

3D Computer Vision

Radim Šára Martin Matoušek

Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

<https://cw.fel.cvut.cz/wiki/courses/tdv/start>

<http://cmp.felk.cvut.cz>

<mailto:sara@cmp.felk.cvut.cz>

phone ext. 7203

rev. November 9, 2021



Open Informatics Master's Course

►5-Point Algorithm for Relative Camera Orientation

Problem: Given $\{m_i, m'_i\}_{i=1}^5$ corresponding image points and calibration matrix \mathbf{K} , recover the camera motion \mathbf{R} , \mathbf{t} .

Obs:

1. \mathbf{E} – homogeneous 3×3 matrix; 9 numbers up to scale
2. \mathbf{R} – 3 DOF, \mathbf{t} – 2 DOF only, in total 5 DOF \rightarrow we need $9 - 1 - 5 = 3$ constraints on \mathbf{E}
3. idea: \mathbf{E} essential iff it has two equal singular values and the third is zero \rightarrow 81

This gives an equation system:

$$\underline{\mathbf{v}}_i^\top \mathbf{E} \underline{\mathbf{v}}'_i = 0 \quad 5 \text{ linear constraints } (\underline{\mathbf{v}} \simeq \mathbf{K}^{-1} \underline{\mathbf{m}})$$

$$\det \mathbf{E} = 0 \quad 1 \text{ cubic constraint}$$

$$\mathbf{E} \mathbf{E}^\top \mathbf{E} - \frac{1}{2} \text{tr}(\mathbf{E} \mathbf{E}^\top) \mathbf{E} = 0 \quad 9 \text{ cubic constraints, 2 independent}$$

⊛ P1; 1pt: verify this equation from $\mathbf{E} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$, $\mathbf{D} = \lambda \text{diag}(1, 1, 0)$

1. estimate \mathbf{E} by SVD from $\underline{\mathbf{v}}_i^\top \mathbf{E} \underline{\mathbf{v}}'_i = 0$ by the null-space method 4D null space
2. this gives $\mathbf{E} \simeq x \mathbf{E}_1 + y \mathbf{E}_2 + z \mathbf{E}_3 + \mathbf{E}_4$
3. at most 10 (complex) solutions for x, y, z from the cubic constraints

- when all 3D points lie on a plane: at most 2 real solutions (twisted-pair) can be disambiguated in 3 views
or by chirality constraint (\rightarrow 83) unless all 3D points are closer to one camera
- 6-point problem for unknown f [Kukelova et al. BMVC 2008]
- resources at <http://aag.ciirc.cvut.cz/minimal/>

► The Triangulation Problem

Problem: Given cameras $\mathbf{P}_1, \mathbf{P}_2$ and a correspondence $x \leftrightarrow y$ compute a 3D point \mathbf{X} projecting to x and y

$$\lambda_1 \underline{\mathbf{x}} = \mathbf{P}_1 \underline{\mathbf{X}}, \quad \lambda_2 \underline{\mathbf{y}} = \mathbf{P}_2 \underline{\mathbf{X}}, \quad \underline{\mathbf{x}} = \begin{bmatrix} u^1 \\ v^1 \\ 1 \end{bmatrix}, \quad \underline{\mathbf{y}} = \begin{bmatrix} u^2 \\ v^2 \\ 1 \end{bmatrix}, \quad \mathbf{P}_i = \begin{bmatrix} (\mathbf{p}_1^i)^\top \\ (\mathbf{p}_2^i)^\top \\ (\mathbf{p}_3^i)^\top \end{bmatrix}$$

Linear triangulation method after eliminating λ_1, λ_2

$$\begin{aligned} u^1 (\mathbf{p}_3^1)^\top \underline{\mathbf{X}} &= (\mathbf{p}_1^1)^\top \underline{\mathbf{X}}, & u^2 (\mathbf{p}_3^2)^\top \underline{\mathbf{X}} &= (\mathbf{p}_1^2)^\top \underline{\mathbf{X}}, \\ v^1 (\mathbf{p}_3^1)^\top \underline{\mathbf{X}} &= (\mathbf{p}_2^1)^\top \underline{\mathbf{X}}, & v^2 (\mathbf{p}_3^2)^\top \underline{\mathbf{X}} &= (\mathbf{p}_2^2)^\top \underline{\mathbf{X}} \end{aligned}$$

Gives

$$\mathbf{D} \underline{\mathbf{X}} = \mathbf{0}, \quad \mathbf{D} = \begin{bmatrix} u^1 (\mathbf{p}_3^1)^\top - (\mathbf{p}_1^1)^\top \\ v^1 (\mathbf{p}_3^1)^\top - (\mathbf{p}_2^1)^\top \\ u^2 (\mathbf{p}_3^2)^\top - (\mathbf{p}_1^2)^\top \\ v^2 (\mathbf{p}_3^2)^\top - (\mathbf{p}_2^2)^\top \end{bmatrix}, \quad \mathbf{D} \in \mathbb{R}^{4,4}, \quad \underline{\mathbf{X}} \in \mathbb{R}^4 \quad (14)$$

- typically, \mathbf{D} has full rank (!)
- what else: back-projected rays will generally not intersect due to image error, see next
- what else: using Jack-knife ($\rightarrow 63$) not recommended sensitive to small error
- idea: we will step back and use SVD ($\rightarrow 90$)
- but the result will not be invariant to projective frame
replacing $\mathbf{P}_1 \mapsto \mathbf{P}_1 \mathbf{H}, \mathbf{P}_2 \mapsto \mathbf{P}_2 \mathbf{H}$ does not always result in $\underline{\mathbf{X}} \mapsto \mathbf{H}^{-1} \underline{\mathbf{X}}$
- note the homogeneous form in (14) can represent points $\underline{\mathbf{X}}$ at infinity

► The Least-Squares Triangulation by SVD

- if \mathbf{D} is full-rank we may minimize the algebraic least-squares error

$$\epsilon^2(\underline{\mathbf{X}}) = \|\mathbf{D}\underline{\mathbf{X}}\|^2 \quad \text{s.t.} \quad \|\underline{\mathbf{X}}\| = 1, \quad \underline{\mathbf{X}} \in \mathbb{R}^4$$

- let \mathbf{d}_i be the i -th row of \mathbf{D} taken as a column vector, then

$$\|\mathbf{D}\underline{\mathbf{X}}\|^2 = \sum_{i=1}^4 (\mathbf{d}_i^\top \underline{\mathbf{X}})^2 = \sum_{i=1}^4 \underline{\mathbf{X}}^\top \mathbf{d}_i \mathbf{d}_i^\top \underline{\mathbf{X}} = \underline{\mathbf{X}}^\top \mathbf{Q} \underline{\mathbf{X}}, \quad \text{where } \mathbf{Q} = \sum_{i=1}^4 \mathbf{d}_i \mathbf{d}_i^\top = \mathbf{D}^\top \mathbf{D} \in \mathbb{R}^{4,4}$$

- we write the SVD of \mathbf{Q} as $\mathbf{Q} = \sum_{j=1}^4 \sigma_j^2 \mathbf{u}_j \mathbf{u}_j^\top$, in which [Golub & van Loan 2013, Sec. 2.5]

$$\sigma_1^2 \geq \dots \geq \sigma_4^2 \geq 0 \quad \text{and} \quad \mathbf{u}_l^\top \mathbf{u}_m = \begin{cases} 0 & \text{if } l \neq m \\ 1 & \text{otherwise} \end{cases}$$

- then $\underline{\mathbf{X}} = \arg \min_{\mathbf{q}, \|\mathbf{q}\|=1} \mathbf{q}^\top \mathbf{Q} \mathbf{q} = \mathbf{u}_4$ the last column of the \mathbf{U} matrix from $\text{SVD}(\mathbf{D}^\top \mathbf{D})$

Proof (by contradiction).

Let $\bar{\mathbf{q}} = \sum_{i=1}^4 a_i \mathbf{u}_i$ s.t. $\sum_{i=1}^4 a_i^2 = 1$, then $\|\bar{\mathbf{q}}\| = 1$, as desired, and

$$\bar{\mathbf{q}}^\top \mathbf{Q} \bar{\mathbf{q}} = \sum_{j=1}^4 \sigma_j^2 \bar{\mathbf{q}}^\top \mathbf{u}_j \mathbf{u}_j^\top \bar{\mathbf{q}} = \sum_{j=1}^4 \sigma_j^2 (\mathbf{u}_j^\top \bar{\mathbf{q}})^2 = \dots = \sum_{j=1}^4 a_j^2 \sigma_j^2 \geq \sum_{j=1}^4 a_j^2 \sigma_4^2 = \sigma_4^2$$

since $\sigma_j \geq \sigma_4$

□

$i=1,2,3,4$

1. compute \mathbf{Q}
2. $\mathbf{U} \mathbf{D} \mathbf{V}^\top = \mathbf{Q}$
3. μ_4

- if $\sigma_4 \ll \sigma_3$, there is a unique solution $\underline{\mathbf{X}} = \mathbf{u}_4$ with residual error $(\mathbf{D} \underline{\mathbf{X}})^2 = \sigma_4^2$
the quality (conditioning) of the solution may be expressed as $q = \sigma_3/\sigma_4$ (greater is better)

Matlab code for the least-squares solver:

```
[U,0,V] = svd(D);  
X = V(:,end);  
q = sqrt(0(end-1,end-1)/0(end,end));
```

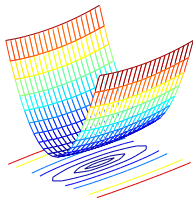
⊗ P1; 1pt: Why did we decompose \mathbf{D} here, and not $\mathbf{Q} = \mathbf{D}^\top \mathbf{D}$?

Numerical Conditioning

- The equation $\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}$ in (14) may be ill-conditioned for numerical computation, which results in a poor estimate for $\underline{\mathbf{X}}$.

Why: on a row of \mathbf{D} there are big entries together with small entries, e.g. of orders projection centers in mm, image points in px

$$\begin{bmatrix} 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \\ 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \end{bmatrix}$$



Quick fix:

- re-scale the problem by a regular diagonal conditioning matrix $\mathbf{S} \in \mathbb{R}^{4,4}$

$$\mathbf{0} = \mathbf{D}\underline{\mathbf{X}} = \mathbf{D}\mathbf{S}\mathbf{S}^{-1}\underline{\mathbf{m}}\underline{\mathbf{X}} = \hat{\mathbf{D}}\underline{\mathbf{m}}\underline{\mathbf{X}}$$

choose \mathbf{S} to make the entries in $\hat{\mathbf{D}}$ all smaller than unity in absolute value:

$$\mathbf{S} = \text{diag}(10^{-3}, 10^{-3}, 10^{-3}, 10^{-6}) \quad \mathbf{S} = \text{diag}(1./\max(\text{abs}(\mathbf{D}), 1))$$

- solve for $\underline{\mathbf{m}}\underline{\mathbf{X}}$ as before
- get the final solution as $\underline{\mathbf{m}}\underline{\mathbf{X}} = \mathbf{S}\underline{\mathbf{m}}\underline{\mathbf{X}}$

$$\hat{\mathbf{D}} = \mathbf{D}\mathbf{S} \rightarrow \mathbf{D} = \hat{\mathbf{D}}\mathbf{S}^{-1}$$

- when SVD is used in camera resection, conditioning is essential for success

→62

Algebraic Error vs Reprojection Error

- algebraic error (c – camera index, (u^c, v^c) – image coordinates)

from SVD →91

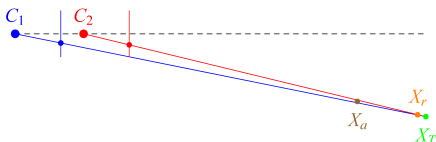
$$e^2(\underline{\mathbf{X}}) = \sigma_4^2 = \sum_{c=1}^2 \left[\left(u^c (\mathbf{p}_3^c)^T \underline{\mathbf{X}} - (\mathbf{p}_1^c)^T \underline{\mathbf{X}} \right)^2 + \left(v^c (\mathbf{p}_3^c)^T \underline{\mathbf{X}} - (\mathbf{p}_2^c)^T \underline{\mathbf{X}} \right)^2 \right]$$

- reprojection error

$$e^2(\underline{\mathbf{X}}) = \sum_{c=1}^2 \left[\left(u^c - \frac{(\mathbf{p}_1^c)^T \underline{\mathbf{X}}}{(\mathbf{p}_3^c)^T \underline{\mathbf{X}}} \right)^2 + \left(v^c - \frac{(\mathbf{p}_2^c)^T \underline{\mathbf{X}}}{(\mathbf{p}_3^c)^T \underline{\mathbf{X}}} \right)^2 \right]$$

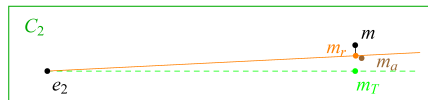
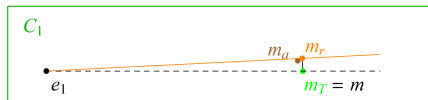
- algebraic error zero \Leftrightarrow reprojection error zero $\sigma_4 = 0 \Rightarrow$ non-trivial null space
- epipolar constraint satisfied \Rightarrow equivalent results
- in general: minimizing algebraic error is cheap but it gives inferior results
- minimizing reprojection error is expensive but it gives good results
- the midpoint of the common perpendicular to both optical rays gives about 50% greater error in 3D
- the golden standard method – deferred to →106

Ex:



- forward camera motion
- error $f/50$ in image 2, orthogonal to epipolar plane

X_T – noiseless ground truth position
 X_r – reprojection error minimizer
 X_a – algebraic error minimizer
 m – measurement (m_T with noise in v^2)



► We Have Added to The ZOO (cont'd from →69)

problem	given	unknown	slide
camera resection	6 world–img correspondences $\{(X_i, m_i)\}_{i=1}^6$	P	62
exterior orientation	K , 3 world–img correspondences $\{(X_i, m_i)\}_{i=1}^3$	R, t	66
relative pointcloud orientation	3 world–world correspondences $\{(X_i, Y_i)\}_{i=1}^3$	R, t	70
fundamental matrix	7 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^7$	F	84
relative camera orientation	K , 5 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^5$	R, t	88
triangulation	P ₁ , P ₂ , 1 img–img correspondence (m_i, m'_i)	X	89

A bigger ZOO at <http://cmp.felk.cvut.cz/minimal/>

calibrated problems

- have fewer degenerate configurations
- can do with fewer points (good for geometry proposal generators →119)
- algebraic error optimization (SVD) makes sense in camera resection and triangulation only
- but it is not the best method; we will now focus on 'optimizing optimally'




Optimization for 3D Vision

- 5.1 The Concept of Error for Epipolar Geometry
- 5.2 The Golden Standard for Triangulation ←
- 5.3 Levenberg-Marquardt's Iterative Optimization
- 5.4 Optimizing Fundamental Matrix ←
- 5.5 The Correspondence Problem
- 5.6 Optimization by Random Sampling

covered by

- [1] [H&Z] Secs: 11.4, 11.6, 4.7
- [2] Fischler, M.A. and Bolles, R.C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6):381–395, 1981

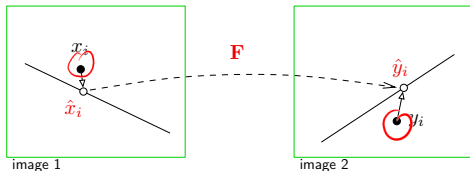
additional references

-  P. D. Sampson. Fitting conic sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm. *Computer Vision, Graphics, and Image Processing*, 18:97–108, 1982.
-  O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Proc DAGM, LNCS 2781:236–243*. Springer-Verlag, 2003.
-  O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In *Proc ICPR, vol 1:112–115, 2004*.

► The Concept of Error for Epipolar Geometry

Background problems: (1) Given at least 8 matched points $x_i \leftrightarrow y_j$ in a general position, estimate the most 'likely' fundamental matrix \mathbf{F} ; (2) given \mathbf{F} triangulate 3D point from $x_i \leftrightarrow y_j$.

$$\mathbf{x}_i = (u_i^1, v_i^1), \quad \mathbf{y}_i = (u_i^2, v_i^2), \quad i = 1, 2, \dots, k, \quad k \geq 8$$



- detected points (measurements) x_i, y_i
- we introduce matches $\mathbf{Z}_i = (u_i^1, v_i^1, u_i^2, v_i^2) \in \mathbb{R}^4$; $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^k$
- corrected points \hat{x}_i, \hat{y}_i ; $\hat{\mathbf{Z}}_i = (\hat{u}_i^1, \hat{v}_i^1, \hat{u}_i^2, \hat{v}_i^2)$; $\hat{\mathbf{Z}} = \{\hat{\mathbf{Z}}_i\}_{i=1}^k$ are correspondences
- correspondences satisfy the epipolar geometry exactly $\hat{\mathbf{y}}_i^\top \mathbf{F} \hat{\mathbf{x}}_i = 0, i = 1, \dots, k$
- small correction is more probable
- let $\mathbf{e}_i(\cdot)$ be the 'reprojection error' (vector) per match i ,

$$\mathbf{e}_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) = \begin{bmatrix} \mathbf{x}_i - \hat{\mathbf{x}}_i \\ \mathbf{y}_i - \hat{\mathbf{y}}_i \end{bmatrix} = \mathbf{e}_i(\mathbf{Z}_i \mid \hat{\mathbf{Z}}_i, \mathbf{F}) = \mathbf{Z}_i - \hat{\mathbf{Z}}_i(\mathbf{F}) \quad (15)$$

$$\|\mathbf{e}_i(\cdot)\|^2 \stackrel{\text{def}}{=} \mathbf{e}_i^2(\cdot) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 = \|\mathbf{Z}_i - \hat{\mathbf{Z}}_i(\mathbf{F})\|^2$$

► cont'd

- the total reprojection error (of all data) then is

$$L(Z | \hat{Z}, \mathbf{F}) = \sum_{i=1}^k e_i^2(x_i, y_i | \hat{x}_i, \hat{y}_i, \mathbf{F}) = \sum_{i=1}^k e_i^2(\mathbf{Z}_i | \hat{\mathbf{Z}}_i, \mathbf{F})$$

- and the optimization problem is

$$(\hat{Z}^*, \mathbf{F}^*) = \arg \min_{\substack{\mathbf{F} \\ \text{rank } \mathbf{F} = 2}} \min_{\substack{\hat{Z} \\ \hat{\mathbf{y}}_i^\top \mathbf{F} \hat{\mathbf{x}}_i = 0}} \sum_{i=1}^k e_i^2(x_i, y_i | \hat{x}_i, \hat{y}_i, \mathbf{F}) \quad (16)$$

Three possible approaches

- they differ in how the correspondences \hat{x}_i, \hat{y}_i are obtained:
 - direct optimization of reprojection error over all variables \hat{Z}, \mathbf{F} →99
 - Sampson optimal correction = partial correction of \mathbf{Z}_i towards $\hat{\mathbf{Z}}_i$ used in an iterative minimization over \mathbf{F} →100
 - ~~removing \hat{x}_i, \hat{y}_i altogether = marginalization of $L(Z, \hat{Z} | \mathbf{F})$ over \hat{Z} followed by minimization over \mathbf{F} not covered, the marginalization is difficult~~

Method 1: Reprojection Error Optimization: Idea

- we need to encode the constraints $\hat{\mathbf{y}}_i \mathbf{F} \hat{\mathbf{x}}_i = 0$, $\text{rank } \mathbf{F} = 2$
- idea: reconstruct 3D point via equivalent projection matrices and use reprojection error
- the equivalent projection matrices are [see \[H&Z, Sec. 9.5\] for complete characterization](#)

$$\mathbf{P}_1 = [\mathbf{I} \quad \mathbf{0}], \quad \mathbf{P}_2 = [[\mathbf{e}_2]_{\times} \mathbf{F} + \mathbf{e}_2 \mathbf{e}_1^{\top} \quad \mathbf{e}_2] \quad (17)$$

$$\mathbf{P}_1 \quad \mathbf{P}_2(\mathbf{F})$$

⊛ H3; 2pt: Given rank-2 matrix \mathbf{F} , let $\mathbf{e}_1, \mathbf{e}_2$ be the right and left nullspace basis vectors of \mathbf{F} , respectively. Verify that such \mathbf{F} is a fundamental matrix of $\mathbf{P}_1, \mathbf{P}_2$ from (17).

Hints:

- (1) consider $\hat{\mathbf{x}}_i = \mathbf{P}_1 \mathbf{X}_i$ and $\hat{\mathbf{y}}_i = \mathbf{P}_2 \mathbf{X}_i$
- (2) \mathbf{A} is skew symmetric iff $\mathbf{x}^{\top} \mathbf{A} \mathbf{x} = 0$ for all vectors \mathbf{x} .

(cont'd) Reprojection Error Optimization: Algorithm

1. compute $\mathbf{F}^{(0)}$ by the 7-point algorithm $\rightarrow 84$; construct camera $\mathbf{P}_2^{(0)}$ from $\mathbf{F}^{(0)}$ using (17)
2. triangulate 3D points $\hat{\mathbf{X}}_i^{(0)}$ from matches (x_i, y_i) for all $i = 1, \dots, k$ $\rightarrow 89$
3. starting from $\mathbf{P}_2^{(0)}$, $\hat{\mathbf{X}}_{1:k}^{(0)}$ minimize the reprojection error (15)

$$(\hat{\mathbf{X}}_{1:k}^*, \mathbf{F}^*) = \arg \min_{\mathbf{F}, \hat{\mathbf{X}}_{1:k}} \sum_{i=1}^k e_i^2(\mathbf{Z}_i \mid \hat{\mathbf{Z}}_i(\hat{\mathbf{X}}_i, \mathbf{P}_2(\mathbf{F})))$$

where

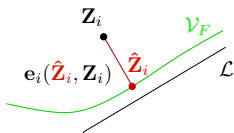
$$\hat{\mathbf{Z}}_i = (\hat{x}_i, \hat{y}_i) \text{ (Cartesian), } \hat{x}_i \simeq \mathbf{P}_1 \hat{\mathbf{X}}_i, \hat{y}_i \simeq \mathbf{P}_2(\mathbf{F}) \hat{\mathbf{X}}_i \text{ (homogeneous)}$$

- non-linear, non-convex problem
- solves \mathbf{F} estimation and triangulation of all k points jointly
- the solver would be quite slow
- $3k + 7$ parameters to be found: latent: $\hat{\mathbf{X}}_i$, for all i (correspondences!), non-latent: \mathbf{F}
- we need minimal representations for $\hat{\mathbf{X}}_i$ and \mathbf{F} $\rightarrow 150$ or introduce constraints
- there are other pitfalls; this is essentially bundle adjustment; we will return to this later $\rightarrow 138$

► Method 2: First-Order Error Approximation

An elegant method for solving problems like (16):

- we will get rid of the latent parameters \hat{X} needed for obtaining the correction [H&Z, p. 287], [Sampson 1982]
- we will recycle the algebraic error $\epsilon = \underline{y}^\top \mathbf{F} \underline{x}$ from $\rightarrow 84$
- consider matches \mathbf{Z}_i , correspondences $\hat{\mathbf{Z}}_i$, and reprojection error $\mathbf{e}_i = \|\mathbf{Z}_i - \hat{\mathbf{Z}}_i\|^2$
- correspondences satisfy $\hat{\underline{y}}_i^\top \mathbf{F} \hat{\underline{x}}_i = 0$, $\hat{\underline{x}}_i = (\hat{u}^1, \hat{v}^1, 1)$, $\hat{\underline{y}}_i = (\hat{u}^2, \hat{v}^2, 1)$
- this is a manifold $\mathcal{V}_F \in \mathbb{R}^4$: a set of points $\hat{\mathbf{Z}} = (\hat{u}^1, \hat{v}^1, \hat{u}^2, \hat{v}^2)$ consistent with \mathbf{F}
- algebraic error vanishes for $\hat{\mathbf{Z}}_i$: $\mathbf{0} = \epsilon_i(\hat{\mathbf{Z}}_i) = \hat{\underline{y}}_i^\top \mathbf{F} \hat{\underline{x}}_i$



Sampson's idea: Linearize the algebraic error $\epsilon(\mathbf{Z})$ at \mathbf{Z}_i (where it is non-zero) and evaluate the resulting linear function at $\hat{\mathbf{Z}}_i$ (where it is zero). The zero-crossing replaces \mathcal{V}_F by a linear manifold \mathcal{L} . The point on \mathcal{V}_F closest to \mathbf{Z}_i is replaced by the closest point on \mathcal{L} .

$$0 = \epsilon_i(\hat{\mathbf{Z}}_i) \approx \epsilon_i(\mathbf{Z}_i) + \frac{\partial \epsilon_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i} (\hat{\mathbf{Z}}_i - \mathbf{Z}_i) = \mathcal{O}$$

► Sampson's Approximation of Reprojection Error

- linearize $\varepsilon(\mathbf{Z})$ at match \mathbf{Z}_i , evaluate it at correspondence $\hat{\mathbf{Z}}_i$

$$\varepsilon_i(\mathbf{Z}_i) + \underbrace{\frac{\partial \varepsilon_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i}}_{\mathbf{J}_i(\mathbf{Z}_i)} \underbrace{(\hat{\mathbf{Z}}_i - \mathbf{Z}_i)}_{\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)} \stackrel{\text{def}}{=} \underbrace{\varepsilon_i(\mathbf{Z}_i) + \mathbf{J}_i(\mathbf{Z}_i)}_{\text{given}} \underbrace{\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)}_{\text{wanted}} = \varepsilon_i(\hat{\mathbf{Z}}_i) \stackrel{!}{=} 0 \quad \in \mathbb{R}$$

- goal: compute function $\mathbf{e}_i(\cdot)$ from $\varepsilon_i(\cdot)$, where $\mathbf{e}_i(\cdot)$ is the distance of $\hat{\mathbf{Z}}_i$ from \mathbf{Z}_i
- we have a linear underconstrained equation for $\mathbf{e}_i(\cdot)$ e.g. $\varepsilon_i \in \mathbb{R}$, $\mathbf{e}_i \in \mathbb{R}^4$
- we look for a minimal $\mathbf{e}_i(\cdot)$ per match i

$$\mathbf{e}_i(\cdot)^* = \arg \min_{\mathbf{e}_i(\cdot)} \|\mathbf{e}_i(\cdot)\|^2 \quad \text{subject to} \quad \varepsilon_i(\cdot) + \mathbf{J}_i(\cdot) \mathbf{e}_i(\cdot) = 0$$

- which has a closed-form solution note that $\mathbf{J}_i(\cdot)$ is not invertible! ⊛ P1; 1pt: derive $\mathbf{e}_i^*(\cdot)$

$$\begin{aligned} \mathbf{e}_i^*(\cdot) &= -\mathbf{J}_i^\top (\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \varepsilon_i(\cdot) && \text{pseudo-inverse} \\ \|\mathbf{e}_i^*(\cdot)\|^2 &= \varepsilon_i^\top(\cdot) (\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \varepsilon_i(\cdot) \end{aligned} \quad (18)$$

- this maps $\varepsilon_i(\cdot)$ to an estimate of $\mathbf{e}_i(\cdot)$ per correspondence
- we often do not need \mathbf{e}_i , just $\|\mathbf{e}_i\|^2$ exception: triangulation $\rightarrow 106$
- the unknown parameters \mathbf{F} are inside: $\mathbf{e}_i = \mathbf{e}_i(\mathbf{F})$, $\varepsilon_i = \varepsilon_i(\mathbf{F})$, $\mathbf{J}_i = \mathbf{J}_i(\mathbf{F})$

► Example: Fitting A Circle To Scattered Points

Problem: Fit an origin-centered circle \mathcal{C} : $\|\mathbf{x}\|^2 - r^2 = 0$ to a set of 2D points $Z = \{\mathbf{x}_i\}_{i=1}^k$

1. consider radial error as the 'algebraic error' $\epsilon(\mathbf{x}) = \|\mathbf{x}\|^2 - r^2$ 'arbitrary' choice
2. linearize it at $\hat{\mathbf{x}}$
 we are dropping i in ϵ_i , \mathbf{e}_i etc for clarity

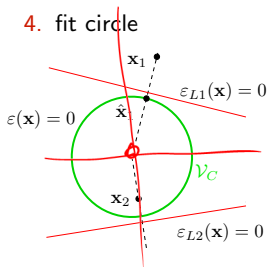
$$\epsilon(\hat{\mathbf{x}}) \approx \epsilon(\mathbf{x}) + \underbrace{\frac{\partial \epsilon(\mathbf{x})}{\partial \mathbf{x}}}_{\mathbf{J}(\mathbf{x})=2\mathbf{x}^\top} \underbrace{(\hat{\mathbf{x}} - \mathbf{x})}_{\mathbf{e}(\hat{\mathbf{x}}, \mathbf{x})} = \dots = 2\mathbf{x}^\top \hat{\mathbf{x}} - (r^2 + \|\mathbf{x}\|^2) \stackrel{\text{def}}{=} \epsilon_L(\hat{\mathbf{x}})$$

$\epsilon_L(\hat{\mathbf{x}}) = 0$ is a line with normal $\frac{\mathbf{x}}{\|\mathbf{x}\|}$ and intercept $\frac{r^2 + \|\mathbf{x}\|^2}{2\|\mathbf{x}\|}$ not tangent to \mathcal{C} , outside!

3. using (18), express error approximation \mathbf{e}^* as

$$\|\mathbf{e}^*\|^2 = \boldsymbol{\epsilon}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \boldsymbol{\epsilon} = \frac{(\|\mathbf{x}\|^2 - r^2)^2}{4\|\mathbf{x}\|^2}$$

4. fit circle



$$r^* = \arg \min_r \sum_{i=1}^k \frac{(\|\mathbf{x}_i\|^2 - r^2)^2}{4\|\mathbf{x}_i\|^2} = \dots = \left(\frac{1}{k} \sum_{i=1}^k \frac{1}{\|\mathbf{x}_i\|^2} \right)^{-\frac{1}{2}}$$

- this example results in a convex quadratic optimization problem
- note that

$$\arg \min_r \sum_{i=1}^k (\|\mathbf{x}_i\|^2 - r^2)^2 = \left(\frac{1}{k} \sum_{i=1}^k \|\mathbf{x}_i\|^2 \right)^{\frac{1}{2}}$$

Thank You

