Optimal Estimation of Matching Constraints

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

$\Rightarrow$ 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

$\Rightarrow$ many fits are to ***inappropriate models*** (strongly biased, degenerate)

$\Rightarrow$ 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 $\boxed{\mathbf{e}(\mathbf{x})}$ and Jacobians $\boxed{\dfrac{\mathbf{de}}{\mathbf{dx}}}$

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

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

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

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:

$\boxed{1}$ ***Implicit constraints*** on some higher dimensional space

$\boxed{2}$ Overlapping ***local coordinate patches***

$\boxed{3}$ ***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 $\boldsymbol{F}$

- Other useful embeddings of matching geometry may exist . . .

- Typical consistency conditions:

— fundamental matrix: $\boxed{\det(\boldsymbol{F}) \;=\; 0}$

— trifocal tensor: $\boxed{\frac{\mathbf{d}^3}{\mathbf{d}\boldsymbol{x}^3}\det(\boldsymbol{G}\cdot\boldsymbol{x}) \;=\; \boldsymbol{0} \quad \text{plus others} \ldots}$

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-Papadopoulo 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 $\boldsymbol{F} = \begin{pmatrix} a & b & c \\ d & e & f \\ ua+vd & ub+ve & uc+vf \end{pmatrix}$ guarantees $\det(\boldsymbol{F}) = 0$

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:

$\boxed{1}$ 3D coordinate frames in reconstruction, projection-based matching constraint parametrizations

$\boxed{2}$ Homogeneous-projective ***scale factors*** $\boldsymbol{F} \;\rightarrow\; \lambda\,\boldsymbol{F}$

$\boxed{3}$ ***Homographic parametrizations*** of epipolar and trifocal geometry

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

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

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 $\mathbf{P}_1 = (\boldsymbol{I}_{3\times 3} \,|\, \boldsymbol{0}), \ \mathbf{e} \cdot \boldsymbol{H} = \boldsymbol{0}$

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

Approaches to Constrained Optimization

$\boxed{1}$ ***Eliminate variables numerically*** using constraint Jacobian

$\boxed{2}$ Introduce ***Lagrange multipliers*** and solve for these too

— for dense systems, $\boxed{2}$ is simpler but $\boxed{1}$ 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 $\begin{cases} \text{rank} = \text{codimension } \textit{on} \text{ the variety} \\ \text{rank} > \text{codimension away from it} \end{cases}$

- Examples

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

$\boxed{2}$ the trifocal consistency constraint $\frac{\mathbf{d}^3}{\mathbf{d}\boldsymbol{x}^3}\det(\boldsymbol{G}\cdot\boldsymbol{x})$ 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'*** $\boxed{\mathbf{x}_i}$

2 An unknown ***true underlying 'model'*** $\boxed{\mathbf{u}}$

3 Exactly satisfied ***model-feature consistency constraints***

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

- *E.g.* for epipolar geometry

— a 'feature' is a pair of corresponding points $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$

— the 'model' $\mathbf{u}$ is the fundamental matrix $\boldsymbol{F}$

— the 'model-feature constraint' is the epipolar constraint $\boldsymbol{x}_i^\top \boldsymbol{F} \boldsymbol{x}'_i \; = \; 0$

2. Error Model

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

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

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

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

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

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

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

3. Model Parametrization

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

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

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

3 ***internal gauge freedoms***

E.g. for the fundamental matrix we can choose

either constraint $\boxed{\det(\boldsymbol{F}) \;=\; 0}$

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

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

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

- $(\hat{\mathbf{u}}, \hat{\mathbf{x}}_i)$ 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***

$\boxed{1}$ Use ***local approximations*** based at the observations $\underline{\mathbf{x}}_i$ to derive an ***effective model-space cost function*** $\boxed{\sum_i \rho_i(\mathbf{u} | \underline{\mathbf{x}}_i)}$

$\boxed{2}$ 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 $\boxed{\chi^2 \approx \mathbf{c}^\top \mathrm{Cov}(\mathbf{c})^{-1}\mathbf{c}}$

- In either case, to first order

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

- *e.g.* for the epipolar constraint

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

- 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

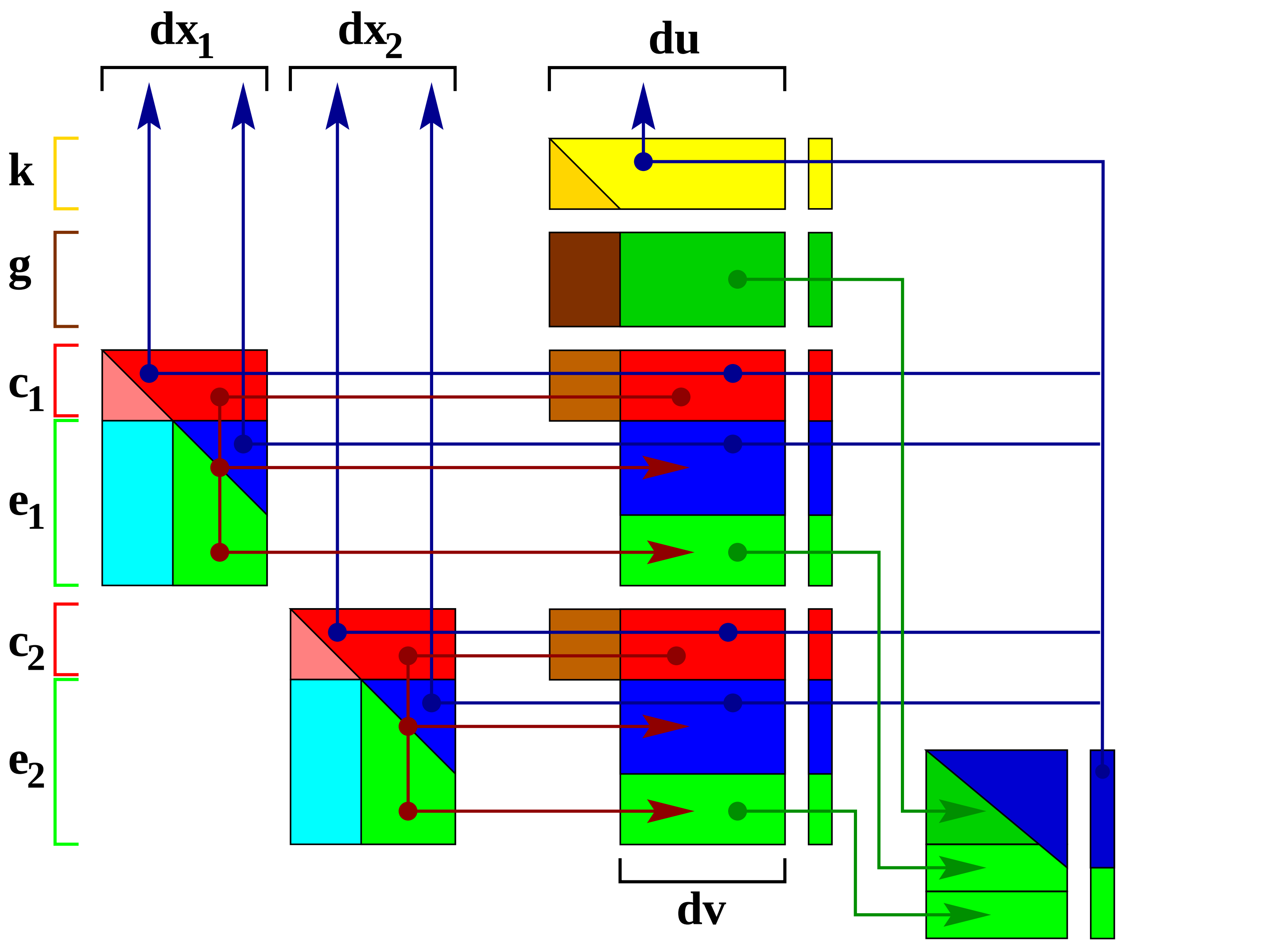
Direct Geometric Fitting

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 $\mathbf{dx}_i$

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 $\boxed{\rho_i(\mathbf{x}_i) = \rho(\|\mathbf{e}_i(\mathbf{x}_i)\|^2)}$
  — $\mathbf{e}_i(\cdot)$ is a ***normalized residual error vector***
  — $\rho(\cdot)$ is a ***robust cost function***
- *e.g.* an outlier polluted Normal distribution $\boxed{\rho = -\log\left(\alpha + \beta \cdot e^{-\|\mathbf{e}\|^2/2}\right)}$

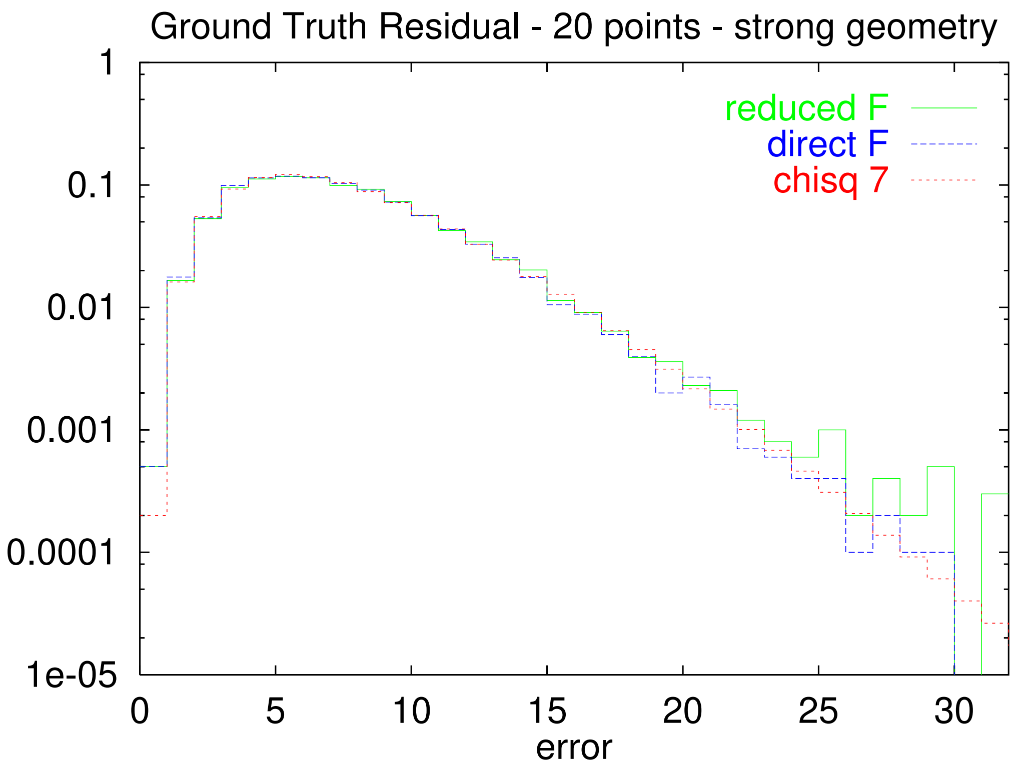
Numerical Problems caused by Robustification

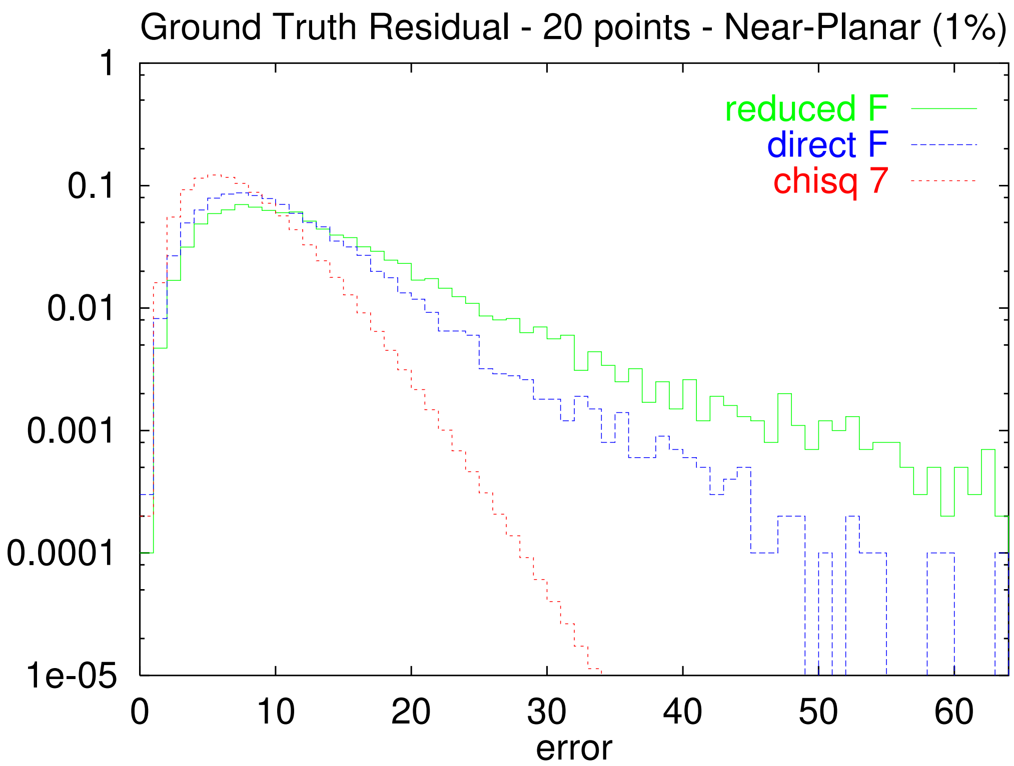
$\boxed{1}$ ***Nonconvex cost function*** — regularization may be needed to guarantee positivity. This can slow convergence

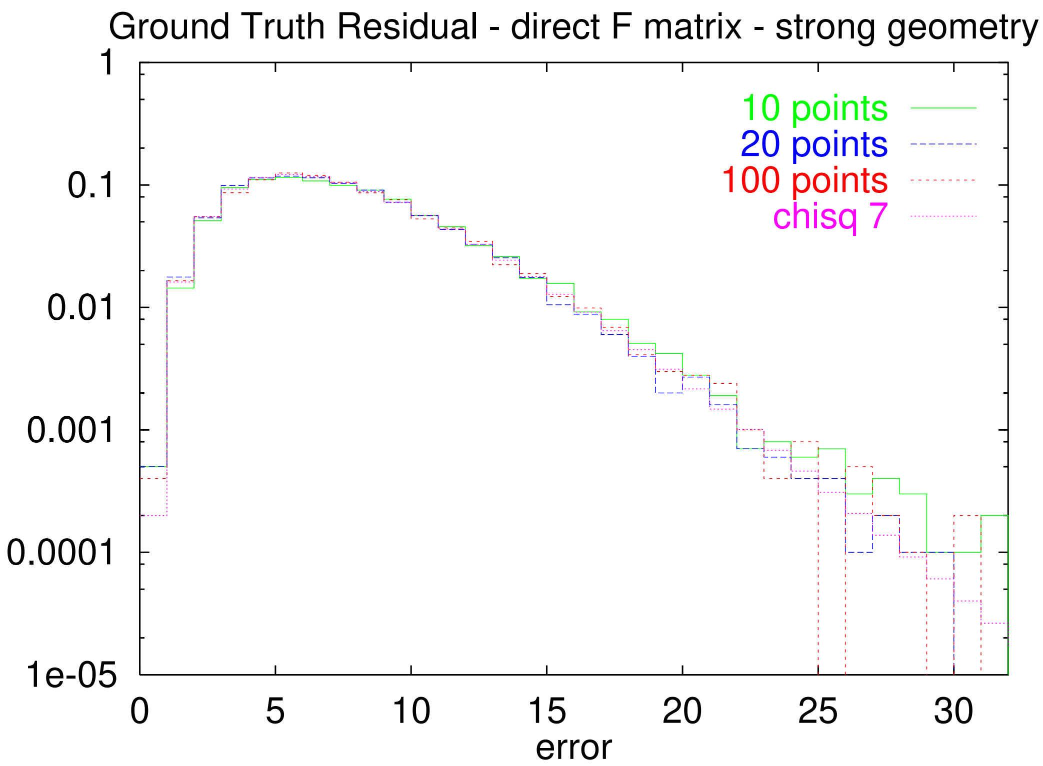
— to partially compensate, correct Jacobians for ***2nd order curvature*** of robustifier $\rho(\cdot)$ — a rank 1 correction along $\mathbf{e}$

$\boxed{2}$ The robust error surface is ***very flat for outliers*** — this can cause poor numerical condition & scaling problems

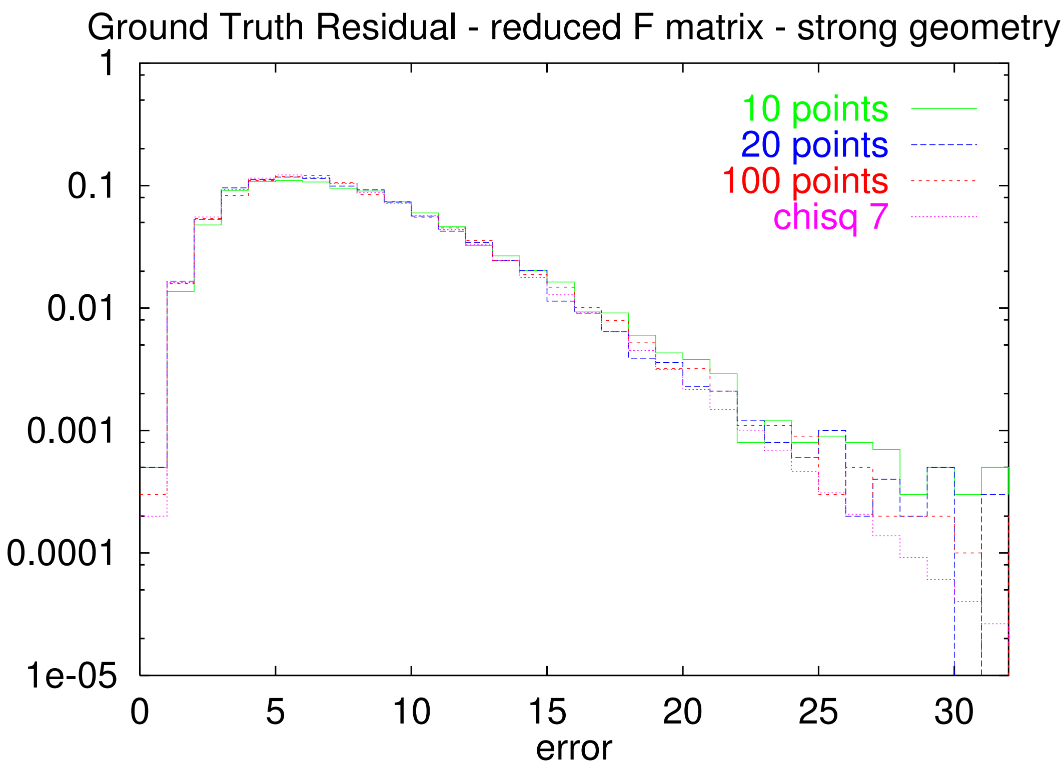
— apply the robust suppression as late in the numerical chain as possible, *i.e.* when the feature contributes to the model's cost function


Ground Truth Residual - 20 points - strong geometry
reduced F
direct F
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
5
10
15
20
25
30
error

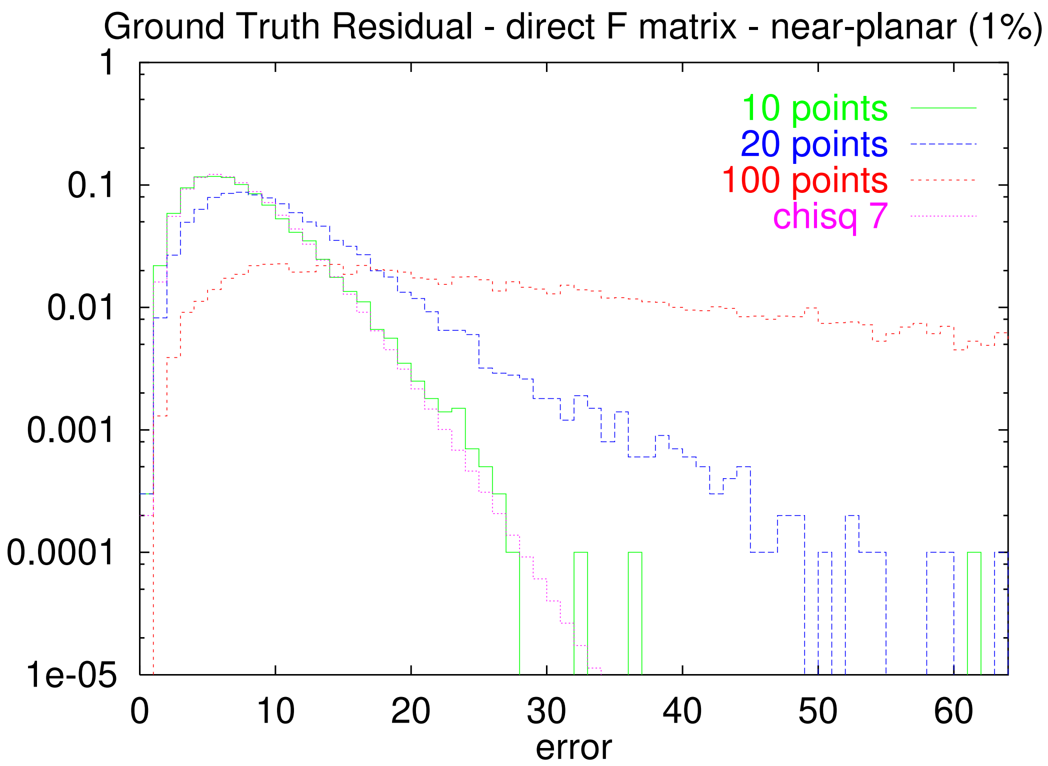

Ground Truth Residual - 20 points - Near-Planar (1%)
reduced F
direct F
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
10
20
30
40
50
60
error

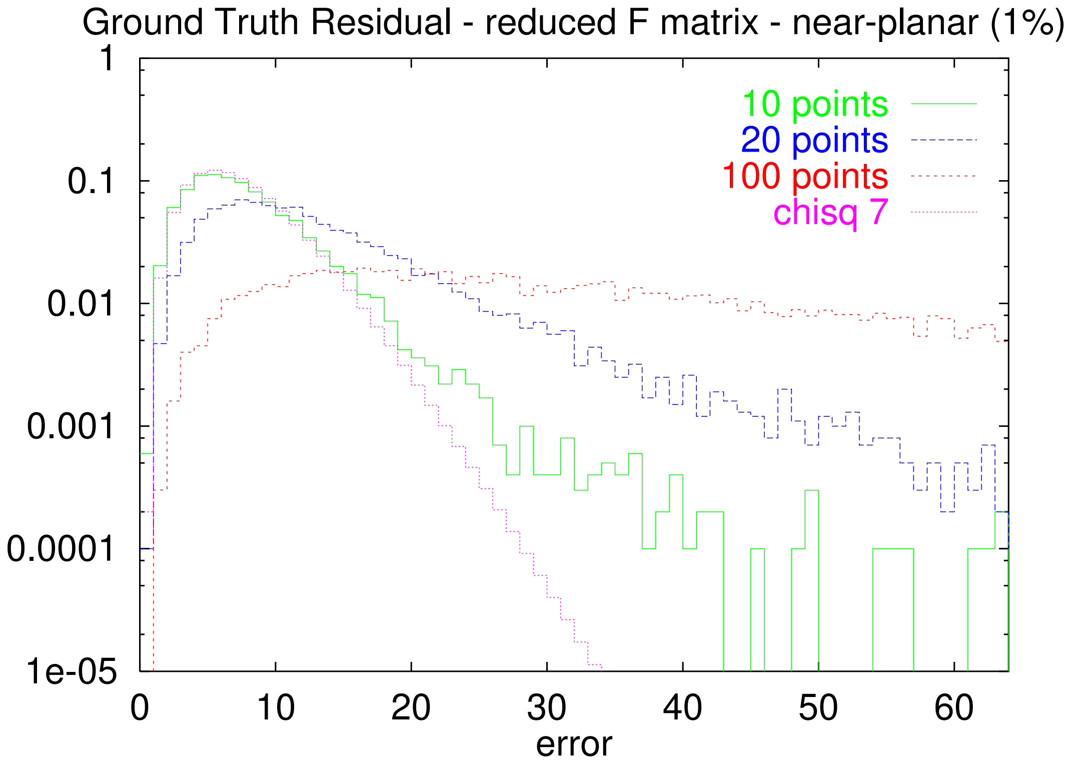


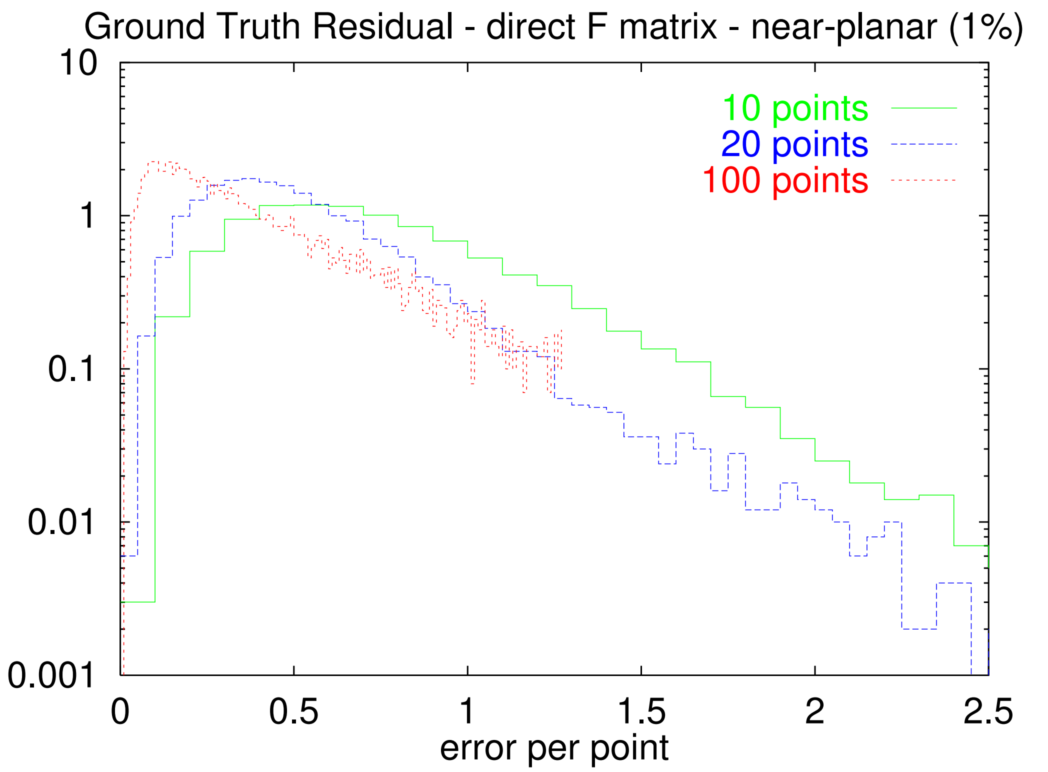
Ground Truth Residual - direct F matrix - strong geometry
10 points
20 points
100 points
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
5
10
15
20
25
30
error

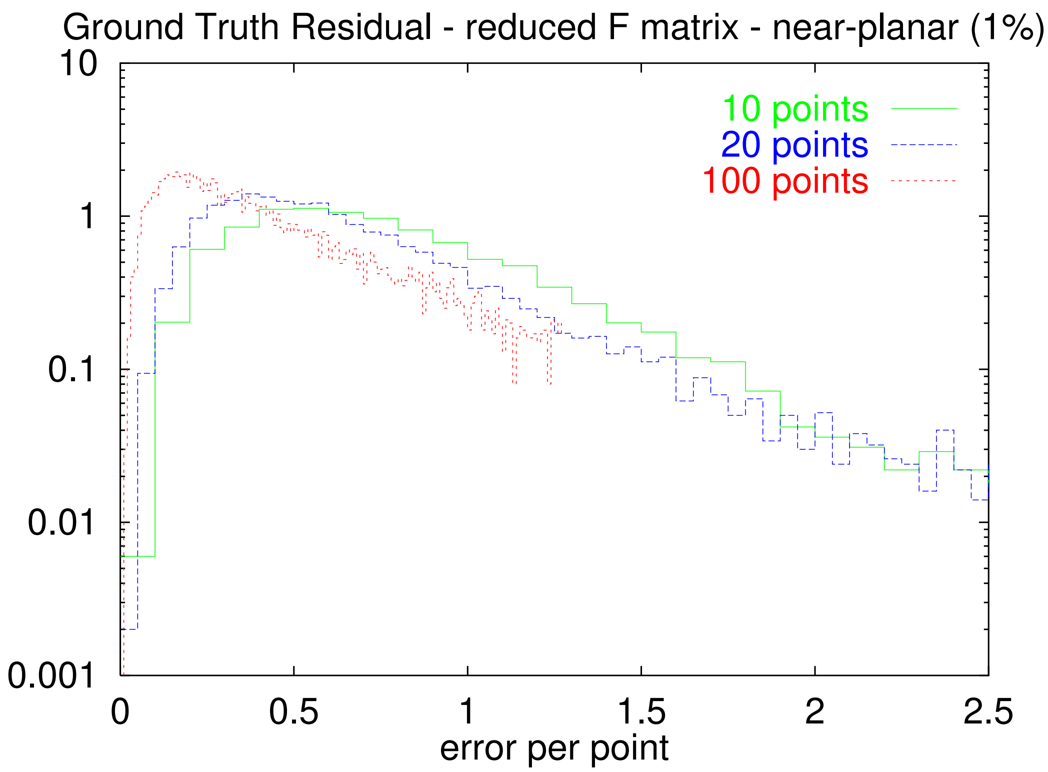


Ground Truth Residual - reduced F matrix - strong geometry
10 points
20 points
100 points
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
5
10
15
20
25
30
error


Ground Truth Residual - direct F matrix - near-planar (1%)
10 points
20 points
100 points
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
10
20
30
40
50
60
error


Ground Truth Residual - reduced F matrix - near-planar (1%)
10 points
20 points
100 points
chisq 7
1
0.1
0.01
0.001
0.0001
1e-05
0
10
20
30
40
50
60
error


Ground Truth Residual - direct F matrix - near-planar (1%)
10 points
20 points
100 points
10
1
0.1
0.01
0.001
0
0.5
1
1.5
2
2.5
error per point


Ground Truth Residual - reduced F matrix - near-planar (1%)
10 points
20 points
100 points
10
1
0.1
0.01
0.001
0
0.5
1
1.5
2
2.5
error per point

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