International Journal of Computer Vision 48(2), 91–114, 2002

Self-Calibration from Image Derivatives

TOMÁŠ BRODSKÝ
Philips Research, 345 Scarborough Road, Briarcliff Manor, NY 10510, USA
tbr@philabs.research.philips.com

CORNELIA FERMÜLLER
Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD 20742-3275, USA
fer@cfar.umd.edu

Received February 26, 1998; Revised December 10, 2001; Accepted December 10, 2001

Abstract. This study investigates the problem of estimating camera calibration parameters from image motion fields induced by a rigidly moving camera with unknown parameters, where the image formation is modeled with a linear pinhole-camera model. The equations obtained show the flow to be separated into a component due to the translation and the calibration parameters and a component due to the rotation and the calibration parameters. A set of parameters encoding the latter component is linearly related to the flow, and from these parameters the calibration can be determined.

However, as for discrete motion, in general it is not possible to decouple image measurements obtained from only two frames into translational and rotational components. Geometrically, the ambiguity takes the form of a part of the rotational component being parallel to the translational component, and thus the scene can be reconstructed only up to a projective transformation. In general, for full calibration at least four successive image frames are necessary, with the 3D rotation changing between the measurements.

The geometric analysis gives rise to a direct self-calibration method that avoids computation of optical flow or point correspondences and uses only normal flow measurements. New constraints on the smoothness of the surfaces in view are formulated to relate structure and motion directly to image derivatives, and on the basis of these constraints the transformation of the viewing geometry between consecutive images is estimated. The calibration parameters are then estimated from the rotational components of several flow fields. As the proposed technique neither requires a special set up nor needs exact correspondence it is potentially useful for the calibration of active vision systems which have to acquire knowledge about their intrinsic parameters while they perform other tasks, or as a tool for analyzing image sequences in large video databases.

Keywords: camera self-calibration, motion estimation, normal flow, depth distortion, minimization of depth variability

1. Introduction

Camera self-calibration, the process of estimating the intrinsic camera parameters without requiring special calibration objects in the scene, has attracted a lot of attention in recent years. Solutions to this problem will contribute to software systems processing previously acquired video data, but, more important from a perceptual point of view, will advance the field of active vision. If active vision systems are constructed that are able to move about and continuously change their parameters, they must also be able to

estimate these parameters while they interact with their environments.

The problem of computing the intrinsic parameters of the camera initially appeared to be very difficult. In order to facilitate it and reduce its complexity, known objects in the scene have usually been employed (Lenz and Tsai, 1988; Tsai, 1986).

Recently, with the introduction of projective geometry as a tool in Computer Vision, researchers achieved a projective reconstruction of the scene (that is, a reconstruction up to an unknown projective transformation) without explicitly computing the intrinsic calibration parameters, which are encoded in the projective transformation. This made it clear that self-calibration is just another aspect of the general structure from motion problem. A series of efforts then started to address the self-calibration problem as a structure from motion problem, that is as the recovery of Euclidean structure, motion parameters and calibration parameters. Naturally, this work followed the traditional style of structure from motion, approaching it in two computational stages. In a first step, the correspondence of points in successive image frames is established; in a second step, this is used to recover the intrinsic parameters and the motion parameters, and subsequently the structure of the scene (Armstrong et al., 1996; Dron, 1993; Faugeras et al., 1992; Hartley, 1994a, 1994b; Maybank and Faugeras, 1992; Pollefeys et al., 1996; Viéville and Faugeras, 1996).

Assuming the camera motion to be discrete, this problem is quite difficult. In Faugeras et al. (1992) and Maybank and Faugeras (1992), the epipolar geometry between pairs of views is computed and projective geometry techniques are used to obtain a set of constraints leading to high-degree polynomial equations. The method developed in Hartley (1994a) computes the parameters of interest in steps using non-iterative and iterative estimation techniques. First a projective reconstruction is derived from which the Euclidean structure and the extrinsic and intrinsic camera parameters are computed by utilizing the constraint of positive depth. Trilinear constraints (Spetsakis and Aloimonos, 1990) are used to provide additional information in Armstrong et al. (1996), but the camera motion is limited to be planar. Several other methods that have appeared either assume known motions of the camera (Dron, 1993), limit the camera motion to rotation (Hartley, 1994b), or make other simplifying assumptions.

In the classical structure from motion literature we also encounter approaches modeling the motion as continuous and using as input optical flow (Barron et al., 1994; Black, 1994; Anandan, 1989; Nagel, 1995; Nagel and Haag, 1998). The only study concerned with reconstruction from flow fields due to uncalibrated camera motion is Viéville and Faugeras (1996), whose projective reconstruction is extended to the differential framework, utilizing the instantaneous form of the epipolar constraint. The paper studies a number of issues, including the question of what can be computed. It provides a comparison between projective reconstruction in the discrete and continuous cases and also describes an algorithm for image stabilization. In this study, however, the problem of estimating the calibration parameters is not considered; the authors consider a very general model which allows for changes of calibration parameters between the frames.

It is well known that the computation of correspondence as well as optical flow is an ill-posed problem. For certain image situations rather good approximations can be obtained at the cost of expensive computations using sophisticated optimization techniques, but in the general case correspondence cannot be obtained without errors. As a remedy to this problem, a number of studies have recently been conducted addressing the motion recovery problem in a direct way, that is by employing as input to the recovery process only normal flow measurements—the components of flow measurements along gradients (Fermüller, 1993; Fermüller and Aloimonos, 1995; Horn and Weldon, 1988; Negahdaripour and Horn, 1987; Bergen et al., 1992). In this spirit this paper addresses the general problem of self-calibration and presents a self-calibration procedure using normal flow as input.

We use a linear pin-hole camera model—that is, we consider the coordinates of the imaging center, the spacing of pixels along the axes of the image grid, and the skewing which denotes the angle between the grid axes—and we assume that the calibration parameters remain constant over several image frames. The proposed method does not use known calibration objects nor does it require the scene to have any particular features, and it can therefore be applied to any image sequence and to unrestricted camera motion. This gives it the potential to be used for fast automatic calibration while the system performs other tasks.

1.1. Organization of the Paper

This paper is devoted both to a theoretical study of image motion fields arising from uncalibrated rigid camera motion and to practical aspects of the problem, proposing and implementing specific self-calibration techniques on the basis of normal flow only.

Continuous motion fields in the uncalibrated case consist of two components. The first component depends on the translation and calibration parameters, but in a way that does not allow recovery of the calibration parameters. The second component depends on the rotation and calibration parameters; it is on the basis of this component that self-calibration can be achieved. In the remainder of the paper we refer to the two components as the translational and the rotational components of the motion field.

Section 2 defines the camera model used, derives the equations describing a motion field induced by a rigidly moving uncalibrated camera and discusses the ambiguities inherent in single flow fields.

Section 3 studies the problem of self-calibration for a rotating camera, and Section 4 examines the problem for a camera undergoing a general motion. Both sections first show how relevant information can be estimated from image measurements and then present self-calibration algorithms that combine information from several flow fields.

Experiments using both artificial and real image sequences are carried out in Section 5 and the paper is summarized in Section 6.

2. Preliminaries

In this section we develop the preliminary equations and we study basic properties of uncalibrated rigid motion fields.

2.1. Uncalibrated Rigid Motion Fields

We use a traditional camera model—the image is formed by perspective projection on a planar imaging surface that is perpendicular to the optical axis. We choose a Cartesian coordinate system $OXYZ$, where O is the projection center, the Z axis is identical to the optical axis, and the X axis is parallel to the horizontal axis in the image coordinate system.

It is convenient to represent image points as three-dimensional vectors $\mathbf{r} = [x, y, f]^{\mathrm{T}}$, where x and y are the image coordinates of the point and f is a positive constant. A suitable choice of f can dramatically improve the numerical stability of the problem (Hartley, 1997).

The mapping between a scene point $\mathbf{R}$ and the corresponding image point $\mathbf{r}$ can be concisely written as (Faugeras et al., 1992)

$$\mathbf{r} = \frac{\mathbf{KR}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \tag{1}$$

where

$$\mathbf{K} = \begin{pmatrix} f_x & s & \Delta_x \\ 0 & f_y & \Delta_y \\ 0 & 0 & f \end{pmatrix} \tag{2}$$

is a matrix describing the intrinsic parameters of the camera. Here Δ_x and Δ_y are the image coordinates of the optical center, f_x and f_y are the focal lengths along the image axes (representing combined effects of the camera focal length, aspect ratio, and skewing angle), s is the skewing parameter, and $\hat{\mathbf{z}}$ is the unit vector in the direction of the Z axis.

Let the camera move in a static environment with instantaneous translation $\mathbf{t}$ and instantaneous rotation $\boldsymbol{\omega}$ (measured in the coordinate system $OXYZ$). Then a scene point $\mathbf{R}$ moves with velocity (relative to the camera)

$$\dot{\mathbf{R}} = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R} = -\mathbf{t} - [\boldsymbol{\omega}]_{\times}\mathbf{R} \tag{3}$$

where $[\boldsymbol{\omega}]_{\times}$ is a skew-symmetric matrix corresponding to the cross product with vector $\boldsymbol{\omega} = [\alpha, \beta, \gamma]^{\mathrm{T}}$:

$$[\boldsymbol{\omega}]_{\times} = \begin{pmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{pmatrix} \tag{4}$$

As we assume $\mathbf{K}$ to be fixed, differentiation of (1) yields

$$\dot{\mathbf{r}} = \frac{\mathbf{K}\dot{\mathbf{R}}}{\mathbf{R} \cdot \hat{\mathbf{z}}} - \frac{(\dot{\mathbf{R}} \cdot \hat{\mathbf{z}})\mathbf{KR}}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} \tag{5}$$

Substituting (3) into (5) and writing $\mathbf{R}$ as $(\mathbf{R} \cdot \hat{\mathbf{z}})(\mathbf{K}^{-1}\mathbf{r})$, we obtain

$$\dot{\mathbf{r}} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}}((\mathbf{t} \cdot \hat{\mathbf{z}})\mathbf{r} - \mathbf{Kt}) + (([\boldsymbol{\omega}]_{\times}\mathbf{K}^{-1}\mathbf{r}) \cdot \hat{\mathbf{z}}) \times \mathbf{r} - \mathbf{K}[\boldsymbol{\omega}]_{\times}\mathbf{K}^{-1}\mathbf{r} \tag{6}$$

To further simplify the equation, we take advantage of the special form of matrix **K**. A simple calculation shows that for any vector **v**,

$$(\mathbf{K}\mathbf{v}) \cdot \hat{\mathbf{z}} = \mathbf{v}^{\mathrm{T}}\mathbf{K}^{\mathrm{T}}\hat{\mathbf{z}} = \mathbf{v}^{\mathrm{T}}(f\,\hat{\mathbf{z}}) = f(\mathbf{v} \cdot \hat{\mathbf{z}})$$

Then we have

$$\dot{\mathbf{r}} = -\frac{1}{f(\mathbf{R} \cdot \hat{\mathbf{z}})}(\hat{\mathbf{z}} \times (\mathbf{K}\mathbf{t} \times \mathbf{r})) + \frac{1}{f}(\hat{\mathbf{z}} \times (\mathbf{r} \times (\mathbf{K}[\boldsymbol{\omega}]_{\times}\mathbf{K}^{-1}\mathbf{r}))) \tag{7}$$

Note that even though (7) contains f, the flow $\dot{\mathbf{r}}$ is independent of f, as can be easily seen by expanding the expressions.

A calibrated camera (with focal length f) can be described as a special case of an uncalibrated camera, in which case $\mathbf{K} = f\ \mathbf{I}$, where **I** is the identity matrix. Equation (7) then becomes the usual (Horn and Weldon, 1988)

$$\dot{\mathbf{r}}_{\mathrm{c}} = -\frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})}(\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})) + \frac{1}{f}\,\hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})) \tag{8}$$

As can be seen by comparing Eqs. (7) and (8), the first component of $\dot{\mathbf{r}}$ is the same as the translational flow generated by a calibrated camera moving with translational velocity $\mathbf{K}\mathbf{t}/f$. We thus call $\mathbf{K}\mathbf{t}$ (or more precisely $\mathbf{K}\mathbf{t}/(\mathbf{t} \cdot \hat{\mathbf{z}})$, the projection of **t** onto the image plane) the apparent focus of expansion/contraction and denote it by $\tilde{\mathbf{t}}$. The second part of $\dot{\mathbf{r}}$ depends linearly on the rotational velocity and non-linearly on the intrinsic parameters of the camera. To hide the non-linear dependence of $\dot{\mathbf{r}}$ on **K** we introduce the matrix

$$\mathbf{A} = \mathbf{K}[\omega]_{\times}\mathbf{K}^{-1} \tag{9}$$

and we call the two parts of (7) the translational and rotational components of flow. As an uncalibrated rigid motion is completely described by its apparent translation $\tilde{\mathbf{t}}$ and the matrix **A** of (9), we will denote the motion by $(\tilde{\mathbf{t}}, \mathbf{A})$.

Substituting $\tilde{\mathbf{t}}$ and **A** into (7) we obtain the uncalibrated motion field as a simple generalization of the calibrated field in the form

$$\dot{\mathbf{r}} = -\frac{1}{f\,Z}\,(\hat{\mathbf{z}} \times (\tilde{\mathbf{t}} \times \mathbf{r})) + \frac{1}{f}\,\hat{\mathbf{z}} \times (\mathbf{r} \times (\mathbf{A}\mathbf{r})) = \frac{1}{Z}\,\mathbf{u}_{\mathrm{tr}}(\tilde{\mathbf{t}}) + \mathbf{u}_{\mathrm{rot}}(\mathbf{A}) \tag{10}$$

where Z denotes the scene depth $(\mathbf{R} \cdot \hat{\mathbf{z}})$.

In the next subsections we discuss what information can in theory be extracted from uncalibrated motion fields and present a useful parameterization of the quantities that can be estimated.

2.2. *What can be Estimated?*

Our primary interest in this paper is the recovery of the intrinsic camera parameters contained in matrix **K** in Eq. (7). As can be seen from the above equations, the translational component of flow, as it is determined by vector $\mathbf{K}\mathbf{t}$, does not allow for the extraction of the calibration parameters. (A change in matrix **K** can be compensated—since **K** is not singular—by a change in **t**, keeping the translational field unchanged.) We therefore perform self-calibration on the basis of the rotational component of flow, as is usually the case in the discrete case.

There are two cases to consider. If the translational flow is negligible, the image flow measurements as well as normal flow measurements are linearly related to the elements of matrix **A** with no additional unknowns present in the equations. This is described in detail in Section 3.

The other case of general camera motion has been studied extensively for discrete camera motions (Faugeras, 1992). This body of work has been extended to the continuous motion case in Viéville and Faugeras (1996), where it was shown what quantities can still be observed and image flow equations identical to (7) were obtained.

In the discrete case, the geometric relationship between two cameras is described by the fundamental matrix (Faugeras, 1992) with 7 degrees of freedom. The direction of translation requires two degrees of freedom and the remaining five parameters provide constraints on the camera rotation as well as the intrinsic calibration parameters.

The situation is identical in the continuous case (Viéville and Faugeras, 1996). What can be estimated from a general uncalibrated motion field is the direction of the apparent translation $\mathbf{K}\mathbf{t}$, which amounts to two parameters, plus five parameters from the rotational component of the flow, encoded in a symmetric 3×3 matrix satisfying one constraint. The remaining component of the rotational flow cannot be disentangled from the translation as it takes the same form as a translational flow field of a planar scene.

From only two frames, both in the discrete and in the continuous case, the structure of the scene can

be recovered up to a projective transformation. The projective ambiguity takes a special form when the intrinsic parameters are fixed, as shown in Hartley (1994a) for the discrete and in Viéville and Faugeras (1996) for the continuous case. If we denote as $\mathbf{R}_0$ in homogeneous coordinates a scene point recovered assuming a standard camera, the same point, $\mathbf{R}$, under any other reconstruction is related to $\mathbf{R}_0$ through

$$\mathbf{R} = \left(\begin{array}{c|c} \mathbf{K}^{-1} & \mathbf{0} \\ \hline -\mathbf{h}_\infty & 1 \end{array}\right)\mathbf{R}_0 \tag{11}$$

The unknown elements of the transformation are vector $\mathbf{h}_\infty$, determining the position of the plane at infinity and $\mathbf{K}$, the intrinsic calibration matrix.

2.3. *Decomposition of* $\mathbf{A}$

Matrix $\mathbf{A}$ in (9) cannot be arbitrary, since it is similar to a skew-symmetric matrix $[\boldsymbol{\omega}]_\times$. The two matrices have the same eigenvalues, namely, 0 and $\pm i\,\|\boldsymbol{\omega}\|$; thus for any $\mathbf{A}$ representing uncalibrated rigid motion, we have

$$\begin{aligned} \det \mathbf{A} &= 0 \\ \operatorname{trace} \mathbf{A} &= 0 \end{aligned} \tag{12}$$

As a 3×3 matrix satisfying two constraints, the matrix depends on seven independent parameters. The conditions (12) are necessary, but not sufficient, because even though almost any matrix satisfying (12) can be decomposed into $\mathbf{K}$ and $[\boldsymbol{\omega}]_\times$, the matrices obtained do not have to be real. In such a case matrix $\mathbf{A}$ does not represent a real camera.

Consider the projections of flow (10) onto the directions perpendicular to the translational component of flow, since such projections are independent of scene depth (Heeger and Jepson, 1992). The perpendicular directions are along vectors $\mathbf{v}_{\mathrm{cp}} = \hat{\mathbf{z}} \times (\hat{\mathbf{z}} \times (\tilde{\mathbf{t}} \times \mathbf{r}))$. We obtain

$$\dot{\mathbf{r}} \cdot \frac{\mathbf{v}_{\mathrm{cp}}}{\|\mathbf{v}_{\mathrm{cp}}\|} = \frac{1}{\|\mathbf{v}_{\mathrm{cp}}\|}\frac{1}{2}\mathbf{r}^{\mathrm{T}}\mathbf{S}(\mathbf{A}, \tilde{\mathbf{t}})\mathbf{r} \tag{13}$$

where

$$\mathbf{S}(\mathbf{A}, \tilde{\mathbf{t}}) = \mathbf{A}^{\mathrm{T}}[\tilde{\mathbf{t}}]_\times - [\tilde{\mathbf{t}}]_\times \mathbf{A} \tag{14}$$

It is known from Viéville and Faugeras (1996) that one flow field allows only for the recovery of the direction of $\tilde{\mathbf{t}}$ together with matrix $\mathbf{S}(\mathbf{A}, \tilde{\mathbf{t}})$. It is useful to explicitly decompose matrix $\mathbf{A}$ into the part that can be estimated from $\mathbf{S}(\mathbf{A}, \tilde{\mathbf{t}})$ and the part that cannot. In Appendix A, we analyze in detail what can be obtained and the results are summarized in the following observation.

Observation. For any given $\hat{\mathbf{s}}$ (representing a candidate direction of translation), matrix $\mathbf{A}$ can be split into two parts $\mathbf{A} = \mathbf{A}_{\mathrm{c}}(\hat{\mathbf{s}}) + \mathbf{A}_{\mathrm{t}}(\hat{\mathbf{s}})$ with the following properties:

- Matrix $\mathbf{A}_{\mathrm{c}}(\hat{\mathbf{s}})$ is the sub-matrix which can be estimated when the direction of apparent translation is $\hat{\mathbf{s}}$. It depends on five independent parameters and encodes the same information as $\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})$. We define a simple linear function f_{c} such that $\mathbf{A}_{\mathrm{c}}(\hat{\mathbf{s}}) = f_{\mathrm{c}}(\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}}))$.
- Matrix $\mathbf{A}_{\mathrm{t}}(\hat{\mathbf{s}})$ is the sub-matrix which cannot be estimated. It can be expressed in the form $\mathbf{A}_{\mathrm{t}}(\hat{\mathbf{s}}) = \hat{\mathbf{s}}\mathbf{w}^{\mathrm{T}} + w_0\,\mathbf{I}$ with $\mathbf{w}$ a vector and w_0 a number. The flow induced by matrix $\mathbf{A}_{\mathrm{t}}(\hat{\mathbf{s}})$ is $(\mathbf{w} \cdot \mathbf{r})\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{s}})$, i.e., it is exactly the same as a translational field with apparent FOE $\hat{\mathbf{s}}$ for a planar scene whose 3D points are defined by equation $(\mathbf{KR}) \cdot \mathbf{w} = 1$.

To simplify the notation, we usually write only $\mathbf{A}_{\mathrm{c}}$ and $\mathbf{A}_{\mathrm{t}}$ instead of $\mathbf{A}_{\mathrm{c}}(\hat{\mathbf{s}})$ and $\mathbf{A}_{\mathrm{t}}(\hat{\mathbf{s}})$.

To summarize the observation:

$$\mathbf{A} = \mathbf{A}_{\mathrm{c}} + \mathbf{A}_{\mathrm{t}} = \mathbf{A}_{\mathrm{c}} + f\,\tilde{\mathbf{t}}\mathbf{w}^{\mathrm{T}} + w_0\,\mathbf{I} \tag{15}$$

To make the reference in later sections easier we use the notation of Fermüller and Aloimonos (1995) and call the vector components perpendicular to translational flow components the copoint projections, and we refer to the matrix $\mathbf{A}_{\mathrm{c}}$ as the copoint matrix of $\mathbf{A}$, or, if no confusion can occur, just the copoint matrix, because it can be estimated from the copoint projections. While $\mathbf{A}_{\mathrm{c}}$ depends on five parameters and $\mathbf{A}_{\mathrm{t}}$ on four parameters, the two matrices together satisfy conditions (12), so there are indeed only seven independent parameters.

2.4. *Estimation of Depth*

Consider the projection of vector Eq. (10) onto normal flow direction $\mathbf{n}$.[1] Assuming candidate motion $(\hat{\mathbf{t}}, \hat{\mathbf{A}})$, inverse scene depth can be estimated as

$$\frac{1}{\hat{Z}} = \frac{\dot{\mathbf{r}} \cdot \mathbf{n} - \mathbf{u}_{\mathrm{rot}}(\hat{\mathbf{A}}) \cdot \mathbf{n}}{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}} \tag{16}$$

Substituting (15) into (16), the depth estimate simplifies into

$$\frac{1}{\hat{Z}} = \frac{\dot{\mathbf{r}} \cdot \mathbf{n} - \mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}}) \cdot \mathbf{n}}{\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}} - \mathbf{w} \cdot \mathbf{r} \tag{17}$$

Since $\hat{\mathbf{t}}$ and $\hat{\mathbf{A}}_{\text{c}}$ can be estimated from uncalibrated flow fields, the only unknown in the equation above is $\mathbf{w}$ in the linear term $\mathbf{w} \cdot \mathbf{r}$. Therefore the projective ambiguity (11) manifest itself as a linear function of the image coordinates added to the estimated inverse depth. While this property was also mentioned in Viéville and Faugeras (1996), it was not used. We utilize the property during the motion estimation stage.

3. Self-Calibration from Rotational Motion

In the previous section, we have shown that in the general case, only a certain part of matrix $\mathbf{A}$, the copoint matrix $\mathbf{A}_{\text{c}}$, can theoretically be estimated from a single flow field. If, however, the camera does not translate (or the translational flow is negligible) the whole matrix $\mathbf{A}$ can be computed and the problem of self-calibration becomes much easier.

As a consequence, in the rest of the paper we deal with two separate cases: the easier case of a non-translating camera in this section and, in Section 4, the more difficult case of a camera that both translates and rotates.

In both cases, we avoid the difficulties of optical flow estimation and use only normal flow (the projection of optical flow on the direction of the image brightness gradient). We describe how to compute either matrix $\mathbf{A}$ (for purely rotational motion) or the copoint matrix $\mathbf{A}_{\text{c}}$ (for general motion) from image measurements and show how to perform self-calibration.

3.1. Direct Estimation of $\mathbf{A}$

If the translational flow is very small (because the depth of the scene is large, or because the translational velocity is small), we can compute matrix $\mathbf{A}$ directly from normal flow using a least squares procedure.

Matrix $\mathbf{A}$ should satisfy the two conditions (12). The latter one is easily satisfied by setting $a_{33} = -a_{11} - a_{22}$ and estimating only the eight remaining elements of $\mathbf{A}$. The singularity constraint is discussed below.

Expanding the rotational part $\mathbf{u}_{\text{rot}}(\mathbf{A}) \cdot \mathbf{n}$ of the normal flow at point $\mathbf{r} = [x, y, f]^{\text{T}}$, where $\mathbf{n} = [\cos\psi, \sin\psi, 0]^{\text{T}}$, we obtain

$$\mathbf{p}_i^{\text{T}} \mathbf{a} = u_{\text{n}} \tag{18}$$

where $\mathbf{a} = [a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}]^{\text{T}}$ is the vector of the unknown elements of $\mathbf{A}$, u_{n} is the normal flow measurement, and $\mathbf{p}_i$ is the vector of coefficients

$$\begin{aligned}\mathbf{p}_i = [&-2x\cos\psi - y\sin\psi, -y\cos\psi, -f\cos\psi,\\ &-x\sin\psi, -x\cos\psi - 2y\sin\psi, -f\sin\psi,\\ &x(x\cos\psi + y\sin\psi)/f,\ y(x\cos\psi\\ &+ y\sin\psi)/f]^{\text{T}}\end{aligned} \tag{19}$$

Given N measurements, we combine the vectors $\mathbf{p}_i$ into an $N \times 8$ matrix $\mathbf{P}$, the values u_{n} into an $N \times 1$ vector $\mathbf{u}$, and solve the over-determined system $\mathbf{Pa} = \mathbf{u}$ using least squares.

It is more difficult to enforce the singularity of $\mathbf{A}$. The same problem is encountered in stereo calibration; the fundamental matrix used to describe the relative orientation of two cameras is also singular. Traditionally, the matrix is first computed using least squares and then singular value decomposition (SVD) is used to enforce the singularity of the matrix.

We use the same method to make $\mathbf{A}$ singular, i.e., compute the SVD of $\mathbf{A}$ and set the smallest singular value to zero. The resulting matrix is the singular matrix closest to the estimated $\mathbf{A}$, but its trace can be non-zero. We could use an iterative procedure to satisfy both conditions (12), but it should not be necessary. The SVD usually gives us a singular matrix with small trace. Since the matrix is then used to compute the calibration parameters (Section 3.2), it is preferable to use matrix $\mathbf{A}_1$ with small, but nonzero, trace instead of matrix $\mathbf{A}_2$, which exactly satisfies both conditions (12), but is not as compatible with the data as $\mathbf{A}_1$, due to the iterative changes.

3.2. Obtaining the Calibration Parameters

In this section we present methods that combine information obtained from several image frames and extract the constant intrinsic parameters of the camera. First we consider the minimum number of flow fields that are necessary to obtain complete calibration, i.e., all five calibration parameters. Of course, the theoretical minimum is sufficient only if the motion of the camera changes between frames, because identical motions provide identical, and thus redundant, equations.

For a non-translating camera, the matrix $\mathbf{A}$ with nine elements satisfies the two conditions (12) and also encodes three rotational parameters. Therefore at most four independent constraints for the calibration parameters can be obtained and at least two flow fields are needed for complete calibration.

For a camera undergoing a general motion, the seven independent parameters include two for the direction of translation and three parameters for the rotation, thus providing only two constraints for the calibration parameters. Consequently, at least three flow fields are necessary for complete calibration in the general case.

Note that for discrete camera motion where the estimation is based on point correspondences, three images (which would correspond to two flow fields) are sufficient to estimate complete calibration. The reason is that there are three fundamental matrices between pairs of views which, even though not completely independent due to the trilinear constraints, provide enough information to recover complete calibration. This information, however, cannot be reliably obtained in the continuous case.

3.2.1. Purely Rotational Motion: Linear Solution. The calibration parameters are related to the observed image flow through matrix $\mathbf{A} = \mathbf{K}[\boldsymbol{\omega}]_\times \mathbf{K}^{-1}$. Having several matrices $\mathbf{A}_i$ estimated from rotational flow fields, we may use a method analogous to that of Hartley (1994b) to compute $\mathbf{K}$ from $\mathbf{A}$.

Since $\mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ should be a skew-symmetric matrix $[\boldsymbol{\omega}]_\times$, the following[2] must hold for matrix $\mathbf{K}$:

$$\mathbf{K}^{-1}\mathbf{A}\mathbf{K} + \mathbf{K}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{K}^{-\mathrm{T}} = \mathbf{0} \tag{20}$$

Equation (20) multiplied by $\mathbf{K}$ and $\mathbf{K}^{\mathrm{T}}$ leads to a simplified condition

$$\mathbf{A}\mathbf{K}\mathbf{K}^{\mathrm{T}} + \mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} = \mathbf{0} \tag{21}$$

Denoting $\mathbf{C} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$, we obtain a set of linear equations for the elements of the symmetric matrix $\mathbf{C}$

$$\mathbf{A}\mathbf{C} + \mathbf{C}\mathbf{A}^{\mathrm{T}} = \mathbf{0} \tag{22}$$

It is well known that camera calibration is closely related to the absolute conic (Maybank and Faugeras, 1992; Faugeras et al., 1992) and its projection in the image plane. Equation $\mathbf{r}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{r} = 0$ defines the image of the absolute conic, so (22) in fact relates matrix $\mathbf{A}$ with the image of the absolute conic.

Equation (22) also provides a very simple way for combining results from multiple frames. Consider an image sequence with N frames and let $\mathbf{A}_i$ be the computed matrix for motion between frames i and $i+1$. Assuming that the calibration parameters are constant, the matrices $\mathbf{A}_i$ depend on the constant calibration matrix $\mathbf{K}$ and, in general, different matrices $[\boldsymbol{\omega}_i]_\times$. Then we can combine the Eq. (22) for all the matrices $\mathbf{A}_i$ and obtain $\mathbf{C}$ from a least squares procedure.

In matrix notation, we could minimize

$$\sum_i \left\| \mathbf{A}_i\mathbf{C} + \mathbf{C}\mathbf{A}_i^{\mathrm{T}} \right\|^2 \tag{23}$$

using the Frobenius matrix norm.

The solution of (23) is easy to obtain, but severely biased in the presence of noise. Matrix $\mathbf{C} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$ is most often diagonally dominant, because the focal length parameters f_x, f_y are much larger than the other parameters. Clearly, criterion (23) will be smaller when $\mathbf{C}$ is smaller and consequently the calibration parameters, especially f_x, f_y, are underestimated.

Obviously, we need to rescale the criterion. As one element of $\mathbf{C}$ is a known constant f^2, we can constrain $\mathbf{C}$ to have norm 1, find the minimum and then rescale the solution appropriately. The criterion we minimize is thus

$$\mathcal{E}_1 = \frac{\sum_i \left\| \mathbf{A}_i\mathbf{C} + \mathbf{C}\mathbf{A}_i^{\mathrm{T}} \right\|^2}{\|\mathbf{C}\|^2} \tag{24}$$

To demonstrate that the scaling in (24) is important, we show the estimated focal length f_x obtained in Experiment 1 (from Section 5) using the unscaled linear criterion in Fig. 1(a) and compare it with the results obtained from the scaled criterion in Fig. 1(b).

The minimum can be found by standard linear algebra algorithms. Let the elements of $\mathbf{C}$ be c_{ij}, where $c_{ij} = c_{ji}$. Then

$$\|\mathbf{C}\|^2 = c_{11}^2 + c_{22}^2 + c_{33}^2 + 2c_{12}^2 + 2c_{13}^2 + 2c_{23}^2$$

We arrange the elements of $\mathbf{C}$ into a vector

$$\mathbf{c} = [c_{11}, c_{22}, c_{33}, \sqrt{2}\,c_{12}, \sqrt{2}\,c_{13}, \sqrt{2}\,c_{23}]^{\mathrm{T}}$$

so that $\|\mathbf{c}\|^2 = \|\mathbf{C}\|^2$.

The elements of matrix $\mathbf{A}_i\mathbf{C} + \mathbf{C}\mathbf{A}_i^{\mathrm{T}}$ can be written as linear combinations (with coefficients defined by $\mathbf{A}_i$) of the elements of vector $\mathbf{c}$. Consequently, there exists a symmetric matrix $\mathbf{M}_i$ (the derivation is long, but

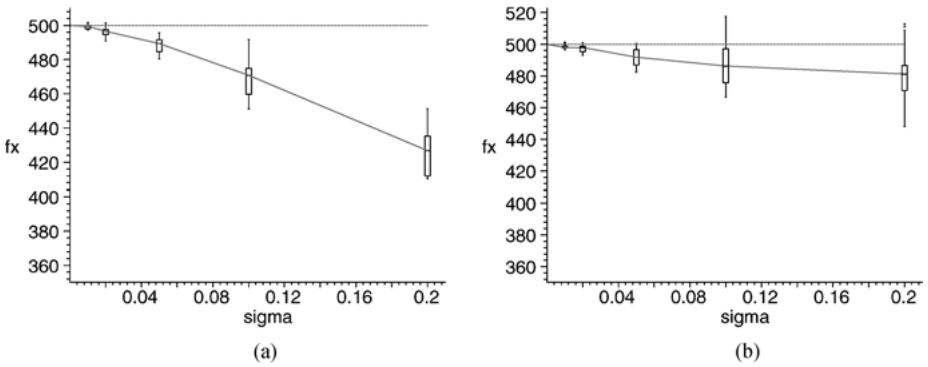

Figure 1. Comparison of (a) unscaled and (b) scaled linear self-calibration methods. Clearly, the scaled method performs better. For detailed description of the graphs see Section 5.

straightforward) such that

$$\left\|\mathbf{A}_i\mathbf{C} + \mathbf{C}\mathbf{A}_i^{\mathrm{T}}\right\|^2 = \mathbf{c}^{\mathrm{T}}\,\mathbf{M}_i\,\mathbf{c}$$

System (24) can thus be rewritten as

$$\mathcal{E}_1 = \frac{\mathbf{c}^{\mathrm{T}}\left(\sum_i \mathbf{M}_i\right)\mathbf{c}}{\|\mathbf{c}\|^2}$$

Vector $\mathbf{c}$ minimizing the criterion is the unit eigenvector corresponding to the smallest eigenvalue of matrix $\sum_i \mathbf{M}_i$. Therefore we call this method the eigenvector method in the sequel.

Each $\mathbf{A}_i$ provides six equations (due to symmetry), but only four of the equations are independent. A unique upper triangular $\mathbf{K}$ is easily obtained from $\mathbf{C}$ by Cholesky decomposition (Strang, 1988). However, since square roots are taken during the decomposition, the computed $\mathbf{K}$ may be complex. Such a solution does not represent a real camera and has to be discarded.

3.2.2. Purely Rotational Motion: Non-Linear Solution. The method in the previous section has several potential drawbacks. It is difficult to enforce additional constraints on $\mathbf{K}$, such as the skewing parameters s being zero, since even simple constraints transform into more complicated conditions for the matrix $\mathbf{C} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$. Also, we may not obtain a real solution when the recovered matrix $\mathbf{C}$ is not positive definite.

As an alternative, we can minimize the deviations from (20) directly. For matrix $\mathbf{A}_i$ and candidate solution $\hat{\mathbf{K}}$, denote the residual matrix by $\mathbf{E}_i$:

$$\mathbf{E}_i = \hat{\mathbf{K}}^{-1}\mathbf{A}_i\hat{\mathbf{K}} + \hat{\mathbf{K}}^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}\hat{\mathbf{K}}^{-\mathrm{T}}$$

The error function for a single matrix $\mathbf{A}_i$ is then

$$\begin{aligned}\text{trace}\,(\mathbf{E}_i\mathbf{E}_i) &= 2\,\text{trace}\,(\mathbf{A}_i\mathbf{A}_i)\\ &\quad + 2\,\text{trace}\left(\mathbf{A}_i\hat{\mathbf{K}}\hat{\mathbf{K}}^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}(\hat{\mathbf{K}}\hat{\mathbf{K}}^{\mathrm{T}})^{-1}\right)\end{aligned} \quad (25)$$

The error function we minimize is simply the sum of the partial errors (25)

$$\mathcal{E}_2(\hat{\mathbf{K}}) = \sum_i \text{trace}\,(\mathbf{E}_i\mathbf{E}_i) \quad (26)$$

Note that (26) does not depend on the scale of $\hat{\mathbf{K}}$. This is not a problem, as one element of $\mathbf{K}$ is a known constant f.

The price we have to pay is that the equations are no longer linear and iterative Levenberg-Marquardt minimization is therefore used. Closed form expressions for the partial derivatives of $\mathcal{E}_2(\hat{\mathbf{K}})$ with respect to the elements of $\hat{\mathbf{K}}$ can be computed by matrix differentiation. Denoting $\mathbf{X} = \hat{\mathbf{K}}^{-1}\mathbf{A}\hat{\mathbf{K}}$, we obtain

$$\frac{\partial\mathcal{E}_2}{\partial\hat{\mathbf{K}}} = 4\hat{\mathbf{K}}^{-\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X} - \mathbf{X}\mathbf{X}^{\mathrm{T}})$$

The linear method may be used to provide a starting solution for the iteration. Our experiments show that the error function is quite well behaved.

Figure 2 shows two density plots of $\mathcal{E}_2(\hat{\mathbf{K}})$ for the data obtained in Experiment 5 (see Section 5). We show two different 2-dimensional subspaces of the 5-dimensional solution space ($f_x - f_y$ space and $f_x - \Delta_x$ space), both passing through the true solution. In the first plot, there is a valley (in the f_x, f_y space) around the true solution. The valley corresponds to cameras with approximately correct aspect ratio and provides an illustration of the fact that has been confirmed by

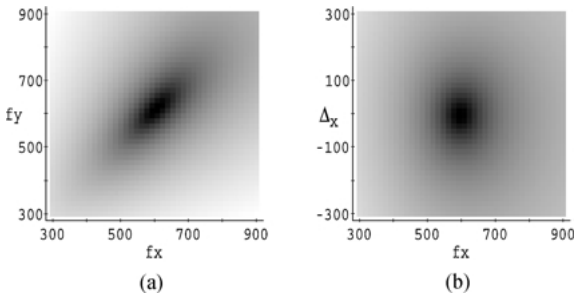

Figure 2. Subspaces of $\mathcal{E}_2(\hat{\mathbf{K}})$ obtained in Experiment 5. The grey-level denotes the value of $\mathcal{E}_2(\hat{\mathbf{K}})$ with black representing the smallest value. (a) The $f_x - f_y$ subspace. (b) The $f_x - \Delta_x$ subspace.

most of the experiments: among all the calibration parameters, the aspect ratio f_x/f_y can be estimated the most robustly.

We have mentioned above that it is simple to enforce additional constraints, such as $s = 0$. Consider a motion sequence with very small change in camera motion between successive frames. Then all the matrices $\mathbf{A}_i$ contain essentially the same data and describe a certain seven-dimensional subspace in the eight dimensional space of rotational and calibration parameters. While in theory the remaining degree of freedom can be obtained even if the motion changes slowly, in practice the information will be unreliable. In such cases one might prefer to impose the additional assumption $s = 0$, which is approximately true for most cameras, in order to estimate the remaining four intrinsic parameters more robustly.

4. Self-Calibration from General Camera Motion

The classic approach to the problem of structure from motion entails a clear separation between structure and motion estimation and between 2D and 3D information. Usually, first 2D-based smoothing constraints are employed to obtain from the image measurements (that is, the image derivatives) the optical flow field; then this information is used to estimate 3D motion and, subsequently, structure.

Here we take a different approach. We formulate constraints on the smoothness of the 3D surfaces and the rigid motion and relate these constraints directly to the image derivatives. This way the processes of smoothing, 3D motion and structure estimation are addressed synergistically and all the information inherent in image derivatives is used for 3D interpretation.

The underlying idea is based on the interaction between 3D motion and scene structure (Cheong et al., 1998). If we have an incorrect 3D motion estimate and we use it to estimate depth, we obtain a distorted version of the depth function. Not only do incorrect estimates of motion parameters lead to incorrect depth estimates, but the distortion is such that the worse the motion estimate, the more likely we are to obtain depth estimates that locally vary much more than the correct ones. The goal, thus, is to find the 3D motion which yields the least varying estimated depth. In practice this is implemented through a search for the 3D motion which minimizes a measure of depth variation within image patches.

The basic approach of the algorithm is quite simple. For a given candidate translation, we perform the following steps: estimate the rotation and then evaluate a measure of depth variation. A search in the space of translations for a minimum of the depth variability measure then yields the best 3D motion.

4.1. Distortions of Depth Estimates

The distortion of visual space due to incorrect motion estimates leads to a very important observation. The estimated and the true scene depth are related by a distortion factor D:

$$\hat{Z} = Z \cdot D$$

with

$$D = \frac{\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{(\mathbf{u}_{\text{tr}}(\mathbf{t}) - Z\mathbf{u}_{\text{rot}}(\delta\mathbf{A}_{\text{c}})) \cdot \mathbf{n}},$$

where $\delta\mathbf{A}_{\text{c}} = \mathbf{A}_{\text{c}} - \hat{\mathbf{A}}_{\text{c}}$. When the motion estimates are correct, the distortion factor simplifies into

$$D = \frac{\|\hat{\mathbf{t}}\|}{\|\mathbf{t}\|}$$

that is, a constant function expressing the overall scale ambiguity, since only the direction of translation can be recovered.

For incorrect motion estimates, the distortion factor for any direction $\mathbf{n}$ corresponds to the ratio of the projections of the two vectors $\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\text{tr}}(\mathbf{t}) - Z\mathbf{u}_{\text{rot}}(\delta\mathbf{A}_{\text{c}})$ on $\mathbf{n}$. The larger the angle between these two vectors is the more the distortion will be spread out over the different directions. Thus, for a smooth surface patch in space, assuming that normal flow measurements are

available along many directions, a rugged (i.e., unsmooth) surface will be computed on the basis of wrong 3D motion estimates.

This observation constitutes the main idea behind our algorithm. For a candidate 3D motion estimate we evaluate the variation of estimated depth within image patches. In contrast to traditional methods that utilize optical flow, all computations are based on normal flow and we thus have available the full statistics of the raw data, providing better weights.

4.2. The Criterion

Consider a small image region $\mathcal{R}$ that contains a set of measurements $\dot{\mathbf{r}}_i$ with directions $\mathbf{n}_i$. Given candidate motion parameters, we can estimate the inverse depth from (16) up to the overall scale ambiguity. To treat different patches equally, we normalize the estimated translation $\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}})$ to be a unit vector in the middle of the region.

One possible measure of depth variation is the variance of the depth values, or, rather, the sum of squared differences of the depth values from a mean $1/\bar{Z}$

$$\sum_i \left(\frac{\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}}) \cdot \mathbf{n}_i}{\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i} - \frac{1}{\bar{Z}} \right)^2 \tag{27}$$

Approaches that directly evaluate variations of estimated depth (or inverse depth) include (Horn and Weldon, 1988; Brodský et al., 1998). However, depth estimates may present a numerical problem, since for many measurements the depth estimate is unreliable (due to division by a small $\mathbf{u}_{\text{tr}} \cdot \mathbf{n}$). Thus we can either ignore many measurements where the depth estimate is unreliable, making comparisons between different translations difficult, or, alternatively, we have to deal with numerical instabilities. We choose a third possibility, defining a whole family of depth variation measures that includes the variance of estimated depth as well as many other measures.

In region $\mathcal{R}$ we compute

$$\Theta_0(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}}, \mathcal{R}) = \sum_i W_i \, (\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}}) \cdot \mathbf{n}_i - (1/\hat{Z})(\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i))^2 \tag{28}$$

where $1/\hat{Z}$ is the depth estimate locally minimizing the measure, i.e., not necessarily the mean $1/\bar{Z}$.

By setting $W_i = 1/(\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2$ we obtain the variation of inverse depth (27). Another natural choice is $W_i = 1$. Then Θ_0 becomes the sum of squared differences of the normal flow measurements and the corresponding projections of the best flow obtained from the motion parameters. This measure has been used in Mendelsohn et al. (1997).

With different choices of W_i we can either emphasize the contributions from the vectors perpendicular to the translational component which are independent of depth, or the vectors parallel to the translation, which are most strongly influenced by the depth. As long as we keep W_i bounded, criterion (28) nicely combines the contribution of the two perpendicular components.

We first minimize Θ_0 with respect to $1/\hat{Z}$. We model the scene patch by a general plane and use a linear approximation $1/\hat{Z} = \mathbf{z} \cdot \mathbf{r}$ (note that the third component of $\mathbf{r}$ is a constant f, so $\mathbf{z} \cdot \mathbf{r}$ is a general linear function in the image coordinates). Then we have

$$\frac{\partial \Theta_0}{\partial \mathbf{z}} = \sum_i W_i \, (\mathbf{z} \cdot \mathbf{r}_i)(\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2 \mathbf{r}_i - \sum_i W_i \times (\dot{\mathbf{r}}_i \cdot \mathbf{n} - \mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}}) \cdot \mathbf{n}_i)(\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i) \, \mathbf{r}_i = 0 \tag{29}$$

a set of three linear equations for the three elements of $\mathbf{z}$.

Substituting the solution of (29) into (28), we obtain $\Theta_1(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}}, \mathcal{R})$, a second order function of $\hat{\mathbf{A}}_{\text{c}}$. Notice that the computation can be performed symbolically even when $\hat{\mathbf{A}}_{\text{c}}$ is not known. This allows us to use the same equations to obtain both the copoint matrix and a measure of depth variation.

To estimate $\hat{\mathbf{A}}_{\text{c}}$, we sum up all the local functions and obtain a global function:

$$\Theta_2(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}}) = \sum_{\mathcal{R}} \Theta_1(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}}, \mathcal{R}) \tag{30}$$

Finally, global minimization yields the best matrix $\hat{\mathbf{A}}_{\text{c}}$ and also a measure of depth variation for the apparent translation $\hat{\mathbf{t}}$:

$$\Phi(\hat{\mathbf{t}}) = \min_{\hat{\mathbf{A}}_{\text{c}}} \Theta_2(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}}) \tag{31}$$

In our algorithm the computation of $\Phi(\hat{\mathbf{t}})$ involves two separate steps. First we estimate the best matrix $\hat{\mathbf{A}}_{\text{c}}$ and in the second step we evaluate the global depth variability measure for the motion $(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{\text{c}})$. In the two steps of computing $\Phi(\hat{\mathbf{t}})$ we choose different weights W_i in function Θ_0.

First we discuss estimation of $\hat{\mathbf{A}}_{\text{c}}$, using weights W_i'. Ideally, the most weight should be given to the

normal flow measurements along directions perpendicular to the translational flow. Such measurements are independent of the scene depth, thus the best source of information about $\hat{\mathbf{A}}_c$ and consequently should have more influence on Θ_0. Direct evaluation of the depth variance, however, means that in Eq. (27) the weighting factor for such vectors tends to infinity.

To prevent numerical instability, weights W_i' should certainly be bounded. For the rotation estimation part, we use

$$W_i' = \frac{1}{\cos^2 \psi_i + \lambda} \tag{32}$$

where ψ_i is the angle between $\mathbf{u}_{tr}(\hat{\mathbf{t}})$ and $\mathbf{n}_i$, λ is a small positive number. Substituting W_i' into (30), we obtain function Θ_2' (and functions Θ_0', Θ_1' can be defined analogously). The copoint matrix is computed as

$$\hat{\mathbf{A}}_{c0} = \underset{\hat{\mathbf{A}}_c}{\operatorname{argmin}}\, \Theta_2'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c)$$

Now we need to evaluate a global depth variation function to obtain $\Phi(\hat{\mathbf{t}})$. As the $\Phi(\hat{\mathbf{t}})$ values are compared for different directions of $\hat{\mathbf{t}}$, we choose constant weights

$$W_i = 1 \tag{33}$$

Then the contribution to $\Phi(\hat{\mathbf{t}})$ of a single normal flow measurement is

$$(\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{rot}(\hat{\mathbf{A}}_c) \cdot \mathbf{n}_i - (1/\hat{Z})(\mathbf{u}_{tr}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i))^2$$

and has a clear geometrical meaning; it is the squared difference of the normal flow and the corresponding projection of the best flow obtained from the motion parameters. More importantly, such squared errors can be easily compared for different directions of $\hat{\mathbf{t}}$.

The global depth variation function $\Phi(\hat{\mathbf{t}})$ is therefore

$$\Phi(\hat{\mathbf{t}}) = \Theta_2(\hat{\mathbf{t}}, \hat{\mathbf{A}}_{c0}) = \Theta_2(\hat{\mathbf{t}}, \underset{\hat{\mathbf{A}}_c}{\operatorname{argmin}}\ \Theta_2'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c)) \tag{34}$$

4.3. Algorithm Description

The translation is found by localizing the minimum of function $\Phi(\hat{\mathbf{t}})$ described in (34). To obtain $\Phi(\hat{\mathbf{t}})$:

1. Partition the image into small regions, in each region compute $\Theta_0'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c, \mathcal{R})$ using weights (32) and perform local minimization of $\hat{Z}$ (the computation is symbolic in the unknown elements of $\hat{\mathbf{A}}_c$). After substitution, the function becomes $\Theta_1'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c, \mathcal{R})$. At the same time, compute Θ_0 and Θ_1 using constant weights (33).
2. Add all the local functions $\Theta_1'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c, \mathcal{R})$ and minimize the resulting $\Theta_2'(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c)$ to obtain $\hat{\mathbf{A}}_{c0}$. Also add $\Theta_1(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c, \mathcal{R})$ to obtain $\Theta_2(\hat{\mathbf{t}}, \hat{\mathbf{A}}_c)$.
3. The final measure $\Phi(\hat{\mathbf{t}})$ is obtained by substituting $\hat{\mathbf{A}}_{c0}$ into Θ_2.

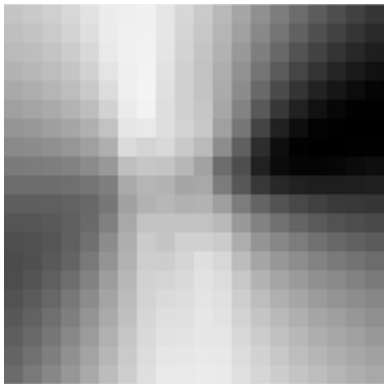

Figure 3. Function $\Phi(\hat{\mathbf{t}})$ for the lab sequence (of Fig.fig:lab-seq). For FOE positions within the image the value of $\Phi(\hat{\mathbf{t}})$ is coded as grey value after logarithmic scaling.

To find the minimum of Φ and thus the apparent translation, we perform a hierarchical search over the two-dimensional space of epipole positions. In practice, function Φ is quite smooth, that is small changes in $\hat{\mathbf{t}}$ give rise to only small changes in Φ (for an example see Fig. 3). One of the reasons for this is that for any $\hat{\mathbf{t}}$, the value of $\Phi(\hat{\mathbf{t}})$ is influenced by all the normal flow measurements and not only by a small subset.

For a majority of motion sequences, the motion of the camera does not change abruptly. Then the translation does not change much between frames and a complete search only has to be performed for the first flow field. In the successive flow fields, we need to search only in a smaller area centered around the previously estimated translation.

4.4. Algorithm Analysis

As the following analysis proves, the depth variability measure for a single image region can be decomposed into two components; one component (independent of the motion estimate) which measures the deviation of the patch from a smooth scene patch (a plane in the given analysis) and a second component which constitutes a multiple of the epipolar constraint.

To show this, consider the function Θ_0 in a small image region $\mathcal{R}$. The vectors $\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}})$ are polynomial functions of image position $\mathbf{r}$ and can usually be approximated by constants within the region. We use a local coordinate system where $\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}})$ is parallel to $[1, 0, 0]^{\text{T}}$. Without loss of generality we can write (in that coordinate system)

$$\begin{aligned} \mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) &= [1, 0, 0]^{\text{T}} \\ \mathbf{u}_{\text{rot}}(\hat{\mathbf{A}}_{\text{c}}) &= [u_{\text{rx}}, u_{\text{ry}}, 0]^{\text{T}} \\ \mathbf{n}_i &= [\cos\psi_i, \sin\psi_i, 0]^{\text{T}} \\ u_{\text{n}i} &= \dot{\mathbf{r}}_i \cdot \mathbf{n}_i \end{aligned} \tag{35}$$

We can rewrite (28) as

$$\begin{aligned} \Theta_0 = \sum_i W_i (u_{\text{n}i} - (\mathbf{z} \cdot \mathbf{r}_i) \cos\psi_i \\ - u_{\text{rx}} \cos\psi_i - u_{\text{ry}} \sin\psi_i)^2 \end{aligned} \tag{36}$$

Note that u_{rx} can be incorporated into $\mathbf{z}$ (writing $\mathbf{z}' = \mathbf{z} + [0, 0, u_{\text{rx}}/f]^{\text{T}}$) and we thus obtain the same minimum for the simplified expression

$$\Theta_0 = \sum_i W_i (u_{\text{n}i} - (\mathbf{z} \cdot \mathbf{r}_i) \cos\psi_i - u_{\text{ry}} \sin\psi_i)^2 \tag{37}$$

Now consider the least squares estimation of optical flow in the region using weights W_i. Allowing linear depth changes, in the local coordinate system we fit flow $(\mathbf{u}_{\text{x}} \cdot \mathbf{r}, u_{\text{y}})$, i.e., a linear function along the direction of $\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}})$ and a constant in the perpendicular direction. We would minimize

$$\sum_i W_i \, (u_{\text{n}i} - (\mathbf{u}_{\text{x}} \cdot \mathbf{r}_i) \cos\psi_i - u_{\text{y}} \sin\psi_i)^2 \tag{38}$$

Expressions (37) and (38) are almost identical, but there is one important difference. The optical flow minimization (38) is strictly local, using only measurements from the region. On the other hand, in (37), the rotational flow $(u_{\text{rx}}, u_{\text{ry}})$ is determined by the global motion parameters.

Let us denote the least squares solution of (38) as $(\hat{\mathbf{u}}_{\text{x}}, \hat{u}_{\text{y}})$ and the residual as E_{F}. After some vector and matrix manipulation we can obtain

$$\Theta_1 = \left(m_{\text{ss}} - \mathbf{m}_{\text{cs}}^{\text{T}} \mathbf{M}_{\text{cc}}^{-1} \mathbf{m}_{\text{cs}}\right) \delta u_{\text{ry}}^2 + E_{\text{F}} = K \, \delta u_{\text{ry}}^2 + E_{\text{F}} \tag{39}$$

where

$$m_{\text{ss}} = \sum_i W_i \, \sin^2\psi_i,$$

$$\mathbf{m}_{\text{cs}} = \sum_i W_i \, \cos\psi_i \sin\psi_i \mathbf{r}_i,$$

$$\mathbf{M}_{\text{cc}} = \sum_i W_i \, \cos^2\psi_i \mathbf{r}_i \mathbf{r}_i^{\text{T}}$$

and

$$K = m_{\text{ss}} - \mathbf{m}_{\text{cs}}^{\text{T}} \mathbf{M}_{\text{cc}}^{-1} \mathbf{m}_{\text{cs}} \tag{40}$$

The expression $\delta u_{\text{ry}} = u_{\text{ry}} - \hat{u}_{\text{y}}$ is the difference of the globally determined rotational component u_{ry} and the best local optical flow component $\hat{u}_{\text{y}}$. Both of the components are in the direction perpendicular to the translational flow and δu_{ry} is therefore the epipolar distance.

In classical approaches to motion estimation the epipolar distance is minimized. Here we are minimizing the sum of two terms. The first component in (39) is related to the epipolar constraint. It amounts to the square of the epipolar distance, which only depends on the 3D motion estimate, times a factor K, which depends on the gradient distribution in the patch. The second component in (39), E_{F}, represents how well the scene is approximated by a plane and it is independent of the 3D motion estimate. In classical approaches this component is derived during the process of optical flow computation, and it is then discarded and not used anymore in the process of estimating 3D motion and structure. This component, however, carries information about the scene discontinuities, and we have used it in related work for segmenting the scene (Brodský et al., 1998b). In the case of a smooth patch, $E_{\text{F}} = 0$ and our technique is similar to epipolar minimization. The difference lies only in the multiplication factor K.

Next let us take a closer look at the multiplication factor (Eq. (40)). It measures the range of normal flow directions within the region. If a region contains only a small range of directions, it may not provide reliable information for all candidate translations and appropriately (40) will be small for such a region. On the other hand, (40) will be large if the region contains a large range of measurement directions. We see that compared to the epipolar constraint, the depth smoothness measure emphasizes regions with larger variation of normal flow directions and can thus be expected to yield better results for noisy data.

4.5. *The* **A** *Matrix Parameterization*

The motion estimation algorithm provides the apparent translation $\hat{\mathbf{t}} = \mathbf{K}\mathbf{t}$ and the matrix $\mathbf{S}(\mathbf{A}, \hat{\mathbf{t}})$ (see Eq. (14)).

The copoint matrix $\mathbf{A}_c$ is then obtained by applying a known linear function f_c (see Appendix A for details)

$$\mathbf{A}_c = f_c(\mathbf{S}(\mathbf{A}, \tilde{\mathbf{t}})) \tag{41}$$

The second part of matrix $\mathbf{A}$ is of the form $\mathbf{A}_t = \hat{\mathbf{t}}\mathbf{w}^T + w_0\mathbf{I}$. Note that w_0 does not influence the flow due to $\mathbf{A}_t$ and can be easily computed using the condition trace $\mathbf{A} = 0$. Since trace $(\hat{\mathbf{t}}\mathbf{w}^T) = \hat{\mathbf{t}} \cdot \mathbf{w}$, we have

$$w_0 = -\frac{1}{3}(\text{trace } (\mathbf{A}_c) + \hat{\mathbf{t}} \cdot \mathbf{w}) \tag{42}$$

We write $\mathbf{w} = \sum_{i=1}^{3} w_i \mathbf{e}_i$, where $\mathbf{e}_1 = [1, 0, 0]^T$, $\mathbf{e}_2 = [0, 1, 0]^T$, $\mathbf{e}_3 = [0, 0, 1]^T$ and define

$$\mathbf{B}_0 = \mathbf{A}_c - \frac{1}{3}\,\text{trace } (\mathbf{A}_c)\,\mathbf{I}$$

and

$$\mathbf{B}_i = \hat{\mathbf{t}}\mathbf{e}_i^T - \frac{1}{3}(\hat{\mathbf{t}} \cdot \mathbf{e}_i)\,\mathbf{I} \quad \text{for } i = 1, \ldots, 3$$

Then

$$\mathbf{A} = \mathbf{B}_0 + \sum_{i=1}^{3} w_i \mathbf{B}_i \tag{43}$$

where $\mathbf{B}_i$ are known matrices (depending on $\hat{\mathbf{t}}$ and $\mathbf{A}_c$ only). In addition, vector $\mathbf{w}$ is not completely arbitrary; there are only two independent unknowns, as matrix $\mathbf{A}$ is singular. However, we choose not to enforce the singularity constraint here, because it amounts to a third order polynomial equation and introduces unnecessary complexity in the sequel.

4.6. Calibration from General Motion

For a general motion sequence, only the copoint matrices can be computed. We assume that a set of partial results in the form of (43) has been obtained with the FOE and/or the camera rotation changing between frames.

The method is again based on the constraint (20). Only partial information about $\mathbf{A}$ is available and thus the error function (26) depends not only on $\hat{\mathbf{K}}$, but also on the unknown vector $\mathbf{w}$. However, for any given $\hat{\mathbf{K}}$, we can choose the $\mathbf{w}$ that minimizes (26) in order to define an error measure that only depends on $\hat{\mathbf{K}}$. The best $\mathbf{w}$ can be expressed in closed form as a function of $\hat{\mathbf{K}}$.

Consider the error function (26)

$$\begin{aligned}\text{trace } (\mathbf{E}_i\mathbf{E}_i) = {} & 2\,\text{trace } (\mathbf{A}_i\mathbf{A}_i) \\ & + 2\,\text{trace } \left(\mathbf{A}_i\hat{\mathbf{K}}\hat{\mathbf{K}}^T\mathbf{A}_i^T(\hat{\mathbf{K}}\hat{\mathbf{K}}^T)^{-1}\right)\end{aligned}$$

and substitute $\mathbf{A}_i$ given by (43)

$$\mathbf{A}_i = \mathbf{B}_{i0} + \sum_{j=1}^{3} w_{ij}\mathbf{B}_{ij}$$

If we denote

$$t_{ijk} = 2\,\text{trace } \left(\mathbf{B}_{ij}\mathbf{B}_{ik}\right) + 2\,\text{trace } \left(\mathbf{B}_{ij}\hat{\mathbf{K}}\hat{\mathbf{K}}^T\mathbf{B}_{ik}^T(\hat{\mathbf{K}}\hat{\mathbf{K}}^T)^{-1}\right)$$

the error function trace $(\mathbf{E}_i\mathbf{E}_i)$ becomes

$$\begin{aligned}\text{trace } (\mathbf{E}_i\mathbf{E}_i) = {} & t_{i00} + \sum_{j=1}^{3} w_{ij}\, t_{ij0} + \sum_{k=1}^{3} w_{ik}\, t_{i0k} \\ & + \sum_{j=1}^{3}\sum_{k=1}^{3} w_{ij}\, w_{ik}\, t_{ijk}\end{aligned} \tag{44}$$

or written in matrix form:

$$\text{trace } (\mathbf{EE}) = t_{i00} + \mathbf{d}_i^T\mathbf{w}_i + \mathbf{w}_i^T\mathbf{D}_i\mathbf{w}_i$$

where $\mathbf{w} = [w_{i1}, w_{i2}, w_{i3}]^T$ is a vector of the unknown parameters, $\mathbf{d}_i = [t_{i01} + t_{i10}, t_{i02} + t_{i20}, t_{i03} + t_{i30}]^T$, and

$$\mathbf{D}_i = \begin{pmatrix} t_{i11} & t_{i12} & t_{i13} \\ t_{i21} & t_{i22} & t_{i23} \\ t_{i31} & t_{i32} & t_{i33} \end{pmatrix}$$

Note that $t_{ij} = t_{ji}$, so matrix $\mathbf{D}$ is symmetric.

Given $\hat{\mathbf{K}}$, quantities t_{ij} are known and the vector $\mathbf{w}$ that minimizes the error function can be expressed in closed form as

$$\mathbf{w}_i = -\frac{1}{2}\,\mathbf{D}_i^{-1}\mathbf{d}_i \tag{45}$$

The global error function is the sum of the local minimization results

$$\begin{aligned}\mathcal{E}_3(\hat{\mathbf{K}}) &= \sum_i (\min\,(\text{trace } (\mathbf{E}_i\mathbf{E}_i))) \\ &= \sum_i \left(t_{i00} - \frac{1}{4}\,\mathbf{d}_i^T(\mathbf{D}_i)^{-1}\mathbf{d}_i\right)\end{aligned} \tag{46}$$

Expression (46) defines an error measure in terms of matrix $\hat{\mathbf{K}}$ alone. We again perform a Levenberg-Marquardt minimization to obtain $\mathbf{K}$. It is possible to obtain closed form (and succinct) formulas for the derivatives of $\mathcal{E}_3(\hat{\mathbf{K}})$ with respect to $\hat{\mathbf{K}}$ using vector and matrix differentiation of scalar functions.

As with $\mathcal{E}_1(\hat{\mathbf{K}})$, the expressions in error function $\mathcal{E}_3(\hat{\mathbf{K}})$ are dominated by the focal lengths f_x, f_y. To find a suitable starting point, we first test diagonal matrices $\hat{\mathbf{K}}$ (with $\Delta_x = \Delta_y = s = 0$) with reasonable values of f_x, f_y, based on the size of the image and expected minimum and maximum field of view. Only a very sparse set of matrices is tested and the matrix yielding the best value is used as a starting point for the minimization in all five calibration parameters. This preprocessing step yields very good results in practice.

4.7. Overview of the Algorithm

Here we summarize the steps performed during the calibration for a general motion sequence. The input is assumed to be a monocular sequence of images taken by a moving camera with constant calibration parameters.

First, for each pair of successive frames, we compute normal flow, estimate the apparent FOE and the copoint matrix (Section 4.3), and then compute the corresponding partial matrices $\mathbf{B}_i$ (Eq. (43)). All the partial matrices are then used in the computation of the calibration parameters (Section 4.6).

5. Experiments

Before presenting the experimental results, we discuss the constant f, which is used as the third component of image points throughout the computation. Even though the value of f does not influence the results in the noiseless case, it can be very important in the presence of noise. For example, it was observed in Dron (1993) that the choice of $f = 1$ leads to numerical instability. As explained in Hartley (1997), the main reason of instability seems to be the inhomogeneity of vectors $[x, y, 1]^{\mathrm{T}}$, since in a typical image, the average coordinates x, y are much larger than 1. A simple solution is to select f comparable with average image coordinates x, y.

In our experiments, we place the origin of the image coordinate system in the middle of the image. If the width of the image is W, the x coordinates range from $-W/2$ to $W/2$ and the average absolute value for x is $W/4$. We compute the average of the image width W and height H and set f to be one half of that value, $f = (W + H)/4$. This value seems to lead to better results than $(W + H)/8$.

We first present experiments using artificially generated data that allow us to compare the various methods and to study the effects of noise. In the second part of this section we show the performance of the algorithms on real image sequences.

5.1. Experiment 1

We used noisy, artificially generated flow fields for a camera with the following parameters: $f_x = 500$, $f_y = 520$, $\Delta_x = -10$, $\Delta_y = 5$, $s = 0$. Eight input matrices were computed from purely rotational fields for different rotations (of the same magnitude but with different rotation axes). In the plots σ denotes the standard deviation of the Gaussian noise that was added to the normal flow measurements (that is, the length of the normal flow vectors). Noise with standard deviation $\sigma = 0.02$ added on average normal flow errors (ratio of noise over actual normal flow value) of approximately 30%. (The normal flow values were in the range of -0.35 to 0.35 pixels in length, with the average absolute value about 0.15 pixels.)

The plots in this section show how the recovered parameters change with respect to noise. In addition to the five calibration parameters, we also plot the aspect ratio f_x/f_y. At each noise level, we repeated the experiment 20 times and we used a statistical package from Maple V for the plots. The results for each variable are drawn as a box with a central line showing the median of the data and two lines showing the first and the third quartile, respectively. The lines extending from the box have maximum length 3/2 of the interquartile length, but not exceeding the range of the data. A horizontal dashed line shows the ground truth value and we also plot a curve connecting the median values at different noise levels.

The results of the eigenvector method that minimizes (24) are illustrated in Fig. 4 with one plot for each calibration parameter plus a plot showing the recovered aspect ratio f_x/f_y.

The performance of the non-linear method for purely rotational motion is comparable, or slightly worse than the performance of the eigenvector method, as can be seen from Fig. 5. Note that in both cases the aspect ratio f_x/f_y is estimated quite robustly even for very noisy inputs.

The image size in the experiments above was 256×256 pixels, so that constant $f = 128$ was used. For the noise level $\sigma = 0.02$, we also tested different values of f and found that both methods are not sensitive to the value of f. We only plot recovered f_x and Δ_x for both

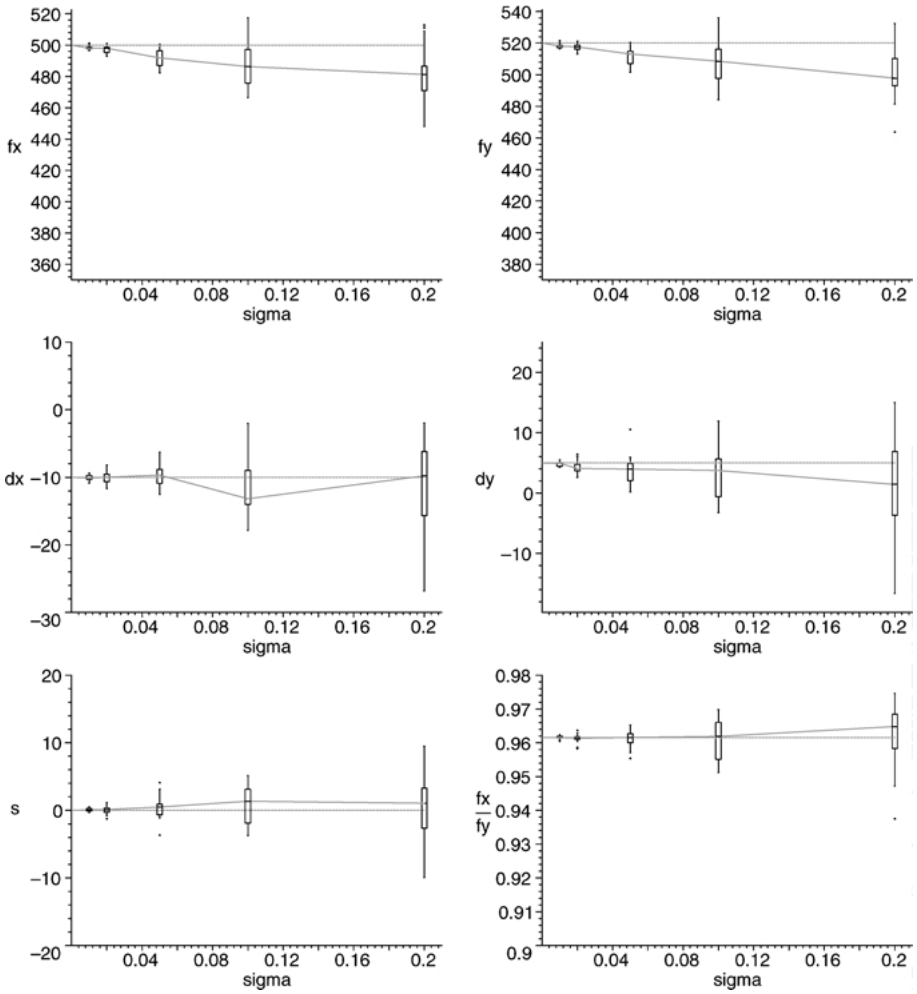

Figure 4. Self-calibration results for rotational motion, eigenvector method.

methods in Fig. 6. The plots for the other parameters are analogous.

5.2. *Experiment 2*

In the second experiment with artificial data we added a translational component to the generated flow field. We first tested an FOE lying in the image (at (40, −100) in image coordinates). In this and all other experiments the size of the patches (to which planes were fitted) was 10×10 pixels. The results for different noise levels are plotted in Fig. 7. It should be noted that a value of $N = 0.1$ corresponds to a quite severe noise level.

We then performed the same set of experiments, but for an FOE lying far from the image (at (4990, −10395) in image coordinates) and present the results in Fig. 8. The data confirms the findings of theoretical studies regarding the confusion between translation and rotation; the estimated FOEs tend to lie along a direction from the center of the image towards the actual FOE.

5.3. *Experiment 3*

In the last experiment with artificial data we added a translational component with apparent FOE lying in the range of (40, −100) and (57, −91) (in the x- and y-image coordinates, where (0, 0) is the center of the image) to the motions used in Experiment 1. The size of the translational flow vectors was on average about 40%

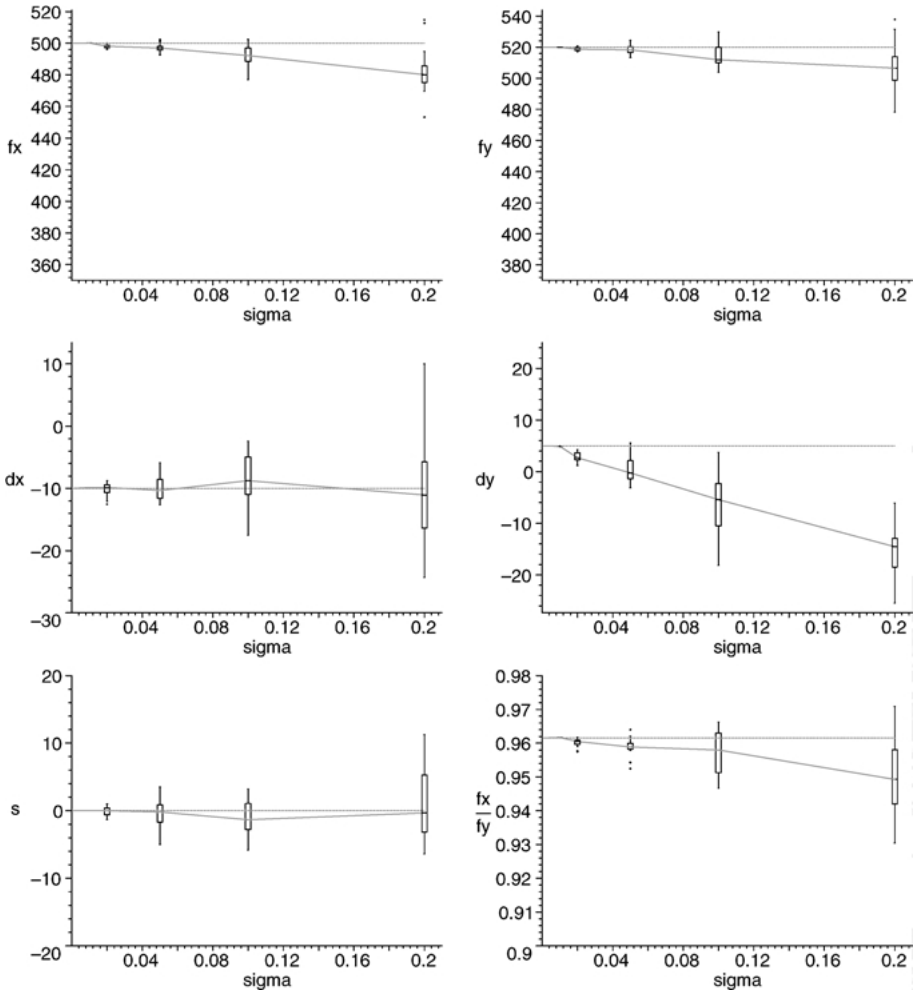

Figure 5. Self-calibration results for rotational motion, non-linear method.

of the size of the rotational vectors. The self-calibration results are presented in Fig. 9.

As in Experiment 1, we also tested different values of f, using the noise level $\sigma = 0.01$, as shown in Fig. 10. Because self-calibration is more noise sensitive for general camera motions, the value of f influences the results much more than for the purely rotational motions. Note that this experimental result confirms that the chosen value $f = 128$ is a suitable choice.

5.4. *Experiment 4*

Performance of the methods for the case of significantly non-square pixels is illustrated in this experiment. We chose a camera with calibration parameters $f_x = 100$, $f_y = 320$, $\Delta_x = 50$, $\Delta_y = -25$, $s = 10$. All the other settings, including the camera motions, were identical to those in Experiment 1.

Only plots for the eigenvector method are shown in Fig. 11; the results for the non-linear method were very similar.

5.5. *Experiment 5*

To test the performance of the method for purely rotational motion on real images, we used a short sequence of computer generated images (see Fig. 12) with calibration parameters $f_x = f_y = 600$, $\Delta_x = \Delta_y = s = 0$.

Again we compared the linear and the non-linear method. Since the input images were highly textured

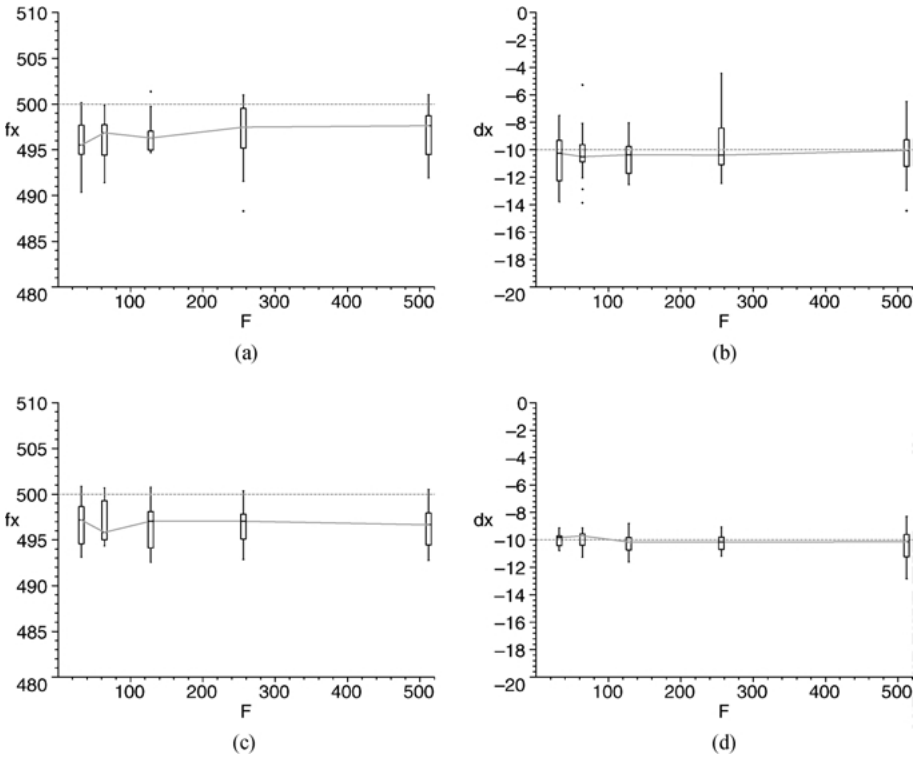

Figure 6. Dependence of the recovered parameters on f for purely rotational motion. The noise level was $\sigma = 0.02$. (a), (b): the eigenvector method, (c), (d): the non-linear method.

and provided many measurements, we performed self-calibration using all the normal flow measurements first and then repeated the experiment with several randomly chosen subsets of the input data to gauge the reliability of the results. The normal flow values were maximally 3.5 pixels long, and the average length was 0.3 pixels.

The linear method gave the results in Table 1, the results of the non-linear method are shown in Table 2.

For the non-linear method, it is also possible to only solve for some of the parameters. In addition to the general calibration, in Table 3 we present the results obtained when solving only for the unknowns f_x, f_y with the remaining parameters Δ_x, Δ_y, and s set to 0. Notice, however, that since the focal lengths dominate the error function, the results were quite close to the results obtained in the general case.

Table 1. Recovered calibration parameters for the purely rotational sequence, using the eigenvector method.

% of data used	f_x	f_y	f_y/f_x	Δ_x	Δ_y	s
100	619.4	619.6	1.000	−1.1	1.4	10.7
80	625.3	624.2	0.998	−0.8	2.1	10.6
60	624.7	624.0	0.999	1.3	1.2	9.9

Table 2. Recovered calibration parameters for the purely rotational sequence, using the non-linear method.

% of data used	f_x	f_y	f_y/f_x	Δ_x	Δ_y	s
100	609.2	624.9	1.026	−0.6	2.5	18.6
80	614.5	630.6	1.026	−0.8	4.8	18.8
60	615.8	629.6	1.022	0.6	2.6	17.7

Table 3. Recovered focal lengths for the purely rotational sequence. The remaining parameters were set to 0.

% of data used	f_x	f_y	f_y/f_x
100	610.1	626.4	1.027
80	616.3	632.6	1.026
60	616.9	631.2	1.023

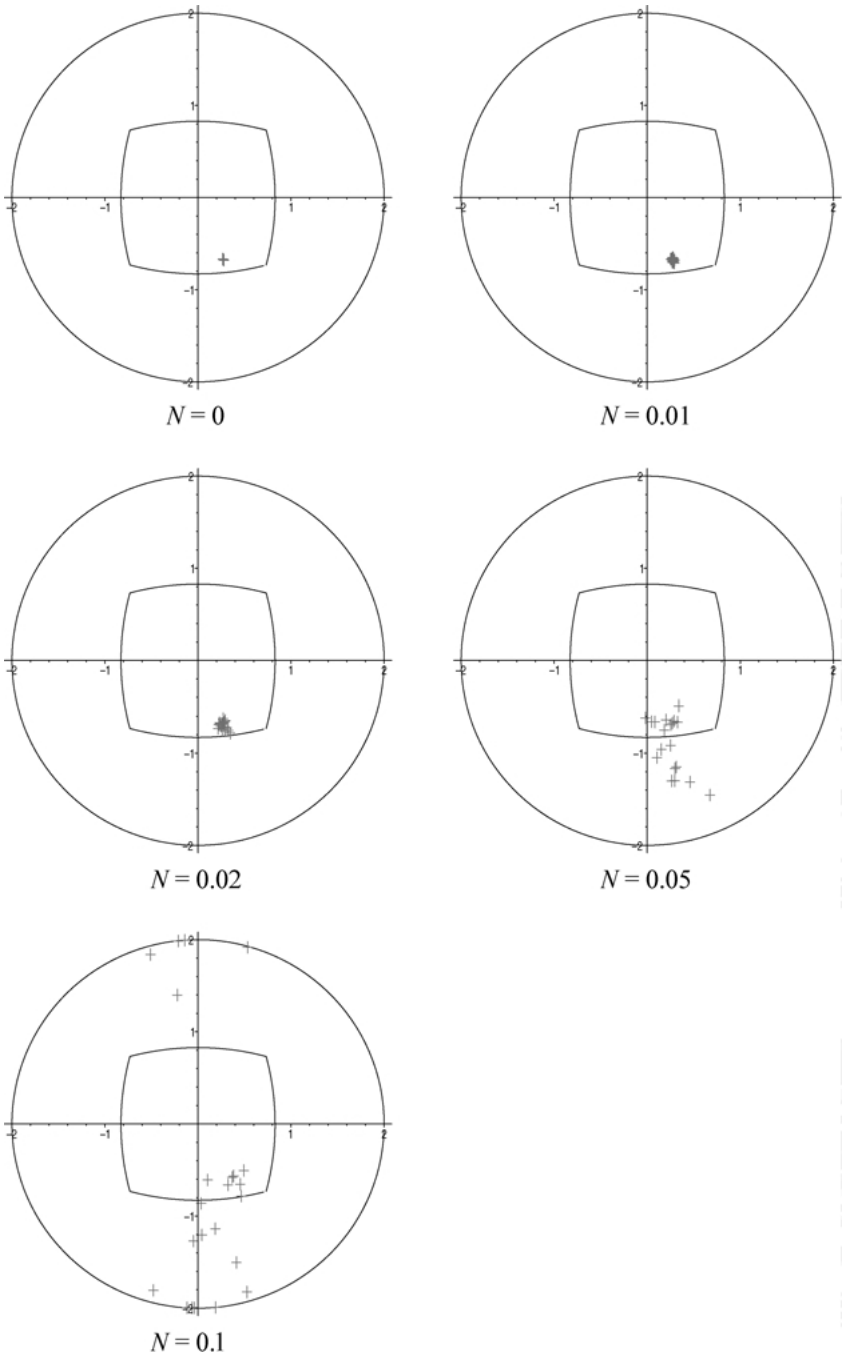

Figure 7. FOE estimation results. Each plot shows the hemisphere of possible translation directions (displayed under stereographic projection), the boundary of the image, and estimated FOE directions for flow fields with a constant translational component and varying rotational components.

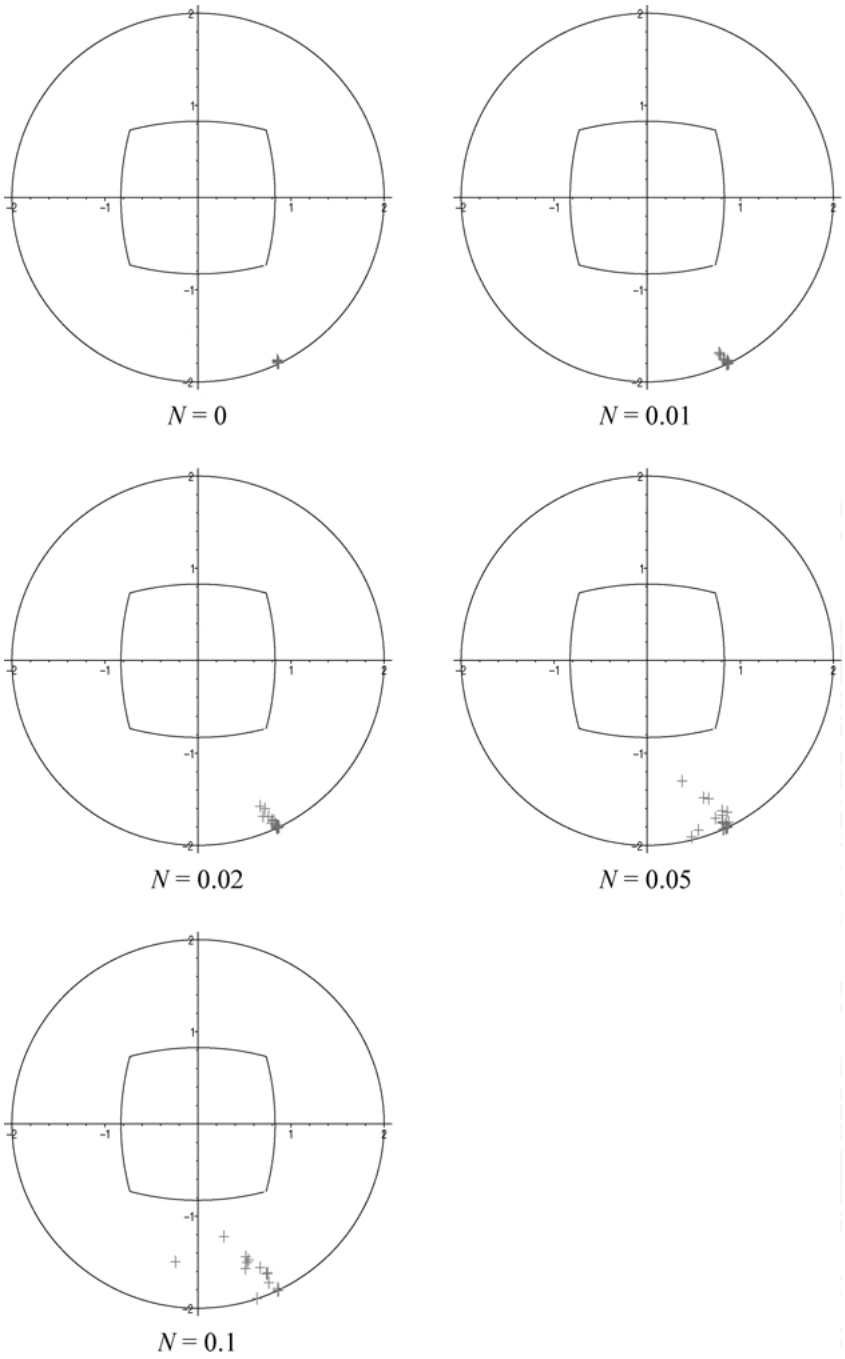

Figure 8. FOE estimation results. Each plot shows the hemisphere of possible translation directions (displayed under stereographic projection), the boundary of the image, and estimated FOE directions for flow fields with a constant translational component and varying rotational components.

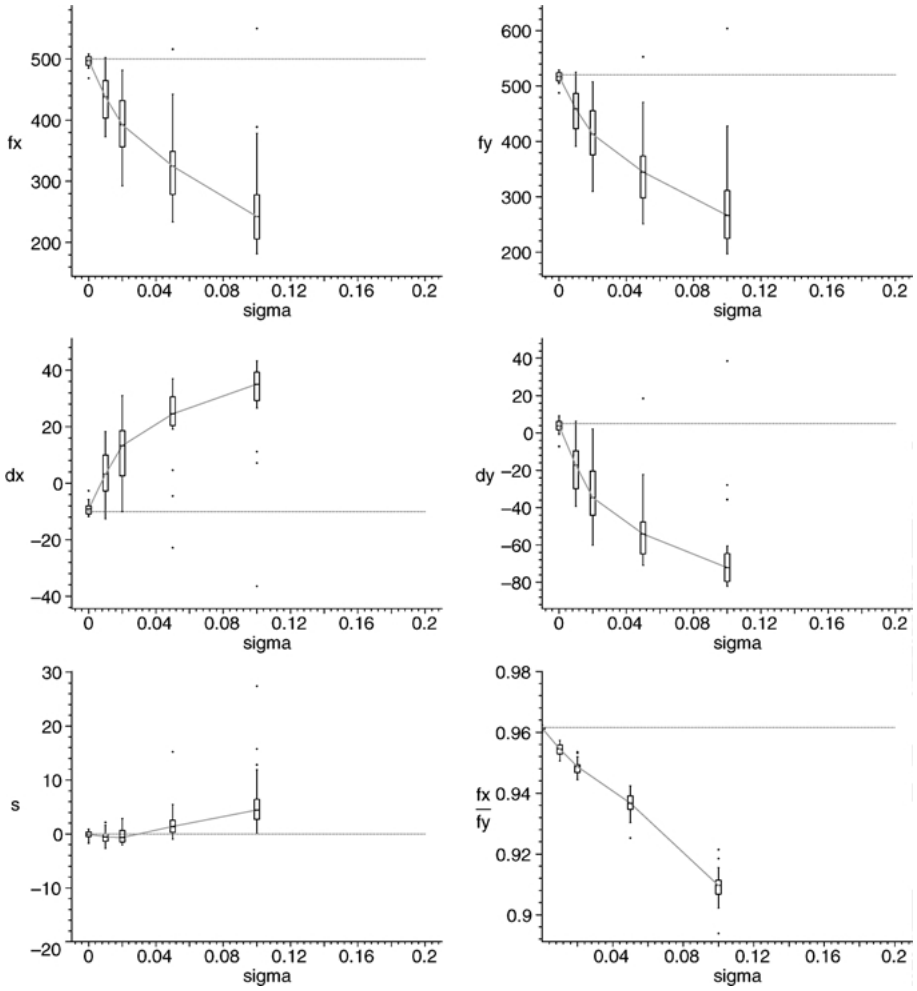

Figure 9. Self-calibration results for a general motion.

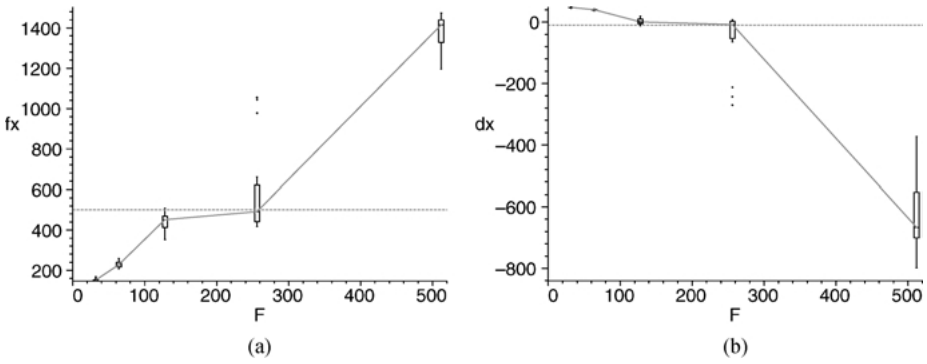

Figure 10. Dependence of the recovered parameters on f for a general motion. The noise level was $\sigma = 0.01$. The program automatically chooses value $f = 128$.

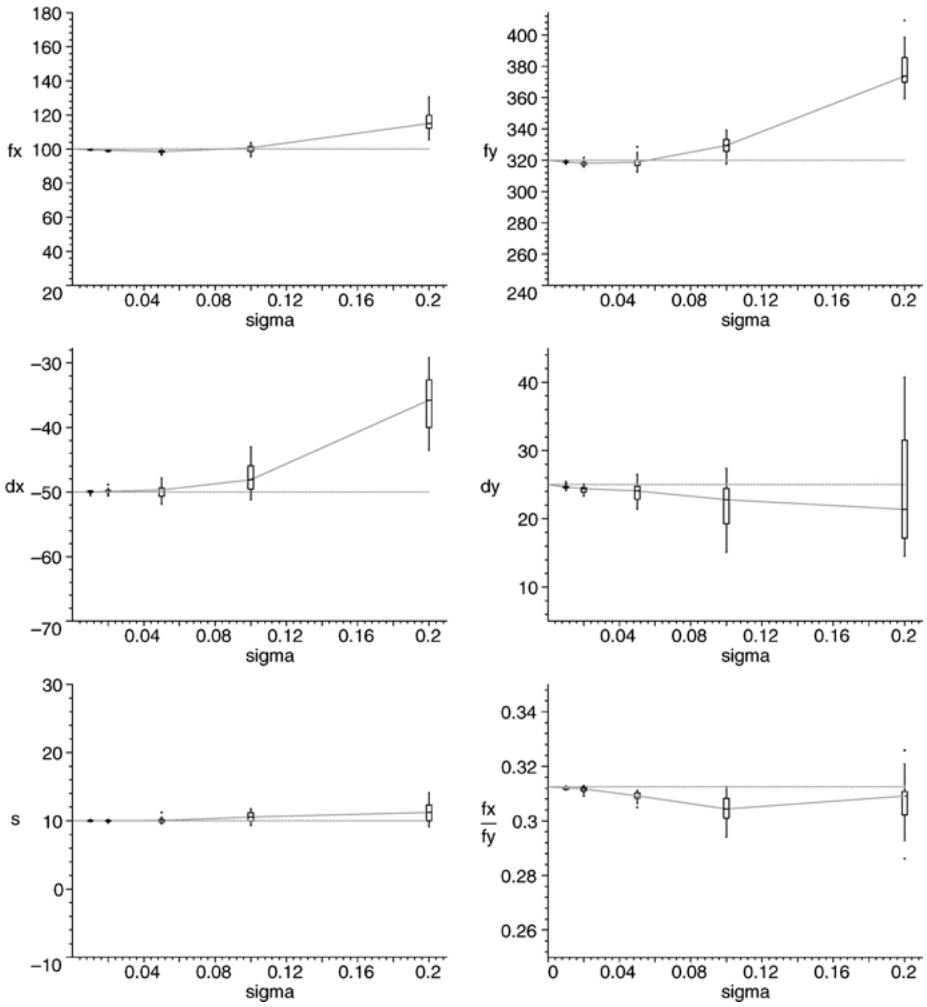

Figure 11. Self-calibration results for rotational motion, eigenvector method.

5.6. Experiment 6

The "lab" sequence (Fig. 13) was taken by a hand-held Panasonic D5000 camera which underwent a general translation and rotation with a zoom setting of approximately 12 mm. Unfortunately, the effective focal length of the pinhole camera model was also influenced by the focus setting and we thus knew the intrinsic parameters only approximately. The internal parameters were fixed and approximately: $f_x = f_y = 450$, $\Delta_x = \Delta_y = s = 0$. In this sequence the normal flow values were up to 4 pixels in length, with an average absolute value of approximately 1.3 pixels. Calibration results using the general algorithm described in Section 4 are summarized in Table 4; the focal lengths were slightly overestimated, but consistent for different parts of the sequence.

From the experimental results, we make the following observations about the feasibility of self-calibration from normal flow input.

The focal length parameters f_x, f_y, and especially the aspect ratio f_x/f_y, can be estimated most reliably. It is more difficult to recover the remaining parameters. One of the reasons is that f_x, f_y are much larger in

Table 4. Self-calibration results for the lab sequence.

Frames	f_x	f_y	Δ_x	Δ_y	s
001–300	536	522	16	26	3
001–100	541	543	−33	6	−25
101–200	544	475	26	−38	14
201–300	548	513	−11	8	6

Figure 12. One frame from the purely rotational sequence used in Experiment 5.

Figure 13. One input frame of the "lab" sequence.

magnitude than the remaining parameters (at least for any real camera) and the error functions depend only on the ratios of the calibration parameters.

6. Summary

We have analyzed the problem of estimating the calibration parameters from image motion fields for an uncalibrated camera moving in a rigid way. Our theoretical analysis of such motion fields has shown that the rotation and calibration parameters are coupled in a way leading to a set of parameters which are linearly related to the image motion measurements, and from these parameters the calibration can be determined. We have analyzed the information about the calibration parameters contained in flow fields and provided a geometrical interpretation of what information can be obtained from a single motion field.

We have given calibration algorithms for cameras in constrained and unconstrained rigid motion. For a camera undergoing a purely rotational motion, we have presented a linear calibration method, as well as a more elaborate non-linear iterative method that can be easily adjusted to specific situations where a reduced set of parameters needs to be estimated.

For the case of general camera motion, we have presented an iterative self-calibration procedure that combines information from several frames to overcome the inherent ambiguities of the problem. This procedure was implemented using a novel FOE estimation approach, utilizing the smoothness of estimated scene depth.

Finally, experiments using both artificial and real input data have been carried out to test the performance of the method.

Appendix A: The Copoint Matrix

In this Appendix we prove the observation from Section 2.3, utilizing a rotated coordinate system. Let $\mathbf{M}$ be a rotation matrix that transforms the unit vector $\hat{\mathbf{s}}$ into $\hat{\mathbf{z}}$, i.e., $\mathbf{M}\hat{\mathbf{s}} = \hat{\mathbf{z}}$. Matrix $\mathbf{M}$ is not uniquely defined, but any such matrix can be used, as long as $\det(\mathbf{M}) = 1$ so that the handedness of the coordinate system is maintained. Then the transformation of axial vectors (Borisenko and Tarapov, 1986) (such as cross-products) does not involve a change of sign.

In the coordinate system rotated by $\mathbf{M}$, vector $\mathbf{r}$ becomes $\mathbf{r}' = \mathbf{M}\mathbf{r}$ and matrix $\mathbf{A}$ transforms into $\mathbf{A}' = \mathbf{M}\mathbf{A}\mathbf{M}^{\mathrm{T}}$. The reason for the rotation becomes clear when we examine $[\hat{\mathbf{s}}]_\times$ in the new coordinate system. As $\mathbf{M}\hat{\mathbf{s}} = \hat{\mathbf{z}}$, we obtain

$$\mathbf{M}[\hat{\mathbf{s}}]_\times \mathbf{M}^{\mathrm{T}} = [\hat{\mathbf{z}}]_\times = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Denoting

$$\mathbf{A}' = \mathbf{M}\mathbf{A}\mathbf{M}^{\mathrm{T}}$$

we can also derive

$$\mathbf{M}\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})\mathbf{M}^{\mathrm{T}} = \mathbf{A}'^{\mathrm{T}}[\hat{\mathbf{z}}]_\times - [\hat{\mathbf{z}}]_\times \mathbf{A}' = \mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}) \quad (47)$$

Let the elements of $\mathbf{A}'$ be a'_{ij}. Conic $\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}})$ is then

$$\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}) = \begin{pmatrix} 2a'_{21} & a'_{22} - a'_{11} & a'_{23} \\ a'_{22} - a'_{11} & -2a'_{12} & -a'_{13} \\ a'_{23} & -a'_{13} & 0 \end{pmatrix} \quad (48)$$

We can split $\mathbf{A}'$ into two parts, a component $\mathbf{A}'_{\mathrm{c}}$ which can be derived from $\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}})$ and a component $\mathbf{A}'_{\mathrm{t}}$ which only affects the flow components parallel to the translational flow due to translation $\hat{\mathbf{s}}$:

$$\mathbf{A}' = \mathbf{A}'_{\mathrm{c}} + \mathbf{A}'_{\mathrm{t}} = \begin{pmatrix} 0 & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} - a'_{11} & a'_{23} \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} a'_{11} & 0 & 0 \\ 0 & a'_{11} & 0 \\ a'_{31} & a'_{32} & a'_{33} \end{pmatrix} \quad (49)$$

Only the difference between the first two diagonal elements of $\mathbf{A}'_{\mathrm{c}}$ is important; other choices just lead to slightly different representations of the same five parameters and do not change the computations in the sequel.

The split of $\mathbf{A}'$ naturally induces a split of $\mathbf{A} = \mathbf{M}^{\mathrm{T}}\mathbf{A}'\mathbf{M}$:

$$\mathbf{A} = \mathbf{A}_{\mathrm{c}} + \mathbf{A}_{\mathrm{t}} = \mathbf{M}^{\mathrm{T}}\mathbf{A}'_{\mathrm{c}}\mathbf{M} + \mathbf{M}^{\mathrm{T}}\mathbf{A}'_{\mathrm{t}}\mathbf{M}$$

Due to linearity, the flow due to $\mathbf{A}'$ is a sum of the flows due to $\mathbf{A}'_{\mathrm{c}}$ and $\mathbf{A}'_{\mathrm{t}}$. Clearly $\mathbf{A}'_{\mathrm{c}}$ encodes the same information as $\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}})$ and consequently matrix $\mathbf{A}_{\mathrm{c}}$ encodes the same information as $\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})$.

Matrix $\mathbf{A}'_{\mathrm{t}}$ contains the remaining parameters. Alternatively, it can be written as

$$\mathbf{A}'_{\mathrm{t}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ a'_{31} & a'_{32} & a'_{33} - a'_{11} \end{pmatrix} + \begin{pmatrix} a'_{11} & 0 & 0 \\ 0 & a'_{11} & 0 \\ 0 & 0 & a'_{11} \end{pmatrix} = \hat{\mathbf{z}}\mathbf{w}'^{\mathrm{T}} + w_0\,\mathbf{I}$$

where $\mathbf{w}' = (a'_{31},\ a'_{32},\ a'_{33} - a'_{11})^{\mathrm{T}}$ and $w_0 = a'_{11}$. Thus

$$\mathbf{A}_{\mathrm{t}} = \mathbf{M}^{\mathrm{T}}\hat{\mathbf{z}}\mathbf{w}'^{\mathrm{T}}\mathbf{M} + w_0\,\mathbf{M}^{\mathrm{T}}\mathbf{I}\mathbf{M} = \hat{\mathbf{s}}\mathbf{w}^{\mathrm{T}} + w_0\,\mathbf{I}$$

where $\mathbf{w} = \mathbf{M}^{\mathrm{T}}\mathbf{w}'$. The flow due to $\mathbf{A}_{\mathrm{t}}$ is

$$\mathbf{u}_{\mathrm{rot}}(\mathbf{A}_{\mathrm{t}}) = \frac{1}{f}(\hat{\mathbf{z}} \times (\mathbf{r} \times (\hat{\mathbf{s}}\mathbf{w}^{\mathrm{T}}\mathbf{r} + w_0\,\mathbf{r}))))$$

$$= -\frac{\mathbf{w}\cdot\mathbf{r}}{f}(\hat{\mathbf{z}} \times (\hat{\mathbf{s}} \times \mathbf{r})) = (\mathbf{w}\cdot\mathbf{r})\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{s}}) \quad (50)$$

As might have been expected, the flow vectors due to $\mathbf{A}_{\mathrm{t}}$ are parallel to the translational flow with apparent FOE $\mathbf{s}$. In fact, the flow due to $\mathbf{A}_{\mathrm{t}}$ is exactly the same as the flow field we would obtain from a translation with apparent FOE $\mathbf{s}$ and scene depth $1/Z = \mathbf{w}\cdot\mathbf{r}$. Such a plane has 3D equation $\mathbf{w}\cdot(\mathbf{KR}) = 1$.

Finally, we derive function f_{c} mapping $\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})$ into $\mathbf{A}_{\mathrm{c}}$. Let $\mathcal{T}$ be a function that converts a symmetric matrix into its upper triangular form (representing the same conic), so that

$$\mathcal{T}(\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}})) = 2\begin{pmatrix} a'_{21} & a'_{22} - a'_{11} & a'_{23} \\ 0 & -a'_{12} & -a'_{13} \\ 0 & 0 & 0 \end{pmatrix}$$

Then matrix $\mathbf{A}'_{\mathrm{c}}$ is simply $\mathbf{A}'_{\mathrm{c}} = \frac{1}{2}[\hat{\mathbf{z}}]_{\times}\mathcal{T}(\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}))$. Therefore

$$\mathbf{A}_{\mathrm{c}} = \frac{1}{2}\mathbf{M}^{\mathrm{T}}[\hat{\mathbf{z}}]_{\times}\mathbf{M}\mathbf{M}^{\mathrm{T}}\mathcal{T}(\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}))\mathbf{M} = \frac{1}{2}[\hat{\mathbf{s}}]_{\times}\mathbf{M}^{\mathrm{T}}\mathcal{T}(\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}))\mathbf{M}$$

Since $\mathbf{S}(\mathbf{A}', \hat{\mathbf{z}}) = \mathbf{M}\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})\mathbf{M}^{\mathrm{T}}$, we can define

$$f_{\mathrm{c}}(\mathbf{X}) = \frac{1}{2}[\hat{\mathbf{s}}]_{\times}\mathbf{M}^{\mathrm{T}}\mathcal{T}(\mathbf{M}\,\mathbf{X}\,\mathbf{M}^{\mathrm{T}})\mathbf{M} \quad (51)$$

and it is straightforward to verify that $f_{\mathrm{c}}(\mathbf{S}(\mathbf{A}, \hat{\mathbf{s}})) = \mathbf{A}_{\mathrm{c}}$.

Acknowledgments

The authors thank Prof. Yiannis Aloimonos for helpful discussions. The support of the Office of Naval Research under Grant N00014-96-1-0587, and IBM under Grant 50000293, is gratefully acknowledged.

Notes

1. Note that normal flow is an image based concept. The image may be distorted by an affine transform, but the component of the flow along some direction is obtained by projecting the flow on that direction. Normal flow does not amount to the transform of the component of flow along some direction in the calibrated image to the corresponding direction in the uncalibrated image.
2. We denote $(\mathbf{K}^{-1})^{\mathrm{T}}$ by $\mathbf{K}^{-\mathrm{T}}$.

References

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310.

Armstrong, M., Zisserman, A., and Hartley, R. 1996. Self-calibration from image triplets. In *Proc. European Conference on Computer Vision*, Cambridge, UK, vol. 1, pp. 3–16.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S. 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77.

Bergen, J.R., Anandan, P., Hanna, K.J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *Proc. European Conference on Computer Vision*, pp. 237–248.

Black, M. 1994. Recursive non-linear estimation of discontinuous flow fields. In *Proc. Third European Conference on Computer Vision*, Springer-Verlag, pp. 138–145.

Borisenko, A.I. and Tarapov, I.E. 1986. *Vector and Tensor Analysis with Applications*. Prentice-Hall: Englewood Cliffs, NJ.

Brodský, T., Fermüller, C., and Aloimonos, Y. 1998a. Self-calibration from image derivatives. In *Proc. International Conference on Computer Vision*, pp. 83–89.

Brodský, T., Fermüller, C., and Aloimonos, Y. 1998b. Simultaneous estimation of 3D motion and structure. In *Proc. European Conference on Computer Vision*, pp. 342–358.

Cheong, L., Fermüller, C., and Aloimonos, Y. 1998. Effects of errors in the viewing geometry on shape estimation. *Computer Vision and Image Understanding*, 71:356–372.

Dron, L. 1993. Dynamic camera self-calibration from controlled motion sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, pp. 501–506.

Faugeras, O.D. 1992. *Three-Dimensional Computer Vision*. MIT Press: Cambridge, MA.

Faugeras, O.D., Luong, Q.-T., and Maybank, S.J. 1992. Camera self-calibration: Theory and experiments. In *Proc. European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 321–334.

Fermüller, C. 1993. Navigational preliminaries. In *Active Perception*, Y. Aloimonos (Ed.). Advances in Computer Vision, Lawrence Erlbaum Associates: Hillsdale, NJ, ch. 3.

Fermüller, C. and Aloimonos, Y. 1995. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976.

Hartley, R.I. 1994a. An algorithm for self calibration from several views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 908–912.

Hartley, R.I. 1994b. Self-calibration from multiple views with a rotating camera. In *Proc. European Conference on Computer Vision*, Stockholm, Sweden, vol. 1, pp. 471–478.

Hartley, R.I. 1997. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:580–593.

Heeger, D.J. and Jepson, A.D. 1992. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7:95–117.

Horn, B.K.P. and Weldon, E.J. Jr. 1988. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76.

Lenz, R.K. and Tsai, R.Y. 1988. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:713–720.

Maybank, S.J. and Faugeras, O.D. 1992. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8:123–151.

Mendelsohn, J., Simoncelli, E., and Bajcsy, R. 1997. Discrete-time rigidity constrained optical flow. In *Proc. International Conference on Computer Analysis of Images and Patterns*, Springer: Berlin, pp. 255–262.

Nagel, H.-H. 1995. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15:271–288.

Nagel, H.-H. and Haag, M. 1998. Bias-corrected optical flow estimation for road vehicle tracking. In *Proc. International Conference on Computer Vision*, Bombay, India, pp. 1006–1011.

Negahdaripour, S. and Horn, B.K.P. 1987. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:163–176.

Pollefeys, M., Van Gool, L., and Oosterlinck, A. 1996. The modulus constraint: A new constraint for self-calibration. In *Proc. International Conference on Pattern Recognition*, Vienna, Austria, vol. A, pp. 349–353.

Spetsakis, M.E. and Aloimonos, J. 1990. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4:171–183.

Strang, G. 1988. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich.

Tsai, R.Y. 1986. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, pp. 364–374.

Viéville, T. and Faugeras, O.D. 1996. The first-order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64:128–146.