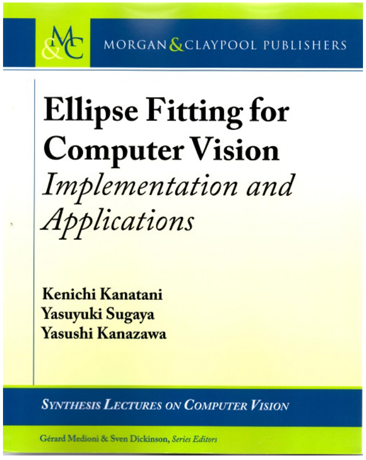
Tutorial

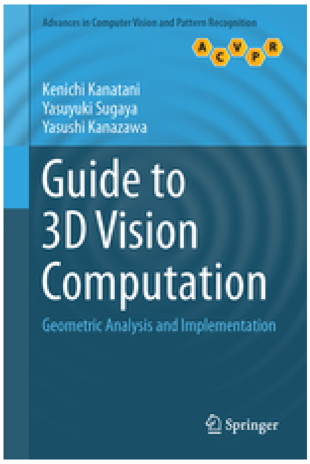
Fitting Ellipse and Computing Fundamental Matrix and Homography

Kenichi Kanatani

Professor Emeritus
Okayama University, Japan

This tutorial is based on



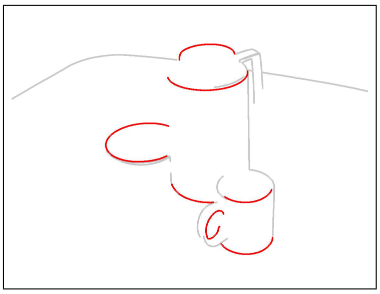


K. Kanatani, Y. Sugaya, and Y. Kanazawa,
Ellipse Fitting for Computer Vision: Implementation and Applications,
Morgan & Claypool Publishers, San Rafael, CA, U.S., April, 2016.
ISBN 9781627054584 (print), ISBN 9781627054980 (E-book)

K. Kanatani, Y. Sugaya, and Y. Kanazawa,
Guide to 3D Vision Computation: Geometric Analysis and Implementation.
Springer International, Cham, Switzerland, December, 2016.
ISBN 978-3-319-48492-1 (print), ISBN 978-3-319-48943-8 (E-book)

Introduction

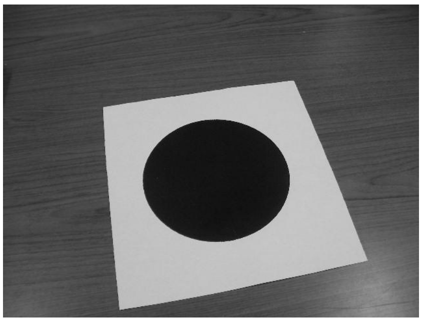
Ellipse fitting

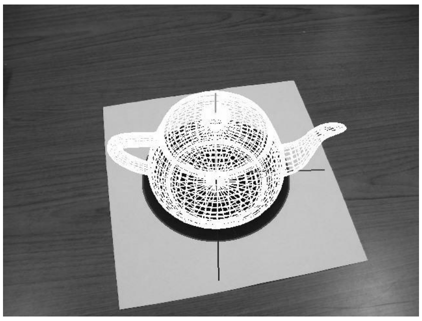


- Circular objects are projected as ellipses in images.
- By fitting ellipses, we can detect circular objects in the scene.
  - It is also used for detecting objects of approximately elliptic shape, e.g., human faces.
- Circles are often used as markers for *camera calibration*.
- Ellipse fitting provides a mathematical basis of various problems, including computation of *fundamental matrices* and *homographies*.

From the fitted ellipse, we can compute the 3-D position of the circular object in the scene.

Ellipse-based 3-D analysis



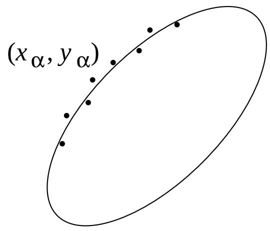


Ellipse representation

Task: Fit an ellipse in the form of

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2 F = 0,$$

to noisy data points (x_α, y_α), $\alpha = 1, ..., N$.



- f_0: scaling constant to make x_α/f_0 and y_α/f_0 have orders $O(1)$.
- For removing scale indeterminacy, the coefficients need to be normalized:

  (1) $F = 1$,
  (2) $A + C = 1$,
  (3) $A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1$, (→ We adopt this)
  (4) $A^2 + B^2 + C^2 + D^2 + E^2 = 1$,
  (5) $A^2 + 2B^2 + C^2 = 1$,
  (6) $AC - B^2 = 1$.

Vector representation

Define

$$\boldsymbol{\xi}_\alpha = \begin{pmatrix} x_\alpha^2 \\ 2x_\alpha y_\alpha \\ y_\alpha^2 \\ 2f_0 x_\alpha \\ 2f_0 y_\alpha \\ f_0^2 \end{pmatrix}, \qquad \boldsymbol{\theta} = \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \end{pmatrix}.$$

Then,

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F = 0 \qquad \Leftrightarrow \qquad (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) = 0,$$

$$A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1 \qquad \Leftrightarrow \qquad \|\boldsymbol{\theta}\| = 1.$$

Task: Find a unit vector $\boldsymbol{\theta}$ such that

$$(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) \approx 0, \qquad \alpha = 1, ..., N.$$

Least squares (LS) approach

The simplest and the most naive method is the *least squares* (*LS*).

---

1. Compute the 6×6 matrix
$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top.$$
2. Solve the eigenvalue problem
$$\boldsymbol{M}\boldsymbol{\theta} = \lambda\boldsymbol{\theta},$$
and return the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.

Motivation: We minimize the *algebraic distance*:

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 = \frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{\theta}^\top \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top \boldsymbol{\theta} = (\boldsymbol{\theta}, \underbrace{\Big(\frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top\Big)}_{\equiv \boldsymbol{M}}\boldsymbol{\theta}) = (\boldsymbol{\theta}, \boldsymbol{M}\boldsymbol{\theta}).$$

- The computation is very easy, and the solution is immediately obtained.
  - Widely used since the 1970s.
- But produces a small and flat ellipse very different from the true shape.
  - In particular, when the input points cover a small part of the ellipse.

How can we improve the accuracy?

- The reason for the poor accuracy is that the *properties of image noise* are not considered.
  - We need to consider the statistical properties of noise.

Noise assumption

Let $\bar{x}_\alpha$ and $\bar{y}_\alpha$ be the true values of observed x_α and y_α:

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, \qquad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha.$$

Then,

$$\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta_1\boldsymbol{\xi}_\alpha + \Delta_2\boldsymbol{\xi}_\alpha.$$

- $\bar{\boldsymbol{\xi}}_\alpha$: the true value of $\boldsymbol{\xi}_\alpha$
- $\Delta_1\boldsymbol{\xi}_\alpha$: noise term linear in Δx_α and Δy_α
- $\Delta_2\boldsymbol{\xi}_\alpha$: noise term quadratic in Δx_α and Δy_α

$$\bar{\boldsymbol{\xi}}_\alpha = \begin{pmatrix} \bar{x}_\alpha^2 \\ 2\bar{x}_\alpha\bar{y}_\alpha \\ \bar{y}_\alpha^2 \\ 2f_0\bar{x}_\alpha \\ 2f_0\bar{y}_\alpha \\ f_0^2 \end{pmatrix}, \qquad \Delta_1\boldsymbol{\xi}_\alpha = \begin{pmatrix} 2\bar{x}_\alpha\Delta x_\alpha \\ 2\Delta x_\alpha\bar{y}_\alpha + 2\bar{x}_\alpha\Delta y_\alpha \\ 2\bar{y}_\alpha\Delta y_\alpha \\ 2f_0\Delta x_\alpha \\ 2f_0\Delta y_\alpha \\ 0 \end{pmatrix}, \qquad \Delta_2\boldsymbol{\xi}_\alpha = \begin{pmatrix} \Delta x_\alpha^2 \\ 2\Delta x_\alpha\Delta y_\alpha \\ \Delta y_\alpha^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Covariance matrix

The noise terms Δx_α and Δy_α are regarded as independent Gaussian random variables of mean 0 and variance σ^2:

$$E[\Delta x_\alpha] = E[\Delta y_\alpha] = 0, \qquad E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = \sigma^2, \qquad E[\Delta x_\alpha \Delta y_\alpha] = 0.$$

The covariance matrix of $\boldsymbol{\xi}_\alpha$ is defined by

$$V[\boldsymbol{\xi}_\alpha] = E[\Delta_1\boldsymbol{\xi}_\alpha \Delta_1\boldsymbol{\xi}_\alpha^\top].$$

Then,

$$V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha], \qquad V_0[\boldsymbol{\xi}_\alpha] = 4\begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & \bar{x}_\alpha^2+\bar{y}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & f_0\bar{y}_\alpha & f_0\bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- σ^2: *noise level*
- $V_0[\boldsymbol{\xi}_\alpha]$: *normalized covariance matrix*
- The true values $\bar{x}_\alpha$ and $\bar{y}_\alpha$ are replaced by their observations x_α and y_α in actual computation.
  - Does not affect the final results.

Ellipse fitting approaches

Algebraic methods

- We *solve* an algebraic equation for computing $\boldsymbol{\theta}$.
  - The solution may or may not minimize any cost function.
- Our task is to find *a good equation to solve*.
  - The resulting solution $\boldsymbol{\theta}$ should be as close to its true value $\bar{\boldsymbol{\theta}}$ as possible.
- We need detailed statistical error analysis.

Geometric methods

- We *minimize* some cost function J.
  - The solution is uniquely determined once the cost J is set.
- Our task is to find *a good cost to minimize*.
  - The minimizing $\boldsymbol{\theta}$ should be close to its true value $\bar{\boldsymbol{\theta}}$.
  - We need to consider the *geometry* of the ellipse and the data points.
- We need a convenient minimization algorithm.
  - Minimization of a given cost is not always easy.

Algebraic Fitting

Iterative reweight

---

1. Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha = 1$, $\alpha = 1$, ..., N.
2. Compute the 6×6 matrix
$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top.$$
3. Solve the eigenvalue problem
$$\boldsymbol{M}\boldsymbol{\theta} = \lambda\boldsymbol{\theta},$$
and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update W_α and $\boldsymbol{\theta}$ to
$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$
and go back to Step 2.

Motivation of iterative reweight

Minimize the weighted sum of squares

$$\frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha(\boldsymbol{\theta}, \boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top\boldsymbol{\theta}) = (\boldsymbol{\theta}, \underbrace{\Big(\frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top\Big)}_{\equiv \boldsymbol{M}}\boldsymbol{\theta}) = (\boldsymbol{\theta}, \boldsymbol{M}\boldsymbol{\theta}).$$

- This is minimized by the unit eigenvector of $\boldsymbol{M}$ for the smallest eigenvalue.
- The weight is W_α is optimal if it is inversely proportional to the variance of each term.
  - Ideally, $W_\alpha = 1/(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})$:

$$E[(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2] = E[(\boldsymbol{\theta}, \Delta_1\boldsymbol{\xi}_\alpha\Delta_1\boldsymbol{\xi}_\alpha^\top\boldsymbol{\theta})] = (\boldsymbol{\theta}, \underbrace{E[\Delta_1\boldsymbol{\xi}_\alpha\Delta_1\boldsymbol{\xi}_\alpha^\top]}_{=\sigma^2 V_0[\boldsymbol{\xi}_\alpha]}\boldsymbol{\theta}) = \sigma^2(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta}),$$

- The true $\boldsymbol{\theta}$ is unknown, so the weight is iteratively updated.
- The iteration starts from the LS solution.

Renormalization (Kanatani 1993)

---

1. Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha = 1$, $\alpha = 1, ..., N$.
2. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \qquad \boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha V_0[\boldsymbol{\xi}_\alpha].$$

3. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

   and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the smallest generalized eigenvalue λ.
4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update W_α and $\boldsymbol{\theta}$ to

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

Motivation of renormalization

- From $(\bar{\boldsymbol{\xi}}_\alpha, \boldsymbol{\theta}) = 0$ or $\bar{\boldsymbol{\xi}}_\alpha^\top \boldsymbol{\theta} = 0$, we see that $\bar{\boldsymbol{M}}\boldsymbol{\theta} = \boldsymbol{0}$ for $\bar{\boldsymbol{M}} = (1/N)\sum_{\alpha=1}^N W_\alpha \bar{\boldsymbol{\xi}}_\alpha \bar{\boldsymbol{\xi}}_\alpha^\top$.
  - If $\bar{\boldsymbol{M}}$ is known, $\boldsymbol{\theta}$ is given by its eigenvector for eigenvalue 0, but $\bar{\boldsymbol{M}}$ is unknown.
- The expectation of $\boldsymbol{M}$ is

$$
\begin{aligned}
E[\boldsymbol{M}] &= E[\frac{1}{N}\sum_{\alpha=1}^N W_\alpha (\bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha)(\bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha)^\top] = \bar{\boldsymbol{M}} + E[\frac{1}{N}\sum_{\alpha=1}^N W_\alpha \Delta\boldsymbol{\xi}_\alpha \Delta\boldsymbol{\xi}_\alpha^\top] \\
&= \bar{\boldsymbol{M}} + \frac{1}{N}\sum_{\alpha=1}^N W_\alpha \underbrace{E[\Delta\boldsymbol{\xi}_\alpha \Delta\boldsymbol{\xi}_\alpha^\top]}_{=\sigma^2 V_0[\boldsymbol{\xi}_\alpha]} = \bar{\boldsymbol{M}} + \sigma^2 \underbrace{\frac{1}{N}\sum_{\alpha=1}^N W_\alpha V_0[\boldsymbol{\xi}_\alpha]}_{=\boldsymbol{N}} = \bar{\boldsymbol{M}} + \sigma^2 \boldsymbol{N}.
\end{aligned}
$$

- $\bar{\boldsymbol{M}} = E[\boldsymbol{M}] - \sigma^2 \boldsymbol{N} \approx \boldsymbol{M} - \sigma^2 \boldsymbol{N}$, so we solve $(\boldsymbol{M} - \sigma^2 \boldsymbol{N})\boldsymbol{\theta} = \boldsymbol{0}$ or $\boldsymbol{M}\boldsymbol{\theta} = \sigma^2 \boldsymbol{N}\boldsymbol{\theta}$.
  - We solve $\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta}$ for the smallest absolute value λ.
- The optimal weight $W_\alpha = 1/(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})$ is unknown, so it is iteratively updated.
- The iterations start from $W_\alpha = 1$, i.e, initially we solve $\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta}$ for $\boldsymbol{M} = (1/N)\sum_{\alpha=1}^N \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top$ and $\boldsymbol{N} = (1/N)\sum_{\alpha=1}^N V_0[\boldsymbol{\xi}_\alpha]$. $\rightarrow$ *Taubin method*.

Taubin method (Taubin 1991)

---

1. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \qquad \boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha].$$

2. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

   and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the smallest generalized eigenvalue λ.

- This method was derived by Taubin (1991) heuristically without considering statistical properties of noise.

Hyper-renormalization (Kanatani et al. 2012)

---

1. Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha = 1$, $\alpha = 1$, ..., N.
2. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top,$$

$$\boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}^\top]\Big) - \frac{1}{N^2}\sum_{\alpha=1}^{N} W_\alpha^2 \Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big).$$

   - $\mathcal{S}[\,\cdot\,]$: symmetrization operator ($\mathcal{S}[\boldsymbol{A}] = (\boldsymbol{A} + \boldsymbol{A}^\top)/2$).
   - $\boldsymbol{e} = (1, 0, 1, 0, 0, 0)^\top$
   - $\boldsymbol{M}_5^-$: pseudoinverse of rank 5:

$$\boldsymbol{M} = \mu_1 \boldsymbol{\theta}_1 \boldsymbol{\theta}_1^\top + \cdots + \underbrace{\mu_6}_{\approx 0} \boldsymbol{\theta}_6 \boldsymbol{\theta}_6^\top \quad \rightarrow \quad \boldsymbol{M}_5^- = \frac{\boldsymbol{\theta}_1 \boldsymbol{\theta}_1^\top}{\mu_1} + \cdots + \frac{\boldsymbol{\theta}_5 \boldsymbol{\theta}_5^\top}{\mu_5}.$$

3. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

   and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$, return $\boldsymbol{\theta}$ and stop. Else, update Else, update W_α and $\boldsymbol{\theta}$ to

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

- This method was derived so that the resulting solution has the *highest accuracy*.
- The iterations start from $W_\alpha = 1$. $\rightarrow$ *HyperLS*.

HyperLS (Rangarajan and Kanatani 2009)

---

1. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top,$$

$$\boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N}\Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}^\top]\Big) - \frac{1}{N^2}\sum_{\alpha=1}^{N}\Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha)V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big).$$

2. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the smallest generalized eigenvalue λ.

- This method was derived so that the *highest accuracy* is achieved among all *non-iterative* schemes.

Summary of algebraic methods

All algebraic methods solve

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ involve observed data. They may or may not involve $\boldsymbol{\theta}$.

$$\boldsymbol{M} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, & \text{(LS, Taubin, HyperLS)} \\ \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} \dfrac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}. & \text{(iterative reweight, renormalization, hyper-renormalization)} \end{cases}$$

$$\boldsymbol{N} = \begin{cases} \boldsymbol{I}, & \text{(LS, iterative reweight)} \\ \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha], & \text{(Taubin)} \\ \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} \dfrac{V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, & \text{(renormalization)} \\ \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} \Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}^\top]\Big) - \dfrac{1}{N^2}\displaystyle\sum_{\alpha=1}^{N} \Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big), & \text{(HypeLS)} \\ \dfrac{1}{N}\displaystyle\sum_{\alpha=1}^{N} \dfrac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})} \Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}^\top]\Big) - \dfrac{1}{N^2}\displaystyle\sum_{\alpha=1}^{N} \dfrac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})^2} \Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big). & \text{(hyper-renormalization)} \end{cases}$$

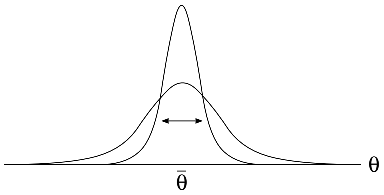
- If $\boldsymbol{M}$ and $\boldsymbol{N}$ do not involve $\boldsymbol{\theta}$, we solve the generalized eigenvalue problem $\boldsymbol{M}\boldsymbol{\theta} = \lambda\boldsymbol{N}\boldsymbol{\theta}$.
  - No iterations are necessary
- If $\boldsymbol{M}$ and $\boldsymbol{N}$ involve $\boldsymbol{\theta}$, we iteratively solve the generalized eigenvalue problem.
  - The weight is iteratively updated.
- $\boldsymbol{N}$ is generally *not* positive definite. $\rightarrow$ We solve $\boldsymbol{N}\boldsymbol{\theta} = (1/\lambda)\boldsymbol{M}\boldsymbol{\theta}$ instead.
  - $\boldsymbol{M}$ is always positive definite for noisy data.

Characterization of algebraic methods

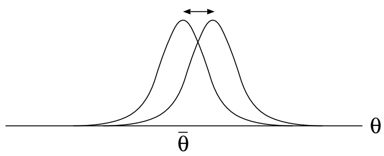
- Problem:

$$\boldsymbol{M}(\boldsymbol{\theta})\boldsymbol{\theta} = \lambda \boldsymbol{N}(\boldsymbol{\theta})\boldsymbol{\theta}.$$

- The data are noisy. → The solution has a distribution.



$\boldsymbol{M}(\boldsymbol{\theta})$ controls the *covariance* of the solution.



$\boldsymbol{N}(\boldsymbol{\theta})$ determines the *bias* of the solution.

- Issue:
  - What $\boldsymbol{M}(\boldsymbol{\theta})$ minimizes the covariance the most?
  - What $\boldsymbol{N}(\boldsymbol{\theta})$ minimizes the bias the most?
- Solution:

$$\boldsymbol{M}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})},$$

The covariance reaches the *theoretical accuracy bound* up to $O(\sigma^4)$

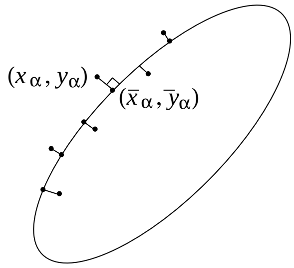
$$\boldsymbol{N}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}\Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}^\top]\Big) - \frac{1}{N^2}\sum_{\alpha=1}^{N} \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})^2}\Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}_5^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big),$$

The bias is 0 up to $O(\sigma^4)$

- *Hyper-renormalization achieves both.*

Geometric Fitting

Geometric approach



Minimize the *geometric distance* S:

$$S = \frac{1}{N}\sum_{\alpha=1}^{N}\Big((x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2\Big) = \frac{1}{N}\sum_{\alpha=1}^{N} d_\alpha^2,$$

i.e., the average of the square distances d_α^2 from data points (x_α, y_α) to the nearest points $(\bar{x}_\alpha, \bar{y}_\alpha)$ on the ellipse.

The computation is very difficult:

- S is minimized subject to the constraint $(\bar{\boldsymbol{\xi}}_\alpha, \boldsymbol{\theta}) = 0$.
  - S does *not* contain $\boldsymbol{\theta}$, for which S is minimized.
  - $\boldsymbol{\theta}$ is contained in the *constraint* $(\bar{\boldsymbol{\xi}}_\alpha, \boldsymbol{\theta}) = 0$.
- The minimization is done in the *joint space* of $\boldsymbol{\theta}$ and $(\bar{x}_1, \bar{y}_1), ..., (\bar{x}_N, \bar{y}_N)$.
  - $\boldsymbol{\theta}$: *structural parameter*
  - $(\bar{x}_\alpha, \bar{y}_\alpha)$: *nuisance parameters*

Sampson error

If (x_α, y_α) is close to the ellipse, the square distance d_α^2 is approximated by

$$d_\alpha^2 = (x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2 \approx \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})},$$

Hence, the geometric distance S is approximated by the *Sampson error*:

$$J = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}.$$

- Minimization is done in the space of $\boldsymbol{\theta}$.
  - *Unconstrained* minimization without nuisance parameters.

FNS: Fundamental Numerical Scheme (Chojnacki et al. 2000)

---

1. Let $\boldsymbol{\theta} = \boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha = 1$.
2. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \qquad \boldsymbol{L} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha^2 (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 V_0[\boldsymbol{\xi}_\alpha].$$

3. Let

$$\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}.$$

4. Solve the eigenvalue problem

$$\boldsymbol{X}\boldsymbol{\theta} = \lambda\boldsymbol{\theta},$$

   and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
5. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update W_α and $\boldsymbol{\theta}$ to

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

Motivation of FNS

- We can see that
$$\nabla_{\boldsymbol{\theta}} J = 2(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{\theta} = 2\boldsymbol{X}\boldsymbol{\theta}.$$
- We iteratively solve the eigenvalue problem $\boldsymbol{X}\boldsymbol{\theta} = \lambda\boldsymbol{\theta}$.
- When the iterations have converged, it can be proved that $\lambda = 0$.
  - The solution satisfies $\nabla_{\boldsymbol{\theta}} J = \mathbf{0}$.
- Initially $\boldsymbol{L} = \boldsymbol{O}$. $\rightarrow$ The iterations start from the LS solution.

Geometric distance minimization (Kanatani and Sugaya 2010)

---

1. Let $J_0^* = \infty$, $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = 0$.
2. Compute the normalized covariance matrix $V_0[\hat{\boldsymbol{\xi}}_\alpha]$ using $\hat{x}_\alpha$ and $\hat{y}_\alpha$, and let

$$\boldsymbol{\xi}_\alpha^* = \begin{pmatrix} \hat{x}_\alpha^2 + 2\hat{x}_\alpha\tilde{x}_\alpha \\ 2(\hat{x}_\alpha\hat{y}_\alpha + \hat{y}_\alpha\tilde{x}_\alpha + \hat{x}_\alpha\tilde{y}_\alpha) \\ \hat{y}_\alpha^2 + 2\hat{y}_\alpha\tilde{y}_\alpha \\ 2f_0(\hat{x}_\alpha + \tilde{x}_\alpha) \\ 2f_0(\hat{y}_\alpha + \tilde{y}_\alpha) \\ f_0 \end{pmatrix}.$$

3. Compute the $\boldsymbol{\theta}$ that minimizes the *modified Sampson error*

$$J^* = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha^*, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})}.$$

4. Update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\hat{x}_\alpha$ and $\hat{y}_\alpha$ to

$$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{2(\boldsymbol{\xi}_\alpha^*, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix}, \qquad \hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \qquad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha.$$

5. Compute

$$J^* = \frac{1}{N}\sum_{\alpha=1}^{N}(\tilde{x}_\alpha^2 + \tilde{y}_\alpha^2).$$

   If $J^* \approx J_0$, return $\boldsymbol{\theta}$ and stop. Else, let $J_0 \leftarrow J^*$ and go back to Step 2.

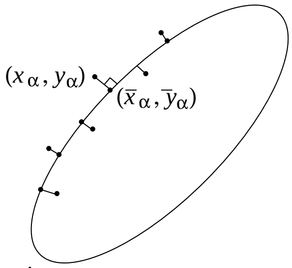
Motivation

- We first minimize the Sampson error J, say by FNS, and modify the data $\boldsymbol{\xi}_\alpha$ to $\boldsymbol{\xi}_\alpha^*$ using the computed solution $\boldsymbol{\theta}$.
- Regarding them as data, we define the modified Sampson error J^* and minimize it, say by FNS.
- If this is repeated, the modified Sampson error J^* eventually coincides with the geometric distance S.
  - We we obtain the solution that minimize S.

However,

- The Sampson error minimization solution and the geometric distance minimization solution usually coincide up to several significant digits.
- Minimizing the Sampson error is *practically the same* as minimizing the geometric distance.

Bias removal



- The geometric fitting solution $\hat{\boldsymbol{\theta}}$ is known to be *biased*:

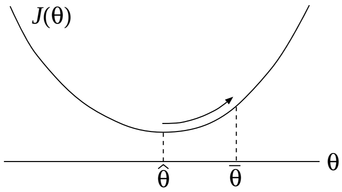
$$E[\boldsymbol{\theta}] \neq \bar{\boldsymbol{\theta}}.$$

  - An ellipse has a *convex* shape.
  - Points are more likely to move outside the ellipse by random noise.
- If we write

$$\hat{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} + \Delta_1\boldsymbol{\theta} + \Delta_2\boldsymbol{\theta} + \cdots, \qquad (\Delta_k\boldsymbol{\theta} : k\text{th order in noise})$$

  we have $E[\Delta_1\boldsymbol{\theta}] = \mathbf{0}$ but $E[\Delta_2\boldsymbol{\theta}] \neq \mathbf{0}$.
- *Hyperaccurate correction*: If we can evaluate $E[\Delta_2\boldsymbol{\theta}]$, we obtain a better solution

$$\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - E[\Delta_2\boldsymbol{\theta}].$$



Hyperaccurate correction (Kanatani 2006)

---

1. Compute $\boldsymbol{\theta}$ by FNS.
2. Estimate σ^2 by
$$\hat{\sigma}^2 = \frac{(\boldsymbol{\theta}, \boldsymbol{M}\boldsymbol{\theta})}{1 - 5/N},$$
using the value of $\boldsymbol{M}$ after the FNS iterations have converged.
3. Compute the correction term
$$\Delta_c\boldsymbol{\theta} = -\frac{\hat{\sigma}^2}{N}\boldsymbol{M}_5^- \sum_{\alpha=1}^{N} W_\alpha(\boldsymbol{e}, \boldsymbol{\theta})\boldsymbol{\xi}_\alpha + \frac{\hat{\sigma}^2}{N^2}\boldsymbol{M}_5^- \sum_{\alpha=1}^{N} W_\alpha^2(\boldsymbol{\xi}_\alpha, \boldsymbol{M}_5^- V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})\boldsymbol{\xi}_\alpha,$$
where using the value of W_α after the FNS iterations have converged, where $\boldsymbol{M}_5^-$ is the pseudoinverse of $\boldsymbol{M}$ of rank 5.
4. Correct $\boldsymbol{\theta}$ to
$$\boldsymbol{\theta} \leftarrow \mathcal{N}[\boldsymbol{\theta} - \Delta_c\boldsymbol{\theta}],$$
where $\mathcal{N}[\,\cdot\,]$ is a normalization operation.

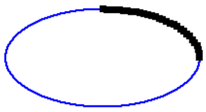
- Since the bias is $O(\sigma^4)$, the solution has the same accuracy as hyper-renormalization.

Experimental Comparisons

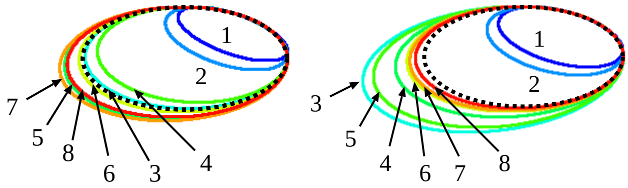
Some examples

Gaussian noise of standard deviation σ is added (the dashed lines: the true shape)

30 data points



Fitting examples for $\sigma = 0.5$

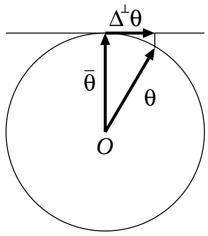


1. LS
2. iterative reweight
3. Taubin
4. renormalization
5. HyperLS
6. hyper-renormalization
7. FNS
8. FNS + hyperaccurate correction

method		2	4	6	7/8
number of iterations	left	4	4	4	9
	right	4	4	4	8

- Methods 1, 3, and 5 are algebraic, hence non-iterative.
- Methods 7 and 8 have the same complexity.
  - Hyperaccurate correction is an analytical procedure.
- FNS requires about twice as many iterations.

Statistical comparison



$\bar{\boldsymbol{\theta}}$: true value (unit vector) $\quad \hat{\boldsymbol{\theta}}$: computed value (unit vector)

- The deviation is measured by the orthogonal error component:

$$\Delta^{\perp}\boldsymbol{\theta} = \boldsymbol{P}_{\bar{\boldsymbol{\theta}}}\hat{\boldsymbol{\theta}}, \qquad \boldsymbol{P}_{\bar{\boldsymbol{\theta}}} \equiv \boldsymbol{I} - \bar{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}}^{\top}.$$

- The bias B and the RMS error D are measured over M (= 10000) trials:

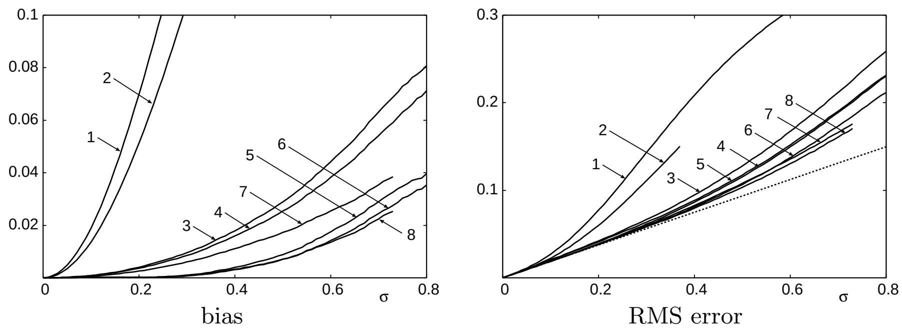
$$B = \left\| \frac{1}{M}\sum_{a=1}^{M} \Delta^{\perp}\boldsymbol{\theta}^{(a)} \right\|, \qquad D = \sqrt{\frac{1}{M}\sum_{a=1}^{M} \|\Delta^{\perp}\boldsymbol{\theta}^{(a)}\|^2}.$$

- *KCR lower bound*:

$$D \geq \frac{\sigma}{\sqrt{N}}\sqrt{\mathrm{tr}\Big(\frac{1}{N}\sum_{\alpha=1}^{N} \frac{\bar{\boldsymbol{\xi}}_\alpha \bar{\boldsymbol{\xi}}_\alpha^{\top}}{(\bar{\boldsymbol{\theta}}, V_0[\boldsymbol{\xi}_\alpha]\bar{\boldsymbol{\theta}})}\Big)^{-}}$$

Bias and RMS error

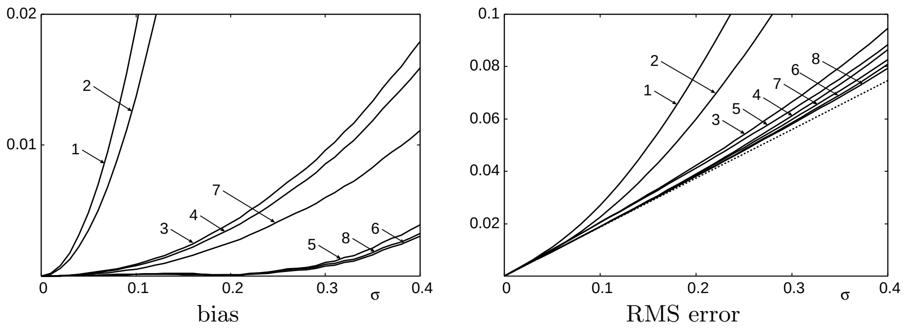
Simulation over independent 10000 trials for different σ.
(the dotted lines: the KCR lower bound)



1. LS
2. iterative reweight
3. Taubin
4. renormalization
5. HyperLS
6. hyper-renormalization
7. FNS
8. FNS + hyperaccurate correction

- LS and iterative reweight has large bias and hence large RMS errors.
- LS has some bias, which is reduced by hyperaccurate correction to a large extent.
- The bias of HyperLS and hyper-renormalization is very small.
- The iterations of iterative reweight and FNS do not converge for large σ.

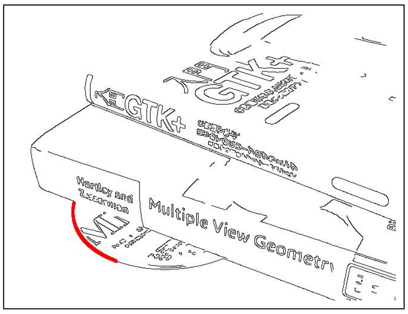
Bias and RMS error (enlargement)

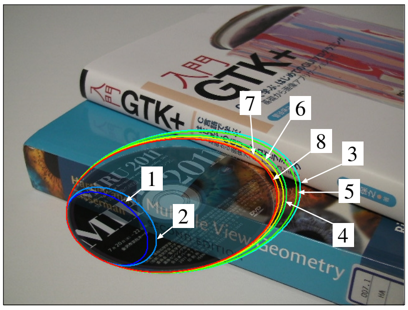


1. LS
2. iterative reweight
3. Taubin
4. renormalization
5. HyperLS
6. hyper-renormalization
7. FNS
8. FNS + hyperaccurate correction

- Hyper-renormalization outperforms FNS for small σ.
- The highest accuracy is given by hyperaccurate correction of FNS.
  - However, the FNS iterations may not converge for large σ.
- Hyper-renormalization is robust to noise.
  - The initial solution (HyperLS) is already very accurate.
  - It is the best method in practice.

Real image example:





1. LS
2. iterative reweight
3. Taubin
4. renormalization
5. HyperLS
6. hyper-renormalization
7. FNS
8. FNS + hyperaccurate correction

method	2	4	6	7/8
# of iter.	4	3	3	6

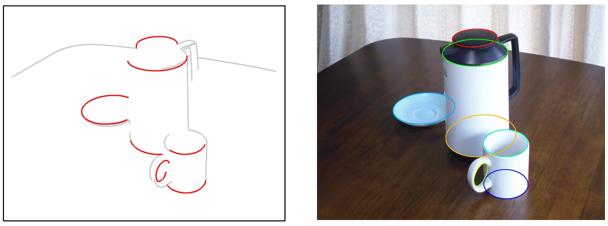
- Methods 1, 3, and 5 are algebraic, hence non-iterative.
- Methods 7 and 8 have the same complexity.
  - Hyperaccurate correction is an analytical procedure.
- ML requires about twice as many iterations.

Robust Fitting

When does ellipse fitting fail?

Superfluous data

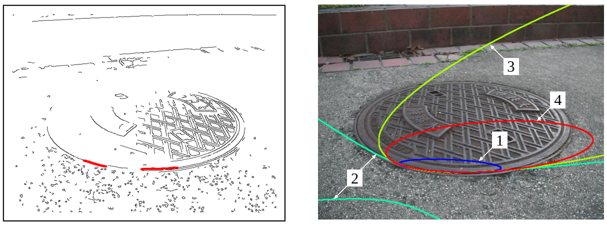
- Some segments may belong to other objects.
  - *Inliers*: segments that belong to the object of interest
  - *Outliers*: segments that belong to different objects.



Difficult to find outliers if they are smoothly connected to inliers

Scarcity of information

- If the segment is too short and/or noisy, a hyperbola can be fit.
  - How can we modify a hyperbola to an ellipse?
  - How can we produce only an ellipse? $\longrightarrow$ *ellipse-specific method*



Information is too scares to produce a good fit by any method.

RANSAC

Find an ellipse such that *the number of points close to it is as large as possible*.

---

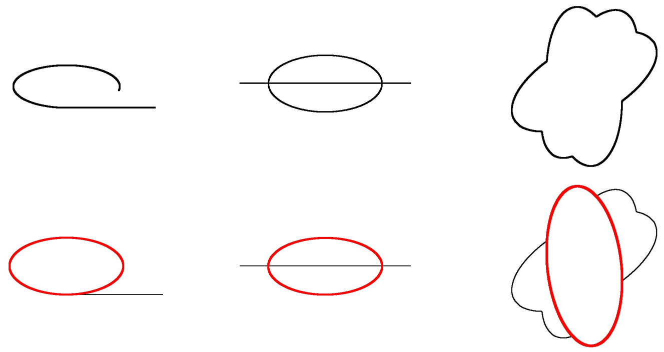
1. Randomly select five points from the input sequence, and let $\boldsymbol{\xi}_1$, ..., $\boldsymbol{\xi}_5$ be their vectors
2. Compute the unit eigenvector $\boldsymbol{\theta}$ of the matrix

$$\boldsymbol{M}_5 = \sum_{\alpha=1}^{5} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top,$$

   for the smallest eigenvalue, and store it as a candidate.
3. Let n be the number of points in the input sequence that satisfy

$$\Big((x-\bar{x})^2 + (y-\bar{y})^2 \approx\Big) \frac{(\boldsymbol{\xi}, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}]\boldsymbol{\theta})} < d^2,$$

   where d is a threshold for admissible deviation from ellipse, e.g., $d = 2$ (pixels). Store that n.
4. Select a new set of five points from the input sequence, and do the same. Repeat this many times, and return from among the stored candidate ellipses the one for which n is the largest.



Ellipse-specific method of Fitzgibbon et al. (1999)

The equation $Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2 F = 0$ represents an ellipse if and only if

$$AC - B^2 > 0.$$

---

1. Compute the 6×6 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \qquad \boldsymbol{N} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

2. Solve the generalized eigenvalue problem

$$\boldsymbol{M\theta} = \lambda \boldsymbol{N\theta},$$

and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the smallest generalized eigenvalue λ.

Motivation

- We minimize the algebraic distance $(1/N)\sum_{\alpha=1}^{N}(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2$ subject to

$$\Big(AC - B^2 =\Big)(\boldsymbol{\theta}, \boldsymbol{N\theta}) = 1.$$

- $\boldsymbol{N}$ is not positive definite.
  → We solve $\boldsymbol{N\theta} = (1/\lambda)\boldsymbol{M\theta}$ instead for the largest eigenvalue.

Random sampling of Masuzaki et al. (2013)

---

1. Fit an ellipse by the standard method. Stop, if the solution $\boldsymbol{\theta}$ satisfies

$$\theta_1\theta_3 - \theta_2^2 > 0.$$

2. Else, randomly select five points among the sequence. Let $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, ..., $\boldsymbol{\xi}_5$ be their vector representations.
3. Compute the unit eigenvector $\boldsymbol{\theta}$ of

$$\boldsymbol{M}_5 = \sum_{\alpha=1}^{5} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top,$$

   for the smallest eigenvalue.
4. If the resulting $\boldsymbol{\theta}$ does not define an ellipse, discard it. Newly select another set of five points randomly and do the same.
5. If the resulting $\boldsymbol{\theta}$ defines an ellipse, keep it as a candidate and compute its Sampson error.
6. Repeat this many times, and return from among the candidates the one with the smallest Sampson error J.

- We can obtain an ellipse less biased than the solution of the method of Fitzgibbon et al.

Penalty method of Szpak et al. (2015)

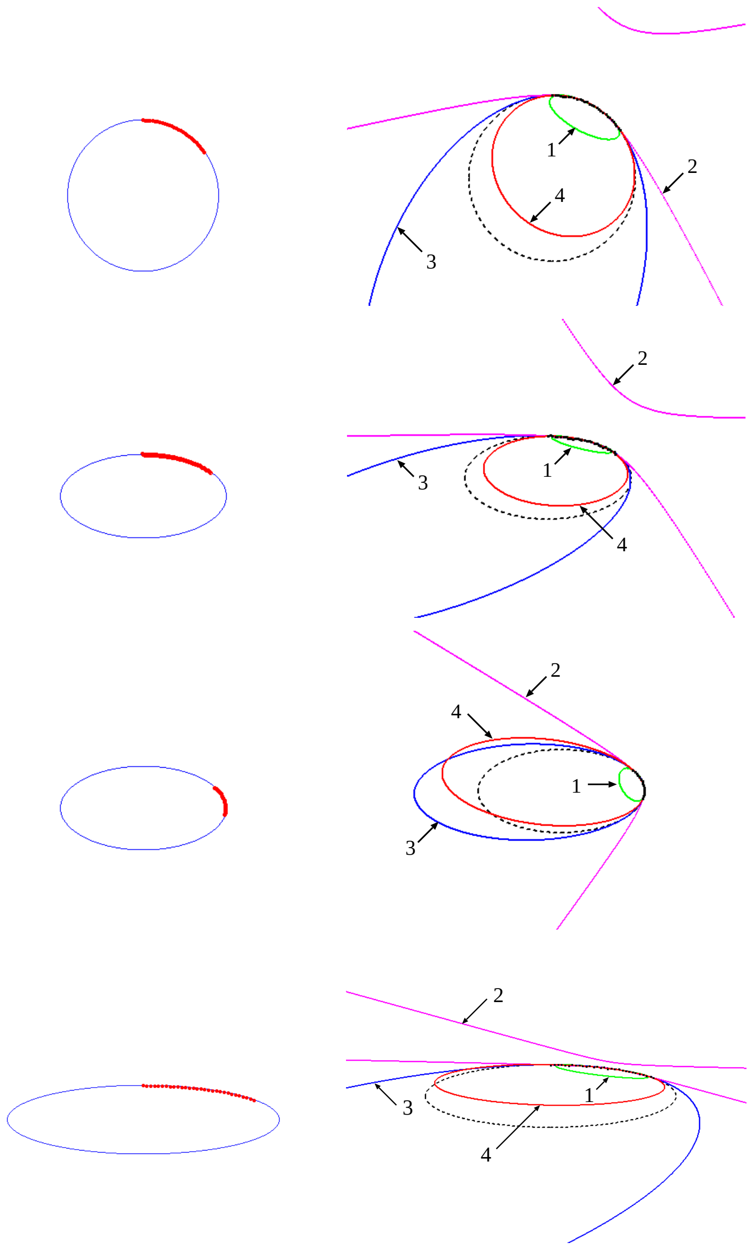
Minimize

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})} + \frac{\lambda\|\boldsymbol{\theta}\|^4}{(\boldsymbol{\theta}, \boldsymbol{N}\boldsymbol{\theta})^2},$$

using the Levenberg–Marquardt method.

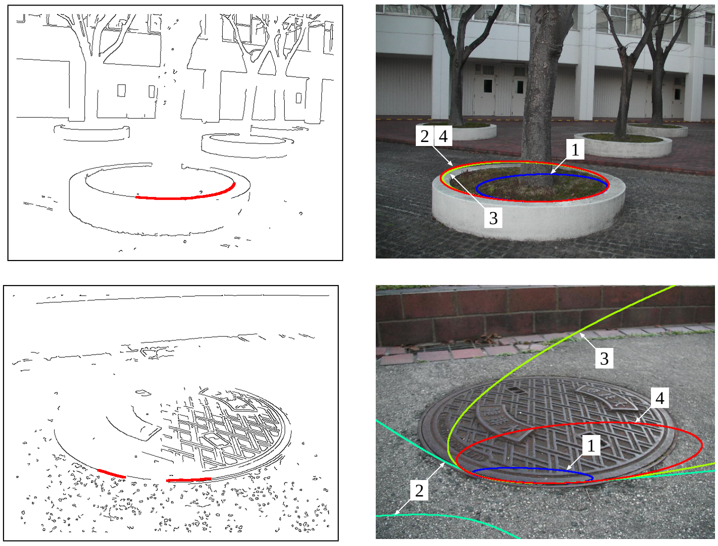
- The first term: the Sampson error.
- $(\boldsymbol{\theta}, \boldsymbol{N}\boldsymbol{\theta}) = 0$ at ellipse-hyperbola boundaries.
- λ: regularization constant

Comparison simulations



1. Fitzgibbon et al.
   → small flat ellipse
2. hyper-renormalization
   → hyperbola
3. penalty method
   → large ellipse close to 2
4. random sampling
   → between 1 and 3.

Real image examples



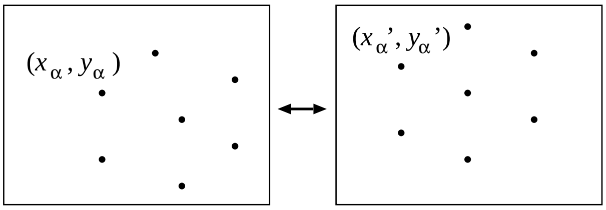
- Fitzgibbon et al. [1] produces a mall flat ellipse.
- If hyper-renormalization [2] returns an ellipse, random sampling [4] returns the same ellipse, and the penalty method [3] fits an ellipse close to it.
- If hyper-renormalization [2] returns a hyperbola, the penalty method [3] fits a large ellipse close to it.
- Random sampling [4] fits somewhat a moderate ellipse.

Conclusion

- If hyper-renormalization returns a hyperbola, any ellipse specific method does not produce a reasonable ellipse.
  - Ellipse specific methods do not make practical sense.
  - Use random sampling if you need an ellipse by all means.

Fundamental Matrix Computation

Fundamental matrix



For two images of the same scene, the following *epipolar equation* holds:

$$(\begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \boldsymbol{F} \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}) = 0.$$

- f_0: scale factor ($\approx$ the size of the image)
- $\boldsymbol{F}$: *fundamental matrix*
- To remove scale indeterminacy, $\boldsymbol{F}$ is normalized to unit norm: $\|\boldsymbol{F}\|$ ($\equiv \sqrt{\sum_{i,j=1,3} F_{ij}^2}$) $= 1$

From the computed $\boldsymbol{F}$, we can reconstruct the 3-D structure of the scene.

Vector representation

$$
\left(\begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \boldsymbol{F}\begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}\right) = 0 \qquad \leftrightarrow \qquad (\boldsymbol{\xi}, \boldsymbol{\theta}) = 0, \qquad \boldsymbol{\xi} \equiv \begin{pmatrix} xx' \\ xy' \\ f_0 x \\ yx' \\ yy' \\ f_0 y \\ f_0 x' \\ f_0 y' \\ f_0^2 \end{pmatrix}, \qquad \boldsymbol{\theta} \equiv \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}.
$$

$$
\|\boldsymbol{F}\| = 1 \qquad \leftrightarrow \qquad \|\boldsymbol{\theta}\| = 1.
$$

Task: From noisy observations $\boldsymbol{\xi}_1$, ..., $\boldsymbol{\xi}_N$, estimate a unit vector $\boldsymbol{\theta}$ such that

$$
(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) \approx 0, \qquad \alpha = 1, ..., N.
$$

Noise assumption

$(\bar{x}_\alpha, \bar{y}_\alpha)$, $(\bar{x}'_\alpha, \bar{y}'_\alpha)$: true values of (x_α, y_α), (x'_α, y'_α).

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, \qquad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha, \qquad x'_\alpha = \bar{x}'_\alpha + \Delta x'_\alpha, \qquad y'_\alpha = \bar{y}'_\alpha + \Delta y'_\alpha.$$

Then,

$$\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta_1\boldsymbol{\xi}_\alpha + \Delta_2\boldsymbol{\xi}_\alpha.$$

- $\bar{\boldsymbol{\xi}}_\alpha$: true value of $\boldsymbol{\xi}_\alpha$
- $\Delta_1\boldsymbol{\xi}_\alpha$: noise term linear in Δx_α, Δy_α, $\Delta x'_\alpha$, and Δy_α'.
- $\Delta_2\boldsymbol{\xi}_\alpha$: noise term quadratic in Δx_α Δy_α, $\Delta x'_\alpha$, and Δy_α'.

$$\bar{\boldsymbol{\xi}} = \begin{pmatrix} \bar{x}_\alpha\bar{x}'_\alpha \\ \bar{x}_\alpha\bar{y}'_\alpha \\ f_0\bar{x}_\alpha \\ \bar{y}_\alpha\bar{x}'_\alpha \\ \bar{y}_\alpha\bar{y}'_\alpha \\ f_0\bar{y}_\alpha \\ f_0\bar{x}'_\alpha \\ f_0\bar{y}'_\alpha \\ f_0^2 \end{pmatrix}, \qquad \Delta_1\boldsymbol{\xi}_\alpha = \begin{pmatrix} \bar{x}'_\alpha\Delta x_\alpha + \bar{x}_\alpha\Delta x'_\alpha \\ \bar{y}'_\alpha\Delta x_\alpha + \bar{x}_\alpha\Delta y'_\alpha \\ f_0\Delta x_\alpha \\ \bar{x}'_\alpha\Delta y_\alpha + \bar{y}_\alpha\Delta x'_\alpha \\ f_0\Delta y_\alpha \\ f_0\Delta x'_\alpha \\ f_0\Delta y'_\alpha \\ 0 \end{pmatrix}, \qquad \Delta_2\boldsymbol{\xi}_\alpha = \begin{pmatrix} \Delta x_\alpha\Delta x'_\alpha \\ \Delta x_\alpha\Delta y'_\alpha \\ 0 \\ \Delta y_\alpha\Delta x'_\alpha \\ \Delta y_\alpha\Delta y'_\alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Covariance matrix

The noise terms Δx_α, Δy_α, $\Delta x'_\alpha$, and Δy_α are regarded as independent Gaussian random variables of mean 0 and variance σ^2:

$$E[\Delta x_\alpha] = E[\Delta y_\alpha] = E[\Delta x'_\alpha] = E[\Delta y'_\alpha] = 0, \quad E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = E[\Delta {x'_\alpha}^2] = E[\Delta {y'_\alpha}^2] = \sigma^2,$$

$$E[\Delta x_\alpha \Delta y_\alpha] = E[\Delta x'_\alpha \Delta y'_\alpha] = E[\Delta x_\alpha \Delta y'_\alpha] = E[\Delta x'_\alpha \Delta y_\alpha] = 0.$$

The covariance matrix of $\boldsymbol{\xi}_\alpha$ is defined by

$$V[\boldsymbol{\xi}_\alpha] = E[\Delta_1 \boldsymbol{\xi}_\alpha \Delta_1 \boldsymbol{\xi}_\alpha^\top].$$

Then,

$$V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha],$$

$$V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} \bar{x}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}'_\alpha \bar{y}'_\alpha & f_0 \bar{x}'_\alpha & \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & f_0 \bar{x}_\alpha & 0 & 0 \\ \bar{x}'_\alpha \bar{y}'_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha'^2 & f_0 \bar{y}'_\alpha & 0 & \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & f_0 \bar{x}_\alpha & 0 \\ f_0 \bar{x}'_\alpha & f_0 \bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & \bar{y}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}'_\alpha \bar{y}'_\alpha & f_0 \bar{x}'_\alpha & f_0 \bar{y}_\alpha & 0 & 0 \\ 0 & \bar{x}_\alpha \bar{y}_\alpha & 0 & \bar{x}'_\alpha \bar{y}'_\alpha & \bar{y}_\alpha^2 + \bar{y}_\alpha'^2 & f_0 \bar{y}'_\alpha & 0 & f_0 \bar{y}_\alpha & 0 \\ 0 & 0 & 0 & f_0 \bar{x}'_\alpha & f_0 \bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ f_0 \bar{x}_\alpha & 0 & 0 & f_0 \bar{y}_\alpha & 0 & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 \bar{x}_\alpha & 0 & 0 & f_0 \bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- σ^2: *noise level*
- $V_0[\boldsymbol{\xi}_\alpha]$: *normalized covariance matrix*

Fundamental matrix computation

algebraic methods

- non-iterative methods
  - least squares (LS), Taubin method, hyperLS
- iterative methods
  - iterative reweight, renormalization, hyper-renormalization

geometric methods

- Sampson error minimization (FNS)
- geometric error minimization
- hyperaccurate correction

However, ...

Rank constraint

The fundamental matrix $\boldsymbol{F}$ must have rank 0:

$$\det \boldsymbol{F} = 0$$

Existing three approaches:

a posteriori correction:

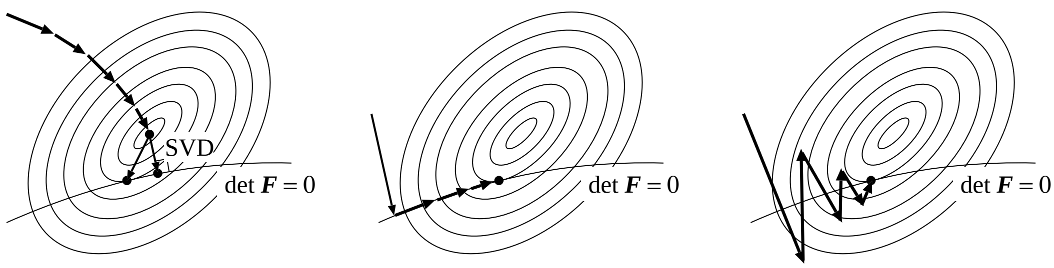
- SVD correction
- optimal correction

internal access:

Parameterize $\boldsymbol{F}$ such that $\det \boldsymbol{F} = 0$ is identically satisfied, and do optimization in the internal parameter space of a smaller dimension.

external access:

Do iteration in the external (redundant) space of $\boldsymbol{\theta}$ in such a way that $\boldsymbol{\theta}$ approaches the true value and yet $\det \boldsymbol{F} = 0$ holds at the time of convergence.



SDV correction

---

1. Compute $\boldsymbol{F}$ without considering the rank constraint.
2. Compute the SDV of $\boldsymbol{F}$:

$$\boldsymbol{F} = \boldsymbol{U} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \boldsymbol{V}^\top$$

3. Correct $\boldsymbol{F}$ to

$$\boldsymbol{F} \leftarrow \boldsymbol{U} \begin{pmatrix} \sigma_1/\sqrt{\sigma_1^2 + \sigma_2^2} & 0 & 0 \\ 0 & \sigma_2/\sqrt{\sigma_1^2 + \sigma_2^2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{V}^\top$$

- The norm $\|\boldsymbol{F}\|$ is scaled to 1

Optimal correction (Kanatani and Sugaya 2007)

---

1. Compute $\boldsymbol{\theta}$ without considering the rank constraint.
2. Compute the 9×9 matrix

$$\hat{\boldsymbol{M}} = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{P}_{\boldsymbol{\theta}}\boldsymbol{\xi}_\alpha)(\boldsymbol{P}_{\boldsymbol{\theta}}\boldsymbol{\xi}_\alpha)^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{P}_{\boldsymbol{\theta}} \equiv \boldsymbol{I} - \boldsymbol{\theta}\boldsymbol{\theta}^\top.$$

   $\boldsymbol{P}_{\boldsymbol{\theta}}$: projection matrix onto the space orthogonal to $\boldsymbol{\theta}$.
3. Compute the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_9$ $(= 0)$ of $\hat{\boldsymbol{M}}$ and the corresponding unit eigenvectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, ..., $\boldsymbol{u}_9$ $(= \boldsymbol{\theta})$. Then, define

$$V_0[\boldsymbol{\theta}] = \frac{1}{N}\Big(\frac{\boldsymbol{u}_1\boldsymbol{u}_1^\top}{\lambda_1} + \cdots + \frac{\boldsymbol{u}_8\boldsymbol{u}_8^\top}{\lambda_8}\Big).$$

4. Modify $\boldsymbol{\theta}$ to

$$\boldsymbol{\theta} \leftarrow \mathcal{N}[\boldsymbol{\theta} - \frac{(\boldsymbol{\theta}^\dagger, \boldsymbol{\theta})V_0[\boldsymbol{\theta}]\boldsymbol{\theta}^\dagger}{3(\boldsymbol{\theta}^\dagger, V_0[\boldsymbol{\theta}]\boldsymbol{\theta}^\dagger)}], \qquad \boldsymbol{\theta}^\dagger = \begin{pmatrix} \theta_5\theta_9 - \theta_8\theta_6 \\ \theta_6\theta_7 - \theta_9\theta_4 \\ \theta_4\theta_8 - \theta_7\theta_5 \\ \theta_8\theta_3 - \theta_2\theta_9 \\ \theta_9\theta_1 - \theta_3\theta_7 \\ \theta_7\theta_2 - \theta_1\theta_8 \\ \theta_2\theta_6 - \theta_5\theta_3 \\ \theta_3\theta_4 - \theta_6\theta_1 \\ \theta_1\theta_5 - \theta_4\theta_2 \end{pmatrix}.$$

   $\mathcal{N}[\,\cdot\,]$: normalization to unit norm
5. If $(\boldsymbol{\theta}^\dagger, \boldsymbol{\theta}) \approx 0$, return $\boldsymbol{\theta}$ and stop. Else, update $V_0[\boldsymbol{\theta}]$ to $\boldsymbol{P}_{\boldsymbol{\theta}}V_0[\boldsymbol{\theta}]\boldsymbol{P}_{\boldsymbol{\theta}}$ and go back to Step 3.

- $V_0[\boldsymbol{\theta}] = \boldsymbol{M}_8^-$ (truncated pseudoinverse of rank 8) = *KCR lower bound*.
- $V_0[\boldsymbol{\theta}]\boldsymbol{\theta} = \boldsymbol{0}$ is always ensured.

Internal access (Sugaya and Kanatani 2007)

SVD of $\boldsymbol{F}$:

$$\boldsymbol{F} = \boldsymbol{U}\begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}\boldsymbol{V}^\top, \qquad \sigma_1 = \cos\phi, \qquad \sigma_2 = \sin\phi.$$

We regard $\boldsymbol{U}$, $\boldsymbol{V}$, σ_1, and σ_2 as independent variables minimize the Sampson error J by Levenberg–Marquardt method.

---

1. Compute an $\boldsymbol{F}$ such that $\det\boldsymbol{F} = 0$, and express its SDV in the form

$$\boldsymbol{F} = \boldsymbol{U}\begin{pmatrix} \cos\phi & 0 & 0 \\ 0 & \sin\phi & 0 \\ 0 & 0 & 0 \end{pmatrix}\boldsymbol{V}^\top.$$

2. Compute the Sampson error J, and let $c = 0.0001$.
3. Compute the 9×3 matrices

$$\boldsymbol{F}_U = \begin{pmatrix} 0 & F_{31} & -F_{21} \\ 0 & F_{32} & -F_{22} \\ 0 & F_{33} & -F_{23} \\ -F_{31} & 0 & F_{11} \\ -F_{32} & 0 & F_{12} \\ -F_{33} & 0 & F_{13} \\ F_{21} & -F_{11} & 0 \\ F_{22} & -F_{12} & 0 \\ F_{23} & -F_{13} & 0 \end{pmatrix}, \qquad \boldsymbol{F}_V = \begin{pmatrix} 0 & F_{13} & -F_{12} \\ -F_{13} & 0 & F_{11} \\ F_{12} & -F_{11} & 0 \\ 0 & F_{23} & -F_{22} \\ -F_{23} & 0 & F_{21} \\ F_{22} & -F_{21} & 0 \\ 0 & F_{33} & -F_{32} \\ -F_{33} & 0 & F_{31} \\ F_{32} & -F_{31} & 0 \end{pmatrix}.$$

4. Compute the 9-D vector

$$\boldsymbol{\theta}_\phi = \begin{pmatrix} \sigma_1 U_{12}V_{12} - \sigma_2 U_{11}V_{11} \\ \sigma_1 U_{12}V_{22} - \sigma_2 U_{11}V_{21} \\ \sigma_1 U_{12}V_{32} - \sigma_2 U_{11}V_{31} \\ \sigma_1 U_{22}V_{12} - \sigma_2 U_{21}V_{11} \\ \sigma_1 U_{22}V_{22} - \sigma_2 U_{21}V_{21} \\ \sigma_1 U_{22}V_{32} - \sigma_2 U_{21}V_{31} \\ \sigma_1 U_{32}V_{12} - \sigma_2 U_{31}V_{11} \\ \sigma_1 U_{32}V_{22} - \sigma_2 U_{31}V_{21} \\ \sigma_1 U_{32}V_{32} - \sigma_2 U_{31}V_{31} \end{pmatrix}.$$

5. Compute the 9×9 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{L} = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})^2}V_0[\boldsymbol{\xi}_\alpha],$$

and let $\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}$.

6. Compute the first derivatives of J

$$\nabla_{\boldsymbol{\omega}}J = 2\boldsymbol{F}_U^\top\boldsymbol{X}\boldsymbol{\theta}, \qquad \nabla_{\boldsymbol{\omega}'}J = 2\boldsymbol{F}_V^\top\boldsymbol{X}\boldsymbol{\theta}, \qquad \frac{\partial J}{\partial\phi} = 2(\boldsymbol{\theta}_\phi, \boldsymbol{X}\boldsymbol{\theta}).$$

and the second derivatives

$$\nabla_{\boldsymbol{\omega}\boldsymbol{\omega}}J = 2\boldsymbol{F}_U^\top\boldsymbol{X}\boldsymbol{F}_U, \qquad \nabla_{\boldsymbol{\omega}'\boldsymbol{\omega}'}J = 2\boldsymbol{F}_V^\top\boldsymbol{X}\boldsymbol{F}_V, \qquad \nabla_{\boldsymbol{\omega}\boldsymbol{\omega}'}J = 2\boldsymbol{F}_U^\top\boldsymbol{X}\boldsymbol{F}_V,$$

$$\frac{\partial J^2}{\partial\phi^2} = 2(\boldsymbol{\theta}_\phi, \boldsymbol{X}\boldsymbol{\theta}_\phi), \qquad \frac{\partial\nabla_{\boldsymbol{\omega}}J}{\partial\phi} = 2\boldsymbol{F}_U^\top\boldsymbol{X}\boldsymbol{\theta}_\phi, \qquad \frac{\partial\nabla_{\boldsymbol{\omega}'}J}{\partial\phi} = 2\boldsymbol{F}_V^\top\boldsymbol{X}\boldsymbol{\theta}_\phi.$$

7. Compute the 9×9 Hessian

$$\boldsymbol{H} = \begin{pmatrix} \nabla_{\boldsymbol{\omega}\boldsymbol{\omega}}J & \nabla_{\boldsymbol{\omega}\boldsymbol{\omega}'}J & \partial\nabla_{\boldsymbol{\omega}}J/\partial\phi \\ (\nabla_{\boldsymbol{\omega}\boldsymbol{\omega}'}J)^\top & \nabla_{\boldsymbol{\omega}'\boldsymbol{\omega}'}J & \partial\nabla_{\boldsymbol{\omega}'}J/\partial\phi \\ (\partial\nabla_{\boldsymbol{\omega}}J/\partial\phi)^\top & (\partial\nabla_{\boldsymbol{\omega}'}J/\partial\phi)^\top & \partial J^2/\partial\phi^2 \end{pmatrix}$$

8. Solve the linear equation

$$(\boldsymbol{H} + cD[\boldsymbol{H}])\begin{pmatrix} \Delta\boldsymbol{\omega} \\ \Delta\boldsymbol{\omega}' \\ \Delta\phi \end{pmatrix} = -\begin{pmatrix} \nabla_{\boldsymbol{\omega}}J \\ \nabla_{\boldsymbol{\omega}'}J \\ \partial J/\partial\phi \end{pmatrix}.$$

$D[\,\cdot\,]$: diagonal matrix of diagonalelements.

9. Update $\boldsymbol{U}$, $\boldsymbol{V}$, and ϕ to

$$\boldsymbol{U}' = \boldsymbol{R}(\Delta\boldsymbol{\omega})\boldsymbol{U}, \qquad \boldsymbol{V}' = \boldsymbol{R}(\Delta\boldsymbol{\omega}')\boldsymbol{V}, \qquad \phi' = \phi + \Delta\phi.$$

   $\boldsymbol{R}(\boldsymbol{w})$: rotation around axis $\boldsymbol{w}$ by angle $\|\boldsymbol{w}\|$.

10. Update $\boldsymbol{F}$ to

$$\boldsymbol{F}' = \boldsymbol{U}' \begin{pmatrix} \cos\phi' & 0 & 0 \\ 0 & \sin\phi' & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{V}'^{\top}.$$

11. Compute the Sampson error J' of $\boldsymbol{F}'$. If $J' < J$ or $J' \approx J$ are not satisfied, let $c \leftarrow 10c$ and go back to Step 8.
12. If $\boldsymbol{F}' \approx \boldsymbol{F}$, return $\boldsymbol{F}'$ and stop. Else, let $\boldsymbol{F} \leftarrow \boldsymbol{F}'$, $\boldsymbol{U} \leftarrow \boldsymbol{U}'$, $\boldsymbol{V} \leftarrow \boldsymbol{V}'$, $\phi \leftarrow \phi'$, and $c \leftarrow c/10$, and go back to Step 3.

External access (Kanatani and Sugaya 2010)

1. Initialize $\boldsymbol{\theta}$.
2. Compute the 9×9 matrices $\boldsymbol{M}$ and $\boldsymbol{L}$.

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{L} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})^2} V_0[\boldsymbol{\xi}_\alpha]$$

3. Compute the 9-D vector $\boldsymbol{\theta}^\dagger$ and the 9×9 matrix $\boldsymbol{P}_{\boldsymbol{\theta}^\dagger}$

$$\boldsymbol{\theta}^\dagger = \begin{pmatrix} \theta_5\theta_9 - \theta_8\theta_6 \\ \theta_6\theta_7 - \theta_9\theta_4 \\ \theta_4\theta_8 - \theta_7\theta_5 \\ \theta_8\theta_3 - \theta_2\theta_9 \\ \theta_9\theta_1 - \theta_3\theta_7 \\ \theta_7\theta_2 - \theta_1\theta_8 \\ \theta_2\theta_6 - \theta_5\theta_3 \\ \theta_3\theta_4 - \theta_6\theta_1 \\ \theta_1\theta_5 - \theta_4\theta_2 \end{pmatrix}, \qquad \boldsymbol{P}_{\boldsymbol{\theta}^\dagger} = \boldsymbol{I} - \frac{\boldsymbol{\theta}^\dagger \boldsymbol{\theta}^{\dagger\top}}{\|\boldsymbol{\theta}^\dagger\|^2}$$

4. Compute the 9×9 matices $\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}$ and $\boldsymbol{Y} = \boldsymbol{P}_{\boldsymbol{\theta}^\dagger}\boldsymbol{X}\boldsymbol{P}_{\boldsymbol{\theta}^\dagger}$. Compute the unit eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of $\boldsymbol{Y}$ for the smallest two eigenvalues, and let $\hat{\boldsymbol{\theta}} = (\boldsymbol{\theta}, \boldsymbol{v}_1)\boldsymbol{v}_1 + (\boldsymbol{\theta}, \boldsymbol{v}_2)\boldsymbol{v}_2$.
5. Compute $\boldsymbol{\theta}' = \mathcal{N}[\boldsymbol{P}_{\boldsymbol{\theta}^\dagger}\hat{\boldsymbol{\theta}}]$.
6. If $\boldsymbol{\theta}' \approx \boldsymbol{\theta}$ up to sign, return $\boldsymbol{\theta}'$ as $\boldsymbol{\theta}$ and stop. Else, let $\boldsymbol{\theta} \leftarrow \mathcal{N}[\boldsymbol{\theta} + \boldsymbol{\theta}']$ and go back to Step 2.

Geometric distance minimization (Kanatani and Sugaya 2010)

---

1. Let $J_0 = \infty$, $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, $\hat{x}'_\alpha = x'_\alpha$, $\hat{y}'_\alpha = y'_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0$.
2. Compute the normalized covariance matrix $V_0[\hat{\boldsymbol{\xi}}_\alpha]$ using $\hat{x}_\alpha$, $\hat{y}_\alpha$, $\hat{x}'_\alpha$, and $\hat{y}'_\alpha$, and let

$$\boldsymbol{\xi}^*_\alpha = \begin{pmatrix} \hat{x}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{x}'_\alpha \\ \hat{x}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{x}_\alpha + \tilde{x}_\alpha) \\ \hat{y}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{x}'_\alpha \\ \hat{y}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{y}_\alpha + \tilde{y}_\alpha) \\ f_0(\hat{x}'_\alpha + \tilde{x}'_\alpha) \\ f_0(\hat{y}'_\alpha + \tilde{y}'_\alpha) \\ f_0^2 \end{pmatrix}.$$

3. Compute the $\boldsymbol{\theta}$ that minimizes the *modified Sampson error*

$$J^* = \frac{1}{N} \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}^*_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})}$$

4. Update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\tilde{x}'_\alpha$, and $\tilde{y}'_\alpha$ to

$$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{(\boldsymbol{\xi}^*_\alpha, \boldsymbol{\theta})}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{pmatrix} \begin{pmatrix} \hat{x}'_\alpha \\ \hat{y}'_\alpha \\ f_0 \end{pmatrix}, \qquad \begin{pmatrix} \tilde{x}'_\alpha \\ \tilde{y}'_\alpha \end{pmatrix} \leftarrow \frac{(\boldsymbol{\xi}^*_\alpha, \boldsymbol{\theta})}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_4 & \theta_7 \\ \theta_2 & \theta_5 & \theta_8 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix},$$

$$\hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \qquad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha, \qquad \hat{x}'_\alpha \leftarrow x'_\alpha - \tilde{x}'_\alpha, \qquad \hat{y}'_\alpha \leftarrow y'_\alpha - \tilde{y}'_\alpha$$

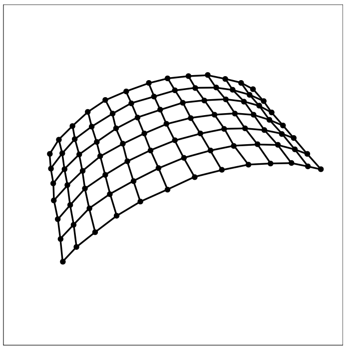
5. Compute

$$J^* = \frac{1}{N} \sum_{\alpha=1}^{N} (\tilde{x}_\alpha^2 + \tilde{y}_\alpha^2 + \tilde{x}'^2_\alpha + \tilde{y}'^2_\alpha).$$

   If $J^* \approx J_0$, return $\boldsymbol{\theta}$ and stop. Else, let $J_0 \leftarrow J^*$ and go back to Step 2.

- The Sampson error minimization solution and the geometric distance minimization solution usually coincide up to several significant digits.
- Minimizing the Sampson error is *practically the same* as minimizing the geometric distance.

Examples



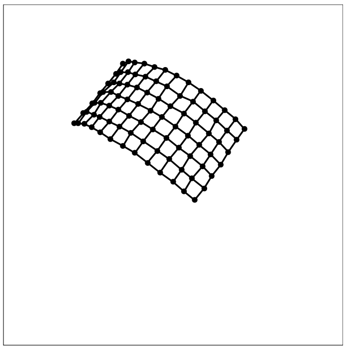


Image size: 600×600, noise level $\sigma = 1.0$, computation error: $E = \sqrt{\sum_{i,j=1}^{3}(F_{ij} - \bar{F}_{ij})^2}$

method	E
LS + SVD	0.370992
FNS + SVD	0.142874
optimal correction	0.026385
internal	0.062475
external	0.026202
geometric distance minimization	0.026149

$$\bar{\boldsymbol{F}} = \begin{pmatrix} 0.07380 & -0.34355 & -0.28357 \\ 0.21858 & 0.41655 & 0.33508 \\ 0.66823 & -0.08789 & -0.09100 \end{pmatrix}$$

LS+SVD: $\begin{pmatrix} 0.21115 & -0.52234 & -0.38029 \\ 0.32188 & 0.32504 & 0.18557 \\ 0.53935 & 0.05232 & -0.02506 \end{pmatrix}$

internal: $\begin{pmatrix} 0.09265 & -0.36657 & -0.30765 \\ 0.24157 & 0.40747 & 0.33578 \\ 0.65177 & -0.05101 & -0.07704 \end{pmatrix}$

FNS+SVD: $\begin{pmatrix} 0.09599 & -0.41151 & -0.34263 \\ 0.25978 & 0.36820 & 0.28133 \\ 0.64538 & -0.02586 & -0.06821 \end{pmatrix}$

external: $\begin{pmatrix} 0.06067 & -0.33702 & -0.27208 \\ 0.21213 & 0.42767 & 0.33980 \\ 0.66834 & -0.10005 & -0.09306 \end{pmatrix}$

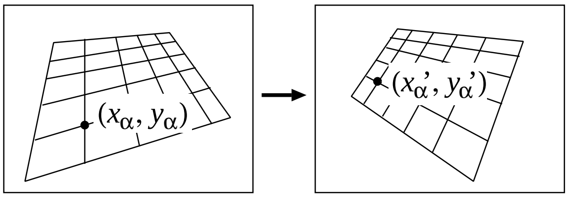
FNS + opt. correc.: $\begin{pmatrix} 0.07506 & -0.34616 & -0.27188 \\ 0.21826 & 0.43547 & 0.33471 \\ 0.65834 & -0.09763 & -0.09158 \end{pmatrix}$

geom. dist.: $\begin{pmatrix} 0.06068 & -0.33706 & -0.27210 \\ 0.21215 & 0.42764 & 0.33979 \\ 0.66833 & -0.10002 & -0.09306 \end{pmatrix}$

- LS + SVD (= *Hartley's 8-point method*) has poor accuracy.
- Optimal correction, internal access, and external access all have almost optimal ($\approx$ KCR lower bound).
- Geometric distance minimization by iterations results in little improvement.

Homography Computation

Homography



Two images of a planar surface are related by a *homography*:

$$x' = f_0 \frac{H_{11}x + H_{12}y + H_{13}f_0}{h_{31}x + H_{32}y + H_{33}f_0}, \qquad y' = f_0 \frac{H_{21}x + H_{22}y + H_{23}f_0}{h_{31}x + H_{32}y + H_{33}f_0}.$$

- f_0: scale factor ($\approx$ the size of the image)

This can be written as

$$\begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix} \simeq \underbrace{\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}}_{\equiv \boldsymbol{H}} \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}.$$

- $\simeq$: equality up to a nonzero constant
- $\boldsymbol{H}$: *homography matrix*
- To remove scale indeterminacy, $\boldsymbol{H}$ is normalized to unit norm: $\|\boldsymbol{H}\|$ $(\equiv \sqrt{\sum_{i,j=1,3} H_{ij}^2}) = 1$

From the computed $\boldsymbol{H}$, we can reconstruct the position and orientation of the plane and compute the relative camera positions.

Vector representation

$$\begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix} \simeq \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix} \quad \leftrightarrow \quad \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix} \times \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The three components of this vector equation are $(\boldsymbol{\xi}^{(1)}, \boldsymbol{\theta}) = 0$, $(\boldsymbol{\xi}^{(2)}, \boldsymbol{\theta}) = 0$, and $(\boldsymbol{\xi}^{(3)}, \boldsymbol{\theta}) = 0$, where

$$\boldsymbol{\theta} = \begin{pmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ H_{33} \end{pmatrix}, \quad \boldsymbol{\xi}^{(1)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -f_0 x \\ -f_0 y \\ -f_0^2 \\ xy' \\ yy' \\ f_0 y' \end{pmatrix}, \quad \boldsymbol{\xi}^{(2)} = \begin{pmatrix} f_0 x \\ f_0 y \\ f_0^2 \\ 0 \\ 0 \\ 0 \\ -xx' \\ -yx' \\ -f_0 x' \end{pmatrix}, \quad \boldsymbol{\xi}^{(3)} = \begin{pmatrix} -xy' \\ -yy' \\ -f_0 y' \\ xx' \\ yx' \\ f_0 x' \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

- $\|\boldsymbol{H}\| = 1 \quad \rightarrow \quad \|\boldsymbol{\theta}\| = 1.$

Task: From noisy observations $\boldsymbol{\xi}_\alpha^{(k)}$, estimate a unit vector $\boldsymbol{\theta}$ such that

$$(\boldsymbol{\xi}_\alpha^{(k)}, \boldsymbol{\theta}) \approx 0, \qquad k = 1, 2, 3, \qquad \alpha = 1, ..., N.$$

- The three equations are *not linearly independent*.
  - If two of them are satisfied, the remaining one is automatically satisfied.

Covariance matrices

The noise terms Δx_α, Δy_α, $\Delta x'_\alpha$, and Δy_α are regarded as independent Gaussian random variables of mean 0 and variance σ^2:

$$E[\Delta x_\alpha] = E[\Delta y_\alpha] = E[\Delta x'_\alpha] = E[\Delta y'_\alpha] = 0, \quad E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = E[\Delta {x'_\alpha}^2] = E[\Delta {y'_\alpha}^2] = \sigma^2,$$

$$E[\Delta x_\alpha \Delta y_\alpha] = E[\Delta x'_\alpha \Delta y'_\alpha] = E[\Delta x_\alpha \Delta y'_\alpha] = E[\Delta x'_\alpha \Delta y_\alpha] = 0.$$

The covariance matrices of $\boldsymbol{\xi}_\alpha^{(k)}$ is defined by

$$V^{(kl)}[\boldsymbol{\xi}_\alpha] = E[\Delta_1 \boldsymbol{\xi}_\alpha^{(k)} \Delta_1 \boldsymbol{\xi}_\alpha^{(l)\top}] \ \ (= \sigma^2 V_0^{(kl)}[\boldsymbol{\xi}_\alpha]).$$

Then,

$$V_0^{(kl)}[\boldsymbol{\xi}_\alpha] = \boldsymbol{T}_\alpha^{(k)} \boldsymbol{T}_\alpha^{(l)\top}, \qquad \boldsymbol{T}_\alpha^{(k)} = \left(\frac{\partial \boldsymbol{\xi}^{(k)}}{\partial x} \quad \frac{\partial \boldsymbol{\xi}^{(k)}}{\partial y} \quad \frac{\partial \boldsymbol{\xi}^{(k)}}{\partial x'} \quad \frac{\partial \boldsymbol{\xi}^{(k)}}{\partial y'} \right)\Bigg|_\alpha .$$

- $\boldsymbol{T}_\alpha^{(k)}$: 9×4 Jacobi matrix
- $(\cdot)|_\alpha$: value for $x = x_\alpha$, $y = y_\alpha$, $x' = x'_\alpha$, and $y' = y'_\alpha$.
- $V_0^{(kl)}[\boldsymbol{\xi}_\alpha]$: the *normalized covariance matrices*

Iterative reweight

---

1. Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha^{(kl)} = \delta_{kl}$, $\alpha = 1, ..., N$, $k, l = 1, 2, 3$.
2. Compute the 9×9 matrices

$$\boldsymbol{M} = \frac{1}{N} \sum_{\alpha=1}^{N} \sum_{k,l=1}^{3} W_\alpha^{(kl)} \boldsymbol{\xi}_\alpha^{(k)} \boldsymbol{\xi}_\alpha^{(l)\top}.$$

3. Solve the eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{\theta},$$

   and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update

$$W_\alpha^{(kl)} \leftarrow \Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\theta}_\alpha]\boldsymbol{\theta}) \Big)_2^-, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

- δ_{kl}: Kronecker delta (1 for $k = l$ and 0 otherwise)
- $\Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\theta}_\alpha]\boldsymbol{\theta}) \Big)$: the matrix whose (k, l) element is $(\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\theta}_\alpha]\boldsymbol{\theta})$.
- $\Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\theta}_\alpha]\boldsymbol{\theta}) \Big)_2^-$: its pseudoinverse of truncated rank 2.
- The initial solution corresponds to least squares.

Renormalization (Kanatani et al. 2000)

---

1. Let $\boldsymbol{\theta}_0 = \boldsymbol{0}$ and $W_\alpha^{(kl)} = \delta_{kl}$, $\alpha = 1, ..., N$, $k, l = 1, 2, 3$.
2. Compute the 9×9 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} \boldsymbol{\xi}_\alpha^{(k)} \boldsymbol{\xi}_\alpha^{(l)\top}, \qquad \boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} V_0^{(kl)}[\boldsymbol{\xi}_\alpha].$$

3. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

   and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the generalized eigenvalue λ of the smallest absolute value.
4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update

$$W_\alpha^{(kl)} \leftarrow \Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})\Big)_2^-, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

- The initial solution corresponds to the Taubin method.

Hyper-renormalization (Kanatani et al. 2014)

---

1. Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha^{(kl)} = \delta_{kl}$, $\alpha = 1, ..., N$, $k, l = 1, 2, 3$.
2. Compute the 9×9

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} \boldsymbol{\xi}_\alpha^{(k)} \boldsymbol{\xi}_\alpha^{(l)\top},$$

$$\boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} V_0^{(kl)}[\boldsymbol{\xi}_\alpha]$$

$$-\frac{1}{N^2}\sum_{\alpha=1}^{N}\sum_{k,l,m,n=1}^{3} W_\alpha^{(kl)} W_\alpha^{(mn)} \Big((\boldsymbol{\xi}_\alpha^{(k)}, \boldsymbol{M}_8^{-} \boldsymbol{\xi}_\alpha^{(m)}) V_0^{(ln)}[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0^{(km)}[\boldsymbol{\xi}_\alpha] \boldsymbol{M}_8^{-} \boldsymbol{\xi}_\alpha^{(l)} \boldsymbol{\xi}_\alpha^{(n)\top}] \Big).$$

3. Solve the generalized eigenvalue problem

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda \boldsymbol{N}\boldsymbol{\theta},$$

and compute the unit generalized eigenvector $\boldsymbol{\theta}$ for the generalized eigenvalue λ of the smallest absolute value.

4. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update

$$W_\alpha^{(kl)} \leftarrow \Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta}) \Big)_2^{-}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

and go back to Step 2.

- The initial solution corresponds to HyperLS

FNS (Kanatani and Niitsuma 2011)

---

1. Let $\boldsymbol{\theta} = \boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha^{(kl)} = \delta_{kl}$, $\alpha = 1, ..., N$, $k, l = 1, 2, 3$.
2. Compute the 9×9 matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} \boldsymbol{\xi}_\alpha^{(k)} \boldsymbol{\xi}_\alpha^{(l)\top}, \qquad \boldsymbol{L} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} v_\alpha^{(k)} v_\alpha^{(l)} V_0^{(kl)}[\boldsymbol{\xi}_\alpha],$$

   where

$$v_\alpha^{(k)} = \sum_{l=1}^{3} W_\alpha^{(kl)} (\boldsymbol{\xi}_\alpha^{(l)}, \boldsymbol{\theta}).$$

3. Compute the 9×9 matrix

$$\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}.$$

4. Solve the eigenvalue problem

$$\boldsymbol{X}\boldsymbol{\theta} = \lambda \boldsymbol{\theta},$$

   and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
5. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update

$$W_\alpha^{(kl)} \leftarrow \Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta}) \Big)_2^-, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta},$$

   and go back to Step 2.

- This minimizes the *Sampson error*:

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)} (\boldsymbol{\xi}_\alpha^{(k)}, \boldsymbol{\theta})(\boldsymbol{\xi}_\alpha^{(l)}, \boldsymbol{\theta}), \qquad W_\alpha^{(kl)} = \Big((\boldsymbol{\theta}, V_0^{(kl)}[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta}) \Big)_2^-,$$

- The initial solution corresponds to least squares.
- This reduces to the FNS of Chojnacki et al. (2000) for a single constraint.

Geometric distance minimization

We strictly minimize the *geometric distance*

$$S = \frac{1}{N}\sum_{\alpha=1}^{N}\Big((x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2 + (x'_\alpha - \bar{x}'_\alpha)^2 + (y'_\alpha - \bar{y}'_\alpha)^2\Big).$$

- We first minimize the Sampson error J by FNS and modify the data $\boldsymbol{\xi}_\alpha^{(k)}$ to $\boldsymbol{\xi}_\alpha^{(k)*}$ using the computed solution $\boldsymbol{\theta}$.
- Regarding them as data, we define the *modified Sampson error* J^* and minimize it by FNS.
- If this is repeated, the modified Sampson error J^* eventually coincides with the geometric distance S.
  - We we obtain the solution that minimize S.
- The iterations do not alter the value of $\boldsymbol{\theta}$ over several significant digits.
  - Sampson error minimization is *practically the same* as geometric distance minimization.

Hyperaccurate correction

- The geometric distance minimization solution is theoretically biased.
- We can theoretically improve the accuracy by evaluating and subtracting the bias.
  $\rightarrow$ *hyperaccurate correction*
- However, the accuracy gain is very small.
  - The bias of the solution is very small.
- The data $\boldsymbol{\xi}_\alpha^{(k)}$ consist of *bilinear* expressions in x_α, y_α, x'_α, and y'_α.
  - Unlike ellipse fitting, no quadratic terms such as x_α^2 are involved,
- Noise in different images are assumed to be independent.
  - The bais of fundamental matrix computation is also small.

Examples

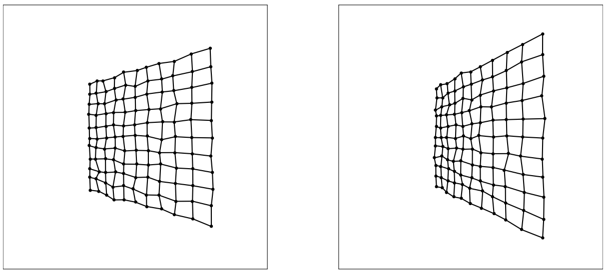


Image size: 500×500, noise level $\sigma = 1.0$, computation error: $E = \sqrt{\sum_{i,j=1}^{3}(H_{ij} - \bar{H}_{ij})^2}$

method	E
LS	1.15042×10^{-2}
iterative reweight	1.07295×10^{-2}
Taubin	0.73568×10^{-2}
renormalization	0.71149×10^{-2}
HyperLS	0.73513×10^{-2}
hyper-renormalization	0.71154×10^{-2}
FNS	0.70337×10^{-2}
geometric distance minimization	0.70304×10^{-2}
hyperaccurate correction	0.70296×10^{-2}

$$\bar{\boldsymbol{H}} = \begin{pmatrix} 0.57773 & 0.00000 & 0.00000 \\ 0.00000 & 0.47171 & 0.00000 \\ 0.00000 & -0.31587 & 0.57773 \end{pmatrix}$$

$$\text{LS:} \begin{pmatrix} 0.21115 & -0.52234 & -0.38029 \\ 0.32188 & 0.32504 & 0.18557 \\ 0.53935 & 0.05232 & -0.02506 \end{pmatrix} \quad \text{hyper-renorm.:} \begin{pmatrix} 0.57690 & -0.00023 & -0.00018 \\ 0.00155 & 0.47284 & 0.00001 \\ -0.00679 & -0.33143 & 0.57768 \end{pmatrix}$$

$$\text{FNS:} \begin{pmatrix} 0.57694 & -0.00020 & -0.00018 \\ 0.00158 & 0.47282 & 0.00001 \\ -0.00671 & -0.33138 & 0.57769 \end{pmatrix} \quad \text{geom dist.:} \begin{pmatrix} 0.57695 & -0.00020 & -0.00018 \\ 0.00158 & 0.47282 & 0.00001 \\ -0.00571 & -0.33135 & 0.57769 \end{pmatrix}$$

- LS and iterative reweight have poor accuracy.
- Taubin and HyperLS improve the accuracy.
- Renormalization and hyper-renormalization further improve the accuracy.
- FNS $\approx$ geometric distance minimization $\approx$ hyperaccurate correction
- FNS is the most suitable in practice.

Acknowledgments

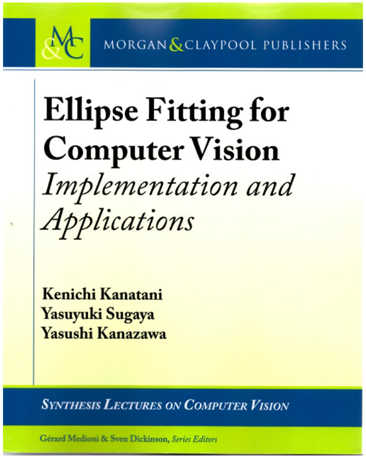
This work has been done in collaboration with:

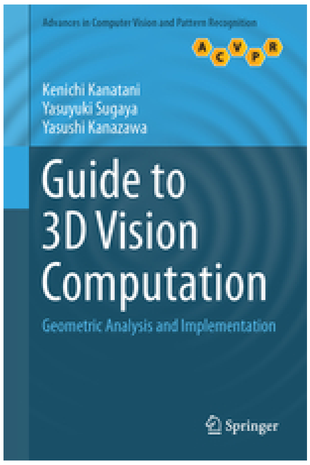
- Prasanna Rangarajan (Southern Methodist University, U.S.A.)
- Ali Al-Sharadqah (California State University, Northridge, U.S.A).
- Nikolai Chernov (University of Alabama at Birmingham, U.S.A.); passed away August 7, 2014

Special thanks are due to:

- Takayuki Okatani (Tohoku University, Japan)
- Michael Felsberg (Linköping University, Sweden)
- Rudolf Mester (University of Frankfurt, Germany)
- Wolfgang Förstner (University of Bonn, Germany)
- Peter Meer (Rutgers University, U.S.A.)
- Michael Brooks (University of Adelaide, Australia)
- Wojciech Chojnacki (University of Adelaide, Australia)
- Alexander Kukush (University of Kiev, Ukraine)

For further details, see





K. Kanatani, Y. Sugaya, and Y. Kanazawa,
Ellipse Fitting for Computer Vision: Implementation and Applications,
Morgan & Claypool Publishers, San Rafael, CA, U.S., April, 2016.
ISBN 9781627054584 (print), ISBN 9781627054980 (E-book)

K. Kanatani, Y. Sugaya, and Y. Kanazawa,
Guide to 3D Vision Computation: Geometric Analysis and Implementation.
Springer International, Cham, Switzerland, December, 2016.
ISBN 978-3-319-48492-1 (print), ISBN 978-3-319-48943-8 (E-book)