Camera Pose and Calibration from 4 or 5 known 3D Points

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 two direct quasilinear methods for camera pose (absolute orientation) and calibration from a single image of 4 or 5 known 3D points. They generalize the 6 point 'Direct Linear Transform' method by incorporating partial prior camera knowledge, while still allowing some unknown calibration parameters to be recovered. Only linear algebra is required, the solution is unique in non-degenerate cases, and additional points can be included for improved stability. Both methods fail for coplanar points, but we give an experimental eigendecomposition based one that handles both planar and nonplanar cases. Our methods use recent polynomial solving technology, and we give a brief summary of this. One of our aims was to try to understand the numerical behaviour of modern polynomial solvers on some relatively simple test cases, with a view to other vision applications.

Keywords: Camera Pose & Calibration, Direct Linear Transform, Polynomial Solving, Multiresultants, Eigensystems.

1 Introduction

This paper describes two quasilinear methods for camera pose (absolute orientation) and calibration from a single image of 4 or 5 known 3D points. The methods are 'direct' (non-iterative) and quasilinear, so: (*i*) only linear algebra is required; (*ii*) they give a unique solution in non-degenerate cases; (*iii*) additional points are easily included to improve stability; and (*iv*) all points are on an equal footing. The classical 'Direct Linear Transform' (DLT) [1, 16] recovers the 5 internal and 6 pose parameters of a fully projective camera from the images of 6 known 3D points. The new methods are analogous to the DLT, but adopt more restrictive calibration models so that: (*i*) the minimum number of points required is reduced to 4 or 5; (*ii*) the results for a given number of points are (at least potentially) more stable, as there is more prior knowledge and hence fewer unknowns to estimate from the input data. The implemented '4 point' method assumes that the focal length f is the only unknown calibration parameter, the '5 point' one that the unknowns are focal length f and principal point (u_0, v_0). Other one (4 point) or three (5 point) parameter linear calibration models could easily be implemented using the same techniques. There are also associated multi-solution methods capable of handling one additional calibration parameter apiece: at most $2^4 = 16$ solutions for pose plus 2 calibration parameters in the 4 point case, $4^2 = 16$ for 4 in the 5 point one. We will not consider these here as they yield too many solutions to be practically useful, and numerical stability is likely to be poor. However we *will* consider a related modification of the quasilinear 4 point method, which has fewer degeneracies but which may return 2 or at most 4 solutions.

Notation: $\mathbf{X}$ denotes 3D points and $\boldsymbol{x}$ image ones. We use homogeneous coordinates and the full projective camera model $\mathbf{P} = \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I}\,|\,-\boldsymbol{t})$ where: $\mathbf{P}$ is the camera's 3×4 projection matrix; the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ give its orientation and position; and $\boldsymbol{K} = \left(\begin{smallmatrix} a & s & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1/f \end{smallmatrix}\right)$ is its internal calibration matrix. The calibration parameters $f, a, s, (u_0, v_0)$ are called *effective focal length*, *aspect ratio*, *skew* and *normalized principal point*. Numerically, we will assume well-normalized image coordinates based on some nominal focal length and principal point (*e.g.* the image centre). Fixed parameters are assumed to have their nominal values $(a, s, u_0, v_0) = (1, 0, 0, 0)$.

Rationale & Existing Work: Our methods use some prior calibration knowledge, and are best seen as intermediate between classical 3–4 point pose-with-known-calibration algorithms [16, 8, 15], and $\geq$ 6 point DLT-like ones which assume completely unknown calibration [1, 16]. They were motivated mainly by the need for approximate camera pose + calibration to initialize bundle adjustment in close range industrial photogrammetry problems. User convenience dictates the use of as few reference points as possible: accurate 3D references are troublesome and expensive to acquire and maintain, and application constraints often mean that only a few points are visible from any given location. As the bundle adjustment can correct quite a lot of residual error, stability is more important than high precision. This suggests the use of simple approximate camera models with minimal free parameters. Aspect ratio a and skew s are both stable and easily measured, so they can usually be pre-calibrated. In contrast, the 'optical scale' parameters focal length f and principal point (u_0, v_0) are difficult to pre-calibrate. Even with a fixed lens they vary slightly

To appear in ICCV'99. This work was supported by Esprit LTR project CUMULI. I would like to thank Peter Sturm for comments.

with focus, aperture, mechanical/thermal motion of the lens mount, and (with lens distortion) image position. Radial lens distortion is also significant in many close range applications, but we will not consider it here as it is difficult to handle in our DLT-like framework. See [16, 2] for extensions of the DLT which partially account for lens distortion.

Degeneracy is a significant problem for all calibration methods using near-minimal data: for certain relative positionings of the points and camera, there are infinitely many solutions and the method fails. Coplanar reference objects are especially easy to manufacture and measure. But all 6 point DLT-like methods fail for planar scenes, and *any* method with free focal length (including all of ours) fails for *frontoparallel* planes, as forward motion is indistinguishable from zoom. This is problematic as near-planarity and frontoparallelism are common in practice. A planar scene gives only two constraints on the calibration ("the images of the plane's two circular points must lie on the image of the absolute conic" [20, 11, 18, 22]). As there are 5 calibration parameters, at least 3 prior constraints are required to recover from planarity. Our 5 point method has only 2 prior constraints, so it must (and does) fail for planes. The 4 point quasilinear method should do better, but in fact it also fails owing to an algorithm-specific rank deficiency. In contrast, relatively simple homography-based methods [21, 10, 18, 22][1] solve the 4 point planar pose + focal length problem rather stably (barring fronto- and other axis parallelisms). Unfortunately, these methods fail for more than about 5% non-coplanarity, so it would be useful to develop algorithms for the difficult (but practically common) near-planar case. I will describe a preliminary version of such a 4 point method below, which uses recent eigenvector-based polynomial solving technology to separate the true root from the false ones. The underlying technique is worth knowing about as it potentially applies to many other vision problems with degeneracies and/or multiple roots.

Contents: §2 outlines our general approach, §3 covers the necessary material on polynomial solving, §4 summarizes the algorithms and gives implementation details, §5 describes experimental tests, and §6 concludes.

2 Approach

Each image of a known 3D point gives two linear constraints on the projection matrix $\mathbf{P}$, or equivalently two *non*linear ones on the camera pose and calibration. So from $n \geq 3$ points we can estimate at most the 6 pose parameters and $2n - 6$ calibration ones. These minimal cases lead to polynomial systems with multiple solutions. But we will see that by estimating one fewer parameter, we can convert such problems to linear null space computations which generically yield a unique solution. Hence, we can estimate pose plus $2n - 7 = 1, 3, 5$ calibration parameters quasilinearly from $4, 5, 6$ points. 6 points is the standard DLT, so we focus on the 4 and 5 point cases. For 4 points we develop methods for pose + focal length f; for 5 points, pose + f + principal point (u_0, v_0). Other selections of 1–3 of the 5 linear camera parameters f, a, s, u_0, v_0 can be handled analogously. The basic idea is to enforce the constraint that the remaining entries of (a, s, u_0, v_0) have their default values $(1, 0, 0, 0)$. '4' and '5 point' really denote the calibration model assumed, not just the minimum number of points required. All of our methods can incorporate further points on an equal footing, if available.

Direct formulations in terms of camera calibration $\boldsymbol{K}$ and (*i*) pose $(\boldsymbol{R}, \boldsymbol{t})$ (using *e.g.* quaternions for $\boldsymbol{R}$), or (*ii*) the camera-point distances (*c.f.* [8, 15]), are possible, but seem to lead to rather unwieldy matrices. Instead, we proceed indirectly as follows: (*i*) find the linear space of 3×4 projection matrices consistent with the given points; (*ii*) recover the estimated projection matrix $\mathbf{P}$ quasilinearly from this subspace using the calibration constraints; (*iii*) extract the calibration and pose $\boldsymbol{K}, \boldsymbol{R}, \boldsymbol{t}$ from $\mathbf{P}$ in the usual way. We focus mainly on step (*ii*) which is the novel contribution.

Step 1 is very similar to the standard 6 point DLT [1, 16]. Given a 3D point $\mathbf{X}$ and its image $\lambda \boldsymbol{x} = \mathbf{P}\,\mathbf{X}$, eliminate the unknown depth λ by forming the cross-product $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{X}) = 0$, and select two independent homogeneous linear constraints on $\mathbf{P}$ from this. (In fact, I project $\mathbf{P}\,\mathbf{X}$ orthogonal to $\boldsymbol{x}$ using $\boldsymbol{x}$'s 3×3 Householder matrix. This is slightly different, but the overall effect is similar). The constraints from n points can be assembled into a $2n \times 12$ matrix which generically has rank $\min(2n, 11)$. With the standard DLT, $n \geq 6$, the rank is generically 11, and the 12 components of the unique null vector directly give the corresponding projection matrix $\mathbf{P}$. For $n = 4, 5$ the rank is generically 8,10 leaving a $d = 4, 2$ dimensional null space. In the noiseless case, this still contains the true projection: $\mathbf{P} = \mathbf{P}(\mu) \equiv \sum_{i=1}^{d} \mu_i \, \mathbf{P}_i$ where the $\mathbf{P}_i$ are 3×4 projections corresponding to the d vectors of a null space basis, and μ_i are unknown parameters. The null space is calculated numerically by SVD. Even if $n > 4, 5$ and the rank is clearly greater than 8,10, we still take the $d = 4, 2$ smallest singular vectors to span the space $\mathbf{P}(\mu)$ used in the next step.

Step 2 recovers $\mathbf{P}(\mu)$ from the $\mathbf{P}_i$ by estimating μ using the calibration constraints. By the decomposition $\mathbf{P} \simeq \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I} \,|\, -\boldsymbol{t})$, the 4×4 Euclidean invariant **absolute dual quadric** matrix $\boldsymbol{\Omega} \equiv \left(\begin{smallmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{smallmatrix}\right)$ projects to the **dual image of the absolute quadric** (DIAC) [19, 9, 13]

$$\boldsymbol{\omega} \equiv \mathbf{P}\,\boldsymbol{\Omega}\,\mathbf{P}^\top \simeq \boldsymbol{K}\boldsymbol{K}^\top \qquad (1)$$

We use this to convert constraints on the calibration $\boldsymbol{K}$ into ones on candidate projections $\mathbf{P}(\mu)$ or their associated

[1] For full 5 parameter calibration from several known planes, [18], [22] and (slightly later) I myself all independently developed essentially the same method, which is highly recommended.

DIAC's $\boldsymbol{\omega} = \boldsymbol{\omega}(\mu) \equiv \mathbf{P}(\mu)\,\boldsymbol{\Omega}\,\mathbf{P}(\mu)^\top$. For the 4 point method the only unknown calibration parameter is f. The remaining parameters take their default values $a = 1$, $s = u_0 = v_0 = 0$ so $\boldsymbol{K} = \mathrm{diag}(f, f, 1)$, $\boldsymbol{K}\boldsymbol{K}^\top = \mathrm{diag}(f^2, f^2, 1)$ and the constraints (1) become

$$\boldsymbol{\omega}_{11} = \boldsymbol{\omega}_{22} \qquad \boldsymbol{\omega}_{12} = \boldsymbol{\omega}_{13} = \boldsymbol{\omega}_{23} = 0 \qquad (2)$$

This overconstrained system of 4 homogeneous quadratics in 4 variables $\mu_1, \ldots, \mu_4$ generically has at most one solution. We will see below how to convert such a system into a rectangular multiresultant matrix $\mathbf{R}$ whose unique null vector encodes the solution. We can then estimate the null vector numerically (*e.g.* using SVD), extract the corresponding μ, substitute into $\mathbf{P}(\mu)$ to obtain $\mathbf{P}$, and decompose $\mathbf{P}$ to obtain full camera pose + calibration. In this case the resultant matrix turns out to be 80×56 — large, but still tractable.

The 5 point method is similar. It recovers (μ_1, μ_2) using the calibration constraints $a = 1, s = 0$. These are no longer linear in the entries of $\boldsymbol{K}\boldsymbol{K}^\top$, but fortunately they *are* linear in those of $\boldsymbol{\omega}^{-1} \simeq (\boldsymbol{K}\boldsymbol{K}^\top)^{-1}$, whose upper 2×2 submatrix is proportional to $\begin{pmatrix} 1 & -s \\ -s & a^2+s^2 \end{pmatrix}$. $\boldsymbol{\omega}^{-1}$ is proportional to the matrix of cofactors of $\boldsymbol{\omega}$, and hence quadratic in $\boldsymbol{\omega} = \boldsymbol{\omega}(\mu)$ or quartic in μ. The system $a = 1, s = 0$ or $\boldsymbol{\omega}^{-1}_{11} = \boldsymbol{\omega}^{-1}_{22}$, $\boldsymbol{\omega}^{-1}_{12} = 0$ becomes

$$\begin{aligned} \boldsymbol{\omega}_{22}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{23}^2 &= \boldsymbol{\omega}_{11}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{13}^2 \\ \boldsymbol{\omega}_{21}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{23}\,\boldsymbol{\omega}_{31} &= 0 \end{aligned} \qquad (3)$$

This overconstrained system of two homogeneous quartics in (μ_1, μ_2) yields an 8×8 (Sylvester) resultant matrix whose null vector again gives the solution quasilinearly.

Notes: The globally optimal $\mathbf{P}$ lies somewhere in the nonlinear variety of projection matrix space cut out by the d calibration constraints. It has low error so it is usually not far from the space spanned by the smallest few singular vectors of the DLT constraint matrix $\mathbf{A}$. This motivates the choice of the subspace $\mathbf{P}(\mu)$. But with noisy data $\mathbf{P}(\mu)$ rarely contains the exact global optimum. In fact, the calibration system has 1 redundant d.o.f. on $\mathbf{P}(\mu)$, so it seldom has *any* exact solution there, let alone an optimal one. Worst still, step 2 finds its "unique" near-solution by roughly minimizing some highly twisted heuristic form of the constraint residual, *regardless of the resulting image error*. The measured data points contribute *only* to the estimation of the "null" space $\mathbf{P}(\mu)$ in step 1. This is fine for minimal point sets where $\mathbf{P}(\mu)$ is the true null space of the DLT constraints. But for noisy, non-minimal, well-conditioned data $\mathbf{P}(\mu)$ generally contains several far-from-null directions and there is a risk that step 2 will return a solution with quite large residual. In summary, the multiresultant solution neither exactly satisfies the constraints, nor minimizes the fitting error even within the $\mathbf{P}(\mu)$ subspace, let alone outside it. Experimentally this is verified: (*i*) nonlinear refinement significantly reduces the residual of the multiresultant solutions; (*ii*) the multiresultant methods are most suited to near-minimal data — as more data is added their performance improves comparatively little, so for well-conditioned high-redundancy data the 6 point DLT is preferable.

3 Solving Polynomial Systems

This section briefly sketches the multiresultant theory required to understand our algorithms. Part of this material is classical, but it has seen a significant revival lately and we will use some recent results. There is no space for details here, but the material deserves to be better known in the vision community as large-scale polynomial solving is rapidly becoming a feasible proposition. See, *e.g.* [4, 14] for references and further reading.

A **polynomial** $\mathbf{p}(\mathbf{x}) = \sum p_\alpha\,\mathbf{x}^\alpha$ in variables $\mathbf{x} = (x_1, \ldots, x_n)$ is a finite sum of **coefficients** p_α times **monomials** $\mathbf{x}^\alpha \equiv \prod_{i=1}^n x_i^{\alpha_i}$, with integer **exponents** $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n$. For **homogeneous** polynomials, all exponents have the same **degree** $|\alpha| \equiv \sum_i \alpha_i$. Any polynomial can be **homogenized** by including an extra variable x_0 at a suitable power in each term, and de-homogenized by setting $x_0 = 1$. The product of polynomials $\mathbf{p}, \mathbf{q}$ is $(\mathbf{p}\,\mathbf{q})(\mathbf{x}) = \sum_\alpha \left(\sum_\beta p_{\alpha-\beta}\,q_\beta\right) \mathbf{x}^\alpha$. By choosing some sufficiently large list of working exponents $\mathcal{A}$ (to be specified below), we can represent polynomials as row vectors $\mathbf{p}_\mathcal{A} \equiv (\ldots p_\alpha \ldots)$ and monomials as columns $\mathbf{x}^\mathcal{A} \equiv (\ldots \mathbf{x}^\alpha \ldots)^\top$, so that $\mathbf{p}(\mathbf{x}) = \mathbf{p}_\mathcal{A} \cdot \mathbf{x}^\mathcal{A}$ is the usual row-column dot product. All of the nonlinearity is hidden in the "simple" monomial evaluation mapping $\mathbf{x} \to \mathbf{x}^\mathcal{A}$. Polynomial multiplication can be represented by matrices $\mathbf{M}_\mathcal{A}(\mathbf{q})$ acting on the right on row vectors $\mathbf{p}$: $(\mathbf{p}\,\mathbf{q})_\mathcal{A} = \mathbf{p}_\mathcal{A}\,\mathbf{M}_\mathcal{A}(\mathbf{q})$. Row α of $\mathbf{M}_\mathcal{A}(\mathbf{q})$ contains the row vector of $\mathbf{x}^\alpha\,\mathbf{q}(\mathbf{x})$, *i.e.* the coefficients of $\mathbf{q}$ 'shifted along' by α. Coefficients shifted outside of $\mathcal{A}$ are truncated, but we will use only untruncated rows.

We want to find the roots of a polynomial system $\{\mathbf{p}_1(\mathbf{x}), \ldots, \mathbf{p}_m(\mathbf{x})\}$, *i.e.* the points $\mathbf{x}$ at which all $\mathbf{p}_i(\mathbf{x}) = 0$. It follows that $\sum_i \mathbf{p}_i(\mathbf{x})\,\mathbf{q}_i(\mathbf{x})$ also vanishes at all roots $\mathbf{x}$, for any other polynomials $\mathbf{q}_i(\mathbf{x})$. As row vectors, such sums are linear combinations of rows $\mathbf{x}^\alpha\,\mathbf{p}_i(\mathbf{x})$ from the multiplication matrices $\mathbf{M}_\mathcal{A}(\mathbf{p}_i)$. Gather the (untruncated) rows of these into a big 'multiresultant' matrix $\mathbf{R}$. The vanishing of $\mathbf{x}^\alpha\,\mathbf{p}_i$ at roots implies that the monomial vector $\mathbf{x}^\mathcal{A}$ of any root is *orthogonal* to all rows of $\mathbf{R}$: *The linear subspace of monomial vectors spanned by the root vectors* $\mathbf{x}^\mathcal{A}$ *is contained in the right null space of* $\mathbf{R}$. It turns out that by making $\mathcal{A}$ larger, this null space can often be made to 'close in' on the space spanned by the roots, until they eventually coincide. If there is only one root $\mathbf{x}$, $\mathbf{x}^\mathcal{A}$ can then be recovered (modulo scale) as the unique null vector of $\mathbf{R}$. $\mathbf{x}$ then follows easily by taking suitable ratios of components, with at most some trivial root extractions. For numerical accuracy, large-modulus components of $\mathbf{x}^{\mathcal{A}}$ should be selected for these ratios.

For homogeneous polynomials, roots are counted projectively in the homogeneous variables $(x_0, \ldots, x_n)$. Bezout's theorem says that a system of n such polynomials of degrees d_i has either exactly $\prod_{i=1}^{n} d_i$ such complex roots (counted with appropriate multiplicities), or (non-generically) an infinite number. Adding further polynomials gives an overconstrained system that generically has no roots at all. But if it does have one it is generically unique and can be recovered by the above construction. In particular, for **dense** homogeneous polynomials (ones whose coefficients of the given degrees are all nonzero and generic), Macaulay's classical multiresultant [12] chooses $\mathcal{A}$ to contain all monomials of degree $D = 1 + \sum_{i=1}^{n+1}(d_i - 1)$.

Taking all untruncated rows of the multiplication matrices as above generally gives a rectangular matrix $\mathbf{R}$. Macaulay gave a prescription for choosing a *minimal* set of rows (a square $\mathbf{R}$) that (generically) suffices to generate the null space. This is useful for theory and most current multiresultant codes adopt it. But numerically it is ill-advised as nothing says that the selected rows are particularly well-conditioned. I prefer to include all available rows and use a stable numerical null space routine, either pivoting to select suitable rows, or using an orthogonal decomposition like QR or SVD that averages errors over all of them. This also allows any available additional polynomials to be included on an equal footing for better stability and/or reduced degeneracy, simply by adding the appropriate rows of their multiplication matrices to $\mathbf{R}$. If some of the polynomial coefficients vanish the Macaulay construction may fail. Sparse 'Newton' multiresultants are available in such cases [7, 6, 4].

The above is all we need for the quasilinear 4 and 5 point methods, as the $\mathbf{P}_i$ and hence (2), (3) are usually dense. However, as mentioned above, the 4 point method fails unnecessarily for coplanar points. $\mathbf{R}$ develops 3 additional null vectors in this case, corresponding roughly to infinite and zero focal lengths (though not necessarily to coherent roots). The true root monomial still lies in this 4D null space, but it is no longer isolated by the null space computation alone. This failure is annoying, as coplanarity is not actually an intrinsic degeneracy of the 4 point problem. Indeed, stable specialized methods exist for the planar case [21, 10, 18, 22]. Unfortunately, these fail even for mildly non-coplanar scenes. It would be useful to develop a method that handled both cases simultaneously, and in particular the difficult near-planar region. To do this we need some more theory.

The columns of the resultant matrix $\mathbf{R}$ are labelled by the exponent set $\mathcal{A}$. If we partition $\mathcal{A}$ into subsets $\mathcal{A}_1 + \mathcal{A}_0$, $\mathbf{R}$ can be partitioned conformably (after column permutation) as $\mathbf{R} = (\mathbf{R}_1 \,|\, \mathbf{R}_0)$. Choose the partition so that: (*i*) $\mathbf{R}_1$ has full column rank $N_1 = |\mathcal{A}_1|$; (*ii*) $\mathcal{A}_0$ is relatively small and compact in the sense given below. For any left pseudoinverse[2] $\mathbf{R}_1^\dagger$ of $\mathbf{R}_1$, the column span of the $N \times N_0$ matrix $\mathbf{U} = \begin{pmatrix} -\mathbf{R}_1^\dagger \mathbf{R}_0 \\ \mathbf{I} \end{pmatrix}$ contains the null space of the columns of $\mathbf{R}$. In fact, $\mathbf{U}$ regenerates null vectors $\mathbf{v}$ from their $\mathcal{A}_0$ components: $\mathbf{R}\,\mathbf{v} = \mathbf{R}_1\,\mathbf{v}_1 + \mathbf{R}_0\,\mathbf{v}_0 = 0$ implies $\mathbf{U}\,\mathbf{v}_0 = \begin{pmatrix} -\mathbf{R}_1^\dagger \mathbf{R}_0\,\mathbf{v}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^\dagger \mathbf{R}_1\,\mathbf{v}_1 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_0 \end{pmatrix} = \mathbf{v}$.

Now choose a non-constant polynomial $\mathbf{q}(\mathbf{x})$ such that the row vectors $\mathbf{x}^\alpha\,\mathbf{q}$ are untruncated in $\mathcal{A}$ for all $\alpha \in \mathcal{A}_0$. (It is to avoid truncation here that $\mathcal{A}_0$ needs to be small and compact. $\mathbf{q}$ can have negative exponents if necessary). Assemble these $\mathcal{A}_0$ rows of $\mathbf{M}_{\mathcal{A}}(\mathbf{q})$ into an $N_0 \times N$ matrix $\mathbf{M} = (\mathbf{M}_1 \,|\, \mathbf{M}_0)$, and form the $N_0 \times N_0$ **reduced multiplication matrix**

$$\mathbf{M}_{\mathcal{A}_0}(\mathbf{q}|\mathbf{p}_1 \ldots \mathbf{p}_m) \;\equiv\; \mathbf{M}\,\mathbf{U} \;=\; \mathbf{M}_0 - \mathbf{M}_1\,\mathbf{R}_1^\dagger\,\mathbf{R}_0$$

What is happening here is that the polynomials $\mathbf{x}^\beta\,\mathbf{p}_i$ (the rows of $\mathbf{R}$, acting via $\mathbf{R}_1^\dagger$) have been used to eliminate the $\mathcal{A}_1$ exponents of the polynomials $\mathbf{x}^\alpha\,\mathbf{q}$, leaving a matrix on the reduced exponent set $\mathcal{A}_0$ representing *multiplication by* $\mathbf{q}$ *followed by reduction modulo (multiples of) the* $\mathbf{p}_i$. The reduction leaves the value of $\mathbf{q}$ unchanged at all roots $\mathbf{x}$ of the $\mathbf{p}_i$, as multiples of $\mathbf{p}_i(\mathbf{x}) = 0$ are added to it. Hence, using the above regeneration property, ***for any root*** $\mathbf{x}$ ***of the system*** $\{\mathbf{p}_1 \ldots \mathbf{p}_m\}$***, the monomial vector*** $\mathbf{x}^{\mathcal{A}_0}$ ***is an eigenvector of*** $\mathbf{M}_{\mathcal{A}_0}(\mathbf{q})$ ***with eigenvalue*** $\mathbf{q}(\mathbf{x})$ ***:***

$$\mathbf{M}_{\mathcal{A}_0}(\mathbf{q})\,\mathbf{x}^{\mathcal{A}_0} \;=\; \mathbf{M}\,\mathbf{x}^{\mathcal{A}} \;=\; \mathbf{q}(\mathbf{x})\,\mathbf{x}^{\mathcal{A}_0}$$

Even if we can't reduce the null space of $\mathbf{R}$ to a single vector owing to multiple roots, ill conditioning, *etc*, we can still obtain roots by solving a nonsymmetric eigenvalue problem. Given $\mathbf{x}^{\mathcal{A}_0}$ we can recover $\mathbf{x}$ as before, if necessary regenerating $\mathbf{x}^{\mathcal{A}} = \mathbf{U}\,\mathbf{x}^{\mathcal{A}_0}$ to do so. Possible problems with this construction are: (*i*) it may be impossible to find an $\mathcal{A}_0$ with well-conditioned $\mathbf{R}_1^\dagger$ and non-constant, untruncated $\mathbf{q}$; (*ii*) if the chosen $\mathbf{q}$ takes similar values at several roots, the eigenvalue routine may fail to separate the corresponding eigenspaces cleanly, leading to inaccurate results; (*iii*) post-processing is required, as some of the recovered eigenvectors may be garbage (*i.e.* vectors that define valid linear forms on polynomials, but whose components do not correspond to the monomials of any root). Beware that nonsymmetric eigenproblems are intrinsically rather delicate, and in this application can become spectacularly unstable for ill-conditioned $\mathbf{R}_1$ or ill-chosen $\mathbf{q}$. This is *not* immediately obvious from the recovered eigenvalues or eigenvectors. However the condition number of the eigenvector matrix is a fairly reliable indicator.

[2] *I.e.* , $\mathbf{R}_1^\dagger\,\mathbf{R}_1 = \mathbf{I}_{N_1 \times N_1}$. Such $\mathbf{R}_1^\dagger$ are easily calculated from most numerical decompositions of $\mathbf{R}_1$.

This multiplication matrix approach to numerical root-finding is quite recent [17, 14, 4], although its roots go back a century. So far as I know, the observation that it continues to work when $\mathcal{A}_0$ and $\mathbf{U}$ span more than the null space of $\mathbf{R}$ is new. This is numerically useful, as it allows eigensystem size to be traded against elimination stability. This approach can be used to find all of Bezout's projective roots of a dense n polynomial system by building a Macaulay matrix with $D = 1+\sum_{i=1}^{n}(d_i-1)$ and choosing $\mathcal{A}_0$ to contain all monomials $\mathbf{x}^\alpha$ with $0 \le \alpha_i < d_i$. Here, $\mathbf{R}_1$ generically spans the column space of $\mathbf{R}$, so there are no extraneous eigenvalues. Sparse analogues also exist.

We will use the eigenvector method to stabilize the 4 point quasilinear one against near-planar scenes. Coplanarity increases the null space dimension of the 4 point multiresultant $\mathbf{R}$ from 1 to 4. So we need to choose four exponents of $\mathcal{A}$ for the reduced exponent set $\mathcal{A}_0$, and the routine will return at most four potential roots. Currently I use the four lowest degree exponents $(\mu_1, \mu_2, \mu_3, 1)$ (where $\mu_4 = 1$ is the homogenizing variable). This choice parametrizes the true root and at least one false null vector stably, but it is not ideal as the remaining 1–2 false null vectors are mainly supported on 'high' exponents deep within $\mathcal{A}_1$. I know of no way around this dilemma: the supports of the null vectors are too widely separated to gather into an $\mathcal{A}_0$ supporting an untruncated $\mathbf{q}$, even if we could isolate which exponents were needed. With the heuristics discussed below, the modified 4 point routine performs tolerably well despite the fact that both $\mathbf{R}_1^\dagger$ and the eigenvalue problem are often fairly ill-conditioned, but a cleaner solution would be desirable.

4 Implementation

The steps of the new pose + calibration algorithms are as follows, where $d = 4, 2$ for the 4,5 point method:

1. Use SVD to estimate the d-D null space $\mathbf{P}(\mu) = \sum_{i=1}^{d} \mu_i \, \mathbf{P}_i$ of the DLT constraints $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{X}) = 0$.
2. Substitute $\mathbf{P}(\mu)$ into the 4 quadratic calibration constraints (2) (4 point) or 2 quartic ones (3) (5 point).
3. Form the rectangular multiresultant matrix $\mathbf{R}$ of the resulting polynomials, use SVD to recover its unique null vector $\mu^{\mathcal{A}}$, and extract μ. For the eigenvector method, choose a splitting $\mathcal{A}_0$ and a compatible random polynomial $\mathbf{q}(\mu)$, use $\mathbf{R}_1^\dagger$ to form $\mathbf{q}$'s reduced multiplication matrix, extract eigenvectors $\mu^{\mathcal{A}_0}$, and recover the solutions μ.
4. (Optional) Refine the recovered roots μ by Newton iteration against the original calibration constraints.
5. Calculate the camera projection matrix $\mathbf{P}(\mu)$ and decompose it as usual to get pose + calibration.

The routines have been implemented in OCTAVE/MATLAB. The necessary multiresultant matrices were calculated using a MAPLE routine similar to [14] (available from the author). The null space methods are straightforward to implement, but the eigenvector one requires some care. The choice of the 'pivoting' exponent set $\mathcal{A}_0$ is critical, and I am not happy with the current heuristic. In fact, I have tried only the μ_4-based exponent set, but varied which of the projection matrices $\mathbf{P}_i$ (the d smallest right singular vectors of the DLT equations) is assigned to μ_4. I tried various permutations and also random orthogonal mixings. None are wholly satisfactory and a more effective pivoting strategy is clearly required before the eigenvalue approach can be routinely used to rescue resultants from multi-root degeneracies. For 4 points and near-planar scenes, making $\mathbf{P}_4$ correspond to the *greatest* of the 4 singular values is by far the best choice. But it performs erratically for non-coplanar scenes and $n > 4$ points. Changing strategies makes enormous differences to the conditioning of $\mathbf{R}_1$, but does not necessarily stop the routine from working. Slight ($\mathcal{O}(10^{-10})$) damping of the pseudoinverse is also essential with the current $\mathcal{A}_0$, as $\mathbf{R}_1$ actually becomes singular for coplanar points.

Another issue for the eigenvector method is the choice of multiplier polynomial $\mathbf{q}(\mathbf{x})$. For simplicity I have used a linear $\mathbf{q}$, although anything up to 4^{th} order could be handled. For maximum stability, it is important that $\mathbf{q}$ should take well-separated values at different roots. In practice, I randomly choose a few $\mathbf{q}$'s and take the one that gives the best conditioned eigensystem. The cost is negligible compared to the calculation of $\mathbf{R}_1^\dagger$.

The current implementations use SVD for all null space computations. This is perhaps overkill, but it guarantees the stablest possible results. Speed is adequate ($<$ 1 second), but might become an issue if the 4 point methods were used in a RANSAC loop.

The roots μ are recovered by selecting suitable large-modulus components of $\mu^{\mathcal{A}}$ and taking their ratios. Optionally, they may then be 'refined' by a simple Newton iteration that minimizes the error in the calibration polynomials (2),(3) over μ. For the best results the original calibration constraints should be used, not their resultant matrix $\mathbf{R}$. Full Newton rather than Gauss-Newton iteration is advisable here, owing to the nonlinearity of the constraints.

5 Experiments

The graphs show some simple experimental tests on synthetic data. The 3D test points are well spread and by default non-coplanar. They are viewed from about 5 scene diameters by a 512×512 camera with $f \approx 1000 \pm 400$ and a default Gaussian noise of 0.5 pixels (which is easily obtainable with marked target points). Median errors over 300 trials are reported. For flat scenes, the plane is viewed

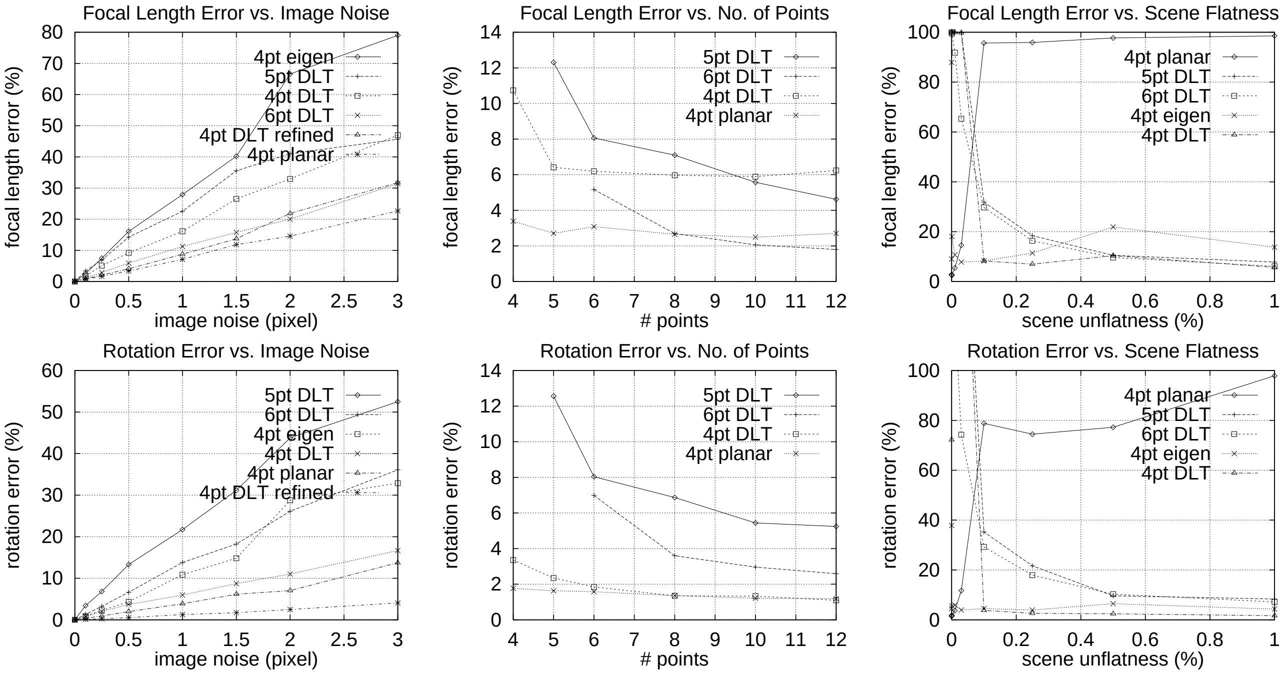

Figure 1: *Left:* Focal length & rotation error *vs.* noise, for each method's minimal point number and preferred scene flatness. *Middle:* Error *vs.* number of points for 0.5 pixels noise. *Right:* Error *vs.* scene flatness for minimal point numbers.

at about $30 \pm 15°$ from normal to avoid the frontoparallel degeneracy, which all of the algorithms here suffer from.

The graphs show that all methods are quite sensitive to noise, but all scale linearly with it up to at least 50% relative error. The planar 4 point f-only method [21, 10, 18, 22] is both simpler and intrinsically stabler than the 3D ones, but it can not tolerate more than about 5% non-coplanarity. Plane + parallax might be an interesting approach for pose + calibration from flat scenes. The 5 and 6 point DLT's fail for scenes within about 20% of planarity, whereas the 4 point DLT one (whose failure is algorithmic not intrinsic) continues to work down to around 10%. The 4 point eigenvector method works even for planar scenes, but overall it is somewhat erratic. (*E.g.* it gives better results for near-planar scenes, and for 4 points rather than $n > 4$). As above, this is due to the lack of a good policy for the choice of the residual exponent set $\mathcal{A}_0$.

The performance of the 5 point DLT is somewhat disappointing. The traditional 6 point DLT is always preferable when there are $n \geq 6$ points, and for $n \geq 10$ even beats the 4 point DLT on f (but not on orientation). In general the relative rankings depend somewhat on the error measure chosen. The fact that the 6 point DLT does better than the 4–5 point ones for large numbers of points is annoying but not unexpected. As discussed in section 2, it happens because the multiresultant step blindly minimizes some sort of twisted constraint residual over the subspace $\mathbf{P}(\mu)$, without any consideration of the image errors produced. For redundant data $\mathbf{P}(\mu)$ usually contains projections with significant image error, hence the problem. I am currently working on this, but for now the 4 and 5 point methods are most useful for minimal and near-minimal data.

The '4pt DLT refined' method runs Newton's method on the output of the linear 4 point one, to minimize the RMS error of the calibration constraints. Such nonlinear refinement is highly recommended, as it reduces the overall residual error by a factor of 2–5. A mini bundle adjustment over the resulting pose estimate would do even better, as it would not be restricted to the d-D 'null space' of the DLT constraints. The large reduction in residual suggests that there is considerable scope for improving the heuristic least squares error function embodied in the multiresultant root estimate. However, except for the initial DLT step, simple rescaling has little effect: the multiresultant is insensitive to the scaling of its input data over a range of at least $10^{\pm 2}$.

Use of the rectangular multiresultant is recommended, as it makes the results significantly more consistent, allows additional points to be incorporated, and reduces errors by 20–40% compared to the square Macaulay resultant.

All of the methods give more accurate relative results as f grows larger and the camera recedes, simply because a larger magnification camera with the same pixel noise is a more accurate angle measurer. Conversely, for small f angular errors and perspective become large and the problem becomes very nonlinear: spurious roots near $f \approx 0$ are common in (auto-)calibration problems. This makes it clear that $1/f$ is a natural expansion parameter, and suggests that pseudo-affine initialization may be a good implementation strategy for pose + calibration methods, *c.f.* [5, 3].

6 Summary and Conclusions

The 4 point quasilinear pose method performs reasonably well considering how much information it extracts from such a small amount of input data. The 5 point method is less good and is probably best reserved for special situations. Both methods are most useful for minimal or near-minimal data. Neither competes with the traditional 6 point DLT when there are ≥ 6 well-spaced points, and hence neither realizes my hopes that calibration constraints could be used to stabilize the 6 point method. The reason is basically the splitting of the problem into 'DLT' and 'multiresultant' parts with different, incompatible error metrics. This sort of subdivision is commonplace in vision geometry, but it is clear that it prevents the data and constraints from being combined very effectively. I am currently reflecting on better ways to handle this. Also, the whole issue of scaling, pivoting, and the effective error metric used by polynomial methods like multiresultants remains very unclear. But the numerical side of this field is very recent, and significant improvements are to be expected over the next few years.

The use of oversized, rectangular multiresultant matrices $\mathbf{R}$ improves the numerical conditioning and also allows redundant data to be included, so it should help to make polynomial-based initialization of many vision optimization problems more feasible. For more difficult cases where there are multiple near-roots and other degeneracies, the eigenvector method has considerable potential. However, if my current experience with the 4 point eigenvector method is any guide, more work on pivoting/exponent choice strategies is essential to make numerically trustworthy.

References

[1] Y. Abdel-Aziz and H. Karara. Direct linear transformation from comparator to object space coordinates in close-range photogrammetry. In *ASP Symp. Close-Range Photogrammetry*, pages 1–18, Urbana, Illinois, 1971.

[2] K. B. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, Scotland, 1996.

[3] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 18(11):1098–1104, 1996.

[4] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics, vol.185. Springer Verlag, 1998.

[5] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *Int. J. Computer Vision*, 15(1/2):123–141, 1995.

[6] I. Emiris and J. Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symbolic Computation*, 20:117–49, 1995.

[7] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhaüser, Boston, 1994.

[8] R.M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 592–8, 1991.

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

[10] K. Kanatani. Camera calibration using a planar surface. Unpublished, November 1998.

[11] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 482–8, 1998.

[12] F. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge University Press, 1916.

[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] B. Mourrain. An introduction to linear algebra methods for solving polynomial equations. In *HERMECA'98*. See also: http://www-sop.inria.fr/saga/logiciels/multires.html.

[15] L. Quan and Z.D. Lan. Linear $n \geq 4$-point pose determination. In *IEEE Int. Conf. Computer Vision*, 1998.

[16] C.C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[17] H. J. Stetter. Eigenproblems are at the heart of polynomial system solving. *SIGSAM Bulletin*, 30:22–5, 1996.

[18] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Conf. Computer Vision & Pattern Recognition*, 1999.

[19] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[20] B. Triggs. Autocalibration from planar scenes. In *European Conf. Computer Vision*, pages I 89–105, Freiburg, June 1998.

[21] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robotics & Automation*, 3(4):323–344, 1987.

[22] Zhengyou Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, December 1998.