CS 395T
Lecture 9: Multi-View Geometry

Qixing Huang

October 1st 2018

Three View Geometry

- Cameras P, P’, P’’ such that

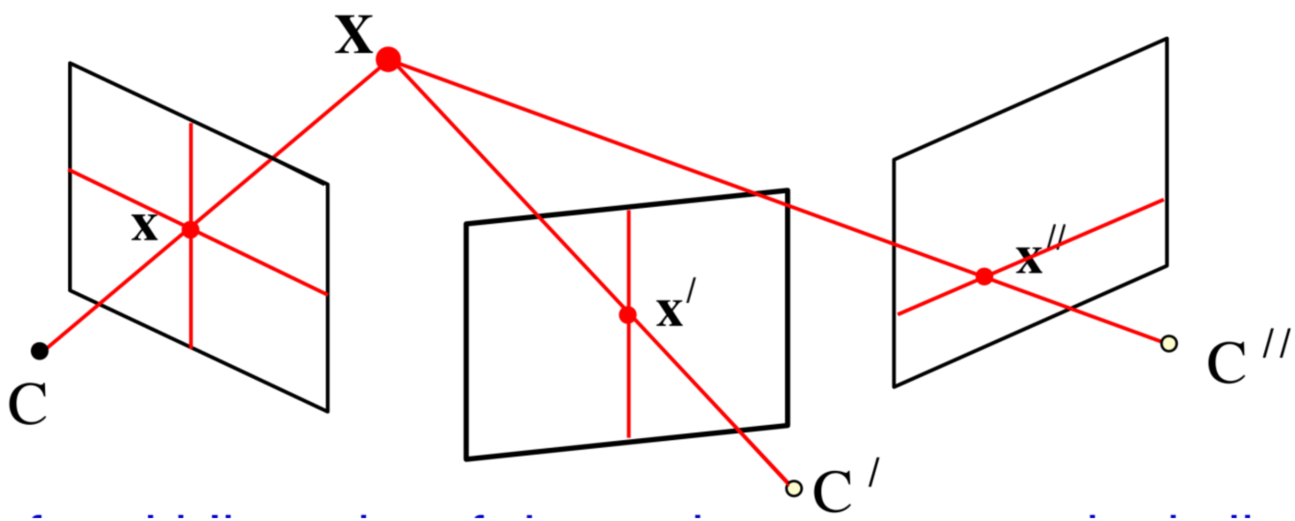
$$\boldsymbol{x} = P\boldsymbol{X} \quad \boldsymbol{x}' = P'\boldsymbol{X} \quad \boldsymbol{x}'' = P''\boldsymbol{X}$$

- Main new result: The Trifocal Tensor
  - Defined for three views
  - Plays a similar role to F for two views
  - Unlike F, trifocal tensor also relates lines in three views

Lines: $\boldsymbol{l}^T\boldsymbol{x} = 0 \quad \boldsymbol{l} = \boldsymbol{x} \times \boldsymbol{x}' \quad \boldsymbol{x} = \boldsymbol{l} \times \boldsymbol{l}'$

Tri-linear Relation

- Back-projected lines passing through corresponding points should all intersect

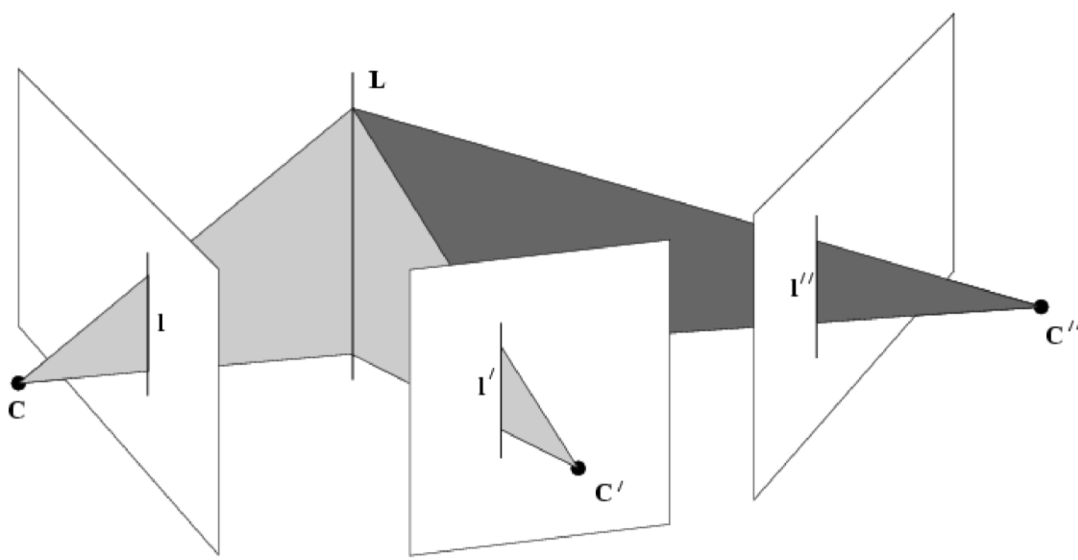


3^3 coefficients: Tfifocal Tensor

Trifocal Tensor

- Tri-linear relation can be represented efficiently as tensor. Easy to express line and point coincidence relations
  - Line-line-line correspondence

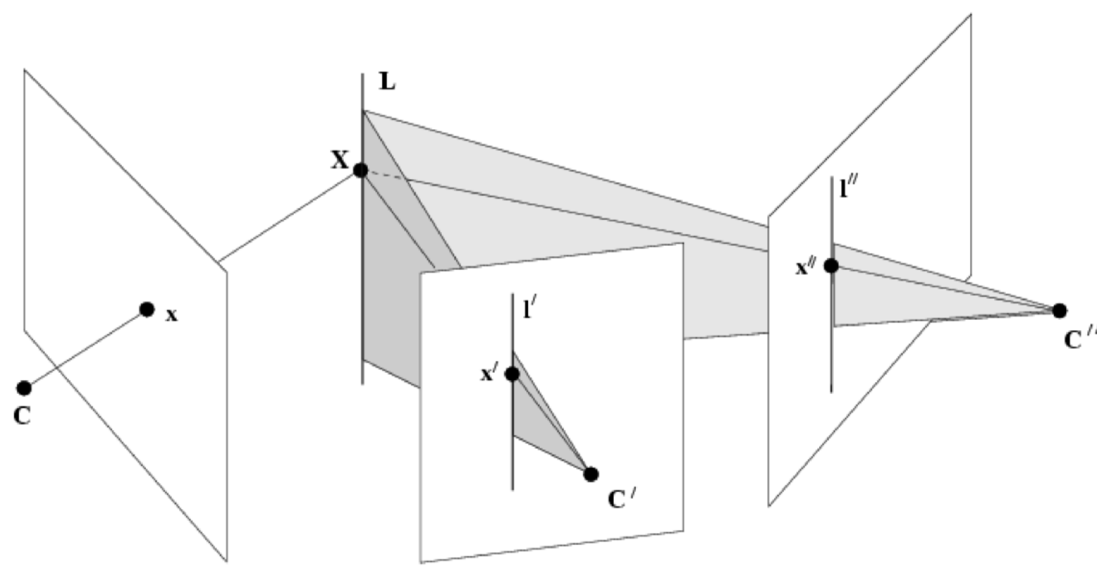
$$\boldsymbol{l}'^T[T_1, T_2, T_3]\boldsymbol{l}'' = \boldsymbol{l}^T \quad \text{or} \quad \boldsymbol{l}'^T[T_1, T_2, T_3]\boldsymbol{l}''(\boldsymbol{l}\times) = \mathbf{0}^T$$



Trifocal Tensor

- Tri-linear relation can be represented efficiently as tensor. Easy to express line and point coincidence relations
  - Point-line-line correspondence

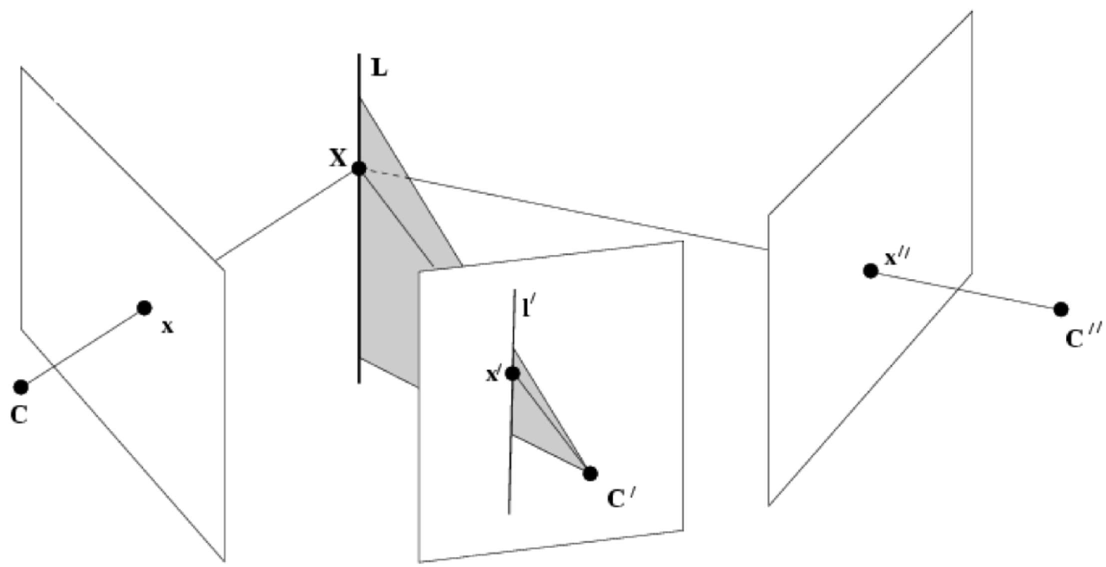
$$\boldsymbol{l}'^T(\sum_i x^i T_i)\boldsymbol{l}'' = 0 \quad \text{for a correspondence} \quad \boldsymbol{x} \Longleftrightarrow \boldsymbol{l}' \Longleftrightarrow \boldsymbol{l}''$$



Trifocal Tensor

- Tri-linear relation can be represented efficiently as tensor. Easy to express line and point coincidence relations
  - Point-line-point correspondence

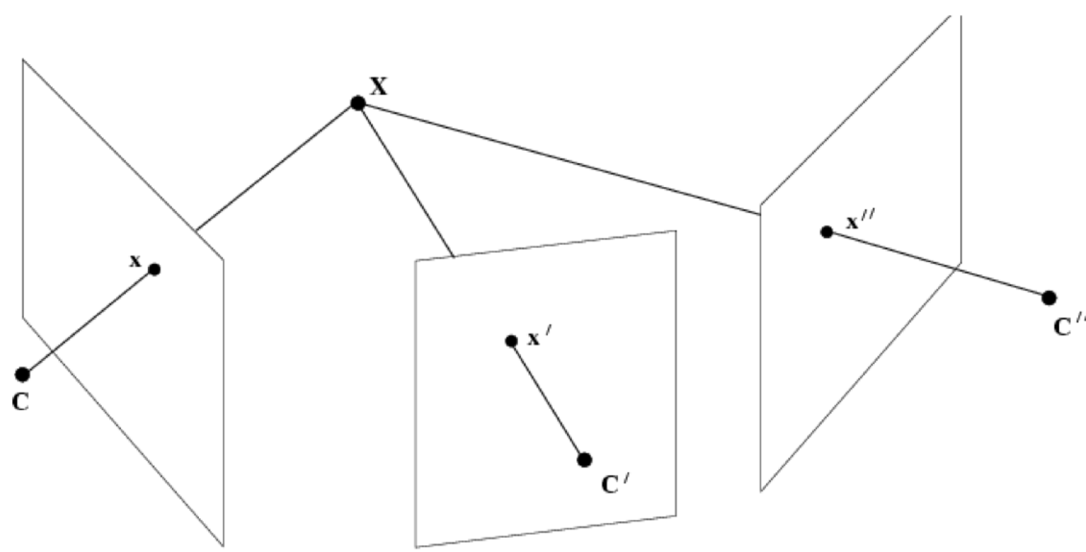
$$\boldsymbol{l}'^T \left(\sum_i x_i T_i\right)(\boldsymbol{x}'' \times) = \mathbf{0}^T \quad \text{for a correspondence} \quad \boldsymbol{x} \Longleftrightarrow \boldsymbol{l}' \Longleftrightarrow \boldsymbol{x}''$$



Trifocal Tensor

- Tri-linear relation can be represented efficiently as tensor. Easy to express line and point coincidence relations
  - Point-point-point correspondence

$$\boldsymbol{l}'^T \left(\sum_i x_i T_i\right)(\boldsymbol{x}'' \times) = 0_{3\times 3}$$



More Views?

Problem Statement-Structure and Motion Estimation

- Given: *n* matching image points $\boldsymbol{x}_j^i$ over *m* views
- Find: the cameras P^i and the 3D points $\boldsymbol{X}_j$ such that $\boldsymbol{x}_j^i = P^i \boldsymbol{X}_j$

$$\min_{P^i, \boldsymbol{X}_j} \sum_{j \in \text{points}} \sum_{i \in \text{views}} d(\boldsymbol{x}_j^i, P^i \boldsymbol{X}_j)^2$$

Minimizing reprojection error corresponds
to a maximum likelihood estimation (MLE) under
the assumption of zero mean Gaussian noise

Factorization

Factorization

- Factorize observations in structure of the scene and motion/calibration of the camera

- Use **all points** in **all images** at the same time
  - Projective factorization

Perspective factorization

- The camera equations

$$\lambda_{ij}\boldsymbol{x}_{ij} = P^i\boldsymbol{X}_j, \qquad i = 1, \cdots, m, j = 1, \cdots, n.$$

- For a fixed image i can be written in matrix form as

$$\boldsymbol{x}_i\Lambda_i = P^i\boldsymbol{X}$$

where $\boldsymbol{x}_i = [\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{im}], \quad \boldsymbol{X} = [\boldsymbol{X}_1, \cdots, \boldsymbol{X}_m]$

$$\Lambda_i = \mathrm{diag}(\lambda_{i1}, \cdots, \lambda_{im}).$$

Perspective factorization

- All equations can be collected for all *i* as

$$\boldsymbol{x} = P\boldsymbol{X}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \Lambda_1 \\ \boldsymbol{x}_2 \Lambda_2 \\ \cdots \\ \boldsymbol{x}_{nn} \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \cdots \\ \boldsymbol{P}_m \end{bmatrix}$$

In these formulas x are known, but Λ_i, **P** and **X** are unknown

Observe that **PX** is a product of a 3mx4 matrix and a 4xn matrix, i.e., it is a rank-4 matrix

Perspective factorization algorithm

- Assume that Λ_i are known, then **PX** is known
- Use the singular value decomposition

$$\boldsymbol{PX} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$$

- In the noise-free case

$$\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \cdots, 0)$$

and a reconstruction can be obtained by setting:

$$\boldsymbol{P} = \text{the first four columns of } \boldsymbol{U}\Sigma$$
$$\boldsymbol{X} = \text{the first four rows of } \boldsymbol{V}$$

Iterative perspective factorization

- When are unknown the following algorithm can be used:
  - 1. Set $\lambda_{ij} = 1$ (affine approximation)
  - 2. Factorize **PX** and obtain an estimate of **P** and **X**
    - If σ_5 is sufficiently small then STOP
  - 3. Use **x**, **P** and **X** to estimate Λ_i from the camera equations (linearly)
  - 4.Go to 2.

Questions: Does it converge?
Does it converge to the global optimal,
assuming that all measurements are clean

Bundle Adjustment

Global refinement of
recovered structure and motion

Bundle adjustment – refining structure and motion

- Minimize reprojection error

$$\min_{P^i, \boldsymbol{X}_j} \sum_{j \in \text{points}} \sum_{i \in \text{views}} d(\boldsymbol{x}_j^i, P^i \boldsymbol{X}_j)^2$$

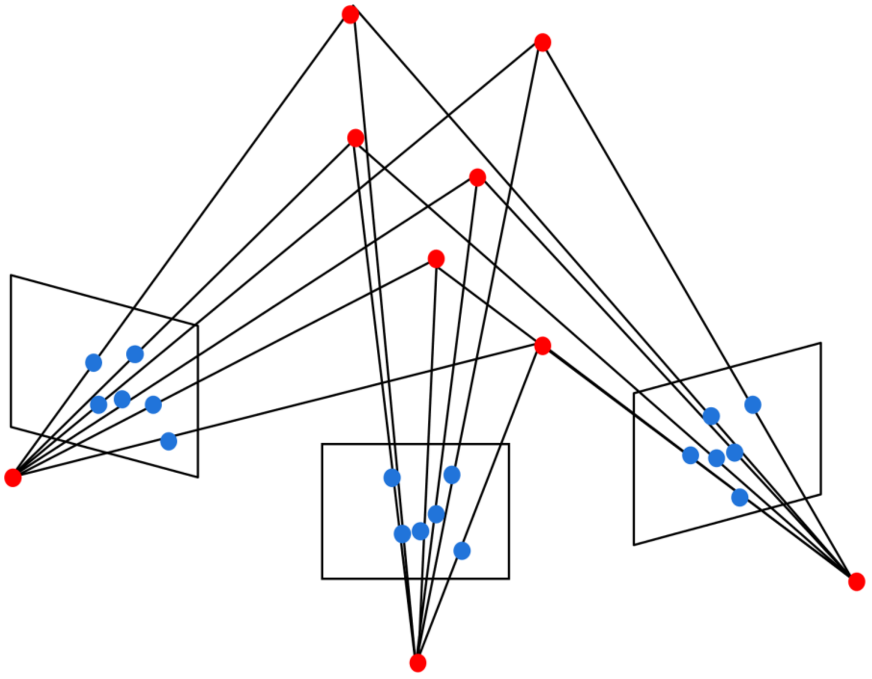
- Maximum likelihood estimation

  (if error zero-mean Gaussian noise)
- Huge problem but can be solved efficiently

  (Bundle adjustment)

Bundle adjustment

- Developed in photogrammetry in 50's



Non-linear least squares

- Linear approximation of residual $e_0 - \mathrm{J}\Delta$
- Allows quadratic approximation of sum-of-squares

$$(e_0 - \mathrm{J}\Delta)^{\mathrm{T}}(e_0 - \mathrm{J}\Delta)$$

- Minimization corresponds to finding zeros of derivatives

$$2\mathrm{J}^{\mathrm{T}}\mathrm{J}\Delta - 2\mathrm{J}^{\mathrm{T}}e_0 = 0$$

$$\Rightarrow \Delta = \underbrace{\boxed{\left(\mathrm{J}^{\mathrm{T}}\mathrm{J}\right)^{-1}}}_{\mathbf{N}}\mathrm{J}^{\mathrm{T}}e_0$$

- Levenberg-Marquardt: extra-term to deal with singular **N** (decrease/increase lambda if success/failure to descent)

$$\mathrm{N}' = \mathrm{J}^{\mathrm{T}}\mathrm{J} + \lambda\,\mathrm{diag}(\mathrm{J}^{\mathrm{T}}\mathrm{J})$$