are relatively inexpensive to compute.

1.1 Thesis outline

Often in optimisation or estimation problems researchers focus on formulating cost functions to minimise some algebraic or geometric error and then employ well-established numerical schemes to perform the minimisation. These methods can sometimes involve significant overhead and be more sophisticated than necessary which comes at an expensive computational cost.

The next chapter in this thesis presents a progressive elaboration of specifically designed algorithms for solving a class of general constrained parameter estimation problems encountered in geometric computer vision and pattern recognition. The remaining chapters exemplify the proposed techniques in the context of realistic vision applications.

In more detail, Chapter 2 introduces necessary theoretical material for discussing parameter estimation and sets forth a parametric model applicable to a wide range of underlying vision problems. The first couple of sections deal with the topic of unconstrained estimation, when the ancillary constraints on the parameters are ignored. Increasingly sophisticated cost functions are developed, including a maximum likelihood cost function but special emphasis is given to an approximated maximum likelihood (AML) cost function. Using an AML cost function over an ML one has significant benefits provided some care is taken. First, a series of three algorithms are devised for minimising the AML function when the parametric model suffers from functional dependencies or none. Second, a reformulation of the multi-objective AML function leads to the development of a novel algorithm which operates in a reduced space than the original three schemes. This algorithm enjoys better stability and convergence properties. The second part of the chapter is then dedicated to post-hoc correction techniques to enforce ancillary constraints to the result of the previous unconstrained minimisation. Several methods with various degrees of complexity are proposed. Specific algorithms are also given to tackle ill-posed constrained problems.

Chapter 3 provides an empirical evaluation of the linear and nonlinear optimisation methods developed for unconstrained parameter estimation. The application chosen is that of computing a planar homography. It will be assumed throughout this thesis that the system of cameras is uncalibrated and that each camera captures the scene under a general projective projection model. Extended Euclidean geometry is then employed to represent the natural geometric relationships between a scene and its projections, or those between corresponding image features across images. Now, because the homography model is described by a multi-component objective vector with linear dependencies, accuracy of the solutions and behaviour of the three AML estimators is examined thoroughly. Experiments are also conducted to compare the accuracy and computational efficiency of other estimation techniques including the ML estimator.

An intermediate chapter is then devoted to expose some theoretical material about the trifocal tensor relating three views of a scene. Following a brief overview of the perspective camera model and the epipolar geometry between a stereo pair of images, Chapter 4 provides a substantial, self-contained study on trifocal geometry. A review of the trilinear relations between image features (lines and/or points) and the trifocal tensor is given. Special care is taken to translate the geometrical relationships into algebraic formulae in a clear and concise manner. The background work on the trifocal tensor establishes notations and relations which lead ultimately to the presentation of the most common internal trifocal constraints.

In a natural extension of this work, an experimental chapter provides details about trifocal tensor computation. The post-hoc correction schemes presented in Chapter 2 are applied and their performance evaluated through both simulated and real image sequences. The trifocal constraints elaborated in the previous chapter are used in two ways, either directly in some of the adjustment techniques or to provide accuracy measures on the final constrained estimate generated. To complete the analysis, optimal fundamental matrices are computed between pairs of images and the accuracy of the resulting estimates compared with that of trifocal tensors.

A final chapter summarises the findings in this thesis. It also opens up several avenues for future research.

1.2 Thesis contributions

This thesis has drawn on expertise from various research disciplines and integrated the knowledge into one piece of work. On one hand there is theoretical material regarding general optimisation techniques and, on the other hand, theory and applications related to computer vision with special attention given to effective computation of the trifocal tensor.

The development of cost functions leading to accurate unconstrained estimators and post-correction methods is exposed neatly within an explicitly defined mathematical model and consistent framework. This presentation aims to facilitate understanding and contributes to a wider effort to place a variety of methods within a global, common context. The parametric model considered in this thesis describes a multi-objective vector-valued principal constraint function, generalising the single-equation case investigated in earlier work [33]. A similar upgrade was achieved when considering ancillary constraints on the parameters. The previous approach was restricted to incorporate a single ancillary constraint so multiple constraints had to be included by summing up the contributions of the squared individual constraints, which is not optimal. In this thesis, proper multi-objective cost functions are defined to tackle problems with several ancillary constraints.

1.2 Thesis contributions

The adoption of a two-stage approach to solving constrained estimation problems offers several advantages. Firstly, it permits to easily discard the ancillary constraints enforcement process if the problem is unconstrained. Secondly, the individual methods, for constrained or unconstrained estimation, can be subject to various standard or customised performance tests, thus facilitating subsequent analysis, comparative studies, and possible refinement. Overall, the new proposed estimators have the advantage of dealing with multi-objective (principal and ancillary) constraint functions, are simply expressed, precise, and computationally efficient. Additionally, they are fairly robust in their ability to cope with linearly dependent cost functions, degenerate camera configurations, and ill-posed problems.

While devising our own algorithms, we also review, improve, or shed new lights on the work of several key contributors to the fundamental algorithmic theories in computer vision. Although the separate development of these algorithms has been reported in earlier work, the originality here comes from describing them in a unified framework. This allows us to put everyone's work into perspective with respect to one another. It is hoped that the contributions from all parties are also better appreciated this way.

Algorithms developed in this thesis are applied to two example problems: planar homography computation and trifocal tensor estimation. With regards to homography fitting, minimisation techniques are evolved from cost functions built from different matrix generalised inverses. The value of the experimental section in Chapter 3 is to show that the selection of an appropriate type of matrix generalised inverse is critical to generate good-quality parameter estimates. Although the cost functions are connected through the use of different generalised inverses, the obtained solutions are not guaranteed to be equivalent. Some of the experiments address this question specifically and reveal that the final estimate of a given estimator is the minimiser not only of its own cost function but also of other functions as long as the cost functions are differentiable or rendered continuous via regularisation.

One major difficulty when dealing with the trifocal tensor is to understand the many different relations and constraints imposed by the trifocal tensor on image measurements, and then realise the connections between these relations. To help with this, we have chosen to expose the trifocal tensor by considering its contractions with an increasing number of image tokens, from a single point/line feature to a combination of three points/lines across the views. Although the work reproduces well-known results, the approach is innovative in that trilinear relations are tackled from an uncommon but consistent manner.

The power of the trifocal tensor comes from its roots in multilinear algebra. In that respect, once defined, it can be manipulated or decomposed in a multitude of ways, or combined with various other entities. One major inconvenience stemming from the generality of the algebra is that there is no unique definition to characterise the trifocal tensor from camera projection

matrices. A considerable amount of literature covers results, rules, constraints, proofs on the trifocal tensor but connections between them are not obvious and one must penibly start from the supplied definition of the tensor and proceed with the analysis.

In this thesis, we have adopted the definition proposed by Hartley [37]. All the main trilinear constraints known to date are then expressed within a common framework based on this particular characterisation of the trifocal tensor. While giving the transfer equations in the usual tensorial notation (with covariant and contravariant indices), a novelty is also to represent them in vector or matrix form. It is hoped that these results elevate the general understanding—and take away some of the burden of dealing with tedious index manipulations. In turn, this work encourages implementation because it is easier in practice to operate at the feature level (using lines and points coordinates) rather than defining every rule or transfer equation through indices, which is more error-prone.

A further area of contributions comes from the work on expressing the ancillary constraints on the trifocal tensor. The original sets of constraints were all described starting from a different definition of the trifocal tensor. Here, all constraints are expressed within a common context, edified from Hartley's trifocal tensor description. The analysis and derivations help discovering their origin, relating them, and ultimately comparing their effect in practice. Literature which provides such detailed and comprehensive summary of the trifocal constraints, for both principal and ancillary constraints, is almost non-existent. This part of the thesis is essential to gather knowledge into one body of work.

It is believed that further contributions come from the methodology employed in the experiments. Often the unconstrained and constrained estimation stages are tested on separate sets of data. Only with real image sequences the computation of both entities is carried out consecutively from one sample data. In this case, there is often no feedback on the accuracy of the unconstrained estimate since the focus is centred exclusively on the quality of the final constrained tensor. In each of our synthetic test, the same data were used to estimate both unconstrained and constrained trifocal tensors. This strategy allows to apply performance measures on both types of estimates and compare the results. For instance, a constrained estimate should typically give a slightly higher cost function value compared to an unconstrained one. So the benefit can be seen to occur at two levels. One is that it permits checking theoretical properties and the other is about testing the performance of individual constrained and unconstrained algorithms.

To conclude, it is recognised that a trifocal tensor provides more accuracy than a fundamental matrix, however, little study or none has been done to show how much gain there is and where the gain takes place exactly. This sort of analysis is quite vital when one needs to decide whether utilising a trifocal tensor or not. This issue is addressed in the experiments carried out on real image data. In general, the simplicity and efficiency of a fundamental matrix is preferred to

a trifocal tensor which is complex to first understand and then compute stably. These major hurdles make it unpopular and abandoned. It is hoped that the results presented in Chapters 4 and 5 will convince and encourage the vision community to use it more often, and believe in the validity of the algorithmic theories set forth in Chapter 2.

Chapter 2

Towards Constrained Parameter Estimators

Determining a global, constrained minimiser is a difficult task which often depends on the characteristics of the objective function, for instance its continuity, convexity, linearity or nonlinearity. This chapter presents several estimation methods, or *estimators*, capable of minimising objective functions derived for a particular parametric model underlying a wide class of problems in computer vision and pattern recognition. The first section describes in detail a class of specific type of parametric model considered in this thesis. The remaining sections will then focus on developing appropriate cost functions and estimation techniques to minimise these functions. Advantages and drawbacks of each technique will be discussed. It is important to understand the limitations of the presented algorithms because knowledge of such limitations will guide their improvement.

2.1 Parametric model

Fitting parametric models to data is a ubiquitous task in computer vision. A parametric model combines parameters and image features to reflect a particular geometric relationships of interest. Image features are typically low-level descriptors of an image contents, for instance the locations of corresponding 2-D points and/or lines in images. Similar features across images can be assembled in a vector $\mathbf{x} = [x_1, \ldots, x_k]^\mathsf{T}$ and considered as a single element of data. Typically, a parameter vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_l]^\mathsf{T}$ and image data pertaining to a model are described by a system of equations

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}, \tag{2.1}$$

where

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = [f_1(\mathbf{x}, \boldsymbol{\theta}), \ldots, f_m(\mathbf{x}, \boldsymbol{\theta})]^\mathsf{T},$$

is a vector of *multi-objective constraints*. Here, the label "multi-objective" is used because the range space of function $\mathbf{f}$ will be assumed to have dimension m greater than one. The case investigating a single objective function was the subject of previous studies [15, 23, 33, 49].

Models with a multi-component constraint vector of type (2.1) include ones in which parameters describe such entities as a planar *homography* [13, 51, 100], a *3-D rotation matrix* [68], a *camera projection matrix* [62], and an *aggregate of affine subspaces* [19, 91]. A vast class of computer vision models, including those mentioned above, employs a constraint vector in which the parameters are bound linearly with the measurements as per

$$f_i(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{u}_i(\mathbf{x})^\mathsf{T}\boldsymbol{\theta},$$

where $\mathbf{u}_i(\mathbf{x})$ is a polynomial function in $[\mathbf{x}^\mathsf{T}, 1]^\mathsf{T}$. In this case, the constraint vector can succinctly be written as

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathtt{U}(\mathbf{x})^\mathsf{T}\boldsymbol{\theta}, \tag{2.2}$$

where $\mathtt{U}(\mathbf{x}) = [\mathbf{u}_1(\mathbf{x}), \ldots, \mathbf{u}_m(\mathbf{x})]$ is an $l \times m$ *carrier matrix* encapsulating the nonlinearity in the data. It is worth noting the fact that the constraint vector depends linearly on $\boldsymbol{\theta}$ implies that equation (2.1) is invariant to the multiplication of $\boldsymbol{\theta}$ by a non-zero scalar. This in turn entails that any useful cost function J for generating estimates of $\boldsymbol{\theta}$ has to be insensitive to scale change, or $\boldsymbol{\theta}$*-homogeneous*, satisfying $J(t\boldsymbol{\theta}) = J(\boldsymbol{\theta})$ for every non-zero scalar t. Clearly, an estimate embodied by the minimiser of a $\boldsymbol{\theta}$-homogeneous cost function is determined only up to a scalar factor.

Noise model

We assume the most general situation where measurement data are affected by anisotropic and inhomogeneous, that is, heteroscedastic, noise. More precisely, we adopt a noise model whereby a datum $\mathbf{x}_i$ is assumed to result from a perturbation of some unobservable *ideal value* $\overline{\mathbf{x}}_i$ such that

$$\mathbf{x}_i = \overline{\mathbf{x}}_i + \boldsymbol{\delta}_{\mathbf{x}_i}, \tag{2.3}$$

where $\boldsymbol{\delta}_{\mathbf{x}_i}$ is seen as a sample from a set of errors distributed according to the Gaussian law with mean zero and *covariance matrix* $\Lambda_{\mathbf{x}_i}$ a proviso that the distributions associated with different points $\mathbf{x}_i$ are independent.

Each ideal point has the property that it satisfies exactly the multi-objective constraints, $\mathtt{U}(\overline{\mathbf{x}}_i)^\mathsf{T}\boldsymbol{\theta} = \mathbf{0}$, for a common value of $\boldsymbol{\theta}$. The set

$$\mathcal{M}_{\boldsymbol{\theta}} = \{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}\}$$

forms a manifold which we term a *model manifold*.

Ancillary constraints

In many vision applications the system (2.1) does not encompass all of the constraints that apply to the problem. These equations which relate data to parameters often constitute a set of

principal constraints in the model. It turns out that the parameters can exhibit some intrinsic properties which are completely independent of any image feature but which must be accounted for in order to conform to the underlying geometry. These additional requirements are described by *ancillary constraints* of the form

$$\boldsymbol{\Phi}(\boldsymbol{\theta}) = \mathbf{0}, \tag{2.4}$$

where $\boldsymbol{\Phi} = [\phi_1(\boldsymbol{\theta}), \ldots, \phi_q(\boldsymbol{\theta})]^\mathsf{T}$ is a nonlinear, vector-valued function of the parameter vector $\boldsymbol{\theta}$ only. In general, each sub-constraint function ϕ_j is *homogeneous* of degree κ_j, or κ_j-homogeneous, with the property that

$$\phi_j(\lambda\boldsymbol{\theta}) = \lambda^{\kappa_j}\phi_j(\boldsymbol{\theta}), \qquad \lambda \neq 0.$$

In this thesis, it will be assumed that each function ϕ_j is smooth enough to satisfy

$$\phi_j(\boldsymbol{\theta} + \mathbf{h}) = \phi_j(\boldsymbol{\theta}) + \mathbf{h}^\mathsf{T}\mathbf{a}_j(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{h}^\mathsf{T}\mathtt{A}_j(\boldsymbol{\theta})\mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3),$$

where $\mathbf{a}_j(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}}\phi_j(\boldsymbol{\theta})^\mathsf{T}$ and $\mathtt{A}_j(\boldsymbol{\theta}) = \partial^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\phi_j(\boldsymbol{\theta})$ are the gradient and Hessian of ϕ_j respectively, for $j = 1, \ldots, q$. The notation $\mathcal{O}(\|\mathbf{h}\|^3)$ stands for an entity that, when divided by $\|\mathbf{h}\|^3$, remains bounded as $\|\mathbf{h}\| \to 0$.

When combined with the principal constraints, the ancillary constraints may restrict the solution to curves, surfaces (or other submanifolds) of lower dimensionality than the unconstrained solution space. Basic examples of models involving principal and ancillary constraints include the stereo and motion problems of estimating the fundamental matrix [1, 17, 84], flow fundamental matrix [52], and coefficients of the differential epipolar equation [14], conic fitting problems [8, 50, 55, 60], and multiple-view structure from motion problems with estimation of the *trifocal* and *quadrifocal tensors* [24, 26, 27, 37, 79].

2.2 Cost functions and estimators

Given a collection $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of image data, we aim to determine $\widehat{\boldsymbol{\theta}} \neq \mathbf{0}$ satisfying (2.4) such that (2.1) holds for each image datum $\mathbf{x}_i$, where $\mathbf{x}$ in (2.1) is now replaced by $\mathbf{x}_i$ for $1 \leq i \leq n$. Our approach to solve this type of constrained optimisation problem is first to design good quality unconstrained estimators and then employ a post-hoc adjustment procedure to accommodate for the ancillary constraints.

We shall use *cost functions* to measure the extent to which the data and candidate estimates fail to adhere to (2.1). Specifically we will consider cost functions that exhibit nonlinearity in both the data and the parameters. It will be assumed that any such function J is smooth enough for the following Taylor expansion to hold

$$J(\boldsymbol{\theta} + \mathbf{h}) = J(\boldsymbol{\theta}) + \mathbf{h}^\mathsf{T}\mathbf{g}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{h}^\mathsf{T}\mathtt{H}(\boldsymbol{\theta})\mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3).$$

Here $\mathbf{g}(\boldsymbol{\theta})$ and $\mathtt{H}(\boldsymbol{\theta})$ represent the gradient and Hessian of J respectively,

$$\mathbf{g}(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} J(\boldsymbol{\theta})^{\mathsf{T}}, \qquad \mathtt{H}(\boldsymbol{\theta}) = \partial^2_{\boldsymbol{\theta}\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

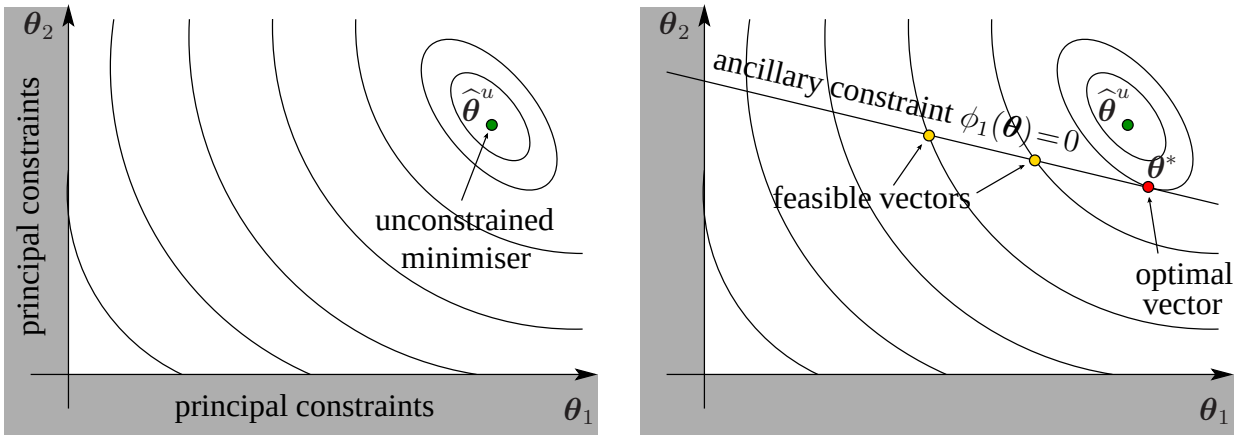
If the ancillary constraints are set aside, then a global *unconstrained* minimiser $\widehat{\boldsymbol{\theta}}^u$ of a cost function $J = J(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n)$ is defined by

$$\widehat{\boldsymbol{\theta}}^u = \underset{\boldsymbol{\theta} \in \mathbb{R}^l}{\arg\min}\ J(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n).$$

When the ancillary constraints are taken into account, then the problem becomes

$$\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta} \in \mathcal{V}}{\arg\min}\ J(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) \\ \text{such that } \mathcal{V} &= \{\boldsymbol{\theta} \in \mathbb{R}^l \,|\, \boldsymbol{\Phi}(\boldsymbol{\theta}) = \mathbf{0}\}. \end{aligned}$$

Clearly, the condition that $\boldsymbol{\theta}$ must belong to the set of *feasible vectors* $\mathcal{V}$ restricts further the solution space. The $\boldsymbol{\theta}$-vector satisfying this requirement and for which the cost function value is minimal is the global *constrained* minimiser of J, denoted $\boldsymbol{\theta}^*$. Figure 2.1 shows two contour plots of a typical quadratic optimisation problem with and without an ancillary constraint.



(a) (b)

Figure 2.1. Contours of a quadratic in $\mathbb{R}^2$: (a) with its global unconstrained minimiser $\widehat{\boldsymbol{\theta}}^u$; (b) with ancillary constraint ϕ_1 and global constrained minimiser $\boldsymbol{\theta}^*$.

The next couple of sections will introduce several cost functions with different algebraic and geometric properties. The minimisation algorithms associated with these functions will find an unconstrained $\boldsymbol{\theta}$-estimate. The second half of the chapter will then present several constraint-enforcement techniques which can be applied to the result of these unconstrained estimators to obtain a final constrained minimiser.

2.2.1 Algebraic least squares

A simple cost function results from considering the system

$$\mathtt{U}(\mathbf{x}_i)^\mathsf{T}\boldsymbol{\theta} = \mathbf{0} \quad (i = 1, \ldots, n),$$

where the i-th equation is a copy of (2.2) in which $\mathbf{x}_i$ replaces $\mathbf{x}$. This system can be expressed as

$$\mathtt{M}^\mathsf{T}\boldsymbol{\theta} = \mathbf{0}, \tag{2.5}$$

where $\mathtt{M} = [\mathtt{U}(\mathbf{x}_1), \ldots, \mathtt{U}(\mathbf{x}_n)]$ is an $l \times mn$ *design matrix*. When $nm > l$, equation (2.5) is over-constrained and, typically, does not admit a non-zero solution. A non-trivial approximate solution in the least-squares format can be obtained by applying the following optimisation rule: minimise $\|\mathtt{M}^\mathsf{T}\boldsymbol{\theta}\|$ subject to $\|\boldsymbol{\theta}\| = 1$. Here $\|\cdot\|$ denotes the Euclidean norm. Given that

$$\|\mathtt{M}^\mathsf{T}\boldsymbol{\theta}\|^2 = \sum_{i=1}^{n} \|\mathtt{U}(\mathbf{x}_i)^\mathsf{T}\boldsymbol{\theta}\|^2 = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}),$$

the solution coincides with the minimiser of the *algebraic least-squares* (ALS) cost function

$$J_{\mathrm{ALS}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \|\boldsymbol{\theta}\|^{-2} \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) = \|\boldsymbol{\theta}\|^{-2}\boldsymbol{\theta}^\mathsf{T}\mathtt{S}\,\boldsymbol{\theta}, \tag{2.6}$$

where $\mathtt{S} = \sum_{i=1}^{n} \mathtt{U}(\mathbf{x}_i)\mathtt{U}(\mathbf{x}_i)^\mathsf{T}$ is the *scatter matrix*. The label 'algebraic least-squares' is reminiscent of the *algebraic distance* between a data point $\mathbf{x}$ and a normalised parameter $\|\boldsymbol{\theta}\|^{-1}\boldsymbol{\theta}$, defined as $\|\mathbf{f}(\mathbf{x}, \|\boldsymbol{\theta}\|^{-1}\boldsymbol{\theta})\| = \|\boldsymbol{\theta}\|^{-1}(\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^\mathsf{T}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}))^{1/2}$, instantiations of which, in the form of the $\|\mathbf{f}(\mathbf{x}_i, \|\boldsymbol{\theta}\|^{-1}\boldsymbol{\theta})\|^2$ terms, appear in the expression for J_{ALS}.

It is easily seen that the ALS estimate, $\widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$, coincides, up to scale, with an eigenvector of $\mathtt{S}$ associated with the smallest eigenvalue, and this can be found by performing *singular value decomposition* (SVD) on $\mathtt{S}^\mathsf{T}$ [54,92]. The ALS method exploiting SVD is essentially equivalent to the Direct Linear Transformation algorithm described in [37].

2.2.2 Generalised total least squares

A *generalised total least-squares* (GTLS) estimator was originally proposed for problems described by a single-objective principal constraint function [55]. A key feature of this estimator is that it utilises a common covariance matrix that weights the carrier vectors $\mathbf{u}_i$. This work was later extended to the multi-objective case [58, 61] and is exposed next.

For each $i = 1, \ldots, n$, and $j = 1, \ldots, m$, let

$$\widehat{\Sigma}_{ji} = [\partial_{\mathbf{x}}\mathbf{u}_j]_{\mathbf{x}=\mathbf{x}_i}\Lambda_{\mathbf{x}_i}[(\partial_{\mathbf{x}}\mathbf{u}_j)^\mathsf{T}]_{\mathbf{x}=\mathbf{x}_i}$$

be the $l \times l$ matrix which is the result of propagating $\Lambda_{\mathbf{x}_i}$ by the mapping $\mathbf{x} \mapsto \mathbf{u}_j(\mathbf{x})$. Suppose that $\tilde{\Sigma}$ is an approximate solution of the equation

$$\widehat{\Sigma}_{ji} = \gamma_{ji}\,\tilde{\Sigma}, \quad \gamma_{ji} > 0, \quad i = 1, \ldots, n,\ j = 1, \ldots, m,$$

obtained by minimising

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\widehat{\Sigma}_{ji} - \gamma_{ji}\,\tilde{\Sigma}\|_{\mathrm{F}}^2,$$

where $\|\mathtt{A}\|_{\mathrm{F}}$ denotes the Frobenius norm of the matrix $\mathtt{A}$. The solution to this problem is characterised by

$$\tilde{\Sigma} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ji}\,\widehat{\Sigma}_{ji}}{\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ji}^2} \quad \text{and} \quad \gamma_{ji} = \frac{\mathrm{tr}(\tilde{\Sigma}\,\widehat{\Sigma}_{ji})}{\mathrm{tr}(\tilde{\Sigma}^2)},$$

where $\mathrm{tr}(\mathtt{A})$ stands for the trace of the matrix $\mathtt{A}$. These equations cannot be solved in closed form. However, a short procedure can be implemented to obtain an estimate of $\tilde{\Sigma}$ (starting from $\gamma_{ji} = 1$) and γ_{ji} by iterating over the above expressions. Now, define a *weighted scatter matrix*

$$\tilde{\mathtt{S}} = \sum_{i=1}^{n} \mathtt{U}(\mathbf{x}_i)\,\Upsilon_i^{-1}\,\mathtt{U}(\mathbf{x}_i)^{\mathsf{T}},$$

where $\Upsilon_i = \mathrm{diag}(\gamma_{1i}, \ldots, \gamma_{mi})$. The GTLS estimate, denoted $\widehat{\boldsymbol{\theta}}_{\mathrm{GTLS}}$, can then be expressed as the minimiser of

$$J_{\mathrm{GTLS}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{\boldsymbol{\theta}^{\mathsf{T}}\tilde{\mathtt{S}}\,\boldsymbol{\theta}}{\boldsymbol{\theta}^{\mathsf{T}}\tilde{\Sigma}\,\boldsymbol{\theta}}. \tag{2.7}$$

Like standard ALS, the solution of the above problem can also be expressed in closed form, in this case requiring a generalised eigendecomposition of $(\tilde{\mathtt{S}}, \tilde{\Sigma})$. The GTLS estimate can be found directly as the generalised eigenvector corresponding to the smallest generalised eigenvalue of

$$\tilde{\mathtt{S}}\boldsymbol{\theta} = \lambda\tilde{\Sigma}\boldsymbol{\theta}.$$

It should be noted that no prior estimate of $\boldsymbol{\theta}$ is required here to solve the eigenproblem since it relies on the pair of matrices $\tilde{\mathtt{S}}$ and $\tilde{\Sigma}$. As a result, this method is often used as initialisation of more sophisticated iterative algorithms.

2.2.3 Maximum likelihood

The ALS cost function is introduced *ad hoc* and is lacking a satisfactory statistical basis. A statistically viable cost function can be derived by adopting a stochastic model for the data generation process. It is natural to assume that the data have been created by choosing a model manifold, selecting points on it, and perturbing these points using Gaussian noise. The observed data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ can then be viewed as a sample drawn from random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$

whose joint distribution is labelled by an *extended* parameter vector $\boldsymbol{\psi} = (\boldsymbol{\theta}; \overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$, where $\boldsymbol{\theta}$ is the *principal* part of $\boldsymbol{\psi}$ representing a particular geometric primitive and $(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$ is the *subsidiary* part representing a specific selection of points on the primitive. The two components of $\boldsymbol{\psi}$ are such that $\boldsymbol{\theta} \neq \mathbf{0}$ and

$$\mathbf{f}(\overline{\mathbf{x}}_1, \boldsymbol{\theta}) = \cdots = \mathbf{f}(\overline{\mathbf{x}}_n, \boldsymbol{\theta}) = \mathbf{0}. \tag{2.8}$$

Using (2.3), the distribution associated with a particular $\boldsymbol{\psi}$ can be described by the probability density function (pdf)

$$p(\overline{\mathbf{x}}_1 + \boldsymbol{\delta}_{\mathbf{x}_1}, \ldots, \overline{\mathbf{x}}_n + \boldsymbol{\delta}_{\mathbf{x}_n} \,|\, \boldsymbol{\psi}) = (2\pi)^{-kn/2} \prod_{i=1}^{n} \det(\Lambda_{\mathbf{x}_i})^{-1/2} \times \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \boldsymbol{\delta}_{\mathbf{x}_i}^{\mathsf{T}} \Lambda_{\mathbf{x}_i}^{-1} \boldsymbol{\delta}_{\mathbf{x}_i} \right\}.$$

Here, k is the common length of the $\mathbf{x}_i$, for each $i = 1, \ldots, n$, $\Lambda_{\mathbf{x}_i}$ is a $k \times k$ symmetric covariance matrix, assumed to be known, that quantifies errors in the measurement of the data point $\mathbf{x}_i$. The above pdfs constitute all the essential ingredients needed for deciding which extended parameter labels the distribution that has most likely produced the data, provided that there are no special preferences as to how a particular geometric primitive and points on it have been chosen. According to the *principle of maximum likelihood*, the optimal extended parameter vector is the one that maximises the *likelihood function* $\boldsymbol{\psi} \mapsto p(\mathbf{x}_1, \ldots, \mathbf{x}_n \,|\, \boldsymbol{\psi})$. In view of the specific form of the (logarithm of the) likelihood function, this *maximum likelihood estimate* of $\boldsymbol{\psi}$, $\widehat{\boldsymbol{\psi}}_{\mathrm{ML}}$, can alternatively be characterised as the extended parameter vector whose subsidiary part minimises the squared *Mahalanobis distance* from the data. The squared Mahalanobis distance between the data points $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and model points $(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$ is given by

$$d^2_{\mathrm{Mahal}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n) = \sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}}_i)^{\mathsf{T}} \Lambda_{\mathbf{x}_i}^{-1} (\mathbf{x}_i - \overline{\mathbf{x}}_i). \tag{2.9}$$

For each $\boldsymbol{\theta} \neq \mathbf{0}$, when restricted to the set of those $(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$ for which (2.8) holds, the function

$$d^2_{\mathrm{Mahal}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \bullet) \colon (\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n) \mapsto d^2_{\mathrm{Mahal}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$$

attains a constrained minimum at some point $(\overline{\mathbf{x}}_1^{\boldsymbol{\theta}}, \ldots, \overline{\mathbf{x}}_n^{\boldsymbol{\theta}})$. All these minima can be assembled into a cost function to yield

$$J_{\mathrm{ML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = d^2_{\mathrm{Mahal}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \overline{\mathbf{x}}_1^{\boldsymbol{\theta}}, \ldots, \overline{\mathbf{x}}_n^{\boldsymbol{\theta}}). \tag{2.10}$$

Denote by $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ the minimiser of J_{ML}. Clearly, $J_{\mathrm{ML}}(\widehat{\boldsymbol{\theta}}_{\mathrm{ML}})$ is the smallest of all the values that $d^2_{\mathrm{Mahal}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \bullet)$ attains at the subsidiary parts of extended parameter vectors. Consequently, $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ and $(\overline{\mathbf{x}}_1^{\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}}, \ldots, \overline{\mathbf{x}}_n^{\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}})$ are the principal and subsidiary parts of $\widehat{\boldsymbol{\psi}}_{\mathrm{ML}}$. This justifies calling J_{ML} the maximum likelihood cost function for $\boldsymbol{\theta}$-estimation, and $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ the maximum

likelihood estimate of $\boldsymbol{\theta}$. Of all candidate parameter vectors, $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ is the preferred vector that makes the observed data as likely as possible. The ML function shown in (2.10) provides an optimal estimator under the defined parametric model of Section 2.1.

The demand for optimality introduces some major drawbacks though. Firstly, an explicit expression for J_{ML} must be derived for each problem specifically. In chapters 3 and 5 we give its form relevant to homography and trifocal tensor estimation respectively. Secondly, finding $(\overline{\mathbf{x}}_1^{\boldsymbol{\theta}}, \ldots, \overline{\mathbf{x}}_n^{\boldsymbol{\theta}})$ for each $\boldsymbol{\theta}$ is a difficult task. Each new measurement $\mathbf{x}_i$ requires the addition of a corresponding $\overline{\mathbf{x}}_i$ to the optimisation set. As a result, the ML estimator effectively operates over a search space of dimension $\sum_{i=1}^{n} \dim(\overline{\mathbf{x}}_i) + \dim(\boldsymbol{\theta}) = kn + l$. The parameters $(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n)$ are only of intermediate use to obtain $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ and not required as final output. For this reason, they are often called the *nuisance parameters*. In practical applications, minimisation of J_{ML} is possible but restricted to relatively small data sets, as the problem quickly becomes intractable. Sparse matrix techniques may be employed to alleviate the computational load, however, this usually complicates greatly the algorithm. A more feasible approach is to seek to minimise an appropriate approximation of J_{ML} that captures near-optimality and which does not optimise over the nuisance parameters. One such approximation is indicated next.

2.2.4 Approximated maximum likelihood

Sampson [75] first proposed and Kanatani [49] next popularised a first-order approximation to J_{ML} defined by

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}} [\partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} \partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}]^{-1} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}). \tag{2.11}$$

Sometimes we may abbreviate this function to $J_{\mathrm{AML}}(\boldsymbol{\theta})$ since $\boldsymbol{\theta}$ is the only unknown. Evidently, the dimension of the search space is now that of $\boldsymbol{\theta}$, as desired. Underlying this *approximated maximum likelihood* (AML) cost function is the assumption that the system of model equations is of full rank, ensuring in particular that $\partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \Lambda_{\mathbf{x}} \partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^{\mathsf{T}}$ is invertible for any $\mathbf{x}$ and $\boldsymbol{\theta}$ satisfying (2.1). Here we give a general form of a AML cost function pertinent to models described by rank-deficient systems of equations. Our development will critically rely on the notion of a generalised inverse of a matrix.

Given an $m \times n$ matrix $\mathtt{A}$, a generalised inverse, or g-inverse, of $\mathtt{A}$ is any $n \times m$ matrix $\mathtt{A}^{-}$ satisfying

$$\mathtt{A}\mathtt{A}^{-}\mathtt{A} = \mathtt{A}.$$

Such matrices are sometimes called *one condition* g-inverses or *equation solving* g-inverses because of their use in solving systems of linear equations. Consider the following conditions for a (real) matrix $\mathtt{X}$:

(1) $\mathtt{AXA} = \mathtt{A}$,
(2) $\mathtt{XAX} = \mathtt{X}$,
(3) $(\mathtt{AX})^{\mathsf{T}} = \mathtt{AX}$,
(4) $(\mathtt{XA})^{\mathsf{T}} = \mathtt{XA}$.
(5) $\mathtt{AX} = \mathtt{XA}$ provided that $\mathtt{A}$ is square.

Let $\mathtt{A}^{(i,j,\ldots,l)}$ be any matrix that satisfies conditions (i), (j), …, (l) of the above itemised conditions. Such matrices are termed $(i, j, \ldots, l)$ g-inverses of $\mathtt{A}$. Particular cases include:

- $\mathtt{A}^{(1,2)}$, a *reflexive* g-inverse, denoted also $\mathtt{A}_r^-$;
- $\mathtt{A}^{(1,4)}$, a *minimum norm* g-inverse, denoted also $\mathtt{A}_m^-$;
- $\mathtt{A}^{(1,3)}$, a *least-squares* g-inverse, denoted also $\mathtt{A}_l^-$;
- $\mathtt{A}^{(1,2,3,4)}$, the *Moore–Penrose* g-inverse, or *pseudo-inverse*, denoted also $\mathtt{A}^+$ or $\mathtt{A}^\dagger$;
- and $\mathtt{A}^{(1,2,5)}$, the *group* or *Drazin* g-inverse, denoted also $\mathtt{A}^\#$ or $\mathtt{A}^D$.

Except for $\mathtt{A}^{(1,2,3,4)}$ and $\mathtt{A}^{(1,2,5)}$, g-inverses are, in general, not unique.

In view of (2.9) and (2.10), to develop an AML function, we need an approximate expression for

$$(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}} \Lambda_{\mathbf{x}_i}^{-1} (\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}).$$

To this end, note that when applied to the constrained minimiser $(\check{\mathbf{x}}_1^{\boldsymbol{\theta}}, \ldots, \check{\mathbf{x}}_n^{\boldsymbol{\theta}})$, the method of Lagrange Multipliers implies that, for each $i = 1, \ldots, n$, the gradient (the column vector of the partial derivatives) of $(\mathbf{x}_i - \mathbf{y})^{\mathsf{T}} \Lambda_{\mathbf{x}_i}^- (\mathbf{x}_i - \mathbf{y})$ with respect to $\mathbf{y}$ is a linear combination of the gradients of the components of (the row vector) $\mathbf{f}(\mathbf{y}, \boldsymbol{\theta})^{\mathsf{T}}$ with respect to $\mathbf{y}$, provided that all the gradients are evaluated at $\check{\mathbf{x}}_i^{\boldsymbol{\theta}}$. Since the first gradient is equal to $-2\Lambda_{\mathbf{x}_i}^- (\mathbf{x}_i - \mathbf{y})$ and the gradients of the components of $\mathbf{f}(\mathbf{y}, \boldsymbol{\theta})^{\mathsf{T}}$ coincide with the columns of $(\partial_{\mathbf{x}} \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}))^{\mathsf{T}}$, it follows that

$$\Lambda_{\mathbf{x}_i}^{-1} (\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}) = (\partial_{\mathbf{x}} \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}))^{\mathsf{T}} \boldsymbol{\lambda}_i \tag{2.12}$$

for some length-m vector of Lagrange multipliers $\boldsymbol{\lambda}_i$. By Taylor expanding $\mathbf{y} \mapsto \mathbf{f}(\mathbf{y}, \boldsymbol{\theta})$ to first order around $\check{\mathbf{x}}_i^{\boldsymbol{\theta}}$

$$\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \partial_{\mathbf{x}} \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})$$

and noting that $\mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \mathbf{0}$, we obtain the approximate equality

$$\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) = \partial_{\mathbf{x}} \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}).$$

This together with (2.12) yields the approximation

$$(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}} \Lambda_{\mathbf{x}_i}^{-1} (\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}) = (\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}} (\partial_{\mathbf{x}} \mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}))^{\mathsf{T}} \boldsymbol{\lambda}_i = \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\lambda}_i. \tag{2.13}$$

Now, using (2.12) restated as

$$\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}} = \Lambda_{\mathbf{x}_i}(\partial_{\mathbf{x}}\mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}))^{\mathsf{T}}\boldsymbol{\lambda}_i,$$

we find that

$$\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) = \Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})\boldsymbol{\lambda}_i, \tag{2.14}$$

where

$$\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \partial_{\mathbf{x}}\mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}(\partial_{\mathbf{x}}\mathbf{f}(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta}))^{\mathsf{T}}.$$

For notational convenience, we contract momentarily $\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})$ and $\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})$ to $\mathbf{f}_i$ and Σ_i, respectively. Then (2.13) reads

$$(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}}\Lambda_{\mathbf{x}_i}^{-1}(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}) = \mathbf{f}_i^{\mathsf{T}}\boldsymbol{\lambda}_i \tag{2.15}$$

and (2.14) reads

$$\mathbf{f}_i = \Sigma_i\boldsymbol{\lambda}_i. \tag{2.16}$$

Clearly,

$$\boldsymbol{\lambda}_i = \Sigma_i^{-}\Sigma_i\boldsymbol{\lambda}_i + (\mathtt{I} - \Sigma_i^{-}\Sigma_i)\boldsymbol{\lambda}_i,$$

where $\mathtt{I} = \mathtt{I}_{m\times m}$ is the $m \times m$ identity matrix. This combined with (2.16) yields

$$\boldsymbol{\lambda}_i = \Sigma_i^{-}\mathbf{f}_i + (\mathtt{I} - \Sigma_i^{-}\Sigma_i)\boldsymbol{\lambda}_i.$$

Hence, again by (2.16),

$$\begin{aligned}\mathbf{f}_i^{\mathsf{T}}\boldsymbol{\lambda}_i &= \mathbf{f}_i^{\mathsf{T}}(\Sigma_i^{-}\mathbf{f}_i + (\mathtt{I} - \Sigma_i^{-}\Sigma_i)\boldsymbol{\lambda}_i)\\ &= \mathbf{f}_i^{\mathsf{T}}\Sigma_i^{-}\mathbf{f}_i + \mathbf{f}_i^{\mathsf{T}}(\mathtt{I} - \Sigma_i^{-}\Sigma_i)\boldsymbol{\lambda}_i\\ &= \mathbf{f}_i^{\mathsf{T}}\Sigma_i^{-}\mathbf{f}_i + \boldsymbol{\lambda}_i^{\mathsf{T}}\Sigma_i^{\mathsf{T}}(\mathtt{I} - \Sigma_i^{-}\Sigma_i)\boldsymbol{\lambda}_i.\end{aligned}$$

Now taking into account that Σ_i is symmetric and using the definition of a generalised inverse, we see that

$$\Sigma_i^{\mathsf{T}}(\mathtt{I} - \Sigma_i^{-}\Sigma_i) = \Sigma_i(\mathtt{I} - \Sigma_i^{-}\Sigma_i) = \mathbf{0}$$

Therefore

$$\mathbf{f}_i^{\mathsf{T}}\boldsymbol{\lambda}_i = \mathbf{f}_i^{\mathsf{T}}\Sigma_i^{-}\mathbf{f}_i$$

which, in view of (2.15), can be rewritten as

$$(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}}\Lambda_{\mathbf{x}_i}^{-1}(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}) = \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})^{-}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}). \tag{2.17}$$

Note that the above equality is only approximate so that each particular g-inverse $\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})^{-}$ leads to a separate first-order approximation of $(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^{\mathsf{T}}\Lambda_{\mathbf{x}_i}^{-1}(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})$, with any pair of such approximations differing by second-order entities.

In a last step we substitute $\Sigma(\check{\mathbf{x}}_i, \boldsymbol{\theta})^-$ for $\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})^-$ in (2.17). For this to be meaningful, it is necessary to assume that $\Sigma(\check{\mathbf{x}}_i, \boldsymbol{\theta})^-$ be close to $\Sigma(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})^-$. Importantly, this requirement is not automatically satisfied. It can be fulfilled by either selecting a g-inverse which is continuous or by regularising appropriately the g-inverse at hand to make it effectively continuous. More details on both approaches will be given later. When Σ^- is chosen to be effectively continuous at $(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})$, the final approximation reads

$$(\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}})^\mathsf{T} \Lambda_{\mathbf{x}_i}^{-1} (\mathbf{x}_i - \check{\mathbf{x}}_i^{\boldsymbol{\theta}}) = \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T} \Sigma(\check{\mathbf{x}}_i, \boldsymbol{\theta})^- \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}).$$

It is clear from the above discussion that a meaninful AML cost function is given by

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T} [\partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} \partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}]^- \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}),$$

provided that Σ^- is effectively continuous at all the $(\check{\mathbf{x}}_i^{\boldsymbol{\theta}}, \boldsymbol{\theta})$.

2.2.5 Equivalent form of the AML cost function

The AML cost function can efficiently be optimised by two closely related procedures. For the presentation of these techniques, an alternative expression for J_{AML} is needed, and this will be given next.

Recall that, for an $m \times n$ matrix $\mathtt{A} = [a_{ij}]$ and a $p \times q$ matrix $\mathtt{B}$, the *Kronecker product* [54] of $\mathtt{A}$ and $\mathtt{B}$, $\mathtt{A} \otimes \mathtt{B}$, is the $mp \times nq$ matrix

$$\mathtt{A} \otimes \mathtt{B} = \begin{bmatrix} a_{11}\mathtt{B} & \ldots & a_{1n}\mathtt{B} \\ \vdots & & \vdots \\ a_{m1}\mathtt{B} & \ldots & a_{mn}\mathtt{B} \end{bmatrix}.$$

For an $m \times n$ matrix $\mathtt{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ with $\mathbf{a}_j$ the j-th column vector of length m, let $\mathrm{vec}(\mathtt{A})$ denote the *vectorisation* of $\mathtt{A}$, that is the column vector of length mn defined by $\mathrm{vec}(\mathtt{A}) = [\mathbf{a}_1^\mathsf{T}, \ldots, \mathbf{a}_n^\mathsf{T}]^\mathsf{T}$. Since (2.2) can be reformulated as

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = (\mathtt{I}_{m \times m} \otimes \boldsymbol{\theta}^\mathsf{T}) \,\mathrm{vec}(\mathtt{U}), \tag{2.18}$$

with $\mathrm{vec}(\mathtt{U}) = [\mathbf{u}_1^\mathsf{T}, \ldots, \mathbf{u}_m^\mathsf{T}]^\mathsf{T}$, we have

$$\partial_{\mathbf{x}} \mathbf{f} = (\mathtt{I}_{m \times m} \otimes \boldsymbol{\theta}^\mathsf{T}) \partial_{\mathbf{x}} \mathrm{vec}(\mathtt{U}),$$

where $\mathtt{I}_{m \times m}$ is the $m \times m$ identity matrix and $\partial_{\mathbf{x}} \mathrm{vec}(\mathtt{U}) = [(\partial_{\mathbf{x}} \mathbf{u}_1)^\mathsf{T}, \ldots, (\partial_{\mathbf{x}} \mathbf{u}_m)^\mathsf{T}]^\mathsf{T}$. Consequently, for each $i = 1, \ldots, n$,

$$\partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} (\partial_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}))^\mathsf{T} = (\mathtt{I}_{m \times m} \otimes \boldsymbol{\theta}^\mathsf{T}) \mathtt{B}_i (\mathtt{I}_{m \times m} \otimes \boldsymbol{\theta}),$$

where

$$\mathtt{B}_i = [\partial_{\mathbf{x}} \mathrm{vec}(\mathtt{U}(\mathbf{x}))]_{\mathbf{x}=\mathbf{x}_i} \Lambda_{\mathbf{x}_i} [(\partial_{\mathbf{x}} \mathrm{vec}(\mathtt{U}(\mathbf{x})))^{\mathsf{T}}]_{\mathbf{x}=\mathbf{x}_i}.$$

Now, with $\mathtt{U}_i = \mathtt{U}(\mathbf{x}_i)$,

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \boldsymbol{\theta}^{\mathsf{T}} \mathtt{U}_i \left[(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}}) \mathtt{B}_i (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}) \right]^{-1} \mathtt{U}_i^{\mathsf{T}} \boldsymbol{\theta}$$

and this is the required expression for J_{AML}.

2.2.6 Optimisation of the AML cost function

The minimiser $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ satisfies the necessary optimality condition

$$[\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n)]_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}} = \mathbf{0}^{\mathsf{T}} \tag{2.19}$$

with $\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}$ the row vector of the partial derivatives of J_{AML} with respect to $\boldsymbol{\theta}$. We term this the *variational equation*. With the aid of (2.18), it can be shown that

$$[\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n)]^{\mathsf{T}} = 2\mathtt{X}_{\boldsymbol{\theta}} \boldsymbol{\theta}, \tag{2.20}$$

where $\mathtt{X}_{\boldsymbol{\theta}}$ is an $l \times l$ symmetric matrix given by

$$\mathtt{X}_{\boldsymbol{\theta}} = \mathtt{M}_{\boldsymbol{\theta}} - \mathtt{N}_{\boldsymbol{\theta}}, \tag{2.21a}$$

$$\mathtt{M}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \mathtt{U}_i \Sigma_i^{-1} \mathtt{U}_i^{\mathsf{T}}, \tag{2.21b}$$

$$\mathtt{N}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} (\boldsymbol{\eta}_i^{\mathsf{T}} \otimes \mathtt{I}_{l\times l}) \mathtt{B}_i (\boldsymbol{\eta}_i \otimes \mathtt{I}_{l\times l}), \tag{2.21c}$$

$$\mathtt{B}_i = \partial_{\mathbf{x}_i} \mathrm{vec}(\mathtt{U}_i) \Lambda_{\mathbf{x}_i} [\partial_{\mathbf{x}_i} \mathrm{vec}(\mathtt{U}_i)]^{\mathsf{T}}, \tag{2.21d}$$

$$\Sigma_i = (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}}) \mathtt{B}_i (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}), \tag{2.21e}$$

$$\boldsymbol{\eta}_i = \Sigma_i^{-1} \mathtt{U}_i^{\mathsf{T}} \boldsymbol{\theta}. \tag{2.21f}$$

The variational equation (2.19) can accordingly be rewritten as

$$\mathtt{X}_{\boldsymbol{\theta}} \boldsymbol{\theta} = \mathbf{0}, \tag{2.22}$$

where the evaluation at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ is dropped for clarity. In this form the variational equation will serve as a basis for isolating $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$.

Two iterative methods for solving this equation have recently been developed. The *fundamental numerical scheme* (FNS), originally designed for optimisation of a single-objective AML cost

function [15], exploits the fact that a vector $\boldsymbol{\theta}$ satisfies (2.22) if and only if it is a solution of the *ordinary* eigenvalue problem

$$\mathtt{X}_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda\boldsymbol{\xi}$$

corresponding to the eigenvalue $\lambda = 0$. This suggests an iterative method for solving (2.22) whereby if $\boldsymbol{\theta}_k$ is a current approximate solution, then an updated solution $\boldsymbol{\theta}_{k+1}$ is a vector chosen from that eigenspace of $\mathtt{X}_{\boldsymbol{\theta}_k}$ which most closely approximates the null-space of $\mathtt{X}_{\boldsymbol{\theta}}$; this eigenspace is, of course, the one corresponding to the eigenvalue closest to zero in absolute value. The algorithm can be seeded with an estimate produced by some non-iterative method such as ALS or GTLS presented in Sections 2.2.1 and 2.2.2 respectively. The overall procedure is summarised in Algorithm 3, assuming an ALS initialisation.

Algorithm 3 Fundamental Numerical Scheme I

Steps to compute an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$ of the minimiser of J_{AML} given in (2.11) :

1. Find an initial estimate $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$ and set $k = 0$.
2. Compute the matrix $\mathtt{X}_{\boldsymbol{\theta}_k}$ as per (2.21a).
3. Take $\boldsymbol{\theta}_{k+1}$ as the normalised eigenvector of $\mathtt{X}_{\boldsymbol{\theta}_k}$ associated with the eigenvalue closest to zero (in absolute value).
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

In view of the representation $\mathtt{X}_{\boldsymbol{\theta}} = \mathtt{M}_{\boldsymbol{\theta}} - \mathtt{N}_{\boldsymbol{\theta}}$, equation (2.22) can also be written as

$$\mathtt{M}_{\boldsymbol{\theta}}\boldsymbol{\theta} = \mathtt{N}_{\boldsymbol{\theta}}\boldsymbol{\theta}. \tag{2.23}$$

The *heteroscedastic errors-in-variables* (HEIV) scheme [55, 59] is based upon the observation that a vector $\boldsymbol{\theta}$ satisfies (2.23) if and only if it is a solution of the *generalised* eigenvalue problem

$$\mathtt{M}_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda\mathtt{N}_{\boldsymbol{\theta}}\boldsymbol{\xi}$$

corresponding to the eigenvalue $\lambda = 1$. Given a current approximate solution $\boldsymbol{\theta}_k$, HEIV in its basic form takes for an updated solution $\boldsymbol{\theta}_{k+1}$ a normalised eigenvector of the eigenvalue problem $\mathtt{M}_{\boldsymbol{\theta}_k}\boldsymbol{\xi} = \lambda\mathtt{N}_{\boldsymbol{\theta}_k}\boldsymbol{\xi}$ corresponding to the eigenvalue closest to 1. Again the iterative process can be seeded with $\widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$.

FNS and the HEIV schemes are locally convergent—to work they require an initial estimate sufficiently close to a solution of the equation underlying a particular method. More stable versions of the algorithms, able to cope with less accurate initial estimates, result from selecting at

each iteration the eigenvector corresponding to the smallest eigenvalue instead of the eigenvector corresponding to the eigenvalue closest to 0 in the case of FNS, and closest to 1 in the case of HEIV. Typically, the minimal eigenvalue computed after a few iterations is also the closest to 0 or 1 depending on the method, and once this stage is reached the modified algorithms act exactly as their original versions. Without these modifications, the schemes may exhibit slow convergence or even divergence.

2.3 Coping with linear dependencies

For some models, like those pertaining to a homography or a camera matrix, the system (2.1) is linear in $\boldsymbol{\theta}$ and consists of *linearly dependent* equations. In this situation, some gradients of the sub-constraint functions can be expressed as a linear combination of other gradients. Figure 2.2 gives an illustration for two gradients.

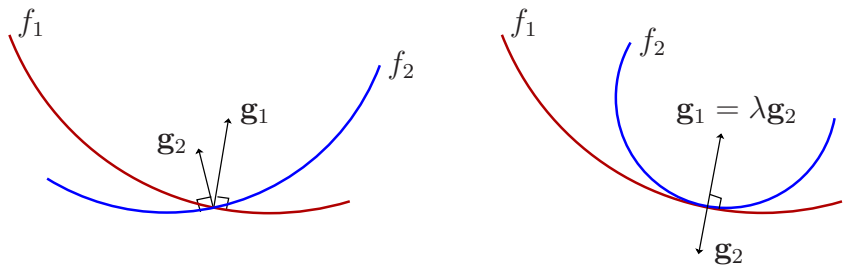
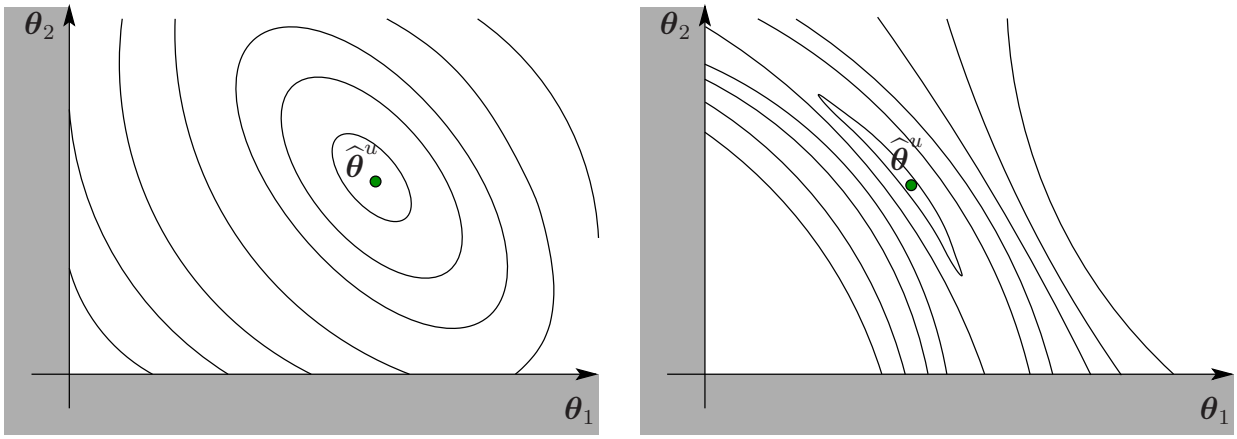


Figure 2.2. Intersection of sub-constraint functions f_1 and f_2 in $\mathbb{R}^2$ when both gradients $\mathbf{g}_1$ and $\mathbf{g}_2$ are (a) linearly independent; (b) proportional. In the latter case this implies that there exists a non-zero scalar λ such that $\mathbf{g}_1 = \lambda\mathbf{g}_2$.

It turns out that for any such model the matrices $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}$, the inverses of which enter the AML cost function, are ill-conditioned and the AML estimates of $\boldsymbol{\theta}$ are, as a rule, inaccurate when noise in the data is small. The purpose of this section is to demonstrate that this deficiency can be overcome if either the cost function or its Jacobian is suitably modified. In general, many modifications are possible. For example, the constraint vector $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ can be curtailed, in a multitude of ways, so that (2.1) becomes a linearly independent system of equations—a system giving rise to a well-behaved AML cost function. Another possibility is to replace the inverses of the $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}$ by truncated versions of the Moore–Penrose generalised inverses. One of the contributions of this work is putting various forms of cost-function modification within a unifying framework. A key concept in this context is a generalised inverse of a matrix. It emerges that any useful variant of J_{AML} is the result of replacing the inverses of the $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}$ by suitable generalised inverses.

2.3.1 Problem description

We now consider a model in which, for any data point $\mathbf{x}$, the carrier matrix $\mathtt{U}(\mathbf{x})$ is not of full rank, having linearly dependent columns. Examples of such a model include the homography and camera matrix models, where in each case the carrier matrix has three linearly dependent components, with only pairs of columns being linearly independent. As we shall see shortly, the problem with any non-full-rank model is that, for small measurement errors in the data, the AML cost function is a poor approximation of the ML cost function and, consequently, the AML estimates of the parameter vector may be unreliable. The difficulty in finding an estimate in this case is depicted in Figure 2.3.



(a) (b)

Figure 2.3. Contours of a quadratic in $\mathbb{R}^2$ when the problem is: (a) well-conditioned; (b) ill-conditioned. The valley around $\widehat{\boldsymbol{\theta}}^u$ becomes longer and narrower as the problem becomes ill-conditioned. This creates some trouble for any method that tries to find the unconstrained minimiser.

A fundamental consequence of the rank deficiency of the carrier matrix is that, for each *ideal* data point $\overline{\mathbf{x}}$ and each parameter vector $\overline{\boldsymbol{\theta}}$ matching $\overline{\mathbf{x}}$, the matrix $\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})^{\mathsf{T}}$ is singular. Indeed, the linear dependence of the columns of $\mathtt{U}(\mathbf{x})$ means that, for each $\mathbf{x}$, there exists $\boldsymbol{\alpha}(\mathbf{x}) = [\alpha_1(\mathbf{x}), \ldots \alpha_m(\mathbf{x})]^{\mathsf{T}} \neq \mathbf{0}$ such that $\sum_{i=1}^{m} \alpha_i(\mathbf{x})\mathbf{u}_i(\mathbf{x}) = \mathbf{0}$, or, more compactly, $\mathtt{U}(\mathbf{x})\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{0}$. It follows that, for each $\boldsymbol{\theta}$,

$$\boldsymbol{\alpha}(\mathbf{x})^{\mathsf{T}}\mathtt{U}(\mathbf{x})^{\mathsf{T}}\boldsymbol{\theta} = \boldsymbol{\alpha}(\mathbf{x})^{\mathsf{T}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta}) = 0$$

whence, upon differentiating with respect to $\mathbf{x}$,

$$\partial_{\mathbf{x}}\boldsymbol{\alpha}(\mathbf{x})^{\mathsf{T}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta}) + \partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{0}. \tag{2.24}$$

Combining this with $\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}}) = \mathbf{0}$ yields

$$\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})^{\mathsf{T}}\boldsymbol{\alpha}(\overline{\mathbf{x}}) = \mathbf{0}.$$

Consequently, $\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})^{\mathsf{T}}$ has $\boldsymbol{\alpha}(\overline{\mathbf{x}})$ in its null-space and as such is singular.

Now, if $\mathbf{x}$ is a noisy variant of $\overline{\mathbf{x}}$ and $\boldsymbol{\theta}$ is close to $\overline{\boldsymbol{\theta}}$, then, as a rule, $\mathbf{f}(\mathbf{x},\boldsymbol{\theta})$ in (2.24) will not vanish and $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}$ will be full-rank (invertible, to be more precise). However, the singularity of $\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})^{\mathsf{T}}$ will influence the behaviour of $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}$. If noisy data points are close to ideal data points, which typically happens when the noise level is low, and if the parameter vector is close to the parameter vector matching the ideal data points, then the latter matrix will be ill-conditioned. In particular, for data exhibiting small errors, J_{AML} will be a poor approximation of J_{ML}.

2.3.2 Sub-constraint vectors and generalised inverses

One way to overcome the deficiency stemming from the presence of ill-conditioned matrices in the expression for J_{AML} is to form a shorter constraint vector $\mathbf{f}'(\mathbf{x},\boldsymbol{\theta}) = \mathtt{U}(\mathbf{x})'^{T}\boldsymbol{\theta}$, where $\mathtt{U}(\mathbf{x})'$ is formed by a linearly independent set of columns of $\mathtt{U}(\mathbf{x})$. The number of columns of $\mathtt{U}(\mathbf{x})'$, or equivalently the length of the sub-constraint vector $\mathbf{f}'(\mathbf{x},\boldsymbol{\theta})$, is set to be equal to the column rank of $\mathtt{U}(\mathbf{x})$. Any sub-constraint vector of this form will be termed a *basic* sub-constraint vector. If $\mathbf{f}'(\mathbf{x},\boldsymbol{\theta})$ is a basic sub-constraint vector, then, under favourable conditions, $\partial_{\mathbf{x}}\mathbf{f}'(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}'(\overline{\mathbf{x}},\overline{\boldsymbol{\theta}})^{\mathsf{T}}$ will be invertible. Various basic sub-constraint vectors can be generated from a full constraint vector. For example, for the homography and camera matrix models, three basic sub-constraint vectors can be formed, each comprising a pair of components of the original constraint vector. Given a basic sub-constraint vector $\mathbf{f}'(\mathbf{x},\boldsymbol{\theta})$, one can define a modified AML cost function by setting

$$J_{\mathrm{AML}}(\boldsymbol{\theta};\mathbf{x}_1,\ldots,\mathbf{x}_n) = \sum_{i=1}^{n}\mathbf{f}'(\mathbf{x}_i,\boldsymbol{\theta})^{\mathsf{T}}[\partial_{\mathbf{x}}\mathbf{f}'(\mathbf{x}_i,\boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}'(\mathbf{x}_i,\boldsymbol{\theta})^{\mathsf{T}}]^{-1}\mathbf{f}'(\mathbf{x}_i,\boldsymbol{\theta}). \tag{2.25}$$

As we shall show next, the multitude of cost functions arising from different basic sub-constraint vectors can be viewed from a unifying perspective once these functions are characterised in terms of a *generalised inverse* of a matrix [11].

Suppose that the constraint vector $\mathbf{f}$ is partitioned as $[\mathbf{f}'^{\mathsf{T}},\mathbf{f}''^{\mathsf{T}}]^{\mathsf{T}}$, where $\mathbf{f}' = [f_1,\ldots,f_r]$ is a basic sub-constraint vector and $\mathbf{f}'' = [f_{r+1},\ldots,f_m]^{\mathsf{T}}$ is the complementary vector. Then associated with $\mathbf{f}'$ there is a g-inverse of $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}$, namely

$$[\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}]_r^- = \begin{bmatrix}[\partial_{\mathbf{x}}\mathbf{f}'(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}'(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}]^{-1} & 0\\ 0 & 0\end{bmatrix}. \tag{2.26}$$

In terms of this g-inverse, the AML cost function based on $\mathbf{f}'$, as given in (2.25), can be written as

$$J_{\mathrm{AML}}(\boldsymbol{\theta};\mathbf{x}_1,\ldots,\mathbf{x}_n) = \sum_{i=1}^{n}\mathbf{f}(\mathbf{x}_i,\boldsymbol{\theta})^{\mathsf{T}}[\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i,\boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i,\boldsymbol{\theta})^{\mathsf{T}}]_r^-\mathbf{f}(\mathbf{x}_i,\boldsymbol{\theta}).$$

Note a similarity with formula (2.11) —the standard inverse appearing there is now replaced by the g-inverse associated with $\mathbf{f}'$.

Different AML cost functions can be obtained by selecting different sub-constraint vectors. To wit, let σ be a permutation of $\{1, 2, \ldots, m\}$ and let $\mathbf{f}_\sigma$ be the result of rearranging the entries of $\mathbf{f}$ according to σ, that is, $\mathbf{f}_\sigma = [f_{\sigma(1)}, \ldots, f_{\sigma(m)}]^\mathsf{T}$. Suppose that in the representation $\mathbf{f}_\sigma = [\mathbf{f}_\sigma'^\mathsf{T}, \mathbf{f}_\sigma''^\mathsf{T}]^\mathsf{T}$, where $\mathbf{f}_\sigma' = [f_{\sigma(1)}, \ldots, f_{\sigma(r)}]^\mathsf{T}$ and $\mathbf{f}_\sigma'' = [f_{\sigma(r+1)}, \ldots, f_{\sigma(m)}]^\mathsf{T}$, $\mathbf{f}_\sigma'$ is a basic sub-constraint vector. Then the following AML cost function can now be defined

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}_\sigma(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T} [\partial_\mathbf{x} \mathbf{f}_\sigma(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} \partial_\mathbf{x} \mathbf{f}_\sigma(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}]_r^- \mathbf{f}_\sigma(\mathbf{x}_i, \boldsymbol{\theta}).$$

Given that $\mathbf{f}_\sigma = \mathtt{P}_\sigma \mathbf{f}$, where $\mathtt{P}_\sigma$ is the $m \times m$ *permutation matrix* defined by $\mathtt{P}_\sigma = [\mathbf{e}_{\sigma(1)}^\mathsf{T}, \ldots, \mathbf{e}_{\sigma(m)}^\mathsf{T}]^\mathsf{T}$, with $\mathbf{e}_i$ the i-th row vector of $\mathtt{I}_{m \times m}$, the above formula can be rewritten as

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T} [\partial_\mathbf{x} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} \partial_\mathbf{x} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}]_{r,\sigma}^- \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}), \tag{2.27}$$

where

$$[\partial_\mathbf{x} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \Lambda_\mathbf{x} \partial_\mathbf{x} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^\mathsf{T}]_{r,\sigma}^- = \mathtt{P}_\sigma^\mathsf{T} [\mathtt{P}_\sigma \partial_\mathbf{x} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \Lambda_\mathbf{x} \partial_\mathbf{x} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^\mathsf{T} \mathtt{P}_\sigma^\mathsf{T}]_r^- \mathtt{P}_\sigma \tag{2.28}$$

is the g-inverse associated with $\mathbf{f}_\sigma'$. Note that, similarly as before, the g-inverse in the AML cost function (which is now based on $\mathbf{f}_\sigma'$) plays the role of the inverse in the formula for the standard AML cost function. Note also that J_{AML} given in (2.27) is differentiable. This follows from (2.26), (2.28) and the fact that the mapping sending an invertible matrix to its inverse is differentiable. The differentiability property guarantees that the function can be optimised by using FNS or HEIV.

In summary, a first solution to bypass problems engendered by the linear dependency of some components of the objective vector $\mathbf{f}$ is to evolve a cost function based on a sub-constraint vector $\mathbf{f}'$. Adapting the original FNS (Algorithm 3) to the present context leads to the procedure outlined in Algorithm 4.

2.3.3 Enter all constraints

An obvious aesthetical, though not fundamental from the point of view of theory, shortcoming of the AML cost functions based on sub-constraint vectors is the asymmetry with which various components of the original constraint vector are treated. Each of these cost functions uses some components and ignore others. It would be more desirable to see all components incorporated into a valid cost function. One conceivable way to achieve this is to use the pseudo-inverse and let

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T} [\partial_\mathbf{x} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}) \Lambda_{\mathbf{x}_i} \partial_\mathbf{x} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}]^+ \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}). \tag{2.29}$$

Algorithm 4 Fundamental Numerical Scheme II

Steps to compute an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$ of the minimiser of J_{AML} given in (2.25) :

1. Find an initial estimate $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$ and set $k = 0$.
2. Compute the matrix $\mathtt{X}_{\boldsymbol{\theta}_k}$ based on a chosen $\mathbf{f}'$ vector instead of the full $\mathbf{f}$ vector.
3. Take $\boldsymbol{\theta}_{k+1}$ as the normalised eigenvector of $\mathtt{X}_{\boldsymbol{\theta}_k}$ associated with the eigenvalue closest to zero (in absolute value).
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

This, however, is not a satisfactory choice—the proposed function is prone to numerical instability, as we now explain.

The point is that the pseudo-inverse may fail to be continuous when the limit matrix is not of full rank. A simple example illustrating this phenomenon is this: Let $\mathtt{T}_n = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1/n \end{smallmatrix}\right]$ and $\mathtt{T} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. Then $\mathtt{T}_n^+ = \left[\begin{smallmatrix} 1 & 0 \\ 0 & n \end{smallmatrix}\right]$ and $\mathtt{T} = \mathtt{T}^+ = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, so $\mathtt{T}_n \to \mathtt{T}$ while $\mathtt{T}_n^+$ diverges as n increases. Note that $\mathtt{T}$ is rank deficient, while all the $\mathtt{T}_n$ are full rank. Given that $\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}}, \overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}}, \overline{\boldsymbol{\theta}})^{\mathsf{T}}$ is singular, the lack of continuity of the pseudo-inverse implies that, for small noise, $[\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^{\mathsf{T}}]^+$ is ill-conditioned. This drawback can be overcome if the formula (2.29) is modified to read

$$J_{\mathrm{AML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}[\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}]_r^+\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta}). \tag{2.30}$$

Here r is the column rank of $\mathtt{U}(\mathbf{x})$ and $\mathtt{A}_r^+$ denotes the *r-truncated pseudo-inverse* of the $m \times m$ matrix $\mathtt{A}$ defined as follows: if $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^{\mathsf{T}}$ is the SVD of $\mathtt{A}$, with $\mathtt{D} = \mathrm{diag}(d_1, \ldots, d_m)$, then $\mathtt{A}_r = \mathtt{U}\mathtt{D}_r\mathtt{V}^{\mathsf{T}}$ with $\mathtt{D}_r = \mathrm{diag}(d_1, \ldots, d_r, 0, \ldots, 0)$ is the r-truncated SVD of $\mathtt{A}$, and $\mathtt{A}_r^+ = \mathtt{V}\mathtt{D}_r^+\mathtt{U}^{\mathsf{T}}$ with $\mathtt{D}_r^+ = \mathrm{diag}(d_1^+, \ldots, d_r^+, 0, \ldots, 0)$, where $d_i^+ = d_i^{-1}$ when $d_i \neq 0$ and $d_i^+ = 0$ otherwise, is the pseudo-inverse of $\mathtt{A}_r$. The truncated SVD forces the *exact* rank of $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^{\mathsf{T}}$ to be reduced to the more adequate *numerical* rank, this being defined as the rank of the underlying matrix $\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}}_i, \overline{\boldsymbol{\theta}})\Lambda_{\overline{\mathbf{x}}}\partial_{\mathbf{x}}\mathbf{f}(\overline{\mathbf{x}}_i, \overline{\boldsymbol{\theta}})^{\mathsf{T}}$. The rank suppression enforces continuity of the g-inverse and is an example of a *regularisation* technique [21].

The modified AML cost function, as specified in (2.30), is still not entirely satisfactory. It is not obvious whether it is differentiable and, even if it is, whether its Jacobian $\partial_{\boldsymbol{\theta}}J_{\mathrm{AML}}$ can be calculated straightforwardly. One troublesome fact is that the singular values of a matrix are not differentiable functions of the matrix. For example, if $\mathtt{A} = a$ is a 1×1 matrix, then its singular value is $|a|$ and the function $a \mapsto |a|$ is not differentiable at 0. To bypass this difficulty, we choose, following Kanatani [49], to regularise the Jacobian rather than the function itself. The

key ingredients of the regularised Jacobian take the form

$$\begin{aligned}
\mathtt{M}_{\boldsymbol{\theta}} &= \sum_{i=1}^{n} \mathtt{U}_i \left[(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{B}_i(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta})\right]_r^+ \mathtt{U}_i^{\mathsf{T}}, \\
\mathtt{N}_{\boldsymbol{\theta}} &= \sum_{i=1}^{n} (\boldsymbol{\theta}^{\mathsf{T}}\mathtt{U}_i \left[(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{B}_i(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta})\right]_r^+ \otimes \mathtt{I}_l) \\
&\quad \times \mathtt{B}_i([(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{B}_i(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta})]_r^+ \mathtt{U}_i^{\mathsf{T}}\boldsymbol{\theta} \otimes \mathtt{I}_l).
\end{aligned} \tag{2.31}$$

Now, with $\mathtt{X}_{\boldsymbol{\theta}} = \mathtt{M}_{\boldsymbol{\theta}} - \mathtt{N}_{\boldsymbol{\theta}}$, the AML estimate of $\boldsymbol{\theta}$, $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$, based on the full constraint vector can be defined as the solution to $\mathtt{X}_{\boldsymbol{\theta}}\boldsymbol{\theta} = \mathbf{0}$ to be found in the vicinity of $\widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$. The computation of this estimate can again be done with the aid of either FNS or HEIV.

Recapitulating the findings, we propose a second method to tackle objective functions with linearly dependent components. The method involves employing a truncated pseudo-inverse of rank r suitable for the problem at hand. A modification of the original FNS (Algorithm 3) is presented next.

Algorithm 5 Fundamental Numerical Scheme III

Steps to compute an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$ of the minimiser of J_{AML} given in (2.30) :

1. Find an initial estimate $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}}$, set $k = 0$ and the rank r.
2. Compute the matrix $\mathtt{X}_{\boldsymbol{\theta}_k}$ from the matrices $\mathtt{M}_{\boldsymbol{\theta}_k}$ and $\mathtt{N}_{\boldsymbol{\theta}_k}$ as per (2.31).
3. Take $\boldsymbol{\theta}_{k+1}$ as the normalised eigenvector of $\mathtt{X}_{\boldsymbol{\theta}_k}$ associated with the eigenvalue closest to zero (in absolute value).
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

2.3.4 General recipe

The previous considerations lead to the following general recipe for generating AML cost functions for models with linear dependencies. The starting point is the standard AML cost function as given in (2.11). This function is modified, the inverses of the matrices of the form $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})\Lambda_{\mathbf{x}}\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x},\boldsymbol{\theta})^{\mathsf{T}}$ being replaced by generalised inverses of some type. Various types of g-inverse can be used for the process. If a particular inverse chosen turns out not to be continuous, then it is next appropriately regularised. If now the resulting AML cost function is

differentiable, then a corresponding optimisation condition is evolved straightforwardly by calculating the Jacobian of the cost function and setting this Jacobian to zero. If the AML cost function is not differentiable, then an optimisation condition is derived based on a regularised Jacobian. Once the optimisation condition is set forth, the computation of AML parameter estimates proceeds by utilising either FNS or HEIV.

Algorithm 3 may be adapted such that the AML cost function is either the one given by (2.25) or (2.30). When a sub-constraint vector $\mathbf{f}'$ is selected, matrix $\mathtt{X}_{\boldsymbol{\theta}}$ is derived based on the matrix $\mathtt{U}(\mathbf{x})'$ associated with $\mathbf{f}'$. If function (2.30) is employed, then $\mathtt{X}_{\boldsymbol{\theta}}$ is formed from $\mathtt{M}_{\boldsymbol{\theta}}$ and $\mathtt{N}_{\boldsymbol{\theta}}$ as described in (2.31).

2.3.5 Equivalence problem

Given the multitude of AML cost functions and related optimisation conditions that can be obtained with use of various g-inverses, it is natural to ask how different AML estimates of $\boldsymbol{\theta}$ relate to one another. Bearing in mind the natural way in which g-inverses arise, one can expect that in terms of accuracy all AML estimates will be essentially equivalent. Chapters 3 and 5 on homography and trifocal tensor estimation will provide some experimental evidence in support of this claim.

A notable implication of this finding is that for the purpose of effective estimation, it suffices to use a simple, differentiable AML cost function such as any one based on a basic sub-constraint vector. More complicated AML cost functions, like the one involving a truncated pseudo-inverse, can safely be ignored.

2.4 Reduced fundamental numerical scheme

The previous sections showed that when the system $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}$ consists of linearly dependent equations two general techniques were available to regularise the associated AML objective function. One approach is to curtail the vector $\mathbf{f}$ so that it includes only linearly independent equations. The second option keeps the original vector $\mathbf{f}$, however, provides a modification of the AML function as follows. When the length m of the $\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})$ surpasses the common codimension r of the submanifolds of the form $\{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}\}$ with $\boldsymbol{\theta}$ representing *ideal* parameters that might have generated the data, the inverses $\Sigma(\mathbf{x}_i, \boldsymbol{\theta})^{-1}$ in expression (2.11) are replaced by the r-truncated pseudo-inverses $\Sigma(\mathbf{x}_i, \boldsymbol{\theta})_r^+$.

Although these modifications improve the robustness of FNS and broaden its range of applicability, the method may show some signs of instability for difficult sets of data. This section will present a reduced form of FNS, where only a subset of the total parameter vector is

estimated iteratively and the remaining parameters are recovered in a single step based on the result of the earlier iterative process. The reduced algorithm in effect replaces the original estimation problem with a couple of problems of lower dimension. The algorithm is an extension to the multi-objective setting of the reduced FNS in the single-objective case given in [16]. The process of dimension reduction leads to significant benefits. Compared to the full form, the reduced form of the algorithm requires a less accurate initial estimate and enjoys better convergence properties. While the work here is primarily concerned with FNS, the optimality condition which underlies the reduced form of this algorithm can readily be exploited to advance a reduced form of HEIV.

2.4.1 Problem reformulation

The starting point for the development of the new algorithm is the constraint function $\mathbf{f}$ given in (2.2). Suppose that the carrier matrix $\mathtt{U}(\mathbf{x})$ can be written as

$$\mathtt{U}(\mathbf{x}) = \begin{bmatrix} \mathtt{Z}(\mathbf{x}) \\ \mathtt{W} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1(\mathbf{x}) & \dots & \mathbf{z}_m(\mathbf{x}) \\ \mathbf{w}_1 & \dots & \mathbf{w}_m \end{bmatrix}, \tag{2.32}$$

where $\mathtt{Z}(\mathbf{x})$ is an $(l-m) \times m$ matrix that depends on $\mathbf{x}$ (a "pure measurement" matrix) and $\mathtt{W}$ is an $m \times m$ *invertible* matrix that does not depend on $\mathbf{x}$. Corresponding to this splitting of $\mathtt{U}(\mathbf{x})$, the parameter vector $\boldsymbol{\theta}$ will be subdivided as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix}, \tag{2.33}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ are vectors of length $l-m$ and m respectively. The partitioning of $\mathtt{U}(\mathbf{x})$ and $\boldsymbol{\theta}$ reflects that fact that some components of $\boldsymbol{\theta}$, considered as indeterminates, appear in each of the equations of (2.2) only with constant coefficients. The vector $\boldsymbol{\alpha}$ collects together those components of $\boldsymbol{\theta}$ that appear in (2.2) with pure constant coefficients. For each $i = 1, \dots, m$, the non-zero entries of the i-th column of $\mathtt{W}$ represent the constant coefficients of the components of $\boldsymbol{\alpha}$ in the i-th equation of (2.2). If, for instance, every equation of (2.2) has exactly one parameter with a unity coefficient, then, after reordering of the equations of (2.2) if necessary, it can be assumed that $\mathtt{W} = \mathtt{I}_{m \times m}$. Notice that matrix $\mathtt{W}$ does not depend on datum $\mathbf{x}$, so its derivative with respect to $\mathbf{x}$, $\partial_{\mathbf{x}} \mathrm{vec}(\mathtt{W})$, is an $ms \times k$ zero matrix. One of the motivations behind the new algorithm is to eliminate these non-informative rows from the estimation process. Upon

differentiation with respect to $\mathbf{x}$,

$$\mathrm{vec}(\mathtt{U}(\mathbf{x})) = \begin{bmatrix} \mathbf{z}_1(\mathbf{x}) \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{z}_m(\mathbf{x}) \\ \mathbf{w}_m \end{bmatrix}, \tag{2.34}$$

so one may see that the derivative matrix does not have all the elements of the last ms rows vanishing because some of these rows correspond to the derivatives of some $\mathbf{z}_i$'s. Therefore, matrix $\mathtt{U}$, as it stands in (2.32), is not an adequate representation.

The shortcoming of $\mathtt{U}$ may be overcome by introducing the $m \times l$ measurement matrix

$$\mathtt{V}(\mathbf{x}) = \mathtt{U}(\mathbf{x})^{\mathsf{T}}. \tag{2.35}$$

Now,

$$\mathrm{vec}(\mathtt{V}(\mathbf{x})) = \begin{bmatrix} \mathrm{vec}(\mathtt{Z}(\mathbf{x})^{\mathsf{T}}) \\ \mathrm{vec}(\mathtt{W}^{\mathsf{T}}) \end{bmatrix} \tag{2.36}$$

and differentiating this expression with respect to $\mathbf{x}$ gives a matrix with the last ms rows vanishing, as desired. The adoption of this new measurement matrix means a reformulation of the objective function $\mathbf{f}$ defined in (2.2) and associated expressions. This is shown next.

The particular relationship $\mathtt{V} = \mathtt{U}^{\mathsf{T}}$ permits to write

$$\mathrm{vec}(\mathtt{U}) = \mathtt{K}_{ml}\ \mathrm{vec}(\mathtt{V}), \tag{2.37}$$

where $\mathtt{K}_{ml}$ denotes the square *commutation matrix* of size $ml \times ml$ [57]. It follows naturally that

$$\mathbf{f} = (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\,\mathrm{vec}(\mathtt{U}) = (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{K}_{ml}\,\mathrm{vec}(\mathtt{V}).$$

By property of commutation matrices,

$$(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{K}_{ml} = (\boldsymbol{\theta}^{\mathsf{T}} \otimes \mathtt{I}_{m\times m}), \tag{2.38}$$

hence

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = (\boldsymbol{\theta}^{\mathsf{T}} \otimes \mathtt{I}_{m\times m})\,\mathrm{vec}(\mathtt{V}). \tag{2.39}$$

If this identity is used in calculating $(\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}})^{\mathsf{T}}$ instead of (2.18), then the ensuing expression for $\mathtt{M}_{\boldsymbol{\theta}}$ will be identical with the one given in (2.21b), but the expression for $\mathtt{N}_{\boldsymbol{\theta}}$ will change as described next.

Inspired by the form of matrix $\mathtt{B}_i$ in (2.21d), let

$$\mathtt{B}_i^* = \partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{V}_i)\mathtt{\Lambda}_{\mathbf{x}_i}[\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{V}_i)]^{\mathsf{T}}, \tag{2.40}$$

where $\mathtt{V}_i$ is short for $\mathtt{V}(\mathbf{x}_i)$. The derivative of $\mathtt{U}$ with respect to $\mathbf{x} = [x^1, \ldots, x^k]$ takes the form

$$\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{U}) = [\partial_{x^1}\mathrm{vec}(\mathtt{U}), \ldots, \partial_{x^k}\mathrm{vec}(\mathtt{U})]$$

and applying rule (2.37) to each of its columns, we deduce that

$$\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{U}) = \mathtt{K}_{ml}\,\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{V}).$$

Starting from (2.21e) and using the previous identity, we can write

$$\Sigma_i = (\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}^{\mathsf{T}})\mathtt{K}_{ml}\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{V}_i)\Lambda_{\mathbf{x}_i}[\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{V}_i)]^{\mathsf{T}}\mathtt{K}_{ml}^{\mathsf{T}}(\mathtt{I}_{m\times m} \otimes \boldsymbol{\theta}).$$

Simplifying with (2.38) and (2.40) gives

$$\Sigma_i = (\boldsymbol{\theta}^{\mathsf{T}} \otimes \mathtt{I}_{m\times m})\mathtt{B}_i^*(\boldsymbol{\theta} \otimes \mathtt{I}_{m\times m}) \tag{2.41}$$

for each $i = 1, \ldots, n$. Recalling (2.21c) and given the fact that

$$(\boldsymbol{\eta}_i^{\mathsf{T}} \otimes \mathtt{I}_{l\times l})\mathtt{K}_{ml} = (\mathtt{I}_{l\times l} \otimes \boldsymbol{\eta}_i^{\mathsf{T}}),$$

it follows that

$$\mathtt{N}_{\boldsymbol{\theta}} = (\mathtt{I}_{l\times l} \otimes \boldsymbol{\eta}_i^{\mathsf{T}})\mathtt{B}_i^*(\mathtt{I}_{l\times l} \otimes \boldsymbol{\eta}_i). \tag{2.42}$$

The matrix $\mathtt{X}_{\boldsymbol{\theta}}$ composed from $\mathtt{M}_{\boldsymbol{\theta}}$ and $\mathtt{N}_{\boldsymbol{\theta}}$ based on $\mathtt{V}$ gives a neat formula for the gradient of the cost function proposed by Matei and Meer [62], where the "scatter matrix" is $\mathtt{M}_{\boldsymbol{\theta}} = \sum_{i=1}^n \mathtt{V}_i^{\mathsf{T}}\Sigma_i^{-1}\mathtt{V}_i$ and the "weighted covariance matrix" is $\mathtt{N}_{\boldsymbol{\theta}} = \sum_{i=1}^n (\mathtt{I}_l \otimes \boldsymbol{\eta}_i^{\mathsf{T}})\mathtt{B}_i^*(\mathtt{I}_l \otimes \boldsymbol{\eta}_i)$. While equation (2.21c) for $\mathtt{N}_{\boldsymbol{\theta}}$ arises more frequently in the literature, equation (2.42) will prove more useful in what follows.

2.4.2 Reduced variational equation

We shall now present a system of two equations that jointly are equivalent to the variational equation (2.22). One of these equations involves only $\boldsymbol{\mu}$ and can be solved separately, and the other expresses $\boldsymbol{\alpha}$ in terms of $\boldsymbol{\mu}$. We begin by noting that, in view of (2.36),

$$\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{V}(\mathbf{x})) = \begin{bmatrix} \partial_{\mathbf{x}}\mathrm{vec}(\mathtt{Z}(\mathbf{x})^{\mathsf{T}}) \\ \mathtt{0}_{m^2\times k} \end{bmatrix}.$$

Hence, for each $i = 1, \ldots, n$, the $lm \times lm$ matrix $\mathtt{B}_i^*$ can be represented as

$$\mathtt{B}_i^* = \begin{bmatrix} \mathtt{B}_i' & \mathtt{0}_{(l-m)m\times m^2} \\ \mathtt{0}_{m^2\times(l-m)m} & \mathtt{0}_{m^2\times m^2} \end{bmatrix}, \tag{2.43}$$

where $\mathtt{B}'_i$ is the $(l-m)m \times (l-m)m$ matrix given by

$$\mathtt{B}'_i = \partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{Z}_i^{\mathsf{T}})\mathtt{\Lambda}_{\mathbf{x}_i}[\partial_{\mathbf{x}_i}\mathrm{vec}(\mathtt{Z}_i^{\mathsf{T}})]^{\mathsf{T}}, \quad \mathtt{Z}_i = \mathtt{Z}(\mathbf{x}_i).$$

It is worth noting that this partitioning of $\mathtt{B}^*_i$, crucial to the subsequent development, results from taking (2.39) rather than (2.18) as a point of departure. It may be difficult to compute the matrix $\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{Z}(\mathbf{x})^{\mathsf{T}})$ directly. An alternative could be to start from an easier expression such as

$$\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{Z}(\mathbf{x})) = \begin{bmatrix} \partial_{\mathbf{x}}\mathbf{z}_1(\mathbf{x}) \\ \vdots \\ \partial_{\mathbf{x}}\mathbf{z}_m(\mathbf{x}) \end{bmatrix}$$

and use the identity

$$\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{Z}(\mathbf{x})^{\mathsf{T}}) = \mathtt{K}_{(l-m)m}\,\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{Z}(\mathbf{x})).$$

As is easily seen from (2.43), the null-space of each matrix $\mathtt{B}^*_i$ is spanned by the length lm canonical vector $[0,\ldots,0,\mathbf{1}]^{\mathsf{T}}$, where $\mathbf{1}$ denotes a length m^2 vector of ones. Consequently, $\mathtt{N}_{\boldsymbol{\theta}}$ is singular and solving the eigenvalue problem associated with (2.22) is susceptible to numerical instability [70]. This difficulty is overcome by reducing the eigenvector problem to a similar problem involving a positive definite matrix replacing $\mathtt{N}_{\boldsymbol{\theta}}$. Such a reduction relies on matrix $\mathtt{B}'_i$ and is exposed next.

Define an $m \times m$ matrix Σ'_i by

$$\Sigma'_i = (\boldsymbol{\mu}^{\mathsf{T}} \otimes \mathtt{I}_{m\times m})\mathtt{B}'_i(\boldsymbol{\mu} \otimes \mathtt{I}_{m\times m}). \tag{2.44}$$

Clearly, Σ'_i is positive semidefinite and depends only on the i-th element of data, its covariance $\mathtt{\Lambda}_{\mathbf{x}_i}$, and the parameter vector $\boldsymbol{\mu}$. Assume henceforth that each Σ'_i is positive definite and hence invertible. The inverses Σ'^{-1}_i can now be used as matricial weights to define a "centroid" of the $\mathtt{Z}_i$ as follows:

$$\tilde{\mathtt{Z}} = \sum_{i=1}^{n} \mathtt{Z}_i\Sigma'^{-1}_i \Big[\sum_{i=1}^{n} \Sigma'^{-1}_i\Big]^{-1}. \tag{2.45}$$

Here $\sum_{i=1}^n \Sigma'^{-1}_i$ is invertible because a sum of positive definite matrices is also positive definite. For each $i = 1,\ldots,n$, let

$$\mathtt{Z}'_i = \mathtt{Z}_i - \tilde{\mathtt{Z}} \tag{2.46}$$

be the i-th pure measurement vector relative to $\tilde{\mathtt{Z}}$. Letting

$$\boldsymbol{\eta}'_i = \Sigma'^{-1}_i \mathtt{Z}'^{\mathsf{T}}_i \boldsymbol{\mu}, \tag{2.47}$$

define the following $(l-m)\times(l-m)$ matrices

$$\mathtt{M}'_{\boldsymbol{\mu}} = \sum_{i=1}^{n} \mathtt{Z}'_i \Sigma_i'^{-1} \mathtt{Z}_i'^{\mathsf{T}}, \tag{2.48a}$$

$$\mathtt{N}'_{\boldsymbol{\mu}} = \sum_{i=1}^{n} (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i'^{\mathsf{T}}) \mathtt{B}'_i (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}'_i), \tag{2.48b}$$

$$\mathtt{X}'_{\boldsymbol{\mu}} = \mathtt{M}'_{\boldsymbol{\mu}} - \mathtt{N}'_{\boldsymbol{\mu}}. \tag{2.48c}$$

A fundamental result that can now be established is that $\boldsymbol{\theta} = [\boldsymbol{\mu}^{\mathsf{T}}, \boldsymbol{\alpha}^{\mathsf{T}}]^{\mathsf{T}}$ satisfies the variational equation (2.22) if and only if the following system of equations holds:

$$\mathtt{X}'_{\boldsymbol{\mu}} \boldsymbol{\mu} = \mathbf{0}, \tag{2.49a}$$

$$\boldsymbol{\alpha} = -(\tilde{\mathtt{Z}}\mathtt{W}^{-1})^{\mathsf{T}} \boldsymbol{\mu}. \tag{2.49b}$$

A proof can be found in Appendix A.1. The first equation constrains solely $\boldsymbol{\mu}$ and, therefore, can be solved separately. Once $\boldsymbol{\mu}$ is determined, $\boldsymbol{\alpha}$ is readily prescribed by the second equation. Of the two constraints, the first plays a leading role and will be termed the *reduced variational equation*. A salient feature of this equation is that the matrix $\mathtt{N}'_{\boldsymbol{\mu}}$, unlike $\mathtt{N}_{\boldsymbol{\theta}}$, is generically positive definite if $n \geq l-m$ in which case the eigenvalue problem associated with (2.49a) is nondegenerate.

With the *reduced AML cost function* defined by

$$J'_{\mathrm{AML}}(\boldsymbol{\mu}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \boldsymbol{\mu}^{\mathsf{T}} \mathtt{Z}'_i \Sigma_i'^{-1} \mathtt{Z}_i'^{\mathsf{T}} \boldsymbol{\mu}, \tag{2.50}$$

(2.49a) can be viewed as the variational equation for an optimiser of J'_{AML}. This formula also reveals that the new estimation problem has a smaller search space, of dimension $\dim(\boldsymbol{\mu}) < \dim(\boldsymbol{\theta})$. Interestingly, the $\boldsymbol{\mu}$-part of $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, which satisfies (2.49a) as $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ satisfies (2.22), turns out to be the minimiser of J'_{AML}, denoted $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, not just a critical point of J'_{AML}. Moreover, both J_{AML} and J'_{AML} attain a common minimum value at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ and $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, respectively (see Appendix A.2). One noteworthy consequence of this link is that the reduced AML cost function can be minimised by any algorithm and the result (a $\boldsymbol{\mu}$-vector) can first be fed into (2.49b) to produce a partial estimate (an $\boldsymbol{\alpha}$-vector) and further combined with this partial estimate (as per (2.33)) to produce the minimiser of the full AML cost function.

2.4.3 Algorithm details

A modification of FNS based on the reduced variational system (2.49a) and (2.49b) is the *reduced fundamental numerical scheme* (RFNS). Its steps are summarised in Algorithm 6.

Given a current approximate solution $\boldsymbol{\mu}_k$, the technique takes for an updated solution $\boldsymbol{\mu}_{k+1}$ a normalised eigenvector of $\mathtt{X}_{\boldsymbol{\mu}_k}$ corresponding to the smallest eigenvalue. Vector $\boldsymbol{\alpha}$ is calculated from the limiting value of $\boldsymbol{\mu}_{k+1}$ obtained upon convergence when solving (2.49a). The iterative process can be started by computing the ALS estimate, $\widehat{\boldsymbol{\mu}}_{\mathrm{ALS}}$.

Algorithm 6 Reduced Fundamental Numerical Scheme

Steps to compute an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}$ of the minimiser of J'_{AML} given in (2.50):

1. Find an initial estimate $\boldsymbol{\mu}_0 = \widehat{\boldsymbol{\mu}}_{\mathrm{ALS}}$ and set $k = 0$.
2. Compute the matrix $\mathtt{X}'_{\boldsymbol{\mu}_k}$ as per (2.48c).
3. Take $\boldsymbol{\mu}_{k+1}$ as the normalised eigenvector of $\mathtt{X}'_{\boldsymbol{\mu}_k}$ associated with the eigenvalue closest to zero (in absolute value).
4. If $\boldsymbol{\mu}_{k+1}$ is sufficiently close to $\boldsymbol{\mu}_k$, then terminate the procedure; otherwise increment k and return to step 2.
5. Compute $\boldsymbol{\alpha}$ as per (2.49b) using the limiting value $\boldsymbol{\mu}_{k+1}$ and the corresponding value $\tilde{\mathtt{Z}}(\boldsymbol{\mu}_{k+1})$ from the previous step. Set $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}} = [\boldsymbol{\mu}_{k+1}^{\mathsf{T}}, \boldsymbol{\alpha}^{\mathsf{T}}]^{\mathsf{T}}$.

In the case that the matrices Σ_i^{-1} are replaced by the matrices $(\Sigma_i)_r^+$ in the expression for J_{AML}, a similar change also affects the matrices $\mathtt{X}_{\boldsymbol{\theta}_k}$ of FNS. Moreover, as $\Sigma_i = \Sigma'_i$ for $i = 1, \ldots, n$ (see Appendix A.1), the $(\Sigma'_i)_r^+$ supercede the $\Sigma_i'^{-1}$ in the expression for J'_{AML} and in the $\mathtt{X}'_{\boldsymbol{\mu}_k}$ of RFNS.

Finally, we remark that a vector $\boldsymbol{\theta}$ satisfying (2.22) can alternatively be viewed as a solution of the *generalised* eigenvalue problem $\mathtt{M}_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda \mathtt{N}_{\boldsymbol{\theta}}\boldsymbol{\xi}$ corresponding to the eigenvalue $\lambda = 1$. This observation provides a starting point for the development of the HEIV scheme in both full and reduced versions [16]. Each version solves successively generalised eigenvalue problems analogous to the ordinary eigenvalue problems solved by a corresponding version of FNS.

2.5 Incorporating ancillary constraints

All of the estimators presented so far, with the exception of the ML one, aim to find a minimiser with no particular restriction on the parameter space other than the requirement that $\boldsymbol{\theta}$ has unit norm. In general these methods will return an estimate which does not satisfy the ancillary constraints. The remaining part of this chapter is dedicated to address this deficiency. Assuming that we have an unconstrained estimate, we propose several cost functions, each leading to a *post-hoc correction* scheme, to enforce the ancillary constraints.

The procedures we are about to present are all iterative and typically employed to solve (unconstrained) nonlinear optimisation problems. By providing these methods with an adequate cost function, they are used here to produce a constrained parameter vector.

2.5.1 Gauss-Newton correction

One way to obtain an estimate consistent with the ancillary constraints is to minimise the geometric distance between a sought vector $\boldsymbol{\theta}$ and the unconstrained estimate $\widehat{\boldsymbol{\theta}}^u_{\text{AML}}$. Under the assumption that $\boldsymbol{\theta}$ lies close to $\widehat{\boldsymbol{\theta}}^u_{\text{AML}}$, this problem can be formulated as a standard least-squares problem of the form

$$\arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \frac{1}{2}\,\mathbf{r}(\boldsymbol{\theta})^{\mathsf{T}}\mathbf{r}(\boldsymbol{\theta}) = \frac{1}{2}\,\|\mathbf{r}(\boldsymbol{\theta})\|^2,$$

where $\mathbf{r}(\boldsymbol{\theta}) = (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}^u_{\text{AML}})/\|\boldsymbol{\theta}\|$ is called the *residual function*. In numerous 3-D computer vision problems [61, 62], $\boldsymbol{\theta}$ is often parameterised through a nonlinear function g acting on a length-s vector $\boldsymbol{\beta}$ such that $\boldsymbol{\theta} = g(\boldsymbol{\beta})$. So, the residual vector is redefined as

$$\mathbf{r}(\boldsymbol{\beta}) = (g(\boldsymbol{\beta}) - \widehat{\boldsymbol{\theta}}^u_{\text{AML}})/\|g(\boldsymbol{\beta})\|, \tag{2.51}$$

and the minimisation criterion becomes a *nonlinear least-squares* problem

$$\arg\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{2}\,\|\mathbf{r}(\boldsymbol{\beta})\|^2. \tag{2.52}$$

Newton's method is a natural choice for solving the above nonlinear problem. From an initial value $\boldsymbol{\beta}_0$, the method constructs a sequence of vectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots$, such that at step k

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{p}_k, \tag{2.53}$$

where $\mathbf{p}_k$, a vector controlling the search direction, satisfies the *Newton's equations*

$$(\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k + \mathtt{B}_k)\mathbf{p}_k = -\mathtt{J}_k^{\mathsf{T}}\mathbf{r}_k, \tag{2.54}$$

where $\mathbf{r}_k = \mathbf{r}(\boldsymbol{\beta}_k)$, $\mathtt{J}_k = \mathtt{J}(\boldsymbol{\beta}_k)$ is the Jacobian matrix of $\mathbf{r}(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}_k$, and $\mathtt{B}_k = \sum_{i=1}^{l} r_i(\boldsymbol{\beta}_k)\mathtt{G}_i(\boldsymbol{\beta}_k)$, with $\mathtt{G}_i(\boldsymbol{\beta}_k)$ the Hessian matrix of $r_i(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}_k$. When the residuals $r_i(\boldsymbol{\beta}_k)$ are small, the quantity $\|\mathtt{B}_k\|$ is small compared to $\|\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k\|$. In such situations, $\mathtt{B}_k$ is typically discarded from (2.54) which gives rise to the so-called *normal equations*

$$\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k\mathbf{p}_k = -\mathtt{J}_k^{\mathsf{T}}\mathbf{r}_k. \tag{2.55}$$

Rearranging this expression, the search direction vector is readily prescribed by

$$\mathbf{p}_k = -[\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k]^{-1}\mathtt{J}_k^{\mathsf{T}}\mathbf{r}_k. \tag{2.56}$$

The Gauss-Newton method is a procedure which updates the search direction in the fashion described by (2.56). Its popularity comes from the fact that the method ultimately converges to a solution at the same (quadratic) rate as Newton's method, however, relying on first derivative information only—encoded in the Jacobian $\mathtt{J}_k$. The value of $\mathbf{p}_k$ in (2.56) can be used in (2.53) to find an update for $\boldsymbol{\beta}_k$.

Under a model employing a quadratic function, the search direction vector given in (2.56) enforces the *descent* condition $F(\boldsymbol{\beta}_{k+1}) < F(\boldsymbol{\beta}_k)$. This requirement is a safety measure to prevent convergence towards a maximiser or a saddle point of the function. Note that F will be monotonically decreasing only if certain favorable conditions are present. This topic is discussed next.

Given a suitable starting point, convergence to a minimiser of F is guaranteed provided that the Jacobian $\mathtt{J}_k$ has full rank in all steps. Unfortunately, this is not the case for the function in (2.52). In general, the problem can be remedied by taking the pseudo-inverse of the quantity $\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k$. If a current approximate solution $\boldsymbol{\theta}_k = g(\boldsymbol{\beta}_k)$ is known, another way to improve performance is to compute an updated estimate $\boldsymbol{\theta}_{k+1}$ as the normal to the hyperplane tangent at $\boldsymbol{\theta}_k$ [61]. When $\boldsymbol{\theta}_k$ has unit norm, this condition is equivalent to requiring that $\boldsymbol{\theta}_{k+1}$ also lies in the unit sphere, see Figure 2.4. For a length-l vector $\boldsymbol{\theta}$, this is ensured by using a *projection matrix*

$$\mathtt{P}(\boldsymbol{\theta}) = \mathtt{I}_{l\times l} - \mathtt{Q}(\boldsymbol{\theta}) = \mathtt{I}_{l\times l} - \|\boldsymbol{\theta}\|^{-2}\boldsymbol{\theta}\boldsymbol{\theta}^{\mathsf{T}}. \tag{2.57}$$

We show next that, with $\boldsymbol{\theta} = g(\boldsymbol{\beta})$, this matrix is given by

$$\mathtt{P}(\boldsymbol{\beta}) = \mathtt{I}_{s\times s} - \frac{(\partial g(\boldsymbol{\beta})/\partial\boldsymbol{\beta})^{\mathsf{T}} g(\boldsymbol{\beta}) g(\boldsymbol{\beta})^{\mathsf{T}} (\partial g(\boldsymbol{\beta})/\partial\boldsymbol{\beta})}{\|(\partial g(\boldsymbol{\beta})/\partial\boldsymbol{\beta})^{\mathsf{T}} g(\boldsymbol{\beta})\|^2}, \tag{2.58}$$

where s refers to the length of $\boldsymbol{\beta}$.

Let $\Delta_{\boldsymbol{\theta}}$ be a small pertubation of $\boldsymbol{\theta}$ and, for simplicity, let $\mathtt{P}_{\boldsymbol{\theta}}$ and $\mathtt{Q}_{\boldsymbol{\theta}}$ be short for $\mathtt{P}(\boldsymbol{\theta})$ and $\mathtt{Q}(\boldsymbol{\theta})$, respectively. Ideally, $\Delta_{\boldsymbol{\theta}}$ should be such that

$$\mathtt{Q}_{\boldsymbol{\theta}}\,\Delta_{\boldsymbol{\theta}} = \mathbf{0}, \tag{2.59}$$

or equivalently,

$$\mathtt{P}_{\boldsymbol{\theta}}\,\Delta_{\boldsymbol{\theta}} = \Delta_{\boldsymbol{\theta}}. \tag{2.60}$$

If (2.59) (or (2.60)) is not satisfied, then $\Delta_{\boldsymbol{\theta}}$ is replaced by

$$\Delta'_{\boldsymbol{\theta}} = \mathtt{P}_{\boldsymbol{\theta}}\,\Delta_{\boldsymbol{\theta}}$$

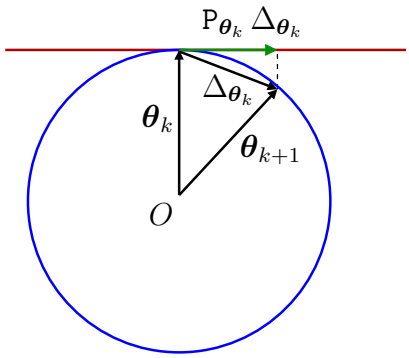


Figure 2.4. Action of the projection matrix on a parameter estimate.

so that $\Delta'_{\boldsymbol{\theta}}$ now satisfies $\mathtt{Q}_{\boldsymbol{\theta}}\,\Delta'_{\boldsymbol{\theta}} = \mathbf{0}$ and $\mathtt{P}_{\boldsymbol{\theta}}\,\Delta'_{\boldsymbol{\theta}} = \Delta'_{\boldsymbol{\theta}}$. In trying to satisfy (2.59), we try to ensure that

$$\|\boldsymbol{\theta} + \Delta_{\boldsymbol{\theta}}\| = \|\boldsymbol{\theta}\|$$

up to second order in $\Delta_{\boldsymbol{\theta}}$. Taking into account that $\boldsymbol{\theta} = g(\boldsymbol{\beta})$ gives

$$\left\| g(\boldsymbol{\beta}) + \frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \Delta_{\boldsymbol{\beta}} \right\| = \|g(\boldsymbol{\beta})\|. \tag{2.61}$$

Since

$$\left\| g(\boldsymbol{\beta}) + \frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \Delta_{\boldsymbol{\beta}} \right\|^2 = \|g(\boldsymbol{\beta})\|^2 + 2g^{\mathsf{T}}(\boldsymbol{\beta}) \frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \Delta_{\boldsymbol{\beta}} + \mathcal{O}(\|\Delta_{\boldsymbol{\beta}}\|^2),$$

condition (2.61) can be rewritten as

$$g^{\mathsf{T}}(\boldsymbol{\beta}) \frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \Delta_{\boldsymbol{\beta}} = \mathbf{0}$$

when neglecting the second order term. By analogy to (2.59), this expression is equivalent to $\mathtt{Q}_{\boldsymbol{\beta}}\,\Delta_{\boldsymbol{\beta}} = \mathbf{0}$, where

$$\mathtt{Q}_{\boldsymbol{\beta}} = \frac{(\partial g(\boldsymbol{\beta})/\partial \boldsymbol{\beta})^{\mathsf{T}} g(\boldsymbol{\beta}) g(\boldsymbol{\beta})^{\mathsf{T}} (\partial g(\boldsymbol{\beta})/\partial \boldsymbol{\beta})}{\|(\partial g(\boldsymbol{\beta})/\partial \boldsymbol{\beta})^{\mathsf{T}} g(\boldsymbol{\beta})\|^2}.$$

The associated projection matrix then takes the form (2.58).

The preceding remarks are included in the final version of the Gauss-Newton method outlined in Algorithm 7. It is assumed that the unconstrained estimator was initialised with the ALS method presented in Section 2.2.1. This implies that an estimate $\widehat{\boldsymbol{\beta}}_{\mathrm{ALS}}$ exists such that $\widehat{\boldsymbol{\theta}}_{\mathrm{ALS}} = g(\widehat{\boldsymbol{\beta}}_{\mathrm{ALS}})$. A constrained estimate is then evolved from $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$ and $\widehat{\boldsymbol{\beta}}_{\mathrm{ALS}}$. Since g is problem specific, we omit the details of computing $\widehat{\boldsymbol{\beta}}_{\mathrm{ALS}}$ here. Examples will be given in Section 4.2.2 when examining different parameterisations of the trifocal tensor. Clearly, if the unconstrained scheme is initialised with a different method, then the corresponding $\boldsymbol{\beta}$-vector must be used in place of $\widehat{\boldsymbol{\beta}}_{\mathrm{ALS}}$.

Algorithm 7 Gauss-Newton correction scheme

Steps to correct an unconstrained estimate $\widehat{\boldsymbol{\theta}}^u_{\text{AML}}$:

1. Set $k = 0$ and $\boldsymbol{\beta}_k = \widehat{\boldsymbol{\beta}}_{\text{ALS}}$.
2. Compute
$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - [\mathtt{P}_k \mathtt{J}_k^\mathsf{T} \mathtt{J}_k \mathtt{P}_k^\mathsf{T}]^+ \mathtt{P}_k \mathtt{J}_k^\mathsf{T} \mathbf{r}_k,$$
where $\mathtt{P}_k = \mathtt{P}(\boldsymbol{\beta}_k)$ as per (2.58).
3. Compute $\boldsymbol{\theta}_{k+1} = g(\boldsymbol{\beta}_{k+1})/\|g(\boldsymbol{\beta}_{k+1})\|$.
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\text{GN}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

2.5.2 Weighted nonlinear least-squares correction

The Gauss-Newton method can be improved by including information about the covariances of parameters. Given a parameter estimate represented by a length-l vector $\boldsymbol{\theta}$, let $\mathtt{C}_{\boldsymbol{\theta}}$ denote the $l \times l$ *covariance matrix* of $\boldsymbol{\theta}$. The aim now is to minimise the geometric distance between $\boldsymbol{\theta}$ and $\widehat{\boldsymbol{\theta}}^u_{\text{AML}}$ under the Mahalanobis metric induced by $\mathtt{C}_{\boldsymbol{\theta}}$. The problem becomes a *weighted nonlinear least-squares* (WNLS) optimisation

$$\arg\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{2}\, \mathbf{r}(\boldsymbol{\beta})^\mathsf{T} \mathtt{W} \mathbf{r}(\boldsymbol{\beta}) = \frac{1}{2}\, \|\mathtt{L}\mathbf{r}(\boldsymbol{\beta})\|^2 \tag{2.62}$$

with $\mathtt{W} = \mathtt{C}^+_{g(\boldsymbol{\beta})}$ and $\mathtt{L}^\mathsf{T}\mathtt{L} = \mathtt{W}$. Letting

$$\|\mathbf{r}(\boldsymbol{\beta})\|^2_{\mathtt{C}_{g(\boldsymbol{\beta})}} = \mathbf{r}(\boldsymbol{\beta})^\mathsf{T} \mathtt{C}^+_{g(\boldsymbol{\beta})} \mathbf{r}(\boldsymbol{\beta}) = \|\mathtt{L}\mathbf{r}(\boldsymbol{\beta})\|^2,$$

problem (2.62) can be rewritten as

$$\arg\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{2}\, \|\mathbf{r}(\boldsymbol{\beta})\|^2_{\mathtt{C}_{g(\boldsymbol{\beta})}} = \frac{1}{2}\, \left\| (g(\boldsymbol{\beta}) - \widehat{\boldsymbol{\theta}}^u_{\text{AML}})/\|g(\boldsymbol{\beta})\| \right\|^2_{\mathtt{C}_{g(\boldsymbol{\beta})}}. \tag{2.63}$$

The above expression emphasises the fact that the residual function $\mathbf{r}(\boldsymbol{\beta})$ is optimised in the metric induced by the parameter covariance matrix $\mathtt{C}_{g(\boldsymbol{\beta})}$. This formulation will turn out useful in Chapter 5 when compared to another cost function underlying a particular method for trifocal tensor estimation. At each iteration step the search direction vector associated with (2.62) is calculated as

$$\mathbf{p}_k = -[\mathtt{J}_k^\mathsf{T} \mathtt{W} \mathtt{J}_k]^{-1} \mathtt{J}_k^\mathsf{T} \mathtt{W} \mathbf{r}_k. \tag{2.64}$$

This expression can be substituted in (2.53) to obtain an updated value of $\boldsymbol{\beta}_k$.

Candidate parameter covariance matrices

Two choices of parameter covariance matrix can be used in problem (2.62). First, the minimisation may be carried out optimally in the metric induced by the covariance matrix of $\widehat{\boldsymbol{\theta}}^u_{\text{AML}}$ [49]:

$$\mathtt{C}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}} = \left[\mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}\right]^-_{l-1}. \tag{2.65}$$

Here constraining the rank of the matrix is needed to ensure that $\mathtt{C}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$ obtained numerically is positive semi-definite. In this case, it will be shown next that the variational equation (2.22), written explicitly as

$$\mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}} \widehat{\boldsymbol{\theta}}^u_{\text{AML}} = \mathbf{0}, \tag{2.66}$$

leads to a simplification of the WNLS problem.

Consider the following first order Taylor expansion of $g(\boldsymbol{\beta}_{k+1})$ in a neighbourhood of $\boldsymbol{\beta}_k$

$$g(\boldsymbol{\beta}_{k+1}) = g(\boldsymbol{\beta}_k) + \frac{\partial g(\boldsymbol{\beta}_k)}{\partial \boldsymbol{\beta}_k}(\boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k).$$

Since

$$\frac{\partial \mathbf{r}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial \mathbf{r}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\beta}} = \frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}},$$

we have

$$\mathtt{J}_k = \frac{\partial \mathbf{r}(\boldsymbol{\beta}_k)}{\partial \boldsymbol{\beta}_k} = \frac{\partial g(\boldsymbol{\beta}_k)}{\partial \boldsymbol{\beta}_k},$$

hence we may write

$$g(\boldsymbol{\beta}_{k+1}) = g(\boldsymbol{\beta}_k) + \mathtt{J}_k(\boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k) = g(\boldsymbol{\beta}_k) + \mathtt{J}_k \mathbf{p}_k. \tag{2.67}$$

Recalling (2.51) and simplifying (2.64) with (2.65) and (2.66) yields

$$\mathbf{p}_k = -[\mathtt{J}_k^\mathsf{T} \mathtt{W} \mathtt{J}_k]^+ \mathtt{J}_k^\mathsf{T} \mathtt{W} g(\boldsymbol{\beta}_k)/\|g(\boldsymbol{\beta}_k)\|.$$

Note that the pseudo-inverse of $\mathtt{J}_k^\mathsf{T}\mathtt{W}\mathtt{J}_k$ must be employed because matrix $\mathtt{W} = \mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$ is rank-deficient and so is the product with $\mathtt{J}_k$. Substituting this form of $\mathbf{p}_k$ in (2.67) shows that $g(\boldsymbol{\beta}_{k+1})$ is fully expressed in terms of $g(\boldsymbol{\beta}_k)$. Therefore, (2.67) turns out to be a special case of linearisation of $r(\boldsymbol{\beta}_{k+1})$ in a neighbourhood of $\boldsymbol{\beta}_k$ arising from the choice of covariance matrix in (2.65).

Another candidate parameter covariance matrix is $\mathtt{M}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$ whose form was originally introduced in (2.21b). This matrix is a good approximation of $\mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$ and has the advantage of being generically positive definite, and therefore invertible, provided $n \geq l$. The general case of the weighted nonlinear least-squares correction scheme is shown in Algorithm 8 for $\mathtt{W} = \mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$. Note that the Gauss-Newton method which solves problem (2.52) can now be seen as a particular case of the WNLS method where the parameter covariance matrix $\mathtt{W}$ is set to the identity, meaning that no weights are applied to the residual vector.

Algorithm 8 Weighted Nonlinear Least-Squares correction scheme

Steps to correct an unconstrained estimate $\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}$:

1. Set $k=0$, $\boldsymbol{\beta}_k = \widehat{\boldsymbol{\beta}}_{\text{ALS}}$, and $\mathtt{W} = \mathtt{X}_{\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}}$.
2. Compute
$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - [\mathtt{P}_k \mathtt{J}_k^{\mathsf{T}} \mathtt{W} \mathtt{J}_k \mathtt{P}_k^{\mathsf{T}}]^{+} \mathtt{P}_k \mathtt{J}_k^{\mathsf{T}} \mathtt{W} \mathbf{r}_k,$$
where $\mathtt{P}_k = \mathtt{P}(\boldsymbol{\beta}_k)$ as per (2.58).
3. Compute $\boldsymbol{\theta}_{k+1} = g(\boldsymbol{\beta}_{k+1})/\|g(\boldsymbol{\beta}_{k+1})\|$.
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\text{WNLS}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

Further analysis

Recall the cost function $F(\boldsymbol{\beta})$ given in (2.62) which underpins the WNLS correction scheme. When $\mathtt{W} = \mathtt{X}_{\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}}$, we demonstrate next that this function enters the expression of a second-order approximation of the unconstrained cost function J_{AML} in the neighbourhood of $\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}$.

Writing out (2.62) explicitly gives

$$F(\boldsymbol{\beta}) = \frac{1}{2\|g(\boldsymbol{\beta})\|^2} \, (g(\boldsymbol{\beta}) - \widehat{\boldsymbol{\theta}}^{u}_{\text{AML}})^{\mathsf{T}} \mathtt{C}^{+}_{g(\boldsymbol{\beta})} (g(\boldsymbol{\beta}) - \widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}),$$

where $\boldsymbol{\theta} = g(\boldsymbol{\beta})$ belongs to the region of the $\boldsymbol{\theta}$-manifold where the constraints imposed by g are satisfied. From (2.65) and (2.66), the above expression reduces to

$$F(\boldsymbol{\beta}) = \frac{1}{2\|g(\boldsymbol{\beta})\|^2} \, g(\boldsymbol{\beta})^{\mathsf{T}} \mathtt{X}_{\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}} g(\boldsymbol{\beta}). \tag{2.68}$$

Now, let $\mathbf{h} = \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}$ and suppose that $\boldsymbol{\theta} = g(\boldsymbol{\beta})$ lies in the vicinity of $\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}$. A second-order Taylor expansion of $J_{\text{AML}}(\boldsymbol{\theta})$ around $\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}$ is given by

$$J_{\text{AML}}(\boldsymbol{\theta}) = J_{\text{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) + \mathbf{h}^{\mathsf{T}} \mathbf{g}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) + \frac{1}{2} \mathbf{h}^{\mathsf{T}} \mathtt{H}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) \mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3),$$

where $\mathbf{g}(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} J_{\text{AML}}(\boldsymbol{\theta})^{\mathsf{T}}$ and $\mathtt{H}(\boldsymbol{\theta}) = \partial^2_{\boldsymbol{\theta}\boldsymbol{\theta}} J_{\text{AML}}(\boldsymbol{\theta})$ are the gradient and Hessian of J_{AML} respectively. Given that $\mathbf{g}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) = \mathbf{0}$ and $\mathtt{H}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) \widehat{\boldsymbol{\theta}}^{u}_{\text{AML}} = \mathbf{0}$, we have

$$\begin{aligned} J_{\text{AML}}(\boldsymbol{\theta}) &\approx J_{\text{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) + \frac{1}{2} \mathbf{h}^{\mathsf{T}} \mathtt{H}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) \mathbf{h}, \\ &= J_{\text{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) + \frac{1}{2} \boldsymbol{\theta}^{\mathsf{T}} \mathtt{H}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) \boldsymbol{\theta}. \end{aligned}$$

Defining the function

$$J_{\text{H}}(\boldsymbol{\beta}) = \frac{1}{2} g(\boldsymbol{\beta})^{\mathsf{T}} \mathtt{H}(\widehat{\boldsymbol{\theta}}^{u}_{\text{AML}}) g(\boldsymbol{\beta}), \tag{2.69}$$

and expressing J_{AML} in terms of $\boldsymbol{\beta}$, it follows that

$$J_{\mathrm{AML}}(g(\boldsymbol{\beta})) \approx J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}) + J_{\mathrm{H}}(\boldsymbol{\beta}).$$

For difficult problems, the Hessian of second derivatives may not be available. Various choices are possible to approximate it. One such choice is $\partial^2_{\boldsymbol{\theta}\boldsymbol{\theta}} J_{\mathrm{AML}}(\boldsymbol{\theta}) = \mathtt{X}_{\boldsymbol{\theta}}$. This with (2.68) and (2.69) gives $J_{\mathrm{H}}(\boldsymbol{\beta}) = \|g(\boldsymbol{\beta})\|^2 F(\boldsymbol{\beta})$, therefore

$$J_{\mathrm{AML}}(g(\boldsymbol{\beta})) \approx J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}) + \|g(\boldsymbol{\beta})\|^2 F(\boldsymbol{\beta}).$$

This analysis shows that function F contributes to the overall J_{AML} cost value. If $\boldsymbol{\beta}^*$ is the minimiser of $F(\boldsymbol{\beta})$ and given that $J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}) \leq J_{\mathrm{AML}}(\boldsymbol{\theta})$, we expect $J_{\mathrm{AML}}(g(\boldsymbol{\beta}^*))$ to be slightly higher than $J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}})$. This fact will be confirmed in the experimental chapters.

2.5.3 Kanatani-like correction

Although the previously described methods do not utilise the values of the ancillary constraints, they produce a constrained estimate by minimising a cost function based on a parameterisation of the $\boldsymbol{\theta}$-manifold where these constraints are satisfied. As an alternative to these techniques, Kanatani [49] proposed a first-order correction scheme which employs a single ancillary constraint directly in its adjustment mechanism. The correction effectively applies to the full length vector $\boldsymbol{\theta}$ rather than $\boldsymbol{\beta}$. Below, Kanatani's original work is expanded to deal with multiple ancillary constraints.

In spirit of Kanatani's writing, the covariance matrix of $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$ is assumed to take the form given in (2.65). Note that Kanatani never defines $\mathtt{C}_{\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}}$ as he prefers working with renormalisation estimates instead of AML estimates.

Recalling the form of $\mathtt{P}(\boldsymbol{\theta})$ given in (2.57), define a *tangential* covariance matrix at $\boldsymbol{\theta}$ by

$$\mathtt{T}_{\boldsymbol{\theta}} = \mathtt{P}(\boldsymbol{\theta})\, \mathtt{C}_{\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}}\, \mathtt{P}(\boldsymbol{\theta}). \tag{2.70}$$

The matrix $\mathtt{T}_{\boldsymbol{\theta}}$ is guaranteed to carry no information in the direction of $\boldsymbol{\theta}$, so that the scale of $\boldsymbol{\theta}$ does not matter.

Let $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_p]^{\mathsf{T}}$ be a length-p column vector of constraints. The ancillary constraints are then given by the system of equations

$$\boldsymbol{\Phi}(\boldsymbol{\theta}) = \mathbf{0}. \tag{2.71}$$

Let

$$\mathtt{D}_{\boldsymbol{\theta}} = [\nabla_{\boldsymbol{\theta}}\phi_1, \ldots, \nabla_{\boldsymbol{\theta}}\phi_p] = \begin{bmatrix} \partial_{\theta_1}\phi_1 & \ldots & \partial_{\theta_1}\phi_p \\ \ldots & \ldots & \ldots \\ \partial_{\theta_l}\phi_1 & \ldots & \partial_{\theta_l}\phi_p \end{bmatrix}$$

be the $l \times p$ gradient matrix of $\boldsymbol{\Phi}$ (the transpose of the Jacobian matrix of $\boldsymbol{\Phi}$). A Kanatani-like correction scheme is shown in Algorithm 9. The method assumes that the rank of $\mathtt{D}_{\boldsymbol{\theta}}$ is t for $\boldsymbol{\theta}$ satisfying (2.71), in other words, t is the number of algebraically independent constraints.

Algorithm 9 Kanatani-like correction scheme

Steps to correct an unconstrained estimate $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$:

1. Set $k = 0$ and $\boldsymbol{\theta}_k = \widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$.
2. Compute the matrix $\mathtt{T}_{\boldsymbol{\theta}_k}$ as per (2.70).
3. Compute
$$\tilde{\boldsymbol{\theta}}_{k+1} = \boldsymbol{\theta}_k - \mathtt{T}_{\boldsymbol{\theta}_k}\mathtt{D}_{\boldsymbol{\theta}_k}\,[\mathtt{D}^{\mathsf{T}}_{\boldsymbol{\theta}_k}\mathtt{T}_{\boldsymbol{\theta}_k}\mathtt{D}_{\boldsymbol{\theta}_k}]^{-}_{t}\,\boldsymbol{\Phi}(\boldsymbol{\theta}_k).$$
4. Compute $\boldsymbol{\theta}_{k+1} = \tilde{\boldsymbol{\theta}}_{k+1}/\|\tilde{\boldsymbol{\theta}}_{k+1}\|$.
5. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\mathrm{KK}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

2.6 Coping with rank-deficient Jacobian and Hessian

One common issue with optimisation techniques is *stability*. Quite often, the stability of a method is affected if the solution is either not unique or very sensitive to noise or measurement errors in the input data of the problem. The success of the Gauss-Newton method is closely related to the sign of the quantity $\delta = \mathbf{p}^{\mathsf{T}}\mathtt{G}\mathbf{p}$, where $\mathbf{p}$ denotes a search direction vector and $\mathtt{G} = (\mathtt{J}^{\mathsf{T}}\mathtt{J} + \mathtt{B})$ is the Hessian matrix of F appearing in Newton's equations (2.54). To guarantee a decrease in the cost function and therefore convergence to a minimiser of F, δ needs to be positive, which is guaranteed when $\mathtt{G}$ is positive definite [92]. When the Jacobian $\mathtt{J}$ is rank-deficient in a neighbourhood of a local minimum, $\mathtt{G}$ is also rank-deficient, $\mathbf{p}$ is not unique and this causes the Gauss-Newton method to fail producing a reliable descent direction. In this situation, the matrix $\mathtt{J}^{\mathsf{T}}\mathtt{J}$ is no longer a good approximation of $\mathtt{G}$. In this case, two approaches are possible. One is to incorporate the second-order matrix $\mathtt{B}$ in $\mathtt{G}$ or an approximation to it. The other is to advance a novel formulation of the problem that deals directly with the rank-deficient Jacobian. Next, we expose two methods which implement each of these strategies. The proposed techniques provide a regularisation of the standard Gauss-Newton algorithm.

2.6.1 The Levenberg-Marquardt method

The Levenberg-Marquardt (LM) algorithm [92] implements a *damped Gauss-Newton* method where the search direction is defined by the following modification of (2.55)

$$(\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k + \mu\mathtt{I}_{n\times n})\mathbf{p}_k = -\mathtt{J}_k^{\mathsf{T}}\mathbf{r}_k, \tag{2.72}$$

where $\mu > 0$ is a *damping parameter*. The behaviour (and stability) of LM is controlled by the value of μ. This parameter ensures that the algorithm stays within a certain *trust region* at each iteration. When $\mathtt{J}_k$ is rank-deficient, μ must be sufficiently large value to assure that the Hessian matrix $\mathtt{G}_k = (\mathtt{J}_k^{\mathsf{T}}\mathtt{J}_k + \mu\mathtt{I}_{n\times n})$ is positive definite. It is only in this case that $\mathbf{p}_k$ is well-defined (unique) and can be stably estimated. By analogy to (2.54), the matrix $\mu\mathtt{I}_{n\times n}$ may be thought as an approximation of matrix $\mathtt{B}_k$ to ensure that $\mathtt{G}_k$ is positive definite. The value of μ is computed dynamically at each iteration and several rules may be implemented to update it. The interested reader can find more details in [92]. When information about $\mathtt{W}$, the inverse covariance matrix of $\boldsymbol{\theta}$, is available, minimising (2.62) may proceed by using the LM algorithm applied to the residual vector

$$\mathbf{r}'(\boldsymbol{\beta}) = \mathtt{L}\,\mathbf{r}(\boldsymbol{\beta})/\sqrt{2}. \tag{2.73}$$

Recall that $\mathtt{L}$ is "half" the covariance matrix $\mathtt{W}$ such that $\mathtt{L}^{\mathsf{T}}\mathtt{L} = \mathtt{W}$. Judicious choices for $\mathtt{W}$ are $\mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$ or $\mathtt{M}_{\widehat{\boldsymbol{\theta}}^u_{\text{AML}}}$, as mentioned in Section 2.5.2. In the simplest scenario, one may take $\mathtt{L} = \mathtt{I}_{l\times l}$ in which case LM solves the minimisation problem (2.52). A programming package such as MATLAB already provides an implementation of LM so the work is limited to defining the residual vector $\mathbf{r}'(\boldsymbol{\beta})$ as input to the built-in function.

2.6.2 The truncated Gauss-Newton method

The cost function given in (2.52) may be modified to accommodate the situation where the Jacobian of $\mathbf{r}$ is rank-deficient. For ill-posed problems of this type, the search direction vector is no longer unique and the computed estimates $\boldsymbol{\beta}_k$ tend to get excessively large. To avoid these situations, a new function can be evolved to solve a *minimum norm nonlinear least-squares* problem defined by coupling two optimisation problems as follows

$$\begin{aligned} \boldsymbol{\beta}^* &= \underset{\boldsymbol{\beta}\,\in\,\mathcal{B}}{\arg\min}\ \frac{1}{2}\,\|\boldsymbol{\beta} - \boldsymbol{\zeta}\|^2 \\ \text{such that } \mathcal{B} &= \{\,\boldsymbol{\beta} \in \mathbb{R}^s \mid \boldsymbol{\beta} = \underset{\tilde{\boldsymbol{\beta}}}{\arg\min}\ \frac{1}{2}\,\|\mathbf{r}(\tilde{\boldsymbol{\beta}})\|^2\,\}. \end{aligned} \tag{2.74}$$

Here $\boldsymbol{\zeta}$ corresponds to a natural centre for the problem and therefore is application dependent. It is either chosen a priori from some reasonable approximation of the solution or set to zero.

The criteria in (2.74) intend to provide some regularisation to the starting ill-conditioned problem in order to achieve a solution of practical value. Similar to the way the original least-squares problem (2.52) can be resolved at each step by determining a search direction vector $\mathbf{p}_k$ using the normal equations (2.55), the minimum norm problem (2.74) can be stated equivalently as

$$\mathbf{p}_k = \underset{\mathbf{p} \in \mathcal{P}_k}{\arg\min} \frac{1}{2} \| \boldsymbol{\beta}_k - \boldsymbol{\zeta} + \mathbf{p} \|^2 \tag{2.75}$$

$$\text{such that } \mathcal{P}_k = \{ \mathbf{p} \in \mathbb{R}^s \mid \mathbf{p} = \underset{\tilde{\mathbf{p}}}{\arg\min} \frac{1}{2} \| \mathbf{r}(\boldsymbol{\beta}_k) + \mathtt{J}_k \tilde{\mathbf{p}} \|^2 \}.$$

To derive an expression for the search direction vector which satisfies the above conditions, it is necessary to consider the SVD decomposition of the Jacobian matrix $\mathtt{J}_k$.

Suppose that at step k the following SVD decomposition of $\mathtt{J}_k$ holds

$$\mathtt{J}_k = \mathtt{E}\mathtt{D}\mathtt{K}^\mathsf{T} = (\mathtt{E}_1, \mathtt{E}_2) \begin{pmatrix} \mathtt{D}_1 & \\ & \mathtt{D}_2 \end{pmatrix} (\mathtt{K}_1, \mathtt{K}_2)^\mathsf{T}, \tag{2.76}$$

where $\mathtt{E}^\mathsf{T}\mathtt{E} = \mathtt{I}_{l \times l}$, $\mathtt{K}^\mathsf{T}\mathtt{K} = \mathtt{I}_{s \times s}$, $\mathtt{D}_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_t)$, and $\mathtt{D}_2 = \mathrm{diag}(\sigma_{t+1}, \ldots, \sigma_s)$ with the singular values σ_i sorted in decreasing order. This splitting of the Jacobian gives

$$\mathtt{J}_k = \mathtt{J}_1 + \mathtt{J}_2 = \mathtt{E}_1\mathtt{D}_1\mathtt{K}_1^\mathsf{T} + \mathtt{E}_2\mathtt{D}_2\mathtt{K}_2^\mathsf{T}.$$

For rank-deficient problems, there exists a clear gap in the SVD-spectrum between σ_t and σ_{t+1}, which allows for such a partitioning to take place. The entity t is called the *grade* of $\mathtt{J}_k$. Since $\mathtt{J}_2$ is negligible or zero, an (approximate) solution to (2.75) is given by

$$\mathbf{p}_k = -(\mathtt{J}_1^+, \mathtt{P}_{\mathcal{N}_1}) \begin{pmatrix} \mathbf{r}(\boldsymbol{\beta}_k) \\ \boldsymbol{\beta}_k - \boldsymbol{\zeta} \end{pmatrix}, \tag{2.77}$$

where $\mathtt{P}_{\mathcal{N}_1}$ is the orthogonal projection on the null-space of $\mathtt{J}_1$. Since we require that at the solution

$$\mathtt{P}_{\mathcal{N}_1}(\boldsymbol{\beta}_k - \boldsymbol{\zeta}) = \mathbf{0},$$

(2.77) reduces to

$$\mathbf{p}_k = -\mathtt{K}_1\mathtt{D}_1^{-1}\mathtt{E}_1^\mathsf{T}\mathbf{r}(\boldsymbol{\beta}_k). \tag{2.78}$$

The obtained search direction vector is such that both minimisation criteria in (2.75) are satisfied in limit as $k \to \infty$, and so are those in (2.74). The *truncated Gauss-Newton* (TGN) method which solves the minimum norm problem (2.74) is detailed in Algorithm 11. If parameter covariance information is available, then $\mathbf{r}(\boldsymbol{\beta}_k)$ in (2.78) may be replaced by $\mathbf{r}'(\boldsymbol{\beta}_k)$ in (2.73).

Algorithm 11 Truncated Gauss-Newton correction scheme

Steps to correct an unconstrained estimate $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$:

1. Set $k = 0$ and $\boldsymbol{\beta}_k = \widehat{\boldsymbol{\beta}}_{\mathrm{ALS}}$.
2. Compute $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{p}_k$, where $\mathbf{p}_k$ is given in (2.78).
3. Compute $\boldsymbol{\theta}_{k+1} = g(\boldsymbol{\beta}_{k+1})/\|g(\boldsymbol{\beta}_{k+1})\|$.
4. If $\boldsymbol{\theta}_{k+1}$ is sufficiently close to $\boldsymbol{\theta}_k$, then terminate the procedure and set $\widehat{\boldsymbol{\theta}}_{\mathrm{TGN}} = \boldsymbol{\theta}_{k+1}$; otherwise increment k and return to step 2.

2.7 Conclusion

The work presented in this chapter relied on a parametric model in which the relationship between parameters and image features is expressed as a system of equations. In a first stage, several multi-objective cost functions were conceived based on this model and used to create unconstrained estimators.

The most ordinary cost functions considered minimising algebraic residual errors. Although their minimiser can be expressed in closed form, these functions exhibit a statistical bias. To overcome this, a maximum likelihood (ML) cost function was evolved and lead to a statistically optimal estimator. In this framework the presence of nuisance parameters create a significant barrier to the scalability of the algorithm to large data sets. Consequently, the approach taken was to generate an approximation to the ML cost function whereby the nuisance parameters are eliminated and only the principal parameters are considered in the estimation. The concocted AML cost function fullfills these criteria and therefore exhibit a much smaller search space, of dimension equal to the size of $\boldsymbol{\theta}$. This distinction considerably reduces the computational complexity of its associated estimator, FNS.

For a sub-class of the parametric model the objective function consists of linearly dependent components. The standard AML cost function corresponding to any of such model involves inverses of ill-conditioned matrices and is not well suited for generating accurate estimates when noise in the data is small. A solution to these deficiencies was to replace the inverses of the critical matrices by generalised inverses. Two variants of FNS were proposed, one relies on a curtailed objective function with linearly independent components, and the other uses the full objective function combined with a rank-constrained pseudo-inverse.

FNS is a local minimiser and so provides a function value which is minimal inside a feasible neighbourhood. The scheme often necessitates a good initialisation and a relatively small noise in the data in order to converge to a minimiser of the AML cost function. The algorithm was

robustified by decoupling the estimation process into two problems of lower dimension. First, a subset of the total parameter vector is estimated iteratively and then the remaining parameters are recovered in a single step based on the result of the earlier iterative process. It will be shown in the experimental chapters that the reduced form of FNS, RFNS, enjoys better converge properties. Finally, it was noted that a companion scheme, RHEIV, can be evolved in a similar fashion to that of RFNS. Subsequent tests will confirm that the performance of RHEIV matches that of RFNS.

The second part of the chapter explored various post-correction techniques to adjust for any ancillary constraints that may apply. The weighted nonlinear least-squares (WNLS) method, which generalised the Gauss-Newton (GN) method, and a Kanatani-like method are classical schemes which accomplish a first-order correction of the constraints. A common feature between the WNLS and GN correction schemes is that they do not handle the ancillary constraints directly because they are based on minimising a cost function. This is a major advantage for applications in which the ancillary constraints are difficult to express in a concise and efficient manner. The Kanatani-like method is the only one to truly involve the value of the ancillary constraints.

For some problems, the solution is not well-defined when the Jacobian and Hessian of the constraint objective function F are rank-deficient. Two other methods were proposed to deal with such cases. One is the Levenberg-Marquardt method and the other is a truncated Gauss-Newton method. In general, these methods attempt to guarantee a positive definite Hessian matrix in order to obtain a search direction with no ambiguity.

The following chapters put these algorithms into practice, first on the unconstrained minimisation problem of fitting a homography to data and then on the constrained problem of estimating a trifocal tensor between three views of a scene.

Chapter 3

Application I: Homography Matrix Estimation

The unconstrained estimation of parameters plays a very important role in deriving optimal constrained parameters. This chapter examines the behaviour and solutions produced by the three Fundamental Numerical Schemes presented in Chapter 2. The core of these schemes relies on an AML cost function which takes various forms depending on the type of generalised inverse used. Adopting the homography model, which is naturally described by a linearly dependent system of three equations, we demonstrate empirically that the choice of a particular type of generalised inverse is irrelevant as long as the inverse is either continuous or is rendered continuous via regularisation. Modified AML cost functions involving continuous generalised inverses are shown to lead to good-quality parameter estimators of comparable accuracy. Our simulations also reveal that a cost function employing a discontinuous generalised inverse can produce a poorly-behaved estimator. To complete the analysis, tests are carried out on real image sequences and computed homographies are used to create panoramic mosaics. In both synthetic and real data experiments, the performance of the AML estimators is compared to that of existing methods.

3.1 Introduction

To evaluate the quality of various types of AML estimates of $\boldsymbol{\theta}$, we have performed a number of experiments on the computation of a homography. This section provides necessary prerequisites concerning homography estimation. A detailed description of experiments and their results will be deferred until the next section.

3.1.1 Two types of homography

If two overlapping images of a *planar* scene are taken from different camera positions and/or orientations, or if a single camera undergoes *pure rotation*, then the images are linked via a planar projective transformation, or homography. A homography is described by a non-singular 3×3 matrix, say $\mathtt{H}$, in terms of which the action of the homography on a 2-D homogeneous point $\mathbf{m} = [u, v, 1]^\mathsf{T}$ to produce a 2-D homogeneous point $\mathbf{m}' = [u', v', 1]^\mathsf{T}$ is given by

$$\mathbf{m}' \simeq \mathtt{H}\mathbf{m}, \tag{3.1}$$

where $\simeq$ denotes equality up to scale. To define the homography uniquely, $\mathtt{H}$ needs to be specified only up to scale—matrices differing by a non-zero scalar factor encode the same transformation. With the common notational conventions as established in [37], the matrix for a homography induced by a scene plane Π is given by the following generalisation of a formula derived by Faugeras and Lustman [25]:

$$\mathtt{H} = \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times 3} - (\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d)^{-1}(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\mathbf{n}^\mathsf{T}]\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}. \tag{3.2}$$

Here the indices $i = 1$ and $i = 2$ refer to the left and right cameras, respectively, $\mathtt{K}_i$ and $\mathtt{R}_i$ denote the calibration and rotation matrices, and $\tilde{\mathbf{C}}_i$ is the inhomogeneous 3-vector representing the camera centre in the world coordinate system. Furthermore, $\mathbf{n}$ is the unit outward normal of Π and d is the Euclidean distance between Π and the origin of the world frame, taken with the minus sign. When a homography arises from a rotation of a single camera, the corresponding matrix is given by

$$\mathtt{H} = \mathtt{K}_2\mathtt{R}_2\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}. \tag{3.3}$$

The derivation of both (3.2) and (3.3) can be found in Appendix B.

3.1.2 Homography model and associated cost functions

The model pertaining to a homography described by a matrix $\mathtt{H}$ is expressed by

$$\mathbf{m}' \times \mathtt{H}\mathbf{m} = \mathbf{0}. \tag{3.4}$$

This vector equation is an immediate consequence of (3.1) —since the vectors $\mathbf{m}'$ and $\mathtt{H}\mathbf{m}$ differ only by a non-zero scalar, their cross product vanishes. With $\mathtt{H}^\mathsf{T} = [\mathtt{h}_1, \mathtt{h}_2, \mathtt{h}_3]$, (3.4) expands to

$$\begin{aligned} v'\mathtt{h}_3^\mathsf{T}\mathbf{m} - \mathtt{h}_2^\mathsf{T}\mathbf{m} &= 0, \\ \mathtt{h}_1^\mathsf{T}\mathbf{m} - u'\mathtt{h}_3^\mathsf{T}\mathbf{m} &= 0, \\ u'\mathtt{h}_2^\mathsf{T}\mathbf{m} - v'\mathtt{h}_1^\mathsf{T}\mathbf{m} &= 0. \end{aligned} \tag{3.5}$$

This system can be brought into the form given in (2.1) by first concatenating the inhomogeneous coordinates of $\mathbf{m}$ and $\mathbf{m}'$ to obtain a single item of data $\mathbf{x} = [u, v, u', v']^\mathsf{T}$, and then letting $\boldsymbol{\theta} = \mathrm{vec}(\mathtt{H}^\mathsf{T})$ and $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = [\, f_1(\mathbf{x}, \boldsymbol{\theta}), f_2(\mathbf{x}, \boldsymbol{\theta}), f_3(\mathbf{x}, \boldsymbol{\theta})\,]^\mathsf{T}$, where f_1, f_2 and f_3 are the corresponding expressions on the left-hand side of (3.5). Furthermore, we have $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathtt{U}(\mathbf{x})^\mathsf{T}\boldsymbol{\theta}$, where $\mathtt{U}(\mathbf{x}) = [\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \mathbf{u}_3(\mathbf{x})]$ with

$$\begin{aligned}
\mathbf{u}_1(\mathbf{x}) &= [0, 0, 0, -u, -v, -1, uv', vv', v']^\mathsf{T}, \\
\mathbf{u}_2(\mathbf{x}) &= [u, v, 1, 0, 0, 0, -uu', -vu', -u']^\mathsf{T}, \\
\mathbf{u}_3(\mathbf{x}) &= [-uv', -vv', -v', uu', vu', u', 0, 0, 0]^\mathsf{T}.
\end{aligned}$$

As a system of linear equations in $\boldsymbol{\theta}$, (3.5) is linearly dependent. Moreover, each of the three possible pairs of equations deriving from (3.5) are linearly independent and can serve as basic sub-constraints, see Section 2.3.2. Accordingly, one can form three AML cost functions corresponding to the three sub-constraint vectors. In addition, one can also form an AML cost function based on all three constraints, as described in Section 2.3.3.

3.1.3 Normalised algebraic least-squares estimate

Four different ALS estimates of a homography can straightforwardly be evolved starting from (3.4): three based on sub-constraint vectors and one derived from the full constraint vector. As pointed out by Hartley [36], the accuracy of each of these estimates can be greatly enhanced if image coordinates are normalised before the estimates are actually computed. Here we briefly describe how to generate improved ALS estimates.

Let

$$\begin{aligned}
\check{\mathbf{m}} = [\check{m}_1, \check{m}_2, 1]^\mathsf{T} &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{m}_i, \\
\check{\mathbf{m}}' = [\check{m}'_1, \check{m}'_2, 1]^\mathsf{T} &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{m}'_i
\end{aligned}$$

be the *centroids* and let

$$\begin{aligned}
s &= \Big[\frac{1}{2n}\sum_{i=1}^{n}(u_i - \check{m}_1)^2 + (v_i - \check{m}_2)^2\Big]^{1/2}, \\
s' &= \Big[\frac{1}{2n}\sum_{i=1}^{n}(u'_i - \check{m}'_1)^2 + (v'_i - \check{m}'_2)^2\Big]^{1/2}
\end{aligned}$$

be the *scales* of the $\mathbf{m}_i = [u_i, v_i, 1]^\mathsf{T}$ and the $\mathbf{m}'_i = [u'_i, v'_i, 1]^\mathsf{T}$, respectively. Following Hartley [36], define the *normalised* data by

$$\begin{aligned}
\tilde{\mathbf{m}}_i &= [(u_i - \check{m}_1)/s, (v_i - \check{m}_2)/s, 1]^\mathsf{T}, \\
\tilde{\mathbf{m}}'_i &= [(u'_i - \check{m}'_1)/s', (v'_i - \check{m}'_2)/s', 1]^\mathsf{T}.
\end{aligned}$$

This amounts to setting $\tilde{\mathbf{m}}_i = \mathtt{T}\mathbf{m}_i$ and $\tilde{\mathbf{m}}'_i = \mathtt{T}'\mathbf{m}'_i$, where

$$\mathtt{T} = \begin{bmatrix} s^{-1} & 0 & -s^{-1}\check{m}_1 \\ 0 & s^{-1} & -s^{-1}\check{m}_2 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathtt{T}' = \begin{bmatrix} s'^{-1} & 0 & -s'^{-1}\check{m}'_1 \\ 0 & s'^{-1} & -s'^{-1}\check{m}'_2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

Let $\tilde{\mathbf{x}}_i = [\tilde{u}_i, \tilde{v}_i, \tilde{u}'_i, \tilde{v}'_i]^\mathsf{T}$ be the result of concatenation of $\tilde{\mathbf{m}}_i$ and $\tilde{\mathbf{m}}'_i$. Fixing a particular ALS method, let $\tilde{\mathtt{H}}_{\mathrm{ALS}}$ be the ALS homography estimate based on the $\tilde{\mathbf{x}}_i$. Then the *normalised algebraic least-squares* (NALS) estimate of $\mathtt{H}$, $\widehat{\mathtt{H}}_{\mathrm{NALS}}$, is defined by $\widehat{\mathtt{H}}_{\mathrm{NALS}} = \mathtt{T}'^{-1}\tilde{\mathtt{H}}_{\mathrm{ALS}}\mathtt{T}$. This estimate is the sought-after enhanced version of $\widehat{\mathtt{H}}_{\mathrm{ALS}}$.

3.2 Experimental evaluation

The remaining part of this chapter is now dedicated to the development and analysis of tests on homography computation.

3.2.1 Experiments with synthetic image data

Repeated experiments were performed in order to collect results of statistical significance. The regime adopted was to generate a planar scene visible by two perspective cameras and project the scene points onto two 500×500 pixel images to provide "true" matches. Each image point was then perturbed by homogeneous Gaussian noise of preset level and the resulting noise-contaminated pairs of corresponding points were used as input to several algorithms.

Scene and camera set-up

The development of a scene and camera set-up for our experimentation relied heavily upon the formula for a plane-induced homography, as given in (3.2). This formula is, as it were, a powerful modelling tool. It enables one to choose freely the position of a plane in 3-D space and permits realistic visualisation of the scene. In addition, with the aid of this formula, camera centres and intrinsic parameters can be set adequately so that scene points, all lying in the plane, are in front of the cameras and within the cameras' fields of view.

In our experiments, the scene and cameras were arranged as follows. After fixing a world coordinate system, we chose for a planar scene the plane parallel to the x–y plane, positioned 6 units away from the world origin. We assumed that 1 unit is equivalent to 250 pixels. 60 points were randomly selected from the scene plane and then viewed by two perspective cameras placed at $\tilde{\mathbf{C}}_1 = [-1.5, -0.1, 0]^\mathsf{T}$ and $\tilde{\mathbf{C}}_2 = [1.5, 0.1, 0]^\mathsf{T}$. The right camera had a -1° rotation about its x-axis and a 4° rotation about its y-axis to look inwards at the scene. The left camera

was subject to a 5° rotation about its y-axis. In radians, the rotation angles for the left camera are $\alpha_1 = 0$ and $\beta_1 = -5 \times 180/\pi$. For the right camera, the corresponding angle values are $\alpha_2 = -1 \times 180/\pi$ and $\beta_2 = 4 \times 180/\pi$. All rotations are counterclockwise about the respective axes and relative to the camera coordinate system. For each view, a rotation was first applied about the y-axis followed by one about the x-axis (there was no z-axis rotation). The composition of the two transformations gave the final rotation matrices

$$\mathtt{R}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \begin{bmatrix} \cos\beta_i & 0 & \sin\beta_i \\ 0 & 1 & 0 \\ -\sin\beta_i & 0 & \cos\beta_i \end{bmatrix}.$$

For both cameras, the focal length f was set to 250 pixels which allowed a 90° viewing angle. The origin of the image coordinate system was set at the lower left corner of the image as in [37]. The principal point was assumed to be located in the centre of the image ($x_0 = 250, y_0 = 250$) and the skew factor s was taken to be zero. So, both cameras used the common calibration matrix

$$\mathtt{K} = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

The scene plane Π had outward unit normal $\mathbf{n} = [0, 0, 1]^\mathsf{T}$ and offset $d = 1500$. Figure 3.1 depicts the scene and camera setup just described. The projection of the scene onto a left and right image provided true matches shown in Figure 3.2. The "true" homography relating the noise-free image points was computed using (3.2) and the aforementioned information.

The need for a truncated g-inverse

We first demonstrate that if the constraint vector $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ is built from all three scalar constraints as per (3.5), then, in the presence of very small noise in the data, the matrices $\Sigma(\mathbf{x}_i, \boldsymbol{\theta}) = \partial_\mathbf{x}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})\Lambda_{\mathbf{x}_i}\partial_\mathbf{x}\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})^\mathsf{T}$ cannot be reliably inverted with the standard inverse operation. We also show that the problem is remedied when the 2-truncated Moore-Penrose g-inverse of Σ, Σ_2^+, replaces the ordinary inverse. More specifically, we show that, with a particular estimate $\widehat{\boldsymbol{\theta}}$ taken as a seed for FNS, if $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ is properly inverted for all data points $\mathbf{x}_i$, then FNS converges and produces a genuine AML estimate. However, if the inverse of some $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ is ill-conditioned, then FNS is unable to produce a stable estimate.

We conducted several series of 200 experiments, where ideal image data were corrupted by small homogeneous Gaussian noise with standard deviation σ varying in steps between 0 and 0.3 pixels. In each experiment we recorded whether or not FNS converged when all $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}})$ were inverted using the standard inverse, the Moore-Penrose g-inverse or the 2-truncated Moore-Penrose g-inverse. Figure 3.3 provides histograms of the results.

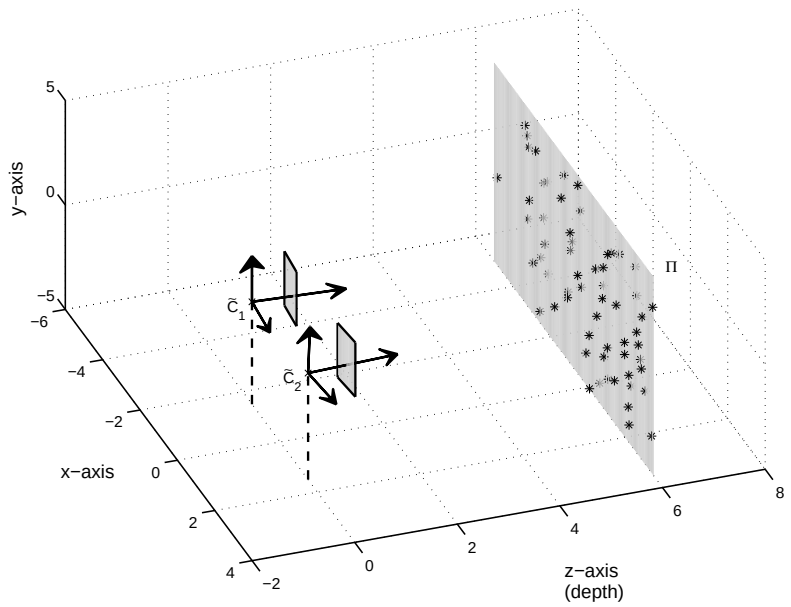


Figure 3.1. A synthetic planar scene of random points photographed by a pair of cameras with non-parallel optical axes.

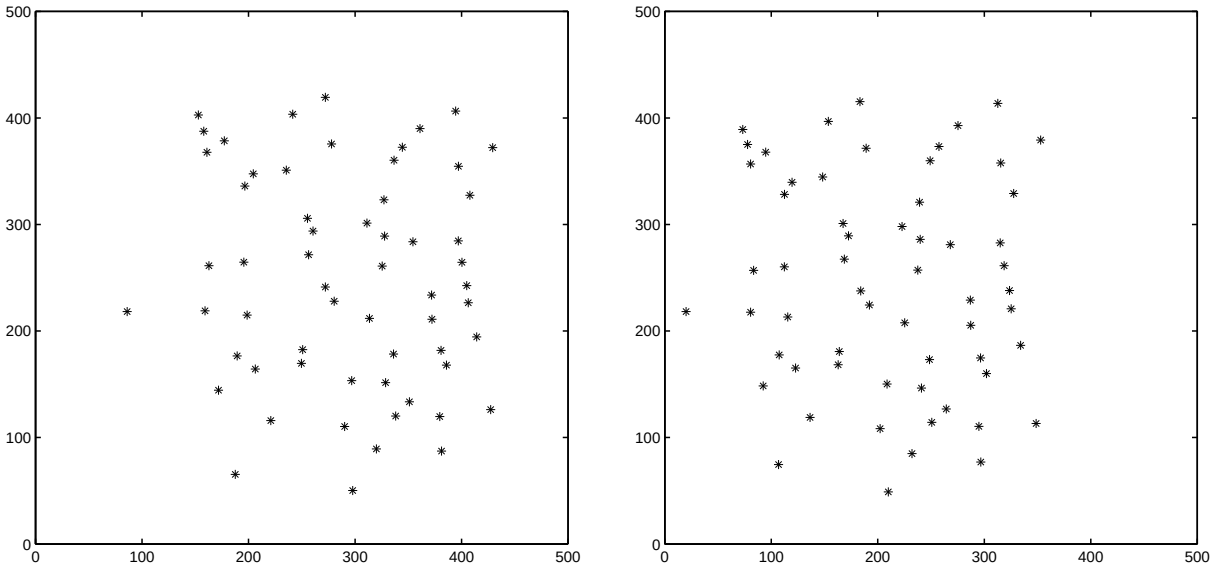


Figure 3.2. Left and right noise-free images acquired by the two cameras shown in Figure 3.1.

For a noise level less than 0.01, the standard inverse applied to the $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}})$ *always* failed to produce a well-conditioned matrix and FNS diverged. As the noise level increased, the Σ's became more often invertible and FNS converged more frequently. In contrast, when the 2-truncated Moore–Penrose g-inverse was employed, the Σ's based on three equations were *always* invertible. The Moore–Penrose g-inverse yielded similar results to those obtained with

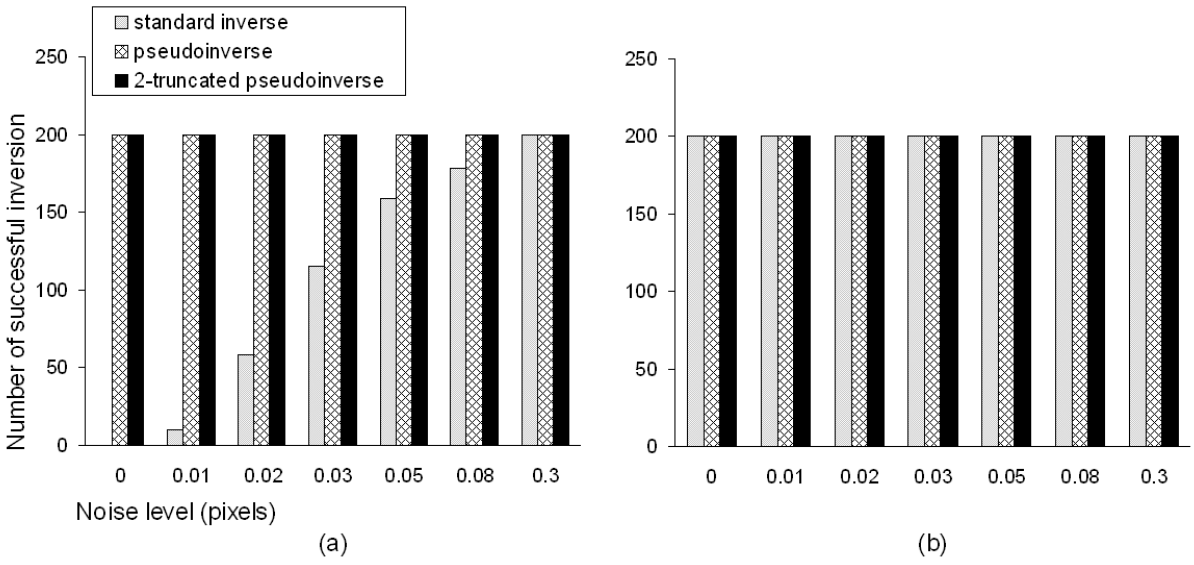


Figure 3.3. Histograms showing the number of times FNS converged over 200 trials at several small noise levels when all $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\mathrm{ALS}})$ were derived from: (a) all three constraint vectors; (b) any choice of two constraint vectors.

the 2-truncated version, which is not surprising given the fact that the MATLAB implementation of the Moore–Penrose g-inverse involves truncation of small singular values. On the other hand, for any choice of pairs of equations of (3.5), the Σ's were always invertible irrespective of the inverse function used or the noise level, and FNS converged to a genuine AML estimate every time.

In each experiment we also tested whether the inversions of the Σ's were affected by the parameter estimate used. More specifically, we inverted $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}})$ and $\Sigma(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}{}^{1-2})$ with the three previously described inverse functions; here $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}{}^{1-2}$ is the AML estimate obtained by choosing the first two equations of (3.5). Results were similar to those shown in Figure 3.3, meaning that the inversion of the Σ's does not depend on the parameter estimate but rather is an intrinsic characteristic of the Σ's.

Performance evaluation

We now present the results of comparative tests carried out to evaluate the accuracy of various AML estimates. The covariances of data were assumed to be the default 4×4 identity matrix ($\Lambda_{\mathbf{x}_i} = \mathtt{I}_{4\times4}$ for all i) corresponding to isotropic homogeneous noise in image point measurement. The basic estimation methods considered were:

- ALS = Algebraic Least Squares,
- NALS = Normalised Algebraic Least Squares,

- FNS = Fundamental Numerical Scheme,
- HEIV = Heteroscedastic Errors-In-Variables scheme,
- GS = Gold Standard.

To describe the GS method, we first note that, with the default covariances, the maximum likelihood cost function given in (2.10) simplifies to

$$J_{\mathrm{ML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \|\mathbf{x}_i - \overline{\mathbf{x}}_i^{\boldsymbol{\theta}}\|^2. \tag{3.8}$$

Recalling that $\mathbf{x}_i = [u_i, v_i, u_i', v_i']$ and $\overline{\mathbf{x}}_i^{\boldsymbol{\theta}} = [\overline{u}_i^{\boldsymbol{\theta}}, \overline{v}_i^{\boldsymbol{\theta}}, \overline{u}_i'^{\boldsymbol{\theta}}, \overline{v}_i'^{\boldsymbol{\theta}}]$ are the concatenation of $\mathbf{m}_i = [u_i, v_i, 1]^\mathsf{T}$ and $\mathbf{m}_i = [u_i', v_i', 1]^\mathsf{T}$, and $\overline{\mathbf{m}}_i^{\boldsymbol{\theta}} = [\overline{u}_i^{\boldsymbol{\theta}}, \overline{v}_i^{\boldsymbol{\theta}}, 1]^\mathsf{T}$ and $\overline{\mathbf{m}}_i'^{\boldsymbol{\theta}} = [\overline{u}_i'^{\boldsymbol{\theta}}, \overline{v}_i'^{\boldsymbol{\theta}}, 1]^\mathsf{T}$, respectively, we see that (3.8) can be written as

$$\begin{aligned} J_{\mathrm{ML}}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n) &= \sum_{i=1}^{n} (u_i - \overline{u}_i^{\boldsymbol{\theta}})^2 + (v_i - \overline{v}_i^{\boldsymbol{\theta}})^2 + (u_i' - \overline{u}_i'^{\boldsymbol{\theta}})^2 + (v_i' - \overline{v}_i'^{\boldsymbol{\theta}})^2 \\ &= \sum_{i=1}^{n} d(\mathbf{m}_i, \overline{\mathbf{m}}_i^{\boldsymbol{\theta}})^2 + d(\mathbf{m}_i', \overline{\mathbf{m}}_i'^{\boldsymbol{\theta}})^2, \end{aligned} \tag{3.9}$$

where $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance between the points $\mathbf{a}$ and $\mathbf{b}$ expressed in inhomogeneous coordinates. The right-hand side of (3.9) is the familiar *reprojection error*. GS is an advanced method [37] for minimising the expression given in (3.9). The scheme seeks to minimise $\sum_{i=1}^{n} d(\mathbf{m}_i, \widehat{\mathbf{m}}_i)^2 + d(\mathbf{m}_i', \widehat{\mathbf{m}}_i')^2$ over all points $\widehat{\mathbf{m}}_i$ and matrices $\widehat{\mathtt{H}}$, with the $\widehat{\mathbf{m}}_i'$ uniquely determined by $\widehat{\mathbf{m}}_i' \simeq \widehat{\mathtt{H}}\widehat{\mathbf{m}}_i$. Our implementation of GS uses the Levenberg-Marquardt algorithm to carry out the minimisation.

FNS and HEIV minimised an AML cost function which incorporated the 2-truncated Moore–Penrose g-inverse function when the full system (3.5) was used or the standard inverse function for any combination of two equations. Both FNS and HEIV estimates were obtained using the stable versions of the algorithms as described in Section 2.2.6 or [18]. The three iterative schemes were initialised with $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$ and operated on raw image data (no Hartley data normalisation applied). It should also be noted that the iterative methods were supplied with similar stopping conditions so as to enable fair comparison.

In each of 200 experiments, one ML estimate was computed as well as four sets of ALS, NALS, AML, and HEIV estimates, each set comprising four different estimates corresponding to the selection of either all three equations of (3.5) or any particular combination of two equations. Figure 3.4 shows the average values of the AML cost function based on three constraint vectors and involving the 2-truncated Moore–Penrose g-inverse, attained at various types of estimate. Our tests revealed that, when evaluated at the estimates delivered by the iterative schemes, the average values of J_{AML} based on three equations, the average values of J_{AML} based on any combination of two equations, and the average values of J_{ML} are practically identical.

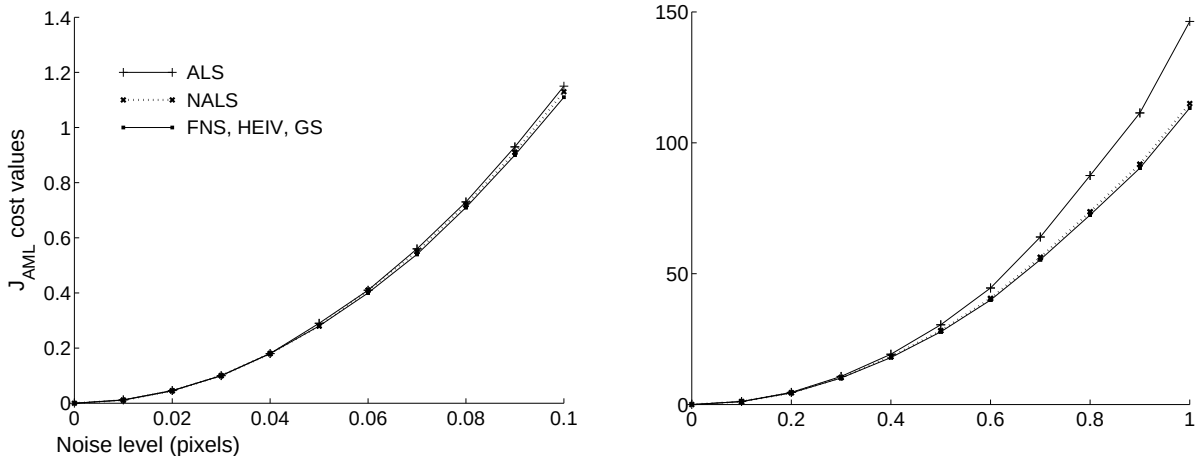


Figure 3.4. Cost values of the J_{AML} function based on three constraint vectors and a 2-truncated g-inverse for: (a) very small noise levels; (b) larger noise levels.

For illustrative purposes, we provide tables of the average performance of the methods when the data were corrupted with 1 pixel of noise.

Table 3.1 shows that the four J_{AML} cost functions achieved very similar values when evaluated at their corresponding minimisers.

	Homography equations			
Methods	$1-2-3$	$1-2$	$1-3$	$2-3$
GS	113.28	113.28	113.28	113.28
ALS	146.38	115.44	247.08	407.74
NALS	114.99	113.29	117.90	119.76
HEIV	113.28	113.28	113.28	113.28
FNS	113.28	113.28	113.28	113.28

Table 3.1. Average J_{AML} cost values for various selection of equations.

On the other hand, Table 3.2 presents the cost values of one particular J_{AML} function evaluated at the four AML estimates. It is seen that each estimate yielded the same cost value. This observation was typical, irrespective of which one of the four AML cost function was selected.

Perhaps the most critical test came from using the maximum likelihood function, J_{ML}. For an estimate $\widehat{\boldsymbol{\theta}}$ obtained by a method other than GS, $J_{\mathrm{ML}}(\widehat{\boldsymbol{\theta}})$ was calculated by minimising $\sum_{i=1}^{n} d(\mathbf{m}_i, \widehat{\mathbf{m}}_i)^2 + d(\mathbf{m}'_i, \widehat{\mathbf{m}}'_i)^2$ over all points $\widehat{\mathbf{m}}_i$, with the $\widehat{\mathbf{m}}'_i$ uniquely determined by the relation $\widehat{\mathbf{m}}'_i \simeq \widehat{\mathtt{H}}\widehat{\mathbf{m}}_i$ and $\widehat{\mathtt{H}}$, representing $\widehat{\boldsymbol{\theta}}$, being kept fixed. Note the difference with GS where the $\widehat{\mathbf{m}}_i$ and $\widehat{\mathtt{H}}$ were allowed to vary simultaneously. Inspecting Table 3.3, we see that FNS

	Homography equations			
Function	$1-2-3$	$1-2$	$1-3$	$2-3$
J_{AML}^{1-2}	113.28	113.28	113.28	113.28

Table 3.2. Average cost values of J_{AML} based on the first two equations of (3.5) and standard inverse.

estimates produced very competitive cost values in comparison to the GS estimate. Now, if Table 3.1 is re-examined, it can be observed that J_{AML} exactly approximated J_{ML} for all types of iterative and non-iterative estimates. Note also that the values of the AML (or ML) function revealed that the data normalisation in the NALS method provided a significant advantage compared to plain ALS. For any combination of equations, the cost values were much lower for the NALS estimates than the ALS ones, and relatively close to that of GS.

		Homography equations			
Methods		$1-2-3$	$1-2$	$1-3$	$2-3$
GS	113.28				
ALS		146.38	115.44	247.08	407.74
NALS		114.99	113.29	117.90	119.76
HEIV		113.28	113.28	113.28	113.28
FNS		113.28	113.28	113.28	113.28

Table 3.3. Average J_{ML} cost values for simulated data.

A timing test is presented in Table 3.4. Unsurprisingly, GS turned out to be by far the slowest of the methods. While it may have been speeded up via the incorporation of sparse-matrix techniques, it was intrinsically slow given the high-dimensionality of its search strategy.

		Homography equations			
Methods		$1-2-3$	$1-2$	$1-3$	$2-3$
GS	1.7				
ALS		0.004	0.004	0.004	0.004
NALS		0.004	0.004	0.004	0.005
HEIV		0.066	0.049	0.049	0.050
FNS		0.093	0.067	0.067	0.067

Table 3.4. Average computation time per homography (in seconds).

3.2.2 Experiments with real image data

Three sequences of images were acquired by rotating a camera about its centre and corresponding points were identified manually. For each sequence, we computed a homography between each pair of consecutive images using five algorithms: ALS, NALS, HEIV, FNS, and GS. As in the simulated experiments of Section 3.2.1, one ML estimate and four sets of ALS, NALS, AML, and HEIV estimates were computed every time, each set comprising four different estimates corresponding to the selection of either all three equations of (3.5) or any particular combination of two equations.

To assess the quality of the various homographies, J_{ML} and J_{AML} cost values were evaluated for the obtained estimates. The algorithms' computation time and number of iterations were also recorded. The three iterative schemes were set as in our synthetic experiments. Note that their timing did not include the initialisation stage. Here, we only report the results for the first homography of each sequence, that is the one computed between the first two views, because the same observations could be made for homographies relating other pairs of images. As illustration, we have used the homographies computed with FNS III (for J_{AML} based on all three equations) to compose panoramic mosaics of the sequences.

IMAX theatre sequence

The three images shown in Figure 3.5 were used to compute a homography. Each input image is 680×450 pixels in size. Starting from the leftmost image, consecutive pairs of images have 258 and 173 matches, respectively.

Figure 3.5. Images of the IMAX theatre sequence.

Numerous contemporary vision applications require real-time processing. To check the potential suitability of FNS for such tasks, another performance indicator that we have examined was the convergence rate of FNS. The convergence rate of an iterative method may be assessed by calculating the relative error

$$e_k = \frac{\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|}{\|\boldsymbol{\theta}_k\|},$$

where $\boldsymbol{\theta}_k$ stands for a parameter estimate at step k and $\boldsymbol{\theta}^*$ is a local minimiser of the cost function underlying the problem of interest. In our tests, $\boldsymbol{\theta}^*$ was taken as the final estimate produced by the FNS method. An algorithm shows linear convergence provided

$$e_{k+1} \leq a\, e_k \qquad \text{when } e_k \text{ is small;} \quad 0 < a < 1.$$

Superlinear convergence is recognised when

$$e_{k+1}/e_k \to 0 \quad \text{for } k \to \infty,$$

and quadratic convergence if

$$e_{k+1} = \mathcal{O}(e_k^2) \quad \text{when } e_k \text{ is small.}$$

Table 3.5 provides information about each iteration step in the FNS minimisation process by listing the J_{AML} residual, the relative norm difference between two successive iterates, and the convergence rate. These results are typical whether all three equations (FNS III) or any two equations (FNS II) were used to compute an AML estimate. It can be seen that the AML residual and the norm difference decreased rapidly. The values of the relative error (third column) indicate that FNS converged quadratically to the solution.

Step	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k
0	224.80	133	1.50×10^{-6}
1	197.76	2.00×10^{-4}	1.40×10^{-3}
2	197.74	2.80×10^{-7}	0.00

Table 3.5. FNS convergence characteristics when computing the first homography in the IMAX sequence. Step 0 gives the initial values for the seed $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$.

When comparing the performance of several methods, Table 3.6 confirms that FNS and HEIV produced estimates on par with GS. Notice that all four FNS estimates yielded almost identical residual values. The J_{ML} cost was computed for each estimate as described in Section 3.2.1. The values were found to match those of their AML counterparts given in Table 3.6. Surprisingly, the NALS method generated a good estimate only when the first two equations of (3.5) were used. For other selections of equations, the NALS solutions were of inferior quality to those of ALS.

GS performed five iterations in total whereas both FNS and HEIV did only two (no matter if two or three equations were employed in the estimation). Timings to compute the various homographies appear in Table 3.7.

	Homography equations			
Methods	$1-2-3$	$1-2$	$1-3$	$2-3$
GS	197.74	197.74	197.74	197.74
ALS	215.60	198.83	292.15	217.65
NALS	224.80	197.76	301.67	226.45
HEIV	197.74	197.74	197.73	197.74
FNS	197.74	197.74	197.73	197.74

Table 3.6. J_{AML} residuals for several homographies between the first two images.

		Homography equations			
Methods		$1-2-3$	$1-2$	$1-3$	$2-3$
GS	46.34				
ALS		0.06	0.03	0.03	0.03
NALS		0.03	0.05	0.05	0.03
HEIV		0.36	0.25	0.23	0.23
FNS		0.51	0.33	0.31	0.33

Table 3.7. Computation time for various homographies (in seconds).

To give a visual appreciation of the accuracy of the AML estimates, we assembled a panorama using the computed homographies derived from all three equations of (3.5), see Figure 3.6.

Figure 3.6. IMAX theatre panoramic mosaic.

Adelaide convention centre sequence

Figure 3.7 shows the images for this sequence. Each image is 640×426 pixels in size. Consecutive pairs of images have 50 and 54 matches, respectively.

Figure 3.7. An Adelaide convention centre sequence.

Some information about each step of FNS III was recorded as the method computed the first homography in this sequence, see Table 3.8. Choosing any pairs of equations (as in FNS II) yielded similar results. Convergence was again quadratic and so only a couple of iterations were needed to reach a solution.

Step	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k
0	116.55	115	3.23×10^{-5}
1	107.68	3.69×10^{-3}	7.60×10^{-3}
2	107.62	2.82×10^{-5}	0.00

Table 3.8. FNS convergence characteristics when computing the first homography in the Adelaide convention centre sequence. Step 0 gives the initial values for the seed $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$.

Table 3.9 confirms that FNS generated as good estimates as GS and HEIV. Note that all four AML estimates yielded closely identical residual values. The value of the reprojection error (from the J_{ML} function) was also checked and found to match its corresponding AML value for each computed estimate. NALS again performed poorer than ALS except when the first two equations of (3.5) were employed. Using a reduced set of equations permits a fractional gain in computation time (Table 3.10) without affecting the accuracy of the solution.

Figure 3.8 shows the mosaic composed from the input images of Figure 3.7.

	Homography equations			
Methods	$1-2-3$	$1-2$	$1-3$	$2-3$
GS	107.63	107.63	107.63	107.63
ALS	112.26	117.36	117.80	112.16
NALS	116.57	107.63	208.48	121.71
HEIV	107.62	107.62	107.62	107.61
FNS	107.62	107.62	107.62	107.61

Table 3.9. J_{AML} residuals for several homographies between the first two images.

		Homography equations			
Methods		$1-2-3$	$1-2$	$1-3$	$2-3$
GS	1.61				
ALS		0.02	0.01	0.00	0.01
NALS		0.02	0.01	0.01	0.00
HEIV		0.09	0.06	0.06	0.05
FNS		0.11	0.06	0.06	0.06

Table 3.10. Computation time for various homographies (in seconds).

Figure 3.8. Panorama of the Adelaide convention centre.

War memorial sequence

The final image sequence considered is shown in Figure 3.9. Each image is 429×640 pixels in size. Consecutive pairs of images have 35, 80, and 67 matches, respectively.

Figure 3.9. Images of the war memorial sequence.

As in previous example sequences, Table 3.11 looks at the behaviour of FNS III when computing the homography between the first two views. Choosing pairs of equations again yielded very similar results to the three-equations case. Only two iterations were necessary for FNS III to attain a solution. The method showed a quadratic convergence rate when applied to these image data.

Step	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k
0	125.58	125	1.28×10^{-5}
1	108.14	1.56×10^{-3}	2.39×10^{-2}
2	108.12	3.83×10^{-5}	0.00

Table 3.11. FNS convergence characteristics when computing the first homography in the war memorial sequence. Step 0 gives the initial values for the seed $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$.

Compared to other iterative methods, FNS produced estimates of similar quality to those of HEIV, and most importantly, to those of GS, see Table 3.12. The J_{ML} values were also found to match those obtained with J_{AML} revealing once more that J_{AML} is a good approximation to J_{ML}. For this particular sequence, NALS always improved on the results of ALS irrespectively of the combination of equations chosen.

In terms of computation time (Table 3.13), HEIV and FNS were the most effective schemes to generate fast solutions while reaching the accuracy of a proper maximum likelihood one.

Figure 3.10 shows the mosaic composed from the input images of Figure 3.9.

	Homography equations			
Methods	$1-2-3$	$1-2$	$1-3$	$2-3$
GS	108.17	108.17	108.17	108.17
ALS	284.90	108.83	285.19	586.43
NALS	125.62	108.14	143.95	256.17
HEIV	108.17	108.12	108.14	108.03
FNS	108.12	108.12	108.14	108.03

Table 3.12. J_{AML} residuals for several homographies between the first two images.

		Homography equations			
Methods		$1-2-3$	$1-2$	$1-3$	$2-3$
GS	1.08				
ALS		0.05	0.01	0.00	0.00
NALS		0.02	0.00	0.01	0.01
HEIV		0.07	0.05	0.04	0.04
FNS		0.10	0.05	0.04	0.04

Table 3.13. Computation time for various homographies (in seconds).

Figure 3.10. War memorial panoramic mosaic.

3.3 Visualising the AML cost function

This section aims at visualising the form of the J_{AML} cost function in vicinity of the AML estimate. The overall plot of the function and its slices are shown through specifically constructed parameter vectors. This analysis can be seen as a safety measure to verify that the solution produced by FNS is indeed the minimiser of the AML cost function. We have decided to use the AML estimate relating the first two images in the Adelaide convention centre sequence to carry out the work. The graphs of other estimates were generated and found to exhibit similar characteristics.

3.3.1 Bracketing the AML minimiser

To see the shape of J_{AML} around $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ a new parameter vector $\tilde{\boldsymbol{\theta}}$ may be defined as

$$\tilde{\boldsymbol{\theta}}(\alpha) = (1-\alpha)\,\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} + \alpha\,\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}.$$

Note that $J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}})$, the cost value at the estimated AML solution, is obtained when $\alpha = 0$. Here, both vectors $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$ are set to unit norm. Choosing the range of α values must be done with care. One might naively think that the minimiser $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ could be bracketed in a range as small as $(1-\varepsilon)\,\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} < \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} < (1+\varepsilon)\,\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, where ε is a tolerance value close to machine floating-point precision. This is not the case! The shape of the AML function near $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ is mandated by its Taylor expansion

$$J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}}) \approx J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) + \frac{1}{2}\tilde{\boldsymbol{\theta}}^{\mathsf{T}}\mathtt{H}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}})\tilde{\boldsymbol{\theta}}.$$

If the required tolerance ε is too small, there will be no gain in bracketing $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ closer [92, Chap. 10]. We now outline a procedure to set upper and lower bounds on α.

Suppose we wish to have a point at a distance k from $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, that is,

$$\begin{aligned}
\| \,[(1-\alpha)\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} + \alpha\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}] - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} \,\| &= k \\
\| \,\alpha(\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) \,\| &= k \\
|\alpha| \,\|\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}\| &= k \\
|\alpha| &= \frac{k}{\|\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}\|}
\end{aligned}$$

so

$$\frac{-k}{\|\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}\|} \le \alpha \le \frac{k}{\|\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}\|}. \tag{3.10}$$

Let

$$f(\alpha) = J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}}(\alpha)) \tag{3.11}$$

be the graph of J_{AML} in vicinity of the AML estimate. The form of f is shown in Figure 3.11 where $k = 0.11$ and α belongs to the interval $[-9.6\times10^{-4}, 9.6\times10^{-4}]$ using the bracketing rule (3.10). For this range of α values, the difference in function values between any two successive points on the curve was very small, about 5×10^{-5}, and matched the termination condition used to compute $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$. As expected, the shape of J_{AML} turned out to be a parabola in the neighbourhood of $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, which confirms the quadratic nature of the function. Clearly, $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ is the lowest point on the curve, and therefore the minimiser of J_{AML}.

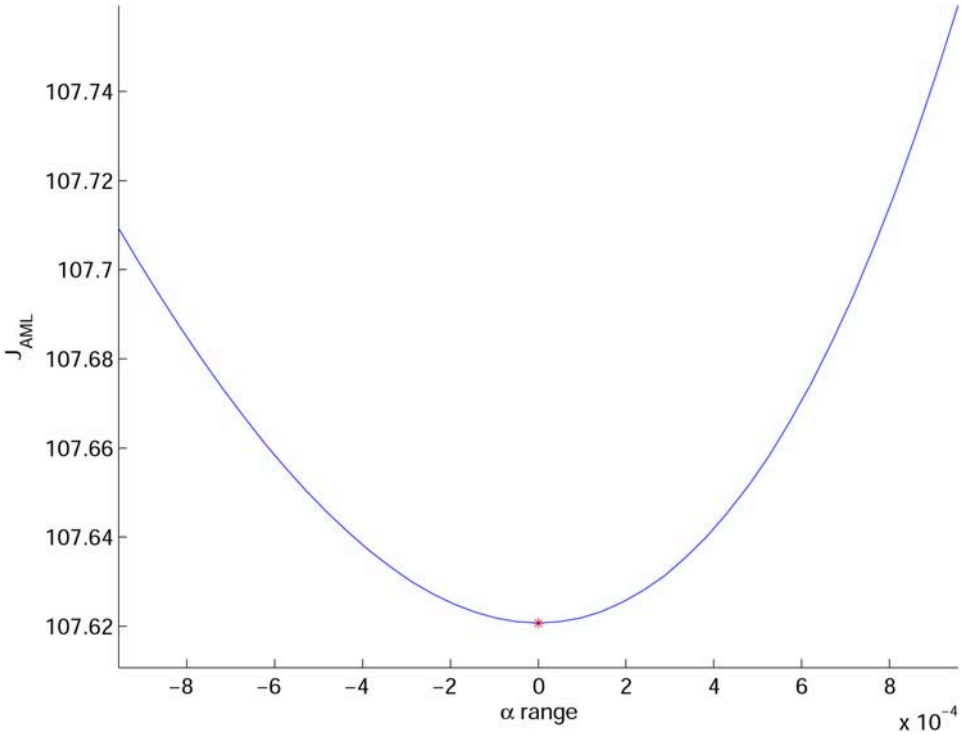


Figure 3.11. Graph of $f(\alpha)$.

It is possible to view 2-D cross-sections of the cost function by varying one component of $\tilde{\boldsymbol{\theta}}$ and fixing others. The search space for the homography fitting problem is 9-dimensional so there are nine graphs, one for each element of $\tilde{\boldsymbol{\theta}}$. The j-th graph shows the value of $J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}})$ where the k-th component of $\tilde{\boldsymbol{\theta}}$ is given by

$$\tilde{\boldsymbol{\theta}}^k(\alpha) = \begin{cases} (1-\alpha)\,\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} + \alpha\,\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} & \text{if } k = j, \\ \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} & \text{otherwise.} \end{cases} \tag{3.12}$$

The resulting cross-sections are presented in Figure 3.12. The same value of $k = 0.11$ was also used here to find a range for α automatically. Each graph shows a parabola with minimum at $\alpha = 0$, that is, where $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ is located.

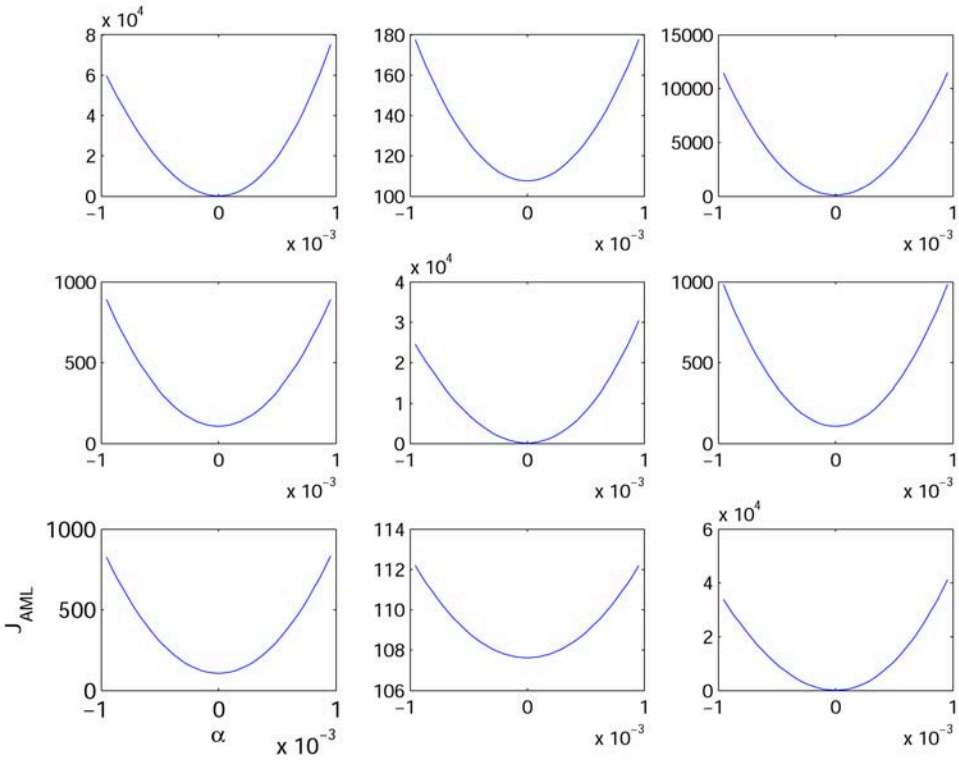


Figure 3.12. Cross-sections of $f(\alpha)$.

3.3.2 Examining the derivative of the AML cost function

We extend the previous work to study the derivative of the AML cost function.

Given a vector $\tilde{\boldsymbol{\theta}}$ satisfying (3.12), the derivative of f can be obtained by applying the chain rule to the right-hand side of (3.11), that is,

$$f'(\alpha) = \partial_{\tilde{\boldsymbol{\theta}}} J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}}) \partial_\alpha \tilde{\boldsymbol{\theta}},$$

hence,

$$f'(\alpha) = \partial_{\tilde{\boldsymbol{\theta}}} J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}})(\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}), \tag{3.13}$$

where

$$\partial_{\tilde{\boldsymbol{\theta}}} J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}}) = 2(\mathtt{X}_{\tilde{\boldsymbol{\theta}}} \tilde{\boldsymbol{\theta}})^{\mathsf{T}}$$

is the Jacobian of J_{AML} at $\tilde{\boldsymbol{\theta}}$ obtained from (2.20). We chose to plot J_{AML} based on all three homography equations, so $\mathtt{X}_{\tilde{\boldsymbol{\theta}}} = \mathtt{M}_{\tilde{\boldsymbol{\theta}}} - \mathtt{N}_{\tilde{\boldsymbol{\theta}}}$, where $\mathtt{M}_{\tilde{\boldsymbol{\theta}}}$ and $\mathtt{N}_{\tilde{\boldsymbol{\theta}}}$ were computed from (2.31) with the rank r set to 2.

The graph of the AML cost function derivative is shown in Figure 3.13 where $f'(\alpha)$ was written as $df/d\alpha$ for clarity. Since the minimum of a function is an inflection point, the derivative changes sign. To locate this point easily on the graph, we have plotted the absolute value of the function derivatives. Upon examination, we see that an inflection occurs at $\alpha = 0$, that is at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, the minimiser of J_{AML}.

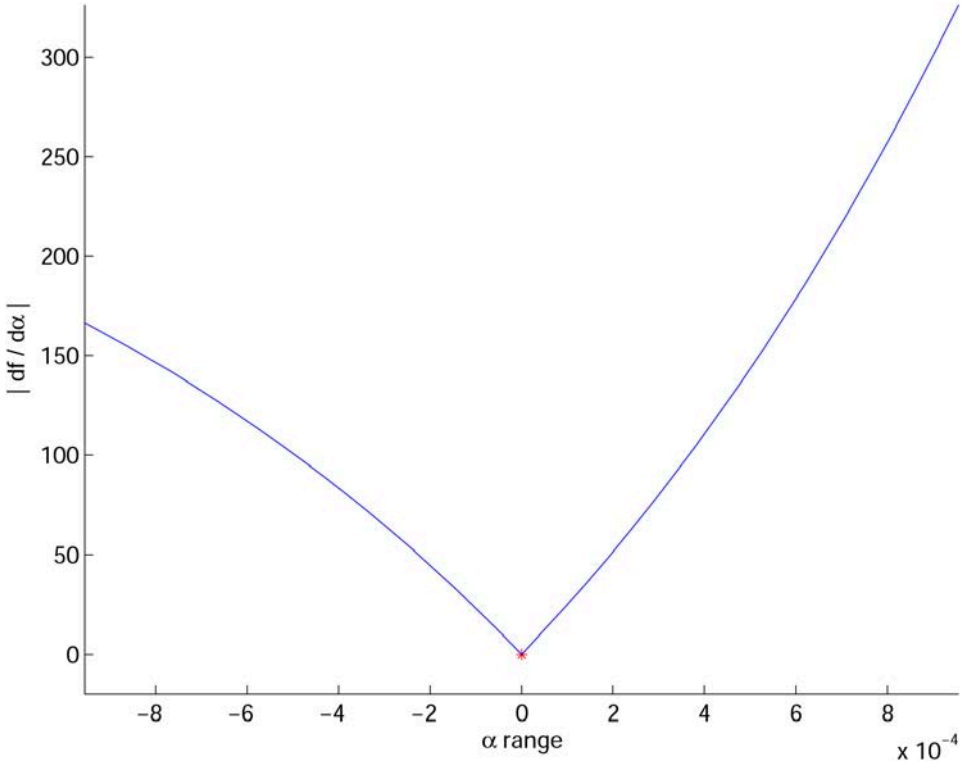


Figure 3.13. Graph of the absolute value of $f'(\alpha)$.

Another formula should be used if we want to examine 2-D cross-sections of the AML cost function derivative. At the difference with (3.13) the derivative applies to a component of the vector $\tilde{\boldsymbol{\theta}}$ and not the whole vector. The expression for the Jacobian $\partial_{\tilde{\boldsymbol{\theta}}} J_{\mathrm{AML}}(\tilde{\boldsymbol{\theta}})$ remains the same, however $\partial_\alpha \tilde{\boldsymbol{\theta}}$ has now k-th component

$$\partial_\alpha \tilde{\boldsymbol{\theta}}^k(\alpha) = \begin{cases} \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} - \widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}} & \text{if } k = j, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3.14 shows the cross-sections of the absolute value of $f'(\alpha)$. The minimum on each graph occurs at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ when $\alpha = 0$.

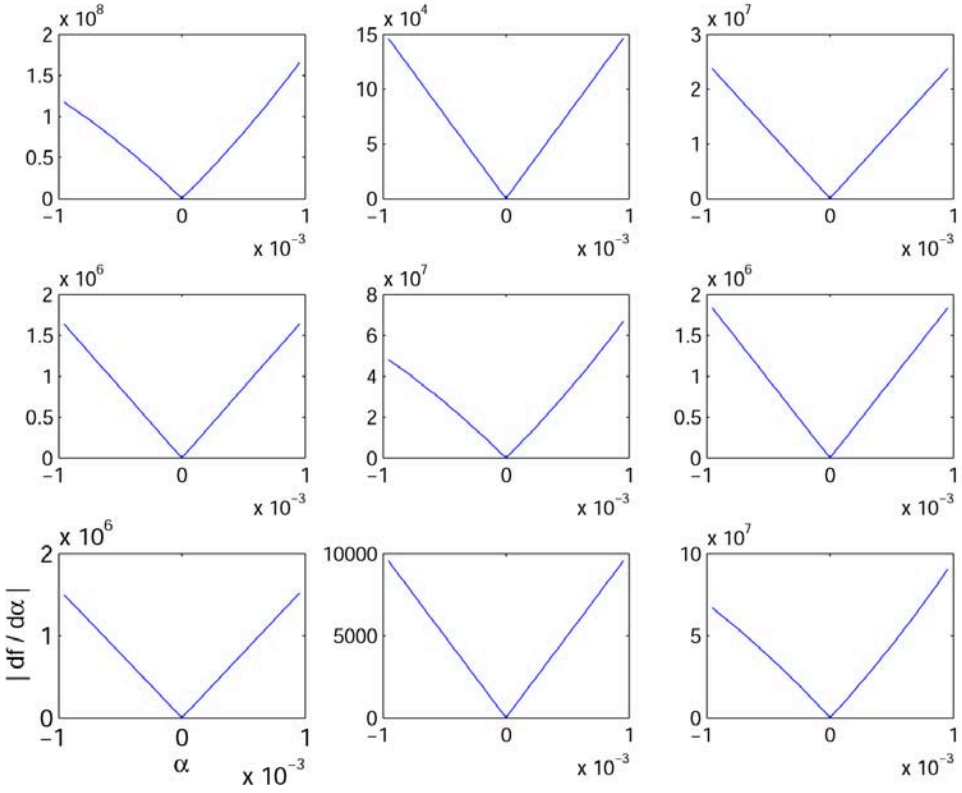


Figure 3.14. Cross-sections of the absolute value of $f'(\alpha)$.

3.4 Conclusion

The aim of this chapter was to demonstrate empirically the validity of the theory put forward when developing the Fundamental Numerical Schemes in Chapter 2. Choosing the homography estimation model, it was shown that the difficulty stemming from the presence of ill-conditioned matrices in the standard AML cost function can be eliminated if the ordinary inverses of the ill-conditioned matrices are replaced by generalised inverses that are continuous. Tests revealed that no specific generalised inverse, as long as it is continuous, is particularly favoured as far as the accuracy of the solution is concerned. This finding lent a particular prominence to cost functions involving generalised inverses that correspond to a minimum number of linearly independent equations, as these functions are differentiable and can be optimised with well-established methods of the likes of FNS or HEIV. It was seen that the modified FNS methods, FNS II and FNS III, produced solutions of similar accuracy and matched that of a maximum likelihood estimate while being faster techniques than GS.

Chapter 4

Geometry and Algebra of the Trifocal Tensor

If we picture the same scene from different viewpoints, we expect the various images to be related in some ways, even if the position, orientation and nature of the cameras differ a lot. The aim of this chapter is to expose the underlying relationships linking three partially overlapping views of a scene. It turns out that a certain valence-3 tensor, the trifocal tensor, serves as an adequate descriptor of the inter-image geometry. This tensor allows the recovery of the scene up to a projective transformation in 3-D and is computable from image correspondences alone without requiring knowledge of the motion between views or calibration of the cameras.

The higher accuracy achieved by using the trifocal tensor of three views compared to a fundamental matrix of a pair of views makes it a powerful tool especially in the domain of structure-and-motion recovery. Applications include accurate camera motion estimation [94, 96], radial distortion [85], augmented reality [95, 99], camera calibration and self-calibration [2, 22, 34], robot navigation [20], scene reconstruction [6, 32, 67, 71], motion segmentation [87, 90, 97], image stabilization [74], novel view synthesis [3], and more.

At the beginning, we briefly expose the perspective camera model and the epipolar geometry which underlie the intrinsic projective geometry of one and two views of a scene respectively. The rest of this chapter is then dedicated to the study of three-view geometry and the presentation of the trifocal tensor. This tensor will be decomposed into smaller, matricial objects called the tensorial slices. These slices are effectively matrices embodying certain geometric operations between canonical basis vectors in the images. We will show that these matrices provide a genuine canonical representation of the trifocal geometry because any general inter-view mapping can be expressed as their linear combination. The intrinsic properties of the slices will then be turned into useful sets of algebraic constraints that must be satisfied to guarantee a consistency between the computed tensor and the underlying trifocal geometry. Our investigation

of the internal trifocal constraints will primarily review the contributions of Hartley, Faugeras, Canterakis, and Ressl on this topic.

An overlap with existing literature is inevitable, however the main value of this chapter is to show a systematic way to derive the tensorial slices and to examine the geometric and algebraic properties of the trifocal tensor by gradually contracting it with one, two, and three image features. The most tangible contribution comes from the work on expressing the various internal constraints in one common framework, the one proposed by Hartley [37], in order to see and understand these constraints from a unified viewpoint.

4.1 Monocular vision and stereo vision

The first section introduces the framework of projective geometry through a presentation of the single image geometry and the perspective camera model. This is extended in the next section where stereo vision is tackled—the case when two perspective views of a scene are available. The first two sections review some of the key concepts while establishing the projective framework necessary to discuss the third and main part of this chapter about trinocular vision.

4.1.1 Single view and the perspective camera

A view is a two-dimensional snapshot of the world taken by a camera at a given time. Mathematically, a camera is a genuine geometric device which constructs planar images of the three-dimensional world by a projection through an optical centre. In the sequel, an image is considered as a two-dimensional projective space $\mathbb{P}^2$ and the scene as a three-dimensional projective space $\mathbb{P}^3$. The phenomenon of projection from 3-D to 2-D can be modelled by the action of a *perspective* camera expressed in the form of a 3×4 matrix

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I}_{3\times 3} \mid -\tilde{\mathbf{C}}], \tag{4.1}$$

where $\mathtt{K}$ is a 3×3 calibration matrix as introduced in (3.7) and $\mathtt{R}$ is a 3×3 matrix representing counterclockwise rotations about the three principal axes with origin the optical centre $\tilde{\mathbf{C}}$ of the camera. The same projection matrix can be written as

$$\mathtt{P} = \begin{bmatrix} \boldsymbol{\Gamma}_1 \\ \boldsymbol{\Gamma}_2 \\ \boldsymbol{\Gamma}_3 \end{bmatrix}, \tag{4.2}$$

where $\boldsymbol{\Gamma}_i$ is the i-th row vector of $\mathtt{P}$. Geometrically, each of these vectors represents the coordinates of a plane in 3-D space. These three planes are called the *principal planes*, or *projection*

planes, of camera $\mathtt{P}$ and are depicted in Figure 4.2(b). These planes will play an important role when deriving internal constraints of the trifocal tensor. With matrix $\mathtt{P}$, the central projection from a scene point $\mathbf{M}$ to its image point $\mathbf{m}$, both represented by homogeneous vectors, is given by

$$\mathbf{m} \simeq \mathtt{P}\mathbf{M}.$$

The symbol $\simeq$ indicates that the equality is true up to a non-zero scalar factor. Note that projection matrix $\mathtt{P}$ and world point $\mathbf{M}$ are always defined up to a 4×4 non-singular homography matrix transformation $\mathtt{H}$, often referred to as the *3-D projective ambiguity*, which leave the image point unchanged

$$\mathbf{m} \simeq \mathtt{P}\mathbf{M} \simeq \mathtt{P}\mathtt{H}^{-1}\mathtt{H}\mathbf{M} \simeq \mathtt{P}'\mathbf{M}'. \tag{4.3}$$

These relations show that different pairs of projection and world point can give the same image point if they transform according to $\mathtt{P}' \simeq \mathtt{P}\mathtt{H}^{-1}$ and $\mathbf{M}' \simeq \mathtt{H}\mathbf{M}$. The remaining part of this section is devoted to the study of several properties of principal planes which will become relevant in Section 4.4.

First, note that the back-projection of an image line $\boldsymbol{l}$ is a plane Π defined by the camera centre $\tilde{\mathbf{C}}$ and the line $\boldsymbol{l}$ as shown in Figure 4.1.

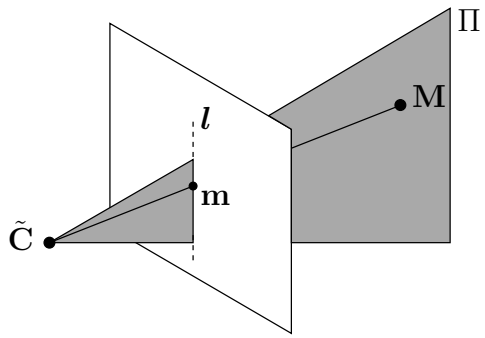


Figure 4.1. An image line back-projects to a plane in space.

A scene point $\mathbf{M}$ lies on a plane Π of homogeneous coordinates $\pi_{\boldsymbol{l}}$ if and only if $\mathbf{M}^{\mathsf{T}}\pi_{\boldsymbol{l}} = 0$. Under this assumption the projected point $\mathbf{m} \simeq \mathtt{P}\mathbf{M}$ belongs to the image line $\boldsymbol{l}$. This means that $\mathbf{m}^{\mathsf{T}}\boldsymbol{l} = \mathbf{M}^{\mathsf{T}}\mathtt{P}^{\mathsf{T}}\boldsymbol{l} = 0$, which suggests that the projection plane Π of line $\boldsymbol{l}$ has coordinates

$$\pi_{\boldsymbol{l}} = \mathtt{P}^{\mathsf{T}}\boldsymbol{l}. \tag{4.4}$$

We now show a particular application of this result. Consider the length-3 canonical basis vectors $\mathbf{e}_i$ ($i = 1, 2, 3$) of $\mathbb{R}^3$, which have their i-th entry unital and all other entries zero. Suppose the line $\boldsymbol{l}$ in (4.4) has homogeneous coordinates given by one of these vectors. Back-projecting $\boldsymbol{l}$ gives the plane

$$\boldsymbol{\Gamma}_i^{\mathsf{T}} = \mathtt{P}^{\mathsf{T}}\mathbf{e}_i.$$

This simple arithmetic demonstrates that the canonical vector $\mathbf{e}_i$ interpreted as an image line belongs to the principal plane $\mathbf{\Gamma}_i$, or alternatively, the back-projection of the line $\mathbf{e}_i$ is the principal plane $\mathbf{\Gamma}_i$. This is summarised in the next proposition.

Proposition 1 *The principal plane $\mathbf{\Gamma}_i$ intersects the image plane in the line with homogeneous coordinates given by the vector $\mathbf{e}_i$ in the canonical basis of $\mathbb{R}^3$.*

The optical centre is the unique point $\tilde{\mathbf{C}}$ which satisfies $\mathtt{P}\,\tilde{\mathbf{C}} = \mathbf{0}$. Therefore, this point lies at the intersection of the three planes $\mathbf{\Gamma}_i$. In the Grassmann-Cayley algebra, the intersection of projective subspaces is defined by the meet operator $\triangle$ [5, 24, 77]. So, the optical centre $\tilde{\mathbf{C}}$ may be computed as $\mathbf{\Gamma}_1 \triangle \mathbf{\Gamma}_2 \triangle \mathbf{\Gamma}_3$, see Figure 4.2(b).

It is essential for later work to also consider the intersection of pairs of projection planes. These intersections give rise to three special world lines,

$$\boldsymbol{\mathcal{R}}_1 = \mathbf{\Gamma}_2 \triangle \mathbf{\Gamma}_3,\ \boldsymbol{\mathcal{R}}_2 = \mathbf{\Gamma}_3 \triangle \mathbf{\Gamma}_1,\ \boldsymbol{\mathcal{R}}_3 = \mathbf{\Gamma}_1 \triangle \mathbf{\Gamma}_2, \tag{4.5}$$

called the *principal rays*, or *projection rays*, of camera $\mathtt{P}$. These three lines meet at the optical centre $\tilde{\mathbf{C}}$ and exhibit the following property.

Proposition 2 *The principal ray $\boldsymbol{\mathcal{R}}_i$ intersects the image plane at the point with homogeneous coordinates given by the vector $\mathbf{e}_i$ in the canonical basis of $\mathbb{R}^3$.*

Proof. A general scene point $\mathbf{M}$, distinct from the optical centre $\tilde{\mathbf{C}}$, belongs to the i-th principal plane $\mathbf{\Gamma}_i$ if and only if $\mathbf{M}^\mathsf{T}\mathbf{\Gamma}_i = 0$. Thus, the image point $\mathbf{m} \simeq \mathtt{P}\mathbf{M}$ has its i-th coordinate equal to zero. Suppose that the zero entries of basis vector $\mathbf{e}_i$ are designated by the indices j and k, the i-index accounting for the unital entry, such that i, j, k take distinct values in the set $\{1, 2, 3\}$. With this convention, it follows that vector $\mathbf{e}_i$, viewed as an image point, corresponds to the projection of a 3-D point $\mathbf{M}$ belonging to both principal planes $\mathbf{\Gamma}_j$ and $\mathbf{\Gamma}_k$ since its j and k entries are zero. With $\mathbf{M} \neq \tilde{\mathbf{C}}$, this situation is only possible if $\mathbf{M}$ lies on the intersection of both planes, that is, the principal ray $\boldsymbol{\mathcal{R}}_i$ by definition. □

An alternative understanding of this propositon is that the optical ray back-projected from the camera centre $\tilde{\mathbf{C}}$ through the image point $\mathbf{e}_i$ is the principal ray $\boldsymbol{\mathcal{R}}_i$. Figure 4.2 illustrates how the geometrical entities presented in Propositions 1 and 2 arise in the camera system.

The geometric interpretations of the canonical vectors $\mathbf{e}_i$ are listed in Table 4.1. Although these considerations seem rather trivial at this stage, they will be important in Section 4.4 as they underpin the geometric properties of the tensorial slices.

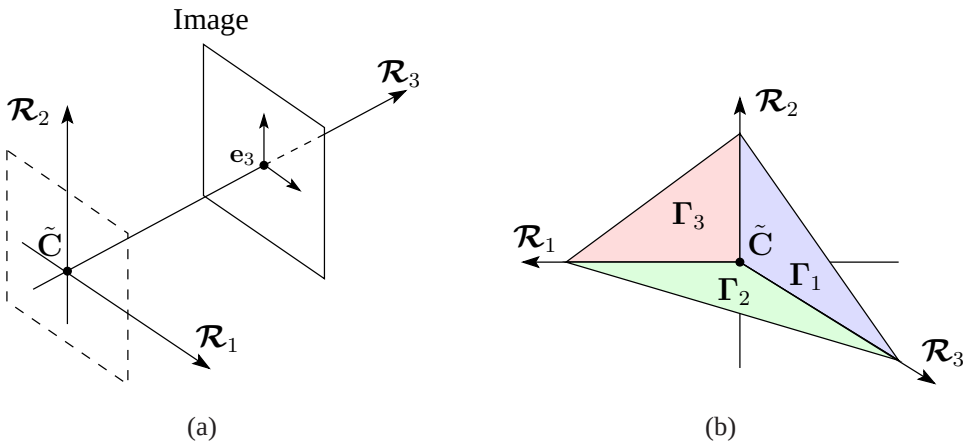


Figure 4.2. The perspective camera model with (a) principal rays $\mathcal{R}_i$; (b) principal planes $\boldsymbol{\Gamma}_i$.

Vector	Image point	Image line
$\mathbf{e}_1 = [1,0,0]^\mathsf{T}$	on the line $\mathcal{R}_1$	on the plane $\boldsymbol{\Gamma}_1$
$\mathbf{e}_2 = [0,1,0]^\mathsf{T}$	on the line $\mathcal{R}_2$	on the plane $\boldsymbol{\Gamma}_2$
$\mathbf{e}_3 = [0,0,1]^\mathsf{T}$	on the line $\mathcal{R}_3$	on the plane $\boldsymbol{\Gamma}_3$

Table 4.1. Geometric interpretation of the canonical basis vectors of $\mathbb{R}^3$.

4.1.2 Stereo vision and the fundamental matrix

A ubiquitous task in computer vision is the computation of the projections that gave rise to a sequence of images. Often it is assumed that the sequence is acquired by a single perspective camera moving along some trajectory in space. Stereo vision is the first instance of multiple view geometry, where a scene is pictured by two cameras. The inherent geometric constraints between image points across the two views are described by the so-called *epipolar geometry*.

Assuming a perspective camera model as introduced in the previous section, suppose that the left and right images arise from projection matrices $\mathtt{P}$ and $\mathtt{P}'$ with origin the camera centres $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{C}}'$ respectively. A homogeneous 3-D point $\mathbf{M}$ is projected onto the left image plane with $\mathtt{P}$ to give an image point $\mathbf{m}$, and onto the right image with $\mathtt{P}'$ to give $\mathbf{m}'$. The pair of image points $\mathbf{m}$ and $\mathbf{m}'$ are said to be in correspondence as they are both projections of the same 3-D point $\mathbf{M}$. This is shown in Figure 4.3.

The projection in one view of the camera centre of the other view is a point known as the *epipole*. We define the left-image epipole as $\mathbf{e} \simeq \mathtt{P}\tilde{\mathbf{C}}'$ and the right-image one as $\mathbf{e}' \simeq \mathtt{P}'\tilde{\mathbf{C}}$. The epipoles may or may not lie in the images depending on the spatial distribution of the cameras. The optical ray going through camera centre $\tilde{\mathbf{C}}$ and image point $\mathbf{m}$, denoted $(\tilde{\mathbf{C}}\mathbf{m})$, projects

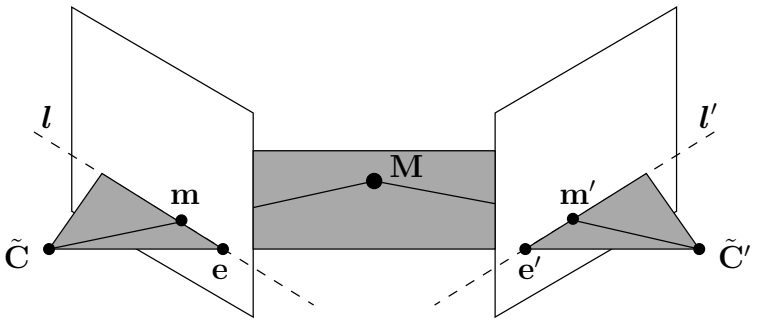


Figure 4.3. Epipolar geometry between two views.

into a line l' in the second view, the *epipolar line* associated with $\mathbf{m}$. Since this line is the projection of both $\tilde{\mathbf{C}}$ and $\mathbf{m}$, l' joins the point $\mathbf{m}'$ and the epipole $\mathbf{e}'$. A similar relation applies to the epipolar line l in the first view which is the image of the ray $(\tilde{\mathbf{C}}'\mathbf{m}')$ joining the point $\mathbf{m}$ and epipole $\mathbf{e}$.

The next two sections will present an algebraic derivation of the fundamental matrix which embodies the mapping between the point $\mathbf{m}$ and its epipolar line l'. Elements in this derivation will come useful later in the chapter.

Point transfer via a plane

Consider a plane Π in 3-D space not passing through either of the camera centres $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{C}}'$. The optical ray back-projected from $\tilde{\mathbf{C}}$ through the image point $\mathbf{m}$ meets the plane Π in a point $\mathbf{M}$, which is then projected into a point $\mathbf{m}'$ in the second image. This procedure is known as a *point transfer via the plane* Π and is depicted in Figure 4.4. Since $\mathbf{M}$ lies on the optical ray through $\mathbf{m}$, the projected point $\mathbf{m}'$ must lie on the epipolar line l' corresponding to the image of that ray. When both points $\mathbf{m}$ and $\mathbf{m}'$ are in homogeneous coordinates, the mapping from $\mathbf{m}$ to $\mathbf{m}'$ is given by a 3×3 homography $\mathtt{H}_\Pi$ induced by the plane Π such that $\mathbf{m}' \simeq \mathtt{H}_\Pi \mathbf{m}$.

Constructing epipolar lines

Because the exact depth of world point $\mathbf{M}$ is not known, it is not possible to predict the position of the correspondent $\mathbf{m}'$ of $\mathbf{m}$. However, geometrically, the location of $\mathbf{M}$ is not arbitrary: $\mathbf{M}$ has to lie on the optical ray $(\tilde{\mathbf{C}}\mathbf{m})$ and therefore $\mathbf{m}'$ is on the projection of that ray in the second image, the epipolar line l'. Since the epipole $\mathbf{e}'$ also belongs to l', we can write this line as

$$l' \simeq \mathbf{e}' \times \mathbf{m}' \simeq [\mathbf{e}']_\times \mathbf{m}', \tag{4.6}$$

where the definition and properties of the matrix $[.]_\times$ can be found in Appendix D. Substituting $\mathbf{m}'$ with $\mathtt{H}_\Pi \mathbf{m}$, we obtain

$$l' \simeq [\mathbf{e}']_\times \mathtt{H}_\Pi \mathbf{m}.$$

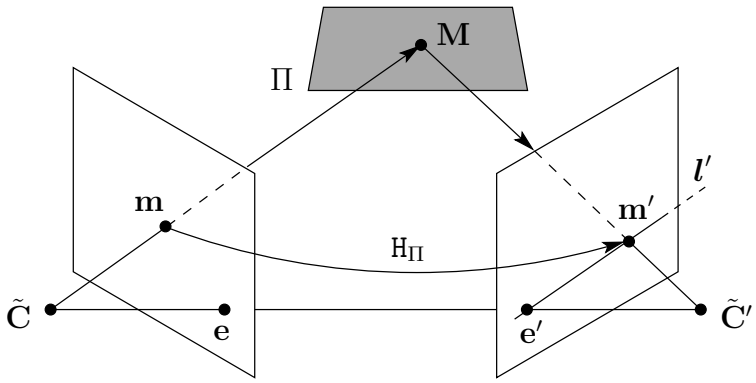


Figure 4.4. Two-view point transfer via a plane in space.

The mapping

$$\mathtt{F}_{21} \simeq [\mathbf{e}']_{\times}\mathtt{H}_{\Pi}$$

is a 3×3 matrix of rank 2 termed the *fundamental matrix*. It follows that

$$\boldsymbol{l}' \simeq \mathtt{F}_{21}\mathbf{m}. \tag{4.7}$$

With $\mathbf{m}'$ being on $\boldsymbol{l}'$, we have $\mathbf{m}'^{\mathsf{T}}\boldsymbol{l}' = 0$ and the *epipolar constraint*

$$\mathbf{m}'^{\mathsf{T}}\mathtt{F}_{21}\mathbf{m} = 0. \tag{4.8}$$

Interchanging the role of the images, a right-image point $\mathbf{m}'$ has associated epipolar line $\boldsymbol{l} \simeq \mathtt{F}_{12}\mathbf{m}'$, where $\mathtt{F}_{12}$ is the fundamental matrix from view 2 to view 1, and epipolar constraint

$$\mathbf{m}^{\mathsf{T}}\mathtt{F}_{12}\mathbf{m}' = 0.$$

Comparison of this expression with (4.8) shows that one equation is the transpose of the other, therefore

$$\mathtt{F}_{21} = \mathtt{F}_{12}^{\mathsf{T}}. \tag{4.9}$$

In practice, the epipolar constraint is a central element in many algorithms designed to accurately estimate the fundamental matrix [17, 23, 36, 48, 59, 66, 86, 98]. Two compelling reasons to solve for this matrix are that it reduces the search space for $\mathbf{m}'$ along the line $\boldsymbol{l}'$ using (4.7) and also that both scene projections can be retrieved from it.

It is worth noting that if only measurements data are available, then it is not possible to recover the true projections that gave rise to the images. This is prevented by the inherent ambiguity of 3-D space whereby projections are defined up to a common projective transformation as described in (4.3). A standard technique to eliminate this problem is to set the projection matrices in *canonical form* such that the first projection has the form

$$\bar{\mathtt{P}} = [\mathtt{I}_{3\times 3} \mid \mathbf{0}].$$

Projections $\mathtt{P}$ and $\mathtt{P}'$ are always amenable to canonical form by carrying out the operation $\bar{\mathtt{P}} \simeq \mathtt{P}\mathtt{H}^{-1}$ and $\bar{\mathtt{P}}' \simeq \mathtt{P}'\mathtt{H}^{-1}$ for

$$\mathtt{H} = \begin{bmatrix} \mathtt{P} \\ \mathbf{w} \end{bmatrix} \tag{4.10}$$

and $\mathbf{w}$ a non-zero row vector of length 4. The matrix $\mathtt{H}$ is a 4×4 matrix defined up to a scale factor and so has 15 *degrees of freedom*. Each projection matrix represents an *uncalibrated* camera with 11 degrees of freedom accounting for the 12 elements of the interior and exterior orientations modulo an overall scale. Because the same projective transformation applies to each of the two projections, the degrees of freedom of two cameras are

$$11 \times 2 - 15 = 7. \tag{4.11}$$

This result may be generalised to an arbitrary number of cameras more than two.

Choosing the first projection in canonical form constrains the form of the second to comply to the epipolar geometry. It can be checked in [7, Chap. 4] or [37, Chap. 8] that the following proposition holds.

Proposition 3 *When* $\mathtt{P} = [\mathtt{I}_{3\times3} \mid \mathbf{0}]$ *and* $\mathtt{P}' = [\mathtt{Q} \mid \mathbf{q}]$*, then the fundamental matrix for such a pair of projections is*

$$\mathtt{F} = [\mathbf{q}]_\times \mathtt{Q}. \tag{4.12}$$

Conversely, if $\mathtt{P} = [\mathtt{I}_{3\times3} \mid \mathbf{0}]$ *and* $\mathtt{F} = [\mathbf{q}]_\times \mathtt{Q}$*, then the second projection has the form*

$$\mathtt{P}' = [\mathtt{Q} \mid \mathbf{q}]. \tag{4.13}$$

4.2 Trinocular vision and the trifocal tensor

Suppose that we acquire a third view of the same scene and label the images from left to right by Ψ_1 to Ψ_3. It seems a priori sufficient to describe the geometry of three views, or *trifocal geometry*, from the perspective of two views by considering the epipolar geometry between each of the three pairs of images. However, it turns out that the number of situations where the scene points cannot be recovered grow rapidly and this causes problems to predict the location of some image points. An interesting analysis of the restrictions of fundamental matrices to describe the trifocal geometry is available in [24, Chap. 8]. The quest for a more global descriptor of the trifocal geometry leads back to analysing the intrinsic relations between image features.

In difference to the case of two cameras, the richness of the trifocal geometry comes from its capacity to capture line correspondences. Relations between triplets of image points or combinations of lines and points spawn from the more general incidence between lines.

Original work on line correspondences appeared in [82, 93] for the special case of calibrated cameras and was further developed in [39] for the uncalibrated case.

In the sections that follow we present a standard derivation of the trifocal tensor edified on the relations between a triplet of image lines and their corresponding world primitive. In addition, we show three parameterisations of the tensor which will serve to derive techniques relevant to its computation.

4.2.1 Parameterisation from three projections

The trifocal tensor naturally arises by considering triplets of lines in correspondence across the images. Consider a world line $\boldsymbol{L}$ imaged in three views in the lines $\boldsymbol{l}$, $\boldsymbol{l}'$, and $\boldsymbol{l}''$ as illustrated in Figure 4.5. Each scene projection is defined up to a 3-D homography as seen in (4.3). This implies that, if we knew the set of projections which gave rise to the images Ψ_1 to Ψ_3, we could apply a 4×4 non-zero transformation in the style of (4.10) to obtain a set in canonical form without changing the original images or their relationships. So, without loss of generality, we may choose the camera projections for the three views as

$$\mathtt{P} = [\mathtt{I}_{3\times 3} \mid \mathbf{0}],\ \mathtt{P}' = [\mathtt{A} \mid \mathbf{e}'] \ = [a_i^j],\ \text{and}\ \mathtt{P}'' = [\mathtt{B} \mid \mathbf{e}''] \ = [b_i^k]. \tag{4.14}$$

Elements a_i^j and b_i^k denote the entries (j, i) and (k, i) in the second and third projections respectively. Quantities $\mathtt{A}$ and $\mathtt{B}$ are 3×3 matrices describing infinite homographies (see Appendix B.2) from the first to the second and third images respectively. Note that, with the first projection in canonical form, the left camera centre expressed in homogeneous coordinates is given by $\tilde{\mathbf{C}} = [0, 0, 0, 1]^\mathsf{T}$. So, we have $\mathtt{P}'\tilde{\mathbf{C}} \simeq \mathbf{e}'$ and $\mathtt{P}''\tilde{\mathbf{C}} \simeq \mathbf{e}''$, that is, the fourth column of the second and third projections readily provide the coordinates of the epipoles in these views.

In light of result (4.4), the three world planes obtained by back-projecting the camera centres through their respective image lines have coordinates

$$\begin{aligned}
\pi_1 &= \mathtt{P}^\mathsf{T}\boldsymbol{l} &&= [\boldsymbol{l}^\mathsf{T}, 0]^\mathsf{T}, \\
\pi_2 &= \mathtt{P}'^\mathsf{T}\boldsymbol{l}' &&= [a_1^j l'_j, a_2^j l'_j, a_3^j l'_j, a_4^j l'_j]^\mathsf{T}, \\
\pi_3 &= \mathtt{P}''^\mathsf{T}\boldsymbol{l}'' &&= [b_1^k l''_k, b_1^k l''_k, b_3^k l''_k, b_4^k l''_k]^\mathsf{T}.
\end{aligned}$$

The above expressions make use of Einstein's summation convention to the effect that a repeated index which appears once as a superscript and once as a subscript implies summation over the range of the index. For instance, $a_1^j l'_j$ is short for $\sum_{j=1}^{3} a_1^j l'_j$. This convention will be used throughout this chapter.

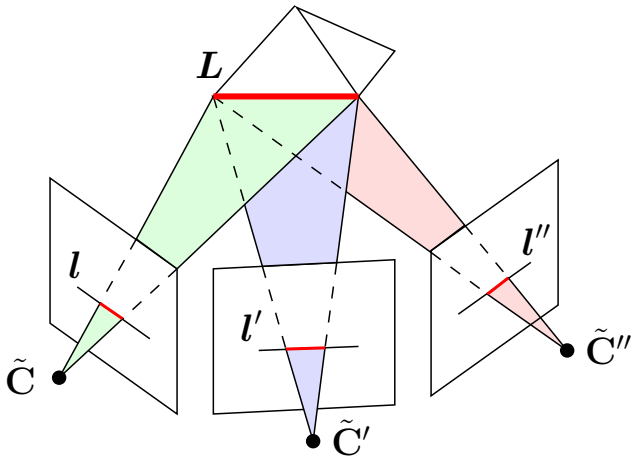


Figure 4.5. Lines l, l', and l'' are the images of a world line $\boldsymbol{L}$ into three distinct views.

In the classical case, three planes intersect in a 3-D point. The situation depicted in Figure 4.5 shows that the planes meet in space in a common line $\boldsymbol{L}$. This particular intersection constrains the planes coordinates in a special way as exposed next.

Let $\mathbf{M} = \lambda_1\mathbf{M}_1 + \lambda_2\mathbf{M}_2$ be a world point on line $\boldsymbol{L}$ where $\mathbf{M}_1, \mathbf{M}_2$ are two linearly independent points on $\boldsymbol{L}$ and λ_1, λ_2 two scalars. Such a point $\mathbf{M}$ lies on all three planes so it satisfies the relations $\mathbf{M}^\mathsf{T}\pi_1 = \mathbf{M}^\mathsf{T}\pi_2 = \mathbf{M}^\mathsf{T}\pi_3 = 0$. With the 4×3 matrix $\mathtt{Z} = [\pi_1, \pi_2, \pi_3]$, these three conditions can be written as $\mathbf{M}^\mathsf{T}\mathtt{Z} = \mathbf{0}^\mathsf{T}$, revealing that $\mathtt{Z}$ has a two dimensional null-space because $\mathbf{M}_1^\mathsf{T}\mathtt{Z} = \mathbf{0}^\mathsf{T}$ and $\mathbf{M}_2^\mathsf{T}\mathtt{Z} = \mathbf{0}^\mathsf{T}$. Therefore, matrix $\mathtt{Z}$ has rank 2 and so the coordinates of the three planes must be linearly dependent. We may write

$$\pi_1 = \alpha\pi_2 + \beta\pi_3,$$

for α, β scalars, which gives the coordinates of $\boldsymbol{l}$ as

$$l_i = \alpha(a_i^j l'_j) + \beta(b_i^k l''_k). \tag{4.15}$$

The last coordinate of π_1 being zero, we have $0 = \alpha(a_4^j l'_j) + \beta(b_4^k l''_k)$, which is satisfied when $\alpha \simeq (b_4^k l''_k)$ and $\beta \simeq -(a_4^j l'_j)$. Substituting these values in (4.15) and rearranging yields

$$l_i \simeq l'_j l''_k (a_i^j b_4^k - a_4^j b_i^k).$$

Defining the $3 \times 3 \times 3$ tensor

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k, \qquad i, j, k = 1, 2, 3, \tag{4.16}$$

now gives

$$l_i \simeq l'_j l''_k \mathcal{T}_i^{jk}. \tag{4.17}$$

Note that, in tensor notation, the order of the elements is not important, only the placement of indices so the previous relation can also be written as $l_i \simeq l''_k \mathcal{T}_i^{jk} l'_j$. The tensor introduced in (4.16) is known in computer vision as the *trinocular* or *trifocal tensor*. This equation provides an explicit parameterisation in terms of the three image projections. Given the range of the indices i,j,k, the trifocal tensor is defined by 27 scalar coefficients. A common visualisation of this tensor is as a $3 \times 3 \times 3$ cube of numbers, as drawn in Figure 4.6. The tensorial elements on the cube can be worked out by looking at the orientation of the coordinate system.

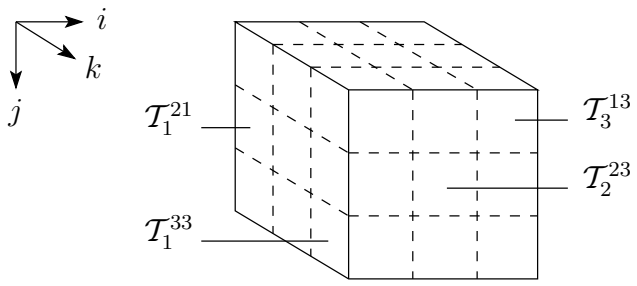


Figure 4.6. A representation of the trifocal tensor as a cube of numbers.

It will be convenient in the sequel to operate on the tensorial coefficients when they are assembled in a vector. Since each index in $\mathcal{T}_i^{jk}$ ranges from 1 to 3, we may list the elements using a polynomial in base 3. For instance, one possibility is to take $\mathcal{T}_i^{jk}$ for the $(3^2 i + 3j + k)$-th entry in a vector $\boldsymbol{\theta}$. However, this rule predicts the first element $\mathcal{T}_1^{11}$ as the 13-th entry in the vector so we must substract 12. In short, we adopt the convention

$$\mathcal{T}_i^{jk} = \theta_{9i+3j+k-12}. \tag{4.18}$$

We conclude this section by outlining two different, but intimately related, techniques to derive the trifocal tensor from three general projections, not necessarily assuming that $\mathtt{P} = [\mathtt{I}_{3\times 3} \mid \mathbf{0}]$. One method relies on the formalism of Grassmann-Cayley algebra [24, 27]. In essence, this method expresses the line relation (4.17) by the meet of principal planes originating from the three images. This procedure relies on three 3×3 critical matrices, the trifocal matrices, which *implicitly* define the trifocal tensor. Each entry in these matrices is obtained by calculating the meet of four planes: two principal planes from the first view and one principal plane from the second and third views. This particular choice of planes will be explained in Section 4.3.3. Using definition (4.2), this is carried out by computing the 4×4 determinant made from two rows of the first projection and one row from the second and third projections. A remarkable fact is that coefficients $\mathcal{T}_i^{jk}$ can be written directly in terms of these 4×4 determinants. This method describes the trifocal tensor *explicitly* [40]. To summarise the argumentation, one method derives the tensor indirectly, from geometric considerations, whereas the second does it directly, from algebraic manipulations.

4.2.2 Parameterisation from two projections

The first projection being in the form $\mathtt{P} = [\mathtt{I}_{3\times3} \mid \mathbf{0}]$ may be discarded to leave a reduced parameterisation which only considers the remaining projections $\mathtt{P}'$ and $\mathtt{P}''$. The complete 27-dimensional parameterisation (4.18) can be curtailed to a 24-dimensional one by vectorising the second and third projections as in

$$\boldsymbol{\beta} = \begin{bmatrix} \mathrm{vec}(\mathtt{P}') \\ \mathrm{vec}(\mathtt{P}'') \end{bmatrix}. \tag{4.19}$$

This parameterisation will be used in the implementation of most correction schemes proposed in Section 2.5. Only Kanatani's method does not exploit any parameterisation of the tensor manifold as it operates with the full, unconstrained parameter vector $\boldsymbol{\theta}$.

Now, notice that (4.16) provides a mapping from elements of the second and third projections to the trifocal tensor. Let this mapping be defined by a function $g : \mathbb{R}^{24} \mapsto \mathbb{R}^{27}$ such that $g(\boldsymbol{\beta}) = \mathcal{T}$, where $\boldsymbol{\beta}$ takes the form (4.19). This function is quadratic and its Jacobian is given by the 24×27 matrix

$$\frac{\partial g}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \mathtt{I}_{3\times3} \otimes \mathbf{e}''^{\mathsf{T}} & & \\ & \mathtt{I}_{3\times3} \otimes \mathbf{e}''^{\mathsf{T}} & \\ & & \mathtt{I}_{3\times3} \otimes \mathbf{e}''^{\mathsf{T}} \\ \mathtt{I}_{3\times3} \otimes -\mathbf{b}_1^{\mathsf{T}} & \mathtt{I}_{3\times3} \otimes -\mathbf{b}_2^{\mathsf{T}} & \mathtt{I}_{3\times3} \otimes -\mathbf{b}_3^{\mathsf{T}} \\ -\mathbf{e}'^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} & & \\ & -\mathbf{e}'^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} & \\ & & -\mathbf{e}'^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} \\ \mathbf{a}_1^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} & \mathbf{a}_2^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} & \mathbf{a}_3^{\mathsf{T}} \otimes \mathtt{I}_{3\times3} \end{bmatrix}, \tag{4.20}$$

where the vectors $\mathbf{a}_i, \mathbf{b}_i$ are the i-th columns of the projections $\mathtt{P}', \mathtt{P}''$, and $\mathbf{e}', \mathbf{e}''$ are the epipoles in images Ψ_2, Ψ_3 respectively. The position of zero elements has been left blank for clarity.

In some situations, the epipoles $\mathbf{e}'$ and $\mathbf{e}''$ may be known. This information yet allows another parameterisation of the tensor. This time, we may define a linear transformation $\tilde{g} : \mathbb{R}^{18} \mapsto \mathbb{R}^{27}$ such that $\tilde{g}(\tilde{\boldsymbol{\beta}}) = \mathcal{T}$, with $\tilde{\boldsymbol{\beta}}$ obtained by vectorising the inner 3×3 matrices of $\mathtt{P}'$ and $\mathtt{P}''$ as in

$$\tilde{\boldsymbol{\beta}} = \begin{bmatrix} \mathrm{vec}(\mathtt{A}) \\ \mathrm{vec}(\mathtt{B}) \end{bmatrix}. \tag{4.21}$$

This partial paramameterisation (of a subset of the trifocal tensor manifold) is 18-dimensional. In this case, transformation $\tilde{g}$ is given by a 27×18 matrix $\mathtt{E}$ such that $\mathcal{T} = \mathtt{E}(\mathbf{e}', \mathbf{e}'')\tilde{\boldsymbol{\beta}}$, and since

$\tilde{g}$ is linear,

$$\mathtt{E}(\mathbf{e}', \mathbf{e}'') = \left(\frac{\partial \tilde{g}}{\partial \tilde{\boldsymbol{\beta}}}\right)^{\mathsf{T}}.$$

One may use (4.20) and remove the rows corresponding to derivatives $\partial\mathcal{T}/\partial\mathbf{e}'$ and $\partial\mathcal{T}/\partial\mathbf{e}''$ to obtain

$$\mathtt{E}(\mathbf{e}', \mathbf{e}'') = \left[\mathtt{I}_{3\times3} \otimes \mathtt{I}_{3\times3} \otimes \mathbf{e}'',\ \mathtt{I}_{3\times3} \otimes -\mathbf{e}' \otimes \mathtt{I}_{3\times3}\right]. \tag{4.22}$$

Hartley has used this type of parameterisation to impose some linear constraints on the trifocal tensor. Details are deferred to Section 5.2.3.

4.3 Contractions of the trifocal tensor

Geometric relations between features in correspondence or features "transferred" from one view(s) to another are related to algebraic operations on the tensor. Since the trifocal tensor is a mixed tensor, its valence can be reduced by multiplying it with lines (covariant vectors) and/or points (contravariant vectors). This reduction of the tensor is formally known as a *contraction* of the tensor. The various combinations of features across three views produce several incidence relations, which can all be captured by a single trifocal tensor. We can distinguish several levels of contraction of the trifocal tensor depending on the number and type of features (line or point) employed. The next sections will present each type of contraction in turn and their associated image geometry.

4.3.1 Single contraction: the tensorial slices

A contraction of the trifocal tensor by one image token reduces its overall valence by one. This operation amounts to fixing one of the indices $i, j,$ or k of the tensor $\mathcal{T}_i^{jk}$. Fixing a particular index isolates three 3×3 matrices from the tensor and, depending on which index is selected, we can imagine the cube being cut in three principal directions: horizontal, vertical, and lateral. This procedure yields three different kinds of matrices called the *tensorial slices*. It will be shown that each type of slices corresponds to a specific image operation. Ultimately, these slices will play an important role in deriving sets of constraints to estimate the trifocal tensor from image measurements.

The correlation slices

Recall that $\{\mathbf{e}_i\}_{i=1}^3$ is a set of canonical basis vectors for $\mathbb{R}^3$. With respect to this basis, any image point $\mathbf{m}$ can be expressed in homogeneous coordinates by a set of contravariant scalar

coordinates $\{m^i\}_{i=1}^3$ such that $\mathbf{m} = m^i\mathbf{e}_i$. Similarly, any image line $\boldsymbol{l}$ can be defined in the contravariant basis $\{\boldsymbol{\epsilon}^i\}_{i=1}^3$ as $\boldsymbol{l} = l_i\boldsymbol{\epsilon}^i$ with l_i also representing scalar coordinates.

The contraction of the trifocal tensor $\mathcal{T}_i^{jk}$ with an image point $\mathbf{m}$ in the reference view eliminates (or fixes) the i-index in $\mathcal{T}_i^{jk}$ by performing the operation $m^i\mathcal{T}_i^{jk}$. When $\mathbf{m} = \mathbf{e}_i$, this contraction gives a special matrix

$$\boldsymbol{\mathcal{I}}_i = \mathcal{T}_{(i)}^{jk} = \mathbf{a}_{(i)}^j\mathbf{b}_4^k - \mathbf{a}_4^j\mathbf{b}_{(i)}^k,$$

where the notation (i) emphasises the fixation of the i-index. The quantity $\mathcal{T}_{(i)}^{jk}$ can be interpreted as a 3×3 matrix $\boldsymbol{\mathcal{I}}_i$ for which the entries are ordered by the j-index for the rows and the k-index for the columns. Cycling through different values of the i-index gives three matrices referred to as the *trifocal matrices* or *correlation slices* of the trifocal tensor. We will see in Section 4.4.1 that these matrices are correlations [77] because they embody mappings from lines to points, hence their name.

In vector notation,

$$\boldsymbol{\mathcal{I}}_i = (\mathtt{A}\mathbf{e}_i)\mathbf{b}_4^\mathsf{T} - \mathbf{a}_4(\mathtt{B}\mathbf{e}_i)^\mathsf{T} = \mathbf{a}_i\mathbf{b}_4^\mathsf{T} - \mathbf{a}_4\mathbf{b}_i^\mathsf{T}, \tag{4.23}$$

which, in terms of the trifocal coefficients, corresponds to

$$\boldsymbol{\mathcal{I}}_i = \begin{bmatrix} \mathcal{T}_i^{11} & \mathcal{T}_i^{12} & \mathcal{T}_i^{13} \\ \mathcal{T}_i^{21} & \mathcal{T}_i^{22} & \mathcal{T}_i^{23} \\ \mathcal{T}_i^{31} & \mathcal{T}_i^{32} & \mathcal{T}_i^{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{(9i-8)} & \boldsymbol{\theta}_{(9i-7)} & \boldsymbol{\theta}_{(9i-6)} \\ \boldsymbol{\theta}_{(9i-5)} & \boldsymbol{\theta}_{(9i-4)} & \boldsymbol{\theta}_{(9i-3)} \\ \boldsymbol{\theta}_{(9i-2)} & \boldsymbol{\theta}_{(9i-1)} & \boldsymbol{\theta}_{(9i)} \end{bmatrix}. \tag{4.24}$$

The definition of the projections matrices in (4.14) sets a particular orientation for the coordinate system of the trifocal tensor (refer to Figure 4.6). In this system, the trifocal matrices correspond to vertical slices of the trifocal cube. A representation is given in Figure 4.7. It turns out that the order of the coefficients $\mathcal{T}_i^{jk}$ on the slices matches the order of the coefficients in the matrices (4.24). For instance, element $\mathcal{T}_1^{21}$ is in position $(2, 1)$ in matrix $\boldsymbol{\mathcal{I}}_1$ and similarly on the first slice of Figure 4.7. The same remark will hold for the other two types of slices that will be examined in the next section.

It will become useful to specify the column and row vectors of the trifocal matrices as follows

$$\boldsymbol{\mathcal{I}}_1 = [\mathbf{a}_1', \mathbf{a}_2', \mathbf{a}_3'], \;\; \boldsymbol{\mathcal{I}}_2 = [\mathbf{b}_1', \mathbf{b}_2', \mathbf{b}_3'], \; \boldsymbol{\mathcal{I}}_3 = [\mathbf{c}_1', \mathbf{c}_2', \mathbf{c}_3'], \tag{4.25}$$

and

$$\boldsymbol{\mathcal{I}}_1^\mathsf{T} = [\mathbf{d}_1'', \mathbf{d}_2'', \mathbf{d}_3''], \boldsymbol{\mathcal{I}}_2^\mathsf{T} = [\mathbf{e}_1'', \mathbf{e}_2'', \mathbf{e}_3''], \; \boldsymbol{\mathcal{I}}_3^\mathsf{T} = [\mathbf{f}_1'', \mathbf{f}_2'', \mathbf{f}_3'']. \tag{4.26}$$

The trifocal tensor can be recovered from the correlations slices defined in (4.24) by setting

$$\boldsymbol{\theta} = \begin{pmatrix} \mathrm{vec}(\boldsymbol{\mathcal{I}}_1^\mathsf{T}) \\ \mathrm{vec}(\boldsymbol{\mathcal{I}}_2^\mathsf{T}) \\ \mathrm{vec}(\boldsymbol{\mathcal{I}}_3^\mathsf{T}) \end{pmatrix}. \tag{4.27}$$

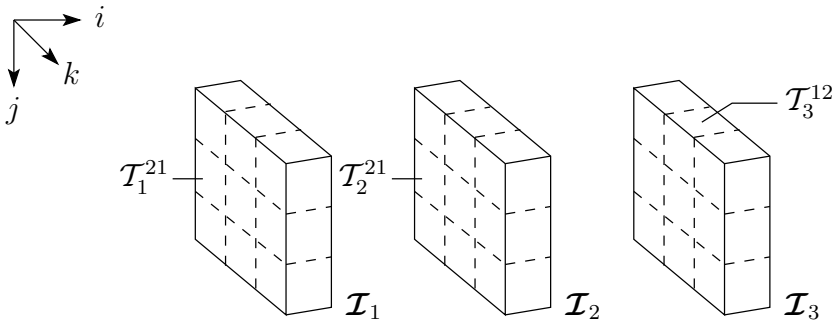


Figure 4.7. A representation of the three correlation slices $\boldsymbol{\mathcal{I}}_i$ and some of their elements.

This reassembling of the tensor from the correlation slices ensures a similar ordering of the trifocal coefficients as that proposed in (4.18).

In general, the contraction of the tensor by an arbitrary image point $\mathbf{m} = m^i\mathbf{e}_i$ in the reference image is a 3×3 matrix

$$\boldsymbol{\mathcal{I}}_{\mathbf{m}} = m^i\mathcal{T}_i^{jk} = m^1\boldsymbol{\mathcal{I}}_1 + m^2\boldsymbol{\mathcal{I}}_2 + m^3\boldsymbol{\mathcal{I}}_3.$$

As it can be seen, this matrix is a linear combination of the trifocal matrices. Using (4.23), we have $m^i\boldsymbol{\mathcal{I}}_i = (\mathtt{A}m^i\mathbf{e}_i)\mathbf{b}_4^\mathsf{T} - \mathbf{a}_4(\mathtt{B}m^i\mathbf{e}_i)^\mathsf{T}$, thus

$$\boldsymbol{\mathcal{I}}_{\mathbf{m}} = (\mathtt{A}\mathbf{m})\mathbf{b}_4^\mathsf{T} - \mathbf{a}_4(\mathtt{B}\mathbf{m})^\mathsf{T}. \tag{4.28}$$

Clearly, if $\mathbf{m} = \mathbf{e}_i$, then $\boldsymbol{\mathcal{I}}_{\mathbf{e}_i} = \boldsymbol{\mathcal{I}}_i$, as given initially in (4.23). The image geometry associated with matrix $\boldsymbol{\mathcal{I}}_{\mathbf{m}}$ will be described in Section 4.3.2.

The homography slices

The j-index (resp. k-index) of the tensor $\mathcal{T}_i^{jk}$ is contravariant which implies that the tensor must be contracted with a covariant vector, that is, a line in the second (resp. third) view.

To begin, consider a contraction of the trifocal tensor (4.16) with an image line $\boldsymbol{l}' = [l'_j]$ in the second view. This operation amounts to fixing the j-index in $\mathcal{T}_i^{jk}$ by carrying out $l'_j\mathcal{T}_i^{jk}$. When the image line $\boldsymbol{l}'$ is a canonical basis vector $\mathbf{e}_j$, we obtain a 3×3 matrix

$$\boldsymbol{\mathcal{J}}_j = \mathcal{T}_i^{(j)k} = \mathbf{a}_i^{(j)}\mathbf{b}_4^k - \mathbf{a}_4^{(j)}\mathbf{b}_i^k. \tag{4.29}$$

Here, indices k and i refer to the entry (k, i) in the matrix. It will be shown in Section 4.4.2 that matrix $\boldsymbol{\mathcal{J}}_j$ represents a homography mapping from points in image Ψ_1 to points in image Ψ_3 via the j-th principal plane in image Ψ_2. Accordingly, these matrices are called *homography slices* or *intrinsic planar morphisms* of the trifocal tensor. They are associated with the horizontal

slices of the cube as depicted in Figure 4.8. Generally, these matrices are regular with rank three but may be singular for particular camera configurations [73]. Equation (4.29) can be formulated in terms of basis vectors as

$$\boldsymbol{\mathcal{J}}_j = \mathbf{b}_4(\mathbf{e}_j^\mathsf{T}\mathtt{A}) - (\mathbf{a}_4^\mathsf{T}\mathbf{e}_j)\mathtt{B}, \tag{4.30}$$

or explicitly, in terms of the tensorial coefficients, as

$$\boldsymbol{\mathcal{J}}_j = \begin{bmatrix} \mathcal{T}_1^{j1} & \mathcal{T}_2^{j1} & \mathcal{T}_3^{j1} \\ \mathcal{T}_1^{j2} & \mathcal{T}_2^{j2} & \mathcal{T}_3^{j2} \\ \mathcal{T}_1^{j3} & \mathcal{T}_2^{j3} & \mathcal{T}_3^{j3} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{(3j-2)} & \boldsymbol{\theta}_{(3j+7)} & \boldsymbol{\theta}_{(3j+16)} \\ \boldsymbol{\theta}_{(3j-1)} & \boldsymbol{\theta}_{(3j+8)} & \boldsymbol{\theta}_{(3j+17)} \\ \boldsymbol{\theta}_{(3j)} & \boldsymbol{\theta}_{(3j+9)} & \boldsymbol{\theta}_{(3j+18)} \end{bmatrix}. \tag{4.31}$$

These matrices have elements in common with the trifocal matrices $\boldsymbol{\mathcal{I}}_i$. One may check that

$$\boldsymbol{\mathcal{J}}_1 = [\mathbf{d}_1'', \mathbf{e}_1'', \mathbf{f}_1''],\ \boldsymbol{\mathcal{J}}_2 = [\mathbf{d}_2'', \mathbf{e}_2'', \mathbf{f}_2''],\ \boldsymbol{\mathcal{J}}_3 = [\mathbf{d}_3'', \mathbf{e}_3'', \mathbf{f}_3'']. \tag{4.32}$$

For convenience, we observe the following convention for the row vectors of the $\boldsymbol{\mathcal{J}}_j$ matrices:

$$\boldsymbol{\mathcal{J}}_1^\mathsf{T} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3],\ \boldsymbol{\mathcal{J}}_2^\mathsf{T} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3],\ \boldsymbol{\mathcal{J}}_3^\mathsf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]. \tag{4.33}$$

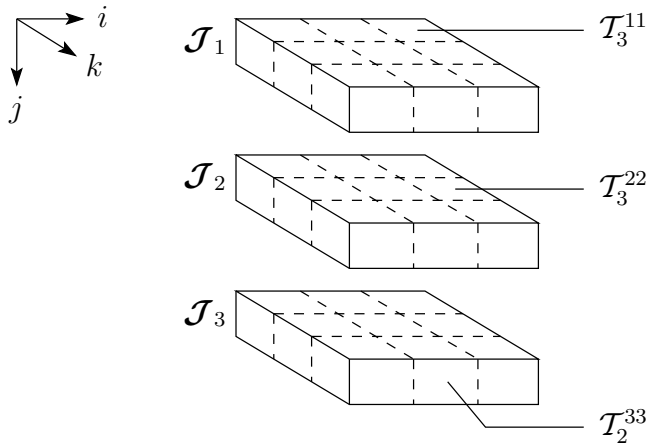


Figure 4.8. A representation of the three homography slices $\boldsymbol{\mathcal{J}}_j$ and some of their elements.

Analogously to the case of matrix $\boldsymbol{\mathcal{I}}_\mathbf{m}$ in (4.28), the contraction of the trifocal tensor by an arbitrary image line $\boldsymbol{l}' = l_j'\boldsymbol{\epsilon}^j$ in the second view is a 3×3 matrix

$$\boldsymbol{\mathcal{J}}_{\boldsymbol{l}'} = l_j'\mathcal{T}_i^{jk} = l_1'\boldsymbol{\mathcal{J}}_1 + l_2'\boldsymbol{\mathcal{J}}_2 + l_3'\boldsymbol{\mathcal{J}}_3, \tag{4.34}$$

which is a linear combination of the homography slices $\boldsymbol{\mathcal{J}}_j$. Using (4.30), we have $l_j'\boldsymbol{\mathcal{J}}_j = \mathbf{b}_4(l_j'\mathbf{e}_j^\mathsf{T}\mathtt{A}) - (\mathbf{a}_4^\mathsf{T}l_j'\mathbf{e}_j)\mathtt{B}$, thus

$$\boldsymbol{\mathcal{J}}_{\boldsymbol{l}'} = \mathbf{b}_4(\boldsymbol{l}'^\mathsf{T}\mathtt{A}) - (\mathbf{a}_4^\mathsf{T}\boldsymbol{l}')\mathtt{B}.$$

An analysis of the image geometry associated with matrix $\mathcal{J}_{l'}$ is deferred to Section 4.4.2.

Finally, the contraction of the trifocal tensor with an image line in the third view fixes the k-index in $\mathcal{T}_i^{jk}$. Choosing lines with canonical coordinates yields three 3×3 matrices of the form

$$\mathcal{K}_k = \mathcal{T}_i^{j(k)} = \mathbf{a}_i^j \mathbf{b}_4^{(k)} - \mathbf{a}_4^j \mathbf{b}_i^{(k)},$$

where indices i and j refer to the entry (j, i) in each matrix $\mathcal{K}_k$. It will be shown in Section 4.4.2 that matrix $\mathcal{K}_k$ represents a homography mapping from points in image Ψ_1 to points in image Ψ_2 via the k-th principal plane in image Ψ_3. These matrices yet constitute another type of homography slices of the trifocal tensor and relate to the lateral slices of the cube, as shown in Figure 4.9. In terms of basis vectors,

$$\mathcal{K}_k = (\mathbf{b}_4^\mathsf{T} \mathbf{e}_k)\mathtt{A} - \mathbf{a}_4(\mathbf{e}_k^\mathsf{T}\mathtt{B}) \tag{4.35}$$

or, in terms of the tensorial coefficients,

$$\mathcal{K}_k = \begin{bmatrix} \mathcal{T}_1^{1k} & \mathcal{T}_2^{1k} & \mathcal{T}_3^{1k} \\ \mathcal{T}_1^{2k} & \mathcal{T}_2^{2k} & \mathcal{T}_3^{2k} \\ \mathcal{T}_1^{3k} & \mathcal{T}_2^{3k} & \mathcal{T}_3^{3k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{(k)} & \boldsymbol{\theta}_{(k+9)} & \boldsymbol{\theta}_{(k+18)} \\ \boldsymbol{\theta}_{(k+3)} & \boldsymbol{\theta}_{(k+12)} & \boldsymbol{\theta}_{(k+21)} \\ \boldsymbol{\theta}_{(k+6)} & \boldsymbol{\theta}_{(k+15)} & \boldsymbol{\theta}_{(k+24)} \end{bmatrix}. \tag{4.36}$$

These matrices have elements in common with both $\mathcal{I}_i$ and $\mathcal{J}_j$ matrices. One may check that

$$\mathcal{K}_1 = [\mathbf{a}'_1, \mathbf{b}'_1, \mathbf{c}'_1],\ \mathcal{K}_2 = [\mathbf{a}'_2, \mathbf{b}'_2, \mathbf{c}'_2],\ \mathcal{K}_3 = [\mathbf{a}'_3, \mathbf{b}'_3, \mathbf{c}'_3], \tag{4.37}$$

and

$$\mathcal{K}_1^\mathsf{T} = [\mathbf{r}_1, \mathbf{s}_1, \mathbf{t}_1],\ \mathcal{K}_2^\mathsf{T} = [\mathbf{r}_2, \mathbf{s}_2, \mathbf{t}_2],\ \mathcal{K}_3^\mathsf{T} = [\mathbf{r}_3, \mathbf{s}_3, \mathbf{t}_3]. \tag{4.38}$$

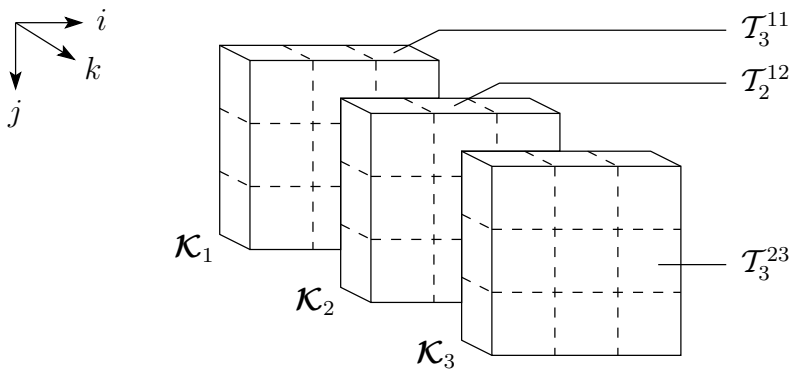


Figure 4.9. A representation of the three homography slices $\mathcal{K}_k$ and some of their elements.

The contraction of the tensor by an arbitrary image line $\boldsymbol{l}'' = l''_k\boldsymbol{\epsilon}^k$ in the third view is a 3×3 matrix

$$\mathcal{K}_{\boldsymbol{l}''} = l''_k\mathcal{T}_i^{jk} = l''_1\mathcal{K}_1 + l''_2\mathcal{K}_2 + l''_3\mathcal{K}_3. \tag{4.39}$$

Using definition (4.35) for $\mathcal{K}_k$ and simplifying gives

$$\mathcal{K}_{\boldsymbol{l}''} = (\mathbf{b}_4^\mathsf{T}\boldsymbol{l}'')\mathtt{A} - \mathbf{a}_4(\boldsymbol{l}''^\mathsf{T}\mathtt{B}).$$

The geometric interpretation of these matrices is also postponed until Section 4.4.2.

4.3.2 Two contractions: transfer between views

We now consider the effect of using two image tokens to simultaneously contract the trifocal tensor. We will see that given a pair of corresponding features in two views, the tensor will give their matching counterpart in the third view [30]. This property of the tensor is widely used in *recognition* applications where it is necessary to predict the location of a feature in one image from its positions in two other images. This problem is sometimes described as the *feature transfer* problem.

Transferring two image lines

The most natural use of the trifocal tensor is to transfer image lines. The situation was first presented in Section 4.2.1 when we derived a parameterisation of the tensor in terms of three camera projections. The primitive that arises in the first view by transferring the pair of lines $(\boldsymbol{l}', \boldsymbol{l}'')$ is a line $\boldsymbol{l}$ given by the equations

$$l_i \simeq l'_j l''_k \mathcal{T}_i^{jk}. \tag{4.40}$$

It turns out that these relations can be written as $\boldsymbol{l} \simeq \mathcal{T}(\boldsymbol{l}', \boldsymbol{l}'')$, where the application $\mathcal{T} : \mathbb{P}^{2*} \times \mathbb{P}^{2*} \rightarrow \mathbb{P}^{2*}$ such that

$$\mathcal{T}(\boldsymbol{l}', \boldsymbol{l}'') = [\boldsymbol{l}'^\mathsf{T}\boldsymbol{\mathcal{I}}_1\boldsymbol{l}'', \boldsymbol{l}'^\mathsf{T}\boldsymbol{\mathcal{I}}_2\boldsymbol{l}'', \boldsymbol{l}'^\mathsf{T}\boldsymbol{\mathcal{I}}_3\boldsymbol{l}'']^\mathsf{T} \tag{4.41}$$

is the trifocal tensor expressed as a bilinear map acting on two lines to produce a third line. This expression will be proved shortly as a consequence of Proposition 5.

A particular case of this transfer occurs when the lines $\boldsymbol{l}'$ and $\boldsymbol{l}''$ are in correspondence with a point $\mathbf{m}$ on line $\boldsymbol{l}$ in the reference image, that is, for a point-line-line incidence in three views. Predicting the point $\mathbf{m}$ cannot be achieved in one tensor contraction as per (4.40). Further information is needed, like its corresponding point $\mathbf{m}'$ on $\boldsymbol{l}'$ in view 2.

Transferring an image point and an image line

Another possible situation is when the tensor is contracted by an image point and an image line in two separate views. The geometric relationships take place as follows.

Suppose that a scene point $\mathbf{M}$ projects into three views in the points $\mathbf{m}$, $\mathbf{m}'$ and $\mathbf{m}''$, and that $\boldsymbol{l}'$ is a line through $\mathbf{m}'$. Although $\boldsymbol{l}'$ may not be an epipolar line, its back-projected plane still contains the scene point $\mathbf{M}$. Without loss of generality, we may assume that the first camera projection is in canonical form, so the ray of sight observing $\mathbf{m}$ gives directly the position of $\mathbf{M}$ in the world frame. Consequently, point $\mathbf{M}$ can be parameterised by the equation

$$\mathbf{M} = \tilde{\mathbf{C}} + \rho \mathtt{P}^{+}\mathbf{m}, \tag{4.42}$$

where ρ is a non-zero scalar indicating the depth of $\mathbf{M}$ in space and $\mathtt{P}^{+} = [\mathtt{I} \mid \mathbf{0}]^{\mathsf{T}}$ is the pseudo-inverse matrix of the canonical projection in the first view. Features $\mathbf{m}'$ and $\boldsymbol{l}'$ in the second view satisfy the relation

$$\boldsymbol{l}'^{\mathsf{T}}\mathbf{m}' = \boldsymbol{l}'^{\mathsf{T}}\mathtt{P}'\mathbf{M} = \boldsymbol{l}'^{\mathsf{T}}(\mathbf{e}' + \rho \mathtt{A}\mathbf{m}) = 0. \tag{4.43}$$

This formula fixes the value of parameter ρ in (4.42), hence the location of the scene point. Now, projecting $\mathbf{M}$ in the third view gives

$$\mathbf{m}'' \simeq \mathtt{P}''\mathbf{M} \simeq \mathbf{e}'' + \rho \mathtt{B}\mathbf{m},$$

and substituting the value of ρ from (4.43) leads to

$$\mathbf{m}'' \simeq \mathbf{e}''(\boldsymbol{l}'^{\mathsf{T}}\mathtt{A}\mathbf{m}) - (\boldsymbol{l}'^{\mathsf{T}}\mathbf{e}')\mathtt{B}\mathbf{m}.$$

Given that $\boldsymbol{l}'^{\mathsf{T}}\mathtt{A}\mathbf{m} = (\mathtt{A}\mathbf{m})^{\mathsf{T}}\boldsymbol{l}'$ and $\boldsymbol{l}'^{\mathsf{T}}\mathbf{e}' = \mathbf{e}'^{\mathsf{T}}\boldsymbol{l}'$, we find that

$$\mathbf{m}'' \simeq \left[\mathbf{e}''(\mathtt{A}\mathbf{m})^{\mathsf{T}} - (\mathtt{B}\mathbf{m})\mathbf{e}'^{\mathsf{T}}\right]\boldsymbol{l}'.$$

The expression between the brackets is the transpose of the 3×3 matrix $\boldsymbol{\mathcal{I}}_{\mathbf{m}}$ in (4.28), hence we can write

$$\mathbf{m}'' \simeq \boldsymbol{\mathcal{I}}_{\mathbf{m}}^{\mathsf{T}}\boldsymbol{l}' \simeq \left[m^i \boldsymbol{\mathcal{I}}_i^{\mathsf{T}}\right]\boldsymbol{l}'.$$

This expression reveals that matrix $\boldsymbol{\mathcal{I}}_{\mathbf{m}}^{\mathsf{T}}$ acts as a correlation mapping from the lines in image Ψ_2 to the points in image Ψ_3. Rearranging this equation gives

$$\mathbf{m}'' \simeq \left[\boldsymbol{\mathcal{I}}_1^{\mathsf{T}}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_2^{\mathsf{T}}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_3^{\mathsf{T}}\boldsymbol{l}'\right]\mathbf{m}.$$

In this form, we see that point $\mathbf{m}$ in the first view is transferred to its correspondent $\mathbf{m}''$ in the third view, and so the 3×3 matrix $[\boldsymbol{\mathcal{I}}_1^{\mathsf{T}}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_2^{\mathsf{T}}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_3^{\mathsf{T}}\boldsymbol{l}']$ is a homography. This is summarised in the following proposition.

Proposition 4 *The 3×3 homography matrix h_i^k, induced by the plane back-projected from a line $\boldsymbol{l}'$ going through $\mathbf{m}'$ in the second image, describes a mapping from $\mathbf{m}$ in the reference image to $\mathbf{m}''$ in the third image such that*

$$m''^k \simeq h_i^k m^i, \quad \textit{where } h_i^k = [\boldsymbol{\mathcal{I}}_1^\mathsf{T}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_2^\mathsf{T}\boldsymbol{l}', \boldsymbol{\mathcal{I}}_3^\mathsf{T}\boldsymbol{l}']. \tag{4.44}$$

Now, rearranging (4.40) as $l_i \simeq l_k''(l_j'\mathcal{T}_i^{jk})$ shows that the quantity $l_j'\mathcal{T}_i^{jk}$ acts as a mapping between lines from the third view to the first. This together with relation (4.44) proves that the result of the contraction $l_j'\mathcal{T}_i^{jk}$ is the homography h_i^k, that is, $l_j'\mathcal{T}_i^{jk} \simeq h_i^k$. Substituting in (4.44) gives

$$m''^k \simeq m^i l_j' \mathcal{T}_i^{jk}. \tag{4.45}$$

The geometric situation is depicted in Figure 4.10. The mapping from $\mathbf{m}$ to $\mathbf{m}''$ in (4.44) is a point transfer via a plane as described in Section 4.1.2; here the plane is readily identified as the back-projection of a line $\boldsymbol{l}'$ in the second view. When the line $\boldsymbol{l}'$ does not go through the point $\mathbf{m}'$, h_i^k is still a homography, however, it maps the point $\mathbf{m}$ to a point $\tilde{\mathbf{m}}''$ on the epipolar line containing $\mathbf{m}''$ [65, 79–81].

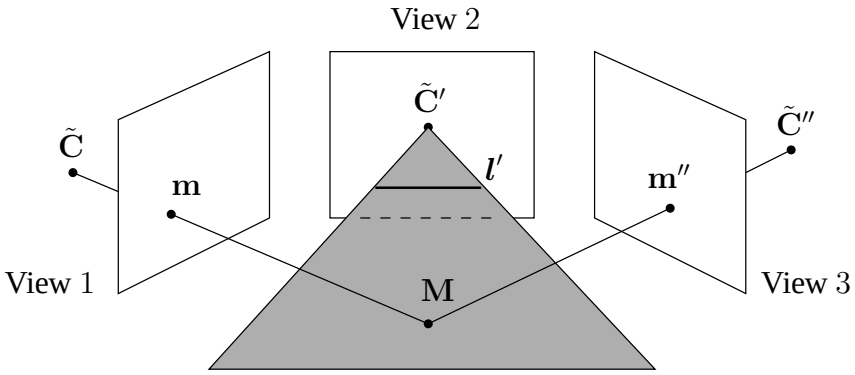


Figure 4.10. Three-view point transfer via a plane in space [37].

Likewise, there exists a homography which maps $\mathbf{m}$ to $\mathbf{m}'$ via the projection plane of a line $\boldsymbol{l}''$ going through $\mathbf{m}''$ in the third view. Projecting the point $\mathbf{M}$ of (4.42) into the second view and substituting the value of ρ from (4.43) gives

$$\mathbf{m}' \simeq \mathbf{e}' + \rho\mathtt{A}\mathbf{m} \simeq (\boldsymbol{l}''^\mathsf{T}\mathbf{e}'')\mathtt{A}\mathbf{m} - \mathbf{e}'(\boldsymbol{l}''^\mathsf{T}\mathtt{B}\mathbf{m}).$$

Since $\boldsymbol{l}''^\mathsf{T}\mathtt{B}\mathbf{m} = (\mathtt{B}\mathbf{m})^\mathsf{T}\boldsymbol{l}''$ and $\boldsymbol{l}''^\mathsf{T}\mathbf{e}'' = \mathbf{e}''^\mathsf{T}\boldsymbol{l}''$, we obtain

$$\mathbf{m}' \simeq \left[(\mathtt{A}\mathbf{m})\mathbf{e}''^\mathsf{T} - \mathbf{e}'(\mathtt{B}\mathbf{m})^\mathsf{T}\right]\boldsymbol{l}'',$$

where the expression between the brackets is exactly the 3×3 matrix in (4.28). Thus, we have

$$\mathbf{m}' \simeq \boldsymbol{\mathcal{I}}_\mathbf{m}\boldsymbol{l}'' \simeq \left[m^i\boldsymbol{\mathcal{I}}_i\right]\boldsymbol{l}''. \tag{4.46}$$

This result shows that matrix $\boldsymbol{\mathcal{T}}_{\mathbf{m}}$ acts as a correlation mapping from the lines in image Ψ_3 to the points in image Ψ_2. After rearranging,

$$\mathbf{m}' \simeq [\boldsymbol{\mathcal{T}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_3\boldsymbol{l}'']\,\mathbf{m}, \tag{4.47}$$

which permits to state the next proposition.

Proposition 5 *The 3×3 homography matrix h_i^j, induced by the plane back-projected from a line $\boldsymbol{l}''$ going through $\mathbf{m}''$ in the third image, provides a mapping from $\mathbf{m}$ in the reference image to $\mathbf{m}'$ in the second image such that*

$$m'^j \simeq h_i^j m^i, \quad \textit{where } h_i^j = [\boldsymbol{\mathcal{T}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_3\boldsymbol{l}'']. \tag{4.48}$$

Here, h_i^j corresponds to the contraction $l_k''\mathcal{T}_i^{jk}$, hence

$$m'^j \simeq m^i l_k'' \mathcal{T}_i^{jk}. \tag{4.49}$$

The proof of (4.41) follows directly from the above results and is exposed next.

If $\boldsymbol{l}'$ is a line through the point $\mathbf{m}'$ in the second view, we have that $\boldsymbol{l}'^{\mathsf{T}}\mathbf{m} = 0$. Substituting (4.47) for $\mathbf{m}'$ gives $\boldsymbol{l}'^{\mathsf{T}}[\boldsymbol{\mathcal{T}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_3\boldsymbol{l}'']\mathbf{m} = 0$. This relation suggests that $\boldsymbol{l}'^{\mathsf{T}}[\boldsymbol{\mathcal{T}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{T}}_3\boldsymbol{l}'']$ can be viewed as the coordinates of a line $\boldsymbol{l}$ going though $\mathbf{m}$ in the first view. Now, the vector $\boldsymbol{l}'^{\mathsf{T}}$ can be inserted in each of the components within the brackets and taking the transpose of the resulting expression yields (4.41), the line $\boldsymbol{l}$ in the first view expressed as a column vector.

In practice, the transfer operations (4.44) and (4.48) are the most useful because they allow to predict the location of corresponding points in images further along in the sequence. It should be noted that, in contrast with using a fundamental matrix, the location of the 3-D point $\mathbf{M}$ needs not be computed to find a corresponding image point. This is a significant advantage with the trifocal tensor.

For completeness, we show next how a point or a line can be obtained in the first view given a point and a line in the second and third views. These situations correspond to yet other possible geometric incidence relations between features in three views and, as before, are associated with specific algebraic contractions of the trifocal tensor.

Suppose that we have a pair of features $(\mathbf{m}', \boldsymbol{l}'')$, and that $\mathbf{m}' = [u', v', 1]^{\mathsf{T}}$. The point $\mathbf{m}'$ can be obtained in infinitely many ways by taking the cross product of any two distinct lines from the pencil of lines passing through $\mathbf{m}'$. Now, any line $\boldsymbol{l}'$ in the pencil can be expressed as a linear combination of two arbitrary, but fixed lines in that pencil, see Figure 4.11. With rectangular images, a natural choice is the horizontal line $\boldsymbol{l}'^h$ and the vertical line $\boldsymbol{l}'^v$ passing through $\mathbf{m}'$:

$$\boldsymbol{l}'^h \simeq [0, -1, v']^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{l}'^v \simeq [1, 0, -u']^{\mathsf{T}}.$$

It turns out that the coordinates of these lines may be expressed in terms of $\mathbf{m}'$ as

$$l'^h_j \simeq m'^q \epsilon_{qj1} \quad \text{and} \quad l'^v_j \simeq m'^q \epsilon_{qj2}, \tag{4.50}$$

where ϵ_{qjs}, $s = 1, 2$, are the coordinates of a covariant kronecker tensor $\boldsymbol{\epsilon}$ [53, 56].

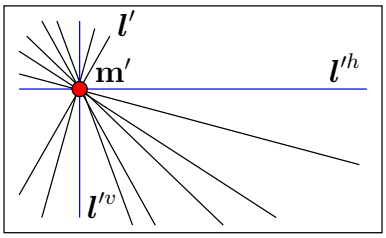


Figure 4.11. Any line l' in the pencil of lines through the image point $\mathbf{m}'$ can be expressed a linear combination of the lines $\boldsymbol{l}'^h$ and $\boldsymbol{l}'^v$.

A line $\boldsymbol{l}$ in the first view is now easily obtained using (4.40) with the lines $\boldsymbol{l}''$ and either one of $\boldsymbol{l}'^h$ or $\boldsymbol{l}'^v$. For instance, choosing $\boldsymbol{l}'^h$ gives $\boldsymbol{l}_i \simeq (m'^q \epsilon_{qj1}) l''_k \mathcal{T}_i^{jk}$, or simply $\boldsymbol{l} \simeq \mathcal{T}(\boldsymbol{l}'^h, \boldsymbol{l}'')$. Note that the computed line $\boldsymbol{l}$ goes through the point $\mathbf{m}$ corresponding to $\mathbf{m}'$. If $\mathbf{m}$ needs to be found explicitly, the tranfer cannot be realised in one operation and both lines $\boldsymbol{l}'^h$ and $\boldsymbol{l}'^v$ are required. Point $\mathbf{m}$ in the reference image can be computed as the intersection of the transferred lines $\boldsymbol{l}^1 \simeq \mathcal{T}(\boldsymbol{l}'^h, \boldsymbol{l}'')$ and $\boldsymbol{l}^2 \simeq \mathcal{T}(\boldsymbol{l}'^v, \boldsymbol{l}'')$ given by

$$\mathbf{m} \simeq \boldsymbol{l}^1 \times \boldsymbol{l}^2. \tag{4.51}$$

Figure 4.12 illustrates the situation. For a pair of features $(\boldsymbol{l}', \mathbf{m}'')$, one may follow a similar approach to find two lines through $\mathbf{m}''$ and transfer them with $\boldsymbol{l}'$ to a line or a point in the first view.

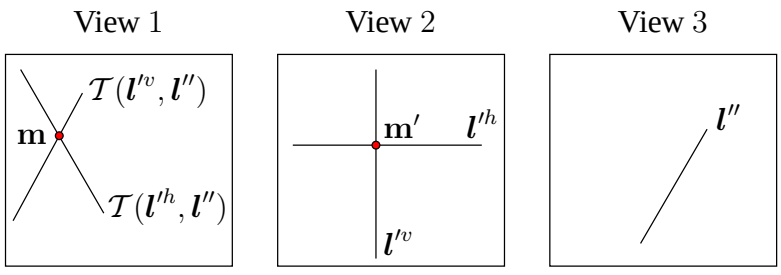


Figure 4.12. Trifocal tensor $\mathcal{T}$ transferring point $\mathbf{m}'$ in view 2 and line l'' in view 3 to point $\mathbf{m}$ in view 1.

Transferring two image points

This section will conclude the study of trifocal tensor contractions from two image tokens. Here, we assume that the locations of two matching points in distinct views are known and we aim to find their point correspondent in the third view. There are three possibilities depending on which two data points are transferred to another view.

First, suppose that we have identified the pair of matches $(\mathbf{m}, \mathbf{m}')$ and wish to obtain $\mathbf{m}''$ in the third image. The trifocal tensor cannot be contracted directly with $\mathbf{m}'$ so one must find a line through this point. Such a line may be taken as $\boldsymbol{l}'^h$ in (4.50) which gives

$$m''^k \simeq m^i (m'^q \epsilon_{qj1}) \mathcal{T}_i^{jk}. \tag{4.52}$$

Similarly, if we have a pair of matches $(\mathbf{m}, \mathbf{m}'')$, their corresponding point $\mathbf{m}'$ in the second view can be computed as

$$m'^j \simeq m^i (m''^q \epsilon_{qj1}) \mathcal{T}_i^{jk}. \tag{4.53}$$

On the other hand, if we have a pair $(\mathbf{m}', \mathbf{m}'')$, we must compute $\mathbf{m}$ as the intersection of two transferred lines corresponding to $\mathbf{m}'$ and $\mathbf{m}''$ respectively. There are infinitely many ways to choose the lines going through these points. A trivial choice is to take $\boldsymbol{l}^1 \simeq \mathcal{T}(\boldsymbol{l}'^h, \boldsymbol{l}''^h)$ with $l'^h_j \simeq m'^q \epsilon_{qj1}$ and $l''^h_k \simeq m''^q \epsilon_{qj1}$, and $\boldsymbol{l}^2 \simeq \mathcal{T}(\boldsymbol{l}'^h, \boldsymbol{l}''^v)$ with $l''^v_k = m''^q \epsilon_{qj2}$. Point $\mathbf{m}$ can then be computed as in (4.51). Figure 4.13 illustrates this transfer.

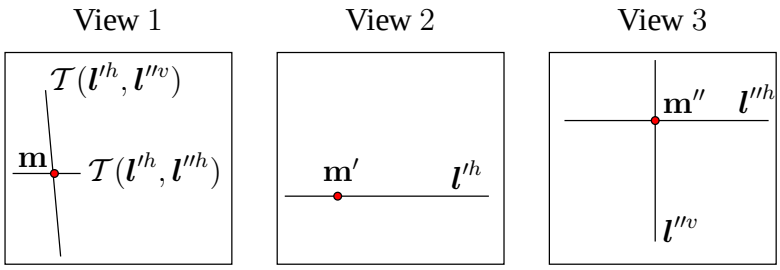


Figure 4.13. Trifocal tensor $\mathcal{T}$ transferring points $\mathbf{m}'$ in view 2 and $\mathbf{m}''$ in view 3 to point $\mathbf{m}$ in view 1.

The above considerations show that image point relations form a subset of image line relations since transferring points can only be done by resorting to lines first. The line transfer technique is used when the three camera centres are aligned, or close to be aligned, thus allowing point $\mathbf{m}$ to be determined. This is not possible from intermediate fundamental matrices between views. More details on point and line transfers can be found in [24, 27, 41]. As in the fundamental matrix case, there exist particular camera configurations and scene structures for which the trifocal tensor is ill-defined. Often these situations appear in the form of critical sets of lines and surfaces in space which give rise to ambiguous or even undefined reconstructions [9, 10, 43, 63, 83].

4.3.3 Three contractions: the trilinearities

Since the trifocal tensor is a valence-3 tensor, it can be reduced to a scalar by using three image tokens. When the image features are in correspondence across the views, the systems of equations describing the various incidence relations can be deduced from the transfer equations elaborated in the previous section. These systems exhibit linear relationships between the image tokens and the trifocal tensor and so are called *trilinearities*. Given enough image measurements, it is possible to estimate the trifocal tensor based on these systems. This topic will be covered in chapter 5.

In this section, we focus on deriving the trilinear systems corresponding to the various possible cases of incidence relations between image features in three views. It will be assumed that lines $\boldsymbol{l}, \boldsymbol{l}'$, and $\boldsymbol{l}''$ are in correspondence across the images and that points $\mathbf{m}, \mathbf{m}'$, and $\mathbf{m}''$ are the respective matches on these lines.

Point-line-line correspondence

The trifocal tensor has the form $\mathcal{T}_i^{jk}$ which suggests that it can be fully contracted by using a point in the first view and a line in the second and third views. Geometrically, the two image lines back-project to planes which intersect in a world line and the optical ray back-projected from the image point intersects this line in a 3-D point. This point-line-line incidence represents the *fundamental trifocal constraint* underpinning all relations between triplets of image points and/or lines. It is the equivalent for the trifocal tensor over three views of what the epipolar constraint is for the fundamental matrix over two views. It will be shown in subsequent sections that the algebraic constraints describing other trinocular incidence relations are variants of this particular one.

In this situation, one may noted that the optical ray in the first view may be obtained as the intersection of two principal planes in this view. Algebraically, this corresponds to selecting two rows from the first projection matrix. Each plane from the back-projected lines in the second and third views contributes one row from the respective projections in these images. This argument justifies the specific choice of four planes used to describe a general trifocal tensor when $\mathtt{P} \neq [\mathtt{I}_{3\times 3} \mid \mathbf{0}]$ as discussed at the end of Section 4.2.1.

To establish the trilinear constraints for the point-line-line relation, recall that, if $\boldsymbol{l}'$ and $\boldsymbol{l}''$ are two corresponding lines in the second and third views respectively, then their matching line $\boldsymbol{l}$ in the reference view is given by (4.40). Any point $\mathbf{m}$ on $\boldsymbol{l}$ satisfies the relation $m^i l_i = 0$ and substituting the definition of l_i from (4.40) yields the trilinear relation

$$m^i l'_j l''_k \mathcal{T}_i^{jk} = 0. \tag{4.54}$$

This formula can also be derived using (4.45) and the fact that $m''^k l''_k = 0$, or using (4.49) with $m'^j l'_j = 0$. Moreover, notice that (4.46) gives the matrix representation of (4.49). Therefore, using the relation $\boldsymbol{l}'^{\mathsf{T}}\mathbf{m}' = 0$, (4.54) can be written in matrix form as

$$\boldsymbol{l}'^{\mathsf{T}} \left[m^i \boldsymbol{\mathcal{I}}_i\right] \boldsymbol{l}'' = 0. \tag{4.55}$$

Point-point-line correspondence

Suppose that the coordinates of points $\mathbf{m}, \mathbf{m}'$, and line $\boldsymbol{l}''$ are known. In order to identify the algebraic constraints which bind these entities, it is necessary to convert the information about point $\mathbf{m}'$ into lines passing through it. A particular line $\boldsymbol{l}'$ can be expressed in terms of $\mathbf{m}'$ as $l'_j \simeq m'^q \epsilon_{qjs}$ for each fixed value of $s = 1, 2, 3$. Substituting for l'_j in (4.54) gives three trilinearities

$$m^i (m'^q \epsilon_{qjs}) l''_k \mathcal{T}_i^{jk} = 0_s. \tag{4.56}$$

When $s = 1$ (resp. $s = 2$), the above equation can be obtained from the transfer equation (4.52) using line $\boldsymbol{l}'^h$ (resp. $\boldsymbol{l}'^v$) and the fact that $m''^k l''_k = 0$. Since the lines $\boldsymbol{l}'^h$ and $\boldsymbol{l}'^v$ are sufficient to describe the pencil of lines through $\mathbf{m}'$, the quantity $m'^q \epsilon_{qj3}$, when $s = 3$, defines a line containing $\mathbf{m}'$ which can be obtained from a linear combination of $\boldsymbol{l}'^h$ and $\boldsymbol{l}'^v$ as depicted in Figure 4.12. Therefore, the three equations in (4.56) must be linearly dependent and only two are independent.

Point-line-point correspondence

The incidence between image features is similar to the previous one except that entities in the second and third views are interchanged. In this case, inserting $l''_k \simeq m''^q \epsilon_{qks}$ in (4.54) yields the relations

$$m^i l'_j (m''^q \epsilon_{qks}) \mathcal{T}_i^{jk} = 0_s, \tag{4.57}$$

Again, only two of these three equations are linearly independent. Alternatively, they may be obtained from (4.53) when $s = 1$ and $s = 2$ and the fact that $m'^j l'_j = 0$.

Point-point-point correspondence

This image correspondence is considered as the most particular case of the trifocal geometry since no lines are known but three points instead. The algebraic constraints which relate a point triplet can be worked out by first considering the lines through points $\mathbf{m}'$ and $\mathbf{m}''$. Three lines supporting each of these points may be defined by the coordinates $l'_j \simeq m'^q \epsilon_{qjs}$ and $l''_k \simeq m''^r \epsilon_{rkt}$ respectively, for $s, t = 1, 2, 3$. Substituting these definitions in (4.54) and taking all possible combinations of indices s and t give a total of nine equations

$$m^i (m'^q \epsilon_{qjs})(m''^r \epsilon_{rkt}) \mathcal{T}_i^{jk} = 0_{st}. \tag{4.58}$$

These trilinearities were first proposed by Shashua in relation to visual recognition tasks [78]. Only four of these equations are linearly independent since each pencil of lines passing through the points $\mathbf{m}'$ and $\mathbf{m}''$ contributes two independent constraints.

Line-line-line correspondence

Any line $\boldsymbol{l}$ in a projective plane is represented by a 3-vector so one may form the 3×3 skew-symmetric matrix $[\boldsymbol{l}]_\times$. By duality, a point $\mathbf{m}$ belonging to a line $\boldsymbol{l}$ may be written as $m^i \simeq l_q \epsilon^{qis}$, for fixed values of $s = 1, 2, 3$, and ϵ^{qis} is the length-3 contravariant kronecker tensor corresponding to lines. It follows that the (matrix) equations $[\boldsymbol{l}]_\times \mathbf{m} = \mathbf{0}$ become $l_i l_q \epsilon^{qis} = 0^s$ in tensor notation. Using (4.40) or replacing m^i in (4.54) by its line equivalent, we obtain three trilinearities of the form

$$(l_q \epsilon^{qis}) l'_j l''_k \mathcal{T}_i^{jk} = 0^s, \tag{4.59}$$

where only two of these three equations are linearly independent. If only the lines $\boldsymbol{l}$ and $\boldsymbol{l}'$ are known, then $\boldsymbol{l}''$ may be computed by solving the set of equations (4.59), hereby transferring $\boldsymbol{l}$ into the third image. A similar technique can be used to transfer lines to the second image.

Summary

One may have realised that not all the points and/or lines in three views are related by a trifocal tensor, only those that are in correspondence and satisfying the trilinear relations summarised in Table 4.2.

Correspondence	Trilinearities	Indep. Eqs.
Point-point-point	$m^i (m'^q \epsilon_{qjs})(m''^r \epsilon_{rkt}) \mathcal{T}_i^{jk} = 0_{st}$	4
Point-point-line	$m^i (m'^q \epsilon_{qjs}) l''_k \mathcal{T}_i^{jk} = 0_s$	2
Point-line-point	$m^i l'_j (m''^q \epsilon_{qks}) \mathcal{T}_i^{jk} = 0_s$	2
Point-line-line	$m^i l'_j l''_k \mathcal{T}_i^{jk} = 0$	1
Line-line-line	$(l_q \epsilon^{qis}) l'_j l''_k \mathcal{T}_i^{jk} = 0^s$	2

Table 4.2. Trilinearities and associated number of independent equations.

The 27 entries of the tensor are defined up to a common scale, so they may be computed provided at least 26 equations. Adapting the result in (4.11), the trifocal tensor associated with three uncalibrated cameras has $11 \times 3 - 15 = 18$ degrees of freedom. This implies that the 27 tensorial coefficients have to fulfil 9 constraints between themselves *in addition* to satisfying the trilinearities. One of these constraints can be eliminated by fixing the scale of the tensor.

This condition can be achieved by requiring that the vectorised form of the tensor, as defined in (4.18), has unit norm. Therefore, 8 constraints are essentially to be considered. Because these constraints apply exclusively to the trifocal elements and do not involve any data, they are referred to as *ancillary* or *internal constraints*, to be distinguished from the *principal constraints* expressed by the trilinearities.

Irrespective of the computation method used, a trifocal tensor which does not satisfy the internal constraints is termed an *unconstrained* tensor. For a point correspondence across three views, such a tensor gives projections for which the back-projected rays through the image points do not intersect in a precise 3-D point, see Figure 4.14(a). Constraining the tensor ensures that the rays intersect in space. Such a constrained tensor is said to be *geometrically valid* and allows a correct triangulation as in Figure 4.14(b).

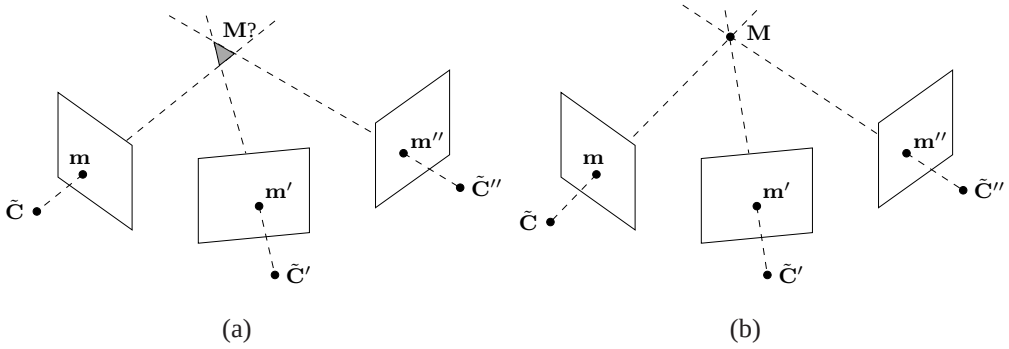


Figure 4.14. A triangulation example for: (a) an unconstrained tensor; (b) a geometrically valid tensor. In case (a), point $\mathbf{M}$ is not at the intersection of the three back-projected rays but lies in the region delimited by these rays. This uncertainty gives a poor reconstruction.

4.4 Properties of the tensorial slices

This section establishes properties of the tensorial slices which are fundamental to subsequently identify sets of ancillary constraints on the trifocal tensor. First, some notation must be introduced.

By analogy to (4.2), let the projection matrices of three views be defined in terms of their principal planes as

$$\mathtt{P} = \begin{bmatrix} \boldsymbol{\Gamma}_1 \\ \boldsymbol{\Gamma}_2 \\ \boldsymbol{\Gamma}_3 \end{bmatrix}, \ \mathtt{P}' = \begin{bmatrix} \boldsymbol{\Gamma}'_1 \\ \boldsymbol{\Gamma}'_2 \\ \boldsymbol{\Gamma}'_3 \end{bmatrix}, \text{ and } \mathtt{P}'' = \begin{bmatrix} \boldsymbol{\Gamma}''_1 \\ \boldsymbol{\Gamma}''_2 \\ \boldsymbol{\Gamma}''_3 \end{bmatrix}.$$

In view of (4.5), the principal rays corresponding to the intersection of pairs of projection planes will be denoted by $\boldsymbol{\mathcal{R}}_h$, $\boldsymbol{\mathcal{R}}'_h$, and $\boldsymbol{\mathcal{R}}''_h$ ($h = 1, 2, 3$). Exact definitions are provided for convenience in Table 4.3.

Image	Principal rays
Ψ_1	$\boldsymbol{\mathcal{R}}_1 \simeq \boldsymbol{\Gamma}_2 \triangle \boldsymbol{\Gamma}_3$, $\boldsymbol{\mathcal{R}}_2 \simeq \boldsymbol{\Gamma}_3 \triangle \boldsymbol{\Gamma}_1$, $\boldsymbol{\mathcal{R}}_3 \simeq \boldsymbol{\Gamma}_1 \triangle \boldsymbol{\Gamma}_2$
Ψ_2	$\boldsymbol{\mathcal{R}}'_1 \simeq \boldsymbol{\Gamma}'_2 \triangle \boldsymbol{\Gamma}'_3$, $\boldsymbol{\mathcal{R}}'_2 \simeq \boldsymbol{\Gamma}'_3 \triangle \boldsymbol{\Gamma}'_1$, $\boldsymbol{\mathcal{R}}'_3 \simeq \boldsymbol{\Gamma}'_1 \triangle \boldsymbol{\Gamma}'_2$
Ψ_3	$\boldsymbol{\mathcal{R}}''_1 \simeq \boldsymbol{\Gamma}''_2 \triangle \boldsymbol{\Gamma}''_3$, $\boldsymbol{\mathcal{R}}''_2 \simeq \boldsymbol{\Gamma}''_3 \triangle \boldsymbol{\Gamma}''_1$, $\boldsymbol{\mathcal{R}}''_3 \simeq \boldsymbol{\Gamma}''_1 \triangle \boldsymbol{\Gamma}''_2$

Table 4.3. Principal rays as intersection of principal planes.

In addition, we observe the naming convention of Table 4.4 to designate the epipolar lines obtained by projecting the principal rays of one image into the other two images. Several comments must be added regarding the information in this table.

Epipolar lines	Geometric incidence
$\boldsymbol{\mathcal{L}}_{2j} \simeq \mathtt{F}_{21}^{\mathsf{T}}\mathbf{e}_j$	Mapping of rays $\boldsymbol{\mathcal{R}}'_j$ of image Ψ_2 into image Ψ_1
$\boldsymbol{\mathcal{L}}_{3k} \simeq \mathtt{F}_{31}^{\mathsf{T}}\mathbf{e}_k$	Mapping of rays $\boldsymbol{\mathcal{R}}''_k$ of image Ψ_3 into image Ψ_1
$\boldsymbol{\mathcal{L}}'_{1i} \simeq \mathtt{F}_{21}\mathbf{e}_i$	Mapping of rays $\boldsymbol{\mathcal{R}}_i$ of image Ψ_1 into image Ψ_2
$\boldsymbol{\mathcal{L}}'_{3k} \simeq \mathtt{F}_{32}^{\mathsf{T}}\mathbf{e}_k$	Mapping of rays $\boldsymbol{\mathcal{R}}''_k$ of image Ψ_3 into image Ψ_2
$\boldsymbol{\mathcal{L}}''_{1i} \simeq \mathtt{F}_{31}\mathbf{e}_i$	Mapping of rays $\boldsymbol{\mathcal{R}}_i$ of image Ψ_1 into image Ψ_3
$\boldsymbol{\mathcal{L}}''_{2j} \simeq \mathtt{F}_{32}\mathbf{e}_j$	Mapping of rays $\boldsymbol{\mathcal{R}}'_j$ of image Ψ_2 into image Ψ_3

Table 4.4. Mapping of the principal rays of one image into epipolar lines in the other two images.

First, the principal rays $\boldsymbol{\mathcal{R}}_h$ in the first image are special optical lines going through the canonical points $\mathbf{e}_h$ and not arbitrary image points (recall Table 4.1). This is also true for principal rays $\boldsymbol{\mathcal{R}}'_h$ and $\boldsymbol{\mathcal{R}}''_h$, so the convention set forth in Table 4.4 applies to specific, "canonical", epipolar lines. It is hoped that this remark will clarify any confusion when we derive properties of the trifocal matrices in Proposition 6 which apply to general epipolar lines. Second, by analogy to (4.7), the canonical epipolar lines in view j may be expressed by the fundamental matrix $\mathtt{F}_{ji}$ acting on points $\mathbf{e}_h$ in view i. Therefore, the column vectors of $\mathtt{F}_{ji}$ give the coordinates of these lines. A similar argument to the derivation of (4.9) establishes that fundamental matrix $\mathtt{F}_{ij}(= \mathtt{F}_{ji}^{\mathsf{T}})$ sends points in view j to epipolar lines in view i.

Finally, two more notations are necessary. One is ${}^{j}\mathbf{c}_h$ to refer to the h-th column of homography $\mathcal{J}_j^{\mathsf{T}}$, or h-th row of $\mathcal{J}_j$. For instance, $\mathbf{r}_3$ and $\mathbf{t}_2$ in (4.33) are represented by ${}^{1}\mathbf{c}_3$ and ${}^{3}\mathbf{c}_2$ respectively. And the other notation is $\mathbf{e}_{ji}$ to represent the epipole obtained by projecting the camera centre of image i into image j. Examples of common epipoles include $\mathbf{e}' = \mathbf{e}_{21}$ and $\mathbf{e}'' = \mathbf{e}_{31}$.

4.4.1 The trifocal matrices

Evolved as special instances of matrix $\mathcal{I}_\mathbf{m}$ in (4.28), the trifocal matrices $\mathcal{I}_i$ are the first type of tensorial slices to be considered. This section provides details about the interframe geometry and 3-D incidence that occur when these correlation transformations are used. Their general properties are first uncovered, followed by some specialisations needed later to deduce internal constraints on the trifocal tensor.

Suppose that we have a point $\mathbf{m}$ in the first view and its corresponding epipolar line $\boldsymbol{l}'$ in the second view. The epipolar plane Π' back-projected from $\boldsymbol{l}'$ passes through the first two camera centres $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{C}}'$ and therefore contains the optical line $(\tilde{\mathbf{C}}\mathbf{m})$. Additionally, the plane back-projected from any line $\boldsymbol{l}''$ in the third view intersect Π' in a 3-D line $\boldsymbol{L}$. Since the ray $(\tilde{\mathbf{C}}\mathbf{m})$ lies entirely in Π', it must intersect the line $\boldsymbol{L}$, which means that $\mathbf{m}$, $\boldsymbol{l}'$, and $\boldsymbol{l}''$ constitute a point-line-line correspondence satisfying the constraint $\boldsymbol{l}'^{\mathsf{T}}\,[m^i\mathcal{I}_i]\,\boldsymbol{l}'' = 0$, see (4.55) in Section 4.3.3. This is true for *any* line $\boldsymbol{l}''$ in the third view, consequently $\boldsymbol{l}'^{\mathsf{T}}\,[m^i\mathcal{I}_i] = \mathbf{0}^{\mathsf{T}}$. In turn, this relation implies that epipolar line $\boldsymbol{l}'$ belongs to the left null-space of matrix $\mathcal{I}_\mathbf{m} = m^i\mathcal{I}_i$. A similar reasoning applied to line $\boldsymbol{l}''$ demonstrates that $[m^i\mathcal{I}_i]\,\boldsymbol{l}'' = \mathbf{0}$. Thus, $\boldsymbol{l}''$ is in the right null-space of matrix $\mathcal{I}_\mathbf{m}$. The next proposition summarises these properties.

Proposition 6 *If $\mathbf{m} = [m^i]$ is a point in the first view with corresponding epipolar lines $\boldsymbol{l}'$ and $\boldsymbol{l}''$ in the second and third views respectively, then $\boldsymbol{l}'$ and $\boldsymbol{l}''$ are members of the left and right null-spaces of the matrix $\mathcal{I}_\mathbf{m} = m^i\mathcal{I}_i$, that is,*

$$\boldsymbol{l}'^{\mathsf{T}}\left[m^i\mathcal{I}_i\right] = \mathbf{0}^{\mathsf{T}} \quad \textit{and} \quad \left[m^i\mathcal{I}_i\right]\boldsymbol{l}'' = \mathbf{0}.$$

Three particular instantiations of this proposition occur when point $\mathbf{m}$ is chosen to have coordinates $\mathbf{e}_1 = [1,0,0]^{\mathsf{T}}, \mathbf{e}_2 = [0,1,0]^{\mathsf{T}}$, and $\mathbf{e}_3 = [0,0,1]^{\mathsf{T}}$. In this setting, the epipolar lines in the left and right null-spaces of matrix $[m^i\mathcal{I}_i]$ are readily computable from the trifocal matrices $\mathcal{I}_1, \mathcal{I}_2$, and $\mathcal{I}_3$ respectively. These lines originate as the projection in the second and third views of the principal rays $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ because these rays are the optical lines through the points $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ in the first image. Referring to Table 4.4, these epipolar lines are $\mathcal{L}'_{1i}$ and $\mathcal{L}''_{1i}$ respectively.

Proposition 7 *When point* $\mathbf{m}$ *is represented by homogeneous coordinates* $\mathbf{e}_i$, $i = 1, 2, 3$, *we have*

$$\boldsymbol{\mathcal{L}}_{1i}'^{\mathsf{T}}\boldsymbol{\mathcal{T}}_i = \mathbf{0}^{\mathsf{T}} \quad \textit{and} \quad \boldsymbol{\mathcal{T}}_i\boldsymbol{\mathcal{L}}_{1i}'' = \mathbf{0}.$$

Since all epipolar lines pass through an epipole, this is true for the lines $\boldsymbol{\mathcal{L}}_{1i}'$ and $\boldsymbol{\mathcal{L}}_{1i}''$ which go through $\mathbf{e}_{21} = \mathbf{e}'$ and $\mathbf{e}_{31} = \mathbf{e}''$. A direct implication of this result is that both of these epipoles can be computed as the intersection of their respective epipolar lines. We can express these intersections using the conditions that $\mathbf{e}'^{\mathsf{T}}\boldsymbol{\mathcal{L}}_{1i}' = 0$ and $\mathbf{e}''^{\mathsf{T}}\boldsymbol{\mathcal{L}}_{1i}'' = 0$ for all $i = 1, 2, 3$.

Proposition 8 *Epipole* $\mathbf{e}'$ *in the second view is the common intersection of epipolar lines* $\boldsymbol{\mathcal{L}}_{11}'$, $\boldsymbol{\mathcal{L}}_{12}'$, *and* $\boldsymbol{\mathcal{L}}_{13}'$, *that is,*

$$\mathbf{e}'^{\mathsf{T}}[\boldsymbol{\mathcal{L}}_{11}', \boldsymbol{\mathcal{L}}_{12}', \boldsymbol{\mathcal{L}}_{13}'] = \mathbf{0}.$$

Similarly, epipole $\mathbf{e}''$ *is the common intersection of epipolar lines* $\boldsymbol{\mathcal{L}}_{11}''$, $\boldsymbol{\mathcal{L}}_{12}''$, *and* $\boldsymbol{\mathcal{L}}_{13}''$, *that is,*

$$\mathbf{e}''^{\mathsf{T}}[\boldsymbol{\mathcal{L}}_{11}'', \boldsymbol{\mathcal{L}}_{12}'', \boldsymbol{\mathcal{L}}_{13}''] = \mathbf{0}.$$

To summarise, it was shown that epipole $\mathbf{e}'$ (resp. $\mathbf{e}''$) is in the null-space of the epipolar lines $\boldsymbol{\mathcal{L}}_{1i}'$ (resp. $\boldsymbol{\mathcal{L}}_{1i}''$), which themselves are in the left (resp. right) null-space of the trifocal matrices $\boldsymbol{\mathcal{T}}_i$. The previous two propositions were originally established by Hartley [37] but also appeared under a different form in [29]. They outline two important properties of the trifocal matrices which will be investigated further in the remaining part of this section.

The relation $\boldsymbol{\mathcal{L}}_{1i}'^{\mathsf{T}}\boldsymbol{\mathcal{T}}_i = \mathbf{0}^{\mathsf{T}}$ implies that the columns of matrix $\boldsymbol{\mathcal{T}}_i$ are three points in the second view[1] lying on epipolar line $\boldsymbol{\mathcal{L}}_{1i}'$ (as the product of each column vector of $\boldsymbol{\mathcal{T}}_i$ with $\boldsymbol{\mathcal{L}}_{1i}'^{\mathsf{T}}$ vanishes). Now, the k-th column of the trifocal matrices can be obtained by carrying out the operation $\boldsymbol{\mathcal{T}}_i\mathbf{e}_k$, where $\mathbf{e}_k$ is a canonical basis vector. This vector may be viewed as representing the coordinates of a particular line $\boldsymbol{l}''$ in the relation $\boldsymbol{l}'^{\mathsf{T}}\,[m^i\boldsymbol{\mathcal{T}}_i]\,\boldsymbol{l}'' = 0$. From Table 4.1, the back-projection of point $\mathbf{e}_i$ in the first image is the principal ray $\boldsymbol{\mathcal{R}}_i$ and the back-projection of line $\mathbf{e}_k$ in the third image is the principal plane $\boldsymbol{\Gamma}_k''$. Therefore, the k-th column of matrix $\boldsymbol{\mathcal{T}}_i$ is an image point which is the projection in the second view of the intersection of $\boldsymbol{\mathcal{R}}_i$ and $\boldsymbol{\Gamma}_k''$, see Figure 4.15.

Analogously, the fact that $\boldsymbol{\mathcal{T}}_i\boldsymbol{\mathcal{L}}_{1i}'' = \mathbf{0}$ means that the rows of matrix $\boldsymbol{\mathcal{T}}_i$ are three points in the third view lying on epipolar line $\boldsymbol{\mathcal{L}}_{1i}''$. The j-th row vector of the trifocal matrices can be obtained as $\boldsymbol{\mathcal{T}}_i^{\mathsf{T}}\mathbf{e}_j$, where $\mathbf{e}_j$ is a canonical basis vector representing a line in the second view.

[1] This result justifies notation (4.25) for the columns of the trifocal matrices denoted with one prime as for entities in the second view.

The line $\mathbf{e}_j$ back-projects as principal plane $\mathbf{\Gamma}'_j$. Therefore, the j-th row of matrix $\boldsymbol{\mathcal{I}}_i$ is an image point which is the projection in the third view of the intersection of $\boldsymbol{\mathcal{R}}_i$ and $\mathbf{\Gamma}'_j$. The findings are summarised in Proposition 9.

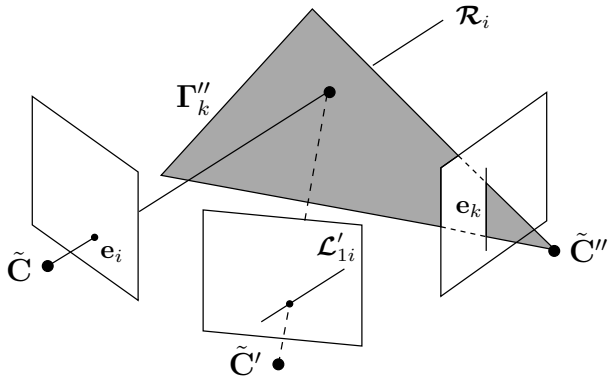


Figure 4.15. Geometric construction to interpret a column vector of matrix $\boldsymbol{\mathcal{I}}_i$.

Proposition 9 *The column vectors of each $\boldsymbol{\mathcal{I}}_i$ matrix are three distinct points in the second view lying on epipolar line $\boldsymbol{\mathcal{L}}'_{1i}$, whereas the row vectors of these matrices are three distinct points in the third view lying on epipolar line $\boldsymbol{\mathcal{L}}''_{1i}$.*

A further interpretation of this proposition is that the h-th column vector of matrices $\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2, \boldsymbol{\mathcal{I}}_3$ represents the vertex of a triangle $\mathbf{T}'_h$ in the second view for each value of $h = 1, 2, 3$, refer to Figure 4.16. A similar remark can be made about the rows of each matrix $\boldsymbol{\mathcal{I}}_i$. The h-th row of the trifocal matrices represents the vertex of a triangle $\mathbf{T}''_h$ in the third view. An illustration of this geometry is given later in Figure 4.19(b) when investigating matrices $\boldsymbol{\mathcal{K}}_k$.

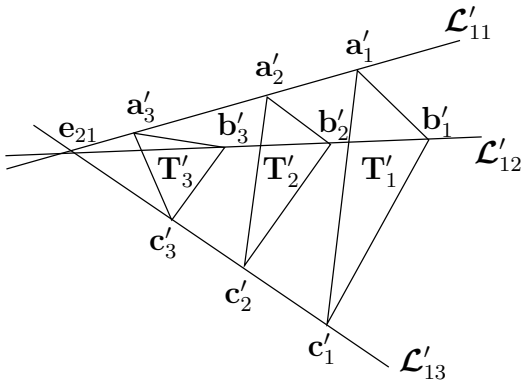


Figure 4.16. Geometric interpretation of the columns of matrices $\boldsymbol{\mathcal{I}}_i$.

Some general properties of the trifocal matrices can now be stated.

Proposition 10 *Matrix $\mathcal{I}_i$ describes a correlation between lines in the third view and points in the second view induced by the principal planes in the third view. Matrix $\mathcal{I}_i^\mathsf{T}$ maps lines in the second view to points in the third view via the principal planes in the second view.*

Clearly, since $\mathcal{I}_\mathbf{m}$ is a linear combination of the trifocal matrices, it ought to perform the same geometrical action. For instance, one may deduce that the operation $\mathcal{I}_\mathbf{m} l''$ gives a point $\mathbf{m}'$ in Ψ_2 (as per (4.46)) which is the projection of the intersection of the optical ray $(\tilde{\mathbf{C}}\mathbf{m})$ of Ψ_1 and the plane back-projected from line l'' in Ψ_3. A formal algebraic proof can be found in [73].

4.4.2 The homography matrices

We now proceed to analyse the properties of homography slices $\mathcal{J}_j$. Derivations related to homographies $\mathcal{K}_k$ are similar and have been placed in Appendix C to avoid redundancy in the chapter.

To begin, note that a special case of Proposition 4 occurs when the line l' is one of the canonical basis vectors $\mathbf{e}_h$ ($h = 1, 2, 3$) of $\mathbb{R}^3$. From Table 4.1, we know that each vector $\mathbf{e}_h$ can be thought of as a line belonging to the h-th principal plane of view Ψ_2. When $l' = [1, 0, 0]^\mathsf{T}$ we have

$$h_i^k = [\mathcal{I}_1^\mathsf{T}\mathbf{e}_1, \mathcal{I}_2^\mathsf{T}\mathbf{e}_1, \mathcal{I}_3^\mathsf{T}\mathbf{e}_1] = \mathcal{J}_1.$$

The above equality can be checked by constructing a matrix with the first column of each of the matrices $\mathcal{I}_1^\mathsf{T}, \mathcal{I}_2^\mathsf{T}$, and $\mathcal{I}_3^\mathsf{T}$ as given in (4.26) and compare it with the matrix $\mathcal{J}_1$ in (4.32). This result shows that matrix $\mathcal{J}_1$ represents a point homography from view Ψ_1 to view Ψ_3 induced by the first principal plane in view Ψ_2. Furthermore, $\mathcal{J}_1^\mathsf{T}$ maps lines in Ψ_3 to lines in Ψ_1 via the first principal plane. Using $l' = \mathbf{e}_2$ (resp. $\mathbf{e}_3$) selects the second (resp. third) principal plane in view Ψ_2 so

$$\mathcal{J}_2 = [\mathcal{I}_1^\mathsf{T}\mathbf{e}_2, \mathcal{I}_2^\mathsf{T}\mathbf{e}_2, \mathcal{I}_3^\mathsf{T}\mathbf{e}_2] \quad \left(\text{resp. } \mathcal{J}_3 = [\mathcal{I}_1^\mathsf{T}\mathbf{e}_3, \mathcal{I}_2^\mathsf{T}\mathbf{e}_3, \mathcal{I}_3^\mathsf{T}\mathbf{e}_3]\right).$$

This leads to the following proposition.

Proposition 11 *Matrix $\mathcal{J}_j$ describes a homography between points in the first view and points in the third view induced by the j-th principal plane in the second view. Matrix $\mathcal{J}_j^\mathsf{T}$ maps lines in the third view to lines in the first view via the same principal plane.*

This proposition sheds light on the underlying geometic transfer implied by matrix $\mathcal{J}_{l'}$, the linear combination of slices $\mathcal{J}_j$, introduced in (4.34). The operation $\mathcal{J}_{l'}\mathbf{m}$ gives a point $\mathbf{m}''$ in Ψ_3 via the plane back-projected from line l' in Ψ_2.

Now, consider $\mathbf{e}_1$ as a point on the first principal ray $\boldsymbol{\mathcal{R}}_1$ of the image Ψ_1. Applying Proposition 11 to transfer $\mathbf{e}_1$ with the homography $\boldsymbol{\mathcal{J}}_j$ gives a point in the third image via the j-th principal plane in view Ψ_2. The particular coordinates of $\mathbf{e}_1$ implies that the transferred point is given by the first column of matrix $\boldsymbol{\mathcal{J}}_j$. A generalisation of this result is straightforward: the homography $\boldsymbol{\mathcal{J}}_j$ acting on the canonical point $\mathbf{e}_h$ produces a point in image Ψ_3 with coordinates given as the h-th column of $\boldsymbol{\mathcal{J}}_j$.

Referring to Proposition 11 again, $\boldsymbol{\mathcal{J}}_j^\mathsf{T}$ is a line homography from view Ψ_3 to view Ψ_1. Considering vectors $\mathbf{e}_h$ as homogeneous lines in view Ψ_3, it can be deduced that the columns of $\boldsymbol{\mathcal{J}}_j^\mathsf{T}$, or equivalently the rows of $\boldsymbol{\mathcal{J}}_j$, are lines in the reference view. These properties are recapitulated below.

Proposition 12 *The column vectors of each $\boldsymbol{\mathcal{J}}_j$ matrix are three distinct points in the third view, whereas the row vectors of these matrices are three distinct lines in the first view.*

The remaining part of this section establishes further properties of matrices $\boldsymbol{\mathcal{J}}_j$ based on the work done so far.

By definition, the principal rays of one image are mapped to epipolar lines in the other images. Now, consider the point $\mathbf{e}_i$ on principal ray $\boldsymbol{\mathcal{R}}_i$ in the first view. The intersection in space of the line $\boldsymbol{\mathcal{R}}_i$ with the three principal planes $\boldsymbol{\Gamma}'_j$ of view Ψ_2 gives three distinct world points $\mathbf{V}_{ij}$, where index i is fixed by choosing the ray $\boldsymbol{\mathcal{R}}_i$ and j varies from 1 to 3 corresponding to the planes $\boldsymbol{\Gamma}'_j$ of Ψ_2. An example is shown in Figure 4.17 for $\boldsymbol{\mathcal{R}}_1$.

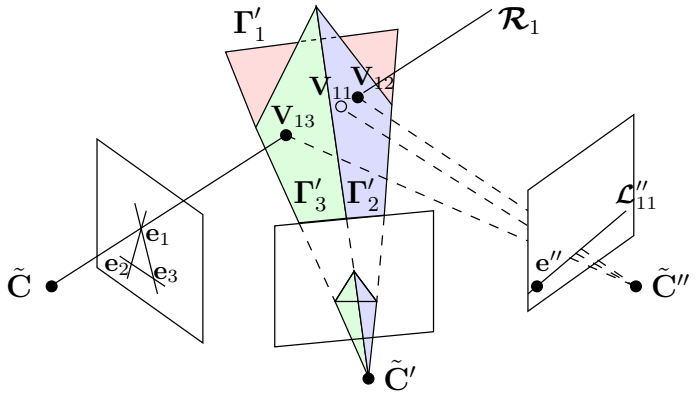


Figure 4.17. Point transfer via the principal planes $\boldsymbol{\Gamma}'_j$ in view Ψ_2.

Since all three world points lie on $\boldsymbol{\mathcal{R}}_i$, they are collinear and so must be their projections in view Ψ_3. From Table 4.4, the ray $\boldsymbol{\mathcal{R}}_i$ maps to the epipolar line $\boldsymbol{\mathcal{L}}''_{1i}$ in Ψ_3 where the corresponding point of $\mathbf{e}_i$ lies. Employing $\boldsymbol{\Gamma}'_j$, the j-th principal plane of Ψ_2, to transfer $\mathbf{e}_i$ means that

homography $\mathcal{J}_j$ carries out the point transfer between the views. Moreover, the i-th coordinate of $\mathbf{e}_i$ being unital implies that the i-th column of $\mathcal{J}_j$ gives the coordinates of the corresponding point in Ψ_3. This result is stated in the next proposition and algebraic details appear in Table 4.5.

Proposition 13 *The three image points obtained by selecting the i-th column vector of matrices $\mathcal{J}_1, \mathcal{J}_2$, and $\mathcal{J}_3$ are collinear and lie on the epipolar line $\mathcal{L}''_{1i}$ in the third view.*

A consequence of this proposition is that the three columns of a particular matrix $\mathcal{J}_j$ represent points (by Proposition 12) lying on different epipolar lines in image Ψ_3, namely the lines $\mathcal{L}''_{11}$, $\mathcal{L}''_{12}$, and $\mathcal{L}''_{13}$. Therefore, for a given matrix $\mathcal{J}_j$, its column vectors define the vertices of a triangle $\mathbf{T}''_j$ in Ψ_3. Since all epipolar lines meet at the epipole, it follows that the triangles associated with $\mathcal{J}_1, \mathcal{J}_2$, and $\mathcal{J}_3$ are in perspective from the epipole $\mathbf{e}_{31} = \mathbf{e}''$. This is illustrated in Figure 4.19(b).

Image points in Ψ_3	On epipolar line	From world points
$\mathbf{d}''_j \simeq \mathcal{J}_j \mathbf{e}_1$	$\mathcal{L}''_{11}$	$\mathbf{V}_{1j} \simeq \mathcal{R}_1 \triangle \mathbf{\Gamma}'_j$
$\mathbf{e}''_j \simeq \mathcal{J}_j \mathbf{e}_2$	$\mathcal{L}''_{12}$	$\mathbf{V}_{2j} \simeq \mathcal{R}_2 \triangle \mathbf{\Gamma}'_j$
$\mathbf{f}''_j \simeq \mathcal{J}_j \mathbf{e}_3$	$\mathcal{L}''_{13}$	$\mathbf{V}_{3j} \simeq \mathcal{R}_3 \triangle \mathbf{\Gamma}'_j$

Table 4.5. Algebraic and geometric properties of the columns of matrices $\mathcal{J}_j$.

Now, recall from Proposition 12 that the row vectors of a particular matrix $\mathcal{J}_j$ represent distinct lines in the first image, hence ${}^j\mathbf{c}_1, {}^j\mathbf{c}_2$, and ${}^j\mathbf{c}_3$ define a triangle $\mathbf{T}_j$ in Ψ_1. Geometrically, these lines correspond to the images of three world lines $\mathbf{L}_{j1}, \mathbf{L}_{j2}$, and $\mathbf{L}_{j3}$ arising as the intersection of the j-th principal plane $\mathbf{\Gamma}'_j$ of view Ψ_2 with the principal planes $\mathbf{\Gamma}''_1, \mathbf{\Gamma}''_2$, and $\mathbf{\Gamma}''_3$ of view Ψ_3 respectively. Figure 4.18 depicts the situation when $\mathbf{\Gamma}'_j$ meets with one of the planes $\mathbf{\Gamma}''_k$. Algebraically, lines ${}^j\mathbf{c}_h$ (for $h = 1, 2, 3$) are obtained by applying homography $\mathcal{J}_j^{\mathsf{T}}$ to the three canonical lines $\mathbf{e}_h$ in Ψ_3.

Looking at the vertices of triangle $\mathbf{T}_j$, one may compute them as the intersection of two lines of $\mathbf{T}_j$, effectively by cross product of two rows of $\mathcal{J}_j$. A particular vertex cannot be calculated from any two arbitrary rows though. Vertex $\mathbf{v}_{jk}$ corresponds to the projection of a world point $\mathbf{V}''_{kj}$, which emanates as the intersection of the ray $\mathcal{R}''_k$ and the plane $\mathbf{\Gamma}'_j$. From Table 4.3, $\mathcal{R}''_k$ is the meet of the planes $\mathbf{\Gamma}''_\alpha$ and $\mathbf{\Gamma}''_\beta$, where k, α, and β take distinct values in the range $\{1, 2, 3\}$. Hence, we can write $\mathbf{V}''_{kj} \simeq \mathbf{\Gamma}'_j \triangle \mathcal{R}''_k \simeq \mathbf{\Gamma}'_j \triangle \mathbf{\Gamma}''_\alpha \triangle \mathbf{\Gamma}''_\beta$. Proposition 14 recapitulates these remarks. Of more practical interest, Table 4.6 gives details about the various algebraic operations and associated geometry which can be elaborated from the rows of matrices $\mathcal{J}_j$.

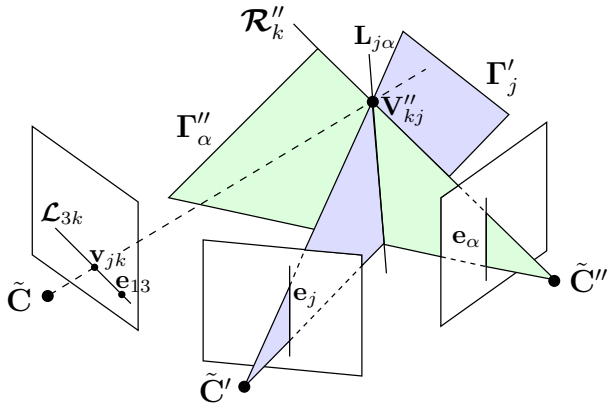


Figure 4.18. Vertex $\mathbf{v}_{jk}$ arises in the first view by projecting the world point $\mathbf{V}''_{kj}$ obtained as the meet of the principal ray $\boldsymbol{\mathcal{R}}''_k \simeq \boldsymbol{\Gamma}''_\alpha \triangle \boldsymbol{\Gamma}''_\beta$ and the world line $\mathbf{L}_{j\alpha} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\Gamma}''_\alpha$.

Proposition 14 *The three image lines given by the row vectors of matrix $\boldsymbol{\mathcal{J}}_j$ form a triangle $\mathbf{T}_j$ in the first view. Vertices $\mathbf{v}_{j1}, \mathbf{v}_{j2}$, and $\mathbf{v}_{j3}$ of $\mathbf{T}_j$ can be computed as the cross product of two lines represented by the appropriate rows of $\boldsymbol{\mathcal{J}}_j$, see Table 4.6.*

Triangle	Edges	From world lines	Vertices	From world points
	${}^j\mathbf{c}_1 \simeq \boldsymbol{\mathcal{J}}_j^\top \mathbf{e}_1$	$\mathbf{L}_{j1} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\Gamma}''_1$	$\mathbf{v}_{j1} \simeq {}^j\mathbf{c}_2 \times {}^j\mathbf{c}_3$	$\mathbf{V}''_{1j} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\mathcal{R}}''_1$
$\mathbf{T}_j$	${}^j\mathbf{c}_2 \simeq \boldsymbol{\mathcal{J}}_j^\top \mathbf{e}_2$	$\mathbf{L}_{j2} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\Gamma}''_2$	$\mathbf{v}_{j2} \simeq {}^j\mathbf{c}_3 \times {}^j\mathbf{c}_1$	$\mathbf{V}''_{2j} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\mathcal{R}}''_2$
	${}^j\mathbf{c}_3 \simeq \boldsymbol{\mathcal{J}}_j^\top \mathbf{e}_3$	$\mathbf{L}_{j3} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\Gamma}''_3$	$\mathbf{v}_{j3} \simeq {}^j\mathbf{c}_1 \times {}^j\mathbf{c}_2$	$\mathbf{V}''_{3j} \simeq \boldsymbol{\Gamma}'_j \triangle \boldsymbol{\mathcal{R}}''_3$

Table 4.6. Algebraic and geometric properties of the rows of matrices $\boldsymbol{\mathcal{J}}_j$.

Further geometric relations can be inferred from the rows of $\boldsymbol{\mathcal{J}}_j$. We show next that corresponding vertices of different triangles are aligned.

First, recall from the derivation preceding Proposition 14 that the vertices $\mathbf{v}_{j1}, \mathbf{v}_{j2}$, and $\mathbf{v}_{j3}$ are the mappings of world points obtained by fixing a plane $\boldsymbol{\Gamma}'_j$ in Ψ_2 and intersecting it with the three rays $\boldsymbol{\mathcal{R}}''_k$ in Ψ_3. Since these 3-D points belong to three different rays in space, their projections in the first image are three non-collinear points which constitute a triangle $\mathbf{T}_j$. Now, suppose that we fix the ray $\boldsymbol{\mathcal{R}}''_k$ in Ψ_3 and intersect it with the three planes $\boldsymbol{\Gamma}'_j$ of Ψ_2. The world points in this case are all collinear on the ray $\boldsymbol{\mathcal{R}}''_k$, hence they project to three vertices which lie on the image of that ray in Ψ_1, that is, the epipolar line $\boldsymbol{\mathcal{L}}_{3k}$ going through the epipole $\mathbf{e}_{13}$. By construction of the world points and definition of vertex $\mathbf{v}_{jk}$, we deduce that the vertices on $\boldsymbol{\mathcal{L}}_{3k}$ are $\mathbf{v}_{1k}, \mathbf{v}_{2k}$, and $\mathbf{v}_{3k}$. Table 4.6 readily provides rules for computation of these points from the rows of matrices $\boldsymbol{\mathcal{J}}_j$. These results are illustrated in Figure 4.19(a) and summed up as follows.

Proposition 15 *Vertices* $\mathbf{v}_{1k}, \mathbf{v}_{2k}$, *and* $\mathbf{v}_{3k}$ *of triangles* $\mathbf{T}_1, \mathbf{T}_2$, *and* $\mathbf{T}_3$ *are collinear and lie on the epipolar line* $\boldsymbol{\mathcal{L}}_{3k}$ *in the first view.*

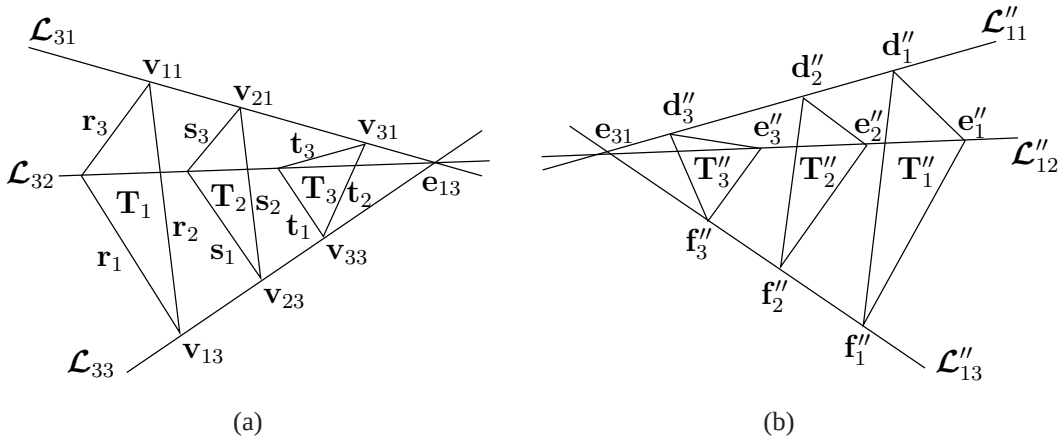


Figure 4.19. Geometric interpretation of the (a) rows and (b) columns of matrices $\boldsymbol{\mathcal{J}}_j$.

The next series of properties are derived from an application of Proposition 11 using the epipoles in the first image.

When the cameras are in general position, the three homographies $\boldsymbol{\mathcal{J}}_j$ relate epipoles $\mathbf{e}_{13}$ in the first image to $\mathbf{e}_{31}$ in the third image according to

$$\mathbf{e}_{31} \simeq \boldsymbol{\mathcal{J}}_j \mathbf{e}_{13} \quad j = 1, 2, 3. \tag{4.60}$$

Given that $\mathbf{e}_{31}$ is the image of $\mathbf{e}_{13}$ by any of the three transformations $\boldsymbol{\mathcal{J}}_j$, we may choose two of these homographies and reformulate (4.60) as a generalised eigenvalue problem of the form

$$(\boldsymbol{\mathcal{J}}_p - \kappa \boldsymbol{\mathcal{J}}_q)\mathbf{x} = \mathbf{0}, \tag{4.61}$$

where κ is a non-zero scalar representing the generalised eigenvalue associated with point $\mathbf{x}$ in the first image. By construction, the solution $\mathbf{x} = \mathbf{e}_{13}$ is a generalised eigenvector for this problem. More details are given in the proposition below. These properties were defined by Ressl [73] but originally proposed by Canterakis [12].

Proposition 16 *The generalised eigenvalue problem* $(\boldsymbol{\mathcal{J}}_p - \kappa \boldsymbol{\mathcal{J}}_q)\mathbf{x} = \mathbf{0}$, *with* $\mathbf{x}$ *a point in the first image and regular homography* $\boldsymbol{\mathcal{J}}_q$, $p \neq q$, *has the following general eigenvalues:*

1. $\kappa_2 = \mathbf{e}_{21}^p / \mathbf{e}_{21}^q$ *is a double generalised eigenvalue with corresponding two-dimensional eigenspace spanned by the epipolar line* $\boldsymbol{\mathcal{L}}_{2r}$, *where* $\{p, q, r\}$ *is a permutation of* $\{1, 2, 3\}$.

2. $\kappa_1 = \mathbf{e}_{23}^p / \mathbf{e}_{23}^q$ *is a single generalised eigenvalue with corresponding one-dimensional eigenspace spanned by the epipole* $\mathbf{e}_{13}$.

Proof. To begin, we show that $\mathbf{e}_{21}^p/\mathbf{e}_{21}^q$ is a double eigenvalue. Upon replacing slices $\boldsymbol{\mathcal{J}}_p$ and $\boldsymbol{\mathcal{J}}_q$ by their algebraic expression evolved from (4.30) and adapting the notation for the epipoles, we obtain

$$\boldsymbol{\mathcal{J}}_p - \kappa\boldsymbol{\mathcal{J}}_q = \mathbf{e}_{31}(\mathbf{e}_p - \kappa\mathbf{e}_q)^\mathsf{T}\mathtt{A} - \mathbf{e}_{21}^\mathsf{T}(\mathbf{e}_p - \kappa\mathbf{e}_q)\mathtt{B}. \tag{4.62}$$

When $\kappa = \mathbf{e}_{21}^p/\mathbf{e}_{21}^q$, the factor preceding matrix $\mathtt{B}$ vanishes so (4.61) reduces to

$$\mathbf{e}_{31}(\mathbf{e}_p - \kappa\mathbf{e}_q)^\mathsf{T}\mathtt{A}\mathbf{x} = \mathbf{0}. \tag{4.63}$$

It turns out that this equation can be simplified by identifying the quantity $(\mathbf{e}_p - \kappa\mathbf{e}_q)^\mathsf{T}\mathtt{A}$ with the epipolar line $\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}$. This is shown next.

Consider the equality

$$(\mathbf{e}_p - \kappa\mathbf{e}_q) = -\frac{1}{\mathbf{e}_{21}^q}[\mathbf{e}_{21}]_\times^\mathsf{T}\mathbf{e}_r, \tag{4.64}$$

where the indices p, q, r take distinct values in the set $\{1, 2, 3\}$ and $\mathbf{e}_r$ is a 3-vector with r-th entry equal to 1 and other entries zero. One may check that the right-hand side of this equality, which can be rewritten as $(\mathbf{e}_{21} \times \mathbf{e}_r)/\mathbf{e}_{21}^q$, produces the same result as the left-hand side for any choice of $\{p, q, r\}$ taken as a permutation of $\{1, 2, 3\}$. Now, using (4.12) in Proposition 3 with $\mathtt{P} = [\mathtt{I}_{3\times3} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{A} \mid \mathbf{e}_{21}]$, the fundamental matrix between views one and two is given by

$$\mathtt{F}_{21} \simeq [\mathbf{e}_{21}]_\times\mathtt{A}. \tag{4.65}$$

Multiplying (4.64) on the left by $\mathtt{A}^\mathsf{T}$ and simplifying with (4.65) shows that

$$\mathtt{A}^\mathsf{T}(\mathbf{e}_p - \kappa\mathbf{e}_q) \simeq \mathtt{A}^\mathsf{T}[\mathbf{e}_{21}]_\times^\mathsf{T}\mathbf{e}_r = \mathtt{F}_{21}^\mathsf{T}\mathbf{e}_r.$$

From Table 4.4, vector $\mathtt{F}_{21}^\mathsf{T}\mathbf{e}_r$ gives the coordinates of epipolar line $\boldsymbol{\mathcal{L}}_{2r}$, the image of the ray $\boldsymbol{\mathcal{R}}_r'$ of the image Ψ_2 into the image Ψ_1. With this, (4.63) can be written as

$$\mathbf{e}_{31}\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}\mathbf{x} = \mathbf{0}. \tag{4.66}$$

As long as $\tilde{\mathrm{C}} \neq \tilde{\mathrm{C}}''$, the epipole $\mathbf{e}_{31}$ exists and is non-zero. Pre-multiplying by $\mathbf{e}_{31}^\mathsf{T}$ gives

$$\mathbf{e}_{31}^\mathsf{T}\mathbf{e}_{31}\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}\mathbf{x} = \|\mathbf{e}_{31}\|^2\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}\mathbf{x} = 0,$$

hence $\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}\mathbf{x} = 0$. This means that the one-dimensional epipolar line $\boldsymbol{\mathcal{L}}_{2r}$ represents a two-dimensional eigenspace in $\mathbb{R}^3$ associated with a double generalised eigenvalue $\kappa = \kappa_2 = \mathbf{e}_{21}^p/\mathbf{e}_{21}^q$. In other words, the points $\mathbf{x}^*$ on epipolar line $\boldsymbol{\mathcal{L}}_{2r}$ are generalised eigenvectors for the pair of homographic slices $\boldsymbol{\mathcal{J}}_p$ and $\boldsymbol{\mathcal{J}}_q$. They span an eigenspace that is orthogonal to the r-th column of the fundamental matrix $\mathtt{F}_{12}(= \mathtt{F}_{21}^\mathsf{T})$ since $\boldsymbol{\mathcal{L}}_{2r}^\mathsf{T}\mathbf{x}^* = (\mathtt{F}_{12}\mathbf{e}_r)^\mathsf{T}\mathbf{x}^* = 0$.

In general, the epipole $\mathbf{e}_{13}$ does not lie on the line $\boldsymbol{\mathcal{L}}_{2r}$ because $\boldsymbol{\mathcal{L}}_{2r}$ is independent of the third image so $(\boldsymbol{\mathcal{J}}_p - \kappa\boldsymbol{\mathcal{J}}_q)\mathbf{e}_{13}$ does not vanish for $\kappa = \kappa_2$. Multiplying (4.62) on the right by $\mathbf{e}_{13}$ gives

$$\mathbf{e}_{31}(\mathbf{e}_p - \kappa\mathbf{e}_q)^\mathsf{T}\mathbf{e}_{23} \tag{4.67}$$

since $\mathtt{A}\mathbf{e}_{13} \simeq \mathbf{e}_{23}$ and $\mathtt{B}\mathbf{e}_{13} = \mathbf{0}$. This expression vanishes when $\kappa = \kappa_1 = \mathbf{e}_{23}^p / \mathbf{e}_{23}^q$. So κ_1 is the generalised eigenvalue corresponding to the eigenvector $\mathbf{e}_{13}$. □

It should be noted that each matrix $\mathcal{J}_q$ must have maximum rank because the denominator in all generalised eigenvalues must be non-zero. Furthermore, the eigenspaces of the matrices $(\mathcal{J}_p - \kappa_i \mathcal{J}_q), i = 1, 2$, represent the null-spaces of these matrices. Indeed, these eigenspaces satisfy the relations $(\mathcal{J}_p - \kappa_i \mathcal{J}_q)\boldsymbol{\xi}_i = \mathbf{0}$, where $\boldsymbol{\xi}_i$ here stands for the eigenspace corresponding to κ_i. Proposition 16 indicates that the eigenspaces/null-spaces are nontrivial so the matrices $(\mathcal{J}_p - \kappa_i \mathcal{J}_q)$ must be rank deficient. In addition, the fact that a particular null-space has dimension equal to the multiplicity of its corresponding eigenvalue suggests that matrices $(\mathcal{J}_p - \kappa_1 \mathcal{J}_q)$ and $(\mathcal{J}_p - \kappa_2 \mathcal{J}_q)$ have rank two and one respectively. It will be proved in Section 4.5.2 that if the conditions of Proposition 16 are met, then the 27 numbers of the $\mathcal{J}_j$ slices constitute a geometrically valid trifocal tensor.

The numerous results derived in this section (and Appendix C) show that the tensorial matrices $\mathcal{I}_i, \mathcal{J}_j$, and $\mathcal{K}_k$ act as *generators* of the trifocal geometry because they underpin any general inter-image relations expressed by the matrices $\mathcal{I}_\mathbf{m}, \mathcal{J}_{l'}$, and $\mathcal{K}_{l''}$. Furthermore, the tensorial matrices are defined from and apply to canonical basis vectors representing either lines or points in the images, so they intrinsically give a canonical representation of the trifocal geometry. Applying constraints on these slices would automatically restrain the entire trinocular geometry underlying the relationships between general line and point matches in three views. This is why they are so important to consider in order to ensure a geometrically valid tensor.

Formulating ancillary constraints on the trifocal tensor now comes as a natural extension after considering the properties of the tensorial matrices. The next section presents the most important sets of algebraic constraints which have been proposed to date in the literature. The description follows their chronological order of appearance and reveals a shift in research focus, from correlation to homography slices, necessary to identify minimal sufficient sets. From an application viewpoint, taking ancillary constraints into account is a mantatory step to compute a final tensor of practical use.

4.5 Ancillary constraints on the trifocal tensor

Although multiple-view geometry is well established [88, 89], its conversion into usable sets of algebraic constraints has turned out to be a majorly difficult task. In the case of three views, some researchers have proposed simplified versions of the full projective trifocal tensor to reduce the number of ancillary constraints. For instance, an affine tensor (corresponding to a weak perspective camera model) can be used in some situations to approximate a generic tensor [64].

The following sections consider the general projective case and describe ancillary constraints underpinning three perspective views of a scene.

4.5.1 Papadopoulo-Faugeras' constraints

Papadopoulo and Faugeras were perhaps the first to propose two sets of equations which fully characterize the trifocal tensor manifold. Both sets are derived from the correlation slices and contain constraints which are neither minimal (≥ 12) nor independent. All constraints hold under the general viewpoint assumption that the three camera centres are not aligned.

It is known from Proposition 9 that the column vectors of the trifocal matrices are three collinear points on the lines $\boldsymbol{\mathcal{L}}'_{1i}$. This collinearity property implies that the columns of each correlation matrix must be linearly dependent so the determinant of matrices $\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2,$ and $\boldsymbol{\mathcal{I}}_3$ must vanish.

Proposition 17 *The trifocal tensor $\mathcal{T}$ satisfies three constraints of degree 3, called the trifocal* ***rank constraints***

$$\det(\boldsymbol{\mathcal{I}}_i) = 0 \qquad i = 1, 2, 3. \tag{4.68}$$

These conditions on the tensor components are generic, that is, independent of the coordinate systems in the images. In other words, constraints (4.68) remain valid if the trifocal matrices are multiplied by non-zero scalars λ_i for $i = 1, 2, 3$. Because the trifocal matrices are 3×3 matrices with vanishing determinants, they must have rank at most equal to 2. It turns out that the sum of the trifocal matrices is also a rank deficient matrix [28, 69]. A generalisation of the rank constraints is stated as follows.

Proposition 18 *The trifocal tensor $\mathcal{T}$ satisfies the ten* ***extended rank constraints***

$$\operatorname{rank}\left(\sum_{i=1}^{3} \lambda_i \boldsymbol{\mathcal{I}}_i\right) \leq 2 \quad \forall \lambda_i \neq 0, i = 1, 2, 3. \tag{4.69}$$

These constraints are equivalent to $\det\left(\sum_{i=1}^{3} \lambda_i \boldsymbol{\mathcal{I}}_i\right) = 0$. To see that (4.69) enforces ten conditions on the tensorial elements, we have to expand the previous determinant with respect to the three unknowns λ_i. The resulting equations are polynomial constraints of order three in the unknowns λ_i. In the determinant expansion, the coefficients of the cubic terms $\lambda_1^3, \lambda_2^3,$ and λ_3^3 correspond to the determinants $\det(\boldsymbol{\mathcal{I}}_1), \det(\boldsymbol{\mathcal{I}}_2),$ and $\det(\boldsymbol{\mathcal{I}}_3)$ respectively. So the extended rank constraints contain the rank constraints.

An interesting result which follows from Proposition 8 is that the 3×3 matrices $[\boldsymbol{\mathcal{L}}'_{11}, \boldsymbol{\mathcal{L}}'_{12}, \boldsymbol{\mathcal{L}}'_{13}]$ and $[\boldsymbol{\mathcal{L}}''_{11}, \boldsymbol{\mathcal{L}}''_{12}, \boldsymbol{\mathcal{L}}''_{13}]$ have rank exactly equal to 2 because they span a one-dimensional null-space given by the epipoles $\mathbf{e}'$ and $\mathbf{e}''$ respectively. This simple observation allows to identify another two internal constraints on the trifocal tensor.

Proposition 19 *The trifocal tensor $\mathcal{T}$ satisfies two constraints of degree 6, called trifocal **epipolar constraints**, given by*

$$\det([\boldsymbol{\mathcal{L}}'_{11}, \boldsymbol{\mathcal{L}}'_{12}, \boldsymbol{\mathcal{L}}'_{13}]) = 0, \tag{4.70}$$

and

$$\det([\boldsymbol{\mathcal{L}}''_{11}, \boldsymbol{\mathcal{L}}''_{12}, \boldsymbol{\mathcal{L}}''_{13}]) = 0. \tag{4.71}$$

Constraint (4.71) on epipolar lines $\boldsymbol{\mathcal{L}}''_{1i}$ was also proposed by Heyden in the context of multiple view tensors [44–46]. His approach was very much algebraic and relied on dependencies of determinants characterising the tensorial components, formally known as the *quadratic p-relations* [47].

We now define the first set of algebraic constraints that fully characterises a trifocal tensor.

Theorem 1 *A bilinear mapping $\mathcal{T}$ from $\mathbb{P}^{2*} \times \mathbb{P}^{2*}$ to $\mathbb{P}^{2*}$ is a genuine trifocal tensor of the form* (4.41) *if and only if $\mathcal{T}$ satisfies the following twelve dependent constraints: the ten extended rank constraints* (4.69) *and the two epipolar constraints* (4.70) *and* (4.71)*.*

Other intrinsic properties of the trifocal tensor may be deduced by considering the 3-D incidence of principal planes in two images and the projection of their intersection in a third view. Suppose that $\mathbf{e}_j = \mathbf{e}_{\alpha_j}$ and $\mathbf{e}_k = \mathbf{e}_{\beta_k}$ represent canonical lines in the second and third views with $j = \alpha_j, k = \beta_k$, and $j, k = 1, 2, 3$. Let $\mathbf{L}_{jk}$ be the 3-D line of intersection of planes $\boldsymbol{\Gamma}'_j$ and $\boldsymbol{\Gamma}''_k$, and $\boldsymbol{l}_{jk} \simeq \mathcal{T}(\mathbf{e}_j, \mathbf{e}_k)$ its image line in the first view. Now, consider the four lines $\boldsymbol{l}_{\alpha_1\alpha_2} \simeq \mathcal{T}(\mathbf{e}_{\alpha_1}, \mathbf{e}_{\alpha_2}), \boldsymbol{l}_{\beta_1\alpha_2} \simeq \mathcal{T}(\mathbf{e}_{\beta_1}, \mathbf{e}_{\alpha_2}), \boldsymbol{l}_{\alpha_1\beta_2} \simeq \mathcal{T}(\mathbf{e}_{\alpha_1}, \mathbf{e}_{\beta_2})$, and $\boldsymbol{l}_{\beta_1\beta_2} \simeq \mathcal{T}(\mathbf{e}_{\beta_1}, \mathbf{e}_{\beta_2})$, such that the pairs of indices (α_1, α_2) and (β_1, β_2) are different and $\alpha_1 \neq \beta_1, \alpha_2 \neq \beta_2$, otherwise identical lines are produced. Clearly, the corresponding world lines are $\mathbf{L}_{\alpha_1\alpha_2} \simeq \boldsymbol{\Gamma}'_{\alpha_1} \triangle \boldsymbol{\Gamma}''_{\alpha_2}, \mathbf{L}_{\beta_1\alpha_2} \simeq \boldsymbol{\Gamma}'_{\beta_1} \triangle \boldsymbol{\Gamma}''_{\alpha_2}, \mathbf{L}_{\alpha_1\beta_2} \simeq \boldsymbol{\Gamma}'_{\alpha_1} \triangle \boldsymbol{\Gamma}''_{\beta_2}$, and $\mathbf{L}_{\beta_1\beta_2} \simeq \boldsymbol{\Gamma}'_{\beta_1} \triangle \boldsymbol{\Gamma}''_{\beta_2}$. An example was shown in Figure 4.18 of Section 4.4.2. From Table 4.6 the image lines can be expressed in terms of the rows of matrices $\boldsymbol{\mathcal{J}}_j$ as $\boldsymbol{l}_{jk} \simeq {}^j\mathbf{c}_k \simeq \boldsymbol{\mathcal{J}}_j^\mathsf{T}\mathbf{e}_k$. Furthermore, using definitions (4.31) and (4.33), we have that $\boldsymbol{l}_{jk} \simeq \left[\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk}\right]^\mathsf{T}$. There are nine possible tuples of such four lines obtained by selecting different basis vectors $\mathbf{e}_j$ and $\mathbf{e}_k$. Each 4-line tuple satisfies some algebraic constraints detailed in the next proposition.

Proposition 20 *The trifocal tensor $\mathcal{T}$ satisfies nine **vertical constraints** of degree 6 given by*

$$\begin{aligned} &\det([\boldsymbol{l}_{\alpha_1\alpha_2}, \boldsymbol{l}_{\alpha_1\beta_2}, \boldsymbol{l}_{\beta_1\beta_2}])\ \det([\boldsymbol{l}_{\alpha_1\alpha_2}, \boldsymbol{l}_{\beta_1\alpha_2}, \boldsymbol{l}_{\beta_1\beta_2}]) \\ &\quad - \det([\boldsymbol{l}_{\beta_1\alpha_2}, \boldsymbol{l}_{\alpha_1\beta_2}, \boldsymbol{l}_{\beta_1\beta_2}])\ \det([\boldsymbol{l}_{\alpha_1\alpha_2}, \boldsymbol{l}_{\beta_1\alpha_2}, \boldsymbol{l}_{\alpha_1\beta_2}]) = 0. \end{aligned} \tag{4.72}$$

A proof of this proposition can be found in [69]. The vertical constraints express the fact that a line joining two corresponding points envelops a degenerate conic whose determinant is given in (4.72). The advantage of showing the representation with principal planes is that it reveals the connection between image features and the 3-D primitive they originate from. Enforcing conditions (4.72) on the image lines constrains the intersection of the principal planes in space and ultimately the form of the projection matrices. A second set of algebraic constraints can now be described. This will conclude the overview of Papadopoulo-Faugeras' constraints.

Theorem 2 *A bilinear mapping $\mathcal{T}$ from $\mathbb{P}^{2*} \times \mathbb{P}^{2*}$ to $\mathbb{P}^{2*}$ is a genuine trifocal tensor of the form* (4.41) *if and only if $\mathcal{T}$ satisfies the following fourteen dependent constraints: the three rank constraints* (4.68)*, the two epipolar constraints* (4.70), (4.71)*, and the nine vertical constraints* (4.72)*.*

4.5.2 Canterakis' constraints

Canterakis was the first to propose a minimal set of eight constraints to describe the intrinsic relationships between the 27 coefficients of the trifocal tensor [12]. His constraints rely upon the properties of homography slices $\boldsymbol{\mathcal{J}}_j$ presented in Proposition 16 and revisited next.

Recall that the matrix $(\boldsymbol{\mathcal{J}}_p - \kappa_i \boldsymbol{\mathcal{J}}_q)$ has a nontrivial null-space for any of the two generalised eigenvalues κ_i. It follows that $\det(\boldsymbol{\mathcal{J}}_p - \kappa_i \boldsymbol{\mathcal{J}}_q) = 0$ so the generalised eigenvalues κ_i can be seen as the roots of a cubic polynomial given by $\det(\boldsymbol{\mathcal{J}}_p - \kappa \boldsymbol{\mathcal{J}}_q)$. For reasons that will be explained shortly, we may only consider two out of three polynomials which can be generated by selecting different pairs of homographies $\boldsymbol{\mathcal{J}}_p$ and $\boldsymbol{\mathcal{J}}_q$. Consequently, the requirements of Proposition 16 on the trifocal coefficients are essentially that

1. The cubic polynomial $\det(\boldsymbol{\mathcal{J}}_2 - \kappa \boldsymbol{\mathcal{J}}_1)$ has a single root κ_1 and a double root κ_2 with matrix $(\boldsymbol{\mathcal{J}}_2 - \kappa_2 \boldsymbol{\mathcal{J}}_1)$ having rank one.

2. The cubic polynomial $\det(\boldsymbol{\mathcal{J}}_3 - \kappa \boldsymbol{\mathcal{J}}_1)$ has a single root $\bar{\kappa}_1$ and a double root $\bar{\kappa}_2$ with matrix $(\boldsymbol{\mathcal{J}}_3 - \bar{\kappa}_2 \boldsymbol{\mathcal{J}}_1)$ having rank one.

3. The general eigenvectors of the single roots κ_1 and $\bar{\kappa}_1$ are the same (equal to $\mathbf{e}_{13}$) modulo a scalar.

We now show how these conditions can be expressed algebraically. A first group of constraints is obtained by defining a third degree polynomial which possesses a double root. Representing the polynomial $\det(\boldsymbol{\mathcal{J}}_2 - \kappa \boldsymbol{\mathcal{J}}_1)$ as $p(\kappa) = a\kappa^3 + b\kappa^2 + c\kappa + d$, where a, b, c, d are scalars made

from the trifocal coefficients of matrices $\mathcal{J}_1$ and $\mathcal{J}_2$, the requirement that $p(\kappa)$ has a double root is satisfied provided

$$B^2 - 4AC = 0, \tag{4.73}$$

where $A = b^2 - 3ac, B = bc - 9ad, C = c^2 - 3bd$. When condition (4.73) holds, the single root κ_1 and the double root κ_2 are given by

$$\kappa_1 = \frac{B}{A} - \frac{b}{a} \quad \text{and} \quad \kappa_2 = -\frac{B}{2A}.$$

Two constraints similar to the one given in (4.73) can be derived by demanding a double root for the polynomials $\det(\mathcal{J}_3 - \kappa\mathcal{J}_1)$ and $\det(\mathcal{J}_3 - \kappa\mathcal{J}_2)$. Thanks to Proposition 16, the two-dimensional eigenspaces of the matrices $(\mathcal{J}_p - \kappa\mathcal{J}_q)$ are readily recognised as the epipolar lines $\boldsymbol{\mathcal{L}}_{21}, \boldsymbol{\mathcal{L}}_{22}$, and $\boldsymbol{\mathcal{L}}_{23}$. Geometrically, all three lines intersect at the epipole $\mathbf{e}_{12}$ but only two of these lines are truly needed to identify the location of this point. So, at best, we can only obtain two independent (internal) constraints of the trifocal tensor by deriving (4.73) for any two of the three matrices $(\mathcal{J}_p - \kappa\mathcal{J}_q)$. Using all three matrices would generate a set of dependent constraints.

Another essential requirement is that the matrix $(\mathcal{J}_2 - \kappa_2\mathcal{J}_1)$ has rank one. When this condition is satisfied, we know from Proposition 16 that the eigenspace of this matrix is represented by the epipolar line $\boldsymbol{\mathcal{L}}_{23}$. The generalised eigenvalue problem $(\mathcal{J}_p - \kappa\mathcal{J}_q)\mathbf{x} = \mathbf{0}$ then reduces to (4.66) and becomes

$$\mathcal{J}_2 - \kappa_2\mathcal{J}_1 = \mathbf{e}_{31}\boldsymbol{\mathcal{L}}_{23}^{\mathsf{T}}.$$

One may see from this relation that any point $\mathbf{x}$ which does not lie on the line $\boldsymbol{\mathcal{L}}_{23}$ is mapped to the epipole $\mathbf{e}_{31}$ by the matrix $(\mathcal{J}_2 - \kappa_2\mathcal{J}_1)$, otherwise $\mathbf{e}_{31}\boldsymbol{\mathcal{L}}_{23}^{\mathsf{T}}\mathbf{x} = \mathbf{0}$. One such point could be $\mathbf{x} = \boldsymbol{\mathcal{L}}_{23}^{*}$ (the dual of $\boldsymbol{\mathcal{L}}_{23}$) because it does not lie on $\boldsymbol{\mathcal{L}}_{23}$. Now, from the homography-epipole relation (4.60), we also know that $\mathbf{e}_{13}$ is sent to $\mathbf{e}_{31}$ by any regular matrix $\mathcal{J}_j$. Therefore, altogether, the rank-1 condition can be imposed by demanding

$$(\mathcal{J}_2 - \kappa_2\mathcal{J}_1)\boldsymbol{\mathcal{L}}_{23} \simeq \mathcal{J}_1\mathbf{e}_{13}.$$

These equations may be expressed in terms of a vector cross product as

$$(\mathcal{J}_2 - \kappa_2\mathcal{J}_1)\boldsymbol{\mathcal{L}}_{23} \times \mathcal{J}_1\mathbf{e}_{13} = \mathbf{0}, \tag{4.74}$$

which provides two independent constraints on the matrices $\mathcal{J}_1$ and $\mathcal{J}_2$. Considering the matrix $(\mathcal{J}_3 - \bar{\kappa}_2\mathcal{J}_1)$, another two constraints can be found in the same way using

$$(\mathcal{J}_3 - \bar{\kappa}_2\mathcal{J}_1)\boldsymbol{\mathcal{L}}_{22} \times \mathcal{J}_1\mathbf{e}_{13} = \mathbf{0}. \tag{4.75}$$

Finally, Proposition 16 also requires equality (up to scale) for the two one-dimensional eigenspaces corresponding to the single roots κ_1 and $\bar{\kappa}_1$. Again, using the homography-epipole relation (4.60), this condition may be enforced by the equations

$$\boldsymbol{\mathcal{J}}_1\mathbf{e}_{13} \times \boldsymbol{\mathcal{J}}_2\mathbf{e}_{13} = \mathbf{0}, \tag{4.76}$$

which yield two independent constraints. Any pair of matrices $\boldsymbol{\mathcal{J}}_j$ could be used to derive such relations. The constraints proposed by Canterakis can now be summarised.

Theorem 3 *A bilinear mapping $\mathcal{T}$ from $\mathbb{P}^{2*} \times \mathbb{P}^{2*}$ to $\mathbb{P}^{2*}$ is a genuine trifocal tensor of the form* (4.41) *if and only if $\mathcal{T}$ satisfies the following eight independent constraints:*

- *Two constraints of the form* (4.73) *by demanding a double root for the polynomials* $\det(\boldsymbol{\mathcal{J}}_2 - \kappa\boldsymbol{\mathcal{J}}_1)$ *and* $\det(\boldsymbol{\mathcal{J}}_3 - \kappa\boldsymbol{\mathcal{J}}_1)$;
- *A total of four constraints from* (4.74) *and* (4.75) *to compel matrices* $(\boldsymbol{\mathcal{J}}_2 - \kappa_2\boldsymbol{\mathcal{J}}_1)$ *and* $(\boldsymbol{\mathcal{J}}_3 - \bar{\kappa}_2\boldsymbol{\mathcal{J}}_1)$ *to have rank one;*
- *Two constraints from* (4.76) *to compel matrices* $(\boldsymbol{\mathcal{J}}_2 - \kappa_1\boldsymbol{\mathcal{J}}_1)$ *and* $(\boldsymbol{\mathcal{J}}_3 - \bar{\kappa_1}\boldsymbol{\mathcal{J}}_1)$ *to have the same eigenspace.*

Since the second and third images play analogous roles with respect to the trifocal tensor, similar constraints can be derived from the homography slices $\boldsymbol{\mathcal{K}}_k$.

4.5.3 Ressl's constraints

Substantial work has also been done by Ressl [73] to formulate a minimal set of ancillary constraints on the trifocal tensor. Similar to the elaboration of equation (4.72), his set is derived by drawing on the properties of 3-D points and lines that arise as the meet of principal rays and planes from the images. As seen in Section 4.4, these canonical relations are encapsulated by the rows and columns of the tensorial slices. Despite considering the correlation slices, much emphasis is dedicated to the homography slices, yet in a different way than Canterakis'. The final constraints impose that the principal rays of one image project into concurrent lines in the other two images. These constraints form a minimal set and are independent.

One inconvenience however is that Ressl's constraints apply to a different tensor than the "standard" ones proposed by Hartley [37] or Faugeras [24]. A significant part of the work in Section 4.4 was realised with the aim to express these constraints in the more familiar framework proposed by Hartley, which is the one chosen in this thesis for trifocal tensor estimation.

Three of the eight constraints are already known and consist of the rank constraints given in (4.68). The second constraint is not entirely new and corresponds to a reformulation of epipolar constraint (4.70) in terms of the correlation slices. A new expression for constraint (4.71) is provided here as a byproduct. The remaining four conditions are founded on properties of triangles $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ defined in Section 4.4.2. These conditions are also a reformulation of the original constraints suggested by Ressl. We begin by revisiting the epipolar constraints.

Epipolar constraints in the second and third views

The inconvenience with epipolar constraints (4.70) and (4.71) comes from the difficulty to write them explicitly in terms of the tensorial components.

Considering the first epipolar constraint, we know from Proposition 8 that the 3×3 matrix $[\boldsymbol{\mathcal{L}}'_{11}, \boldsymbol{\mathcal{L}}'_{12}, \boldsymbol{\mathcal{L}}'_{13}]$ has a one-dimensional null-space, the epipole $\mathbf{e}'$. In light of Proposition 9, the epipolar line $\boldsymbol{\mathcal{L}}'_{1i}$ may be computed from any two column vectors of trifocal matrix $\boldsymbol{\mathcal{I}}_i$. Therefore, constraint $\det([\boldsymbol{\mathcal{L}}'_{11}, \boldsymbol{\mathcal{L}}'_{12}, \boldsymbol{\mathcal{L}}'_{13}]) = 0$ may be reformulated by selecting, for instance, the first two columns of matrices $\boldsymbol{\mathcal{I}}_i$. Using convention (4.25), the new form of this constraint is

$$\det([\mathbf{a}'_1 \times \mathbf{a}'_2,\ \mathbf{b}'_1 \times \mathbf{b}'_2,\ \mathbf{c}'_1 \times \mathbf{c}'_2]) = 0. \tag{4.77}$$

A similar reasoning can be applied to epipolar lines $\boldsymbol{\mathcal{L}}''_{1i}$. We deduce from Proposition 9 that each line $\boldsymbol{\mathcal{L}}''_{1i}$ may be obtained as the cross product of any two row vectors of the trifocal matrices. Choosing the first two rows of these matrices and following convention (4.26), the constraint $\det([\boldsymbol{\mathcal{L}}''_{11}, \boldsymbol{\mathcal{L}}''_{12}, \boldsymbol{\mathcal{L}}''_{13}]) = 0$ becomes

$$\det([\mathbf{d}''_1 \times \mathbf{d}''_2,\ \mathbf{e}''_1 \times \mathbf{e}''_2,\ \mathbf{f}''_1 \times \mathbf{f}''_2]) = 0. \tag{4.78}$$

Collinearity constraints

The next series of constraints develops from Proposition 15 and expresses the fact that vertices $\mathbf{v}_{1k}, \mathbf{v}_{2k}$, and $\mathbf{v}_{3k}$ of triangles $\mathbf{T}_1, \mathbf{T}_2$, and $\mathbf{T}_3$ lie on epipolar line $\boldsymbol{\mathcal{L}}_{3k}$ for $k = 1, 2, 3$, as illustrated in Figure 4.19(a). The collinearity property of these vertices means that the 3×3 matrix formed by these points has rank two and therefore may be expressed as a determinant constraint in the style of equation (4.78) but involving lines $\boldsymbol{\mathcal{L}}_{3k}$. According to Proposition 14, the triangle vertices can be computed in terms of the rows of matrices $\boldsymbol{\mathcal{J}}_j$. Using Table 4.6, the collinearity of vertices $\mathbf{v}_{11}, \mathbf{v}_{21}$, and $\mathbf{v}_{31}$ on line $\boldsymbol{\mathcal{L}}_{31}$ implies the constraint

$$\det([\mathbf{r}_2 \times \mathbf{r}_3,\ \mathbf{s}_2 \times \mathbf{s}_3,\ \mathbf{t}_2 \times \mathbf{t}_3]) = 0. \tag{4.79}$$

Similarly, vertices $\mathbf{v}_{12}, \mathbf{v}_{22}$, and $\mathbf{v}_{32}$ on line $\boldsymbol{\mathcal{L}}_{32}$ give the constraint

$$\det([\mathbf{r}_3 \times \mathbf{r}_1,\ \mathbf{s}_3 \times \mathbf{s}_1,\ \mathbf{t}_3 \times \mathbf{t}_1]) = 0, \tag{4.80}$$

and vertices $\mathbf{v}_{13}, \mathbf{v}_{23}$, and $\mathbf{v}_{33}$ on line $\boldsymbol{\mathcal{L}}_{33}$ mean

$$\det([\mathbf{r}_1 \times \mathbf{r}_2,\ \mathbf{s}_1 \times \mathbf{s}_2,\ \mathbf{t}_1 \times \mathbf{t}_2]) = 0. \tag{4.81}$$

Papadopoulo *et al.* already discovered the role of the vertices entering the above constraints when they derived the vertical constraints in (4.72). However, the authors miss to evolve any conditions from their collinearity property. The new constraints are stated in the following proposition.

Proposition 21 *Trifocal tensor $\mathcal{T}$ satisfies three constraints of degree 6, called trifocal **collinearity constraints**, given by the equations* (4.79), (4.80), *and* (4.81).

Epipolar constraints in the first view

If the essence of epipolar constraint (4.70) (resp. (4.71)) is that epipolar lines $\boldsymbol{\mathcal{L}}'_{1i}$ (resp. $\boldsymbol{\mathcal{L}}''_{1i}$) go through the epipole $\mathbf{e}'$ (resp. $\mathbf{e}''$), then the essence of epipolar lines $\boldsymbol{\mathcal{L}}_{3k}$ is to go through the epipole $\mathbf{e}_{13}$. This latter remark implies that

$$\det([\boldsymbol{\mathcal{L}}_{31}, \boldsymbol{\mathcal{L}}_{32}, \boldsymbol{\mathcal{L}}_{33}]) = 0. \tag{4.82}$$

From Proposition 15, we know that epipolar lines $\boldsymbol{\mathcal{L}}_{3k}$ can be computed from any two of the three vertices $\mathbf{v}_{1k}, \mathbf{v}_{2k}$, and $\mathbf{v}_{3k}$, $k = 1, 2, 3$. Moreover, these vertices can themselves be computed from the rows of matrices $\boldsymbol{\mathcal{J}}_j$ as claimed in Proposition 14. Therefore, we can write

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_{31} &\simeq \mathbf{v}_{11} \times \mathbf{v}_{21} \simeq (\mathbf{r}_2 \times \mathbf{r}_3) \times (\mathbf{s}_2 \times \mathbf{s}_3), \\
\boldsymbol{\mathcal{L}}_{32} &\simeq \mathbf{v}_{12} \times \mathbf{v}_{22} \simeq (\mathbf{r}_3 \times \mathbf{r}_1) \times (\mathbf{s}_3 \times \mathbf{s}_1), \\
\boldsymbol{\mathcal{L}}_{33} &\simeq \mathbf{v}_{13} \times \mathbf{v}_{23} \simeq (\mathbf{r}_1 \times \mathbf{r}_2) \times (\mathbf{s}_1 \times \mathbf{s}_2).
\end{aligned}$$

Altogether, it follows that epipolar constraint (4.82) is given by

$$\det([\mathbf{r}_{23} \times \mathbf{s}_{23},\ \mathbf{r}_{31} \times \mathbf{s}_{31},\ \mathbf{r}_{12} \times \mathbf{s}_{12}]) = 0, \tag{4.83}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i \times \mathbf{r}_j$ and $\mathbf{s}_{ij} = \mathbf{s}_i \times \mathbf{s}_j$. Here, we have chosen the first two vertices on each line $\boldsymbol{\mathcal{L}}_{3k}$ (using the rows of $\boldsymbol{\mathcal{J}}_1$ and $\boldsymbol{\mathcal{J}}_2$), but in general this constraint can be set up with any two vertices on these lines (using any two of the three matrices $\boldsymbol{\mathcal{J}}_j$). This constraint is summarised in the next proposition.

Proposition 22 *Trifocal tensor $\mathcal{T}$ satisfies an **epipolar constraint** of degree 12 given by the equation* (4.83).

The conditions on the tensor proposed by Ressl can be summarised in the following theorem. This will conclude our presentation of the internal constraints on the trifocal tensor.

Theorem 4 *A bilinear mapping $\mathcal{T}$ from $\mathbb{P}^{2*} \times \mathbb{P}^{2*}$ to $\mathbb{P}^{2*}$ is a genuine trifocal tensor of the form* (4.41) *if and only if $\mathcal{T}$ satisfies the following eight independent constraints: the three rank constraints* (4.68)*, the epipolar constraint* (4.77)*, the three collinearity constraints* (4.79) *to* (4.81)*, and another epipolar constraint of the form* (4.83)*.*

It is possible to find sets with less than 8 constraints, however, these sets are derived by direct computation of the projections from a minimum of six point correspondences [72, 76]. These constraints are useful in methods such as RANSAC to eliminate outliers in the data by iteratively testing the quality of tensors computed from sets of six points at a time. This type of algebraic constraints were not considered here because they do not give any indication on how to correct an unconstrained tensor.

4.6 Recovering 3-D information from the trifocal tensor

One of many objectives behind computing the trifocal tensor is to recover the projectivity of the scene and the camera centres. We have seen in equation (4.16) that the trifocal tensor may be computed from three camera projection matrices. We now show the converse, that projection matrices may be computed from the tensor up to projective equivalence. Knowledge of the projections will then allow the relative camera positions to be worked out. We begin by showing how the epipoles and fundamental matrices can be obtained from the trifocal tensor.

4.6.1 Retrieving the epipoles

Suppose that we have a trifocal tensor $\boldsymbol{\theta}$ as given in (4.16). Its corresponding trifocal matrices $\boldsymbol{\mathcal{I}}_i$ are readily obtained from expression (4.24). Let $\mathbf{w}_i$ be the unit vector that minimises $\|\boldsymbol{\mathcal{I}}_i \mathbf{w}_i\|$, that is, $\mathbf{w}_i$ is the eigenvector corresponding to the third column of $\mathtt{V}_i$ in the SVD decomposition of $\boldsymbol{\mathcal{I}}_i = \mathtt{U}_i \mathtt{D}_i \mathtt{V}_i^\mathsf{T}$. Let $\mathtt{W}$ be the matrix with i-th row made of $\mathbf{w}_i^\mathsf{T}$. The epipole $\mathbf{e}''$ is the unit vector that minimises $\|\mathtt{W}\mathbf{e}''\|$, that is, $\mathbf{e}''$ is the eigenvector corresponding to the third column of $\mathtt{V}'$ in the SVD decomposition of $\mathtt{W} = \mathtt{U}'\mathtt{D}'\mathtt{V}'^\mathsf{T}$ [37]. The epipole $\mathbf{e}'$ can be computed in a similar manner, starting from $\boldsymbol{\mathcal{I}}_i^\mathsf{T}$ in place of $\boldsymbol{\mathcal{I}}_i$. This method works well in the case of general motion. If the camera displacement is degenerate, the trifocal matrices have rank less than 2, which means that the epipoles must be estimated from a more robust method as outlined in [73, Chap. 7].

4.6.2 Retrieving the fundamental matrices

With knowledge of the epipoles $\mathbf{e}'$ and $\mathbf{e}''$, it is an easy task to work out the fundamental matrices $\mathtt{F}_{21}$ and $\mathtt{F}_{31}$ from the trifocal tensor.

Recalling formula (4.48), a line $\boldsymbol{l}''$ in the third view back-projects to a plane in space which induces a homography such that $\mathbf{m}' \simeq [\boldsymbol{\mathcal{I}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_3\boldsymbol{l}'']\mathbf{m}$. Substituting this result in (4.6) gives

$$\boldsymbol{l}' \simeq [\mathbf{e}']_\times[\boldsymbol{\mathcal{I}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_3\boldsymbol{l}'']\mathbf{m}$$

and now comparison with (4.7) shows that

$$\mathtt{F}_{21} \simeq [\mathbf{e}']_\times[\boldsymbol{\mathcal{I}}_1\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_2\boldsymbol{l}'', \boldsymbol{\mathcal{I}}_3\boldsymbol{l}''].$$

Since $\boldsymbol{l}''$ is arbitrary, Hartley recommands to use $\boldsymbol{l}'' = \mathbf{e}''$ to avoid a critical situation where $\boldsymbol{l}''$ lies in the null-space of any of the trifocal matrices. Therefore,

$$\mathtt{F}_{21} \simeq [\mathbf{e}']_\times[\boldsymbol{\mathcal{I}}_1\mathbf{e}'', \boldsymbol{\mathcal{I}}_2\mathbf{e}'', \boldsymbol{\mathcal{I}}_3\mathbf{e}'']. \tag{4.84}$$

A similar proof can be evolved for $\mathtt{F}_{31}$ using homography h_i^k in (4.44). Setting $\boldsymbol{l}' = \mathbf{e}'$ produces

$$\mathtt{F}_{31} \simeq [\mathbf{e}'']_\times[\boldsymbol{\mathcal{I}}_1^\mathsf{T}\mathbf{e}', \boldsymbol{\mathcal{I}}_2^\mathsf{T}\mathbf{e}', \boldsymbol{\mathcal{I}}_3^\mathsf{T}\mathbf{e}']. \tag{4.85}$$

Fundamental matrix $\mathtt{F}_{32}$ between images Ψ_2 and Ψ_3 is not as straightforward to compute but a couple of methods exist. One technique which utilises the formalism of Grassmann-Cayley algebra [27] allows to express and recover $\mathtt{F}_{32}$ in terms of the trifocal tensor [26]. A second method proposes to compute this matrix linearly from six matching points across three views given fundamental matrix $\mathtt{F}_{21}$ and tensorial coefficients $\mathcal{T}_i^{jk}$ [4]. Perhaps an easier method than the previous two is to compute homography $\mathtt{H}_{32}$ and epipole $\mathbf{e}_{32}$ such that $\mathtt{F}_{32} \simeq \mathtt{H}_{32}^\mathsf{T}[\mathbf{e}_{32}]_\times$. The interested reader is conferred to [73, Chap. 7] for more details.

4.6.3 Retrieving the projections and camera centres

Because projections are defined only up to a projective 3-D transformation, the first projection may be chosen as $\mathtt{P} = [\mathtt{I}_{3\times3} \mid \mathbf{0}]$. Applying (4.13) in Proposition 3 to matrix $\mathtt{F}_{21}$ in (4.84) gives the second projection as

$$\mathtt{P}' = [[\boldsymbol{\mathcal{I}}_1\mathbf{e}'', \boldsymbol{\mathcal{I}}_2\mathbf{e}'', \boldsymbol{\mathcal{I}}_3\mathbf{e}''] \mid \mathbf{e}'].$$

Fixing the first and second projections in this manner defines a specific *projective frame*. The third projection cannot be derived from (4.85) directly because the final triplet of cameras $\mathtt{P}$, $\mathtt{P}'$, and $\mathtt{P}''$ is not expressed in the same world coordinate system and therefore is *inconsistent*.

Setting the scale of both epipoles $\mathbf{e}'$ and $\mathbf{e}''$ so that they have unit norm, Hartley showed that the third projection must be taken as

$$\mathtt{P}'' = [(\mathbf{e}''\mathbf{e}''^{\mathsf{T}} - \mathtt{I}_{3\times 3})[\boldsymbol{\mathcal{I}}_1^{\mathsf{T}}\mathbf{e}', \boldsymbol{\mathcal{I}}_2^{\mathsf{T}}\mathbf{e}', \boldsymbol{\mathcal{I}}_3^{\mathsf{T}}\mathbf{e}'] \mid \mathbf{e}'']$$

to ensure projective consistency between all three projections. This is summarised next.

Proposition 23 *When the epipoles $\mathbf{e}'$ and $\mathbf{e}''$ are normalised to unit norm, 3-D information can be retrieved from the trifocal tensor in the form of a set of three consistent projections*

$$\begin{aligned}
\mathtt{P} &= [\mathtt{I}_{3\times 3} \mid \mathbf{0}], \\
\mathtt{P}' &= [[\boldsymbol{\mathcal{I}}_1\mathbf{e}'', \boldsymbol{\mathcal{I}}_2\mathbf{e}'', \boldsymbol{\mathcal{I}}_3\mathbf{e}''] \mid \mathbf{e}'], \\
\mathtt{P}'' &= [(\mathbf{e}''\mathbf{e}''^{\mathsf{T}} - \mathtt{I}_{3\times 3})[\boldsymbol{\mathcal{I}}_1^{\mathsf{T}}\mathbf{e}', \boldsymbol{\mathcal{I}}_2^{\mathsf{T}}\mathbf{e}', \boldsymbol{\mathcal{I}}_3^{\mathsf{T}}\mathbf{e}'] \mid \mathbf{e}''].
\end{aligned}$$

Given the special form of matrix $\mathtt{P}$, the first camera centre is located at $(0, 0, 0)^{\mathsf{T}}$. As mentioned in Section 4.1.1, an optical centre is the (unique) point at the intersection of the principal planes of a projection matrix. So, centres $\tilde{\mathbf{C}}'$ and $\tilde{\mathbf{C}}''$ may be computed as the null-spaces of the second and third projections. More details can be found in [37, Chap. 5].

4.7 Conclusion

The description of three-view geometry can be approached from the standpoint of two-view geometry, with fundamental matrices as bindings between points of each image pairs. However, the restrictions of fundamental matrices to properly encompass the multilinear relations between three images have lead to the elaboration of a new object, the trifocal tensor. A valuable feature of this tensor is its ability to handle point as well as line correspondences across the views. If the essence of stereo vision is the epipolar constraint, then the essence of trinocular vision is the point-line-line trilinear constraint. In general, information about a point must first be converted into lines passing through that point before it can be handled by the trifocal tensor. The various incidence relations between lines and points over three views give different, but intimately related, trilinear expressions. These matching constraints tell whether features in different images could possibly be the projections of a single world primitive.

A fundamental result is that the geometric notion of transfer between views is captured by the algebraic contraction of the trifocal tensor. A single contraction of the tensor with canonical basis vectors gives rise to tensorial slices, which are groups of three 3×3 matrices representing either correlations or homographies between pairs of images. A contraction with an arbitrary

image point or line produces a matrix which is a linear combination of the slices in one group and so exhibits similar geometric properties than its canonical constituents. The consecutive contraction of the tensor by two feature matches across two views permits finding a corresponding feature in the third view. The trilinearities come as natural extensions of two-feature transfer equations, when the tensor is fully contracted by three corresponding image tokens.

Because the tensorial slices may be seen as canonical representatives of the trinocular geometry, they are ideal to characterise ancillary constraints that the trifocal tensor is subject to. Initial work by Hartley and Faugeras *et al.* focused on the correlation matrices and established several sets of dependent constraints. In recent years, Canterakis first, and Ressl later, utilised the homography matrices to derive minimal sets of eight independent constraints. This number of constraints matches the expected theoretical value. The various sets of constraints were examined in turn and expressed in one common framework.

The next chapter provides an implementation and testing of many results established here.

Chapter 5

Application II: Trifocal Tensor Estimation

This chapter is devoted to the estimation of the trifocal tensor from triplets of corresponding image points. The computation process is very sensitive to noise and outliers in the data. Only an estimate which satisfies the internal constraints is accurate enough to be usable in any application. It is highly probable that the initial approximation of the trifocal tensor, arising exclusively from solving the incidence relations between views, does not adhere to the ancillary constraints. A refinement of the tensor must be carried out as a post-process to enforce these conditions. The corrected estimate will then be consistent with the underlying trifocal geometry.

For this task, general estimation methods described in Chapter 2 are employed and combined with the trinocular constraints derived in Chapter 4. An additional non-iterative method is presented whereby a trifocal tensor is computed by imposing linear constraints. The FNS and RFNS algorithms will be the major tools to generate accurate unconstrained estimates. The trifocal tensors obtained are then corrected a posteriori using the schemes presented in Section 2.5. Various performance measures of the devised (constrained) estimators are evaluated through experiments on both simulated and real image data, and compared to that of other existing methods.

5.1 Point incidence

Throughout the chapter estimation of the trifocal tensor will be based on the case of a point incidence in three views. In general, there are two courses of action available. The estimation problem may be modelled by an objective function which includes all nine equations generated from each correspondence triplet or a minimum selection of four independent and orthonormal equations.

Including all nine equations improves the conditioning of the system [40] but increases the complexity of the solution. It was shown that the trilinear equations correspond to depth errors weighted by a function of the data [31]. For an uncalibrated image sequence, the weights vary substantially for different equations. Pre-normalising the data and selecting a minimum of four equations helps reducing the variation between weights. This is the approach we will follow.

A typical triplet of corresponding points is assumed to have the form $\mathbf{m} = [m^1, m^2, m^3]^\mathsf{T}$, $\mathbf{m}' = [m'^1, m'^2, 1]^\mathsf{T}$, and $\mathbf{m}'' = [m''^1, m''^2, 1]^\mathsf{T}$. The points are related through the trifocal tensor by the four trilinear constraints [79]:

$$\begin{aligned}
&\sum_{i=1}^{3}(m^i\mathcal{T}_i^{11} - m^i m'^1\mathcal{T}_i^{31} + m^i m'^1 m''^1\mathcal{T}_i^{33} - m^i m''^1\mathcal{T}_i^{13}) = 0,\\
&\sum_{i=1}^{3}(m^i\mathcal{T}_i^{12} - m^i m'^1\mathcal{T}_i^{32} + m^i m'^1 m''^2\mathcal{T}_i^{33} - m^i m''^2\mathcal{T}_i^{13}) = 0,\\
&\sum_{i=1}^{3}(m^i\mathcal{T}_i^{21} - m^i m'^2\mathcal{T}_i^{31} + m^i m'^2 m''^1\mathcal{T}_i^{33} - m^i m''^1\mathcal{T}_i^{23}) = 0,\\
&\sum_{i=1}^{3}(m^i\mathcal{T}_i^{22} - m^i m'^2\mathcal{T}_i^{32} + m^i m'^2 m''^2\mathcal{T}_i^{33} - m^i m''^2\mathcal{T}_i^{23}) = 0.
\end{aligned} \tag{5.1}$$

This system will be used as a basis for computation of the trifocal tensor. Letting $m^3 = 1$, the system can be brought into the form given in (2.2) by first concatenating the inhomogeneous coordinates of $\mathbf{m}$, $\mathbf{m}'$, and $\mathbf{m}''$ to obtain a single item of data $\mathbf{x} = [m^1, m^2, m'^1, m'^2, m''^1, m''^2]$, next rearranging the tensor entries into a length-27 vector $\boldsymbol{\theta}$, and then setting

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = [f_1(\mathbf{x}, \boldsymbol{\theta}), \ldots, f_4(\mathbf{x}, \boldsymbol{\theta})]^\mathsf{T}, \tag{5.2}$$

where $f_1, \ldots, f_4$ are the corresponding expressions on the left-hand side of the above system.

5.2 Hartley's method

This section will show that a trifocal tensor can be estimated by an equivalent procedure to that of the normalised eight-point algorithm for the essential matrix [36]. An unconstrained estimate is first generated using the NALS method and then corrected using *linear* constraints. In a final step, the algorithm requires transformation rules to express the computed tensor back into the original space of measurements. These transformations are stated in the next section before presenting the algorithm's details. We will refer to this method as *Hartley's method*, named so after its inventor.

5.2.1 Trifocal tensor normalisation

Suppose that triplet of corresponding points $\mathbf{m}_i, \mathbf{m}'_i$, and $\mathbf{m}''_i$ are normalised using transformations $\mathtt{T}$, $\mathtt{T}'$, and $\mathtt{T}''$ as in (3.6) such that $\tilde{\mathbf{m}}_i = \mathtt{T}\mathbf{m}_i$, $\tilde{\mathbf{m}}'_i = \mathtt{T}'\mathbf{m}'_i$, and $\tilde{\mathbf{m}}''_i = \mathtt{T}''\mathbf{m}''_i$. Denoting the original tensorial coefficients by $\mathcal{T}_i^{jk}$ for $i, j, k = 1, 2, 3$, the trifocal tensor relating the *normalised* data points is expressed as

$$\widetilde{\mathcal{T}}_r^{st} = (\mathtt{T}^{-1})_r^i \, \mathtt{T}_j'^s \, \mathtt{T}_k''^t \, \mathcal{T}_i^{jk} \qquad r, s, t = 1, 2, 3. \tag{5.3}$$

Here summations over the indices i, j, and k are implicit. A more compact (and perhaps convenient) way to perform this operation is by acting on the trifocal matrices directly. Expression (5.3) becomes

$$\widetilde{\mathcal{T}}_r = \mathtt{T}' \left(\sum_{i=1}^{3} (\mathtt{T}^{-1})_r^i \, \mathcal{T}_i \right) \mathtt{T}''^{\mathsf{T}} \qquad r = 1, 2, 3.$$

The original tensor can be recovered from its normalised version by applying the transformation

$$\mathcal{T}_i^{jk} = \mathtt{T}_i^r \, (\mathtt{T}'^{-1})_s^j \, (\mathtt{T}''^{-1})_t^k \, \widetilde{\mathcal{T}}_r^{st}, \tag{5.4}$$

or equivalently,

$$\mathcal{T}_i = \mathtt{T}'^{-1} \left(\sum_{r=1}^{3} \mathtt{T}_i^r \, \widetilde{\mathcal{T}}_r \right) \mathtt{T}''^{-\mathsf{T}} \qquad i = 1, 2, 3.$$

In general, trifocal tensors (5.3) and (5.4) do not have unit norm.

5.2.2 Normalised algebraic least-squares estimate

As in the case of homography or fundamental matrix computation, the accuracy of the ALS estimate can be greatly enhanced if image coordinates are normalised before the estimate is actually computed. The normalisation ensures that the entries of the design matrix $\mathtt{M}$ are of comparable size.

Suppose that the original 2-D homogeneous points $\mathbf{m}_i$, $\mathbf{m}'_i$, and $\mathbf{m}''_i$ of the left, central and right images respectively are converted to *normalised* 2-D homogeneous points $\tilde{\mathbf{m}}_i$, $\tilde{\mathbf{m}}'_i$, and $\tilde{\mathbf{m}}''_i$ as described in Section 5.2.1. Moreover, let $\tilde{\mathbf{x}}_i = [\tilde{u}_i, \tilde{v}_i, \tilde{u}'_i, \tilde{v}'_i, \tilde{u}''_i, \tilde{v}''_i]^{\mathsf{T}}$ be the result of concatenating the inhomogeneous coordinates of $\tilde{\mathbf{m}}_i$, $\tilde{\mathbf{m}}'_i$, and $\tilde{\mathbf{m}}''_i$. If $\widetilde{\mathcal{T}}_r^{st}$ designates the ALS trifocal tensor estimate based on the $\tilde{\mathbf{x}}_i$, then the corresponding *normalised algebraic least-squares* (NALS) estimate of $\boldsymbol{\theta}$, $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$, is defined by

$$\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}} = \mathcal{T}_i^{jk} = \mathtt{T}_i^r (\mathtt{T}'^{-1})_s^j (\mathtt{T}''^{-1})_t^k \widetilde{\mathcal{T}}_r^{st}. \tag{5.5}$$

5.2.3 Imposing linear constraints

The 27 entries of the tensor are defined up to a common scale so they may be computed given at least 26 equations involving points and lines as data input such that

$$2\ \#\text{lines} + 4\ \#\text{points} \geq 26.$$

Originally, solutions were computed from a set of 7 point matches or 13 line correspondences in three views. Hartley was first to propose a linear algorithm allowing for a mixture of both points and lines [38].

The NALS estimate obtained in the previous step does not satisfy the ancillary constraints for the trifocal tensor. This means that the projections recovered from this tensor would give a crude 3-D reconstruction of the scene. Imposing the constraints ensures that the rays back-projected in space from each triplet of image points intersect in a single 3-D point, as seen in Figure 4.14. To guarantee such a geometrically valid tensor, Hartley proposed a scheme which applies *linear* constraints on to the NALS estimate (5.5). This method is exposed next.

Having isolated epipoles $\mathbf{e}'$ and $\mathbf{e}''$ from $\widehat{\boldsymbol{\theta}}_{\text{NALS}}$ (see Section 4.6.1), one may solve the following quadratic minimisation problem with linear constraints

$$\widehat{\boldsymbol{\theta}}_{\text{HRT}} = \underset{\tilde{\boldsymbol{\beta}} \in \mathcal{V}}{\arg\min} \| \mathtt{M}\,\mathtt{E}\,\tilde{\boldsymbol{\beta}} \|^2 \tag{5.6}$$

$$\text{such that} \quad \mathcal{V} = \{\tilde{\boldsymbol{\beta}} \in \mathbb{R}^w \,|\, \mathtt{C}\tilde{\boldsymbol{\beta}} = \mathbf{0}\}, \tag{5.7}$$

where $\mathtt{M}$ is the design matrix given in (2.5), $\mathtt{E} = \mathtt{E}(\mathbf{e}', \mathbf{e}'')$ is the transformation (4.22) and $\tilde{\boldsymbol{\beta}}$ is the associated length-18 vector containing the inner 3×3 matrices $\mathtt{A}$ and $\mathtt{B}$ of the second and third projections as defined in (4.21).

The constraints $\mathtt{C}\tilde{\boldsymbol{\beta}} = \mathbf{0}$ in (5.7) represent the three conditions

$$\sum_{i=1}^{3} \mathbf{a}_j^i \mathbf{a}_4^i = \mathbf{0}, \tag{5.8}$$

which is the requirement that the 4-th column of $\mathtt{P}'$ is orthogonal to all other columns. It is preferable to enforce these constraints, otherwise performing minimisation (5.6) alone may be unstable due to the fact that $\mathtt{E}$ does not have full rank [35, 41]. The estimate resulting from this minimisation process will be referred to as the *Hartley* trifocal tensor estimate and denoted $\widehat{\boldsymbol{\theta}}_{\text{HRT}}$.

The SVD of $\mathtt{C} = \hat{\mathtt{U}}\hat{\mathtt{D}}\hat{\mathtt{V}}^{\mathsf{T}}$ enables to express $\tilde{\boldsymbol{\beta}}$ as a linear combination of the columns of $\hat{\mathtt{V}}$, that is, $\tilde{\boldsymbol{\beta}} = \hat{\mathtt{V}}\hat{\mathbf{y}}$. The diagonal matrix $\hat{\mathtt{D}}$ has the property that its r non-zero diagonal entries precede the zero ones; for constraints (5.8), $r = 3$. Consequently, $\mathtt{C}\tilde{\boldsymbol{\beta}} = \mathbf{0}$ may be written as $\hat{\mathtt{U}}\hat{\mathtt{D}}\hat{\mathbf{y}} = \mathbf{0}$,

which in turn is equivalent to $\hat{\mathtt{D}}\hat{\mathbf{y}} = \mathbf{0}$ since $\hat{\mathtt{U}}$ is orthogonal. This last constraint is satisfied only if $\hat{\mathbf{y}}$ has r leading zeros. In this case, we may write $\tilde{\boldsymbol{\beta}} = \hat{\mathtt{V}}\hat{\mathbf{y}} = \tilde{\mathtt{V}}\tilde{\mathbf{y}}$, where $\tilde{\mathbf{y}}$ is the vector created from $\hat{\mathbf{y}}$ by discarding the leading r zeros and $\tilde{\mathtt{V}}$ is formed from $\hat{\mathtt{V}}$ by omitting the first r columns. Since $\hat{\mathtt{V}}$ is orthogonal, the condition $\| \tilde{\mathtt{V}}\tilde{\mathbf{y}} \| = 1$ is equivalent to $\| \tilde{\mathbf{y}} \| = 1$ and the original problem comprising (5.6) and (5.7) is reduced to a standard least-squares minimisation

$$\tilde{\boldsymbol{\beta}}_{\mathrm{HRT}} = \underset{\tilde{\mathbf{y}} \in \mathbb{R}^{w-r}}{\arg\min} \| \mathtt{M}\,\mathtt{E}\,\tilde{\mathtt{V}}\,\tilde{\mathbf{y}} \|^2 \quad \text{with } \| \tilde{\mathbf{y}} \| = 1. \tag{5.9}$$

The solution $\tilde{\boldsymbol{\beta}}_{\mathrm{HRT}}$ can be found by computing the SVD of the matrix $\mathtt{M}\,\mathtt{E}\,\tilde{\mathtt{V}}$ and choosing the eigenvector corresponding to the smallest singular value. Finally, $\widehat{\boldsymbol{\theta}}_{\mathrm{HRT}} = \tilde{\mathtt{V}}\tilde{\boldsymbol{\beta}}_{\mathrm{HRT}}$.

Levenberg-Marquardt correction

The epipoles, which serve to compute the constrained tensor, are fixed and not updated in the minimisation (5.6). This can be remedied by applying the Levenberg-Marquardt algorithm to minimise the function

$$(\mathbf{e}', \mathbf{e}'') \mapsto \| \mathtt{M}\,\mathtt{E}\,\tilde{\boldsymbol{\beta}} \| \quad \text{with } \| \mathtt{E}\,\tilde{\boldsymbol{\beta}} \| = 1. \tag{5.10}$$

The steps of the overall estimation procedure are as follows:

Algorithm 14 Hartley-LM algorithm

Steps to compute an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{HRT-LM}}$ which minimises (5.9) and (5.10) :

1. Compute the trifocal tensor $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$ and retrieve the epipoles $\mathbf{e}'$ and $\mathbf{e}''$; set $k = 0$.
2. Solve problem (5.9) for the current epipoles and let $\tilde{\boldsymbol{\beta}}_k = \tilde{\boldsymbol{\beta}}_{\mathrm{HRT}}$.
3. Compute the norm of the residual error vector $\boldsymbol{\epsilon}_k = \mathtt{M}\,\mathtt{E}\,\tilde{\boldsymbol{\beta}}_k$.
4. If $\|\boldsymbol{\epsilon}_k\|$ is greater than a user-defined threshold, then apply the Levenberg-Marquardt algorithm to find new values for the two epipoles, increment k and go back to step 2. Otherwise, terminate the procedure.
5. Compute $\widehat{\boldsymbol{\theta}}_{\mathrm{HRT-LM}} = \mathtt{E}\,\tilde{\boldsymbol{\beta}}_k$ where $\mathtt{E}$ is made from the final estimates of the epipoles and the limiting value of $\tilde{\boldsymbol{\beta}}_k$.

This minimisation problem is of modest size since only six parameters, the homogeneous coordinates of the two epipoles, are involved so the computational cost remains small. The improved epipoles yield an optimal estimate of the trifocal tensor in terms of algebraic error associated with the input data [42].

Discussion

Compared to the minimisation problem (2.63), the problem (5.9) involves the metric induced by the identity covariance matrix of the parameter vector. Moreover, note that $\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}$ in (2.63) acts as a centre for the estimated entity $g(\boldsymbol{\beta})$ in the same way that $\boldsymbol{\zeta}$ is a centre for $\boldsymbol{\beta}$ in (2.74). In contrast, (5.9) can be interpreted as minimising the quantity $(\mathtt{M}\,\mathtt{E}\,\tilde{\mathtt{V}}\,\tilde{\mathbf{y}})$ with no centering, that is, the centre is *always* the null vector. The advantage however in Hartley's formulation resides in the fact that the ancillary constraints are linear and therefore can be expressed conveniently in matrix form. After some algebraic manipulations, these constraints can be included in the main objective function and solved at once to yield the inner (constrained) projections.

As we saw in Chapter 4 the elements of the trifocal tensor must satisfy 8 nonlinear internal constraints to represent a geometrically valid entity. These constraints are only partially met by Hartley's linear solution and a major drawback of this solution is that the (geometric) reprojection error is not fully reduced. Nevertheless, the method is fast and can be a good precursor for an iterative procedure. In some instances, it can even produce an estimate which is competitive with nonlinear solutions. The real image sequences we have considered give an example of both possibilities, when Hartley's method produces some results which are "far" from those of the iterative techniques and some which are of comparable quality.

5.3 FNS: Full and reduced forms

Hartley's estimation method is simple but from its simplicity raises its imperfection. The fundamental numerical scheme (FNS) and its reduced version (RFNS) offer a good alternative as they produce near-optimal unconstrained solutions and are also fast. It will be shown through experiments that when combined with post-correction methods they form good constrained estimators.

The implementation of the FNS method was analogous to that for estimating homography matrices. The only difference was to supply the scheme with a carrier matrix $\mathtt{U}$ relevant to the trifocal tensor and adopt the appropriate gradient matrix $\partial_{\mathbf{x}}\mathrm{vec}(\mathtt{U})$. Elements relevant to RFNS can be derived from those of FNS by selecting the appropriate rows in the matrices. The following sections give technical details specific to the estimation of the trifocal tensor with both methods.

5.3.1 Parameterisation for FNS and RFNS

The objective function $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ is taken as specified in Section 5.1. For FNS, the entries in $\boldsymbol{\theta}$ are chosen such that $\mathcal{T}_i^{jk}$ is the $(9i + 3j + k - 12)$-th component of $\boldsymbol{\theta}$ as set initially in

(4.18). For RFNS, $\mathtt{U}(\mathbf{x})$ and $\boldsymbol{\theta}$ can be partitioned as in (2.32) and (2.33) with $\mathtt{W} = \mathtt{I}_{4\times4}$, $\boldsymbol{\alpha} = [\,\mathcal{T}_3^{11}, \mathcal{T}_3^{12}, \mathcal{T}_3^{21}, \mathcal{T}_3^{22}\,]^\mathsf{T}$, and

$$\boldsymbol{\mu} = [\,\mathcal{T}_1^{13}, \mathcal{T}_1^{23}, \mathcal{T}_1^{31}, \mathcal{T}_1^{32}, \mathcal{T}_1^{33}, \mathcal{T}_2^{13}, \mathcal{T}_2^{23}, \mathcal{T}_2^{31}, \mathcal{T}_2^{32}, \mathcal{T}_2^{33}, \mathcal{T}_3^{13}, \mathcal{T}_3^{23}, \mathcal{T}_3^{31}, \mathcal{T}_3^{32}, \mathcal{T}_3^{33}, \mathcal{T}_1^{11}, \mathcal{T}_1^{12}, \mathcal{T}_1^{21}, \mathcal{T}_1^{22}, \mathcal{T}_2^{11}, \mathcal{T}_2^{12}, \mathcal{T}_2^{21}, \mathcal{T}_2^{22}\,]^\mathsf{T}.$$

5.3.2 Curtailing or truncating ?

The linear dependency of the gradients $\partial_{\mathbf{x}} f_j$ of the objective function (5.2) means that the length of the $\mathbf{f}(\mathbf{x}_i, \boldsymbol{\theta})$ surpasses the common codimension of the submanifolds of the form $\{\mathbf{x} \in \mathbb{R}^6 \mid \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}\}$ with $\boldsymbol{\theta}$ representing *ideal* parameters that might have generated the data. Curtailing the objective vector is not an acceptable compromise because the corresponding $\boldsymbol{\theta}$-parameterisation does not encompass all of the sought tensorial coefficients.

The linear dependency is tackled in this case by using FNS III (Algorithm 5). The singularity is accommodated by replacing the 4×4 matrices Σ_i^{-1} and $\Sigma_i'^{-1}$ by their 3-truncated pseudo-inverses $(\Sigma_i)_3^+$ and $(\Sigma_i')_3^+$ in the expressions for J_{AML} and J'_{AML}, respectively, and in related corresponding entities.

5.3.3 Data covariances

Given a data set $\{\mathbf{x}_i\}_{i=1}^n$, the covariance of each datum $\mathbf{x}_i$ takes the form of a symmetic matrix

$$\Lambda_{\mathbf{x}_i} = \begin{bmatrix} \Lambda_{\mathbf{m}_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_{\mathbf{m}'_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Lambda_{\mathbf{m}''_i} \end{bmatrix},$$

where $\Lambda_{\mathbf{m}_i}$, $\Lambda_{\mathbf{m}'_i}$, and $\Lambda_{\mathbf{m}''_i}$ are 2×2 symmetric covariance matrices associated with the inhomogeneous coordinates of the points $\mathbf{m}_i$, $\mathbf{m}'_i$, and $\mathbf{m}''_i$ respectively.

Although noise information was available in our synthetic tests, data covariance information was not exploited. In experiments involving real images $\Lambda_{\mathbf{x}_i}$ was taken to be the default 6×6 identity matrix corresponding to isotropic homogeneous noise in image point measurement.

5.4 Gold Standard method

Optimal results can be obtained by utilising the maximum likelihood (ML) estimator. When all data covariances are assumed to be the default identity matrix, the ML cost function (2.10) is given by the reprojection error

$$\sum_{i=1}^{n} \left(d(\mathbf{m}_i, \hat{\mathbf{m}}_i)^2 + d(\mathbf{m}'_i, \hat{\mathbf{m}}'_i)^2 + d(\mathbf{m}''_i, \hat{\mathbf{m}}''_i)^2 \right), \tag{5.11}$$

where points $\hat{\mathbf{m}}_i = \mathcal{N}(\mathrm{P}\mathbf{M}_i)$, $\hat{\mathbf{m}}'_i = \mathcal{N}(\hat{\mathrm{P}}'\mathbf{M}_i)$, and $\hat{\mathbf{m}}''_i = \mathcal{N}(\hat{\mathrm{P}}''\mathbf{M}_i)$ and $\hat{\mathrm{P}}'$ and $\hat{\mathrm{P}}''$ are the second and third projection matrices retrieved from the trifocal tensor estimate $\widehat{\boldsymbol{\theta}}$ as described in Section 4.6.3. Here $\mathcal{N}(\mathbf{a}) = \mathbf{a}/a^3$ is a normalisation procedure whose application ensures that the third (homogeneous) coordinate of a given planar point is unity, and $d(\mathbf{a}, \mathbf{b})$ denotes the Euclidean distance between the image points $\mathbf{a}$ and $\mathbf{b}$ that have been normalised in the above sense. The $\mathbf{M}_i$ are initially obtained by triangulating from the $\mathbf{m}_i, \mathbf{m}'_i$, and $\mathbf{m}''_i$, and are then recomputed in each optimisation step of an iterative scheme (typically, and in our case, the Levenberg-Marquardt algorithm) that minimises the reprojection error. The overall procedure constitutes the Gold Standard (GS) method.

For a trifocal tensor estimate $\widehat{\boldsymbol{\theta}}$ obtained by a method other than GS, $J_{\mathrm{ML}}(\widehat{\boldsymbol{\theta}})$ is calculated by minimising (5.11) over the $3n$ reprojected points $\hat{\mathbf{m}}_i$, $\hat{\mathbf{m}}'_i$, and $\hat{\mathbf{m}}''_i$ and keeping the projections fixed. Note the difference with the GS algorithm, in which—for finding $\widehat{\boldsymbol{\theta}}$ that minimises the reprojection error—the $\hat{\mathbf{m}}_i, \hat{\mathbf{m}}'_i, \hat{\mathbf{m}}''_i$, *and* $\hat{\mathrm{P}}'$ and $\hat{\mathrm{P}}''$ are allowed to vary simultaneously.

A commonly used accuracy measure which can be derived from (5.11) is the root-mean-squared (RMS) error. Given a trifocal tensor $\widehat{\boldsymbol{\theta}}$ it is taken to be

$$\mathrm{RMS} = \sqrt{J_{\mathrm{ML}}(\widehat{\boldsymbol{\theta}})/(6n)},$$

with 6 representing the number of elementary degrees of freedom expressible in units of length: (three images) $\times$ (two image dimensions). For an optimal estimate $\widehat{\boldsymbol{\theta}}$, its value is a good indication of the average noise contained in the data.

5.5 Experiments with synthetic data

Repeated experiments were performed in order to collect results of statistical significance. The regime adopted was to generate a 3-D scene visible by three perspective cameras and project the scene points onto images to provide "true" matches. Each image point was then perturbed by homogeneous Gaussian noise of two pixels and the resulting noise-contaminated triples of corresponding points were used as input to several algorithms.

5.5.1 Scene and camera configuration

In a standard experiment, the scene and cameras were arranged as follows. After fixing a world coordinate system, a set of 3-D points were synthetised in a cuboid of dimensions $3 \times 1.5 \times 3$ m^3 with 5 points equally spaced along each direction. The images were 3000×2000 pixels, with a pixel size of 9×9 μm^2. The centre of the cuboid was first located 5m away from the

world origin. We then applied a 15° rotation about the point $[1.5, 0.75, 4.5]^\mathsf{T}$ (front lower right corner of the cuboid) to all 3-D points.

The 125 (rotated) world points were captured by three perspective cameras placed at $\tilde{\mathbf{C}}_1 = [-5, 3, 1.5]^\mathsf{T}$, $\tilde{\mathbf{C}}_2 = [0, 0, 0]^\mathsf{T}$, and $\tilde{\mathbf{C}}_3 = [3, 3, 1.5]^\mathsf{T}$. The right camera was rotated by $-31^\circ, -36^\circ$, and -5° about the x, y, and z-axis, respectively. The central camera only had a 10° rotation about its y-axis. The left camera was subject to rotations of $-32^\circ, 35^\circ$, and 5° about its x, y, and z-axis in that order. All rotations were counterclockwise about the respective axes and relative to the camera coordinate system. For each view, rotations were applied about the z, y, and x-axis consecutively.

The three cameras used a common calibration matrix. The focal length f was set to 3600 pixels (about 32mm) which allowed an approximate $45^\circ \times 31^\circ$ viewing angles in the x and y-direction respectively. The origin of the image coordinate system was set in the centre of the image where the principal point was assumed to be located and the skew factor was taken to be zero. Figure 5.1 shows the simulated 3-D scene and camera configuration just described. The projection of the scene onto a left, central and right image provided true matches shown in Figure 5.2. The "true" trifocal tensor relating the noise-free image points was computed using (4.16) based on the knowledge of the camera projections for each of the three views.

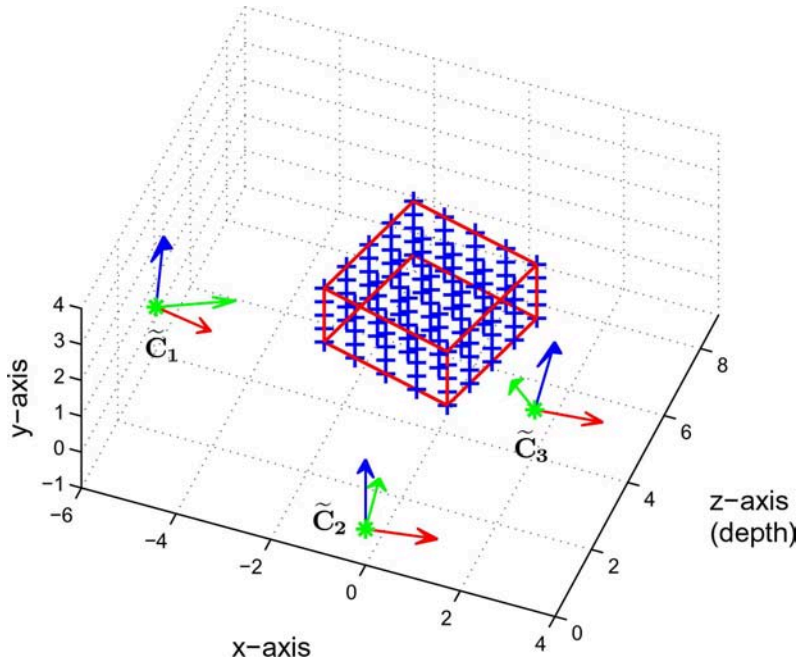


Figure 5.1. A synthetic 3-D scene made of equally spaced points inside a cuboid and three cameras viewing the scene.

To determine the difference in accuracy and computational efficiency between different algorithms, each unconstrained and constrained stage was examined. The methods' performance

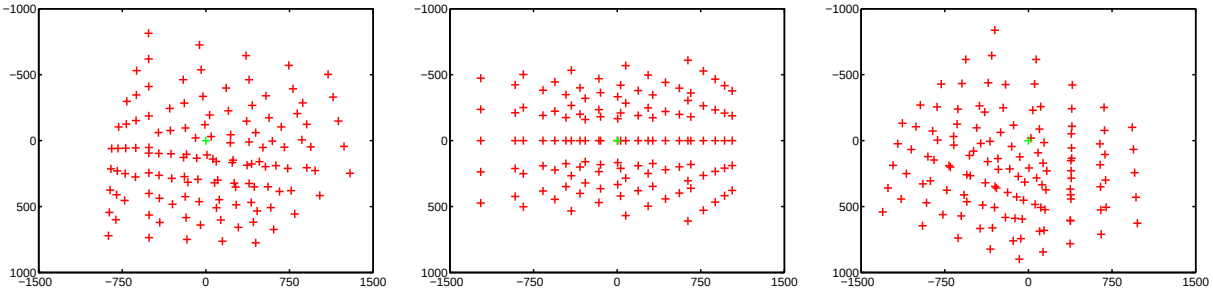


Figure 5.2. Noise-free images acquired by the three cameras shown in Figure 5.1.

was assessed through two types of measures. One type included *error* measures based on cost functions and the other included *computational* measures such as number of iterations, convergence rate, and timing.

5.5.2 Performance measures part I: Unconstrained estimation

The performance of the estimators was evaluated over a series of 200 experiments to check their average statistical behaviour in the long term.

Table 5.1 shows averages over the total number of trials for five unconstrained schemes: NALS, GTLS, FNS, RFNS, and RHEIV. The three iterative schemes were seeded with a GTLS estimate obtained as described in Section 2.2.2. The first two columns consider the J_{AML} and J'_{AML} objective functions respectively. To calculate J'_{AML} residuals for the non-reduced methods, the $\boldsymbol{\mu}$-component of each final $\boldsymbol{\theta}$-vector was retrieved and plugged into the J'_{AML} expression.

Methods	J_{AML}	J'_{AML}	Iter.	Time (sec)
NALS	1415.4	1415.3	1	0.05
GTLS	1415.4	1415.3	1	0.36
FNS	1398.1	1398.1	7.8	2.41
RFNS	1398.1	1398.1	1.5	1.02
RHEIV	1398.1	1398.1	1.5	1.2

Table 5.1. Average residual errors and computational performance of five unconstrained algorithms.

It is clearly seen that the estimates produced by the iterative schemes give all similar values of J_{AML} and J'_{AML} residuals. This in particular provides an empirical confirmation of the identity $J_{\mathrm{AML}}(\boldsymbol{\theta}) = J'_{\mathrm{AML}}(\boldsymbol{\mu})$ (see Appendix A.2). RFNS and RHEIV achieved a better convergence rate over FNS—the last scheme was almost three times slower and required about five times

as many iterations compared to the two other schemes. It can also be seen that the iterative methods are tangibly more accurate than the basic non-iterative NALS and GTLS procedures. This suggests potential utility of the iterative methods, as only accurate unconstrained estimates can be upgraded to accurate constrained estimates suitable for practical applications.

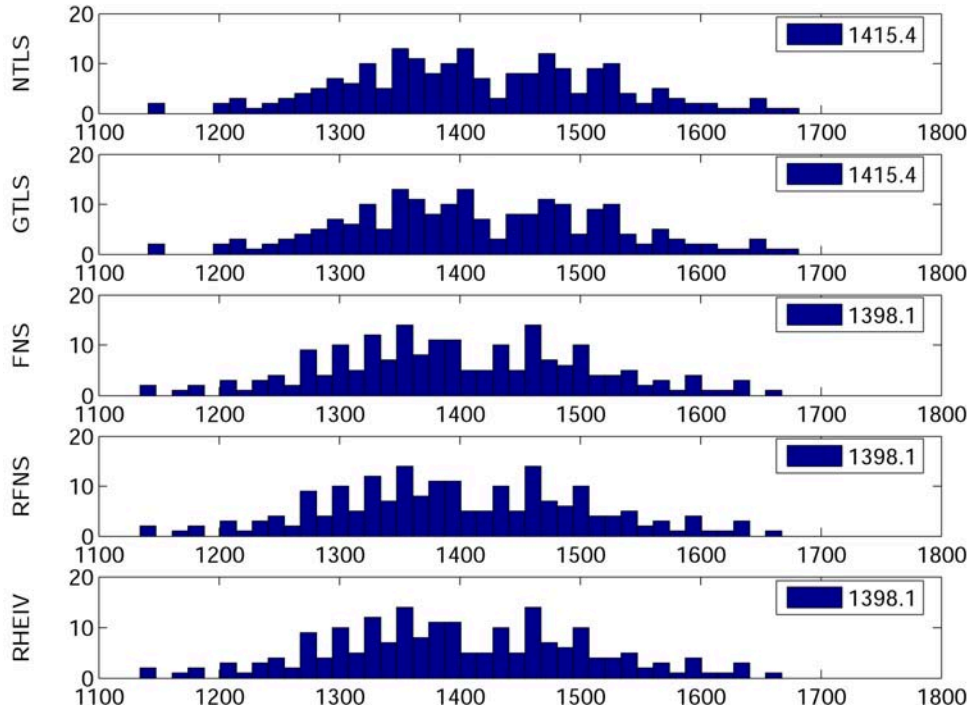


Figure 5.3. Histograms of J_{AML} values for five unconstrained methods.

To complete the analysis, Figure 5.3 shows histograms of J_{AML} values for each estimator over the 200 simulations (the average value appears in the top right corner of each histogram). Two distinct groups of histogram profiles can be distinguished. One profile corresponds to NALS and GTLS estimates (top two histograms) whereas FNS, RFNS, and RHEIV estimates generated another type of distribution (bottom three histograms).

5.5.3 Performance measures part II: Constrained estimation

This section shows the results of testing the constraint adjustment schemes presented in Section 2.5. The following acronyms will designate the type of post-correction employed: LM for Levenberg-Marquardt, GN for Gauss-Newton, TGN for Truncated Gauss-Newton, and WNLS for Weighted Nonlinear Least-Squares. Thus, the composition of FNS and LM correction will be denoted by FNS-LM. To use Kanatani's extended method, let

$$\boldsymbol{\Phi}(\boldsymbol{\theta}) = [\phi_1(\boldsymbol{\theta}), \ldots, \phi_8(\boldsymbol{\theta})]^{\mathsf{T}}, \tag{5.12}$$

where $\phi_1, \ldots, \phi_8$ are the values of the constraints (4.68), (4.77), (4.79) to (4.81), and (4.83) respectively.

With the exception of GN, all other techniques can be supplied with a non-identity covariance matrix for $\boldsymbol{\theta}$. This covariance matrix effectively induces a different metric in the parameter space. The choice of such a matrix is therefore important since different covariances may lead an algorithm to different solutions. Two main questions arise. One is to determine whether a particular choice of covariance matrix leads to a better and/or faster correction of the parameters. Secondly, given a covariance matrix, do the correction schemes converge, and if they do, is it to the same solution ? The next section presents the experimental results of this investigation.

Influence of the parameter covariance matrix in estimation

Since FNS and RHEIV produced equivalent estimates to those of RFNS, results in this section concern correction of the RFNS estimates only. The five constraint methods were applied to each of the 200 $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}$ estimates to assess the effect of incorporating different parameter covariances in the correction mechanism. Apart from GN, the methods were run on three input covariance matrices: a "default" identity matrix $\mathtt{I}_{27\times 27}$, $\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$, and $\mathtt{X}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$. Recall that GN is a simplified version of WNLS where, by definition, the covariance matrix is fixed to $\mathtt{W} = \mathtt{I}_{27\times 27}$.

Several outcomes were immediately observable. The matrix $\mathtt{X}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$ was not often positive definite and therefore made every algorithm diverge.

Kanatani's extended method turned out to be inadequate to correct an unconstrained trifocal tensor, irrespective of the covariance matrix used. For low levels of noise in the data (less than 0.2 pixels), the value of the constraint $\boldsymbol{\Phi}$ was systematically small so the method reached machine accuracy in one or two steps without providing much correction to the input estimate. The result of this operation remained very much an unconstrained vector. For higher noise levels, the method diverged as the iterations progress. Although the technique is efficient for simpler problems such as ellipse fitting or fundamental matrix estimation [49], the degree of difficulty involved in correcting an unconstrained trifocal tensor revealed to be too significant for the method to work properly.

Table 5.2 presents the average J_{AML} residuals for the various methods set with either the default matrix $\mathtt{I}_{27\times 27}$ or matrix $\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$. Clearly, whenever the identity matrix was used, the final constrained estimates were of mediocre quality no matter which method performed the correction. Although the algorithms converged, they all reached a distant local minimum where the J_{AML} cost value is significantly high. On the other hand, when the metric was induced by the matrix $\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$, all methods yielded equivalent constrained vectors. It will be shown in Section 5.5.3 that these solutions are commensurate in quality with GS's solutions.

Covariance	GN	WNLS	LM	TGN
$\mathtt{I}_{27\times 27}$	5544.1	–	5496.3	5655.8
$\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$	–	1427.6	1427.6	1427.6

Table 5.2. Mean J_{AML} values when using different parameter covariances.

Perhaps a more appealing test comes from looking at the RMS error in Table 5.3. This error approximately doubled whenever the identity matrix was used.

Covariance	GN	WNLS	LM	TGN
$\mathtt{I}_{27\times 27}$	2.53	–	2.52	2.56
$\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$	–	1.38	1.38	1.38

Table 5.3. Mean RMS errors when using different parameter covariances.

To summarise the findings, matrix $\mathtt{X}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$ was unusable as parameter covariance. The Kanatani-like method was an inefficient post-correction scheme for the trifocal tensor. The default identity matrix induced an inadequate metric to adjust the parameters so the methods either diverged or found a distant local minimum of the function. Best corrections were obtained when the matrix $\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$ was used. Operated with this covariance, WNLS, LM, and TGN converged to a similar solution.

Experimental results

We now compare the accuracy of the previous corrected estimates to optimal constrained vectors obtained from GS. Results for Hartley's method are also given for reference. In this method, no iterative correction of the epipoles was performed, this is deferred until using real image data where the benefit is more noticeable. Table 5.4 provides feedback on the algorithms performance. Every correction scheme which could use a non-identity parameter covariance matrix was supplied with $\mathtt{M}_{\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}}$.

Upon inspection, it is immediately apparent that RFNS-GN produced estimates with high cost function values. For any other combination of RFNS with a correction scheme, the cost values of the computed estimates were of comparable accuracy to the GS estimates. These values were all very similar, if slightly higher than those for the estimates generated by the respective unconstrained schemes (Table 5.1), as expected. Hartley's method yielded reasonable results and was especially fast.

The RMS error (fourth column) can be derived from the J_{ML} cost value and so, not surprisingly, values for the $\mathtt{M}$-based correction methods match those of GS. The fifth column indicates the

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
Hartley	1442.6	1442.5	1442.6	1.39	1	0.05
RFNS-GN	5544.1	3580.9	5543.9	2.53	2.7	1.47
RFNS-WNLS	1427.6	1427.5	1427.6	1.38	2.5	1.45
RFNS-LM	1427.6	1427.5	1427.6	1.38	4.3	1.08
RFNS-TGN	1427.6	1427.5	1427.6	1.38	6.22	1.69
GS	1427.8	1427.7	1433.5	1.38	12.4	22.37

Table 5.4. Average residual errors and computational performance of several constrained algorithms. Hartley and GS methods are added for reference.

number of iterations achieved in the constrained minimisation stage of the specified method. WNLS performed the least amount of iterations followed by LM and then TGN. The sixth column corresponds to the overall timing to produce a constrained parameter vector, that is, the times for both unconstrained and constrained stages were added together. Timing for a specific constraint adjustment scheme may be deduced by calculating the time difference with the results given in Table 5.1.

LM turned out to provide faster correction than WNLS. This result may be explained by the fact that LM relies on an termination condition based on the difference of successive estimates whereas WNLS (and other post-correction schemes) uses cost function values, which need to be computed at each iteration. We found experimentally that this latter choice was safer to guarantee a good constrained estimate.

Overall, RFNS followed by any post-correction operated very quickly compared to GS. Note that the search space for GS has dimension $125 \times 3 + 27 = 402$, whereas the search space for RFNS has dimensions 23—since the method does not optimise over the data points. The post-corrections typically involved a few steps and therefore executed rapidly too.

Figure 5.4 shows histograms of J_{AML} values for the constrained methods. The RFNS-based estimators and GS produced identical profiles whereas Hartley's histogram shows some differences particularly around the 1400 mark and above 1550. Compared to the histograms of the unconstrained estimators (Figure 5.3), these ones are more compact. The J_{AML} cost function and its slices were also examined; results were similar to those for homography estimation shown in Section 3.3.

We performed a metric reconstruction of the cuboid from the RFNS-TGN trifocal tensor. Figure 5.5(a) shows the corners of both the original object (red) and its reconstruction obtained from the projective tensor (blue). The "projective" cuboid is clearly different from the original one. After metric reconstruction (Figure 5.5(b)), the final object is very much aligned with the

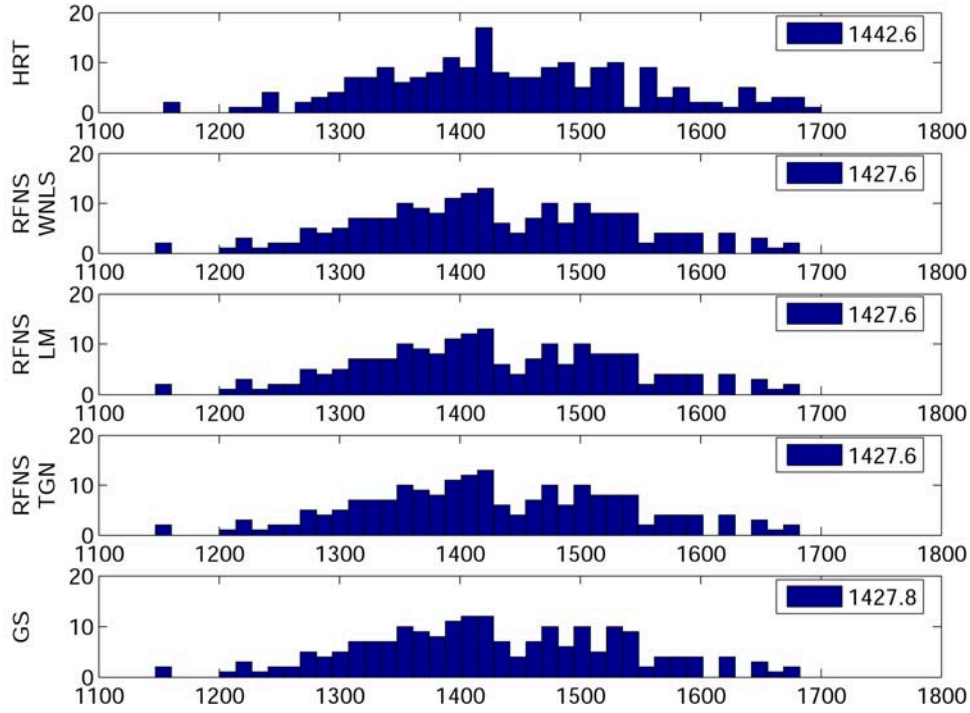
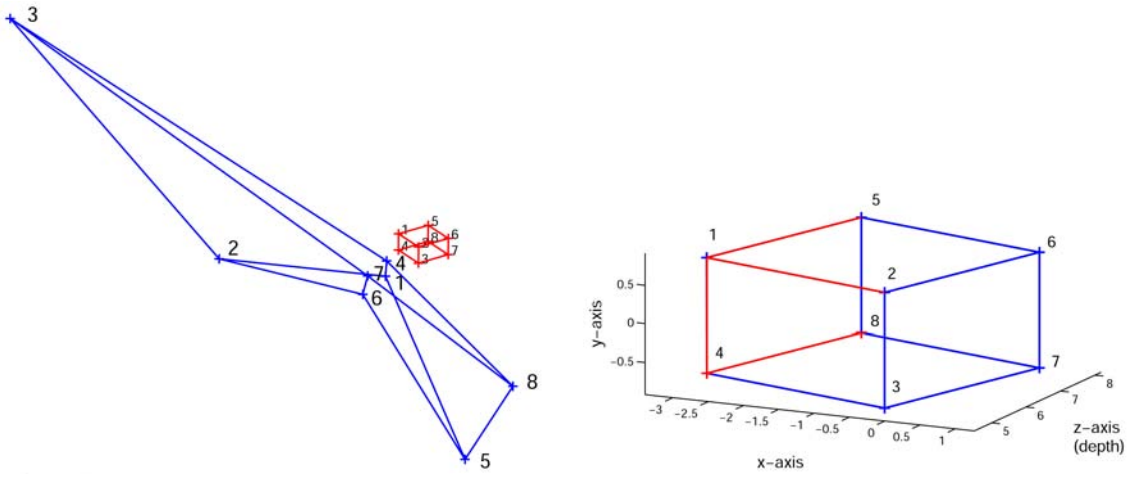


Figure 5.4. Histograms of J_{AML} values for five constrained methods. Not shown are the values for RFNS-GN and RFNS-KK, as they are off the scale.



(a) (b)

Figure 5.5. Cuboid 3-D models. The red cuboids are the original objects whereas the blue ones represent computed cuboids from the (a) projective and (b) metric RFNS-TGN trifocal tensor estimates.

starting cuboid. The mixture of red and blue edges is due to the scene viewing angle as the two cuboids are now very similar.

The values of the ancillary constraints are of great interest but will be omitted in this section. Details are delayed until Section 5.6.1 and 5.6.2 when working with real image data. This information may be more pertinent then and was found to agree with the results for synthetic data.

5.6 Considering real images

We now extend the algorithms test framework to perform experiments on data collected from two real image sequences.

5.6.1 Chemistry department sequence

Three images were acquired by a hand-held camera and 44 corresponding points were manually identified as shown in Figure 5.6. Due to a small baseline distance between any two camera positions, the trifocal plane here is not firmly defined and has potential to trigger numerical instabilities. The problem is often due to several small eigenvalues being close to zero when solving the eigenvalue problem $\mathtt{X}_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda\boldsymbol{\xi}$. The eigenvector associated with the smallest eigenvalue is no longer the sought solution and actually yields to a drastic increase in the objective function. In turn, this typically makes the algorithm diverge. A similar behaviour is observable when the noise in the data is high or outliers are present.

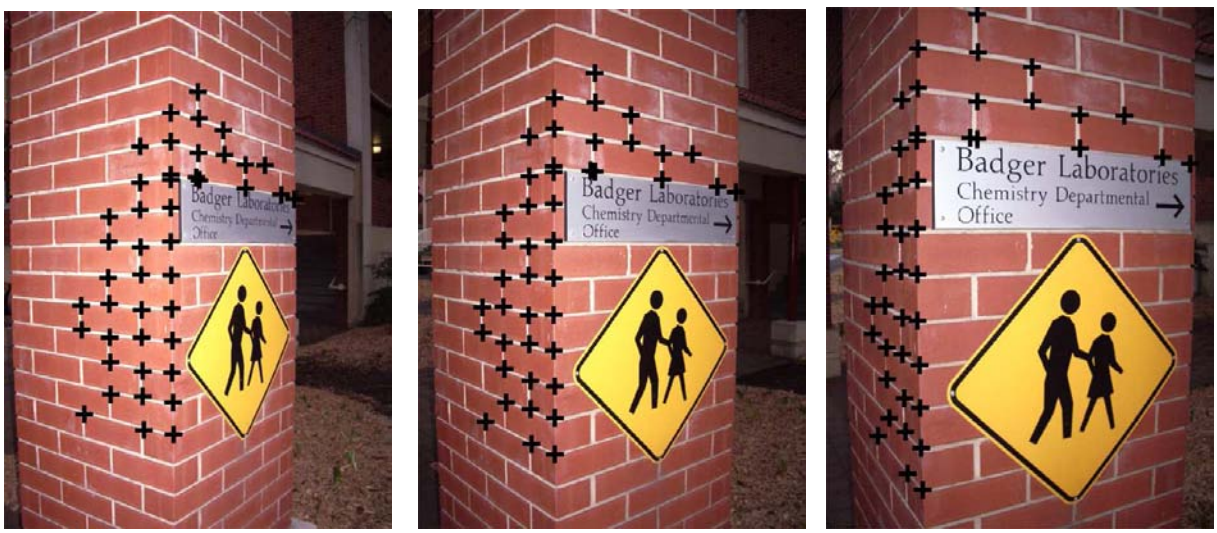


Figure 5.6. Chemistry department sequence. Each image is 600×800 pixels in size.

To ensure convergence, a modification of FNS in step 3 of Algorithm 5 became necessary. With $\mathbf{v}_{i,k}$ the normalised eigenvector corresponding to the i-th smallest eigenvalue of $\mathtt{X}_{\boldsymbol{\theta}_{k-1}}$, the update $\boldsymbol{\theta}_k$ was defined as the result of normalising $\sum_{i=1}^{3}(\boldsymbol{\theta}_{k-1}^{\mathsf{T}}\mathbf{v}_{i,k})\mathbf{v}_{i,k}$. A similar adjustment was made in Algorithm 6 describing RFNS. As far as projections on *multi-dimensional* rather than *one-dimensional* eigenspaces are concerned, it should be stressed that FNS, RFNS, or any similar scheme involving such projections converge to a solution *different* from the solution obtained by using the respective methods in their original form. The point here is that the multi-dimensional eigenspaces bring stability but potentially loose accuracy. For this reason, the dimensionality of the space spanned by the eigenvectors $\mathbf{v}_{i,k}$ should be kept as low as possible. In particular, if there are no essential gains in stability, the multi-dimensional eigenspaces should be abandoned.

Both unconstrained and constrained algorithms were run on Hartley-normalised data and normalised trifocal tensor respectively. GS was the only method to operate on the original (unnormalised) measurement data. The various iterative methods were seeded with the GTLS estimate, $\widehat{\boldsymbol{\theta}}_{\mathrm{GTLS}}$, except the Hartley-LM method which, by definition, used an NALS estimate.

Performance results part I: Unconstrained estimation

To begin, Table 5.5 shows the results of applying five unconstrained algorithms to the image data points. In each class of iterative and non-iterative estimators the methods produced very close J_{AML} and J'_{AML} values for both functions, and, given any $\boldsymbol{\theta}$-vector and its $\boldsymbol{\mu}$-component coming from a particular method, the two functions attained a similar cost. FNS and RFNS performed the same number of iterations and executed in about the same time. RHEIV lagged fractionally behind.

Methods	J_{AML}	J'_{AML}	Iter.	Time (sec)
NALS	33.2	33.2	1	0.08
GTLS	33.8	33.8	1	0.23
FNS	28.5	28.3	3	0.48
RFNS	28.6	28.6	3	0.58
RHEIV	28.7	28.6	5	0.64

Table 5.5. Residual errors and computational performance of five unconstrained schemes.

As for homography matrix estimation in Chapter 2, we examine the methods convergence rates, more particularly those of FNS and RFNS listed in Table 5.6 below. The J_{AML} cost value is given at each step as well as the relative norm difference between two successive iterates. Note here that, although FNS and RFNS operated on Hartley-normalised data, the cost values are

given for the original measurement data so that they can be compared to the final values in Table 5.5.

	FNS			RFNS		
Step	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k
0	33.8	0.433	0.53	33.8	1.180	0.43
1	36.2	0.217	0.36	34.2	0.232	0.41
2	28.7	0.072	0.17	29.1	0.098	0.39
3	28.5	0.011	0.00	28.7	0.059	0.00

Table 5.6. FNS and RFNS convergence characteristics when computing a trifocal tensor from to the chemistry department data. Step 0 gives the initial values corresponding to the seed $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{GTLS}}$.

The norm difference between successive estimates is tangibly smaller for RFNS and clearly decreases as the iterations progress. FNS exhibits a much more steady pace. This behaviour is typical in optimisation methods where the seed $\boldsymbol{\theta}_0$ is far from the solution. In this situation, the methods produce iterates which move steadily towards the local minimum and it is common that the error increases in the very first step(s). This last remark is applicable to both FNS and RFNS here. It was already noted in the case of ellipse fitting and fundamental matrix estimation [33] that FNS was not producing monotonically decreasing J_{AML} values for successive estimates.

Overall, both FNS and RFNS showed a modest linear convergence. RFNS proved to have faster convergence rate than FNS, as expected, despite a slightly higher final J_{AML} cost. In the next section, we will see that the constrained vectors produced by these two methods are commensurate in accuracy so the slightly inferior RFNS estimate has not adversely affected the constraint correction stage.

Performance results part II: Constrained estimation

The solutions delivered by FNS, RFNS, and RHEIV were then corrected with the constraint adjustment schemes presented in Section 2.5. As for synthetic data, the extended Kanatani method failed to converge, irrespective of the covariance matrix used, $\mathtt{W} = \mathtt{I}_{27\times 27}, \mathtt{M}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$, or $\mathtt{X}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$, where $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ was an AML estimate generated from the above three iterative schemes.

To facilitate reading, the results are split up into several tables. Each one highlights an (iterative) unconstrained scheme followed by a post-hoc correction. The three methods, WNLS, TGN, and LM were supplied with parameter covariance $\mathtt{M}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ coming from the unconstrained estimator they were combined with.

First, Table 5.7 gives the results of constraining $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$. For clarity, the fifth column indicates the number of iterations achieved in the constrained minimisation stage of the specified method and the timing result (sixth column) encompasses both unconstrained and constrained stages.

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
FNS-GN	131.5	112.9	131.6	0.70	4	0.62
FNS-WNLS	37.2	37.2	37.2	0.37	7	0.70
FNS-LM	37.2	37.2	37.2	0.37	13	0.90
FNS-TGN	37.2	37.2	37.2	0.37	6	0.62

Table 5.7. Residual errors and computational performance of FNS followed by four post-correction schemes.

Not surprisingly, GN provided a limited correction to the parameters. Although the algorithm converged, and quite rapidly, it had reached a distant local minimum of the function. Other schemes performed well and matched each other in all tests that measured accuracy of the final (constrained) solutions. The fastest method was TGN, followed by WNLS, and then LM.

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
RHEIV-GN	154.5	143.9	154.6	0.76	4	0.73
RHEIV-WNLS	37.5	37.3	37.5	0.38	10	0.86
RHEIV-LM	37.4	37.3	37.3	0.38	9	0.67
RHEIV-TGN	37.3	37.3	37.3	0.38	7	0.81

Table 5.8. Residual errors and computational performance of RHEIV followed by four post-correction schemes.

The same correction procedures were carried out after RHEIV estimation. Results appear in Table 5.8. As before, GN turned out to be the weakest constraining scheme with even higher cost values and RMS error than when coupled with FNS. The residual errors for other correction types are good but slightly higher than those obtained with FNS indicating that the marginally inferior unconstrained estimate did influence in this case the performance of the adjustment schemes.

The final and best results were obtained for the RFNS estimate, shown in Table 5.9. Performance of Hartley, Hartley-LM, and GS methods are also included for comparison. Despite that GN produced its best results here, the error values remain significant. The other correction procedures performed very well with TGN being the fastest again, seconded by WNLS, and LM in third place. Compared to Hartley's method, the three RFNS-corrected estimates from these

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
Hartley	57.1	57.1	57.1	0.46	1	0.08
Hartley-LM	37.3	37.3	37.3	0.38	3	0.61
RFNS-GN	116.9	105.5	116.9	0.66	5	0.66
RFNS-WNLS	37.2	37.2	37.2	0.37	7	0.69
RFNS-LM	37.2	37.2	37.2	0.37	15	0.59
RFNS-TGN	37.2	37.2	37.2	0.37	6	0.67
GS	37.2	37.2	37.2	0.37	233	43.16

Table 5.9. Residual errors and computational performance of RFNS followed by four post-correction schemes. Hartley, Hartley-LM, and GS results are added for reference.

schemes yielded marquedly better constrained solutions which are actually inseparable from that of GS. One noticeable difference is that GS's larger search space meant that the method run considerably slower than any of the three correction schemes combined with FNS, RFNS, or RHEIV. For this sequence, Hartley-LM method produced excellent results compared to its basic form with no epipole correction. Its results are close to those of the optimal iterative schemes.

The benefit of using a parameter covariance matrix is clearly visible. This is best noted in situations where any estimate resulting from applying a WNLS correction achieved optimal results compared to their GN-corrected counterpart.

Another important factor to check is the effectiveness of the correction schemes in enforcing the multi-objective constraints and ancillary constraints. For each of the final estimates obtained, the following constraints were evaluated:

1. $\psi(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}) = \sum_{j=1}^{4} |f_j(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})|$, the sum of the absolute value of the principal constraints f_j given in (5.1). This is approximately the equivalent of the epipolar constraint for the fundamental matrix.

2. $\varphi(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{8} \phi_i^2(\widehat{\boldsymbol{\theta}})$, where $\phi_1, \ldots, \phi_8$ were defined in (5.12).

3. Faugeras's epipolar constraints (4.70) and (4.71) which will be denoted by ξ_1 and ξ_2 respectively.

Table 5.10 shows example values of the above constraints for RFNS-TGN and GS parameter vectors. Combinations of FNS and RHEIV with the available constrained estimators, except with the Gauss-Newton scheme, achieved similar performance. We also give the values for $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}$ so that the accuracy of the final constrained vectors can be contrasted to some AML unconstrained estimates. The results for $\widehat{\boldsymbol{\theta}}_{\mathrm{FNS}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}$ give an appreciation of how

much correction was needed to get a fully constrained vector. GS is an ultimate benchmark to compare the efficiency of the post-correction methods.

Methods	$\psi(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$	$\varphi(\widehat{\boldsymbol{\theta}})$	ξ_1	ξ_2
FNS	0.230	4.20×10^{-6}	2.78×10^{-3}	1.29×10^{-4}
RFNS	0.197	1.70×10^{-5}	1.97×10^{-2}	1.61×10^{-2}
RFNS-TGN	0.059	5.76×10^{-37}	1.71×10^{-19}	4.62×10^{-19}
GS	0.058	1.17×10^{-38}	6.92×10^{-19}	1.08×10^{-22}

Table 5.10. Some constraint values for two unconstrained and two constrained estimates.

An important fact to note in these results is that the value of φ for the unconstrained estimates is relatively small. This explains why the Kanatani-like scheme struggled to provide any substantial correction. That it is for the constraint φ, ξ_1, or ξ_2, the difference in magnitude between unconstrained and constrained estimates is significant, hence the need to apply a powerful post-correction to reach an adequate accuracy. Although FNS and RFNS should theoretically produce equivalent estimates, these tests reveal that their respective output vectors are not guaranteed to satisfy the constraints in the same way.

Finally, to get a visual impression of the accuracy of the tested methods, the three input images of Figure 5.6 were registered using the RFNS-TGN trifocal tensor. A 3-D model was then built as shown in Figure 5.7.

Performance results part III: Accuracy of fundamental matrices

We now carry out a qualitative comparison between the fundamental matrices which can be retrieved from the trifocal tensor and an optimum estimate coming from applying GS to pairs of feature points.

In the following, fundamental matrices $\widehat{\mathrm{F}}_{21}$ and $\widehat{\mathrm{F}}_{31}$ were computed from the RFNS-TGN tensor as explained in Section 4.6.2. For comparison, optimum fundamental matrices $\widetilde{\mathrm{F}}_{21}$ and $\widetilde{\mathrm{F}}_{31}$ were computed with GS applied to each pair of views $1-2$ and $1-3$ respectively. GS was seeded with a GTLS estimate to match the initialisation condition of the RFNS-TGN trifocal tensor. The resulting four estimates were SVD-corrected to ensure that they satisfy the rank-two constraint. Given an estimated fundamental matrix $\widehat{\mathrm{F}}$, we then computed the reprojection error (ML cost), the epipolar error $\delta = \mathbf{m}'^{\mathsf{T}}\widehat{\mathrm{F}}\mathbf{m}$, and the absolute value of the ancillary constraint $\varrho(\widehat{\mathrm{F}}) = \det(\widehat{\mathrm{F}})$. Results are presented in Table 5.11.

The fundamental matrices of corresponding views have a similar ML residual and an ancillary constraint value. The matrices retrieved from the trifocal tensor seemed to be more advantageous though since they produced lower epipolar error than those of GS.

Figure 5.7. Two views of the 3-D model obtained from the RFNS-TGN trifocal tensor estimate.

F-matrix	J_{ML}	δ	$\lvert\varrho\rvert$
$\widetilde{\mathrm{F}}_{21}$	9.5	0.129	1.59×10^{-22}
$\widehat{\mathrm{F}}_{21}$	9.6	0.095	2.65×10^{-23}
$\widetilde{\mathrm{F}}_{31}$	13.2	0.225	4.23×10^{-22}
$\widehat{\mathrm{F}}_{31}$	13.2	0.054	2.12×10^{-22}

Table 5.11. Reprojection error and constraint values for various fundamental matrices.

A visual appreciation of the accuracy of $\widehat{\mathrm{F}}_{21}$ and $\widehat{\mathrm{F}}_{31}$ can be attained by inspecting Figure 5.8. Depicted is a typical set of epipolar lines obtained from transferring an image point from one view to another via these matrices. In addition, the epipolar lines from $\widetilde{\mathrm{F}}_{21}$ and $\widetilde{\mathrm{F}}_{31}$ are drawn. To contrast the difference with a trifocal tensor which does not satisfy the ancillary constraints, a third final set of lines was drawn, coming from fundamental matrices associated with the NALS trifocal tensor. The distinction between GS and RFNS-TGN epipolar lines is not visible and GS's lines are occluded by those of RFNS-TGN. The view is improved in Figure 5.9 which shows the images at sub-pixel accuracy, focusing around the feature point.

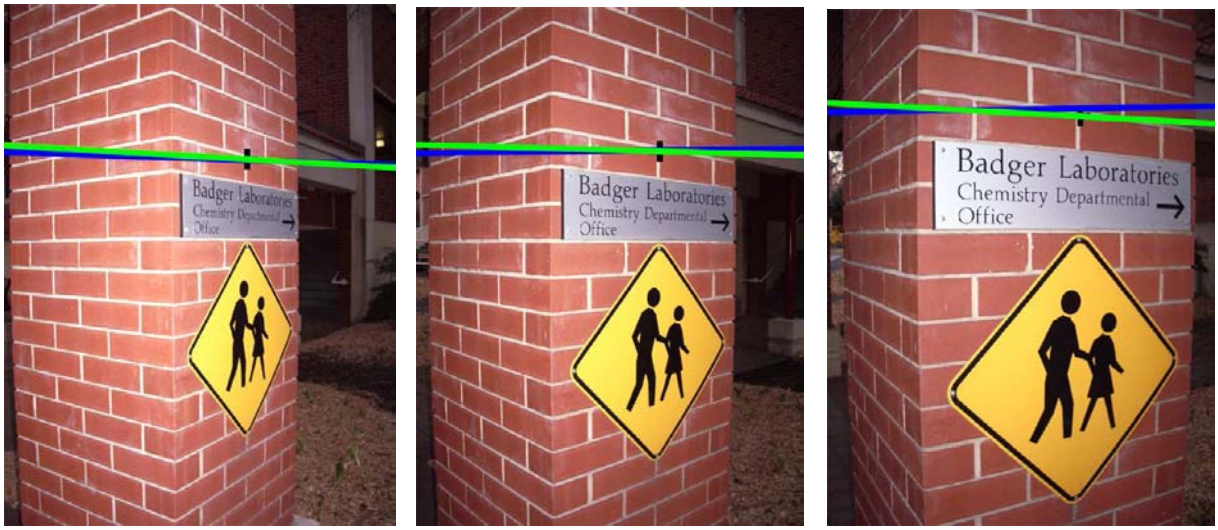


Figure 5.8. An example of epipolar lines for a particular data point. The blue and green lines are obtained from fundamental matrices retrieved from the NALS and RFNS-TGN trifocal tensors respectively. The yellow epipolar lines from GS fundamental matrices are not visible here because they lie underneath the green epipolar lines.

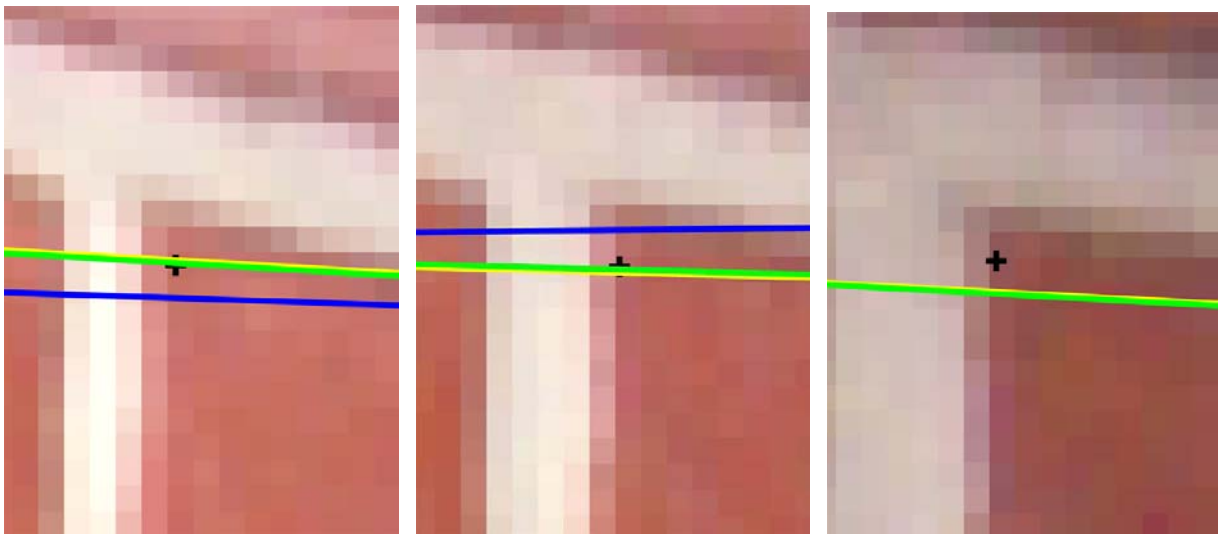


Figure 5.9. Epipolar lines of Figure 5.8 viewed at sub-pixel level. The same magnification factor has been used in all three images. The difference between RFNS-TGN and GS starts to show up but is negligible. The lines from the NALS fundamental matrices do not fit the feature point in any image; the line is even out of sight in the third image.

5.6.2 Dredger sequence

Another three images were acquired by a hand-held camera and 127 corresponding points were manually selected as shown in Figure 5.10. Once again, the trifocal tensor is not well defined because the three camera centres are close to be aligned which made the estimation process unstable. We have used the same modification as in the previous image sequence. With $\mathbf{v}_{i,k}$ the normalised eigenvector corresponding to the i-th smallest eigenvalue of $\mathtt{X}_{\boldsymbol{\theta}_{k-1}}$, the update $\boldsymbol{\theta}_k$ was defined as the result of normalising $\sum_{i=1}^{5}(\boldsymbol{\theta}_{k-1}^{\mathsf{T}}\mathbf{v}_{i,k})\mathbf{v}_{i,k}$. A similar adjustment was made in RFNS.

For this sequence, FNS, RFNS, RHEIV, and GS were initialised with the NALS estimate, $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$. Algorithms for both stages were supplied with Hartley-normalised data and normalised trifocal tensor respectively. GS was again the only method to execute on un-normalised data.

Figure 5.10. Dredger sequence. Each image is 720×576 pixels in size.

Performance results part I: Unconstrained estimation

Five unconstrained algorithms were also applied to the image points of this sequence. RHEIV did not converge, irrespective of the seed used, so it does not appear in Table 5.12.

Unlike in the previous sequence, there is more variability in the accuracy of the estimates generated. RFNS performed much less iterations than FNS but its performance, in terms of cost function values, is somewhat lower than expected. The effect of projecting on multi-dimensional eigenspaces may have lead to this poorer performance. Overall, for any of the estimates, J_{AML} and J'_{AML} residuals matched, which agrees with our expectations.

Similar convergence tests to those in Section 5.6.1 were also carried out. Results are summarised in Table 5.13. RFNS demonstrated fast convergence obtaining a small norm difference in a single step. The fact that the method quickly reached a solution indicates that the estimate obtained from the initialisation procedure was in vicinity of a local minimum of J'_{AML}.

Methods	J_{AML}	J'_{AML}	Iter.	Time (sec)
NALS	176.9	176.9	1	0.11
GTLS	185.4	185.4	1	0.48
FNS	172.9	172.9	3	1.23
RFNS	175.4	175.4	1	1.11

Table 5.12. Residual errors and computational performance of four unconstrained schemes.

However, this solution is not an optimal minimiser of J_{AML}. Started with the same seed, FNS behaved differently. The method first showed an increase in function value then steadily moved towards a local minimum. In this case, the solution yielded lower J_{AML} residual than RFNS final estimate. The algorithm's behaviour is reminescent of the steps it followed earlier when applied to the chemistry department data.

	FNS			RFNS		
Step	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k	J_{AML}	$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$	e_{k+1}/e_k
0	176.9	0.257	0.15	176.9	0.150	0.01
1	182.1	0.075	2.39	175.4	7.61×10^{-4}	0.00
2	174.3	0.101	0.08			
3	173.4	0.008	0.00			

Table 5.13. FNS and RFNS convergence characteristics when computing a trifocal tensor from to the dredger data. Step 0 gives the initial values corresponding to the seed $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$.

Performance results part II: Constrained estimation

Attention is now focused on constraining the previously obtained AML solutions. Kanatani's extended method was once again to no avail and so four correction schemes remained to be tested: GN, WNLS, LM, and TGN. Table 5.14 gives the results of coupling the FNS estimate with these schemes.

As noted in the experiments of Section 5.6.1, GN converged but produced a poor constrained solution. The other adjustment schemes performed equally well across all tests.

Table 5.15 summarises the results of constraining $\widehat{\boldsymbol{\theta}}_{\mathrm{RFNS}}$. The table also includes the performance of Hartley, Hartley-LM, and GS methods. For this sequence, Hartley's method performed exceptionally well both in terms of accuracy and execution speed. Note here that the

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
FNS-GN	614.4	587.7	614.4	0.90	3	1.30
FNS-WNLS	177.2	177.2	177.2	0.48	1	1.22
FNS-LM	177.1	177.1	177.1	0.48	4	1.67
FNS-TGN	177.3	177.3	177.3	0.48	3	1.67

Table 5.14. Residual errors and computational performance of FNS followed by four post-correction schemes.

epipole correction with LM did not improve the quality of the estimates generated. In separate tests, it actually turned that a slightly lower termination condition value and a Hartley estimate as initialisation for FNS produced an unconstrained estimate which, after correction, yielded results that matched those of the corresponding corrected RFNS estimates. Nevertheless, considering the number of matching points, the difference in the results compared to FNS seeded with $\widehat{\boldsymbol{\theta}}_{\mathrm{NALS}}$ (Table 5.14) is negligible.

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time (sec)
Hartley	177.9	177.9	177.9	0.48	1	0.11
Hartley-LM	177.9	177.9	177.9	0.48	2	0.61
RFNS-GN	179.5	178.8	179.5	0.48	1	1.22
RFNS-WNLS	177.0	177.0	177.0	0.48	1	1.26
RFNS-LM	177.0	177.0	177.0	0.48	3	1.03
RFNS-TGN	177.0	177.0	177.0	0.48	4	1.41
GS	176.7	176.7	179.3	0.48	576	625.12

Table 5.15. Residual errors and computational performance of RFNS followed by four post-correction schemes. Hartley, Hartley-LM, and GS results are added for reference.

RFNS was very competitive with GS in terms of cost function values, reprojection error and RMS error. Although GS's stopping condition was very low (1×10^{-14}), the method could not reduce the J_{ML} residual further which resulted in a discrepancy with the J_{AML} and J'_{AML} residuals. Setting a lower termination condition did not help because function values and norm differences became close to the machine precision and the method started behaving erratically. A striking observation in these results come from the methods' execution time. GS converged very slowly compared to the other techniques. In addition to its high-dimensional search space, GS here made very small steps towards a solution and therefore used up many iterations—a well known phenomenon of the Levenberg-Marquardt method when the estimation process is unstable. This behaviour was avoided in FNS and RFNS, and subsequent correction of the parameters delivered optimal constrained estimates in considerably less time.

Once again, in light of these experiments, there is no doubt that employing a non-identity parameter covariance matrix provides a distinctive advantage. Any estimate resulting from a WNLS correction achieved better accuracy than a GN-corrected one. The difference margin between their respective residuals was not as significant as for the chemistry data though.

Finally, we have checked the contraint values produced by the solutions of FNS, RFNS, RFNS-TGN, and GS. The statistics obtained are recapitulated in Table 5.16. Any corrected FNS estimate, except the one based on a Gauss-Newton correction, achieved similar accuracy. For these images the unconstrained estimates are less disparate from one another and their constraint values are of smaller order of magnitude than for the previous sequence. The amount of correction required to get a constrained vector is visibly substantial.

Methods	$\psi(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$	$\varphi(\widehat{\boldsymbol{\theta}})$	ξ_1	ξ_2
FNS	0.190	7.34×10^{-6}	4.68×10^{-4}	3.62×10^{-4}
RFNS	0.088	2.27×10^{-6}	1.01×10^{-3}	2.59×10^{-4}
RFNS-TGN	0.060	7.35×10^{-38}	3.48×10^{-19}	7.38×10^{-20}
GS	0.057	2.94×10^{-39}	9.90×10^{-20}	5.24×10^{-19}

Table 5.16. Some constraint values for two unconstrained and two constrained estimates.

As for the image sequence in Section 5.6.1, the three input images of Figure 5.10 were registered and a 3-D model was built using the projections retrieved from the RFNS-TGN trifocal tensor estimate, see Figure 5.11.

Performance results part III: Accuracy of fundamental matrices

In this last section, we investigate the precision of fundamental matrices retrieved from the RFNS-TGN trifocal tensor. Fundamental matrices computed with GS (seeded with an NALS estimate) were also used for comparison - results were no better if GS was initialised with a GTLS seed. All estimates were rank-two corrected. The notation convention is kept identical to that in Section 5.6.1. Results appear in Table 5.17.

F-matrix	J_{ML}	δ	$\lvert\varrho\rvert$
$\tilde{\mathrm{F}}_{21}$	49.4	0.082	1.22×10^{-25}
$\widehat{\mathrm{F}}_{21}$	50.6	0.090	0
$\widetilde{\mathrm{F}}_{31}$	71.7	0.42	2.65×10^{-22}
$\widehat{\mathrm{F}}_{31}$	71.7	0.066	3.18×10^{-22}

Table 5.17. Reprojection error and constraint values for various fundamental matrices.

Figure 5.11. Two views of the 3-D model obtained from the RFNS-TGN trifocal tensor estimate.

For the first two views, results from GS are closely related to those of RFNS-TGN. The main discrepancy occured in the epipolar error for views $1-3$, where the trifocal tensor clearly minimised this error better. It is obvious in this example that the trifocal tensor encapsulates fundamental matrices which are approximately of equal quality as far as ancillary constraints are satisfied. This may not be true for separate fundamental matrices, even when they are computed with GS, hence the advantage of using a trifocal tensor.

We have also looked at the fit of several sets of epipolar lines through the image data. An example for a specific point is shown in Figure 5.12 and at sub-pixel accuracy in Figure 5.13.

Figure 5.12. An example of epipolar lines for a particular data point. The blue and green lines are obtained from fundamental matrices retrieved from the NALS and RFNS-TGN trifocal tensors respectively and the yellow lines are from GS. In this case all three lines are distinct except in the third view where they are grouped closer together.

Figure 5.13. Epipolar lines of Figure 5.12 viewed at sub-pixel level. The same magnification factor has been used in all three images. The epipolar lines from GS do not superimpose on those of RFNS-TGN but still provide an excellent fit to the feature point. The lines from NALS miss the point in every image.

5.7 Conclusion

This chapter offered an application of the unconstrained minimisation algorithms and constraint post-correction schemes in the realm of trifocal tensor estimation. It was first shown through experiments on simulated data that the parameter covariance matrix plays an important role in finding a good constrained solution. Taking $\mathtt{M}_{\widehat{\boldsymbol{\theta}}^{u}_{\mathrm{AML}}}$ for a parameter covariance matrix revealed to be an adequate choice.

Supplied with this parameter covariance, the Truncated Gauss-Newton (TGN) scheme provided the most efficient constraint correction for the experimental data considered. It was optimal in terms of accuracy as it generated constrained vector of quality equivalent to that of GS. Two other post-hoc corrections, Weighted Nonlinear Least-Squares and Levenberg-Marquardt, also produced remarkably accurate and consistent results and were practically as good as TGN. The fourth adjustment scheme considered was Gauss-Newton. Although the method converged, it suffered from utilising a basic identity parameter covariance matrix. Its final solution vector was not a good constrained estimate in any of the two image sequences considered. The last method implemented and tested was Kanatani's extended method. This was the only correction scheme to act directly on the ancillary constraints and not on the projection matrices. Unfortunately, the method failed to converge in every situation because the constraint values were small and therefore prevented any substantial correction to be made.

Overall, FNS and RFNS algorithms coupled with adequate post-correction schemes were shown to produce optimal constrained solutions. These methods have simple update rules, are inexpensive to compute and therefore provide significant speed-up over traditional parameter estimation methods like the GS method.

Chapter 6

Conclusion

The thesis has described some minimisation techniques to solve a wide class of parameter estimation problems in geometric computer vision. For each method exposed or proposed, the motivating ideas were discussed, analytical calculations were presented, and final algorithmic steps were detailed. In example applications the numerical performance of the new estimators was discussed and compared to existing methods. The next sections summarise the main conclusions from each chapter and highlight a number of future research areas.

6.1 Thesis review

Initially, a specific form of a parametric model was defined. The relationships between image data and parameters pertaining to the model were expressed through a multi-objective principal constraint function. To accomodate the case where model parameters are further subject to ancillary constraints, a second constraint function was presented for use in conjunction with the principal constraints. The parametric model thus defined permits covering both unconstrained and constrained optimisation tasks. The applications considered in this thesis, homography fitting and trifocal tensor estimation, are instances of each type of problem respectively.

The development of constrained parameter estimators comprises two main stages. In the first part, an approach was adopted to solve the principal system of equations in terms of an unconstrained minimum of a cost function. The estimate generated was then used as input to an adjustment procedure which aimed to enforce the ancillary constraints. In this setting, it is assumed that the sought constrained vector lies close to the unconstrained minimum and therefore that it would be a good estimate of a global constrained minimum for the problem.

Several cost functions and associated estimation techniques have been exposed. A framework was established where objective functions and estimators could be built and analysed systematically. It started with classical least squares methods such as TLS, NTLS, and GTLS, which minimise some image algebraic error, to progress towards more statistically sophisticated functions like the maximum likelihood (ML) cost function measuring geometric errors in images.

The minimiser of the ML cost function is typically isolated by the Levenberg-Marquardt (LM) algorithm. In computer vision, LM, under the label of the Gold Standard (GS) scheme, is a pivotal procedure to compute optimal parameters by minimising maximum likelihood cost functions. The high dimensionality of the search space combined with a subtle iterative update strategy typically makes LM operate fairly slowly. These considerations led to the study of an alternative cost function. Through a variational argument, the approximated maximum likelihood (AML) cost function was proposed. Critically, the gradient of the AML function was shown to have a special form. This was exploited to compute its minimiser by a specifically devised fundamental numerical scheme (FNS).

Next, the original AML cost function was refined to cope with situations where the principal objective function has linearly dependent components. Analysis revealed that the deficiency could be overcome by the use of generalised inverses. In turn, these inverses were shown to act effectively as regularisation tools. Each type of generalised inverse gave rise to a different AML cost function and consequently to a variant of the original FNS.

Motivated by the desire to robustify these schemes even further, a reformulation of the main constraint function led to a reduced form of FNS, RFNS. The new algorithm replaces the original (unconstrained) estimation problem by two problems of lower dimension. A first iterative procedure computes only a subset of the parameters and the remaining parameters are recovered in a final single step based on the result of the earlier process. RFNS offers better convergence properties and requires a less accurate initial estimate. It was shown that the optimality condition underlying this scheme is based on a reduced AML cost function, J'_{AML}. It was further demonstrated that the minimiser of J'_{AML} could be used to produce a minimiser of the full AML cost function. While the emphasis was primarily on FNS, it was seen that a vector satisfying the reduced variational equation could alternatively be viewed as a solution of a generalised eigenvalue problem. This observation was readily exploited to advance a reduced form of the HEIV scheme.

Until this point, ancillary constraints were ignored. A new section then explored ways of integrating these constraints in an adjustment mechanism to produce constrained parameters. Novel cost functions were devised, again, within a consistent framework. A simple residual function was first proposed and immediately refined to accomodate problems where a linear relationship exists between the complete set of parameters and a subset of it. This change only affects the dimensionality of the solution space but not the constraint correction process. So, if no subset can be identified, the overall correction procedure can be applied directly to the full set of parameters. On the other hand, if a subset exists, then the linear relationship between the two sets must be taken advantage of.

The first method proposed to minimise the residual function was the Gauss-Newton (GN) method. It was shown that GN represents a particular case of a weighted nonlinear least squares (WNLS) method where the covariance matrix of the parameters is the identity. Several examples of parameter covariances were mentioned. Radically differing from the previous two methods is Kanatani-like (KK) correction scheme. The method properly uses the ancillary constraints in its adjustment mechanism. A common feature of the methods so far is that they are based only on the Jacobian of the constraints to find the next best estimate. To handle situations where the ancillary objective function is ill-conditioned, more powerful methods need to be employed. It was shown how the LM method could be applied to a suitable residual function to produce a constrained estimate. Another method capable of tackling rank-deficient function is the Truncated Gauss-Newton (TGN) method. In this case, a splitting of the Jacobian allows annihilation of elements which cause instability in the algorithm. Any of these five post-hoc correction schemes can be combined with the previous unconstrained methods to create genuine constrained estimators.

As a first application, the unconstrained algorithms were used to the computation of a planar homography. Initial synthetic experiments were conducted to check the ability of the FNS methods to deal with objective functions with linearly dependent components. These estimators were tested on data corrupted by increasingly larger noise levels, starting with a very small noise of less than a hundredth of a pixel. The influence of the initial estimate on the inversion of the problematic matrices was also measured. The results revealed that good-quality AML estimates were obtained when an adequate generalised pseudo-inverse was used. It was seen that the standard inverse failed systematically and typically made FNS diverge. Subsequent experiments on different sequences of images compared the accuracy of the AML estimates to those of existing methods. Across all sequences, it was observed that AML solutions commensurate in quality with maximum likelihood ones. A distinct advantage of FNS was that it converged at a quadratic rate and produced estimates marginally faster than GS. A confidence region around the AML solution was also visualised to check that FNS computed a minimiser of the AML function.

The second application considered was that of the estimation of the trifocal tensor. This is a much harder problem than fitting a homography to data. The difficulty might be attributed to the fact that the (standard) parameterisation of the trifocal tensor manifold is given by a nonlinear (quadratic) mapping and also to the higher number of ancillary constraints compared to the number of principal constraints. The trifocal tensor was first defined geometrically from line correspondences across three images. The trilinear constraints were then derived progressively starting from a contraction of the tensor with a single image feature, then two, and finally three image features. Geometric and algebraic properties of the tensorial slices were presented to provide necessary background before considering the ancillary constraints on the tensor. The work

of Hartley, Faugeras et al., Canterakis, and Ressl on this topic were examined. The different sets of constraints were all expressed within a common framework. The chapter concluded with practical information about recovering 3-D structure from the tensor.

The majority of the trifocal constraints derived were then employed in the unconstrained AML estimators and post-hoc correction techniques to either compute trifocal tensors or test the accuracy of given such entities. One additional method, the Hartley-LM method, was also presented as a specific technique devoted to the estimation of the trifocal tensor. The method is non-iterative, imposes linear constraints on the tensorial coefficients, and is reasonably accurate. It can therefore complement the array of precursor methods for the non-iterative schemes. In simulated tests, we began by evaluating the performance of unconstrained estimators, namely TLS, GTLS, FNS, RFNS, and RHEIV. This was followed by a similar analysis for the adjustment schemes, GN, WNLS, KK, TGN, and LM. For the latter algorithms we showed the importance of the parameter convariance matrix on the estimation. The synthetic tests already provided a statistical appreciation of the robustness of the correction schemes. Experiments on real image data confirmed earlier deductions. Convergence of FNS was linear for the sequences tested but rapid nonetheless since only a few steps were necessary. As for homography, accuracy of the resulting constrained estimates were compared to that of several other estimators, including GS. For the cost functions used and ancillary constraints considered, the results of the AML-based constrained estimates matched those of the ML estimates. Again, the FNS and RFNS-based constrained estimators executed tangibly faster than GS. The RFNS-TGN estimator turned out to be the best combination scheme in terms of accuracy and execution speed. Finally, for each image sequence, an interesting analysis was also carried out to compare the accuracy of RFNS-TGN trifocal tensors to that of optimal fundamental matrices between intermediate views.

In general, FNS and RFNS are geared to find (local) minimisers of differentiable cost functions or functions that can be rendered smooth via regularisation. These methods were designed to operate with a minimum number of elements. FNS and RFNS are Newton-Raphson-like iterative methods where each step is equivalent to solving a quadratic optimisation problem with the help of first derivative information only. Upon careful initialisation, they yield a minimiser of the AML cost functions which commensurate in accuracy with a maximum likelihood solution. Other salient features of the methods are that they have simple update rules and exhibit fast convergence. This was demonstrated here on homography fitting and trifocal tensor estimation and is in accord with earlier findings concerning such estimation problems as ellipse fitting, fundamental matrix estimation, camera resectioning, or constrained generalised principal component analysis.

6.2 Future research

Since FNS and RFNS are fast and accurate methods, an immediate extension of this work could be to incorporate them as modules in tracking algorithms or robust outlier rejection estimators. A lot of these methods rely currently on fundamental matrices to establish cross-correlation measures between images. By means of the estimators developed here, the inter-image connectors could be upgraded to trifocal tensors. This would result in stronger image geometry and increased efficiency in various applications, especially if object occlusions are a frequent problem. The capacity of the trifocal tensor to handle line correspondences could be relevant in these circumstances. Note that FNS and RFNS-based constrained algorithms could also be applied to estimate a quadrifocal tensor connecting four views of a scene. This would complete the task of computing the major building blocks of image sequences.

On a more theoretical level, desirable properties of numerical methods can be summarised in the adjectives reliable, accurate, and fast. These three aspects may all be seen to have a common denominator that relates to the convergence of the algorithm. Although rules were given to ensure convergence of FNS and RFNS, reliability may be improved by incorporating Hessian information in the minimisation process. The search direction of the next iterate would be defined more precisely which would increase the stability of the estimators. In turn, higher accuracy of the computed solution would be guaranteed. It may be envisaged that, for an application such as trifocal tensor estimation, this strategy would boost convergence to superlinear or even quadratic. A subsequent analysis could examine whether the gain in convergence speed would reduce the number of iterations. The tradeoff is to identify whether the enhanced numerical robustness compensates for the higher computational cost of calculating the Hessian of J_{AML}. Certainly for relatively simple applications like homography fitting, this addition is superfluous and should be discarded.

Regarding the post-hoc correction schemes, a line search strategy could be implemented in GN and WNLS method although our expectation is that it would provide limited benefit for the increased computational load. Now, when an unconstrained trifocal tensor is corrected with the TGN scheme, the grade of the Jacobian J_k is set manually. A convenient arrangement would be to automatically and reliably determine its value. Further attention could also be given to improve the Kanatani-like correction scheme. This work would require a deeper mathematical analysis and may lead to a novel algorithm.

Having identified many of the pitfalls in dealing with an AML cost function and ill-conditioned ancillary constraint function, it is possible to contemplate integrating the ancillary constraints into a single optimisation problem. This extension is already available in the form of the Constrained Fundamental Numerical Scheme for problems described by a single-objective principal function with a single ancillary constraint. The task would be to generalise this procedure to

the case of multiple constraints for both principal and ancillary functions. By construction, a single constrained estimator searches through a smaller parameter space and therefore may be prone to isolating a local extremum of the cost function rather than a global minimum, especially if the objective function is complex. The two-stage approach adopted in this thesis was chosen deliberately as a safety precaution to avoid such situations and unravel some of the numerical complications that arise when dealing with multi-equation functions. The accumulated knowledge should now permit to devise a fully integrated estimator.

Appendix A

Complementary Proofs for Reduced FNS

This appendix provides further proofs involved in the derivation of the Reduced Fundamental Numerical Scheme presented in Section 2.4 of Chapter 2.

A.1 Problem equivalence

We begin by showing the equivalence of (2.22) and the system which comprises (2.49a) and (2.49b). Recalling definitions (2.41) and (2.44), first note that, by (2.33) and (2.43),

$$\Sigma_i = \Sigma_i' \tag{A.1}$$

for each $i = 1, \ldots, n$. Consequently, definition (2.21b) can be rephrased as

$$\mathtt{M}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \mathtt{U}_i \Sigma_i'^{-1} \mathtt{U}_i^{\mathsf{T}}. \tag{A.2}$$

Again by (2.43), for each $i = 1, \ldots, n$,

$$\begin{aligned}
(\mathtt{I}_{l\times l} \otimes \boldsymbol{\eta}_i^{\mathsf{T}}) \mathtt{B}_i^{*} (\mathtt{I}_{l\times l} \otimes \boldsymbol{\eta}_i) = &\begin{bmatrix} \mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i^{\mathsf{T}} & \mathtt{0}_{(l-m)\times m^2} \\ \mathtt{0}_{m\times m(l-m)} & \mathtt{I}_{m\times m} \otimes \boldsymbol{\eta}_i^{\mathsf{T}} \end{bmatrix} \\
&\times \begin{bmatrix} \mathtt{B}_i' & \mathtt{0}_{m(l-m)\times m^2} \\ \mathtt{0}_{m^2\times m(l-m)} & \mathtt{0}_{m^2\times m^2} \end{bmatrix} \\
&\times \begin{bmatrix} \mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i & \mathtt{0}_{m(l-m)\times m} \\ \mathtt{0}_{m^2\times(l-m)} & \mathtt{I}_{m\times m} \otimes \boldsymbol{\eta}_i \end{bmatrix} \\
= &\begin{bmatrix} (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i^{\mathsf{T}}) \mathtt{B}_i' (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i) & \mathtt{0}_{(l-m)\times m} \\ \mathtt{0}_{m\times(l-m)} & \mathtt{0}_{m\times m} \end{bmatrix}.
\end{aligned}$$

It follows that $\mathtt{N}_{\boldsymbol{\theta}}$ given in (2.42) takes the form

$$\mathtt{N}_{\boldsymbol{\theta}} = \begin{bmatrix} \mathtt{N}_{\boldsymbol{\theta}}' & \mathtt{0}_{(l-m)\times m} \\ \mathtt{0}_{m\times(l-m)} & \mathtt{0}_{m\times m} \end{bmatrix}, \tag{A.3}$$

where

$$\mathtt{N}'_{\boldsymbol{\theta}} = \sum_{i=1}^{n} (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i^{\mathsf{T}}) \mathtt{B}'_i (\mathtt{I}_{(l-m)\times(l-m)} \otimes \boldsymbol{\eta}_i). \tag{A.4}$$

Now, if $\boldsymbol{\theta}$ satisfies (2.22), then, in view of (2.32), (A.2), and (A.3), the equivalent condition $\mathtt{M}_{\boldsymbol{\theta}}\boldsymbol{\theta} = \mathtt{N}_{\boldsymbol{\theta}}\boldsymbol{\theta}$ on $\boldsymbol{\theta}$ can be written as

$$\sum_{i=1}^{n} \begin{bmatrix} \mathtt{Z}_i \\ \mathtt{W} \end{bmatrix} \Sigma_i'^{-1} [\mathtt{Z}_i^{\mathsf{T}}, \mathtt{W}^{\mathsf{T}}] \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathtt{N}'_{\boldsymbol{\theta}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix}, \tag{A.5}$$

which in turn expands into the system

$$\sum_{i=1}^{n} \mathtt{Z}_i \Sigma_i'^{-1} (\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \mathtt{Z}_i^{\mathsf{T}}\boldsymbol{\mu}) = \mathtt{N}'_{\boldsymbol{\theta}}\boldsymbol{\mu}, \tag{A.6a}$$

$$\sum_{i=1}^{n} \mathtt{W} \Sigma_i'^{-1} (\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \mathtt{Z}_i^{\mathsf{T}}\boldsymbol{\mu}) = \mathbf{0}. \tag{A.6b}$$

By our standing assumption that $\mathtt{W}$ is invertible, the second of the above equations reduces to

$$\sum_{i=1}^{n} \Sigma_i'^{-1} (\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \mathtt{Z}_i^{\mathsf{T}}\boldsymbol{\mu}) = \mathbf{0}. \tag{A.7}$$

Now, since the Σ'_i and hence the $\Sigma_i'^{-1}$ are symmetric, it immediately follows from (2.45) that $\tilde{\mathtt{Z}}^{\mathsf{T}} = (\sum_{i=1}^{n} \Sigma_i'^{-1})^{-1} \sum_{i=1}^{n} \Sigma_i'^{-1} \mathtt{Z}_i^{\mathsf{T}}$. Hence (A.7) can be rewritten as

$$\Big(\sum_{i=1}^{n} \Sigma_i'^{-1} \Big) (\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \tilde{\mathtt{Z}}^{\mathsf{T}}\boldsymbol{\mu}) = \mathbf{0}$$

and further as

$$\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \tilde{\mathtt{Z}}^{\mathsf{T}}\boldsymbol{\mu} = \mathbf{0}. \tag{A.8}$$

As $\mathtt{W}$ is invertible, this immediately implies (2.49b).

To show that (2.49a) also holds, note that, by (2.32) and (2.33), for each $i = 1, \ldots, n$,

$$\mathtt{U}_i^{\mathsf{T}}\boldsymbol{\theta} = \mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \mathtt{Z}_i^{\mathsf{T}}\boldsymbol{\mu},$$

and by (2.46) and (A.8),

$$\mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + \mathtt{Z}_i^{\mathsf{T}}\boldsymbol{\mu} = \mathtt{W}^{\mathsf{T}}\boldsymbol{\alpha} + (\mathtt{Z}_i'^{\mathsf{T}} + \tilde{\mathtt{Z}}^{\mathsf{T}})\boldsymbol{\mu} = \mathtt{Z}_i'^{\mathsf{T}}\boldsymbol{\mu},$$

whence

$$\mathtt{U}_i^{\mathsf{T}}\boldsymbol{\theta} = \mathtt{Z}_i'^{\mathsf{T}}\boldsymbol{\mu}. \tag{A.9}$$

Recalling definitions (2.21f) and (2.47), we see that (A.9) combined with (A.1) implies that $\boldsymbol{\eta}_i = \boldsymbol{\eta}'_i$. Comparison of (2.48b) and (A.4) now yields $\mathtt{N}'_{\boldsymbol{\theta}} = \mathtt{N}'_{\boldsymbol{\mu}}$. Thus, in particular,

$$\mathtt{N}'_{\boldsymbol{\theta}}\boldsymbol{\mu} = \mathtt{N}'_{\boldsymbol{\mu}}\boldsymbol{\mu}. \tag{A.10}$$

Furthermore, in view of (2.46),

$$\sum_{i=1}^{n} \mathtt{Z}_i \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \mathtt{Z}_i^\mathsf{T} \boldsymbol{\mu}) = \sum_{i=1}^{n} (\mathtt{Z}_i' + \tilde{\mathtt{Z}}) \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \mathtt{Z}_i^\mathsf{T} \boldsymbol{\mu}).$$

By (A.7),

$$\sum_{i=1}^{n} \tilde{\mathtt{Z}}\, \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \mathtt{Z}_i^\mathsf{T} \boldsymbol{\mu}) = \tilde{\mathtt{Z}} \sum_{i=1}^{n} \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \mathtt{Z}_i^\mathsf{T} \boldsymbol{\mu}) = \mathbf{0},$$

and by (2.46), (2.48a), and (A.8),

$$\begin{aligned} \sum_{i=1}^{n} \mathtt{Z}_i' \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \mathtt{Z}_i^\mathsf{T} \boldsymbol{\mu}) &= \sum_{i=1}^{n} \mathtt{Z}_i' \Sigma_i'^{-1} (\mathtt{W}^\mathsf{T} \boldsymbol{\alpha} + \tilde{\mathtt{Z}}^\mathsf{T} \boldsymbol{\mu} + \mathtt{Z}_i'^\mathsf{T} \boldsymbol{\mu}) \\ &= \sum_{i=1}^{n} \mathtt{Z}_i' \Sigma_i'^{-1} \mathtt{Z}_i'^\mathsf{T} \boldsymbol{\mu} = \mathtt{M}_{\boldsymbol{\mu}}' \boldsymbol{\mu}. \end{aligned}$$

Putting the last three expressions together, we see that the left-hand side of (A.6a) is equal to $\mathtt{M}_{\boldsymbol{\mu}}' \boldsymbol{\mu}$. This jointly with (A.10) yields (2.49a), as required.

Working backwards, one can easily infer that if $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ satisfy (2.49a) and (2.49b) respectively, then $\boldsymbol{\theta} = [\boldsymbol{\mu}^\mathsf{T}, \boldsymbol{\alpha}^\mathsf{T}]^\mathsf{T}$ satisfies the original expression (2.22).

A.2 Common minimisers

Let $\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ and $\boldsymbol{\alpha}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ be the parts of $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ as per (2.33). Here we show that $\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ can be identified with $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, and, moreover, that both J_{AML} and J'_{AML} attain a common minimum value at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ and $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, respectively.

First note that, in view of (A.1), the expression for J_{AML} given by

$$J_{\mathrm{AML}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \boldsymbol{\theta}^\mathsf{T} \mathtt{U}_i \Sigma_i^{-1} \mathtt{U}_i^\mathsf{T} \boldsymbol{\theta}$$

can be restated as

$$J_{\mathrm{AML}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \boldsymbol{\theta}^\mathsf{T} \mathtt{U}_i \Sigma_i'^{-1} \mathtt{U}_i^\mathsf{T} \boldsymbol{\theta}. \tag{A.11}$$

Next, given an arbitrary $\boldsymbol{\mu}$, let $\boldsymbol{\alpha}$ be such that (2.49b) holds, and let $\boldsymbol{\theta} = [\boldsymbol{\mu}^\mathsf{T}, \boldsymbol{\alpha}^\mathsf{T}]^\mathsf{T}$. Then, as the calculation in Appendix A.1 immediately preceding (A.9) reveals, (A.9) holds, and this equality combined with (2.50) and (A.11) yields

$$J_{\mathrm{AML}}(\boldsymbol{\theta}) = J'_{\mathrm{AML}}(\boldsymbol{\mu}). \tag{A.12}$$

Since $J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) \le J_{\mathrm{AML}}(\boldsymbol{\theta})$, we see that $J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) \le J'_{\mathrm{AML}}(\boldsymbol{\mu})$, and since $\boldsymbol{\mu}$ can in particular be taken to be $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, we have

$$J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) \le J'_{\mathrm{AML}}(\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}). \tag{A.13}$$

On the other hand, as (2.49b) holds for $\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ and $\boldsymbol{\alpha}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ (recall that $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ satisfies (2.22), which, as shown earlier in Appendix A.1, implies (2.49b)), (A.12) can be explicitly written is this case as

$$J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}) = J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}). \tag{A.14}$$

But $J'_{\mathrm{AML}}(\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}) \leq J'_{\mathrm{AML}}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu}$, so in particular

$$J'_{\mathrm{AML}}(\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}) \leq J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}). \tag{A.15}$$

Putting (A.13), (A.14), and (A.15) together, we obtain

$$J'_{\mathrm{AML}}(\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}) = J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}) = J_{\mathrm{AML}}(\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}).$$

Hence it first follows that $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$ is equal to $\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}}$ (up to scale), as, generically, the minimiser of J'_{AML} is uniquely defined (up to scale). Furthermore, we see that J_{AML} and J'_{AML} attain a common minimum value at $\widehat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ and $\widehat{\boldsymbol{\mu}}^u_{\mathrm{AML}}$, respectively. Our claims have thus been established.

Appendix B

Homography Types

It is possible to distinguish between two types of homography matrix depending on the camera configuration capturing the scene. Algebraic details of their derivation are presented next.

B.1 Plane-induced homography

Suppose that a scene plane Π is imaged from two viewpoints. Let $\mathtt{P}_i = \mathtt{K}_i\mathtt{R}_i[\mathtt{I}_{3\times 3}| - \tilde{\mathbf{C}}_i]$, $i = 1, 2$, be the projection matrices for the two views. Assume that Π does not contain either of the camera centres $\tilde{\mathbf{C}}_1$ or $\tilde{\mathbf{C}}_2$. We shall show that corresponding image points are related by a homography.

Let $\mathbf{X}$ be a point on Π, and let $\mathbf{x}$ and $\mathbf{x}'$ be its left and right images. Consider the projection ray in 3D passing through $\mathbf{x}$ and $\tilde{\mathbf{C}}_1$. Any point $\mathbf{X}' = [\tilde{\mathbf{X}}'^{\mathsf{T}}, 1]^{\mathsf{T}}$ along this ray satisfies

$$\mathtt{K}_1\mathtt{R}_1[\mathtt{I}_{3\times 3}| - \tilde{\mathbf{C}}_1]\begin{bmatrix}\tilde{\mathbf{X}}'\\ 1\end{bmatrix} = \lambda\mathbf{x}$$

for some scalar λ, implying that

$$\tilde{\mathbf{X}}' = \tilde{\mathbf{C}}_1 + \lambda\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}.$$

If the scene plane Π has unit outward normal $\mathbf{n}$ and is situated at a distance $-d$ from the origin of the world coordinate system, then any point $\mathbf{X}'' = [\tilde{\mathbf{X}}''^{\mathsf{T}}, 1]^{\mathsf{T}}$ on the plane satisfies

$$\mathbf{n}^{\mathsf{T}}\tilde{\mathbf{X}}'' + d = 0.$$

The point $\mathbf{X} = [\tilde{\mathbf{X}}^{\mathsf{T}}, 1]^{\mathsf{T}}$ lies at the intersection of the ray and the plane. Hence

$$\mathbf{n}^{\mathsf{T}}\tilde{\mathbf{X}} + d = 0$$

and the value of λ for $\mathbf{X}$ can now be read off from the equation

$$\mathbf{n}^{\mathsf{T}}(\tilde{\mathbf{C}}_1 + \lambda\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}) + d = 0,$$

the result being

$$\lambda = -\frac{\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d}{\mathbf{n}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}}.$$

Consequently,

$$\tilde{\mathbf{X}} = \tilde{\mathbf{C}}_1 - \frac{\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d}{\mathbf{n}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}.$$

As $\tilde{\mathbf{C}}_1 \notin \Pi$, we have $\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d \neq 0$, implying $\lambda \neq 0$. Now the projection of $\mathbf{X}$ in the right image satisfies

$$\mathbf{x}' \simeq \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times3}|-\tilde{\mathbf{C}}_2]\begin{bmatrix}\tilde{\mathbf{C}}_1 + \lambda\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\\ 1\end{bmatrix}.$$

This relation does not change if the right-hand side is multiplied by the scalar $1/\lambda$. Putting $\boldsymbol{v}^\mathsf{T} = \mathbf{n}^\mathsf{T}/(\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d)$, we have

$$\mathbf{X}/\lambda = \begin{bmatrix}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x} - \boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\tilde{\mathbf{C}}_1\\ -\boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\end{bmatrix},$$

whence

$$\begin{aligned}\mathbf{x}' &\simeq \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times3}|-\tilde{\mathbf{C}}_2]\begin{bmatrix}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x} - \boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\tilde{\mathbf{C}}_1\\ -\boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\end{bmatrix}\\ &= \mathtt{K}_2\mathtt{R}_2[(\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x} - \boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\tilde{\mathbf{C}}_1) + \boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}\tilde{\mathbf{C}}_2],\\ &= \mathtt{K}_2\mathtt{R}_2[\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x} - (\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\boldsymbol{v}^\mathsf{T}\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}],\\ &= \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times3} - (\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\boldsymbol{v}^\mathsf{T}]\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x},\\ &= \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times3} - (\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d)^{-1}(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\mathbf{n}^\mathsf{T}]\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathbf{x}. \qquad \text{(B.1)}\end{aligned}$$

Since $\tilde{\mathbf{C}}_2 \notin \Pi$, we also have $\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_2 + d \neq 0$. A simple calculation now shows that

$$\begin{aligned}&[\mathtt{I}_{3\times3} - (\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d)^{-1}(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\mathbf{n}^\mathsf{T}]\\ &\qquad\qquad \times [\mathtt{I}_{3\times3} - (\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_2 + d)^{-1}(\tilde{\mathbf{C}}_2 - \tilde{\mathbf{C}}_1)\mathbf{n}^\mathsf{T}] = \mathtt{I}_{3\times3}.\end{aligned}$$

Hence the matrix

$$\mathtt{H} = \mathtt{K}_2\mathtt{R}_2[\mathtt{I}_{3\times3} - (\mathbf{n}^\mathsf{T}\tilde{\mathbf{C}}_1 + d)^{-1}(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\mathbf{n}^\mathsf{T}]\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}$$

is invertible. Finally, rewriting (B.1) as $\mathbf{x}' \simeq \mathtt{H}\mathbf{x}$, we see that $\mathbf{x}'$ is the image of $\mathbf{x}$ by the homography associated with $\mathtt{H}$.

B.2 Rotation-induced homography

A homography of simpler form is applicable in the situation when both cameras have a common centre. Such a configuration of cameras arises, for example, when a single camera rotates, possibly changing its intrinsics during the motion—the two cameras then represent the rotating camera at two different instants. With $\tilde{\mathbf{C}}$ the cameras' common centre, we have $\mathtt{P}_i = \mathtt{K}_i\mathtt{R}_i[\mathtt{I}_{3\times3}|-\tilde{\mathbf{C}}]$,

and so

$$\mathtt{P}_2 = \mathtt{K}_2\mathtt{R}_2(\mathtt{K}_1\mathtt{R}_1)^{-1}\mathtt{K}_1\mathtt{R}_1[\mathtt{I}_{3\times 3}| - \tilde{\mathbf{C}}] = \mathtt{K}_2\mathtt{R}_2\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}\mathtt{P}_1,$$

or $\mathtt{P}_2 = \mathtt{H}\mathtt{P}_1$, where

$$\mathtt{H} = \mathtt{K}_2\mathtt{R}_2\mathtt{R}_1^{-1}\mathtt{K}_1^{-1}$$

is an invertible 3×3 matrix. If a point $\mathbf{X}$ in the 3D scene gives rise to two images $\mathbf{x}$ and $\mathbf{x}'$, then $\mathbf{x} \simeq \mathtt{P}_1\mathbf{X}$ and

$$\mathbf{x}' \simeq \mathtt{P}_2\mathbf{X} = \mathtt{H}\mathtt{P}_1\mathbf{X} \simeq \mathtt{H}\mathbf{x},$$

showing that $\mathbf{x}'$ is the image of $\mathbf{x}$ via the homography associated with $\mathtt{H}$. Note that $\mathtt{H}$ in this case is the limit of the plane-induced homography matrices as d tends to $-\infty$ with $\mathbf{n}^\mathsf{T}$, $\tilde{\mathbf{C}}_1$, and $\tilde{\mathbf{C}}_2$ kept fixed. Thus the rotation-induced homography described by $\mathtt{H}$ from (3.3) coincides with the homography induced by the plane at infinity and is therefore termed the *infinite homography*.

Appendix C

Trifocal Tensor Homography Matrices

Additional material is now presented to complete the work in Section 4.3.1 on the trifocal tensor homography matrices. Here, homography slices $\mathcal{K}_k$ are considered and analogous results to Propositions 11 to 15 are derived specifically for these slices. Proofs will be succint because the rôle of matrices $\mathcal{J}_j$ and $\mathcal{K}_k$ is similar, they are both point homographies, except for the views where they map.

C.1 Column properties

A particular instantiation of Proposition 5 occurs when the line $\boldsymbol{l}'' = [1, 0, 0]^\mathsf{T}$, which yields

$$h_i^j = [\mathcal{I}_1\mathbf{e}_1, \mathcal{I}_2\mathbf{e}_1, \mathcal{I}_3\mathbf{e}_1] = \mathcal{K}_1.$$

Choosing $\boldsymbol{l}'' = \mathbf{e}_1$ reveals that matrix $\mathcal{K}_1$ represents a point homography from view Ψ_1 to view Ψ_2 via the first principal plane in view Ψ_3. The geometric properties of slices $\mathcal{K}_2$ (resp. $\mathcal{K}_3$) can be deduced if $\mathbf{e}_2$ (resp. $\mathbf{e}_3$) is used in place of $\mathbf{e}_1$, which suggests the next proposition.

Proposition 24 *Matrix $\mathcal{K}_k$ describes a homography between points in the first view and points in the second view induced by the k-th principal plane in the third view. Matrix $\mathcal{K}_k^\mathsf{T}$ maps lines in the second view to lines in the first view via the same principal plane.*

Similar to $\mathcal{J}_{l'}$, matrix $\mathcal{K}_{l''}$ in (4.39) is constructed as a linear combination of the coordinates of line $\boldsymbol{l}''$ and homography slices $\mathcal{K}_k$. The operation $\mathcal{K}_{l''}\mathbf{m}$ gives a point $\mathbf{m}'$ in Ψ_2 via the plane back-projected from line $\boldsymbol{l}''$ in Ψ_3.

Now, according to Proposition 24, employing homography $\mathcal{K}_k$ to transfer canonical point $\mathbf{e}_h$ ($h = 1, 2, 3$) in image Ψ_1 gives a point in the image Ψ_2 via the plane $\boldsymbol{\Gamma}_k''$ in Ψ_3. The special coordinates of $\mathbf{e}_h$ selects the h-th column of matrix $\mathcal{K}_k$ which implies that the columns of each matrix $\mathcal{K}_k$ represents a point in the view Ψ_2. Conversely, thinking of $\mathbf{e}_h$ as a line in Ψ_2, homography $\mathcal{K}_k^\mathsf{T}$ yields a line in Ψ_1 which has coordinates given by the h-th column of $\mathcal{K}_k^\mathsf{T}$, or h-th row of $\mathcal{K}_k$. These considerations are gathered in the following proposition.

Proposition 25 *The column vectors of each $\mathcal{K}_k$ matrix are three distinct points in the second view, whereas the row vectors of these matrices are three distinct lines in the first view.*

By interchanging entities in the image Ψ_2 with those in the image Ψ_3 in the derivation preceding Proposition 13 in Chapter 4, one may deduce some properties about the columns of matrices $\mathcal{K}_k$.

Consider point $\mathbf{e}_i$ on principal ray $\mathcal{R}_i$ in Ψ_1. The three homographies $\mathcal{K}_k$ send $\mathbf{e}_i$ to three points in the image Ψ_2 which correspond to the projection of world points $\mathbf{W}_{ik} \simeq \mathcal{R}_i \triangle \mathbf{\Gamma}''_k$ for $k = 1, 2, 3$. These image points must lie on the projection of $\mathcal{R}_i$ in Ψ_2, that is, the epipolar line $\mathcal{L}'_{1i}$ according to Table 4.4. Because $\mathbf{e}_i$ has unital i-th component with other components being zero, the images of $\mathbf{e}_i$, points $\mathcal{K}_k\mathbf{e}_i$, have coordinates given by the i-th column of matrices $\mathcal{K}_k$. This result leads to the next proposition.

Proposition 26 *The three image points obtained by selecting the i-th column vector of matrices $\mathcal{K}_1, \mathcal{K}_2$, and $\mathcal{K}_3$ are collinear and lie on the epipolar line $\mathcal{L}'_{1i}$ in the second view.*

A similar argument to the one following Proposition 13 proves that the column vectors of a given matrix $\mathcal{K}_k$ define the vertices of a triangle $\mathbf{T}'_k$ in the image Ψ_2, each vertex being on one of the epipolar lines $\mathcal{L}'_{11}, \mathcal{L}'_{12}, \mathcal{L}'_{13}$. The three triangles associated with $\mathcal{K}_1, \mathcal{K}_2$, and $\mathcal{K}_3$ are in perspective from the epipole $\mathbf{e}_{21} = \mathbf{e}'$.

The properties of matrices $\mathcal{K}_k$ elaborated so far are summarised in Table C.1. The remaining part of this section carries on from these results.

Image points in Ψ_2	On epipolar line	From world points
$\mathbf{a}'_k \simeq \mathcal{K}_k\mathbf{e}_1$	$\mathcal{L}'_{11}$	$\mathbf{W}_{1k} \simeq \mathcal{R}_1 \triangle \mathbf{\Gamma}''_k$
$\mathbf{b}'_k \simeq \mathcal{K}_k\mathbf{e}_2$	$\mathcal{L}'_{12}$	$\mathbf{W}_{2k} \simeq \mathcal{R}_2 \triangle \mathbf{\Gamma}''_k$
$\mathbf{c}'_k \simeq \mathcal{K}_k\mathbf{e}_3$	$\mathcal{L}'_{13}$	$\mathbf{W}_{3k} \simeq \mathcal{R}_3 \triangle \mathbf{\Gamma}''_k$

Table C.1. Algebraic and geometric properties of the columns of matrices $\mathcal{K}_k$.

C.2 Row properties

In Chapter 4 the convention ${}^j\mathbf{c}_h$ was used to refer to the h-th column of homography $\mathcal{J}_j^\mathsf{T}$, or h-th row of $\mathcal{J}_j$. Since matrices $\mathcal{J}_j$ and $\mathcal{K}_k$ are related, this notation also gives the j-th column of homography $\mathcal{K}_h^\mathsf{T}$, or j-th row of $\mathcal{K}_h$. It can be checked in (4.38) that $\mathbf{r}_3$ and $\mathbf{t}_2$ are represented by ${}^1\mathbf{c}_3$ and ${}^3\mathbf{c}_2$ respectively.

Now, recall from Proposition 25 that the rows of a given matrix $\mathcal{K}_k$ represent distinct lines in image Ψ_1. Following (4.38), this means that $\mathbf{r}_k, \mathbf{s}_k$, and $\mathbf{t}_k$ are edges of a triangle $\mathbf{S}_k$ in the first image where $\mathbf{r}_k = {}^1\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_1, \mathbf{s}_k = {}^2\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_2$, and $\mathbf{t}_k = {}^3\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_3$. Here, entities $\mathbf{e}_h$ ($h = 1, 2, 3$) represent lines in view Ψ_2. Geometrically, the lines $\mathbf{r}_k, \mathbf{s}_k, \mathbf{t}_k$ are obtained by projection of the world lines $\mathbf{L}_{1k}, \mathbf{L}_{2k}$, and $\mathbf{L}_{3k}$ which are the intersections of the three principal planes $\mathbf{\Gamma}'_1, \mathbf{\Gamma}'_2$, and $\mathbf{\Gamma}'_3$ (back-projection of $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ in Ψ_2) with the k-th principal plane $\mathbf{\Gamma}''_k$ (by choosing $\mathcal{K}_k^\mathsf{T}$).

Now, let $\mathbf{w}_{jk}$ be a vertex on triangle $\mathbf{S}_k$ and point $\mathbf{W}'_{kj}$ its corresponding world primitive defined as the meet of the ray $\boldsymbol{\mathcal{R}}'_j$ and the plane $\mathbf{\Gamma}''_k$. Explicitly, $\mathbf{W}'_{kj} \simeq \boldsymbol{\mathcal{R}}'_j \triangle \mathbf{\Gamma}''_k \simeq \mathbf{\Gamma}'_\alpha \triangle \mathbf{\Gamma}'_\beta \triangle \mathbf{\Gamma}''_k$, where j, α, and β are distinct values in the range $\{1, 2, 3\}$. The planes $\mathbf{\Gamma}'_\alpha$ and $\mathbf{\Gamma}'_\beta$ are the back-projection of the canonical lines $\mathbf{e}_\alpha$ and $\mathbf{e}_\beta$ in view Ψ_2. This means that these lines select rows α and β from the matrix $\mathcal{K}_k$. Therefore, we have $\mathbf{w}_{jk} \simeq {}^\alpha\mathbf{c}_k \times {}^\beta\mathbf{c}_k$. The main facts to remember from these derivations are stated in Proposition 27. Table C.2 lists information about the algebraic and geometric operations deduced from the rows of matrices $\mathcal{K}_k$.

Proposition 27 *The three image lines given by the row vectors of matrix $\mathcal{K}_k$ form a triangle $\mathbf{S}_k$ in the first view. Vertices $\mathbf{w}_{1k}, \mathbf{w}_{2k}$, and $\mathbf{w}_{3k}$ of $\mathbf{S}_k$ can be computed as the cross product of two lines represented by the appropriate rows of $\mathcal{K}_k$, see Table C.2.*

Triangle	Edges	From world lines	Vertices	From world points
	${}^1\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_1$	$\mathbf{L}_{1k} \simeq \mathbf{\Gamma}'_1 \triangle \mathbf{\Gamma}''_k$	$\mathbf{w}_{1k} \simeq {}^2\mathbf{c}_k \times {}^3\mathbf{c}_k$	$\mathbf{W}'_{k1} \simeq \boldsymbol{\mathcal{R}}'_1 \triangle \mathbf{\Gamma}''_k$
$\mathbf{S}_k$	${}^2\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_2$	$\mathbf{L}_{2k} \simeq \mathbf{\Gamma}'_2 \triangle \mathbf{\Gamma}''_k$	$\mathbf{w}_{2k} \simeq {}^3\mathbf{c}_k \times {}^1\mathbf{c}_k$	$\mathbf{W}'_{k2} \simeq \boldsymbol{\mathcal{R}}'_2 \triangle \mathbf{\Gamma}''_k$
	${}^3\mathbf{c}_k \simeq \mathcal{K}_k^\mathsf{T}\mathbf{e}_3$	$\mathbf{L}_{3k} \simeq \mathbf{\Gamma}'_3 \triangle \mathbf{\Gamma}''_k$	$\mathbf{w}_{3k} \simeq {}^1\mathbf{c}_k \times {}^2\mathbf{c}_k$	$\mathbf{W}'_{k3} \simeq \boldsymbol{\mathcal{R}}'_3 \triangle \mathbf{\Gamma}''_k$

Table C.2. Algebraic and geometric properties of the rows of matrices $\mathcal{K}_k$.

Finally, suppose that we fix the ray $\boldsymbol{\mathcal{R}}'_j$ in the view Ψ_2 and consider its intersection with the three principal planes $\mathbf{\Gamma}''_k$ of the view Ψ_3. This process yields three world points $\mathbf{W}'_{kj} \simeq \boldsymbol{\mathcal{R}}'_j \triangle \mathbf{\Gamma}''_k$, for $k = 1, 2, 3$. By construction, these 3-D points are on the line of sight $\boldsymbol{\mathcal{R}}'_j$ and so, according to Table 4.4, project into points on the epipolar line $\boldsymbol{\mathcal{L}}_{2j}$ in the first view. Using Table C.2, these image points can be identified as the vertices $\mathbf{w}_{j1}, \mathbf{w}_{j2}$, and $\mathbf{w}_{j3}$. It follows that these vertices lie on the epipolar line $\boldsymbol{\mathcal{L}}_{2j}$ going through the epipole $\mathbf{e}_{12}$.

Proposition 28 *Vertices $\mathbf{w}_{j1}, \mathbf{w}_{j2}$, and $\mathbf{w}_{j3}$ of triangles $\mathbf{S}_1, \mathbf{S}_2$, and $\mathbf{S}_3$ are collinear and lie on the epipolar line $\boldsymbol{\mathcal{L}}_{2j}$ in the first view.*

This proposition concludes the investigation of the properties of matrices $\mathcal{K}_k$.

Appendix D

Vector Cross-product

This appendix describes some useful operations derived from the vector cross-product.

Given a length-3 vector $\mathbf{p} = [p^1, p^2, p^3]^\mathsf{T}$, define the 3×3 skew-symmetric matrix

$$[\mathbf{p}]_\times = \begin{bmatrix} 0 & -p^3 & p^2 \\ p^3 & 0 & -p^1 \\ -p^2 & p^1 & 0 \end{bmatrix}.$$

This definition is motivated by the fact that, for any length-3 vector $\mathbf{q}$, we have

$$\mathbf{p} \times \mathbf{q} = [\mathbf{p}]_\times \mathbf{q} \quad \text{and} \quad \mathbf{q} \times \mathbf{p} = \mathbf{q}^\mathsf{T} [\mathbf{p}]_\times.$$

The vector $\mathbf{p}$ spans both the left and right null-spaces of the matrix $[\mathbf{p}]_\times$, which implies that $[\mathbf{p}]_\times$ has rank 2 for any non-zero vector $\mathbf{p}$. This matrix is also defined up to scale by $\mathbf{p}$ since $\lambda\mathbf{p}$ is also a null-vector of $[\mathbf{p}]_\times$ for any non-zero λ.

The matrix $([\mathbf{p}]_\times)_{sr}$ is written as $p^i \epsilon_{irs}$ in tensor notation where the indices r and s represent the (s, r)-th entry of the matrix $[\mathbf{p}]_\times$. By definition, $[\mathbf{p}]_\times \mathbf{p} = \mathbf{0}$ is equivalent to $\sum_r p^r ([\mathbf{p}]_\times)_{sr} = 0_s$, for each fixed value of $s = 1, 2, 3$. In turn, this may be written compactly as

$$p^r p^i \epsilon_{irs} = 0_s.$$

Denoting the s-th row of $[\mathbf{p}]_\times$ by $\boldsymbol{l}^{s\mathsf{T}}$, we have

$$\mathbf{p}^\mathsf{T} \boldsymbol{l}^s = 0 \qquad \text{for all } s = 1, 2, 3. \tag{D.1}$$

If $\mathbf{p}$ represents the homogeneous coordinates of an image point, then, from (D.1), each $\boldsymbol{l}^s$ may be thought as a line going through $\mathbf{p}$. The line $\boldsymbol{l}^1$ with coordinates

$$\boldsymbol{l}^1_r = p^i \epsilon_{ir1} = [0, -p^3, p^2]^\mathsf{T},$$

is a horizontal line going through the point $\mathbf{p}$ since points of the form $\mathbf{y} = [p^1 + \lambda, p^2, p^3]^\mathsf{T}$ satisfy $\mathbf{y}^\mathsf{T} \boldsymbol{l}^1 = 0$ for any λ. Similarly, the line $\boldsymbol{l}^2$ with coordinates

$$\boldsymbol{l}^2_r = p^i \epsilon_{ir2} = [p^3, 0, -p^1]^\mathsf{T},$$

is a vertical line going through $\mathbf{p}$. Finally, the line $\boldsymbol{l}^3$ with coordinates

$$\boldsymbol{l}_r^3 = p^i \epsilon_{ir3} = [-p^2, p^1, 0]^\mathsf{T},$$

is a line going through the image coordinate origin (the point $[0, 0, 1]^\mathsf{T}$).

Bibliography

[1] X. Armangué and J. Salvi. Overall view regarding fundamental matrix estimation. *Image and Vision Computing*, 21(2):205–220, 2003.

[2] M. Armstrong, A. Zisserman, and R. I. Hartley. Self-calibration from image triplets. In *Proc. European Conf. on Comput. Vision*, LNCS 1064/5, pages 3–16. Springer-Verlag, 1996.

[3] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Trans. on Visualization and Computer Graphics*, 4(4):293–306, 1998.

[4] S. Avidan and A. Shashua. Threading fundamental matrices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(1):73–77, 2001.

[5] M. Barnabei, A. Brini, and G.-C. Rota. On the exterior calculus of invariant theory. *Journal of Algebra*, 96:120–160, 1985.

[6] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. European Conf. on Comput. Vision*, pages 683–695, April 1996.

[7] M. Bergtholdt. *Auto-calibration with convex constraints*. PhD thesis, University of Mannheim, October 2002.

[8] F. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9(1):56–71, 1979.

[9] T. Buchanan. Critical sets for 3D reconstruction using lines. In *Proc. European Conf. on Comput. Vision*, pages 730–738. Springer-Verlag, 1992.

[10] T. Buchanan. On the critical set for photogrammetric reconstruction using line tokens in $P_3(\mathbb{C})$. *Geometriae Dedicata*, 44(2):223–232, November 1992.

[11] S. L. Campbell and C. D. Meyer. *Generalized Inverses of Linear Transformations*. Pitman, London-San Francisco-Melbourne, 1979.

[12] N. Canterakis. A minimal set of constraints for the trifocal tensor. In *Proc. European Conf. on Comput. Vision*, volume 1, pages 84–99, 2000.

[13] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 885–891, June 1998.

[14] W. Chojnacki, M. J. Brooks, and L. Baumela. Determining the egomotion of an uncalibrated camera from instantaneous optical flow. *Journal of the Optical Society of America A*, 14(10):2670–2677, 1997.

[15] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. On the fitting of surfaces to data with covariances. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1294–1303, 2000.

[16] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. From FNS to HEIV: A link between two vision parameter estimation methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):264–268, 2004.

[17] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. A new constrained parameter estimator for computer vision applications. *Image and Vision Computing*, 22(2):85–91, 2004.

[18] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. FNS, CFNS and HEIV: A unifying approach. *J. Math. Imaging and Vision*, 23(2):175–183, 2005.

[19] W. Chojnacki, A. van den Hengel, and M. J. Brooks. Generalised principal component analysis: Exploiting inherent parameter constraints. In *Advances in Computer Graphics and Computer Vision*, volume 4 of *Communications in Computer and Information Science*, pages 217–228. Springer, 2008.

[20] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):865–880, 2002.

[21] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, Dordrecht-Boston-London, 1996.

[22] 0. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *Proc. European Conf. on Comput. Vision*, pages 321–334, 1992.

[23] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, Mass.: MIT Press, 1993.

[24] O. D. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, Cambridge, Mass., 2001.

[25] O. D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. J. Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[26] O. D. Faugeras and B. Mourrain. About the correspondences of points between N images. In *Proc. of the IEEE Workshop on Representation of Visual Scenes*, pages 37–44, 1995.

[27] O. D. Faugeras and B. Mourrain. On the geometry and algebra of point and line correspondences between N images. In *Proc. Fifth Int. Conf. Computer Vision*, pages 951–956, June 1995.

[28] O. D. Faugeras and T. Papadopoulo. Grassmann-Cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. Technical Report RR-3225, Institut National de Recherche en Informatique et en Automatique, July 1997.

[29] O. D. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of three views. Technical Report RR-3221, Institue National de Recherche en Informatique et en Automatique, July 1997.

[30] O. D. Faugeras and L. Robert. What can two images tell us about a third one? In *Proc. European Conf. on Comput. Vision*, volume 1, pages 485–492. Springer-Verlag New York, Inc., 1994.

[31] X. Feng, J-Y. Bouguet, and P. Perona. A new geometrical interpretation of trilinear constraints. In *Proc. Asian Conf. on Computer Vision*, January 2000.

[32] A. W. Fitzgibbon and A. Zisserman. Automatic 3D model acquisition and generation of new images from video sequences. In *Proc. of European Signal Processing Conference*, pages 1261–1269, 1998.

[33] D. Gawley. *Towards an estimation framework for some problems in computer vision*. PhD thesis, University of Adelaide, September 2004.

[34] X. Hao and H. Mayer. Orientation and auto-calibration of image triplets and sequences. In *Proc. Int. Soc. of Photogrammetry and Remote Sensing*, volume XXXIV, September 2003.

[35] R. Hartley. A linear method for reconstruction from lines and points. In *Proc. Fifth Int. Conf. Computer Vision*, pages 882–887, June 1995.

[36] R. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997.

[37] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[38] R. I. Hartley. Lines and points in three views: An integrated approach. In *ARPA Image Understanding Workshop*, volume 2, pages 1009–1016, 1994.

[39] R. I. Hartley. Projective reconstruction from line correspondences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 903–907, 1994.

[40] R. I. Hartley. Multilinear relationships between coordinates of corresponding image points and lines. In *Proc. Int. Workshop on Computer Vision and Applied Geometry*, August 1995.

[41] R. I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Comput. Vision*, 22(2):125–140, 1997.

[42] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Proc. Sixth Int. Conf. Computer Vision*, pages 469–476, 1998.

[43] R. I. Hartley. Ambiguous configurations for 3-view projective reconstruction. In *Proc. European Conf. on Comput. Vision*, pages 922–935. Springer-Verlag, 2000.

[44] A. Heyden. A common framework for multiple view tensors. In *Proc. European Conf. on Comput. Vision*, LNCS 1406, pages 3–19. Springer Berlin/Heidelberg, June 1998.

[45] A. Heyden. Multiple view geometry using multifocal tensors. In *DSAGM*, pages 169–202, 1999.

[46] A. Heyden. Tensorial properties of multiple view constraints. *Mathematical Methods in the Applied Sciences*, 23(2):169–202, January 2000.

[47] W. V. D. Hodge and D. Pedoe. *Methods of Algebraic Geometry*, volume I. Cambridge University Press, 1952.

[48] K. Kanatani. Unbiased estimation and statistical analysis of 3-D rigid motion from two views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):37–50, January 1993.

[49] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, Amsterdam, 1996.

[50] K. Kanatani. Ellipse fitting with hyperaccuracy. In *Proc. European Conf. on Comput. Vision*, volume 1, pages 484–495, 2006.

[51] K. Kanatani and N. Ohta. Accuracy bounds and optimal computation of homography for image mosaicing applications. In *Proc. Seventh Int. Conf. Computer Vision*, pages 73–78, 1999.

[52] K. Kanatani, Y. Shimizu, N. Ohta, M. J. Brooks, W. Chojnacki, and A. van den Hengel. Fundamental matrix from optical flow: optimal computation and reliability evaluation. *J. Electronic Imaging*, 9:194–202, 2000.

[53] A. I. Kostrikin and Yu. I. Manin. *Linear Algebra and Geometry*, volume 1. Gordon and Breach Science Publishers, ISBN: 2-88124-683-4, 1989.

[54] P. Lancaster and M. Tismenetsky. *The Theory of Matrices with Applications*. Academic Press, San Diego, 2nd edition, 1985.

[55] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: problems with bilinear constraint. *Int. J. Comput. Vision*, 37(2):127–150, 2000.

[56] D. Lovelock and H. Rund. *Tensors, Differential Forms, and Variational Principles*. A Wiley-Interscience publication, ISBN: 0-471-54840-5, 1975.

[57] J. R. Magnus and H. Neudecker. The commutation matrix: some properties and applications. *Annals of Statistics*, 7(2):381–394, 1979.

[58] B. Matei. *Heteroscedastic errors-in-variables models in computer vision*. PhD thesis, Rutgers University, May 2001.

[59] B. Matei and P. Meer. A general method for errors-in-variables problems in computer vision. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 18–25, 2000.

[60] B. Matei and P. Meer. Reduction of bias in maximum likelihood ellipse fitting. In *Proc. Int. Conf. Pattern Recognition*, volume 3, pages 3802–3806, September 2000.

[61] B. Matei and P. Meer. A versatile method for trifocal tensor estimation. In *Proc. Eighth Int. Conf. Computer Vision*, volume 2, pages 578–585, 2001.

[62] B. Matei and P. Meer. Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1537–1552, 2006.

[63] S. Maybank. The critical line congruence for reconstruction from three images. *Applicable Algebra in Engineering, Communication and Computing*, 6(2):89–113, March 1995.

[64] P. Mendonça and R. Cipolla. Analysis and computation of an affine trifocal tensor. In *Proc. British Mach. Vision Conf.*, pages 125–133, 1998.

[65] T. Moons. A guided tour through multiview relations. *Lecture Notes in Computer Science*, 1506:304–346, June 1998.

[66] M. Mühlich and R. Mester. Subspace methods and equilibration in computer vision. Technical Report XP-TR-C-21, Inst. for Applied Physics, Johann Wolfgang Goethe-Universität, Frankfurt, Germany, November 1999.

[67] D. Nister. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proc. European Conf. on Comput. Vision*, pages 649–663, June 2000.

[68] N. Ohta and K. Kanatani. Optimal estimation of three-dimensional rotation and reliability evaluation. In *Proc. European Conf. on Comput. Vision*, volume 1, pages 175–187, 1998.

[69] T. Papadopoulo and O. D. Faugeras. A new characterization of the trifocal tensor. In *Proc. European Conf. on Comput. Vision*, pages 109–123, June 1998.

[70] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[71] P. Pritchett and A. Zisserman. Matching and reconstruction from widely separated views. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506*, pages 78–92. Springer-Verlag, June 1998.

[72] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(1):34–46, 1995.

[73] C. Ressl. *Geometry, Constraints and Computation of the Trifocal Tensor*. PhD thesis, Institute of Photogrammetry and Remote Sensing, University of Technology, Austria, June 2003.

[74] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, page 796, June 1996.

[75] P. D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18(1):97–108, 1982.

[76] F. Schaffalitzky, A. Zisserman, R. I. Hartley, and P. H. S. Torr. A six point solution for structure and motion. *Lecture Notes in Computer Science*, 1842:632–648, 2000.

[77] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[78] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):779–789, 1995.

[79] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. In *Proc. Int. Workshop Algebraic Frames for The Perception-Action Cycle*, volume 1315 of *Lecture Notes in Computer Science*, pages 190–206. Springer, 1997.

[80] A. Shashua and M. Werman. Fundamental tensor: On the geometry of three perspective views. In *Proc. Fifth Int. Conf. Computer Vision*, June 1995.

[81] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proc. Fifth Int. Conf. Computer Vision*, June 1995.

[82] M. E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *Int. J. Comput. Vision*, 4(3):171–183, 1990.

[83] G. P. Stein and A. Shashua. On degeneracy of linear reconstruction from three views: Linear line complex and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(3):244–251, 1999.

[84] Y. Sugaya and K. Kanatani. High accuracy computation of rank-constrained fundamental matrix. In *Proc. British Mach. Vision Conf.*, volume 1, pages 282–291, 2007.

[85] S. Thirthala and M. Pollefeys. The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 321–328, June 2005.

[86] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. Comput. Vision*, 24(3):271–300, 1997.

[87] P. H. S. Torr, A. Zisserman, and D. W. Murray. Motion clustering using the trilinear constraint over three views. In *Workshop on Geometrical Modeling and Invariants for Computer Vision*. Xidian University Press, 1995.

[88] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. *Int. J. Comput. Vision*, pages 338–343, July 1995.

[89] B. Triggs. Matching constraints and the joint image. In *Proc. Fifth Int. Conf. Computer Vision*, pages 338–343. IEEE Computer Society Press, June 1995.

[90] R. Vidal and R. Hartley. The multibody trifocal tensor: Motion segmentation from 3 perspective views. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 769–775, June 2004.

[91] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1945–1959, 2005.

[92] W. T. Vetterling W. H. Press, S. A. Teukolsky and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1995.

[93] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from line correspondences: Closed-form solution, uniqueness and optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(3):318–336, 1992.

[94] Y. K. Yu, S. H. Or, K. H. Wong, and M. M. Y. Chang. Accurate 3-D motion tracking with an application to super-resolution. In *Proc. Int. Conf. Pattern Recognition*, volume 3, pages 730–733, August 2006.

[95] Y. K. Yu, K. H. Wong, and M. M. Y. Chang. A fast and robust simultaneous pose tracking and structure recovery algorithm for augmented reality applications. In *Proc. Int. Conf. on Image Processing*, volume 2, pages 1029–1032, October 2004.

[96] Y. K. Yu, K. H. Wong, M. M. Y. Chang, and S. H. Or. Recursive camera-motion estimation with the trifocal tensor. *IEEE Transactions on Systems, Man, and Cybernetics - part B*, 36(5):1081–1090, October 2006.

[97] J. Zhang, F. Shi, and Y. Liu. Motion segmentation by multibody trifocal tensor using line correspondence. In *Proc. Int. Conf. Pattern Recognition*, pages 599–602, August 2006.

[98] Z. Zhang. On the optimization criteria used in two-view motion analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(7):717–729, July 1998.

[99] A. Zisserman, A. W. Fitzgibbon, and G. Cross. VHS to VRML: 3D graphical models from video sequences. In *IEEE Int. Conf. on Multimedia and Systems*, pages 51–57, June 1999.

[100] I. Zoghlami, O. D. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 420–425, 1997.