Structure from Motion

Projections and image formation

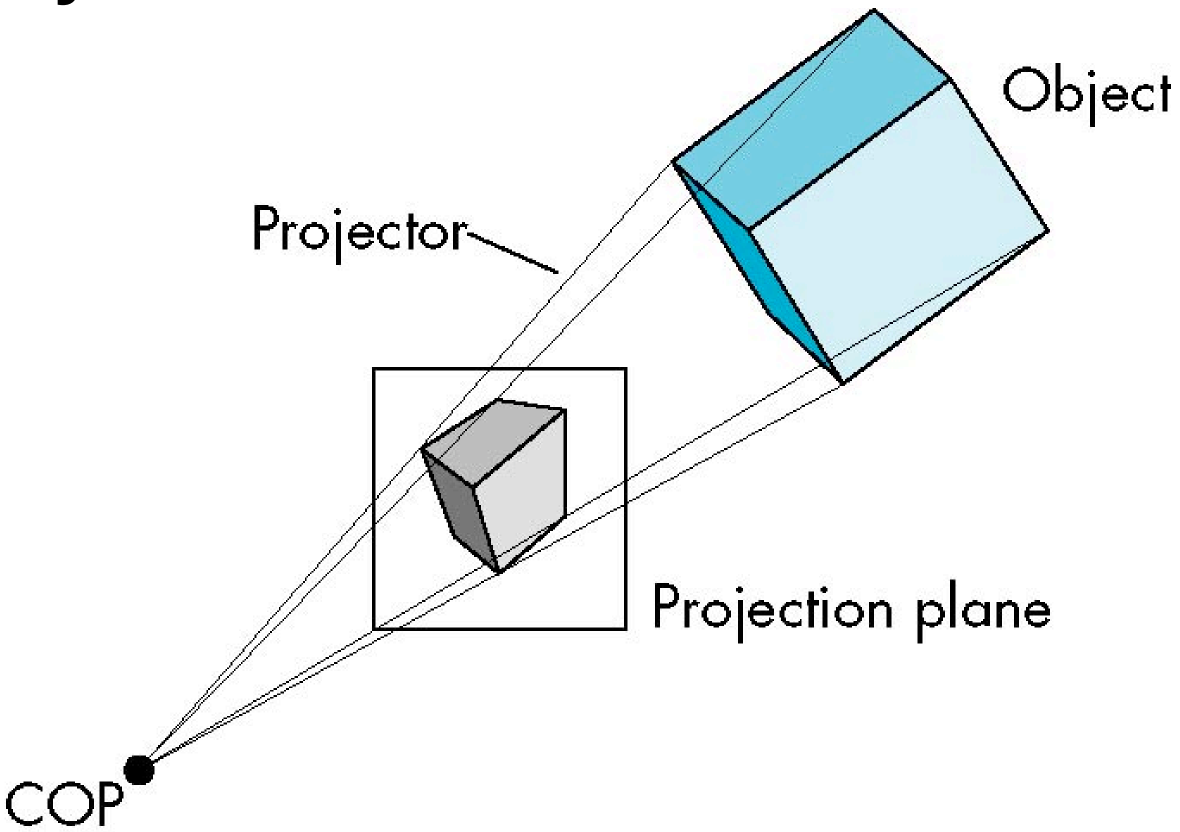
Problem statement

Rank theorem

Factorization method

Perspective Projection

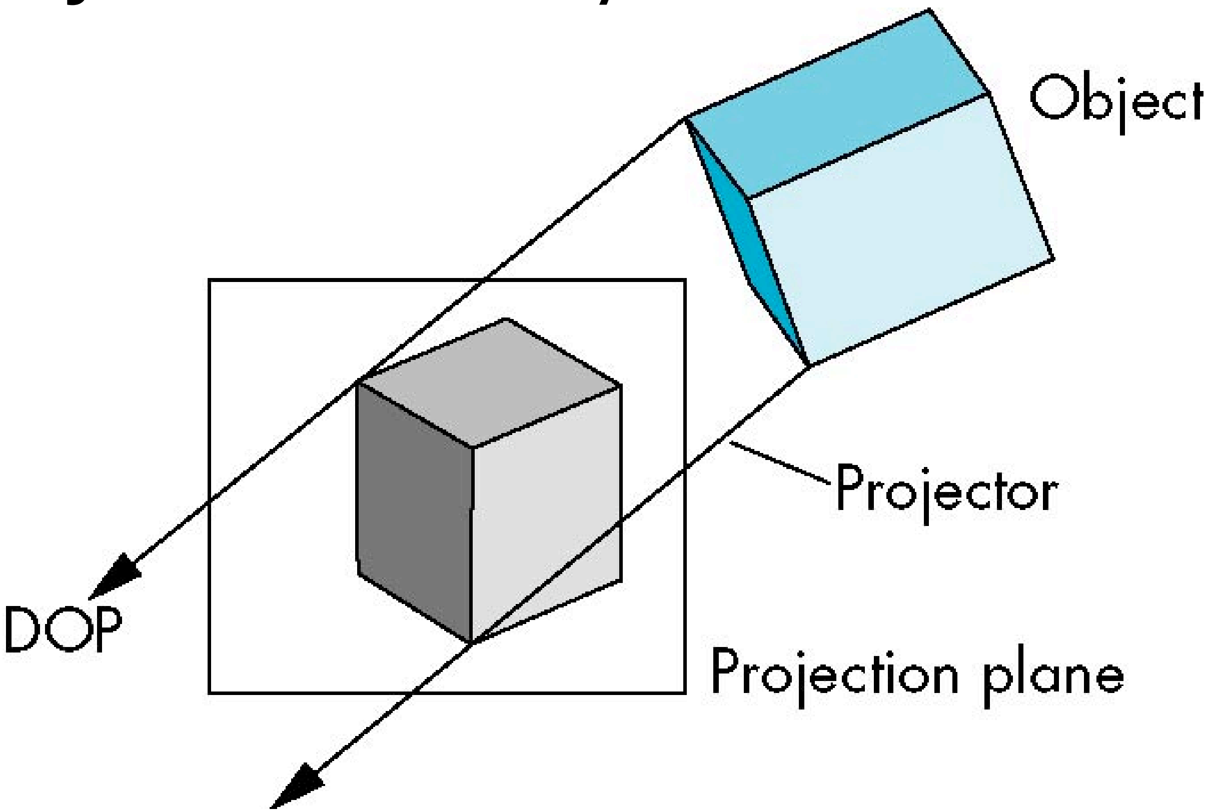
- Projectors converge at a single point: center of projection.



2

Parallel Projection

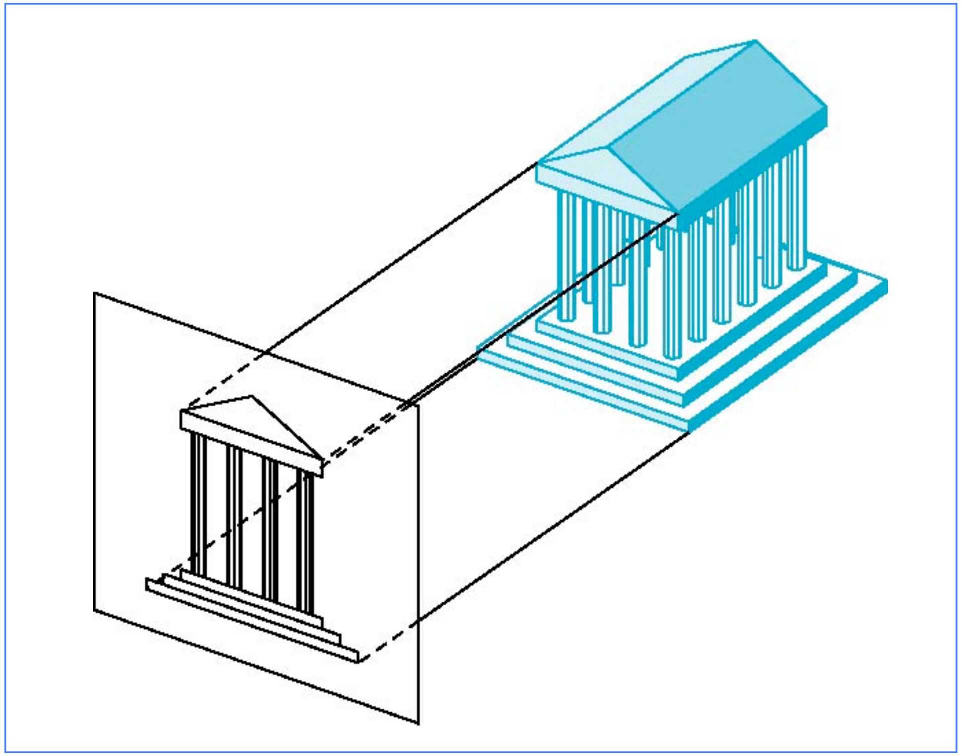
- Projectors are parallel to each other: center of projection at infinity.



3

Orthographic Projection

- Projectors are parallel to each other and perpendicular to projection surface:



4

Orthographic Image Formation

- Model the position of an object with respect to the camera by a rigid transformation (rotation + translation).
- Apply projection matrix (typically, drop one coordinate).

$$\begin{bmatrix} x_{\text{img}} \\ y_{\text{img}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{t} \right)$$

Questions

- Is it possible to recover the original (3D) scene coordinates (structure)?
- Is it possible to recover the relative transformation (motion)?

- Structure from Motion: yes* to both questions, if we track a number of points in a sequence of images (frames)!

*Certain restrictions may apply!

Motion Field

- **Definition**: the motion field is the 2D vector field of velocities of the image points, induced by the relative motion between the viewing camera and the observed scene.

- Dense motion field (optical flow): a motion vector at (nearly) every pixel.
- Sparse motion field: a motion vector for a sparse set of landmarks in the image.

Structure from Motion

- **Problem statement**: Given the 2D motion field estimated from an image sequence, compute the shape, or 3D structure, of the visible objects, and their 3D motion with respect to the viewing camera.

- Today: focus on the case of a sparse 2D motion field: Tomasi and Kanade (1992).

Assumptions

- Orthographic camera model.
- The position of n image points, corresponding to n scene points: $P_1, \ldots, P_n$, not all coplanar, have been tracked in N frames ($N \geq 3$).

Measurement Matrix

- The j-th image point (j = 1,…,n) in the i-th frame (i = 1,…,N):

$$\mathbf{p}_{ij} = [x_{ij} \;\; y_{ij}]^{\mathrm{T}}$$

- Measurement matrix (2N × n):

$$W = \begin{bmatrix} X = \{x_{ij}\} \\ Y = \{y_{ij}\} \end{bmatrix}$$

Registered Measurement Matrix

$$\tilde{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \qquad \tilde{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij}$$

$$\tilde{x}_{ij} = x_{ij} - \tilde{x}_i \qquad \tilde{y}_{ij} = y_{ij} - \tilde{y}_i$$

$$\tilde{W} = \begin{bmatrix} \tilde{X} = \{\tilde{x}_{ij}\} \\ \tilde{Y} = \{\tilde{y}_{ij}\} \end{bmatrix}$$

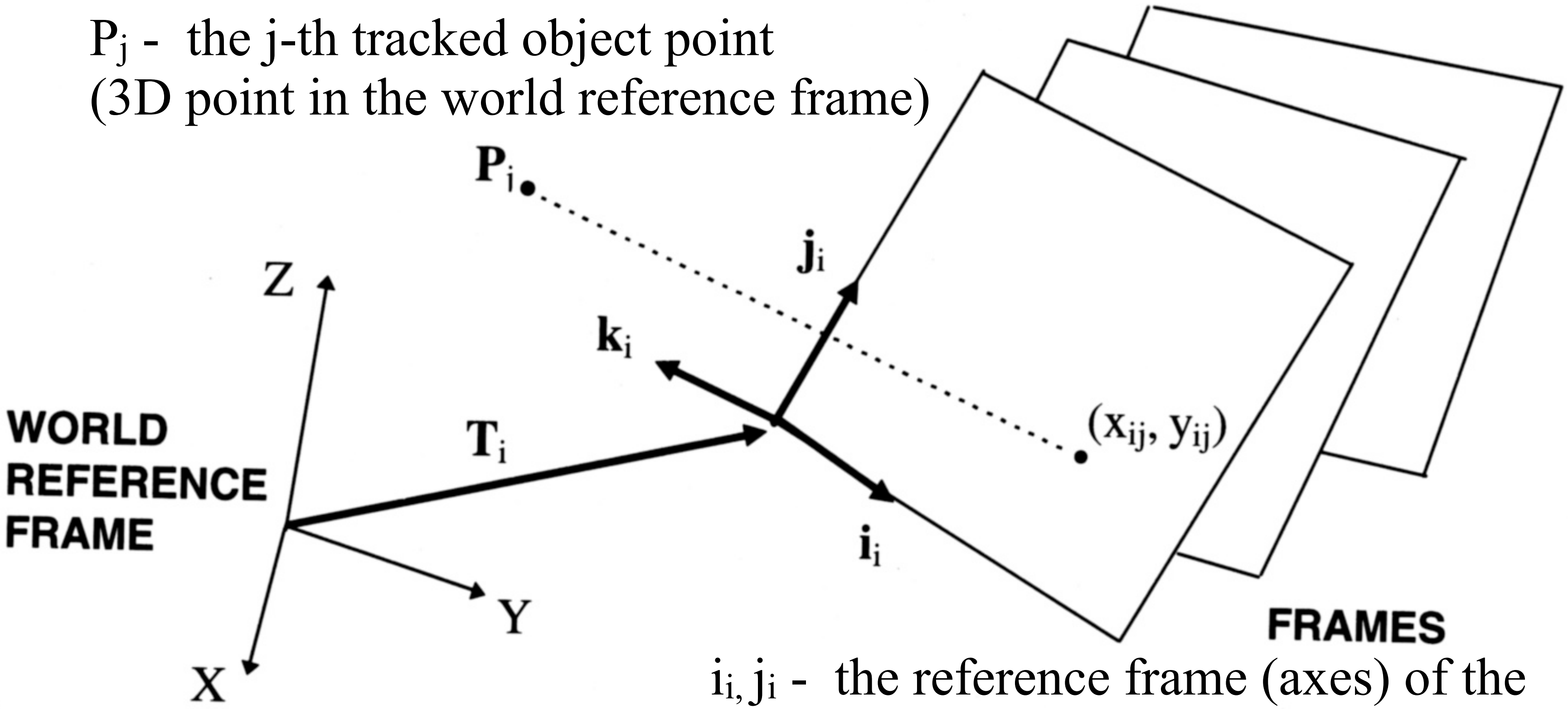
The Rank Theorem

- The registered measurement matrix W (without noise) has at most rank 3.

- Proof idea: the registered measurement matrix may be decomposed into a product of two matrices, rotation R, and shape S, each of which has rank 3.

The Rank Theorem - Proof

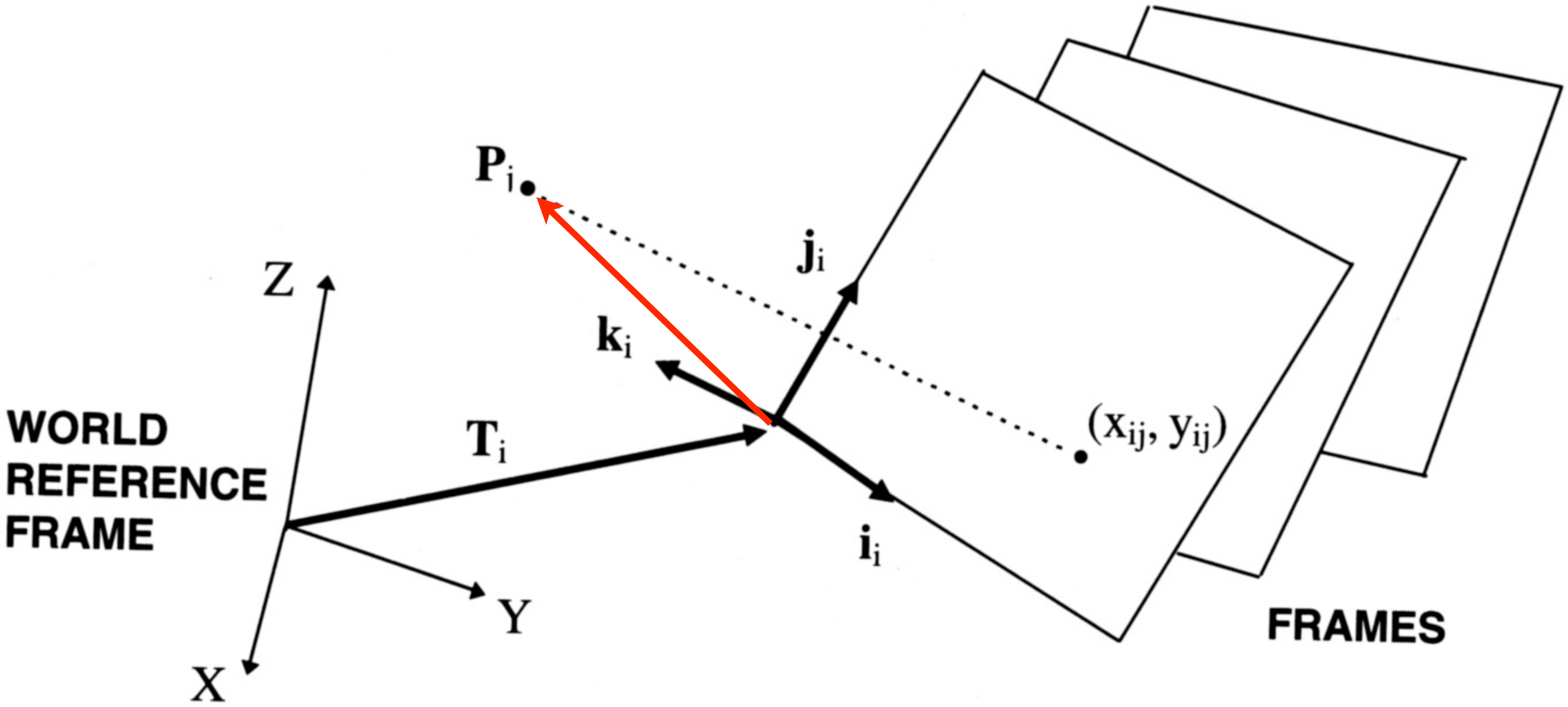
P_j - the j-th tracked object point
(3D point in the world reference frame)



i_i, j_i - the reference frame (axes) of the i-th image in the sequence (3D vectors in the world reference frame).

T_i - the vector from the world origin to the origin of the i-th image reference frame

The Rank Theorem - Proof



$$x_{ij} = \mathbf{i}_i^{\mathrm{T}} (\mathbf{P}_j - \mathbf{T}_i)$$

$$y_{ij} = \mathbf{j}_i^{\mathrm{T}} (\mathbf{P}_j - \mathbf{T}_i)$$

The Rank Theorem - Proof

$$\tilde{x}_{ij} = \mathbf{i}_i^{\mathrm{T}} (\mathbf{P}_j - \mathbf{T}_i) - \frac{1}{n} \sum_{m=1}^{n} \mathbf{i}_i^{\mathrm{T}} (\mathbf{P}_m - \mathbf{T}_i)$$

$$= \mathbf{i}_i^{\mathrm{T}} (\mathbf{P}_j - \mathbf{T}_i) - \mathbf{i}_i^{\mathrm{T}} \left(\frac{1}{n} \sum_{m=1}^{n} \mathbf{P}_m - \mathbf{T}_i \right)$$

by definition: $\frac{1}{n} \sum_{m=1}^{n} \mathbf{P}_m = 0$

$$\tilde{x}_{ij} = \mathbf{i}_i^{\mathrm{T}} \mathbf{P}_j \qquad \tilde{y}_{ij} = \mathbf{j}_i^{\mathrm{T}} \mathbf{P}_j$$

The Rank Theorem - Proof

$$\underset{2N \times n}{\tilde{W}} = \underset{2N \times 3}{\begin{bmatrix} \mathbf{i}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{i}_N^{\mathrm{T}} \\ \mathbf{j}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{j}_N^{\mathrm{T}} \end{bmatrix}} \underset{3 \times n}{\begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \ldots & \mathbf{P}_n \end{bmatrix}} = R\,S$$

Factorization

- The factorization is not unique. Let Q be any invertible 3x3 matrix, then:

$$\tilde{W} = R\,S = R\,QQ^{-1}S = (RQ)(Q^{-1}S)$$

- Add constraints:
  - Rows of R must have unit norm;
  - First n rows of R must be orthogonal to last (corresponding) n rows.

Factorization Algorithm

- Consider the singular value decomposition:

$$\tilde{W} = U\,D\,V^{\mathrm{T}}$$

- Rank 3 implies that only 3 singular values are non-zero!

$$\tilde{W} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_{2N} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \vdots \\ 0 & 0 & \sigma_3 & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix}^{\mathrm{T}}$$

$2N \times 2N$ $\quad$ $2N \times n$ $\quad$ $n \times n$

Factorization Algorithm

- Rewrite the decomposition as:

$$\tilde{W} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^T$$

$2N \times 3$ 3×3 $3 \times n$

U' D' V'^{T}

$$\hat{R} = U' D'^{1/2} \qquad \hat{S} = D'^{1/2} V'^{T}$$

Factorization Algorithm

- We get a decomposition with the desired dimensions:

$$\tilde{W} = \hat{R}\hat{S} = (U' D'^{1/2})(D'^{1/2} V'^{\mathrm{T}})$$

- Look for Q, s.t.:

$$\hat{\mathbf{i}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{i}}_i = 1$$

$$\hat{\mathbf{j}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{j}}_i = 1$$

$$\hat{\mathbf{i}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{j}}_i = 0$$

$$\boxed{\tilde{W} = RS = (\hat{R}Q)(Q^{-1}\hat{S})}$$

Solving for Q: method A

- Use a non-linear solver (e.g., Newton's method) to directly solve for the variables in the non-linear (quadratic) system of equations:

$$\hat{\mathbf{i}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{i}}_i = 1$$

$$\hat{\mathbf{j}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{j}}_i = 1$$

$$\hat{\mathbf{i}}_i^{\mathrm{T}} Q Q^{\mathrm{T}} \hat{\mathbf{j}}_i = 0$$

Solving for Q: method B

- Solve the linear system of equations:

$$\hat{\mathbf{i}}_i^{\mathrm{T}} C \hat{\mathbf{i}}_i = 1$$

$$\hat{\mathbf{j}}_i^{\mathrm{T}} C \hat{\mathbf{j}}_i = 1$$

$$\hat{\mathbf{i}}_i^{\mathrm{T}} C \hat{\mathbf{j}}_i = 0$$

- Use Cholesky to factor the matrix C:

$$Q Q^{\mathrm{T}} = C$$

- But beware: C might not be s.p.d!