Three View Geometry

Computer Vision II

CSE 252B

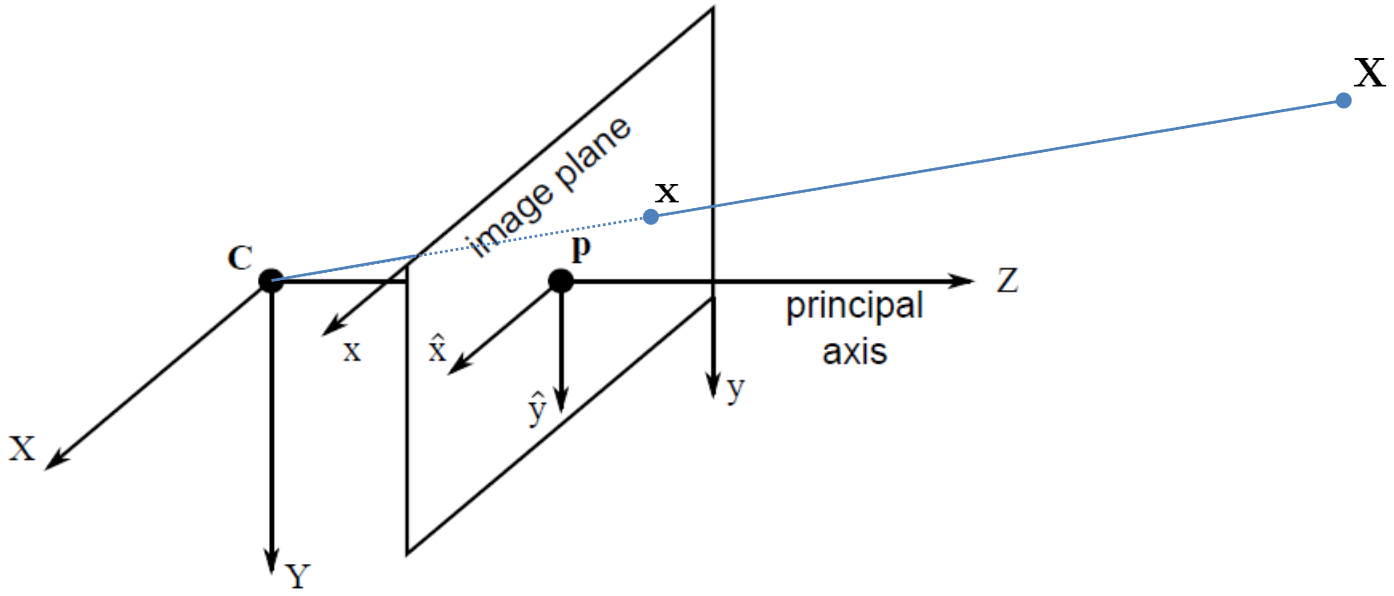
Lecture 16

Announcements

- Assignment 4 is due today, 11:59 PM
- Assignment 5 will be released today
  - Due Mar 18, 11:59 PM
- Reading
  - Chapters 15 and 16

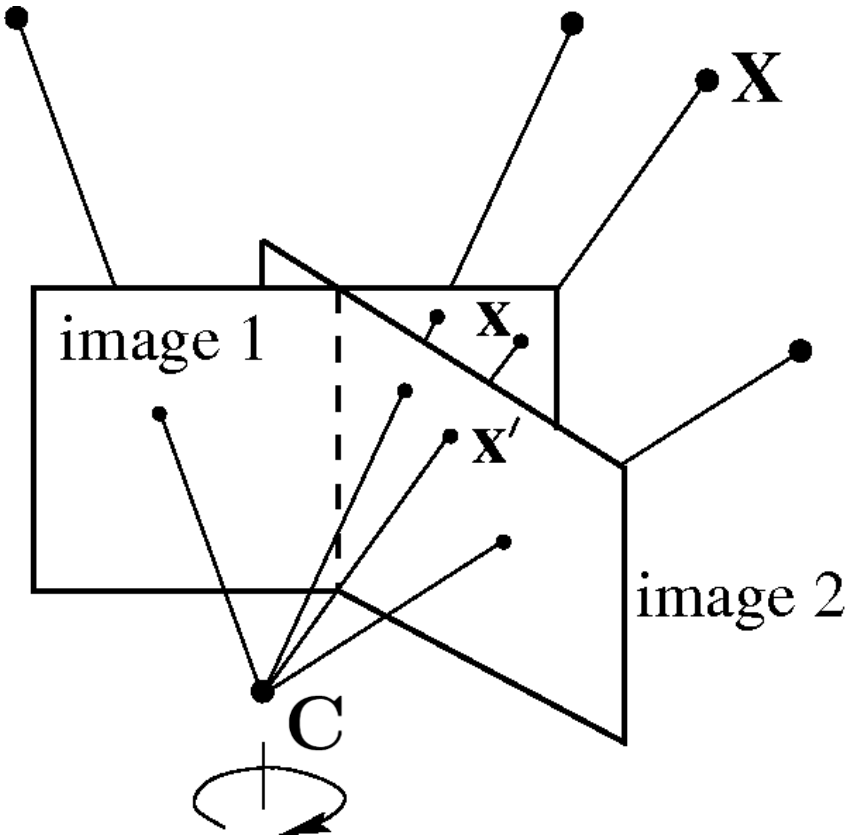
Single view geometry

- Imaging a scene



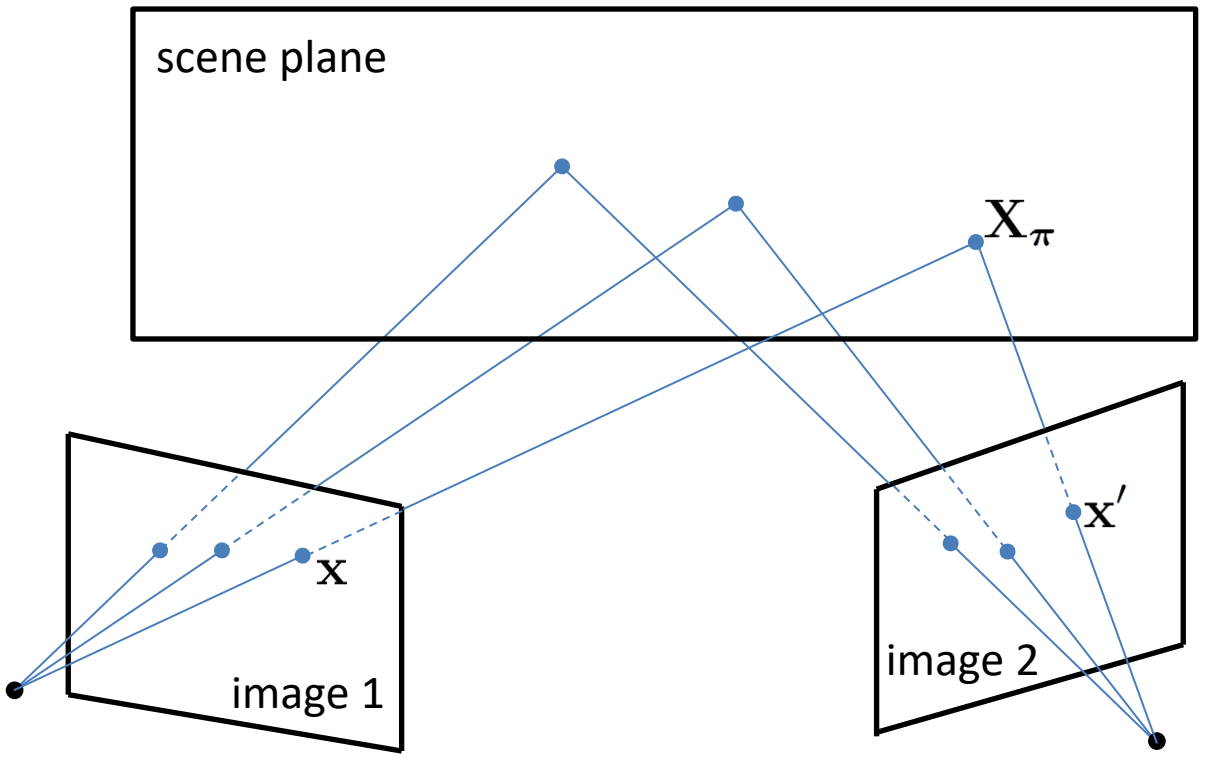
Two view geometry

- Rotation about the same camera center



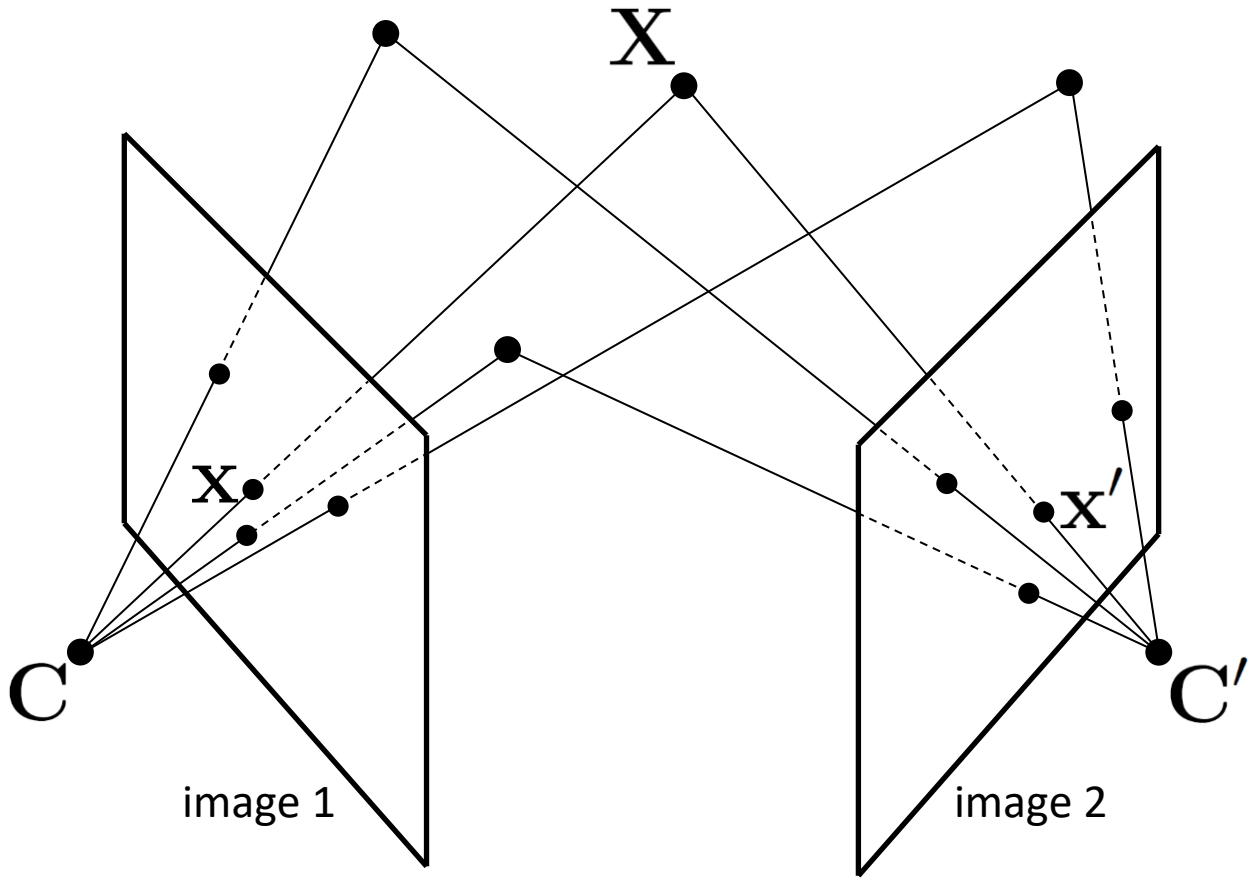
Two view geometry

- Imaging a plane



Two view geometry

- Imaging a 3D scene



Imaging geometry models

- Single view
  - Uncalibrated: camera projection matrix
    - 12 elements, 11 degrees of freedom
  - Calibrated: normalized camera projection matrix
    - 12 elements, 6 degrees of freedom
- Two views
  - Rotation about the same camera center
    - Uncalibrated: 2D projective transformation matrix
      - 9 elements, 8 degrees of freedom
    - Calibrated: 3D rotation matrix
      - 9 elements, 3 degrees of freedom

Imaging geometry models

- Two views
  - Imaging a plane
    - Uncalibrated: 2D projective transformation matrix
      - 9 elements, 8 degrees of freedom
    - Calibrated: 2D projective transformation matrix
      - 9 elements, 8 degrees of freedom
  - Imaging a 3D scene
    - Uncalibrated: fundamental matrix
      - 9 elements, 7 degrees of freedom
    - Calibrated: essential matrix
      - 9 elements, 5 degrees of freedom

Imaging geometry models

- Three views
  - Uncalibrated: trifocal tensor
    - 27 elements, 18 degrees of freedom
  - Calibrated: calibrated trifocal tensor
    - 27 elements, 11 degrees of freedom

Not covered in CSE 252B

Tensors

- Notation

A tensor with rank $r + s$ may be of mixed type (r, s), consisting of r contravariant (upper) indices and s covariant (lower) indices.

(Scalars are) zeroth-rank tensors	s
(Vectors are) first-rank tensors	v_i or v^i
Second-rank tensors	a_{ij}, a^{ij}, or a^j_i
Third-rank tensors	$b_{ijk}, b^k_{ij}, b^{jk}_i$, or b^{ijk}
	$\vdots$

- Matrix as a tensor

The matrix `A` where a_{ij} is the element at ith row and jth column of `A` corresponds to the second-rank tensor a^i_j with one contravariant (upper) index and one covariant (lower) index.

Tensors

- Contraction

Einstein summation is the convention that repeated indices are implicitly summed over. For tensors, the sum over a repeated upper and lower index in a product is a contraction, e.g.,

$$a = b_i c^i \qquad a = \sum_i b_i c^i \qquad a = \mathbf{b}^\top \mathbf{c}$$

$$a^i = b^i_j c^j \qquad a^i = \sum_j b^i_j c^j \qquad \mathbf{a} = \mathtt{B}\mathbf{c}$$

$$a_i = b^j_i c_j \qquad a_i = \sum_j b^j_i c_j \qquad \mathbf{a} = \mathtt{B}^\top \mathbf{c}$$

$$a^j_i = b_k c^{jk}_i \qquad a^j_i = \sum_k b_k c^{jk}_i$$

Unlike matrices and vectors, when multiplying tensors, the order of the entries does not matter, e.g., $a^j_i = b_k c^{jk}_i = c^{jk}_i b_k$

Tensors

- Points and hyperplanes as tensors

A point as a (first-rank) tensor has a contravariant (upper) index. A hyperplane as a (first-rank) tensor has a covariant (lower) index, e.g., 2D points and 2D lines in three images as tensors are

Point in image 1	$x^i = (x^1, x^2, x^3)$	Line in image 1	$\ell_i = (\ell_1, \ell_2, \ell_3)$
Point in image 2	$x'^j = (x'^1, x'^2, x'^3)$	Line in image 2	$\ell'_j = (\ell'_1, \ell'_2, \ell'_3)$
Point in image 3	$x''^k = (x''^1, x''^2, x''^3)$	Line in image 3	$\ell''_k = (\ell''_1, \ell''_2, \ell''_3)$

Tensors

- 2D points and 2D lines in skew symmetric matrix form as tensors

The skew symmetric matrix form of a 2D point or 2D line corresponds to the (second-rank) tensor resulting from the tensor product of the 2D point or 2D line with a $3 \times 3 \times 3$ permutation tensor $\epsilon_{ijk} = \epsilon^{ijk}$, e.g.,

Line in image 1	$\ell_r \epsilon^{ris}$	$[\boldsymbol{\ell}]_\times$
Point in image 2	$x'^j \epsilon_{jpr}$	$[\mathbf{x}']_\times$
Point in image 3	$x''^k \epsilon_{kqs}$	$[\mathbf{x}'']_\times$

where

$$\epsilon_{ijk} = \epsilon^{ijk} = \begin{cases} 0 & \text{for } i = j, j = k, \text{ or } i = k \\ 1 & \text{for } (i,j,k) \in \{(1,2,3),(2,3,1),(3,1,2)\} \\ -1 & \text{for } (i,j,k) \in \{(1,3,2),(3,2,1),(2,1,3)\} \end{cases}$$

Tensors

- 2D projective transform as a contraction

A 2D projective transformation as a (second-rank) tensor maps points and lines (first-rank tensors) as a contraction, e.g.,

Map point in image 1 to point in image 2	$x'^j = x^i h_i^j$	$\mathbf{x}' = \mathtt{H}_{1,2}\mathbf{x}$
Map point in image 1 to point in image 3	$x''^k = x^i h_i^k$	$\mathbf{x}'' = \mathtt{H}_{1,3}\mathbf{x}$
Map line in image 2 to line in image 1	$\ell_i = \ell'_j h_i^j$	$\boldsymbol{\ell} = \mathtt{H}_{1,2}^\top \boldsymbol{\ell}'$
Map line in image 3 to line in image 1	$\ell_i = \ell''_k h_i^k$	$\boldsymbol{\ell} = \mathtt{H}_{1,3}^\top \boldsymbol{\ell}''$

Trifocal tensor

- The trifocal tensor embodies the projective geometry of three views
- Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  - A 2D projective transformation maps a point to a point
  - A 2D projective transformation maps a line to a line

Trifocal tensor

- The trifocal tensor embodies the projective geometry of three views
- Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  - A 2D projective transformation maps a line to a line

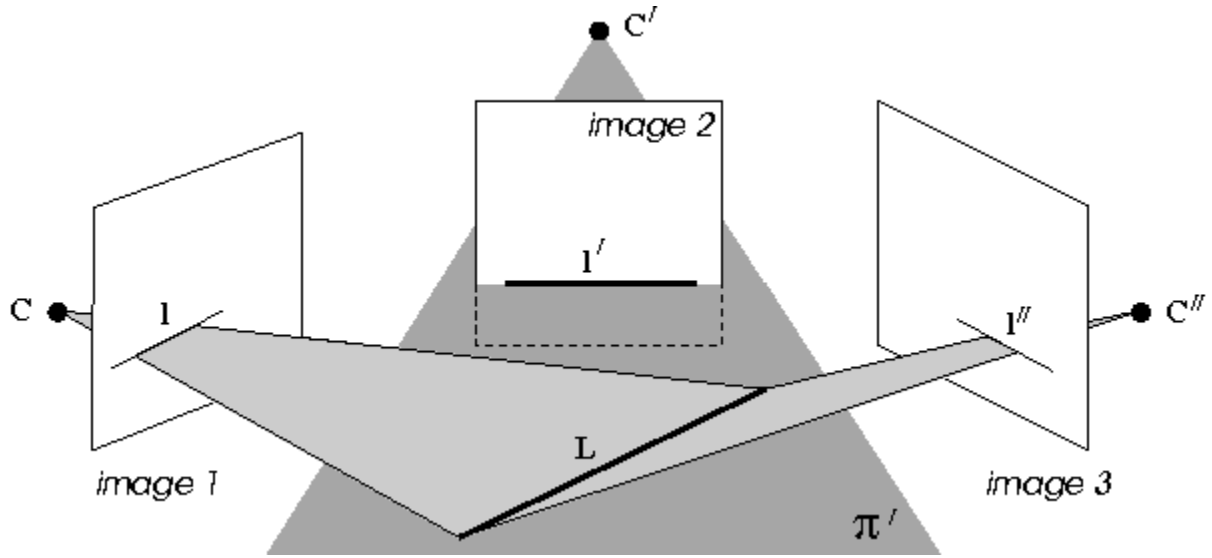
The trifocal tensor $\mathcal{T}_i^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

$$\ell_i = \ell'_j \ell''_k \mathcal{T}_i^{jk}$$

$$\ell_i = \ell''_k h_i^k, \text{ where } h_i^k = \ell'_j \mathcal{T}_i^{jk}$$

Trifocal tensor

- The 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view



$$\ell_i = \ell''_k h^k_i, \text{ where } h^k_i = \ell'_j \mathcal{T}_i^{jk}$$

Induced by a plane backprojected from a line in image 2

Trifocal tensor

- The trifocal tensor embodies the projective geometry of three views
- Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  - A 2D projective transformation maps a point to a point

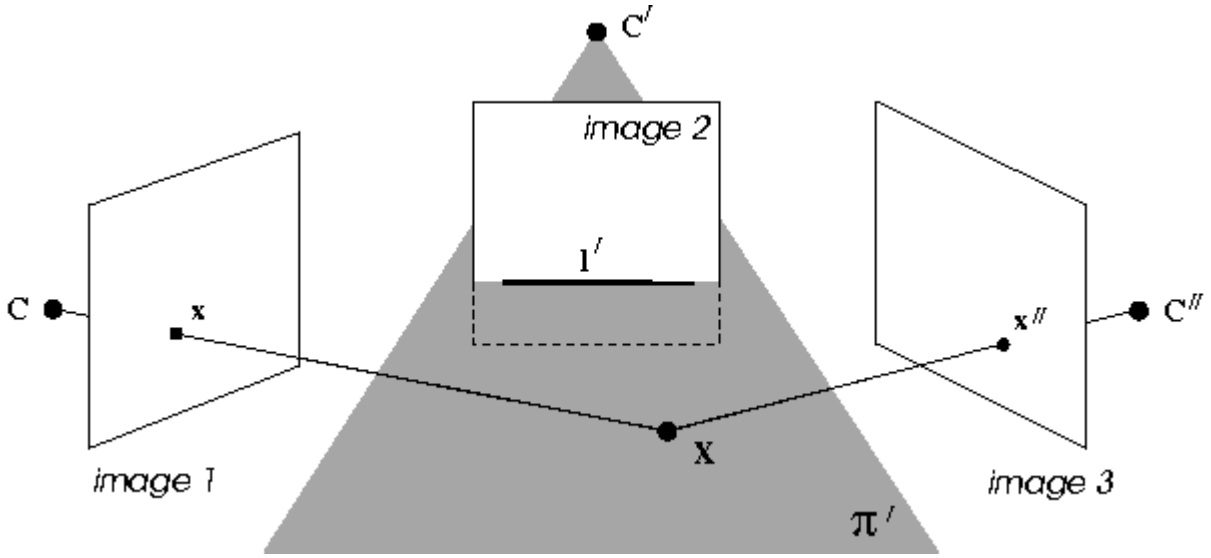
The trifocal tensor $\mathcal{T}_i^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

$$x''^k = x^i \ell'_j \mathcal{T}_i^{jk}$$

$$x''^k = x^i h_i^k, \text{ where } h_i^k = \ell'_j \mathcal{T}_i^{jk}$$

Trifocal tensor

- The 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view



$$x''^k = x^i h_i^k, \text{ where } h_i^k = \ell'_j \mathcal{T}_i^{jk}$$

Induced by a plane backprojected from a line in image 2

Trifocal tensor

- The 2D projective transformation induced by a plane backprojected from a line in image 2 maps a point in image 1 on the line in image 2 the same

$$\ell_i = \ell'_j \ell''_k \mathcal{T}_i^{jk} \text{ and } x^i \ell_i = 0$$

$$x^i \ell'_j \ell''_k \mathcal{T}_i^{jk} = 0 \quad \text{Trilinearity}$$

Trifocal tensor

- The trifocal tensor embodies the projective geometry of three views
- Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view

The trifocal tensor $\mathcal{T}_i^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

$$x'^j = x^i \ell''_k \mathcal{T}_i^{jk} \qquad x''^k = x^i \ell'_j \mathcal{T}_i^{jk}$$

$$x'^j = x^i h_i^j, \text{ where } h_i^j = \ell''_k \mathcal{T}_i^{jk} \qquad x''^k = x^i h_i^k, \text{ where } h_i^k = \ell'_j \mathcal{T}_i^{jk}$$

$$\ell_i = \ell'_j \ell''_k \mathcal{T}_i^{jk} \qquad \ell_i = \ell'_j \ell''_k \mathcal{T}_i^{jk}$$

$$\ell_i = \ell'_j h_i^j, \text{ where } h_i^j = \ell''_k \mathcal{T}_i^{jk} \qquad \ell_i = \ell''_k h_i^k, \text{ where } h_i^k = \ell'_j \mathcal{T}_i^{jk}$$

Trifocal tensor

- The trilinearities, i.e., incidence relations

Line-line-line correspondence	$\ell_r \epsilon^{ris} \ell'_j \ell''_k \mathcal{T}_i^{jk} = 0^s$
Point-line-line correspondence	$x^i \ell'_j \ell''_k \mathcal{T}_i^{jk} = 0$
(Point-line-point correspondence	$x^i \ell'_j x''^k \epsilon_{kqs} \mathcal{T}_i^{jq} = 0_s$)
Point-point-line correspondence	$x^i x'^j \epsilon_{jpr} \ell''_k \mathcal{T}_i^{pk} = 0_r$
Point-point-point correspondence	$x^i x'^j \epsilon_{jpr} x''^k \epsilon_{kqs} \mathcal{T}_i^{pq} = 0_{rs}$

  - Remember

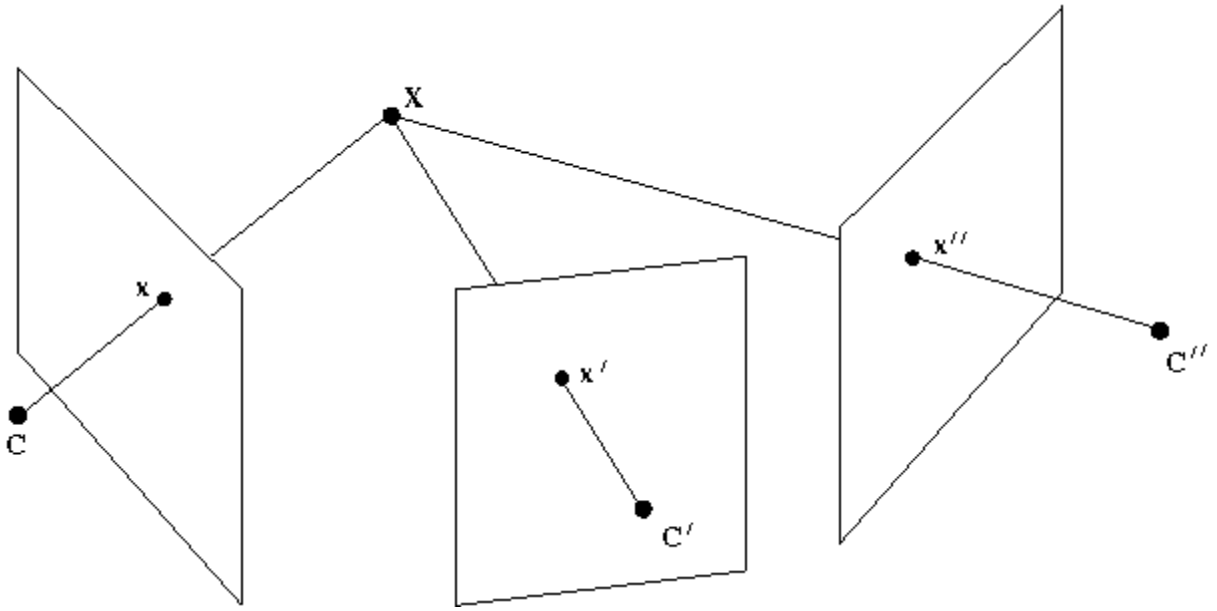
Line in image 1	$\ell_r \epsilon^{ris}$	$[\boldsymbol{\ell}]_\times$
Point in image 2	$x'^j \epsilon_{jpr}$	$[\mathbf{x}']_\times$
Point in image 3	$x''^k \epsilon_{kqs}$	$[\mathbf{x}'']_\times$

where

$$\epsilon_{ijk} = \epsilon^{ijk} = \begin{cases} 0 & \text{for } i=j, j=k, \text{ or } i=k \\ 1 & \text{for } (i,j,k) \in \{(1,2,3),(2,3,1),(3,1,2)\} \\ -1 & \text{for } (i,j,k) \in \{(1,3,2),(3,2,1),(2,1,3)\} \end{cases}$$

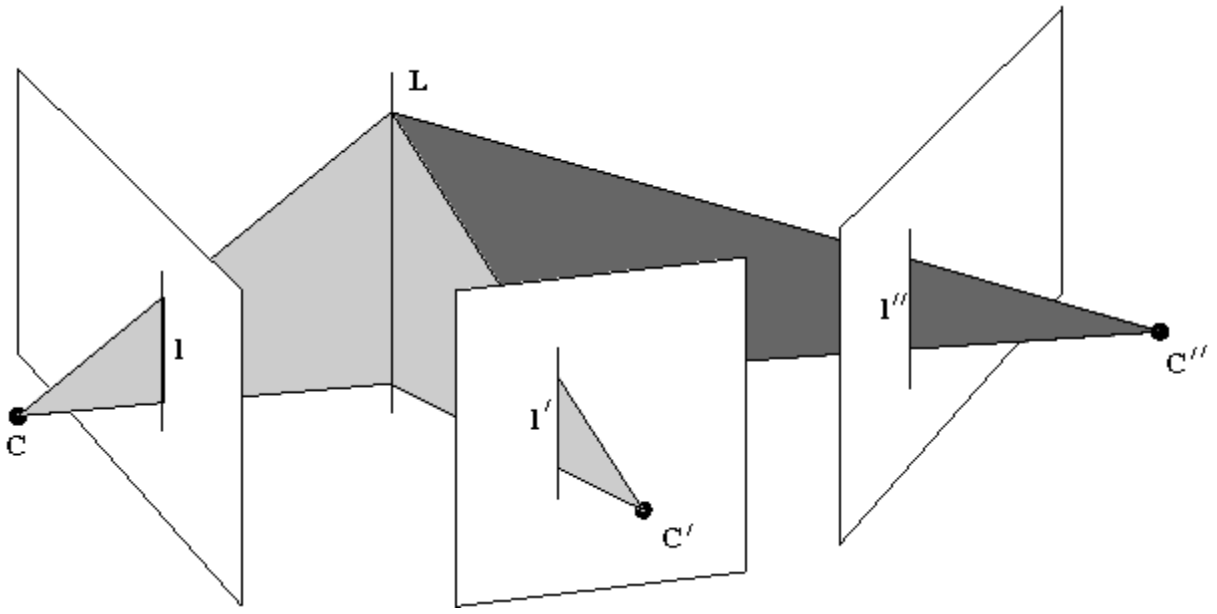
Transfer using the trifocal tensor

- Point-Point-Point



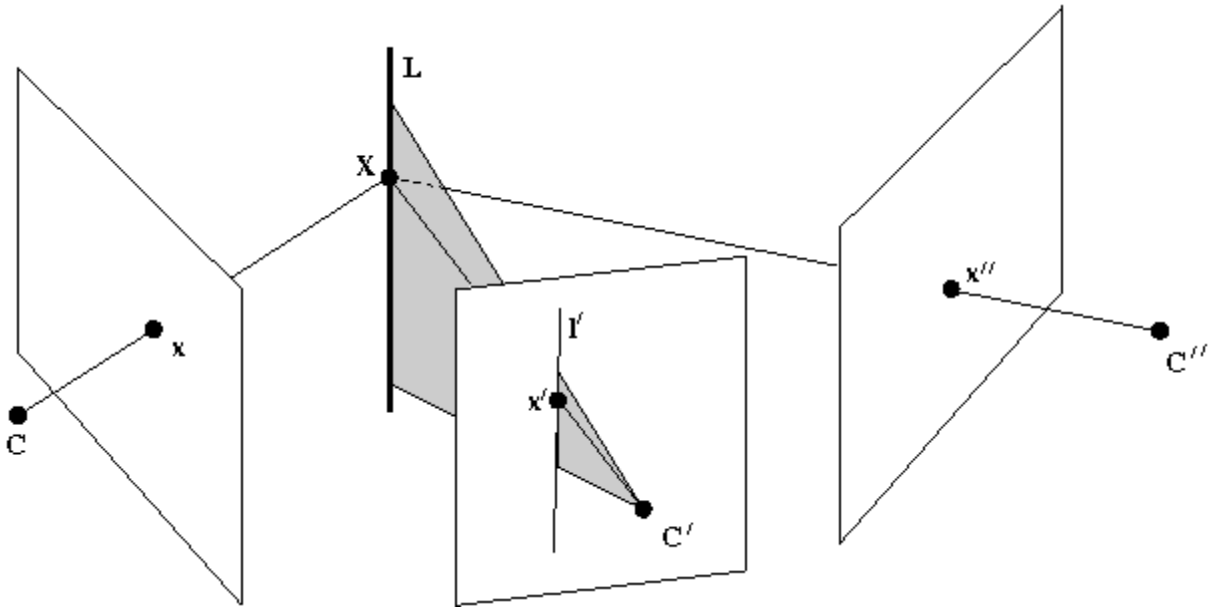
Transfer using the trifocal tensor

- Line-Line-Line



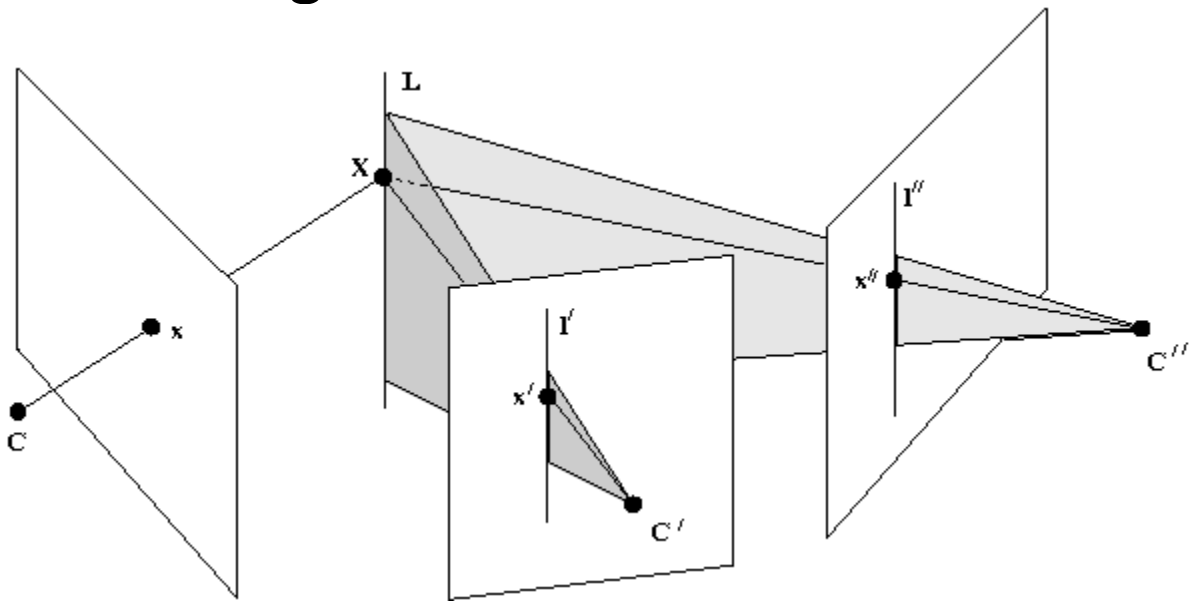
Additional trilinearity constraints

- Point-Line-Point
  - Note: line in image 2 must pass through corresponding point in image 2



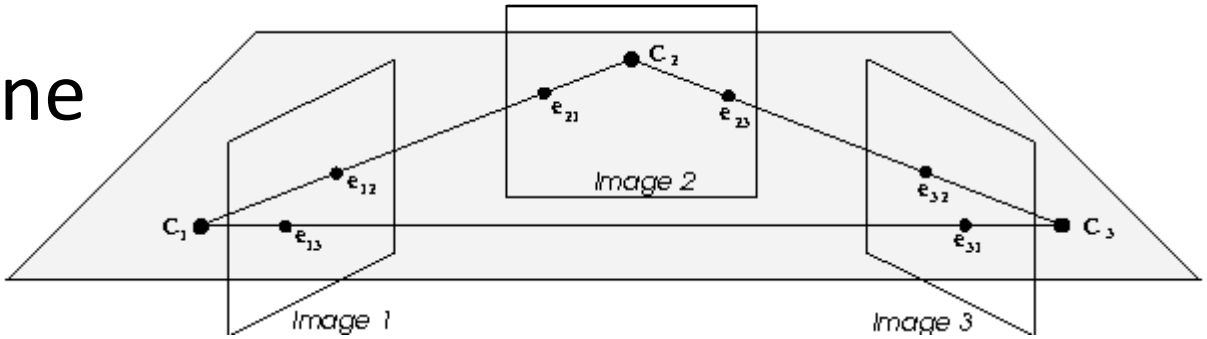
Additional trilinearity constraints

- Point-Line-Line
  - Note: lines in images 2 and 3 do not need to correspond, but must pass through corresponding points in images 2 and 3



Trifocal tensor

- Trifocal plane



- Epipoles

The trifocal tensor $\mathcal{T}_i^{jk}$ can also be represented as a set of three matrices $\{\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3\}$, where

$$\mathtt{T}_1 = \begin{bmatrix} \mathcal{T}_1^{11} & \mathcal{T}_1^{12} & \mathcal{T}_1^{13} \\ \mathcal{T}_1^{21} & \mathcal{T}_1^{22} & \mathcal{T}_1^{23} \\ \mathcal{T}_1^{31} & \mathcal{T}_1^{32} & \mathcal{T}_1^{33} \end{bmatrix}, \mathtt{T}_2 = \begin{bmatrix} \mathcal{T}_2^{11} & \mathcal{T}_2^{12} & \mathcal{T}_2^{13} \\ \mathcal{T}_2^{21} & \mathcal{T}_2^{22} & \mathcal{T}_2^{23} \\ \mathcal{T}_2^{31} & \mathcal{T}_2^{32} & \mathcal{T}_2^{33} \end{bmatrix}, \text{ and } \mathtt{T}_3 = \begin{bmatrix} \mathcal{T}_3^{11} & \mathcal{T}_3^{12} & \mathcal{T}_3^{13} \\ \mathcal{T}_3^{21} & \mathcal{T}_3^{22} & \mathcal{T}_3^{23} \\ \mathcal{T}_3^{31} & \mathcal{T}_3^{32} & \mathcal{T}_3^{33} \end{bmatrix}$$

Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the left and right null vectors, respectively, of $\mathtt{T}_i$ (i.e., $\mathbf{u}_i^\top \mathtt{T}_i = \mathbf{0}^\top$ and $\mathtt{T}_i \mathbf{v}_i = \mathbf{0}$). The epipoles $\mathbf{e}_{21}$ and $\mathbf{e}_{31}$ are calculated as follows.

$$\begin{bmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \\ \mathbf{u}_3^\top \end{bmatrix} \mathbf{e}_{21} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{bmatrix} \mathbf{e}_{31} = \mathbf{0}$$

Trifocal tensor

The trifocal tensor $\mathcal{T}_i^{jk}$ from the canonical cameras $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}' = \mathtt{B}$, and $\mathtt{P}'' = \mathtt{C}$

$$\mathcal{T}_i^{jk} = b_i^j c_4^k - b_4^j c_i^k$$

Given general cameras

$$\mathtt{P} = \begin{bmatrix} \mathbf{a}^{1\top} \\ \mathbf{a}^{2\top} \\ \mathbf{a}^{3\top} \end{bmatrix}, \mathtt{P}' = \begin{bmatrix} \mathbf{b}^{1\top} \\ \mathbf{b}^{2\top} \\ \mathbf{b}^{3\top} \end{bmatrix}, \text{ and } \mathtt{P}'' = \begin{bmatrix} \mathbf{c}^{1\top} \\ \mathbf{c}^{2\top} \\ \mathbf{c}^{3\top} \end{bmatrix}$$

where $\mathbf{a}^{i\top}$, $\mathbf{b}^{j\top}$, and $\mathbf{c}^{k\top}$ are the ith, jth, and kth rows of $\mathtt{P}$, $\mathtt{P}'$, and $\mathtt{P}''$, respectively, the trifocal tensor $\mathcal{T}_i^{jk}$ is given by

$$\mathcal{T}_i^{jk} = (-1)^{i+1} \det \left(\begin{bmatrix} \sim\mathbf{a}^{i\top} \\ \mathbf{b}^{j\top} \\ \mathbf{c}^{k\top} \end{bmatrix} \right)$$

where $\sim\mathbf{a}^{i\top}$ is the matrix $\mathtt{P}$ with the ith row omitted

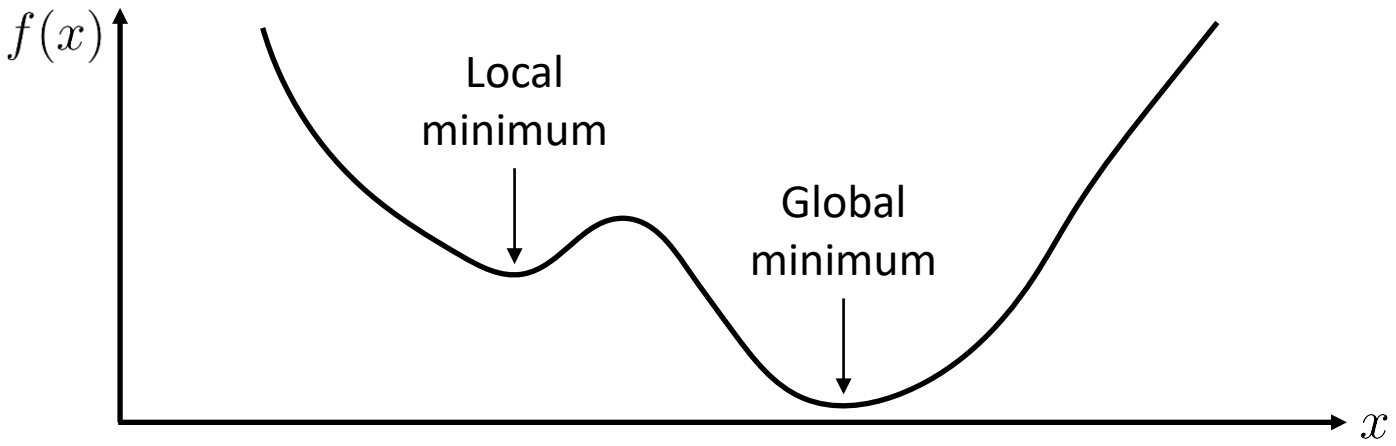
Trifocal tensor

- Error in image point and line measurements
  - Transfer and trilinearity constraints
  - Measured points and lines, and projected points and lines

Estimation of trifocal tensor

$\mathbf{x}_i = \mathtt{P}\mathbf{X}_i \,\forall\, i, \mathbf{x}'_i = \mathtt{P}'\mathbf{X}_i \,\forall\, i$, and $\mathbf{x}''_i = \mathtt{P}''\mathbf{X}_i \,\forall\, i$, set $\mathtt{P} = [\mathtt{I} \,|\, \mathbf{0}]$ and solve for $\mathtt{P}'$ and $\mathtt{P}''$

- Given point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$
- Minimize geometric error $\sum_i \left(d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 + d(\mathbf{x}'_i, \mathtt{P}'\mathbf{X}_i)^2 + d(\mathbf{x}''_i, \mathtt{P}''\mathbf{X}_i)^2 \right)$
- 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



Outlier rejection

- Even the presence of a single outlier may result in an inaccurate estimate
- Linear estimate
  - Minimizes sum of squared error and, in general, outliers have large error relative to inliers
  - Resulting estimate may not be near global minimum
- Nonlinear estimate
  - Cost function minimizes sum of squared error, so may not converge to accurate solution
    - Alternatively, use a robust cost function

Trifocal tensor, estimation from minimum number of correspondences

- 6 image point correspondences between three images
  - The last 4 image points must be in general position
    - 6 choose 4 = 15 combinations (rearrange, if needed)
- Carlsson-Weinshall duality (interchange camera and points)
  - Dualize first 2 image points
    - Results in 3 dual point correspondences
    - Additional 4 dual point correspondences are 2D projective basis
  - Calculate reduced dual fundamental matrix (5 degrees of freedom) from 7 dual point correspondences
    - 1 or 3 solutions
- Solve for three camera projection matrices
  - 5 3D points are 3D projective basis
  - 1 or 3 solutions for 6th 3D point using dual fundamental matrix
- Calculate trifocal tensor from three camera projection matrices
  - 1 or 3 solutions

See textbook for details

Random sample consensus (RANSAC) and M-estimator sample consensus (MSAC)

Objective is to determine the consensus set with the minimum cost

tol is the tolerance for establishing datum/model compatibility
τ_{cost} is the upper bound on the cost of an acceptable consensus set
maxTrials is the maximum number of attempts to find a consensus set

$\text{consensus}_{\text{cost}} = \infty$
for (trials $= 0$; trials $<$ maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)
    Select a random sample of unique data points
    Calculate the model using the random sample
    Calculate the error for each data point using model
    Calculate the cost
    if cost $< \text{consensus}_{\text{cost}}$
        $\text{consensus}_{\text{cost}} = \text{cost}$
        $\text{consensus}_{\text{model}} = \text{model}$
Calculate the error for each data point using $\text{consensus}_{\text{model}}$
Calculate the set of inliers (i.e., data points with error $\leq$ tol)

If multiple solutions, try each one, but it is still a single trial

Outlier rejection

- RANSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? 0 : 1
```

- MSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? error[n] : tol
```

RANSAC and MSAC

- Maximum number of trials
  - Adaptive maximum number of trials
- The tolerance for establishing datum/model compatibility

RANSAC and MSAC

- Maximum number of trials

The smaller the sample size, the smaller maximum number of trials. Use minimum solution!

Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Table 4.3. *The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers, ϵ.*

Adaptive maximum number of trials

s is the sample size

p is the assumed probability that at least one of the random samples does not contain any outliers

tol is the tolerance for establishing datum/model compatibility

τ_{cost} is the upper bound on the cost of an acceptable consensus set

$\text{maxTrials} = \infty$

$\text{consensus}_{\text{cost}} = \infty$

for (trials = 0; trials < maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)

- Select a random sample of unique data points
- Calculate the model using the random sample
- Calculate the error for each data point using model
- Calculate the cost
- if cost < $\text{consensus}_{\text{cost}}$
  - $\text{consensus}_{\text{cost}} = \text{cost}$
  - $\text{consensus}_{\text{model}} = \text{model}$
  - Calculate the number of inliers
  - $w = \dfrac{\text{number of inliers}}{\text{total number of data points}}$
  - $\text{maxTrials} = \dfrac{\log(1-p)}{\log(1-w^s)}$

If multiple solutions, try each one, but it is still a single trial

Calculate the error for each data point using $\text{consensus}_{\text{model}}$

Calculate the set of inliers (i.e., data points with error $\leq$ tol)

RANSAC and MSAC

- The tolerance for establishing datum/model compatibility

Probability α that a data point is an inlier (usually α is chosen to be 0.95)
Variance σ^2 of the measurement error (if unkonwn, assumed to be 1)
Codimension m
Tolerance is square distance $t^2 = F_m^{-1}(\alpha)\sigma^2$, where $F_m^{-1}(\alpha)$ is the inverse chi-squared cumulative distribution function with m degress of freedom at the probability α

- Codimension

If w is a subspace of a finite-dimensional vector space v, then $\text{codim}(w) = \dim(v) - \dim(w)$
The trifocal tensor is a variety of dimension 3 in $(\tilde{x}, \tilde{y}, \tilde{x}', \tilde{y}', \tilde{x}'', \tilde{y}'') \in \mathbb{R}^6$, so has codimension 3

Use the square Sampson error

See textbook for details

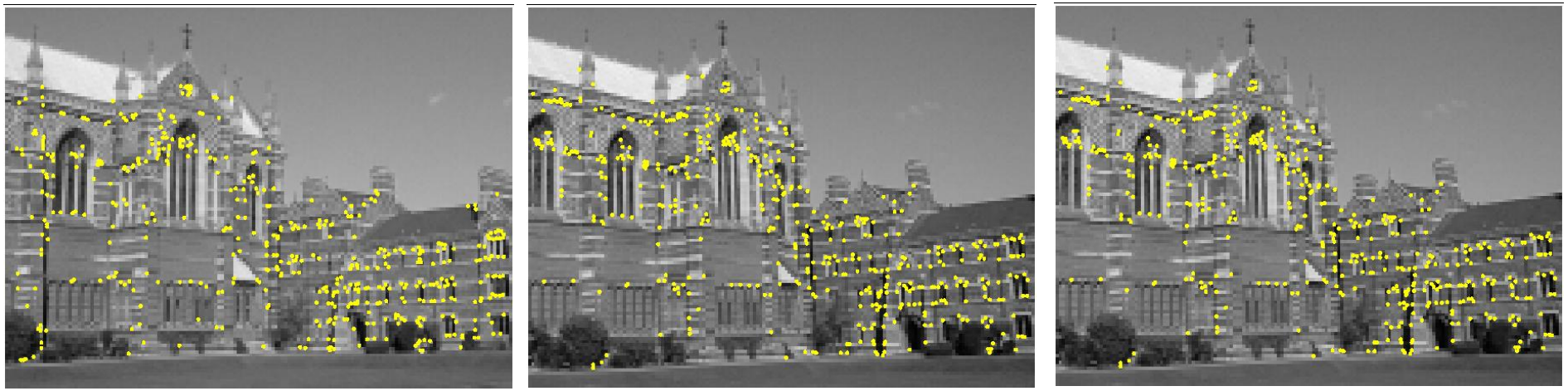
Feature detection and matching





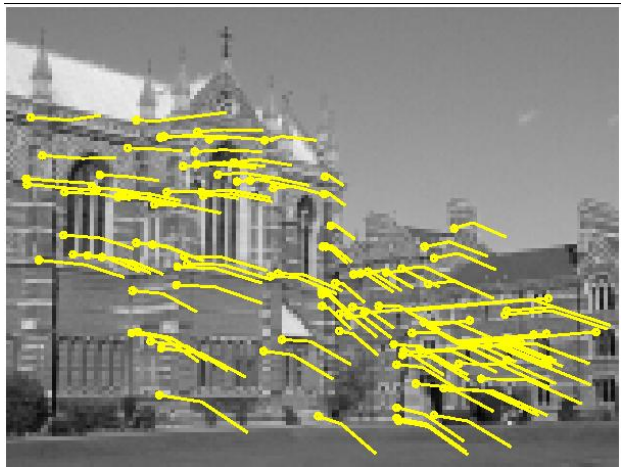
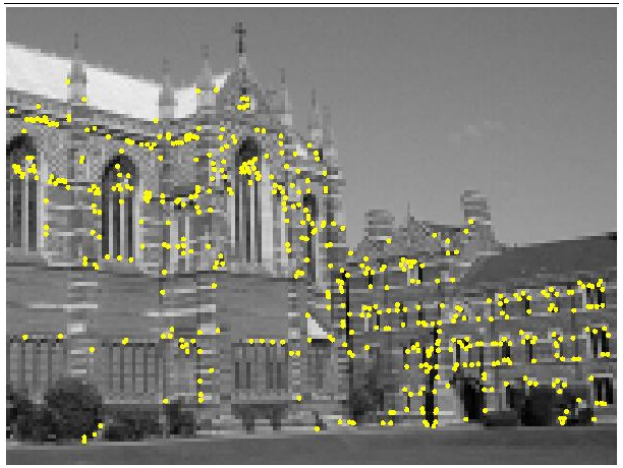
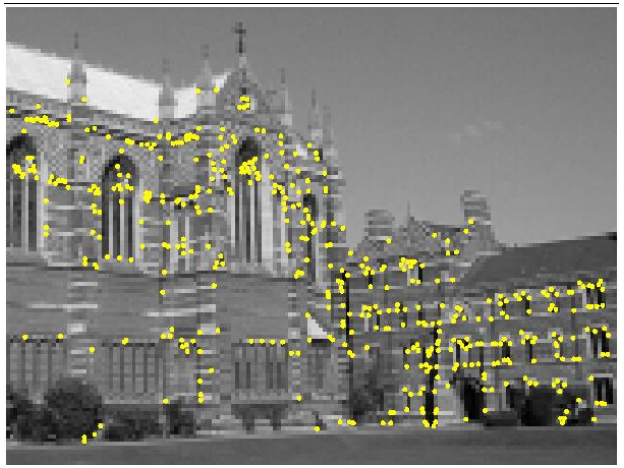
Input Images

Feature detection and matching



Detected Corners

Feature detection and matching

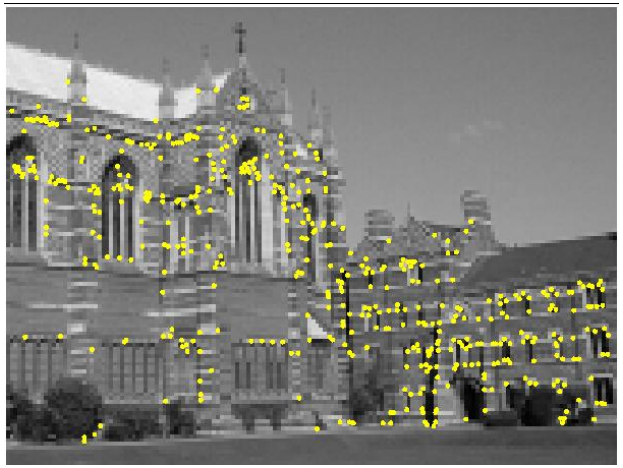




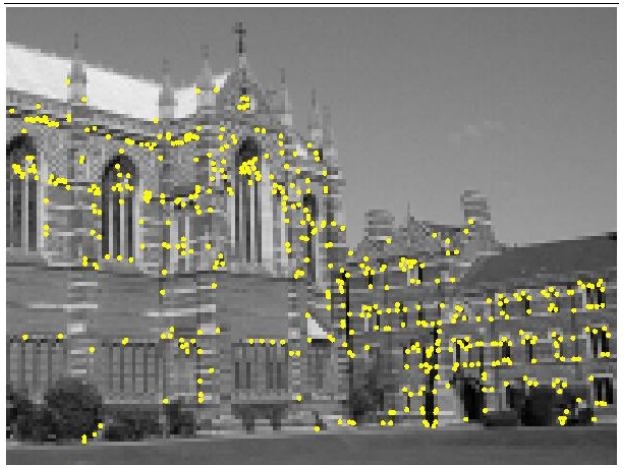
Simple Matching

Expect some false matches

Feature detection and matching







No false matches

Simple Matching
Including Outlier Rejection

Linear estimation of trifocal tensor using the direct linear transformation (DLT) algorithm

- Minimize algebraic error

$$x^i \ell'_j \ell''_k \mathcal{T}_i^{jk} = 0 \quad \text{Trilinearity}$$

$$(\mathbf{x}^\top \otimes (\boldsymbol{\ell}'^\top \otimes \boldsymbol{\ell}''^\top))\mathbf{t} = 0$$

where $\otimes$ denotes the Kronecker product and

$$\mathbf{t} = (\mathcal{T}_1^{11}, \mathcal{T}_1^{12}, \mathcal{T}_1^{13}, \mathcal{T}_1^{21}, \mathcal{T}_1^{22}, \mathcal{T}_1^{23}, \mathcal{T}_1^{31}, \mathcal{T}_1^{32}, \mathcal{T}_1^{33}, \mathcal{T}_2^{11}, \mathcal{T}_2^{12}, \mathcal{T}_2^{13}, \mathcal{T}_2^{21}, \mathcal{T}_2^{22},$$
$$\mathcal{T}_2^{23}, \mathcal{T}_2^{31}, \mathcal{T}_2^{32}, \mathcal{T}_2^{33}, \mathcal{T}_3^{11}, \mathcal{T}_3^{12}, \mathcal{T}_3^{13}, \mathcal{T}_3^{21}, \mathcal{T}_3^{22}, \mathcal{T}_3^{23}, \mathcal{T}_3^{31}, \mathcal{T}_3^{32}, \mathcal{T}_3^{33})^\top$$

Linear estimation of trifocal tensor using the direct linear transformation (DLT) algorithm

Given $n \geq 7$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$

$$(\mathbf{x}_i^\top \otimes ([\mathbf{x}'_i]^\perp \otimes [\mathbf{x}''_i]^\perp))\mathbf{t} = \mathbf{0}\,\forall\, i, \text{ solve for } \mathbf{t}, \text{ where}$$

$$\mathbf{t} = (\mathcal{T}_1^{11}, \mathcal{T}_1^{12}, \mathcal{T}_1^{13}, \mathcal{T}_1^{21}, \mathcal{T}_1^{22}, \mathcal{T}_1^{23}, \mathcal{T}_1^{31}, \mathcal{T}_1^{32}, \mathcal{T}_1^{33}, \mathcal{T}_2^{11}, \mathcal{T}_2^{12}, \mathcal{T}_2^{13}, \mathcal{T}_2^{21}, \mathcal{T}_2^{22}, \mathcal{T}_2^{23}, \mathcal{T}_2^{31}, \mathcal{T}_2^{32}, \mathcal{T}_2^{33}, \mathcal{T}_3^{11}, \mathcal{T}_3^{12}, \mathcal{T}_3^{13}, \mathcal{T}_3^{21}, \mathcal{T}_3^{22}, \mathcal{T}_3^{23}, \mathcal{T}_3^{31}, \mathcal{T}_3^{32}, \mathcal{T}_3^{33})^\top$$

$$\begin{bmatrix} \mathbf{x}_1^\top \otimes ([\mathbf{x}'_1]^\perp \otimes [\mathbf{x}''_1]^\perp) \\ \mathbf{x}_2^\top \otimes ([\mathbf{x}'_2]^\perp \otimes [\mathbf{x}''_2]^\perp) \\ \vdots \\ \mathbf{x}_n^\top \otimes ([\mathbf{x}'_n]^\perp \otimes [\mathbf{x}''_n]^\perp) \end{bmatrix} \mathbf{t} = \mathbf{0}$$

$4n$ x 27

$$\mathtt{A}\mathbf{t} = \mathbf{0}, \text{ where } \mathtt{A} = \begin{bmatrix} \mathbf{x}_1^\top \otimes ([\mathbf{x}'_1]^\perp \otimes [\mathbf{x}''_1]^\perp) \\ \mathbf{x}_2^\top \otimes ([\mathbf{x}'_2]^\perp \otimes [\mathbf{x}''_2]^\perp) \\ \vdots \\ \mathbf{x}_n^\top \otimes ([\mathbf{x}'_n]^\perp \otimes [\mathbf{x}''_n]^\perp) \end{bmatrix}$$

Solve for $\mathbf{t}$ using a constrained minimization, subject to a span-space constraint. Minimize $\|\mathtt{A}\mathbf{t}\|$ subject to $\|t\| = 1$ and $\mathbf{t} = \mathtt{G}\hat{\mathbf{t}}$, where $\mathtt{G}$ has rank $r \leq 18$.

See textbook for details

But, data normalization must be used

Retrieve camera projection matrices from trifocal tensor

See textbook for details

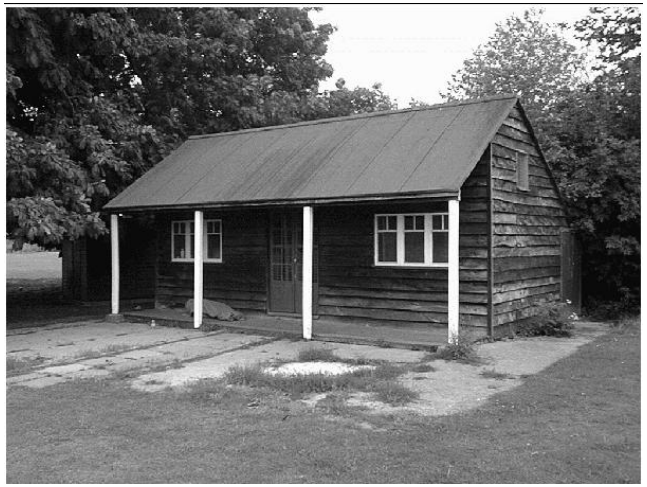
Retrieve epipoles $\mathbf{e}_{21}$ and $\mathbf{e}_{31}$, and fundamental matrix $\mathtt{F}_{1,2}$ from the trifocal tensor.
Then, retrieve the camera projection matrices $\mathtt{P} = [\mathtt{I} \,|\, \mathbf{0}]$ and $\mathtt{P}' = [\mathbf{a}_1 \,|\, \mathbf{a}_2 \,|\, \mathbf{a}_3 \,|\, \mathbf{a}_4]$, where $[\mathbf{a}_1 \,|\, \mathbf{a}_2 \,|\, \mathbf{a}_3]$ is full rank, from $\mathtt{F}_{1,2}$.
Two possible choices of third camera projection matrix $\mathtt{P}''$

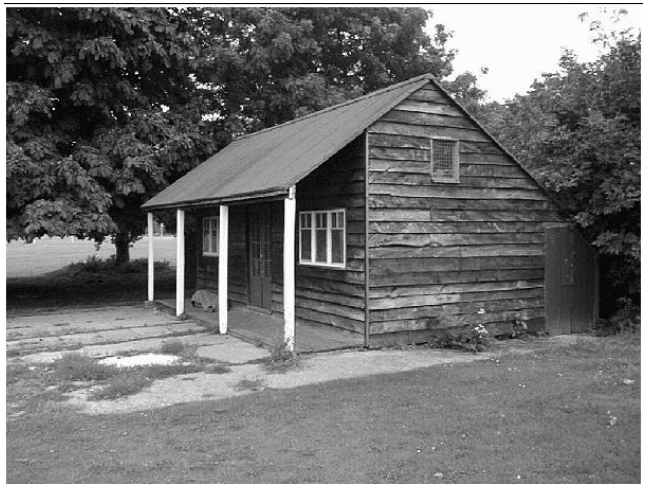
$$\mathtt{P}''_1 = [\mathbf{b}_1 \,|\, \mathbf{b}_2 \,|\, \mathbf{b}_3 \,|\, \mathbf{b}_4], \text{ where } \mathbf{b}_{\{1,2,3\}} = (\mathbf{e}_{31}\mathbf{a}^\top_{\{1,2,3\}} - \mathtt{T}^\top_{\{1,2,3\}})\mathbf{a}_4 \text{ and } \mathbf{b}_4 = \mathbf{e}_{31}$$

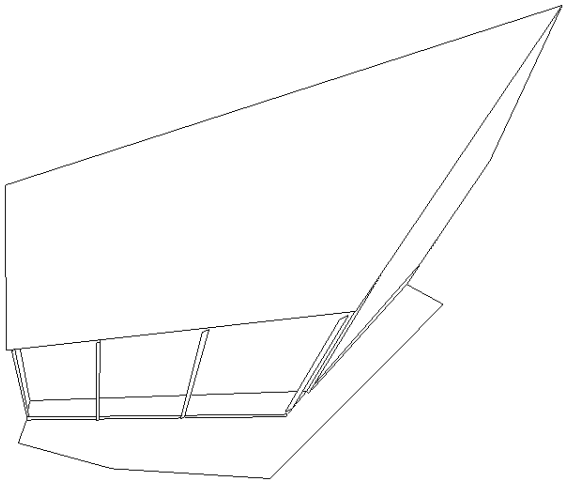
$$\text{or } \mathtt{P}''_2 = [\mathbf{c}_1 \,|\, \mathbf{c}_2 \,|\, \mathbf{c}_3 \,|\, \mathbf{c}_4], \text{ where } \mathbf{c}_{\{1,2,3\}} = (-\mathbf{e}_{31}\mathbf{a}^\top_{\{1,2,3\}} - \mathtt{T}^\top_{\{1,2,3\}})\mathbf{a}_4 \text{ and } \mathbf{c}_4 = -\mathbf{e}_{31}$$

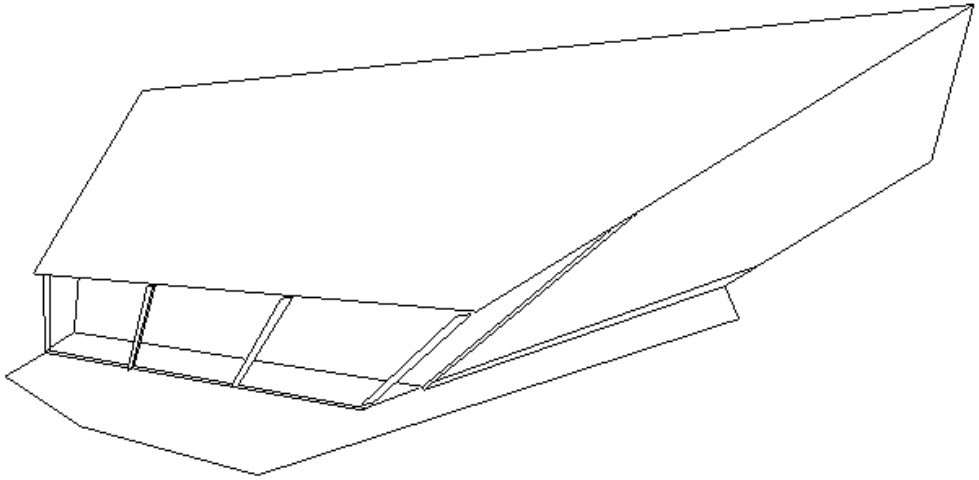
Choose the $\mathtt{P}''$ compatible with the trifocal tensor.

Up to 3D projective transformation









Nonlinear estimation using the Levenberg-Marquardt algorithm

Given

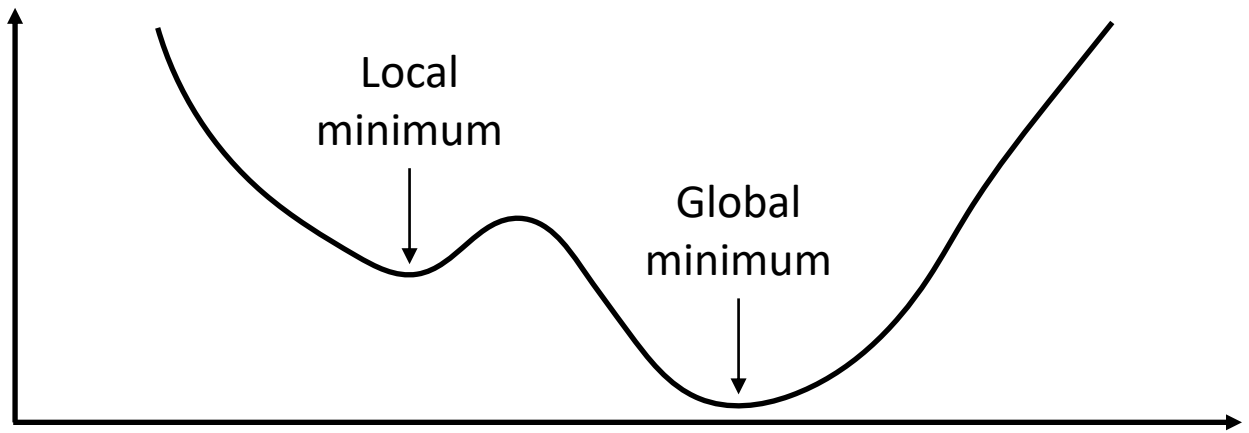
Measurement vector $\mathbf{X}$ with associated covariance matrix $\Sigma_{\mathbf{X}}$

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

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

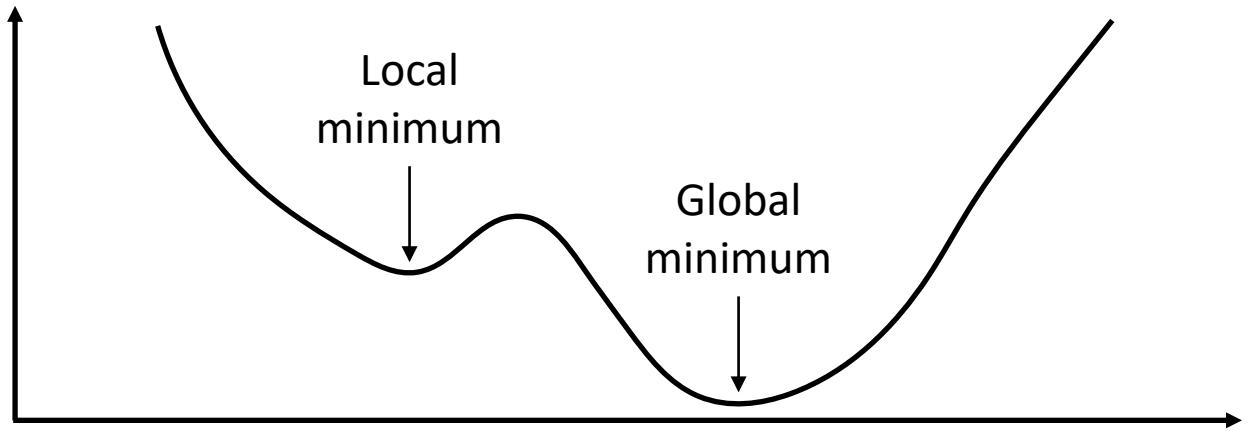
Objective

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**



Parameter vector

- The trifocal tensor has 18 degrees of freedom, so can be parameterized with 18 parameters
  - Advanced topic (come to office hours)
- Alternatively, fix the first camera projection matrix and adjust the second and third ones

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 trifocal tensor using the Levenberg-Marquardt algorithm

Parameter vector $(22 + 3n)$ x 1

Parameterization of homogeneous vector

$$\left(\hat{\mathbf{p}}'^{\top}, \hat{\mathbf{p}}''^{\top}, \hat{\mathbf{X}}_1^{\top}, \hat{\mathbf{X}}_2^{\top}, \ldots, \hat{\mathbf{X}}_n^{\top}\right)^{\top}$$

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

Mapping

$$\hat{\mathbf{x}}_i = \mathrm{P}\hat{\mathbf{X}}_i \,\forall\, i$$
$$\hat{\mathbf{x}}'_i = \hat{\mathrm{P}}'\hat{\mathbf{X}}_i \,\forall\, i$$
$$\hat{\mathbf{x}}''_i = \hat{\mathrm{P}}''\hat{\mathbf{X}}_i \,\forall\, i$$

Measurement vector $6n$ x 1

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

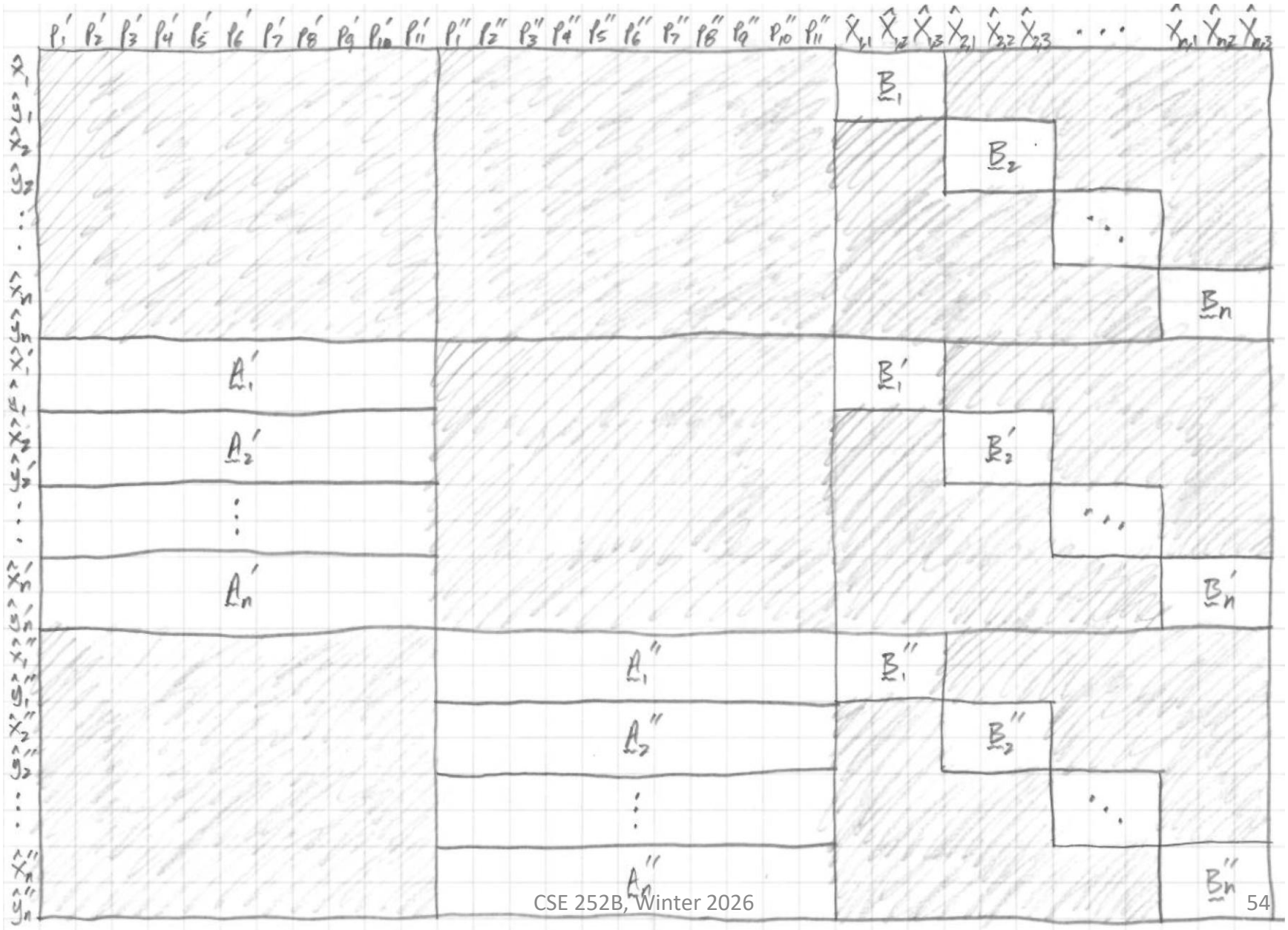
with associated covariance matrices $\Sigma_{\tilde{\mathbf{x}}_1}, \Sigma_{\tilde{\mathbf{x}}_2}, \ldots, \Sigma_{\tilde{\mathbf{x}}_n}, \Sigma_{\tilde{\mathbf{x}}'_1}, \Sigma_{\tilde{\mathbf{x}}'_2}, \ldots, \Sigma_{\tilde{\mathbf{x}}'_n}, \Sigma_{\tilde{\mathbf{x}}''_1}, \Sigma_{\tilde{\mathbf{x}}''_2},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

Cost

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

But, use data normalization

Jacobian



Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

- Jacobian

2 x 12 12 x 11

$$\mathtt{A}'_i = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \hat{\mathbf{p}}'} = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \bar{\mathbf{p}}'} \frac{\partial \bar{\mathbf{p}}'}{\partial \hat{\mathbf{p}}'}$$

$$\mathtt{A}''_i = \frac{\partial \hat{\bar{\mathbf{x}}}''_i}{\partial \hat{\mathbf{p}}''} = \frac{\partial \hat{\bar{\mathbf{x}}}''_i}{\partial \bar{\mathbf{p}}''} \frac{\partial \bar{\mathbf{p}}'}{\partial \hat{\mathbf{p}}'}$$

$$\mathtt{B}_i = \frac{\partial \hat{\bar{\mathbf{x}}}_i}{\partial \hat{\mathbf{X}}_i} = \frac{\partial \hat{\bar{\mathbf{x}}}_i}{\partial \bar{\mathbf{X}}_i} \frac{\partial \bar{\mathbf{X}}_i}{\partial \hat{\mathbf{X}}_i}$$

$$\mathtt{B}'_i = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \hat{\mathbf{X}}_i} = \frac{\partial \hat{\bar{\mathbf{x}}}'_i}{\partial \bar{\mathbf{X}}_i} \frac{\partial \bar{\mathbf{X}}_i}{\partial \hat{\mathbf{X}}_i}$$

$$\mathtt{B}''_i = \frac{\partial \hat{\bar{\mathbf{x}}}''_i}{\partial \hat{\mathbf{X}}_i} = \frac{\partial \hat{\bar{\mathbf{x}}}''_i}{\partial \bar{\mathbf{X}}_i} \frac{\partial \bar{\mathbf{X}}_i}{\partial \hat{\mathbf{X}}_i}$$

2 x 4 4 x 3

Projection of a point under the camera projection matrix

The homogeneous 3D point $\mathbf{X}$ is projected to the homogeneous 2D point $\mathbf{x}$ under the (homogeneous) camera projection matrix $\mathtt{P}$ by

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$

Dehomogenizing the 2D point results in the mapping $\mathbf{X} \mapsto \tilde{\mathbf{x}}$. For this mapping

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}} = \frac{1}{w}\begin{bmatrix} \mathbf{X}^\top & \mathbf{0}^\top & -\tilde{x}\mathbf{X}^\top \\ \mathbf{0}^\top & \mathbf{X}^\top & -\tilde{y}\mathbf{X}^\top \end{bmatrix} \qquad 2 \text{ x } 12$$

and

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{X}} = \frac{1}{w}\begin{bmatrix} \mathbf{p}^{1\top} - \tilde{x}\mathbf{p}^{3\top} \\ \mathbf{p}^{2\top} - \tilde{y}\mathbf{p}^{3\top} \end{bmatrix} \qquad 2 \text{ x } 4$$

where $w = \mathbf{p}^{3\top}\mathbf{X}$ and $\mathbf{p}^{i\top}$ is the ith row of $\mathtt{P}$.

Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

- Normal equations

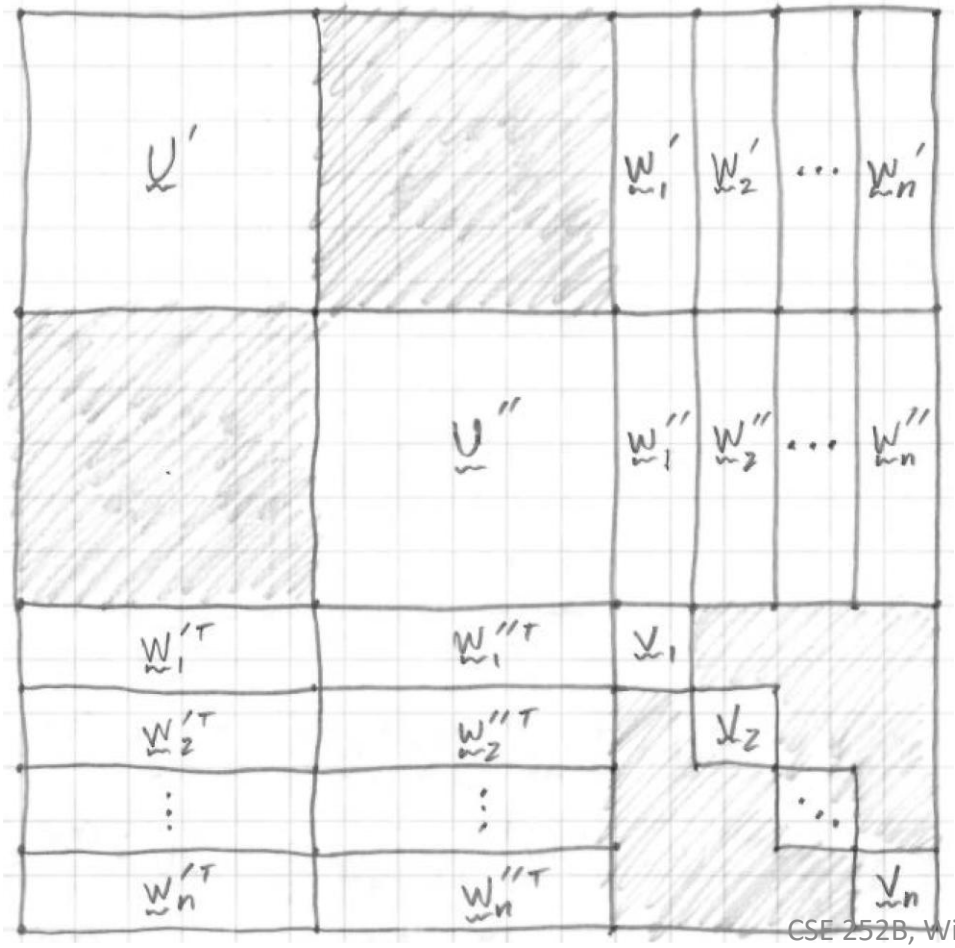
$$\left(\mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \mathtt{J}\right) \boldsymbol{\delta} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$$

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

Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

- Normal equations matrix

$$\begin{bmatrix} \mathrm{U}' & \mathrm{O} & \mathrm{W}' \\ \mathrm{O} & \mathrm{U}'' & \mathrm{W}'' \\ \mathrm{W}'^{\top} & \mathrm{W}''^{\top} & \mathrm{V} \end{bmatrix}$$



$$\mathrm{U}' = \sum_i \mathrm{A}_i'^{\top} \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \mathrm{A}_i' \qquad 11 \text{ x } 11$$

$$\mathrm{U}'' = \sum_i \mathrm{A}_i''^{\top} \Sigma_{\tilde{\mathbf{x}}_i''}^{-1} \mathrm{A}_i'' \qquad 11 \text{ x } 11$$

$$\mathrm{V}_i = \sum_{j=1}^{3} \mathrm{B}_i^{(j)\top} \Sigma_{\tilde{\mathbf{x}}_i^{(j)}}^{-1} \mathrm{B}_i^{(j)} \qquad 3 \text{ x } 3$$

$$\mathrm{W}_i' = \mathrm{A}_i'^{\top} \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \mathrm{B}_i' \qquad 11 \text{ x } 3$$

$$\mathrm{W}_i'' = \mathrm{A}_i''^{\top} \Sigma_{\tilde{\mathbf{x}}_i''}^{-1} \mathrm{B}_i'' \qquad 11 \text{ x } 3$$

Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

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

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

where

$$\boldsymbol{\epsilon}_{\mathbf{a}'} = \sum_i \mathtt{A}_i'^\top \Sigma_{\tilde{\mathbf{x}}_i'}^{-1} \boldsymbol{\epsilon}_i' \qquad 11 \text{ x } 1$$

$$\boldsymbol{\epsilon}_{\mathbf{a}''} = \sum_i \mathtt{A}_i''^\top \Sigma_{\tilde{\mathbf{x}}_i''}^{-1} \boldsymbol{\epsilon}_i'' \qquad 11 \text{ x } 1$$

$$\boldsymbol{\epsilon}_{\mathbf{b}_i} = \sum_{j=1}^{3} \mathtt{B}_i^{(j)\top} \Sigma_{\tilde{\mathbf{x}}_i^{(j)}}^{-1} \boldsymbol{\epsilon}_i^{(j)} \qquad 3 \text{ x } 1$$

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

Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

Augmented normal equations

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

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

where

$$\mathtt{U}^{*} = \begin{bmatrix} \mathtt{U}'^{*} & \mathtt{0} \\ \mathtt{0} & \mathtt{U}''^{*} \end{bmatrix}, \mathtt{W} = \begin{bmatrix} \mathtt{W}' \\ \mathtt{W}'' \end{bmatrix}, \boldsymbol{\delta}_{\mathbf{a}} = \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{a}'} \\ \boldsymbol{\delta}_{\mathbf{a}''} \end{bmatrix}, \text{ and } \boldsymbol{\epsilon}_{\mathbf{a}} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{a}'} \\ \boldsymbol{\epsilon}_{\mathbf{a}''} \end{bmatrix}$$

where $\mathtt{U}'^{*} = \mathtt{U}' + \lambda\mathtt{I}$, $\mathtt{U}''^{*} = \mathtt{U}'' + \lambda\mathtt{I}$, 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 trifocal tensor 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 $\boldsymbol{\delta}_\mathbf{a}$

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

$$\mathtt{S}\boldsymbol{\delta}_\mathbf{a} = \mathbf{e}, \text{ solve for } \boldsymbol{\delta}_\mathbf{a} = (\boldsymbol{\delta}_{\mathbf{a}'}^\top, \boldsymbol{\delta}_{\mathbf{a}''}^\top)^\top \quad 22 \text{ x } 1$$

where

$$\mathtt{S} = \begin{bmatrix} \mathtt{S}^{(2,2)} & \mathtt{S}^{(2,3)} \\ \mathtt{S}^{(3,2)} & \mathtt{S}^{(3,3)} \end{bmatrix}$$

where

$$\mathtt{S}^{(2,2)} = \mathtt{U}'^* - \sum_i \mathtt{W}_i'\mathtt{V}_i^{*-1}\mathtt{W}_i'^\top \quad 11 \text{ x } 11$$

$$\mathtt{S}^{(3,3)} = \mathtt{U}''^* - \sum_i \mathtt{W}_i''\mathtt{V}_i^{*-1}\mathtt{W}_i''^\top \quad 11 \text{ x } 11$$

$$\mathtt{S}^{(2,3)} = -\sum_i \mathtt{W}_i'\mathtt{V}_i^{*-1}\mathtt{W}_i''^\top \quad 11 \text{ x } 11$$

$$\mathtt{S}^{(3,2)} = \mathtt{S}^{(2,3)\top} \quad 11 \text{ x } 11$$

and

$$\mathbf{e} = (\mathbf{e}'^\top, \mathbf{e}''^\top)^\top \quad 22 \text{ x } 1$$

where

$$\mathbf{e}' = \boldsymbol{\epsilon}_{\mathbf{a}'} - \sum_i \mathtt{W}_i'\mathtt{V}_i^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}_i} \quad 11 \text{ x } 1$$

$$\mathbf{e}'' = \boldsymbol{\epsilon}_{\mathbf{a}''} - \sum_i \mathtt{W}_i''\mathtt{V}_i^{*-1}\boldsymbol{\epsilon}_{\mathbf{b}_i} \quad 11 \text{ x } 1$$

Nonlinear estimation of trifocal tensor 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

$$\boldsymbol{\delta}_{\mathbf{b}_i} = \mathtt{V}_i^{*-1}(\boldsymbol{\epsilon}_{\mathbf{b}_i} - \mathtt{W}_i'^\top\boldsymbol{\delta}_{\mathbf{a}'} - \mathtt{W}_i''^\top\boldsymbol{\delta}_{\mathbf{a}''}) \qquad 3 \times 1$$

Candidate parameter vector

$$\begin{aligned} \hat{\mathbf{p}}_0' &= \hat{\mathbf{p}}' + \boldsymbol{\delta}_{\mathbf{a}'} \\ \hat{\mathbf{p}}_0'' &= \hat{\mathbf{p}}'' + \boldsymbol{\delta}_{\mathbf{a}''} \\ \hat{\mathbf{X}}_{i_0} &= \hat{\mathbf{X}}_i + \boldsymbol{\delta}_{\mathbf{b}_i} \end{aligned}$$

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 trifocal tensor using the Levenberg-Marquardt algorithm

1. Initialize the scene points using triangulation Covered next lecture
2. Data normalize the image points

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

scene points

$$\mathbf{X}_{\mathrm{DN},i} = \mathtt{U}\mathbf{X}_i \,\forall\, i$$

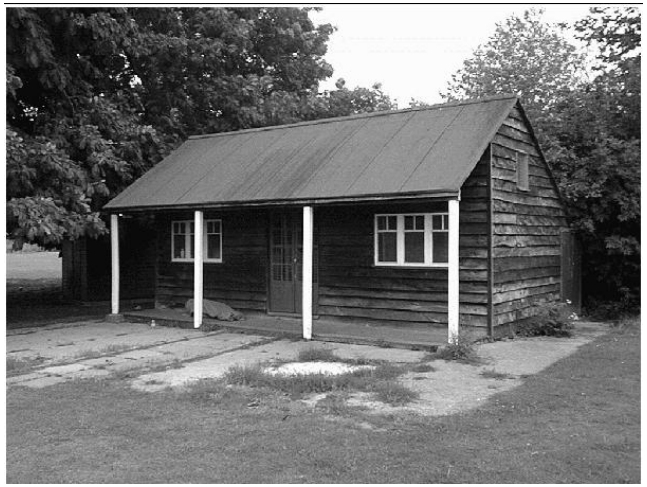
and camera projection matrices

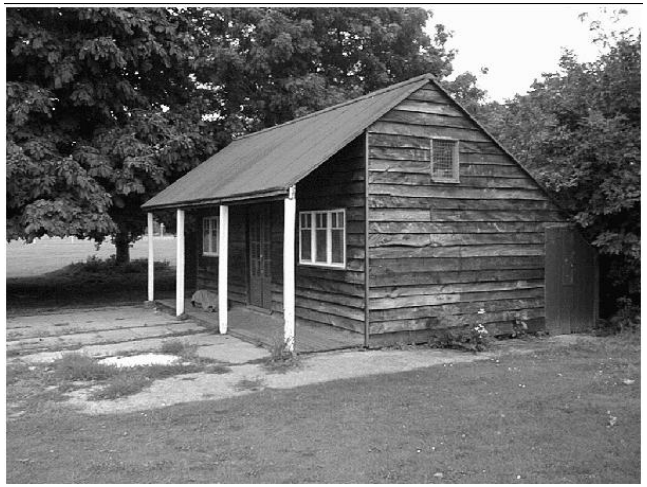
$$\mathtt{P}_{\mathrm{DN}} = \mathtt{T}[\mathtt{I} \mid \mathbf{0}]\mathtt{U}^{-1}$$
$$\mathtt{P}'_{\mathrm{DN}} = \mathtt{T}'\mathtt{P}'\mathtt{U}^{-1}$$
$$\mathtt{P}''_{\mathrm{DN}} = \mathtt{T}''\mathtt{P}''\mathtt{U}^{-1}$$

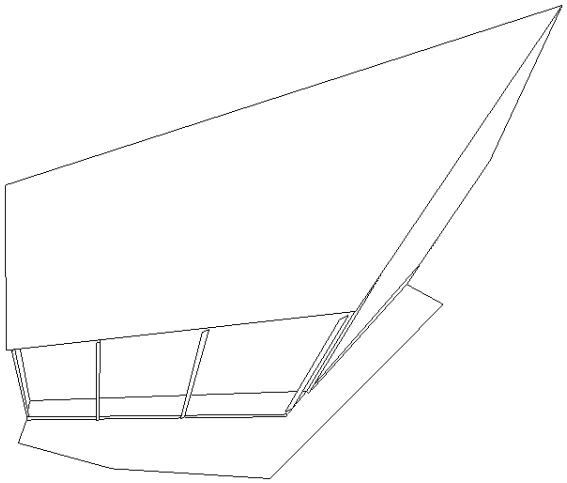
3. Estimate the data normalized camera projection matrices $\mathtt{P}'_{\mathrm{DN}}$ and $\mathtt{P}''_{\mathrm{DN}}$, and data normalized scene points from the data normalized image points
4. Data denormalize the final estimate of the data normalized camera projection matrices

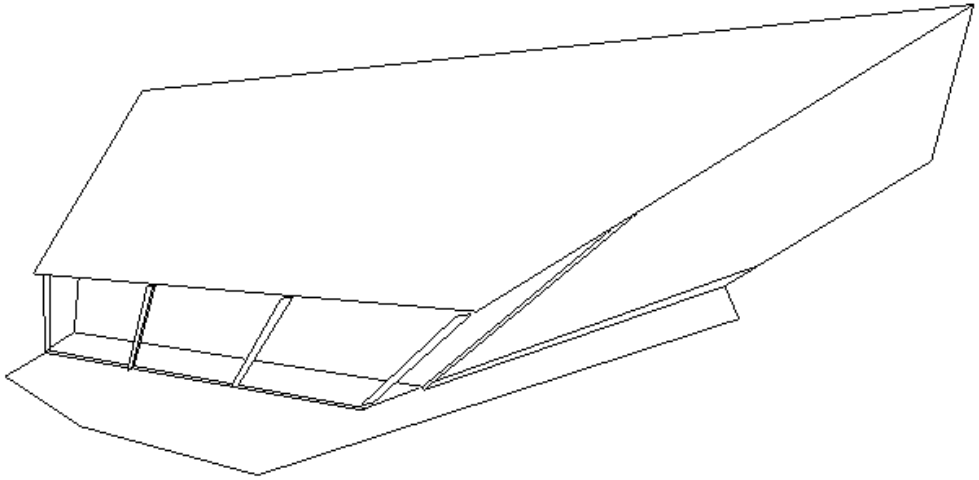
$$\mathtt{P}' = \mathtt{T}'^{-1}\mathtt{P}'_{\mathrm{DN}}\mathtt{U}$$
$$\mathtt{P}'' = \mathtt{T}''^{-1}\mathtt{P}''_{\mathrm{DN}}\mathtt{U}$$

Up to 3D projective transformation









Next lecture

- n view geometry
- Reading
  - Sections 12.1, 12.2, 12.3, and 18.1