Robust estimation of structure from motion in the uncalibrated case

Anton van den Hengel, M.Comp.Sci., L.L.B., B.Sc.(Ma.Sc.)

Department of Computer Science
The University of Adelaide

Cooperative Research Centre for
Sensor Signal and Information Processing

A thesis submitted for the degree of
Doctor of Philosophy
in the Department of Computer Science
University of Adelaide

May, 2000

Abstract

A picture of a scene is a 2-dimensional representation of a 3-dimensional world. In the process of projecting the scene onto the 2-dimensional image plane, some of the information about the 3-dimensional scene is inevitably lost. Given a series of images of a scene, typically taken by a video camera, it is sometimes possible to recover some of this lost 3-dimensional information. Within the computer vision literature this process is described as that of recovering structure from motion. If some of the information about the internal geometry of the camera is unknown, then the problem is described as that of recovering structure from motion in the uncalibrated case. It is this uncalibrated version of the problem that is the concern of this thesis.

Optical flow represents the movement of points across the image plane over time. Previous work in the area of structure from motion has given rise to a so-called differential epipolar equation which describes the relationship between optical flow and the motion and internal parameters of the camera. This equation allows the calibration of a camera undergoing unknown motion and having an unknown, and possibly varying, focal length. Obtaining accurate estimates of the camera motion and internal parameters in the presence of noisy optical flow data is critical to the structure recovery process.

We present and compare a variety of methods for estimating the coefficients of the differential epipolar equation. The goal of this process is to derive a tractable total least squares estimator of structure from motion robust to the presence of inaccuracies in the data. Methods are also presented for rectifying optical flow to a particular motion estimate, eliminating outliers from the data, and calculating the relative motion of a camera over an image sequence. The thesis thus explores the application of numerical and statistical techniques for estimation of structure from motion in the uncalibrated case.

Publications

In carrying out the research that underlies this thesis, a number of papers were published [1, 2, 3, 4, 5, 6, 7]. These papers have largely been co-authored with my supervisors M. J. Brooks and W. Chojnacki. Aspects of the introductory sections of the papers appear in Chapters 1 and 2. The reconstruction formulae upon which Section 3.1 is based appeared originally in [3], as did the exact methods presented in Section 4.1 and the least median of squares scheme from Section 6.2. Aspects of the iteratively reweighted least squares method derived in Section 5.9.1 appeared in [3] and were developed further in [1, 4]. The gradient weighted least squares cost function presented in Section 5.2 was derived in [2], although for the case in which covariance information about the data is available. The error measure based on the smallest angle between the true and estimated motion matrices was used in [2] to measure the performance of different schemes. The rectification procedure for enforcing the cubic constraint on the matrices discussed in Section 2.4.1 appeared in [3] and was used in [1]. The Newton-like method first appeared in [1]. Some of the ideas presented in this thesis have been applied to the case in which covariance information about the data is available [2, 5, 6, 7].

[1] L. Baumela, M. J. Brooks, W. Chojnacki, and A. van den Hengel. Robust techniques for the estimation of structure for motion. In H. Burkhardt and B. Neumann, editors, *Computer Vision—ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, pages 281–295, Fifth European Conference on Computer Vision, Freiburg, Germany, June 2–6, 1998. Springer, Berlin.

[2] M. J. Brooks, W. Chojnacki, A. Dick, A. van den Hengel, K. Kanatani, and N. Ohta. Incorporating optical flow information into a self-calibration procedure for a moving camera. In S. F. El-Hakim and A. Gruen, editors, *Videometrics VI*, volume 3641 of *Proceedings of SPIE*, pages 183–192, San Jose, California, USA, January 28–29, 1999.

[3] M. J. Brooks, W. Chojnacki, A. van den Hengel, and L. Baumela. 3D reconstruction from optical flow generated by an uncalibrated camera undergoing unknown motion. In H. Pan, M. J. Brooks, D. McMichael, and G. Newsam, editors, *Image Analysis and Information Fusion, Proceedings of the International Workshop IAIF'97*, pages 35–42, Adelaide, Australia, November 1997. Cooperative Research Centre for Sensor Signal and Information Processing, The Levels, South Australia.

[4] M. J. Brooks, W. Chojnacki, A. van den Hengel, and L. Baumela. Estimation of structure from motion in the uncalibrated case. In *Proceedings of the IPSJ Workshop on Computer Vision and Image Media*, volume PRMU97-180 (1997-12) of *Technical Report of IEICE*, pages 49–56, Utsunomiya, Japan, November 1997. The Institute of Electronics, Information and Communication Engineers.

[5] W. Chojnacki, M. J. Brooks, and A. van den Hengel. Fitting surfaces to data with covariance information: fundamental methods applicable to computer vision. Technical Report TR99-03, Department of Computer Science, University of Adelaide, August 1999.

[6] W. Chojnacki, M. J. Brooks, and A. van den Hengel. Rationalising Kanatani's method of renormalisation in computer vision. In *Statistical Methods for Image Processing*, pages 61–63, Uppsala, Sweden, August 1999. International Statistical Institute.

[7] 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. *Journal of Electronic Imaging*, 9(2):194–202, April 2000.

Declaration

This thesis contains no material that 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 for this copy of my thesis, when deposited in the University Library, to be available for loan and photocopying.

Anton van den Hengel
April, 2000

Acknowledgments

Primarily I would like to thank Professor M. Brooks and Professor W. Chojnacki not only for their supervision over the course of my candidature, but for having made the Ph.D. process interesting and even enjoyable. I would also like to acknowledge the support provided by The Cooperative Research Centre for Sensor Signal and Information Processing. For the epic patience of my friends and family, I am extremely grateful, but it is really the confidence of my mother in the value of persistence that has made all of this possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the problem of determining structure from motion, including some background information and a brief survey of the relevant literature. We then provide an outline of the structure of the remainder of the thesis.

1.1 Goal: Robust estimation of structure from motion

The problem of determining structure from motion is that of recovering 3-dimensional information about a scene from an image sequence. Figure 1.1 shows a series of images of the Yosemite Valley from a sequence synthetically generated by Lynn Quam at SRI. Viewing such a sequence as a static group of images makes detecting changes between frames difficult. When viewed in succession as an image stream, however, these differences give rise to a compelling sense of the shape of the object viewed. In the case of Figure 1.1 the nature of the static images gives a clue as to the shape of the valley. The movement from image to image, however, provides independent information about the shape of the scene. From this we infer that the motion of points across the image plane of a moving camera is related to the shape of the object viewed. It is this observation that underlies the determination of structure from motion.

There are many methods for determining structure from motion, some of which are described in Section 1.4. This thesis concentrates on a method developed by Viéville and Faugeras [135], and Brooks et al. [16–18] based on a differential version of the epipolar equation for uncalibrated cameras. For a detailed description of the method, see Chapter 2.

Any method of estimating scene structure from the motion of points through an image sequence is necessarily limited by the accuracy of measurement of this motion. Unfortunately, inaccuracies in this measurement process are unavoidable, meaning that calculating structure from motion becomes a question of statistical estimation. The majority of this thesis is concerned with improving the robustness of the statistical estimation of structure. Some new auxiliary

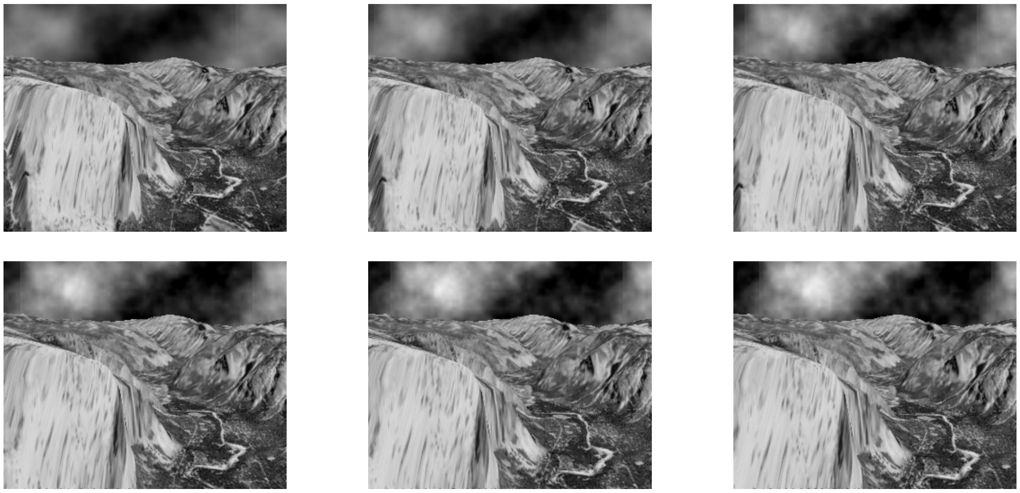

Figure 1.1: Images from the Yosemite Valley sequence

results are also generated relating to the reconstruction of viewed scenes, and the trajectory taken by the camera over a time interval.

1.2 Notation

Before proceeding any further we introduce some notation. In this thesis the following conventions will be adopted:

- scalars are represented by lower case italic letters, e.g. a,
- the absolute value of a scalar a is denoted by $|a|$,
- vectors will be denoted by boldface lower case letters, e.g. $\boldsymbol{a}$,
- vector elements will be represented in lower case italic font, as in $\boldsymbol{a} = [a_1, a_2, a_3]^T$,
- the norm of a vector $\boldsymbol{a}$ will be denoted $||\boldsymbol{a}||$, i.e., $||\boldsymbol{a}|| = (\sum_i a_i^2)^{\frac{1}{2}}$,
- points in space will, in general, be represented as vectors, but points in 3-dimensional 'scene' space may be represented as upper case letters, such as A, to distinguish them from image points,
- matrices will be denoted by boldface upper case letters, e.g. $\boldsymbol{A}$, with their elements in lower case italic: a_{ij},
- the determinant of a matrix $\boldsymbol{A}$ will be denoted $|\boldsymbol{A}|$,
- the norm of a matrix $\boldsymbol{A}$ is represented by $||\boldsymbol{A}||$, i.e., $||\boldsymbol{A}|| = \left(\sum_{ij} a_i^2\right)^{\frac{1}{2}}$

If $\boldsymbol{a} = [a_1, \ldots, a_n]^T$ and $\boldsymbol{b} = [b_1, \ldots, b_n]^T$, then let $\boldsymbol{ab} = \sum_{i=1}^{n} a_i b_i$ denote the inner product of $\boldsymbol{a}$ and $\boldsymbol{b}$. We now define the cross product $\boldsymbol{a} \times \boldsymbol{b}$ to be a vector $\boldsymbol{c}$ of n elements such that $c_i = a_i b_i$. The norm $||\boldsymbol{a}||$ of a vector $\boldsymbol{a}$ of n elements is given by $(a_1^2 + \cdots + a_n^2)^{\frac{1}{2}}$, the norm $||\boldsymbol{A}||$ of an $n \times m$ matrix $\boldsymbol{A}$ by $\left(\sum_{i,j=1}^{n,m} a_{ij}^2\right)^{\frac{1}{2}}$.

1.3 Camera model

A model of the camera used to capture an image sequence is essential to the process of recovering the structure of the scene viewed. In this work the model used is that of the ubiquitous pinhole camera (see Figure 1.2). Within this model light is projected through a small aperture located at the *optical centre* of the camera (point C in the figure) onto the *image plane*. The image plane is located at a distance f from the optical centre. The distance f is labelled the *focal length* of the camera. Associated with the camera is a coordinate frame Γ_c, with the x and y axes in a plane parallel to the image plane, and the z axis perpendicular to it. The origin of this coordinate frame is the optical centre of the camera. The point on the image plane closest to the optical centre has coordinates $[0, 0, -f]^T$, this being the *principal point* of the camera, labelled D in Figure 1.2. The frame Γ_c provides the basis for all image-based measurements. The pinhole model

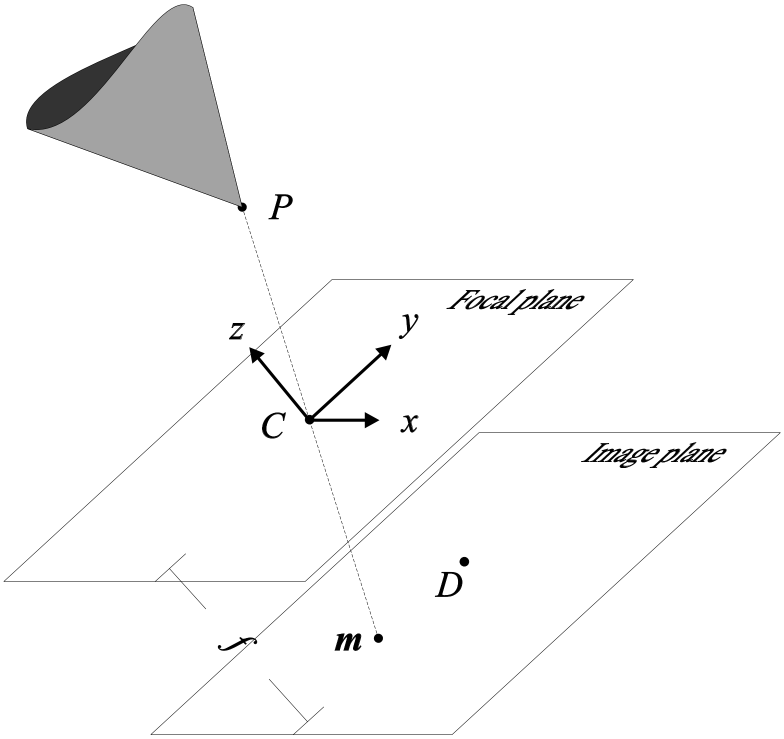

Figure 1.2: The pinhole camera model

may seem simplistic, given the complex nature of modern cameras and more particularly lenses, but it provides a good abstraction and a reasonably accurate representation of the imaging process for many cameras. A detailed exposition of the pinhole model, including an explanation of the terminology, can be found in Ref. [47, Section 3].

1.4 Background

Much of the relevant literature is discussed, in context, throughout the remainder of this thesis. We present at this stage, however, a brief background to the process of estimating structure from image sequences, and the statistical estimation techniques applicable. We begin with a description of the related process of recovering 3-dimensional information about a scene from images taken by two cameras at the same instant. This is the problem tackled in stereo vision.

The process of reconstructing 3-dimensional shape from a pair of images is based on triangulation, whereby the intersection of two lines issuing from two cameras is computed (see Section 2.1). This triangulation requires that each image feature should provide a vector along which the corresponding scene point must lie. Calculation of the intersection of these lines is only possible if we know the situation in which the images were taken. It is useful to break up the parameters describing a particular imaging situation into those internal to the cameras and those describing the relative positions of the cameras. The parameters internal to a camera, such as the focal length and the location of the principal point, are labelled the *intrinsic parameters*. The *extrinsic parameters*, in contrast, describe the relationship between two cameras. These two parameter sets have been termed collectively the *key parameters* [84].

Cameras for which the intrinsic parameters are known are termed calibrated cameras. In contrast, for uncalibrated cameras, some, or all, of this information is unknown. If we assume that the intrinsic parameters of a pair of cameras are known, then all that remains, in order to enable reconstruction, is to determine the extrinsic parameters. This information, relating the positions of two cameras, is embedded within the epipolar equation for calibrated cameras, which is covered in more detail in Section 2.1.1.

The epipolar equation for calibrated cameras was introduced by Longuet-Higgins [77] as an extension of the work of Kruppa [38, 75] at the beginning of this century. The epipolar equation for calibrated cameras encompasses the extrinsic parameters of the cameras within the essential matrix [89]. A number of methods for determining the essential matrix have been developed [77, 130]. Regardless of the method used, reconstruction cannot be carried out until all five extrinsic parameters have been recovered from the matrix [47]. Only the direction component of the translation between the cameras is recoverable by this method due to an ambiguity inherent in the imaging geometry, the magnitude of the translation is not recoverable. This causes a scale indeterminacy in the subsequent reconstruction.

The epipolar equation for uncalibrated cameras is of the same form as that for calibrated cameras, except that the essential matrix is replaced by the fundamental matrix. This change reflects the fact that the intrinsic parameters of the cameras must now be represented. The form of this equation is given in section 2.1.2. Recovering the key parameters from the fundamental matrix is considerably more complex than recovering the extrinsic parameters from the essential matrix. This is partly because there may be more key parameters than degrees of freedom in the fundamental matrix. In addition to this, the convoluted way in which the parameters are combined in the fundamental matrix complicates the extraction process. An iterative method for recovering these parameters from the fundamental matrix has been developed by Horn [61,62]. Unfortunately not all stereo imaging situations allow recovery of the key parameters. There has been significant work carried out in the area of determining degeneracies in the stereo imaging process [56,78,79,132,133], particularly in relation to scene shapes that lead to multiple valid image interpretations.

Both the calibrated and uncalibrated epipolar equations assume a perspective projection camera model such as the pinhole camera described in Section 1.3. This perspective projection camera model is described in more detail in Refs. [42, 83,138], but other camera models exist such as those based on affine [74,146] and orthographic [126] projection. For a good introduction to projective geometry in computer vision, see Ref. [103].

If, rather than considering two static cameras, we contemplate the case of one camera in motion taking a sequence of images over time, it turns out to be useful to alter the representation of the key parameters to reflect this change. The extrinsic parameters would therefore no longer describe the relative position of two cameras, but the relative motion of one. Similarly, the intrinsic parameters would describe not only the internal state of the camera, but also the rate of change of some aspects of this state. The endeavour of recovering 3-dimensional information about the world from images taken by a camera in motion is described as the structure from motion problem. This process of determining structure from motion is based on the same principles as stereo vision, but relates to the case in which images are taken sequentially, rather than simultaneously [82,90].

The use of image sequences rather than pairs provides the possibility of increasing robustness by tracking the location of image features over longer sequences of images [45]. This work has typically involved extension of the epipolar equation to multiple images as initiated by Shashua [111, 112], and followed by many [41,44,46,53,58,113,131,141]. Tomasi and Kanade [125,126] developed a method of determining structure and motion under orthographic projection based on the factorisation technique developed by Debrunner and Ahuja [35, 36]. This method has been extended to incorporate least squares techniques by Szeliski and Kang [122], and again to cope with non-rigid scenes by

Debrunner and Ahuja [37]. For an overview of structure from motion techniques, see Refs. [47, 63, 103].

In the case of video sequences taken by a moving camera, the difference in camera position between subsequent images is typically quite small. This results in a similarity of subsequent image pairs which renders determination of corresponding points more tractable. The reconstruction process, however, relies upon significant translation of the camera between image pairs for accurate 3-dimensional reconstruction. So, although points may be tracked more accurately than in the stereo case, the underlying geometry makes reconstruction more sensitive to errors. One method used to overcome this problem has been to track image points over a sequence of images. This enables calculation of the velocity of the point through an image, rather than its movement between images. The collection of these image point velocities is called optical flow. A more detailed description of optical flow is provided in Section 1.6. The structure from motion problem suffers from the same types of degeneracies as does general stereo. This degeneracy is represented in particular camera motions and scene shapes, and therefore certain kinds of optical flow fields (see Refs. [80, 81, 91]).

Recovery of structure from motion should not be confused with the problem of recovering scene shape from a moving stereo head, although the problems are related. For techniques that address this problem, see [145] and [23].

The majority of this thesis is based on a means of describing the changing state of a moving camera on the basis of an optical flow field developed by Brooks et al. [16–18]. One of the advantages of the method proposed by these authors is the availability of closed form solutions for the key parameters, thus alleviating the need for an iterative determination as in the general stereo case.

It has been suggested by Soatto et al. [115,116,118] and Heeger and Jepson [57] that motion and structure should be computed separately, and, more particularly, that 3-dimensional structure is not necessary to compute motion estimates. In contrast to this, a number of authors including McLauchlan and Murray [93] have stated that structure and motion are so inter-related that the separation is artificial and leads to a compounding of errors [76, 122]. The method of Brooks et al. follows this latter line of thinking in that it recovers structure and motion simultaneously.

Alongside the recent progress in the understanding of the underlying mathematics of computer vision, there has also developed an increasing awareness of the importance of statistical methods to improve estimation techniques. For example, the 8-point method of Longuet-Higgins for estimating the essential matrix, although improved by data normalisation techniques [54], has been largely superseded by techniques based on statistical methods capable of incorporating more data (see Refs. [84, 85, 128] for reviews of statistical methods for estimating the fundamental matrix). So far, within the computer vision literature, there has been little work in the area of statistical estimation of structure from motion in the uncalibrated case using a projective camera model. It is this area which is the concern of this thesis.

Optical flow information may be utilised for a number of purposes other than determining structure from motion. There have been a number of methods developed, some of which are capable of using optical flow contaminated by noise [2, 8, 123, 124]. Ohta and Kanatani [101] have applied statistical techniques to the structure from motion problem, but their work is concerned with the calibrated case and hence is not immediately applicable to optical flow generated by an uncalibrated camera. The work of Ohta and Kanatani also assumes that only the velocity component of the optical flow is perturbed by noise, which simplifies the problem [68, Chap. 12]. This is similar to the approach of Mühlich and Mester [95] to the stereo problem in which they applied least squares techniques assuming that all error occurs in one image. For more examples of work dealing with the ego-motion of a calibrated camera see Refs. [52, 57]. A contrasting approach is given by Beardsley et al. [13], involving the computation of projective and affine (rather than Euclidean) structure from motion. Other related papers include Refs. [3–5, 10, 86, 105, 117, 136, 137].

In some cases it is possible to estimate the covariance of particular optical flow measurements, so, as part of the measurement process, we would obtain not only the optical flow, but also an indication of its reliability. Some analysis of the use of covariance estimates (describing the uncertainty of the data) in the estimation of structure from motion has been carried out; see for example Refs. [19, 20, 68, 101]. For the purposes of this thesis we will assume that, as is commonly the case, no covariance information is available.

1.5 The nature of the differential epipolar equation

The differential epipolar equation is an algebraic relationship between the location and velocity of an image point and two matrices representing the motion and internal parameters of the camera. The motion across an image of a point $\boldsymbol{m}$ is denoted by the vector $\dot{\boldsymbol{m}}$. The relationship between $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ is shown in Figure 1.3.

The vectors $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ are represented in homogeneous form, thus

$$\boldsymbol{m} = [m_1, m_2, 1]^T \text{ and } \dot{\boldsymbol{m}} = [\dot{m}_1, \dot{m}_2, 0]^T. \tag{1.1}$$

Homogeneity refers to the property by which these 2-dimensional entities are represented as vectors with three entries. This representation simplifies the mathematics by allowing particular nonlinear operations to be specified by linear relations [103]. The homogeneous form and the values given to the last elements of the vectors $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ are explained in Section 1.6. If, as we have suggested, the flow vector itself is represented by $\dot{\boldsymbol{m}}$, and its position in the image by the vector $\boldsymbol{m}$, then by a particular encoding of the key parameters in two matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ we have that

$$\boldsymbol{m}^T\boldsymbol{C}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{W}\dot{\boldsymbol{m}} = 0. \tag{1.2}$$

This is the differential epipolar equation defined by Brooks et al. [17,18].

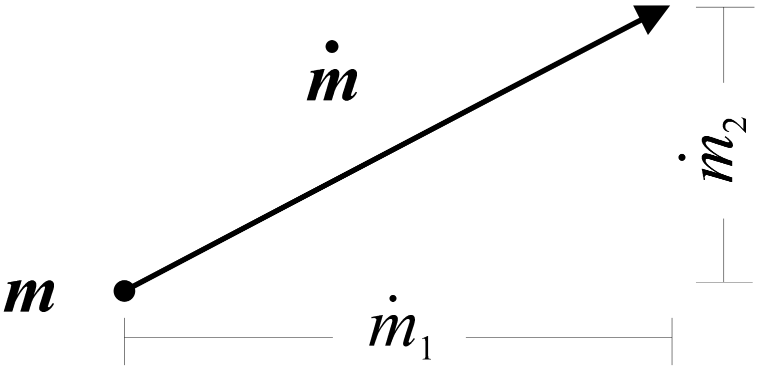

Figure 1.3: An optical flow vector

In order to fully understand the differential epipolar equation we first need to describe in more detail both the optical flow and the means of encoding the key parameters of the moving camera.

1.6 Optical flow

The term optical flow refers to the motion of points across the image plane due to the relative motion between camera and scene. If we assume that the imaging process is instantaneous then an image is a static representation of the viewed scene at a particular time. If there is any movement in the scene, or if the camera itself is in motion with respect to the scene, this may be visible by comparing successive images. Optical flow is a representation of this motion. It is quite possible to imagine a situation in which no difference will be discernible between successive images by judicious selection of camera motion, scene shape and object texture. In general, however, relative motion between camera and scene will result in variation over successive images.

An optical flow field is a set of optical flow vectors representing the velocity of points across an image at some instant. The location of a point $\boldsymbol{m}$ in an image is specified by its x and y coordinates labelled m_1 and m_2 respectively. This image point represents the intersection with the image plane of a ray from the scene point passing through the optical centre of the camera (see figure 1.2). In fact, it is these rays that we are interested in. The exact location of the image plane determines only the size of the image, the relationship between the rays being unaltered. As has been pointed out above, it is useful to represent image points in homogeneous coordinates, by the addition of a third component to the position vector. The position of our image point $\boldsymbol{m}$ in homogeneous coordinates is

$$\boldsymbol{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}.$$

The image plane is most simply regarded as a 2-dimensional space of reals $\mathbb{R}^2$. Some of the mathematics involved in image formation becomes simpler, however, if the image plane is represented as part of the 2-D projective space $\mathbb{P}^2$. A point in $\mathbb{R}^2$ is represented by a 2-vector $\boldsymbol{r} = [r_1, r_2]^T$. A point in $\mathbb{P}^2$ is represented by an equivalence class of 3-vectors $\boldsymbol{p} = [p_1, p_2, p_3]^T$, with $||\boldsymbol{p}|| \neq 0$. The vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ are members of the same class if there exists a non-zero scalar λ such that $\boldsymbol{p} = \lambda\boldsymbol{q}$. Our image points in $\mathbb{R}^2$ may thus be represented as part of $\mathbb{P}^2$ by the conversion to homogeneous coordinates, which requires only the addition of a third element to the vector. This additional element may be chosen freely, but must not be 0 and must be consistent across the image plane. For simplicity, we select 1, thus representing the vector $\boldsymbol{r}$ as $[r_1, r_2, 1]^T$. Those parts of $\mathbb{P}^2$ which are not part of $\mathbb{R}^2$ (that is $\mathbb{P}^2 \setminus \mathbb{R}^2$) coincide with the set of vectors of the form $[r_1, r_2, 0]^T$.

Having made this change to homogeneous coordinates we are no longer interested only in the values of m_1 and m_2 but in the ratio $m_1 : m_2 : m_3$. This property of the homogeneous representation simplifies the representation of the projection of scene points onto the image plane which occurs within our pinhole camera model. If the coordinates of a particular scene point Q are $[x, y, z]^T$ with respect to the camera based coordinate system, then the projection onto the image plane will generate an image point $\boldsymbol{r}$ given by

$$\boldsymbol{r} = \begin{bmatrix} -fx/z \\ -fy/z \end{bmatrix},$$

where f represents the distance between the image plane and the optical centre, or the focal length of the camera. This projection is represented in 2-dimensional form in Figure 1.4.

Projection onto the image plane cannot be represented as a linear equation in terms of the vectors Q and $\boldsymbol{r}$ as they stand. If the vectors are represented in homogeneous coordinates, however, the projection may be represented by an equation of the form

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

if we let

$$r_1 = \frac{m_1}{m_3} \text{ and } r_2 = \frac{m_2}{m_3}.$$

Representing Q, and more particularly $\boldsymbol{r}$, in homogeneous coordinates thus enables linear representation of a non-linear relationship. On the basis of the

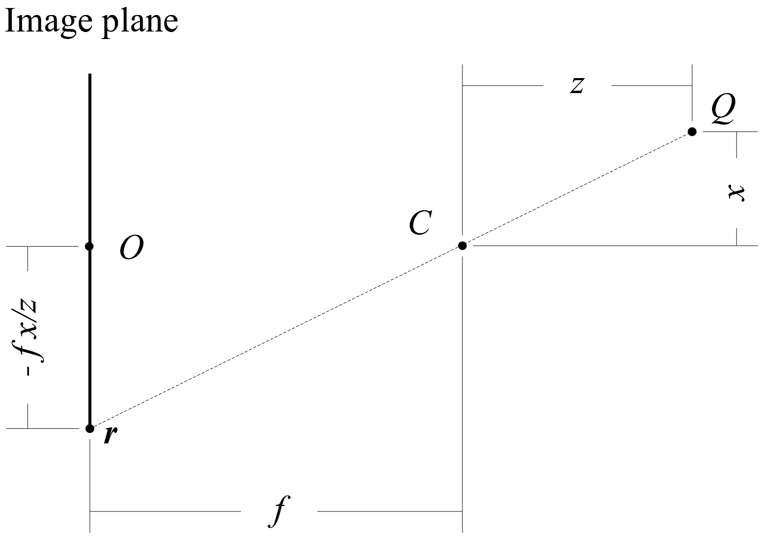

Figure 1.4: Projection of image points

above we represent the location of an image feature with x and y coordinates of m_1 and m_2 respectively as $\boldsymbol{m} = [m_1, m_2, 1]^T$. For a more detailed description of the advantages of homogeneous representation, see Refs. [47, 103].

An optical flow vector represents the motion of a point across an image and thus the derivative of point such as $\boldsymbol{m}$ above. We see from our representation of $\boldsymbol{m}$ that only the first two elements may vary, the third element being 1. This clearly reflects the fact that image points are constrained to the image plane. We thus represent an optical flow vector with x and y dimensions of $\dot{m}_1$ and $\dot{m}_2$ respectively as $\dot{\boldsymbol{m}} = [\dot{m}_1, \dot{m}_2, 0]^T$.

An optical flow vector is thus described by two vectors of length 3 such as $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$. If we denote a particular optical flow vector by $\{\boldsymbol{m}_i, \dot{\boldsymbol{m}}_i\}$, the field of n such pairs may be represented as the set $\mathcal{S} = \{\{\boldsymbol{m}_i, \dot{\boldsymbol{m}}_i\} \mid i = 1 \ldots n\}$.

1.6.1 The motion field

It proves useful, at this point, to consider what B.K.P. Horn in *Robot Vision* [60] defines a *motion field*. The motion field corresponding to a camera moving with respect to the viewed scene is made up of velocity vectors describing the velocity of each visible point across the image plane (see also Ref. [134]). A motion field thus represents the projection of the scene motion relative to the camera onto the image plane for every visible scene point. The motion field is thus not defined in terms of the properties of the image created within a particular camera, but in terms of the relative motion between camera and scene. Optical flow, being an image based measurement, may thus, at times, diverge from the motion field.

If we consider the location of a particular world point relative to the camera as a function of time, denoted by $P(t)$, then the position of its projection onto the image plane may be regarded similarly as a function of time. We label

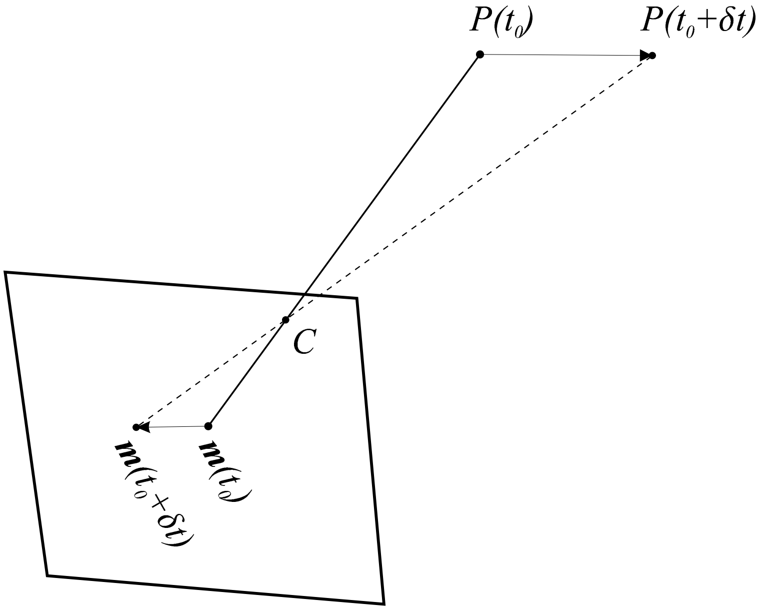

Figure 1.5: Motion projection

this point $\boldsymbol{m}(t)$. Figure 1.5 shows the nature of this projection onto the image plane. In the interval from t_0 to $t_0 + \delta t$ that part of the scene at point $P(t_0)$ moves to $P(t_0 + \delta t)$. Correspondingly, the image of point $P(t_0)$ moves from $\boldsymbol{m}(t_0)$ to $\boldsymbol{m}(t_0 + \delta t)$. Following this process for all points in the scene visible at time t, we arrive at a full description of the motion of every point in the image over the interval δt. By dividing the length of each vector by δt we arrive at a representation of the velocity of every point across the image plane

$$\frac{\boldsymbol{m}(t_0 + \delta t) - \boldsymbol{m}(t_0)}{\delta t}. \tag{1.3}$$

In the limit as δt approaches 0, the set of these velocity vectors approaches the motion field at time t_0. The differential epipolar equation holds for every vector in such a motion field.

The motion field is thus a vector field corresponding to the projection of the vectors representing scene motion relative to the camera onto the image plane. It is generally not possible to compute the motion field from a sequence of images. It may, however, be possible to determine the optical flow field which often constitutes a good approximation to the motion field. We now look at two methods of measuring optical flow fields.

1.6.2 Intensity based optical flow

The notion of optical flow was introduced by J. J. Gibson in *The Perception of the Visual World* [51], and has traditionally been calculated using methods based on image intensity derivatives. In order to show why image intensity gradient based optical flow measurements do not, in general, satisfy the differential epipolar equation we now give an outline of the means of their calculation.

The basis for the measurement of optical flow by image intensity gradients is the *optical flow constraint equation* [59]. This equation is based on the idea that if we represent the intensity of a point $\boldsymbol{p} = [x, y]^T$ in an image at time t by $I(x, y, t)$, then at time $t + \delta t$ the intensity $I(x + \delta x, y + \delta y, t + \delta t)$ at some neighboring point $\boldsymbol{p}' = [x + \delta x, y + \delta y]^T$ satisfies $I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)$. Taking into account that

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

we see that

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + O(\delta^2) = 0. \tag{1.4}$$

If we rearrange (1.4) and divide by δt then, in the limit as δt approaches 0, the $O(\delta^2)$ terms disappear and

$$\frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} = -\frac{\partial I}{\partial t}.$$

The derivatives $\partial I/\partial x$ and $\partial I/\partial y$ represent the spatial derivatives in each direction across the image and $\partial I/\partial t$ is the intensity change in a pixel over time. We thus have one equation in two unknowns. These two unknowns $\partial x/\partial t$ and $\partial y/\partial t$ represent the movement across the image plane, or optical flow. In order to solve this under-determined system a constraint based on the assumption that velocity changes smoothly across the image is usually used, although other constraints have been suggested [31,39,114,121]. For a more detailed introduction to gradient based optical flow, see Refs. [59, 70–72, 119], and for a survey of other optical flow measurement techniques, and particularly methods robust to the presence of noise in the data, see Refs. [6, 7, 11, 14].

The advantages of gradient based optical flow are that it is relatively easily obtained and that it provides an estimate of the motion field at every point in the image. The disadvantage is that there are a number of situations in which it is not necessarily a very good estimate [43, 96]. Any method of calculating the optical flow will suffer from certain conditions under which the estimated image motion do not correspond to the motion field. In the case of gradient based optical flow estimation there are three main situations which can cause problems: when there is a lack of scene texture, when there are surface discontinuities, and under particular lighting conditions.

If an object in an image has no surface texture and moves with no visible change in its boundary, then no gradient based optical flow will be measured. This is the case for a rotating smooth sphere for example [59]. The motion field would reflect the rotation of the sphere, but images of the sphere at time t_0 and time $t_0 + \delta t$ would be identical, so $\partial I/\partial t = 0$ and no intensity gradient based optical flow would be registered. A similar problem occurs when a change in the position of an object occurs, but the visible motion does not reflect the object motion. This is called the aperture problem. The simplest example of the aperture problem comes about when the image of the edge of an object moves in such a way that neither end of the edge is visible. This situation is illustrated in Figure 1.6. Only the component of object motion perpendicular to the visible edge is observable through the viewing aperture.

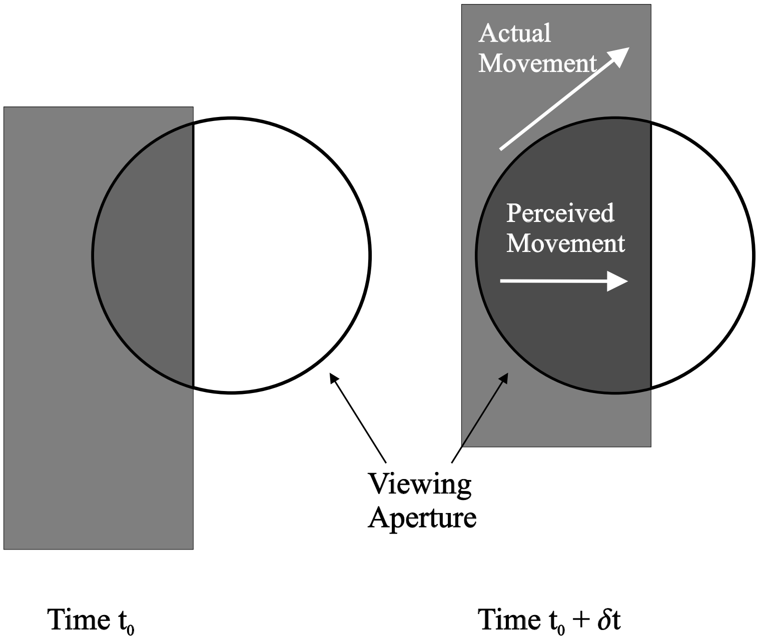

Figure 1.6: The aperture problem

Surface discontinuities in a scene pose a problem for intensity based optical flow due to the common assumption within the methods that image point velocity changes smoothly across an image. Surface discontinuities in the scene usually cause image point velocity discontinuities, thus violating this assumption. For possible solutions to this problem see Refs. [94, 97–100].

The final problem with intensity gradient based optical flow estimation is that of lighting. If altering the lighting of a scene produces visible changes in its image, the gradient based method will register optical flow despite the fact that no movement has taken place. More complex than this, however, is the problem of violating the assumption fundamental to the method that an object has the same apparent intensity over time. The apparent intensity of a matte surface is dependent only on the orientation and proximity of the surface to the light source. Neither of these factors need change as the camera moves through a rigid scene. Unfortunately most real surfaces are not perfectly matte, meaning that the apparent intensity may change with viewing angle, thus violating the assumption. For these reasons intensity gradient based optical flow may not provide a good representation of the required motion field, and thus may not satisfy the differential epipolar equation.

1.6.3 Feature based optical flow

Feature based optical flow is based upon detection and matching of specific features in successive images from a sequence. If this is carried out effectively then the result will be a good estimate of the motion field at the matched feature points. These feature points will, however, be sparsely spread across the images, as opposed to the dense field obtained from gradient based methods. In order to obtain a good estimate of the motion field it is necessary to design a detector of features that allows accurate determination of location that is orientation independent. The usual choice is to detect corners in the image for this reason. Edges or lines, for example, allow good spatial location perpendicular to their direction [29,88], but not parallel to their direction. Use of such features, therefore, requires the use of additional assumptions about the nature of the flow [26,27,30]. It is possible to track edge end points, but this is not a popular approach as end points are hard to detect reliably, and, if generated by occlusion, may change with viewpoint. It is the definition of what constitutes a feature that determines the density of the measured optical flow field. One means of increasing the density of the field is to describe every point in the image and match the resulting descriptors. This method has been developed with some success in the stereo case in Ref. [87] (using wavelets) although it has yet to be determined whether the results are accurate enough for self-calibration purposes.

Detecting which features in two images represent the same scene point has been termed the *correspondence problem.* The correspondence problem in the case of video sequences is typically simpler than in conventional stereo due to the small time difference, and therefore smaller displacement in feature position, between image pairs.

By tracking features through an image sequence we attain a discrete approximation to the motion field. The formulation of the gradient based scheme is differential in nature, and therefore may be said to be closer in essence to the differential epipolar equation. Ultimately, however, we are bound by the nature of image sequences to estimate discrete displacements whichever method is used.

1.7 Representing the camera parameters

Recall the differential epipolar equation

$$\boldsymbol{m}^T \boldsymbol{C} \boldsymbol{m} + \boldsymbol{m}^T \boldsymbol{W} \dot{\boldsymbol{m}} = 0,$$

within which the key parameters are encoded in the motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$. The exact nature of these matrices is described in Section 2.1.7. The motion field generated by a camera in motion will satisfy the differential epipolar equation

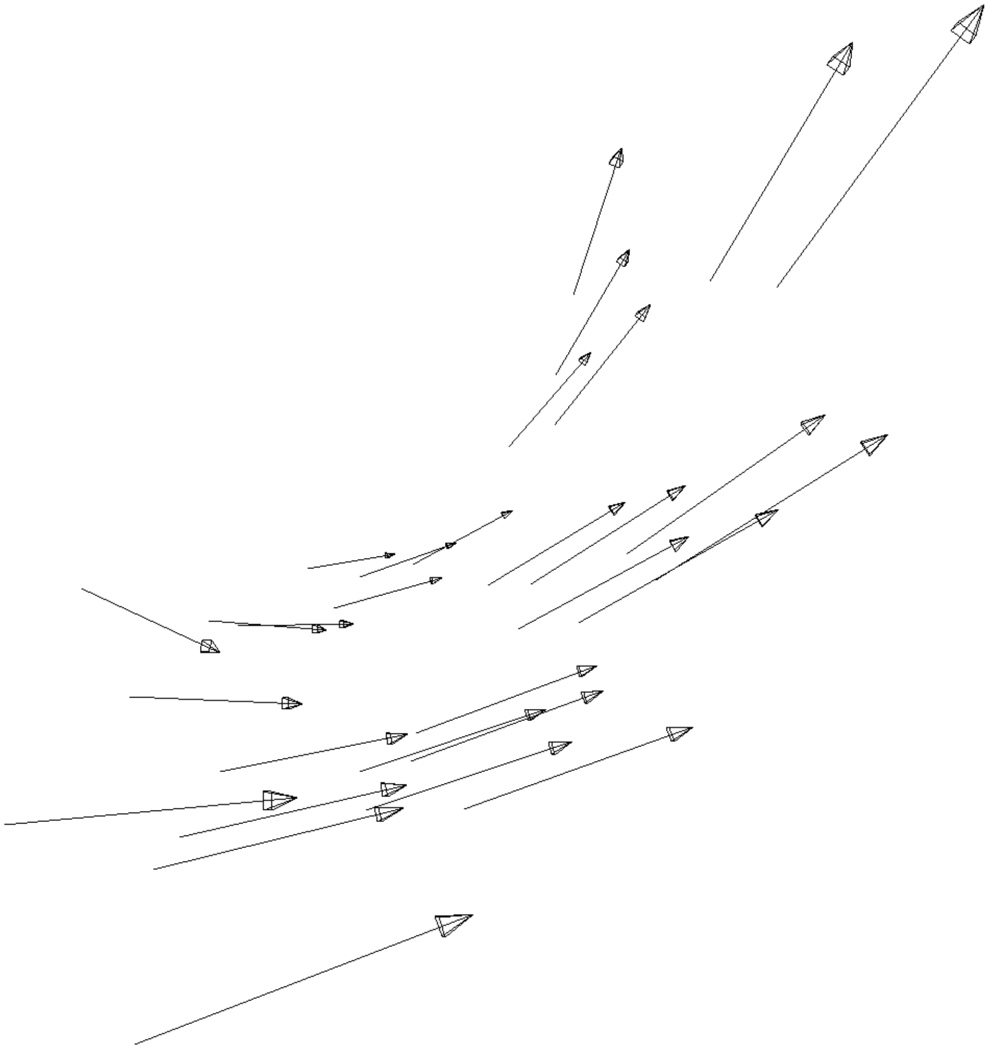

Figure 1.7: An optical flow field

for the corresponding motion matrices. The matrix $\boldsymbol{C}$ is symmetric and $\boldsymbol{W}$ antisymmetric, so they have 6 and 3 free parameters respectively:

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}.$$

The differential epipolar equation holds if both sides are multiplied through by a scalar, thus the motion matrices are defined only up to a scale factor. The effect of this scale indeterminacy is to reduce the degrees of freedom exhibited by the motion matrices by 1. Section 2.1.7 introduces an additional constraint on the forms of $\boldsymbol{C}$ and $\boldsymbol{W}$ thus reducing the system to 7 degrees of freedom.

Each flow vector pair $\{\boldsymbol{m}, \dot{\boldsymbol{m}}\}$ when substituted into the differential epipolar equation provides one constraint, which is linear in the elements of $\boldsymbol{C}$ and $\boldsymbol{W}$. By considering these constraints it is possible to recover the motion matrices (up to a common scale factor) and therefore some of the imaging parameters. The implication of the number of degrees of freedom of the system is that at least 7 optical flow vectors are necessary to recover the motion matrices. The process of recovering estimates of the motion matrices from optical flow fields is covered in detail in Sections 4 and 5.

The key parameters, describing a particular imaging situation, can be divided into two categories: those internal to the camera—the intrinsic parameters—and those describing the motion of the camera—the extrinsic parameters. In the case of general stereo vision these parameters represent the state of the cameras, and the geometric relationship between them. In the case of the differential epipolar equation these parameters represent the state and motion of the camera, and the change in the state and motion of the camera.

The motion matrices exhibit 7 degrees of freedom, thus we can recover at most 7 parameters, whether intrinsic or extrinsic. This recovery process is predicated on knowing the values of the remaining parameters. It is, however, not possible to recover all possible combinations of 7 parameters given those remaining. Section 2.5, however, describes a method for recovering the translation, rotation, focal length and rate of change in focal length of the camera. The rotation is described by 3 parameters, the translation by 2, and the focal length and its rate of change by 1 each giving a total of 7.

One measurement that cannot be recovered from the motion matrices is the rate of translation (i.e. the speed) of the camera. This is reflected in the fact that the translation recovered from the motion matrices is described by 2 parameters rather than 3, giving the direction, but not the magnitude of the vector. This inability to recover translational velocity is due to an inherent ambiguity in the problem. It is impossible to tell from a video sequence whether a camera is moving quickly towards a large object, or slowly towards a closer, but smaller object that is otherwise identical. For instance, footage generated by a camera moving towards a sphere gives no clue as to whether the sphere is planet sized, or football sized. The sequence could thus have been generated by a camera moving slowly towards a small sphere, or rapidly towards a larger one. The determination of the speed translation requires a priori information about the size of the objects in the scene, and is therefore indeterminable from such an image sequence alone. This corresponds to the baseline length indeterminacy in general stereo.

1.8 Outline

Chapter 1 describes the structure from motion problem and gives some background information. In Chapter 2 we derive the differential epipolar equation and show a method of self-calibration based on the equation. Chapter 3 describes a means of reconstructing the viewed scene from ego-motion information and optical flow. Methods for determining the trajectory of the camera over time are also presented. In Chapter 4 a number of methods of estimating the coefficients of the differential epipolar equation are presented and the merits of the various schemes discussed. Chapter 5 continues by deriving more tractable means of estimating the coefficients and considering the value of the gradient weighted least squares approach to the problem. Two means of filtering optical flow fields are presented in Chapter 6, one based on removing optical flow vectors and the other on improving their quality individually. Chapter 7 describes a number of experiments on real and synthetic image sequences. Our conclusions and suggestions for future research directions are presented in Chapter 8.

1.9 Contribution

A detailed analysis of the differential epipolar equation and its place in the literature is presented. This is followed by the development and testing of a means of reconstructing depth from optical flow and ego-motion information. Formulae are presented enabling the calculation of the change in camera position over an interval for both general camera motions and for a camera undergoing constant rotation. A number of methods for estimating the coefficients of the differential epipolar equation from optical flow are presented. The first methods rely on algebraic solution of systems of equations, following which a number of least squares methods are presented. A method based on ordinary least squares estimation, and therefore algebraic distance measures, is described first. Subsequently a number of methods based on orthogonal distances and total least squares principles are developed. As a part of this process two cost functions are derived and their accuracy tested. Direct minimisation of these full cost functions is shown to be impractical and more tractable approximations are developed. One of these algebraic approximations is shown to be equally justifiable in terms of the gradient weighted least squares approach. A method of minimising the approximated cost functions is developed based on Sampson's approach to finding the best fitting ellipse to a set of points. The statistical bias of Sampson's method is subsequently shown, and an alternative Newton-like method derived. The least median of squares approach is applied to the problem of estimating the coefficients of the differential epipolar equation. This leads to a method of eliminating gross errors from an optical flow field. A method of altering particular optical flow vectors such that they are better aligned with the ego-motion of a camera is presented. This method is shown to reduce the noise contamination in an observed optical flow field.

Chapter 2

The differential epipolar equation

This chapter provides two alternative derivations of the differential epipolar equation of Brooks et al. [17, 18]. The first exploits the epipolar equation from stereo vision, the second provides a derivation from first principles.

2.1 Differentiating the epipolar equation

A principle aim in general stereo vision is to recover 3-dimensional shape of an object or scene from two images. One method of achieving this manipulates the epipolar equation which we now describe.

2.1.1 The epipolar equation for calibrated cameras

Figure 2.1 shows that if we know the positions of the optical centres C and C', and of the image points $\boldsymbol{p}$ and $\boldsymbol{p}'$, we can recover the position of the scene point P as the intersection of the vectors from $\boldsymbol{p}$ through C and from $\boldsymbol{p}'$ through C'. The epipolar equation is simply an algebraic relationship between the location of a point in one image and that of the corresponding point in the other image. Encoded in this relationship, however, is the information necessary for reconstruction. Given enough point correspondences we can estimate this information in the form of the coefficients of the equation.

The epipolar equation for calibrated cameras, as derived by Longuet-Higgins in Ref. [77], relates the position of a point $\boldsymbol{p}$ in one image to the corresponding point $\boldsymbol{p}'$ in another via the equation

$$\boldsymbol{p}^T \boldsymbol{E}\, \boldsymbol{p}' = 0, \tag{2.1}$$

where $\boldsymbol{E}$ is the 3×3 *essential matrix*, and $\boldsymbol{p}$ and $\boldsymbol{p}'$ are the image points in homogeneous coordinates (having third elements of unity). The essential matrix is decomposable into two matrices $\boldsymbol{R}$ and $\boldsymbol{T}$ which represent the rotation and translation of the right camera relative to the left

$$\boldsymbol{E} = \boldsymbol{T}\, \boldsymbol{R}. \tag{2.2}$$

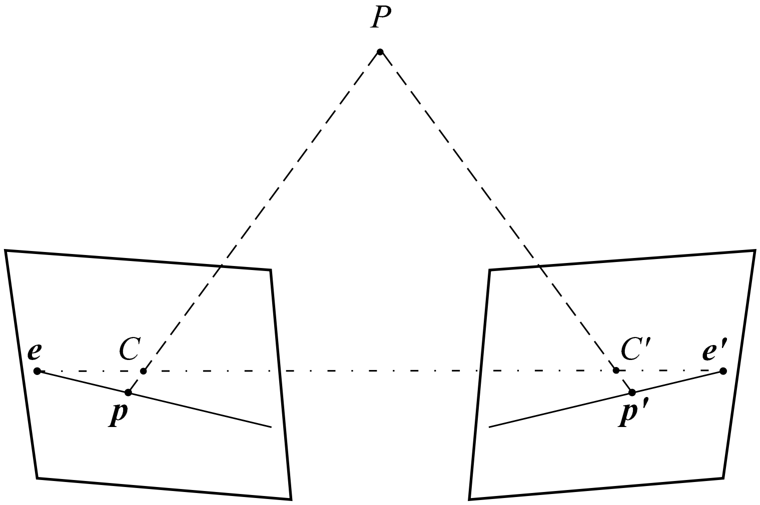

Figure 2.1: Epipolar geometry

If we define the baseline vector $\boldsymbol{t}$, connecting the optical centres of the two cameras (C to C') such that

$$\boldsymbol{t} = [t_1, t_2, t_3]^T ,$$

then the associated translation matrix, $\boldsymbol{T}$, is then given by

$$\boldsymbol{T} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} .$$

Note that $\boldsymbol{T}$ is antisymmetric and that $\boldsymbol{t} \times \boldsymbol{x} = \boldsymbol{T}\boldsymbol{x}$ for any vector $\boldsymbol{x}$.

The rotation matrix $\boldsymbol{R}$ describes the rotation of the right camera relative to the left and takes the form

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where α, β and γ represent the rotations about the x, y and z axes respectively, using the right hand rule to determine direction. The matrix $\boldsymbol{R}$ is, by definition, orthogonal, and of determinant 1; that is,

$$\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I} \text{ and } |\boldsymbol{R}| = 1.$$

The matrices $\boldsymbol{R}$ and $\boldsymbol{T}$ describe only the relative orientation of the two cameras. For the purposes of this analysis it is assumed that the internal geometry of the cameras is known, and therefore that measurements taken in the image plane may be used to determine the direction of rays through the optical centre of the camera.

Returning to Figure 2.1, the epipoles ($\boldsymbol{e}$ and $\boldsymbol{e}'$) of a pair of cameras are the points at which the ray passing through the two optical centres intersect the image planes. For any scene point P, the plane that passes through P, C and C' must pass through these two epipoles. We can define this plane knowing only $\boldsymbol{p}$, C and C'. The intersection of this plane with the image plane of the right camera defines a line on which the point $\boldsymbol{p}'$ must lie. On this basis, for any point in the left image $\boldsymbol{p}$, we can specify a line in the right image, passing through $\boldsymbol{e}'$, on which $\boldsymbol{p}'$ must fall. Lines of this form are called epipolar lines. There is of course no theoretical distinction between the left and right cameras so it is equally possible, given $\boldsymbol{p}'$, to define a line in the left image plane on which $\boldsymbol{p}$ must lie.

The above relies on the fact that the cameras are calibrated, so it is possible to interpret $\boldsymbol{p}$ and $\boldsymbol{p}'$ as vectors through C and C'. We know therefore that the vectors $\boldsymbol{p}$ and $\boldsymbol{p}'$ and the baseline vector $\boldsymbol{t}$ lie in the same plane. By definition the cross product of two vectors returns a vector perpendicular to both, and the inner product of two perpendicular vectors is 0. Three vectors are therefore in the same plane if the inner product of the first vector and the cross product of the second and third is 0. That is, coplanarity is proven if

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{a}^T(\boldsymbol{b} \times \boldsymbol{c}) = 0$$

where $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{c}$ are the three vectors in question. The vectors $\boldsymbol{p}$ and $\boldsymbol{p}'$ are defined in terms of the two camera based coordinate frames. Any comparison of the two requires that we represent them both in one frame, and we choose that of the left camera. The transition from the frame of the right camera to that of the left requires only that we multiply the vector by $\boldsymbol{R}$. The vectors $\boldsymbol{p}$, $\boldsymbol{R}\boldsymbol{p}'$ and $\boldsymbol{t}$ must therefore be coplanar so

$$\boldsymbol{p}^T(\boldsymbol{t} \times \boldsymbol{R}\boldsymbol{p}') = 0.$$

By the definition of $\boldsymbol{T}$ we have $\boldsymbol{t} \times \boldsymbol{R}\boldsymbol{p}' = \boldsymbol{T}\,\boldsymbol{R}\,\boldsymbol{p}'$ and so

$$\boldsymbol{p}^T(\boldsymbol{T}\,\boldsymbol{R}\,\boldsymbol{p}) = 0.$$

We see, therefore, from the definition of $\boldsymbol{E}$ in equation (2.2), that

$$\boldsymbol{p}^T\boldsymbol{E}\,\boldsymbol{p}' = 0.$$

2.1.2 The epipolar equation for uncalibrated cameras

Camera calibration is the process of measuring the internal geometry of a camera by taking images of scene points with known locations, and comparing these to the corresponding image point locations. To fully calibrate a camera thus requires both effort and apparatus. Once this calibration has taken place for both cameras it is possible to represent image points in a way that is camera independent. It is for points represented in this manner that the calibrated epipolar equation (2.1) holds. The epipolar equation for uncalibrated cameras is of the same form as (2.1); however, the essential matrix is replaced by the *fundamental matrix* $\boldsymbol{F}$, so

$$\boldsymbol{m}^T \boldsymbol{F} \boldsymbol{m}' = 0, \tag{2.3}$$

where $\boldsymbol{m}$ and $\boldsymbol{m}'$ (again in homogeneous coordinates) represent corresponding points in the images obtained by left and right cameras, respectively. The difference between the two epipolar equations is that the fundamental matrix, $\boldsymbol{F}$, embodies both extrinsic and intrinsic parameters. This means that $\boldsymbol{m}$ and $\boldsymbol{m}'$ in equation (2.3) refer to the uncalibrated rather than the calibrated image feature positions, which allows the use of uncalibrated, rather than calibrated cameras.

Given that our measurements are image based, it is useful to adopt an image-related coordinate frame Γ_{i}, with origin O and basis of vectors $\{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2\}$, in the image plane. It is natural to align the $\boldsymbol{\epsilon}_i$ along the sides of pixels and take one of the four corners of the rectangular image boundary for O. Suppose that a point in the image plane has coordinates $\boldsymbol{p} = [p_1, p_2, -f]^T$ and $[m_1, m_2]^T$ relative to the camera and image based frames Γ_{c} and Γ_{i} respectively. If $[m_1, m_2]^T$ is represented in homogeneous coordinates as $\boldsymbol{m} = [m_1, m_2, 1]^T$, then the relation between $\boldsymbol{p}$ and $\boldsymbol{m}$ can be conveniently written as

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{m}, \tag{2.4}$$

where $\boldsymbol{A}$ is a 3×3 invertible matrix called the *intrinsic-parameter matrix.* If we assume, for simplicity, that the camera has square pixels, that $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are also used as basis vectors in the camera frame, and that $[i_1, i_2]^T$ is the Γ_{i} based coordinate representation of the principal point D, then $\boldsymbol{A}$ takes the form

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -i_1 \\ 0 & 1 & -i_2 \\ 0 & 0 & -f \end{bmatrix}.$$

The matrix $\boldsymbol{A}$ takes a more complicated form if we relax the restrictions on Γ_{c} and Γ_{i} (see Ref. [47] for a more detailed description).

The fundamental matrix represents both the intrinsic and extrinsic characteristics of the particular stereo camera setup. With the intrinsic parameter matrix $\boldsymbol{A}$ as described above, the fundamental matrix may be represented as

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{A}^T \boldsymbol{E} \boldsymbol{A}' && (2.5) \\ &= \boldsymbol{A}^T \boldsymbol{T} \boldsymbol{R} \boldsymbol{A}'. && (2.6) \end{aligned}$$

It follows from equations (2.3) and (2.5) that

$$\boldsymbol{m}\boldsymbol{A}^T \boldsymbol{T} \boldsymbol{R} \boldsymbol{A}'\boldsymbol{m}' = 0$$

and, given (2.4), that

$$\boldsymbol{p}^T \boldsymbol{T} \boldsymbol{R} \boldsymbol{p}' = 0.$$

The relationship between the two camera coordinate frames is supplemented by the presence of the matrix $\boldsymbol{A}$ making it complicated to represent $\boldsymbol{m}$ and $\boldsymbol{m}'$ as vectors from their respective optical centres. This in turn makes the explanation of epipolar geometry carried out for the calibrated case less practical for the uncalibrated case. It is, however, useful to note that if a point $\boldsymbol{m}$ falls on the line $\boldsymbol{l}$ then $\boldsymbol{m}^T \boldsymbol{l} = 0$. So if we see $\boldsymbol{F}$ as describing the projective linear transformation from point $\boldsymbol{m}'$ in the right image to line $\boldsymbol{l}$ in the left then

$$\boldsymbol{l} = \boldsymbol{F} \boldsymbol{m}'.$$

We already know that $\boldsymbol{m}$ lies on $\boldsymbol{l}$ so

$$\boldsymbol{m}^T \boldsymbol{F} \boldsymbol{m}' = 0.$$

In Section 1.4 we defined the key parameters which describe the internal state of the cameras and the geometric relationship between them. Given sufficiently many, non-degenerate corresponding points, it is sometimes possible, via a process of self-calibration, to determine various combinations of the key parameters [48, 92]. Using corresponding points extracted from a single image pair, at most 7 key parameters may be determined. These might, for example, comprise 5 relative orientation parameters and two focal lengths (see Refs. [55, 102]).

2.1.3 The time dependent epipolar equation

Our interest is in determining structure from motion, and therefore in representing the motion of one camera rather than the relationship between two. Towards this goal we now introduce into the epipolar equation for uncalibrated cameras (equation (2.3)) a dependency on time in order to develop equations based on differential forms. Our aim is to develop closed-form expressions for the changes in key parameters as a function of optical flow. This section presents the work of Brooks et al in Ref. [17] and Ref. [18], which, in turn, can be seen as a recasting of the research of Viéville and Faugeras [135] into an analytical framework.

If we allow $\boldsymbol{m}$, $\boldsymbol{m}'$ and $\boldsymbol{F}$ to vary over time, we need to add a dependency upon time to (2.3), which leads to *the time dependent epipolar equation for uncalibrated stereo cameras*

$$\boldsymbol{m}^T(t) \boldsymbol{F}(t)\, \boldsymbol{m}'(t) = 0. \tag{2.7}$$

This equation is simply an instance of the epipolar equation (2.3) at a particular time. We thus describe the key parameters of the stereo setup as they change over time. Note that consideration of time will be of no benefit if we have a pair of cameras in a fixed relationship, with unchanged relative orientation and intrinsic parameters. This applies even if the stereo cameras are in motion relative to some global frame, as each camera remains stationary relative to the other. In this situation the key parameters, and thus $\boldsymbol{F}$, will not change over time.

If we assume that the cameras are not in a fixed relationship, but that they are free to move independently, equation (2.7) then offers the possibility of recovering some of the key parameters as a function of time. Note, therefore, that this implicitly conveys information about the motion of one camera relative to another. Again, however, no information is available about the motion of either camera relative to a fixed frame of reference.

We now consider the nature of the epipolar equation arising from images taken by a single camera at successive time instants. The limiting case, where the time interval between the acquisition of the images tends to zero, might then permit computation of both the ego-motion and the intrinsic parameters of the camera. Note that results pertaining to camera ego-motion and a stationary scene are equally applicable to a stationary camera and a moving, rigid scene.

In contemplating the limiting case in which the time difference between images tending to zero (as in Ref. [135]) we see immediately, that the following equation holds little value:

$$\boldsymbol{m}^T(t)\boldsymbol{F}(t)\,\boldsymbol{m}(t) = 0. \tag{2.8}$$

This deals merely with identical left and right images and points. In this situation, we will clearly have $\boldsymbol{F}(t) = \mathbf{0}$. In contrast we seek a fundamental matrix relating a pair of images captured at different times.

We now consider an alternative formulation of the time-dependent epipolar equation

$$\boldsymbol{m}^T(t_1)\,\boldsymbol{F}(t_1, t_2)\,\boldsymbol{m}(t_2) = 0. \tag{2.9}$$

This equation makes explicit the dependencies of the fundamental matrix $\boldsymbol{F}$. It is important to note here that the fundamental matrix associated with images obtained from a single camera (in contrast with that associated with a pair of cameras) is dependent upon *two* times. It is this that enables the derivative of $\boldsymbol{F}$ with respect to time to be defined. This equation has been termed *the time-dependent epipolar equation for an uncalibrated camera.* It is this equation which forms the basis for the subsequent analysis.

2.1.4 Differential forms of the time-dependent epipolar equation

We now confine our attention to (2.9), seeking differential forms that enable instantaneous changes in the key parameters to be related to instantaneous changes in positions of corresponding points.

Assume that a camera undergoes some arbitrary motion over a period of time, thereby generating an image stream. At two different times t_1 and t_2, equation (2.9) will constrain the relationship between the coordinates of the corresponding points and the image formation parameters bound up in $\boldsymbol{F}$. As t_1 and t_2 vary, we therefore expect that $\boldsymbol{F}(t_1, t_2)$ will also vary.

Observe that as $t_2 \to t_1$, then $\boldsymbol{F}(t_1, t_2) \to \mathbf{0}$. Nevertheless, the derivative of $\boldsymbol{F}(t_1, t_2)$ will at all times be defined, including at time $t_1 = t_2$. Of particular interest to us here is to determine the time-derivatives of $\boldsymbol{F}(t, t)$, for these will be central to the consideration of ego-motion of a single, moving camera.

We now introduce notation to simplify the representation of the derivative of the epipolar equation. The first and second derivatives of a single parameter function $f(t)$ with respect to t are represented as $\dot{f}(t)$ and $\ddot{f}(t)$ respectively. Given a function g of two times, we let

$$\begin{aligned} g(t) &= g(t, t), \\ \overset{\circ}{g}(t) &= \left.\frac{\partial g}{\partial t_2}(t_1, t_2)\right|_{(t_1, t_2) = (t, t)} \\ \overset{\circ\circ}{g}(t) &= \left.\frac{\partial^2 g}{\partial t_2^2}(t_1, t_2)\right|_{(t_1, t_2) = (t, t)} \end{aligned}$$

With this notation

$$\begin{aligned} \boldsymbol{T}(t) &= \mathbf{0}, \\ \boldsymbol{R}(t) &= \boldsymbol{I}, \end{aligned} \tag{2.10}$$

which immediately implies that

$$\boldsymbol{F}(t) = \mathbf{0}. \tag{2.11}$$

We now differentiate (2.9) with respect to t_2,

$$\boldsymbol{m}^T(t_1)\,\frac{\partial \boldsymbol{F}}{\partial t_2}(t_1, t_2)\,\boldsymbol{m}(t_2) + \boldsymbol{m}^T(t_1)\,\boldsymbol{F}(t_1, t_2)\,\dot{\boldsymbol{m}}(t_2) = 0,$$

whence, on letting $t_1 = t_2 = t$, we have

$$\boldsymbol{m}^T(t)\,\overset{\circ}{\boldsymbol{F}}(t)\,\boldsymbol{m}(t) + \boldsymbol{m}^T(t)\,\boldsymbol{F}(t)\,\dot{\boldsymbol{m}}(t) = 0.$$

Omitting the notational dependency on time, and using (2.11), we may rewrite this equation as

$$\boldsymbol{m}^T\,\overset{\circ}{\boldsymbol{F}}\,\boldsymbol{m} = 0. \tag{2.12}$$

This is the first differential form of the epipolar equation, as it has arisen by once differentiating (2.9).

We may now follow a similar path to obtain the second form. Differentiating equation (2.9) twice with respect to t_2, we obtain

$$\begin{aligned} &\boldsymbol{m}^T(t_1)\,\frac{\partial^2 \boldsymbol{F}}{\partial t_2^2}(t_1, t_2)\,\boldsymbol{m}(t_2) \\ &\qquad + 2\,\boldsymbol{m}^T(t_1)\,\frac{\partial \boldsymbol{F}}{\partial t_2}(t_1, t_2)\,\dot{\boldsymbol{m}}(t_2) + \boldsymbol{m}^T(t_1)\,\boldsymbol{F}(t_1, t_2)\,\ddot{\boldsymbol{m}}(t_2) = 0, \end{aligned}$$

thus, on letting $t_1 = t_2 = t$, we have

$$\boldsymbol{m}^T(t)\,\overset{\circ\circ}{\boldsymbol{F}}(t)\,\boldsymbol{m}(t) + 2\,\boldsymbol{m}^T(t)\,\overset{\circ}{\boldsymbol{F}}(t)\,\dot{\boldsymbol{m}}(t) + \boldsymbol{m}^T(t)\,\boldsymbol{F}(t)\,\ddot{\boldsymbol{m}}(t) = 0,$$

and accordingly

$$\boldsymbol{m}^T\,\overset{\circ\circ}{\boldsymbol{F}}\,\boldsymbol{m} + 2\,\boldsymbol{m}^T\,\overset{\circ}{\boldsymbol{F}}\,\dot{\boldsymbol{m}} = 0. \tag{2.13}$$

This is the second differential form of the epipolar equation. Note that this equation contains both location and velocity of an image point, but not its acceleration, $\ddot{\boldsymbol{m}}$ having fallen away in the derivation.

2.1.5 The matrices of relative orientation

If the matrices $\boldsymbol{R}$ and $\boldsymbol{T}$ are constructed as in Section 2.1.1 we observe that

$$\begin{aligned}
\overset{\circ}{\boldsymbol{T}}(t) &= \frac{\partial \boldsymbol{T}}{\partial t_1}\overset{\circ}{t}_1(t) + \frac{\partial \boldsymbol{T}}{\partial t_2}\overset{\circ}{t}_2(t) + \frac{\partial \boldsymbol{T}}{\partial t_3}\overset{\circ}{t}_3(t)\\
\overset{\circ}{\boldsymbol{R}}(t) &= \frac{\partial \boldsymbol{R}}{\partial \alpha}\overset{\circ}{\alpha}(t) + \frac{\partial \boldsymbol{R}}{\partial \beta}\overset{\circ}{\beta}(t) + \frac{\partial \boldsymbol{R}}{\partial \gamma}\overset{\circ}{\gamma}(t),
\end{aligned}$$

where the derivatives $\overset{\circ}{t}_1(t)$, $\overset{\circ}{t}_2(t)$, $\overset{\circ}{t}_3(t)$, $\overset{\circ}{\alpha}(t)$, $\overset{\circ}{\beta}(t)$ and $\overset{\circ}{\gamma}(t)$ are defined in terms of $t_1(t_1, t_2)$, $t_2(t_1, t_2)$, $t_3(t_1, t_2)$, $\alpha(t_1, t_2)$, $\beta(t_1, t_2)$ and $\gamma(t_1, t_2)$ as described in Section 2.1.4. Noting that $t_1(t) = t_2(t) = t_3(t) = \alpha(t) = \beta(t) = \gamma(t) = 0$, we are left with the simple forms

$$\overset{\circ}{\boldsymbol{T}}(t) = \begin{bmatrix} 0 & -\overset{\circ}{t}_3(t) & \overset{\circ}{t}_2(t) \\ \overset{\circ}{t}_3(t) & 0 & -\overset{\circ}{t}_1(t) \\ -\overset{\circ}{t}_2(t) & \overset{\circ}{t}_1(t) & 0 \end{bmatrix} \tag{2.14}$$

$$\overset{\circ}{\boldsymbol{R}}(t) = \begin{bmatrix} 0 & -\overset{\circ}{\gamma}(t) & \overset{\circ}{\beta}(t) \\ \overset{\circ}{\gamma}(t) & 0 & -\overset{\circ}{\alpha}(t) \\ -\overset{\circ}{\beta}(t) & \overset{\circ}{\alpha}(t) & 0 \end{bmatrix} \tag{2.15}$$

We observe that both $\overset{\circ}{\boldsymbol{T}}$ and $\overset{\circ}{\boldsymbol{R}}$ are anti-symmetric. Additionally, matrix $\overset{\circ\circ}{\boldsymbol{T}}$ is readily shown to be anti-symmetric.

2.1.6 Elaborating the Second Differential Form

We now seek to determine $\overset{\circ}{\boldsymbol{F}}(t)$ and $\overset{\circ\circ}{\boldsymbol{F}}(t)$ in terms of the component matrices of $\boldsymbol{F}$. The fundamental matrix for a single camera $\boldsymbol{F}(t_1, t_2)$ may be expressed as

$$\boldsymbol{F}(t_1, t_2) = \boldsymbol{A}^T(t_1)\boldsymbol{E}(t_1, t_2)\boldsymbol{A}(t_2). \tag{2.16}$$

Differentiating this equation, and taking into account that

$$\boldsymbol{E}(t) = \boldsymbol{0}, \tag{2.17}$$

we obtain

$$\overset{\circ}{\boldsymbol{F}}(t) = \boldsymbol{A}^T(t)\overset{\circ}{\boldsymbol{T}}(t)\boldsymbol{A}(t), \tag{2.18}$$

due to the nature of the matrix $\boldsymbol{E}$. We observe, therefore, that $\overset{\circ}{\boldsymbol{F}}(t)$ is dependent only upon the values of the intrinsic parameters and the derivatives of the translation parameters. The rotation parameters and their derivatives are not represented.

Differentiating once more, and dropping henceforth the dependency on t, we obtain

$$\overset{\circ\circ}{\boldsymbol{F}} = \boldsymbol{A}^T(\overset{\circ\circ}{\boldsymbol{T}} + 2\overset{\circ}{\boldsymbol{T}}\overset{\circ}{\boldsymbol{R}})\boldsymbol{A} + 2\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\dot{\boldsymbol{A}}, \tag{2.19}$$

and therefore that

$$\begin{aligned} \boldsymbol{m}^T\overset{\circ\circ}{\boldsymbol{F}}\boldsymbol{m} = {} & \boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ\circ}{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{m} \\ & + 2\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\overset{\circ}{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m} \\ & + 2\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\dot{\boldsymbol{A}}\boldsymbol{m}. \end{aligned} \tag{2.20}$$

Since $\overset{\circ\circ}{\boldsymbol{T}}$ is antisymmetric, it follows that

$$\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ\circ}{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{m} = 0.$$

Therefore (2.20) can be rewritten in the form

$$\boldsymbol{m}^T\overset{\circ\circ}{\boldsymbol{F}}\boldsymbol{m} = 2\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\overset{\circ}{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m} + 2\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\dot{\boldsymbol{A}}\boldsymbol{m}. \tag{2.21}$$

Equation (2.13) thus becomes

$$\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\overset{\circ}{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\dot{\boldsymbol{A}}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\boldsymbol{A}\dot{\boldsymbol{m}} = 0. \tag{2.22}$$

Even though this equation incorporates the first and second derivatives of the fundamental matrix, we observe that no second derivatives of its component matrices survive the elaboration. We also note that, in the event that the intrinsic parameters are fixed, this equation reduces to

$$\boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\overset{\circ}{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{A}^T\overset{\circ}{\boldsymbol{T}}\boldsymbol{A}\dot{\boldsymbol{m}} = 0. \tag{2.23}$$

2.1.7 An Alternative Second Differential Form

We now derive an alternative form of (2.22) that is more amenable to numerical solution.

If we let

$$\boldsymbol{B} = \dot{\boldsymbol{A}}\boldsymbol{A}^{-1} \tag{2.24}$$

then (2.22) can rewritten as

$$\boldsymbol{m}^T\boldsymbol{A}^T\mathring{\boldsymbol{T}}(\mathring{\boldsymbol{R}} + \boldsymbol{B})\boldsymbol{A}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{A}^T\mathring{\boldsymbol{T}}\boldsymbol{A}\dot{\boldsymbol{m}} = 0. \tag{2.25}$$

Given a matrix $\boldsymbol{X}$, denote by $\boldsymbol{X}_{\text{sym}}$ and $\boldsymbol{X}_{\text{asym}}$ the symmetric and antisymmetric parts of $\boldsymbol{X}$ defined, respectively, by

$$\boldsymbol{X}_{\text{sym}} = \frac{1}{2}(\boldsymbol{X} + \boldsymbol{X}^T), \tag{2.26}$$

and

$$\boldsymbol{X}_{\text{asym}} = \frac{1}{2}(\boldsymbol{X} - \boldsymbol{X}^T). \tag{2.27}$$

By definition we see that

$$\boldsymbol{m}^T\boldsymbol{X}_{\text{sym}}\boldsymbol{m} = \boldsymbol{m}^T\boldsymbol{X}\boldsymbol{m}, \text{ and} \tag{2.28}$$

and

$$\boldsymbol{m}^T\boldsymbol{X}_{\text{asym}}\boldsymbol{m} = 0. \tag{2.29}$$

Since $\mathring{\boldsymbol{R}}$ and $\mathring{\boldsymbol{T}}$ are antisymmetric, we get

$$(\mathring{\boldsymbol{T}}\mathring{\boldsymbol{R}})_{\text{sym}} = \frac{1}{2}(\mathring{\boldsymbol{T}}\mathring{\boldsymbol{R}} + \mathring{\boldsymbol{R}}\mathring{\boldsymbol{T}}), \tag{2.30}$$

and

$$(\mathring{\boldsymbol{T}}\boldsymbol{B})_{\text{sym}} = \frac{1}{2}(\mathring{\boldsymbol{T}}\boldsymbol{B} - \boldsymbol{B}^T\mathring{\boldsymbol{T}}). \tag{2.31}$$

If we denote the symmetric part of $\boldsymbol{A}^T\mathring{\boldsymbol{T}}(\mathring{\boldsymbol{R}} + \boldsymbol{B})\boldsymbol{A}$ by $\boldsymbol{C}$, then, we see that

$$\boldsymbol{C} = \frac{1}{2}\boldsymbol{A}^T(\mathring{\boldsymbol{T}}\mathring{\boldsymbol{R}} + \mathring{\boldsymbol{R}}\mathring{\boldsymbol{T}} + \mathring{\boldsymbol{T}}\boldsymbol{B} - \boldsymbol{B}^T\mathring{\boldsymbol{T}})\boldsymbol{A}. \tag{2.32}$$

Let

$$\boldsymbol{W} = \boldsymbol{A}^T\mathring{\boldsymbol{T}}\boldsymbol{A}. \tag{2.33}$$

On account of (2.25), (2.28) and (2.32), we can write

$$\boldsymbol{m}^T\boldsymbol{C}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{W}\dot{\boldsymbol{m}} = 0. \tag{2.34}$$

A constraint similar to that of (2.34), termed the first-order expansion of the fundamental motion equation, is derived by Viéville and Faugeras in Ref. [135]. In contrast with the above, however, it is derived using Taylor series expansions and approximations. The reader is also referred to Ref. [66], where a similar derivation (though not involving any special differentiation procedure) is presented in the context of images formed on a sphere.

It is important to realise that by applying equation (2.34), the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ can be determined, to within a common scalar factor, directly from image data. So if, at any given instant t, we supply sufficiently many independent $\boldsymbol{m}_i$ and $\dot{\boldsymbol{m}}_i$, then $\boldsymbol{C}$ and $\boldsymbol{W}$ can be determined, up to a common scalar factor, from the following system of equations:

$$\boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_i + \boldsymbol{m}_i^T \boldsymbol{W} \dot{\boldsymbol{m}}_i = 0 \ \ (i = 1 \ldots n). \tag{2.35}$$

These equations are linear in $\boldsymbol{C}$ and $\boldsymbol{W}$.

2.2 Derivation from first principles

Having shown the relationship between the differential epipolar equation and the epipolar equation for uncalibrated cameras, we now give a derivation of the differential epipolar equation based more directly on the nature of the projection of scene motion onto the image plane. This alternate derivation provides a perspective which is useful in understanding the relationships between the various coordinate frames used, and ultimately an avenue for reconstructing the viewed scene.

2.2.1 Scene motion in the camera frame

To describe the position, orientation and internal geometry of the camera as well as the image formation process, it is convenient to introduce two coordinate frames. We define a Cartesian ("world") coordinate frame Γ_{w} whose scene configuration will be fixed throughout. We also define an independent Cartesian coordinate frame Γ_{c} associated with the camera. This frame has origin C and basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ of unit orthogonal vectors, so that C coincides with the optical centre, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ span the image plane, and $\boldsymbol{e}_3$ lies along the optical axis (see Figure 1.2). By ensuring that Γ_{c} and Γ_{w} are equi-oriented we guarantee that the value of the cross product of two vectors is independent of whether the basis of unit orthogonal vectors associated with Γ_{w} or that associated with Γ_{c} is used for calculation. For reasons of tractability, C will be identified with the point in $\boldsymbol{R}^3$ formed by the coordinates of the optical centre of the camera relative to Γ_{w}. Similarly, for each $i \in \{1, 2, 3\}$, $\boldsymbol{e}_i$ will be identified with the point in $\boldsymbol{R}^3$ formed by the components of $\boldsymbol{e}_i$ relative to the vector basis of Γ_{w}.

Suppose that the camera undergoes smooth motion with respect to Γ_{w}. At each time instant t, the location of the camera relative to Γ_{w} is given by

$$(C(t), \boldsymbol{e}_1(t), \boldsymbol{e}_2(t), \boldsymbol{e}_3(t)) \in \boldsymbol{R}^3 \times \boldsymbol{R}^3 \times \boldsymbol{R}^3 \times \boldsymbol{R}^3.$$

The motion of the camera is then described by the differentiable function

$$t \mapsto (C(t), \boldsymbol{e}_1(t), \boldsymbol{e}_2(t), \boldsymbol{e}_3(t)).$$

The derivative $\dot{C}(t)$ captures the instantaneous translational velocity of the camera relative to Γ_{w} at t. Expanding this derivative with respect to the basis $\{\boldsymbol{e}_i(t)\}_{1\leq i\leq 3}$

$$\dot{C}(t) = \sum_i t_i(t)\boldsymbol{e}_i(t) \tag{2.36}$$

defines $\boldsymbol{v}(t) = [t_1(t), t_2(t), t_3(t)]^T$. This vector represents the instantaneous translational velocity of the camera relative to Γ_{c} at t. Each of the derivatives $\dot{\boldsymbol{e}}_i(t)$ can be expanded in a similar fashion yielding

$$\dot{\boldsymbol{e}}_i(t) = \sum_j \omega_{ji}(t)\boldsymbol{e}_j(t). \tag{2.37}$$

Let P be a point in space. The location of P relative to Γ_{c} can be expressed in terms of a coordinate vector $\boldsymbol{z} = [z_1, z_2, z_3]^T$ determined from the equation

$$P = \sum_i z_i\boldsymbol{e}_i + C. \tag{2.38}$$

Suppose that P is static with respect to Γ_{w}. As the camera moves, the position of P relative to Γ_{c} will change accordingly and will be recorded in the function $t \mapsto \boldsymbol{z}(t)$. This function satisfies an equation reflecting the kinematics of the moving camera. We derive this equation next.

Differentiating (2.38) and taking into account that $\dot{P} = \mathbf{0}$, we obtain

$$\sum_i (\dot{z}_i\boldsymbol{e}_i + z_i\dot{\boldsymbol{e}}_i) + \dot{C} = \mathbf{0}.$$

From this and equations (2.36) and (2.37) we see that

$$\dot{\boldsymbol{z}} + \boldsymbol{\omega} \times \boldsymbol{z} + \boldsymbol{v} = \mathbf{0}. \tag{2.39}$$

We define an operator such that, for some vector $\boldsymbol{a} = [a_1, a_2, a_3]^T$,

$$\widehat{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{2.40}$$

Using this definition we write equation (2.39) as

$$\dot{\boldsymbol{z}} + \widehat{\boldsymbol{\omega}}\boldsymbol{z} + \boldsymbol{v} = \mathbf{0}. \tag{2.41}$$

2.2.2 Differential epipolar equation: the second form

Under our camera model the image is formed by projection of light from the viewed scene, through an aperture at C, onto the image plane (again, see Figure 1.2). In coordinates relative to the camera based frame Γ_{c} the image plane is described by all points with depth $-f$ (so $\{\boldsymbol{z} \in \boldsymbol{R}^3 \colon z_3 = -f\}$), where f is the focal length.

Let the vector $\boldsymbol{z}$ describe the coordinates of a particular scene point P relative to Γ_{c}. The location of the image of P is given by a vector $\boldsymbol{p}$ as defined in equation (2.4)

$$\boldsymbol{p} = -f\frac{\boldsymbol{z}}{z_3}. \tag{2.42}$$

Suppose again that P is static and the camera moves with respect to Γ_{w}. The evolution of the image of P will then be described by the function $t \mapsto \boldsymbol{p}(t)$. This function is subject to a constraint deriving from equation (2.41). We proceed to determine this constraint.

First, note that (2.42) can be equivalently rewritten as

$$\boldsymbol{z} = -\frac{z_3\boldsymbol{p}}{f}, \tag{2.43}$$

which immediately leads to

$$\dot{\boldsymbol{z}} = \frac{z_3\dot{f} - \dot{z}_3 f}{f^2}\boldsymbol{p} - \frac{z_3}{f}\dot{\boldsymbol{p}}. \tag{2.44}$$

Next, applying the matrix $\widehat{\boldsymbol{v}}$ to both sides of (2.41) and noting that $\widehat{\boldsymbol{v}}\boldsymbol{v} = \mathbf{0}$, we get

$$\widehat{\boldsymbol{v}}\dot{\boldsymbol{z}} + \widehat{\boldsymbol{v}}\widehat{\boldsymbol{\omega}}\boldsymbol{z} = \mathbf{0}.$$

Now, in view of (2.43) and (2.44),

$$\frac{z_3\dot{f} - \dot{z}_3 f}{f^2}\widehat{\boldsymbol{v}}\boldsymbol{p} - \frac{z_3}{f}\widehat{\boldsymbol{v}}\dot{\boldsymbol{p}} - \frac{z_3}{f}\widehat{\boldsymbol{v}}\widehat{\boldsymbol{\omega}}\boldsymbol{p} = \mathbf{0}. \tag{2.45}$$

In view of the antisymmetry of $\widehat{\boldsymbol{v}}$, we have $\boldsymbol{p}^T\widehat{\boldsymbol{v}}\boldsymbol{p} = 0$. Applying $\boldsymbol{p}^T$ to both sides of equation (2.45) we obtain

$$\boldsymbol{p}^T\widehat{\boldsymbol{v}}\dot{\boldsymbol{p}} + \boldsymbol{p}^T\widehat{\boldsymbol{v}}\widehat{\boldsymbol{\omega}}\boldsymbol{p} = 0. \tag{2.46}$$

This is the sought-after differential epipolar equation.

The differential epipolar equation is not the only constraint that can be imposed on functions of the form $t \mapsto \boldsymbol{p}(t)$. As shown by Åström and Heyden [4], for every $n \geq 2$, such functions satisfy an nth order differential equation that reduces to the differential epipolar equation when $n = 2$. The nth equation in the series is the infinitesimal version of the analogue of the standard epipolar equation satisfied by a set of corresponding points, identified within a sequence of n images, depicting a common scene point. This work rests solely on the differential epipolar equation which is the simplest of these equations.

2.2.3 Relating the two forms of the differential epipolar equation

The two derivations above have provided two different forms of the differential epipolar equation. We now show the relationship between them by making the transition from equation (2.46) to equation (2.21)

By their definitions in equations (2.40), (2.14) and (2.14) we see that

$$\widehat{\boldsymbol{v}} = \mathring{\boldsymbol{T}} \quad \text{and} \quad \widehat{\boldsymbol{\omega}} = \mathring{\boldsymbol{R}}. \tag{2.47}$$

It follows from our definition of $\boldsymbol{p}$ in terms of $\boldsymbol{m}$ in equation (2.4) that the derivative of $\boldsymbol{p}$ is

$$\dot{\boldsymbol{p}} = \dot{\boldsymbol{A}}\boldsymbol{m} + \boldsymbol{A}\dot{\boldsymbol{m}}. \tag{2.48}$$

By equations (2.47) and (2.48)

$$\begin{aligned} \boldsymbol{p}^T\widehat{\boldsymbol{v}}\dot{\boldsymbol{p}} &= \boldsymbol{m}^T\boldsymbol{A}^T\mathring{\boldsymbol{T}}\dot{\boldsymbol{A}}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{A}^T\mathring{\boldsymbol{T}}\boldsymbol{A}\dot{\boldsymbol{m}} \\ \boldsymbol{p}^T\widehat{\boldsymbol{v}}\widehat{\boldsymbol{\omega}}\boldsymbol{p} &= \boldsymbol{m}^T\boldsymbol{A}^T\mathring{\boldsymbol{T}}\mathring{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m}, \end{aligned}$$

so (2.46) can be rewritten as

$$\boldsymbol{m}^T\boldsymbol{A}\mathring{\boldsymbol{T}}\boldsymbol{A}\dot{\boldsymbol{m}} + \boldsymbol{m}^T\big(\boldsymbol{A}^T\mathring{\boldsymbol{T}}\mathring{\boldsymbol{R}}\boldsymbol{A} + \boldsymbol{A}^T\mathring{\boldsymbol{T}}\dot{\boldsymbol{A}}\big)\boldsymbol{m} = 0,$$

which is equation (2.21).

2.3 A projective form of the motion matrices

In view of its definition in (2.33) and the antisymmetry of $\mathring{\boldsymbol{T}}$, we see that $\boldsymbol{W}$ is antisymmetric, and so $\boldsymbol{W} = \widehat{\boldsymbol{w}}$ for some vector $\boldsymbol{w} = [w_1, w_2, w_3]^T$ (that is $\boldsymbol{w}$ as opposed to $\widehat{\boldsymbol{\omega}}$). $\boldsymbol{C}$ is symmetric, and hence it is uniquely determined by the entries $c_{11}, c_{12}, c_{13}, c_{22}, c_{23}, c_{33}$. Let $\boldsymbol{C} : \boldsymbol{W}$ be the *joint projective form* of $\boldsymbol{C}$ and $\boldsymbol{W}$, that is, the point in the 8-dimensional real projective space $\mathbb{P}^8$ with homogeneous coordinates given by the composite ratio

$$\boldsymbol{C} : \boldsymbol{W} = (c_{11} : c_{12} : c_{13} : c_{22} : c_{23} : c_{33} : w_1 : w_2 : w_3).$$

Clearly, $\lambda\boldsymbol{C} : \lambda\boldsymbol{W} = \boldsymbol{C} : \boldsymbol{W}$ for any non-zero scalar λ. Thus knowing $\boldsymbol{C} : \boldsymbol{W}$ amounts to knowing $\boldsymbol{C}$ and $\boldsymbol{W}$ to within a common scalar factor. We see from this that a normalising condition is necessary to compare different $\boldsymbol{C} : \boldsymbol{W}$ pairs, but that the exact form of the normalising condition is somewhat arbitrary.

2.4 A cubic constraint on the motion matrices

We now show that $\boldsymbol{C} : \boldsymbol{W}$ lies on a hypersurface of $\mathbb{P}^8$, so rather than being able to take any value from this space, $\boldsymbol{C} : \boldsymbol{W}$ is confined to a 7-dimensional manifold. We thus define a constraint on the possible forms of the motion matrices.

By the definitions of the motion matrices in equations (2.32) and (2.33), we know that

$$\boldsymbol{C} = \frac{1}{2}\big[\boldsymbol{W}\boldsymbol{A}^{-1}(\mathring{\boldsymbol{R}} + \boldsymbol{B})\boldsymbol{A} + \boldsymbol{A}^T(\mathring{\boldsymbol{R}} - \boldsymbol{B}^T)(\boldsymbol{A}^T)^{-1}\boldsymbol{W}\big]. \tag{2.49}$$

Taking into account that $\boldsymbol{w}^T\boldsymbol{W} = \mathbf{0}$ and $\boldsymbol{W}\boldsymbol{w} = \mathbf{0}$, we see that

$$\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0. \tag{2.50}$$

The left-hand side is a homogeneous polynomial of degree 3 in the entries of $\boldsymbol{C}$ and $\boldsymbol{W}$, and so the equation defines a hypersurface in $\mathbb{P}^8$. Clearly, $\boldsymbol{C} : \boldsymbol{W}$ is a member of this hypersurface. Thus $\boldsymbol{C} : \boldsymbol{W}$ is not an arbitrary point in $\mathbb{P}^8$ but is constrained to a 7-dimensional submanifold of $\mathbb{P}^8$, a fact noted in Ref. [135].

2.4.1 Enforcing the cubic constraint

A number of methods of estimating the motion matrices are provided in Sections 4 and 5, but there is no guarantee that an estimate $\{\boldsymbol{C}, \boldsymbol{W}\}$ produced by such a procedure will satisfy equation (2.50). A rectification procedure for modifying estimates to accommodate this constraint is therefore needed.

Given a pair of motion matrices $\{\boldsymbol{C}, \boldsymbol{W}\}$, let

$$\boldsymbol{C}_\rho = \frac{\boldsymbol{C} - \boldsymbol{PCP}}{\|\boldsymbol{C} - \boldsymbol{PCP}\|^2 + \|\boldsymbol{W}\|^2},$$
$$\boldsymbol{W}_\rho = \frac{\boldsymbol{W}}{\|\boldsymbol{C} - \boldsymbol{PCP}\|^2 + \|\boldsymbol{W}\|^2},$$

where

$$\boldsymbol{P} = \boldsymbol{I} + \|\widehat{\boldsymbol{w}}\|^{-2}\boldsymbol{W}^2.$$

It is easily verified that if $\{\boldsymbol{C}, \boldsymbol{W}\}$ satisfies (2.50), then $\boldsymbol{PCP} = \mathbf{0}$. Hence $\boldsymbol{W} = \boldsymbol{W}_\rho$ whenever (2.50) holds for $\{\boldsymbol{C}, \boldsymbol{W}\}$. Since $\boldsymbol{P}\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{w}}$ and $\widehat{\boldsymbol{w}}^T\boldsymbol{P} = \widehat{\boldsymbol{w}}^T$, it follows that $\widehat{\boldsymbol{w}}^T\boldsymbol{C}_\rho\widehat{\boldsymbol{w}} = 0$, which in turn immediately implies that $\widehat{\boldsymbol{w}}_\rho{}^T\boldsymbol{C}_\rho\widehat{\boldsymbol{w}}_\rho = 0$. Thus passing from $\{\boldsymbol{C}, \boldsymbol{W}\}$ to $\{\boldsymbol{C}_\rho, \boldsymbol{W}_\rho\}$ gives the required modification procedure.

2.5 Self-calibration with free focal length

As we have outlined in Section 2.1.2 only 5 ego-motion parameters can be determined from image data, as one parameter is lost due to scale indeterminacy. Given that $\boldsymbol{C} : \boldsymbol{W}$ is a member of a 7-dimensional hypersurface in $\mathbb{P}^8$, the total number of key parameters that can be recovered by exploiting $\boldsymbol{C} : \boldsymbol{W}$ cannot exceed 7. If we want to recover all 5 computable ego-motion parameters, we have to accept that not all intrinsic parameters can be retrieved. Accordingly, we have to adopt a particular form of $\boldsymbol{A}$, deciding which intrinsic parameters will be known and which will be unknown, and also which will be fixed and which will be free. A *free* parameter is defined in [17] to be one that may vary continuously with time.

We assume that the focal length of the camera is unknown and free, and that pixels are square with unit length (in terms of Γ_{c}). We further assume that the principal point is fixed and known, and that the data is represented with respect to this fixed principle point. In this situation, for each time instant t, $\boldsymbol{A}(t)$ is given by

$$\boldsymbol{A}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -f(t) \end{bmatrix}, \tag{2.51}$$

where $f(t)$ is the unknown focal length at time t. From now on we shall omit in notation the dependence upon time. Let $\pi(\boldsymbol{v})$ be the *projective form* of $\boldsymbol{v}$, that is, the point in the 2-dimensional real projective space $\mathbb{P}^2$ with homogeneous coordinates given by the composite ratio

$$\pi(\boldsymbol{v}) = (v_1 : v_2 : v_3).$$

As is clear, $\pi(\boldsymbol{v})$ captures the direction of $\boldsymbol{v}$. It emerges that, with the adoption of the above form of $\boldsymbol{A}$, one can conduct self-calibration by explicitly expressing the entities $\boldsymbol{\omega}$, $\pi(\boldsymbol{v})$, f and $\dot{f}$ in terms of $\boldsymbol{C} : \boldsymbol{W}$. Of these entities, $\boldsymbol{\omega}$ and $\pi(\boldsymbol{v})$ account for 5 ego-motion parameters ($\boldsymbol{\omega}$ accounting for 3 parameters and $\pi(\boldsymbol{v})$ accounting for 2 parameters), and f and $\dot{f}$ account for 2 intrinsic parameters. Note that $\boldsymbol{v}$ is not wholly recoverable, the length of $\boldsymbol{v}$ being indeterminate. Retrieving $\boldsymbol{\omega}$, $\pi(\boldsymbol{v})$, f and $\dot{f}$ from $\boldsymbol{C} : \boldsymbol{W}$ has as its counterpart in stereo vision Hartley's procedure [55] to determine 5 relative orientation parameters and 2 focal lengths from a fundamental matrix whose intrinsic-parameter parts have a form analogous to that given in equation (2.51).

Let $\boldsymbol{S}$ be the matrix defined as

$$\boldsymbol{S} = \boldsymbol{A}^{-1}(\overset{\circ}{\boldsymbol{R}} + \boldsymbol{B})\boldsymbol{A}.$$

A straightforward calculation shows that

$$\boldsymbol{S} = \begin{bmatrix} 0 & -\omega_3 & -f\omega_2 \\ \omega_3 & 0 & f\omega_1 \\ \omega_2/f & -\omega_1/f & \dot{f}/f \end{bmatrix}. \tag{2.52}$$

With the use of $\boldsymbol{S}$, equation (2.49) can be rewritten as

$$\boldsymbol{C} = \frac{1}{2}(\boldsymbol{W}\boldsymbol{S} - \boldsymbol{S}^T\boldsymbol{W}). \tag{2.53}$$

Regarding $\boldsymbol{C}$ and $\boldsymbol{W}$ as being known and $\boldsymbol{S}$ as being unknown, and taking into account the fact that $\boldsymbol{C}$—a 3×3 symmetric matrix—has only six independent entries, equation (2.53) can be seen as a system of six inhomogeneous linear equations in the entries of $\boldsymbol{S}$. Of these only five equations are independent, as $\boldsymbol{C}$

and $\boldsymbol{W}$ are interrelated. Solving for the entries of $\boldsymbol{S}$ one can express $\boldsymbol{\omega}$, f and $\dot{f}$ in terms of $\boldsymbol{C} : \boldsymbol{W}$. Once f and hence $\boldsymbol{A}$ is represented as a function of $\boldsymbol{C} : \boldsymbol{W}$, the matrix $\overset{\circ}{\boldsymbol{T}}$ can be found from

$$\overset{\circ}{\boldsymbol{T}} = (\boldsymbol{A}^T)^{-1}\boldsymbol{W}\boldsymbol{A}^{-1}, \tag{2.54}$$

which immediately follows from (2.33). Note that $\boldsymbol{W}$ is known only up to a scalar factor, and so $\overset{\circ}{\boldsymbol{T}}$ (and hence $\boldsymbol{v}$), cannot be fully determined. However, as $\boldsymbol{W}$ depends linearly on $\overset{\circ}{\boldsymbol{T}}$, it is clear that $\pi(\boldsymbol{v})$ can be regarded as being a function of $\boldsymbol{C} : \boldsymbol{W}$. In this way, all the parameters $\boldsymbol{\omega}$, $\pi(\boldsymbol{v})$, f, and $\dot{f}$ are determined from $\boldsymbol{C} : \boldsymbol{W}$.

We now give explicit formulae for $\boldsymbol{\omega}$, $\pi(\boldsymbol{v})$, f, and $\dot{f}$. Set

$$\eta_1 = -\frac{\omega_1}{f}, \qquad \eta_2 = -\frac{\omega_2}{f}, \qquad \eta_3 = -\omega_3, \qquad \eta_4 = f^2, \qquad \eta_5 = \frac{\dot{f}}{f}. \tag{2.55}$$

In view of equations (2.52) and (2.53), we have

$$\begin{aligned} c_{11} &= -w_2\eta_2 + w_3\eta_3, \\ 2c_{12} &= w_2\eta_1 + w_1\eta_2, \\ c_{22} &= -w_1\eta_1 + w_3\eta_3. \end{aligned}$$

Hence

$$\begin{aligned} \eta_1 &= \frac{2c_{12}w_2 - (c_{22} - c_{11})w_1}{w_1^2 + w_2^2}, \\ \eta_2 &= \frac{2c_{12}w_1 + (c_{22} - c_{11})w_2}{w_1^2 + w_2^2}, \\ \eta_3 &= \frac{c_{11}w_1^2 + 2c_{12}w_1w_2 + c_{22}w_2^2}{w_3(w_1^2 + w_2^2)}. \end{aligned} \tag{2.56}$$

The expressions on the right-hand side are homogeneous of degree 0 in the entries of $\boldsymbol{C}$ and $\boldsymbol{W}$; that is, they do not change if $\boldsymbol{C}$ and $\boldsymbol{W}$ are multiplied by a common scalar factor. Therefore the above equations can be regarded as formulae for η_1, η_2, and η_3 in terms of $\boldsymbol{C} : \boldsymbol{W}$. Assuming—as we now may—that η_1, η_2, η_3 are known, we again use (2.52) and (2.53) to derive the following formulae for η_4 and η_5:

$$\begin{aligned} 2c_{13} &= w_3\eta_1\eta_4 + w_2\eta_5 - w_1\eta_3, \\ 2c_{23} &= w_3\eta_2\eta_4 - w_1\eta_5 - w_2\eta_3, \\ c_{33} &= -(w_1\eta_1 + w_2\eta_2)\eta_4. \end{aligned} \tag{2.57}$$

These three equations in η_4 and η_5 are not linearly independent. To determine η_4 and η_5 we proceed as follows. Let $\boldsymbol{\delta} = [\eta_4, \eta_5]^T$, let $\Xi = [d_1, d_2, d_3]^T$ be such that

$$d_1 = 2c_{13} + w_1\eta_3, \qquad d_2 = 2c_{23} + w_2\eta_3, \qquad d_3 = c_{33},$$

and let

$$\boldsymbol{D} = \begin{bmatrix} w_3\eta_1 & w_2 \\ w_3\eta_2 & -w_1 \\ -w_1\eta_1 - w_2\eta_2 & 0 \end{bmatrix}.$$

With this notation, (2.57) can be rewritten as

$$\boldsymbol{D}\boldsymbol{\delta} = \Xi,$$

Now $\boldsymbol{\delta}$ is given by

$$\boldsymbol{\delta} = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\Xi.$$

More explicitly, we have the following formulae:

$$\begin{aligned} \eta_4 &= \frac{1}{\Gamma}\left(w_1w_3d_1 + w_2w_3d_2 - (w_1^2 + w_2^2)d_3\right), \\ \eta_5 &= \frac{1}{\Gamma}\big((w_1w_2\eta_1 + (w_2^2 + w_3^2)\eta_2)d_1 - ((w_1^2 + w_3^2)\eta_1 + w_1w_2\eta_2)d_2 \\ &\qquad + (w_2w_3\eta_1 - w_1w_3\eta_2)d_3\big), \end{aligned} \tag{2.58}$$

where $\Gamma = (w_1^2 + w_2^2 + w_3^2)(w_1\eta_1 + w_2\eta_2)$. Again the expressions on the right-hand side are homogeneous of degree 0 in the entries of $\boldsymbol{C}$ and $\boldsymbol{W}$, and so the above equations can be regarded as formulae for η_4 and η_5 in terms of $\boldsymbol{C} : \boldsymbol{W}$.

Combining (2.55), (2.56) and (2.58), we obtain

$$\omega_1 = -\eta_1\sqrt{\eta_4}, \qquad \omega_2 = -\eta_2\sqrt{\eta_4}, \qquad \omega_3 = -\eta_3, \qquad f = \sqrt{\eta_4}, \qquad \dot{f} = \eta_5\sqrt{\eta_4}.$$

Rewriting (2.54) as

$$t_1 = -\frac{w_1}{f}, \qquad t_2 = -\frac{w_2}{f}, \qquad t_3 = w_3, \tag{2.59}$$

and taking into account that f has already been specified, we find that

$$\pi(\boldsymbol{v}) = (-w_1 : -w_2 : fw_3).$$

In this way, all the parameters $\boldsymbol{\omega}$, $\pi(\boldsymbol{v})$, f and $\dot{f}$ are determined from $\boldsymbol{C} : \boldsymbol{W}$.

2.6 Degeneracies

There has been significant work in the area of determining degeneracies in the stereo imaging process [23, 56, 78, 79, 132, 133]. The differential epipolar equation suffers correspondingly, as the same epipolar geometry underlies both schemes. We consider here the degeneracies in the process of self-calibration outlined in the previous section.

Inspecting (2.56) we see the need to assume that $t_3 \neq 0$ and also that either $t_1 \neq 0$ or $t_2 \neq 0$. Furthermore, Γ appearing in (2.58) also has to be non-zero. If we assume that $t_3 \neq 0$ and also that either $t_1 \neq 0$ or $t_2 \neq 0$, we see that $\Gamma \neq 0$ if and only if $w_1\eta_1 + w_2\eta_2 \neq 0$. Taking into account the first two equations of (2.55) and the first two equations of (2.59), we see that the latter condition is equivalent to $t_1\omega_1 + t_2\omega_2 \neq 0$. Altogether we have then to assume that $t_3 \neq 0$, that either $t_1 \neq 0$ or $t_2 \neq 0$, and, furthermore, that $t_1\omega_1 + t_2\omega_2 \neq 0$.

Fundamentally this restriction means that, in order to calculate structure from motion via this method, the camera must exhibit some movement along its optical axis, and that the vector describing the translation of the camera and the optical axis must not lie on the same plane. This corresponds to the self-calibration degeneracy in general stereo that occurs when the optical axes of the cameras are coplanar.

2.7 Conclusion

We have related above two derivations of the differential epipolar equation originally provided by Brooks et al. [17, 18]. Additionally we have shown a method for enforcing the cubic constraint on the motion matrices and derived simple degeneracy conditions.

We are now in a position to consider reconstruction from optical flow and subsequently statistical estimation of the motion matrices.

Chapter 3

Reconstruction and relative position

Within this chapter we determine a means of reconstructing a viewed scene given the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ and an optical flow field, we also provide a method for estimating the relative position of a camera from one instant to another. Determining the movement of a camera from one time instant to another is essential if reconstructions calculated at these times are to be compared or merged. Finally, we relate motion measured from a world-centred coordinate system to the ego-motion measured from a camera-centered coordinate system.

3.1 Scene reconstruction

Reconstruction of a viewed scene is the process of calculating (an estimate of) the coordinates of the point in space corresponding to a particular image point. Reconstruction thus requires that we relate an object's position in the scene to the associated optical flow vector and the motion of the camera. This relationship necessarily relies on the nature of the projection of points onto the image plane as described in Section 2.2.2.

From equation (2.44) we know that

$$\dot{\boldsymbol{z}} = \frac{z_3 \dot{f} - \dot{z}_3 f}{f^2} \boldsymbol{p} - \frac{z_3}{f} \dot{\boldsymbol{p}}.$$

It follows from (2.41) that $\dot{\boldsymbol{z}} = -\boldsymbol{v} + \dot{\boldsymbol{R}} \boldsymbol{z}$, therefore

$$\boldsymbol{v} + \frac{z_3 \dot{f} - \dot{z}_3 f}{f^2} \boldsymbol{p} - \frac{z_3}{f} \dot{\boldsymbol{p}} - \dot{\boldsymbol{R}} \boldsymbol{z} = 0.$$

Substituting the expression given in equation (2.43) for $\boldsymbol{z}$ we obtain

$$\boldsymbol{v} + \frac{z_3 \dot{f} - \dot{z}_3 f}{f^2} \boldsymbol{p} - \frac{z_3}{f} \dot{\boldsymbol{p}} + \frac{z_3}{f} \dot{\boldsymbol{R}} \boldsymbol{p} = 0.$$

This leads to a system of three equations in two unknowns:

$$\boldsymbol{v} = -\frac{z_3\dot{f} - \dot{z}_3 f}{f^2}\boldsymbol{p} + \frac{z_3}{f}\dot{\boldsymbol{p}} - \frac{z_3}{f}\dot{\boldsymbol{R}}\boldsymbol{p}. \tag{3.1}$$

Clearly, $\dot{f}, f, \boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ are known, $\boldsymbol{v}$ is partially known (up to a scale factor), and z_3 and $\dot{z}_3$ are unknown. Assume temporarily that $\boldsymbol{v}$ is known. Then (3.1) can immediately be employed to find z_3 and $\dot{z}_3$. Bearing in mind that $\boldsymbol{m}$, $\dot{\boldsymbol{m}}$, $\boldsymbol{v}$, and $\dot{\boldsymbol{R}}\boldsymbol{m}$ are column vectors with three entries, one can regard (3.1) as being a system of three linear equations in z_3 and $\dot{z}_3$. Upon solving this system for z_3 and $\dot{z}_3$, we use (2.43) and (2.44) to determine $\boldsymbol{z}$ and $\dot{\boldsymbol{z}}$. With $\boldsymbol{z}$ thus specified, scene reconstruction is complete.

Note that this method breaks down when $(\dot{\boldsymbol{p}} - \dot{\boldsymbol{R}}\boldsymbol{p})$ and $\boldsymbol{p}$ are linearly dependent, or equivalently, in view of (2.43) and (2.44), if

$$\boldsymbol{z} \times (\dot{\boldsymbol{z}} + \dot{\boldsymbol{R}}\boldsymbol{z}) = \boldsymbol{0}.$$

This, by (2.41), is equivalent to $\boldsymbol{z} \times \boldsymbol{v} = \boldsymbol{0}$. We need therefore to assume that $\boldsymbol{z} \times \boldsymbol{v} \neq \boldsymbol{0}$, or equivalently that $\boldsymbol{z}$ and $\boldsymbol{v}$ are linearly independent, whenever $z_3 \neq 0$. In particular, this means that $\boldsymbol{v} \neq \boldsymbol{0}$ holds. We have assumed above that we know the scale of $\boldsymbol{v}$. In fact, we see from equation (3.1) that the scale of $\boldsymbol{v}$ affects only the scale of the reconstructed object. The shape of the reconstruction is unaltered. The scale indeterminacy of the recovered translation vector described in Section 1.7 thus leads to a scale indeterminacy in the recovered reconstruction. This phenomenon parallels the well known result from stereo analysis to the effect that it is impossible to recover the scale of a reconstructed object without prior knowledge of the separation of the cameras.

3.1.1 Testing the reconstruction formulae

In order to test the reconstruction formulae, a realistic model of a camera and its motion were generated, and then the implied motion matrices and a corresponding set of optical flow were calculated. The optical flow was generated so as to correspond to points on three surfaces of a cube. The calculated optical flow and motion parameters determined from the motion matrices were then used to generate a reconstruction by the method described above. The results are illustrated in Figure 3.1. The reconstruction calculated matches the original data precisely once the scale indeterminacy has been taken into account. This is as would be expected given that the exact motion matrices have been used to calculate the key parameters implying that no noise has been introduced into the system.

3.2 Calculating relative position

Determining the change in position of an object over a given time period when only velocity information is available naturally requires integration. The differ-

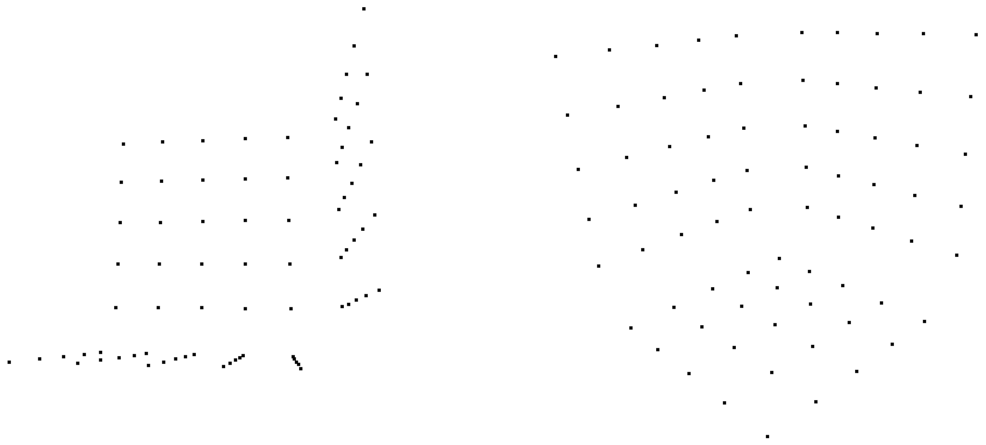

Figure 3.1: Cube reconstruction

ential epipolar equation provides information about the velocity of the camera (via the motion matrices), but only for a specific time instant. Determining the change in position over time thus requires integration on the basis of successive motion estimates.

In order to integrate over a time period the velocity of the camera must be fully described at each intervening instant. Unfortunately in the course of estimating motion matrices and subsequent reconstruction of the scene from instantaneous optical flow, the speed of translation $||\boldsymbol{v}||$ remains undetermined and can take any positive value. This reflects the aforementioned scale indeterminacy of the reconstruction problem; namely, that it is impossible to tell from an image sequence whether the camera is moving very quickly past a large object, or slowly past a small one, without reference to prior knowledge about the scene. Rotational velocity on the other hand is fully recoverable from the motion matrices. Given that it is not possible to recover absolute translational velocity from $\boldsymbol{C}$ and $\boldsymbol{W}$ we now provide a means of determining relative translational velocity.

3.2.1 Determining relative translational velocity

Suppose that we are given an optical flow field that evolves over a period of time, from which we calculate estimates of the motion matrices at a number of instants. When calculating camera translation as described above, it is quite conceivable that the scale factor corresponding to $||\boldsymbol{v}||$ may change in an uncontrollable way from one time instant to another [4]. This indeterminacy can be significantly reduced, however, if we are able to track a single feature over a period of time. Given two time instants s and t with $s < t$, suppose that we are given a function $[s,t] \ni \sigma \mapsto \{\boldsymbol{m}(\sigma), \dot{\boldsymbol{m}}(\sigma)\}$ that represents a moving feature. The relative translational velocity $\|\boldsymbol{v}(\sigma)\|/\|\boldsymbol{v}(s)\|$ may be uniquely determined for all $\sigma \in [s,t]$, once the initial velocity $\|\boldsymbol{v}(s)\|$ is fixed. That is, the velocity $\|\boldsymbol{v}(\sigma)\|$ becomes uniquely determined for all $\sigma \in [s,t]$.

Using (2.4) and (2.48), we first determine $\boldsymbol{p}(\sigma)$ and $\dot{\boldsymbol{p}}(\sigma)$ for each $\sigma \in [s,t]$. Omitting in notation the dependence upon σ, let

$$\begin{aligned} \boldsymbol{k} &= \widehat{\boldsymbol{v}}(\dot{f}\boldsymbol{p} - f(\dot{\boldsymbol{p}} + \widehat{\boldsymbol{\omega}}\boldsymbol{p})), \\ \boldsymbol{l} &= f\widehat{\boldsymbol{v}}\boldsymbol{p}. \end{aligned}$$

Applying $\widehat{\boldsymbol{v}}$ to both sides of (3.1) and taking into account that $\widehat{\boldsymbol{v}}\boldsymbol{v} = 0$, we see that

$$z_3\boldsymbol{k} - \dot{z}_3\boldsymbol{l} = 0.$$

So

$$\frac{\dot{z}_3}{z_3} = \frac{\boldsymbol{l}^T\boldsymbol{k}}{\|\boldsymbol{l}\|^2}. \tag{3.2}$$

Here we tacitly assume that $z_3 \neq 0$ and $\boldsymbol{l} \neq 0$. In fact, it suffices to assume only that $z_3 \neq 0$; when this assumption holds, then the inequality $\boldsymbol{l} \neq 0$ follows from our standing assumption that $\boldsymbol{v} \neq \boldsymbol{0}$, the definition of $\boldsymbol{l}$, and equation (2.43). Note that the right-hand side of equation (3.2) does not change if $\boldsymbol{v}$ is multiplied by a non-zero scalar. It can therefore be regarded as being a function of $\pi(\boldsymbol{v})$, $\boldsymbol{\omega}$, f, $\dot{f}$, $\boldsymbol{p}$ and $\dot{\boldsymbol{p}}$, and can be treated as known. Similarly, $\boldsymbol{q}$ defined by

$$\boldsymbol{q} = f^{-2}\left(\frac{\boldsymbol{l}^T\boldsymbol{k}}{\|\boldsymbol{l}\|^2}f\boldsymbol{p} - \dot{f}\boldsymbol{p} + f(\dot{\boldsymbol{p}} + \widehat{\boldsymbol{\omega}}\boldsymbol{p})\right) \tag{3.3}$$

can be regarded as known. In view of (3.1) and (3.2), we have

$$\boldsymbol{v} = z_3\boldsymbol{q}$$

and further

$$\|\boldsymbol{v}\| = |z_3|\,\|\boldsymbol{q}\|. \tag{3.4}$$

To simplify the notation, let

$$\begin{aligned} v &= ||\boldsymbol{v}|| \\ q &= ||\boldsymbol{q}||\,. \end{aligned} \tag{3.5}$$

Taking the logarithmic derivative of both sides of (3.4) and using (3.2), we see that

$$\frac{\dot{v}}{v} = \frac{\dot{z}_3}{z_3} + \frac{\dot{q}}{q} = \frac{\boldsymbol{l}^T\boldsymbol{k}}{\|\boldsymbol{l}\|^2} + \frac{\dot{q}}{q}. \tag{3.6}$$

Let

$$g = \frac{\boldsymbol{l}^T\boldsymbol{k}}{\|\boldsymbol{l}\|^2} + \frac{\dot{q}}{q}.$$

The scalars q and $\dot{q}$ are derivable from equations (3.3) and (3.5) respectively, and $\boldsymbol{l}^T\boldsymbol{k}/\left|\left|\boldsymbol{l}\right|\right|^2$ is made up of known objects, so, in light of equation (3.2.1), g can be regarded as known. In view of (3.6), we finally find that

$$\frac{v(\sigma)}{v(s)} = \exp\left(\int_s^\sigma g(u)\,du\right), \tag{3.7}$$

which is the desired formula for the relative translational velocity. The fact that we can calculate relative translational velocity enables the resizing of reconstructions calculated at different time instants such that they share a common scale factor to the original scene. This result, therefore, does not give us the absolute scale of such reconstructions, but allows the relative scaling between them to be calculated. Section 3.4.2 describes a method for calculating the trajectory of a camera based on this result.

3.2.2 Testing the accuracy of relative velocity determination

In order to test the accuracy of the formula for the determination of relative translational velocity, a time interval (10 seconds) over which to integrate was selected. From this interval were selected 150 equally-spaced instants, each representing the time at which a particular image was taken. These settings matched the 15 images per second frame rate of the Pulnix TM9701 progressive scan camera. We then generated motion matrices as a function of time over this interval to represent a camera undergoing changing motion. The projection of a fixed scene point in every image was then calculated at every instant.

Figure 3.2 depicts the magnitude of the true and estimated translational velocities over this time period. We could calculate relative velocity over any interval, but for simplicity we choose the interval starting at time 0. In order to enable comparison, the correct value for $v(0)$ (the velocity at time 0) was used to generate subsequent estimates from (3.7). As is generally the case with numerical integration, increasing the number of steps over a given interval increased the accuracy of the result. Figure 3.2 shows that the integration method of calculating relative translational velocity performs well; in fact it performs so well that the lines are barely distinguishable. As expected, however, it does diverge from the true solution over time. Tests on real optical flow data have not been carried out because the imaging equipment used for real data capture does not allow accurate measurement of the true relative translational velocity. Without this information no meaningful comparison is possible. Adding noise to the data leads to decreased accuracy of the relative velocity estimate, as would be expected. This effect may be mitigated by generating estimates based on a number of scene points and calculating their average.

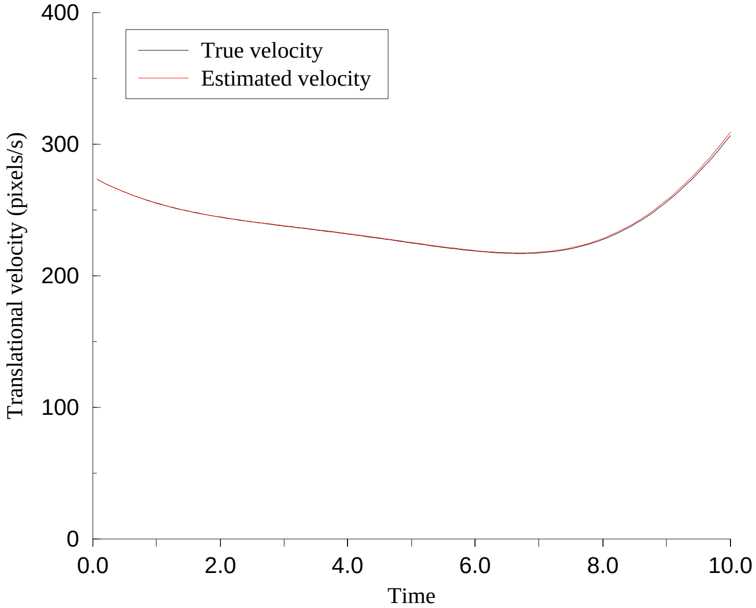

Figure 3.2: Relative velocity estimation

3.3 Ego-motion from change in position

In real imaging situations, we often define the motion of the camera relative to the scene based coordinate system. The motion matrices, however, describe motion in terms of the frame attached to the camera itself. It is thus useful to be able to represent motion in one frame that has been measured in another.

Firstly we consider the problem of representing motion measured in Γ_{w} the frame fixed to the viewed scene in terms of Γ_{c} the frame attached to the camera. If the rotation and translation of the camera relative to the fixed scene frame is denoted by $\boldsymbol{R}^{-1}$ and $\boldsymbol{v}$ respectively, then, by definition, the instantaneous rotation $\mathring{\boldsymbol{R}}$ in the camera frame is given by

$$\mathring{\boldsymbol{R}} = \boldsymbol{R}^{-1}\dot{\boldsymbol{R}}, \tag{3.8}$$

and the corresponding translation $\mathring{\boldsymbol{T}}$ by

$$\mathring{\boldsymbol{T}} = \boldsymbol{R}^{-1}(\dot{\boldsymbol{v}} - \boldsymbol{v}). \tag{3.9}$$

We are particularly interested in the instantaneous speed of rotation and translation of the camera because it is this information that may be recovered directly from the differential epipolar equation.

3.4 Relative motion from ego-motion

We seek to recover the position of the camera in the frame attached to the scene at some time t, given its ego-motion (described in camera centered coordinates).

This requires solving the differential equations (3.8) and (3.9) given that we know only $\overset{\circ}{\boldsymbol{R}}$ and $\overset{\circ}{\boldsymbol{T}}$. We thus seek to perform the reverse of the transformation described in the previous section.

3.4.1 Recovering rotation

Rather than solve equations (3.8) and (3.9) directly, we initially consider the problem of recovering $f(t)$ from the linear system $c(t)f(t) = \dot{f}(t)$ when only $\dot{f}(t)$ is known. We solve this system using the method of variation of a constant.

If we express the equation in question as

$$c(t)f(t) = \frac{df}{dt},$$

we can then rewrite it as

$$c(t)dt = \frac{df}{f(t)}.$$

Since

$$\frac{d(\ln f(t))}{dt} = \frac{df}{dt}\frac{1}{f(t)},$$

it follows that

$$d(\ln f(t)) = \frac{df}{f(t)}$$

and hence

$$c(t)dt = d(\ln f(t)).$$

By the fundamental theorem of calculus

$$\int_{t_0}^{t} dg = g(t) - g(t_0)$$

so

$$\begin{aligned}\int_{t_0}^{t} c(s)ds &= \ln f(t) - \ln f(t_0) \\ &= \ln \frac{f(t)}{f(t_0)}.\end{aligned}$$

Taking the exponential of both sides we find that

$$\exp\left(\int_{t_0}^{t} c(s)ds\right) = \frac{f(t)}{f(t_0)}.$$

and thus that

$$f(t) = f(t_0)\exp\left(\int_{t_0}^{t} c(s)ds\right). \tag{3.10}$$

If we assume that $c(s)$ is constant, i.e. that $c(s) = c$, then

$$f(t) = f(t_0)\exp((t - t_0)c). \tag{3.11}$$

This result, although derived using scalar functions, applies equally to the case where $c(t), f(t)$ and $d(t)$ represent matrix and vector valued functions of appropriate dimensions. On this basis we find that we may determine $\boldsymbol{R}$ from equation (3.8) using the form of equation (3.10)

$$\boldsymbol{R}(t) = \boldsymbol{R}(t_0)\exp\left(\int_{t_0}^{t} \overset{\circ}{\boldsymbol{R}}(s)ds\right). \tag{3.12}$$

3.4.1.1 Recovering rotation from constant ego-rotation

If the rotation component of the ego-motion of the camera, or ego-rotation, is constant, we can solve equation (3.12) for $\boldsymbol{R}(t)$ without the need for integration.

Constant ego-rotation implies that $\overset{\circ}{\boldsymbol{R}}$ is constant. If we set our reference point as the orientation of the camera at time $t = 0$, then $\boldsymbol{R}(0) = \boldsymbol{I}$. We thus know, from equation (3.11), that the solution $\boldsymbol{R}(t)$ is given by

$$\boldsymbol{R}(t) = \exp(t\overset{\circ}{\boldsymbol{R}}) \tag{3.13}$$

where the exponential of a matrix $\boldsymbol{X}$ is defined such that

$$\exp(\boldsymbol{X}) = \sum_{n=0}^{\infty} \frac{\boldsymbol{X}^n}{n!}.$$

Given that $\overset{\circ}{\boldsymbol{R}}$ is antisymmetric by definition, we know from the theory of matrix exponentials [107] that

$$\exp(\overset{\circ}{\boldsymbol{R}}) = \boldsymbol{I} + \frac{\sin\theta}{\theta}\overset{\circ}{\boldsymbol{R}} + \frac{1-\cos\theta}{\theta^2}\overset{\circ}{\boldsymbol{R}}^2$$

where $\theta = \sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2}$, and ω_i are the non-zero elements of the antisymmetric matrix $\overset{\circ}{\boldsymbol{R}}$. From this and equation (3.13) we see that, given a constant rotation $\overset{\circ}{\boldsymbol{R}}$, we can recover the rotation of the camera relative to its position at time 0 by the equation

$$\boldsymbol{R}(t) = \boldsymbol{I} + t\frac{\sin\theta}{\theta}\overset{\circ}{\boldsymbol{R}} + t^2\frac{1-\cos\theta}{\theta^2}\overset{\circ}{\boldsymbol{R}}^2. \tag{3.14}$$

This method of calculating the rotation of the camera relative to the scene over an interval has been tested synthetically by simulating a camera undergoing constant motion through a rigid scene. The rotation calculated using equation (3.14) on the basis of the true motion matrices matched the true rotation to the accuracy of the calculations.

The above thus provides a method of recovering the rotation of a camera relative to the scene viewed over an interval. The simplification that leads to equation (3.13) and therefore equation (3.14), however, is valid only for a camera undergoing rotation of constant direction and magnitude. This may occur in certain circumstances, but in general, the motion of the camera will not be so constrained. If the rotation of the camera is not constant we revert to equation (3.12) and therefore to numerical integration over the rotation estimates recovered from the differential epipolar equation.

3.4.1.2 Estimating rotation by integration

If the ego-rotation of the camera is not constant over time our only means of recovering rotation relative to the scene over time is to integrate. We therefore carry out the integral from equation (3.12). In order to determine the errors arising from integration rather than those from the estimation process we show results generated using the true values for the motion matrices. The rotation of the camera over time is difficult to visualise, thus Figure 3.3 shows the error in the rotation estimate. The error measure used is based on the difference between the true and estimated camera based coordinate frames. The angle between the true and estimated direction of each axis of the frames is summed to arrive at this indicator of estimate accuracy. Figure 3.3 shows that the error in the estimate of the rotation of the camera is small, but that it increases over time.

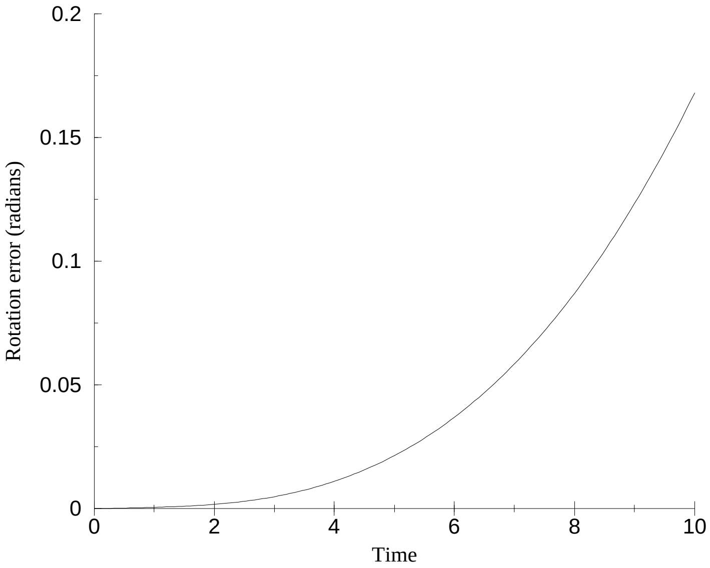

Figure 3.3: Rotation estimation by integration

The ability to recover the rotation of the camera over an interval means that we are now in a position to be able to relate reconstructions from differing camera positions. This ability is essential to the process of joining reconstructions from multiple images in a sequence, but also allows us to integrate over velocity and thus to calculate the trajectory of the camera.

3.4.2 Calculating translation

We have shown in Section 3.2.1 that by tracking one point across a series of images we can calculate relative translational speed thus partially solving the scale indeterminacy problem. Subsequently, in Section 3.4.1 we provided a method for recovering the rotation of a camera over an interval. It is the confluence of these capacities which now allows us to tackle the problem of determining translation over an interval.

Recovering the translation of the camera relative to its original position is complicated by the fact that equation (3.9), which is that

$$\mathring{\boldsymbol{T}} = \boldsymbol{R}^{-1}(\dot{\boldsymbol{v}} - \boldsymbol{v}),$$

diverges from the purely exponential form of equation (3.8). Expanding on the method used to solve equation (3.8), we guess that the translation of the camera may be recovered by setting

$$\boldsymbol{v}(t) = \exp(t)\boldsymbol{u}(t)$$

for some function $\boldsymbol{u}(t)$. Differentiating this formulation for $\boldsymbol{v}$ we find that

$$\begin{aligned}\dot{\boldsymbol{v}}(t) &= \exp(t)\boldsymbol{u}(t) + \exp(t)\dot{\boldsymbol{u}}(t)\\ &= \boldsymbol{v}(t) + \exp(t)\dot{\boldsymbol{u}}(t)\end{aligned}$$

and thus that

$$\dot{\boldsymbol{v}}(t) - \boldsymbol{v}(t) = \exp(t)\,\dot{\boldsymbol{u}}(t).$$

Rearranging equation (3.9), we get

$$\dot{\boldsymbol{v}} - \boldsymbol{v} = \boldsymbol{R}\mathring{\boldsymbol{T}}$$

so

$$\dot{\boldsymbol{v}}(t) + \boldsymbol{v}(t) = \exp(t)\,\dot{\boldsymbol{u}}(t) = \boldsymbol{R}(t)\,\mathring{\boldsymbol{T}}$$

and therefore

$$\dot{\boldsymbol{u}}(t) = \boldsymbol{R}(t)\mathring{\boldsymbol{T}}\exp(-t).$$

Unfortunately, the simplification leading to equation (3.13) in the rotation case is not possible here. The translation of the camera at time t relative to its position at time t_0 is therefore given by

$$\boldsymbol{v}(t) = \exp(t)\int_{t_0}^{t} \boldsymbol{R}(s)\mathring{\boldsymbol{T}}\exp(-s)ds.$$

This formulation of the translation over an interval requires that we know the magnitude of $\mathring{\boldsymbol{T}}$ and thus relies on the relative velocity result from Section 3.2.1.

We have thus derived an equation allowing us to calculate the translation of the camera over time by integration. Using this, and the rotation recovery mechanism from Section 3.4.1.2, we can now recover the trajectory of the camera over an interval. Unfortunately, and unavoidably, this trajectory will suffer from the scale indeterminacy inherent in the translation recovery process. Solving this problem requires prior knowledge about the scene or the initial motion of the camera.

3.4.3 Testing the recovered trajectory

In order to enable comparison of recovered and true camera trajectories we have used the correct initial velocity to seed the estimation process. As stated, the result of this integration will depend on the choice of $v(t_0)$, the velocity at time t_0. In testing we have used the true value of $v(t_0)$ to calculate the scale of the trajectory so as to facilitate comparison with the true camera trajectory. The selection of a particular value for $v(t_0)$ determines only the scale of the recovered translation of the camera and therefore the scale of the recovered trajectory.

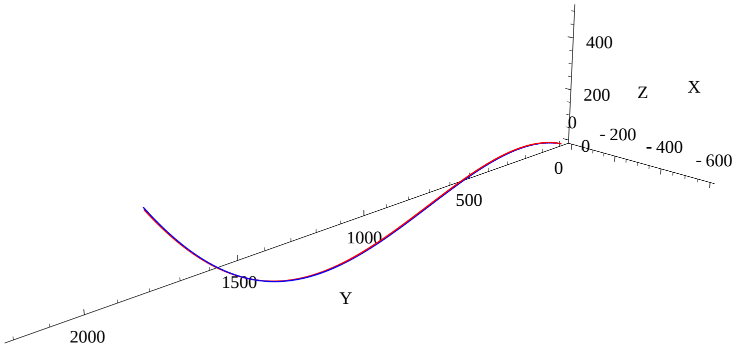

Figure 3.4: Estimating the camera trajectory

Figure 3.4 shows both the correct and estimated trajectories for the 10 seconds for which relative translational velocity was estimated in Section 3.2.2. The correct trajectory is shown in red, the estimate in blue. The two trajectories are so close together as to make their distinction almost impossible. Figure 3.5 depicts the results of the same tests, but shows only the error in position determination over the interval. The shape of the curve is somewhat counterintuitive, but is due to the fact that the translation direction changes over the interval. The error in any numerical integration process is affected by the shape of the integrated curve. In this case the particular velocity of the camera produces a curve of such shape that the errors generated between 4 and 8 seconds almost cancel those of the previous 4 seconds.

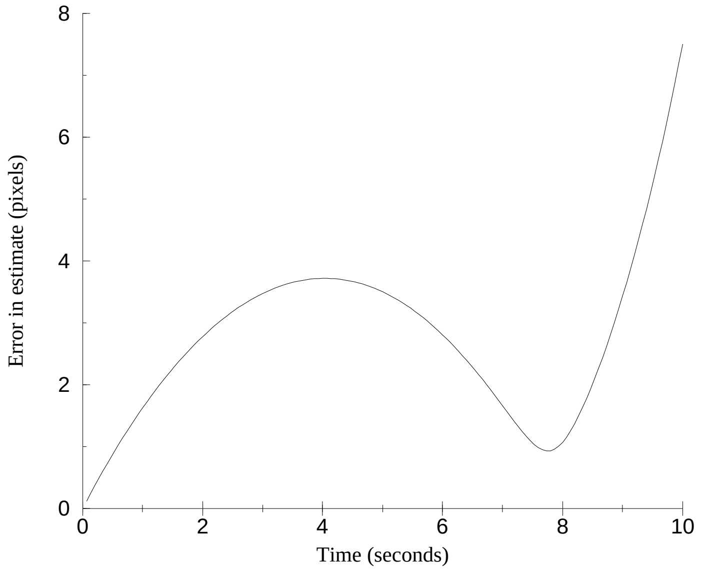

Figure 3.5: Error in estimated trajectory

3.5 Trajectory calculation without integration

Due to the cumulative nature of the error in relative translation estimation the estimated and actual positions of the camera will diverge over time. It is possible, using the reconstruction formulae from Section 3.1 and the relative translational velocity estimation procedure from Section 3.2.1, to calculate the position of a scene point relative to the camera at any two instants. Using this result and the fact that we can calculate the relative rotation from one time instant to another using the result from Section 3.4.1.2, we can calculate relative translation without integration. The integration necessary to calculate relative rotation and relative translational speed is unavoidable for a camera with varying ego-motion. Having determined these two quantities, however, we can use the reconstructed position of a scene point at two time instants to calculate the translation occurring between them.

Recovering translation from two reconstructions requires only that we subtract the vector representing a particular scene point's position at the first instant from that at the second instant. The vector representing the position of the scene point at the second time instant needs to be represented in the frame corresponding to the camera at the first time instant. This transformation requires only that we multiply the vector by the rotation matrix resulting from the integration process set out in Section 3.4.1.2.

Figure 3.6 shows the errors occurring in the process of relative position determination by reconstruction. The method is much more accurate than the integration-based method presented in Section 3.4.2.

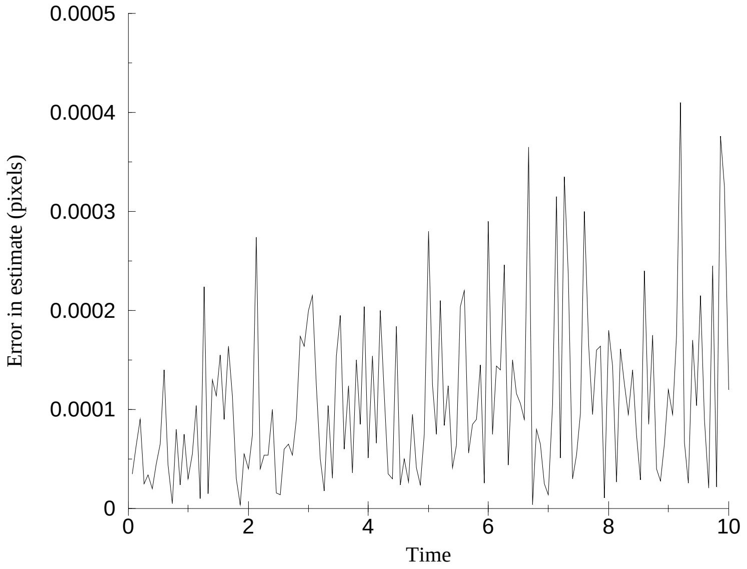

Figure 3.6: Estimating translation without integration

3.6 Conclusion

We have derived reconstruction formulae in Section 3.1, and shown that the reconstructions are accurate. We have also provided a means of estimating the trajectory of the camera over time from the motion matrices. This is significant as the extrinsic parameters therein encode only velocity information. The ability to recover a trajectory is important in that it allows reconstructions generated at different instants to be compared and combined, thus improving the quality of reconstructions from image sequences.

Chapter 4

Solving for C and W

In the previous chapters, we have described the differential epipolar equation and given methods for reconstructing the viewed scene and calculating the camera trajectory. Both scene reconstruction and trajectory calculation require knowledge of the motion matrices. This chapter presents various techniques for estimating these matrices, the goal being to determine methods robust to the presence of noise in the optical flow data.

4.1 Exact methods

Initially we consider two methods based on solving a system of equations. These methods are directly applicable on their own, but also form the basis for the statistical techniques presented in subsequent sections.

4.1.1 Eight-point estimator

Let $\mathcal{S}$ be the set of optical flow vectors corresponding to a particular image. The differential epipolar equation provides a constraint on the values of the elements of $\boldsymbol{C}$ and $\boldsymbol{W}$ for each optical flow vector, and therefore for each element of $\mathcal{S}$. The differential epipolar equation expands to

$$\begin{aligned} & m_1^2 c_{11} + 2m_1 m_2 c_{12} + 2m_1 c_{13} + m_2^2 c_{22} + 2m_2 c_{23} \\ & \quad + c_{33} + \dot{m}_2 v_1 - \dot{m}_1 v_2 + v_3(\dot{m}_1 m_2 - \dot{m}_2 m_1) = 0. \end{aligned} \tag{4.1}$$

For a set $\mathcal{S}$ of n optical flow vectors there are n such equations, each of which is linear in the elements of $\boldsymbol{C}$ and $\boldsymbol{W}$. We have seen above (Section 2.4) that the elements of the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ are constrained such that the system has seven degrees of freedom. One of these constraints is non-linear in the elements of the matrices, namely the requirement that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$. Ignoring this non-linear complication increases the number of degrees of freedom of the system to eight but allows estimation of $\{\boldsymbol{C}, \boldsymbol{W}\}$ by solving the set of differential epipolar equations algebraically. In this simplified case eight optical flow vectors are required to form a solution. This method is the fastest of those presented but suffers not only from the instability inherent in any method based on such a small sample, but from the obvious auxiliary disadvantage that there is no guarantee that the constraint $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$ is satisfied.

4.1.2 Seven-point estimator

By including the cubic constraint, an estimate of $\boldsymbol{C} : \boldsymbol{W}$ may be obtained from seven points by solving the system

$$\boldsymbol{m}_i^T\boldsymbol{W}\dot{\boldsymbol{m}}_i + \boldsymbol{m}_i^T\boldsymbol{C}\boldsymbol{m}_i = 0 \quad i = 1\ldots7, \tag{4.2a}$$
$$\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0. \tag{4.2b}$$

Equations (4.2a) are homogeneous in the entries of $\boldsymbol{C}$ and $\boldsymbol{W}$, and effectively provide seven constraints for the ratio $\boldsymbol{C} : \boldsymbol{W}$. If we identify $\boldsymbol{C} : \boldsymbol{W}$ with the vector

$$\boldsymbol{\Theta} = [c_{11}, c_{12}, c_{13}, c_{22}, c_{23}, c_{33}, w_{32}, w_{13}, w_{21}]^T,$$

then the space of solutions to system (4.2a) is spanned by two normalised linearly independent vectors $\widehat{\boldsymbol{\Theta}}_\alpha$ and $\widehat{\boldsymbol{\Theta}}_\beta$. These vectors may be calculated by singular value decomposition using the method employed in Section 4.5. An un-normalised solution $\widehat{\boldsymbol{\eta}}$ to the full system of equations can therefore be represented as a weighted sum of these vectors

$$\widehat{\boldsymbol{\eta}} = \lambda\,\widehat{\boldsymbol{\Theta}}_\alpha + (1-\lambda)\widehat{\boldsymbol{\Theta}}_\beta \tag{4.3}$$

for some scalar parameter λ. Substituting (4.3) into equation (4.2b) leads to a cubic constraint on λ. This equation has either one or three real solutions λ_i, which in turn give rise to one or three normalised estimates

$$\widehat{\boldsymbol{\Theta}}_i = \frac{\widehat{\boldsymbol{\eta}}_i}{||\widehat{\boldsymbol{\eta}}_i||}.$$

If three real estimates are obtained, we select the estimate $\widehat{\boldsymbol{\Theta}}_i$ satisfying the differential epipolar equation for all seven optical flow vectors, otherwise the solution corresponds to the single real estimate.

4.2 Least squares methods

We have seen that we can select a $\{\boldsymbol{C}, \boldsymbol{W}\}$ pair that will satisfy the differential epipolar equation for any set of seven or eight optical flow vectors. Naturally, if there is no noise in the measurement process, it would be quite possible to select a $\{\boldsymbol{C}, \boldsymbol{W}\}$ pair that will satisfy the differential epipolar equation for any number of flow vectors. Unfortunately, measuring optical flow from real image sequences introduces significant noise into the data, and consequently, if there are more than eight vectors, it is unlikely that there exist motion matrices such that the differential epipolar equation is satisfied for all of them. One method we can utilise in this situation is to select seven or eight optical flow vectors from the optical flow field, and calculate the motion matrices from these points alone. This proposition has been supported by Hartley [54] as a means of estimating the fundamental matrix. The problem with this approach, however, is that, because it does not utilise all of the data available to us (and therefore all of the information available to us), our estimates may not coincide with those that are in some sense 'most likely' given the data. We now tackle the problem of selecting the motion matrices which are most likely given the data.

4.2.1 Maximum likelihood estimation

Maximum likelihood estimation, like any estimation method, is concerned with selecting the *model* which best fits a given set of data. A classical example of a model fitting problem is that of fitting a line through a set of scattered points. The model describes a particular form the data may take, but, in doing so, may also represent some information about the process by which the data was generated. The set of all possible models from which the selection is made, is usually constrained to some class of functions, or distributions. In the line fitting example, this is the set of all lines. The set of all possible models is parameterised in order that each model may be identified individually. Maximum likelihood estimation differs from other estimation methods in that it requires that we seek the model for which the probability of the data having been generated is greatest.

The probability that a continuous random variable α will have any particular value ξ is always zero by definition. If we denote the probability that α falls within the interval between ξ and $\xi + \Delta\xi$ by $P(\xi < \alpha < \xi + \Delta\xi)$, then $P(\xi < \alpha < \xi + \Delta\xi) \geq 0$. In fact, in the limit, as $\Delta\xi$ approaches 0, the expression

$$\frac{P(\xi < \alpha < \xi + \Delta\xi)}{\Delta\xi}$$

approaches some value, which is not necessarily 0. In the following it is this limit which is taken as the value of the probability density function $p(\xi)$.

If a continuous random variable α has a Gaussian distribution of mean $\bar{\alpha}$ and variance σ, then the probability that it takes a particular value ξ is given by

$$p(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\xi-\bar{\alpha})^2/2\sigma^2}$$

If the continuous random variable is vector valued then the probability that $\boldsymbol{\alpha} = \boldsymbol{\xi}$ is

$$p(\boldsymbol{\xi}) = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} e^{-\sum_{i=1}^{n}(\xi_i-\bar{\alpha})^2/2\sigma^2}, \tag{4.4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$. The mean of each α_i is assumed to be $\bar{\alpha}$ and the variance σ.

In terms of optical flow, each measured data point is represented by a vector, which we label $\boldsymbol{x}_i$. The set of these measured data points is thus $\mathcal{S} = \{\boldsymbol{x}_i | \, i = 1 \ldots n\}$. We regard this set $\mathcal{S}$ as a sample taken from an aggregate of vector-valued random variables $\{\mathsf{x}_1 \ldots \mathsf{x}_n\}$. These vector-valued random variables are assumed to be stochastically independent, and the elements of each such vector are taken to be of variance σ. Any particular vector x_i is assumed to be of mean $\bar{\boldsymbol{x}}_i$, which represents the true value of the vector $\boldsymbol{x}_i$, so

$$\mathsf{E}\mathsf{x}_i = \bar{\boldsymbol{x}}_i \tag{4.5}$$

and

$$\mathsf{E}(\mathsf{x}_i - \bar{\boldsymbol{x}}_i)(\mathsf{x}_i - \bar{\boldsymbol{x}}_i)^T = \boldsymbol{\Sigma} \tag{4.6}$$

where $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma, \sigma, 0, \sigma, \sigma, 0)$.

If we parameterise the set of all possible models by the vector $\boldsymbol{\Theta}$, then maximum likelihood estimation becomes the problem of selecting the model $\boldsymbol{\Theta}$ under which the observed data set $\mathcal{S}$ is most likely to occur. The probability that our observed data $\mathcal{S}$ will occur given a particular model $\boldsymbol{\Theta}$ is represented by the conditional probability $p(\mathcal{S}|\boldsymbol{\Theta})$. We thus seek

$$\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} p(\mathcal{S}|\boldsymbol{\Theta}).$$

The probability $p(\mathcal{S}|\boldsymbol{\Theta})$ is the product of the probabilities $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$ of each of the points arising so, given equation (4.4),

$$p(\mathcal{S}|\boldsymbol{\Theta}) \propto \prod_{i=1}^{n} \left\{ \exp\left[-\frac{1}{2} \left((\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i)^T \, \boldsymbol{\Sigma}^- \, (\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i) \right)^2 \right] \right\}$$

where $\boldsymbol{\Sigma}^- = \operatorname{diag}(\sigma^{-1}, \sigma^{-1}, 0, \sigma^{-1}, \sigma^{-1}, 0)$ and $\boldsymbol{x}(\boldsymbol{\Theta})_i$ is the expected location of $\boldsymbol{x}_i$ based on the model $\boldsymbol{\Theta}$. Maximising this probability is equivalent to minimising its logarithm, so, since σ and $\boldsymbol{n}$ are not dependent on $\boldsymbol{\Theta}$, our maximum likelihood estimate is given by

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^{n} \left((\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i)^T \, \boldsymbol{\Sigma}^- \, (\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i) \right)^2 . \tag{4.7}$$

The maximum likelihood estimate is statistically optimal when the variance of each $\boldsymbol{x}_i$ is the same, and the errors in the $\boldsymbol{x}_i$ are uncorrelated. The term

$$(\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i)^T \, \boldsymbol{\Sigma}^- \, (\boldsymbol{x}_i - \boldsymbol{x}(\boldsymbol{\Theta})_i)$$

represents a measure of the distance between the observed data point $\boldsymbol{x}_i$ and its expected value $\boldsymbol{x}(\boldsymbol{\Theta})_i$. In this sense the maximum likelihood estimate corresponds to the model which minimises the sum of the squares of the distances between the data and their expected values. For a more detailed introduction to probability and maximum likelihood estimates, see Refs. [9, 50, 106, 120].

In terms of the current estimation problem the motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ represent the model $\boldsymbol{\Theta}$. We wish to maximise $p(\mathcal{S}|\{\boldsymbol{C},\boldsymbol{W}\})$, the probability that the measured optical flow $\mathcal{S}$ would occur given a moving camera with key parameters described by $\{\boldsymbol{C},\boldsymbol{W}\}$. We therefore minimise the sum of the squares of the distances between the elements of $\mathcal{S}$ and their expected values given a particular $\{\boldsymbol{C},\boldsymbol{W}\}$ pair. Representing this sum of squares measure as a function J of $\boldsymbol{C}$, $\boldsymbol{W}$ and $\mathcal{S}$, we seek the motion matrices for which $J(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ is minimal. Given that it is $J(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ which is minimised, we label this the cost function. We now consider the selection and application of an appropriate cost function.

4.3 The manifold of consistent optical flow

We have seen in Section 2.5 that the ratios of the elements of the motion matrices $\boldsymbol{C}:\boldsymbol{W}$ describe the instantaneous state and motion of a camera. We now define the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ of all optical flow vectors which satisfy the differential epipolar equation for these matrices. If

$$f_{\boldsymbol{C},\boldsymbol{W}}(\{\boldsymbol{m},\dot{\boldsymbol{m}}\}) = \boldsymbol{m}^T\boldsymbol{C}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{W}\dot{\boldsymbol{m}}$$

then the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ can be defined as the set of all flow vectors $\boldsymbol{x} = \{\boldsymbol{m},\dot{\boldsymbol{m}}\}$ consistent with $\{\boldsymbol{C},\boldsymbol{W}\}$. That is, the manifold describes the set of all vectors $\boldsymbol{x}$ for which $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = 0$. By this definition

$$\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}} = \{\boldsymbol{x} : f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = 0\}. \tag{4.8}$$

We term an optical flow field *consistent* if there exist motion matrices $\{\boldsymbol{C},\boldsymbol{W}\}$ such that the differential epipolar equation is satisfied for every vector. An optical flow field is thus consistent if it forms part of a manifold of the form described above. We define also, at this point, the associated manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W},k}$ such that $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W},k} = \{\boldsymbol{x} : f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = k\}$.

From the expansion of $f_{\boldsymbol{C},\boldsymbol{W}}(\{\boldsymbol{m},\dot{\boldsymbol{m}}\})$ in (4.1) and from (4.8) notice that the equation $f_{\boldsymbol{C},\boldsymbol{W}}(\{\boldsymbol{m},\dot{\boldsymbol{m}}\}) = 0$ is quadratic in the elements of $\boldsymbol{m}$ and linear in the elements of $\dot{\boldsymbol{m}}$. In fact, due to the nature of (4.1), the differential epipolar equation, and therefore the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$, can be seen as representing a generalised conic section in the 6-dimensional space of $\boldsymbol{x}$.

In Section 4.1.1 it was shown that, given any eight optical flow vectors, we can find motion matrices such that the differential epipolar equation is satisfied for every vector, provided that the cubic constraint $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$ is ignored. Consequently, it is possible to construct a manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ which passes through any set of eight optical flow vectors by a judicious selection of the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$. Whether or not these matrices satisfy the cubic constraint is another matter.

4.4 The problem statement

We now restate the problem of the estimation of motion matrices from optical flow in a manner more suitable to visualisation in terms of distance measures.

Firstly, we assume that there is some underlying 'true' optical flow field that represents the true motion of the image points across the image plane. Measurement of optical flow necessarily introduces noise into the data. It is this noise that the following robust estimation techniques seek to overcome.

If we represent a component of the true optical flow field for an image as $\bar{\boldsymbol{x}}$ then the corresponding observed optical flow vector $\boldsymbol{x}$ can be written as $\boldsymbol{x} = \bar{\boldsymbol{x}} + \Delta\bar{\boldsymbol{x}}$ where $\Delta\bar{\boldsymbol{x}}$ is the error in the estimate. Letting $\bar{\boldsymbol{C}}$ and $\bar{\boldsymbol{W}}$ represent the true motion matrices, we have

$$f_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}(\bar{\boldsymbol{x}}) = 0,$$

but in general

$$f_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}(\boldsymbol{x}) \neq 0.$$

This is due to the fact that the observed optical flow vector $\boldsymbol{x}$, having been contaminated by noise (represented by $\Delta\bar{\boldsymbol{x}}$), does not, in general, lie on the manifold $\mathcal{F}_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}$.

It is not possible to recover the true motion flow vector $\bar{\boldsymbol{x}}$ from its measurement $\boldsymbol{x}$ because $\Delta\bar{\boldsymbol{x}}$ is unknown. It is, however, possible to recover an estimate of the true value. We label this estimate $\widehat{\boldsymbol{x}}$. The true motion matrices are similarly unrecoverable, but it is possible to recover estimates of these matrices from a set $\mathcal{S}$ of n optical flow vectors $\mathcal{S} = \{\boldsymbol{x}_i : i = 1 \ldots n\}$ where $n \geq 7$. Label these estimates $\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{W}}$.

The motion matrices are defined only up to a scale factor so, as described in Section 2.3, it is the ratio of the elements of $\boldsymbol{C} : \boldsymbol{W}$ that is important rather than their particular values. It is important to note that the ratio $\boldsymbol{C} : \boldsymbol{W}$ can be identified with the pair $\{\boldsymbol{C}, \boldsymbol{W}\}$ by the specification of a particular normalisation condition, so that estimates of $\boldsymbol{C} : \boldsymbol{W}$ can always be expressed in terms of normalised pairs $\{\boldsymbol{C}, \boldsymbol{W}\}$. The choice of normalisation condition does not affect the ratio $\boldsymbol{C} : \boldsymbol{W}$ and is therefore somewhat arbitrary.

The manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ describes the set of all optical flow corresponding to the motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$. The manifold therefore represents the expected value of the optical flow given the motion matrices. We have shown in Section 4.2.1 that the maximum likelihood estimate corresponds to the model minimising the sum of the squares of the distance between the observed data and their expected values given the model. We therefore define a manifold to be the best fit to the data when the sum of the squares of distances between the data and the manifold is minimal (see Figure 4.1). There is no limit to the number of distance measures that may be constructed. All that is required is that they provide some measure of the degree to which a particular data element fails to fit the model in question. Unfortunately all distance measures are not equally appropriate. The problem of selecting the best estimate of the motion matrices thus becomes that of the selection and application of an appropriate distance measure.

The process of determining the maximum likelihood estimate of $\boldsymbol{C}$ and $\boldsymbol{W}$ can be restated as that of finding the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ which best fits the observed optical flow data.

We stated in Section 4.3 that the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ is a generalised conic section. We see now that the problem of estimating the motion matrices from optical flow is thus a generalisation of the problem of fitting conic sections to a set of points.

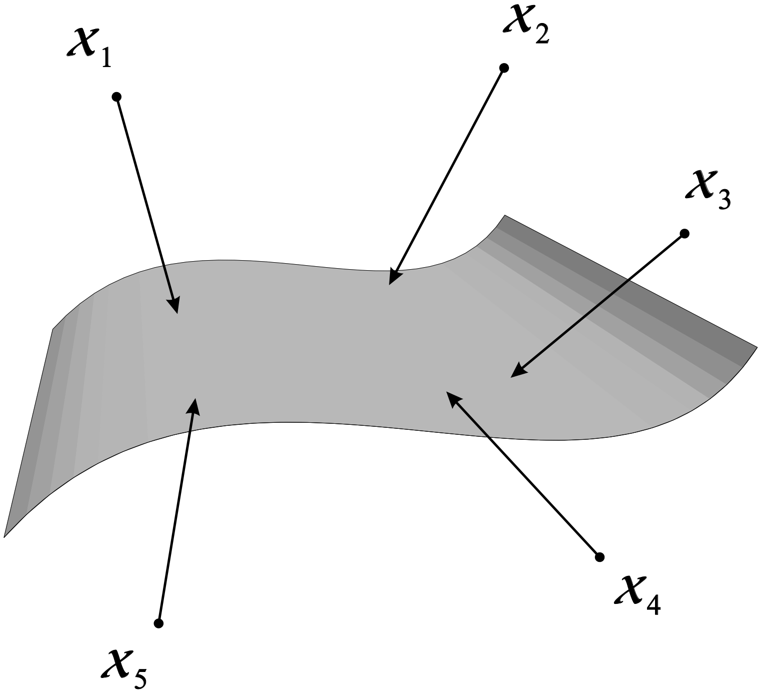

Figure 4.1: The manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$

4.5 The ordinary least squares solution

Substituting each vector in an optical flow field of size n into the differential epipolar equation generates a system of linear homogeneous equations

$$\boldsymbol{m}_i^T \boldsymbol{W} \dot{\boldsymbol{m}}_i + \boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_i = 0, \quad i = 1 \ldots n.$$

If $n \geq 8$ this system provides $n - 1 \geq 7$ constraints for $\boldsymbol{C} : \boldsymbol{W}$ as only the ratio of the elements is important. Unfortunately, in the presence of noise, it is unlikely that there exist normalised motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ such that $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_i) = 0$ for $i = 1 \ldots n$ when $n \geq 8$. We have stated the need for a measure of the degree to which a data element does not conform to a particular model– a distance measure. A simple expression for the distance between a particular optical flow vector $\boldsymbol{x}_i$ and its expected value as defined by $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ would therefore be $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_i)$, the residual of the differential epipolar equation at $\boldsymbol{x}_i$. We label this distance measure δ_1 where

$$\delta_1(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x}_i) = \delta_1(\boldsymbol{C}, \boldsymbol{W}; \{\boldsymbol{m}_i, \dot{\boldsymbol{m}}_i\}) = \boldsymbol{m}_i^T \boldsymbol{W} \dot{\boldsymbol{m}}_i + \boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_i.$$

The corresponding cost function $\mathcal{J}_1$ is

$$\mathcal{J}_1(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S}) = \sum_{i=1}^{n} \delta_1(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x}_i)^2. \tag{4.9}$$

Minimising $\mathcal{J}_1$ thus serves as a basis for estimating $\boldsymbol{C} : \boldsymbol{W}$ but does not guarantee that the solution will satisfy the cubic constraint (2.50).

The distance δ_1 is linear in the entries of $\boldsymbol{C}$ and $\boldsymbol{W}$, so it is possible to separate the data from the model. Let

$$\boldsymbol{u}_i = \begin{bmatrix} m_{i,1}^2 \\ 2m_{i,1}m_{i,2} \\ 2m_{i,1}m_{i,3} \\ m_{i,2}^2 \\ 2m_{i,2}m_{i,3} \\ m_{i,3}^3 \\ m_{i,3}\dot{m}_{i,2} - m_{i,2}\dot{m}_{i,3} \\ m_{i,1}\dot{m}_{i,3} - m_{i,3}\dot{m}_{i,1} \\ m_{i,2}\dot{m}_{i,1} - m_{i,1}\dot{m}_{i,2} \end{bmatrix}.$$

Using the definition of $\boldsymbol{\Theta}$ from Section 4.1.2 it is possible to rephrase δ_1 as

$$\delta_1(\boldsymbol{\Theta}; \boldsymbol{u}_i) = \boldsymbol{u}_i^T \boldsymbol{\Theta}$$

and $\mathcal{J}_1$ as

$$\mathcal{J}_1(\boldsymbol{\Theta}; \mathcal{S}) = \sum_{i=1}^{n} \left(\boldsymbol{u}_i^T \boldsymbol{\Theta}\right)^2.$$

Constructing the matrix

$$\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^T,$$

$\mathcal{J}_1$ becomes

$$\mathcal{J}_1(\boldsymbol{\Theta}; \mathcal{S}) = (\boldsymbol{U}\boldsymbol{\Theta})^T (\boldsymbol{U}\boldsymbol{\Theta}). \tag{4.10}$$

We seek the normalised $\{\boldsymbol{C}, \boldsymbol{W}\}$ for which $\mathcal{J}_1$ is minimal. As stated above, we are at liberty to choose any particular normalisation condition. In order to simplify the mathematics, we select the condition that $\frac{1}{2}\left|\left|\boldsymbol{\Theta}\right|\right|^2 = 1$. We then use the Lagrange multiplier technique to find the $\boldsymbol{\Theta}$ satisfying the constraint that $\frac{1}{2}\left|\left|\boldsymbol{\Theta}\right|\right|^2 = 1$ for which $\mathcal{J}_1(\boldsymbol{\Theta}; \mathcal{S})$ is minimal. We label this estimate $\widehat{\boldsymbol{\Theta}}$.

The Lagrange multiplier method is a commonly used technique for finding the extrema of an objective function within a region described by a constraint equation. The method is based on the knowledge that the extrema of the system in question occur at points at which gradient of the objective function is perpendicular to the surface represented by the constraint equation. At these points the derivatives of the objective function and the constraint equation are parallel but may not be of the same magnitude. In seeking out these points we set the derivative of the cost function equal to the normal of the constraint equation multiplied by some unknown constant λ. In this case this yields

$$\boldsymbol{U}^T\boldsymbol{U}\widehat{\boldsymbol{\Theta}} = \lambda\widehat{\boldsymbol{\Theta}}.$$

Note that $\widehat{\boldsymbol{\Theta}}$ is therefore an eigenvector of $\boldsymbol{U}^T\boldsymbol{U}$, and λ the corresponding eigenvalue, so from (4.10)

$$\mathcal{J}_1(\widehat{\boldsymbol{\Theta}};\mathcal{S}) = \widehat{\boldsymbol{\Theta}}\boldsymbol{U}^T\boldsymbol{U}\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Theta}}\lambda\widehat{\boldsymbol{\Theta}} = \lambda\left|\left|\widehat{\boldsymbol{\Theta}}\right|\right|^2 = 2\lambda.$$

The estimate $\widehat{\boldsymbol{\Theta}}$ which minimises the cost function $\mathcal{J}_1(\boldsymbol{\Theta};\mathcal{S})$ is thus the eigenvector corresponding to the least eigenvalue of $\boldsymbol{U}^T\boldsymbol{U}$. This eigenvector can be efficiently calculated by employing the method of singular value decomposition on the matrix $\boldsymbol{U}$. It is important in determining this eigenvector to avoid calculating $\boldsymbol{U}^T\boldsymbol{U}$ since the condition number of this matrix is the square of that of $\boldsymbol{U}$. This higher condition number significantly decreases the accuracy possible in the determination of $\widehat{\boldsymbol{\Theta}}$.

The vector $\widehat{\boldsymbol{\Theta}}$ thus corresponds to the ordinary least squares estimate of $\bar{\boldsymbol{\Theta}}$, representing the true motion matrices. Unfortunately there is no guarantee that the estimate will satisfy the constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$.

4.5.1 The problem with algebraic distances

Section 4.5 describes the ordinary least squares method of estimating $\boldsymbol{C} : \boldsymbol{W}$; that is, it provides a means of finding $\boldsymbol{\Theta}$, which minimises

$$\mathcal{J}_1(\boldsymbol{\Theta};\mathcal{S}) = \sum_{i=1}^{n}\left(\boldsymbol{u}_i^T\boldsymbol{\Theta}\right)^2.$$

The process of measuring optical flow is imperfect and thus necessarily introduces some error, or noise, into the data. Optical flow exists in the image plane and thus is measured with reference to the image based coordinate frame. Any associated noise is therefore most easily characterised with reference to this frame. Unfortunately, the residual $\delta_1(\boldsymbol{\Theta};\boldsymbol{u}_i) = \boldsymbol{u}_i^T\boldsymbol{\Theta}$ has no obvious geometric significance in this frame because the relationship between the elements of $\boldsymbol{u}_i$ and those of $\{\boldsymbol{m},\dot{\boldsymbol{m}}\}$ is non-linear.

Fundamentally, a residual, or distance measure, represents the degree to which a specific data element does not conform to a particular model. Due to its lack of geometric significance in the image based coordinate system, we label $\delta_1(\boldsymbol{\Theta};\boldsymbol{u}_i)$ as an algebraic residual, or equivalently an algebraic distance measure. All useful distance measures are of course algebraic in nature. We label this one as such only to signify its lack of geometric significance.

Interestingly, in the 9-dimensional space spanned by the vector $\boldsymbol{u}$, the distance

$$\delta_1(\boldsymbol{\Theta};\boldsymbol{u}_i) = \boldsymbol{u}_i^T\boldsymbol{\Theta}$$

can be represented geometrically. The residual represents the perpendicular distance from the point $\boldsymbol{u}_i$ to the hyperplane perpendicular to the vector $\boldsymbol{\Theta}$. Figure 4.2 depicts the hyperplane $\mathcal{F}_{\boldsymbol{\Theta}}$, its normal vector $\boldsymbol{\Theta}$ and a number of data points $\boldsymbol{u}_i$. The perpendicular distance from a point to the hyper-plane corresponds to the inner product of the normal vector $\boldsymbol{\Theta}$ and the location of the point $\boldsymbol{u}$. This is predicated on the fact that the hyper-plane passes through the origin O of the coordinate frame in which $\boldsymbol{u}$ is represented. This representation is put forward in the stereo case by Torr and Murray [128]. Unfortunately, as is stated above, the mapping between optical flow vectors in the image space and the points $\boldsymbol{u}_i$ in this 9-dimensional space is non-linear. One of the consequences of this is that a simple model of the noise associated with the location of points in an image will take on this non-linearity when transfered to the 9-dimensional data space.

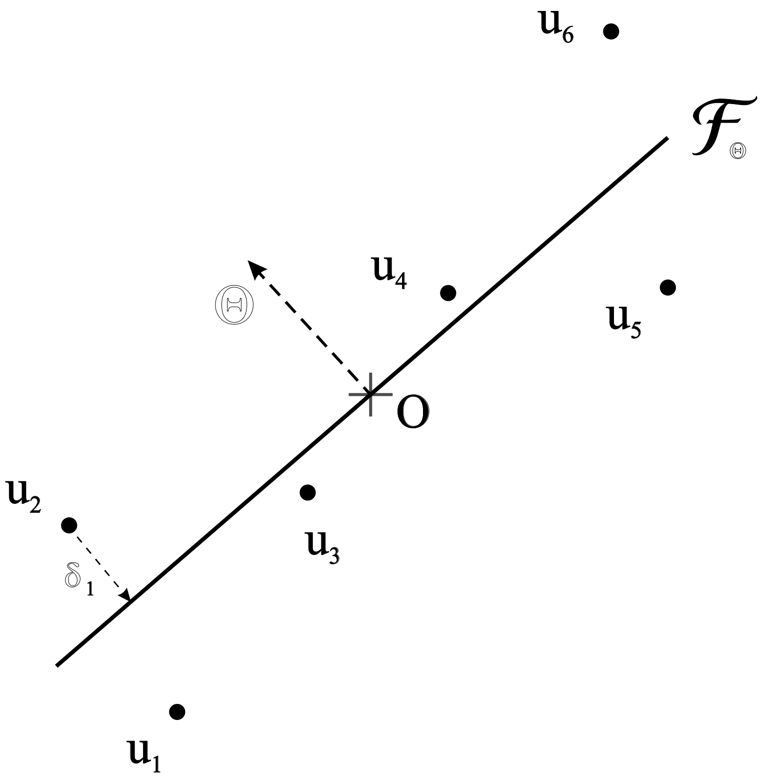

Figure 4.2: Perpendicular distance to the hyperplane in 9 dimensions

Total least squares minimisation requires that we know the orthogonal distance from each data point to the manifold representing the model. Having determined that δ_1 is an algebraic distance measure in the frame in which the data is measured we term the process outlined in Section 4.5 an ordinary least squares method for estimating the motion matrices from optical flow. This is despite the fact that in the coordinate frame spanned by the vector $\boldsymbol{u}$, the residual δ_1 is an orthogonal distance measure which underlies total least squares schemes.

There have been many reports of the advantages of using geometric rather than algebraic distance measures in the computer vision literature, see for example Refs. [84, 95, 128, 129, 144]. Some of these results have been based on the results of Pearson's [104] work on fitting lines and planes suggesting that orthogonal distance measures are essential when noise affects every element of a set of measurements. Within this literature there are two major problems that have been identified with algebraic distance measures. First, that they are not necessarily invariant to Euclidean transformations, and, second, that they have no obvious geometric significance [68, 127, 128, 143].

We have discussed the issue of lack of geometric significance in the previous section. We now consider invariance to Euclidean transformations. A residual which is invariant to Euclidean transformations returns the same result before and after the application of a Euclidean transformation to data and the proposed model. Such a residual is thus sensitive only to the relative orientation of model and data. If an estimator is invariant to Euclidean transformations the only effect of rotating the data and moving it sideways will be that the estimate produced will be similarly transformed. If the method is not so invariant then the estimate based on the transformed data will not be such a simple representation of that produced from the original data. The distinction between relative and absolute orientation is made in the space in which the data is measured, so, the fact that δ_1 is not easily representable in this space implies that it is unlikely to exhibit the required invariance.

A subsidiary problem with algebraic distances also described in the computer vision literature is that induced by the varying curvature of the manifold to be fitted. In Section 4.4 we showed that recovering motion matrices from optical flow is a generalisation of one of the fundamental problems in the field of computer vision, namely that of fitting conic sections to scattered data. It was shown by Bookstein [15] that the method of fitting conic sections using the algebraic residual is more sensitive to points close to low curvature areas of the conic. Bookstein showed that the algebraic distance of a point to a conic section is proportional to d_1^2/d_2^2-1, where d_1 is the distance of the point to the conic's centre O, and d_2 is the distance to the conic along the line towards O (see Figure 4.3). From this it may be seen that points which are the same distance from the conic will register greater algebraic distances as they approach its minor axis. The algebraic distance used in that case is just the equation of the conic section, which is much the same as the use of the differential epipolar equation in the ordinary least squares method given above. The conclusions reached by Bookstein therefore transfer directly to the current problem implying that optical flow data close to low curvature areas of the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ have a greater impact on the

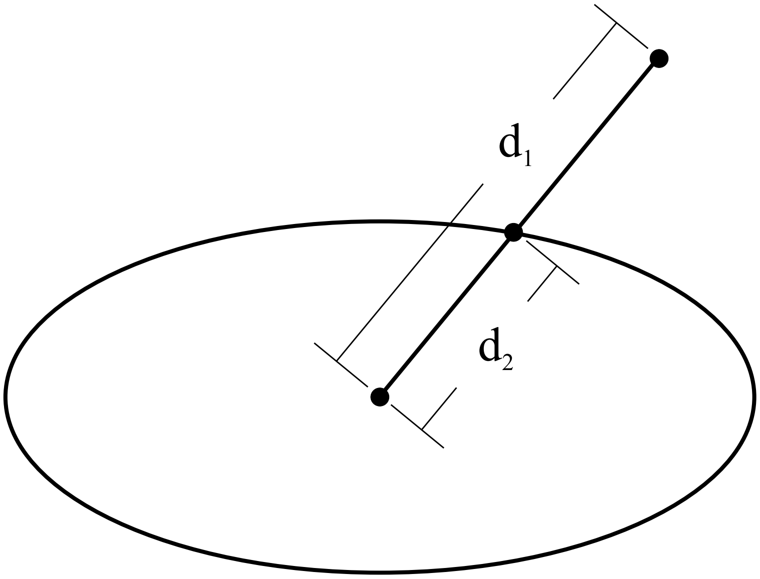

Figure 4.3: The Bookstein distances

final solution than data close to areas exhibiting greater surface curvature. See Section 5.2 for a more detailed explanation of this effect. Differential weighting of data on the basis of proximity to low curvature parts of the manifold is obviously an undesirable property in an estimator.

4.6 Total least squares

We have seen that the ordinary least squares approach minimises the sum of the squares of an algebraic distance measure. It is well known, however, that the maximum likelihood estimate in the quadratic curve fitting problem is the one that minimises the sum of squares of geometric distances to the data points [73]. The total least squares approach thus seeks to minimise the sum of the squares of the geometric distances. On this basis we now derive a geometric distance measure.

4.6.1 A geometric distance measure

In Section 4.3 we defined $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ to be the manifold of all optical flow conforming to the model represented by the motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$. In Section 4.5 we "flattened" the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ to the vector $\boldsymbol{\Theta}$. The two representations are equivalent, so we let $\mathcal{F}_{\boldsymbol{\Theta}} = \mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$. The true motion matrices representing the actual key parameters may thus be represented as $\bar{\boldsymbol{\Theta}}$ and the associated manifold containing the true optical flow field $\mathcal{F}_{\bar{\boldsymbol{\Theta}}}$. This manifold encompasses not only the true optical flow field but every optical flow vector satisfying the differential epipolar equation based on the true motion matrices. Define $\tilde{\boldsymbol{x}}_i$ to be the point on this manifold closest to an optical flow vector $\boldsymbol{x}_i$ (see Figure 4.4). The Euclidean distance between $\boldsymbol{x}_i$ and the manifold is therefore $||\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i||$. We label this distance as

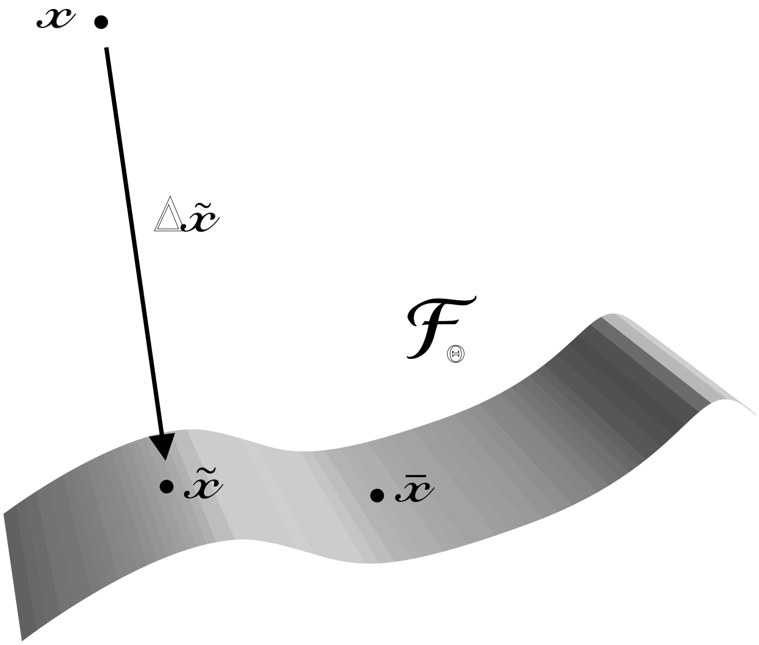

Figure 4.4: The closest point on the manifold

$$\delta_2(\boldsymbol{\Theta}; \boldsymbol{x}_i) = ||\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i||$$

On this basis we define $\mathcal{J}_2(\boldsymbol{\Theta}, \mathcal{S})$ to be the sum of the squares of all Euclidean distances between the manifold defined by $\boldsymbol{\Theta}$, and the set of points $\mathcal{S} = \{\boldsymbol{x}_i | i = 1 \ldots n\}$, so

$$\begin{aligned}\mathcal{J}_2(\boldsymbol{\Theta}, \mathcal{S}) &= \sum_{i=1}^{n} \delta_2(\boldsymbol{\Theta}; \boldsymbol{x}_i) = \sum_{i=1}^{n} ||\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i||^2 \\ &= \sum_{i=1}^{n} (\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i)^T (\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i) = \sum_{i=1}^{n} (\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i)^2.\end{aligned}$$

The total least squares solution is thus $\widehat{\boldsymbol{\Theta}}$ such that

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \mathcal{J}_2(\boldsymbol{\Theta}, \mathcal{S})$$

for a given set of vectors $\mathcal{S}$, subject to the constraints that $f_{\boldsymbol{\Theta}}(\tilde{\boldsymbol{x}}_i) = 0$ for $i = 1 \ldots n$ and $||\boldsymbol{\Theta}||^2 = 2$. Once again there is no guarantee that this solution will satisfy the cubic constraint from equation (2.50).

4.6.2 An image based residual

We described the process of measuring optical flow in Section 1.6.3, and we noted that it is this process which introduces the error into our flow data. Rather than minimise the sum of the squares of the distances to the manifold we now estimate the motion matrices by trying to compensate for this noise in the data.

If the optical flow measurement procedure is reasonably accurate then we know that our n-element observed flow field will be close to the true one. We also know that the true optical flow field satisfies the differential epipolar equation. On this basis we create the set of all n-element optical flow fields which satisfy the differential epipolar equation for any $\boldsymbol{C}$ and $\boldsymbol{W}$. We then select the member of this set which is closest to our observed field in the hope that this is either the true optical flow field or very close to it. This optical flow field will be consistent for some pair of motion matrices. It is these matrices that we select as our estimate.

We thus, by the method above, calculate the minimal change we would have to make to the observed optical flow field in order that it satisfy the differential epipolar equation. It is important to note that this change is an estimate of the error in the optical flow estimation process and thus must be limited to the image plane. The vectors $\boldsymbol{m}_i$ and $\dot{\boldsymbol{m}}_i$ have three elements, but only the first two are represented in the image plane. We therefore restrict the change to the first two elements in both cases. So, finally, we seek the optical flow field closest to our data (but still in the image plane) and which satisfies the differential epipolar equation for some $\boldsymbol{C}$ and $\boldsymbol{W}$. We label the elements of this closest optical flow field $\left\{\tilde{\boldsymbol{m}}, \tilde{\dot{\mathrm{m}}}\right\}$ noting that the third elements of $\tilde{\boldsymbol{m}}$ and $\tilde{\dot{\mathrm{m}}}$ are fixed at 1 and 0 respectively.

We define $\Delta\tilde{\boldsymbol{m}}_i$ and $\Delta\tilde{\dot{\mathrm{m}}}_i$ such that

$$\begin{aligned} \boldsymbol{m}_i &= \tilde{\boldsymbol{m}}_i + \Delta\tilde{\boldsymbol{m}}_i \\ \dot{\boldsymbol{m}}_i &= \tilde{\dot{\mathrm{m}}}_i + \Delta\tilde{\dot{\mathrm{m}}}_i. \end{aligned} \tag{4.11}$$

The vectors $\tilde{\boldsymbol{m}}$ and $\tilde{\dot{\mathrm{m}}}$ are defined such that they satisfy the differential epipolar equation so there exist $\boldsymbol{C}$ and $\boldsymbol{W}$ such that

$$\tilde{\boldsymbol{m}}_i^T \boldsymbol{W} \tilde{\dot{\mathrm{m}}}_i + \tilde{\boldsymbol{m}}_i^T \boldsymbol{C} \tilde{\boldsymbol{m}}_i = 0, \forall i.$$

There is an infinity of $\{\boldsymbol{C}, \boldsymbol{W}\}$ pairs and for each pair an infinity of sets of n optical flow vectors which satisfy the differential epipolar equation. Each of these flow fields is specified with reference to our observed optical flow by selecting different sets of $\Delta\tilde{\boldsymbol{m}}_i$ and $\Delta\tilde{\dot{\mathrm{m}}}_i$ vectors. The magnitude of the change to each vector in the field is

$$d_2\left(\left\{\tilde{\boldsymbol{m}}_i, \tilde{\dot{\mathrm{m}}}_i\right\}\right) = \sqrt{||\Delta\boldsymbol{m}_i||^2 + ||\Delta\dot{\boldsymbol{m}}_i||^2}$$

and, therefore, due to (4.11)

$$d_2\left(\left\{\tilde{\boldsymbol{m}}_i, \tilde{\dot{\mathrm{m}}}_i\right\}\right) = \sqrt{\left\|\boldsymbol{m}_i - \tilde{\boldsymbol{m}}_i\right\|^2 + \left\|\dot{\boldsymbol{m}}_i - \tilde{\dot{\mathrm{m}}}_i\right\|^2}.$$

If we let $\tilde{\boldsymbol{x}}_i = \left\{\tilde{\boldsymbol{m}}_i, \tilde{\dot{\mathrm{m}}}_i\right\}$ and $\Delta\tilde{\boldsymbol{x}}_i = \left\{\Delta\tilde{\boldsymbol{m}}_i, \Delta\tilde{\dot{\mathrm{m}}}_i\right\}$ then

$$\begin{aligned} d_2(\tilde{\boldsymbol{x}}_i) &= \sqrt{\|\Delta\tilde{\boldsymbol{x}}_i\|^2} \\ &= \sqrt{\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|^2} \\ &= \delta_2(\boldsymbol{\Theta}; \boldsymbol{x}_i). \end{aligned}$$

So the measure of the magnitude of change is the same as for the total least squares case in Section 4.6.1. This does not mean that by minimising $\mathcal{J}_2(\boldsymbol{\Theta}; \mathcal{S})$ we find the required estimate. In Section 4.6.1 the vectors $\tilde{\boldsymbol{x}}_i$ were not constrained to lie in the image plane, and could vary freely over the 6 dimensions in which the manifold $\mathcal{F}_{\boldsymbol{\Theta}}$ was defined. In the case of this image based residual we consider only movement in the image plane, so only the first two elements of $\Delta\tilde{\boldsymbol{m}}_i$ and $\Delta\tilde{\dot{\mathrm{m}}}_i$ are of interest. These elements of $\Delta\tilde{\boldsymbol{m}}_i$ and $\Delta\tilde{\dot{\mathrm{m}}}_i$ correspond to elements $1, 2, 4$ and 5 of $\Delta\tilde{\boldsymbol{x}}_i$. We thus define a new norm such that

$$|||\boldsymbol{a}||| = \sqrt{a_1^2 + a_2^2}$$

for a vector $\boldsymbol{a} = [a_1, a_2, a_3]^T$ and

$$|||\boldsymbol{a}||| = \sqrt{a_1^2 + a_2^2 + a_4^2 + a_5^2}$$

for a vector $\boldsymbol{a} = [a_1, a_2, a_3, a_4, a_5, a_6]^T$. Using this notation the required distance measure is

$$\begin{aligned} \delta_3(\boldsymbol{\Theta}; \boldsymbol{x}_i) &= \sqrt{|||\Delta\tilde{\boldsymbol{m}}_i|||^2 + \left|\left|\left|\Delta\tilde{\dot{\mathrm{m}}}_i\right|\right|\right|^2} \\ &= \sqrt{|||\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i|||^2}. \end{aligned}$$

The optical flow field representing the smallest deviation, in the image plane, from our observed flow is thus the field for which the sum of the squares of the $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x}_i)$ is minimal. So we seek the set $\tilde{\mathcal{S}}$ of points $\{\tilde{\boldsymbol{x}}_i = \left\{\tilde{\boldsymbol{m}}_i, \tilde{\dot{\mathrm{m}}}_i\right\} | i = 1 \ldots n\}$, which minimises

$$\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S}) = \sum_i \delta_3(\boldsymbol{\Theta}; \boldsymbol{x}_i)^2$$

subject to the condition that

$$f_{\boldsymbol{\Theta}}(\tilde{\boldsymbol{x}}_i) = 0, \quad \forall\, \tilde{\boldsymbol{x}}_i \in \tilde{\mathcal{S}}. \tag{4.12}$$

The resulting estimate of the motion matrices is the $\boldsymbol{\Theta}$ for which equation (4.12) holds.

4.7 Total least squares minimisation

We have constructed two residuals based on the Euclidean, and therefore geometric, distance between a data point and a manifold. We now consider methods for finding the motion matrices which minimise the associated cost functions $\mathcal{J}_2(\mathbf{\Theta}; \mathcal{S})$ and $\mathcal{J}_3(\mathbf{\Theta}; \mathcal{S})$. The methods apply equally to both cost functions. In order to indicate this fact the generic form $\mathcal{J}(\mathbf{\Theta}; \mathcal{S})$ is minimised.

4.7.1 Finding the set of closest points

An obvious approach to solving a problem of this form would be to find the set $\tilde{\mathcal{S}} = \{\tilde{\boldsymbol{x}}_i : i = 1 \ldots n\}$ which minimises $\mathcal{J}(\mathbf{\Theta}; \mathcal{S})$ subject to the constraint that $f_{\mathbf{\Theta}}(\tilde{\boldsymbol{x}}_i) = 0$ for $i = 1 \ldots n$. The estimate of the motion matrices, represented by $\widehat{\mathbf{\Theta}}$, is that $\mathbf{\Theta}$ implied by the particular set $\tilde{\mathcal{S}}$ selected. In algorithmic terms the method is as follows:

1. Generate the set of all possible normalised 9-vectors $\mathbf{\Theta}$

2. For every element $\mathbf{\Theta}_i$ of this set;

   2.1 Generate the set of all possible n element optical flow fields consistent with $\mathbf{\Theta}_i$,

   2.2 For every such flow field calculate $\mathcal{J}(\mathbf{\Theta}_i; \mathcal{S})$,

3. Select as our estimate the $\mathbf{\Theta}_i$ with the smallest value for $\mathcal{J}(\mathbf{\Theta}_i; \mathcal{S})$.

The problem with this approach is that it requires generating every possible $\mathbf{\Theta}$ and then for each $\mathbf{\Theta}$ generating every possible n element set of data. Generating every possible set of data for each $\mathbf{\Theta}$ is necessary because we have no way of telling which point on the manifold $\mathcal{F}_{\mathbf{\Theta}}$ is closest to a particular data point $\boldsymbol{x}_i$. Unfortunately generating every possible n element set of data is impractical, if not impossible.

In order to alleviate this problem, we now devise a means of determining the closest point $\tilde{\boldsymbol{x}}_i$, on the manifold $\mathcal{F}_{\mathbf{\Theta}}$ to $\boldsymbol{x}_i$. The location of this closest point then leads to a measure of the distance between $\boldsymbol{x}_i$ and the manifold. We then devise a method of alleviating the necessity of generating every possible $\mathbf{\Theta}$.

4.7.2 The distance to the manifold

Our current formulae for the distance from a point $\boldsymbol{x}$ to a manifold $\mathcal{F}_{\mathbf{\Theta}}$ are

$$\delta_2(\mathbf{\Theta}; \boldsymbol{x}) = \sqrt{\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2},$$
$$\delta_3(\mathbf{\Theta}; \boldsymbol{x}) = \sqrt{|||\boldsymbol{x} - \tilde{\boldsymbol{x}}|||^2},$$

where $\tilde{\boldsymbol{x}}$ is the closest point on the manifold to $\boldsymbol{x}$. Rather than apply these formulae to every point on a manifold to determine the closest one, we seek a method of determining $\tilde{\boldsymbol{x}}$, and therefore $\delta(\tilde{\boldsymbol{x}})$, directly. For simplicity we select $\delta_2(\boldsymbol{\Theta};\boldsymbol{x})$ as the distance formula to be minimised, but the method applies equally to $\delta_3(\boldsymbol{\Theta};\boldsymbol{x})$.

The point $\tilde{\boldsymbol{x}}$ is defined such that $f_{\boldsymbol{\Theta}}(\tilde{\boldsymbol{x}}) = 0$ and $\delta_2(\boldsymbol{\Theta};\boldsymbol{x})$ is minimal. In order to simplify the mathematics we choose to minimise $\frac{1}{2}(\boldsymbol{x}-\tilde{\boldsymbol{x}})^2$ rather than $\sqrt{\left\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\right\|^2}$. This change of objective function obviously has no effect on the result. Introducing the Lagrange multiplier λ we have

$$(\boldsymbol{x}-\tilde{\boldsymbol{x}})+\lambda\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})}{\partial\tilde{\boldsymbol{x}}}=\mathbf{0}. \tag{4.13}$$

Substituting into our constraint

$$f_{\boldsymbol{C}\boldsymbol{W}}(\tilde{\boldsymbol{x}})=0$$

to eliminate $\tilde{\boldsymbol{x}}$ we arrive at an expression that is polynomial in λ. If we restrict $\tilde{\boldsymbol{m}}$ and $\tilde{\mathrm{m}}$ to the image plane as suggested in Section 4.6.2, this polynomial is of order eight, otherwise it is of order 10. These polynomials are not presented here due to the complicated nature of the coefficients. Obviously, we cannot find the roots of such polynomials algebraically. We must rely on a numerical polynomial solver. Such a solver will calculate either eight or ten roots as appropriate, each real root corresponding to a possible $\tilde{\boldsymbol{x}}$. We then select the value for $\tilde{\boldsymbol{x}}$ which is closest to our data point $\boldsymbol{x}$ as our estimate which allows us to calculate $\delta_2(\boldsymbol{\Theta};\boldsymbol{x})$ or $\delta_2(\boldsymbol{\Theta};\boldsymbol{x})$ as required.

4.7.3 A total least squares algorithm

Having determined a means of calculating the distance of a point to the manifold we need to clarify how we will utilise it. Recall that we wish to select the motion matrices minimising the sum of the squares of the distances from data to the corresponding manifold. Ideally we would like the distance measure to be of such a form as to enable algebraic determination not only of each distance, but also of the minimal sum of squares of distances given the data. Obviously this is not possible when our distance measure is solvable only by numerical algorithms. We thus require some other method of determining the motion matrices for which the sum of squares of distances is minimal. One possible approach would be to determine this sum for all values of $\boldsymbol{\Theta}$:

1. Generate the set of all possible normalised 9-vectors $\boldsymbol{\Theta}$

2. For every element $\boldsymbol{\Theta}_i$ of this set

   2.1 For every data point $\boldsymbol{x}_j$

      2.1.1 Calculate $\delta(\boldsymbol{\Theta}_i;\boldsymbol{x}_j)$ the distance to manifold $\mathcal{F}_{\boldsymbol{\Theta}_i}$

   2.2 Calculate sum of squares of distances

3. Select $\boldsymbol{\Theta}_i$ corresponding to smallest sum of squares of distances

Obviously the time required to cycle through every possible $\boldsymbol{\Theta}$ is infinite. We can reduce this to a finite interval by using numerical minimisation in place of steps 1 and 2. We are already committed to using a numerical polynomial solver to determine the distance to the manifold in step 2.1.1. In the course of the numerical minimisation step to replace steps 1 and 2 we may have to calculate the distance $\delta(\boldsymbol{\Theta}; \boldsymbol{x}_j)$ many times. Unfortunately this renders the numerical minimisation so slow as to prohibit any detailed testing.

The fact that it is not possible to estimate the motion matrices using either $\mathcal{J}_2(\boldsymbol{\Theta}; \mathcal{S})$ or $\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S})$ renders meaningful comparison of their merits as cost functions difficult. This comparison is carried out in Chapter 5 on the basis of algebraic approximations to $\delta_2(\boldsymbol{\Theta}; \boldsymbol{x})$ or $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x})$.

4.7.4 An end to direct minimisation

The direct approach to geometric distance minimisation may have failed to provide a feasible means of estimating the motion matrices, but this does not mean that the process has been in vain. The method suggested in Section 4.7.3 has successfully reduced the search space to the 8-dimensional space of all $\boldsymbol{C}$ and $\boldsymbol{W}$ pairs. Using the constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$, this may be further reduced to a 7-dimensional space. This is to be contrasted with the dimensionality of the domain of the method from Section 4.7.2 which was at least four times the number of optical flow vectors. Section 5.1 follows on from this in the formulation of an algebraic approximation to the geometric distance which does not suffer from the problems associated with the polynomial formulation. The accuracy of the polynomial formulation of the distance to the manifold which has so far been assumed is demonstrated in Section 4.8.1.

4.8 A geometric measure of performance

Kendall and Stuart [73], amongst others, have shown that, in the case of fitting conic sections, the conic which minimises the sum of the squares of the orthogonal distances to the data is the maximum likelihood estimate. The sum of squares of geometric distances, therefore, constitutes a good measure of the quality of a particular solution. Despite the fact that using the sum of the squares of the geometric distances as a means of estimating the motion matrices has failed, we can use it as a measure of the success of subsequent algorithms. We show in Section 5.7 that the best results are achieved when the distance between a point and a manifold is measured only in the image plane. For this reason we select $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x})$ as our preferred performance measure rather than $\delta_2(\boldsymbol{\Theta}; \boldsymbol{x})$.

One disadvantage of using the sum of squares of geometric distances as a measure of accuracy of fit is that it has an indirect relationship to our desired result, the key parameters of the camera or the structure of the scene viewed. The effect of this is that it is difficult to tell whether a particular value of the distance represents a small or large error in estimation. These problems are addressed in more detail below, but, for the moment, the advantages of the geometric measure of performance outweigh the disadvantages.

4.8.1 Confirming the accuracy of the geometric distance measure

The accuracy of the polynomial method of determining the geometric distance to a manifold was tested as follows: first 125 manifolds were generated from sets of eight randomly determined optical flow vectors. Each of these original flow vectors was then perturbed by k or $-k$ pixels in the direction of the normal to the manifold. This perturbation was carried out in only the four directions corresponding to movement within the image plane as described in Section 4.6.2. Whether the movement was k or $-k$ pixels was selected randomly, the reason being to ensure that some of the perturbed points fell on both sides of the manifold. The value of k was varied from 0.01 to 100 (this value being represented along the x-axis in Figures 4.5 and 4.6). The distance back to the manifold was then measured for each point, using the polynomial method above, and compared to the known distance k. The average error in the distance estimates for these 1000 points is depicted in Figure 4.5, the variance in Figure 4.6. The graphs show that the polynomial representation of the distance to the manifold is accurate and that solving the polynomial numerically produces acceptable results.

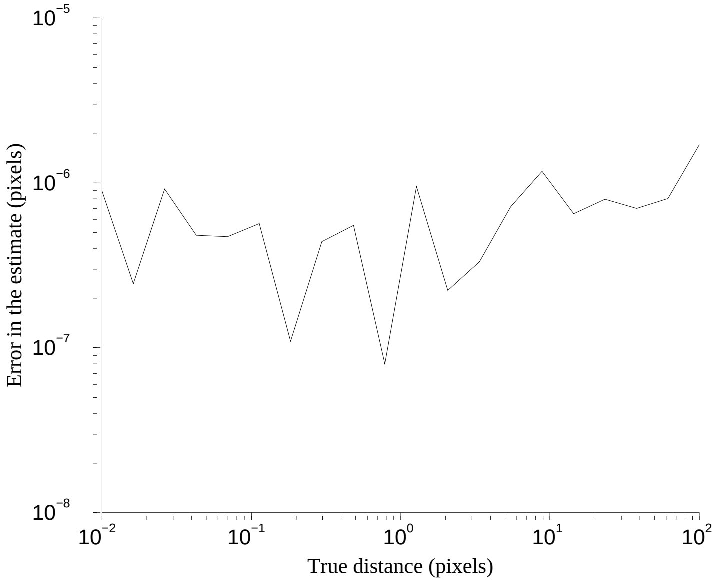

Figure 4.5: Error in the polynomial estimate of the geometric distance measure

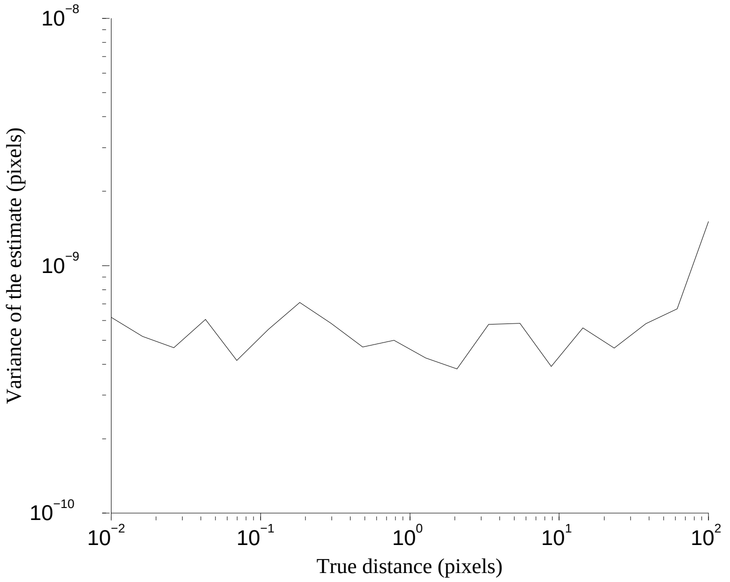

Figure 4.6: Variance of the polynomial estimate of the geometric distance measure

The original optical flow was generated so as to reflect the properties of a real camera, in this case a Pulnix 9701 with a telephoto lens, undergoing reasonable motion. This is important because it limits the shape of the manifold corresponding to the true motion matrices $\mathcal{F}_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}$. Figures 4.7 and 4.8 represent a 100-fold magnification of the space around particular optical flow vectors. The lines in each image represent the projection of the manifold $\mathcal{F}_{\widehat{\boldsymbol{C}},\widehat{\boldsymbol{W}}}$ onto the 2-dimensional image plane achieved by freezing $\dot{\boldsymbol{m}}$ at its true value. The background shading represents the value of the algebraic residual at that point in the image. Once again, this is based on the true value of $\dot{\boldsymbol{m}}$. The darker background colour indicates points with lower residuals.

A priori we would expect that, in some cases, after perturbing an optical flow vector, it would be moved closer to a part of the manifold other than that from which it came (see Figure 4.8). That is, we would expect that, if the point is perturbed far enough, the closest point on the manifold would not be the point's original position. In fact, when the optical flow was generated as specified above, the closest point on the manifold to the perturbed point was always its original position.

Figure 4.8 shows the results of polynomial determination of the closest point on the manifold when the underlying optical flow is generated from general motion matrices. The process that led to Figure 4.7 differed only in the motion matrices from which optical flow was generated. The test represented in Figure 4.7 used camera-based rather than general motion matrices. The differences between these two methods of determining motion matrices are detailed in Appendix A.

During the many thousands of tests carried out, the situation corresponding to Figure 4.8 never occurred when using camera-based motion matrices. For

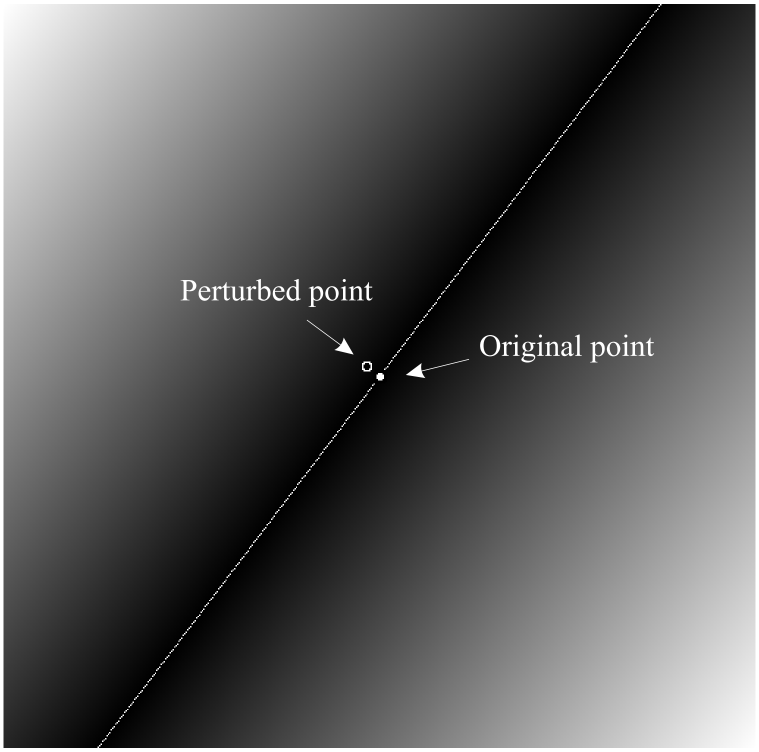

Figure 4.7: A typical perturbation

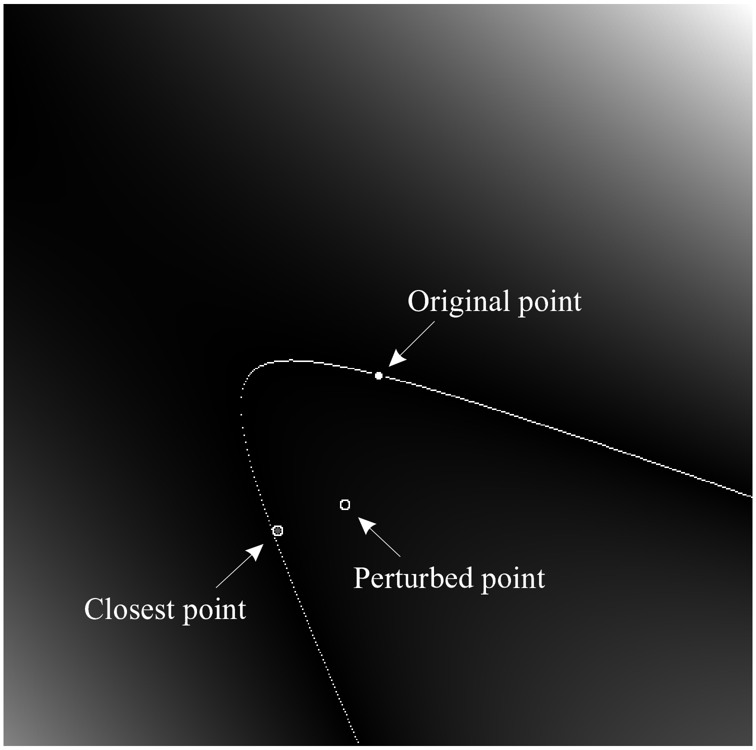

Figure 4.8: An awkward perturbation

general matrices it occurred not often, but repeatably. This would seem to imply that the camera-based motion matrices lead to flatter manifolds than do general motion matrices. The implications of this are discussed in Section 5.11.

4.9 Conclusion

We have defined two methods of estimating the motion matrices based on algebraic techniques applicable when little data is available, or when the data is not corrupted by noise. We have also defined two cost functions, the minimisation of which may provide a means of estimating the motion matrices if the data is corrupted by noise. Unfortunately the brute force method of generating estimates from these cost functions has failed to produce practical estimation methods. We have, however, developed a means of comparing the performance of estimation methods that will later prove useful.

Chapter 5

Approximating the geometric distance

In Section 4.5.1 we showed that, in general, there are advantages in using a geometric rather than an algebraic measure of the distance of a point to a manifold in model fitting problems. In Section 4.7 we provided a total least squares estimation scheme for the motion matrices based on a geometric distance measure. The limitation of this scheme is the fact that calculation of the expression given for the geometric distance requires the use of a numerical polynomial solver. We now seek a measure of the geometric distance for which such a solver is not required. In the course of finding this algebraic representation of the geometric distance from a point to the manifold, we determine a means of comparing $\mathcal{J}_2(\Theta; \mathcal{S})$ and $\mathcal{J}_3(\Theta; \mathcal{S})$.

5.1 An algebraic formulation

We have defined the point $\tilde{\boldsymbol{x}}$ to be the closest point on the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ to $\boldsymbol{x}$. We now seek an algebraic expression of the distance from the point $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ which does not require the use of a numerical polynomial solver. Our final goal is a method of calculating the motion matrices for which the sum of the squares of these distances is minimal. It would be advantageous, therefore, if the algebraic form of the geometric distance were such that it would be possible to calculate these matrices directly. One means of achieving this goal would be to arrive at a formulation within which we may separate our model parameters from our data. Our approach is based on linearising the differential epipolar equation in order to avoid the formation of the polynomial described above, in the hope that this will facilitate direct determination of the distance.

The Taylor series expansion of $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ about the point $\tilde{\boldsymbol{x}}$ can be written as

$$f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}) + \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \tilde{\boldsymbol{x}}) + O((\boldsymbol{x} - \tilde{\boldsymbol{x}})^2). \tag{5.1}$$

We have defined $\tilde{\boldsymbol{x}}$ to lie on the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ so we know that $f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}) = 0$. If the observed flow is sufficiently close to the true flow, we can assume that the

O^2 term is negligible, and thus, to a good approximation that

$$f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \tilde{\boldsymbol{x}}). \tag{5.2}$$

We know that the vector corresponding to the shortest Euclidean distance between a point and a surface strikes the surface at right angles. The vector from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ is, therefore, perpendicular to the surface $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = 0$ at $\tilde{\boldsymbol{x}}$ and parallel to its gradient $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})$, so

$$||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \tilde{\boldsymbol{x}})|| = ||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|| \; ||(\boldsymbol{x} - \tilde{\boldsymbol{x}})|| \,. \tag{5.3}$$

On combining 5.2 and 5.3, we see that

$$||\boldsymbol{x} - \tilde{\boldsymbol{x}}|| = \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||}, \tag{5.4}$$

which is the required Euclidean distance. More generally, if $\mathcal{F}$ is a hypersurface in $\boldsymbol{R}^k$ defined by $\mathcal{F} = \{x \in \boldsymbol{R}^k \mid f(x) = 0\}$ and $z \in \boldsymbol{R}^k$ is a point close to $\mathcal{F}$, then the Euclidean distance between z and $\mathcal{F}$ is, to a first-order approximation, equal to $|f(z)|/\|\nabla f(z)\|$. This fact was first exploited in vision-related statistical formulations by Sampson [109] and later by a number of authors (see for example Refs. [68, 85, 128, 139]).

Unfortunately we do not, a priori, know $||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||$ but we do know $||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})||$, since it may be expressed as

$$\begin{aligned} ||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|| &= \left|\left|\left(\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial \boldsymbol{m}}, \frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial \dot{\boldsymbol{m}}}\right)\right|\right| \\ &= \left|\left|\left(2\boldsymbol{m}^T\boldsymbol{C} - \dot{\boldsymbol{m}}^T\boldsymbol{W}, \boldsymbol{m}^T\boldsymbol{W}\right)\right|\right| \\ &= \sqrt{\left|\left|2\boldsymbol{m}^T\boldsymbol{C} - \dot{\boldsymbol{m}}^T\boldsymbol{W}\right|\right|^2 + ||\boldsymbol{m}^T\boldsymbol{W}||^2} \end{aligned} \tag{5.5}$$

because $\boldsymbol{x} = \{\boldsymbol{m}, \dot{\boldsymbol{m}}\}$. If, as we have assumed, $\boldsymbol{x}$ is sufficiently close to $\tilde{\boldsymbol{x}}$ then it is reasonable to assume that

$$||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|| \approx ||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||\,, \tag{5.6}$$

so we can approximate the distance $\delta_2(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$ by

$$\delta_4(\boldsymbol{C}, \boldsymbol{W}, \boldsymbol{x}) = \frac{\left|\boldsymbol{m}^T\boldsymbol{C}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{W}\dot{\boldsymbol{m}}\right|}{\sqrt{\left|\left|2\boldsymbol{m}^T\boldsymbol{C} - \dot{\boldsymbol{m}}^T\boldsymbol{W}\right|\right|^2 + ||\boldsymbol{m}^T\boldsymbol{W}||^2}}. \tag{5.7}$$

This is a direct algebraic distance measure, based on known quantities, which serves as an approximation to $\delta_2(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$. We have thus determined a geometric residual, the calculation of which does not require a numerical polynomial solver. The new distance measure $\delta_4(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$ is, of course, only an approximation to our desired distance $\delta_2(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$, but it is an approximation which may be easily calculated. The accuracy of the approximation is shown in Section 5.4.

Given $\delta_4(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$ we construct $\mathcal{J}_4(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ which is an approximation to the sum of squares of distances $\mathcal{J}_2(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$

$$\mathcal{J}_4(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S}) = \sum_i \frac{\left(\boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_i + \boldsymbol{m}_i^T \boldsymbol{W} \dot{\boldsymbol{m}}_i\right)^2}{\left|\left|2\boldsymbol{m}_i^T \boldsymbol{C} - \dot{\boldsymbol{m}}_i^T \boldsymbol{W}\right|\right|^2 + \left|\left|\boldsymbol{m}_i^T \boldsymbol{W}\right|\right|^2}. \tag{5.8}$$

The process of linearising $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ for every optical flow vector has therefore led to a replacement for $\mathcal{J}_2(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ with no reference to the points $\tilde{\boldsymbol{x}}$, and thus no longer requiring the use of a numerical polynomial solver. This is significant in that it leads to the possibility of a practical means of estimating the motion matrices based on geometric rather than algebraic distances.

5.1.1 An algebraic approximation to the image based residual

We have made no assumptions about the form of $\tilde{\boldsymbol{x}}$ in the above, but if this closest optical flow vector is constrained to lie in the image plane, as we have suggested, we must set

$$\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial m_3} = \frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial \dot{m}_3} = 0.$$

This is equivalent to projecting the 6-dimensional gradient vector $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ onto the 4-dimensional space corresponding to the first two elements of the vectors $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$. This projection can be carried out by multiplying the components of the gradient corresponding to the derivatives with respect to $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ (from (5.5)) by a matrix $\boldsymbol{P}$ where

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

Recall the notation from Section 4.6.2 whereby

$$|||\boldsymbol{a}||| = \sqrt{a_1^2 + a_2^2}$$

for a vector $\boldsymbol{a} = [a_1, a_2, a_3]^T$ and

$$|||\boldsymbol{a}||| = \sqrt{a_1^2 + a_2^2 + a_4^2 + a_5^2}$$

for a vector $\boldsymbol{a} = [a_1, a_2, a_3, a_4, a_5, a_6]^T$. We see therefore that

$$\boldsymbol{a}\boldsymbol{P}\boldsymbol{a} = |||\boldsymbol{a}|||^2$$

if $\boldsymbol{a}$ is a vector of length 3. Applying this projection to our gradient vector from (5.5) we have

$$\begin{aligned}|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})||| &= \left|\left|\left|\left(\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial \boldsymbol{m}}, \frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{\partial \dot{\boldsymbol{m}}}\right)\right|\right|\right| \\ &= \left|\left|\left|(2\boldsymbol{m}^T\boldsymbol{C} - \dot{\boldsymbol{m}}^T\boldsymbol{W}, \boldsymbol{m}^T\boldsymbol{W})\right|\right|\right| \\ &= \sqrt{\left|\left|\left|2\boldsymbol{m}^T\boldsymbol{C} - \dot{\boldsymbol{m}}^T\boldsymbol{W}\right|\right|\right|^2 + |||\boldsymbol{m}^T\boldsymbol{W}|||^2}\end{aligned} \tag{5.9}$$

This leads to an approximation to δ_3 of the form

$$\delta_5(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x}_i) = \frac{\left|\boldsymbol{m}_i^T\boldsymbol{C}\boldsymbol{m}_i + \boldsymbol{m}_i^T\boldsymbol{W}\dot{\boldsymbol{m}}_i\right|}{\sqrt{\left|\left|\left|2\boldsymbol{m}_i^T\boldsymbol{C} - \dot{\boldsymbol{m}}_i^T\boldsymbol{W}\right|\right|\right|^2 + |||\boldsymbol{m}_i^T\boldsymbol{W}|||^2}}. \tag{5.10}$$

On the basis of δ_5 we construct $\mathcal{J}_5$ which is an approximation to the sum of the squares of the image based geometric residuals $\mathcal{J}_3$,

$$\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S}) = \sum_i \frac{\left(\boldsymbol{m}_i^T\boldsymbol{C}\boldsymbol{m}_i + \boldsymbol{m}_i^T\boldsymbol{W}\dot{\boldsymbol{m}}_i\right)^2}{\left|\left|\left|2\boldsymbol{m}_i^T\boldsymbol{C} - \dot{\boldsymbol{m}}_i^T\boldsymbol{W}\right|\right|\right|^2 + |||\boldsymbol{m}_i^T\boldsymbol{W}|||^2}. \tag{5.11}$$

5.2 Gradient weighted least squares

The ordinary least squares procedure outlined in Section 4.5 minimises the sum of the squares of the algebraic residuals. We showed in Section 4.2.1 that minimising the sum of squares of distances produces the maximum likelihood solution if all data elements have the same variance and are uncorrelated. Unfortunately, assuming equal variance in optical flow elements does not guarantee equal variance in algebraic residuals. We now illustrate this point, in the process providing an alternative derivation for $\mathcal{J}_4(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ and $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$.

The Taylor expansion of $f_{\boldsymbol{C}\boldsymbol{W}}(\boldsymbol{x})$ about a point $\tilde{\boldsymbol{x}}$ on the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ is

$$f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}) + \nabla f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \tilde{\boldsymbol{x}}) + O((\boldsymbol{x} - \tilde{\boldsymbol{x}})^2). \tag{5.12}$$

We know, by the definition of $\tilde{\boldsymbol{x}}$, that $f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}) = 0$, so, if we assume that the $O((\boldsymbol{x} - \tilde{\boldsymbol{x}})^2)$ term is negligible, we expand (5.12) to get

$$f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = \sum_i \frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}}{\partial x_i}(x_i - \tilde{x}_i).$$

If we represent the variance of the elements of an optical flow vector by σ_{x}^2, then the variance σ_f^2 of the algebraic residual $\delta_1(\boldsymbol{C}, \boldsymbol{W}; \mathsf{x}) = f_{\boldsymbol{C},\boldsymbol{W}}(\mathsf{x})$ is

$$\begin{aligned}\sigma_f^2 &= \mathsf{E}(f_{\boldsymbol{C},\boldsymbol{W}}(\mathsf{x}) - f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\mathsf{x}}))^2 \\ &= \sum_i \left(\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}}{\partial \mathsf{x}_i}\right)^2 \mathsf{E}(\mathsf{x}_i - \tilde{\mathsf{x}}_i)^2 \\ &= \left(\sum_i \left(\frac{\partial f_{\boldsymbol{C},\boldsymbol{W}}}{\partial \mathsf{x}_i}\right)^2\right)\sigma_{\mathsf{x}}^2 \\ &= \left|\left|f'_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\mathsf{x}})\right|\right|^2 \sigma_{\mathsf{x}}^2,\end{aligned} \tag{5.13}$$

where $\mathsf{E}(\mathsf{x})$ denotes the expected value of a random variable x. See Zhang [144] for a similar method applied to the general stereo case.

The differential epipolar equation is quadratic in the elements of $\boldsymbol{m}$ so the derivative of $f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})$ is dependent on the value of $\tilde{\boldsymbol{x}}$. Given equation (5.13), this means that the variance of the residual $\delta_1(\boldsymbol{\Theta};\boldsymbol{x})$ is dependent on the value of $\boldsymbol{x}$. Residuals with variance dependent on the value of the associated data point are called *heteroscedastic*. Thus $f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})$ is heteroscedastic. This corresponds to the finding of Sampson [109] that the variance of the algebraic residual of a conic section at a particular point is dependent on the location of that point. The least squares solution is only optimal when residuals are *homoscedastic*, that is, when the residuals exhibit constant and equal variance. This is obviously not the case when the variance of the residual is a function of the data. We see from equation (5.13) that the non-linear representation of the data in the differential epipolar equation leads to heteroscedastic residuals. This non-linearity is represented in the form of the data vector $\boldsymbol{u}$ in Section 4.5.

In Section 4.5.1 we showed that the algebraic residual δ_1 could be seen as measuring the perpendicular distance to the manifold defined by $\boldsymbol{\Theta}$ in the 9-dimensional space of the elements of $\boldsymbol{u}$. Despite this, the associated minimisation method was termed an ordinary, rather than total, least squares method. We have now shown that, in the space in which δ_1 is an orthogonal distance measure, the variance of the data representation, $\boldsymbol{u}$, is heteroscedastic and thus violates the assumptions on which the least squares methods are based. We must determine whether or not a distance measure is geometric in nature in the frame of the original data. It is in this sense that minimising $\mathcal{J}_1$ is an ordinary least squares method.

Given equation (5.13), we see that a first-order approximation to the required correction can be achieved by dividing each residual by its gradient. Once again the derivative of $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ at $\tilde{\boldsymbol{x}}$ is unknown so we construct an approximation to the solution by dividing each residual by the gradient of $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ at $\boldsymbol{x}$:

$$\delta = \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||}. \tag{5.14}$$

This is the gradient weighted least squares method as applied to the stereo case by Weng [140]; it corresponds to the residual δ_4 in (5.7) and therefore to the cost function $\mathcal{J}_4(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ in (5.8). Given that the variances of the third elements of the vectors $\boldsymbol{m}$ and $\dot{\boldsymbol{m}}$ are 0, we apply the projection from (5.9) arriving at δ_5 from (5.10), and $\mathcal{J}_5(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ from (5.11)

5.3 Geometric interpretation

Figure 5.1 gives a geometric interpretation of the approximation to the geometric distance presented in Sections 5.1 and 5.2. For the purposes of visualisation, we have mapped the higher dimensional space of flow tuples into three dimensions.

Recall that we are trying to find a formulation for the distance between the point $\boldsymbol{x}$ and the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ defined by the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$. Our true flow tuple $\bar{\boldsymbol{x}}$ lies on the unknown manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ as does our desired closest point $\tilde{\boldsymbol{x}}$. The observed optical flow tuple $\boldsymbol{x}$ has a residual with respect to $\boldsymbol{C}$ and $\boldsymbol{W}$ of $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) = k$, and lies on the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W},k}$ of all points with that residual. In equation (5.1) we construct a linearisation of $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ about $\boldsymbol{x}$, which corresponds to finding the equation of the plane tangent to the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W},k}$ at this point. We label the linearised $f_{\boldsymbol{C},\boldsymbol{W}}$ as $g_{\boldsymbol{C},\boldsymbol{W}}$. As we have noted, trying to calculate the distance to the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ for which $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{z}) = 0$ is too difficult, but we can find, as an approximation, the distance to the planar manifold $\mathcal{G}_{\boldsymbol{C},\boldsymbol{W}}$ for which $g_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{z}) = 0$. The approximation in equation (5.6) that $||\nabla f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|| \approx ||\nabla f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||$ is equivalent to assuming that the normal to the plane $\mathcal{G}_{\boldsymbol{C},\boldsymbol{W}}$ is also normal to the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ at $\tilde{\boldsymbol{x}}$.

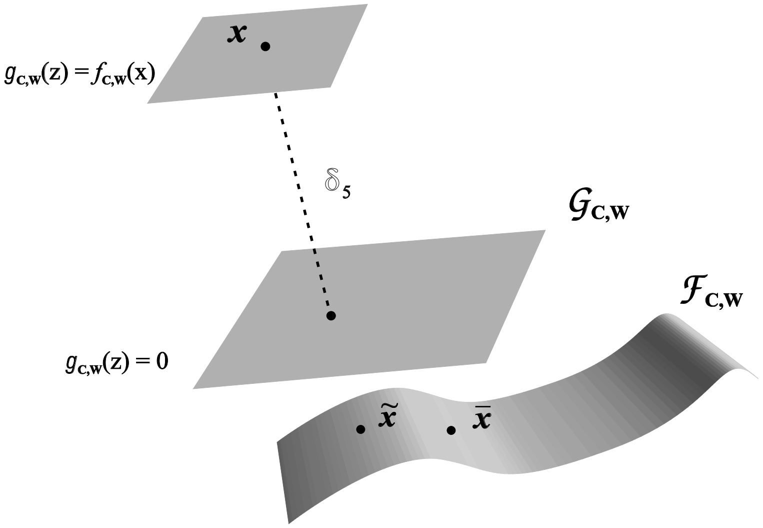

Figure 5.1: Gradient weighted least squares

5.4 Testing the approximated geometric distance

Figures 5.2 and 5.3 were generated in the same manner as those in Section 4.8.1 in that they show the results of perturbing 1000 points by a known distance k from the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$. This perturbation occurred only within the image plane. The distance $\delta_5(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$ was then compared to the known distance k. The graphs show that the average error in the estimate increases with the distance from the manifold, but that even at 100 pixels from the manifold the distance estimate is accurate to 3 significant figures. Similar results have been measured for $\delta_4(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x})$. Comparing the accuracy of the distance approximation as represented in Figures 5.2 and 5.3 with that of the polynomial method represented in Figures 4.5 and 4.6 leads to a surprising result: the approximation is in fact more accurate that the polynomial method for small perturbations. This seems to be an artifact of the numerical process used to determine the roots of the polynomial described in Section 4.8.1.

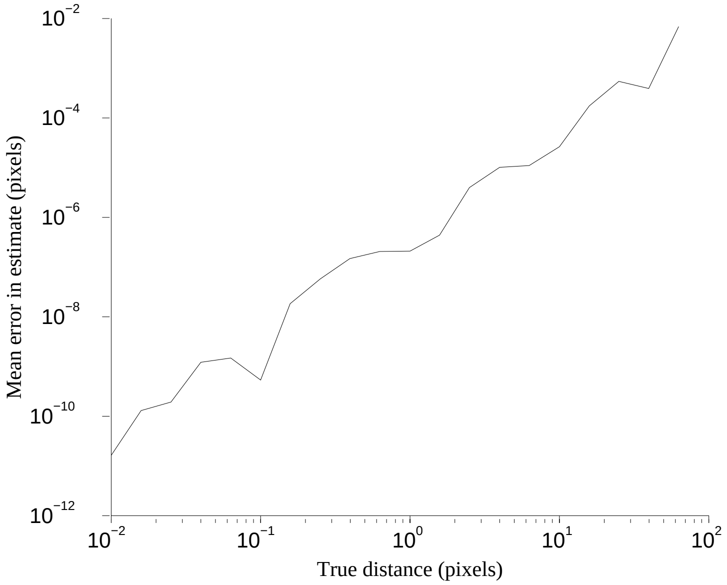

Figure 5.2: Accuracy of the approximation to the geometric distance measure

5.5 Numerical minimisation

We have demonstrated the accuracy of δ_4 and δ_5, the algebraic approximations to the geometric distances δ_2 and δ_3, in Section 5.4. The relative performance of these measures is discussed in Section 5.7 after the derivation of a means of comparison is developed. The result of this comparison is that the distance δ_5 is more appropriate than δ_4. For this reason we henceforth concentrate on this image based residual, but the results are equally applicable to either distance measure. The comparison of distance measures must wait until Section 5.7 because, at present, we have no practical means of using them to generate an estimate, and no independent means of comparing the results such a procedure would produce.

The distance measure δ_5 can now be substituted into the algorithm for estimating the motion matrices developed in Section 4.7.3. This algorithm performs numerical minimisation of a cost function over the range of $\mathbf{\Theta}$, so it calculates an estimate $\widehat{\mathbf{\Theta}}$ such that

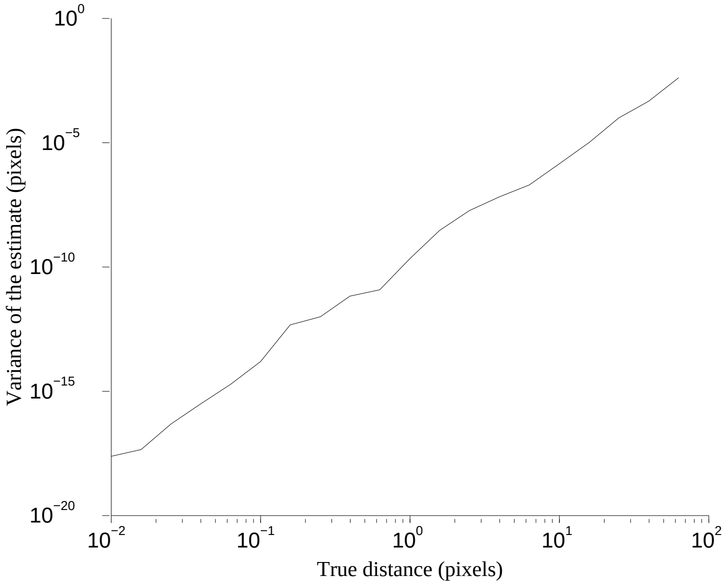

Figure 5.3: Variance of the approximation to the geometric distance measure

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \mathcal{J}(\boldsymbol{\Theta}; \mathcal{S}).$$

The numerical minimisation of $\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S})$ was so slow as to be impractical because calculation of $\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S})$ requires repeated use of a numerical polynomial solver. Calculating $\mathcal{J}_5(\boldsymbol{\Theta}; \mathcal{S})$, in contrast, requires only a simple algebraic operation.

Figure 5.4 shows the results of numerical minimisation of $\mathcal{J}_5(\boldsymbol{\Theta}; \mathcal{S})$, the sum of the squares of this approximated distance measure. The tests were carried out using the procedure outlined in Appendix A.3, and the error measure used is $\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S})$ as suggested by the reading of Section 4.8 in conjunction with Section 5.7. Numerical minimisation of $\mathcal{J}_5(\boldsymbol{\Theta}; \mathcal{S})$ was carried out using the multidimensional direction set method of Powell [106, Chapter10] with the stopping condition being that the residual does not decrease by more than 10^{-8} in any direction. The sum of the squares of the distances to the manifold corresponding to each estimate was then calculated by the method described in Section 4.7.2 as a measure of the accuracy of the estimation process. It is the average of this measure over 50 tests at each noise level that is represented on the y-axis. The ordinary least squares solution from Section 4.5 is presented for comparison. The variance of the added noise is represented along the x-axis, but the absolute magnitude should not be considered indicative. It is one of the consequences of using general, rather than camera-based, motion matrices that no comparison can be made to real cameras and therefore real pixels. Tests involving motion matrices representing more realistic cameras are presented below.

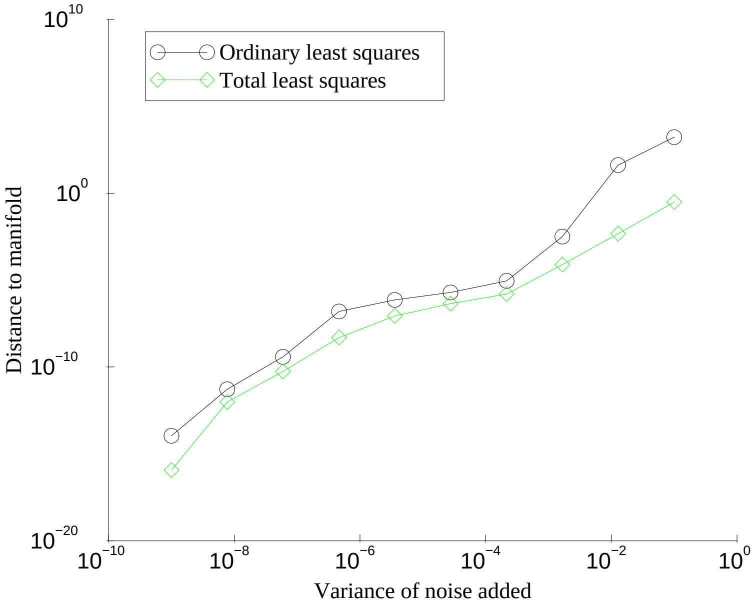

Figure 5.4: Minimising the approximated distance

Figure 5.4 shows the advantages of minimising the geometric distance, but only in terms of geometric distances. It is to be expected that a total least squares approach would result in a lower geometric distance measure. To some extent Figure 5.4, therefore, depicts a self-fulfilling prophecy by virtue of the choice of the measure of the accuracy of the estimates. Figure 5.5 depicts the results of the same tests, but the comparison is made using an alternative accuracy measure, namely, the inner product measure described in Section 5.6. This measure is used to determine the difference between the estimated and the true values of the motion matrices, whereas the measure based on geometric distances refers only to the data and the solution determined. Figure 5.5 shows that minimising the sum of the squares of the geometric distances produces an estimate of the motion matrices which is closer (in terms of the inner product measure) to the true matrices than the ordinary least squares estimate. From Figures 5.4 and 5.5 we conclude not only that $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ is a good estimate of $\mathcal{J}_3(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$, but also that the total least squares approach holds some promise.

5.6 Comparing estimates with the inner product

In Section 4.7.2 we proposed the sum of the squares of the geometric distances as a means of comparing the accuracy of different methods of estimating the motion matrices. This is based on the fact that the sum of the squares of the geometric distances is minimised for the maximum likelihood solution, which is

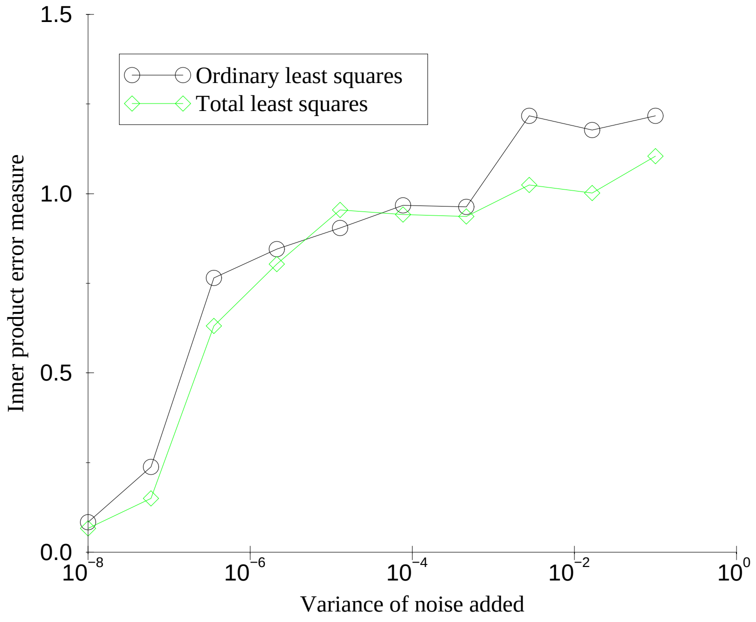

Figure 5.5: Total least squares and the inner product

a definite advantage. Our ultimate goal, however, must always be to recover the estimate of the motion matrices which is closest to the true value. The fact that the geometric distance measure bears no reference to the true solution is both its triumph and its downfall. It can be applied with no knowledge of the true solution, which is essential for tests on real imagery where no ground truth is available. In the case of synthetic testing, however, the true solution is readily available. Provided below is a method of comparing estimated motion matrices with their known true values.

The motion matrices are only defined up to a scale factor, thus when comparing $\{\boldsymbol{C}, \boldsymbol{W}\}$ pairs it is only the ratio $\boldsymbol{C} : \boldsymbol{W}$ which is useful. There are many possible methods of comparing two entities defined only up to a scale factor, some of which are based around the inner product. If we represent each $\boldsymbol{C} : \boldsymbol{W}$ as the vector from the origin to a point on the unit sphere, the inner product of the two vectors represents the cosine of the angle between them. The scale indeterminacy means that we are only interested in the absolute value of the cosine, but even the absolute value of the cosine bears a non-linear relationship to the angle between the two vectors. It is for this reason that we select the arc cosine of the absolute value of the inner product of the two normalised vectors as our error measure.

5.6.1 The scale of the inner product

In order for an error measure to be useful we need a sense of scale; that is, some way of knowing whether a particular value represents a good or a bad result. One useful indicator is the average value of the measure when applied to a randomly selected series of estimates. By calculating this average we generate the value we would expect of the error measure if our estimation process were based on random selection. Naturally, if an estimation method achieves results worse than this value, it is of little use. We now determine the value we would expect to get by applying this inner product error measure to a random guess at the motion matrices.

The expected value of a random variable may be calculated by integrating over the product of the value of the variable and its probability, for all values of the variable. The expected value $E(\phi)$ of a random variable ϕ is thus

$$E(\phi) = \int \phi \, p(\phi) \, d\phi.$$

In order to carry out this calculation we need to know the range of ϕ and the probability of each particular value occurring.

If we repeatedly produce pairs of 2-dimensional random vectors on the unit circle, we expect that the smallest angle between them will vary between 0 and π. If we define these vectors to be invariant of scale, and therefore sign, then the range of values becomes $[0, \pi/2]$. The smallest angle between two scale invariant vectors as described is thus a random variable occurring in the range $[0, \pi/2]$. The average value of such a random variable over a large number of trials is its expected value. Intuitively, the expected value of the smallest angle between two scale invariant vectors is $\pi/4$.

Representing the smallest angle between the 2-dimensional vectors specified above as ϕ, we now calculate its expected value. We have defined ϕ to lie in the range $[0, \pi/2]$ so we need now only calculate the relative probabilities of each angle occurring. The probabilities $p(\phi)$ must sum to 1 by definition, so

$$\int_0^{\frac{\pi}{2}} p(\phi) \, d\phi = 1.$$

We know that all angles in the range are equally likely to occur so the probability of any particular ϕ occurring is $2/\pi$. The expected value of a randomly generated angle ϕ in this range would thus be

$$\begin{aligned} E(\phi) &= \int_0^{\frac{\pi}{2}} \phi \, \frac{2}{\pi} \, d\phi \\ &= \frac{\pi}{4}. \end{aligned}$$

Recall that our inner product error measure is based on the angle between two vectors and so may equally be applied to 2-dimensional vectors. We have shown that the expected value of this error measure when applied to 2 randomly selected 2-dimensional vectors is $\pi/4$. We now extend this result to the 9-dimensional space of the motion matrices.

Specifying the relative position of two vectors in a 9-dimensional sphere requires 8 angles. If we label these angles as ϕ_i for $i = 1 \dots 8$, and the smallest angle between the vectors as ψ, then

$$\psi(\phi_1 \dots \phi_8) = \arccos(\cos\phi_1 \dots \cos\phi_8).$$

We may determine the expected value of $\psi(\phi_1 \dots \phi_8)$ by calculating

$$\begin{aligned} E(\psi) &= \int_{\phi_1=0}^{\phi_1=\frac{\pi}{2}} \dots \int_{\phi_8=0}^{\phi_8=\frac{\pi}{2}} \psi(\phi_1 \dots \phi_8)\, p(\phi_1) \dots p(\phi_8)\; d\phi_1 \dots d\phi_8 \\ &= \int_{\phi_1=0}^{\phi_1=\frac{\pi}{2}} \dots \int_{\phi_8=0}^{\phi_8=\frac{\pi}{2}} \arccos(\cos\phi_1 \dots \cos\phi_8) \left(\frac{2}{\pi}\right)^8 \; d\phi_1 \dots d\phi_8. \end{aligned}$$

This integral is complicated to solve algebraically but using numerical integration techniques we find that $E(\psi) = 1.54362$. This result has been confirmed by repeatedly generating random vectors and measuring the smallest angle between them.

The conclusion drawn from the above is that, when evaluating the merit of a particular estimate of the motion matrices, an inner product error measure greater than 1.54362 is unacceptable. An inner product error measure less than this number does not mean that we have an accurate estimate, but accuracy does increase with decreasing error values.

5.7 Comparing distance-based residuals

In Sections 4.6.1 and 4.6.2 we developed the distance measures $\delta_2(\boldsymbol{\Theta}; \boldsymbol{x})$ and $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x})$ leading to the cost functions $\mathcal{J}_2(\boldsymbol{\Theta}, \mathcal{S})$ and $\mathcal{J}_3(\boldsymbol{\Theta}, \mathcal{S})$. We could not estimate the motion matrices from these cost functions because a numerical polynomial solver was required to calculate every $\delta(\boldsymbol{\Theta}; \boldsymbol{x}_i)$. In Sections 5.1 and 5.2 we derived approximations to these distance measures $\delta_4(\boldsymbol{\Theta}, \boldsymbol{x})$ and $\delta_5(\boldsymbol{\Theta}, \boldsymbol{x})$ which do not require the use of a numerical polynomial solver. On the basis of these approximated distances we generated new cost functions $\mathcal{J}_4(\boldsymbol{\Theta}, \mathcal{S})$ and $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$ finally allowing us, in Section 5.5, to derive a method of estimating the motion matrices. Until our derivation of the inner product error measure in Section 5.6 we had no method for measuring the performance of such estimation methods other than the cost functions themselves. The problem with using the cost functions as an error measure is that it provides no means of comparing between cost functions. The inner product error measure now makes this comparison possible.

Figure 5.5 shows that numerical minimisation of $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$ provides estimates of the motion matrices with a smaller inner product error measure than the ordinary least squares estimate. Importantly this figure also shows that the error measure tends towards 0 as the noise in the data diminishes. Figure 5.6 shows the results of tests carried out using the same methods but over a smaller range of noise magnitudes and for both $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$ and $\mathcal{J}_4(\boldsymbol{\Theta}, \mathcal{S})$. It can be seen from Figure 5.6 that numerical minimisation of the image based cost function $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$ generally produces better results than does numerical minimisation of the full geometric cost function $\mathcal{J}_4(\boldsymbol{\Theta}, \mathcal{S})$. The smaller range of noise magnitudes in Figure 5.6 as compared to Figure 5.5 is due to the fact that the results of the three minimisation processes converge as noise diminishes. Every test produces slightly different output but, in general, it is the results in the range depicted that most distinguish the methods.

On the basis of the above, and the fact that $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x})$ more closely corresponds to the process under which the noise in the data is generated, we henceforth use $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$ as our preferred cost function. The results to follow, however, generally apply to both cost functions. We have shown that the process of numerical minimisation of our selected cost function produces estimates of the motion matrices which are closer to the true value than the ordinary least squares estimates. What has not been shown is that the ordinary least squares method is significantly faster than the numerical minimisation process. In fact the results depicted in Figure 5.6 required 3 days of processing to generate on an AlphaStation 5/266. Creating the same data and calculating only the ordinary least squares estimate requires less than 30 seconds. In Section 5.9.1 we develop a more efficient method of minimising $\mathcal{J}_5(\boldsymbol{\Theta}, \mathcal{S})$.

5.8 Rectifying motion matrices

Having developed the inner product as a method of measuring the distance between sets of motion matrices we are able to measure the effects of the rectification procedure given in Section 2.4.1. The method modifies motion matrices so that they satisfy $\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} = 0$, but does not guarantee to select the matrices satisfying the constraint that are closest to the originals. The effect of rectification on motion matrices was analysed by performing a series of trials using the same methodology as that described in Appendix A.3. Figures 5.7 and 5.8 show the effect of rectification as reported by the geometric distance based measure (from Section 4.8) and the inner product measure respectively.

Figure 5.7 shows the results of ordinary least squares estimation of $\boldsymbol{C}$ and $\boldsymbol{W}$ over many tests at a range of noise levels. In each test the ordinary least squares estimate was calculated on the basis of a new set of synthetically generated data, and the value of the geometric accuracy measure recorded. This process was repeated 50 times for each noise level. The average of these accuracy measures is represented in Figure 5.7. This average was simultaneously calculated for the rectified ordinary least squares estimate. As can be seen, the rectification procedure causes an increase in this distance measure, although the magnitude of the increase is small. Figure 5.8 shows the results of the same process, but as measured by the inner product rather than the geometric error measure. This graph thus depicts the average difference between the estimated and the true

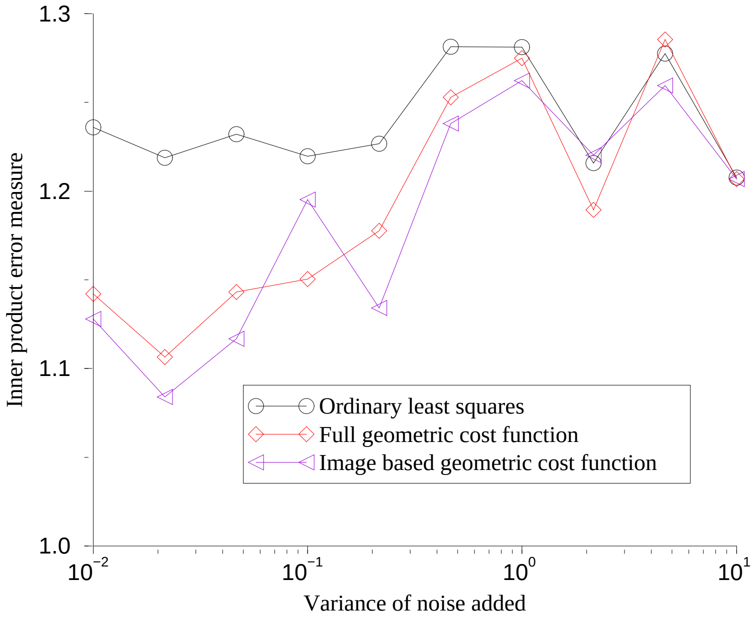

Figure 5.6: Distance based cost functions and the inner product

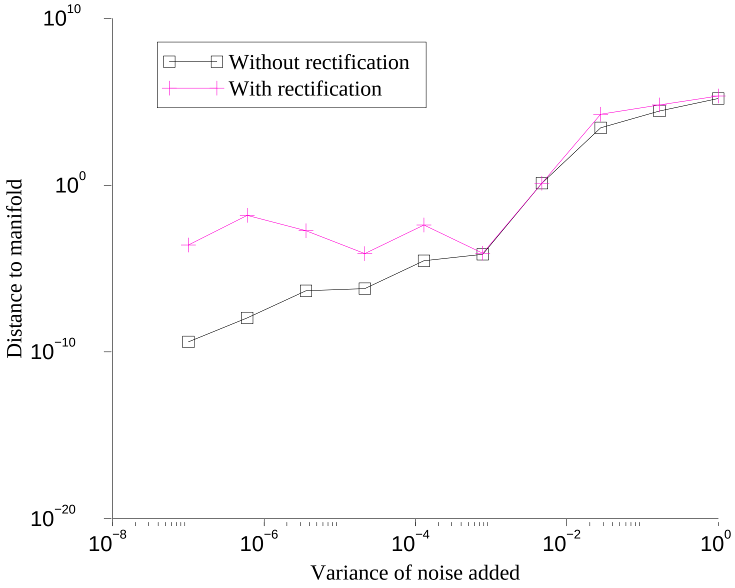

Figure 5.7: Rectified ordinary least squares - distance-based error measure

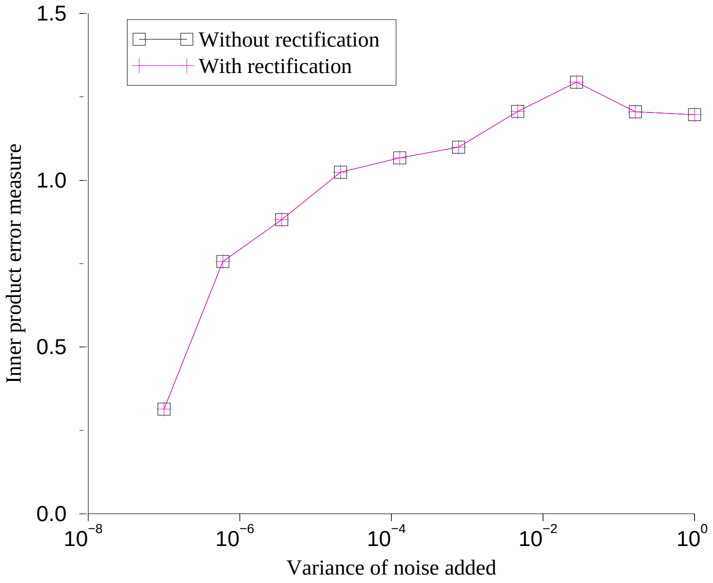

Figure 5.8: Rectified ordinary least squares - inner product error measure

motion matrices over a number of trials. Despite the fact that Figures 5.7 and 5.8 appear on different scales (as necessitated by their ranges) it can be seen that the rectification procedure has little effect on the inner product measure. The implication of this result is that the rectification procedure is having neither significant positive nor negative effects on the quality of estimates produced in terms of the inner product measure.

5.9 Sampson's method

We have shown in Section 5.4 that $\delta_5(\boldsymbol{\Theta}; \boldsymbol{x})$ is a good approximation to $\delta_3(\boldsymbol{\Theta}; \boldsymbol{x})$ and therefore that $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ is a good approximation to $\mathcal{J}_3(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$. The advantage of calculating the sum of the squares of the distances with $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ is that it is an algebraic process, rather than one involving numerical routines. Our overriding aim, however, is to speed the process of determining the motion matrices for which this sum is minimal. Optimally we would be able to determine a formulation for the geometric distance which allows not only algebraic determination of the sum of squares of distances, but also algebraic determination of the ratio $\boldsymbol{C} : \boldsymbol{W}$ corresponding to the minimal sum of squares of distances. This would be feasible if it were possible to separate the model from the data as in the ordinary least squares solution given in Section 4.5. This may be possible of $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$, but, unfortunately, the way ahead is far from clear. Rather than follow this path we therefore seek to eliminate the use of Powell's method thus creating a minimisation scheme capable of delivering the same results but using far less processing time.

5.9.1 Iteratively re-weighted least squares estimator

The process of numerical minimisation of $\mathcal{J}_5(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ is faster than that of numerical minimisation of $\mathcal{J}_3(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$, because each distance calculation is algebraic rather than involving numerical solution of a polynomial. Despite this the minimisation remains a slow process. Rather than embark upon a brute force minimisation technique such as that used in Section 5.5 we present the following as a technique of incrementally updating our estimate of $\boldsymbol{C}$ and $\boldsymbol{W}$.

Assuming that we already have estimates of the motion matrices $\{\underline{\boldsymbol{C}},\underline{\boldsymbol{W}}\}$, let

$$\underline{\mathcal{J}_5}(\boldsymbol{C},\boldsymbol{W};\mathcal{S})=\sum_{i=1}^{n}\underline{\delta_5}(\boldsymbol{C},\boldsymbol{W};\boldsymbol{m}_i,\dot{\boldsymbol{m}}_i)^2,$$

where

$$\underline{\delta_5}(\boldsymbol{C},\boldsymbol{W};\boldsymbol{m},\dot{\boldsymbol{m}})=\frac{|\boldsymbol{m}^T\boldsymbol{W}\dot{\boldsymbol{m}}+\boldsymbol{m}^T\boldsymbol{C}\boldsymbol{m}|}{\sqrt{|||2\underline{\boldsymbol{C}}\boldsymbol{m}+\underline{\boldsymbol{W}}\dot{\boldsymbol{m}}|||^2+|||\underline{\boldsymbol{W}}\boldsymbol{m}|||^2}}.$$

The denominator of the right hand side does not depend on $(\boldsymbol{C},\boldsymbol{W})$, and so minimisation of $\underline{\mathcal{J}_5}(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ subject to the constraint $||\boldsymbol{C}||^2+||\boldsymbol{W}||^2=1$ falls into the category of weighted least squares techniques. Using the Lagrange multiplier technique, as in Section 4.5, we see that the least-square estimate based on $\underline{\mathcal{J}_5}(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ can be identified with the eigenvector of $\boldsymbol{U}^T\boldsymbol{H}_{\underline{\boldsymbol{C}},\underline{\boldsymbol{W}}}\boldsymbol{U}$ corresponding to the smallest eigenvalue. Here, we define the weight matrix $\boldsymbol{H}_{\underline{\boldsymbol{C}},\underline{\boldsymbol{W}}}$ as

$$\boldsymbol{H}_{\underline{\boldsymbol{C}},\underline{\boldsymbol{W}}}=\begin{bmatrix} h_1 & \ldots & 0\\ \multicolumn{3}{c}{\ldots\ldots\ldots\ldots} \\ 0 & \ldots & h_n\end{bmatrix},\ \text{where}$$
$$h_i=\left(|||2\underline{\boldsymbol{C}}\boldsymbol{m}_i+\underline{\boldsymbol{W}}\dot{\boldsymbol{m}}_i|||^2+|||\underline{\boldsymbol{W}}\boldsymbol{m}_i|||^2\right)^{-1}, \tag{5.15}$$

and the data matrix $\boldsymbol{U}$ is as in Section 4.5. We now propose the following iteratively re-weighted least squares procedure that simultaneously seeks to minimise $\mathcal{J}_5(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ and to accommodate the cubic constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w}=0$. The method proceeds by repeatedly calculating the motion matrices which minimise $\underline{\mathcal{J}_5}(\boldsymbol{C},\boldsymbol{W};\mathcal{S})$ and substituting them back into the matrix $\boldsymbol{H}$. The cubic constraint is enforced by application of the rectification method outlined in Section 2.4.1. The iteratively re-weighted least squares procedure is thus as follows:

1. Set $(\boldsymbol{C}_0,\boldsymbol{W}_0)$ to the rectified ordinary least squares solution for our data $\mathcal{S}$ and set $k=0$.

2. Compute the weight matrix $\boldsymbol{H}_{\boldsymbol{C}_k,\boldsymbol{W}_k}$ on the basis of $\boldsymbol{C}_k$, $\boldsymbol{W}_k$ and $\mathcal{S}$.

3. Compute the eigenvector of $\boldsymbol{U}^T\boldsymbol{H}_{\boldsymbol{C}_k,\boldsymbol{W}_k}\boldsymbol{U}$ corresponding to the smallest eigenvalue.

4. Using the eigenvector and the rectification procedure, calculate $(\boldsymbol{C}_{k+1}, \boldsymbol{W}_{k+1})$.
5. If $(\boldsymbol{C}_{k+1}, \boldsymbol{W}_{k+1})$ is sufficiently close to $(\boldsymbol{C}_k, \boldsymbol{W}_k)$, then terminate the procedure; otherwise increment k and return to Step 2.

This method represents a slight modification of that applied to the problem of fitting conic sections by Sampson [109]. The differences between this method and Sampson's are the domain, and the rectification at each step. The domain of Sampson's original method was a set of 2-dimensional points, to which a conic section was to be fitted. The addition of rectification at every iteration (in steps 1 and 4) guides the method towards motion matrices which satisfy the cubic constraint. It is possible that this method will not converge, and that $(\boldsymbol{C}_{k+1}, \boldsymbol{W}_{k+1})$ will never be sufficiently close to $(\boldsymbol{C}_k, \boldsymbol{W}_k)$. In order to prevent this we limit the number of possible iterations, and, when that limit is reached, we compare $\mathcal{J}_5(\boldsymbol{C}_0, \boldsymbol{W}_0, \mathcal{S})$ and $\mathcal{J}_5(\boldsymbol{C}_{k+1}, \boldsymbol{W}_{k+1}, \mathcal{S})$ returning the motion matrices corresponding to the lower value.

5.9.2 Testing Sampson's method

Figures 5.9 and 5.10 show the results of tests comparing two versions of Sampson's method to the numerical minimisation technique from Section 5.5 and the ordinary least squares technique from Section 4.5. The repeated application of the rectification procedure in the method outlined above ensures that any solution will satisfy $\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} = 0$. In order to show the effects of rectification on the minimisation procedure, we have included in the figures the results generated with no rectification applied. We have used the label *unconstrained weighted least squares* to distinguish this method from that utilising the rectification procedure. The tests were performed using the methodology presented in appendix A.3. The stopping condition for Powell's numerical minimisation method is that the residual does not decrease by more than a particular value over an iteration. The stopping condition for Sampson's method however relates to the difference between the sum of the squares of the elements of the motion matrices corresponding to two successive estimates. The performance of each scheme will be affected by the value of the thresholds chosen, but testing suggests that improvements gained by reducing the value below 10^{-8} and 10^{-6}, respectively, are minimal.

The difference between Figure 5.9 and 5.10 is the error measure used. In Figure 5.9 the comparison is made on the basis of the error measure presented in Section 4.8; that is, how well it minimises the sum of the squares of the distances of the data to the manifold. It can be seen that neither version of the iteratively re-weighted estimation scheme reduces $\mathcal{J}_3(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ as well as does the numerical minimisation procedure for higher noise levels.

Figure 5.9 shows the performance of the three estimation schemes according to the inner product error measure. This comparison shows that the results of

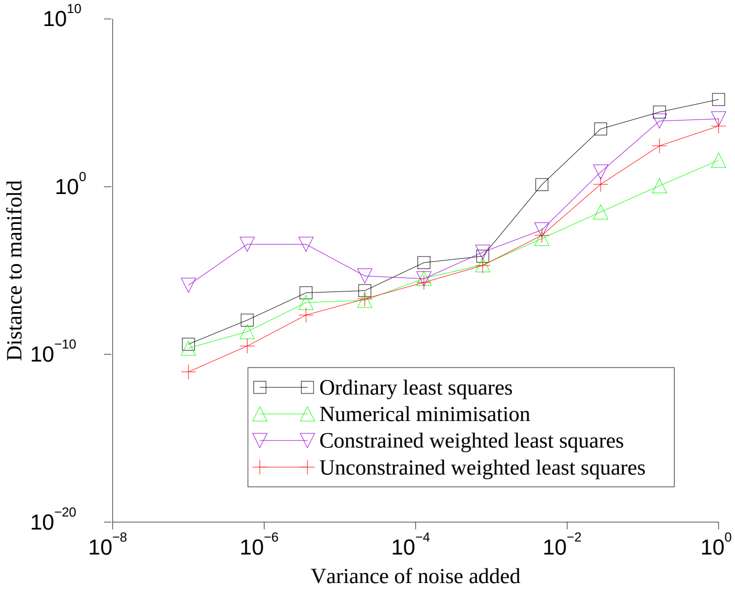

Figure 5.9: Sampson's method - distance to the manifold

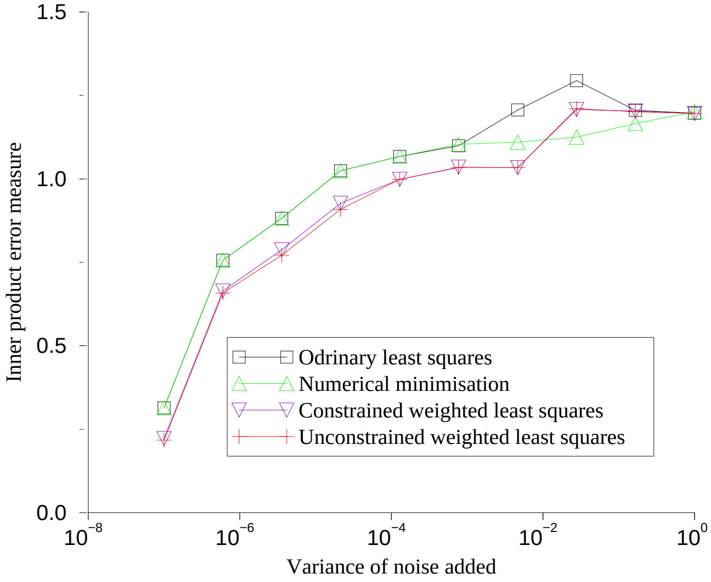

Figure 5.10: Sampson's method - inner product error measure

Sampson's method are closer to the original motion matrices than are those of the numerical minimisation scheme. Thus, although the constrained version of Sampson's method does not appear to minimise $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$) (and therefore $\mathcal{J}_3(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$) as well as the other methods, it produces better estimates of the true motion matrices. It might be thought that this is due to the fact that the estimates are guaranteed to satisfy the cubic constraint that $\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} = 0$. This explanation is, however, contradicted by the fact that the unconstrained version of Sampson's method also out-performs the numerical minimisation scheme. The advantage of Sampson's method disappears as the noise level increases, which suggests that Powell's method may not be suitable for minimising such small residuals. The advantage of the iteratively weighted scheme not shown by either graph is that it requires significantly less execution time than does the numerical minimisation technique.

5.9.3 The problem with Sampson's method

It has been shown by Kanatani [67] and Zhang [144] amongst others that the process of repeatedly fixing the denominator when minimising an expression of the same form as $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ leads to statistical bias. Suppose we were to repeatedly generate noisy data according to some fixed model and apply some process to the data in order to recover an estimate of the model. We would expect that, over a large number of trials, the average of the model estimates would converge to the value of the true model. Statistical bias in an estimator describes the situation in which the average of the estimates converges to some other model. An estimation process exhibiting bias is obviously not statistically optimal. The fact that Sampson's method is biased implies that the estimate generated does not necessarily lead to the $\boldsymbol{C}$ and $\boldsymbol{W}$ for which $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ is minimal. This is to be expected as step 3 operates on $\underline{\mathcal{J}_5}(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ which has a fixed denominator. The variation in the denominator of the cost function is of influence only as steps 2 through 5 are repeated.

The results shown in Figures 5.9 and 5.10 are not affected by this bias because they represent the average over a number of trials, each using a different model. Statistical bias affects every estimate produced under a biased estimator but the magnitude of the bias is dependent on the data present and thus the model under which it is generated. If repeated tests use the same model this bias will be evidenced in the deviation of the average estimate from the true value. If repeated tests use different models then the magnitude will be different in each test, thus canceling out any cumulative effect. The use of multiple models, however, enables analysis of the performance of a method over a range of data models. The process of testing using repeated trials with the same model is susceptible to delivering model dependent results, as some methods exhibit different performance levels on different model types. This is particularly true when the selected model is close to being degenerate.

5.10 A Newton-like method

We now seek an estimation process better able to find the minimum of $\mathcal{J}_5$. Eventually we wish to differentiate $\mathcal{J}_5(\boldsymbol{C}, \boldsymbol{W}; \mathcal{S})$ so our first step is to express $|||2\boldsymbol{C}\boldsymbol{m} + \boldsymbol{W}\dot{\boldsymbol{m}}|||^2 + |||\boldsymbol{W}\boldsymbol{m}|||^2$ as a product of matrices. Towards this goal we let

$$\boldsymbol{\Phi}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{\Phi}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{\Phi}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{\Psi}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\boldsymbol{\Psi}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and}$$

$$\boldsymbol{\Psi}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

A fundamental property of these matrices is that, for each $\alpha \in \{1, 2, 3\}$,

$$\boldsymbol{\Phi}_\alpha \boldsymbol{\Theta} = [c_{\alpha 1}, c_{\alpha 2}, c_{\alpha 3}]^T \quad \text{and} \quad \boldsymbol{\Psi}_\alpha \boldsymbol{\Theta} = [w_{\alpha 1}, w_{\alpha 2}, w_{\alpha 3}]^T.$$

and therefore that

$$\begin{aligned} \boldsymbol{m}^T \boldsymbol{C}^T \boldsymbol{P}^T \boldsymbol{P} \boldsymbol{C} \boldsymbol{m} &= \sum_{\alpha,\beta,\gamma=1}^{3} m_\beta c_{\beta\alpha} p_{\alpha\alpha} p_{\alpha\alpha} c_{\alpha\gamma} m_\gamma \\ &= \sum_{\alpha,\beta,\gamma=1}^{3} c_{\beta\alpha} p_{\alpha\alpha} m_\beta m_\gamma p_{\alpha\alpha} c_{\alpha\gamma} \\ &= \sum_{\delta=1}^{3} \boldsymbol{\Theta}^T \boldsymbol{\Phi}_\delta{}^T \boldsymbol{m}\boldsymbol{m}^T \boldsymbol{\Phi}_\delta \boldsymbol{\Theta} \end{aligned}$$

where c_{ij}, w_{ij} and p_{ij} represent the elements of the matrices $\boldsymbol{C}$, $\boldsymbol{W}$ and $\boldsymbol{P}$ respectively. The matrix $\boldsymbol{P}$ is as described in Section 5.1.1. By the same process we see that

$$\begin{aligned}
\boldsymbol{m}^T\boldsymbol{C}^T\boldsymbol{P}\boldsymbol{W}\dot{\boldsymbol{m}} &= \sum_{\delta=1}^{3} \boldsymbol{\Theta}^T{\boldsymbol{\Phi}_\delta}^T\boldsymbol{m}\dot{\boldsymbol{m}}^T\boldsymbol{\Psi}_\delta\boldsymbol{\Theta},\\
\dot{\boldsymbol{m}}^T\boldsymbol{W}^T\boldsymbol{P}\boldsymbol{W}\dot{\boldsymbol{m}} &= \sum_{\delta=1}^{3} \boldsymbol{\Theta}^T{\boldsymbol{\Psi}_\delta}^T\dot{\boldsymbol{m}}\dot{\boldsymbol{m}}^T\boldsymbol{\Psi}_\delta\boldsymbol{\Theta},\\
\boldsymbol{m}^T\boldsymbol{W}^T\boldsymbol{P}\boldsymbol{W}\boldsymbol{m} &= \sum_{\delta=1}^{3} \boldsymbol{\Theta}^T{\boldsymbol{\Psi}_\delta}^T\boldsymbol{m}\boldsymbol{m}^T\boldsymbol{\Psi}_\delta\boldsymbol{\Theta}.
\end{aligned}$$

Combining these four identities with the fact that

$$\begin{aligned}
|||2\boldsymbol{C}\boldsymbol{m}+\boldsymbol{W}\dot{\boldsymbol{m}}|||^2 + |||\boldsymbol{W}\boldsymbol{m}|||^2 = 4\boldsymbol{m}^T\boldsymbol{C}^T\boldsymbol{P}\boldsymbol{C}\boldsymbol{m} &+ 4\boldsymbol{m}^T\boldsymbol{C}^T\boldsymbol{P}\boldsymbol{W}\dot{\boldsymbol{m}}\\
&+ \dot{\boldsymbol{m}}^T\boldsymbol{W}^T\boldsymbol{P}\boldsymbol{W}\dot{\boldsymbol{m}} + \boldsymbol{m}^T\boldsymbol{W}^T\boldsymbol{P}\boldsymbol{W}\boldsymbol{m},
\end{aligned}$$

we see that

$$|||2\boldsymbol{C}\boldsymbol{m}+\boldsymbol{W}\dot{\boldsymbol{m}}|||^2 + |||\boldsymbol{W}\boldsymbol{m}|||^2 = \boldsymbol{\Theta}^T\boldsymbol{N}_i\boldsymbol{\Theta}, \tag{5.16}$$

where

$$\begin{aligned}
\boldsymbol{N}_i = 4\sum_{\delta=1}^{3} {\boldsymbol{\Phi}_\delta}^T\boldsymbol{m}_i\boldsymbol{m}_i^T\boldsymbol{\Phi}_\delta &+ 4\sum_{\delta=1}^{3} {\boldsymbol{\Phi}_\delta}^T\boldsymbol{m}_i\dot{\boldsymbol{m}}_i^T\boldsymbol{\Psi}_\delta\\
-\sum_{\delta=1}^{3} {\boldsymbol{\Psi}_\delta}^T\dot{\boldsymbol{m}}_i\dot{\boldsymbol{m}}_i^T\boldsymbol{\Psi}_\delta &- \sum_{\delta=1}^{3} {\boldsymbol{\Psi}_\delta}^T\boldsymbol{m}_i\boldsymbol{m}_i^T\boldsymbol{\Psi}_\delta.
\end{aligned}$$

For each data point $\{\boldsymbol{m}_i, \dot{\boldsymbol{m}}_i\}$ we define the matrix $\boldsymbol{M}_i$ such that

$$\boldsymbol{M}_i = \boldsymbol{u}_i{\boldsymbol{u}_i}^T$$

where $\boldsymbol{u}_i$ is as defined in Section 4.5.

Now, in view of equation (5.16),

$$\delta_5(\boldsymbol{\Theta};\boldsymbol{m}_i,\dot{\boldsymbol{m}}_i)^2 = \frac{\boldsymbol{\Theta}^T\boldsymbol{M}_i\boldsymbol{\Theta}}{\boldsymbol{\Theta}^T\boldsymbol{N}_i\boldsymbol{\Theta}}$$

implying that

$$\mathcal{J}_5(\boldsymbol{\Theta};\mathcal{S}) = \sum_{i=1}^{n} \frac{\boldsymbol{\Theta}^T\boldsymbol{M}_i\boldsymbol{\Theta}}{\boldsymbol{\Theta}^T\boldsymbol{N}_i\boldsymbol{\Theta}}. \tag{5.17}$$

Hence, immediately,

$$[\nabla_{\boldsymbol{\Theta}}\mathcal{J}_5(\boldsymbol{\Theta};\mathcal{S})]^T = 2\boldsymbol{X}_{\boldsymbol{\Theta}}\boldsymbol{\Theta}, \tag{5.18}$$

where

$$\boldsymbol{X}_{\boldsymbol{\Theta}} = \sum_{i=1}^{n} \frac{\boldsymbol{M}_i}{\boldsymbol{\Theta}^T\boldsymbol{N}_i\boldsymbol{\Theta}} - \sum_{i=1}^{n} \frac{\boldsymbol{\Theta}^T\boldsymbol{M}_i\boldsymbol{\Theta}}{(\boldsymbol{\Theta}^T\boldsymbol{N}_i\boldsymbol{\Theta})^2}\boldsymbol{N}_i. \tag{5.19}$$

Again the minimiser $\widehat{\boldsymbol{\Theta}}$ satisfies

$$[\nabla_{\boldsymbol{\Theta}} \mathcal{J}_5(\widehat{\boldsymbol{\Theta}}; \mathcal{S})]^T = 2\lambda \widehat{\boldsymbol{\Theta}}$$

for some Lagrange multiplier λ and our selected normalising condition

$$\left|\left|\widehat{\boldsymbol{\Theta}}\right|\right|^2 = 1. \tag{5.20}$$

Combining this with (5.18), we conclude that $\widehat{\boldsymbol{\Theta}}$ is an eigenvector of $\boldsymbol{X}_{\widehat{\boldsymbol{\Theta}}}$ with λ as the associated eigenvalue, so

$$\widehat{\boldsymbol{\Theta}}^T \boldsymbol{X}_{\widehat{\boldsymbol{\Theta}}} \widehat{\boldsymbol{\Theta}} = \lambda \widehat{\boldsymbol{\Theta}}^T \widehat{\boldsymbol{\Theta}} = \lambda.$$

On the other hand, recourse to (5.19) reveals that $\widehat{\boldsymbol{\Theta}}^T \boldsymbol{X}_{\widehat{\boldsymbol{\Theta}}} \widehat{\boldsymbol{\Theta}} = 0$. Therefore $\lambda = 0$ and, consequently,

$$\boldsymbol{X}_{\widehat{\boldsymbol{\Theta}}} \widehat{\boldsymbol{\Theta}} = \boldsymbol{0}. \tag{5.21}$$

We see by comparing equations (5.21) and (5.18) that in fact the normalising condition of equation (5.20) has no effect on the result. The solution to the constrained minimisation problem is the same as that for the unconstrained problem. This is explained by the fact that $\mathcal{J}_5(\widehat{\boldsymbol{\Theta}}; \mathcal{S})$ is immune to scale changes in $\boldsymbol{\Theta}$ as evidenced by its form in (5.17).

Equation (5.21) is a non-linear constraint on $\widehat{\boldsymbol{\Theta}}$ which one might hope to resolve by employing a method of successive approximations of some kind. The following scheme is based on Newton's method:

1. Compute $\boldsymbol{\Theta}_0$ using least-square fitting based on $\mathcal{J}_5(\boldsymbol{\Theta}; \mathcal{S})$.

2. Assuming that $\boldsymbol{\Theta}_{k-1}$ is known, compute the matrix $\boldsymbol{X}_{\boldsymbol{\Theta}_{k-1}}$.

3. Compute a normalised eigenvector of $\boldsymbol{X}_{\boldsymbol{\Theta}_{k-1}}$ corresponding to the smallest eigenvalue and take this eigenvector for $\boldsymbol{\Theta}_k$.

4. If $\boldsymbol{\Theta}_k$ is sufficiently close to $\boldsymbol{\Theta}_{k-1}$, then terminate the procedure; otherwise increment k and return to Step 2.

Observe that, on account of (5.15) and (5.19),

$$\boldsymbol{X}_{\boldsymbol{\Theta}} = \boldsymbol{U}_{\boldsymbol{\Theta}}^T \boldsymbol{H}_{\boldsymbol{\Theta}} \boldsymbol{U}_{\boldsymbol{\Theta}} - \boldsymbol{E}_{\boldsymbol{\Theta}},$$

where

$$\boldsymbol{E}_{\boldsymbol{\Theta}} = \sum_{i=1}^{n} \frac{\boldsymbol{\Theta}^T \boldsymbol{M}_i \boldsymbol{\Theta}}{(\boldsymbol{\Theta}^T \boldsymbol{N}_i \boldsymbol{\Theta})^2} \boldsymbol{N}_i.$$

Therefore $\boldsymbol{X}_{\boldsymbol{\Theta}}$ can be viewed as a modification of $\boldsymbol{U}_{\boldsymbol{\Theta}}^T \boldsymbol{H}_{\boldsymbol{\Theta}} \boldsymbol{U}_{\boldsymbol{\Theta}}$. Accordingly, the estimator embodied by the above algorithm can be viewed as a modification of the iteratively re-weighted least squares estimator from Section 5.9.1.

Unfortunately, when general motion matrices are used rather than those based on a realistic camera model, this algorithm sometimes fails to converge. One of the methods we have used to remedy this is simply to retain the best estimate produced (in terms of $\mathcal{J}_5(\boldsymbol{\Theta}; \mathcal{S})$) rather than the final one in cases where it fails to converge. It is this algorithm which was used to generate the data depicted in Figures 5.11 and 5.12. The results of this modified algorithm when camera-based motion matrices are used are given in Section 5.11. More refined schemes for solving (5.21) may readily be developed. One possibility is a fixed point method obtained by linearising the left-hand side of (5.21) to incorporate the matrix-valued derivative of the mapping $\boldsymbol{\Theta} \mapsto \boldsymbol{X}_{\boldsymbol{\Theta}}$. This work has yet to be carried out.

While in some aspects this Newton-like method resembles Kanatani's technique of renormalisation [68], it differs in that it is formulated in a purely deterministic, probability-free fashion, and that it utilises standard, rather than generalised, eigenvalue analysis. This Newton-like method is also more simply derived and implemented. For a more detailed comparison of Kanatani's method and this Newton-like method see Ref. [33].

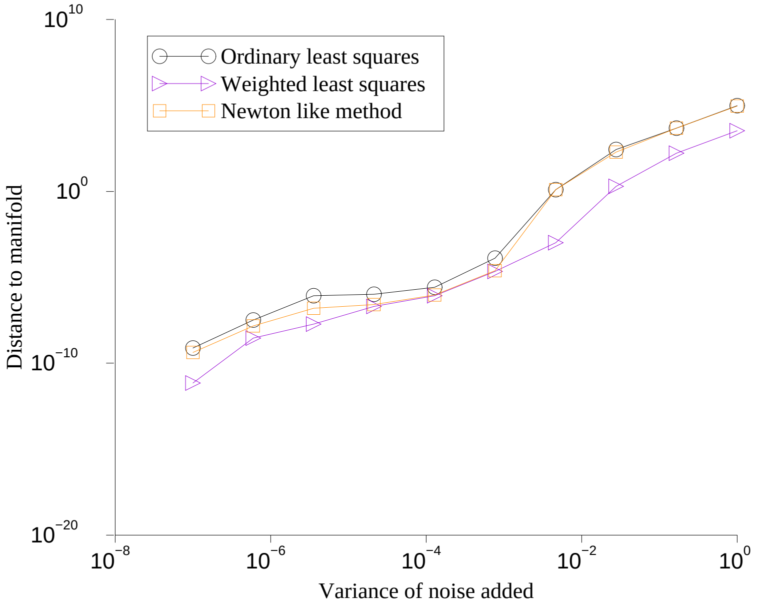

Figure 5.11: Newton-like method - distance-based error measure

Figures 5.11 and 5.12 show the results of applying the ordinary least squares procedure, the weighted least squares procedure and the Newton-like method to the same data. Figure 5.11 uses the $\mathcal{J}_3(\boldsymbol{\Theta}; \mathcal{S})$ based error measure and Figure 5.12 the inner product based error measure. The tests were carried out using the methodology described in Appendix A.3.

Figure 5.11 shows that the Newton-like method minimises $\mathcal{J}_5(\mathbf{\Theta}; \mathcal{S})$ more effectively than the ordinary least squares procedure but, due to the fact that it often fails to converge, it does not perform as well as the weighted least squares method. Figure 5.12 shows similarly that, in terms of the inner product based error measure, the Newton-like method performs better than the ordinary least squares procedure, although not as well as the weighted least squares method. Numerical solution of (5.21) has been tested and converges more quickly than does numerical minimisation of $\mathcal{J}_5(\mathbf{\Theta}; \mathcal{S})$. This increase in speed of convergence was not significant enough to render this numerical method faster than the weighted least squares iterative scheme.

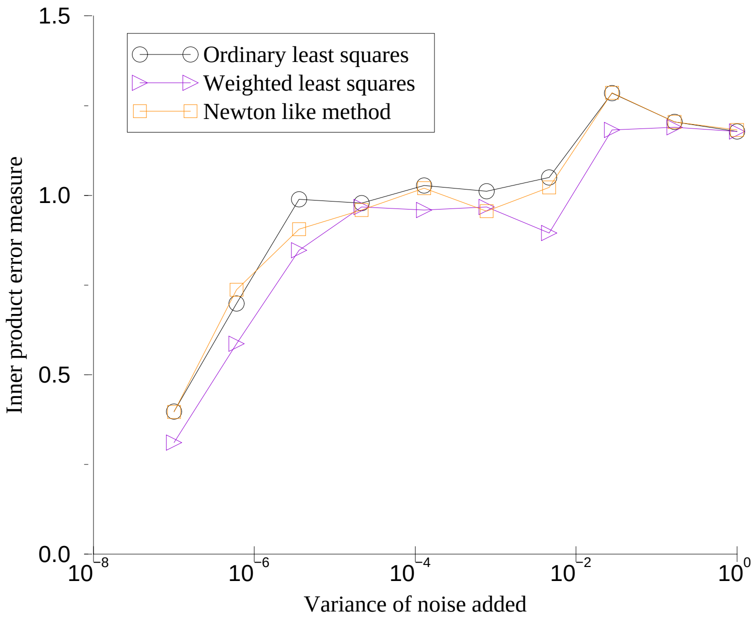

Figure 5.12: Newton-like method - inner product error measure

It is possible to extend this Newton-like method by applying the procedure for the rectification of motion matrices at every step. This is equivalent to the extension to Sampson's method given in Section 5.9.1. The detrimental effects of this constraint on the Newton-like method at low noise levels were greater than those on Sampson's method. This result is depicted in Figure 5.13.

5.11 The applicability of total least squares

The figures above have shown the advantages of using geometric distance measures when calculating motion matrix estimates. These tests have, however, been carried out using data generated under a slightly unrealistic model. The figures represent tests carried out using general motion matrices generated according to the process described in Section A.1.1. There can be no guarantee

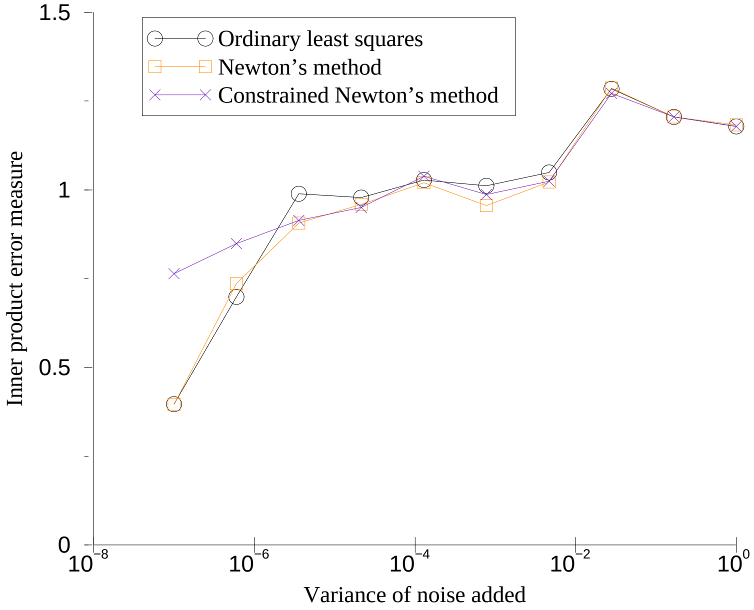

Figure 5.13: Rectification and the Newton-like method

that motion matrices generated in this manner will bear any relationship to any realistic camera. The only test applied after generating such motion matrices is that the associated focal length is real rather than imaginary. The reasons for using general, rather than camera-specific, motion matrices were to ensure that the selection of a particular camera model did not affect the results, and because the results for the general matrices enable better discrimination between estimation methods. Figures 5.14 and 5.15 represent the results of testing the methods outlined above using motion matrices based on the Pulnix 9701 with zoom lens as described in Appendix A.1.2. Figures 5.16 and 5.17 similarly represent the results of the same tests using motion matrices based on a Pulnix TM-6CN with a lens of focal length 8mm. The differences between the two camera models are the CCD sizes, which are 752×582 for the TM-6CN and 1024×1024 for the 9701, and the focal length of the lens, namely 8mm for the TM-6CN and between 8.5 and 51 for the 9701. The CCD size of the 9701 is larger than usual for that model due to a factory installed modification.

The figures in previous sections have shown that significant gains in the accuracy of the estimates of general motion matrices can be achieved using the methods provided. Unfortunately, Figures 5.14, 5.15, 5.16 and 5.17 show that this is not the case when more realistic motion matrices are used. In this situation, gains provided by these procedures are barely enough to justify the increase in complexity and execution time. Figures 5.15 and 5.17 also show that the Newton-like method sometime diverges from its course, thus producing quite erroneous estimates.

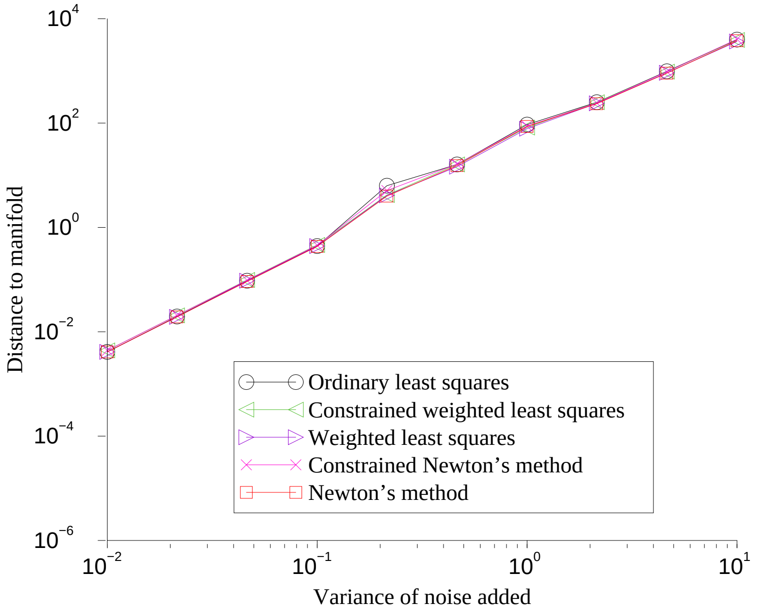

Figure 5.14: The 9701 and the distance based error measure

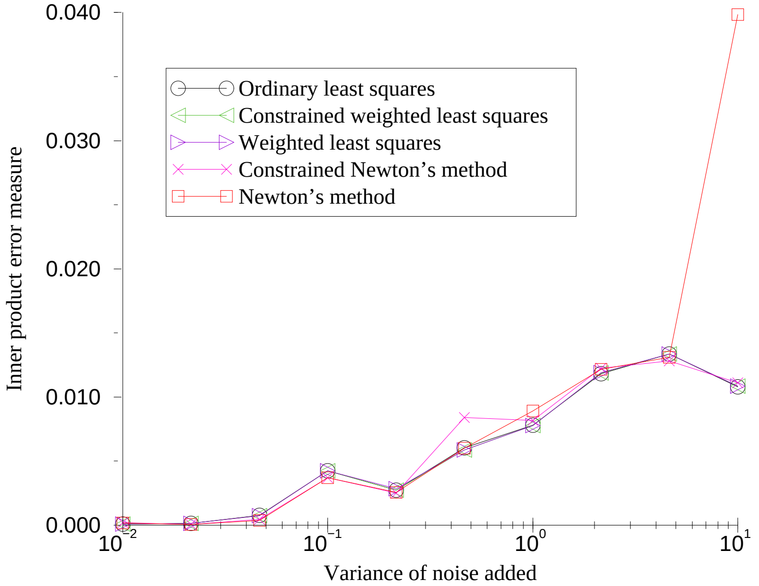

Figure 5.15: The 9701 and the inner product error measure

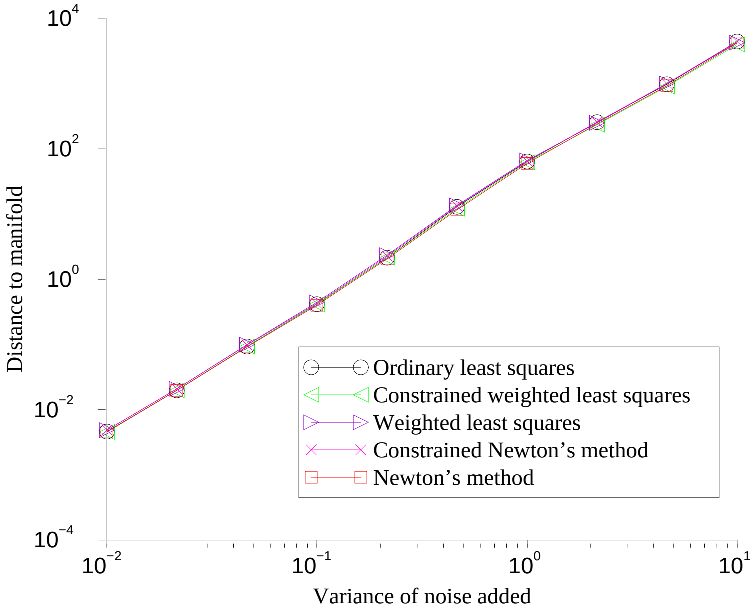

Figure 5.16: The TM6CN and the distance based error measure

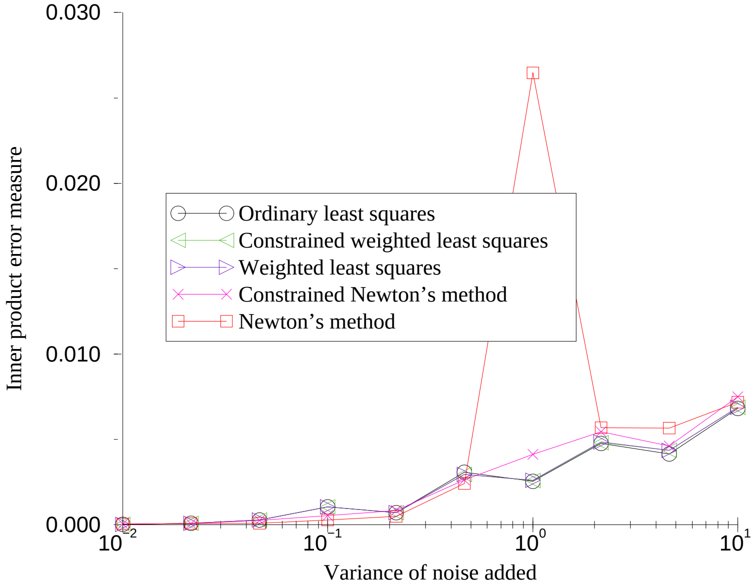

Figure 5.17: The TM6CN and the inner product error measure

5.11.1 How flat is the manifold?

We have seen in Section 5.11 that the results for methods based on geometric distances are practically identical to those for the ordinary least squares method when motion matrices based on real camera models are used. We now show that this is due to the fact that, for realistic camera models, the manifold of all consistent optical flow is relatively flat.

We have shown in Section 5.1 that the geometric distance to a manifold can be approximated by the algebraic distance divided by the norm of the gradient of the manifold at the closest point. Specifically the approximation to the distance of a point $\boldsymbol{x}$ to the manifold is given by

$$\delta_5(\boldsymbol{C}, \boldsymbol{W}; \boldsymbol{x}) = \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})|||}, \tag{5.22}$$

where $\tilde{\boldsymbol{x}}$ is the closest point on the manifold. The role of the denominator in this expression is to compensate for the differences between the algebraic and geometric distances due to the curvature of the manifold. In fact, for tests using realistic motion matrices, this denominator does not vary significantly enough between optical flow vectors to affect the outcome of the minimisation process. Table 5.1 presents the norms of the gradient vectors (the denominator of $\mathcal{J}_5(\Theta; \mathcal{S})$) for five optical flow fields. Each field contains 20 optical flow vectors randomly generated from camera-based motion matrices according to the procedure set out in Appendix A. The table shows that, within the same test, the norms are identical to 1 or 2 significant figures. That is, for a particular pair of motion matrices, the curvature of the manifold does not change significantly over the range of the data. Table 5.2 presents the same results for general motion matrices, as a result of which it shows a far greater range of values for each pair of motion matrices.

The advantage of the total least squares method over the ordinary least squares method is that it is unaffected by Euclidean transformations of the data. This is due to its reliance on the geometric, rather than the algebraic distance to the manifold. We see from Table 5.1 and the nature of the approximation in equation (5.22) that the geometric distance is just a multiple of the algebraic distance, the multiplication factor being determined by the parameters of the camera used to generate the data. The motion matrices that minimise the sum of the squares of the residuals will thus be the same for ordinary and total least squares methods when data is generated according to a realistic camera model. The variation in the norm of the gradient seen in the columns of Table 5.2 explains why the total least squares approach proves superior for data generated according to general motion matrices.

Trial				
1	2	3	4	5
0.0136538	0.00480119	0.00144263	0.00322330	0.00895554
0.0135596	0.00480689	0.00144121	0.00323210	0.00978067
0.0135738	0.00477333	0.00144201	0.00319464	0.00846607
0.0135843	0.00479188	0.00144203	0.00321045	0.00998705
0.0135609	0.00480352	0.00144347	0.00323202	0.00854463
0.0135990	0.00480188	0.00144283	0.00320743	0.00939436
0.0135919	0.00480881	0.00144215	0.00324183	0.00981198
0.0135859	0.00479356	0.00144327	0.00319444	0.00938259
0.0136230	0.00479612	0.00144339	0.00321914	0.00993624
0.0135424	0.00482369	0.00144215	0.00320256	0.00833310
0.0135618	0.00482368	0.00144218	0.00323732	0.00981202
0.0136047	0.00481028	0.00144161	0.00319905	0.00900025
0.0135737	0.00478577	0.00144213	0.00320626	0.00943115
0.0135451	0.00480763	0.00144173	0.00322777	0.00926702
0.0136475	0.00483650	0.00144164	0.00321713	0.00974576
0.0136100	0.00478920	0.00144249	0.00318398	0.00974753
0.0135651	0.00480153	0.00144280	0.00322075	0.00896593
0.0135811	0.00480822	0.00144330	0.00320687	0.00962225
0.0135448	0.00477644	0.00144363	0.00318938	0.00922709
0.0135179	0.00478305	0.00144295	0.00318569	0.00868575

Table 5.1: Comparing derivatives - camera-based motion matrices

5.12 Conclusion

We have shown a method for constructing an approximation to the geometric distance based on linearising the manifold and that the approximation is accurate. We have also shown that the approach is worthwhile when the curvature of the manifold varies significantly over the data space. Unfortunately, in the case of the problem at hand, the gains made in using geometric distance measures are not significant enough to justify the extra computational effort required.

One situation in which distance measures of this form may provide more significant advantages is where the variances or covariances of the individual data elements are known. In equation (5.13) we assumed that the variance of the data elements was identical. If the variances are not identical, and we have some information about the nature of the variances, then we can incorporate this into equation (5.13). Some work in this direction has been carried out in Ref. [33], and shows promising results.

Trial				
1	2	3	4	5
0.0104021	0.0039190	0.0131172	0.0126732	0.0064201
0.0736075	0.0459063	0.0207778	0.0214896	0.0648416
0.0744701	0.0754716	0.0950387	0.0669086	0.0276325
0.1275970	0.0181690	0.0287859	0.0196260	0.2252070
0.1478640	0.0677369	0.0658131	0.0394017	0.0057206
0.0345513	0.0359932	0.1279310	0.0754019	0.0046749
0.0231267	0.0216485	0.0194550	0.0382806	0.0098342
0.0117736	0.0074273	0.0274975	0.0276933	0.0295883
0.0090052	0.0006961	0.0354355	0.0550518	278.70300
0.0294153	0.0272826	0.0654270	0.0513257	0.0123012
0.1334450	0.0266337	0.0095053	0.0231625	0.0154115
0.0566409	0.0302338	0.0146664	0.0147253	0.0050022
0.0931793	0.0376214	0.0132076	0.0195027	0.0410249
0.0250815	0.0225418	2.9871200	2.9718000	0.0064256
0.0291315	0.0219863	0.1441900	0.0794377	0.0533637
0.0175879	0.0082110	0.0626285	0.0355358	0.0266798
0.0352154	0.0322963	0.0463474	0.0258569	0.0084419
0.3220600	0.2641270	0.0721973	0.0474931	0.0453012
0.0243351	0.0225654	0.0360002	0.0270858	0.0151616
0.0303701	0.0270414	0.0200172	0.0142241	0.0042912

Table 5.2: Comparing derivatives - general motion matrices

Chapter 6

Filtering optical flow fields

We now consider two methods for altering optical flow fields on the basis of the degree to which their elements satisfy the differential epipolar equation.

6.1 Projecting optical flow onto a manifold

Regardless of the means used to estimate the motion matrices, or the curvature of the corresponding manifold, we will eventually arrive at a $\{\boldsymbol{C}, \boldsymbol{W}\}$ pair representing the key parameters of a moving camera. As has been shown in Section 4.3, the estimation process can be seen as determining the consistent optical flow field closest to the original data. Consistency refers to the property of an optical flow field whereby there exist motion matrices such that the differential epipolar equation is satisfied for all optical flow vectors therein. Having estimated the motion matrices on the basis of this closest consistent field of optical flow vectors, we are naturally led to the idea of reconstructing on the same basis. This method assumes that the consistent optical flow vectors should be a better representation of the true data than is provided by our observed data. There are many possible projections onto the manifold. We seek the projection that requires the smallest change in the data.

6.1.1 The closest point on a manifold

We require a method of projecting an optical flow vector $\boldsymbol{x}$ onto a manifold defined by $\boldsymbol{C}$ and $\boldsymbol{W}$. The projection that maps $\boldsymbol{x}$ to its closest point on $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ is achieved by linearising the manifold in much the same way as in Section 5.1. That section, however, proceeded with the aim of removing $\tilde{\boldsymbol{x}}$ from the calculation in order to facilitate estimation of the motion matrices. Now that such an estimate has been calculated, we seek $\tilde{\boldsymbol{x}}$ for other purposes.

In Section 5.1 we derived equation (5.4)

$$||\boldsymbol{x} - \tilde{\boldsymbol{x}}|| = \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||},$$

for the (approximate) Euclidean distance between a point $\boldsymbol{x}$ and the closest point on the manifold $\tilde{\boldsymbol{x}}$. If we let $\gamma = \text{sign}\,(f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}))$ then

$$||\boldsymbol{x} - \tilde{\boldsymbol{x}}|| = \gamma \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||}. \tag{6.1}$$

We know that

$$\boldsymbol{x} - \tilde{\boldsymbol{x}} = ||\boldsymbol{x} - \tilde{\boldsymbol{x}}||\, \frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{||\boldsymbol{x} - \tilde{\boldsymbol{x}}||}, \tag{6.2}$$

and that

$$\frac{\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||} = \gamma \frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{||\boldsymbol{x} - \tilde{\boldsymbol{x}}||}, \tag{6.3}$$

and so, substituting (6.2) into (6.3), we get

$$\boldsymbol{x} - \tilde{\boldsymbol{x}} = ||\boldsymbol{x} - \tilde{\boldsymbol{x}}||\, \gamma \frac{\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||}.$$

Using this and (6.1), we see that

$$\begin{aligned} \boldsymbol{x} - \tilde{\boldsymbol{x}} &= \gamma \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})|}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||} \gamma \frac{\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||} \\ &= \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||^2} \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}), \end{aligned}$$

and therefore that

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})||^2} \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}}).$$

As in Section 5.1, we do not know $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})$ but if $\boldsymbol{x}$ is sufficiently close to $\tilde{\boldsymbol{x}}$ then we can assume that $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}) \approx \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\tilde{\boldsymbol{x}})$. We can therefore generate an estimate $\widehat{\boldsymbol{x}}$ of $\tilde{\boldsymbol{x}}$:

$$\widehat{\boldsymbol{x}} = \boldsymbol{x} - \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})||^2} \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}). \tag{6.4}$$

So for a particular $\boldsymbol{C}$ and $\boldsymbol{W}$ we now have a means of generating $\widehat{\boldsymbol{x}}$, an approximation to the closest point on the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ to our original data $\boldsymbol{x}$. The fact that we have used a linear approximation of the function $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ in equation (5.2) means that $\widehat{\boldsymbol{x}}$ is close to, but not necessarily on, the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$. We therefore define an iterative scheme whereby $\boldsymbol{x}_0 = \boldsymbol{x}$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)}{||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)||^2} \Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k), \tag{6.5}$$

and $\lim_{k \to \infty} \boldsymbol{x}_k = \widehat{\boldsymbol{x}}$.

The vector $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)$ has six elements. Only four of these elements correspond to directions in the image plane. If we use only these directions the iteration above becomes

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|||^2}\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k). \tag{6.6}$$

This corresponds to the projection applied in calculating the final estimate of the residual in Section 5.1.

6.1.2 The effect of the projection

In order to test the effects of projecting optical flow onto a manifold we randomly selected camera based motion matrices $\bar{\boldsymbol{C}}$ and $\bar{\boldsymbol{W}}$ as described in Appendix A.1.2. We then generated $n = 20$ optical flow vectors

$$\bar{\mathcal{S}} = \{\bar{\boldsymbol{x}}_i \, : \, f_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}(\bar{\boldsymbol{x}}_i) = 0, i = 1 \ldots n\}$$

to represent the true underlying data. Random noise of standard deviation 1 pixel was then added to the first two elements of the vectors $\bar{\boldsymbol{m}}_i$ and $\dot{\bar{\mathrm{m}}}_i$ (where $\bar{\boldsymbol{x}}_i = \left\{\bar{\boldsymbol{m}}_i, \dot{\bar{\mathrm{m}}}_i\right\}$) to create the set $\mathcal{S} = \{\boldsymbol{x}_i \, : \, i = 1 \ldots n\}$. The projection onto the manifold $\mathcal{F}_{\bar{\boldsymbol{C}},\bar{\boldsymbol{W}}}$ was then performed for each $\boldsymbol{x}_i$ to create the vector $\widehat{\boldsymbol{x}}_i$.

In order to determine the effect of the projection, we need to provide a measure of the difference between the true optical flow vectors and the estimates. The difference between an original "true" optical flow vector $\bar{\boldsymbol{x}}_i$ and the estimate $\widehat{\boldsymbol{x}}_i$ can be calculated by the square root of the sum of the squares of the differences between their elements. The average of these differences across the field is then simply

$$\frac{1}{n}\sum^{n} ||\widehat{\boldsymbol{x}} - \bar{\boldsymbol{x}}|| \, . \tag{6.7}$$

Table 6.1 shows the results of this process, each row representing the values of (6.7) for 1 of 10 tests.

No stopping condition is given in (6.6) so two candidates have been tested. The first stopping condition used was merely that only one iteration was performed. The justification for this is that the manifold has been shown to be relatively flat in Section 5.11.1, and so the linearisation in equation (5.2) should have little effect on its local shape. If this is true of the initial linearisation there would be little to be gained by repeating the process because the projected optical flow vector would already be very close to the manifold.

The second stopping condition is based on the magnitude of the effect of projection on each optical flow vector. The magnitude of the change to $\boldsymbol{x}_k$ is

$$\begin{aligned}\left|\left|\left|\frac{f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|||^2}\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)\right|\right|\right| &= \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|||^2}\,|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)||| \\ &= \frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|||},\end{aligned}$$

where the norm $|||\boldsymbol{z}|||$ of a vector $\boldsymbol{z}$ is as defined in Section 4.6.2. We stop the iterative process when

$$\frac{|f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|}{|||\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x}_k)|||} < \epsilon \tag{6.8}$$

for some ϵ. The fraction in (6.8) corresponds to the algebraic approximation to the geometric distance to the manifold for the point $\boldsymbol{x}_k$. By experimentation we have found that multiple iterations of this projection process provide improved results, but that reducing the value of ϵ below 10^{-3} is of little benefit irrespective of the data.

original	1 step	$\epsilon = 10^{-3}$
0.460992	0.40273	0.391154
0.456702	0.420822	0.410976
0.484189	0.437897	0.440952
0.454711	0.420504	0.41422
0.448227	0.371859	0.353968
0.473782	0.4245	0.422104
0.415976	0.406389	0.411816
0.45242	0.375349	0.374864
0.429816	0.396757	0.379033
0.439965	0.368901	0.345763

Table 6.1: Average error in projected optical flow over 10 tests

The results in Table 6.1 are based on the projection back onto the manifold corresponding to the original true motion matrices. Unfortunately, these matrices are not generally available, so we must estimate the motion matrices before we can do the projection.

Table 6.2 shows the results of projecting onto the manifold corresponding to motion matrices estimated from the data. The weighted least squares procedure was used to generate the estimated motion matrices. The reason for this is given in the next Section. Results are shown only for the second stopping condition. From Table 6.2 we see that projection onto a manifold corresponding to estimate motion matrices is beneficial in most cases. The improvement in accuracy is small, but worthwhile.

6.1.3 Recursive weighted least squares

Section 5.9.1 provides a method for estimating the motion matrices by linearising $f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$. Section 6.1.1 provides a method of updating optical flow on the basis of this estimate. An obvious step would be to combine the two into a recursive procedure for estimating the motion matrices and true optical flow in tandem.

In fact, a fully recursive procedure requires a more complex approach, but gains can be achieved by the following three step method:

original	$\epsilon = 10^{-3}$
0.405179	0.393833
0.426152	0.418075
0.422579	0.40462
0.37691	0.403343
0.43827	0.400847
0.4502	0.445082
0.435966	0.418036
0.417274	0.398374
0.436871	0.41717
0.457	0.447132

Table 6.2: Projecting using estimated motion matrices

1. Construct an estimate $\{\boldsymbol{C}, \boldsymbol{W}\}$ of the motion matrices using the weighted least squares procedure.
2. Project the measured optical flow onto the manifold defined by $\boldsymbol{C}$ and $\boldsymbol{W}$.
3. Repeat the weighted least squares estimation process on the basis of this new, projected, optical flow to generate a final $\{\boldsymbol{C}, \boldsymbol{W}\}$.

It could be argued that, since the projection in step 2 takes the optical flow to the manifold defined in step 1, the estimate provided by step 3 would be the same as that from step 1. This is not the case because the projection in step 2 uses a linearisation of the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$ rather than the manifold itself. The result of this linearisation is that the position of the optical flow vector after the projection is somewhere between its original position and the manifold. That is, the projection is not perfect. If the linearised version of the manifold differs significantly from the underlying manifold, the projection will be far from the surface of this underlying manifold. There are two cases in which the linearised version of the manifold will differ markedly from the original: firstly, if the underlying manifold has high curvature in the area of the linearisation point; and secondly, if the gradient of the differential epipolar equation at the data point is significantly different to the normal of the manifold at the closest point (see assumption leading to equation (6.4)). In either case the reapplication of the estimation procedure will provide a better estimate.

We have used weighted least squares estimation here rather than ordinary least squares because it uses the same technique involving the linearisation of the manifold in calculating estimates of motion matrices. An auxiliary advantage is that it is easily extended to the case of more than two applications as detailed below. Figure 6.1 depicts the results of tests carried out using the recursive weighted least squares procedure as measured by the inner product error measure from Section 5.6. The testing procedure is as described in Appendix A.3. Figure 6.1 shows that the recursive procedure produces better estimates than

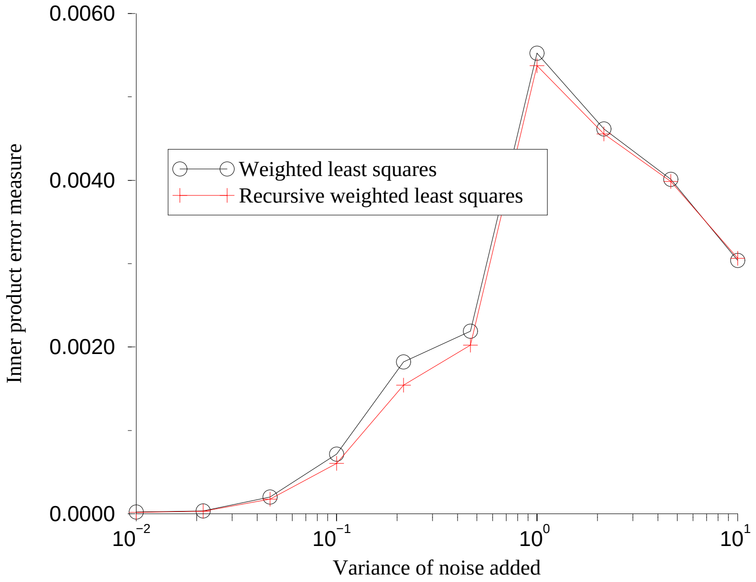

Figure 6.1: Recursive weighted least squares

the weighted least squares procedure would by itself, but that the improvement is marginal.

6.1.4 A fully recursive method

Having benefited from updating optical flow once, we now investigate the possibility of repeating the process. We have described above a three-step process involving estimation of the motion matrices $\boldsymbol{C}$ and $\boldsymbol{W}$, and projecting optical flow onto the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$. If we wish to extend this type of procedure to more steps, we have a problem updating the optical flow a second time.

When we project an optical flow vector onto the manifold $\mathcal{F}_{\boldsymbol{C},\boldsymbol{W}}$, we seek the point on that manifold closest to our original data, not the point closest to the last projection of the data. For the first data projection (step 2 above) the data point and the last projected point are the same. For further projections the data point remains the same but the projected point is updated every time.

In terms of the algebra, the linearisation in (5.2) is about the point $\widehat{\boldsymbol{x}}$, but the substitution of $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\boldsymbol{x})$ for $\Delta f_{\boldsymbol{C},\boldsymbol{W}}(\widehat{\boldsymbol{x}})$ in (6.4) is no longer the best possible. Figure 6.2 highlights this difference, showing that the linearisation point $\underline{\boldsymbol{x}}$ and the data point $\boldsymbol{x}$ are not necessarily the same points. At every iteration we improve the estimation of the manifold, but still require the point on that manifold closest to our original datum.

It is possible to provide a new optical flow projection routine based on minimising the distance to the data rather than to the last projection. This work is being carried out, but preliminary results show that the improvement in results does not justify the increase in complexity unless information about the covariance of the data is included. Kanatani [68] has shown that a similar

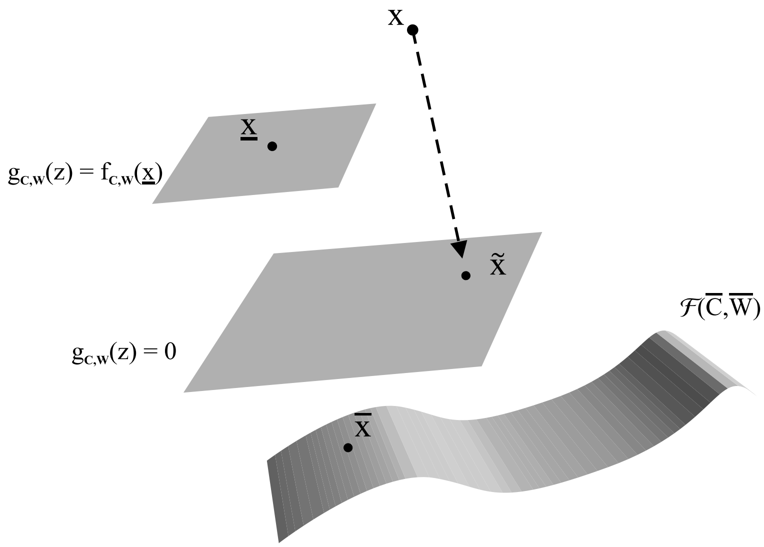
x
$\underline{x}$
$g_{c,w}(z) = f_{c,w}(\underline{x})$
$\tilde{x}$
$\mathcal{F}(\overline{C},\overline{W})$
$g_{c,w}(z) = 0$
$\overline{x}$

Figure 6.2: A fully recursive scheme

method applied to a similar problem produced an improved estimate in the case where the covariances of the data were available.

6.2 Least median of squares filtering

In this section we present a median filtering scheme for removal of outliers in the data and some results from real image sequences.

Typically, a real data set comprises two subsets: a large, dominant subset of valid data or *inliers*, and a relatively small subset of *outliers* or *contaminants*. Least squares minimisation is global in nature and hence vulnerable to distortion by outliers. To obtain robust estimates, outliers have to be detected and rejected. To identify the outliers, we use the method of least median of squares (LMedS) [108]. Once an LMedS fit is generated, the outliers can then be identified (if necessary) as those data which are inconsistent with the fit. The remaining inliers can then be processed with the use of a least squares technique, which results in a final, relatively robust, estimate [49].

The LMedS process requires repeated sampling of the data set, the calculation of a statistic from each sample, and a selection mechanism to determine the most appropriate estimate. The robustness of this method stems from the fact that at any one time only a subset of the data set is considered, and the fact that there is a high probability that one such set will contain only inliers.

In order to maximise the probability of selecting a set containing only inliers, we need to keep the set size as small as possible. The smallest set from which it is possible to calculate $\boldsymbol{C} : \boldsymbol{W}$ contains seven elements. Ideally, the estimator should consider the set of all seven-element samples. In practice, to make the search computationally feasible, the sample space is reduced to a family of m

randomly chosen samples. The number m is determined as follows. If we label the set of observed optical flow data $\mathcal{S}$, we assume that the proportion of outliers in $\mathcal{S}$ does not exceed ϵ, where $0 \leq \epsilon \leq 1$. Then the probability P that a family of m samples contains at least one element that is outlier-free is approximatively given by

$$P = 1 - (1 - (1 - \epsilon)^7)^m.$$

Consequently,

$$m = \left[\frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^7)} \right],$$

where $[x]$ denotes the integral part of x [108].

If we set $\epsilon = 0.2$ and $P = 0.95$ then, using our seven-point method from Section 4.1.2, we require 12 samples, whereas if we use our eight point method 16 samples would be required. These values must be set to represent the data available; overestimating is always safer, but implies more samples and therefore more processing time.

Once m is fixed by selecting ϵ and P, the LMedS estimate of $\boldsymbol{C} : \boldsymbol{W}$ is obtained in the following steps:

1. Select a family $\mathcal{S}_0$ consisting of m subsets of $\mathcal{S}$, each subset containing seven elements which are evenly spaced around the image.

2. For each $s \in \mathcal{S}_0$, compute three estimates $(\widehat{\boldsymbol{C}}_{s,k}, \widehat{\boldsymbol{W}}_{s,k})$ $(k \in \{1, 2, 3\})$ by using the seven-point algorithm (see Section 4.1.2).

3. For each $(s, k) \in \mathcal{S}_0 \times \{1, 2, 3\}$, determine the median

$$M_{s,k} = \mathrm{med}\{\delta(\boldsymbol{m}_i, \dot{\boldsymbol{m}}_i, \widehat{\boldsymbol{C}}_{s,k}, \widehat{\boldsymbol{W}}_{s,k})^2 \mid i = 1, \ldots, n\}.$$

4. Letting $(s_m, k_m) \in \mathcal{S}_0 \times \{1, 2, 3\}$ be such that

$$M_{s_m,k_m} = \min\{M_{s,k} \mid (s, k) \in \mathcal{S}_0 \times \{1, 2, 3\}\},$$

   take $(\widehat{\boldsymbol{C}}_{s_m,k_m}, \widehat{\boldsymbol{W}}_{s_m,k_m})$ for the LMedS estimate of $\boldsymbol{C} : \boldsymbol{W}$.

It is important in generating the subsets of $\mathcal{S}$ in step 1 to provide a means of ensuring that the vectors selected are evenly spaced around the image. If this is not the case the estimates calculated in step 2 are less likely to reflect the true motion matrices because of the instabilities introduced. With the LMedS estimate at hand, we proceed to identify outliers by applying the following procedure:

1. Take

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n - 7}\right) \sqrt{M_{s_m,k_m}}$$

   for the *robust standard deviation* of the distance measurements [108].

2. Declare $[\boldsymbol{m}_i^T, \dot{\boldsymbol{m}}_i^T]^T$ to be an outlier if and only if

$$\delta(\boldsymbol{m}, \dot{\boldsymbol{m}}, \widehat{\boldsymbol{C}}_{s_m,k_m}, \widehat{\boldsymbol{W}}_{s_m,k_m}) > 2.5\hat{\sigma}.$$

Once the outliers have been detected and removed, we can apply one of the least-squares techniques proposed earlier to the remaining elements of $\mathcal{S}$ and thereby obtain a robust estimate of $\boldsymbol{C} : \boldsymbol{W}$.

Chapter 7

Experimental results

We have, in previous sections, shown that it is possible to estimate the motion matrices from data contaminated with noise. We have also shown that, from a sufficiently accurate estimate of the motion matrices, it is possible to reconstruct the scene viewed. In this section we estimate the motion matrices from real image sequences, and calculate the corresponding reconstructions.

7.1 Experimental results on synthetic image sequences

Figure 3.1 in Section 3.1.1 shows a reconstruction of three surfaces of a cube. This reconstruction was generated by selecting a pair of motion matrices and specifying the three-dimensional shape of the points in the scene. An optical flow field was then calculated on this basis. The reconstruction process described in Section 3.1 was then carried out using the true optical flow and the true motion matrices. This verifies that the reconstruction formulae work for perfect inputs. In order to provide a more realistic test, we now give an example using exact, synthetically generated data, but estimated motion matrices.

7.1.1 Yosemite Valley image sequence

The Yosemite Valley image sequence has been generated synthetically, and generously distributed by Lynn Quam at SRI. Six images from the sequence are shown in Figure 1.1, several of which reappear in Figure 7.1.

As the sequence was synthetically generated the true optical flow is known, although only that corresponding to the ninth image is available. This flow field is depicted in Figure 7.2.

Within the files distributed by SRI, the optical flow vectors are encoded in eight bits per dimension, so $\dot{m}_1$ and $\dot{m}_2$ are represented as one byte each. The location information, $\boldsymbol{m}$, is know precisely because the flow is sampled on a regular grid. The points above the horizon represent the optical flow of the

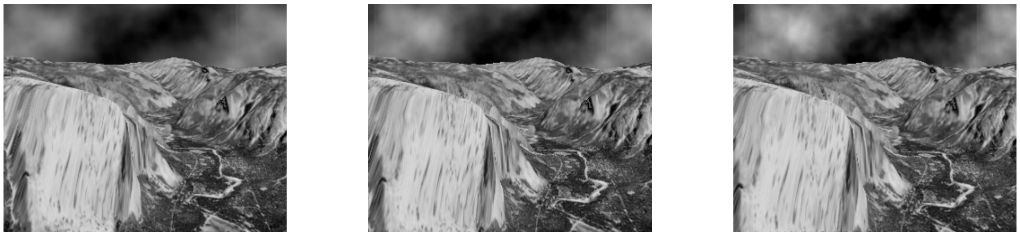

Figure 7.1: Images from the Yosemite Valley sequence

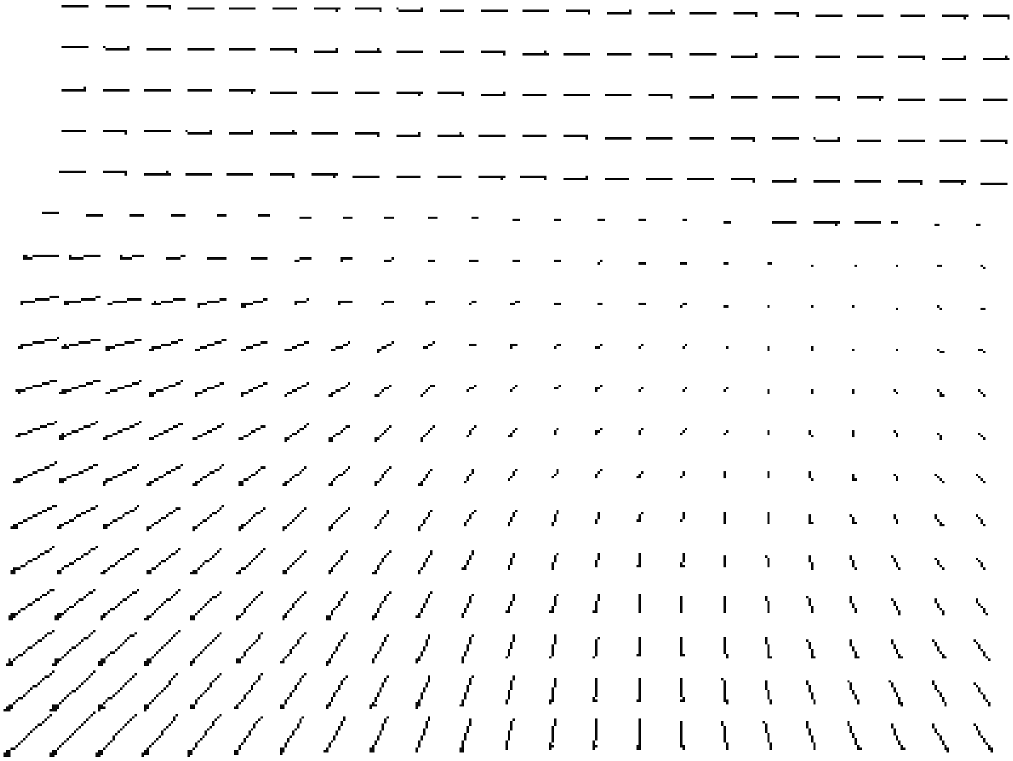

Figure 7.2: The Yosemite Valley sequence optical flow field

clouds visible in the image sequence. The clouds are not part of the rigid scene, so the corresponding vectors must be removed from the flow field. An optical flow vector is available for each pixel in the image. This represents more data than is necessary, or even practical. The field is thus randomly sampled in order to select a smaller number of flow vectors. This subset of the optical flow field is used to estimate the motion matrices for the ninth image in the sequence. The method used is the gradient weighted least squares procedure outlined in Section 5.2. Using the true optical flow and the estimated motion matrices, the reconstruction presented in Figure 7.4 was generated.

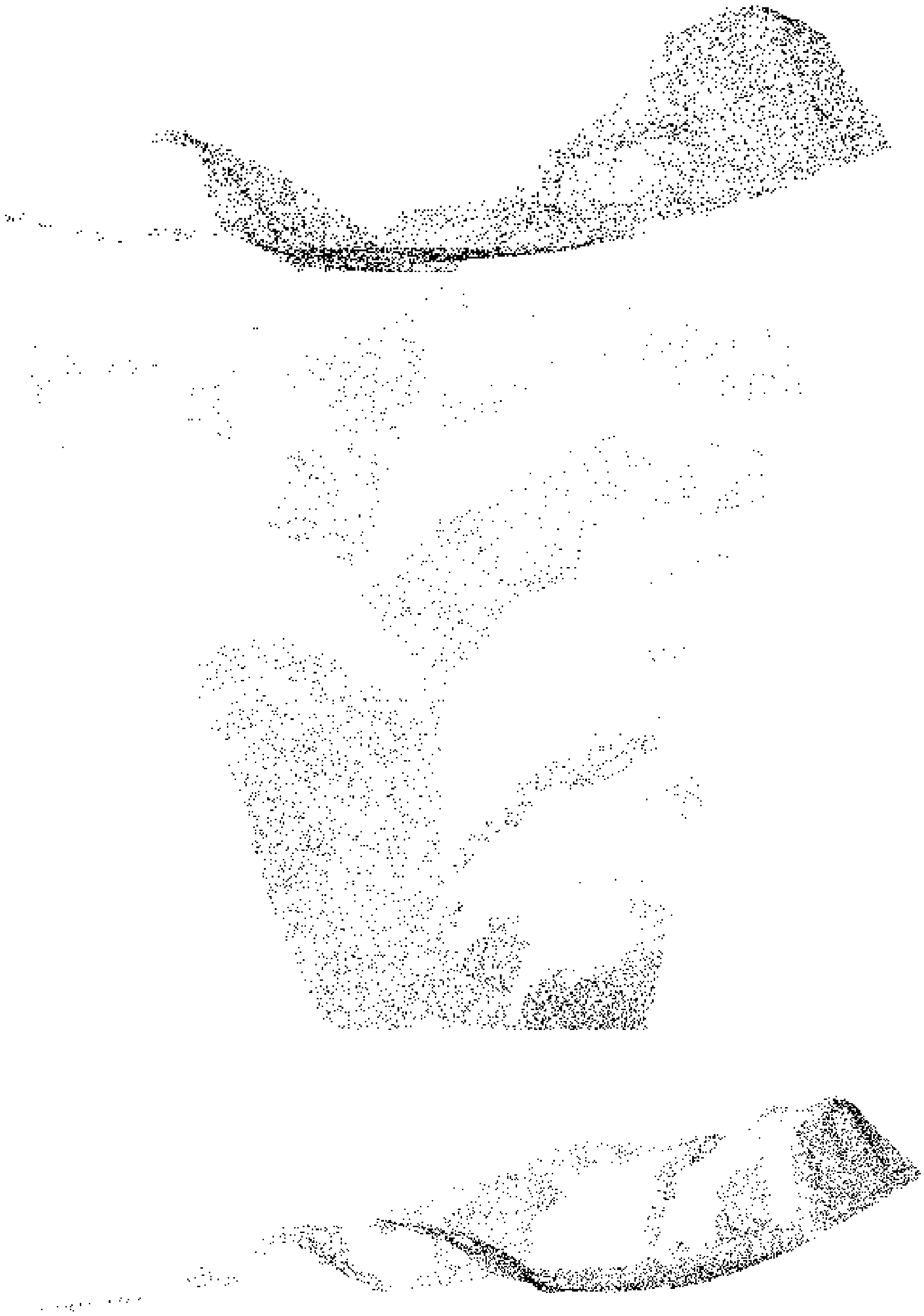

Figure 7.3: Yosemite Valley reconstructed points

The point clouds in Figure 7.4 give an impression of the shape of the valley, and of the distribution of image points across the surface. In order to make the shape of the valley more easily identifiable we have repeated the estimation and reconstruction process for a smaller number of optical flow vectors and triangulated a surface across the resulting points. The surface was calculated using the Delaunay triangulation package created by Ian Curington of Advanced Visual Systems. Figure 7.4 shows the results of this process with the ninth image of the sequence projected onto the reconstructed surface.

Unfortunately the true shape of the valley has not been made available, so we cannot compare the reconstruction with the original. The true motion of the camera has been provided by Lynn Quam, and matches the estimated motion to 2 significant figures. The most convincing argument in favour of the reconstructed shape of the Yosemite Valley, however, is that, when viewed and manipulated in 3-dimensions it looks as you would expect it to given the image sequence.

7.2 Experimental results on real images

We now present the results of estimation and reconstruction from a number of real image sequences.

7.2.1 Calibration object sequence

Figure 7.5 shows three images of a calibration object. The sequence was taken using a Kodak DCS420, which is a digital camera based on a Nikon single lens reflex camera. The calibration object sequence has the advantage of showing clear corners amenable to sub-pixel accuracy measurement. The method used to determine corner locations was based on computing the intersections of lines found in the image. The optical flow generated by this process is depicted in Figure 7.2.1.

The motion matrices were estimated using the recursive weighted least squares procedure given in section 6.1.3. The recursive weighted least squares procedure updates the optical flow field in the course of estimating the motion matrices. It is this updated optical flow field which was used to generate the reconstruction. No information about the shape or appearance of the calibration object, or of features on the grid, has been used in estimating the optical flow or the motion matrices. The process does not rely on the fact that the images are of a calibration object at all.

In order to aid visualisation, lines connecting the points at the corners of the squares on the grid have been added. Two views of the reconstruction of the calibration object are shown in Figure 7.2.1. The angle between the faces of the calibration object when the images were taken was 90 degrees, and the markings on the faces of the object are obviously coplanar. The overhead view of the reconstruction, in Figure 7.7, shows that the points on the faces of the

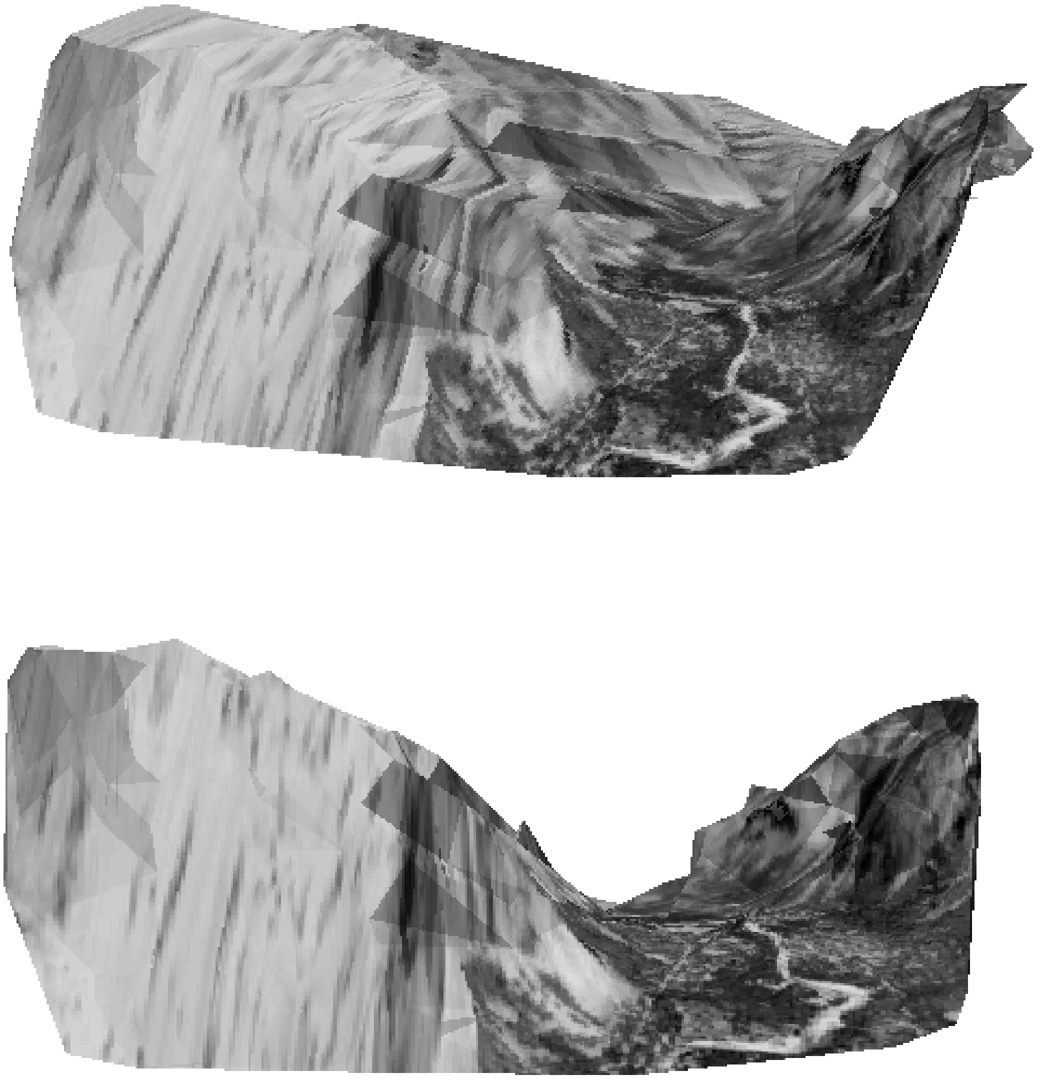

Figure 7.4: Yosemite Valley rendered reconstruction

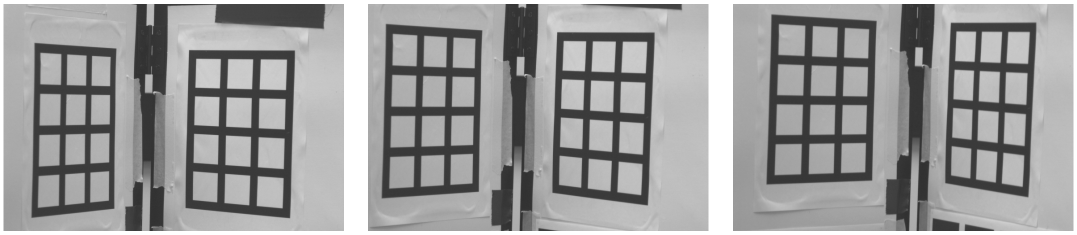

Figure 7.5: Images from the calibration object sequence

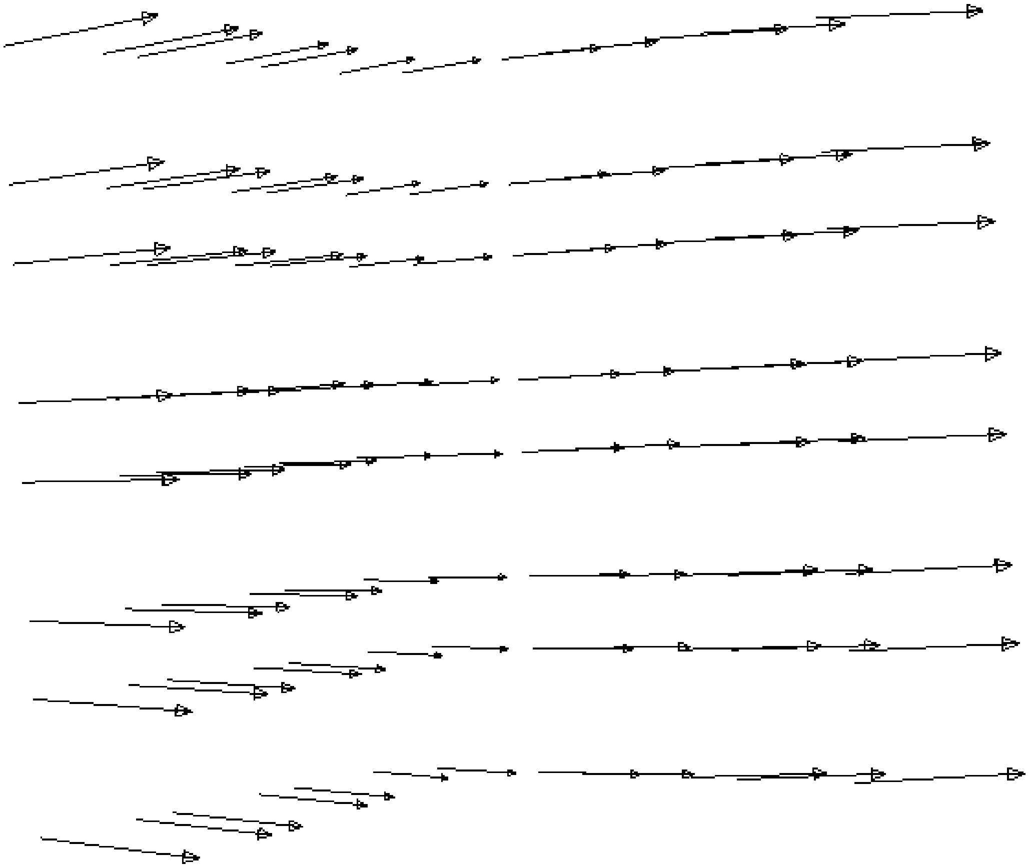

Figure 7.6: Optical flow from the calibration object sequence

calibration object are relatively coplanar and that the angle between the faces of the object is close to ninety degrees.

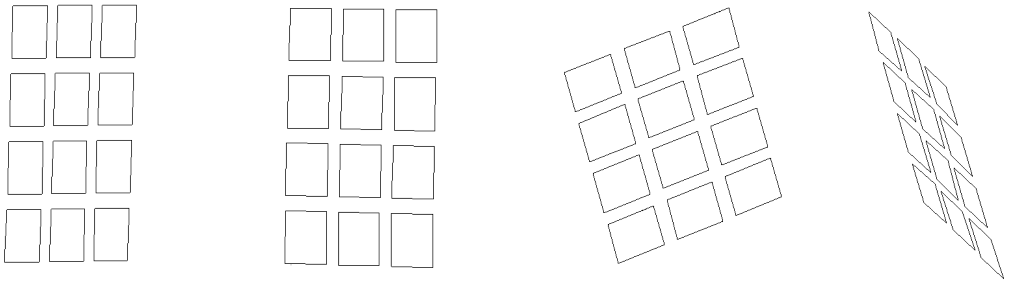

Figure 7.7: Calibration grid reconstructions

7.2.2 Office sequence

The office sequence, two images of which are shown in Figure 7.9, was taken using the Pulnix 9701 described in Section 5.11. This was one of the cameras which formed the basis of the camera-based motion matrices used in synthetic testing in previous chapters (as described in Appendix A.1.2). Feature location was carried out using the intersection of lines method described above. Estimation of the motion matrices was performed using the recursive weighted least squares method on the points selected as inliers using the least median of squares approach. Reconstruction was thus performed on this set of inliers, as updated by the recursive weighted least squares scheme.

Figures 7.10 and 7.11 show the reconstruction of the office scene. As in the calibration grid example, lines have been added joining points in the reconstructions of the office scene in order to enhance the representation. In Figure 7.9 we see a number of sheets of paper on the wall and on the column. Figure 7.11 shows that the corners of these rectangles are well reconstructed. The sheets on the column are roughly perpendicular, and the side of the column is roughly parallel to the wall.

7.2.3 Soccer ball sequence

The soccer ball image sequence shown in Figure 7.12 was taken using the Pulnix 9701 camera. The feature detector used calculates the intersections of interpolated lines as described above, and the calculated optical flow was again filtered using the least median of squares technique from Section 6.2. The motion matrices were calculated using the recursive weighted least squares technique and the corresponding reconstruction generated from the updated optical flow field.

Having reconstructed the point locations, a surface was interpolated using the Delaunay triangulation method described in Section 7.1.1 for the Yosemite Valley sequence. The reconstruction is shown in Figure 7.13.

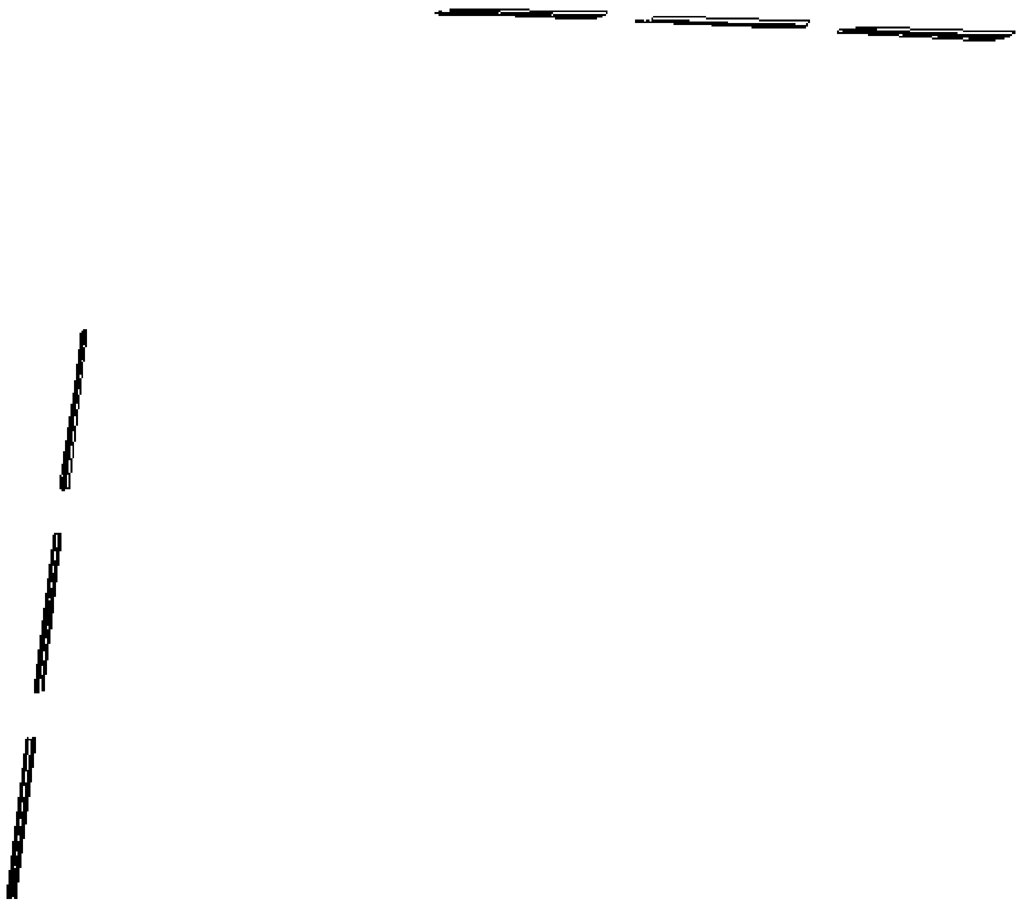

Figure 7.8: Calibration grid reconstruction, overhead view

7.3 Conclusion

We have shown a number of reconstructions above, all of which represent the shape of the viewed scene reasonably accurately. In section 1.4 we noted that reconstructing a scene from neighboring images of a video stream is difficult because the camera will not have moved far in the period between frames. This means that there is generally little spatial separation between the positions of the optical centre of the camera for one image and that for the next. This lack of separation increases the sensitivity of the triangulation process fundamental to reconstruction. On the other hand it leads to very similar images which renders the correspondence problem easier to solve. The sensitivity of the process of reconstruction from optical flow may indicate that it is not the most appropriate use to which estimated motion matrices may be put. The least median of squares procedure from Section 6.2 for example can be used to filter optical flow that is

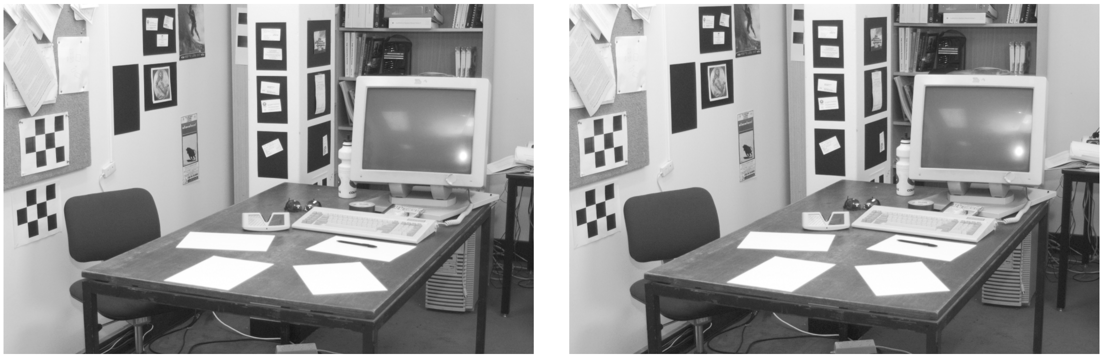

Figure 7.9: Office scene image sequence

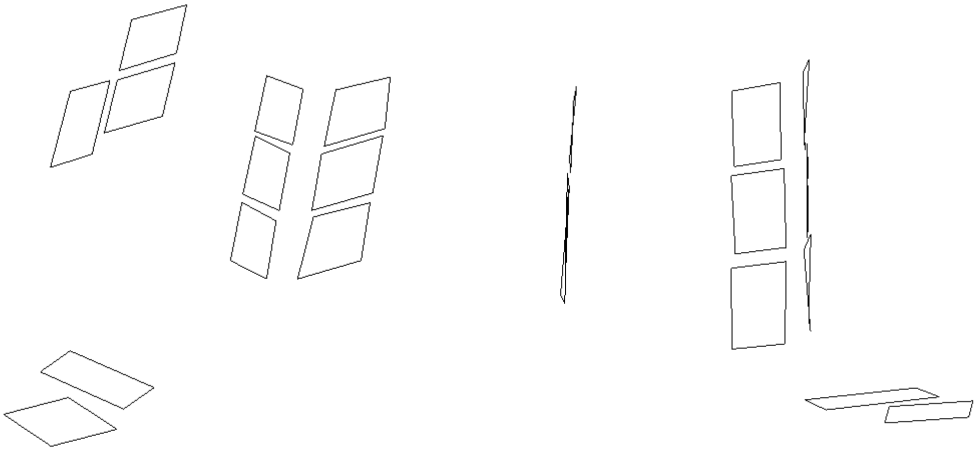

Figure 7.10: Office scene reconstructions

to be used for other purposes. The remaining inliers could be used for purposes such as collision avoidance or motion segmentation.

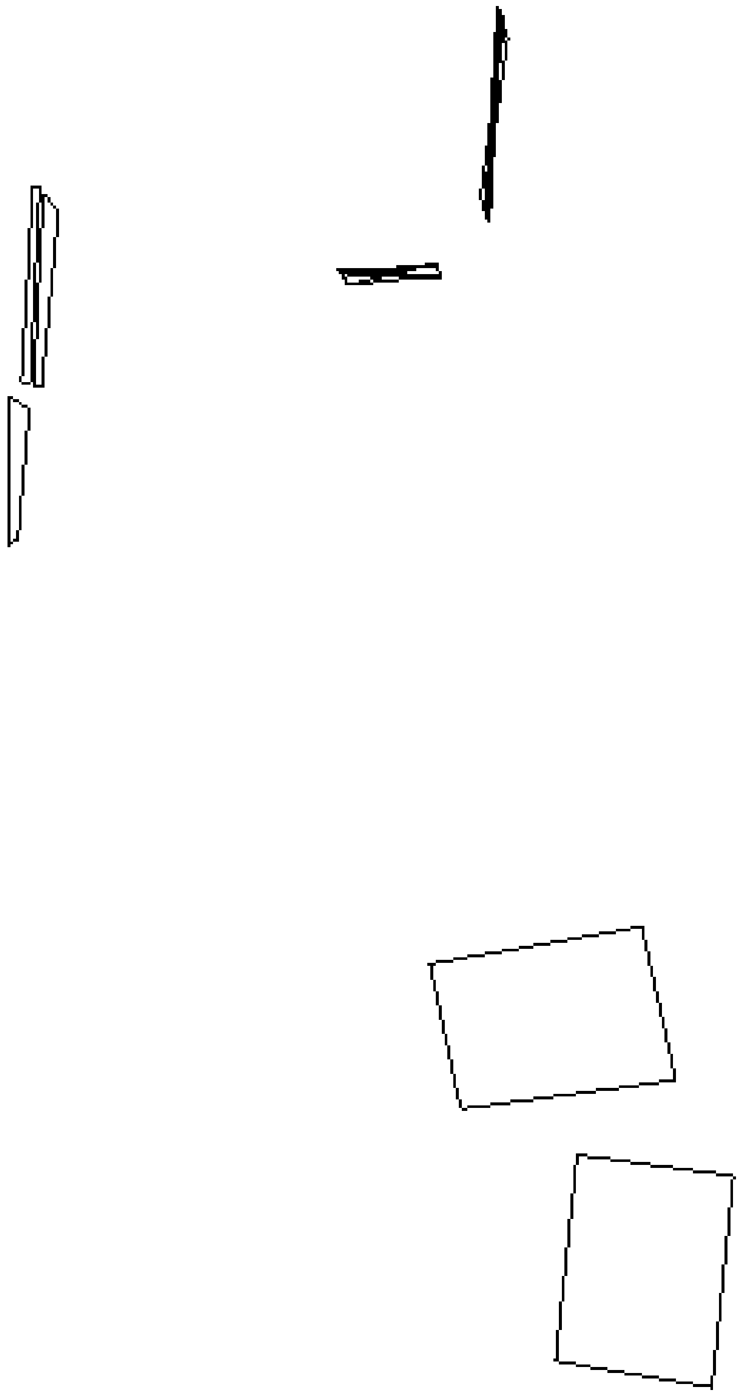

Figure 7.11: Office scene reconstruction, overhead view

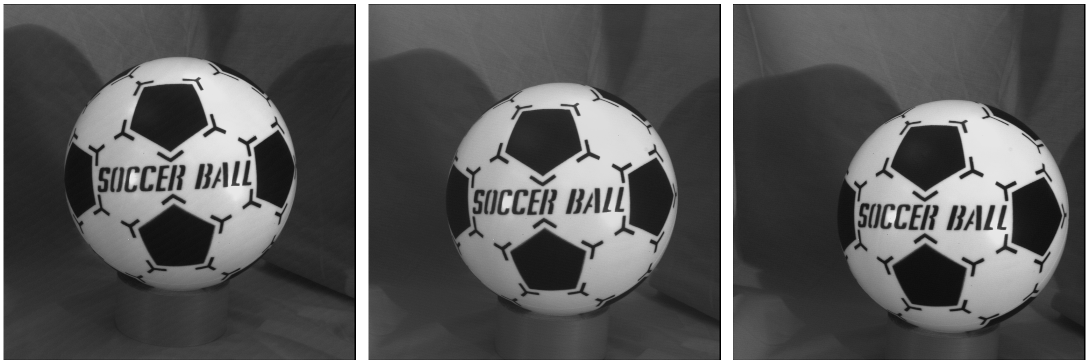

Figure 7.12: Images from the soccer ball sequence

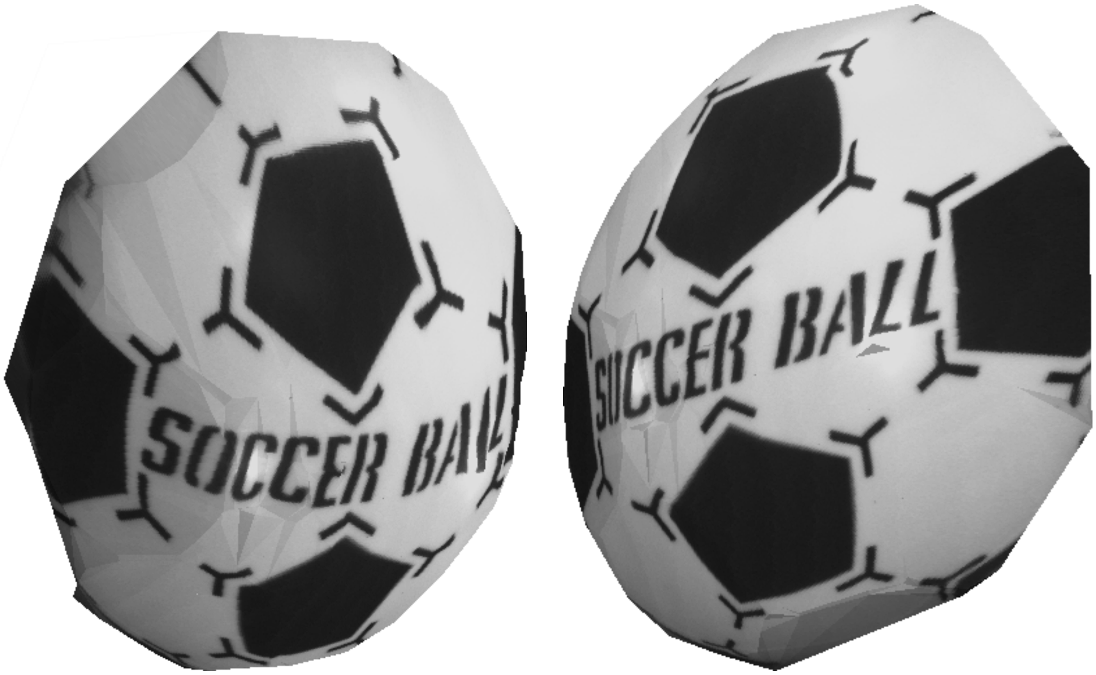

Figure 7.13: The reconstructed soccer ball

Chapter 8

Conclusion

We have carried out an investigation into the problem of determining structure from optical flow. This has involved a comparison of a number of different means of estimating the coefficients of the differential epipolar equation. The methods used are based on maximum likelihood estimation techniques and, more specifically, minimising the sum of the squares of certain residuals. Residuals, based on geometric distances, were derived, leading to the determination of certain cost functions. The form of these total least squares cost functions, however, rendered direct minimisation intractable. To alleviate this problem algebraic approximations to the cost functions were derived. A comparison of methods for minimising these algebraic cost functions was then carried out. This comparison lead to an analysis of the applicability of gradient weighted least squares approaches to the problem of estimating the coefficients of the differential epipolar equation. The conclusion of this analysis was that maximum likelihood based estimation procedures produced results that were marginally better than the ordinary least squares procedure presented. The similarity in the results was found to be due to the fact that the range of the gradients of the residuals over the data space was small.

Deriving the algebraic approximation to the total least squares cost functions necessitated estimating the closest point on a manifold to a measured optical flow vector. This estimation procedure led to a method of updating an optical flow vector such that it better matched the estimated motion matrices. This method, applied to each vector, allows the removal of some of the noise in the measured optical flow field. A rectification procedure for enforcing the cubic constraint on the motion matrices was presented and its performance measured. The effect of the procedure on the quality of estimates produced by various means was also measured.

Reconstruction formulae, based on the differential epipolar equation, were given, and the problem of estimating the trajectory of the camera over time investigated. The differential epipolar equation provides a means of estimating the motion of the camera, but the translation information is determinable only up to a scale factor. It is thus not possible to estimate the trajectory of the camera by simply integrating over its velocity. A means of registering the scale of the velocity of the camera was developed based on a reference to a single scene point. This method does not allow the scale of the translation to be recovered but does provide a means of ensuring a consistent scale amongst a set of estimated translation vectors. The consistent scaling of the estimated translation vectors enabled the development of two methods for calculating the trajectory of the camera, applicable to different classes of motions.

8.1 Future directions

An interesting extension to this work would be to compare the processes described in this thesis with the bootstrap method described in Refs. [28, 32, 142]. This bootstrap method is particularly interesting as confidence intervals for the estimates are produced as part of the process. Similarly, testing of the Hough transform methods of Kiryati [65] should be investigated.

An extension of the methods to the case where covariance information describing the uncertainty of individual data elements has been partly carried out (see [20,33,34,69]). An extension of this work involving estimating the covariances associated with real data should be investigated. A comparison of covariance estimation methods would enable an assessment of the accuracy with which the covariances of real data may be measured. This information is required in order to determine the value of including covariances in real estimation problems.

In Section 2.4.1 we presented a method for enforcing the cubic constraint on the motion matrices. This method took the form of a post process, and thus made no reference to the cost function. The change made to the motion matrices in order that they might satisfy the constraint was thus somewhat arbitrary. A means of incorporating the constraint into the estimation method would be more appropriate. A preliminary investigation has been carried out into reparameterisation of the motion matrices in order to accommodate the cubic constraint. The results of this investigation are not reported in this thesis. This procedure requires further investigation, along with the possibility of reparameterising for the case in which the focal length of the camera is fixed. It is envisaged that this will aid the estimation process as, in reality, the focal length of a camera taking a video sequence generally changes slowly. There are of course some situations in which this is not the case.

In Section 6.1.3 we suggested the possibility of a fully recursive scheme based on repeatedly estimating the motion matrices and updating the optical flow field. Some of this work has been carried out, and shows promise, but requires significant further investigation.

Appendix A

Modelling a moving camera

Evaluating the benefits of different estimation techniques requires the ability to generate synthetic data. In the case of the differential epipolar equation this data is optical flow. Data generation must take place according to a model, here the model being a camera moving through a static scene. Our model is described by the motion matrices, so, as a first step, we must generate the matrices $\boldsymbol{C}$ and $\boldsymbol{W}$.

A.1 Randomly generating motion matrices

Different methods of estimating the motion matrices have shown sensitivities to different sets of key parameters. Performing all tests with data generated according to a set of parameters may advantage one method over others. A means of randomly generating motion matrices is thus essential so that tests can be carried out over multiple sets of key parameters.

The disadvantage of generating new key parameters for every test is that comparisons of bias and variance of estimators become more difficult. Such tests can really only be carried out with fixed motion matrices and fixed data with random noise added.

A.1.1 General motion matrices

The matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ are symmetric and antisymmetric respectively, so they have nine independent elements. The matrices are, however, defined only up to a scale factor, and subject to the constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$. The most general method of generating motion matrices is to randomly generate values for each free element of the matrices, and normalise the result. In order to avoid bias in the set of motion matrices produced, we want the probability of each possible pair of motion matrices occurring to be the same. To this end, we generate nine instances of a uniformly distributed random variable, from which we construct the vector $\boldsymbol{\Theta}$. From the vector $\boldsymbol{\Theta}$ we can generate $\boldsymbol{C}$ and $\boldsymbol{W}$. The range of the random variable used to generate the elements of $\boldsymbol{\Theta}$ is somewhat arbitrary as it is only the ratio between elements that is significant (see Section 2.3). For simplicity we select here the range $-\alpha$ to α.

The vector $\boldsymbol{\Theta}$ is defined only up to a scale factor, so an appropriate normalising condition must be selected as described in Section 2.3. If we select the normalising condition that $||\boldsymbol{\Theta}||^2 = 1$ then particular vectors $\boldsymbol{\Theta}$ may be represented as points on a 9-dimensional sphere. This normalisation condition corresponds to the projection from $\mathbb{R}^9$ onto the surface of the sphere. Randomly generating points in a 9-dimensional cube and projecting onto the sphere as suggested above creates a biased distribution of points on the sphere. It is for this reason that points lying outside the sphere are discarded.

To constitute a valid set of motion matrices, $\boldsymbol{C}$ and $\boldsymbol{W}$ must satisfy the cubic constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$. Applying the method from Section 2.4.1 will distort the distribution of points on the 9-dimensional sphere. The hope is, however, that the distortion is not significant. One final constraint applied to the generated matrices is that the focal length at the time at which tests are carried out is real rather than imaginary. If this constraint is not met, then the candidate matrices are discarded and a new pair generated.

A.1.2 Camera-based motion matrices

An alternative to randomly selecting the elements of the matrices is to generate a set of key parameters from which $\boldsymbol{C}$ and $\boldsymbol{W}$ may be determined. In this vein we randomly select the internal and external parameters of a camera in motion according to two possible templates. These templates have been determined by analysis of two cameras: the Pulnix 9701 with telephoto lens, and the Pulnix 601C with 8mm fixed lens. Data generated under these models provide a more realistic test of the capabilities of the estimation methods because they correspond to optical flow by a realistic camera undergoing realistic motion. Generating motion matrices in this manner represents the reverse of the process described in section 2.5 for determining parameters from motion matrices. The nature of this process ensures that the motion matrices generated satisfy the constraint that $\boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 0$.

A.2 Generating noisy optical flow

Having determined the motion matrices it remains to calculate the corresponding optical flow. This is carried out by randomly generating a set of points in space. We then calculate the optical flow that such a set of points would generate on the image plane of a camera undergoing the motion described by the selected motion matrices using the equations from Chapter 2. These synthetic optical flow tuples are then perturbed in the elements corresponding to $m_1, m_2, \dot{m}_1$ and $\dot{m}_2$, by normally distributed random variables of mean 0 and variance as required.

A.3 Testing estimation methods

In testing the performance of each of the methods for estimating the motion matrices we have used the following procedure:

1. For each of 10 noise levels:

   1.1 For each of 50 trials:

      1.1.1 Randomly generate motion matrices using one of the methods above

      1.1.2 Randomly generate 50 optical flow vectors corresponding to the motion matrices

      1.1.3 Add noise to each optical flow vector according to the noise level selected

      1.1.4 Apply each of the methods to be tested to the noisy optical flow

      1.1.5 Calculate error in each estimate according to each metric.

   1.2 Calculate average error over the 50 trials for each of the methods using each of the error metrics

2. Report results.

Each test, therefore, has independent motion matrices, optical flow and noise.

Bibliography

[1] *Proceedings, CVPR '96, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[2] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, May 1989.

[3] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In Buxton and Cipolla [25], pages 3–16.

[4] K. Åström and A. Heyden. Multilinear forms in the infinitesimal-time case. In *Proceedings, CVPR '96, IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [1], pages 833–838.

[5] K. Åström and A. Heyden. Continuous time matching constraints for image streams. *International Journal of Computer Vision*, 28(1):85–96, 1998.

[6] A. Bab-Hadiashar and D. Suter. Robust optic flow estimation using least median of squares. In *Proceedings of the Third International Conference on Image Processing*, Lausanne, Switzerland, September 1996.

[7] A. Bab-Hadiashar and D. Suter. Optic flow calculation using robust statistics. In *Proceedings, CVPR '97*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, June 1997. IEEE Computer Society Press, Los Alamitos, CA.

[8] A. Bab-Hadiashar and D. Suter. Outlier resistant GAIC based image data segmentation. Technical Report MECSE-99-1, Monash University, Clayton 3168, Australia, 1999.

[9] Y. Bar-Shalom and X. Li. *Estimation and Tracking: Principles, Techniques and Software.* Artech House, Norwood, MA, 1993.

[10] J. L. Barron and R. Eagleson. Recursive estimation of time-varying motion and structure parameters. *Pattern Recognition*, 29(5):797–818, 1996.

[11] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[12] L. Baumela, M. J. Brooks, W. Chojnacki, and A. van den Hengel. Robust techniques for the estimation of structure for motion. In Burkhardt and Neumann [24], pages 281–295.

[13] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.

[14] S. S. Beauchemin and J. L. Barron. The computation of optical-flow. *Surveys*, 27(3):433–467, September 1995.

[15] F. L Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.

[16] M. J. Brooks, L. Baumela, and W. Chojnacki. An analytical approach to determining the egomotion of a camera having free intrinsic parameters. Technical Report 96-04, Department of Computer Science, University of Adelaide, South Australia, June 1996.

[17] M. J. Brooks, L. Baumela, and W. Chojnacki. Egomotion from optical flow with an uncalibrated camera. In J. Biemond and E. J. Delp, editors, *Visual Communications and Image Processing '97*, volume 3024 of *Proceedings of SPIE*, pages 220–228, San Jose, CA, USA, February 12–14, 1997.

[18] M. J. Brooks, W. Chojnacki, 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.

[19] M. J. Brooks, W. Chojnacki, L. Baumela, and A. van den Hengel. Robust techniques for the estimation of structure for motion. In Burkhardt and Neumann [24], pages 281–295.

[20] M. J. Brooks, W. Chojnacki, A. Dick, A. van den Hengel, K. Kanatani, and N. Ohta. Incorporating optical flow information into a self-calibration procedure for a moving camera. In S. F. El-Hakim and A. Gruen, editors, *Videometrics VI*, volume 3641 of *Proceedings of SPIE*, pages 183–192, San Jose, California, USA, January 28–29, 1999.

[21] M. J. Brooks, W. Chojnacki, A. van den Hengel, and L. Baumela. 3D reconstruction from optical flow generated by an uncalibrated camera undergoing unknown motion. In H. Pan, M. J. Brooks, D. McMichael, and G. Newsam, editors, *Image Analysis and Information Fusion, Proceedings of the International Workshop IAIF'97*, pages 35–42, Adelaide, Australia,

November 1997. Cooperative Research Centre for Sensor Signal and Information Processing, The Levels, South Australia.

[22] M. J. Brooks, W. Chojnacki, A. van den Hengel, and L. Baumela. Estimation of structure from motion in the uncalibrated case. In *Proceedings of the IPSJ Workshop on Computer Vision and Image Media*, volume PRMU97-180 (1997-12) of *Technical Report of IEICE*, pages 49–56, Utsunomiya, Japan, November 1997. The Institute of Electronics, Information and Communication Engineers.

[23] M. J. Brooks, L. de Agapito, D. Q. Huynh, and L. Baumela. Direct methods for self-calibration of a moving stereo head. In Buxton and Cipolla [25], pages 415–426.

[24] H. Burkhardt and B. Neumann, editors. *Computer Vision—ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, Fifth European Conference on Computer Vision, Freiburg, Germany, June 2–6, 1998. Springer, Berlin.

[25] B. Buxton and R. Cipolla, editors. *Computer Vision—ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, Fourth European Conference on Computer Vision, Cambridge, UK, April 14–18, 1996, 1996. Springer, Berlin.

[26] B. F. Buxton and H. Buxton. Monocular depth perception from optical flow by space time signal processing. *Philosophical Transactions of Royal Society of London*, B-218:27–47, 1983.

[27] B. F. Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2:59–75, 1984.

[28] J. Cabrera and P. Meer. Unbiased estimation of ellipses by bootstrapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:752–756, 1996.

[29] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[30] D. A. Castelow, D. W. Murray, G. L. Scott, and B. F. Buxton. Matching canny edgels to compute the principal components of optic flow. *Image and Vision Computing*, 6:129–136, 1988.

[31] K. Chaudhury and R. Mehrotra. Optical-flow estimation using smoothness of intensity trajectories. *Computer Vision, Graphics, and Image Processing*, 60(2):230–244, September 1994.

[32] K. Cho, P. Meer, and J. Cabrera. Performance assessment through bootstrap. Technical Report CE-123, Department of electrical and computer engineering, Rutgers University, New Jersey, USA, September 1995.

[33] W. Chojnacki, M. J. Brooks, and A. van den Hengel. Fitting surfaces to data with covariance information: fundamental methods applicable to computer vision. Technical Report TR99-03, Department of Computer Science, University of Adelaide, August 1999.

[34] W. Chojnacki, M. J. Brooks, and A. van den Hengel. Rationalising Kanatani's method of renormalisation in computer vision. In *Statistical Methods for Image Processing*, pages 61–63, Uppsala, Sweden, August 1999. International Statistical Institute.

[35] C. Debrunner. *Structure and Motion from Long Image Sequences.* PhD thesis, University of Illinois at Urbana-Champaign, 1990.

[36] C. Debrunner and N. Ahuja. A direct data approximation based motion estimation algorithm. In *Proceedings 10'th International Conference on Pattern Recognition*, pages 384–389, Atlantic City, N.J., June 1990.

[37] C. Debrunner and N. Ahuja. Segmentation and factorization-based motion and structure estimation for long image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):206–211, February 1998.

[38] M. Demazure. Sur deux problems de reconstruction. Technical Report 882, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, July 1988.

[39] D. Devleeschauwer. On the smoothness constraint in the intensity-based estimation of the parallax field. *Multidimensional Systems and Signal Processing*, 6(2):113–135, April 1995.

[40] J. O. Eklundh, editor. *Computer Vision—ECCV '94*, Third European Conference on Computer Vision, Stockholm, Sweden, May 2-6, 1994. Springer-Verlag,, Berlin.

[41] O. Faigeras and L. Robert. What can two images tell us about a third one? Technical Report 2018, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, September 1993.

[42] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In Sandini [110].

[43] O. Faugeras. A theory of the motion fields of curves. *International Journal of Computer Vision*, 10(2):125–156, 1993.

[44] O. Faugeras and B. Mourrain. About the correspondences of points between N images. In ICCV '95 [64], pages 37–44.

[45] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In ICCV '95 [64], pages 951–956.

[46] O. Faugeras, T. Vieville, and Q. Luong. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17(1):7–41, January 1996.

[47] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Mass., 1993.

[48] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In Sandini [110], pages 321–334.

[49] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[50] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland. *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts, 1974.

[51] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, 1950.

[52] N. C. Gupta and L. N. Kanal. 3-D motion estimation from motion field. *Artificial Intelligence*, 78(1-2):45–86, 1995.

[53] R. Hartley. Lines and points in three views: A unified approach. In *Image Understanding Workshop*, volume II, pages 1009–1016, Monterey, CA, November 1994.

[54] R. Hartley. In defence of the 8-point algorithm. In ICCV '95 [64], pages 1064–1070.

[55] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In Sandini [110], pages 579–587.

[56] J. C. Hay. Optical motions and space perception: an extension of Gibson's analysis. *Psychology Review*, 73:550–565, 1966.

[57] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, January 1992.

[58] A. Heyden. Reconstruction from image sequences by means of relative depths. In ICCV '95 [64], pages 1058–1063.

[59] B. K. P. Horn and B. G. Schunck. Determining optical flow. 17:185–203, 1981.

[60] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, Ma, 1986.

[61] B.K.P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, January 1990.

[62] B.K.P. Horn. Relative orientation revisited. *Journal of the Optical Society of America A*, 8(10):1630–1638, October 1991.

[63] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of IEEE*, 82(2):252–268, February 1994.

[64] *Proceedings of the Fifth International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press, Los Alamitos, CA.

[65] H. Kalviainen, N. Kiryati, and S. Alaoutinen. Randomized or probabilistic Hough transform: Unified performance evaluation. In *Proceedings of the 11th Scandinavian Conference on Image Analysis*, pages 259–266, Kangerlussuaq, Greenland, June 1999.

[66] K. Kanatani. 3-D interpretation of optical flow by renormalization. *International Journal of Computer Vision*, 11(3):267–282, 1993.

[67] K. Kanatani. Statistical bias of conic fitting and renormalisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):320–326, March 1994.

[68] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier Science, Amsterdam, 1996.

[69] 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. *Journal of Electronic Imaging*, 9(2):194–202, April 2000.

[70] J. K. Kearney and W. B. Thompson. Gradient-based estimation of optical flow with global optimization. In *Proceedings of Artificial Intelligence and Applications*, pages 376–380, 1984.

[71] J. K. Kearney, W. B. Thompson, and D. L. Boley. Gradient based estimation of disparity. In *Proceedings of Pattern Recognition and Image Processing*, pages 246–251, 1982.

[72] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):229–244, March 1987.

[73] M. Kendall and A. Stuart. *The Theory of Advanced Statistics*, volume 2. Charles Griffin and Co, Bucks, London, 5th edition, 1977.

[74] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, 1991.

[75] E. Kruppa. Zur ermittlung eines objektes aus zwie perspekriven mit innerer orientierung. *Sitz.-Ber. Akad. Wiss., Wien, Math. Naturw, Abt. Ila*, 122:1939–1948, 1913.

[76] A. N. Lasenby, J. Lasenby, W. J. Fitzgerald, and C. J. L. Doran. New geometric methods for computer vision: An application to structure and motion extimation. *International Journal of Computer Vision*, 26(3):191–213, 1998.

[77] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[78] H. C. Longuet-Higgins. Configurations that defeat the 8-point algorithm. In S. Ullman and W. A. Richards, editors, *Image Understanding, 1984*, Ablex, pages 314–319, 1984.

[79] H. C. Longuet-Higgins. The reconstruction of a scene from two projections: configurations that defeat the 8-point algorithm. In *IEEE: Proceedings of The First Conference on Artificial Intelligence Applications*, pages 395–397, Denver, Colorado, December 1984.

[80] H. C. Longuet-Higgins. Multiple interpretations of a pair of images of a surface. *Proceeedings of The Royal Society of London*, A(418):1–15, 1998.

[81] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of The Royal Society of London*, B(208):385–397, 1980.

[82] Q. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, March/April 1997.

[83] Q. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. In Eklundh [40], pages 589–599.

[84] Q.-T. Luong, R. Deriche, O. D. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: analysis of different methods and experimental results. Technical Report 1894, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, April 1993.

[85] Q.-T. Luong and O. D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996.

[86] Q.-T. Luong and O. D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.

[87] J. Magarey, A. Dick, M. J. Brooks, G. Newsam, and A. van den Hengel. Incorporating the epipolar constraint into a multiresolution algorithm for stereo image matching. In *Applied Informatics '99, 17th IASTED International Conference*, Innsbruck, Austria, February 15–18, 1999. Acta Press.

[88] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, 1982.

[89] S. Maybank. Properties of essential matrices. *International Journal of Imaging Systems and Technology*, 2:380–384, 1990.

[90] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[91] S. J. Maybank. The angular velocity associated with the optical flow field arising from motion through a rigid environment. *Proceedings of The Royal Society of London*, A(401):317–326, 1985.

[92] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[93] P. F. McLauchlan and D. W Murray. A unifying framework for structure and motion recovery from image sequences. In ICCV '95 [64], pages 314–320.

[94] A. Mitiche, R. Grisell, and J. K. Aggarwal. On smoothness of a vector field – application to optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):943–949, November 1988.

[95] M. Mühlich and R. Mester. The role of total least squares in motion analysis. In Burkhardt and Neumann [24], pages 305–321.

[96] H.-H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[97] H. H. Nagel. On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):13–30, January 1989.

[98] H. H. Nagel. Extending the "Oriented Smoothness Constraint" into the temporal domain and the estimation of derivatives of optical flow. In *Computer Vision—ECCV '90*, pages 139–148, First European Conference on Computer Vision, Antibes, France, April 1990. Springer, Berlin.

[99] H. H. Nagel and W. Enkelmann. Towards the estimation of displacement vector fields by "Oriented Smoothness" constraints. In *Proceedings of the 7th International Conference on Pattern Recognition*, pages 6–8, Montreal, July 30 - Aug 2, 1984. IEEE Computer Society Press.

[100] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, September 1986.

[101] N. Ohta and K. Kanatani. Optimal structure-from-motion algorithm for optical flow. *IEICE Transactions on Information and Systems*, E78-D(12):1559–1566, December 1995.

[102] H.-P. Pan, M. J. Brooks, and G. N. Newsam. Image resituation: initial theory. In S. F. El-Hakim, editor, *Videometrics IV*, volume 2598 of *Proceedings of SPIE*, pages 162–173, Philadelphia, Pennsylvania, USA, October 25–26, 1995.

[103] L. F. Pau, C. H. Chen, and P. S. P. Wang, editors. *Handbook of Pattern Recognition and Computer Vision*, chapter 2.4, pages 369–393. World Scientific Publishing Company, 1996.

[104] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572, 1901.

[105] M. Pollefeys, M. Van Gool, and M. Proesmans. Euclidean 3D reconstruction from image sequences with variable focal length. In Buxton and Cipolla [25], pages 31–42.

[106] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[107] O. Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui puevant les produire. *Journal de Mathématiques Pures et Appliquées*, 5:380–440, 1840.

[108] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Regression.* John Wiley & Sons, New York, 1987.

[109] P. D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Brookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.

[110] G. Sandini, editor. *Computer Vision—ECCV '92*, volume 588 of *Lecture Notes in Computer Science*, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19–22, 1992. Springer, Berlin.

[111] A. Shashua. Trilinearity in visual recognition by alignment. In Eklundh [40], pages 479–484.

[112] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:779–789, 1995.

[113] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In ICCV '95 [64], pages 920–925.

[114] M. A. Snyder. The mathematical foundations of smoothness constraints: A new class of coupled constraint. In *Proceedings of Image Understanding Workshop*, pages 154–161, Pittsburgh, PA, September 1990.

[115] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. In Eklundh [40], pages 211–216.

[116] S. Soatto and P. Perona. Reducing structure from motion: A general framework for dynamic vision with experimental evaluation. In *Proceedings, CVPR '96, IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [1], pages 825–832.

[117] S. Soatto and P. Perona. Recursive 3D visual motion estimation using subspace constraints. *International Journal of Computer Vision*, 22(3):235–259, 1997.

[118] S. Soatto and P. Perona. Reducing "Structure from Motion": a general framework for dynamic vision, part 2: Implementation and experimental assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9), September 1998.

[119] M. Sonka, M. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*, chapter 14, pages 512–524. Chapman and Hall, London, 1993.

[120] A. Stuart and J. Keith Ord. *Kendall's Advanced Theory of Statistics*, volume 1, chapter 8. Charles Griffin and Co, London, 5th edition, 1987.

[121] D. Suter. Motion estimation and vector splines. In *Proceedings, CVPR '94*, pages 939–942, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 1994. IEEE Computer Society Press, Los Alamitos, CA.

[122] R. Szelizki and S. B. Kang. Recovering 3D shape and motion from image sequences using non-linear least squares. Technical Report CLR 93/3, Digital Cambridge research laboratory, March 1993.

[123] R. Tissainayagam and D. Suter. Performance of visual tracking algorithms. In *Fifth International/National Conference on Digital Image Computing, Techniques, and Applications*, pages 206–211, Perth, Australia, 1999.

[124] R. Tissainayagam and D. Suter. Performance prediction and analysis for linear visual trackers. In *Irish Machine Vision and Image Processing Conference IMVIP'99*, pages 131–147, 1999.

[125] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. *IEEE Workshop on Visual Motion*, 91:21–28, 1991.

[126] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[127] P. H. S. Torr. *Outlier Detection and Motion Segmentation.* PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.

[128] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, September/October 1997.

[129] P. H. S. Torr and D. W. Murray. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[130] G. Toscani, R. Deriche R. Vaillant, and O. Faugeras. Stereo camera callibation using the environment. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, pages 953–960, Oulu, Finland, 1989.

[131] B. Triggs. Matching constraints in the joint image. In ICCV '95 [64], pages 338–343.

[132] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):13–27, January 1984.

[133] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, Massachusetts, 1979.

[134] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, May 1989.

[135] T. Viéville and O. D. Faugeras. Motion analysis with a camera with unknown, and possibly varying intrinsic parameters. In ICCV '95 [64], pages 750–756.

[136] T. Viéville, O. D. Faugeras, and Q.-T. Luong. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17(1):7–41, 1996.

[137] T. Viéville, C. Zeller, and L. Robert. Using collineations to compute motion and structure in an uncalibrated image sequence. *International Journal of Computer Vision*, 20(3):213–242, 1996.

[138] J. Weng, N. Ahuja, and T. S. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, September 1993.

[139] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, 1989.

[140] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):451–476, May 1989.

[141] M Werman and A. Shashua. Elimination: An approach to the study of 3-D-from-2-D. In ICCV '95 [64].

[142] Y. Leedan Y. Genc, J. Ponce and P. Meer. Parameterized image varieties and estimation with bilinear constraints. In *Proceedings of 1999 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–8, Fort Collins, Co, USA, June 1999.

[143] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, July 1996.

[144] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15:59–76, 1997.

[145] Z. Zhang, Q.-T. Luong, and O. D. Faugeras. Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction. Technical Report 2079, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, June 1994.

[146] A. Zisserman, P. Beardsley, and D. Murray. Navigation using affine structure from motion. In Eklundh [40], pages 85–96.