3D Data Processing

Structure from Motion
Bundle Adjustment

Hyoseok Hwang

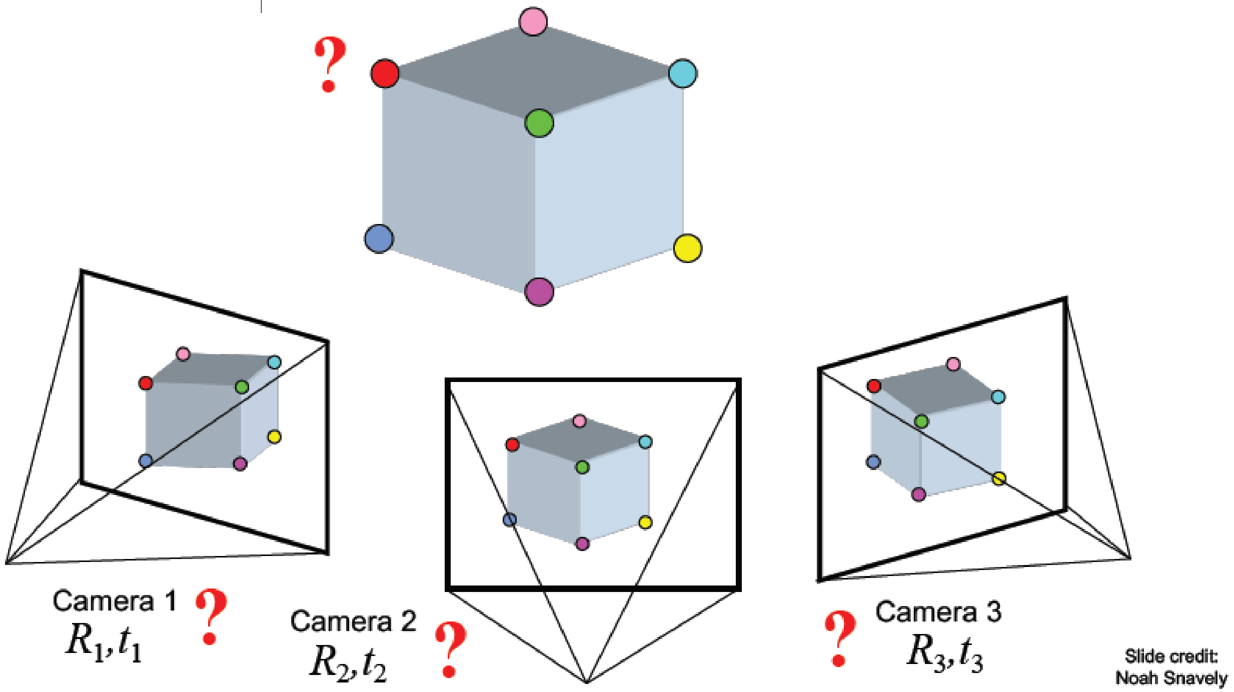
Lecture notes are from Marc Pollefeys

Today

- Structure from Motion
- Bundle Adjustment

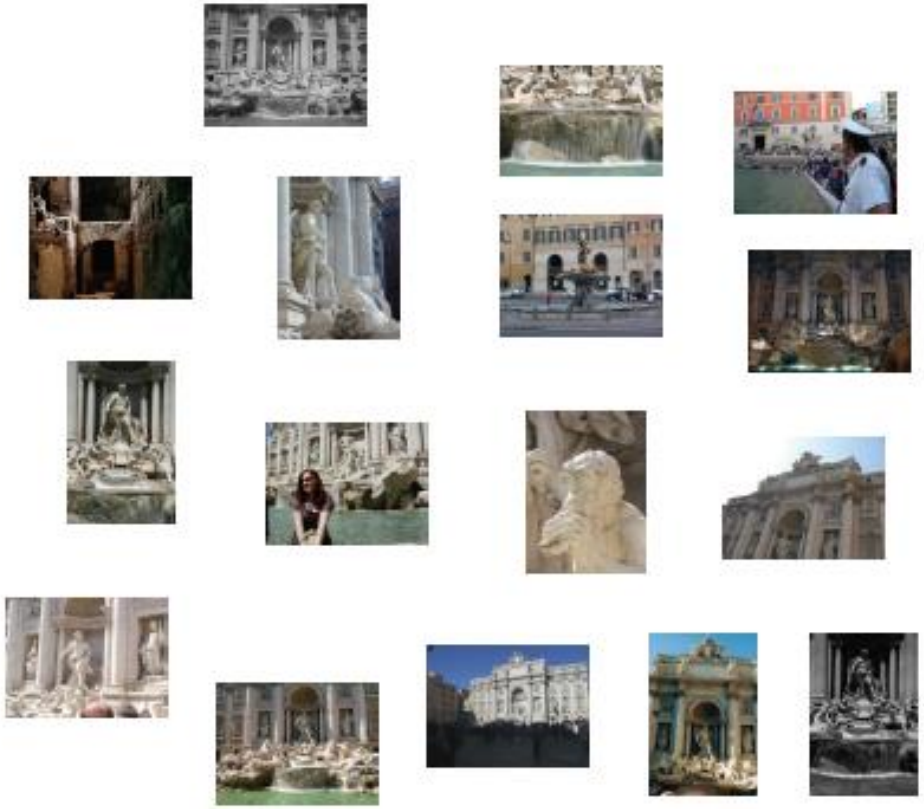
Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Feature detection

- Images from the same scene

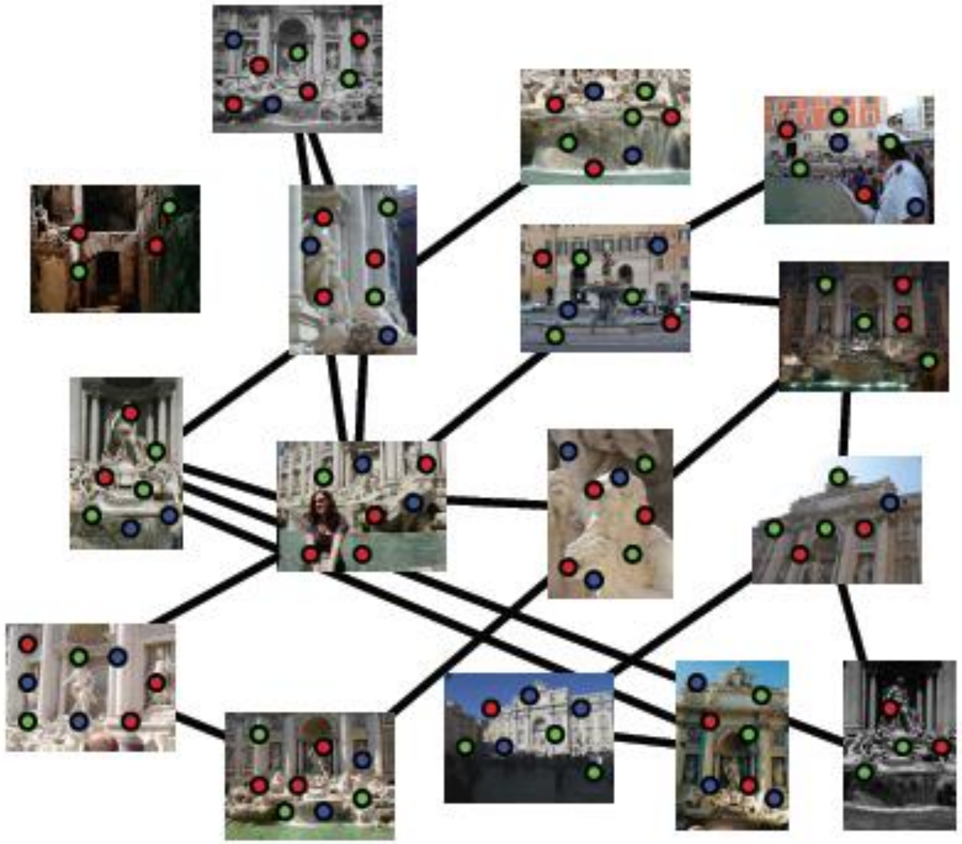


Feature detection

- Detect SIFT (or other) features

Feature matching

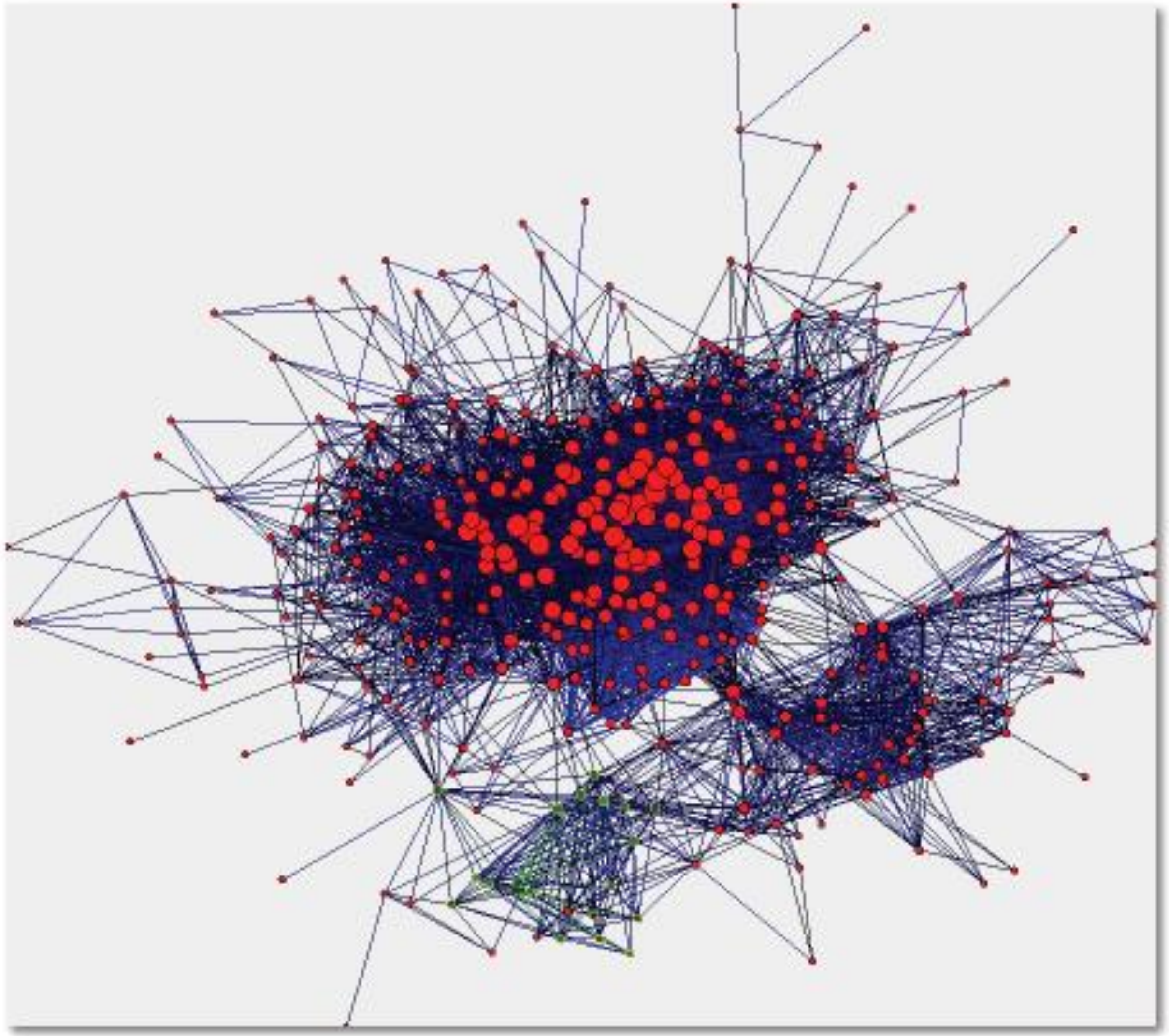
- Match features between each pair of images



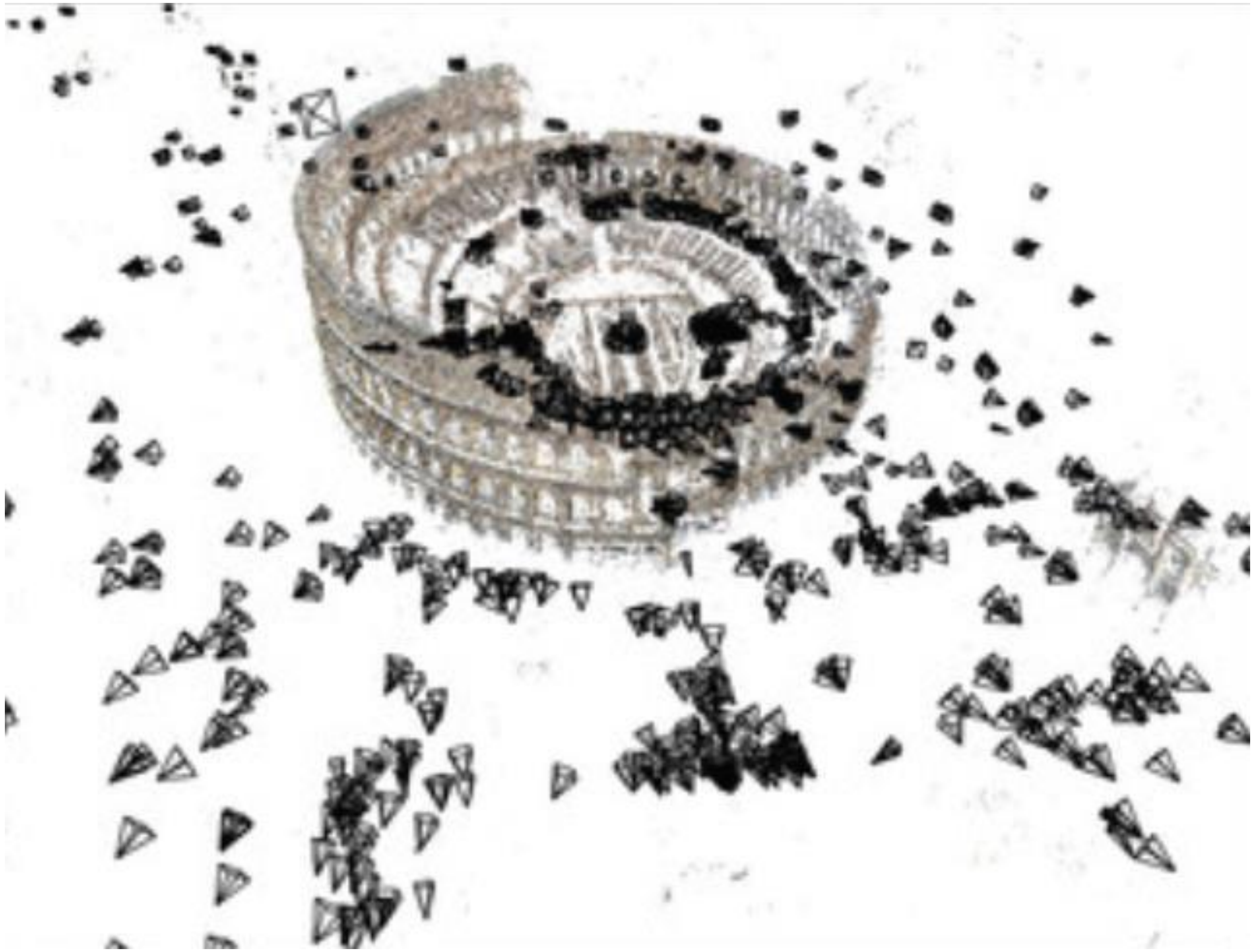
Feature matching

- Use RANSAC to estimate fundamental matrix between each pair

Image connectivity graph

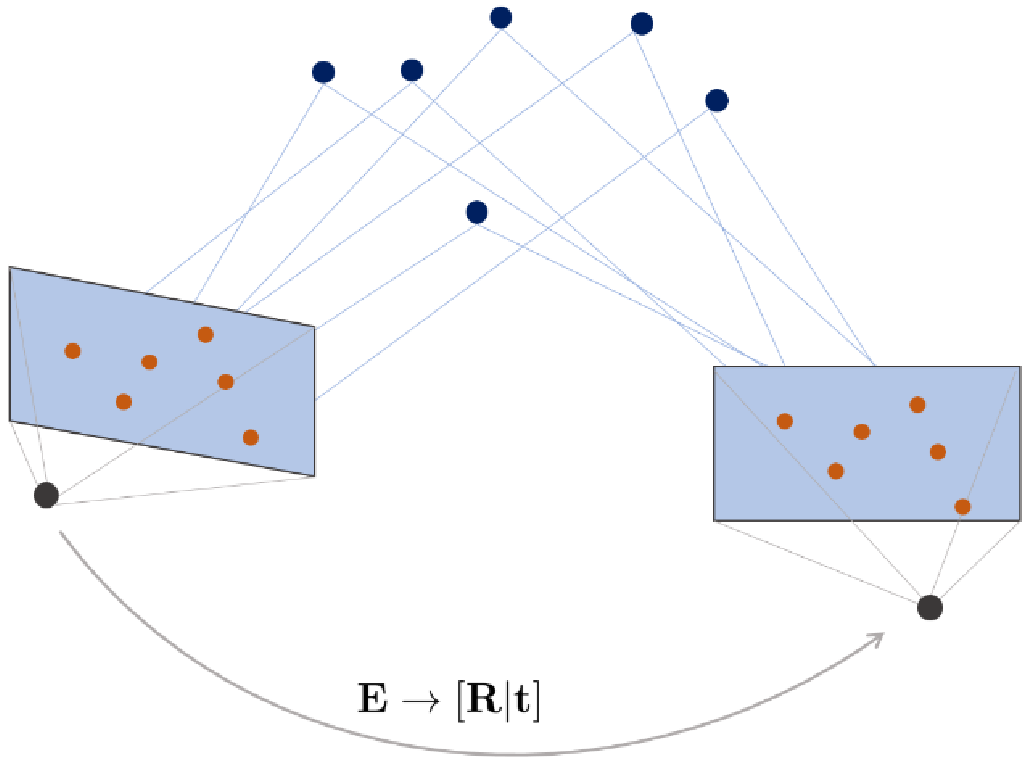


3D reconstruction



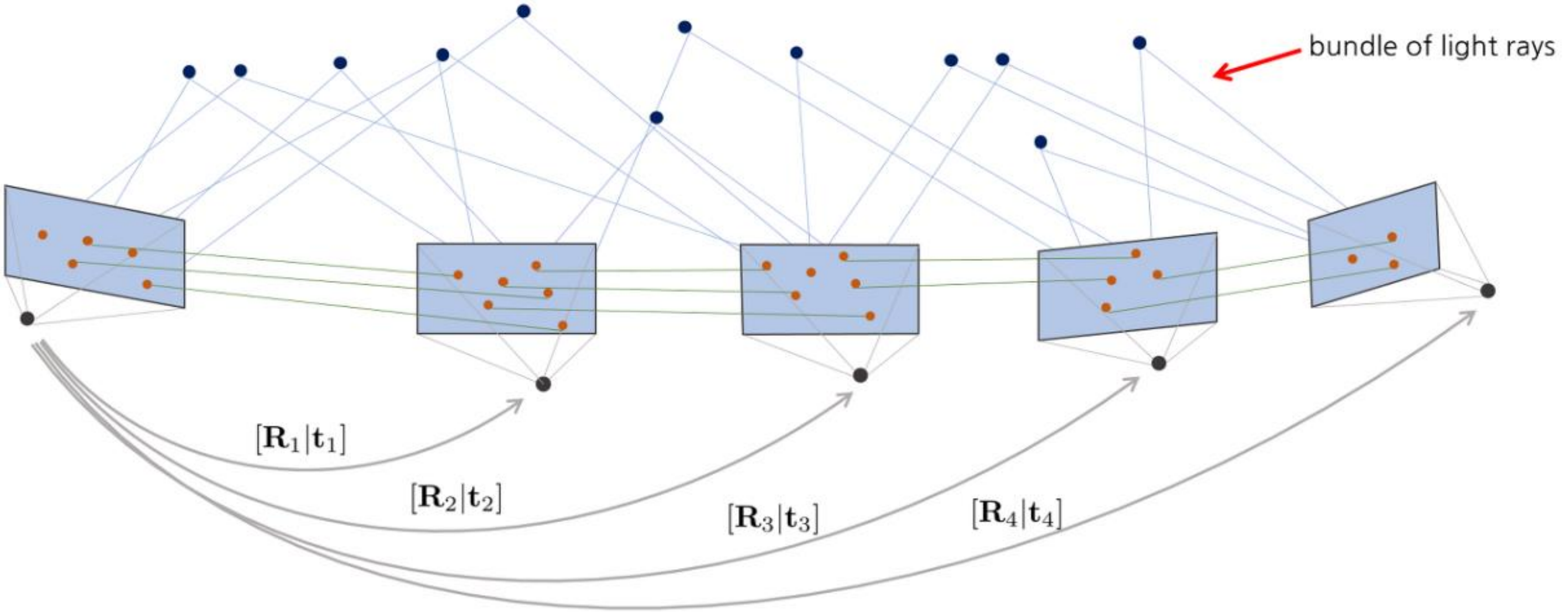
Bundle adjustment

- Where can I found corresponding point?



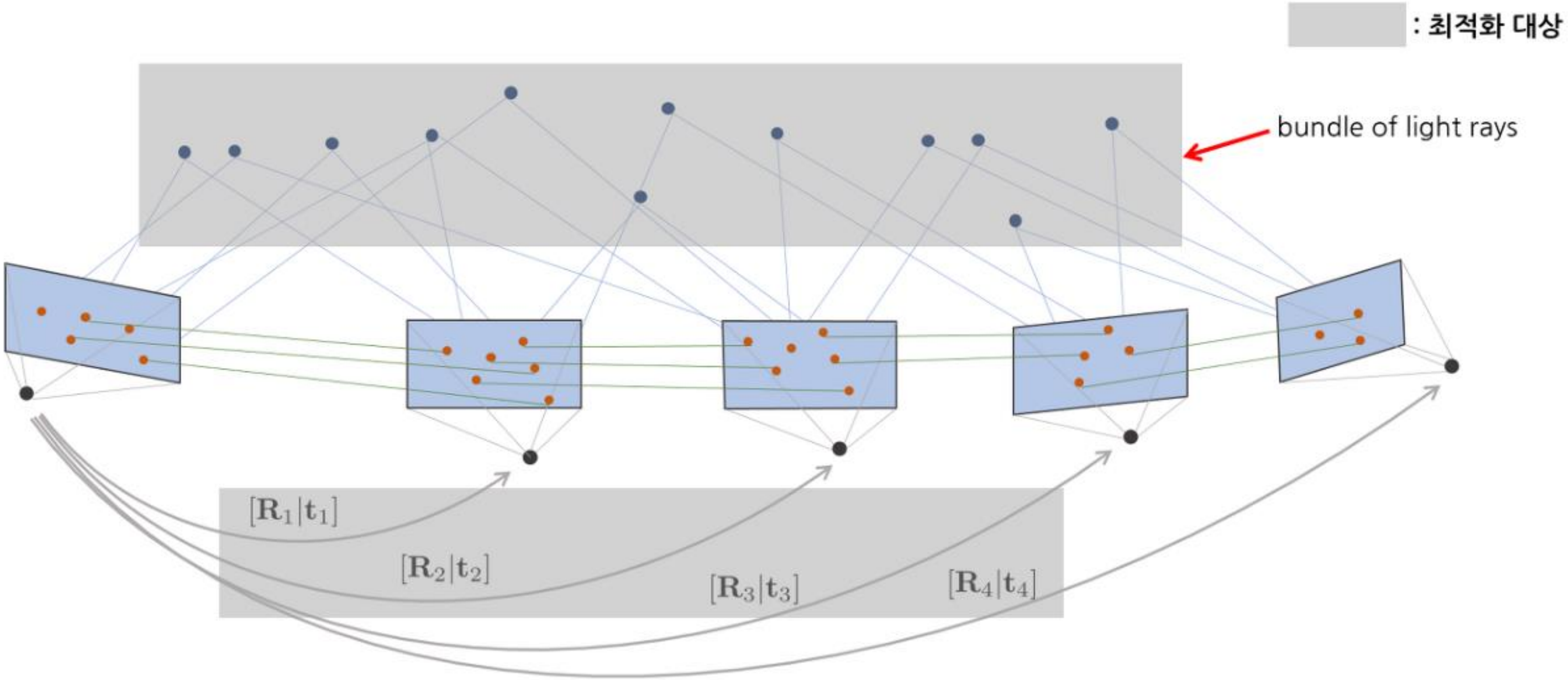
Bundle adjustment

- bundle



Bundle adjustment

- bundle



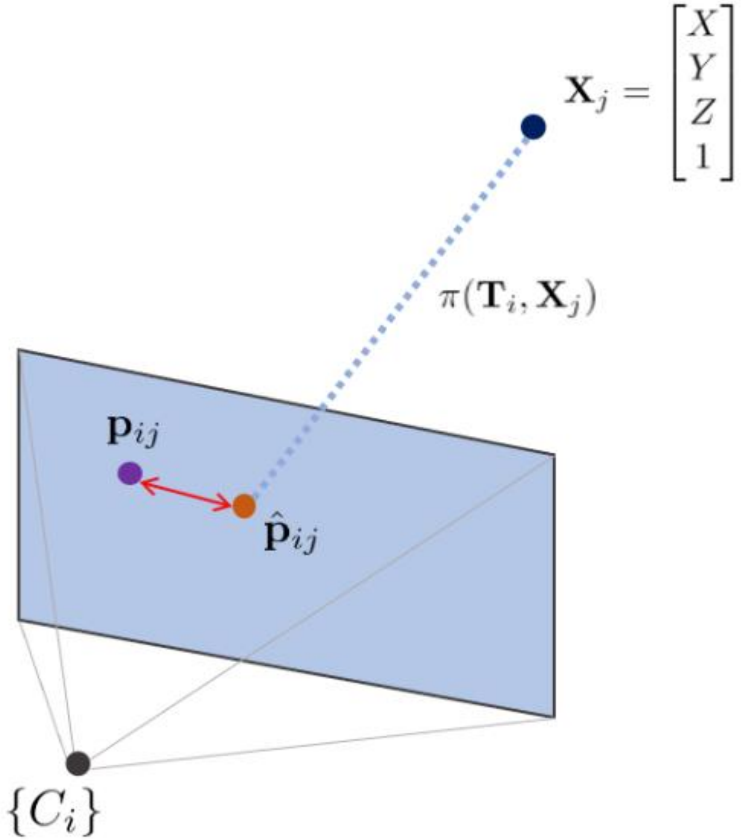
Reprojection error

- Projection 3D point to the image plane

$$\tilde{\mathbf{p}} = \pi_h(\cdot) : \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} X'/Z' \\ Y'/Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{bmatrix}$$

- Transform a point on the image plane to the pixel

$$\hat{\mathbf{p}} = \pi_k(\cdot) = \tilde{\mathbf{K}}\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{bmatrix} = \begin{bmatrix} f\tilde{u} + c_x \\ f\tilde{v} + c_y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

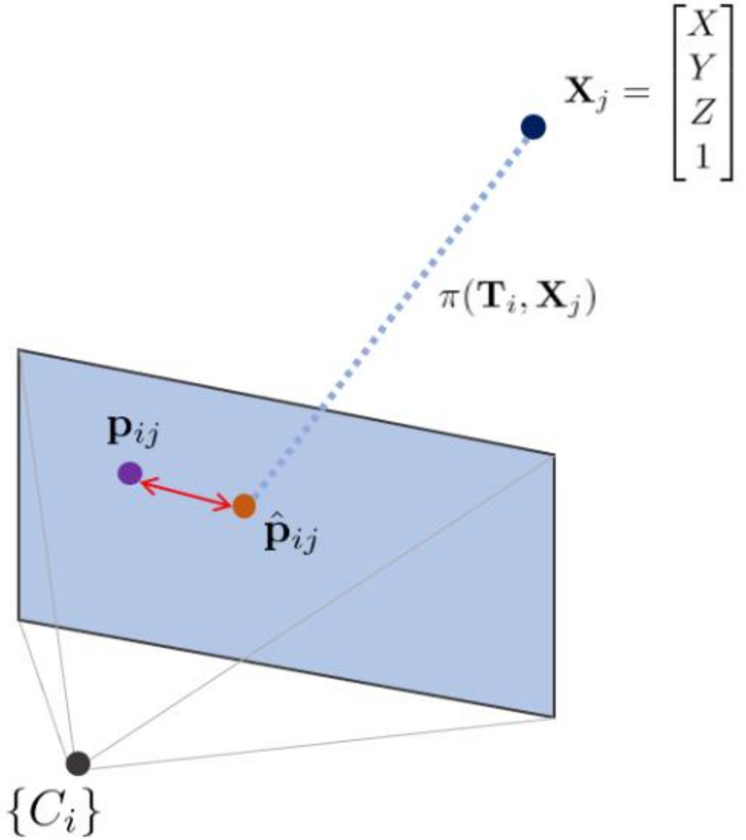


Reprojection error

- $K, \widetilde{K}$: camera intrinsic parameter (3x3, 3x2)

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \tilde{\mathbf{K}} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \end{bmatrix}$$

- State variable $\mathcal{X} = [\mathcal{T}_1, \cdots, \mathcal{T}_m, \mathbf{X}_1, \cdots, \mathbf{X}_n]^\top$
- m: the number of cameras
- n: the number of 3D points
- $\mathcal{T}_i = [\mathbf{R}_i, \mathbf{t}_i]$
- p_{ij}: observed points
- $\hat{p}_{ij}$: projected points



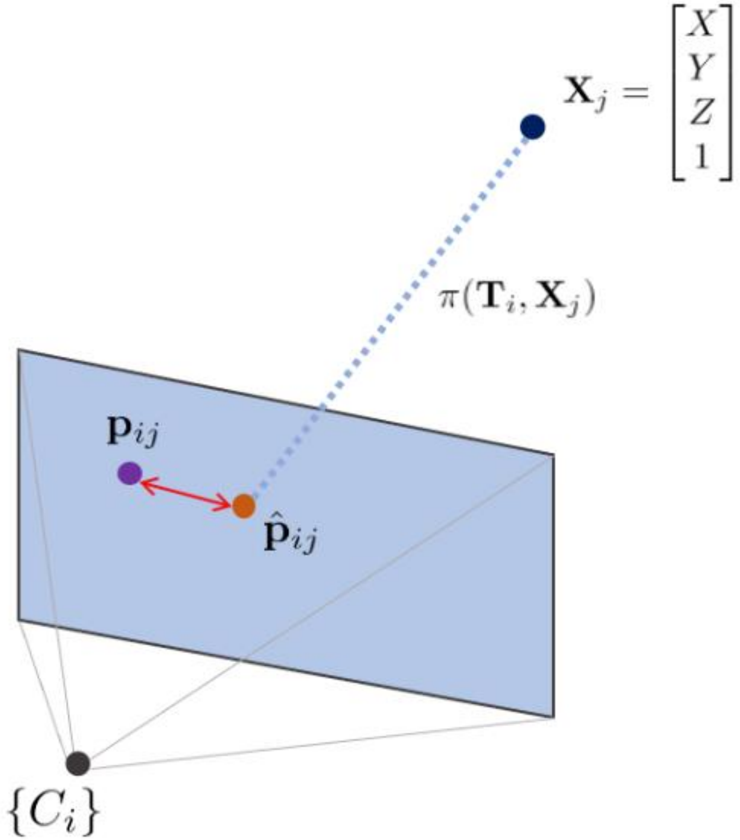
Reprojection error

- Projection model

$$\hat{\mathbf{p}}_{ij} = \pi(\mathbf{T}_i, \mathbf{X}_j)$$

  - $\hat{p}_{ij}$ is a projected point of the j-th 3D point to the i-th camera

  - p_{ij} is the corresponding feature of the j-th 3D point on the i-th image



Reprojection error

- Undistortion
  - All operations assume linearity.
  - Correct for camera distortion during projection to the image plane

$$\mathbf{X}_j \quad [X_j, Y_j, Z_j, 1]^\intercal$$

Transformation $\mathbf{T}_i \mathbf{X}_j$

$$\mathbf{X}'_j = \begin{bmatrix} \mathbf{R}_i \mathbf{X}_j + \mathbf{t}_i \\ 1 \end{bmatrix} = [X'_j, Y'_j, Z'_j, 1]^\intercal$$

Non-homogeneous $\tilde{\mathbf{p}} = \pi_h(\cdot)$

$$\tilde{\mathbf{p}}_{ij} = [X'_j/Z'_j,\ Y'_j/Z'_j,\ 1]^\intercal = [\tilde{u}_{ij}, \tilde{v}_{ij}, 1]^\intercal$$

Undistort

$$\begin{cases} \bar{u}_{ij} = \tilde{u}_{ij}(1 + k_1 r^2 + k_2 r^4) \\ \bar{v}_{ij} = \tilde{v}_{ij}(1 + k_1 r^2 + k_2 r^4) \end{cases}$$

Project to Image Plane $\hat{\mathbf{p}} = \pi_k(\cdot)$

$$\begin{cases} u_{ij} = f_x \bar{u}_{ij} + c_x \\ v_{ij} = f_y \bar{v}_{ij} + c_y \end{cases}$$

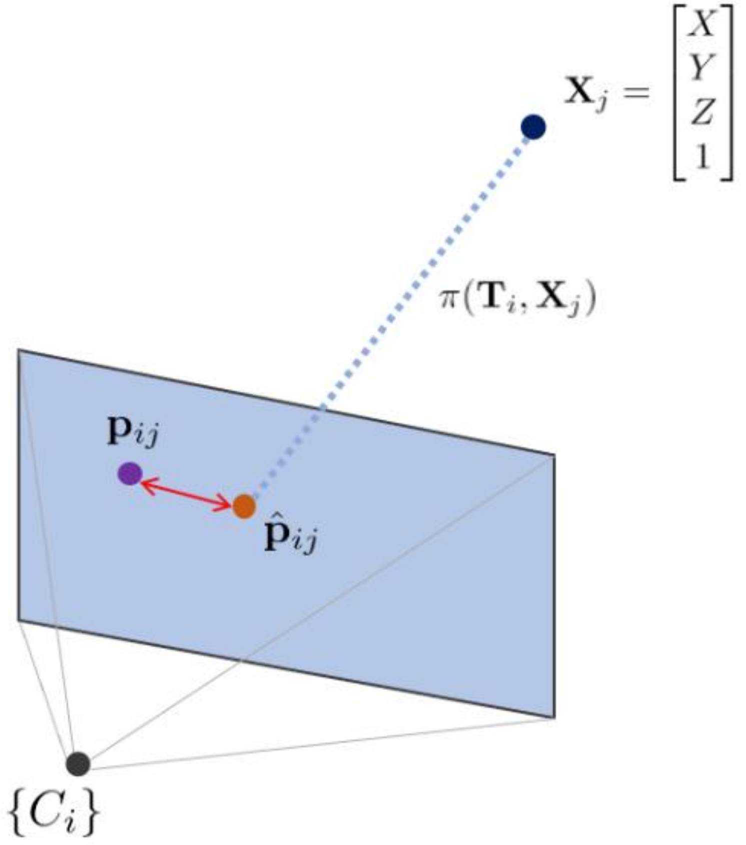
$$\hat{\mathbf{p}}_{ij} \quad [u_{ij}, v_{ij}]^\intercal$$

Reprojection error

- Reprojection error
  - Distance between projected points and feature points

$$\boxed{\begin{aligned}\mathbf{e}_{ij} &= \mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij} \\ &= \mathbf{p}_{ij} - \pi(\mathbf{T}_i, \mathbf{X}_j) \\ &= \mathbf{p}_{ij} - \pi_k(\pi_h(\mathbf{T}_i \mathbf{X}_j))\end{aligned}}$$

$$\begin{aligned}\mathbf{E}(\mathcal{X}) &= \arg\min_{\mathcal{X}^*} \sum_i \sum_j \|\mathbf{e}_{ij}\|^2 \\ &= \arg\min_{\mathcal{X}^*} \sum_i \sum_j \mathbf{e}_{ij}^\intercal \mathbf{e}_{ij} \\ &= \arg\min_{\mathcal{X}^*} \sum_i \sum_j (\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij})^\intercal (\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij})\end{aligned}$$



Reprojection error

- Example
  - Camera: 1~C
  - 3D Points: 1~P

$$E(\mathcal{X}) = \arg\min_{\mathcal{X}^*} \|e(\mathcal{X})\|^2$$

$$e(\mathcal{X}) = e_{11}^2 + e_{12}^2 + \cdots + e_{1p}^2 + e_{21}^2 + \cdots e_{cp}^2$$

$$e(\mathcal{X}) = \left(p_{11} - \pi_k\left(\pi_h(T_{1,} X_1)\right)\right)^2 + \left(p_{12} - \pi_k\left(\pi_h(T_{1,} X_2)\right)\right)^2 + \cdots + \left(p_{cp} - \pi_k\left(\pi_h(T_{c,} X_p)\right)\right)^2$$

$$e(\mathcal{X}) = \left(\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \begin{bmatrix} f_x \tilde{u} + c_x \\ f_y \tilde{v} + c_y \\ 1 \end{bmatrix}\right)^2 + \ldots$$

$$e(\mathcal{X}) = \left(\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \begin{bmatrix} f_x\left((T_1X_1)[1]/(T_1X_1)[3]\right) + c_x \\ f_y\left((T_1X_1)[2]/(T_1X_1)[3]\right) + c_y \\ 1 \end{bmatrix}\right)^2 + \ldots$$

$(X)[n]$: the n-th row vector of X

Reprojection error

- Bundle adjustment
  - Minimize the reprojection error by adjusting both 3D points and camera poses (sometimes intrinsic parameters)

  - Non-linear optimization
    - Gradient descent
    - Newton Method
    - Gauss-Newton
    - Levenberg-Marquardt

Warm-up

- Gradient
  - Derivative to the multivariate functions

$$\nabla f = \left[\frac{\partial f(x_1,x_2,x_3)}{\partial x_1}, \frac{\partial f(x_1,x_2,x_3)}{\partial x_2}, \frac{\partial f(x_1,x_2,x_3)}{\partial x_3}\right]$$

- Jacobian
  - The matrix of all its first-order partial derivatives

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

- Hessian
  - Second-order partial derivatives

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Reprojection error

- Non-linear least squares
  - Update small increments $\Delta\mathcal{X}$ to $\mathcal{X}$ iteratively.

$$\mathbf{E}(\mathcal{X}+\Delta\mathcal{X}) = \arg\min_{\mathcal{X}^*} \sum_i \sum_j \|\mathbf{e}(\mathcal{X}+\Delta\mathcal{X})\|^2$$

- Using the Taylor first-order approximation, this can be expressed as

$$\begin{aligned} \mathbf{e}(\mathcal{X}+\Delta\mathcal{X}) &\approx \mathbf{e}(\mathcal{X}) + \mathbf{J}\Delta\mathcal{X} \\ &= \mathbf{e}(\mathcal{X}) + \mathbf{J}_c\Delta\mathcal{T} + \mathbf{J}_p\Delta\mathbf{X} \\ &= \mathbf{e}(\mathcal{X}) + \frac{\partial\mathbf{e}}{\partial\mathcal{T}}\Delta\mathcal{T} + \frac{\partial\mathbf{e}}{\partial\mathbf{X}}\Delta\mathbf{X} \end{aligned}$$

$$\mathbf{E}(\mathcal{X}+\Delta\mathcal{X}) \approx \arg\min_{\mathcal{X}^*} \sum_i \sum_j \|\mathbf{e}(\mathcal{X}) + \mathbf{J}\Delta\mathcal{X}\|^2$$

Reprojection error

- non-linear least squares

$$\mathbf{E}(\mathcal{X}+\Delta\mathcal{X}) \approx \arg\min_{\mathcal{X}^*} \sum_i \sum_j \|\mathbf{e}(\mathcal{X}) + \mathbf{J}\Delta\mathcal{X}\|^2$$

$$= (\mathrm{e} + \mathrm{J}\Delta\mathrm{x})^\top (\mathrm{e} + \mathrm{J}\Delta\mathrm{x})$$

$$= \mathrm{e}^\top \mathrm{e} + 2\mathrm{e}\mathrm{J}\Delta\mathrm{x} + \Delta\mathrm{x} J^\top J \Delta\mathrm{x}$$

$$H = J^\top J$$

$$= a + 2\mathrm{b}\Delta\mathrm{x} + \Delta\mathrm{x} H \Delta\mathrm{x}$$

Quadratic equation for $\Delta\mathrm{x}$

H is positive definite

→ convex

$$\frac{\partial \mathbf{E}(\mathbf{x}+\Delta\mathbf{x})}{\partial \Delta\mathbf{x}} \approx 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x} = 0$$

$$\mathbf{H}\Delta\mathbf{x} = -\mathbf{b}$$

$$\Delta\mathcal{X}^* = -\mathbf{H}^{-1}\mathbf{b}$$

Jacobian of state

- State variable

$$\mathcal{X} = [\mathcal{X}_c \ \mathcal{X}_p]^T \text{ where } \begin{aligned} \mathcal{X}_c &= [\mathcal{T}_1, \cdots, \mathcal{T}_m]^T \in \mathbb{R}^{6m} \\ \mathcal{X}_p &= [\mathbf{X}_1, \cdots, \mathbf{X}_n]^T \in \mathbb{R}^{4n} \end{aligned}$$

- Jacobian

$$\mathbf{J} = [\mathbf{J_c} \ \mathbf{J_p}]$$

$$\mathbf{J} = \begin{array}{c} \\ 2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 3 & 3 & 3 \end{array} \\ \left[\underbrace{\frac{\partial \mathbf{e}}{\partial \mathbf{R}} \quad \frac{\partial \mathbf{e}}{\partial \mathbf{t}}}_{\mathbf{J_c}} \quad \underbrace{\frac{\partial \mathbf{e}}{\partial \mathbf{X}}}_{\mathbf{J_p}}\right] \end{array}$$

$\mathrm{J_c}: \frac{\partial e}{\partial \mathcal{T}}$ camera pose components

$\mathrm{J_P}: \frac{\partial e}{\partial \mathrm{X}}$ 3D points components

- Hessian

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} = \begin{bmatrix} \mathbf{J_c}^T\mathbf{J_c} & \mathbf{J_c}^T\mathbf{J_p} \\ \mathbf{J_p}^T\mathbf{J_c} & \mathbf{J_p}^T\mathbf{J_p} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cp} \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix}$$

Appendix

$$
\mathbf{J}_c = \frac{\partial \hat{\mathbf{p}}}{\partial \tilde{\mathbf{p}}} \frac{\partial \tilde{\mathbf{p}}}{\partial \mathbf{X}'} \frac{\partial \mathbf{X}'}{\partial [\Delta \mathbf{w}, \mathbf{t}]}
$$

$$
= \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \end{bmatrix} \begin{bmatrix} \frac{1}{Z'} & 0 & \frac{-X'}{Z'^2} & 0 \\ 0 & \frac{1}{Z'} & \frac{-Y'}{Z'^2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & Z' & -Y' & 1 & 0 & 0 \\ -Z' & 0 & X' & 0 & 1 & 0 \\ Y' & -X' & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} \frac{f}{Z'} & 0 & -\frac{fX}{Z'^2} & 0 \\ 0 & \frac{f}{Z'} & -\frac{fY}{Z'^2} & 0 \end{bmatrix} \begin{bmatrix} 0 & Z' & -Y' & 1 & 0 & 0 \\ -Z' & 0 & X' & 0 & 1 & 0 \\ Y' & -X' & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} -\frac{fX'Y'}{Z'^2} & \frac{f(1+X'^2)}{Z'^2} & -\frac{fY'}{Z'} & \frac{f}{Z'} & 0 & -\frac{fX'}{Z'^2} \\ -\frac{f(1+y^2)}{Z'^2} & \frac{fX'Y'}{Z'^2} & \frac{fX'}{Z'} & 0 & \frac{f}{Z'} & -\frac{fY'}{Z'^2} \end{bmatrix} \in \mathbb{R}^{2\times 6}
$$

Appendix

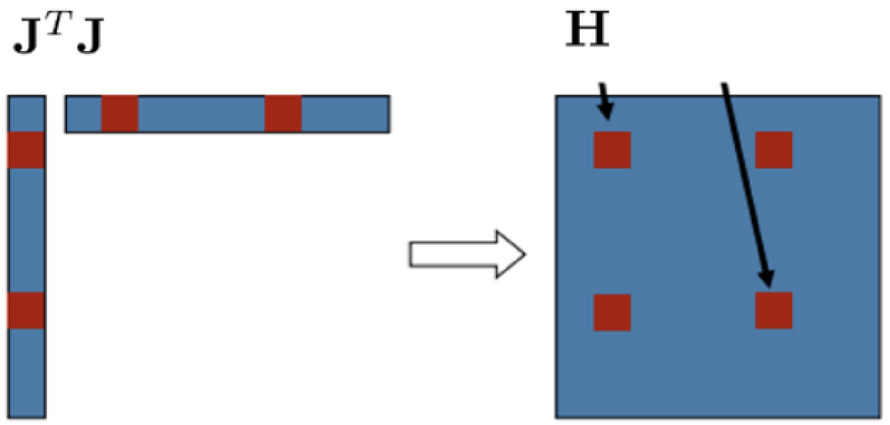
$$\mathbf{J}_p = \begin{bmatrix} \frac{f}{Z'} & 0 & -\frac{fX'}{Z'^2} \\ 0 & \frac{f}{Z'} & -\frac{fY'}{Z'^2} \end{bmatrix} \mathbf{R} \in \mathbb{R}^{2\times 3}$$

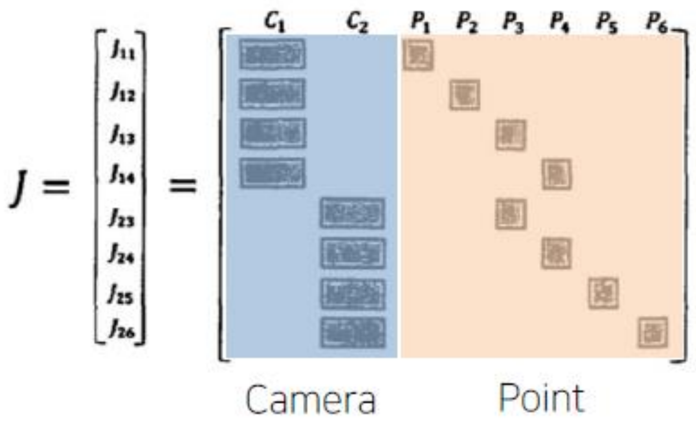
Multiple cameras and 3D points

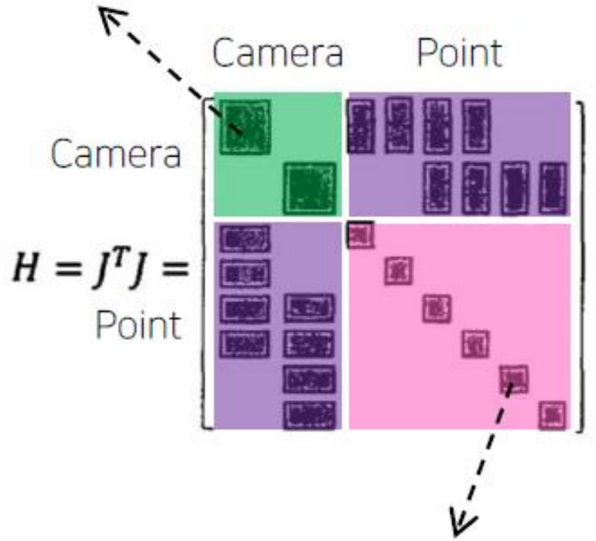
- Jacobian

$$\mathbf{J}_{ij} = \left(\mathbf{0}, \cdots, \mathbf{0}, \frac{\partial \mathbf{e}_{ij}}{\partial \mathcal{T}_i}, \mathbf{0}, \cdots, \mathbf{0}, \frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{X}_j}, \mathbf{0}, \cdots, \mathbf{0}\right)$$

- Hessian



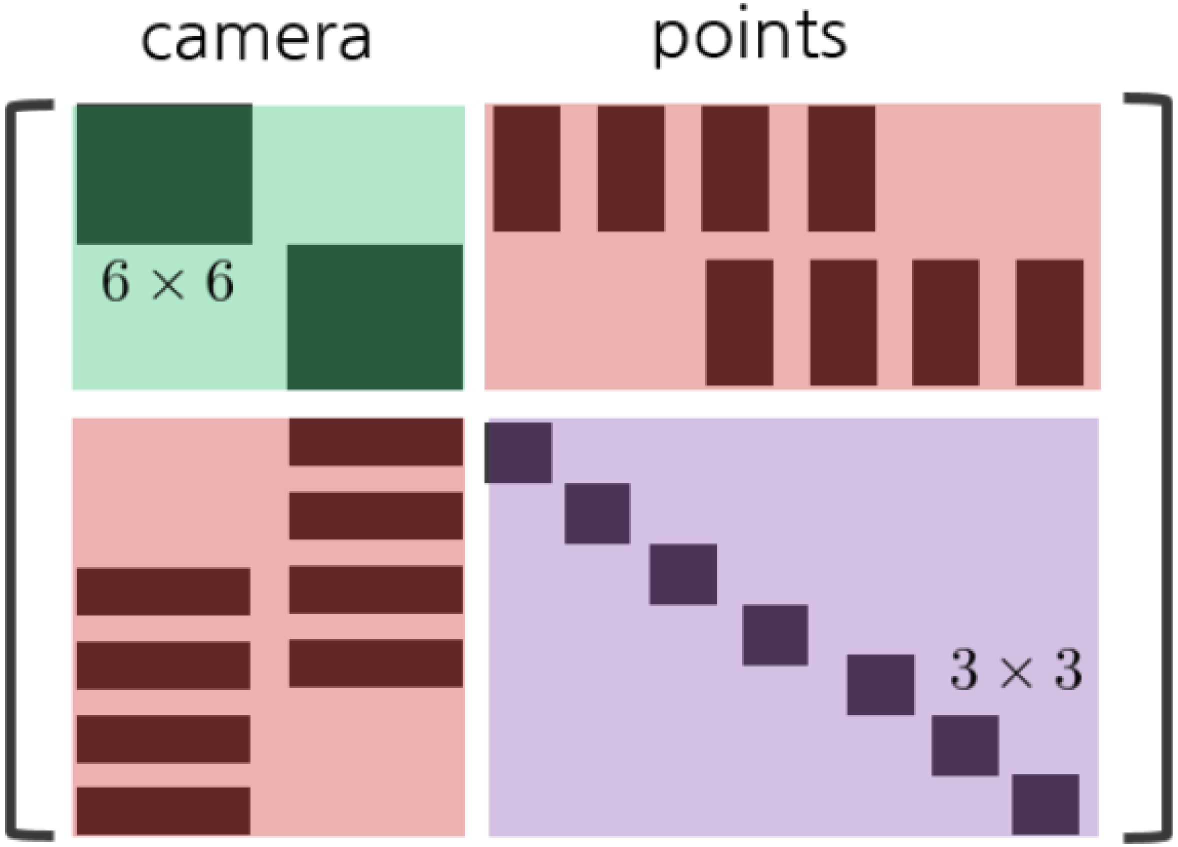




Multiple cameras and 3D points

- Hessian

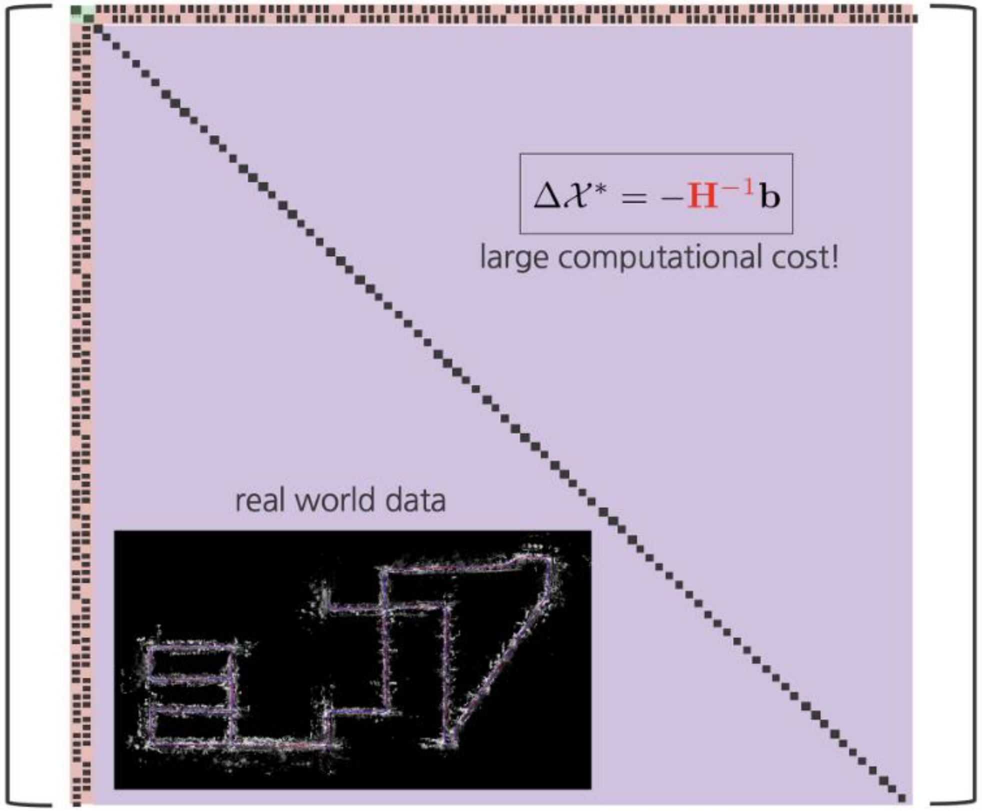
$$\mathbf{H} = \mathbf{J}^T\mathbf{J} = \begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cp} \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} =$$



Multiple cameras and 3D points

- Hessian
  - General case : The number of 3D points are much larger than cameras
  - Compute inverse H is very computationally intensive.

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} = \begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cp} \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} =$$



Schur complement

- Schur complement allow us to derive several useful formulae for the inversion and the factorization of the block matrix
- Using the Schur complement reduce the complexity
  - Compute $\Delta\mathcal{X}_c, \Delta\mathcal{X}_P$ sequentially

$$\mathbf{H}\Delta\mathcal{X}^* = -\mathbf{b}$$

$$\begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cp} \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} \begin{bmatrix} \Delta\mathcal{X}_c \\ \Delta\mathcal{X}_p \end{bmatrix} = \begin{bmatrix} \mathbf{b}_c \\ \mathbf{b}_p \end{bmatrix}$$

Schur complement

- Forward substitution

$$\begin{bmatrix} \mathbf{I} & -\mathbf{H}_{cp}\mathbf{H}_{pp}^{-1} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cp} \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} \begin{bmatrix} \Delta\mathcal{X}_c \\ \Delta\mathcal{X}_p \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{H}_{cp}\mathbf{H}_{pp}^{-1} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{b}_c \\ \mathbf{b}_p \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{H}_{cc} - \mathbf{H}_{cp}\mathbf{H}_{pp}^{-1}\mathbf{H}_{pc} & 0 \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} \begin{bmatrix} \Delta\mathcal{X}_c \\ \Delta\mathcal{X}_p \end{bmatrix} = \begin{bmatrix} \mathbf{b}_c - \mathbf{H}_{cp}\mathbf{H}_{pp}^{-1}\mathbf{b}_p \\ \mathbf{b}_p \end{bmatrix}$$

$$\mathbf{H}_m \Delta\mathcal{X}_c = \mathbf{b}_m$$

- $\mathbf{H}_m = \mathbf{H}_{cc} - \mathbf{H}_{cp}\boxed{\mathbf{H}_{pp}^{-1}}\mathbf{H}_{pc}$
- $\mathbf{b}_m = \mathbf{b}_c - \mathbf{H}_{cp}\mathbf{H}_{pp}^{-1}\mathbf{b}_p$

Schur complement

- Backward substitution

$$\begin{bmatrix} \mathbf{H}_{cc} - \mathbf{H}_{cp}\mathbf{H}_{pp}^{-1}\mathbf{H}_{pc} & 0 \\ \mathbf{H}_{pc} & \mathbf{H}_{pp} \end{bmatrix} \begin{bmatrix} \Delta\mathcal{X}_c \\ \Delta\mathcal{X}_p \end{bmatrix} = \begin{bmatrix} \mathbf{b}_c - \mathbf{H}_{cp}\mathbf{H}_{pp}^{-1}\mathbf{b}_p \\ \mathbf{b}_p \end{bmatrix}$$

$$\Delta\mathcal{X}_p = \mathbf{H}_{pp}^{-1}(\mathbf{b}_p - \mathbf{H}_{pc}\Delta\mathcal{X}_c)$$

Thank you
for this semester