Photogrammetry & Robotics Lab

3D Coordinate Systems
(Bsc Geodesy & Geoinformation)

12. ML solution for 3D Similarity

Wolfgang Förstner

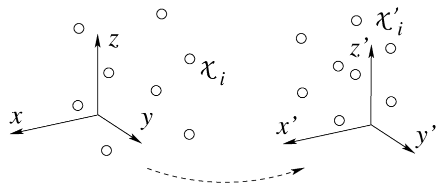
The slides have been created by Wolfgang Förstner.

1

Outline

ML estimation of parameters of similarities

- Iterative solution for nonlinear Gauss-Markov model
- 3D similarity with simplified stochastical model
  for finding CovM of direct solution
- 3D similarity with full covariance matrices per point



For this video see Förstner/Wrobel (2016), p. 102-104

2

Nonlinear Gauss-Markov model

3

Non-linear Gauss-Markov problem

Model

$$\underline{\boldsymbol{l}} \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{x}), \Sigma_{ll})$$

Observed $\boldsymbol{l}$

Given $\boldsymbol{f}, \Sigma_{ll}$

Task Minimize w.r.t. $\boldsymbol{x}$

$$\Omega(\boldsymbol{x}) = (\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}))^{\mathsf{T}} \Sigma_{ll}^{-1} (\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}))$$

- Nonlinear problem → iterative
- Approximate values → linear substitute problem
- Estimates after convergence

4

Gauss-Markov model: iterative solution

Principle

1. Provide approximate values $\boldsymbol{x}^a$ for parameters $\boldsymbol{x}$
2. Iteratively improve approximate values $\boldsymbol{x}^a$:
   1. Derive linear substitute problem $\underline{\Delta \boldsymbol{l}} \sim \mathcal{N}(A\Delta\boldsymbol{x}, \Sigma_{ll})$
   2. Solve for corrections, e.g. $\Delta\boldsymbol{x} := (A^{\mathsf{T}}\Sigma_{ll}A)^{-1}A^{\mathsf{T}}\Sigma_{ll}\Delta\boldsymbol{l}$
   3. Update approximate values $\boldsymbol{x}^a := u(\boldsymbol{x}^a, \Delta\boldsymbol{x})$
   4. If corrections too large → iterate, otherwise stop loop
3. Last approximate parameters as estimates $\widehat{\boldsymbol{x}} := \boldsymbol{x}^a$
4. Derive quality measures, e.g. $\mathbb{D}(\widehat{\underline{\boldsymbol{x}}})$

5

Comments to iterative solution

- Approximate values sometimes hard to obtain
- Linear substitute problem by Taylor expansion
- Linear substitute problem = linear GM-model

$$\boldsymbol{l} + \boldsymbol{v} \approx \boldsymbol{f}(\boldsymbol{x}^a) + A\,\Delta\boldsymbol{x}$$

$$\overbrace{(\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}^a))}^{\Delta\boldsymbol{l}} + \boldsymbol{v} \approx A\,\Delta\boldsymbol{x} \qquad \text{with} \quad A = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{x=x^a}$$

- Updates: usually additive, sometimes multiplicative
- Termination of iteration: use ratios $|\Delta x_u|/\sigma_{x_u}$
- Uncertainty: transfer from linear substitute problem

6

Gauss-Markov model of 3D similarity model

7

3D similarity

- Approximate values from direct solution
  if no outliers
- Derive substitute problem by Taylor expansion
  use multiplicative corrections for scaled rotation
- Use normal equation system (7x7)
- Update: (translation, scaled rotation)
- Number or iterations: usually 2-3
- Uncertainty by variance propagation

8

Model of 3D similarity

- Functional model

$$\boldsymbol{l}_i + \boldsymbol{v}_i = \boldsymbol{f}_i(\boldsymbol{x}) \quad \text{or} \quad \boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = \boldsymbol{t} + \lambda R \boldsymbol{x}_i \quad i = 1, \ldots, I$$

- Approximate values (taking sequence of beginning)

$$\boldsymbol{x}^a \quad \text{or} \quad \{R^a, \boldsymbol{t}^a, \lambda^a\}$$

- Updates

$$\boldsymbol{x}^a := u(\boldsymbol{x}^a, \widehat{\Delta \boldsymbol{x}}) \quad \text{or} \quad \boldsymbol{p}^a = \left\{ \begin{array}{c} R^a \\ \boldsymbol{t}^a \\ \lambda^a \end{array} \right\} := \left\{ \begin{array}{c} \mathrm{e}^{S(\Delta r)} R^a \\ \boldsymbol{t}^a + \widehat{\Delta \boldsymbol{t}} \\ \mathrm{e}^{\widehat{\Delta \lambda}} \lambda^a \end{array} \right\}$$

or

$$\begin{bmatrix} \lambda^a R^a & \boldsymbol{t}^a \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix} := \begin{bmatrix} \mathrm{e}^{\widehat{\Delta \lambda}} \mathrm{e}^{S(\widehat{\Delta \boldsymbol{r}})} & \widehat{\Delta \boldsymbol{t}} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \lambda^a R^a & \boldsymbol{t}^a \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix}$$

Model of 3D similarity

- Linearized substitute model

$$\Delta \boldsymbol{l}_i + \boldsymbol{v}_i = A_i^\mathsf{T} \Delta \boldsymbol{x} \quad \text{or} \quad \Delta \boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = A_i^\mathsf{T} \Delta \boldsymbol{p}, \quad i = 1, \ldots, I$$

- With

$$\Delta \boldsymbol{x}'_i = \boldsymbol{x}'_i - (\underbrace{\boldsymbol{t}^a + \lambda^a R^a \boldsymbol{x}_i}_{\boldsymbol{x}'^a_i}) \quad \text{and} \quad \Delta \boldsymbol{p} = \begin{bmatrix} \Delta \boldsymbol{r} \\ \Delta \boldsymbol{t} \\ \Delta \lambda \end{bmatrix}$$

- And Jacobian, design matrix, coefficient matrix,…

$$\underbrace{A_i^\mathsf{T}(\boldsymbol{x}'^a_i)}_{3\times 7} = \left.\frac{\partial \boldsymbol{f}_i(\Delta \boldsymbol{p})}{\partial \Delta \boldsymbol{p}}\right|_{p=p^a} = [-S(\boldsymbol{x}'^a_i),\, I_3,\, \boldsymbol{x}'^a_i]$$

Estimation with simplified stochastical model

Isotropic uncertainty model

Assumption

Simplified uncertainty model of observed coordinates

$$\Sigma_{l_i l_i} := \mathbb{D}(\underline{\boldsymbol{x}}'_i) = \Sigma_{x'_i x'_i} = \sigma^2_{x'_i} I_3 = \frac{1}{w_i} I_3$$

→ use centred coordinates

$${}^c\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{x}_C \quad \text{and} \quad {}^c\boldsymbol{x}'_i = \boldsymbol{x}'_i - \boldsymbol{x}'_C$$

with centroids

$$\boldsymbol{x}_C = \frac{\sum_i w_i \boldsymbol{x}_i}{\sum_i w_i} \quad \text{and} \quad \boldsymbol{x}'_C = \frac{\sum_i w_i \boldsymbol{x}'_i}{\sum_i w_i}$$

Model for centred coordinates

- Nonlinear model → centred nonlinear model

$$(\boldsymbol{x}'_i - \boldsymbol{x}'_C) + \boldsymbol{v}_{x'_i} = {}^c\boldsymbol{t} + \lambda R(\boldsymbol{x}_i - \boldsymbol{x}_C)$$

$${}^c\boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = {}^c\boldsymbol{t} + \lambda R\, {}^c\boldsymbol{x}_i$$

- Approximate values, translation = 0

$$(R^a,\, \boldsymbol{0},\, \lambda^a)$$

- Linearized model

$$\Delta^c\boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = \left[-S({}^c\boldsymbol{x}'^a_i),\, I_3,\, {}^c\boldsymbol{x}'^a_i\right] \begin{bmatrix} \widehat{\Delta \boldsymbol{r}} \\ \widehat{\Delta^c \boldsymbol{t}} \\ \widehat{\Delta\lambda} \end{bmatrix}$$

$$\Delta^c\boldsymbol{x}'_i = {}^c\boldsymbol{x}'_i - \lambda^a R^a\, {}^c\boldsymbol{x}_i = {}^c\boldsymbol{x}'_i - {}^c\boldsymbol{x}'^a_i$$

Normal equation system

- 7x7 Normal equation matrix

$$N = A^{\mathsf{T}} W_{ll} A := \sum_{i=1}^{I} \begin{bmatrix} S({}^c\boldsymbol{x}'^a_i) \\ I_3 \\ {}^c\boldsymbol{x}'^{a\mathsf{T}}_i \end{bmatrix} \left[-S({}^c\boldsymbol{x}'^a_i),\, I_3,\, {}^c\boldsymbol{x}'^a_i\right] w_i$$

$$= \begin{bmatrix} \sum_{i=1}^{I} S({}^c\boldsymbol{x}'^a_i) S^{\mathsf{T}}({}^c\boldsymbol{x}'^a_i) w_i & 0_{3\times 3} & \boldsymbol{0} \\ 0_{3\times 3} & \sum_{i=1}^{I} w_i I_3 & \boldsymbol{0} \\ \boldsymbol{0}^{\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} & \sum_{i=1}^{I} {}^c\boldsymbol{x}'^{a\mathsf{T}}_i\, {}^c\boldsymbol{x}'^a_i w_i \end{bmatrix}$$

→ block diagonal matrix, one 3x3 block for rotation

Normal equation system

- 7-vector of right hand sides

$${}^c\boldsymbol{x}'_i = \boldsymbol{x}'_i - \boldsymbol{x}'_C$$

$$\boldsymbol{h} = A^{\mathsf{T}} W_{ll}\, {}^c\Delta \boldsymbol{l} := \sum_{i=1}^{I} \begin{bmatrix} S({}^c\boldsymbol{x}'^a_i) \\ I_3 \\ {}^c\boldsymbol{x}'^{a\mathsf{T}}_i \end{bmatrix} ({}^c\boldsymbol{x}'_i - {}^c\boldsymbol{x}'^a_i)\, w_i$$

$${}^c\boldsymbol{x}'^a_i = \lambda^a R^a\, (\boldsymbol{x}_i - \boldsymbol{x}_C)$$

$$= \begin{bmatrix} \sum_{i=1}^{I} S({}^c\boldsymbol{x}'^a_i)({}^c\boldsymbol{x}'_i - {}^c\boldsymbol{x}'^a_i) w_i \\ \boldsymbol{0} \\ \sum_{i=1}^{I} {}^c\boldsymbol{x}'^{a\mathsf{T}}_i ({}^c\boldsymbol{x}'_i - {}^c\boldsymbol{x}'^a_i) w_i \end{bmatrix}$$

→ estimated translation = 0

Estimated parameters

- Update of scale factor

$$\lambda^a := (1 + \widehat{\Delta\lambda})\lambda^a$$

with

$$\widehat{\Delta\lambda} = \frac{\sum_{i=1}^{I} {}^c\boldsymbol{x}'^{a\mathsf{T}}_i ({}^c\boldsymbol{x}'_i - {}^c\boldsymbol{x}'^a_i) w_i}{\sum_{i=1}^{I} |{}^c\boldsymbol{x}'^a_i|^2\, w_i}$$

Remark: in order to guarantee positivity use $1 + x \approx \mathrm{e}^x$

$$\lambda^a := \mathrm{e}^{\widehat{\Delta\lambda}}\, \lambda^a$$

→ compare to

$$R^a := \mathrm{e}^{\mathsf{S}(\Delta r)}\, R^a$$

Estimated parameters

- Updated rotation matrix

$$R^a := R(\widehat{\Delta r})R^a$$

with parameters from 3x3 system

$$\underbrace{\left(\sum_{i=1}^{I} S({}^c\boldsymbol{x}_i'^a) S^{\mathsf{T}}({}^c\boldsymbol{x}_i'^a) w_i\right)}_{3\times3} \widehat{\Delta r} = \underbrace{\sum_{i=1}^{I} S({}^c\boldsymbol{x}_i'^a)({}^c\boldsymbol{x}_i' - {}^c\boldsymbol{x}_i'^a) w_i}_{3\times1}$$

- Estimated translation parameter

$$\boldsymbol{t}^a := \boldsymbol{x}'_C - \lambda^a R^a \boldsymbol{x}_C$$

17

3D similarity: Matlab code (1/3)

```
function A_est = estimate_3d_similarity(X,Xt,R0,l0,niter)
%% xt_i = l * R * x_i + t, i=1,...,I, weights = 1
%  X, Xt     3xI matrices of given and observed points
%  R0, l0    3x3 matrix approximate R, lambda
%  niter     number of iterations
I    = size(X,2);                          % number I
X    = [X ;ones(1,I)];Xt =[Xt;ones(1,I)]; % -> homogeneous
Xmean  = mean(X,2);                        % centroids
Xtmean = mean(Xt,2);
cX   = X  - [  Xmean(1:3);0]*ones(1,I);   % centred data
cXt  = Xt - [ Xtmean(1:3);0]*ones(1,I);
A0   = [l0*R0,[0,0,0]';[0,0,0],1];         % initial A
...
```

18

3D similarity: Matlab code (2/3)

```
...
for iter = 1:niter  % ======== begin of iterations ========
  cXt0   = A0 * cX;  dl = Xtq-Xtq0;  % appr. transf. points, corr.
  Norm   = zeros(3); hv = zeros(3,1);% initialize NE(dr), h
  suml_z = 0;  suml_d = 0;           % ... sums for lambda
  for i = 1:I    % ......... begin of NE building .........
    A    = -calc_S(cXt0(1:3,i));     % Jacobian for rotation
    Norm = Norm + A' * A;            % add contribution to N
    hv   = hv + A' * dl(1:3,i);      % ... to h, z, n
    suml_z = suml_z + cXt0(1:3,i)'*(cXt(1:3,i)-cXt0(1:3,i));
    suml_d = suml_d + cXt0(1:3,i)'* cXt0(1:3,i);
  end            %  .......... end of NE building .........
...
```

19

3D similarity: Matlab code (3/3)

```
...
% solve normal equations and update
  dr    = Norm\hv;                       % corr. dr to u
  R0    = calc_Rot_r(dr) * R0;           % updated r
  l0    = exp(suml_z/suml_d) * l0;       % updated l
  A0    = [l0*R0,[0,0,0]';[0,0,0],1]; % new transform
end % ============ end of iterations ==============
% provide final results
t_est = Xtmean(1:3)-l0*R0*Xmean(1:3);  % translation
A_est = [l0*R0,t_est;[0,0,0],1];       % similarity
end
```

20

Comments

- Add criterium for stopping iterations,
  estimated scale and rotation changes: unitless

$$\max(|\Delta\lambda|, |\Delta r_i|) < T \quad \text{with} \quad T = 10^{-6}$$

- Add output of residuals
- Add output of uncertainty measures

Estimation with full covariance matrices

General 3D similarity

- Normal equation matrix

$$N = A^{\mathsf{T}} W A := \sum_i A_i(\boldsymbol{x}_i'^a)\, W_{ii}\, A^{\mathsf{T}}(\boldsymbol{x}_i'^a)$$

- Right hand side

$$\boldsymbol{h} = A^{\mathsf{T}} W \Delta \boldsymbol{l} := \sum_i A_i(\boldsymbol{x}_i'^a)\, W_{ii}\, \Delta\boldsymbol{x}_i'$$

- Corrections

$$\widehat{\Delta\boldsymbol{x}} := \begin{bmatrix} \widehat{\Delta\boldsymbol{\theta}} \\ \widehat{\Delta^c\boldsymbol{t}} \\ \widehat{\Delta\lambda} \end{bmatrix}$$

General 3D similarity

Using centroids → numerically more stable

- Centroids, choose any, translation generally not 0

$$\boldsymbol{x}_C = \left(\sum_i W_{ii}\right)^{-1} \sum_i W_{ii}\boldsymbol{x}_i \quad \text{and} \quad \boldsymbol{x}_C' = \ldots$$

- Centred coordinates

$$^c\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{x}_C \quad \text{and} \quad {}^c\boldsymbol{x}_i' = \boldsymbol{x}_i - \boldsymbol{x}_C$$

- In normal equations: replace

$$\begin{aligned} \boldsymbol{x}_i'^a &\mapsto {}^c\boldsymbol{x}_i'^a = \lambda^a R^a\, {}^c\boldsymbol{x}_i \\ \Delta\boldsymbol{x}_i' &\mapsto {}^c\Delta\boldsymbol{x}_i' = {}^c\boldsymbol{x}_i' - {}^c\boldsymbol{x}_i'^a \\ \widehat{\boldsymbol{t}} &\mapsto {}^c\widehat{\boldsymbol{t}} \end{aligned}$$

Updates and undoing centering

- Updates

$$\boldsymbol{x}^a := u(\boldsymbol{x}^a, \widehat{\Delta \boldsymbol{x}}) \qquad \begin{aligned} R^a &:= e^{S(\widehat{\Delta\theta})}\, R^a \\ {}^c\boldsymbol{t}^a &:= {}^c\boldsymbol{t}^a + \widehat{\Delta {}^c\boldsymbol{t}} \\ \lambda^a &:= e^{\widehat{\Delta\lambda}}\, \lambda^a \end{aligned}$$

- Estimates

$${}^c\widehat{\boldsymbol{x}} := \left\{ \begin{array}{c} \widehat{R} \\ {}^c\widehat{\boldsymbol{t}} \\ \widehat{\lambda} \end{array} \right\}$$

- Undo centering for translation

$$\widehat{\boldsymbol{t}} = {}^c\widehat{\boldsymbol{t}} + \boldsymbol{x}'_C - \widehat{\lambda}\widehat{R}\boldsymbol{x}_C$$

Couldn't we have done everything with quaternions?

Model of 3D similarity: quaternion form

Given pairs of pure quaternions (only vector part $\neq 0$)

$$\mathbf{x}_i = (0, \boldsymbol{x}_i) \quad \text{and} \quad \mathbf{x}'_i = (0, \boldsymbol{x}'_i)$$

Unknown parameters

translation: pure quaternion

scaled rotation: unconstrained quaternion

$$\mathbf{t} = (0, \boldsymbol{t}) \qquad \mathbf{q} = (q, \boldsymbol{q}) = \sqrt{\lambda}\, \underbrace{\frac{\mathbf{q}}{|\mathbf{q}|}}_{\mathcal{R}} = \sqrt{\lambda}\, (\cos\tfrac{\theta}{2}, \sin\tfrac{\theta}{2}\, \boldsymbol{r})$$

Model $\boxed{\mathbf{x}'_i = \mathbf{t} + \mathbf{q}\mathbf{x}_i\overline{\mathbf{q}}}$

Model of 3D similarity: matrix form

- **Scaled rotation** matrix

$$Q = \lambda R$$

- Use quaternion for rotation $R_Q(\mathbf{q})$, scale $\lambda = |\mathbf{q}|^2$

$$Q(\mathbf{q}) := |\mathbf{q}|^2 R_Q(\mathbf{q}) = \left[\begin{array}{ccc} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\,(q_1 q_2 - q_0 q_3) & 2\,(q_1 q_3 + q_0 q_2) \\ 2\,(q_2 q_1 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\,(q_2 q_3 - q_0 q_1) \\ 2\,(q_3 q_1 - q_0 q_2) & 2\,(q_3 q_2 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{array} \right]$$

- Nonlinear model

see video 5

$$\boxed{\boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = \boldsymbol{t} + Q(\mathbf{q})\boldsymbol{x}_i}$$

→ 2D similarity with scaled rotation $Z(s) = \left[\begin{array}{cc} a & -b \\ b & a \end{array} \right]$

$$\boldsymbol{x}'_i + \boldsymbol{v}_{x'_i} = \boldsymbol{r} + Z(\mathbf{s})\boldsymbol{x}_i$$

Linearize 3D similarity

- Multiplicative update of quaternions/scaled rotation
  small quaternion s , to be estimated
  $$\mathbf{q} = (\mathbf{1}+\mathrm{s})\mathbf{q}^a \quad \text{or} \quad Q(\mathbf{q}) = Q(\mathbf{1}+\mathrm{s})Q(\mathbf{q}^a)$$
- General scaled rotation (! No normalization)
  $$Q(\mathbf{q}) = ((q^2 - \boldsymbol{q}^\mathsf{T}\boldsymbol{q})I_3 + 2qS_q + 2\boldsymbol{q}\boldsymbol{q}^\mathsf{T})$$
- Small scaled rotation
  $$Q(\mathbf{1}+\mathrm{s}) = ((1+s)^2 - \boldsymbol{s}^\mathsf{T}\boldsymbol{s})I_3 + 2(1+s)S_s + 2\boldsymbol{s}\boldsymbol{s}^\mathsf{T}$$
- → linearized, omit quadratic terms
  $$\approx \quad I_3 + 2sI_3 + 2S_s$$

29

Linearized model

- Approximately transformed points
  $$\boldsymbol{x}_i'^a = \boldsymbol{t}^a + Q(\mathbf{q}^a)\,\boldsymbol{x}_i$$
- Linearized observations
  $$\Delta\boldsymbol{x}_i' = \boldsymbol{x}_i' - \boldsymbol{x}_i'^a$$
- Linearized model
  $$\Delta\boldsymbol{x}_i' + \boldsymbol{v}_{x_i'} = \Delta\boldsymbol{t} + (2\Delta sI_3 + 2S_{\Delta s})\boldsymbol{x}_i'^a = A_i^\mathsf{T}\begin{bmatrix}\Delta\boldsymbol{t}\\ \Delta\mathrm{s}\end{bmatrix}$$

30

Linearized model

- Linearized model
  $$\Delta\boldsymbol{x}_i' + \boldsymbol{v}_{x_i'} = \Delta\boldsymbol{t} + (2\Delta sI_3 + 2S_{\Delta s})\boldsymbol{x}_i'^a = A_i^\mathsf{T}\begin{bmatrix}\Delta\boldsymbol{t}\\ \Delta\mathrm{s}\end{bmatrix}$$
- Design matrix for translation and scaled rotation
  $$A_i^\mathsf{T} := \left[\left\{I_3 \mid Y^\mathsf{T}(\boldsymbol{x}_i'^a)\right\}\right] \quad \text{with} \quad Y^\mathsf{T}(\boldsymbol{x}) = 2\,[\boldsymbol{x},\ -S(\boldsymbol{x})]$$

→see 2D similarity (linear)
$${}^2A_i^\mathsf{T} := [\{I_2 \mid Z(\boldsymbol{x}_i'^a)\}] \qquad \text{with} \quad Z(\boldsymbol{x}) = [\boldsymbol{x}, \boldsymbol{x}^\perp]$$

31

… comparison to Jacobian with $(\lambda, \boldsymbol{\theta})$

- Scale factor, small factor
  $$\lambda = |\mathbf{q}|^2 \quad \text{and} \quad 1 + \mathrm{d}\lambda = |\mathbf{1}+\mathrm{ds}|^2$$
- Relation to small factor $1+\mathrm{d}\lambda$ and rotation vector $\mathrm{d}\boldsymbol{\theta}$
  $$\begin{aligned}\mathbf{1}+\mathrm{ds} &= (\sqrt{1+\mathrm{d}\lambda}, \mathbf{0})(1, \tfrac{1}{2}\mathrm{d}\boldsymbol{\theta}) \approx \\ &\approx (1+\tfrac{1}{2}\mathrm{d}\lambda)(1, \tfrac{1}{2}\mathrm{d}\boldsymbol{\theta}) \\ &\approx (1+\tfrac{1}{2}\mathrm{d}\lambda, \tfrac{1}{2}\mathrm{d}\boldsymbol{\theta})\end{aligned}$$
- Differential quaternion = ½ scale and rotation
  $$\mathrm{ds} = \tfrac{1}{2}(\mathrm{d}\lambda, \mathrm{d}\boldsymbol{\theta})$$

32

… comparison to Jacobian with $(\lambda, \boldsymbol{\theta})$

- Linearized model with **scale** / **rotation** parameters

$$\Delta \mathbf{x}'_i + \mathbf{v}_{x'_i} = J_{x'_i \Delta p} \begin{bmatrix} \Delta \boldsymbol{t} \\ \Delta \boldsymbol{s} \end{bmatrix} \quad \text{with} \quad \Delta \boldsymbol{s} = \begin{bmatrix} \Delta \lambda \\ \Delta \boldsymbol{\theta} \end{bmatrix}$$

- Design matrix w.r.t. $(\mathrm{d}\boldsymbol{t}, \mathrm{d}\lambda, \mathrm{d}\boldsymbol{\theta})$

$$J_{x'_i \Delta p} = \begin{bmatrix} I_3 & \boldsymbol{x}'^a_i & -S(\boldsymbol{x}'^a_i) \\ \mathbf{0}^\mathsf{T} & 0 & \mathbf{0}^\mathsf{T} \end{bmatrix}$$

Equivalence to classical scale and rotation

Compare $\underset{4\times 7}{J_{x'\Delta p}} = \left.\dfrac{\partial \mathbf{x}'}{\partial(\Delta \boldsymbol{p})}\right|_{x'=x'^a}$ here $\Delta \boldsymbol{p} = \begin{bmatrix} \Delta \boldsymbol{\theta} \\ \Delta \lambda \\ \Delta \boldsymbol{t} \end{bmatrix}$

$$= \left.\begin{bmatrix} -S(\boldsymbol{x}') & I_3 & \boldsymbol{x}' \\ \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}\right|_{x'=x'^a}$$

$$= \left.\begin{bmatrix} 0 & x'_3 & -x'_2 & 1 & 0 & 0 & x'_1 \\ -x'_3 & 0 & x'_1 & 0 & 1 & 0 & x'_2 \\ x'_2 & -x'_1 & 0 & 0 & 0 & 1 & x'_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right|_{x'=x'^a}$$

from video 7 (! Sorting)

with (nonhomogeneous coordinates)

$$A_i^\mathsf{T} = \left[I_3 \mid Y^\mathsf{T}(\boldsymbol{x}'^a_i) \right] \quad \text{with} \quad Y^\mathsf{T}(\boldsymbol{x}) = [\underset{\lambda}{\boldsymbol{x}} \mid \underset{\theta}{-S(\boldsymbol{x})}]$$

Summary

1. General solution for arbitrary covariance matrix
   anisotropic uncertainty of coordinates
   - use centroids for improving numerical stability
   - Required for testing of residuals, deformations, …

2. Specific solution for isotropic uncertainty
   → Useful for deriving covariance matrix for direct solution

3. Model and linearization using quaternions

Next lecture

5.4 Uncertainty analysis

References of video series

Arun, K. S., T. S. Huang, and S. B. Blostein (1987). Least-Squares Fitting of Two 3D Point Sets. IEEE T-PAMI 9 (5), 698-700.

Bishop, C. (2006). Pattern Recognition and Machine Learning. Springer.

Boyd, S. and L. Vandenberghe (2004). Convex optimization. Cambridge University Press.

Cayley, A. (1846). Sur quelques proprétés des déterminants gauches. Journal für die reine und angewandte Mathematik 32, 119-123

Cover, T. and J. A. Thomas (1991). Elements of Information Theory. John Wiley & Sons.

H. S. M. Coxeter (1946): Quaternions and Reflections, The American Mathematical Monthly, 53(3), 136-146

Fallat, S. M. and M. J. Tsatsomeros (2002). On the Cayley Transform of Positivity Classes of Matrices. Electronic Journal of Linear Algebra 9, 190-196.

Fischler, M. A. and R. C. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24 (6), 381-395.

Förstner, W. and B. P. Wrobel (2016). Photogrammetric Computer Vision Statistics, Geometry, Orientation and Reconstruction. Springer.

Horn, B. K. B. (1987). Closed-form Solution of Absolute Orientation Using Unit Quaternions. Optical Soc. of America.

Howell, T. D. and J.-C. Lafon (1975). The Complexity of the Quaternion Product. Technical Report TR75-245, Cornell University.

Kanatani, K. (1990). Group Theoretical Methods in Image Understanding. New York: Springer.

Koch, K.-R. (1999). Parameter Estimation and Hypothesis Testing in Linear Models (2nd ed.). Springer.

Li, S. Z. (2000). Markov random field modeling in computer vision. Springer.

McGlone, C. J., E. M. Mikhail, and J. S. Bethel (2004). Manual of Photogrammetry (5th ed.). Maryland, USA: American Society of Photogrammetry and Remote Sensing.

Mikhail, E. M. and F. Ackermann (1976). Observations and Least Squares. University Press of America

Mikhail, E. M., J. S. Bethel, and J. C. McGlone (2001). Introduction to Modern Photogrammetry. Wiley.

Palais, B. and R. Palais (2007). Euler's fixed point theorem: The axis of a rotation. J. fixed point theory appl. 2, 215-220.

Raguram, R., O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm (2013). USAC: A Universal Framework for Random Sample Consensus. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8), 2022-2038.

Rao, R. C. (1973). Linear Statistical Inference and Its Applications. New York: Wiley.

Rodriguez, O. (1840). Des lois géometriques qui regissent les deplacements d'un système solide independament des causes qui peuvent les produire. Journal de mathématiques pures et appliquées 1 (5), 380440.

Sansò, F. (1973). An Exact Solution of the Roto-Translation Problem. Photogrammetria 29 (6), 203-206.

Vaseghi, S. V. (2000). Advanced Digital Signal Processing and Noise Reduction. Wiley.

Weber, M. (2003). Rotation between two vectors. Personal communication, Bonn.