Optimal Estimation of Matching Constraints

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 work in progress on a numerical library for estimating multi-image matching constraints, or more precisely the multi-camera geometry underlying them. The library will cover several variants of homographic, epipolar, and trifocal constraints, using various different feature types. It is designed to be modular and open-ended, so that (*i*) new feature types or error models, (*ii*) new constraint types or parametrizations, and (*iii*) new numerical resolution methods, are relatively easy to add. The ultimate goal is to provide practical code for stable, reliable, statistically optimal estimation of matching geometry under a choice of robust error models, taking full account of any nonlinear constraints involved. More immediately, the library will be used to study the relative performance of the various competing problem parametrizations, error models and numerical methods. The paper focuses on the overall design, parametrization and numerical optimization issues. The methods described extend to many other geometric estimation problems in vision, *e.g.* curve and surface fitting.
Keywords: Matching constraints, multi-camera geometry, geometric fitting, statistical estimation, constrained optimization.

1 Introduction and Motivation

This paper describes work in progress on a numerical library for the estimation of multi-image matching constraints. The library will cover several variants of homographic, epipolar, and trifocal constraints, using various common feature types. It is designed to be modular and open-ended, so that new feature types or error models, new constraint types or parametrizations, and new numerical resolution methods are relatively easy to add. The ultimate goal is to provide practical code for stable, reliable, statistically optimal estimation of matching geometry under a choice of robust error models, taking full account of any nonlinear constraints involved. More immediately, the library is being used to study the relative performance of the various competing problem parametrizations, error models and numerical methods. Key questions include: (*i*) how much difference does an *accurate statistical error model* make; (*ii*) which *constraint parametrizations*, *initialization methods* and *numerical optimization schemes* offer the best reliability/speed/simplicity. The answers are most interesting for *near-degenerate*

To appear in SMILE'98, European Workshop on 3D Structure from Multiple Images of Large-scale Environments, Springer-Verlag LNCS, 1998. [7/7/98]

problems, as these are the most difficult to handle reliably. This paper focuses on architectural, parametrization and numerical optimization issues. I have tried to give an overview of the relevant choices and technology, rather than going into too much detail on any one subject. The methods described extend to many other geometric estimation problems, such as curve and surface fitting.

After motivating the library and giving notation in this section, we develop a general statistical framework for geometric fitting in §2 and discuss parametrization issues in §3. §4 summarizes the library architecture and numerical techniques, §5 discusses experimental testing, and §6 concludes.

Why study matching constraint estimation? — Practically, matching constraints are central to both feature grouping and 3D reconstruction, so better algorithms should immediately benefit many geometric vision applications. But there are many variations to implement, depending on the feature type, number of images, image projection model, camera calibration, and camera and scene geometry. So a systematic approach seems more appropriate than an *ad hoc* case-by-case one. Matching constraints also have a rather delicate algebraic structure which makes them difficult to estimate accurately. Many common camera and scene geometries correspond to degenerate cases whose special properties need to be detected and exploited for stability. Even in stable cases it is not yet clear how best to parametrize the constraints — usually, they belong to fairly complicated algebraic varieties and redundant or constrained parametrizations are required. Some numerical sophistication is needed to implement these efficiently, and the advantages of different models and parametrizations need to be studied experimentally: the library is a vehicle for this.

It is also becoming clear that in many cases no single model suffices. One should rather think in terms of a continuum of nested models linked by specialization/generalization relations. For example, rather than simply assuming a generic fundamental matrix, one should use inter-image homographies for small camera motions or large flat scenes, affine fundamental matrices for small, distant objects, essential matrices for constant intrinsic parameters, fundamental matrices for wide views of large close objects, lens distortion corrections for real images, *etc.* Ideally, the model should be chosen to maximize the statistically expected end-to-end system performance, given the observed input data. Although there are many specific decision criteria (ML, AIC, BIC, ...), the key issue is always the *bias* of over-restrictive models versus the *variability* of over-general ones with superfluous parameters poorly controlled by the data. Any model selection approach requires several models to be fitted so that the best can be chosen. Some of the models must always be inappropriate — either biased or highly variable — so fast, reliable, accurate fitting in difficult cases is indispensable for practical model selection.

Terminology and notation: We use homogeneous coordinates throughout, with upright bold for 3D quantities and italic bold for image ones. Image projections are described by 3×4 perspective ***projection matrices*** $\boldsymbol{P}$, with specialized forms for calibrated or very distant cameras. Given m images of a static scene, our goal is to recover as much information as possible about the camera calibrations and poses, using only image measurements. We will call the recoverable information the ***inter-image geometry*** to emphasize that no explicit 3D structure is involved. The ensemble of projection matrices is defined only up to a 3D coordinate transformation (projectivity or similarity) $\mathbf{T}$: $(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_m) \to (\boldsymbol{P}_1\mathbf{T}, \ldots, \boldsymbol{P}_m\mathbf{T})$. We call such coordinate freedoms ***gauge freedoms***. So our first representation of the inter-image geometry is as ***projection matrices modulo a transformation group***. In the uncalibrated case this gives an $11m$ parameter representation with 15 gauge freedoms, leaving $11m - 15$ essential d.o.f. $(= 7, 18, 29$ for $m = 2, 3, 4)$. In the calibrated case there are $6m - 7$ essential degrees of freedom.

Any set of four (perhaps not distinct) projection matrices can be combined to form a ***matching tensor*** [14, 5] — a multi-image object independent of the 3D coordinates. The possible types are: ***epipoles*** $\boldsymbol{e}_i^j$; 3×3 ***fundamental matrices*** $\boldsymbol{F}_{ij}$; $3 \times 3 \times 3$ ***trifocal tensors*** $\boldsymbol{G}_i^{jk}$; and $3 \times 3 \times 3 \times 3$ ***quadrifocal tensors*** $\boldsymbol{H}^{ijkl}$. Their key property is that they are the coefficients of inter-image ***matching constraints*** — the consistency relations linking corresponding features in different images. *E.g.*, for images $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}''$ of a 3D point we have the 2-image ***epipolar constraint*** $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{F}\, \boldsymbol{x}' = 0$; the 3-image ***trinocular constraint*** which can be written symbolically as $[\,\boldsymbol{x}'\,]_\times (\boldsymbol{G} \cdot \boldsymbol{x})\, [\,\boldsymbol{x}''\,]_\times = \boldsymbol{0}$ where $[\,\boldsymbol{x}\,]_\times$ is the matrix generating the cross product $[\,\boldsymbol{x}\,]_\times \boldsymbol{y} \equiv \boldsymbol{x} \wedge \boldsymbol{y}$; and a 4-image ***quadrinocular constraint***. The matching tensors also characterize the inter-image geometry. This is attractive because they are intimately connected to the image measurements — it is much easier to get linearized initial estimates of matching tensors than of projection matrices. Unfortunately, this linearity is deceptive. Matching tensors are not really linear objects: they only represent a valid, realizable inter-image geometry if they satisfy a set of nonlinear algebraic ***consistency constraints***. These rapidly become intractable beyond 2–3 images, and are still only partially understood [4, 14, 5, 9, 6]. Our second parametrization of the inter-image geometry is as *matching tensors subject to consistency constraints*.

We emphasize that camera matrices or matching tensors are only a means to an end: it is the underlying inter-image geometry that we are really trying to estimate. Unfortunately, this is abstract and somewhat difficult to pin down because it is a ***nontrivial algebraic variety*** — there *are* no simple, minimal, global parametrizations.

2 Optimal Geometric Fitting

2.1 Direct Approach

Matching constraint estimation is an instance of an ***abstract geometric fitting problem*** which also includes curve and surface fitting and many other geometric estimation problems: estimate the parameters of a model $\mathbf{u}$ defining implicit constraints $\mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0}$ on underlying features $\mathbf{x}_i$, from noisy measurements of the features. More specifically we assume:

1. There are unknown ***true underlying features*** $\overline{\mathbf{x}}_i$ and an unknown ***true underlying model*** $\overline{\mathbf{u}}$ which exactly satisfy implicit ***model-feature consistency constraints*** $\mathbf{c}_i(\overline{\mathbf{x}}_i, \overline{\mathbf{u}}) = \mathbf{0}$. (For matching constraint estimation, these 'features' are actually ensembles of several corresponding image ones).
2. Each underlying feature $\overline{\mathbf{x}}_i$ is linked to observations $\underline{\mathbf{x}}_i$ or other prior information by an additive ***posterior statistical error measure*** $\rho_i(\mathbf{x}_i) \equiv \rho_i(\mathbf{x}_i \,|\, \underline{\mathbf{x}}_i)$. For example, ρ_i might be (robustified, bias corrected) ***posterior log likelihood***. There may also be a ***model prior*** $\rho_{\text{prior}}(\mathbf{u})$. These distributions are independent.
3. The model parametrization $\mathbf{u}$ may itself be complex, *e.g.* with internal constraints $\mathbf{k}(\mathbf{u}) = \mathbf{0}$, gauge freedoms, *etc.*
4. We want to find ***optimal consistent point estimates*** $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}})$ of the true underlying model $\overline{\mathbf{u}}$ and features $\overline{\mathbf{x}}_i$

$$(\hat{\mathbf{x}}_i, \ldots, \hat{\mathbf{u}}) \;\equiv\; \arg\min \left(\rho_{\text{prior}}(\mathbf{u}) + \sum_i \rho_i(\mathbf{x}_i|\underline{\mathbf{x}}_i) \;\middle|\; \mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0},\, \mathbf{k}(\mathbf{u}) = \mathbf{0} \right)$$

   Consistent means that $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}})$ exactly satisfy all the constraints. ***Optimal*** means that they minimize the total error over all such estimates. ***Point estimate*** means that we are attempting to "summarize" the joint posterior distribution $\rho(\mathbf{x}_i, \ldots, \mathbf{u}|\underline{\mathbf{x}}_i, \ldots)$ with just the few numbers $(\hat{\mathbf{x}}_i, \ldots, \hat{\mathbf{u}})$.

We call this the ***direct approach*** to geometric fitting because it involves direct numerical optimization over the "natural" variables $(\mathbf{x}_i, \mathbf{u})$. Its most important characteristics are: (i) It gives exact, optimal results — no approximations are involved. (ii) It produces optimal consistent estimates $\hat{\mathbf{x}}_i$ of the underlying features $\overline{\mathbf{x}}_i$. These are useful whenever the measurements need to be made coherent with the model. For matching constraint estimation such feature estimates are "pre-triangulated" or "implicitly reconstructed" in that they have already been made exactly consistent with exactly one reconstructed 3D feature. (iii) Natural variables are used and the error function is relatively simple, typically just a sum of (robustified, covariance weighted) squared deviations $\|\mathbf{x}_i - \underline{\mathbf{x}}_i\|^2$. (iv) However, a sparse constrained nonlinear optimization routine is required: the problem is large, constrained and usually nonlinear, but the features couple only to the model, not to each other.

As an example, for the uncalibrated epipolar geometry: the "features" are pairs of corresponding underlying image points $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$; the "model" $\mathbf{u}$ is the fundamental matrix $\boldsymbol{F}$ subject to the consistency constraint $\det(\boldsymbol{F}) = 0$; the "model-feature constraints" are the epipolar constraints $\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{F}\, \boldsymbol{x}'_i = 0$; and the "feature error model" $\rho_i(\mathbf{x}_i)$ might be (a robustified, covariance-weighted variant of) the squared feature-observation distance $\|\boldsymbol{x} - \underline{\boldsymbol{x}}\|^2 + \|\boldsymbol{x}' - \underline{\boldsymbol{x}}'\|^2$.

2.2 Reduced Approach

If explicit estimates of the underlying features are not required, one can attempt to replace step 4 above with an optimization over $\mathbf{u}$ alone:

4′. Find an ***optimal consistent point estimate*** $\hat{\mathbf{u}}$ of the true underlying model $\overline{\mathbf{u}}$

$$\hat{\mathbf{u}} \;\equiv\; \arg\min \left(\rho_{\text{prior}}(\mathbf{u}) + \sum_i \rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \;\middle|\; \mathbf{k}(\mathbf{u}) = \mathbf{0} \right)$$

Here, the ***reduced error functions*** $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i)$ are obtained by freezing $\mathbf{u}$ and eliminating the unknown features from the problem using either: (i) ***point estimates*** $\mathbf{x}_i(\underline{\mathbf{x}}_i, \mathbf{u}) \;\equiv\; \arg\min\left(\rho_i(\mathbf{x}_i|\underline{\mathbf{x}}_i) \,|\, \mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0}\right)$ of $\mathbf{x}_i$ given $\underline{\mathbf{x}}_i$ and $\mathbf{u}$, with $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \;\equiv\; \rho_i(\mathbf{x}_i(\underline{\mathbf{x}}_i, \mathbf{u})|\underline{\mathbf{x}}_i)$; or (ii) ***marginalization*** with respect to $\mathbf{x}_i$: $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \;\equiv\; \int_{\mathbf{c}_i(\mathbf{x}_i,\mathbf{u})=0} \rho_i(\mathbf{x}_i|\underline{\mathbf{x}}_i)\, \mathbf{dx}_i$. These two methods are not equivalent in general, although their answers happen to agree in the linear/Gaussian limit. But both represent reasonable estimation techniques.

We call this the ***reduced approach*** to geometric fitting, because the problem is ***reduced*** to one involving only the model parameters $\mathbf{u}$. The main advantage is that the optimization is over relatively few variables $\mathbf{u}$. The constraints $\mathbf{c}_i$ do not appear, so a non-sparse and (perhaps) unconstrained optimization routine can be used. The disadvantage is that the reduced cost $\rho(\mathbf{u})$ is seldom available in closed form. Usually, it can only be evaluated to first order in a linearized + central distribution approximation. In fact, the direct method (with $\mathbf{u}$ frozen, and perhaps limited to a single iteration) is often the easiest way to evaluate the point-estimate-based reduced cost. The only real difference is that the direct method explicitly calculates and applies feature updates $\mathbf{dx}_i$, while the reduced method restarts each time from $\mathbf{x}_i \equiv \underline{\mathbf{x}}_i$. But the feature updates are relatively easy to calculate given the factorizations needed for cost evaluation, so it seems a pity not to use them.

The first order reduced cost can be estimated in two ways, either (i) directly from the definition by projecting $\underline{\mathbf{x}}_i$ Mahalanobis-orthogonally onto the local first-order constraint surface $\mathbf{c}_i + \frac{\mathbf{dc}_i}{\mathbf{dx}_i} \cdot \mathbf{dx}_i = \mathbf{0}$; or (ii) by treating $\mathbf{c}_i \equiv \mathbf{c}_i(\underline{\mathbf{x}}_i, \mathbf{u})$ as a random variable, using covariance propagation w.r.t. $\underline{\mathbf{x}}_i$ to find its covariance, and calculating the χ^2-like variable $\mathbf{c}_i^{\top} \mathrm{Cov}(\mathbf{c}_i)^{-1} \mathbf{c}_i$. In either case we obtain the ***gradient weighted least squares*** cost function[1] [13]

$$\rho(\mathbf{u}) \;=\; \sum_i \mathbf{c}_i^{\top} \left(\frac{\mathbf{dc}_i}{\mathbf{dx}_i} \left(\frac{\mathbf{d}^2\rho_i}{\mathbf{dx}_i^2} \right)^{-1} \frac{\mathbf{dc}_i}{\mathbf{dx}_i}^{\top} \right)^{-1} \mathbf{c}_i \;\;\Bigg|_{(\underline{\mathbf{x}}_i, \mathbf{u})}$$

This is simplest for problems with scalar constraints. *E.g.* for the uncalibrated epipolar constraint we get the well-known form [10]

$$\rho(\mathbf{u}) \;=\; \sum_i \frac{(\underline{\boldsymbol{x}}_i^{\top}\, \boldsymbol{F}\, \underline{\boldsymbol{x}}_i')^2}{\underline{\boldsymbol{x}}_i^{\top}\, \boldsymbol{F}\, \mathrm{Cov}(\underline{\boldsymbol{x}}_i')\, \boldsymbol{F}^{\top}\, \underline{\boldsymbol{x}}_i + \underline{\boldsymbol{x}}_i'^{\top}\, \boldsymbol{F}^{\top}\, \mathrm{Cov}(\underline{\boldsymbol{x}}_i)\, \boldsymbol{F}\, \underline{\boldsymbol{x}}_i'}$$

[1] If any of the covariance matrices is singular (which happens for redundant constraints or homogeneous data $\mathbf{x}_i$), the matrix inverses can be replaced with pseudo-inverses.

2.3 Robustification — Total Distribution Approach

Outliers are omnipresent in vision data and it is essential to protect against them. In general, they are distinguished only by their failure to agree with the consensus established by the inliers, so one should really think in terms of *inlier* or *coherence* detection. The hardest part is establishing a reliable initial estimate, *i.e.* the combinatorial problem of finding enough inliers to estimate the model, without being able to tell in advance that they *are* inliers. Exhaustive enumeration is usually impracticable, so one falls back on either RANSAC-like random sampling or (in low dimensions) Hough-like voting. Initialization from an outlier-polluted linear estimate is seldom completely reliable.

Among the many approaches to robustness, I prefer M-like estimators and particularly the ***total distribution*** approach: hypothesize a parametric form for the ***total observation distribution*** — *i.e.* including *both* inliers *and* outliers — and fit this to the data using some standard criterion, *e.g.* maximum likelihood. No explicit inlier/outlier decision is needed: the correct model is located simply because it provides an explanation more probable than randomness for the coherence of the inliers[2]. The total approach is really just classical parametric statistics with a more realistic or "robust" choice of parametric family. Any required distribution parameters can in principle be estimated during fitting (*e.g.* covariances, outlier densities). For centrally peaked mixtures one can view the total distribution as a kind of M-estimator, although it long predates these and gives a much clearer meaning to the rather arbitrary functional forms usually adopted for them. As with other M-like-estimators, the estimation problem is nonlinear and numerical optimization is required. With this approach, both of the above geometric fitting methods are 'naturally' robust — we just need to use an appropriate total likelihood.

Reasons for preferring M-like estimators over trimmed ones like RANSAC's consensus and rank-based ones like least median squares include: (*i*) to the extent that the total distribution is realistic, the total approach is actually the statistically optimal one; (*ii*) only M-like cost functions are smooth and hence easy to optimize; (*iii*) the 'soft' transitions of M-like estimators allow better use of weak 'near outlier' data, *e.g.* points which are relatively uncertain owing to feature extraction problems, or "false outliers" caused by misestimated covariances or a skewed, biased, or badly initialized model; (*iv*) including an explicit covariance scale makes the results more reliable and increases the *expected* breakdown point — 'scale free' rank based estimators can not tell whether the measurements they are including are "plausible" or not; (*v*) all of these estimators assume an underlying ranking of errors 'by relative size', and none are robust against mismodelling of this — rank based estimators only add a little extra robustness against the likelihood *vs.* error size assignment.

[2] If the total distribution happens to be an inlier/outlier *mixture* — *e.g.* Gaussian peak + uniform background — posterior inlier/outlier probabilities are easily extracted as a side effect.

3 Parametrizing the Inter-image Geometry

As discussed above, what we are really trying to estimate is the ***inter-image geometry*** — the part of the multi-camera calibration and pose that is recoverable from image measurements alone. However, this is described by a nontrivial algebraic variety — it has *no* simple, minimal, concrete, global parametrization. For example, the uncalibrated epipolar geometry is “the variety of all homographic mappings between line pencils in the plane”, but it is unclear how best to parametrize this. We will consider three general parametrization strategies for algebraic varieties: (i) redundant parametrizations with internal gauge freedoms; (ii) redundant parametrizations with internal constraints; (iii) overlapping local coordinate patches. *Mathematically* these are all equivalent — they only differ in relative convenience and numerical properties. Different methods are convenient for different uses, so it is important to be able to convert between them. Even the numerical differences are slight for strong geometries and careful implementations, but for weak geometries there can be significant differences.

3.1 Redundant Parametrizations with Gauge Freedom

In many geometric problems, ***arbitrary choices of coordinates*** are required to reduce the problem to a concrete algebraic form. Such choices are called ***gauge freedoms*** — ‘gauge’ just means coordinate system. They are associated with an internal ***symmetry*** or ***coordinate transformation*** group and its representations. Formulae expressed in gauged coordinates reflect the symmetry by obeying well-defined transformation rules under changes of coordinates, *i.e.* by belonging to well-defined group representations. 3D Cartesian coordinates are a familiar example: the gauge group is the group of rigid motions, and the representations are (roughly speaking) Cartesian tensors.

Common gauge freedoms include: (i) 3D projective or Euclidean coordinate freedoms in reconstruction and projection-matrix-based camera parametrizations; (ii) arbitrary homogeneous-projective scale factors; and (iii) choice-of-plane freedoms in ***homographic parametrizations*** of the inter-image geometry. These latter represent matching tensors as products of epipoles and inter-image homographies induced by an arbitrary 3D plane. The gauge freedom is the 3 d.o.f. choice of plane. The fundamental matrix can be written as $\boldsymbol{F} \simeq [\,\boldsymbol{e}\,]_\times \boldsymbol{H}$ where $\boldsymbol{e}$ is the epipole and $\boldsymbol{H}$ is any inter-image homography [11,3]. Redefining the 3D plane changes $\boldsymbol{H}$ to $\boldsymbol{H} + \boldsymbol{e}\,\boldsymbol{a}^{\mathrm{T}}$ for some image line 3-vector $\boldsymbol{a}$. This leaves $\boldsymbol{F}$ unchanged, as do rescalings $\boldsymbol{e} \to \lambda \boldsymbol{e}$, $\boldsymbol{H} \to \mu \boldsymbol{H}$. So there are $3+1+1$ gauge freedoms in the $3 + 3 \times 3 = 12$ variable parametrization $\boldsymbol{F} \simeq \boldsymbol{F}(\boldsymbol{e}, \boldsymbol{H})$, leaving the correct $12 - 5 = 7$ degrees of freedom of the uncalibrated epipolar geometry. Similarly [8], the image $(1,2,3)$ trifocal tensor $\boldsymbol{G}$ can be written in terms of the epipoles $(\boldsymbol{e}', \boldsymbol{e}'')$ and inter-image homographies $(\boldsymbol{H}', \boldsymbol{H}'')$ of image 1 in images 2 and 3

$$\boldsymbol{G} \simeq \boldsymbol{e}' \otimes \boldsymbol{H}'' - \boldsymbol{H}' \otimes \boldsymbol{e}'' \quad \text{with freedom} \quad \left(\begin{smallmatrix} \boldsymbol{H}' \\ \boldsymbol{H}'' \end{smallmatrix}\right) \to \left(\begin{smallmatrix} \boldsymbol{H}' \\ \boldsymbol{H}'' \end{smallmatrix}\right) + \left(\begin{smallmatrix} \boldsymbol{e}' \\ \boldsymbol{e}'' \end{smallmatrix}\right) \boldsymbol{a}^{\mathrm{T}}$$

The gauge freedom corresponds to the choice of 3D plane and 3 scale d.o.f. — the relative scaling of $(\boldsymbol{e}', \boldsymbol{H}')$ *vs.* $(\boldsymbol{e}'', \boldsymbol{H}'')$ being significant — so the 18 d.o.f. of the uncalibrated trifocal geometry are parametrized by $3+3+9+9 = 24$ parameters modulo $3+1+1+1 = 6$ gauge freedoms. For calibrated cameras it is useful to place the 3D plane at infinity so that the resulting absolute homographies are represented by 3×3 rotation matrices. This gives well-known 6 and 12 parameter representations of the calibrated epipolar and trifocal geometries, each with just one redundant scale d.o.f.: $\boldsymbol{E} \simeq [\,\boldsymbol{e}\,]_\times \boldsymbol{R}$, $\boldsymbol{G} \simeq \boldsymbol{e}' \otimes \boldsymbol{R}'' - \boldsymbol{R}' \otimes \boldsymbol{e}''$. All of these homography + epipole parametrizations can also be viewed as projection matrix based ones, in a 3D frame where the first projection takes the form $(\boldsymbol{I}_{3\times 3}|\boldsymbol{0})$. The plane position freedom $\boldsymbol{a}$ corresponds to the 3 remaining d.o.f. of the 3D projective frame [8]. These methods seem to be a good compromise: compared to 'free' projections, they reduce the number of extraneous d.o.f. from 15 to 3. However their numerical stability does depend on that of the key image.

Gauged parametrizations have the following advantages: (i) they are very natural when the inter-image geometry is derived from the 3D one; (ii) they are close to the underlying geometry, so it is relatively easy to derive further properties from them (projection matrices, reconstruction methods, matching tensors); (iii) a single homogeneous coordinate system covers the whole variety; (iv) they are numerically fairly stable. Their main disadvantage is that they include extraneous, strictly irrelevant degrees of freedom which have no effect at all on the residual error. Hence, gauged Jacobians are exactly rank deficient: specially stabilized numerical methods are needed to handle them. The additional variables and stabilization also tend to make gauged parametrizations slow.

3.2 Constrained Parametrizations

Another way to define a variety is in terms of ***consistency constraints*** that "cut the variety out of" a larger, usually linear space. Any coordinate system in the larger space then parametrizes the variety, but this is an over-parametrization subject to nonlinear constraints. Points which fail to satisfy the constraints have no meaning in terms of the variety. ***Matching tensors*** are the most familiar example. In the 2- and 3-image cases a single fundamental matrix or trifocal tensor suffices to characterize the inter-image geometry. But this is a linear over-parametrization, subject to the tensor's nonlinear consistency constraints — only so is a coherent, realizable inter-image geometry represented. Such parametrizations are valuable because they are close to the image data, and (inconsistent!) linear initial estimates of the tensors are easy to obtain. Their main disadvantages are: (i) the consistency conditions rapidly become complicated and non-obvious; (ii) the representation is only implicit — it is not immediately obvious how to go from the tensor to other properties of the geometry such as projection matrices. The first problem is serious and puts severe limitations on the use of (ensembles of) matching tensors to represent camera geometries, even in transfer-type applications where explicit projection matrices are not required. Three images seems to be about the practical limit if a guaranteed-consistent geometry is required, although — at the peril of a build-up of rounding error — one can chain together a series of such three image solutions [12, 15, 1].

For the fundamental matrix the codimension is 1 and the consistency constraint is $\det(\boldsymbol{F}) = 0$ — this is perhaps the simplest of all representations of the uncalibrated epipolar geometry. For the essential matrix $\boldsymbol{E}$ the codimension is 3, spanned either by the requirement that $\boldsymbol{E}$ should have two equal (which counts for 2) and one zero singular values, or by a local choice of 3 of the 9 Demazure constraints $(\boldsymbol{E}\boldsymbol{E}^{\top} - \frac{1}{2}\mathrm{trace}(\boldsymbol{E}\boldsymbol{E}^{\top}))\ \boldsymbol{E} = \mathbf{0}$ [4]. For the uncalibrated trifocal tensor $\boldsymbol{G}$ we locally need $26 - 18 = 8$ linearly independent constraints. Locally (only!) these can be spanned by the 10 determinantal constraints $\frac{\mathbf{d}^3}{\mathbf{dx}^3}\det(\boldsymbol{G}\cdot\mathbf{x}) = \boldsymbol{0}$ — see [6] for several global sets. For the quadrifocal tensor $\boldsymbol{H}$ the codimension is $80 - 29 = 51$ which is locally (but almost certainly not globally) spanned by the $3!\cdot 3\cdot 3 = 54$ determinantal constraints $\det_{ij}(\boldsymbol{H}^{ijkl}) = \boldsymbol{0}$ + permutations.

Note that the redundancy and complexity of the matching tensor representation rises rapidly as more images or calibration constraints are added. Also, ***constraint redundancy*** is common. Many algebraic varieties require a number of generators greater than their codimension. Intersections of the minimal number of polynomials *locally* give the correct variety, but typically have other, unwanted components elsewhere in the space. Extra polynomials must be included to suppress these, and it rapidly becomes difficult to say which sets of polynomials are globally sufficient.

3.3 Local Coordinate Patches / Minimal Parametrizations

Both gauged and constrained parametrizations are redundant and require specialized numerical methods. Why not simplify life by using a ***minimal set of independent parameters***? — The basic problem is that no such parametrization can cover the whole of a topologically nontrivial variety without singularities. Minimal parametrizations are intrinsically *local*: to cover the whole variety we need several such partially overlapping 'local coordinate patches', and also code to select the appropriate patch and manage any inter-patch transitions that occur. This can greatly complicate the optimization loop.

Also, although infinitely many local parametrizations exist, they are not usually very 'natural' and finding one with good properties may not be easy. Basically, starting from some 'natural' redundant representation, we must either come up with some inspired nonlinear change of variables which locally removes the redundancy, or algebraically eliminate variables by brute force using consistency or gauge fixing constraints. For example, Luong *et al* [10] guarantee $\det(\boldsymbol{F}) = 0$ by writing each row of the fundamental matrix as a linear combination of the other two. Each parametrization fails when its two rows are linearly dependent, but the three of them suffice to cover the whole variety. In more complicated situations, intuition fails and we have to fall back on algebraic elimination, which rapidly leads to intractable results. Elimination-based parametrizations are usually highly anisotropic: they do not respect the symmetries of the underlying geometry. This tends to mean that they are messy to implement, and numerically ill-behaved, particularly near the patch boundaries.

The above comments apply only to *algebraically* derived parametrizations. Many of the numerical techniques for gauged or constrained problems eliminate redundant variables *numerically* to first order, using the constraint Jacobians. Such local parametrizations are much better behaved because they are always used at the centre of their valid region, and because stabilizing techniques like pivoting can be used. *It is usually preferable to eliminate variables locally and numerically rather than algebraically.*

4 Library Architecture and Numerical Methods

The library is designed to be modular so that different problems and approaches are easy to implement and compare. We separate: (*i*) the matching geometry type and parametrization; (*ii*) each contributing feature-group type, parametrization and error model; (*iii*) the numerical optimization method, and its associated linear algebra; (*iv*) the search controller (step acceptance and damping, convergence tests). This decomposition puts some constraints on the types of algorithms that can be implemented, but these do not seem to be too severe in practice. Modularization also greatly simplifies the implementation.

Perhaps the most important assumption is the adoption throughout of a "square root" or normalized residual vector based framework, and the associated use of Gauss-Newton techniques. ***Normalized residual vectors*** are quantities $\mathbf{e}_i$ for which the squared norm $\|\mathbf{e}_i\|^2$ — or more generally a robust, nonlinear function $\rho_i(\|\mathbf{e}_i\|^2)$ — is a meaningful statistical error measure. *E.g.* $\mathbf{e}_i(\mathbf{x}_i) \equiv \mathrm{Cov}(\underline{\mathbf{x}}_i)^{-\frac{1}{2}}(\mathbf{x}_i - \underline{\mathbf{x}}_i)$. This allows a nonlinear-least-squares-like approach. Whenever possible, we work directly with the residual $\mathbf{e}$ and its Jacobian $\frac{\mathbf{de}}{\mathbf{dx}}$ rather than with $\|\mathbf{e}\|^2$, its gradient $\frac{\mathbf{d}(\|\mathbf{e}\|^2)}{\mathbf{dx}} = \mathbf{e}^\top \frac{\mathbf{de}}{\mathbf{dx}}$ and its Hessian $\frac{\mathbf{d}^2(\|\mathbf{e}\|^2)}{\mathbf{dx}^2} = \mathbf{e}^\top \frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2} + \frac{\mathbf{de}}{\mathbf{dx}}^\top \frac{\mathbf{de}}{\mathbf{dx}}$. We use the ***Gauss-Newton approximation***, *i.e.* we discard the second derivative term $\mathbf{e}^\top \frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2}$ in the Hessian. This buys us simplicity (no second derivatives are needed) and also numerical stability because we can use stable ***linear least squares*** methods for step prediction: by default we use ***QR decomposition with column pivoting*** of $\frac{\mathbf{de}}{\mathbf{dx}}$, rather than Cholesky decomposition of the normal matrix $\frac{\mathbf{de}}{\mathbf{dx}}^\top \frac{\mathbf{de}}{\mathbf{dx}}$. This is potentially slightly slower, but for ill-conditioned Jacobians it has much better resistance to rounding error. (The default implementation is intended for use as a reference, so it is deliberately rather conservative). The main disadvantage of Gauss-Newton is that convergence may be slow if the problem has both *large residual* and *strong nonlinearity* — *i.e.* if the ignored Hessian term $\mathbf{e}^\top \frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2}$ is large. However, *geometric vision problems usually have small residuals* — the noise is usually much smaller than the scale of the geometric nonlinearities.

4.1 Numerical Methods for Gauge Freedom

The basic numerical difficulty with gauge freedom is that because gauge motions represent exact redundancies that have no effect at all on the residual error, in a classical optimization framework there is nothing to say what they should be: the error gradient and Hessian in a gauge direction both vanish, so the Newton step is undefined. If left undamped, this leads to ***large gauge fluctuations*** which can destabilize the rest of the system, prevent convergence tests from operating, *etc.* There are two ways around this problem:

1. Gauge fixing conditions break the degeneracy by adding ***artificial constraints***. Unless we are clever enough to choose constraints that eliminate variables in closed form, this reduces the problem to constrained optimization. The constraints are necessarily non-gauge-invariant, *i.e.* non-tensorial under the gauge group. For example, to fix the 3D projective coordinate freedom, Hartley [8] sets $\boldsymbol{P}_1 \equiv (\boldsymbol{I}_{3\times3}|\boldsymbol{0})$ and $\sum_i \boldsymbol{e}^i \boldsymbol{H}^i_j = 0$ where $\boldsymbol{P}_2 = (\boldsymbol{H}|\boldsymbol{e})$. Neither of these constraints is tensorial — the results depend on the chosen image coordinates.

2. Free gauge methods — like photogrammetric ***free bundle*** ones — leave the gauge free to drift, but ensure that it does not move too far at each step. Typically, it is also monitored and reset "by hand" when necessary to ensure good conditioning. The basic tools are ***rank deficient least squares*** methods (*e.g.* [2]). These embody some form of damping to preclude large fluctuations in near-deficient directions. The popular ***regularization*** method minimizes $\|\text{residual}\|^2 + \lambda^2\|\text{step size}\|^2$ for some small $\lambda > 0$ — an approach that fits very well with Levenberg-Marquardt-like search control schemes. Alternatively, a ***basic solution*** — a solution where certain uncontrolled components are set to zero — can be calculated from a standard pivoted QR or Cholesky decomposition, simply by ignoring the last few (degenerate) columns. One can also find vectors spanning the local gauge directions and treat them as 'virtual constraints' with zero residual, so that the gauge motion is locally zeroed. ***Householder reduction***, which orthogonalizes the rows of $\frac{\mathbf{de}}{\mathbf{dx}}$ w.r.t. the gauge matrix by partial QR decomposition, is a nice example of this.

4.2 Numerical Methods for Constrained Optimization

There are at least three ways to handle linear constraints numerically: (*i*) ***eliminate variables*** using the constraint Jacobian; (*ii*) introduce ***Lagrange multipliers*** and solve for these too; (*iii*) ***weighting methods*** treat the constraints as heavily weighted residual errors. Each method has many variants, depending on the matrix factorization used, the ordering of operations, *etc.* As a rough rule of thumb, for dense problems variable elimination is the fastest and stablest method, but also the most complex. Lagrange multipliers are slower because there are more variables. Weighting is simple, but slow and inexact — stable orthogonal decompositions are needed as weighted problems are ill-conditioned.

For efficiency, direct geometric fitting requires a sparse implementation — the features couple to the model, but not to each other. The above methods all extend to sparse problems, but the implementation complexity increases by about one order of magnitude in each case. My initial implementation [16] used Lagrange multipliers and Cholesky decomposition, but I currently prefer a stabler, faster 'multifrontal QR' elimination method. There is no space for full details here, but it works roughly as follows (NB: the implementation orders the steps differently for efficiency): For each constrained system, the constraint Jacobian $\frac{\mathbf{dc}}{\mathbf{dx}}$ is factorized and the results are propagated to the error Jacobian $\frac{\mathbf{de}}{\mathbf{dx}}$. This eliminates the $\dim(\mathbf{c})$ variables best controlled by the constraints from $\frac{\mathbf{de}}{\mathbf{dx}}$, leaving a 'reduced' $\dim(\mathbf{e}) \times (\dim(\mathbf{x}) - \dim(\mathbf{c}))$ least squares problem. Many factorization methods can be used for the elimination and the reduced problem. I currently use column pivoted QR decomposition for both, which means that the elimination step is essentially Gaussian elimination. All this is done for each feature system. The elimination also carries the $\frac{\mathbf{dc}}{\mathbf{du}}$ columns into the reduced system. The residual error of the reduced system can not be reduced by changing $\mathbf{x}$, but it is affected by changes in $\mathbf{u}$ acting via these reduced $\frac{\mathbf{dc}}{\mathbf{du}}$ columns, which thus give contributions to an effective reduced error Jacobian $\frac{\mathbf{de(u)}}{\mathbf{du}}$ for the model $\mathbf{u}$. (This is the reduced geometric fitting method's error function). The resulting model system is reduced against any model constraints and factorized by pivoted QR. Back-substitution through the various stages then gives the required model update and finally the feature updates.

4.3 Search Control

All of the above techniques are linear. For nonlinear problems they must be used in a loop with appropriate step damping and search control strategies. This has been an unexpectedly troublesome part of the implementation — there seems to be a lack of efficient, reliable search control heuristics for constrained optimization. The basic problem is that the dual goals of reducing the constraint violation and reducing the residual error often conflict, and it is difficult to find a compromise that is good in all circumstances. Traditionally, a ***penalty function*** [7] is used, but all such methods have a 'stiffness' parameter which is difficult to set — too weak and the constraints are violated, too strong and the motion along the constraints towards the cost minimum is slowed. Currently, rather than a strict penalty function, I use a heuristic designed to allow a reasonable amount of 'slop' during motions along the constraints. The residual/constraint conflict also affects ***step damping*** — the control of step length to ensure acceptable progress. The principle of a ***trust region*** — a dynamic local region of the search space where the local function approximations are thought to hold good — applies, but interacts badly with ***quadratic programming*** based step prediction routines which try to satisfy the constraints exactly no matter how far away they are. Existing heuristics for this seemed to be poor, so I have developed a new 'dual control' strategy which damps the towards-constraint and along-constraint parts of the step separately using two Levenberg-Marquardt parameters linked to the same trust region.

Another difficulty is ***constraint redundancy***. Many algebraic varieties require a number of generators greater than their codimension to eliminate spurious components elsewhere in the space. The corresponding constraint Jacobians theoretically have rank = codimension on the variety, but usually rank > codimension away from it. Numerically, a reasonably complete and well-conditioned set of generators is advisable to reduce the possibility of convergence to spurious solutions, but the high degree of rank degeneracy on the variety, and the rank transition as we approach it, are numerically troublesome. Currently, my only effective way to handle this is to assume known codimension r and numerically project out and enforce only the r strongest constraints at each iteration. This is straightforward to do during the constraint factorization step, once r is known. As examples: the trifocal point constraints $[\,\boldsymbol{x}'\,]_\times(\boldsymbol{G}\cdot\boldsymbol{x})[\,\boldsymbol{x}''\,]_\times = \boldsymbol{0}$ have rank 4 in $(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'')$ for most invalid tensors, but only rank 3 for valid ones; and the trifocal consistency constraints $\frac{\mathbf{d}^3}{\mathbf{d}\boldsymbol{x}^3}\det(\boldsymbol{G}\cdot\boldsymbol{x}) = \boldsymbol{0}$ have rank 10 for most invalid tensors, but only rank 8 for valid ones. In both cases, overestimating the rank causes severe ill-conditioning.

4.4 Robustification

We assume that each feature has a ***central*** robust cost function $\rho_i(\mathbf{x}_i) \equiv \rho_i(\|\mathbf{e}_i(\mathbf{x}_i)\|^2)$ defined in terms of a covariance-weighted ***normalized residual error*** $\mathbf{e}_i(\mathbf{x}_i) \equiv \mathbf{e}_i(\mathbf{x}_i|\underline{\mathbf{x}}_i)$. This defines the 'granularity' — entire 'features' (for matching constraints, ensembles of corresponding image features) are robustified, not their individual components. The robust cost ρ_i is usually some M-estimator, often a total log likelihood. For a uniform-outlier-polluted Gaussian it has the form $\rho(z) \equiv -2\log\left(e^{-z/2} + \beta\right)$, where β is related to outlier density. Typically, $\rho(z)$ is linear near 0, monotonic but sublinear for $z > 0$ and tends to a constant at $z \to \infty$ if distant outliers have vanishing influence. Hence, $\rho' \equiv \frac{d\rho}{dz}$ decreases monotonically to 0 and $\rho'' \equiv \frac{d^2\rho}{dz^2}$ is negative.

Robustification can lead to numerical problems, so care is needed. Firstly, since the cost is often nonconvex for outlying points, strong regularization may be required to guarantee a positive Hessian and hence a cost reducing step. This can slow convergence. To partially compensate for this curvature, and to allow us to use a 'naïve' Gauss-Newton step calculation while still accounting for robustness, we define a weighted, rank-one-corrected ***effective residual*** $\tilde{\mathbf{e}} \equiv \frac{\sqrt{\rho'}}{1-\alpha}\mathbf{e}$ and ***effective Jacobian*** $\widetilde{\frac{\mathbf{de}}{\mathbf{dx}}} \equiv \sqrt{\rho'}\,(\mathbf{I} - \frac{\alpha}{\|\mathbf{e}\|^2}\,\mathbf{e}\,\mathbf{e}^\top)\frac{\mathbf{de}}{\mathbf{dx}}$ where $\alpha \equiv \mathrm{RootOf}(\frac{1}{2}\alpha^2 - \alpha - \frac{\rho''}{\rho'}\|\mathbf{e}\|^2)$. These definitions ensure that to second order in ρ and $\mathbf{dx}$ and up to an irrelevant constant, the true robust cost $\rho(\|\mathbf{e} + \frac{\mathbf{de}}{\mathbf{dx}}\mathbf{dx}\|^2)$ is the same as the naïve effective squared error $\|\tilde{\mathbf{e}} + \widetilde{\frac{\mathbf{de}}{\mathbf{dx}}}\mathbf{dx}\|^2$. *I.e.* the same step $\mathbf{dx}$ is generated, so if we use effective quantities, we need think no further about robustness[3]. Here the $\sqrt{\rho'}$ weighting is the first order correction, and the α terms are the second order one. Usually $\rho' \to 0$ for distant outliers. Since the whole feature system is scaled by $\sqrt{\rho'}$, this might cause numerical conditioning or scaling problems in the direct method. To avoid this, we actually apply the $\sqrt{\rho'}$-weighting at the last possible moment — the contribution of the feature to the model error — and leave the feature systems themselves unweighted.

[3] If $\frac{\rho''}{\rho'}\|\mathbf{e}\|^2 < -\frac{1}{2}$ the robust Hessian has negative curvature and there is no real solution for α. In practice we limit $\alpha < 1 - \epsilon$ to prevent too much ill-conditioning. We would have had to regularize this case away anyway, so nothing is lost.

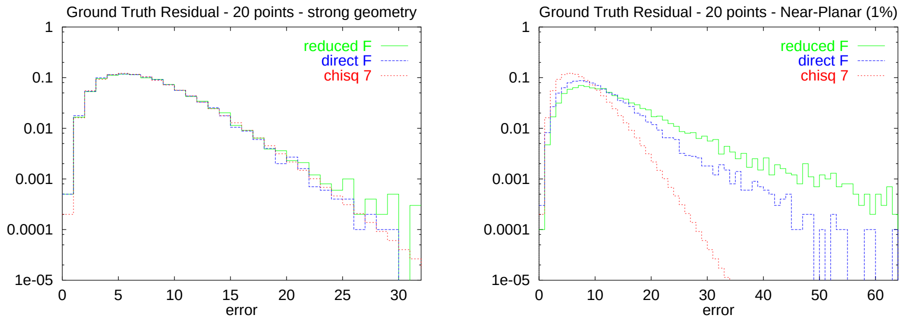

Fig. 1. Ground feature residuals for strong and near-coplanar epipolar geometries.

5 Measuring Performance

We currently test mainly on synthetic data, to allow systematic comparisons over a wide range of problems. We are particularly concerned with verifying theoretical statistical performance bounds, as these are the best guarantee that we are doing as well as could reasonably be expected. Any tendency to return occasional outliers is suspect and needs to be investigated. Histograms of the ***ground-truth-feature residual*** (GFR) have proven particularly useful for this. These plot frequency *vs.* size of the total squared deviation of the *ground truth* values of the noisy features used in the estimate, from the estimated matching relations. This measures how *consistent* the estimated geometry is with the underlying noise-free features. For weak feature sets the geometry might still be far from the true one, but consistency is the most we can expect given the data. In the linear approximation the GFR is χ^2_ν distributed for any sufficient model and number of features, where ν is the number of d.o.f. of the underlying inter-image geometry. This makes GFR easy to test and very sensitive to residual biases and oversized errors, as these are typically proportional to the number of features n and hence easily seen against the fixed χ^2_ν background for $n \gg \nu$. For example, fig.1 shows GFR histograms for the 7 d.o.f. uncalibrated epipolar geometry for direct and reduced $\boldsymbol{F}$-matrix estimators and strong and weak (1% non-coplanar) feature sets. For the strong geometry both methods agree perfectly with the theoretical χ^2_7 distribution without any sign of outliers, so both methods do as well as could be hoped. This holds for any number of points from 9 to 1000 — the estimated geometry (error per point) becomes more accurate, but the total GFR error stays constant. For the weak geometry both methods do significantly worse than the theoretical limit — in fact they turn out to have a small but roughly constant residual error *per point* rather than in total — with the direct method being somewhat better than the reduced one. We are currently investigating this: in theory it should be possible to get near the limit, even for exactly singular geometries.

6 Summary

We have described work in progress on a generic, modular library for the optimal nonlinear estimation of matching constraints, discussing especially the overall approach, parametrization and numerical optimization issues. The library will cover many different constraint types & parametrizations and feature types & error models in a uniform framework. It aims to be efficient and stable even in near-degenerate cases, *e.g.* so that it can be used reliably for model selection. Several fairly sophisticated numerical methods are included, including a sparse constrained optimization method designed for ***direct geometric fitting***. Future work will concentrate mainly on (i) implementing and comparing different constraint types and parametrizations, feature types, and numerical resolution methods; and (ii) improving the reliability of the initialization and optimization stages, especially in near-degenerate cases.

References

1. S. Avidan and A. Shashua. Threading fundamental matrices. In *European Conf. Computer Vision*, pages 124–140, Freiburg, 1998.
2. Åke Björk. *Numerical Methods for Least Squares Problems.* SIAM Press, Philadelphia, PA, 1996.
3. B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1030–1036, Cambridge, MA, June 1995.
4. O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, 1993.
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. O. Faugeras and T. Papadopoulo. Grassmann-Cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. *Transactions of the Royal society A*, 1998.
7. R. Fletcher. *Practical Methods of Optimization.* John Wiley, 1987.
8. R.I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Computer Vision*, 22(2):125–140, 1997.
9. Anders Heyden. *Geometry and Algebra of Multiple Projective Transformations.* Ph.D. Thesis, University of Lund, 1995.
10. Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical Report RR-1894, INRIA, Sophia Antipolis, France, 1993.
11. Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. In *European Conf. Computer Vision*, pages 589–599, 1994.
12. 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.
13. G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 13(11):1115–38, 1991.

14. B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.
15. B. Triggs. Linear projective reconstruction from matching tensors. In *British Machine Vision Conference*, pages 665–74, Edinburgh, September 1996.
16. B. Triggs. A new approach to geometric fitting. Available from http://www.inrialpes.fr/movi/people/Triggs, 1997.