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

I will describe work in progress on a library of routines for estimating multi-image matching relations. The library covers 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 codes for stable, reliable, statistically optimal estimation of matching relations under various different robust error models, taking full account of any nonlinear constraints involved. In the shorter term, the library will be used to study the relative performance of the various competing parametrizations, error models and numerical methods. The talk will focus on the constraint parametrization and numerical optimization issues (rather than, say, feature extraction, grouping or initialization). The methods described are also highly effective for many other geometric estimation problems in vision, such as curve and surface fitting.

Motivation: Why study matching constraints in such detail? – Practically, they are central to both feature grouping and 3D reconstruction, so better algorithms should be of immediate benefit to many geometric vision applications. However they have a surprisingly rich and delicate algebraic structure which makes them rather difficult to estimate accurately. Many practically common camera and scene geometries correspond to degenerate or near-degenerate cases whose special properties need to be detected and exploited for stability, and even in stable cases it is not yet clear how best to parametrize the constraints — usually, they belong to fairly complicated algebraic varieties and either redundant parametrizations or nonlinear constraints are required.

In fact, it is becoming clear that more often than not, a single model does not suffice. 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, projective fundamental matrices for wide views of large close objects, lens distortion corrections for real images, *etc.*. The model should ideally be chosen to optimize some measure of statistically expected end-to-end system performance, given the observed input data. The key question is always whether a more general model improves the expected performance (owing to more accurate modelling) or worsens it (owing, *e.g.*, to fluctuations induced by poor estimates of superfluous model parameters). Although there are many specific decision criteria (maximum likelihood, AIC, BIC, . . .), the essential choice is always between the potential bias of an over-restrictive model and the potential variability of an over-general one. Any such model selection approach requires several models to be fitted to the data so that the best can be chosen. Many models may need to be fitted and marginal cases are very common, so fast, reliable, accurate fitting in near-degenerate cases is indispensable for practical model selection.

Finally, as the wide applicability of the techniques developed illustrates, matching constraints are an excellent testing ground for general statistical and numerical techniques that may be useful in other parts of vision.

The estimation of matching constraints is a complicated business. There are at least three main axes of variation: (*i*) feature type and error model; (*ii*) constraint type and parametrization; (*iii*) numerical optimization and initialization method. Most published algorithms represent a single point in this space. There has been little comparative work, so it is not yet clear which error models, parametrizations and numerical methods will provide the best practical compromise between accuracy, stability, speed, and simplicity. The immediate aim of this work is to create an experimental platform on which to study this issue. More generally, I wanted to move away from *ad hoc* algorithm designs towards a more systematic coverage of the space of possibilities. Both goals suggested that the above three axes needed to be decoupled, so that different features, constraint types and numerical methods could be more freely 'mixed and matched'. The code is a fairly direct reflection of this.

Feature models: It is clearly desirable to be able to use various different types of feature grouping to estimate matching relations — points, lines, conics, homographies... There has also been some recent debate about (*i*) the extent to which accurate statistical error modelling is necessary or desirable; and (*ii*) the best means of robustifying against outliers (mismatches, poorly localized features, ...). Hence, it is useful to be able to compare different robust and conventional error models (algebraic, uniform geometric, full image covariance...).

Matching relations: Each basic class of relation (epipolar, trifocal constraint, ...) has various specialized or degenerate forms depending on the camera model (projective, affine, calibrated...), motion (general, linear, rotational...) and scene geometry (general, planar/distant, small object...). For reliable results, it is essential to use an appropriate model, *i.e.* one which can be estimated stably from the given data without too much variability or residual bias. Current model selection techniques require multiple models to be fitted and compared, so speed is important. As many of the fits will necessarily be either strongly biased or ill-conditioned, stable, well-thought-out numerical methods are essential. Bias increases the fraction of apparent outliers (which may cause numerical problems if fitting occurs in the tails of the robust distributions), and the residual (which may slow convergence). Ill-conditioning puts a premium on good problem parametrization and numerical accuracy.

Parametrization: In most cases the space of possible matching relations of a given type is a nonlinear algebraic variety which can not be parametrized by a single, simple, minimal coordinate system. Several competing parametrization paradigms exist: (*i*) overlapping sets of local, *ad hoc*, nonlinear coordinate systems; (*ii*) redundant parametrizations with internal gauge freedoms (*iii*) linear embeddings subject to nonlinear consistency constraints. None of these is ideal and all complicate the fitting process. **Local nonlinear coordinates** are essentially based on algebraic elimination of variables. Hence, they tend to be ill-conditioned, particularly near the boundaries of the coordinate patch. Many patches may be required, along with extra code to manage the transitions between them. **Redundant parametrizations** are usually much stabler, at the cost of extra variables describing irrelevant internal 'gauge' degrees of freedom which have no effect on the fit. Examples are the 3D coordinate frame freedoms in projection matrix based parametrizations, or the incompletely specified homography in homography+epipole epipolar geometry ones. The Hessians of redundant parametrizations are exactly rank deficient, so specialized numerical methods must be used. **Nonlinear constraints** can be introduced, either as consistency conditions on a linear (tensor) embedding of the matching constraint, or as 'gauge fixing' conditions on a redundant parametrization. But then more complicated, constrained optimization algorithms must be used. All three possibilities are considered in the library.

Numerical Methods: Fairly sophisticated numerical optimization methods are needed to handle the nonlinear robust error function induced by the features on the matching constraint, under its nonlinear, redundant or constrained parametrization, while maintaining efficiency, stability under degeneracies, *etc.*. To ensure good numerical conditioning, the current implementation always works in a 'square root' (residual vector based) Gauss-Newton formalism. For example, its default linear algebra module uses column pivoted QR decomposition rather than the normal/Cholesky method. A new variant of trust region based step length control was developed for constrained problems, as existing methods in the literature gave poor performance. Initialization remains a difficult problem, especially from sparse or near-degenerate data — more work is required on this.

Geometric Fitting Methods: The key to geometric fitting is the synthesis of a model parameter error function from the individual feature errors. Roughly speaking the posterior likelihood of a model depends on the summed statistical distances of the observations from the *consistent set* in feature space (*i.e.* the submanifold defined by all feature groups consistent with the model). The method traditionally used to calculate this is 'orthogonal regression' — project the observations Mahalanobis-orthogonally onto the consistent submanifold, and calculate how far each projection is from its parent observation. For one-codimensional models (curves in 2D, epipolar geometry, …) the projection can be done analytically ('inverse gradient weighting'), but in higher codimensions (curves in 3D, trifocal geometry, …) this rapidly becomes intractable and numerical projection is required. In either case iteration is needed, the projection must be recalculated at each iteration, and the result is only ever correct to first order. Instead of this, the library takes a new 'direct' approach, including the projection positions as live variables in the optimization. The result is simpler to use, gives exactly optimal rather than first order results, and provides useful additional information in the form of optimal self-consistent estimates of the 'true underlying feature positions'. The traditional approach can be recovered by simply failing to update the running projection position estimates.

Robust Error Model: The oldest and — in my opinion — still the most effective approach to robustness is direct statistical fitting (maximum likelihood, MAP,…) to a model of the *total* observation distribution (*i.e.* including *both* inliers *and* outliers). The library supports this, but currently makes no attempt to estimate global (outlier frequency or noise level) parameters on-line. The actual distributions are defined internally by the feature models (I often use mixtures of Gaussian inliers and uniform outliers). Robustification often makes the numerical optimization problem significantly harder. In particular, for outliers the error gradient can become very small (which can cause numerical conditioning and scaling problems) and the Hessian non-positive (which requires additional damping and hence slows convergence). The library avoids the worst of these problems by careful scaling and correcting Jacobians for second order robust error function derivatives, but reliable robust fitting in near-degenerate cases remains a difficult problem.