

Optimal Estimation of Matching Constraints

- 1 Motivation & general approach
- 2 Parametrization of matching constraints
- 3 Direct vs. reduced fitting
- 4 Numerical methods
- 5 Robustification
- 6 Summary

Why Study Matching Constraint Estimation?

- ① They are *practically useful*, both for correspondence and reconstruction
- ② They are *algebraically complicated*, so the best algorithm is not obvious
 - a good testing ground for new ideas . . .
- ③ There are *many variants*
 - different constraint & feature types, camera models
 - special forms for degenerate motions and scene geometries

⇒ Try a systematic approach rather than an *ad hoc* case-by-case one

Model selection

- For practical reliability, it is essential to use an appropriate model
- **Model selection** methods fit several models, choose the best
 - ⇒ many fits are to **inappropriate models** (strongly biased, degenerate)
 - ⇒ the fitting algorithm must be efficient and reliable, even in difficult cases

Questions to Study

- 1 How much difference does an *accurate statistical error model* make ?
- 2 Which types of *constraint parametrization* are the most reliable ?
- 3 Which *numerical method* offers the best stability/speed/simplicity ?

The answers are most interesting for *nearly degenerate* cases, as these are the most difficult to handle reliably.

Design of Library

1. Modular Architecture

- Separate modules for
 - 1 matching geometry type & parametrization
 - 2 feature type, parametrization & error model
 - 3 linear algebra implementation
 - 4 loop controller (step damping, convergence tests)

Stable Gauss-Newton Approach

① Work with residual error vectors $\mathbf{e}(\mathbf{x})$ and Jacobians $\frac{d\mathbf{e}}{d\mathbf{x}}$

— not gradient and Hessian of squared error $\frac{d(|\mathbf{e}|^2)}{d\mathbf{x}} = \mathbf{e}^\top \frac{d\mathbf{e}}{d\mathbf{x}}, \quad \frac{d\mathbf{e}^\top}{d\mathbf{x}} \frac{d\mathbf{e}}{d\mathbf{x}}$

• e.g. simplest residual is $\mathbf{e} = \mathbf{x} - \underline{\mathbf{x}}$ for observations $\underline{\mathbf{x}}$

② Discard 2nd derivatives, e.g. $\mathbf{e}^\top \frac{d^2\mathbf{e}}{d\mathbf{x}^2}$

③ For stability use QR decomposition, not normal equations + Cholesky

Advantages of Gauss-Newton

⊕ Simple to use — no 2nd derivatives required

⊕ Stable *linear least squares methods* can be used for step prediction

⊖ Convergence may be slow if problem has both *large residual* and *strong nonlinearity* — but in vision, *residuals are usually small*

Parametrization of Matching Geometry

- The underlying geometry of matching constraints is parametrized by *nontrivial algebraic varieties* — there are *no* single, simple, minimal parametrizations
 - e.g. epipolar geometry \approx the variety of all homographic mappings between line pencils in two images

There are (at least) three ways to parametrize varieties:

- 1 *Implicit constraints* on some higher dimensional space
- 2 Overlapping *local coordinate patches*
- 3 *Redundant parametrizations* with internal gauge freedoms

Constrained Parametrizations

- 1 Embed the variety in a larger (e.g. linear, tensor) space
- 2 Find *consistency conditions* that characterize the embedding

Matching Tensors are the most familiar embeddings

- coefficients of *multilinear feature matching relations*
- e.g. the fundamental matrix F
- Other useful embeddings of matching geometry may exist . . .
- Typical consistency conditions:
 - fundamental matrix: $\det(F) = 0$
 - trifocal tensor: $\frac{d^3}{dx^3} \det(G \cdot x) = 0$ plus others . . .

Advantages of Constrained Parametrizations

- ⊕ Very natural when matching geometry is derived from image data
- ⊕ “*Linear methods*” give (inconsistent!) initial estimates
- ⊖ *Reconstruction problem* — how to go from tensor to other properties of matching geometry
- ⊖ The consistency conditions rapidly become complicated and non-obvious
 - Demazure for essential matrix
 - Faugeras-Papadopoulos for the trifocal tensor
- ⊖ *Constraint redundancy* is common: #generators $>$ codimension

Local Coordinates / Minimal Parametrizations

Express the geometry in terms of a *minimal set of independent parameters*

- e.g. describe some components of a matching tensor as *nonlinear functions* of the others (or of some other parameters)

- c.f. Z. Zhang's $\mathbf{F} = \begin{pmatrix} a & b & c \\ d & e & f \\ ua+vd & ub+ve & uc+vf \end{pmatrix}$ guarantees $\det(\mathbf{F}) = 0$

Advantages of Minimal Parametrizations

- ⊕ Simple unconstrained optimization methods can be used
- ⊖ They are usually *highly anisotropic*
 - they don't respect symmetries of the underlying geometry so they are messy to implement, and hard to optimize over
- ⊖ They are usually *only valid locally*
 - many coordinate patches may be needed to cover the variety, plus code to manage inter-patch transitions
- ⊖ They must usually be found by *algebraic elimination* using the constraints
 - numerically ill-conditioned, and rapidly becomes intractable

It is usually preferable to eliminate variables *numerically* using the constraint Jacobians — *i.e.* constrained optimization

Redundant Parametrizations / Gauge Freedom

In many geometric problems, the simplest approach requires an *arbitrary choice of coordinate system*

Common examples:

- 1 3D coordinate frames in reconstruction, projection-based matching constraint parametrizations
- 2 Homogeneous-projective *scale factors* $F \rightarrow \lambda F$
- 3 *Homographic parametrizations* of epipolar and trifocal geometry

$$F \simeq [e]_{\times} H \quad \text{with freedom} \quad H \rightarrow H + e a \quad \text{for any } a$$

$$G \simeq e' \otimes H'' - H' \otimes e'' \quad \text{with freedom} \quad \begin{pmatrix} H' \\ H'' \end{pmatrix} \rightarrow \begin{pmatrix} H' \\ H'' \end{pmatrix} + \begin{pmatrix} e' \\ e'' \end{pmatrix} a^{\top}$$

Gauge Freedoms

Gauge Freedoms are *internal symmetries* associated with a free choice of internal “coordinates”

- **Gauge** just means *(internal) coordinate system*
- There is an associated *symmetry group* and its *representations*
- Expressions derived in gauged coordinates reflect the symmetries
- A familiar example: ordinary 3D Cartesian coordinates
 - the gauge group is the rigid motions
 - the gauged representations are Cartesian tensors

Advantages of Gauged Parametrizations

- ⊕ Very natural when the matching geometry is derived from the 3D one
- ⊕ Close to the geometry, so it is easy to derive further properties from them
- ⊕ Numerically *much stabler* than minimal parametrizations
- ⊕ One coordinate system covers the whole variety
- ⊖ Symmetry implies *rank degeneracy* — special numerical methods are needed
- ⊖ They may be slow as there are additional, redundant variables

Handling Gauge Freedom Numerically

Gauge motions don't change the residual, so there is nothing to say what they should be

- If left undamped, *large gauge fluctuations* can destabilize the system
 - e.g. Hessians are exactly rank deficient in the gauge directions
- Control fluctuations by **gauge fixing conditions** or **free gauge** methods
- C.f. *'Free Bundle'* methods in photogrammetry

1. Gauge Fixing Conditions

- Remove the degeneracy by adding **artificial constraints**
 - e.g. Hartley's gauges $\mathbf{P}_1 = (\mathbf{I}_{3 \times 3} \mid \mathbf{0})$, $\mathbf{e} \cdot \mathbf{H} = \mathbf{0}$
- Constrained optimization is (usually) needed
- Poorly chosen constraints can *increase* ill-conditioning

2. 'Free Gauge' Methods

- 1] Leave the gauge "free to drift" — but take care not to push it too hard!
 - **rank deficient least squares** methods (basic or min. norm solutions)
 - **Householder reduction** projects motion orthogonally to gauge directions
- 2] Monitor the gauge and reset it "by hand" as necessary (e.g. each iteration)

Constrained Optimization

Constraints arise from

- 1 **Matching relations** on features, e.g. $\mathbf{x}^\top \mathbf{F} \mathbf{x} = 0$
- 2 **Consistency conditions** on matching tensors, e.g. $\det(\mathbf{F}) = 0$
- 3 **Gauge fixing conditions**, e.g. $\mathbf{e} \cdot \mathbf{H} = 0$, $\|\mathbf{F}\|^2 = 1$

Approaches to Constrained Optimization

- 1 *Eliminate variables numerically* using constraint Jacobian
- 2 Introduce *Lagrange multipliers* and solve for these too
 - for dense systems, 2 is simpler but 1 is usually faster and stabler
 - each has many variants: linear algebra method, operation ordering, . . .

Difficulties

- The linear algebra gets complicated, especially for sparse problems
- A lack of efficient, reliable *search control heuristics*
- *Constraint redundancy*

Constraint Redundancy

Many algebraic varieties have #generators $>$ codimension

The constraint Jacobian has $\begin{cases} \text{rank} = \text{codimension on the variety} \\ \text{rank} > \text{codimension away from it} \end{cases}$

- Examples

1 the trifocal point constraint $[\mathbf{x}']_{\times} (\mathbf{G} \cdot \mathbf{x}) [\mathbf{x}'']_{\times}$ has rank 3 for valid trifocal tensors, 4 otherwise

2 the trifocal consistency constraint $\frac{d^3}{dx^3} \det(\mathbf{G} \cdot \mathbf{x})$ has rank 8 for valid tensors, 10 otherwise

- It seems difficult to handle such **localized redundancies** numerically
- Currently, I assume known codimension r , project out the strongest r constraints and enforce only these

Abstract Geometric Fitting Problem

1. Model-Feature Constraints

There are

- 1 Unknown *true underlying 'features'* \mathbf{x}_i
- 2 An unknown *true underlying 'model'* \mathbf{u}
- 3 Exactly satisfied *model-feature consistency constraints*

$$\mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = 0$$

- *E.g.* for epipolar geometry
 - a 'feature' is a pair of corresponding points $(\mathbf{x}_i, \mathbf{x}'_i)$
 - the 'model' \mathbf{u} is the fundamental matrix \mathbf{F}
 - the 'model-feature constraint' is the epipolar constraint $\mathbf{x}_i^\top \mathbf{F} \mathbf{x}'_i = 0$

2. Error Model

1 There is an additive **posterior statistical error metric** linking the underlying features to observations and other prior information

$$\rho_i(\mathbf{x}_i) = \rho_i(\mathbf{x}_i | \text{observations } i)$$

— e.g. (robustified, bias corrected) **posterior log likelihood**

2 There may also be a **model-space prior** $\rho_{\text{prior}}(\mathbf{u})$

• For epipolar geometry, given observed points $(\underline{\mathbf{x}}, \underline{\mathbf{x}}')$, we could take

$$\rho(\mathbf{x}, \mathbf{x}') = \rho \left(\|\mathbf{x} - \underline{\mathbf{x}}\|^2 + \|\mathbf{x}' - \underline{\mathbf{x}}'\|^2 \right)$$

where $\rho(\cdot)$ is some robustifier

3. Model Parametrization

The model \mathbf{u} may have a *nontrivial parametrization*

① *internal constraints* $\mathbf{k}(\mathbf{u}) = \mathbf{0}$

② *local parametrization* $\mathbf{u} = \mathbf{u}(\mathbf{v})$ with free parameters \mathbf{v}

③ *internal gauge freedoms*

E.g. for the fundamental matrix we can choose

either constraint $\det(\mathbf{F}) = 0$

or gauge freedom $\mathbf{F} \simeq [\mathbf{e}]_{\times} \mathbf{H}$

4. Estimation Method

- We want to find **point estimates** of the model \mathbf{u} and (maybe) the underlying features \mathbf{x}_i , which **minimize the total error** subject to all of the constraints

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

- $(\hat{\mathbf{u}}, \hat{\mathbf{x}}_i)$ are **optimal self-consistent estimates** of the underlying model and features $(\mathbf{u}, \mathbf{x}_i)$

Fitting by Reduction to Model Space

- The traditional approach to geometric fitting is *reduction*
- ① Use *local approximations* based at the observations $\underline{\mathbf{x}}_i$ to derive an *effective model-space cost function* $\sum_i \rho_i(\mathbf{u} | \underline{\mathbf{x}}_i)$
- ② Numerically optimize over \mathbf{u} (subject to any constraints, *etc*, on it)

Advantages

- ⊕ The optimization is (nominally) over relatively few variables \mathbf{u}
- ⊖ The cost function $\rho(\mathbf{u})$ is complicated and only correct to 1st order
- ⊖ If $\dim(\mathbf{c}) > 1$, even the approximation has to be evaluated numerically

Estimating the Reduced Cost

The reduced error $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i)$ is given by **Gradient Weighted Least Squares**

either Project each observation Mahalanobis-orthogonally onto the estimated local constraint surface, and work out the error ρ_i there

or Find the covariance in \mathbf{c}_i due to $\underline{\mathbf{x}}_i$, and work out $\chi^2 \approx \mathbf{c}^\top \text{Cov}(\mathbf{c})^{-1} \mathbf{c}$

- In either case, to first order

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

- e.g. for the epipolar constraint

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

- If \mathbf{c}_i is linear in \mathbf{u} and the dependence of the Jacobians on \mathbf{u} is ignored, $\rho(\mathbf{u})$ is a simple quadratic in \mathbf{u} which can be worked out once and for all

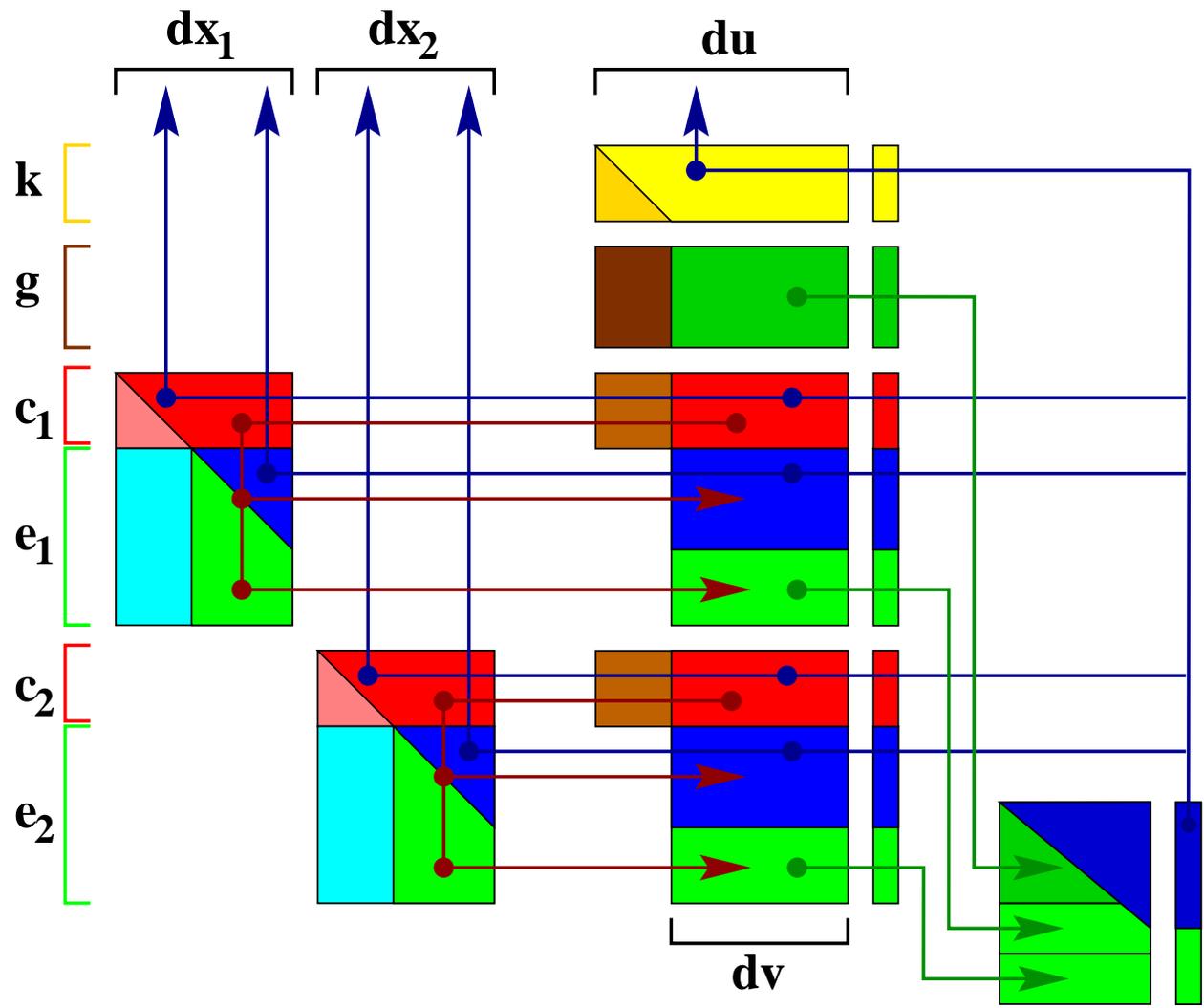
Direct Geometric Fitting

Fit the model by *direct constrained numerical optimization* over the natural variables $(\mathbf{u}, \mathbf{x}_i)$

- ⊕ Simple to use, even for complex problems
 - only the ‘natural’ error and constraint Jacobians are required
- ⊕ Gives *exact, optimal results*
- ⊕ Generates useful estimates of *true underlying features* \mathbf{x}_i
- ⊖ Requires a *sparse constrained optimization* routine

The only difference between the direct and reduced methods is that the reduced one throws away the easily calculated feature updates $d\mathbf{x}_i$

Direct Geometric Fitting — QR Method



Robustification

- Use standard statistical fitting (e.g. max. likelihood) to a model of the **total observed data distribution** — i.e. including **both inliers and outliers**
- Use numerical optimization, with initialization e.g. by consensus search
- All distribution parameters can (in principle) be estimated
 - e.g. covariances, outlier percentages

Implementation

- Assume a **central** robust cost function

$$\rho_i(\mathbf{x}_i) = \rho(\|\mathbf{e}_i(\mathbf{x}_i)\|^2)$$

- $\mathbf{e}_i(\cdot)$ is a **normalized residual error vector**
- $\rho(\cdot)$ is a **robust cost function**

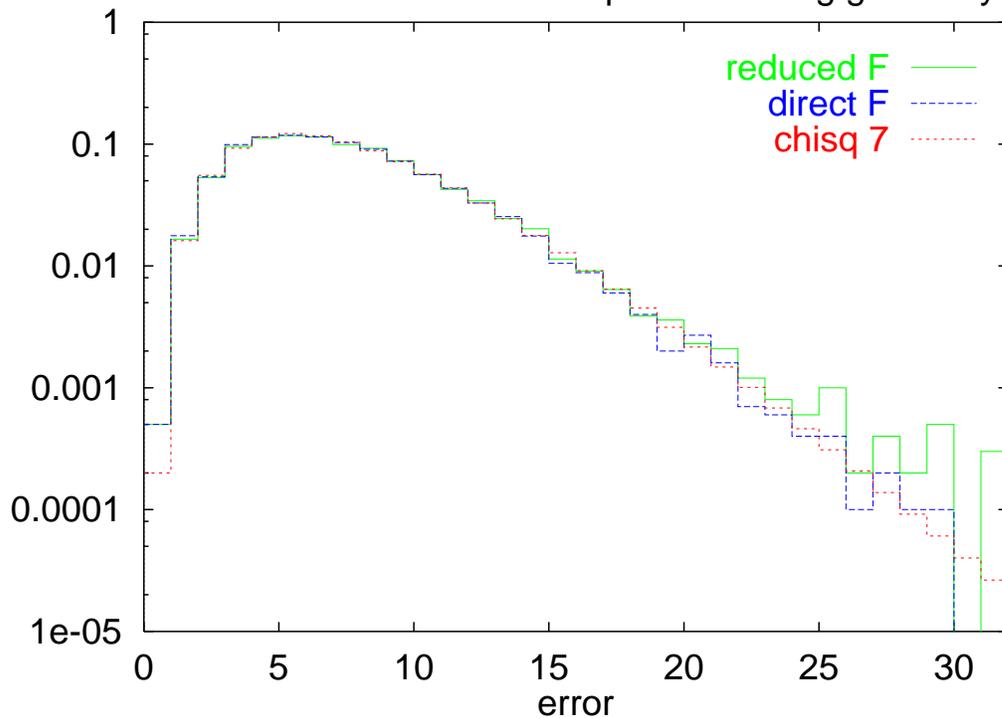
- e.g. an outlier polluted Normal distribution

$$\rho = -\log\left(\alpha + \beta \cdot e^{-\|\mathbf{e}\|^2/2}\right)$$

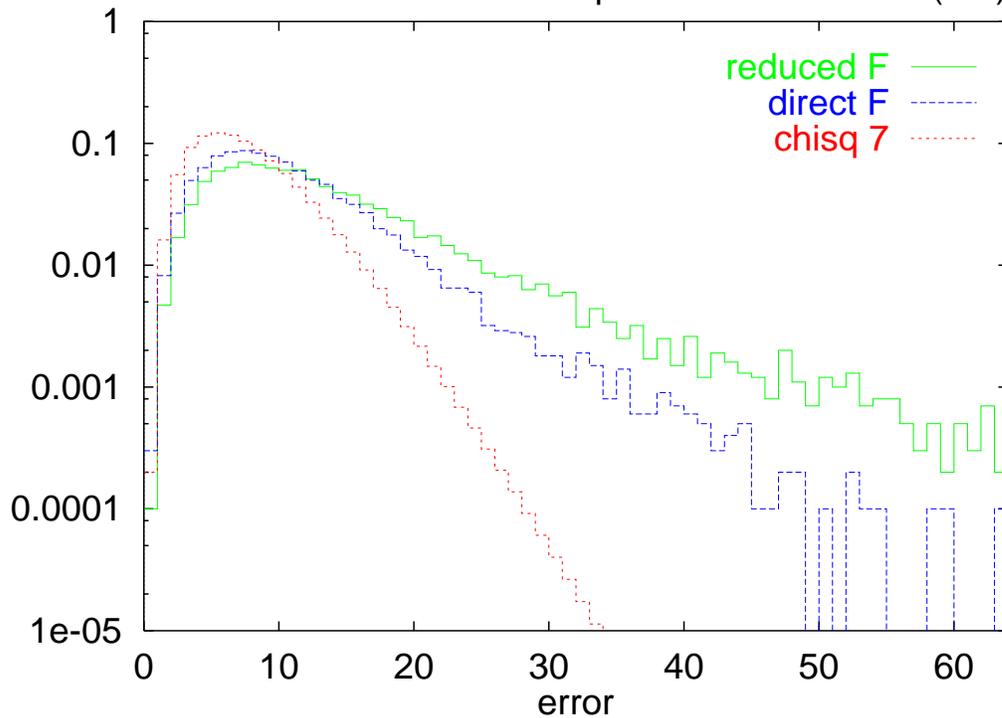
Numerical Problems caused by Robustification

- 1 **Nonconvex cost function** — regularization may be needed to guarantee positivity. This can slow convergence
 - to partially compensate, correct Jacobians for **2nd order curvature** of robustifier $\rho(\cdot)$ — a rank 1 correction along e
- 2 The robust error surface is **very flat for outliers** — this can cause poor numerical condition & scaling problems
 - apply the robust suppression as late in the numerical chain as possible, *i.e.* when the feature contributes to the model's cost function

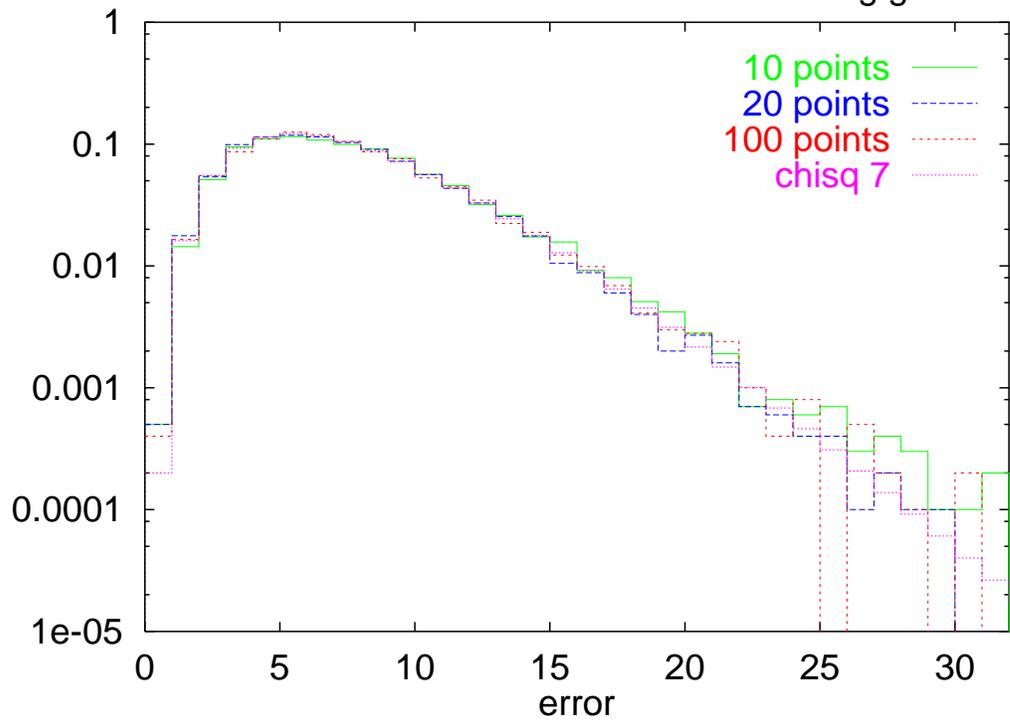
Ground Truth Residual - 20 points - strong geometry



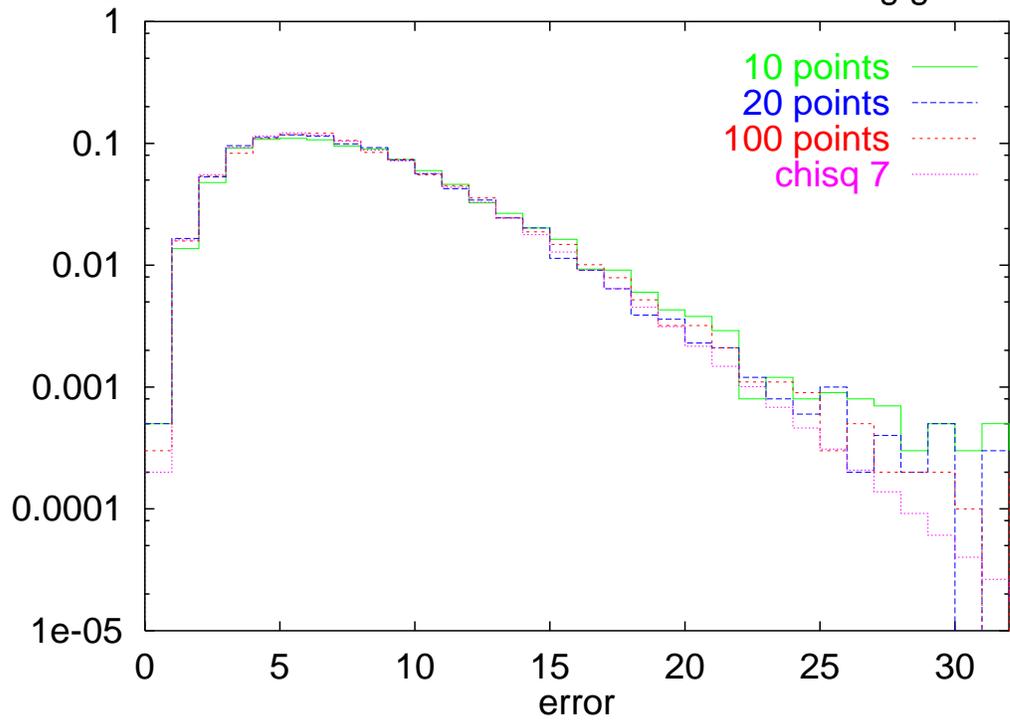
Ground Truth Residual - 20 points - Near-Planar (1%)



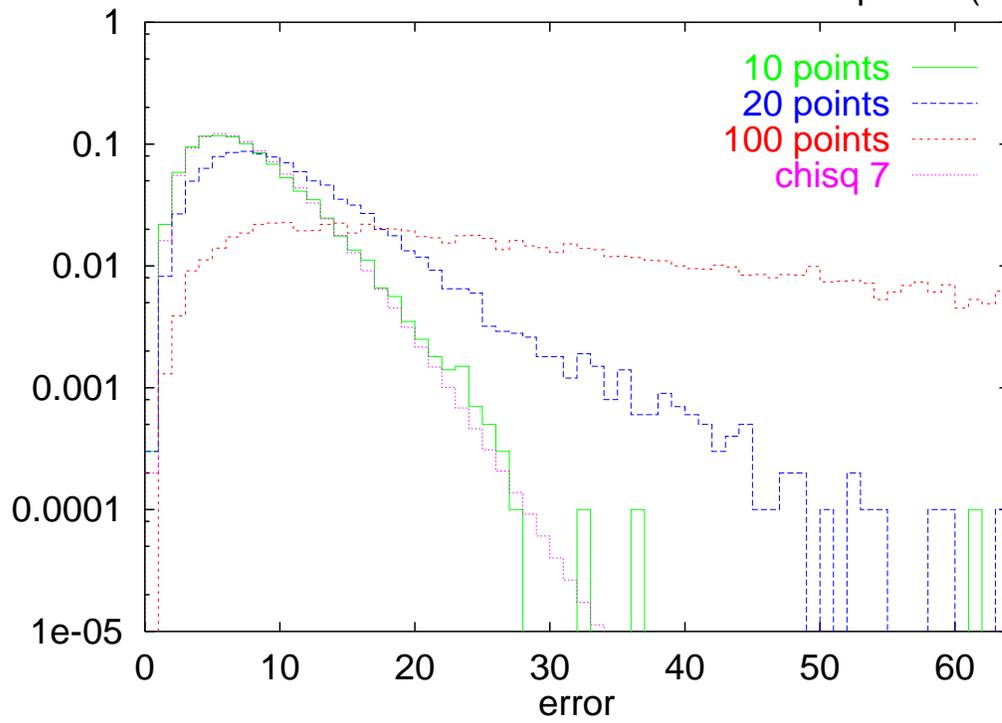
Ground Truth Residual - direct F matrix - strong geometry



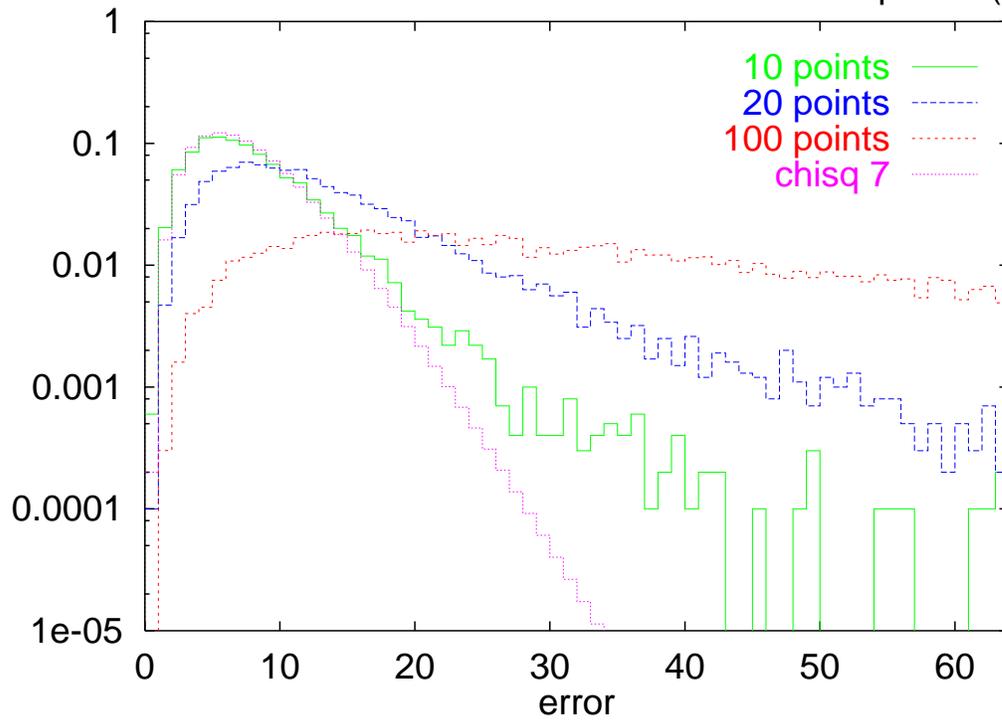
Ground Truth Residual - reduced F matrix - strong geometry



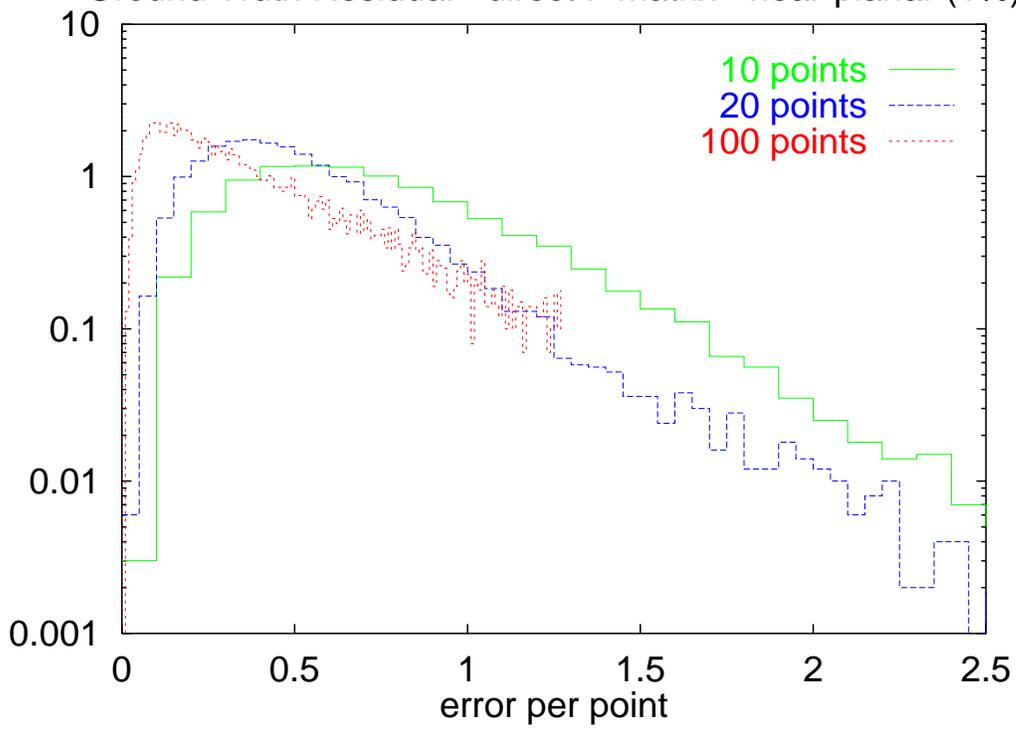
Ground Truth Residual - direct F matrix - near-planar (1%)



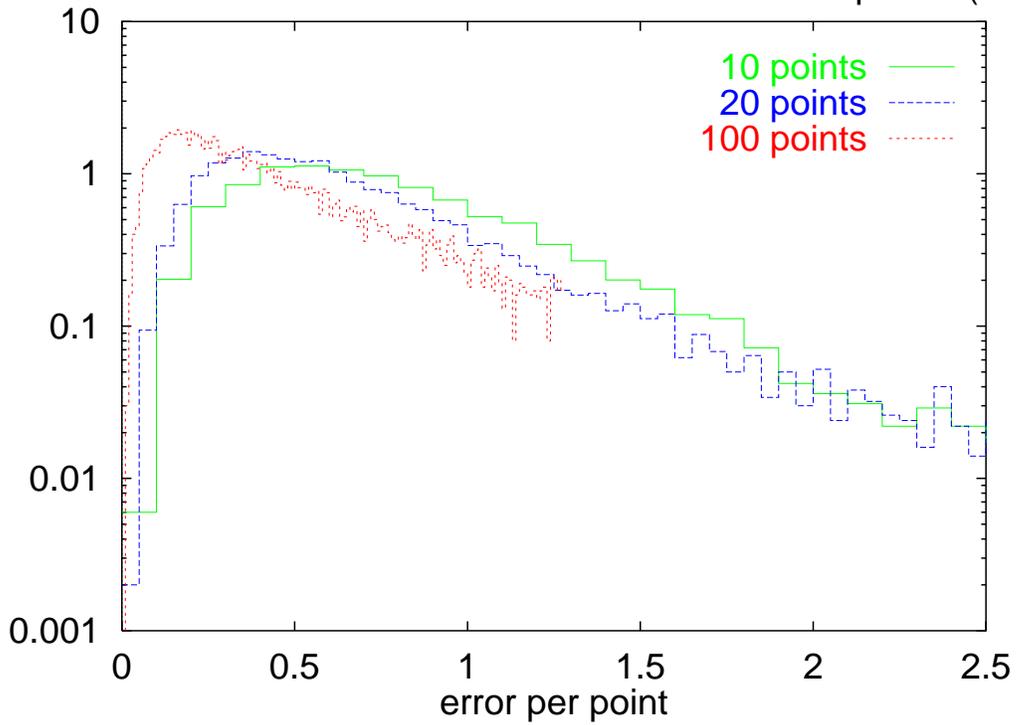
Ground Truth Residual - reduced F matrix - near-planar (1%)



Ground Truth Residual - direct F matrix - near-planar (1%)



Ground Truth Residual - reduced F matrix - near-planar (1%)



Summary

- A generic, modular library for *matching constraint estimation*
- Aims to be *efficient* and *stable*, even in *near-degenerate cases*
- Will be used to compare different
 - feature error models
 - constraint parametrizations
 - numerical resolution methods
- Central numerical method is *direct geometric fitting*

<http://www.inrialpes.fr/movi/people/Triggs/home.html>