

Autocalibration from Planar Scenes

Bill Triggs

INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France
Bill.Triggs@inrialpes.fr — <http://www.inrialpes.fr/movi/people/Triggs>

Abstract. This paper describes the theory and a practical algorithm for the autocalibration of a moving projective camera, from $m \geq 5$ views of a *planar* scene. The unknown camera calibration, motion and scene geometry are recovered up to scale, from constraints encoding the motion-invariance of the camera's internal parameters. This extends the domain of autocalibration from the classical non-planar case to the practically common planar one, in which the solution can not be bootstrapped from an intermediate projective reconstruction. It also generalizes Hartley's method for the internal calibration of a rotating camera, to allow camera translation and to provide 3D as well as calibration information. The basic constraint is that orthogonal directions (points at infinity) in the plane must project to orthogonal directions in the calibrated images. Abstractly, the plane's two circular points (representing its Euclidean structure) lie on the 3D absolute conic, so their projections must lie on the absolute image conic (representing the camera calibration). The resulting algorithm optimizes this constraint numerically over all circular points and all projective calibration parameters, using the inter-image homographies as a projective scene representation.

Keywords: Autocalibration, Euclidean structure, Absolute Conic & Quadric, Planar Scenes.

1 Introduction

This paper describes a method of autocalibrating a moving projective camera with general, unknown motion and unknown intrinsic parameters, from $m \geq 5$ views of a *planar* scene. Autocalibration is the recovery of metric information from uncalibrated images using geometric self-consistency constraints. For example, the internal and external calibration of a moving projective camera can be recovered from the knowledge that the internal parameters are constant during the motion, and the inter-image consistency constraints that this entails. Since the seminal work of Maybank & Faugeras [14, 3], a number of different approaches have been developed [5, 6, 1, 26, 25, 2, 13, 9, 16, 15, 22, 10]. For the 'classical' problem of a single perspective camera with constant but unknown internal

Work supported by Esprit LTR project CUMULI. Many thanks to P. Sturm for discussions, G. Csurka and A. Ruf for test images and calibration data, and the anonymous reviewers for comments. An extended version is available from my web page.

parameters moving with a general but unknown motion in a 3D scene, the original ‘Kruppa equation’ approach [14] seems to be being displaced by methods based on the ‘rectification’ of an intermediate projective reconstruction [5, 9, 15, 22, 10]. More specialized methods exist for particular types of motion and simplified calibration models [6, 23, 1, 16]. Stereo heads can also be autocalibrated [26, 11]. Limited variations of the intrinsic parameters can also be handled [9, 15]. Hartley [6] gives a particularly simple internal calibration method for a ‘rotating camera’ — *i.e.* whose translation is known to be negligible relative to $n \geq 4$ distant, identifiable, real or synthetic points in the scene. Faugeras [2] advocates a ‘stratification’ paradigm based on such ‘plane at infinity’ constructs. The numerical conditioning of autocalibration is historically delicate. Although recent algorithms have improved the situation significantly [9, 15, 22], classical autocalibration still has some restrictive intrinsic degeneracies — classes of motion for which no algorithm can recover a full unique solution. Sturm [18, 19] gives a catalogue of these. At least 3 views, some translation and some rotation about at least two non-aligned axes are required.

Planar Autocalibration: Existing approaches to classical autocalibration rely on information equivalent to a 3D projective scene reconstruction. In the Kruppa approach this is the fundamental matrices and epipoles, while for most other methods it is an explicit 3D reconstruction. This is a problem whenever planar or near-planar scenes occur, as projective reconstructions, fundamental matrices, *etc.*, can not be estimated stably in the near-planar case. In contrast, *calibrated* reconstruction from near-planar scenes is not difficult, so it is exactly here where projective methods fail, where autocalibration would be most useful. The current paper aims to rectify this, by providing methods which autocalibrate planar scenes by ‘straightening’ the inter-image homographies induced by the plane. A longer term goal is to find ways around the ill-conditioning of projective reconstruction for near-planar scenes, and also to develop ‘structure-free’ internal calibration methods similar to Hartley’s rotating camera one [6], but which work for non-zero translations. The hope is that planar methods may offer one way to attack these problems.

Planar autocalibration has other potential advantages. Planes are very common in man-made environments, and often easily identifiable and rather accurately planar. They are simple to process and allow very reliable and precise feature-based or intensity-based matching, by fitting the homographies between image pairs. They are also naturally well adapted to the calibration of lens distortion, as some of the subtleties of 3D geometry are avoided¹.

The main *disadvantage* of planar autocalibration (besides the need for a nice, flat, textured plane) seems to be the number of images required. Generically, $m \geq \lceil \frac{n+4}{2} \rceil$ images are needed for an internal camera model with n free parameters, *e.g.* $m \geq 5$ for the classical 5 parameter projective model (focal length f , aspect ratio a , skew s , principal point (u_0, v_0)), or $m \geq 3$ if only focal length is

¹ We will ignore lens distortion throughout this paper. If necessary it can be corrected by a nominal model or — at least in theory — estimated up to an overall 3×3 projectivity by a bundled adjustment over all the inter-image homographies.

estimated. However for good accuracy and reliability, at least 8–10 images are recommended in practice. Almost any attempt at algebraic elimination across so many images rapidly leads to a combinatorial explosion. Hence, the approach is resolutely numerical, and it seems impracticable to initialize the optimization from a minimal algebraic solution. Although in many cases the numerical domain of convergence is sufficient to allow moderately reliable convergence from a fixed default initialization, and we have also developed several initialization search methods which may be useful in some cases, occasional convergence to false minima remains a problem.

Organization: Section 2 gives a direction-vector based formulation of auto-calibration, and discusses how planar and non-planar autocalibration can be approached within this framework. Section 3 describes the statistically-motivated cost function we optimize. Section 4 discusses the numerical algorithm, and the method used to initialize it. Section 5 gives experimental results on synthetic and real images, and section 6 concludes the paper.

Notation will be introduced as required. Briefly we use bold upright \mathbf{x}, \mathbf{M} for homogeneous 3D (4 component) vectors and matrices; bold italic \mathbf{x}, \mathbf{M} for 3 component ones (homogeneous image, inhomogeneous 3D, 3-component parts of homogeneous 4-component objects); \mathbf{P} for image projections and \mathbf{H} for inter-image homographies; $\mathbf{K}, \mathbf{C} = \mathbf{K}^{-1}$ for upper triangular camera calibration and inverse calibration matrices; $\mathbf{\Omega}^*$ and $\mathbf{\Omega}$ for the absolute conic and dual (hyperplane) quadric; and $\boldsymbol{\omega}^{-1} = \mathbf{C}^T \mathbf{C}$, $\boldsymbol{\omega} = \mathbf{K} \mathbf{K}^T = \mathbf{P} \mathbf{\Omega} \mathbf{P}^T$ for their images.

2 Euclidean Structure and Autocalibration

To recover the metric information implicit in projective images, we need a projective encoding of Euclidean structure. The key to this is the dot product between direction vectors (“points at infinity”), or dually the dot product between hyperplane normals. The former leads to the stratified “plane at infinity + absolute conic” (affine + metric structure) formulation [17], the latter to the “absolute dual quadric” one [22]. These are just dual ways of saying the same thing. The dual quadric formalism is preferable for ‘pure’ autocalibration where there is no prior decomposition into affine and metric strata, while the point one may be simpler if such a stratification is given.

Generalities: Consider k -dimensional Euclidean space. We will need the cases $k = 2$ (the planar scene and its 2D images) and $k = 3$ (ordinary 3D space). Introducing homogeneous Euclidean coordinates, points, displacement vectors and hyperplanes are encoded respectively as homogeneous $k + 1$ component column vectors $\mathbf{x} = (\mathbf{x}, 1)^T$, $\mathbf{t} = (\mathbf{t}, 0)^T$ and row vectors $\mathbf{p} = (\mathbf{n}, d)$. Here \mathbf{x} , \mathbf{t} and \mathbf{n} are the usual k -D coordinate vectors of the point, the displacement, and the hyperplane normal, and d is the hyperplane offset. Points and displacements on the plane satisfy respectively $\mathbf{p} \cdot \mathbf{x} = \mathbf{n} \cdot \mathbf{x} + d = 0$ and $\mathbf{p} \cdot \mathbf{t} = \mathbf{n} \cdot \mathbf{t} = 0$. Displacement directions can be appended to the point space, as a **hyperplane at infinity** \mathbf{p}_∞ of **vanishing points**. Projectively, \mathbf{p}_∞ behaves much like any other hyperplane. In Euclidean coordinates, $\mathbf{p}_\infty = (0, 1)$ so that $\mathbf{p}_\infty \cdot \mathbf{t} = 0$ for any

displacement $\mathbf{t} = (t, 0)$. **Projective transformations** mix finite and infinite points. Under a projective transformation encoded by an arbitrary nonsingular $(k+1) \times (k+1)$ matrix \mathbf{T} , points and directions (column vectors) transform **contravariantly**, *i.e.* by \mathbf{T} acting on the left: $\mathbf{x} \rightarrow \mathbf{T}\mathbf{x}$, $\mathbf{v} \rightarrow \mathbf{T}\mathbf{v}$. To preserve the point-on-plane relation $\mathbf{p} \cdot \mathbf{x} = \mathbf{n} \cdot \mathbf{x} + d = 0$, hyperplanes (row vectors) transform **covariantly**, *i.e.* by \mathbf{T}^{-1} acting on the right: $\mathbf{p} \rightarrow \mathbf{p}\mathbf{T}^{-1}$.

Absolute Conic & Dual Quadric: The Euclidean dot product between hyperplane normals is $\mathbf{n}_1 \cdot \mathbf{n}_2 = \mathbf{p}_1 \boldsymbol{\Omega} \mathbf{p}_2^\top$ where the symmetric, rank k , positive semidefinite matrix $\boldsymbol{\Omega} = \begin{pmatrix} \mathbf{I}_{k \times k} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix}$ is called the **absolute dual quadric**. $\boldsymbol{\Omega}$ encodes the Euclidean structure in projective coordinates. Under projective transformations it transforms contravariantly (*i.e.* like a point) in each of its two indices so that the dot product between plane normals is invariant: $\boldsymbol{\Omega} \rightarrow \mathbf{T} \boldsymbol{\Omega} \mathbf{T}^\top$ and $\mathbf{p}_i \rightarrow \mathbf{p}_i \mathbf{T}^{-1}$, so $\mathbf{p}_i \boldsymbol{\Omega} \mathbf{p}_i^\top = \mathbf{n}_i \cdot \mathbf{n}_i$ is constant. $\boldsymbol{\Omega}$ is invariant under Euclidean transformations, but in a general projective frame it loses its diagonal form and becomes an arbitrary symmetric positive semidefinite rank k matrix. In any frame, the Euclidean angle between two hyperplanes is $\cos \theta = (\mathbf{p} \boldsymbol{\Omega} \mathbf{p}'^\top) / \sqrt{(\mathbf{p} \boldsymbol{\Omega} \mathbf{p}^\top)(\mathbf{p}' \boldsymbol{\Omega} \mathbf{p}'^\top)}$, and the plane at infinity is $\boldsymbol{\Omega}$'s unique null vector: $\mathbf{p}_\infty \boldsymbol{\Omega} = \mathbf{0}$. When restricted to coordinates on \mathbf{p}_∞ , $\boldsymbol{\Omega}$ becomes nonsingular and can be dualized (inverted) to give the $k \times k$ symmetric positive definite **absolute (direction) conic** $\boldsymbol{\Omega}^*$. This measures dot products between displacement vectors, just as $\boldsymbol{\Omega}$ measures them between hyperplane normals. $\boldsymbol{\Omega}^*$ is defined *only* on direction vectors, not on finite points, and unlike $\boldsymbol{\Omega}$ it has no unique canonical form in terms of the *unrestricted* coordinates.

Direction bases: In Euclidean coordinates, $\boldsymbol{\Omega}$ can be decomposed as a sum of outer products of any orthonormal (in terms of $\boldsymbol{\Omega}^*$) basis of displacement vectors: $\boldsymbol{\Omega} = \sum_{i=1}^k \mathbf{x}_i \mathbf{x}_i^\top$ where $\mathbf{x}_i \boldsymbol{\Omega}^* \mathbf{x}_j = \delta_{ij}$. For example in 2D, $\boldsymbol{\Omega} = \begin{pmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix} = \hat{\mathbf{x}} \hat{\mathbf{x}}^\top + \hat{\mathbf{y}} \hat{\mathbf{y}}^\top$ where $\hat{\mathbf{x}} = (1, 0, 0)^\top$, $\hat{\mathbf{y}} = (0, 1, 0)^\top$, are the usual unit direction vectors. Gathering the basis vectors into the columns of a $(k+1) \times k$ orthonormal rank k matrix \mathbf{U} we have $\boldsymbol{\Omega} = \mathbf{U} \mathbf{U}^\top$, $\mathbf{p}_\infty \mathbf{U} = \mathbf{0}$ and $\mathbf{U}^\top \boldsymbol{\Omega}^* \mathbf{U} = \mathbf{I}_{k \times k}$. The columns of \mathbf{U} span \mathbf{p}_∞ . These relations remain valid in an arbitrary projective frame \mathbf{T} and with an arbitrary choice of representative for $\boldsymbol{\Omega}^*$, except that $\mathbf{U} \rightarrow \mathbf{T} \mathbf{U}$ ceases to be orthonormal.

\mathbf{U} is defined only up to an arbitrary $k \times k$ orthogonal mixing of its columns (redefinition of the direction basis) $\mathbf{U} \rightarrow \mathbf{U} \mathbf{R}_{k \times k}$. In a Euclidean frame $\mathbf{U} = \begin{pmatrix} \mathbf{V} \\ \mathbf{0} \end{pmatrix}$ for some $k \times k$ rotation matrix \mathbf{V} , so the effect of a Euclidean space transformation is $\mathbf{U} \rightarrow \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{U} = \mathbf{U} \mathbf{R}'$ where $\mathbf{R}' = \mathbf{V}^\top \mathbf{R} \mathbf{V}$ is the conjugate rotation: Euclidean transformations of direction bases (*i.e.* on the left) are equivalent to orthogonal re-mixings of them (*i.e.* on the right). This remains true in an arbitrary projective frame, even though \mathbf{U} and the space transformation no longer *look* Euclidean. Given a projective frame, this mixing freedom can be used to choose an associated direction basis in which the columns of \mathbf{U} are numerically orthonormal up to a diagonal rescaling: simply take the SVD $\mathbf{U}' \mathbf{D} \mathbf{V}^\top$ of \mathbf{U} and discard the mixing rotation \mathbf{V}^\top . Equivalently, the eigenvectors and square roots

of eigenvalues of Ω can be used. Such orthogonal parametrizations of \mathbf{U} make good sense numerically, and we will use them below.

Circular points: Given any two orthonormal direction vectors \mathbf{x}, \mathbf{y} , the complex conjugate vectors $\mathbf{x}_{\pm} \equiv \frac{1}{\sqrt{2}}(\mathbf{x} \pm i\mathbf{y})$ satisfy $\mathbf{x}_{\pm}^{\top} \Omega^* \mathbf{x}_{\pm} = 0$. Abstractly, these complex directions lie on the absolute conic. Conversely, given any complex projective point lying on Ω^* , its real and imaginary parts are two orthogonal direction vectors. In the 2D case there is only one such conjugate pair up to complex phase, and these **circular points** characterize the Euclidean structure of the plane. The phase freedom in \mathbf{x}_{\pm} corresponds to the 2×2 orthogonal mixing freedom of \mathbf{x} and \mathbf{y} . In the implementation we prefer to avoid complex numbers by using \mathbf{x} and \mathbf{y} rather than \mathbf{x}_{\pm} .

The above parametrizations of Euclidean structure are theoretically equivalent. Which is practically best depends on the problem. Ω is easy to use, except that constrained optimization is required to handle the rank k constraint $\det \Omega = 0$. Direction bases \mathbf{U} eliminate the constraint at the cost of a $k \times k$ orthogonal gauge freedom. The absolute conic Ω^* has neither constraint nor gauge freedom, but has significantly more complicated image projection properties and can only be defined once the plane at infinity \mathbf{p}_{∞} is known and a projective coordinate system on it has been chosen (*e.g.* induced from one of the images). One can also use Cholesky-like decompositions $\Omega = \mathbf{L} \mathbf{L}^{\top}$ (*e.g.* the \mathbf{L} part of the LQ decomposition of \mathbf{U}), although pivoting is needed to avoid singularities at maximally non-Euclidean frames.

Image Projections: Since the columns of a 3D direction basis \mathbf{U} are *bona fide* 3D direction vectors, its image projection is simply $\mathbf{P} \mathbf{U}$, where \mathbf{P} is the usual 3×4 point projection matrix. Hence, the projection of $\Omega = \mathbf{U} \mathbf{U}^{\top}$ is the 3×3 symmetric positive definite contravariant image matrix $\omega = \mathbf{P} \Omega \mathbf{P}^{\top}$. Abstractly, this is the image line quadric dual to the image of the absolute conic. Concretely, given any two image lines l_1, l_2 , ω encodes the 3D dot product between their 3D visual planes $\mathbf{p}_i \equiv l_i \mathbf{P}$: $\mathbf{p}_1 \Omega \mathbf{p}_2^{\top} = l_1 \mathbf{P} \Omega \mathbf{P}^{\top} l_2^{\top} = l_1 \omega l_2^{\top}$. With the traditional Euclidean decomposition $\mathbf{K} \mathbf{R} (\mathbf{I} | -\mathbf{t})$ of \mathbf{P} into an upper triangular **internal calibration matrix** \mathbf{K} , a 3×3 **camera orientation** (rotation) \mathbf{R} and an **optical centre** \mathbf{t} , ω becomes simply $\mathbf{K} \mathbf{K}^{\top}$. ω is invariant under camera displacements so long as \mathbf{K} remains constant. \mathbf{K} can be recovered from ω by Cholesky decomposition. Similarly, the Euclidean scene structure (in the form of a ‘rectifying’ projectivity) can be recovered from Ω . The upper triangular **inverse calibration matrix** $\mathbf{C} = \mathbf{K}^{-1}$ converts homogeneous pixel coordinates to optical ray directions in the Euclidean camera frame. $\omega^{-1} = \mathbf{C}^{\top} \mathbf{C}$ is the image of the absolute conic.

Autocalibration: Given several images taken with projection matrices $\mathbf{P}_i = \mathbf{K}_i \mathbf{R}_i (\mathbf{I} | -\mathbf{t}_i)$, and (in the same Euclidean frame) a orthogonal direction basis $\mathbf{U} = \begin{pmatrix} \mathbf{V} \\ \mathbf{o} \end{pmatrix}$, we find that

$$\mathbf{C}_i \mathbf{P}_i \mathbf{U} = \mathbf{R}'_i \quad (1)$$

where $\mathbf{C}_i = \mathbf{K}_i^{-1}$ and $\mathbf{R}'_i = \mathbf{R}_i \mathbf{V}$ is a rotation matrix depending on the camera pose. This is perhaps the most basic form of autocalibration constraint. It

says that the calibrated images (*i.e.* 3D directions in the camera frame) of an orthogonal direction basis must remain orthogonal. It remains true in arbitrary projective 3D and image frames, as the projective deformations of \mathbf{P}_i *vs.* \mathbf{U} and \mathbf{C}_i *vs.* \mathbf{P}_i cancel each other out. Unfortunately, for projectively reconstructed projection matrices it is seldom possible to estimate scale factors consistent with those of their unknown Euclidean parents. So in practice the autocalibration constraint can only be applied up to an unknown scale factor: $\mathbf{C}_i \mathbf{P}_i \mathbf{U} \sim \mathbf{R}_i'$. As always, the direction basis \mathbf{U} is only defined up to an arbitrary 3×3 orthogonal mixing $\mathbf{U} \rightarrow \mathbf{U} \mathbf{R}$.

2.1 Autocalibration for Non-Planar Scenes

The simplest approaches to non-planar autocalibration are based on (1), an intermediate projective reconstruction \mathbf{P}_i , and some sort of knowledge about the \mathbf{C}_i (classically that they are constant, $\mathbf{C}_i = \mathbf{C}$ for some unknown \mathbf{C}). Non-linear optimization or algebraic elimination are used to estimate the Euclidean structure $\mathbf{\Omega}$ or \mathbf{U} , and the free parameters of the \mathbf{C}_i . Multiplying (1) either on the left or on the right by its transpose to eliminate the unknown rotation, and optionally moving the \mathbf{C} 's to the right hand side, gives several equivalent symmetric 3×3 constraints linking $\mathbf{\Omega}$ or \mathbf{U} to ω_i , \mathbf{K}_i or \mathbf{C}_i

$$\mathbf{U}^\top \mathbf{P}_i^\top \omega_i^{-1} \mathbf{P}_i \mathbf{U} \sim \mathbf{I}_{3 \times 3} \quad (2)$$

$$\mathbf{C}_i \mathbf{P}_i \mathbf{\Omega} \mathbf{P}_i^\top \mathbf{C}_i^\top \sim \mathbf{I}_{3 \times 3} \quad (3)$$

$$\mathbf{P}_i \mathbf{\Omega} \mathbf{P}_i^\top \sim \omega_i = \mathbf{K}_i \mathbf{K}_i^\top \quad (4)$$

In each case there are 5 independent constraints per image on the 8 non-Euclidean d.o.f. of the 3D projective structure² and the ≤ 5 d.o.f. per image of the internal calibrations \mathbf{C}_i . For example, three images in general position suffice for classical constant- \mathbf{C} autocalibration. In each case, the unknown scale factors can be eliminated by treating the symmetric 3×3 left and right hand side matrices as $\frac{3 \cdot 4}{2} = 6$ component vectors, and either (i) cross-multiplying, or (ii) projecting (say) the left hand side vectors orthogonally to the right hand ones (hence deleting the proportional components and focusing on the constraint-violating non-proportional ones). Cross-multiplication gives

$$\mathbf{u}_i \cdot \mathbf{v}_i = \mathbf{u}_i \cdot \mathbf{w}_i = \mathbf{v}_i \cdot \mathbf{w}_i = 0 \quad \|\mathbf{u}_i\|^2 = \|\mathbf{v}_i\|^2 = \|\mathbf{w}_i\|^2 \quad (5)$$

$$\begin{aligned} (\mathbf{C}_i \mathbf{P}_i \mathbf{\Omega} \mathbf{P}_i^\top \mathbf{C}_i^\top)^{AA} &= (\mathbf{C}_i \mathbf{P}_i \mathbf{\Omega} \mathbf{P}_i^\top \mathbf{C}_i^\top)^{BB} \\ (\mathbf{C}_i \mathbf{P}_i \mathbf{\Omega} \mathbf{P}_i^\top \mathbf{C}_i^\top)^{AB} &= 0 \end{aligned} \quad (6)$$

² These can be counted as follows: 15 for a 3D projective transformation modulo 7 for a scaled Euclidean one; or 12 for a 4×3 \mathbf{U} matrix modulo 1 scale and 3 d.o.f. for a 3×3 orthogonal mixing; or $4 \cdot 5/2 = 10$ d.o.f. for a 4×4 symmetric quadric matrix $\mathbf{\Omega}$ modulo 1 scale and 1 d.o.f. for the rank 3 constraint $\det \mathbf{\Omega} = 0$; or 3 d.o.f. for \mathbf{p}_∞ and 5 for the $3 \cdot 4/2 = 6$ components of $\mathbf{\Omega}^*$ modulo 1 scale.

$$(\mathbf{P}_i \boldsymbol{\Omega} \mathbf{P}_i^\top)^{CD} (\boldsymbol{\omega})^{EF} = (\boldsymbol{\omega})^{CD} (\mathbf{P}_i \boldsymbol{\Omega} \mathbf{P}_i^\top)^{EF} \quad (7)$$

where $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i) \equiv \mathbf{C}_i \mathbf{P}_i \mathbf{U}$ and $A < B, C \leq D, E \leq F = 1 \dots 3$. Several recent autocalibration methods (*e.g.* [22, 9]) are based implicitly on these constraints, parametrized by \mathbf{K} or $\boldsymbol{\omega}$ and something equivalent³ to $\boldsymbol{\Omega}$ or \mathbf{U} . All of these methods seem to work well provided the intrinsic degeneracies of the autocalibration problem [18] are avoided. In contrast, methods based on the Kruppa equations [14, 3, 25] are not recommended for general use, because they add a serious additional singularity to the already-restrictive ones intrinsic to the problem: if any 3D point projects to the same pixel and is viewed from the same distance in each image, one focal length parameter can not be recovered [19].

2.2 Autocalibration from Planar Scenes

Now consider autocalibration from *planar* scenes. Everything above remains valid, except that no intermediate 3D projective reconstruction is available from which to bootstrap the process. However autocalibration is still possible using the inter-image homographies. The Euclidean structure of the scene plane is given by any one of (i) a 3×3 rank 2 absolute dual (line) quadric \mathbf{Q} ; (ii) a 3 component line at infinity \mathbf{l}_∞ and its associated 2×2 absolute (direction) conic matrix; (iii) a 3×2 direction basis matrix $\mathbf{U} = (\mathbf{x} \ \mathbf{y})$; (iv) two complex conjugate circular points $\mathbf{x}_\pm = \frac{1}{\sqrt{2}}(\mathbf{x} \pm i\mathbf{y})$ which are also the two roots of the absolute conic on \mathbf{l}_∞ and the factors of the absolute dual quadric $\mathbf{Q} = \mathbf{x} \mathbf{x}^\top + \mathbf{y} \mathbf{y}^\top = \mathbf{x}_+ \mathbf{x}_+^\top + \mathbf{x}_- \mathbf{x}_-^\top$. In each case the structure is the natural restriction of the corresponding 3D one, re-expressed in the planar coordinate system. In each case it projects isomorphically into each image, either by the usual 3×4 3D projection matrix (using 3D coordinates), or by the corresponding 3×3 world-plane to image-plane homography \mathbf{H} (using scene plane coordinates). Hence, each image inherits a pair of circular points $\mathbf{H}_i \mathbf{x}_\pm$ and the corresponding direction basis $\mathbf{H}_i (\mathbf{x} \ \mathbf{y})$, line at infinity $\mathbf{l}_\infty \mathbf{H}_i^{-1}$ and 3×3 rank 2 absolute dual quadric $\mathbf{H}_i \mathbf{Q} \mathbf{H}_i^\top$. The columns of the planar \mathbf{U} matrix still represent *bona fide* 3D direction vectors (albeit expressed in planar coordinates), so their images still satisfy the autocalibration constraints (1):

$$\mathbf{C}_i \mathbf{H}_i \mathbf{U} \sim \mathbf{R}_{3 \times 2} \quad (8)$$

where $\mathbf{R}_{3 \times 2}$ contains the first two columns of a 3×3 rotation matrix. Left multiplication by the transpose eliminates the unknown rotation (*c.f.* (2)):

$$\mathbf{U}^\top \mathbf{H}_i^\top \boldsymbol{\omega}_i^{-1} \mathbf{H}_i \mathbf{U} \sim \mathbf{I}_{2 \times 2} \quad (9)$$

Splitting this into components gives the form of the constraints used by our planar autocalibration algorithm:

$$\|\mathbf{u}_i\|^2 = \|\mathbf{v}_i\|^2, \quad \mathbf{u}_i \cdot \mathbf{v}_i = 0 \quad \text{where} \quad (\mathbf{u}_i, \mathbf{v}_i) \equiv \mathbf{C}_i \mathbf{H}_i (\mathbf{x}, \mathbf{y}) \quad (10)$$

³ If the first camera projection is taken to be $(\mathbf{I} | \boldsymbol{\theta})$ [5, 9], \mathbf{U} can be chosen to have the form $\begin{pmatrix} \mathbf{I} \\ -\mathbf{p}^\top \end{pmatrix} \mathbf{K}$ where $\mathbf{p}_\infty \sim (\mathbf{p}^\top, 1)$, whence $\boldsymbol{\Omega} \sim \begin{pmatrix} \boldsymbol{\omega} & -\boldsymbol{\omega} \mathbf{p} \\ -\mathbf{p}^\top \boldsymbol{\omega} & \mathbf{p}^\top \boldsymbol{\omega} \mathbf{p} \end{pmatrix}$ and $\begin{pmatrix} \mathbf{C} & \boldsymbol{\theta} \\ \mathbf{p}^\top & 1 \end{pmatrix}$ is a Euclideanizing projectivity.

These constraints say that any two orthonormal directions in the world plane project (via $\mathbf{C}_i \mathbf{H}_i$) to orthonormal directions in the calibrated camera frame. Equivalently, the images of the circular points $\mathbf{x}_{\pm} = \frac{1}{\sqrt{2}}(\mathbf{x} \pm i\mathbf{y})$ lie on the image of the absolute conic:

$$(\mathbf{H}_i \mathbf{x}_{\pm})^{\top} \boldsymbol{\omega}^{-1} (\mathbf{H}_i \mathbf{x}_{\pm}) = \|\mathbf{u}_{i\pm}\|^2 = 0 \quad \text{where} \quad \mathbf{u}_{i\pm} \equiv \mathbf{C}_i \mathbf{H}_i \mathbf{x}_{\pm} \quad (11)$$

All of the above constraints are valid in arbitrary projective image and world-plane frames, except that $\mathbf{U} = (\mathbf{x} \ \mathbf{y})$ is no longer orthonormal. As always, (\mathbf{x}, \mathbf{y}) are defined only up to a 2×2 orthogonal mixing, and we can use this gauge freedom to require that $\mathbf{x} \cdot \mathbf{y} = 0$.

Our planar autocalibration method is based on direct numerical minimization of the residual error in the constraints (10) from several images, over the unknown direction basis (\mathbf{x}, \mathbf{y}) and any combination of the five intrinsic calibration parameters f, a, s, u_0 and v_0 . The input data is the set of world plane to image homographies \mathbf{H}_i , expressed with respect to an arbitrary projective frame for the world plane. In particular, if the plane is coordinatized by its projection into some key image (say image 1), the inter-image homographies \mathbf{H}_{i1} can be used as input.

Four independent parameters are required to specify the Euclidean structure of a projective plane: the 6 components of (\mathbf{x}, \mathbf{y}) modulo scale and the phase of a 2×2 rotation; or the $3 \cdot 4/2 = 6$ components of a 3×3 absolute dual quadric \mathbf{Q} modulo scale and $\det \mathbf{Q} = 0$; or the 2 d.o.f. of the plane's line at infinity, plus the 2 d.o.f. of two circular points on it. Since equations (9), (10) or (11) give two independent constraints for each image, $\lceil \frac{n+4}{2} \rceil$ images are required to estimate the Euclidean structure of the plane and n intrinsic calibration parameters. For known calibration, two images suffice (classical plane-based relative camera orientation). Three are required if the focal length is also estimated, four for the perspective f, u_0, v_0 model, and five if all 5 intrinsic parameters are unknown.

2.3 Camera Parametrization

We have not yet made the camera parametrization explicit, beyond saying that it is given by the upper triangular matrices \mathbf{K} or $\mathbf{C} = \mathbf{K}^{-1}$. For autocalibration methods which fix some parameters while varying others, it makes a difference which parametrization is used. I prefer the following form motivated by a zoom lens followed by an affine image-plane coordinatization:

$$\mathbf{K} = \begin{pmatrix} f & fs & u_0 \\ 0 & fa & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{C} = \mathbf{K}^{-1} = \frac{1}{fa} \begin{pmatrix} a & -s & sv_0 - au_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & fa \end{pmatrix}$$

Here, if standard pixel coordinates are used, $f = \alpha_u$ is the focal length in u -pixels, $s = -\tan \theta_{\text{skew}}$ is the dimensionless geometric skew, $a = \alpha_v / (\alpha_u \cos \theta_{\text{skew}})$ is the dimensionless $v : u$ aspect ratio, and (u_0, v_0) are the pixel coordinates of the principal point. However pixel coordinates are *not* used in the optimization routine

below. Instead, a nominal calibration is used to standardize the parameters to nominal values $f = a = 1$, $s = u_0 = v_0 = 0$, and all subsequent fitting is done using the above model with respect to these values.

3 Algebraic vs. Statistical Error

Many vision problems reduce to minimizing the residual violation of some vector of nonlinear constraints $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu}) \approx \mathbf{0}$ over parameters $\boldsymbol{\mu}$, given noisy measurements \mathbf{x} with known covariance \mathbf{V}_x . Often, heuristic **algebraic error** metrics like $\|\mathbf{e}(\mathbf{x}, \boldsymbol{\mu})\|^2$ are minimized. However this is statistically sub-optimal, and if done uncritically can lead to both (i) very significant bias in the results and (ii) severe constriction of the domain of convergence of the optimization. Appropriate **balancing** or **preconditioning** (numerical scaling of the variables and constraints, as advocated in [7, 8] and any numerical optimization text) is one step towards eliminating such problems, but it is not the whole story. In any case it begs the question of what *is* “balanced”. It is *not* always appropriate to scale all variables to $\mathcal{O}(1)$. In fact, in the context of parameter estimation, balanced simply means “close to an underlying *statistical* error metric” such as $\chi_e^2 \approx \mathbf{e}^\top \mathbf{V}_e^{-1} \mathbf{e}$, where $\mathbf{V}_e \approx \frac{\mathbf{D}\mathbf{e}}{\mathbf{D}\mathbf{x}} \mathbf{V}_x \frac{\mathbf{D}\mathbf{e}}{\mathbf{D}\mathbf{x}}^\top$ is the covariance⁴ of \mathbf{e} .

Ideally one would like to minimize the underlying statistical error, but this can be complicated owing to the matrix products and (pseudo-)inverse. Simple approximations often suffice, and I feel that this is the *only* acceptable way to introduce algebraic error measures — as explicit, controlled approximations to an underlying statistical metric. The extra computation required for a suitable approximation is usually minimal while the results can be substantially more accurate, so it makes little sense to use an arbitrary *ad hoc* error metric.

One useful simplification ignores the dependence of \mathbf{V}_e^{-1} on $\boldsymbol{\mu}$ in cost function derivatives. This gives **self-consistent** or **iterative re-weighting** schemes (e.g. [12]), where \mathbf{V}_e is treated as a constant within each optimization step, but updated at the end of it. One can show that the missing terms effectively displace the cost derivative evaluation point from the measured \mathbf{x} to a first order estimate of the true underlying value \mathbf{x}_0 [20]. For the most part this makes little difference unless the constraints are strongly curved on the scale of \mathbf{V}_x .

However, as I feel that a final bundle adjustment is an essential part of any reconstruction or autocalibration technique, my current implementation uses a rather simplistic error model for the \mathbf{H}_i . Firstly, I ignore the fact that they are correlated through their mutual dependence on the base image, which is treated just like any other in the sum. This undoubtedly introduces some bias. Correcting it would significantly complicate the method, and require users to supply an inconvenient amount of inter- \mathbf{H}_i covariance data. Secondly, the components of

⁴ \mathbf{e} is a random variable through its dependence on \mathbf{x} . Assuming that the uncertainties are small enough to allow linearization and that \mathbf{x} is centred on some underlying \mathbf{x}_0 satisfying $\mathbf{e}(\mathbf{x}_0, \boldsymbol{\mu}_0) = \mathbf{0}$ for underlying parameters $\boldsymbol{\mu}_0$, $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu}_0)$ has mean $\mathbf{0}$ and the above covariance. It follows that $\mathbf{e}^\top \mathbf{V}_e^{-1} \mathbf{e}$ is approximately a $\chi_{\text{rank}(\mathbf{e})}^2$ variable near $\boldsymbol{\mu}_0$, which can be minimized to find a maximum likelihood estimate of $\boldsymbol{\mu}$.

\mathbf{H}_i are assumed to be i.i.d. in the nominally calibrated coordinate system: $\mathbf{V}_H = \langle \Delta \mathbf{H}_B^A \Delta \mathbf{H}_D^C \rangle \sim \epsilon \cdot \delta^{AC} \cdot \delta_{BD}$ where ϵ is a noise level⁵. This could easily be corrected at the cost of a little extra linear algebra.

From here, it is straightforward to find and invert the constraint covariance. For the planar autocalibration constraint (10), and assuming that we enforce the gauge constraint $\mathbf{x} \cdot \mathbf{y} = 0$, the constraint covariance is

$$\mathbf{V}_e \approx \epsilon \cdot \begin{pmatrix} \mathbf{x}^2 \mathbf{a}_i^2 + \mathbf{y}^2 \mathbf{b}_i^2 & (\mathbf{x}^2 - \mathbf{y}^2) \mathbf{a}_i \cdot \mathbf{b}_i \\ (\mathbf{x}^2 - \mathbf{y}^2) \mathbf{a}_i \cdot \mathbf{b}_i & \mathbf{x}^2 \mathbf{b}_i^2 + \mathbf{y}^2 \mathbf{a}_i^2 \end{pmatrix}, \quad \begin{aligned} (\mathbf{a}_i, \mathbf{b}_i) &\equiv \mathbf{C}_i^\top (\mathbf{u}_i, \mathbf{v}_i) \\ &= \omega_i^{-1} \mathbf{H}_i (\mathbf{x}, \mathbf{y}) \end{aligned}$$

In this case, numerical experience indicates that the off-diagonal term is seldom more than a few percent of the diagonal ones, which themselves are approximately equal for each image, but differ by as much as a factor of 2–3 between images⁶. Hence, we drop the off-diagonal term to give an autocalibration method based on self-consistent optimization of the diagonal cost function

$$\sum_{i=1}^m \left(\frac{(\|\mathbf{u}_i\|^2 - \|\mathbf{v}_i\|^2)^2}{\mathbf{x}^2 \|\mathbf{C}_i^\top \mathbf{u}_i\|^2 + \mathbf{y}^2 \|\mathbf{C}_i^\top \mathbf{v}_i\|^2} + \frac{(\mathbf{u}_i \cdot \mathbf{v}_i)^2}{\mathbf{x}^2 \|\mathbf{C}_i^\top \mathbf{v}_i\|^2 + \mathbf{y}^2 \|\mathbf{C}_i^\top \mathbf{u}_i\|^2} \right) \quad (12)$$

where $(\mathbf{u}_i, \mathbf{v}_i) \equiv \mathbf{C}_i \mathbf{H}_i (\mathbf{x}, \mathbf{y})$. In our synthetic experiments, this statistically motivated cost function gives ground-truth standard deviations about 10% lower than even the most carefully normalized algebraic ones. This is a modest but useful improvement, obtained without any measurable increase in run time. The improvement would have been *much* larger had the error model been less uniform in the standardized coordinates. Perhaps most importantly, the statistical cost is almost completely immune to mis-scaling of the variables. This is certainly not true of the algebraic ones, which deteriorate very rapidly for mis-scaling factors greater than about 3.

4 Planar Autocalibration Algorithm

Numerical Method: Our planar autocalibration algorithm is based on direct numerical minimization of the m -image cost function (12), with respect to the direction basis (\mathbf{x}, \mathbf{y}) and any subset of the 5 internal calibration parameters focal length f , aspect ratio a , skew s , and principal point (u_0, v_0) . There are 4 d.o.f. in (\mathbf{x}, \mathbf{y}) — 6 components defined up to an overall rescaling and a 2×2

⁵ Although balancing should make their variances similar, in reality the components are unlikely to be independent. At very least we should subtract a ‘scale’ term $\mathbf{H}_B^A \mathbf{H}_D^C$, as variations proportional to \mathbf{H} make no projective difference. However this makes no difference here. A correctly weighted error metric \mathbf{e} must be insensitive to rescalings of its projective-homogeneous parameters \mathbf{H} . Hence it has homogeneity 0 in \mathbf{H} — $\frac{\mathbf{D}\mathbf{e}}{\mathbf{D}\mathbf{H}} \cdot \mathbf{H} = 0$ — and \mathbf{V}_e is unaffected by the ‘scale’ term.

⁶ This was to be expected: we chose everything to be well-scaled except that the \mathbf{H} normalizations may differ somewhat from their ‘correct’ Euclidean ones, and our noise model is uniform in an approximately calibrated frame. Relaxing any of these conditions would make the differences *much* greater.

orthogonal mixing — so the optimization is over 5–9 parameters in all. Numerically, the 6 component (\mathbf{x}, \mathbf{y}) vector is projected onto the subspace orthogonal to its current scaling and mixing d.o.f. by Householder reduction (*i.e.* effectively a mini QR decomposition). As mentioned in section 2, the mixing freedom allows us to enforce the gauge condition $\mathbf{x} \cdot \mathbf{y} = 0$. Although not essential, this costs very little (one Jacobi rotation) and we do it at each iteration as an aid to numerical stability.

A fairly conventional nonlinear least squares optimization method is used: Gauss-Newton iteration based on Cholesky decomposition of the normal equations. As always, forming the normal equations gives a fast, relatively simple method but effectively squares the condition number of the constraint Jacobian. This is not a problem so long as intermediate results are stored at sufficiently high precision: double precision proves more than adequate here.

As with any numerical method, care is needed to ensure stability should the numerical conditioning become poor. Our parametrization guarantees that all variables are of $\mathcal{O}(1)$ and fairly well decoupled, so preconditioning is not necessary. The Cholesky routine uses diagonal pivoting and Gill & Murray’s [4] minimum-diagonal-value regularization to provide local stability. The regularizer is manipulated in much the same way as a Levenberg-Marquardt parameter to ensure that each step actually reduces the cost function. We also limit the maximum step size for each variable, relatively for the positive, multiplicative parameters f and a and absolutely for the others. Both the regularizer and the step size limits are activated fairly often in practice, the regularizer at any time, and the step limit usually only during the first 1–2 iterations. The method terminates when the step size converges to zero, with additional heuristics to detect thrashing. Convergence within 5–10 iterations is typical.

Prior over Calibrations: We also allow for a simple user-defined prior distribution on the calibration parameters. Even if there is no strong prior knowledge, it is often advisable to include a weak prior in statistical estimation problems as a form of regularization. If there are nearly unobservable parameter combinations (*i.e.* which make little or no difference to the fit), optimal, unbiased estimates of these are usually extremely sensitive to noise. A weak prior has little influence on strong estimates, but significantly reduces the variability of weak ones by biasing them towards reasonable default values. A desire to keep the results unbiased is understandable, but limiting the impact of large fluctuations on the rest of the vision system is often more important in practice.

Priors can also be used to ensure that parameters retain physically meaningful values. For example, the multiplicative parameters f and a must lie in the range $(0, \infty)$, and we include weak heuristic priors of the form $(f/f_0 - f_0/f)^2$ (*i.e.* multiplicatively symmetric about f_0) to ensure this. This is particularly important for autocalibration problems, where degenerate motions occur frequently. In such cases the calibration can not be recovered uniquely and a numerical method will converge to an arbitrary member of the family of possible solutions. For sanity, it pays to ensure that this is a physically feasible solution not too far from the plausible range of values. A weak default prior is an effec-

tive means of achieving this, and seems no more unprincipled than any other method. (This is not to say that degeneracies should be left unflagged, simply that recovering from them is easier if it starts from reasonable default values).

Initialization: The domain of convergence of the numerical method is reasonable and for many applications it will probably be sufficient to initialize from fixed default values. The most critical parameters are the focal length f and the number and angular spread of the views. For example, if f can only be guessed within a factor of 2 and all 5 parameters f, a, s, u_0, v_0 are left free, about 9–10 images spread by more than about 10° seem to be required for reliable convergence to the true solution. Indeed, with 5 free parameters and the theoretical minimum of only 5–6 images, even an *exact* initialization is not always sufficient to eliminate false solutions (*i.e.* whose residuals happen to be slightly smaller than the true one). These figures assume that the direction basis \mathbf{x}, \mathbf{y} is completely unknown: constraints on it are potentially very valuable and should be used if available. Knowledge of the world-plane’s horizon (line at infinity) removes 2 d.o.f. from \mathbf{x}, \mathbf{y} and hence reduces the number of images required by one, and knowledge of its Euclidean structure (but not the positions of points on it) eliminates another image. Even if not directly visible, horizons can be recovered from known 3D parallelism or texture gradients, or bounded by the fact that visible points on the plane must lie inside them. We will not consider such constraints further here.

If a default initialization is insufficient to guarantee convergence, several strategies are possible. One quite effective technique is simply to use a preliminary optimization over \mathbf{x}, \mathbf{y} or $\mathbf{x}, \mathbf{y}, f$ to initialize a full one over all parameters. More global searches over $f, \mathbf{x}, \mathbf{y}$ are also useful. Perhaps the easiest way to approach this is to fix nominal values for all of the calibration parameters except f , and to recover estimates for \mathbf{x}, \mathbf{y} as a function of f from a single pair of images as f varies. These values can then be substituted into the autocalibration constraints for the other images, and the overall most consistent set of values chosen to initialize the optimization routine. The estimation of $\mathbf{x}(f), \mathbf{y}(f)$ reduces to the classical photogrammetric problem of the relative orientation of two calibrated cameras from a planar scene, as the Euclidean structure is easily recovered once the camera poses are known. In theory this problem could be solved in closed form (the most difficult step being a 3×3 eigendecomposition) and optimized over f analytically. But in practice this would be rather messy and I have preferred to implement a coarse numerical search over f . The search uses a new SVD-based planar relative orientation method [21] related to Wunderlich’s eigendecomposition approach [24]. The camera pose and planar structure are recovered directly from the SVD of the inter-image homography. As always with planar relative orientation, there is a two-fold ambiguity in the solution, so both solutions are tested. In the implemented routine, the solutions for each image against the first one, and for each f in a geometric progression, are substituted into the constraints from all the other images, and the most consistent overall values are chosen.

If the full 5 parameter camera model is to be fitted, Hartley’s ‘rotating camera’ method [6] can also be used for initialization. It works well *provided* (i) the camera translations are smaller than or comparable to the distance to the plane; (ii) no point on the plane is nearly fixated from a constant distance. (For such a point \mathbf{x} , $\boldsymbol{\omega} + \mu \mathbf{x} \mathbf{x}^\top$ is an approximate solution of Hartley’s equation $\mathbf{H} \boldsymbol{\omega} \mathbf{H}^\top = \boldsymbol{\omega}$ for any μ , i.e. $\boldsymbol{\omega}$ can not be estimated uniquely, even for small translations).

5 Experiments

Synthetic data: The method has been implemented in C and tested on both real and synthetic images. For the synthetic experiments, the camera roughly fixates a point on the plane from a constant distance, from randomly generated orientations varying by (by default) $\pm 30^\circ$ in each of the three axes. The camera calibration varies randomly about a nominal focal length of 1024 pixels and unit aspect ratio, by $\pm 30\%$ in focal length f , $\pm 10\%$ in aspect ratio a , ± 0.01 in dimensionless skew s , and ± 50 pixels in principal point (u_0, v_0) . (These values are standard deviations of log-normal distributions for f , a and normal ones for s , u_0 , v_0). The scene plane contains by default 40 visible points, projected into the 512×512 images with a Gaussian noise of ± 1 pixel. Before the homographies are estimated and the method is run, the pixel coordinates are centred and scaled to a nominal focal length of 1: $(u, v) \rightarrow (u - 256, v - 256)/1024$. The output is classed as a ‘success’ or ‘failure’ according to fixed thresholds on the size of its deviation from the true value. Only successes count towards the accuracy estimates. The usual mode of failure is convergence to a false solution with extremely short focal length (say < 50 pixels). However when the angular spread of the views is small or there are only a few images, random fluctuations sometimes take a “correct” but highly variable solution outside the (generously set) thresholds. Conversely, there is occasionally convergence to a false solution within the threshold. Thus, when the failure rate is high, neither it nor the corresponding precision estimates are accurate. The optimization typically converges within 5–10 iterations, although more may be needed for degenerate problems. The run time is negligible: on a Pentium 133, about 0.5 milliseconds per image if the default initialization is used, or 2.0 with a fairly fine initialization search over f .

Figure 1 gives some illustrative accuracy and reliability results, concentrating on the estimation of focal length f . First consider the plots where all 5 calibration parameters are estimated. The error scales roughly linearly with noise and inversely with the angular spread of the views. It drops rapidly as the first few images are added, but levels off after about 10 images. The failure rate increases rapidly for more than about 2–3 pixels noise, and is also unacceptably high for near-minimal numbers of images (within 1–2 of the minimum) and small angular spreads (less than about 10°). However, it decreases rapidly as each of these variables is increased. It seems to be difficult to get much below about 1% failure rate with the current setup. Some of the failures probably result from degeneracies in the randomly generated problems, but most of them are caused by convergence to a false solution with implausible parameters, either very small f

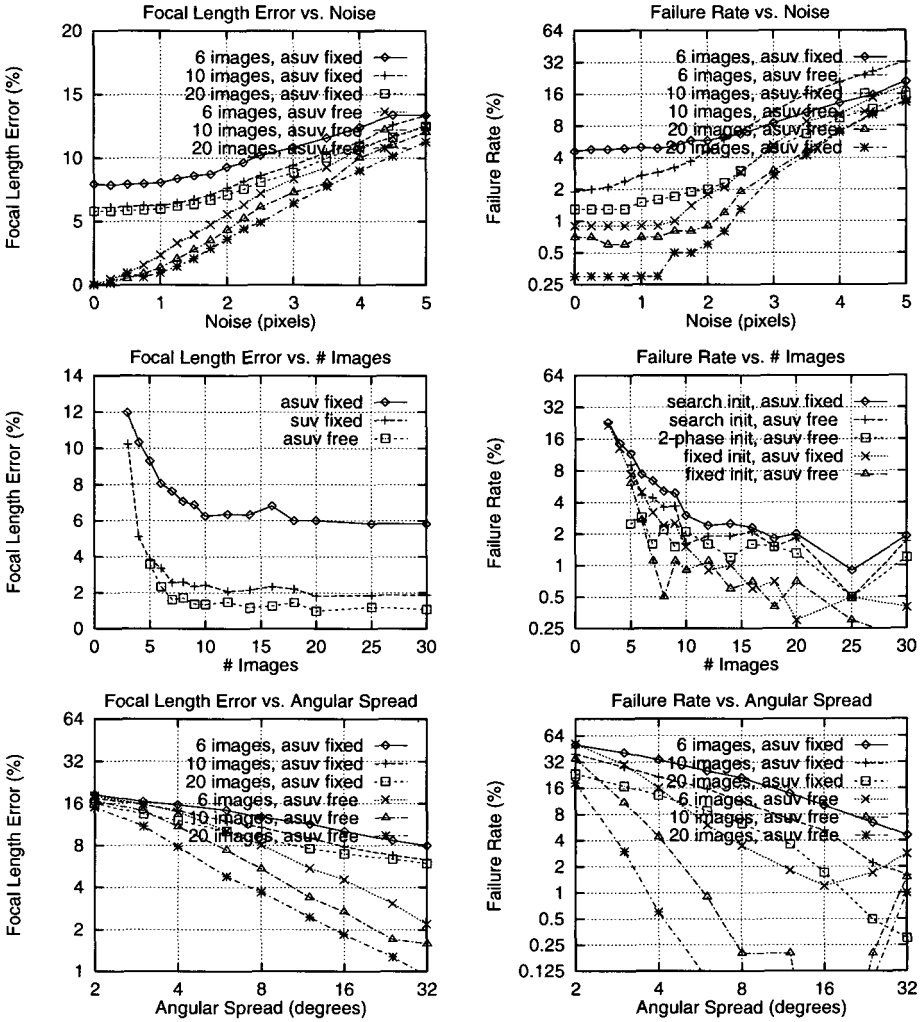


Fig. 1. Focal length error and failure rate vs. image noise, number of images and angular spread of cameras, averaged over 1000 trials. The aspect ratio a , skew s , and principal point (u_0, v_0) are either fixed at their nominal values, or allowed to vary freely, as indicated. The method is initialized from the nominal calibration, except that in the failure vs. images plot we also show the results for initialization by numerical search over f , and by a preliminary fit over f alone ('2-phase').

(less than about 50) or a far from 1. The initialization method has little impact on the reliability. In fact, in these experiments the default initialization proved more reliable than either numerical search over f , or an initial optimization over f alone. The reason is simply that we do not assume prior knowledge of *any* of the calibration parameters. An initialization search over f must fix a, s, u_0, v_0 at

their inaccurate nominal values, and this is sometimes enough to make it miss the true solution entirely. This also explains the poor performance of the methods which hold a, s, u_0, v_0 fixed and estimate f alone. As the graphs of error *vs.* noise and number of images show, errors in a, s, u_0, v_0 lead to a significant bias in f , but most of this can be eliminated by estimating a as well as f . The initialization search over f also becomes much more reliable (*e.g.* 0.05% failure rate for 10 images, 30° spread and 1 pixel noise) if a and s are accurate to within a few percent. Here and elsewhere, it is only worthwhile to fix parameters if they are reliably known to an accuracy better than their measured variabilities, *e.g.* here for 1 pixel noise and 10 images, to about 0.003 for a, s or 20 pixels for u_0, v_0 .

For conventional calibration, f is often said the most difficult parameter to estimate, and also the least likely to be known *a priori*. In contrast, a and s are said to be estimated quite accurately, while u_0 and v_0 — although variable — are felt to have little effect on the overall results. A more critical, quantitative view is to compare the *relative* accuracy $|\Delta f/f|$ to the dimensionless quantities $|\Delta a|$, $|\Delta s|$, $|\Delta u_0/f|$ and $|\Delta v_0/f|$. Errors in these contribute about equally to the overall geometric accuracy (*e.g.* reconstruction errors of 3D visual ray directions). Conversely, other things being equal, geometric constraints such as the autocalibration ones typically constrain each of these quantities to about the same extent. A good rule of thumb is that for auto- (and many other types of) calibration, $|\Delta u_0/f|$ and $|\Delta v_0/f|$ are of the same order of magnitude as $|\Delta f/f|$, while $|\Delta a|$ and $|\Delta s|$ are usually somewhat smaller if there are strong aspect ratio constraints (*e.g.* cyclotorsion), but larger if there are none (*e.g.* motions that leave a direction in the optical plane fixed). These rules are well borne out in all of the experiments reported here: we always find $|\Delta u_0| \approx |\Delta v_0| \approx |\Delta f|$, while $|\Delta a|$ and $|\Delta s|$ are respectively about one fifth, one half, and one tenth of $|\Delta f/f|$ for the synthetic experiments, the real experiments below, and the Faugeras-Toscani calibration used in the real experiments.

Real data: We have also run the method on parts of a sequence of about 40 real images of our tri-planar calibration grid. Only the 49 (at most) points on the base plane of the grid are used. (The algorithm could easily be expanded to handle several planes, but there seems little point as a non-planar method could be used in this case). The motion was intended to be general within the limits of the 5 d.o.f. robot used to produce it, but is fairly uniform within each subsequence. Visibility considerations limited the total angular displacement to about 40°, and significantly less within each subsequence. Here are the sample means and standard deviations over a few non-overlapping subsequences for (i) f alone, and (ii) all 5 parameters. The errors are observed sample scatters, *not* estimates of absolute accuracy.

	f only	f	a	s	u_0	v_0
calibration	-	1515 ± 4	0.9968 ± 0.0002	-	271 ± 3	264 ± 4
6 images	1584 ± 63	1595 ± 63	0.9934 ± 0.0055	0.000 ± 0.001	268 ± 10	271 ± 22
8 images	1619 ± 25	1614 ± 42	0.9890 ± 0.0058	-0.005 ± 0.005	289 ± 3	320 ± 26
10 images	1612 ± 19	1565 ± 41	1.0159 ± 0.0518	-0.004 ± 0.006	273 ± 5	286 ± 27

The ‘calibrated’ values are the averaged results of several single-image Faugeras-Toscanni calibrations using all visible points on the grid. Looking at the table, the results of the autocalibration method seem usable but not quite as good as I would have expected on the basis of the synthetic experiments. This may just be the effect of the small angular range within each subsequence, but the estimates of f seem suspiciously high and it may be that some small systematic error has occurred during the processing. Further work is required to check this. Note that in this case, fixing a, s, u_0, v_0 appears to have the desired effect of decreasing the variability of the estimated f without perturbing its value very much.

6 Summary

In summary, we have shown how autocalibration problems can be approached using a projective representation of orthogonal 3D direction frames, and used this to derive a practical numerical algorithm for the autocalibration of a moving projective camera viewing a planar scene. The method is based on the ‘rectification’ of inter-image homographies. It requires a minimum of 3 images if only the focal length is estimated, or 5 for all five internal parameters. Using 9–10 images significantly increases both reliability and accuracy. An angular spread between the cameras of at least 15–20° is recommended.

Priorities for future work are the initialization problem and the detection of false solutions (or possibly the production of multiple ones). Although the current numerical method is stable even for degenerate motions (and hence gives a possible solution), it does not attempt to detect and flag the degeneracy. This can be done, *e.g.* by extracting the null space of the estimated covariance matrix. It would also be useful to have autocalibration methods that could estimate lens distortion. This should be relatively simple in the planar case, as distortion can be handled during homography estimation.

References

1. M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 3–16, Cambridge, U.K., April 1996.
2. O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.
3. O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
4. P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.
5. R. Hartley. Euclidean reconstruction from multiple views. In *2nd Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.
6. R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conf. Computer Vision*, pages 471–8. Springer-Verlag, 1994.
7. R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

8. R. Hartley. Minimizing algebraic error. In *IEEE Int. Conf. Computer Vision*, Bombay, January 1998.
9. A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–43, Vienna, 1996.
10. A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.
11. R. Horaud and G. Csurka. Self-calibration and euclidean reconstruction using motions of a stereo rig. In *IEEE Int. Conf. Computer Vision*, January 1998.
12. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, Amsterdam, 1996.
13. Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. Technical Report UCB/CSD-93-772, Dept. EECS, Berkeley, California, 1993.
14. S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.
15. M. Pollefeys and L Van Gool. A stratified approach to metric self-calibration. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 407–12, San Juan, Puerto Rico, June 1997.
16. M. Pollefeys, L Van Gool, and M. Proesmans. Euclidean 3d reconstruction from image sequences with variable focal length. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 31–42, Cambridge, U.K., April 1996.
17. J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.
18. P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.
19. P. Sturm. *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. Ph.D. Thesis, Institut National Polytechnique de Grenoble, December 1997.
20. B. Triggs. A new approach to geometric fitting. Available from <http://www.inrialpes.fr/movi/people/Triggs>.
21. B. Triggs. SVD based relative orientation from planar scenes. Available from <http://www.inrialpes.fr/movi/people/Triggs>.
22. B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.
23. T. Viéville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 750–6, Cambridge, MA, June 1995.
24. W. Wunderlich. Rechnerische rekonstruktion eines ebenen objekts aus zwei photographien. In *Mittsilungen Geodät. Inst. TU Gras*, Folge 40 (festschrift K. Rimmer sum 70. Geburtstag), pages 365–77, 1982.
25. C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Technical Report 2793, INRIA, INRIA Sophia-Antipolis, France, 1996.
26. Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. Technical Report 2079, INRIA, Sophia-Antipolis, France, 1993.