Autocalibration and the Absolute Quadric

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

We describe a new method for camera autocalibration and scaled Euclidean structure and motion, from three or more views taken by a moving camera with fixed but unknown intrinsic parameters. The motion constancy of these is used to rectify an initial projective reconstruction. Euclidean scene structure is formulated in terms of the **absolute quadric** — the singular dual 3D quadric (4×4 rank 3 matrix) giving the Euclidean dot-product between plane normals. This is equivalent to the traditional absolute conic but simpler to use. It encodes both affine and Euclidean structure, and projects very simply to the dual absolute image conic which encodes camera calibration. Requiring the projection to be constant gives a bilinear constraint between the absolute quadric and image conic, from which both can be recovered nonlinearly from $m \geq 3$ images, or quasi-linearly from $m \geq 4$. Calibration and Euclidean structure follow easily. The nonlinear method is stabler, faster, more accurate and more general than the quasi-linear one. It is based on a general constrained optimization technique — sequential quadratic programming — that may well be useful in other vision problems.

Keywords: autocalibration, absolute quadric, multiple images, Euclidean reconstruction, constrained optimization.

1 Introduction

Camera calibration is traditionally based on explicit 3D scene or motion measurements, but even for unknown motions in an unknown scene there are strong rigidity constraints relating the calibration to the image, scene and motion. **Autocalibration** is the recovery of calibration and motion from an unknown scene using rigidity. Structure follows easily from this.

With arbitrary cameras, structure can only be recovered up to an overall projectivity. Additional constraints are required to 'Euclideanize' it. We will focus on the traditional case of a single camera with fixed but unknown intrinsic parameters moving arbitrarily in the scene [13, 4, 7], but our formalism easily extends to handle multiple cameras and prior calibration, motion or scene constraints. Alternative approaches restrict the motion to a pure rotation [8] or a plane [1]; handle zoom modulo an initial pre-calibration [15, 16]; or assume a rigidly moving stereo head [22]. For practical applications it is important to exploit any constraints that may be available, as this both increases stability and allows autocalibration from more restricted types of motion.

Used on its own, autocalibration has several notable weaknesses: (*i*) scene scale can not be recovered — small motions in a small scene are indistinguishable from large motions in a large one; (*ii*) *generic* motions — independent rotations and some translation — are required for a unique (up to scale) solution: many common types of motion are degenerate cases; (*iii*) past formulations have tended to be complex and ill-conditioned, often adding further degeneracies of their own; (*iv*) it has been hard to incorporate additional knowledge except during a final bundle adjustment, exacerbating the degeneracy and ill-conditioning problems.

This paper focuses on the last two points, contributing a simpler, more direct problem formulation and a well-behaved numerical algorithm that easily handles additional constraints.

To appear in CVPR'97. This work was supported by INRIA Rhône-Alpes and Esprit LTR project CUMULI. I would like to thank P. Sturm for useful discussions and R. Horaud and G. Csurka for supplying calibration data.

2 The Absolute Quadric

We work in homogeneous coordinates, initially Euclidean, later projective. Finite points and asymptotic directions ('points at infinity') are given by column vectors $\mathbf{x} = (\boldsymbol{x} \;\; 1)^\top$ and $\mathbf{v} = (\boldsymbol{v} \;\; 0)^\top$. A *row* vector $\mathbf{p} = (\boldsymbol{n} \;\; d)$ specifies a plane with normal $\boldsymbol{n}$ and offset $-d$. $\mathbf{x}$ lies on $\mathbf{p}$ iff its signed distance from it vanishes: $\mathbf{p}\,\mathbf{x} = \boldsymbol{n} \cdot \boldsymbol{x} + d = 0$. The **plane at infinity** $\mathbf{p}_\infty = (\boldsymbol{0} \;\; 1)$ contains the infinite points $(\boldsymbol{d} \;\; 0)$ and no finite ones.

Change-of-basis transformations are 4×4 matrices acting by left multiplication on points ($\mathbf{x} \to \mathbf{T}\,\mathbf{x}$) and by right multiplication by the *inverse* on planes ($\mathbf{p} \to \mathbf{p}\,\mathbf{T}^{-1}$) so that point-plane products are preserved: $\mathbf{p}\,\mathbf{x} = (\mathbf{p}\,\mathbf{T}^{-1})(\mathbf{T}\,\mathbf{x})$. **Euclidean** transformations take the form $\begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}$ where $\boldsymbol{R}$ is a 3×3 rotation matrix ($\boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}$) and $\boldsymbol{t}$ a translation vector. $\boldsymbol{R}$ becomes a rescaled rotation for **scaled Euclidean** or **similarity** transformations, and an arbitrary nonsingular 3×3 matrix for **affine** ones. For **projective** transformations $\mathbf{T}$ is an arbitrary nonsingular 4×4 matrix.

To distinguish their very different transformation laws, points are called **contravariant**, and planes **covariant**. Matrices and higher dimensional arrays (tensors) have a different transformation law associated with each index. **Contraction** ('projective dot product' or sum over products of components) is only meaningful between contravariant-covariant index pairs (*e.g.* a point and a plane). Otherwise the result is completely basis-dependent.

The **absolute quadric** is the symmetric 4×4 rank 3 matrix $\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}$. It is defined to be contravariant (point-like) in each index, so $\boldsymbol{\Omega} \to \mathbf{T}\,\boldsymbol{\Omega}\,\mathbf{T}^\top$ under change-of-basis transforms $\mathbf{x} \to \mathbf{T}\,\mathbf{x}$. It follows that $\boldsymbol{\Omega}$ is invariant under Euclidean transformations, is rescaled under similarities, takes the form $\begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}$ (symmetric 3×3 nonsingular Q) under affine ones, and becomes an arbitrary symmetric 4×4 rank 3 matrix under projective ones.

Being contravariant, $\mathbf{\Omega}$ can be contracted against plane vectors. Given a finite plane $\mathbf{p}$, $\mathbf{\Omega}\,\mathbf{p}^\top$ is the point at infinity representing its Euclidean normal direction. The plane at infinity is $\mathbf{\Omega}$'s unique null vector: $\mathbf{\Omega}\,\mathbf{p}_\infty^\top = \mathbf{0}$. The Euclidean dot product of the normals of two finite planes $\mathbf{p}$ and $\mathbf{p}'$ is $\boldsymbol{n} \cdot \boldsymbol{n}' = \mathbf{p}\,\mathbf{\Omega}\,\mathbf{p}'^\top$, and the angle between them is $\cos\theta = (\mathbf{p}\,\mathbf{\Omega}\,\mathbf{p}'^\top)/\sqrt{(\mathbf{p}\,\mathbf{\Omega}\,\mathbf{p}^\top)(\mathbf{p}'\,\mathbf{\Omega}\,\mathbf{p}'^\top)}$. These formulae apply in any basis provided the corresponding $\mathbf{\Omega}$ is used. So $\mathbf{\Omega}$ is a projective encoding of both scaled Euclidean (angle between planes) and affine (plane at infinity) structure. Using $\mathbf{\Omega}$, it is straightforward to define further Euclidean concepts such as spheres, angles between lines and lines or planes, relative distances, and even (fixing a scale) absolute distances.

In contrast to planes, there is no meaningful "Euclidean dot product" between finite points. However, introducing 3-component coordinates on the plane at infinity, the dot product of two direction vectors becomes $\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}^\top \boldsymbol{C}\, \boldsymbol{v}$ where the 3×3 symmetric doubly covariant **absolute conic** matrix $\boldsymbol{C}$ becomes $\boldsymbol{I}$ in any Euclidean basis. The need for separate coordinates on $\mathbf{p}_\infty$ is inconvenient. In world coordinates the direction dot product can be written $\mathbf{u}^\top \mathbf{Q}\,\mathbf{v}$, where $\mathbf{Q}$ is *any* doubly covariant symmetric 4×4 matrix of the form $\begin{pmatrix} I & * \\ * & * \end{pmatrix}$. However there is no *canonical* choice of $\mathbf{Q}$: it *cannot* be invariant under translations. Only the upper 3×3 submatrix (the restriction of $\mathbf{Q}$ to $\mathbf{p}_\infty$) is invariant. Such a $\mathbf{Q}$ converts a point at infinity (direction vector) $\mathbf{d}$ into some finite plane $\mathbf{d}^\top \mathbf{Q}$ orthogonal to it, but there is no canonical choice of such a plane.

The absolute quadric is also much simpler to project into images than the absolute conic. Any doubly contravariant world matrix $\mathbf{M}$ can be projected to a doubly contravariant image one $\boldsymbol{m}$ according to $\boldsymbol{m} \sim \mathbf{P}\,\mathbf{M}\,\mathbf{P}^\top$, where $\mathbf{P}$ is the usual 3×4 point projection $\mathbf{x} \to \mathbf{P}\mathbf{x}$. This applies both to skew Plücker line matrices $\mathbf{L}$ and symmetric dual quadric matrices $\mathbf{Q}$. In each case the result represents the actual image of the 3D object (skew matrix representation $[\,\boldsymbol{l}\,]_\times$ of image line $\boldsymbol{l}$, and dual image conic $\boldsymbol{q}$ representing the image of the dual quadric $\mathbf{Q}$'s occluding contour). $\mathbf{\Omega}$'s projection $\boldsymbol{\omega} \equiv \mathbf{P}\,\mathbf{\Omega}\,\mathbf{P}^\top$ is the **dual absolute image conic** — a symmetric 3×3 rank 3 image matrix. Using 3×3 RQ decomposition to expand the projection $\mathbf{P} = \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I}\,|-\boldsymbol{t})$ into the traditional upper triangular **calibration matrix** $\boldsymbol{K}$, rotation $\boldsymbol{R}$ and translation to the optical centre $\boldsymbol{t}$, we find that $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top$ is invariant under rigid motions and encodes the camera's intrinsic parameters. $\boldsymbol{K}$ can be recovered from $\boldsymbol{\omega}$ by Choleski factorization.

The dual and non-dual absolute image conics $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^{-1}$ encode the 3D angular structure implicit in the image measurements. The 3D angle between the visual planes of image lines $\boldsymbol{l}$ and $\boldsymbol{m}$ is $\cos\theta = (\boldsymbol{l}\,\boldsymbol{\omega}\,\boldsymbol{m}^\top)/\sqrt{(\boldsymbol{l}\,\boldsymbol{\omega}\,\boldsymbol{l}^\top)(\boldsymbol{m}\,\boldsymbol{\omega}\,\boldsymbol{m}^\top)}$, while that between the visual rays of image points $\mathbf{x}$ and $\mathbf{y}$ is $\cos\theta = (\mathbf{x}\,\boldsymbol{\omega}^{-1}\mathbf{y}^\top)/\sqrt{(\mathbf{x}\,\boldsymbol{\omega}^{-1}\mathbf{x}^\top)(\mathbf{y}\,\boldsymbol{\omega}^{-1}\mathbf{y}^\top)}$.

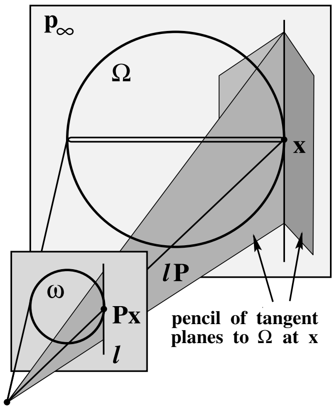

Figure 1: The absolute quadric $\mathbf{\Omega}$ is a very flat dual quadric "squashed onto" the plane at infinity, whose rim is the absolute conic $\boldsymbol{C}$.

The above algebra is all we will need to use $\mathbf{\Omega}$, but a geometric picture may help intuition. Temporarily allow $\mathbf{x}$ to be complex. Then a symmetric covariant matrix $\mathbf{Q}$ uniquely defines a non-empty **quadric**: a quadratic hypersurface (ellipsoid, hyperboloid,...) given by homogeneous equations $\mathbf{x}^\top \mathbf{Q}\,\mathbf{x} = 0$. The plane $\mathbf{x}^\top \mathbf{Q}$ is called the **dual plane** of $\mathbf{x}$ in $\mathbf{Q}$. $\mathbf{x}$ lies on $\mathbf{Q}$ iff it lies in its own dual plane: $(\mathbf{x}^\top \mathbf{Q})\,\mathbf{x} = 0$. This happens iff $\mathbf{x}^\top \mathbf{Q}$ is tangent to the quadric at $\mathbf{x}$. The **dual** of $\mathbf{Q}$ is the quadric $\mathbf{p}\,\mathbf{Q}^{-1}\mathbf{p}^\top = 0$ in the projective space of all planes. The 'points' of $\mathbf{Q}^{-1}$ are exactly the tangent planes of $\mathbf{Q}$, as is easily seen by replacing $\mathbf{p} \leftrightarrow \mathbf{x}^\top \mathbf{Q}$.

For regular $\mathbf{Q}$ the duality relation is symmetric. For singular $\mathbf{Q}$ the point quadric 'stretches out' to a cone then a plane pair, while in dual-space the quadric collapses onto a plane then a line until only its 'rim' remains (*i.e.* it becomes a dual-space plane conic curve or a point pair). The cone vertex and its dual space supporting plane correspond to the kernel of $\mathbf{Q}$.

Dually, a singular *dual* quadric $\mathbf{Q}^{-1}$ defines a dual-space cone and a point-space conic curve whose dual-space vertex or point-space supporting plane is the null space of $\mathbf{Q}^{-1}$. This is the case with the absolute quadric $\mathbf{\Omega}$: it is the degenerate dual-space quadric whose 'rim' is the absolute conic $\boldsymbol{C}$ in $\mathbf{p}_\infty$ (see fig. 1). Dual quadric projection $\mathbf{Q} \to \mathbf{P}\,\mathbf{Q}\,\mathbf{P}^\top$ is also easy to picture: an image line $\boldsymbol{l}$ is tangent to the image conic iff the pulled back visual plane $\boldsymbol{l}\,\mathbf{P}$ is tangent to the 3D quadric: $\boldsymbol{l}\,(\mathbf{P}\,\mathbf{Q}\,\mathbf{P}^\top)\,\boldsymbol{l}^\top = (\boldsymbol{l}\,\mathbf{P})\,\mathbf{Q}\,(\boldsymbol{l}\mathbf{P})^\top = 0$ (*c.f.* fig. 1).

3 Autocalibration

There are essentially three current approaches to autocalibration, all based on the motion constancy of $\boldsymbol{\omega}$. Multilinear **matching constraints** exist relating 2–10 images of any dual quadric, including $\mathbf{\Omega}$. The **Kruppa constraint** is the two image case, originally used to find epipolar geometry for relative orientation from *known* calibration. It essentially says that since epipolar lines correspond via epipolar planes, the above angle-between-visual-planes formula must give the same result for corresponding epipolar lines in either image. A compact derivation applies the **closure identity** [19] $\boldsymbol{F}_{21}\,\mathbf{P}_1 \sim$

$[\boldsymbol{e}_{12}]_\times \mathbf{P}_2$ to either side of $\boldsymbol{\Omega}$ to derive the quadric matching constraint $\boldsymbol{F}_{21}\,\boldsymbol{\omega}\,\boldsymbol{F}_{21}^\top \sim [\boldsymbol{e}_{12}]_\times\,\boldsymbol{\omega}\,[\boldsymbol{e}_{12}]_\times^\top$. Allowing for symmetry and rank deficiency, this amounts to 3 linearly or (cross multiplying to eliminate the unknown scale) 2 algebraically independent equations. $\boldsymbol{\omega}$ has 5 d.o.f. so at least 3 images are required. Various resolution procedures exist. Maybank, Faugeras & Luong [13, 4] use algebraic elimination in well-chosen coordinates, Zeller & Faugeras [21] apply least squares optimization over many images, and Hartley (reported in [14]) uses a preliminary SVD based simplification.

The second approach **stratifies** [12, 3] the problem into affine and Euclidean parts. Affine structure is encoded in $\mathbf{p}_\infty$ or the **absolute homography** $\boldsymbol{H}_\infty$ — the inter-image mapping defined by projecting pixels up onto $\mathbf{p}_\infty$. For fixed calibration, $\boldsymbol{H}_\infty = \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}$ is conjugate to a rotation and $\boldsymbol{\omega}$ turns out to be invariant: $\boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top \sim \boldsymbol{\omega}$ (with equality if $\det(\boldsymbol{H}_\infty) = 1$). This gives a linear constraint on the "Kruppa matrix" $\boldsymbol{\omega}$, sometimes also (misleadingly) called the Kruppa constraint. Since $\boldsymbol{H}_\infty$ fixes the direction $\boldsymbol{d}$ of the rotation axis, $\boldsymbol{\omega} + \lambda\,\boldsymbol{d}\,\boldsymbol{d}^\top$ also satisfies the constraint for any λ. So two rotations with different axes are needed to solve for $\boldsymbol{\omega}$.

If there is negligible translation compared to a visible 'background', $\boldsymbol{H}_\infty$ is an observable inter-image homography so autocalibration is straightforward (but not structure!) [8]. $\boldsymbol{H}_\infty$ can also be found from known vanishing points or 3D parallelism [3]. But for pure autocalibration on finite points, the only constraints on $\mathbf{p}_\infty$ and $\boldsymbol{H}_\infty$ are their relations to $\boldsymbol{\Omega}$, $\boldsymbol{\omega}$, and $\boldsymbol{K}$. Given a plane $(\boldsymbol{n}\;\; d)$ and an image projection $\mathbf{P} = \boldsymbol{A}\,(\boldsymbol{I}\,|\,-\boldsymbol{t})$, the image-to-plane homography is $\begin{pmatrix} (\boldsymbol{n}\cdot\boldsymbol{t}+d)\,\boldsymbol{I}-\boldsymbol{t}\,\boldsymbol{n} \\ -\boldsymbol{n} \end{pmatrix} \boldsymbol{A}^{-1}$. Specializing to coordinates $\mathbf{P} = (\boldsymbol{I}\,|\,\boldsymbol{0})$ and projecting into another image $\boldsymbol{A}'\,(\boldsymbol{I}\,|\,-\boldsymbol{t}')$ gives a homography $\boldsymbol{H} = \boldsymbol{A}'\,(d\boldsymbol{I} + \boldsymbol{t}'\,\boldsymbol{n})$. If $(\boldsymbol{n}\;\; d)$ represents $\mathbf{p}_\infty$ in some projective frame, applying this to $\boldsymbol{\omega} \sim \boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top$ gives equations relating the unknowns $(\boldsymbol{n}\;\; d)$ and $\boldsymbol{\omega}$. These can be solved iteratively given a reasonable initial guess for $\mathbf{p}_\infty$ or $\boldsymbol{K}$.

Hartley pioneered this sort of approach using bounds on $\mathbf{p}_\infty$ [7]. Most other authors start from an approximate prior calibration [12, 10]. Heyden & Åström's formulation [10] also partially (but independently) foreshadows ours given below. The **modulus constraint** [12, 15] — that $\boldsymbol{H}_\infty = \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}$ being conjugate to a rotation matrix must have the same unit modulus eigenvalues — focuses on $(\boldsymbol{n}\;\; d)$ by implicitly eliminating $\boldsymbol{\omega}$ or $\boldsymbol{K}$. Armstrong *et. al.* [1] take a more eclectic approach, restricting attention to planar motion and using both parallelism to constrain $\boldsymbol{H}_\infty$ and the motion constancy of the circular points (the 1D analogue of $\boldsymbol{\omega}$).

The Kruppa (epipolar constraint) approach avoids the need to deduce $\boldsymbol{H}_\infty$ indirectly from the constraints, but it can not distinguish $\boldsymbol{\Omega}$ from any other quadric with constant image: planarity ($\operatorname{rank} \boldsymbol{\Omega} = 3$) is not directly enforced.

3.1 Absolute Quadric Method

This paper introduces a third approach to autocalibration, which explicitly locates the absolute quadric in an initial projective reconstruction and uses it to 'straighten' the projective structure. $\boldsymbol{\Omega}$ is recovered using the motion constancy of its projection $\boldsymbol{\omega} \sim \mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$, where $\mathbf{P}_i \sim \boldsymbol{K}\boldsymbol{R}_i\,(\boldsymbol{I}\,|\,-\boldsymbol{t}_i)\,\mathbf{T}^{-1}$ for fixed unknown 3×3 and 4×4 transformations $\boldsymbol{K}$ and $\mathbf{T}$ and normalized rotations $\boldsymbol{R}_i$. If we knew the correct relative scaling for the projections, $\boldsymbol{\omega} = \mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$ would be linear in the unknowns $\boldsymbol{\omega}$ and $\boldsymbol{\Omega}$ and could be solved trivially. Instead, we eliminate the unknown scale by taking ratios of components and cross-multiplying, in much the same way as the point projection $\boldsymbol{x} \sim \mathbf{P}\,\mathbf{x}$ can be rewritten as $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{x}) = \mathbf{0}$:

$$\boldsymbol{\omega}^{AB}\,(\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)^{CD} - \boldsymbol{\omega}^{CD}\,(\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)^{AB} = \mathbf{0}$$

This **absolute quadric projection constraint** is the basis of our autocalibration method. The antisymmetrization interchanges *both* indices AB and CD of the 3×3 symmetric matrices $\boldsymbol{\omega}$ and $\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$. Viewing these as abstract 6D vectors, we will write this symbolically as

$$\boldsymbol{\omega} \wedge (\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top) = \mathbf{0}$$

For each image, this amounts to $\binom{6}{2} = 15$ bilinear equations (5 linearly independent) in the $10 + 6 = 16$ independent components of $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$, with coefficients quadratic in the image's reconstructed projection matrix. It can also be written as 9 bilinear equations in $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}^{-1}$ (8 linearly independent):

$$\boldsymbol{\omega}^{-1}\,\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top \;=\; \tfrac{1}{3}\operatorname{trace}(\boldsymbol{\omega}^{-1}\,\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)\cdot \boldsymbol{I}$$

The constraint says that angles between visual planes measured using $\boldsymbol{\Omega}$ must agree with those measured from the corresponding image lines using $\boldsymbol{\omega}$. Roughly speaking, the Kruppa constraint is the projection of the restriction of this to epipolar planes, while the homography constraint $\boldsymbol{\omega}\wedge(\boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top) = \mathbf{0}$ is the projection of the rotational part of it. At least 3 images are required for a unique solution. For maximum stability it is advisable to include further images, and to enforce $\operatorname{rank}(\boldsymbol{\Omega}) = 3$ (*i.e.* $\det(\boldsymbol{\Omega}) = 0$) and any known scene or calibration constraints.

We will describe two methods of resolving the absolute quadric projection constraints. Both use all $15m$ equations from m images and solve the system in algebraic least squares. The **nonlinear method** uses constrained numerical optimization on $m \geq 3$ images, while the **quasi-linear** method uses SVD based factorization on $m \geq 4$. Only the nonlinear method directly enforces $\det(\boldsymbol{\Omega}) = 0$. It requires a (very approximate) initialization, but turns out to be more accurate, stabler, faster and simpler than the quasi-linear method.

Once $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$ are known, the camera calibration $\boldsymbol{K}$ is easily found by Choleski decomposition of $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top$. Similarly, a Euclideanizing homography $\mathbf{x} \to \mathbf{T}^{-1}\,\mathbf{x}$, $\mathbf{P} \to \mathbf{P}\,\mathbf{T}$ can be found from the eigen-decomposition $\mathbf{E}\,\boldsymbol{\Lambda}\,\mathbf{E}^\top$ of $\boldsymbol{\Omega} \sim \mathbf{T}\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}\mathbf{T}^\top$ by setting $\mathbf{T} \sim \mathbf{E}\,\boldsymbol{\Lambda}^{1/2}$ (with the 0 eigenvalue in $\boldsymbol{\Lambda}$ replaced by 1). The columns of $\mathbf{T}$ are an absolute Euclidean basis in projective coordinates (*i.e.* 3 orthogonal directions and an origin). If required, the rotational part of each rectified projection $\boldsymbol{K}^{-1}\,\mathbf{P}_i\,\mathbf{T} \sim \boldsymbol{R}_i\,(\boldsymbol{I}\,|\,-\boldsymbol{t}_i)$ can be perturbed to be precisely orthonormal (*e.g.* using quaternions and SVD [11]). As always, a final, close-lying-outlier-insensitive bundle adjustment over all parameters is recommended for precise work.

3.2 Degeneracy

Autocalibration has some intrinsic limitations that apply uniformly to all algorithms. In particular, if the axes of all the camera rotations are parallel (say, vertical), the horizontal-to-vertical aspect ratio of neither the camera nor the scene can be recovered. Intuitively, a narrow scene taken with a wide aspect ratio lens is indistinguishable from a wide scene taken with a narrow lens. This is unfortunate as many real image sequences do preserve a vertical. To avoid this problem, one must either include images with 3 substantially different tilts or cyclotorsions, or rely on prior scene, motion or camera knowledge (*e.g.* aspect ratios). $90°$ rotations provide the maximum stability, but feature extraction and matching limitations mean that these are usually only possible with pure cyclotorsion.

Formally, if $\mathbf{d} = (\boldsymbol{d}\ \ 0)^\top$ is the 3D direction (common point at infinity) of the rotation axes and $\mathbf{P}_i\,\mathbf{d} = \boldsymbol{K}\boldsymbol{R}_i\,\boldsymbol{d} = \boldsymbol{K}\boldsymbol{d}$ (independent of i) is the corresponding image point, adding any multiple of $\mathbf{d}\mathbf{d}^\top$ to $\boldsymbol{\Omega}$ and the same multiple of $(\mathbf{P}\mathbf{d})(\mathbf{P}\mathbf{d})^\top$ to $\boldsymbol{\omega}$ maintains both $\boldsymbol{\omega} \sim \mathbf{P}\,\boldsymbol{\Omega}\,\mathbf{P}^\top$ and $\det(\boldsymbol{\Omega}) = 0$, so it gives another feasible solution. This corresponds to a vertical stretching of both $\boldsymbol{K}$ and the scene.

Pure translation is an even more degenerate case as it fixes *all* points at infinity: affine structure follows easily, but $\boldsymbol{\Omega}$ is essentially arbitrary so autocalibration is impossible. Various other types of motion lead to further degeneracies: Sturm [17] gives a detailed catalog. Such ambiguities must typically be handled by imposing further constraints (known skew, aspect ratio, motion...). This can be difficult with algebraic approaches, but is very easy in our numerical formalism below.

Euclidean structure and motion follow directly from autocalibration, provided only that there is sufficient translation to give a stereo baseline. Translation-neutral internal calibration methods would be useful: Hartley's method [8] requires zero translation, while reconstruction based methods require fairly substantial ones and nonplanar scenes.

3.3 Nonlinear Solution

Now consider how to solve the quadric projection constraints $\boldsymbol{\omega} \wedge (\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top) = \mathbf{0}$ for $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$, with $\det(\boldsymbol{\Omega}) = 0$. By far the most effective approach turns out to be direct constrained numerical optimization. Numerical approaches are sometimes undervalued in the vision community. Empirically, algebraic elimination on coordinate expressions provides valuable theoretical insight but almost inevitably leads to poor numerical conditioning, while numerical resolution based directly on the original, physically meaningful variables tends to be significantly more stable in practical applications, but too 'opaque' to provide much theoretical insight. At present it is hard to relate the two approaches, but progress in tensorial and Grassmann-Cayley-like formalisms [19, 5] and computational nonlinear algebra (*e.g.* [2]) may soon make this much easier.

Many constrained optimization schemes exist [6]. I will give a brief outline of the simple one used here, as I think that it has considerable potential for other constrained problems in vision. **Sequential Quadratic Programming** [6] is a general numerical scheme for optimizing smooth non-linear cost functions under smooth non-linear constraints. It is Newton-like in that it requires second derivatives of the cost function and potentially provides quadratic convergence. The version presented below is trivial to implement and adequate for our needs. More elaborate versions provide inequality constraints, stabilization and step control schemes.

The goal is to extremize a scalar cost function $f(\mathbf{x})$ subject to a vector of constraints $\mathbf{c}(\mathbf{x}) = \mathbf{0}$. Lagrange multipliers $\mathbf{z}$ give an implicit solution:

$$\nabla f + \mathbf{z}\cdot\nabla\mathbf{c} = \mathbf{0} \qquad \text{with} \qquad \mathbf{c}(\mathbf{x}) = \mathbf{0}$$

Resolve this iteratively starting from some initial guess $\mathbf{x}_0$. Approximate the cost to second order and the constraint to first order at $\mathbf{x}_0$, giving a quadratic optimization subproblem with linear constraints:

$$\min_{\delta\mathbf{x}} \left(\nabla f\cdot\delta\mathbf{x} + \frac{1}{2}\,\delta\mathbf{x}^\top\cdot\nabla^2 f\cdot\delta\mathbf{x} \right)\Big|_{\mathbf{c}+\nabla\mathbf{c}\cdot\delta\mathbf{x}=\mathbf{0}}$$

This subproblem has an exact linear solution:

$$\begin{pmatrix} \nabla^2 f & \nabla\mathbf{c}^\top \\ \nabla\mathbf{c} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \mathbf{z} \end{pmatrix} = -\begin{pmatrix} \nabla f \\ \mathbf{c} \end{pmatrix}$$

Solve for $\delta\mathbf{x}$, update $\mathbf{x}_0$ to $\mathbf{x}_1 = \mathbf{x}_0 + \delta\mathbf{x}$, re-estimate derivatives, and iterate to convergence.

In the current application, $\mathbf{x}$ contains the $10 + 6 = 16$ components of $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$. The cost function is the sum of squared violations of the projection constraints $\sum_i \|\boldsymbol{\omega} \wedge (\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)\|^2$. The constraint vector $\mathbf{c}$ enforces rank-3-ness $\det(\boldsymbol{\Omega}) = 0$ and normalization $\|\boldsymbol{\omega}\|^2 = \|\boldsymbol{\Omega}\|^2 = 3$. Further knowledge or constraints are easily added (*e.g.* known skew, aspect ratio, principal point,...). A Gauss-Newton approximation (ignoring second derivatives of the quadric projection constraints) was used for the Hessian $\nabla^2 f$.

Initial guesses $\boldsymbol{\Omega}_0$ and $\boldsymbol{\omega}_0$ are required. Using $\boldsymbol{\omega}_0 \wedge (\mathbf{P}\,\boldsymbol{\Omega}_0\mathbf{P}^\top) = \mathbf{0}$, $\boldsymbol{\Omega}_0$ can be estimated in linear least squares from an approximate calibration $\boldsymbol{\omega}_0 = \boldsymbol{K}_0\,\boldsymbol{K}_0^\top$, or $\boldsymbol{\omega}_0$ by projecting an estimated $\boldsymbol{\Omega}_0$ derived from approximate scene constraints. In fact, for $m \geq 4$ images and reasonably well placed cameras (*i.e.* several independent rotations and translations), spurious solutions seem to be rare and any initialization will do. The choices $\boldsymbol{\omega}_0 = \boldsymbol{I}$ and $\boldsymbol{\Omega}_0 = \mathbf{I}$ or $\left(\begin{smallmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{smallmatrix}\right)$ often suffice, although for 3 images, long focal lengths or highly constrained motions they can sometimes lead to local minima.

Convergence is rapid (4–10 iterations) unless the problem is degenerate, and even then failure to converge to *some* feasible solution is rare. It is worth using a fairly accurate (*e.g.* nonlinear least squares) projective reconstruction, especially in the unstable 3 image case. Omitting the $\det(\boldsymbol{\Omega}) = 0$ constraint significantly reduces both accuracy and stability.

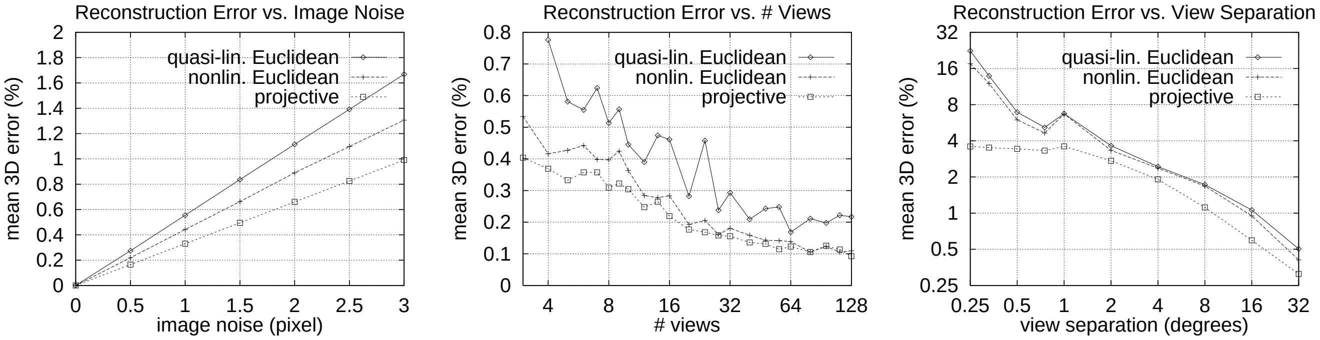

Figure 2: Mean 3D reconstruction error *vs.* image noise, number of images and angular spread of cameras for quasi-linear Euclidean, nonlinear Euclidean and projective reconstructions of point clouds.

3.4 Quasi-Linear Approach

It is also possible to solve the quadric projection constraints using a "quasi-linear" approach. No initialization is required, but at least 4 images are needed and the method is slower, less stable and less accurate than SQP.

The basic idea is to write the independent components of $\mathbf{\Omega}$ and $\boldsymbol{\omega}$ as vectors and work with the $10 \times 6 = 60$ components of their outer product matrix. The absolute quadric projection constraints are linear and have rank 15 in these variables, so the matrix can be recovered linearly from $m \geq \lceil \frac{59}{15} \rceil = 4$ images. A 10×6 SVD projects the result to rank 1 and factorizes it into vectors $\mathbf{\Omega}$ and $\boldsymbol{\omega}$. Finally, $\mathbf{\Omega}$ (rewritten as a matrix) is projected to rank 3 by annulling its smallest eigenvalue, and the method proceeds with $\mathbf{\Omega}$ and $\boldsymbol{\omega}$ as above.

Since it only enforces the rank 1 and $\det(\mathbf{\Omega}) = 0$ constraints indirectly, the quasi-linear method introduces degeneracies that are not intrinsic to the underlying problem. In particular, it fails whenever *any* point — even a finite one — is fixed in all images (*e.g.* a fixating camera).

4 Algorithm

The full algorithm for autocalibration and scaled Euclidean reconstruction is as follows:

1) Standardize all image coordinates.

2) Find the projections $\mathbf{P}_i$ by projective reconstruction.

3) Find the absolute quadric $\mathbf{\Omega}$ and image conic $\boldsymbol{\omega}$ by solving $15m$ bilinear quadric projection constraints $\boldsymbol{\omega} \wedge (\mathbf{P}_i \, \mathbf{\Omega} \, \mathbf{P}_i^\top) = \mathbf{0}$ (nonlinear and quasi-linear methods).

4) Recover the camera calibration $\boldsymbol{K}$ by Choleski decomposition of $\boldsymbol{\omega} = \boldsymbol{K} \boldsymbol{K}^\top$.

5) Find a 4×4 Euclideanizing homography $\mathbf{T}$ by eigendecomposition of $\mathbf{\Omega}$.

6) Perturb $\boldsymbol{K}^{-1} \, \mathbf{P}_i \, \mathbf{T}^{-1} \sim \boldsymbol{R}_i(\boldsymbol{I} \,|\, -\boldsymbol{t}_i)$ to be exactly Euclidean.

7) Recover Euclidean structure by $\mathbf{x} \rightarrow \mathbf{T}\,\mathbf{x}$ or back-projecting with the corrected projections.

8) Optional bundle adjustment.

Standardization rescales image pixel coordinates to lie in the unit box $[-1, 1] \times [-1, 1]$. It is *absolutely indispensable*. Otherwise, different equations of $\|\boldsymbol{\omega} \wedge (\mathbf{P}_i \, \mathbf{\Omega} \, \mathbf{P}_i^\top)\|^2 \approx 0$ have a difference in scale of (say) $256^6 \approx 10^{14}$. Their numerical conditioning is terrible and severe floating point truncation error leads to further loss of precision. This is perhaps the major reason for the observed instability of some previous autocalibration approaches. Standardization ('preconditioning') is *essential* whenever there is an implicit least squares trade-off (as here), particularly with equations of high degree. It is discussed in every text on numerical methods, but does not seem to have been widely known in vision before Hartley made the point for fundamental matrix estimation [9].

5 Experiments

To give a rough idea of the performance of the algorithm, we briefly report on numerical experiments with synthetic data. Images of random point clouds were taken with identical wide-angle cameras placed randomly within a fixed cone of viewing angles, approximately on a sphere surrounding the scene. Several other configurations have also been tried with success. Uniform random noise was added to the image points. The initial projective reconstruction was projective factorization [18, 20] followed by projective bundle adjustment (not indispensable). The nonlinear method was initialized with a calibration wrong by about 50%. Mean 3D reconstruction error over 10 trials was estimated by projective least squares alignment for projective reconstructions and scaled Euclidean alignment for Euclidean ones. There was no final Euclidean bundle adjustment, although this is recommended for real applications. Default values were ± 1 pixel noise, 6 views, 50 points, with a wide ($\pm 30°$) range of viewing directions and cyclotorsions.

Figure 2(a) shows that all errors scale linearly with noise, and that the un-adjusted nonlinear Euclidean reconstruction (with $3 + 3 + 1 = 7$ free parameters) is very nearly as good as the underlying projective one (with 15). Figure 2(b) suggests that this applies for any number of images, while the quasi-linear method is somewhat less stable. Figure 2(c) shows that the error scales smoothly as the viewing angles are decreased.

In an informal test on real images of a calibration grid, we compared un-bundle-adjusted autocalibration with the scatter of results from conventional calibration using known 3D point positions. It was within: 0.1% (0.3σ) on α_u and α_v; 0.01% (1.5σ) on α_u/α_v; and 5 pixels ($\sim$1–2σ) on u_0 and v_0 (the σ estimates here are very imprecise).

6 Discussion & Conclusions

We have described a new method for autocalibrating a moving camera with fixed but unknown intrinsic parameters, moving arbitrarily in an unknown scene. An initial projective reconstruction is rectified to give calibration and scaled Euclidean structure and motion. The method is based on a new projective encoding of metric structure: the **absolute quadric**. This is equivalent to the absolute conic, but considerably easier to use. It projects very simply to the dual absolute image conic which encodes camera calibration. The absolute quadric and conic are recovered simultaneously using an efficient constrained nonlinear optimization technique (**sequential quadratic programming**) or a quasi-linear method. The results are stable and accurate for generic camera motions, and the formalism clarifies the reasons for autocalibration's intrinsic degeneracies. A major practical advantage of the nonlinear approach is the ease with which it incorporates any further constraints that may be available, potentially significantly reducing the problems of degeneracy.

Future work will examine several topics. In the one camera case, priorities are techniques to detect and handle degeneracy, and a study of the advantages of incorporating various additional constraints. Problems with several cameras (*i.e.* several $\boldsymbol{\omega}$'s) are easily handled, as are rigidly moving stereo heads ($\boldsymbol{\omega}$ is replaced by a 'local' $\boldsymbol{\Omega}$ in the head frame, invariant under motion induced 4×4 homographies). Non-reconstruction based autocalibration techniques that work whether or not the translations are zero would be useful. Finally, SQP is being successfully applied to several other constrained statistical fitting problems in vision.

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] J. Canny. A toolkit for nonlinear algebra. Report available from http://http.cs.berkeley.edu/∼jfc, August 1993.

[3] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[4] 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.

[5] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[6] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[7] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[8] R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conf. Computer Vision*, pages 471–8. Springer-Verlag, 1994.

[9] 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.

[10] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–43, Vienna, 1996.

[11] B. Horn. Relative orientation. *Int. J. Computer Vision*, 4:59–78, 1990.

[12] 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.

[13] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[14] R. Mohr and B. Triggs. Projective geometry for image analysis. Tutorial given at *Int. Symp. Photogrammetry and Remote Sensing*, July 1996.

[15] 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.

[16] P. Sturm. Self-calibration of a moving camera by precalibration. In *British Machine Vision Conference*, pages 675–84, Edinburgh, September 1996. British Machine Vision Association.

[17] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[18] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[19] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[20] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 845–51, San Francisco, CA, 1996.

[21] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the kruppa equations revisited. Technical Report 2793, INRIA, INRIA Sophia-Antipolis, France, 1996.

[22] Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. Technical Report 2079, INRIA, INRIA Sophia-Antipolis, France, 1993.