

3D Structure and Camera Motion

6.1 Introduction

6.2 Reconstructing Camera Systems

6.3 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.

additional references



D. Martinec and T. Pajdla. Robust Rotation and Translation Estimation in Multiview Reconstruction. In *Proc CVPR*, 2007



M. I. A. Lourakis and A. A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans Math Software* 36(1):1–30, 2009.

► 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

$$\mathbf{P}_1 = [\mathbf{I} \quad \mathbf{0}]$$

See [H&Z, p. 256]

$$\mathbf{P}_2 = [[\mathbf{e}_2]_{\times} \mathbf{F} + \mathbf{e}_2 \underline{\mathbf{v}}^{\top} \quad \lambda \mathbf{e}_2]$$

where

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

Proof

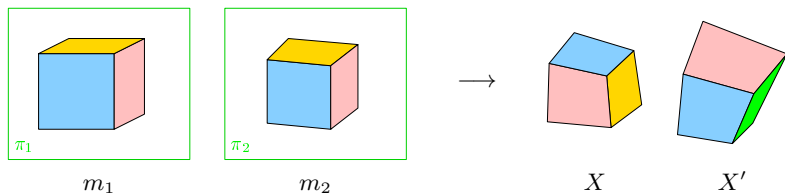
1. \mathbf{S} is skew-symmetric iff $\mathbf{x}^{\top} \mathbf{S} \mathbf{x} = 0$ for all \mathbf{x} look-up the proof!
2. we have $\underline{\mathbf{x}} \simeq \mathbf{P} \underline{\mathbf{X}}$
3. a non-zero \mathbf{F} is a f.m. of $(\mathbf{P}_1, \mathbf{P}_2)$ iff $\mathbf{P}_2^{\top} \mathbf{F} \mathbf{P}_1$ is skew-symmetric
4. if $\mathbf{P}_1 = [\mathbf{I} \quad \mathbf{0}]$ and $\mathbf{P}_2 = [\mathbf{S} \mathbf{F} \quad \mathbf{e}_2]$ 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 $\mathbf{s} = \mathbf{e}_2$ [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 $i = 1, \dots, k$

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

- 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)

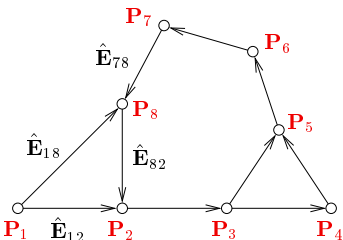


- when cameras are internally calibrated (\mathbf{K}_i known) then \mathbf{H} is restricted to a similarity since it must preserve the calibrations \mathbf{K}_i [H&Z, Secs. 10.2, 10.3], [Longuet-Higgins 1981]
(translation, rotation, scale)

► 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 $\mathbf{P}_i, i = 1, \dots, k$

→81 and →146 on representing \mathbf{E}



We construct calibrated camera pairs $\hat{\mathbf{P}}_{ij} \in \mathbb{R}^{6,4}$ →129

$$\hat{\mathbf{P}}_{ij} = \begin{bmatrix} \mathbf{K}_i^{-1} \hat{\mathbf{P}}_i \\ \mathbf{K}_j^{-1} \hat{\mathbf{P}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix} \in \mathbb{R}^{6,4}$$

- singletons i, j correspond to graph nodes k nodes
- pairs ij correspond to graph edges p edges

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

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix}}_{\mathbb{R}^{6,4}} \underbrace{\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^{\top} & s_{ij} \end{bmatrix}}_{\mathbf{H}_{ij} \in \mathbb{R}^{4,4}} \stackrel{!}{=} \underbrace{\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{R}_j & \mathbf{t}_j \end{bmatrix}}_{\mathbb{R}^{6,4}} \quad (29)$$

- (29) 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 node \mathbf{P}_i eg. P_5 3-times

► cont'd

Eq. (29) implies
$$\begin{bmatrix} \mathbf{R}_{ij} \\ \hat{\mathbf{R}}_{ij} \mathbf{R}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i \\ \mathbf{R}_j \end{bmatrix} \quad \begin{bmatrix} \mathbf{t}_{ij} \\ \hat{\mathbf{R}}_{ij} \mathbf{t}_{ij} + s_{ij} \hat{\mathbf{t}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_j \end{bmatrix}$$

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

$$\hat{\mathbf{R}}_{ij} \mathbf{R}_i = \mathbf{R}_j, \quad \hat{\mathbf{R}}_{ij} \mathbf{t}_i + s_{ij} \hat{\mathbf{t}}_{ij} = \mathbf{t}_j, \quad s_{ij} > 0 \quad (30)$$

- 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, e.g. by selecting

$$\mathbf{R}_1 = \mathbf{I}, \quad \sum_{i=1}^k \mathbf{t}_i = \mathbf{0}, \quad \frac{1}{p} \sum_{i,j} s_{ij} = 1 \quad (31)$$

- rotation equations are decoupled from translation equations
- in principle, s_{ij} could correct the sign of $\hat{\mathbf{t}}_{ij}$ from essential matrix decomposition →81
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 $s_{ij} > 0$; →83

+ pairwise correspondences are sufficient

- suitable for well-distributed cameras only (dome-like configurations)

otherwise intractable or numerically unstable

► Solving Eq. (30) by Stepwise Gluing

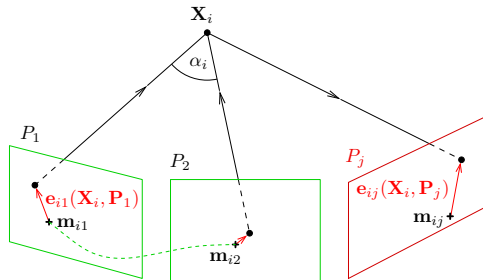
Given: Calibration matrices \mathbf{K}_j and tentative correspondences per camera triples.

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 →88
3. construct camera pair

$$\mathbf{P}_1 = \mathbf{K}_1 [\mathbf{I} \quad \mathbf{0}], \quad \mathbf{P}_2 = \mathbf{K}_2 [\mathbf{R} \quad \mathbf{t}]$$

4. compute 3D reconstruction $\{X_i\}$ per match from M_{12} →105
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 e_{ij} in \mathcal{X}_j →68
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 →137

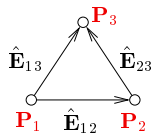
Finding The Rotation Component in Eq. (30): A Global Algorithm

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 :

$$\hat{\mathbf{R}}_{ij}\mathbf{r}_i^c - \mathbf{r}_j^c = \mathbf{0}, \quad \text{for all } i, j \quad (32)$$

- fix c and denote $\mathbf{r}^c = [\mathbf{r}_1^c, \mathbf{r}_2^c, \dots, \mathbf{r}_k^c]^\top$ c -th columns of all rotation matrices stacked; $\mathbf{r}^c \in \mathbb{R}^{3k}$
- then (32) becomes $\mathbf{D}\mathbf{r}^c = \mathbf{0}$ $\mathbf{D} \in \mathbb{R}^{3p, 3k}$
- $3p$ equations for $3k$ unknowns $\rightarrow p \geq k$ in a 1-connected graph we have to fix $\mathbf{r}_1^c = [1, 0, 0]$

Ex: ($k = p = 3$)



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

- 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 (32) \mathbf{D} is sparse, use $[V, E] = \text{eigs}(\mathbf{D}^* \mathbf{D}, 3, 0)$; (Matlab)
 2. choose 3 unit orthogonal vectors in this space 3 smallest eigenvectors
 3. find closest rotation matrices per cam. using SVD 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. (30)

From (30) and (31):

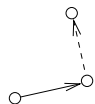
$d \leq 3$ – rank of camera center set, p – #pairs, k – #cameras

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

- in rank d : $d \cdot p + d + 1$ equations for $d \cdot k + p$ unknowns $\rightarrow p \geq \frac{d(k-1)-1}{d-1} \stackrel{\text{def}}{=} Q(d, k)$

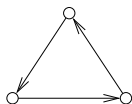
Ex: Chains and circuits construction from sticks of known orientation and unknown length?

$p = k - 1$



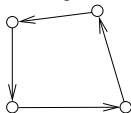
$k \leq 2$ for any d

$k = p = 3$



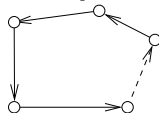
$3 \geq d \geq 2$: non-collinear ok

$k = p = 4$



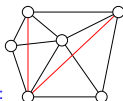
$3 \geq d \geq 3$: non-planar ok

$k = p > 4$



$3 \geq d \geq k - 1$: impossible

- equations insufficient for chains, trees, or when $d = 1$ collinear cameras
- 3-connectivity implies sufficient equations 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 $p \geq \lceil \frac{3k}{2} \rceil \geq Q(3, k) = \frac{3k}{2} - 2$
- 4-connectivity implies sufficient eqns. for any k when $d = 2$ coplanar cams
 - since $p \geq \lceil 2k \rceil \geq Q(2, k) = 2k - 3$
 - maximal planar triangulated graphs have $p = 3k - 6$ maximal planar triangulated graph example:
 - and give a solution for $k \geq 3$



Linear equations in (30) and (31) can be rewritten to

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

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 (mind the 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!

► Bundle Adjustment

Given:

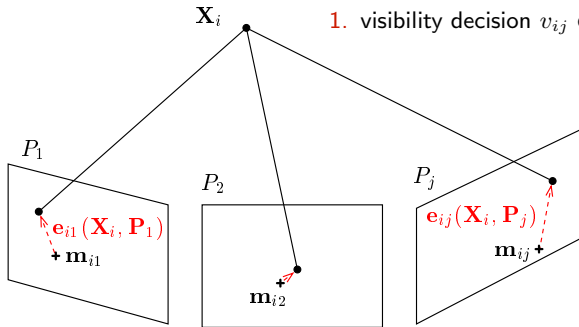
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^P$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^C$
3. fixed tentative projections \mathbf{m}_{ij}

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 $\mathbf{e}_{ij}(\mathbf{X}_i, P_j) = \mathbf{x}_i - \mathbf{m}_i$ per image feature, where $\mathbf{x}_i = P_j \mathbf{X}_i$
- 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 →113

$$p(\{e\} | \{\mathbf{P}, \mathbf{X}\}) = \prod_{\text{pts}:i=1}^p \prod_{\text{cams}:j=1}^c \left((1 - P_0)p_1(e_{ij} | \mathbf{X}_i, \mathbf{P}_j) + P_0p_0(e_{ij} | \mathbf{X}_i, \mathbf{P}_j) \right)$$

marginalized negative log-density is (→114)

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

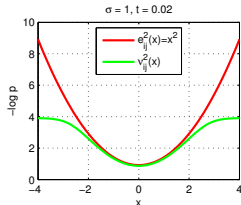
- e_{ij} is the projection error (not Sampson error)
- ν_{ij} is a 'robust' error fcn.; it is non-robust ($\nu_{ij} = e_{ij}$) when $t = 0$
- $\rho(\cdot)$ is a 'robustification function' we often find in M-estimation
- the \mathbf{L}_{ij} in Levenberg-Marquardt changes to vector

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t e^{e_{ij}^2(\theta)/(2\sigma_1^2)}}}_{\text{small for big } e_{ij}} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \quad (33)$$

but the LM method stays the same as before →107–108

- outliers: almost no impact on \mathbf{d}_s in normal equations because the red term in (33) scales contributions to both sums down for the particular ij

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

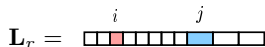


► Sparsity in Bundle Adjustment

We have $q = 3p + 11k$ parameters: $\theta = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_k)$ points, cameras
 We will use a running index $r = 1, \dots, z$, $z = p \cdot k$. Then each r corresponds to some i, j

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

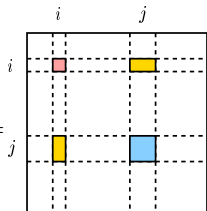
The block form of \mathbf{L}_r in Levenberg-Marquardt ($\rightarrow 107$) is zero except in columns i and j :
 r -th error term is $\nu_r^2 = \rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j))$



$r = (i, j)$ blocks:

$\color{red}\square$: $\mathbf{X}_i, 1 \times 3$

$\color{blue}\square$: $\mathbf{P}_j, 1 \times 11$

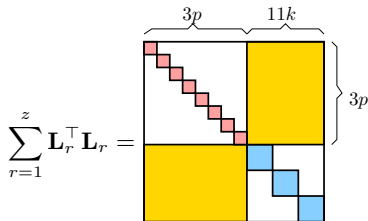


blocks:

$\color{red}\square$: $\mathbf{X}_i - \mathbf{X}_i, 3 \times 3$

$\color{yellow}\square$: $\mathbf{X}_i - \mathbf{P}_j, 3 \times 11$

$\color{blue}\square$: $\mathbf{P}_j - \mathbf{P}_j, 11 \times 11$



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

► Choleski Decomposition for B. A.

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

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

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 $\underbrace{\mathbf{L}\mathbf{L}^\top}_{\mathbf{c}} \mathbf{x} = \mathbf{b}$

2. solve for \mathbf{x} in two passes:

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

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

- 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} [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 must be 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 must be positive definite'; end
    L(i,i) = sqrt(a);
end
end
```

► Gauge Freedom

1. The external frame is not fixed: See Projective Reconstruction Theorem →130

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

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. $\mathbf{P}_{3,4} = 1$
 - points, e.g. $\|\underline{\mathbf{X}}_i\|^2 = 1$
- this excludes affine cameras
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

Thank You