►Solving Eq. (29) by Stepwise Gluing

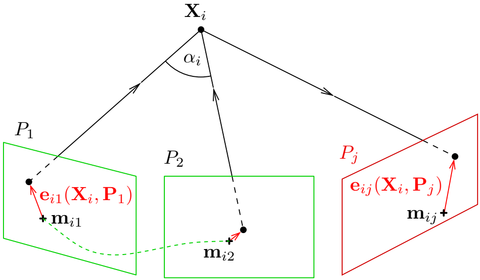
Given: Calibration matrices $\mathbf{K}_j$ and tentative correspondences per camera triples.

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 →87
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} →104
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 $\mathcal{X}_j$ →66
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 →136

▶Bundle Adjustment

Given:

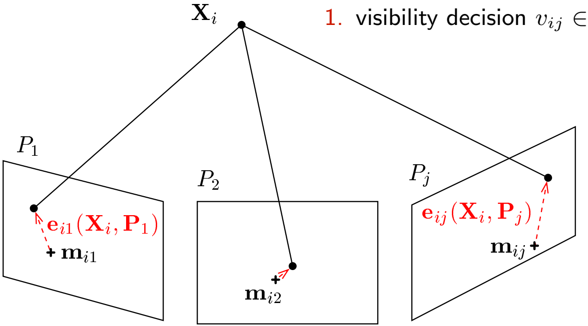
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^{p}$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^{c}$
3. fixed tentative projections $\mathbf{m}_{ij}$

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 Cartesian (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

The data model is constructed by marginalization, as in Robust Matching Model →112

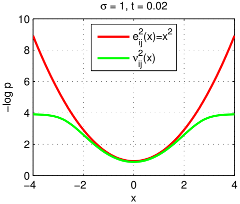
$$p(\{\mathbf{e}\} \mid \{\mathbf{P}, \mathbf{X}\}) = \prod_{\text{pts}:i=1}^{p} \prod_{\text{cams}:j=1}^{c} \Big((1-P_0) p_1(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) + P_0\, p_0(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) \Big)$$

marginalized negative log-density is (→113)

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

- e_{ij} is the projection error (not Sampson error)
- 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

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t\, e^{e_{ij}^2(\theta)/(2\sigma_1^2)}}}_{\text{small for big } e_{ij}} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \qquad (32)$$



but the LM method stays the same as before →106–107

- outliers: almost no impact on $\mathbf{d}_s$ in normal equations because the red term in (32) scales contributions to both sums down for the particular ij

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

▶Sparsity in Bundle Adjustment

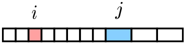
We have $q = 3p + 11k$ parameters: $\theta = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k)$ points, cameras

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

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

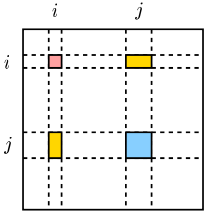
The block form of $\mathbf{L}_r$ in Levenberg-Marquardt (→106) is zero except in columns i and j:

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

$\mathbf{L}_r =$ 

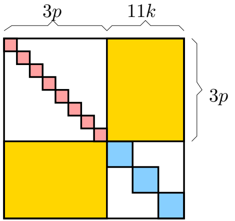
$r = (i, j)$ blocks:

- (pink): $\mathbf{X}_i$, 1×3
- (blue): $\mathbf{P}_j$, 1×11

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

blocks:

- (pink): $\mathbf{X}_i - \mathbf{X}_i$, 3×3
- (yellow): $\mathbf{X}_i - \mathbf{P}_j$, 3×11
- (blue): $\mathbf{P}_j - \mathbf{P}_j$, 11×11

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

- “points first, then cameras” scheme

▸Choleski Decomposition for B. A.

The most expensive computation in B. A. is solving the normal eqs:

$$\text{find } \mathbf{d}_s \text{ such that} \qquad -\sum_{r=1}^{z} \mathbf{L}_r^\top \nu_r(\theta^s) = \Big(\sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r + \lambda \operatorname{diag} \mathbf{L}_r^\top \mathbf{L}_r\Big)\mathbf{d}_s$$

This is a linear set of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, where

- $\mathbf{A}$ is very large — approx. $3 \cdot 10^4 \times 3 \cdot 10^4$ for a small problem of 10000 points and 5 cameras
- $\mathbf{A}$ is sparse and symmetric, $\mathbf{A}^{-1}$ is dense — direct matrix inversion is prohibitive

> Choleski: Every symmetric positive definite matrix $\mathbf{A}$ can be decomposed to $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, where $\mathbf{L}$ is lower triangular. If $\mathbf{A}$ is sparse then $\mathbf{L}$ is sparse, too.

1. decompose $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ — transforms the problem to solving $\mathbf{L}\underbrace{\mathbf{L}^\top\mathbf{x}}_{\mathbf{c}} = \mathbf{b}$
2. solve for $\mathbf{x}$ in two passes:

$$\mathbf{L}\,\mathbf{c} = \mathbf{b} \qquad \mathbf{c}_i := \mathbf{L}_{ii}^{-1}\Big(\mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{c}_j\Big) \qquad \text{forward substitution, } i = 1, \dots, q$$

$$\mathbf{L}^\top\mathbf{x} = \mathbf{c} \qquad \mathbf{x}_i := \mathbf{L}_{ii}^{-1}\Big(\mathbf{c}_i - \sum_{j>i} \mathbf{L}_{ji}\mathbf{x}_j\Big) \qquad \text{back-substitution}$$

- Choleski decomposition is fast (does not touch zero blocks)
  non-zero elements are $9p + 121k + 66pk \approx 3.4 \cdot 10^6$; ca. $250\times$ fewer than all elements
- it can be computed on single elements or on entire blocks
- use profile Choleski for sparse $\mathbf{A}$ and diagonal pivoting for semi-definite $\mathbf{A}$ — see above; [Triggs et al. 1999]
- λ controls the definiteness

Profile Choleski Decomposition is Simple

```
function L = pchol(A)
%
% PCHOL profile Choleski factorization,
%    L = PCHOL(A) returns lower-triangular sparse L such that A = L*L'
%    for sparse square symmetric positive definite matrix A,
%    especially useful for arrowhead sparse matrices.

% (c) 2010 Radim Sara (sara@cmp.felk.cvut.cz)

 [p,q] = size(A);
 if p ~= q, error 'Matrix A is not square'; end

 L = sparse(q,q);
 F = ones(q,1);
 for i=1:q
  F(i) = find(A(i,:),1); % 1st non-zero on row i; we are building F gradually
  for j = F(i):i-1
   k = max(F(i),F(j));
   a = A(i,j) - L(i,k:(j-1))*L(j,k:(j-1))';
   L(i,j) = a/L(j,j);
  end
  a = A(i,i) - sum(full(L(i,F(i):(i-1))).^2);
  if a < 0, error 'Matrix A is not positive definite'; end
  L(i,i) = sqrt(a);
 end
end
```

▶Gauge Freedom

1. The external frame is not fixed: See Projective Reconstruction Theorem →129

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

2. Some representations are not minimal, e.g.
   - $\mathbf{P}$ is 12 numbers for 11 parameters
   - we may represent $\mathbf{P}$ in decomposed form $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$
   - but $\mathbf{R}$ is 9 numbers representing the 3 parameters of rotation

As a result

- there is no unique solution
- matrix $\sum_r \mathbf{L}_r^\top \mathbf{L}_r$ is singular

Solutions

1. fixing the external frame (e.g. a selected camera frame) explicitly or by constraints

2a. either imposing constraints on projective entities
   - cameras, e.g. $\mathbf{P}_{3,4} = 1$ — this excludes affine cameras
   - points, e.g. $\|\underline{\mathbf{X}}_i\|^2 = 1$ — this way we can represent points at infinity

2b. or using minimal representations
   - points in their Euclidean representation $\mathbf{X}_i$ — but finite points may be an unrealistic model
   - rotation matrix can be represented by axis-angle or the Cayley transform — see next

Implementing Simple Constraints

What for?

1. fixing external frame as in $\theta_i = \mathbf{t}_i$ — 'trivial gauge'
2. representing additional knowledge as in $\theta_i = \theta_j$ — e.g. cameras share calibration matrix $\mathbf{K}$

Introduce reduced parameters $\hat{\theta}$ and replication matrix $\mathbf{T}$:

$$\theta = \mathbf{T}\,\hat{\theta} + \mathbf{t}, \quad \mathbf{T} \in \mathbb{R}^{p,\hat{p}}, \quad \hat{p} \le p$$

then $\mathbf{L}_r$ in LM changes to $\mathbf{L}_r\,\mathbf{T}$ and everything else stays the same →106

$$\mathbf{T} = \begin{array}{c|cccc|} & \hat{\theta}_1 & \hat{\theta}_2 & \hat{\theta}_3 & \hat{\theta}_4 \\ \theta_1 & 1 & & & \\ \theta_2 & & 1 & & \\ \theta_3 & & & & \\ \theta_4 & & & & 1 \\ \theta_5 & & & & 1 \end{array} \qquad \mathbf{t} = \begin{array}{|c|} \\ \\ 1 \\ \\ \\ \end{array}$$

these $\mathbf{T}$, $\mathbf{t}$ represent	
$\theta_1 = \hat{\theta}_1$	no change
$\theta_2 = \hat{\theta}_2$	no change
$\theta_3 = t_3$	constancy
$\theta_4 = \theta_5 = \hat{\theta}_4$	equality

- $\mathbf{T}$ deletes columns of $\mathbf{L}_r$ that correspond to fixed parameters — it reduces the problem size
- consistent initialisation: $\theta^0 = \mathbf{T}\,\hat{\theta}^0 + \mathbf{t}$ — or filter the init by pseudoinverse $\theta^0 \mapsto \mathbf{T}^\dagger \theta^0$
- no need for computing derivatives for θ_j corresponding to all-zero rows of $\mathbf{T}$ — fixed θ

- constraining projective entities →144–145
- more complex constraints tend to make normal equations dense
- implementing constraints is safer than explicit renaming of the parameters, gives a flexibility to experiment
- other methods are much more involved, see [Triggs et al. 1999]
- **BA resource:** `http://www.ics.forth.gr/~lourakis/sba/` [Lourakis 2009]

Matrix Exponential

- for any square matrix we define

$$\operatorname{expm}\mathbf{A} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \qquad \text{note: } \mathbf{A}^0 = \mathbf{I}$$

- some properties:

$$\operatorname{expm}\mathbf{0} = \mathbf{I}, \quad \operatorname{expm}(-\mathbf{A}) = \left(\operatorname{expm}\mathbf{A}\right)^{-1},$$

$$\operatorname{expm}(a\,\mathbf{A})\operatorname{expm}(b\,\mathbf{A}) = \operatorname{expm}((a+b)\mathbf{A}), \quad \operatorname{expm}(\mathbf{A}+\mathbf{B}) \neq \operatorname{expm}(\mathbf{A})\operatorname{expm}(\mathbf{B})$$

$\operatorname{expm}(\mathbf{A}^\top) = (\operatorname{expm}\mathbf{A})^\top$ hence if $\mathbf{A}$ is skew symmetric then $\operatorname{expm}\mathbf{A}$ is orthogonal:

$$\left(\operatorname{expm}(\mathbf{A})\right)^\top = \operatorname{expm}(\mathbf{A}^\top) = \operatorname{expm}(-\mathbf{A}) = \left(\operatorname{expm}(\mathbf{A})\right)^{-1}$$

$$\det\operatorname{expm}\mathbf{A} = \operatorname{expm}(\operatorname{tr}\mathbf{A})$$

Ex:

- homography can be represented via exponential map with 8 numbers e.g. as

$$\mathbf{H} = \operatorname{expm}\mathbf{Z} \quad \text{such that} \quad \operatorname{tr}\mathbf{Z} = 0, \text{ eg. } \mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & -(z_{11}+z_{22}) \end{bmatrix}$$

- rotation can be represented by skew-symmetric matrix (3 numbers), see next

▶Minimal Representations for Rotation

- $\mathbf{o}$ – rotation axis, $\|\mathbf{o}\| = 1$, φ – rotation angle
- **wanted**: simple mapping to/from rotation matrices

1. Matrix exponential. Let $\boldsymbol{\omega} = \varphi\,\mathbf{o}$, $0 \leq \varphi < \pi$, then

$$\mathbf{R} = \operatorname{expm}\,[\boldsymbol{\omega}]_\times = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\omega}]_\times^n}{n!} = \overset{\circledast\,1}{\cdots} = \mathbf{I} + \frac{\sin\varphi}{\varphi}\,[\boldsymbol{\omega}]_\times + \frac{1-\cos\varphi}{\varphi^2}\,[\boldsymbol{\omega}]_\times^2$$

   - for $\varphi = 0$ we take the limit and get $\mathbf{R} = \mathbf{I}$
   - this is the Rodrigues' formula for rotation
   - inverse (the principal logarithm of $\mathbf{R}$) from

$$0 \leq \varphi < \pi, \quad \cos\varphi = \frac{1}{2}(\operatorname{tr}\mathbf{R} - 1), \quad [\boldsymbol{\omega}]_\times = \frac{\varphi}{2\,\sin\varphi}(\mathbf{R} - \mathbf{R}^\top),$$

2. Cayley's representation; let $\mathbf{a} = \mathbf{o}\tan\frac{\varphi}{2}$, then

$$\mathbf{R} = (\mathbf{I} + [\mathbf{a}]_\times)(\mathbf{I} - [\mathbf{a}]_\times)^{-1}, \quad [\mathbf{a}]_\times = (\mathbf{R} + \mathbf{I})^{-1}(\mathbf{R} - \mathbf{I})$$

$$\mathbf{a}_1 \circ \mathbf{a}_2 = \frac{\mathbf{a}_1 + \mathbf{a}_2 - \mathbf{a}_1 \times \mathbf{a}_2}{1 - \mathbf{a}_1^\top \mathbf{a}_2}$$

composition of rotations $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2$

   - again, cannot represent rotations for $\phi \geq \pi$
   - no trigonometric functions
   - explicit composition formula

▶Minimal Representations for Other Entities

with the help of rotation we can minimally represent

1. fundamental matrix

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \quad \mathbf{D} = \operatorname{diag}(1, d^2, 0), \quad \mathbf{U}, \mathbf{V} \text{ are rotations}, \quad 3 + 1 + 3 = 7 \text{ DOF}$$

2. essential matrix

$$\mathbf{E} = [-\mathbf{t}]_\times \mathbf{R}, \quad \mathbf{R} \text{ is rotation}, \quad \|\mathbf{t}\| = 1, \qquad 3 + 2 = 5 \text{ DOF}$$

3. camera

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}, \qquad 5 + 3 + 3 = 11 \text{ DOF}$$

Interestingly, let [Eade 2017]

$$\mathbf{B} = \begin{bmatrix}[\boldsymbol{\omega}]_\times & \mathbf{u} \\ \mathbf{0}^\top & 0\end{bmatrix}, \qquad \mathbf{B} \in \mathbb{R}^{4,4}$$

then, assuming $\|\boldsymbol{\omega}\| = \phi > 0$ — for $\phi = 0$ we take the limits

$$\begin{bmatrix}\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1\end{bmatrix} = \operatorname{expm}\mathbf{B} = \mathbf{I}_4 + \mathbf{B} + h_2(\phi)\,\mathbf{B}^2 + h_3(\phi)\,\mathbf{B}^3 = \begin{bmatrix}\operatorname{expm}[\boldsymbol{\omega}]_\times & \mathbf{V}\,\mathbf{u} \\ \mathbf{0}^\top & 1\end{bmatrix}$$

$$\mathbf{V} = \mathbf{I}_3 + h_2(\phi)\,[\boldsymbol{\omega}]_\times + h_3(\phi)\,[\boldsymbol{\omega}]_\times^2, \quad \mathbf{V}^{-1} = \mathbf{I}_3 - \frac{1}{2}[\boldsymbol{\omega}]_\times + h_4(\phi)[\boldsymbol{\omega}]_\times^2$$

$$h_1(\phi) = \frac{\sin\phi}{\phi}, \quad h_2(\phi) = \frac{1-\cos\phi}{\phi^2}, \quad h_3(\phi) = \frac{\phi - \sin\phi}{\phi^3}, \quad h_4(\phi) = \frac{1}{\phi^2}\left(1 - \frac{1}{2}\,\phi\cot\frac{\phi}{2}\right)$$

Thank You