Part VI

3D Structure and Camera Motion

1 Introduction

2 Reconstructing Camera Systems

3 Bundle Adjustment

covered by

[1] [H&Z] Secs: 9.5.3, 10.1, 10.2, 10.3, 12.1, 12.2, 12.4, 12.5, 18.1

[2] Triggs, B. et al. Bundle Adjustment—A Modern Synthesis. In *Proc ICCV Workshop on Vision Algorithms.* Springer-Verlag. pp. 298–372, 1999.

►Constructing Cameras from the Fundamental Matrix

Given $\mathbf{F}$, construct some cameras $\mathbf{P}_1$, $\mathbf{P}_2$ such that $\mathbf{F}$ is their fundamental matrix.

Solution

See [H&Z, p. 256]

$$\mathbf{P}_1 = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix}[\underline{\mathbf{e}}_2]_\times \mathbf{F} + \underline{\mathbf{e}}_2 \, \underline{\mathbf{v}}^\top & \lambda \, \underline{\mathbf{e}}_2\end{bmatrix}$$

$v = e_1$

$\left([e_2]_\times F + e_2 e_1^T\right) e_1 =$ ~~if $v = 0$~~ $= e_2 \|e_1\|^2$

where

- $\underline{\mathbf{v}}$ is any 3-vector, e.g. $\underline{\mathbf{v}} = \underline{\mathbf{e}}_1$ to make the camera finite
- $\lambda \neq 0$ is a scalar,
- $\underline{\mathbf{e}}_2 = \mathrm{null}(\mathbf{F}^\top)$, i.e. $\underline{\mathbf{e}}_2^\top \mathbf{F} = 0$

Proof

1. $\mathbf{S}$ is antisymmetric iff $\mathbf{x}^\top \mathbf{S} \mathbf{x} = 0$ for all $\mathbf{x}$ — look-up the proof!
2. we have $\underline{\mathbf{x}} \simeq \mathbf{P} \underline{\mathbf{X}}$
3. a non-zero $\mathbf{F}$ is a f.m. iff $\mathbf{P}_2^\top \mathbf{F} \mathbf{P}_1$ is antisymmetric
4. if $\mathbf{P}_1 = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$ and $\mathbf{P}_2 = \begin{bmatrix}\mathbf{S}\mathbf{F} & \underline{\mathbf{e}}_2\end{bmatrix}$ then $\mathbf{F}$ corresponds to $(\mathbf{P}_1, \mathbf{P}_2)$ by Step 3
5. we can write $\mathbf{S} = [\mathbf{s}]_\times$
6. a suitable choice is $\mathbf{s} = \underline{\mathbf{e}}_2$ — [Luong96]
7. for the full the class including $\underline{\mathbf{v}}$, see [H&Z, Sec. 9.5]

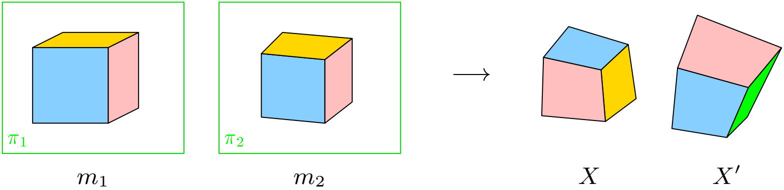
▶The Projective Reconstruction Theorem

Observation: Unless $\mathbf{P}_i$ are constrained, then for any number of cameras $i = 1, \ldots, k$

$$\underline{\mathbf{m}}_i = \mathbf{P}_i \underline{\mathbf{X}} = \underbrace{\mathbf{P}_i \mathbf{H}^{-1}}_{\mathbf{P}'_i} \underbrace{\mathbf{H} \underline{\mathbf{X}}}_{\underline{\mathbf{X}}'} = \mathbf{P}'_i \underline{\mathbf{X}}'$$

motion → $\mathbf{P}_i$, structure → $\underline{\mathbf{X}}$

- when $\mathbf{P}_i$ and $\underline{\mathbf{X}}$ are both determined from correspondences (including calibrations $\mathbf{K}_i$), they are given up to a common 3D homography $\mathbf{H}$
  (translation, rotation, scale, shear, pure perspectivity)

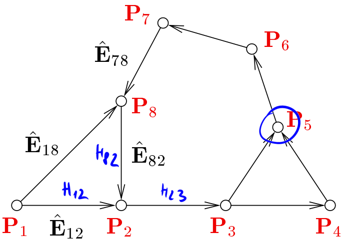


- when cameras are internally calibrated ($\mathbf{K}_i$ known) then $\mathbf{H}$ is restricted to a similarity since it must preserve the calibrations $\mathbf{K}_i$ [H&Z, Secs. 10.2, 10.3], [Longuet & Higgins 81]
  (translation, rotation, scale)

▶Reconstructing Camera Systems

Problem: Given a set of p decomposed pairwise essential matrices $\hat{\mathbf{E}}_{ij} = [\hat{\mathbf{t}}_{ij}]_\times \hat{\mathbf{R}}_{ij}$ and calibration matrices $\mathbf{K}_i$ reconstruct the camera system $\mathbf{P}_i$, $i = 1, \ldots, k$

→ Slides 78 and 138 on representing $\mathbf{E}$

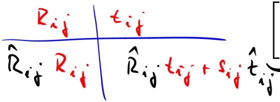


We construct camera pairs $\hat{\mathbf{P}}_{ij} \in \mathbb{R}^{6,4}$ → Slide 123

$$\hat{\mathbf{P}}_{ij} = \begin{bmatrix} \hat{\mathbf{P}}_i \\ \hat{\mathbf{P}}_j \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{K}}_i \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \\ \hat{\mathbf{K}}_j \begin{bmatrix} \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{6,4}$$

- singletons i, j correspond to vertices V — k vertices
- pairs ij correspond to graph edges E — p edges

$\hat{\mathbf{P}}_{ij}$ are in different coordinate systems but these are related by similarities $\hat{\mathbf{P}}_{ij}\mathbf{H}_{ij} = \mathbf{P}_{ij}$



$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix}}_{\mathbb{R}^{6,4}} \underbrace{\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & s_{ij} \end{bmatrix}}_{\mathbf{H}_{ij} \in \mathbb{R}^{4,4}} \overset{!}{=} \underbrace{\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{R}_j & \mathbf{t}_j \end{bmatrix}}_{\mathbb{R}^{6,4}} \qquad (24)$$

- $\mathbf{K}_i$ removed on both sides of eq. (24)
- (24) is a linear system of $24p$ eqs. in $7p + 6k$ unknowns — $7p \sim (\mathbf{t}_{ij}, \mathbf{R}_{ij}, s_{ij})$, $6k \sim (\mathbf{R}_i, \mathbf{t}_i)$
- each $\mathbf{P}_i$ appears on the right side as many times as is the degree of vertex $\mathbf{P}_i$ — eg. P_5 3-times

▶cont'd

Eq. (24) implies $\begin{bmatrix} \mathbf{R}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{R}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i \\ \mathbf{R}_j \end{bmatrix} \qquad \begin{bmatrix} \mathbf{t}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{t}_{ij} + s_{ij}\hat{\mathbf{t}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_j \end{bmatrix}$

- $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$ can be eliminated:

$$\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j, \qquad \hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} = \mathbf{t}_j, \qquad s_{ij} > 0 \tag{25}$$

- note transformations that do not change these equations — assuming no error in $\hat{\mathbf{R}}_{ij}$

  1. $\mathbf{R}_i \mapsto \mathbf{R}_i\mathbf{R}$, 2. $\mathbf{t}_i \mapsto \sigma\,\mathbf{t}_i$ and $s_{ij} \mapsto \sigma s_{ij}$, 3. $\mathbf{t}_i \mapsto \mathbf{t}_i + \mathbf{R}_i\mathbf{t}$

- the global frame is fixed by e.g. selecting

$$\mathbf{R}_1 = \mathbf{I}, \qquad \sum_{i=1}^{k} \mathbf{t}_i = \mathbf{0}, \qquad \frac{1}{p}\sum_{i,j} s_{ij} = 1 \tag{26}$$

- rotation equations are decoupled from translation equations
- in principle, s_{ij} could correct the sign of $\hat{\mathbf{t}}_{ij}$ from essential matrix decomposition Slide 78
  but $\mathbf{R}_i$ cannot correct the α sign in $\hat{\mathbf{R}}_{ij}$
  → therefore make sure all points are in front of cameras and constrain $s_{ij} > 0$; see Slide 80

+ pairwise correspondences are sufficient
− suitable for well-located cameras only (dome-like configurations)

otherwise intractable or numerically unstable

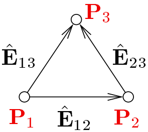
Finding The Rotation Component in Eq. (25)

Task: Solve $\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j$, $i, j \in V$, $(i,j) \in E$ where $\mathbf{R}$ are a 3×3 rotation matrix each.
Per columns $c = 1, 2, 3$ of $\mathbf{R}_j$:

$$\hat{\mathbf{R}}_{ij}\mathbf{r}_i^c - \mathbf{r}_j^c = \mathbf{0}, \qquad \text{for all } i,\ j \tag{27}$$

- fix c and denote $\mathbf{r}^c = \left[\mathbf{r}_1^c, \mathbf{r}_2^c, \dots, \mathbf{r}_k^c\right]^\top$ c-th columns of all rotation matrices stacked; $\mathbf{r}^c \in \mathbb{R}^{3k}$
- then (27) becomes $\mathbf{D}\,\mathbf{r}^c = \mathbf{0}$ $\quad \mathbf{D} \in \mathbb{R}^{3p,3k}$
- $3p$ equations for $3k$ unknowns $\to p \geq k$ $\quad$ in a 1-connected graph we have to fix $\mathbf{r}_1^c = [1, 0, 0]$

Ex: $(k = p = 3)$



$\rightarrow$

$$\hat{\mathbf{R}}_{12}\mathbf{r}_1^c - \mathbf{r}_2^c = \mathbf{0}$$
$$\hat{\mathbf{R}}_{23}\mathbf{r}_2^c - \mathbf{r}_3^c = \mathbf{0}$$
$$\hat{\mathbf{R}}_{13}\mathbf{r}_1^c - \mathbf{r}_3^c = \mathbf{0}$$

$\rightarrow$

$$\mathbf{D}\,\mathbf{r}^c = \begin{bmatrix} \hat{\mathbf{R}}_{12} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_{23} & -\mathbf{I} \\ \hat{\mathbf{R}}_{13} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^c \\ \mathbf{r}_2^c \\ \mathbf{r}_3^c \end{bmatrix} = \mathbf{0}$$

- must hold for any c

Idea: [Martinec & Pajdla CVPR 2007]

1. find the space of all $\mathbf{r}^c \in \mathbb{R}^{3k}$ that solve (27) $\quad \mathbf{D}$ is sparse, use `[V,E] = eigs(D'*D,3,0);` (Matlab)
2. choose 3 unit orthogonal vectors in this space $\quad$ 3 smallest eigenvectors
3. find closest rotation matrices per cam. using SVD $\quad$ because $\|\mathbf{r}^c\| = 1$ is necessary but insufficient
   $\mathbf{R}_i^* = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{R}_i = \mathbf{U}\mathbf{D}\mathbf{V}^\top$

- global world rotation is arbitrary

Finding The Translation Component in Eq. (25)

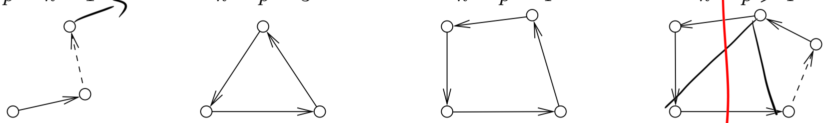
From eqs. (25) and (26): d – rank of camera center set p – No. of pairs, k – No. of cameras

$$\hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} - \mathbf{t}_j = \mathbf{0}, \qquad \sum_{i=1}^{k} \mathbf{t}_i = \mathbf{0}, \qquad \sum_{i,j} s_{ij} = p, \qquad s_{ij} > 0, \qquad \mathbf{t}_i \in \mathbb{R}^d$$

- in rank d: $d \cdot p + d + 1$ equations for $d \cdot k + p$ unknowns $\to p \geq \frac{d(k-1)-1}{d-1}$

Ex: Chains and circuits construction from sticks of known orientation and unknown length?

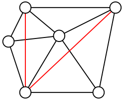
$p = k - 1$ | $k = p = 3$ | $k = p = 4$ | $k = p > 4$



$k \leq 2$ for any d | $d \geq 2$: non-collinear ok | $d \geq 3$: non-planar ok | $d \geq k - 1$: not possible

without chords

- rank is not sufficient for chains, trees, or when $d = 1$ (collinear cameras)
- 3-connectivity gives a sufficient rank for $d = 3$ (cams. in general pos. in 3D)
  - s-connected graph has $p \geq \lceil \frac{sk}{2} \rceil$ edges for $s \geq 2$, hence $p \geq \lceil \frac{3k}{2} \rceil \geq \frac{3k}{2} - 2$
- 4-connectivity gives a sufficient rank for any k for $d = 2$ (coplanar cams)
  - since $p \geq \lceil 2k \rceil \geq 2k - 3$
  - maximal planar (tringulated) graphs have $p = 3k - 6$ and give the rank for $k \geq 3$



cont'd

Linear equations in (25) and (26) can be rewritten to

$$\mathbf{D}\mathbf{t} = \mathbf{0}, \qquad \mathbf{t} = \left[\mathbf{t}_1^\top, \mathbf{t}_2^\top, \dots, \mathbf{t}_k^\top, s_{12}, \dots, s_{ij}, \dots\right]^\top$$

for $d = 3$: $\mathbf{t} \in \mathbb{R}^{3k+p}$, $\mathbf{D} \in \mathbb{R}^{3p,3k+p}$ is sparse

$$\mathbf{t}^* = \arg\min_{\mathbf{t},\, s_{ij}>0} \mathbf{t}^\top \mathbf{D}^\top \mathbf{D}\, \mathbf{t}$$

- this is a quadratic programming problem (constraints!)

```
z = zeros(3*k+p,1);
t = quadprog(D'*D, z, diag([zeros(3*k,1); -ones(p,1)]), z);
```

- but check the rank first!

▶Solving Eq. (25) by Stepwise Gluing

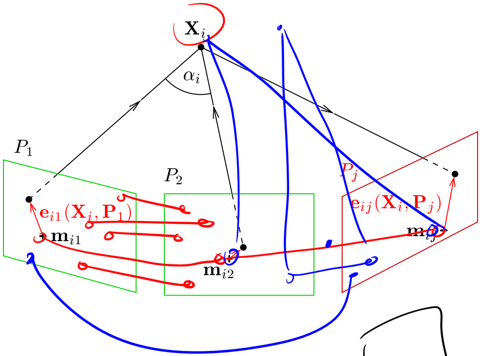
Given: Calibration matrices $\mathbf{K}_j$ and tentative correspondences per camera triples.

Initialization

1. initialize camera cluster $\mathcal{C}$ with P_1, P_2,
2. find essential matrix $\mathbf{E}_{12}$ and matches M_{12} by the 5-point algorithm Slide 84
3. construct camera pair

$$\mathbf{P}_1 = \mathbf{K}_1 \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \ \mathbf{P}_2 = \mathbf{K}_2 \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

4. compute 3D reconstruction $\{X_i\}$ per match from M_{12} Slide 90
5. initialize point cloud $\mathcal{X}$ with $\{X_i\}$ satisfying chirality constraint $z_i > 0$ and apical angle constraint $|\alpha_i| > \alpha_T$



Attaching camera $P_j \notin \mathcal{C}$

1. select points $\mathcal{X}_j$ from $\mathcal{X}$ that have matches to P_j
2. estimate $\mathbf{P}_j$ using $\mathcal{X}_j$, RANSAC with the 3-pt alg. (P3P), projection errors $\mathbf{e}_{ij}$ in $\mathcal{X}_j$ Slide 69
3. reconstruct 3D points from all tentative matches from P_j to all P_l, $l \neq k$ that are not in $\mathcal{X}$
4. filter them by the chirality and apical angle constraints and add them to $\mathcal{X}$
5. add P_j to $\mathcal{C}$
6. perform bundle adjustment on $\mathcal{X}$ and $\mathcal{C}$ coming next

▶Bundle Adjustment

Given:

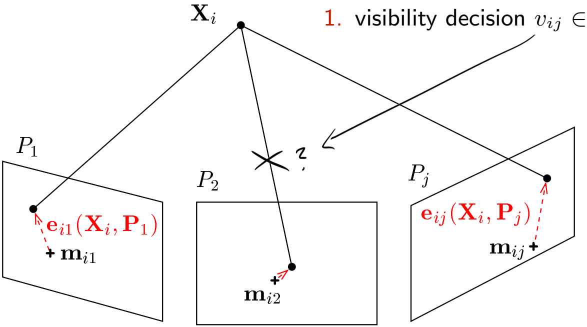
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^{p}$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^{c}$
3. fixed tentative projections $\mathbf{m}_{ij}$

Required:

1. corrected 3D points $\{\mathbf{X}'_i\}_{i=1}^{p}$
2. corrected cameras $\{\mathbf{P}'_j\}_{j=1}^{c}$

Latent:

1. visibility decision $v_{ij} \in \{0, 1\}$ per $\mathbf{m}_{ij}$



- for simplicity, $\mathbf{X}$, $\mathbf{m}$ are considered direct (not homogeneous)
- we have projection error $\mathbf{e}_{ij}(\mathbf{X}_i, \mathbf{P}_j) = \mathbf{x}_i - \mathbf{m}_i$ per image feature, where $\underline{\mathbf{x}}_i = \mathbf{P}_j \underline{\mathbf{X}}_i$
- for simplicity, we will work with scalar error $e_{ij} = \|\mathbf{e}_{ij}\|$

Robust Objective Function for Bundle Adjustment

Data likelihood is constructed by marginalization, as in Robust Matching Model, Slide 107

$$p(\{\mathbf{m}\} \mid \{\mathbf{P}\}) = \prod_{\text{pts:}i=1}^{p} \prod_{\text{cams:}j=1}^{c} \Big((1-\alpha_0) p_1(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) + \alpha_0\, p_0(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) \Big)$$

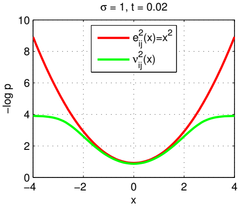
the simplified log-likelihood is (as on Slide 108)

$$V(\{\mathbf{m}\} \mid \{\mathbf{P}\}) = -\log p(\{\mathbf{m}\} \mid \{\mathbf{P}\}) = \sum_i \sum_j \underbrace{-\log\left(e^{-\frac{e_{ij}^2(\mathbf{X}_i,\mathbf{P}_j)}{2\sigma_1^2}} + t \right)}_{\rho(e_{ij}^2(\mathbf{X}_i,\mathbf{P}_j)) = \nu_{ij}^2(\mathbf{X}_i,\mathbf{P}_j)} \stackrel{\text{def}}{=} \sum_i \sum_j \nu_{ij}^2(\mathbf{X}_i, \mathbf{P}_j)$$

- ν_{ij} is a 'robust' error fcn.; it is non-robust ($\nu_{ij} = e_{ij}$) when $t = 0$
- $\rho(\cdot)$ is a 'robustification function' we often find in M-estimation
- the $\mathbf{L}_{ij}$ in Levenberg-Marquardt changes to vector

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t\, e^{e_{ij}^2(\theta)/(2\sigma_1^2)}}}_{\text{small for big } e_{ij}} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \qquad (28)$$

but the LM method stays the same as on Slides 101–102



- outliers have virtually no impact on $\mathbf{d}_s$ in normal equations because of the red term in (28) that scales contributions to the sums down

$$-\sum_{i,j} \mathbf{L}_{ij}^\top \nu_{ij}(\theta^s) = \Big(\sum_{i,j}^{k} \mathbf{L}_{ij}^\top \mathbf{L}_{ij} \Big) \mathbf{d}_s$$

▶Sparsity in Bundle Adjustment

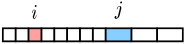
We have $q = 3p + 11c$ parameters: $\theta = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_c)$ points, cameras

We will use a running index $r = 1, \ldots, k$, $k = p \cdot c$. Then   each r corresponds to some i, j

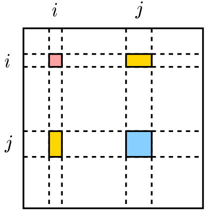
$$\theta^* = \arg\min_\theta \sum_{r=1}^{k} \nu_r^2(\theta),\ \boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s,\ -\sum_{r=1}^{k} \mathbf{L}_r^\top \nu_r(\theta^s) = \left(\sum_{r=1}^{k} \mathbf{L}_r^\top \mathbf{L}_r + \lambda \operatorname{diag} \mathbf{L}_r^\top \mathbf{L}_r \right) \mathbf{d}_s$$

The block form of $\mathbf{L}_r$ in Levenberg-Marquardt (Slide 101) is zero except in columns i and j:

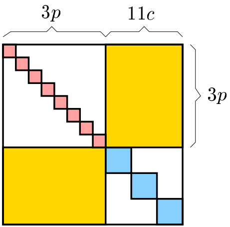
r-th error term is $\nu_r^2 = \rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j))$

$\mathbf{L}_r =$ 

blocks:
- ■: $\mathbf{X}_i, 1 \times 3$
- ■: $\mathbf{P}_j, 1 \times 11$

$\mathbf{L}_r^\top \mathbf{L}_r =$ 

blocks:
- ■: $\mathbf{X}_i \sim \mathbf{X}_i,\ 3 \times 3$
- ■: $\mathbf{X}_i \sim \mathbf{P}_j,\ 3 \times 11$
- ■: $\mathbf{P}_j \sim \mathbf{P}_j,\ 11 \times 11$

$\sum_{r=1}^{k} \mathbf{L}_r^\top \mathbf{L}_r =$ 

- "points first, then cameras" scheme
- standard bundle adjustment eliminates points and solves cameras, then back-substitutes