Geometry for computer Vision

Lecture 5b: Calibrated multi-view Geometry

Lecture 5b: calibrated multi-view geometry

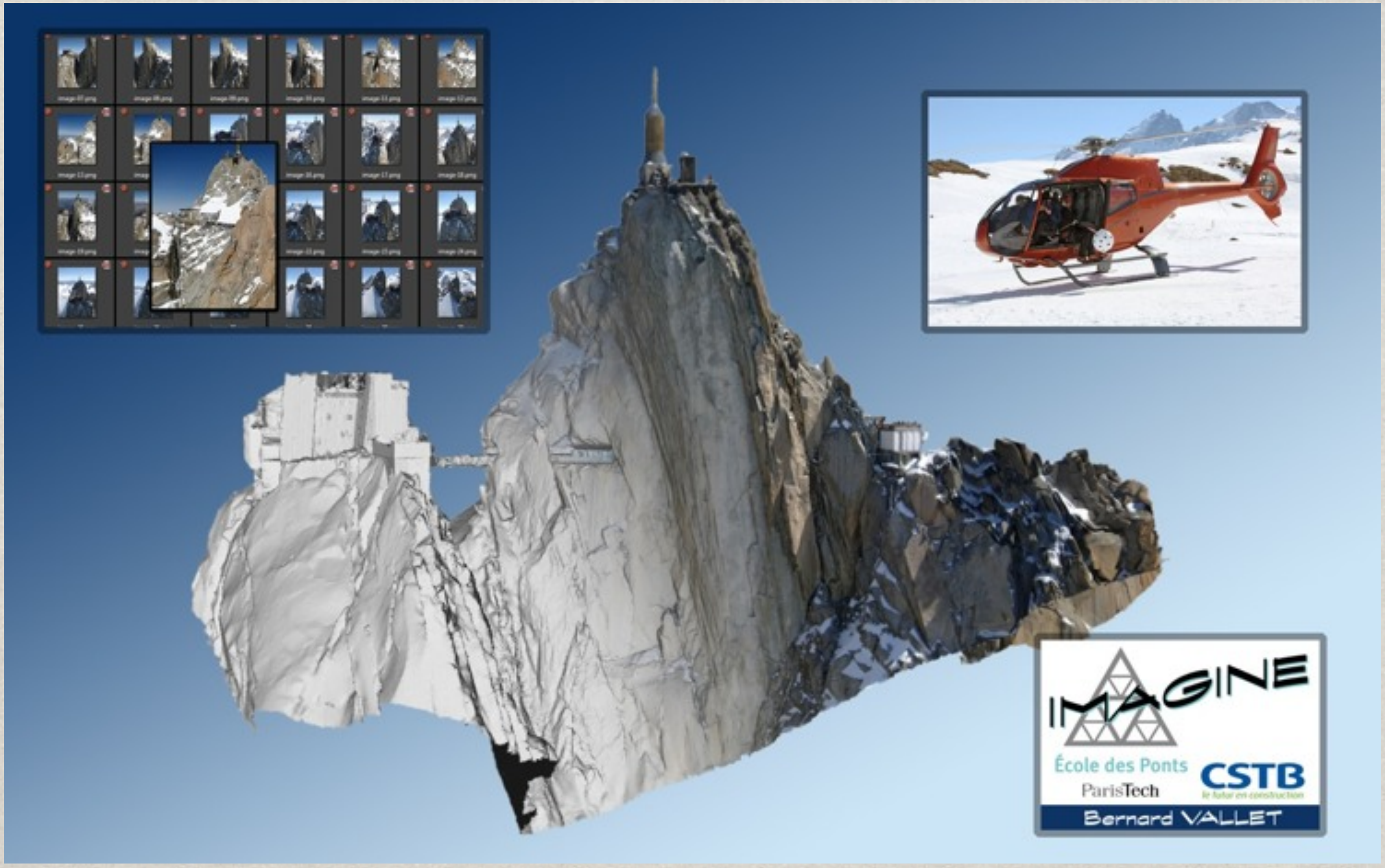
- The 5-point Algorithm
- P3P
- Bundle Adjustment

Tools for image-based 3D models

- E.g. Photo Tourism from University of Washington. (has a web demo)

Tools for image-based 3D models

- E.g. Pons et al. at Inria Sophia-Antipolis CVPR'09



Planar degeneracy

- In the uncalibrated case, two view geometry is encoded by the fundamental matrix

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

Planar degeneracy

- In the uncalibrated case, two view geometry is encoded by the fundamental matrix

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

- If all scene points lie on a plane, or if the camera has undergone a pure rotation (no translation), we also have:

$$\mathbf{x}_1 = \mathbf{H} \mathbf{x}_2$$

- Big trouble!

Planar degeneracy

- If $\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$, then the epipolar constraint becomes $\mathbf{x}_1^T\mathbf{F}\mathbf{x}_2 = \mathbf{x}_1^T\mathbf{F}\mathbf{H}^{-1}\mathbf{x}_1 = 0$
- For $\mathbf{M} = \mathbf{F}\mathbf{H}^{-1}$, this is true whenever $\mathbf{M}$ is skew-symmetric, i.e.

$$\mathbf{M}^T + \mathbf{M} = 0 \quad \Leftrightarrow \quad \mathbf{M} = [\mathbf{m}]_\times$$

Planar degeneracy

- If $\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$, then the epipolar constraint becomes $\mathbf{x}_1^T\mathbf{F}\mathbf{x}_2 = \mathbf{x}_1^T\mathbf{F}\mathbf{H}^{-1}\mathbf{x}_1 = 0$

- For $\mathbf{M} = \mathbf{F}\mathbf{H}^{-1}$, this is true whenever $\mathbf{M}$ is skew-symmetric, i.e.
$$\mathbf{M}^T + \mathbf{M} = 0 \quad \Leftrightarrow \quad \mathbf{M} = [\mathbf{m}]_\times$$

- Thus $\mathbf{F} = [\mathbf{s}]_\times\mathbf{H}$ where $\mathbf{s}$ may be chosen freely!

- A two-parameter family of solutions.

The 5-point algorithm

- Recap from last weeks lecture...
- In the calibrated case, epipolar geometry is encoded by the *essential matrix*, $\mathbf{E}$ according to:

$$\hat{\mathbf{x}}_1^T \mathbf{E} \hat{\mathbf{x}}_2 = 0$$

The 5-point algorithm

- Recap from last weeks lecture...
- In the calibrated case, epipolar geometry is encoded by the *essential matrix*, **E** according to:

$$\hat{\mathbf{x}}_1^T \mathbf{E} \hat{\mathbf{x}}_2 = 0$$

- In the calibrated setting there are just two possibilities if a plane is seen. See Negahdaripour, *Closed-form relationship between the two interpretations of a moving plane*. JOSA90

The 5-point algorithm

- **E** can be estimated from 5 corresponding points (see today's paper).
- A small sample is useful for RANSAC (le 3).
- The plane degeneracy is essentially avoided.
- There are however up to 10 solutions for **E** to test.

Perspective 3-point problem

- If we have the calibrated two view geometry, and want to add another view to the model.
- Or, in general from N views to N+1 views...
- First triangulate image points $\hat{\mathbf{x}}$ in two views to get 3D points $\mathbf{X}$
- Then relate $\mathbf{X}$ to image points in the new view

$$\hat{\mathbf{y}} \sim [\mathbf{R}|\mathbf{t}]\mathbf{X}$$

Perspective 3-point problem

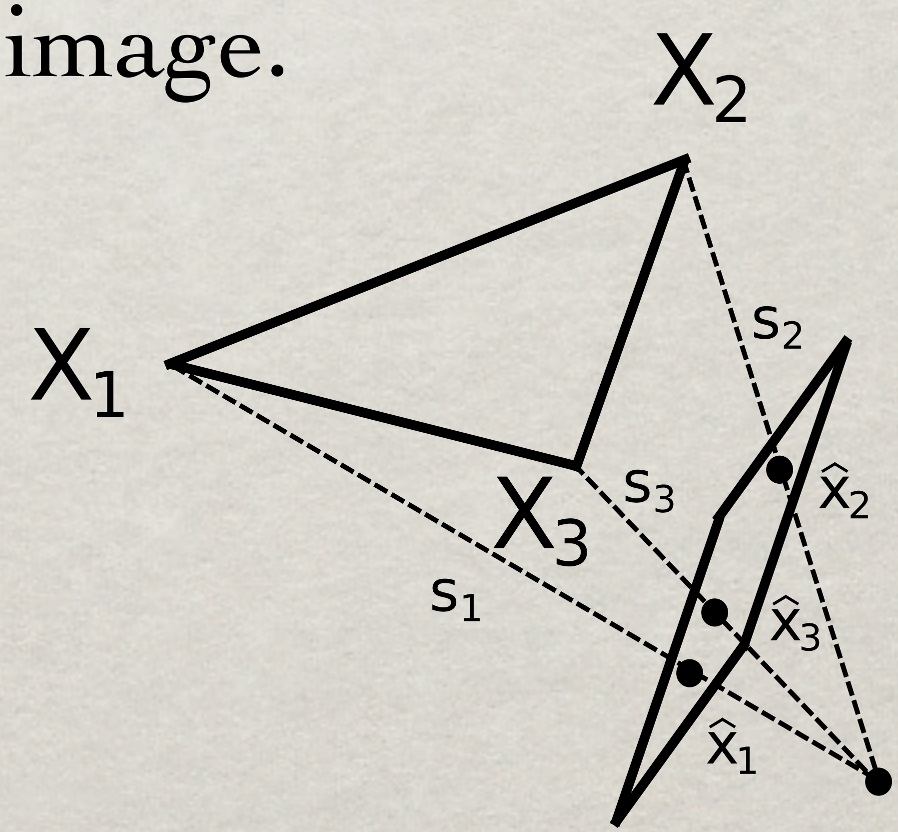
- Each correspondence $\hat{\mathbf{y}} \sim [\mathbf{R}|\mathbf{t}]\mathbf{X}$ gives us 2 equations.
- We have 6 unknowns.
- $\Rightarrow$ at least 3 points are needed

Perspective 3-point problem

- Each correspondence $\hat{\mathbf{y}} \sim [\mathbf{R}|\mathbf{t}]\mathbf{X}$ gives us 2 equations.
- We have 6 unknowns.
- $\Rightarrow$ at least 3 points are needed
- If we have outliers, we want to use RANSAC, with a minimal sample set.

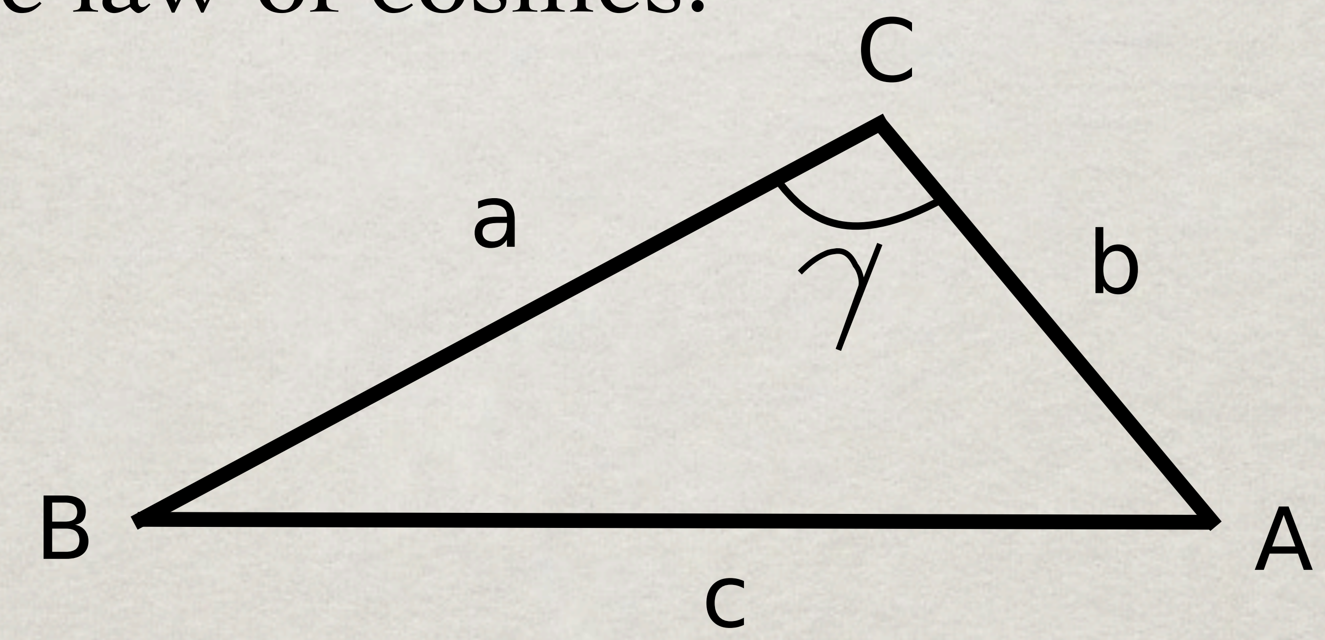
Perspective 3-point problem

- Instead of determining **R**, **t** directly one typically computes the distances to the 3D points **X** from the new camera centre given the side lengths of the 3D triangle, and the projections in the image.



Perspective 3-point problem

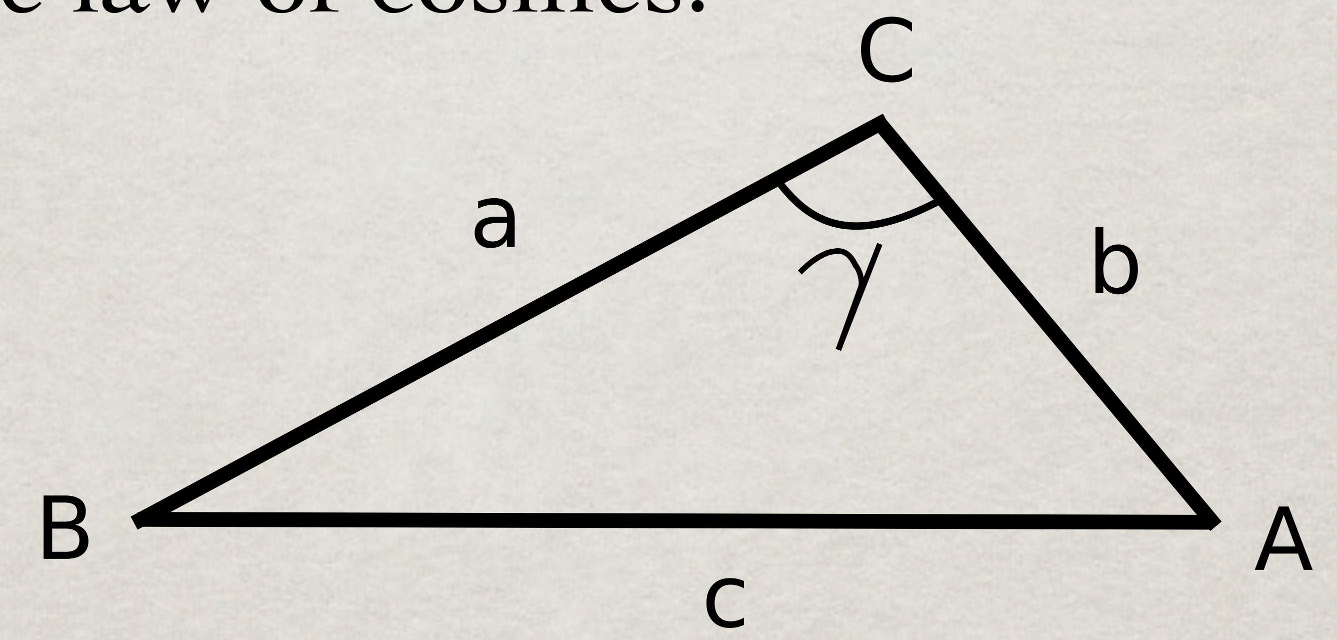
- Recall the law of cosines:



$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

Perspective 3-point problem

- Recall the law of cosines:



$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

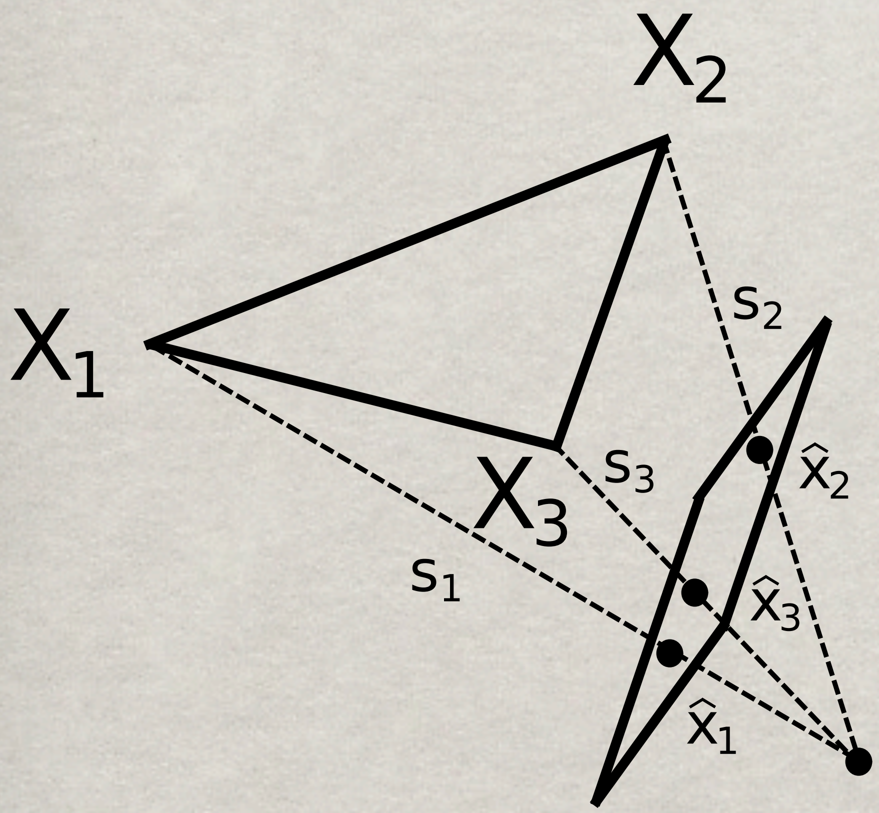
- By expressing the angles via scalar products, we get 3 equations with 3 unknowns.

Perspective 3-point problem

- Define angles between rays as:

$$\cos\gamma_{kl} = \frac{\hat{\mathbf{x}}_k^T \hat{\mathbf{x}}_l}{||\hat{\mathbf{x}}_k||||\hat{\mathbf{x}}_l||}$$

- and the side lengths as: $l_{kl} = ||\mathbf{X}_k - \mathbf{X}_l||$



$$\begin{cases} s_1^2 + s_2^2 - 2s_1 s_2 \cos\gamma_{12} & = & l_{12}^2 \\ s_1^2 + s_3^2 - 2s_1 s_3 \cos\gamma_{13} & = & l_{13}^2 \\ s_2^2 + s_3^2 - 2s_2 s_3 \cos\gamma_{23} & = & l_{23}^2 \end{cases}$$

Perspective 3-point problem

- Define angles between rays as:

$$\cos\gamma_{kl} = \frac{\hat{\mathbf{x}}_k^T \hat{\mathbf{x}}_l}{||\hat{\mathbf{x}}_k|| \, ||\hat{\mathbf{x}}_l||}$$

- and the side lengths as: $l_{kl} = ||\mathbf{X}_k - \mathbf{X}_l||$

$$\begin{cases} s_1^2 + s_2^2 - 2s_1 s_2 \cos\gamma_{12} & = & l_{12}^2 \\ s_1^2 + s_3^2 - 2s_1 s_3 \cos\gamma_{13} & = & l_{13}^2 \\ s_2^2 + s_3^2 - 2s_2 s_3 \cos\gamma_{23} & = & l_{23}^2 \end{cases}$$

- This can be converted into a fourth degree polynomial, which is then solved.

Perspective 3-point problem

- Numerical issues when solving the 4th degree polynomial.

- Various approaches compared by Haralick et al. *Analysis and Solutions of the Three Point Perspective Pose Estimation Problem*, IJCV94

- P3P has up to four real solutions that have to be checked inside RANSAC.

Perspective 3-point problem

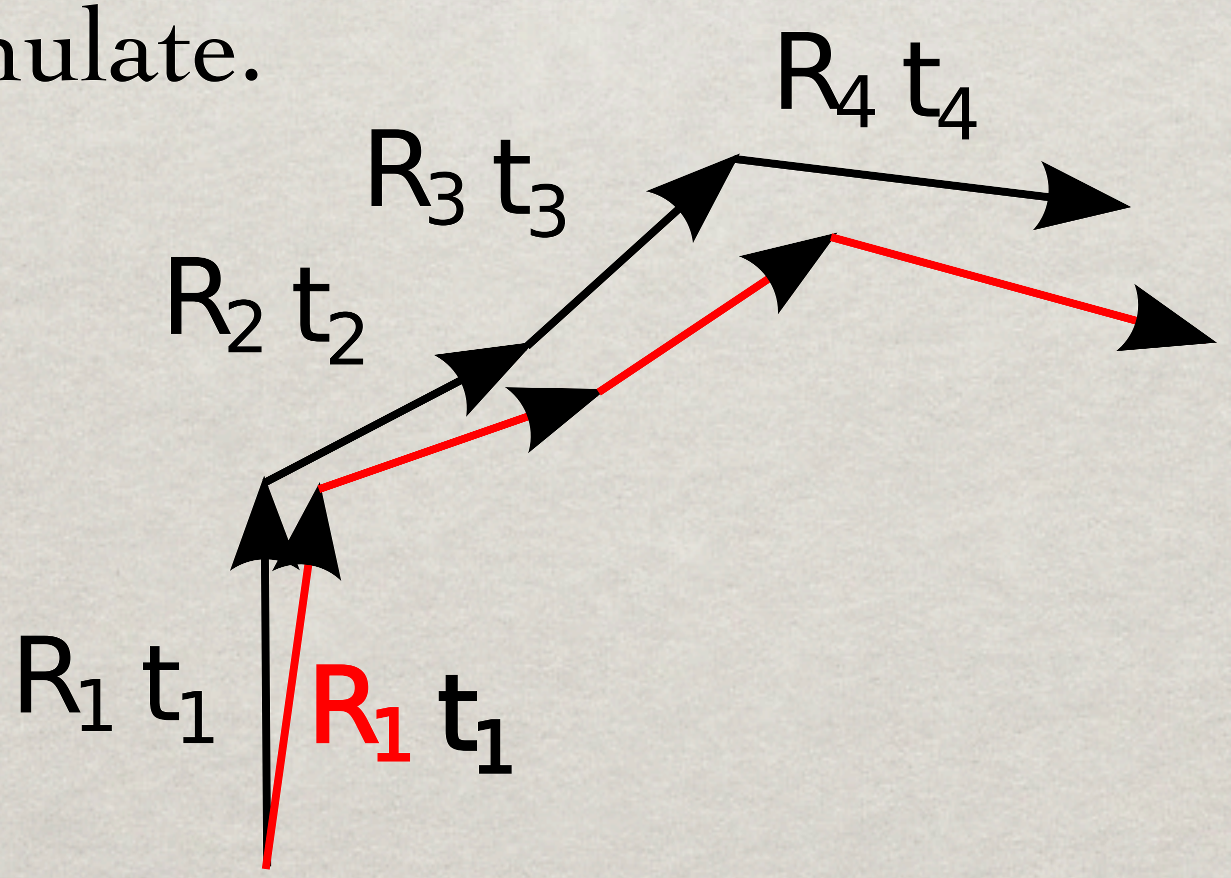
- Once we have solved P3P we have 3D points in the new camera.
- By relating these to the known 3D points in the world coordinate system, **R**, **t** are uniquely defined.

Bundle adjustment

- We can now build a decent 3D model by incrementally adding new cameras using P3P.
- But...

Bundle adjustment

- We can now build a decent 3D model by incrementally adding new cameras using P3P.
- But for long trajectories, errors will start to accumulate.



Bundle adjustment

- BA is essentially **ML** over all image correspondences given all cameras, and all 3D points.

$$\{\mathbf{R}^*, \mathbf{t}^*, \mathbf{X}^*\} = \arg \min_{\{\mathbf{R},\mathbf{t},\mathbf{X}\}} \sum_{k,l} d(\mathbf{x}_{kl}, \mathbf{K}[\mathbf{R}_k|\mathbf{t}_k]\mathbf{X}_l)^2$$

Bundle adjustment

- BA is essentially **ML** over all image correspondences given all cameras, and all 3D points. (Optionally also intrinsics.)

$$\{\mathbf{R}^*, \mathbf{t}^*, \mathbf{X}^*\} = \arg \min_{\{\mathbf{R},\mathbf{t},\mathbf{X}\}} \sum_{k,l} d(\mathbf{x}_{kl}, \mathbf{K}[\mathbf{R}_k|\mathbf{t}_k]\mathbf{X}_l)^2$$

- **Needs initial guess**. (Obtained by RANSAC on 5-point method and P3P)

- **Open source** SBA, by M. Lourakis et al.

Bundle adjustment

- The choice of **parametrisation** of 3D points, and camera rotations is important.
- If both near and far points are seen, it might be better to use $\mathbf{X} = [X_1, X_2, X_3, X_4]^T$ than $\mathbf{X} = [X_1, X_2, X_3, 1]^T$
- Good choices for rotations are unit quarternions, and axis-angle (see le 7)

Bundle adjustment

- Descent on the cost function is typically done using a regularized Newton method, such as Levenberg-Marquardt

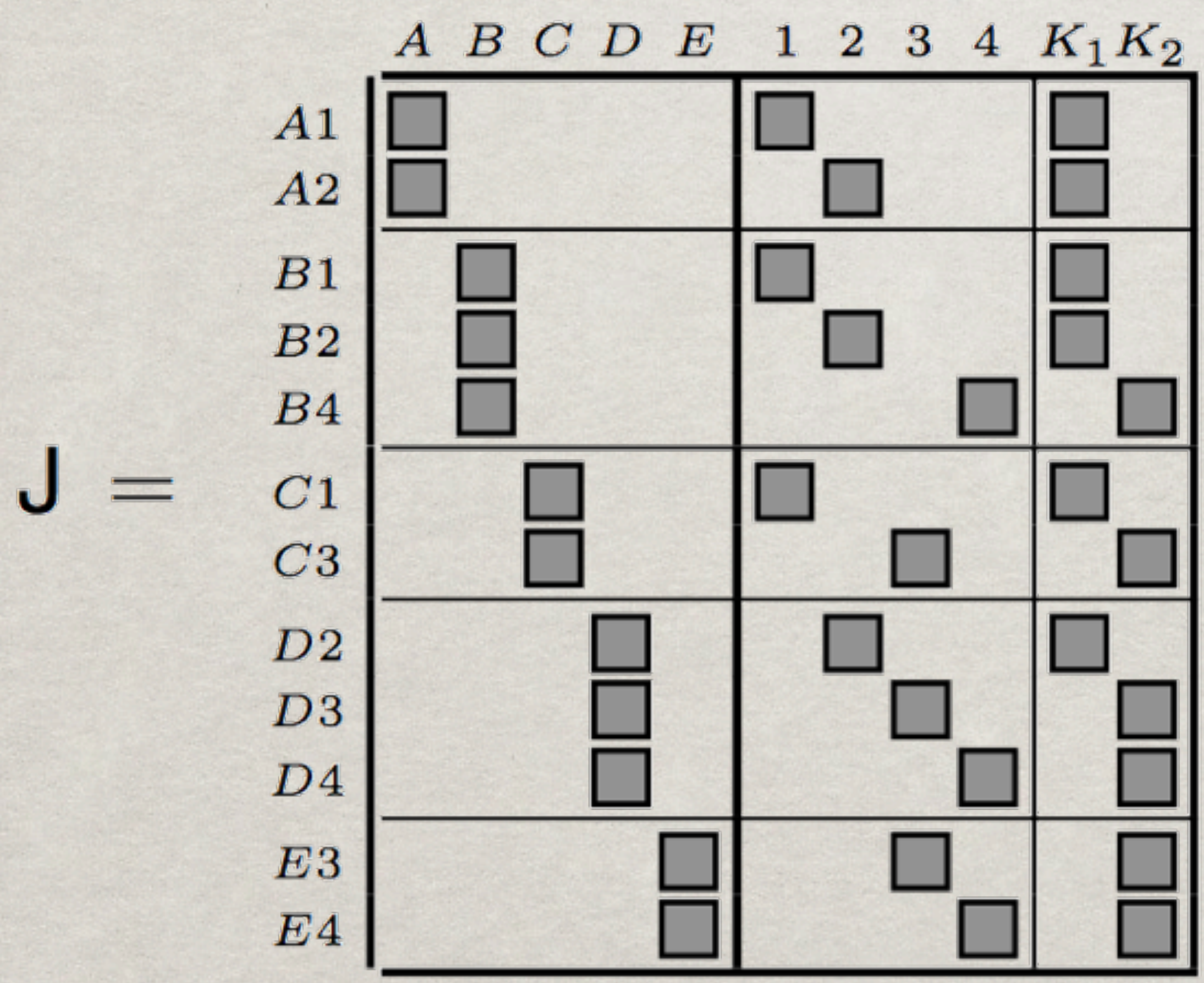
$$\min \sum_{k,l} (\mathbf{x}_{kl} - \mathbf{f}(\mathbf{X}_k, \mathbf{S}_l))^2$$

$$\sum_{k,l} (\mathbf{x}_{kl} - f(\mathbf{X}_k, \mathbf{S}_l + \delta_l))^2 \approx \sum_{k,l} (\mathbf{x}_{kl} - f(\mathbf{X}_k, \mathbf{S}_l) + \mathbf{J}_{kl}\delta_l)^2$$

- Block structure. Should be utilised for speed!

Bundle adjustment

- Block structure. Should be utilised for speed!



A-E	features
1-4	cameras
K_1,K_2	intrinsics

Bundle adjustment

- Too many details to mention.
- See the paper: Triggs et al., *Bundle Adjustment - A Modern Synthesis*, LNCS Book chapter, 2000

Discussion

- Discussion of the paper:
  David Nistér, *An Efficient Solution to the Five-Point Relative Pose Problem*, CVPR'03

FOR NEXT WEEK...

- Quan *Invariants of Six Points and Projective Reconstruction From Three Uncalibrated Images* PAMI'95. Sec 1-3
- Ondrej Chum and Jiri Matas, *Matching with PROSAC – Progressive Sample Consensus,* CVPR'05