# Estimation of 2D Projective Transformation

Computer Vision II

CSE 252B

Lecture 11

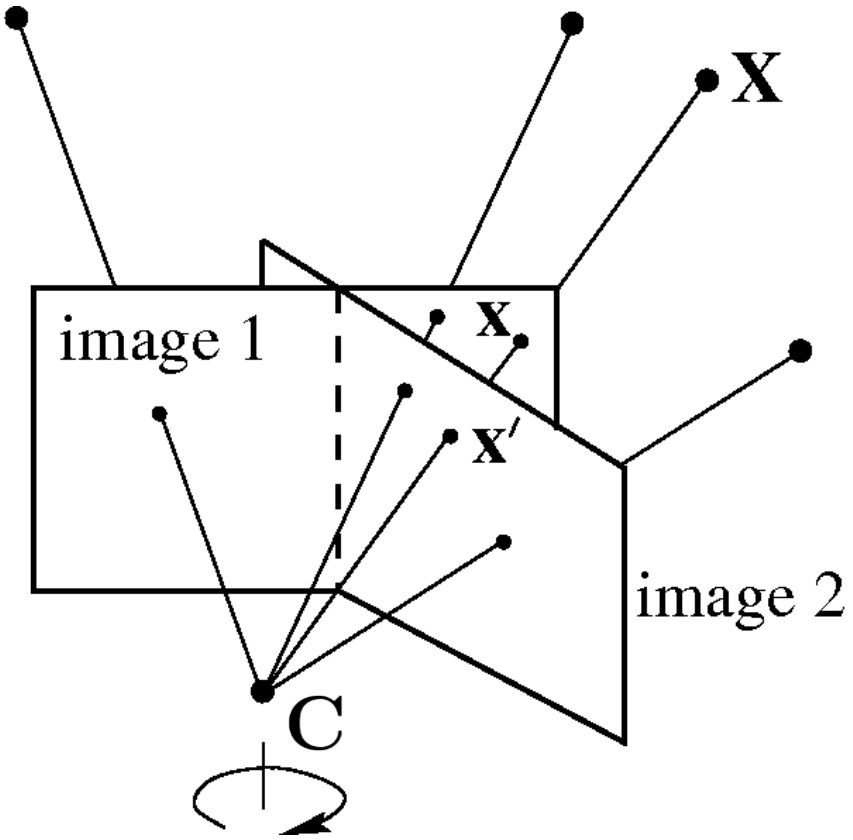
# Announcements

- Assignment 3 is due Feb 18, 11:59 PM
- Reading
  - Sections 4.1, 4.4, and 4.5

# Two view geometry

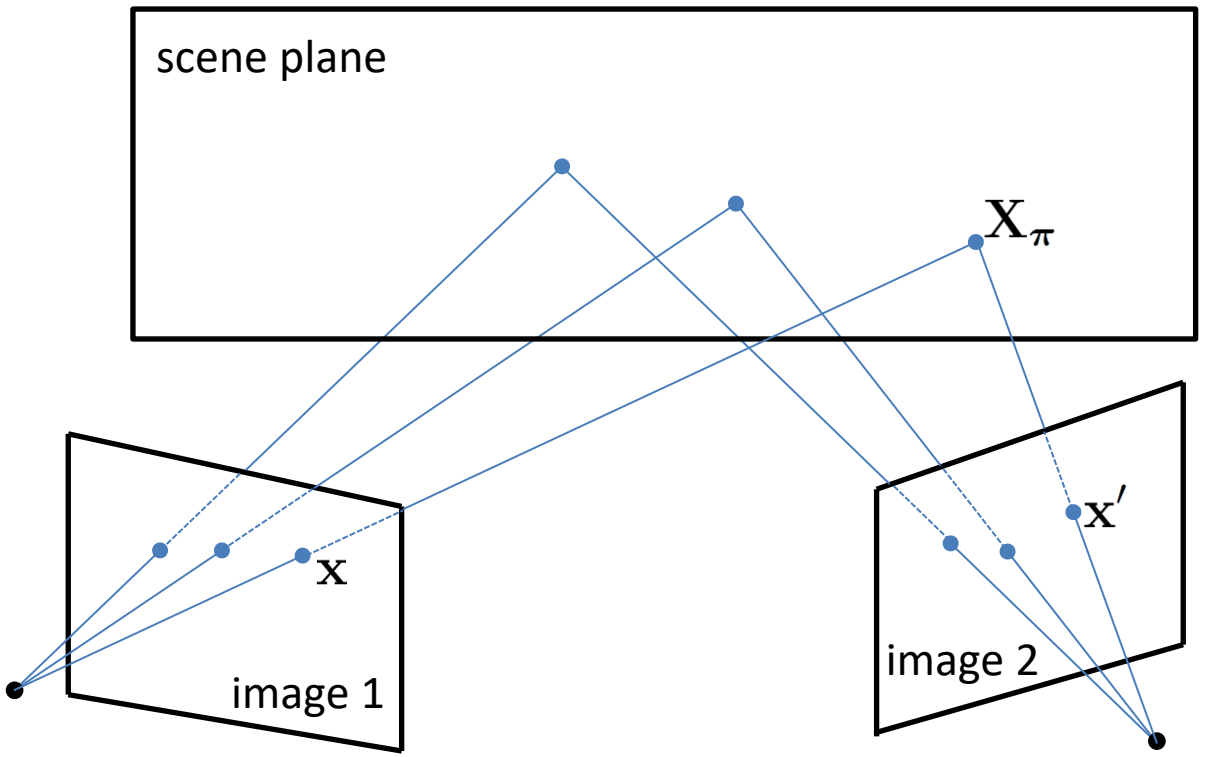
# Two view geometry

- Rotation about the same camera center



# Two view geometry

- Imaging a plane



# Two view geometry

- Same model for rotation about the same camera center and imaging a plane
  - 2D projective transformation

# 2D geometric transformations

- Projective

$3 \times 3$ homogeneous (i.e., defined up to nonzero scale) matrix (8 degrees of freedom)

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ v_1 & v_2 & u \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & u \end{bmatrix} \mathbf{x}, \text{ where } \mathtt{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \mathbf{t} = (t_1, t_2)^\top, \text{ and } \mathbf{v} = (v_1, v_2)^\top$$

$$\mathbf{x}' = \mathtt{H}_\mathrm{P}\mathbf{x}, \text{ where } \mathtt{H}_\mathrm{P} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & u \end{bmatrix}$$

# 2D geometric transformations

Linear transformation of points in homogeneous coordinates

$$\mathbf{x}' = \mathtt{H}\mathbf{x}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} xh_{11} + yh_{12} + wh_{13} \\ xh_{21} + yh_{22} + wh_{23} \\ xh_{31} + yh_{32} + wh_{33} \end{bmatrix}$$

Nonlinear transformation of points in inhomogeneous coordinates

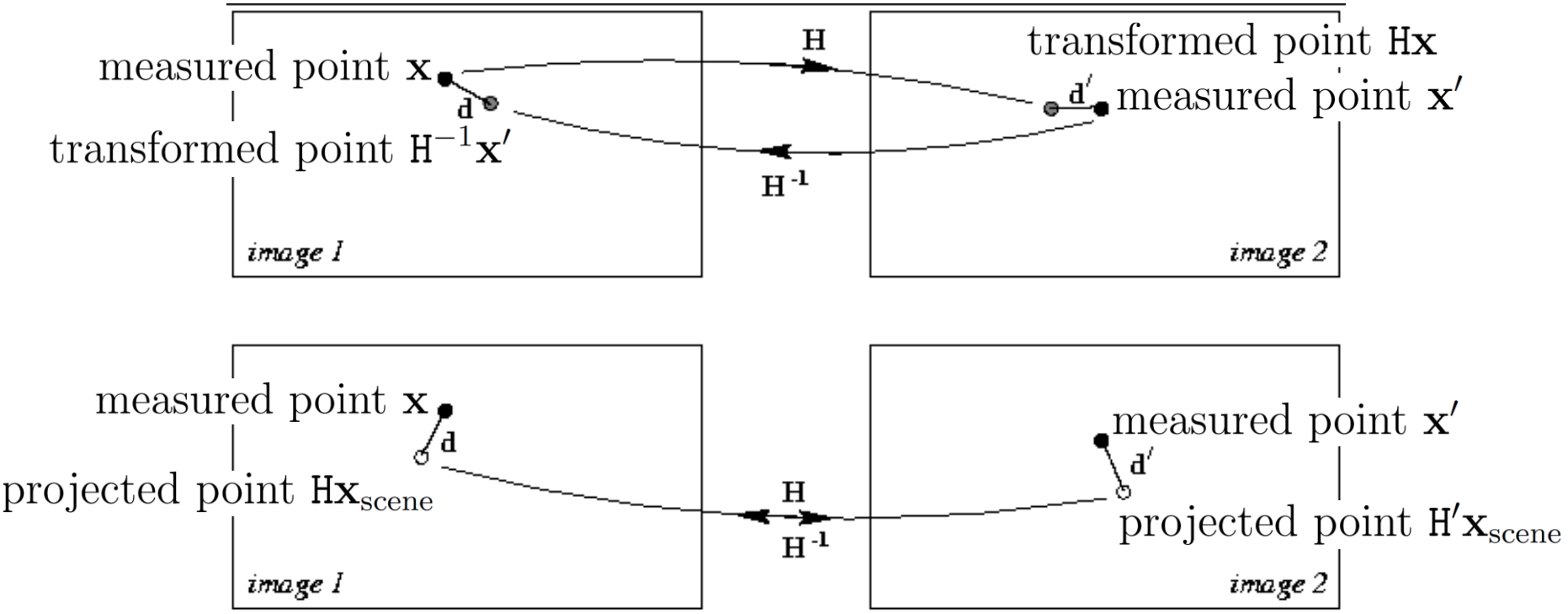
$$\tilde{x} = \frac{x'}{w'} = \frac{xh_{11} + yh_{12} + wh_{13}}{xh_{31} + yh_{32} + wh_{33}} \text{ and}$$

$$\tilde{y} = \frac{y'}{w'} = \frac{xh_{21} + yh_{22} + wh_{23}}{xh_{31} + yh_{32} + wh_{33}}$$

Do not forget this is a nonlinear transformation

# 2D projective transformation

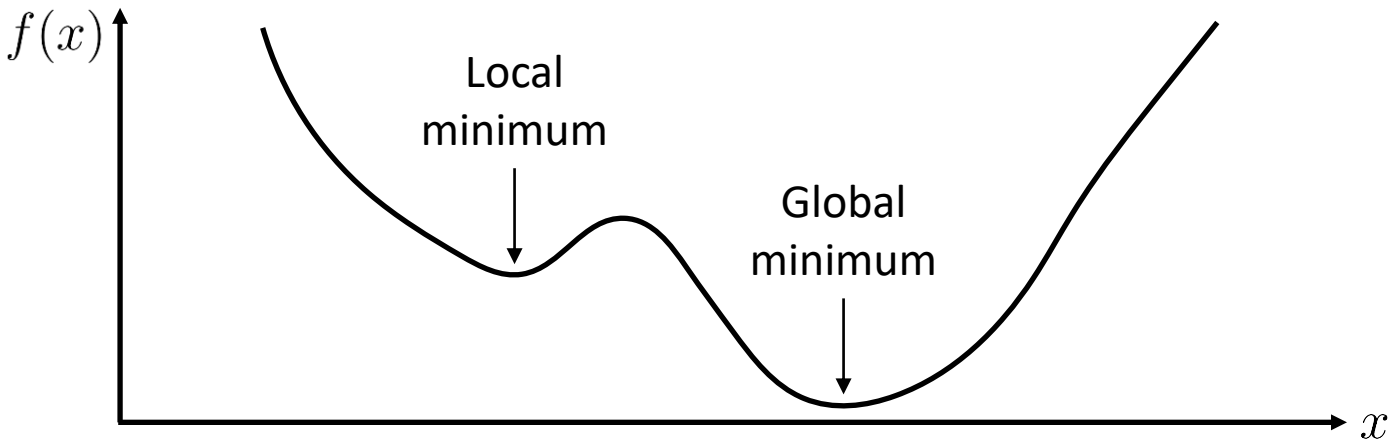
- Error in image point measurements



# Estimation of 2D projective transformation

$$\mathbf{x}_i = \mathtt{H}\mathbf{x}_{\text{scene},i}\ \forall\, i \text{ and } \mathbf{x}'_i = \mathtt{H}'\mathbf{x}_{\text{scene},i}\ \forall\, i, \text{ set } \mathtt{H} = \mathtt{I} \text{ and solve for } \mathtt{H}'$$

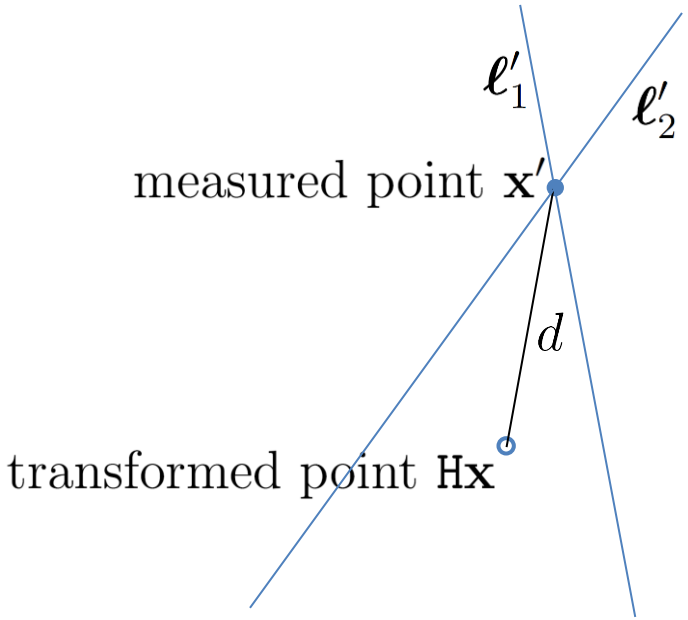
- Given point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
- Minimize geometric error $\sum_i \left(d(\mathbf{x}_i, \mathtt{H}\mathbf{x}_{\text{scene},i})^2 + d(\mathbf{x}'_i, \mathtt{H}'\mathbf{x}_{\text{scene},i})^2\right)$
- A nonlinear mapping requires a nonlinear optimization problem solver
  - Use linear estimation for initial estimate (to mitigate converging to local optimum)
  - Iterative process to determine global optimum



# Estimation of 2D projective transformation, linear

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

- Minimize algebraic error between the transformed point and two lines intersecting at the measured point



# Points and hyperplanes

- Two 2D lines define a 2D point

2D point at intersection of two 2D lines in general position is (right) null space of matrix

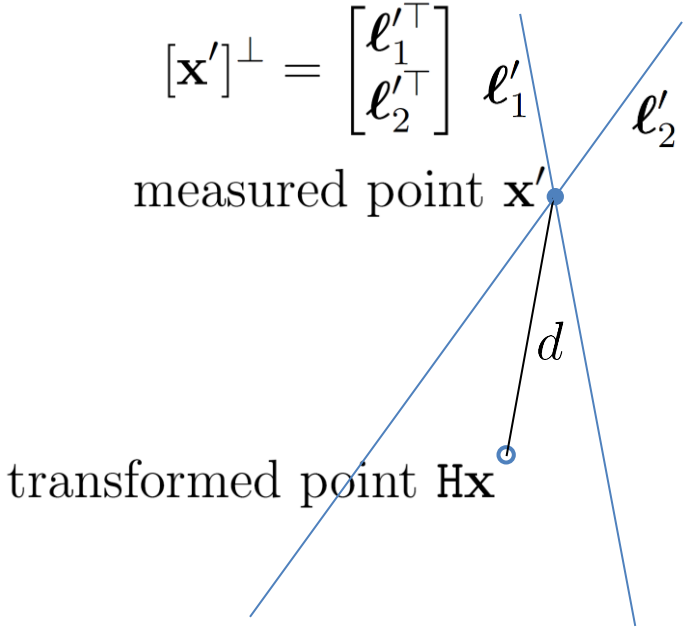
$$\begin{bmatrix} \boldsymbol{\ell}_1^\top \\ \boldsymbol{\ell}_2^\top \end{bmatrix} \mathbf{x} = \mathbf{0}$$

Two 2D lines intersecting at a 2D point is the left null space of 2D point

$$[\mathbf{x}]^\perp = \begin{bmatrix} \boldsymbol{\ell}_1^\top \\ \boldsymbol{\ell}_2^\top \end{bmatrix}$$

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

- Minimize algebraic error between the transformed point and two lines intersecting at the measured point




$$[\mathbf{x}']^{\perp}\mathbf{x}' = \mathbf{0} \qquad \mathbf{x}' = \mathtt{H}\mathbf{x}$$
$$[\mathbf{x}']^{\perp}\mathtt{H}\mathbf{x} = \mathbf{0}$$

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

$$[\mathbf{x}']^{\perp}\mathtt{H}\mathbf{x} = \mathbf{0}$$

$$\begin{bmatrix}\boldsymbol{\ell}_1'^{\top}\\ \boldsymbol{\ell}_2'^{\top}\end{bmatrix}\begin{bmatrix}\mathbf{h}^{1\top}\\ \mathbf{h}^{2\top}\\ \mathbf{h}^{3\top}\end{bmatrix}\mathbf{x} = \mathbf{0}, \text{ where } [\mathbf{x}']^{\perp} = \begin{bmatrix}\boldsymbol{\ell}_1'^{\top}\\ \boldsymbol{\ell}_2'^{\top}\end{bmatrix} \text{ and } \mathtt{H} = \begin{bmatrix}\mathbf{h}^{1\top}\\ \mathbf{h}^{2\top}\\ \mathbf{h}^{3\top}\end{bmatrix}$$

$$\begin{bmatrix}a_1' & b_1' & c_1'\\ a_2' & b_2' & c_2'\end{bmatrix}\begin{bmatrix}\mathbf{h}^{1\top}\mathbf{x}\\ \mathbf{h}^{2\top}\mathbf{x}\\ \mathbf{h}^{3\top}\mathbf{x}\end{bmatrix} = \mathbf{0}, \text{ where } \boldsymbol{\ell}_1' = (a_1', b_1', c_1')^{\top} \text{ and } \boldsymbol{\ell}_2' = (a_2', b_2', c_2')^{\top}$$

$$\begin{bmatrix}a_1'\mathbf{h}^{1\top}\mathbf{x} + b_1'\mathbf{h}^{2\top}\mathbf{x} + c_1'\mathbf{h}^{3\top}\mathbf{x}\\ a_2'\mathbf{h}^{1\top}\mathbf{x} + b_2'\mathbf{h}^{2\top}\mathbf{x} + c_2'\mathbf{h}^{3\top}\mathbf{x}\end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix}a_1'\mathbf{x}^{\top} & b_1'\mathbf{x}^{\top} & c_1'\mathbf{x}^{\top}\\ a_2'\mathbf{x}^{\top} & b_2'\mathbf{x}^{\top} & c_2'\mathbf{x}^{\top}\end{bmatrix}\begin{bmatrix}\mathbf{h}^1\\ \mathbf{h}^2\\ \mathbf{h}^3\end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix}\boldsymbol{\ell}_1'^{\top}\otimes\mathbf{x}^{\top}\\ \boldsymbol{\ell}_2'^{\top}\otimes\mathbf{x}^{\top}\end{bmatrix}\mathbf{h} = \mathbf{0}, \text{ where } \otimes \text{ denotes the Kronecker product}$$

$$\text{and } \mathbf{h} = \mathrm{vec}(\mathtt{H}^{\top}) = \begin{bmatrix}\mathbf{h}^1\\ \mathbf{h}^2\\ \mathbf{h}^3\end{bmatrix}$$

$$\left(\begin{bmatrix}\boldsymbol{\ell}_1'^{\top}\\ \boldsymbol{\ell}_2'^{\top}\end{bmatrix}\otimes\mathbf{x}^{\top}\right)\mathbf{h} = \mathbf{0}$$

$$\left([\mathbf{x}']^{\perp}\otimes\mathbf{x}^{\top}\right)\mathbf{h} = \mathbf{0}$$

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

Given $n \geq 4$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

$$\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ solve for } \mathtt{H}$$

$$[\mathbf{x}'_i]^{\perp}\mathtt{H}\mathbf{x}_i = \mathbf{0} \,\forall\, i, \text{ solve for } \mathtt{H}$$

$$\left([\mathbf{x}'_i]^{\perp} \otimes \mathbf{x}_i^{\top}\right)\mathbf{h} = \mathbf{0} \,\forall\, i, \text{ solve for } \mathbf{h} = \mathrm{vec}(\mathtt{H}^{\top})$$

$$\begin{bmatrix} [\mathbf{x}'_1]^{\perp} \otimes \mathbf{x}_1^{\top} \\ [\mathbf{x}'_2]^{\perp} \otimes \mathbf{x}_2^{\top} \\ \vdots \\ [\mathbf{x}'_n]^{\perp} \otimes \mathbf{x}_n^{\top} \end{bmatrix} \mathbf{h} = 0$$

$$\mathtt{A}\mathbf{h} = \mathbf{0}, \text{ where } \mathtt{A} = \begin{bmatrix} [\mathbf{x}'_1]^{\perp} \otimes \mathbf{x}_1^{\top} \\ [\mathbf{x}'_2]^{\perp} \otimes \mathbf{x}_2^{\top} \\ \vdots \\ [\mathbf{x}'_n]^{\perp} \otimes \mathbf{x}_n^{\top} \end{bmatrix}$$

But, data normalization must be used

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

Given $n \geq 5$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

$$\mathtt{A}\mathbf{h} = \mathbf{0}, \text{ where } \mathtt{A} = \begin{bmatrix} [\mathbf{x}'_1]^{\perp} \otimes \mathbf{x}_1^{\top} \\ [\mathbf{x}'_2]^{\perp} \otimes \mathbf{x}_2^{\top} \\ \vdots \\ [\mathbf{x}'_n]^{\perp} \otimes \mathbf{x}_n^{\top} \end{bmatrix}$$

In general, due to error in measurements, the (right) null space of the matrix $\mathtt{A}$ will be empty (i.e., $\mathtt{A}$ will be full rank). Instead, solve for the vector $\mathbf{h}$ such that $\|\mathtt{A}\mathbf{h}\|$ is minimized.

$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^{\top}$$

The solution vector $\mathbf{h}$ is the column of $\mathtt{V}$ corresponding to the smallest singular value (i.e., the last column of $\mathtt{V}$ (or last row of $\mathtt{V}^{\top}$)).

But, data normalization must be used

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

Given $n = 4$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

But, if $n = 4$, then use minimal solution (see previous lecture)

$$\mathtt{A}\mathbf{h} = \mathbf{0}, \text{ where } \mathtt{A} = \begin{bmatrix} [\mathbf{x}'_1]^{\perp} \otimes \mathbf{x}_1^{\top} \\ [\mathbf{x}'_2]^{\perp} \otimes \mathbf{x}_2^{\top} \\ [\mathbf{x}'_3]^{\perp} \otimes \mathbf{x}_3^{\top} \\ [\mathbf{x}'_4]^{\perp} \otimes \mathbf{x}_4^{\top} \end{bmatrix}$$

The (right) null space of the matrix $\mathtt{A}$ is the vector $\mathbf{h}$.

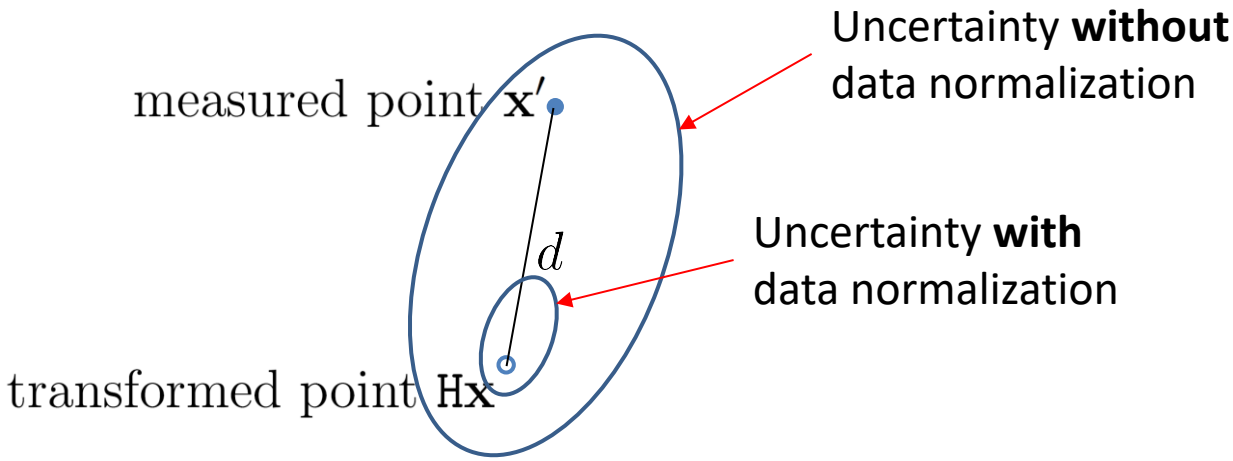
$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^{\top}$$

The solution vector $\mathbf{h}$ is the column of $\mathtt{V}$ not corresponding to a singular value (i.e., the last column of $\mathtt{V}$ (or last row of $\mathtt{V}^{\top}$)).

But, data normalization must be used

# Data normalization

- Data normalization is required to reduce the propagated uncertainty under the nonlinear projection



# Data normalizing transformations

- 2D

Determine the similarity transformation $\mathtt{T}$ such that the mean (i.e., centroid) of the transformed points is at the origin and their standard deviation from the origin is $\sqrt{2}$.

$$\mathtt{T} = \begin{bmatrix} s & 0 & -s\mu_{\tilde{x}} \\ 0 & s & -s\mu_{\tilde{y}} \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } s = \sqrt{\frac{2}{\sigma_{\tilde{x}}^2 + \sigma_{\tilde{y}}^2}}$$

where $\mu_{\tilde{x}}$ and $\sigma_{\tilde{x}}^2$, and $\mu_{\tilde{y}}$ and $\sigma_{\tilde{y}}^2$ are the mean and variance of the $\tilde{x}$ and $\tilde{y}$ coordinates, respectively.

# Linear estimation of 2D projective transformation using the direct linear transformation (DLT) algorithm

- Algorithm

1. Data normalize the points

$$\mathbf{x}_{\mathrm{DN},i} = \mathtt{T}\mathbf{x}_i \,\forall\, i$$
$$\mathbf{x}'_{\mathrm{DN},i} = \mathtt{T}'\mathbf{x}'_i \,\forall\, i$$

2. Estimate the data normalized 2D projective transformation $\mathtt{H}_{\mathrm{DN}}$ from the data normalized points

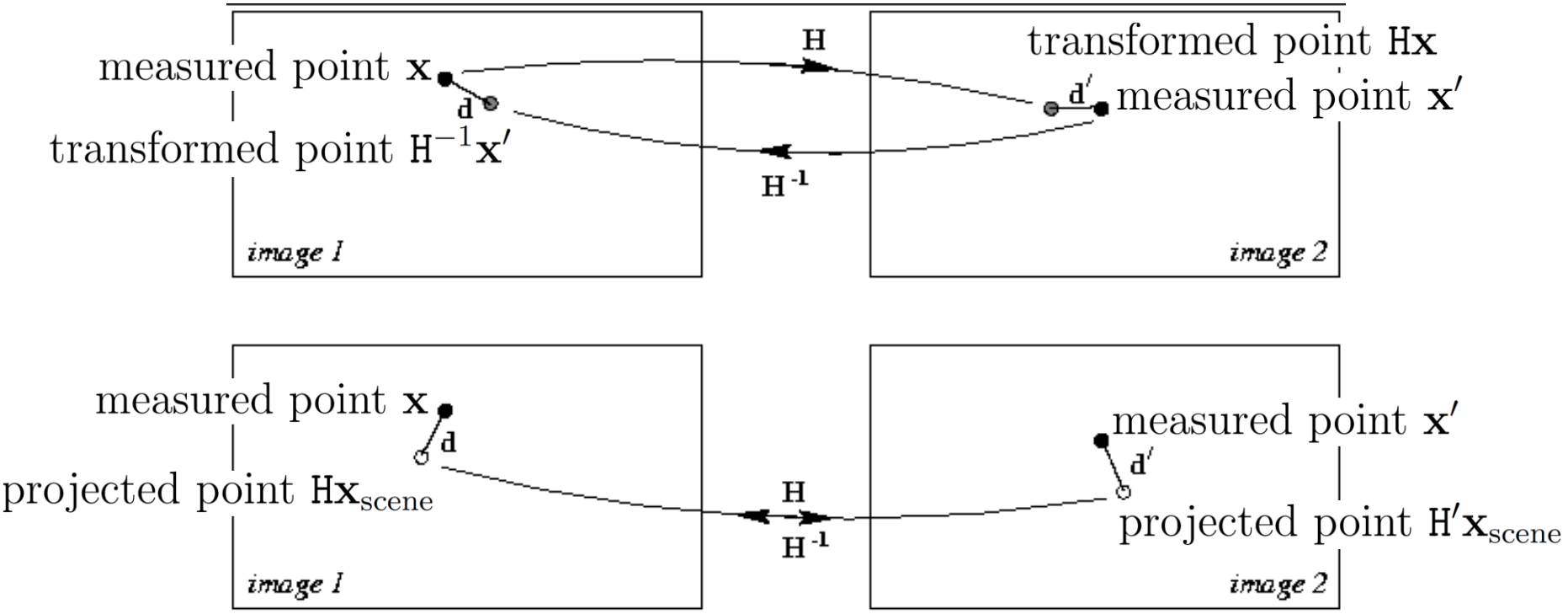
3. Data denormalize the data normalized 2D projective transformation

$$\begin{aligned}
\mathbf{x}'_{\mathrm{DN},i} &= \mathtt{H}_{\mathrm{DN}}\mathbf{x}_{\mathrm{DN},i} \,\forall\, i \\
\mathtt{T}'\mathbf{x}'_i &= \mathtt{H}_{\mathrm{DN}}\mathtt{T}\mathbf{x}_i \\
\mathbf{x}'_i &= \mathtt{T}'^{-1}\mathtt{H}_{\mathrm{DN}}\mathtt{T}\mathbf{x}_i \\
\mathbf{x}'_i &= \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ where } \mathtt{H} = \mathtt{T}'^{-1}\mathtt{H}_{\mathrm{DN}}\mathtt{T}
\end{aligned}$$

# Sampson correction and Sampson error

# 2D projective transformation

- Error in image point measurements



# Sampson cost

Sampson error uses a cost function that lies between algebraic and geometric cost functions (in terms of complexity) and gives a close approximation to geometric error.
Measurement vector $\mathbf{X} = (\tilde{x}, \tilde{y}, \tilde{x}', \tilde{y}')^\top$
$\mathcal{C}_{\mathtt{H}}(\mathbf{X})$ is the cost associated with $\mathbf{X}$

$$\mathcal{C}_{\mathtt{H}}(\mathbf{X}) = \mathbf{0}$$

$$\mathcal{C}_{\mathtt{H}}(\mathbf{X} + \boldsymbol{\delta}_{\mathbf{X}}) = \mathcal{C}_{\mathtt{H}}(\mathbf{X}) + \frac{\partial \mathcal{C}_{\mathtt{H}}(\mathbf{X})}{\partial \mathbf{X}} \boldsymbol{\delta}_{\mathbf{X}} \qquad \text{first order Taylor expansion}$$

$$\text{if } \hat{\mathbf{X}} = \mathbf{X} + \boldsymbol{\delta}_{\mathbf{X}}$$

$$\hat{\mathbf{X}} - \mathbf{X} = \boldsymbol{\delta}_{\mathbf{X}} \text{ and desire } \hat{\mathbf{X}} \text{ such that } \mathcal{C}_{\mathtt{H}}(\hat{\mathbf{X}}) = \mathbf{0}, \text{ then}$$

$$\mathcal{C}_{\mathtt{H}}(\mathbf{X}) + \frac{\partial \mathcal{C}_{\mathtt{H}}(\mathbf{X})}{\partial \mathbf{X}} \boldsymbol{\delta}_{\mathbf{X}} = \mathbf{0}$$

$$\boldsymbol{\epsilon} + \mathtt{J}\boldsymbol{\delta}_{\mathbf{X}} = \mathbf{0}, \text{ where } \boldsymbol{\epsilon} = \mathcal{C}_{\mathtt{H}}(\mathbf{X}) \text{ and } \mathtt{J} = \frac{\partial \mathcal{C}_{\mathtt{H}}(\mathbf{X})}{\partial \mathbf{X}}$$

$$\mathtt{J}\boldsymbol{\delta}_{\mathbf{X}} = -\boldsymbol{\epsilon}, \text{ solve for } \boldsymbol{\delta}_{\mathbf{X}} \text{ that minimizes } \|\boldsymbol{\delta}_{\mathbf{X}}\|$$

# Sampson corrected and Sampson error

$$\mathtt{J}\boldsymbol{\delta}_{\mathbf{X}} = -\boldsymbol{\epsilon}, \text{ solve for } \boldsymbol{\delta}_{\mathbf{X}} \text{ that minimizes } \|\boldsymbol{\delta}_{\mathbf{X}}\|$$

Use a vector $\boldsymbol{\lambda}$ of Lagrange multipliers and find extrema of $\boldsymbol{\delta}_{\mathbf{X}}^{\top}\boldsymbol{\delta}_{\mathbf{X}} - 2\boldsymbol{\lambda}^{\top}(\mathtt{J}\boldsymbol{\delta}_{\mathbf{X}} + \boldsymbol{\epsilon})$

$$\mathtt{J}\mathtt{J}^{\top}\boldsymbol{\lambda} = -\boldsymbol{\epsilon}, \text{ solve for } \boldsymbol{\lambda}$$

then substitute $\boldsymbol{\lambda}$ to solve for $\boldsymbol{\delta}_{\mathbf{X}}$

$$\boldsymbol{\delta}_{\mathbf{X}} = \mathtt{J}^{\top}\boldsymbol{\lambda}$$

| | |
|---|---|
| Sampson corrected | $\hat{\mathbf{X}} = \mathbf{X} + \boldsymbol{\delta}_{\mathbf{X}}$ |
| Squared sampson error | $\|\boldsymbol{\delta}_{\mathbf{X}}\|^2 = \boldsymbol{\delta}_{\mathbf{X}}^{\top}\boldsymbol{\delta}_{\mathbf{X}}$ |

Determine $\mathtt{J}$ and $\boldsymbol{\epsilon}$

$$\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i \,\forall\, i$$

$$\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \mathbf{0}$$

$$\begin{bmatrix} x'_i \\ y'_i \\ w'_i \end{bmatrix} \times \begin{bmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} y'_i\mathbf{h}^{3\top}\mathbf{x}_i - w'_i\mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\top}\mathbf{x}_i - y'_i\mathbf{h}^{1\top}\mathbf{x}_i \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = \mathbf{0}$$

Last row is only needed if $w'_i = 0$

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = \mathbf{0}$$

$$\mathtt{A}_i\mathbf{h} = \mathbf{0}\,\forall\, i, \text{ where } \mathtt{A}_i = \begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \end{bmatrix} \text{ and } \mathbf{h} = \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix}$$

# Sampson corrected and Sampson error

$$\mathtt{A}_i\mathbf{h} = \begin{bmatrix} \mathbf{0}^\top & -\mathbf{x}_i^\top & \tilde{y}'_i\mathbf{x}_i^\top \\ \mathbf{x}_i^\top & \mathbf{0}^\top & -\tilde{x}'_i\mathbf{x}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} \quad \forall\, i$$

$$\mathtt{A}_i\mathbf{h} = \begin{bmatrix} -\mathbf{x}_i^\top\mathbf{h}^2 + \tilde{y}'_i\mathbf{x}_i^\top\mathbf{h}^3 \\ \mathbf{x}_i^\top\mathbf{h}^1 - \tilde{x}'_i\mathbf{x}_i^\top\mathbf{h}^3 \end{bmatrix} \quad \forall\, i$$

$$\mathbf{X}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{x}'_i, \tilde{y}'_i)^\top$$

$$\boldsymbol{\epsilon}_i = \mathcal{C}_{\mathtt{H}}(\mathbf{X}_i) = \mathtt{A}_i\mathbf{h} = \begin{bmatrix} -(\tilde{x}_i h_{21} + \tilde{y}_i h_{22} + h_{23}) + \tilde{y}'_i(\tilde{x}_i h_{31} + \tilde{y}_i h_{32} + h_{33}) \\ \tilde{x}_i h_{11} + \tilde{y}_i h_{12} + h_{13} - \tilde{x}'_i(\tilde{x}_i h_{31} + \tilde{y}_i h_{32} + h_{33}) \end{bmatrix}$$

$$\mathtt{J}_i = \frac{\partial \mathcal{C}_{\mathtt{H}}(\mathbf{X}_i)}{\partial \mathbf{X}_i} = \frac{\partial(\mathtt{A}_i\mathbf{h})}{\partial(\tilde{x}_i, \tilde{y}_i, \tilde{x}'_i, \tilde{y}'_i)}$$

$$\mathtt{J}_i = \begin{bmatrix} -h_{21} + \tilde{y}'_i h_{31} & -h_{22} + \tilde{y}'_i h_{32} & 0 & \tilde{x}_i h_{31} + \tilde{y}_i h_{32} + h_{33} \\ h_{11} - \tilde{x}'_i h_{31} & h_{12} - \tilde{x}'_i h_{32} & -(\tilde{x}_i h_{31} + \tilde{y}_i h_{32} + h_{33}) & 0 \end{bmatrix}$$

$\mathtt{J}_i\mathtt{J}_i^\top\boldsymbol{\lambda}_i = -\boldsymbol{\epsilon}_i$, solve for $\boldsymbol{\lambda}_i$, then $\boldsymbol{\delta}_{\mathbf{X}_i} = \mathtt{J}_i^\top\boldsymbol{\lambda}_i$

Sampson corrected points $\quad \hat{\mathbf{X}}_i = (\hat{\tilde{x}}_i, \hat{\tilde{y}}_i, \hat{\tilde{x}}'_i, \hat{\tilde{y}}'_i)^\top = \mathbf{X}_i + \boldsymbol{\delta}_{\mathbf{X}_i}$

Squared Sampson error $\quad \|\boldsymbol{\delta}_{\mathbf{X}_i}\|^2 = \boldsymbol{\delta}_{\mathbf{X}_i}^\top\boldsymbol{\delta}_{\mathbf{X}_i} = \boldsymbol{\epsilon}_i^\top(\mathtt{J}_i\mathtt{J}_i^\top)^{-1}\boldsymbol{\epsilon}_i$

# Estimation of 2D projective transformation, nonlinear

# Nonlinear estimation using the Levenberg-Marquardt algorithm

Given

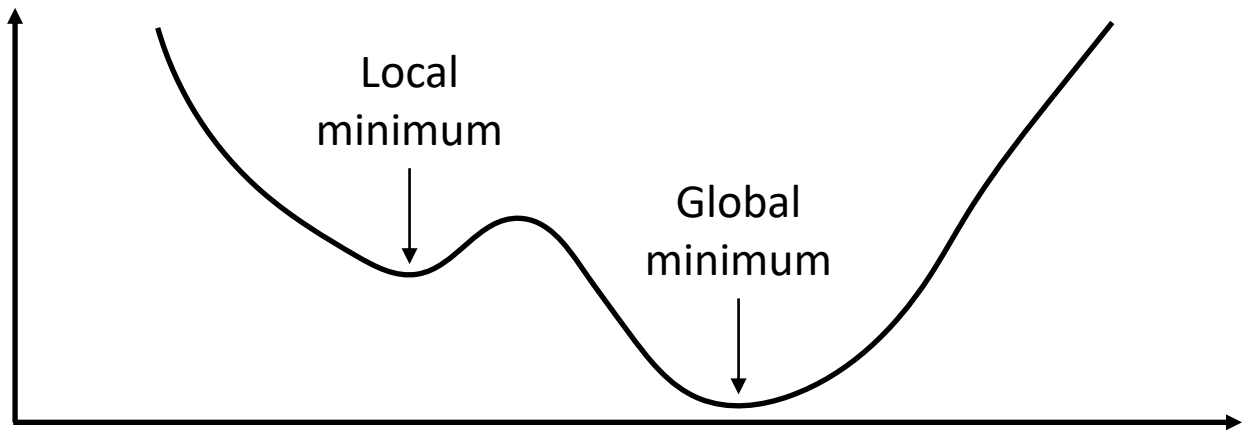
Measurement vector $\mathbf{X}$ with associated covariance matrix $\Sigma_{\mathbf{X}}$

Parameter vector $\hat{\mathbf{P}}$ (initial estimate)

Nonlinear function $f$, where $\hat{\mathbf{X}} = f(\hat{\mathbf{P}})$

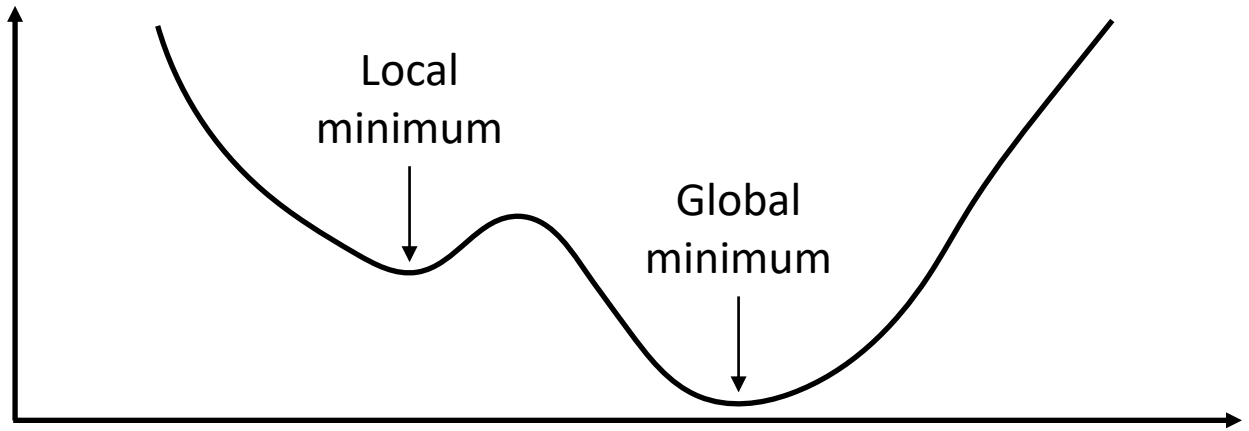
Objective

Find $\hat{\mathbf{P}}$ that minimizes $\boldsymbol{\epsilon}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = \mathbf{X} - \hat{\mathbf{X}}$



# Parameter vector

- May get stuck in local minimum of high dimensional space
- **Use minimal parameterizations towards mitigating this**



# Parameterization of a homogeneous vector

Let the homogeneous vector $\bar{\mathbf{v}} = (a, \mathbf{b}^\top)^\top \in \mathbb{R}^n$, where $\|\bar{\mathbf{v}}\| = 1$ (i.e., $\bar{\mathbf{v}}$ is a unit vector), be parameterized as

$$\mathbf{v} = \frac{2}{\operatorname{sinc}(\cos^{-1}(a))}\mathbf{b} \in \mathbb{R}^{n-1}$$

then, if $\|\mathbf{v}\| > \pi$, normalized by

$$\mathbf{v} = \left(1 - \frac{2\pi}{\|\mathbf{v}\|}\left\lceil \frac{\|\mathbf{v}\| - \pi}{2\pi} \right\rceil\right)\mathbf{v}$$

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix

# Parameterization of a homogeneous vector

The parameterized homogeneous vector $\mathbf{v}$ is deparameterized as the homogeneous vector

$$\bar{\mathbf{v}} = \left( \cos\left(\frac{\|\mathbf{v}\|}{2}\right), \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathbf{v}^\top \right)^\top \in \mathbb{R}^n$$

Unit vector with nonnegative first element

$$\bar{\mathbf{v}} = \left(a, \mathbf{b}^\top\right)^\top, \text{ where } a = \cos\left(\frac{\|\mathbf{v}\|}{2}\right) \text{ and } \mathbf{b} = \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathbf{v}$$

where $\|\bar{\mathbf{v}}\| = 1$ and $a$ is nonnegative. For the deparameterization,

$$\frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{v}} = \frac{\partial (a, \mathbf{b}^\top)}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\mathrm{d}a}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{b}}{\partial \mathbf{v}} \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$$

where

$$\frac{\mathrm{d}a}{\partial \mathbf{v}} = \begin{cases} \mathbf{0}^\top & \text{if } \|\mathbf{v}\| = 0 \\ -\frac{1}{2}\mathbf{b}^\top & \text{otherwise} \end{cases}$$

and

$$\frac{\partial \mathbf{b}}{\partial \mathbf{v}} = \begin{cases} \frac{1}{2}\mathtt{I} & \text{if } \|\mathbf{v}\| = 0 \\ \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathtt{I} + \frac{1}{4\|\mathbf{v}\|}\frac{\mathrm{d}\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{\mathrm{d}\frac{\|\mathbf{v}\|}{2}}\mathbf{v}\mathbf{v}^\top & \text{otherwise} \end{cases}$$

# Parameterization of a homogeneous vector

- Sinc function

The sinc function

$$\operatorname{sinc}(x) = \begin{cases} 1 & \text{if } x = 0 \\ \frac{\sin(x)}{x} & \text{otherwise} \end{cases}$$

The derivative is given by

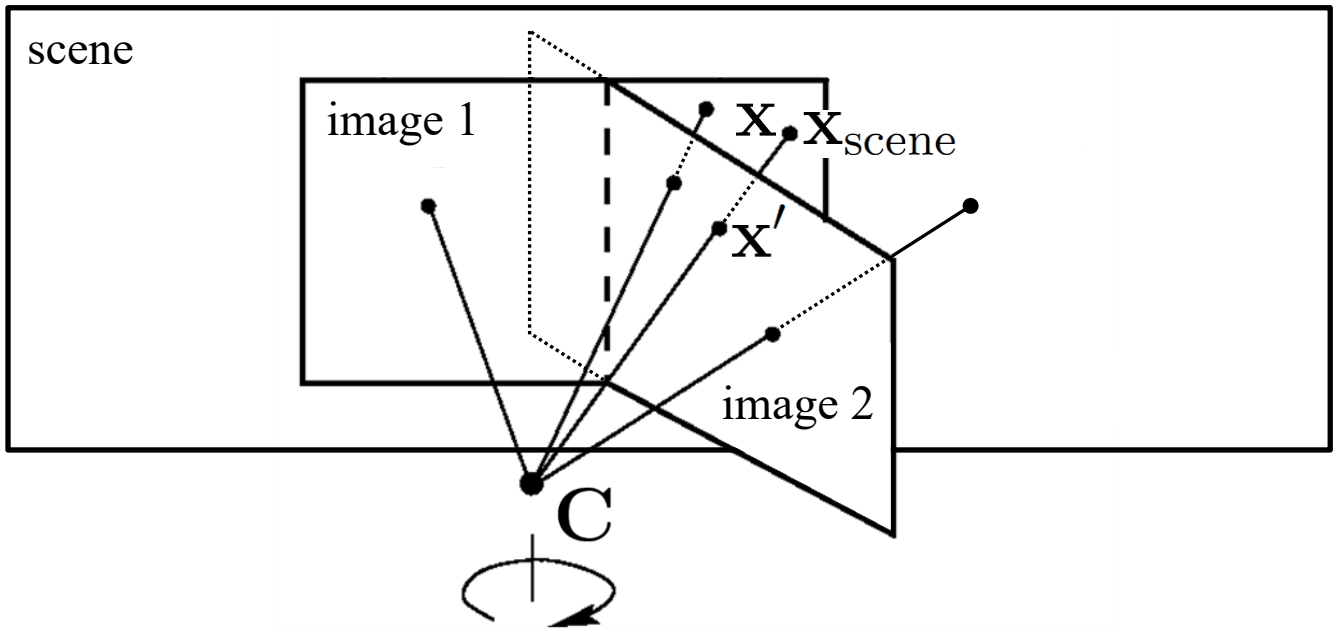
$$\frac{\mathrm{d}\operatorname{sinc}(x)}{\mathrm{d}x} = \begin{cases} 0 & \text{if } x = 0 \\ \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2} & \text{otherwise} \end{cases}$$

# Nonlinear estimation of 2D projective transformation

- Rotation about the same camera center

$$\mathbf{x}_i = \mathtt{H}\mathbf{x}_{\text{scene},i}\ \forall\, i \text{ and } \mathbf{x}'_i = \mathtt{H}'\mathbf{x}_{\text{scene},i}\ \forall\, i, \text{ set } \mathtt{H} = \mathtt{I}$$

(i.e., image 1 plane is scene plane)




Similar figure for imaging a plane

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

Parameterization of homogeneous vector

Parameter vector $(8 + 2n)$ x 1

$$(\hat{\mathbf{h}}'^{\top}, \hat{\mathbf{x}}_{\text{scene},1}^{\top}, \hat{\mathbf{x}}_{\text{scene},2}^{\top}, \ldots, \hat{\mathbf{x}}_{\text{scene},n}^{\top})^{\top}$$

Mapping

$$\hat{\mathbf{x}}_i = \mathtt{H}\hat{\mathbf{x}}_{\text{scene},i} \; \forall \, i$$

$$\hat{\mathbf{x}}'_i = \hat{\mathtt{H}}'\hat{\mathbf{x}}_{\text{scene},i} \; \forall \, i$$

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix

Measurement vector $4n$ x 1

$$\left(\tilde{\mathbf{x}}_1^{\top}, \tilde{\mathbf{x}}_2^{\top}, \ldots, \tilde{\mathbf{x}}_n^{\top}, \tilde{\mathbf{x}}_1'^{\top}, \tilde{\mathbf{x}}_2'^{\top}, \ldots, \tilde{\mathbf{x}}_n'^{\top}\right)^{\top}$$

with associated covariance matrices $\Sigma_{\tilde{\mathbf{x}}_1}, \Sigma_{\tilde{\mathbf{x}}_2}, \ldots, \Sigma_{\tilde{\mathbf{x}}_n}, \Sigma_{\tilde{\mathbf{x}}'_1}, \Sigma_{\tilde{\mathbf{x}}'_2}, \ldots, \Sigma_{\tilde{\mathbf{x}}'_n}$

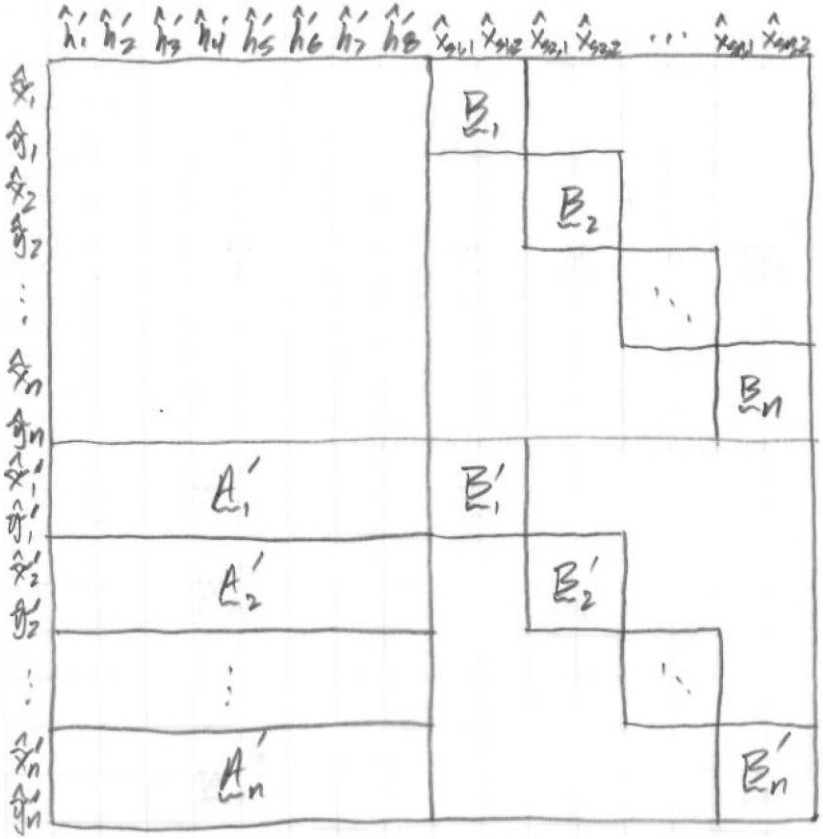
Cost

$$\sum_i \left(\boldsymbol{\epsilon}_i^{\top}\Sigma_{\tilde{\mathbf{x}}_i}^{-1}\boldsymbol{\epsilon}_i + \boldsymbol{\epsilon}_i'^{\top}\Sigma_{\tilde{\mathbf{x}}'_i}^{-1}\boldsymbol{\epsilon}'_i\right), \text{ where } \boldsymbol{\epsilon}_i = \tilde{\mathbf{x}}_i - \hat{\tilde{\mathbf{x}}}_i \text{ and } \boldsymbol{\epsilon}'_i = \tilde{\mathbf{x}}'_i - \hat{\tilde{\mathbf{x}}}'_i$$

But, use data normalization

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

- Jacobian



2 x 9 9 x 8

$$\mathrm{A}'_i = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \hat{\mathbf{h}}'} = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \bar{\mathbf{h}}'} \frac{\partial \bar{\mathbf{h}}'}{\partial \hat{\mathbf{h}}'}$$

$$\mathrm{B}_i = \frac{\partial \hat{\bar{\mathbf{x}}}_i}{\partial \hat{\mathbf{x}}_{\mathrm{scene},i}} = \frac{\partial \hat{\bar{\mathbf{x}}}_i}{\partial \bar{\mathbf{x}}_{\mathrm{scene},i}} \frac{\partial \bar{\mathbf{x}}_{\mathrm{scene},i}}{\partial \hat{\mathbf{x}}_{\mathrm{scene},i}}$$

$$\mathrm{B}'_i = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \hat{\mathbf{x}}_{\mathrm{scene},i}} = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \bar{\mathbf{x}}_{\mathrm{scene},i}} \frac{\partial \bar{\mathbf{x}}_{\mathrm{scene},i}}{\partial \hat{\mathbf{x}}_{\mathrm{scene},i}}$$

2 x 3 3 x 2

# Mapping of a point under a 2D projective transformation

The homogeneous 2D point $\mathbf{x}$ is mapped to the homogeneous 2D point $\mathbf{x}'$ under the (homogeneous) 2D projective transformation matrix $\mathtt{H}$ by

$$\mathbf{x}' = \mathtt{H}\mathbf{x}$$

Dehomogenizing $\mathbf{x}'$ results in the mapping $\mathbf{x} \mapsto \tilde{\mathbf{x}}'$. For this mapping

$$\frac{\partial \tilde{\mathbf{x}}'}{\partial \mathbf{h}} = \frac{1}{w'} \begin{bmatrix} \mathbf{x}^\top & \mathbf{0}^\top & -\tilde{x}'\mathbf{x}^\top \\ \mathbf{0}^\top & \mathbf{x}^\top & -\tilde{y}'\mathbf{x}^\top \end{bmatrix} \qquad 2 \text{ x } 9$$

and

$$\frac{\partial \tilde{\mathbf{x}}'}{\partial \mathbf{x}} = \frac{1}{w'} \begin{bmatrix} \mathbf{h}^{1\top} - \tilde{x}'\mathbf{h}^{3\top} \\ \mathbf{h}^{2\top} - \tilde{y}'\mathbf{h}^{3\top} \end{bmatrix} \qquad 2 \text{ x } 3$$

where $w' = \mathbf{h}^{3\top}\mathbf{x}$ and $\mathbf{h}^{i\top}$ is the $i$th row of $\mathtt{H}$.

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

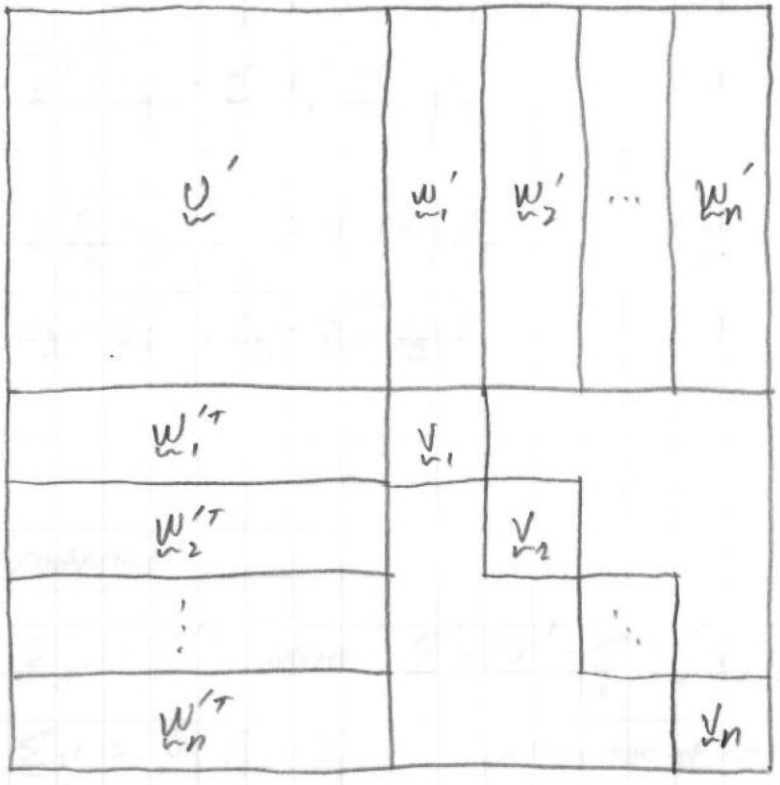
- Normal equations

$$\left(\mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \mathtt{J}\right) \boldsymbol{\delta} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$$

$$\begin{bmatrix} \mathtt{U}' & \mathtt{W}' \\ \mathtt{W}'^\top & \mathtt{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

- Normal equations matrix $\begin{bmatrix} \mathrm{U}' & \mathrm{W}' \\ \mathrm{W}'^{\top} & \mathrm{V} \end{bmatrix}$



$$\mathrm{U}' = \sum_i \mathrm{A}_i'^{\top} \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \mathrm{A}_i' \qquad 8 \times 8$$

$$\mathrm{V}_i = \mathrm{B}_i^{\top} \Sigma_{\tilde{\mathbf{x}}_i}^{-1} \mathrm{B}_i + \mathrm{B}_i'^{\top} \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \mathrm{B}_i' \qquad 2 \times 2$$

$$\mathrm{W}_i' = \mathrm{A}_i'^{\top} \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \mathrm{B}_i' \qquad 8 \times 2$$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

- Normal equations vector $\begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$

$$\left(\boldsymbol{\epsilon}_{\mathbf{a}'}^\top, \boldsymbol{\epsilon}_{\mathbf{b}_1}^\top, \boldsymbol{\epsilon}_{\mathbf{b}_2}^\top, \ldots, \boldsymbol{\epsilon}_{\mathbf{b}_n}^\top\right)^\top$$

where

$$\boldsymbol{\epsilon}_{\mathbf{a}'} = \sum_i \mathtt{A}_i'^\top \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \boldsymbol{\epsilon}_i' \qquad 8 \text{ x } 1$$

$$\boldsymbol{\epsilon}_{\mathbf{b}_i} = \mathtt{B}_i^\top \Sigma_{\tilde{\mathbf{x}}_i}^{-1} \boldsymbol{\epsilon}_i + \mathtt{B}_i'^\top \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \boldsymbol{\epsilon}_i' \qquad 2 \text{ x } 1$$

where $\boldsymbol{\epsilon}_i = \tilde{\mathbf{x}}_i - \hat{\tilde{\mathbf{x}}}_i$ and $\boldsymbol{\epsilon}_i' = \tilde{\mathbf{x}}_i' - \hat{\tilde{\mathbf{x}}}_i'$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

Augmented normal equations

$$\begin{bmatrix} \mathtt{U}'^{*} & \mathtt{W}' \\ \mathtt{W}'^{\top} & \mathtt{V}^{*} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

where $\mathtt{U}'^{*} = \mathtt{U}' + \lambda \mathtt{I}$ and $\mathtt{V}_i^{*} = \mathtt{V}_i + \lambda \mathtt{I}$

Solve for $\left(\boldsymbol{\delta}_{\mathbf{a}'}^{\top}, \boldsymbol{\delta}_{\mathbf{b}}^{\top}\right)^{\top}$ using blockwise methods

$$\begin{bmatrix} \mathtt{I} & -\mathtt{W}'\mathtt{V}^{*-1} \\ \mathtt{0} & \mathtt{I} \end{bmatrix} \begin{bmatrix} \mathtt{U}'^{*} & \mathtt{W}' \\ \mathtt{W}'^{\top} & \mathtt{V}^{*} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \mathtt{I} & -\mathtt{W}'\mathtt{V}^{*-1} \\ \mathtt{0} & \mathtt{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

$$\begin{bmatrix} \mathtt{U}'^{*} - \mathtt{W}'\mathtt{V}^{*-1}\mathtt{W}'^{\top} & \mathtt{0} \\ \mathtt{W}'^{\top} & \mathtt{V}^{*} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} - \mathtt{W}'\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

$$\begin{bmatrix} (\mathtt{U}'^{*} - \mathtt{W}'\mathtt{V}^{*-1}\mathtt{W}'^{\top})\boldsymbol{\delta}_{\mathbf{a}'} \\ \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'} + \mathtt{V}^{*}\boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} - \mathtt{W}'\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

$$\begin{bmatrix} (\mathtt{U}'^{*} - \mathtt{W}'\mathtt{V}^{*-1}\mathtt{W}'^{\top})\boldsymbol{\delta}_{\mathbf{a}'} \\ \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'} + \mathtt{V}^{*}\boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} - \mathtt{W}'\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

1. Solve for $\boldsymbol{\delta}_{\mathbf{a}'}$

$$(\mathtt{U}'^{*} - \mathtt{W}'\mathtt{V}^{*-1}\mathtt{W}'^{\top})\boldsymbol{\delta}_{\mathbf{a}'} = \boldsymbol{\epsilon}_{\mathbf{a}'} - \mathtt{W}'\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}}$$

$$\mathtt{S}'\boldsymbol{\delta}_{\mathbf{a}'} = \mathbf{e}', \text{ solve for } \boldsymbol{\delta}_{\mathbf{a}'} \quad 8 \text{ x } 1$$

where

$$\mathtt{S}' = \mathtt{U}'^{*} - \sum_i \mathtt{W}'_i \mathtt{V}_i^{*-1} \mathtt{W}_i'^{\top} \quad 8 \text{ x } 8$$

$$\mathbf{e}' = \boldsymbol{\epsilon}_{\mathbf{a}'} - \sum_i \mathtt{W}'_i \mathtt{V}_i^{*-1} \boldsymbol{\epsilon}_{\mathbf{b}_i} \quad 8 \text{ x } 1$$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

$$\begin{bmatrix} (\mathtt{U}'^{*} - \mathtt{W}'\mathtt{V}^{*-1}\mathtt{W}'^{\top})\boldsymbol{\delta}_{\mathbf{a}'} \\ \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'} + \mathtt{V}^{*}\boldsymbol{\delta}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} - \mathtt{W}'\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}} \\ \boldsymbol{\epsilon}_{\mathbf{b}} \end{bmatrix}$$

2. Solve for $\boldsymbol{\delta}_{\mathbf{b}}$

$$\begin{aligned} \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'} + \mathtt{V}^{*}\boldsymbol{\delta}_{\mathbf{b}} &= \boldsymbol{\epsilon}_{\mathbf{b}} \\ \mathtt{V}^{*}\boldsymbol{\delta}_{\mathbf{b}} &= \boldsymbol{\epsilon}_{\mathbf{b}} - \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{b}} &= \mathtt{V}^{*-1}\left(\boldsymbol{\epsilon}_{\mathbf{b}} - \mathtt{W}'^{\top}\boldsymbol{\delta}_{\mathbf{a}'}\right) \end{aligned}$$

where

$$\boldsymbol{\delta}_{\mathbf{b}_i} = \mathtt{V}_i^{*-1}(\boldsymbol{\epsilon}_{\mathbf{b}_i} - \mathtt{W}_i'^{\top}\boldsymbol{\delta}_{\mathbf{a}'}) \quad 2 \text{ x } 1$$

Candidate parameter vector

$$\hat{\mathbf{h}}_0' = \hat{\mathbf{h}}' + \boldsymbol{\delta}_{\mathbf{a}'}$$

$$\hat{\mathbf{x}}_{\text{scene},i_0} = \hat{\mathbf{x}}_{\text{scene},i} + \boldsymbol{\delta}_{\mathbf{b}_i}$$

Adjustment

Normalize parameterization of homogeneous vector after adjustment

# Data normalization with covariance propagation

Determine the similarity transformation $\mathtt{T}$ such that the mean (i.e., centroid) of the transformed points is at the origin and their standard deviation from the origin is $\sqrt{2}$.

$$\mathtt{T} = \begin{bmatrix} s & 0 & -s\mu_{\tilde{x}} \\ 0 & s & -s\mu_{\tilde{y}} \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } s = \sqrt{\frac{2}{\sigma_{\tilde{x}}^2 + \sigma_{\tilde{y}}^2}}$$

where $\mu_{\tilde{x}}$ and $\sigma_{\tilde{x}}^2$, and $\mu_{\tilde{y}}$ and $\sigma_{\tilde{y}}^2$ are the mean and variance of the $\tilde{x}$ and $\tilde{y}$ coordinates, respectively.

$$\mathbf{x}_{\text{DN}} = \mathtt{T}\mathbf{x}$$

$$\begin{bmatrix} \tilde{\mathbf{x}}_{\text{DN}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ 1 \end{bmatrix}, \text{ where } \mathtt{A} = s\mathtt{I} \text{ and } \mathbf{t} = \begin{bmatrix} -s\mu_{\tilde{x}} \\ -s\mu_{\tilde{y}} \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{x}}_{\text{DN}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{A}\tilde{\mathbf{x}} + \mathbf{t} \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{x}}_{\text{DN}} = \mathtt{A}\tilde{\mathbf{x}} + \mathbf{t}$$

If unknown, then assume to be identity

Covariance propagation

$$\Sigma_{\tilde{\mathbf{x}}_{\text{DN}}} = \mathtt{J}\Sigma_{\tilde{\mathbf{x}}}\mathtt{J}^\top, \text{ where } \mathtt{J} = \frac{\partial \tilde{\mathbf{x}}_{\text{DN}}}{\partial \tilde{\mathbf{x}}} = \mathtt{A} = s\mathtt{I}$$

# Nonlinear estimation of 2D projective transformation using the Levenberg-Marquardt algorithm

- Algorithm

1. Data normalize the image points

$$\tilde{\mathbf{x}}_{\mathrm{DN},i} = \mathtt{A}\tilde{\mathbf{x}}_i + \mathbf{t} \text{ and associated covariance matrices } \Sigma_{\tilde{\mathbf{x}}_{\mathrm{DN},i}} = s^2\Sigma_{\tilde{\mathbf{x}}_i}$$
$$\tilde{\mathbf{x}}'_{\mathrm{DN},i} = \mathtt{A}'\tilde{\mathbf{x}}'_i + \mathbf{t}' \text{ and associated covariance matrices } \Sigma_{\tilde{\mathbf{x}}'_{\mathrm{DN},i}} = s'^2\Sigma_{\tilde{\mathbf{x}}'_i}$$

   and 2D projective transformation

$$\mathtt{H}'_{\mathrm{DN}} = \mathtt{T}'\mathtt{H}'\mathtt{T}^{-1}$$

2. Initialize the scene points as the Sampson corrected data normalized points in the first image

3. Estimate the data normalized 2D projective transformation $\mathtt{H}'_{\mathrm{DN}}$ and data normalized scene points from the data normalized image points

4. Data denormalize the final estimate of the data normalized 2D projective transformation

$$\mathtt{H}' = \mathtt{T}'^{-1}\mathtt{H}'_{\mathrm{DN}}\mathtt{T}$$

# Next lecture

- Camera rotation and translation, two views
- Estimation of fundamental matrix, linear
- Reading
  - Sections 9.1, 9.2, 11.1, 12.4, 11.2, and 11.3