Fundamental Matrix & Structure from Motion

Instructor - Simon Lucey

16-423 - Designing Computer Vision Apps

Carnegie Mellon
THE ROBOTICS INSTITUTE

Today

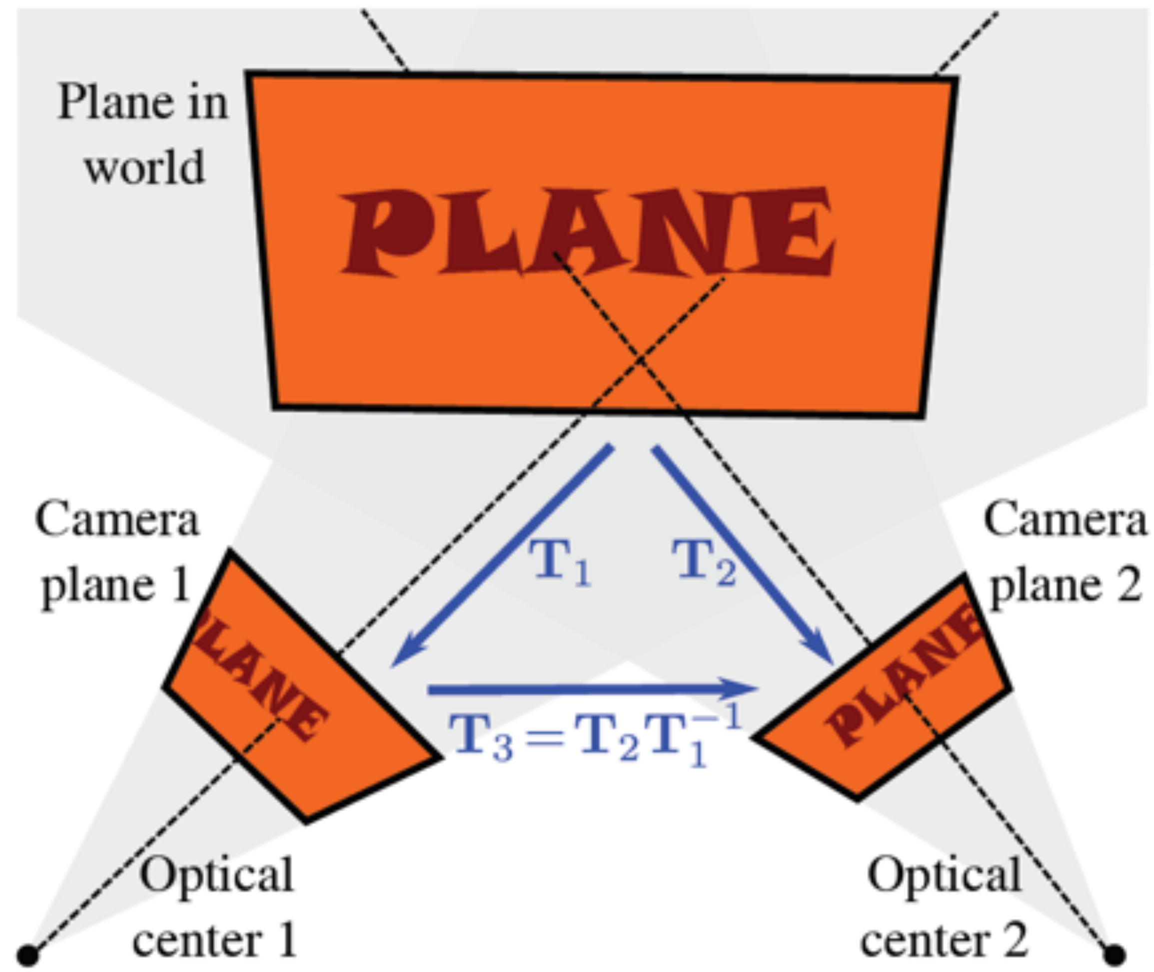
- Transformations between images
- Structure from Motion
- The Essential Matrix
- The Fundamental Matrix

Transformations between images

- So far we have considered transformations between the image and a plane in the world

- Now consider two cameras viewing the same plane

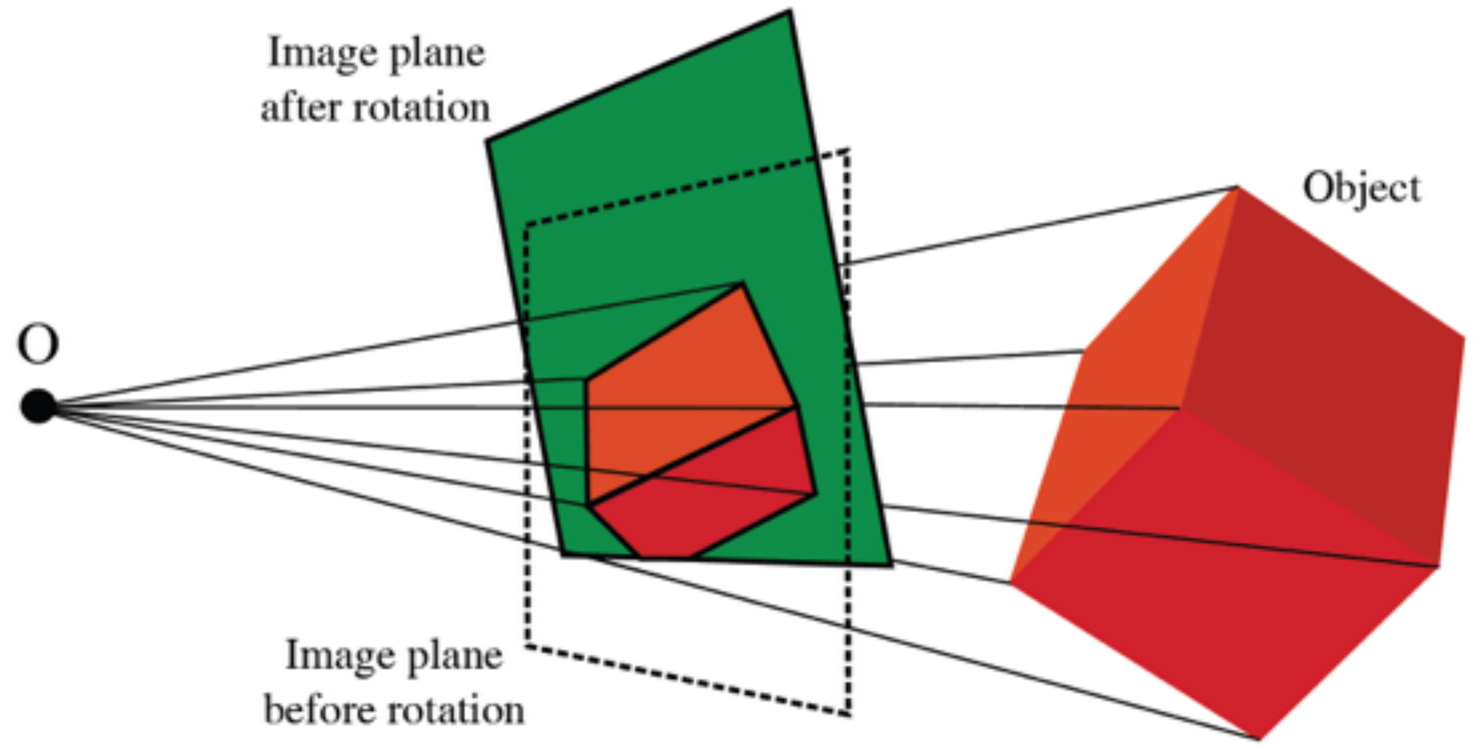
- There is a homography between camera 1 and the plane and a second homography between camera 2 and the plane



- It follows that the relation between the two images is also a homography

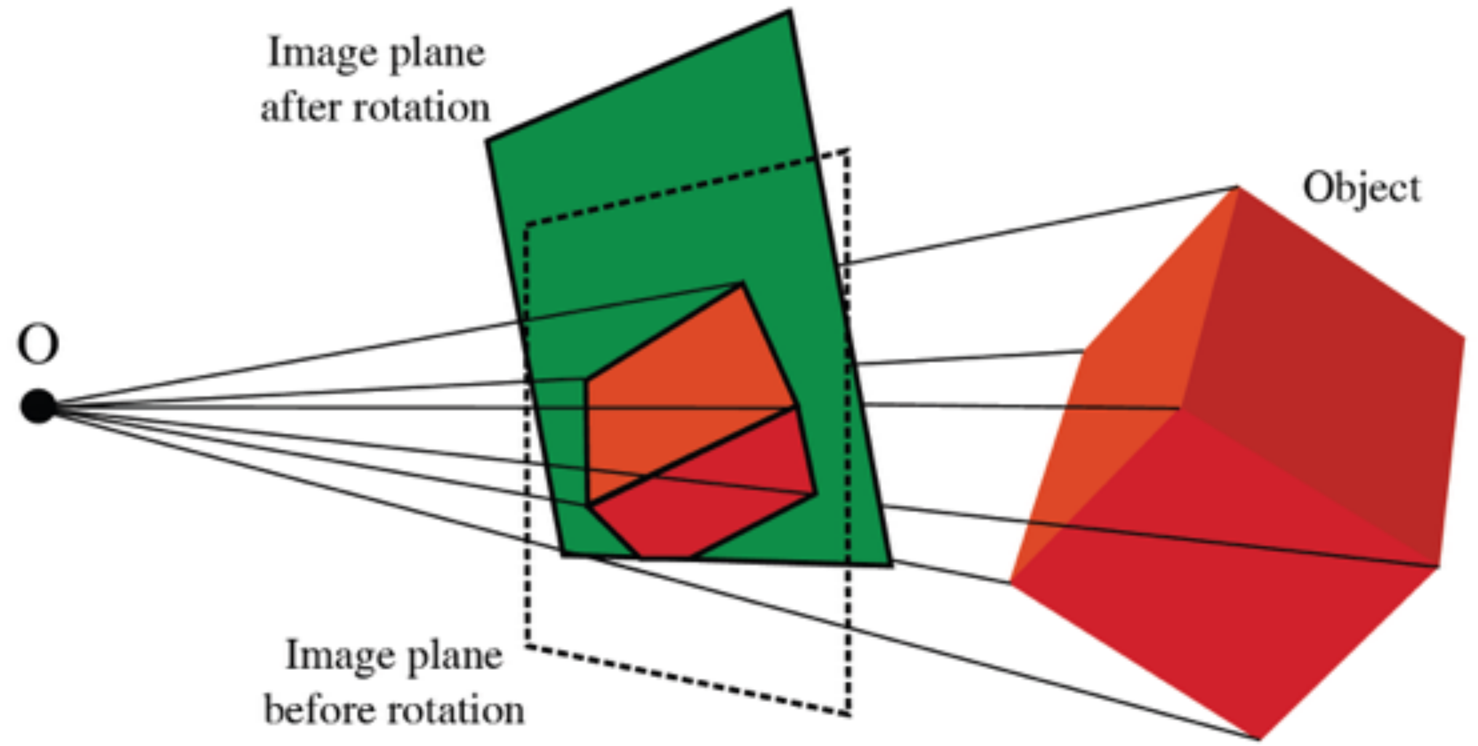
Camera under pure rotation

Special case is camera under pure rotation.
Homography can be showed to be $\boldsymbol{\Phi} = \boldsymbol{\Lambda}\boldsymbol{\Omega}_2\boldsymbol{\Lambda}^{-1}$



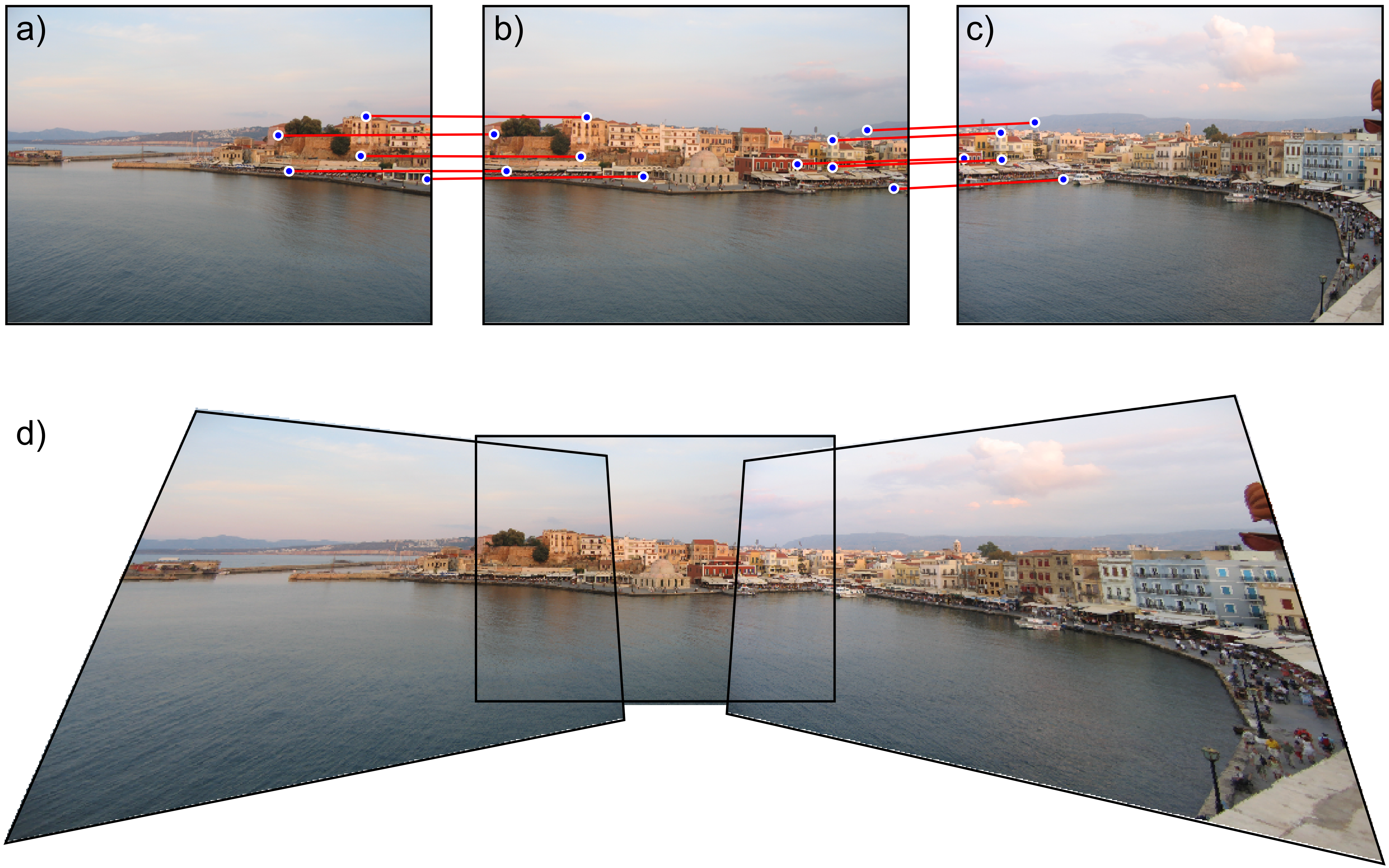
Camera under pure rotation

Special case is camera under pure rotation.
Homography can be showed to be $\mathbf{\Phi} = \mathbf{\Lambda}\mathbf{\Omega}_2\mathbf{\Lambda}^{-1}$



Why is this?

Panorama Example



Review - Homography

- Start with basic projection equation:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \tau_x \\ \omega_{21} & \omega_{22} & \omega_{23} & \tau_y \\ \omega_{31} & \omega_{32} & \omega_{33} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ 0 \\ 1 \end{bmatrix}$$

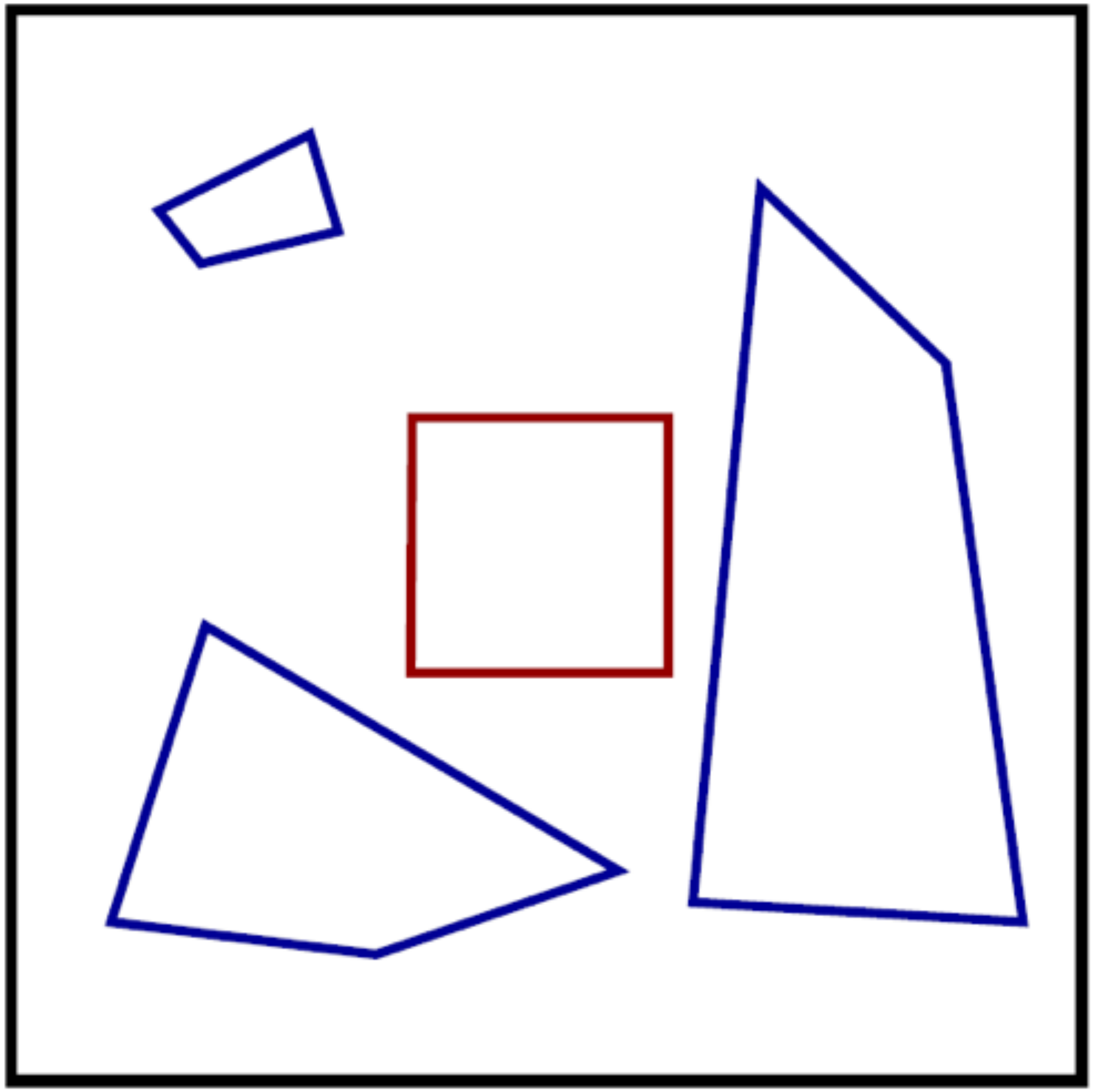
$$= \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ \omega_{31} & \omega_{32} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Combining these two matrices we get:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Review - Homography



Homogeneous:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Cartesian:

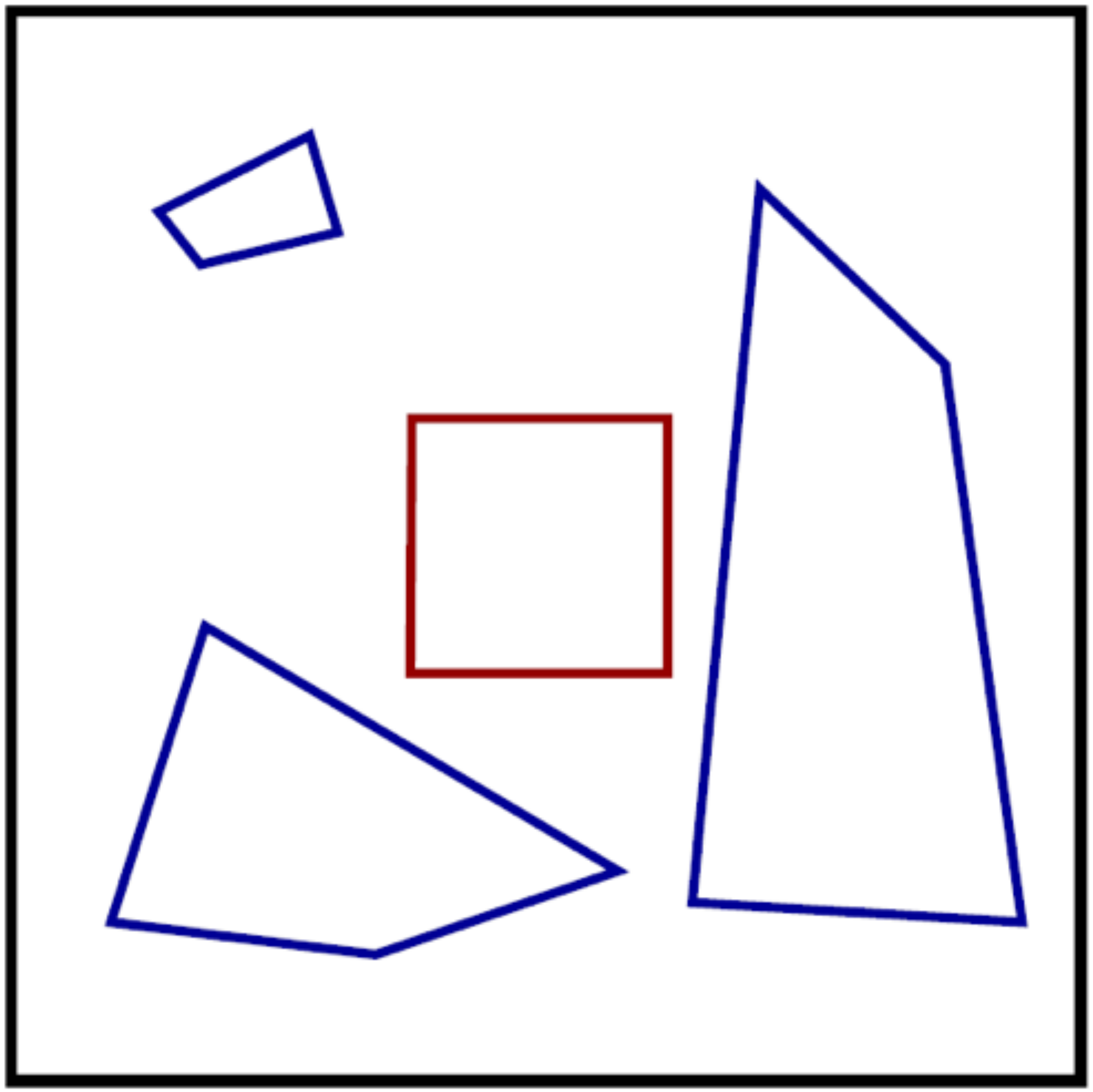
$$x = \frac{\phi_{11}u + \phi_{12}v + \phi_{13}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

$$y = \frac{\phi_{21}u + \phi_{22}v + \phi_{23}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

For short:

$$\mathbf{x} = \mathbf{hom}[\mathbf{w}, \boldsymbol{\Phi}]$$

Review - Homography



Homogeneous:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Cartesian:

$$x = \frac{\phi_{11}u + \phi_{12}v + \phi_{13}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

$$y = \frac{\phi_{21}u + \phi_{22}v + \phi_{23}}{\phi_{31}u + \phi_{32}v + \phi_{33}}$$

For short:

$$\mathbf{x} = \mathbf{hom}[\mathbf{w}, \mathbf{\Phi}]$$

How many unknowns?

Homography Estimation

- Re-arrange cartesian equations,

$$x(\phi_{31}u + \phi_{32}v + \phi_{33}) = \phi_{11}u + \phi_{12}v + \phi_{13}$$

$$y(\phi_{31}u + \phi_{32}v + \phi_{33}) = \phi_{21}u + \phi_{22}v + \phi_{23}$$

- Form linear system $\mathbf{A}\phi = \mathbf{0}$ s.t $||\phi||_2^2 = 1$

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & y_1u_1 & y_1v_1 & y_1 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 & -x_1 \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & y_2u_2 & y_2v_2 & y_2 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -x_2v_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_I & -v_I & -1 & y_Iu_I & y_Iv_I & y_I \\ u_I & v_I & 1 & 0 & 0 & 0 & -x_Iu_I & -x_Iv_I & -x_I \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \phi_{13} \\ \phi_{21} \\ \phi_{22} \\ \phi_{23} \\ \phi_{31} \\ \phi_{32} \\ \phi_{33} \end{bmatrix} = \mathbf{0},$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Homography Estimation

- In MATLAB this becomes,

```
>> [U,S,V] = svd(A);
>> Phi = reshape(V(:,end),[3,3])';
```

- Both sides are 3x1 vectors; should be parallel, so cross product will be zero

$$\tilde{\mathbf{x}} \times \mathbf{\Phi}\tilde{\mathbf{w}} = \mathbf{0}$$

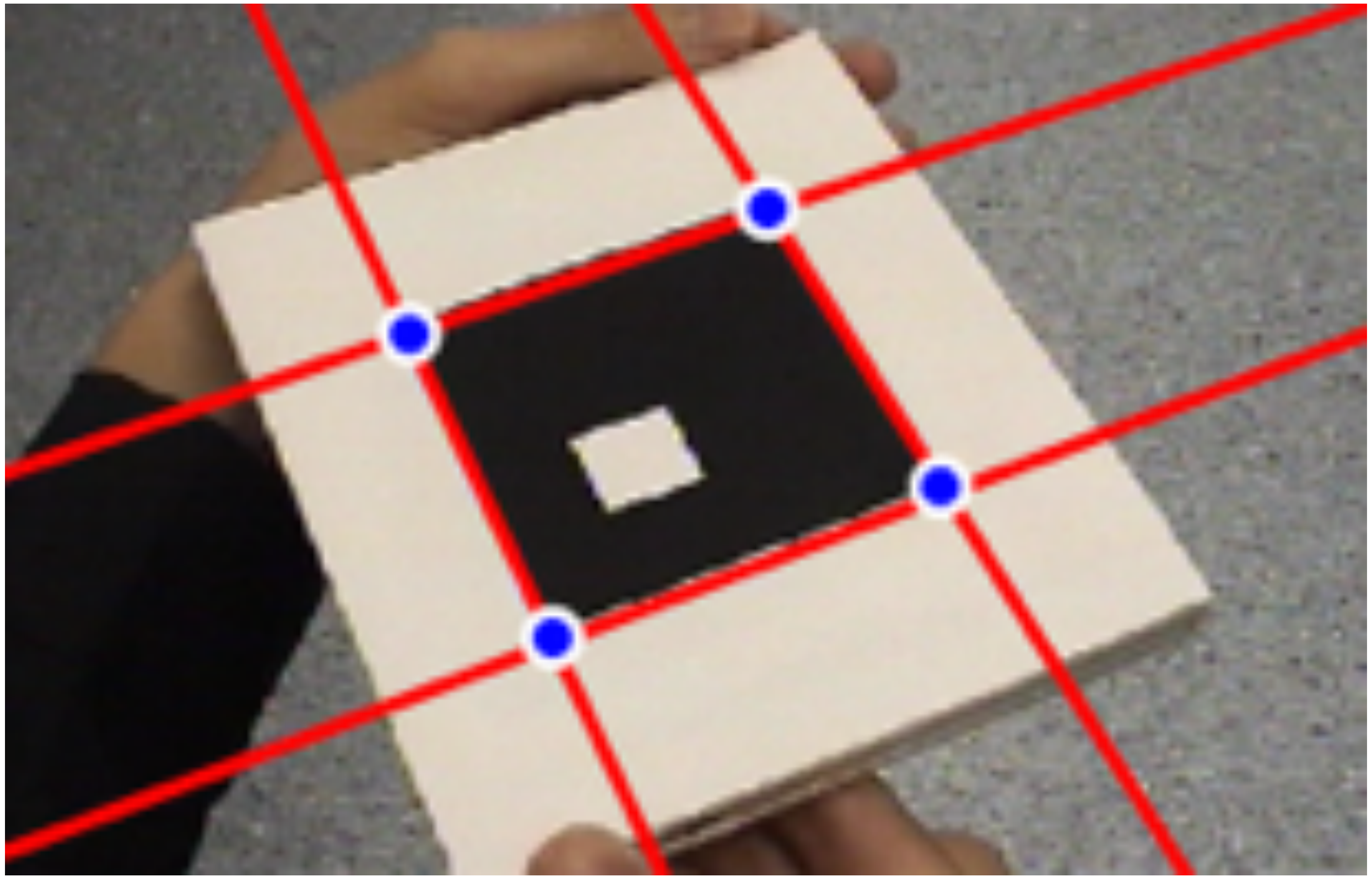
- For you to try MATLAB,

```
>> x = [randn(2,1);1]; cross(x,4*x)
```

Caution…

- Approach only minimizes algebraic error **NOT** the re-projection error!!!
- Need to employ non-linear optimization.

Estimating Extrinsics



$$\hat{\Omega}, \hat{\tau}$$

Estimating Extrinsics

- Writing out the camera equations in full

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & D \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ \omega_{31} & \omega_{32} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Estimate the homography from matched points
- Factor out the intrinsic parameters

$$\begin{bmatrix} \phi'_{11} & \phi'_{12} & \phi'_{13} \\ \phi'_{21} & \phi'_{22} & \phi'_{23} \\ \phi'_{31} & \phi'_{32} & \phi'_{33} \end{bmatrix} = \lambda' \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ \omega_{31} & \omega_{32} & \tau_z \end{bmatrix}$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Estimating Extrinsics

- Find the last column using the cross product of first two columns
- Make sure the determinant is 1. If it is -1, then multiply last column by -1.
- Find translation scaling factor between old and new values

$$\lambda' = \frac{\sum_{m=1}^{3}\sum_{n=1}^{2}\phi'_{mn}/\omega_{mn}}{6}$$

- Finally, set $\boldsymbol{\tau} = [\phi'_{13}, \phi'_{23}, \phi'_{33}]^T/\lambda'$

Augmented Reality

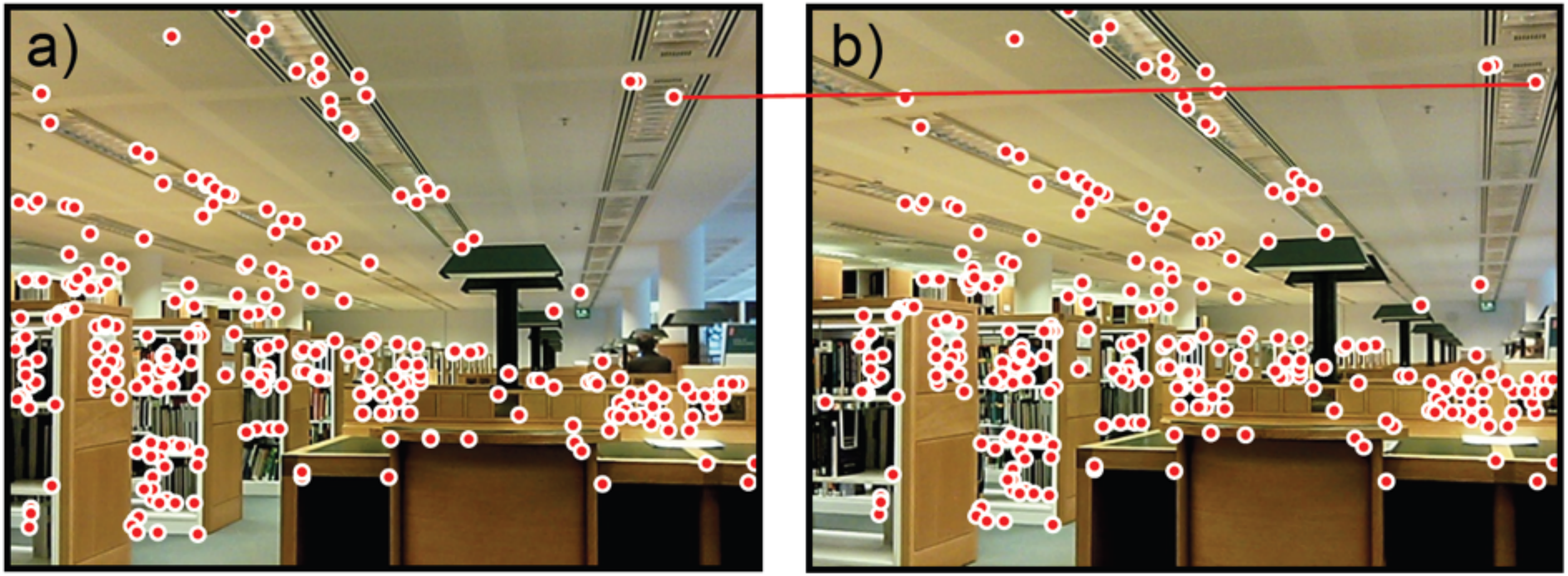
Today

- Transformations between images
- Structure from Motion
- The Essential Matrix
- The Fundamental Matrix

Taken from Agarwal et al. “Building Rome in a Day”, ICCV 2009.

Taken from Agarwal et al. "Building Rome in a Day", ICCV 2009.

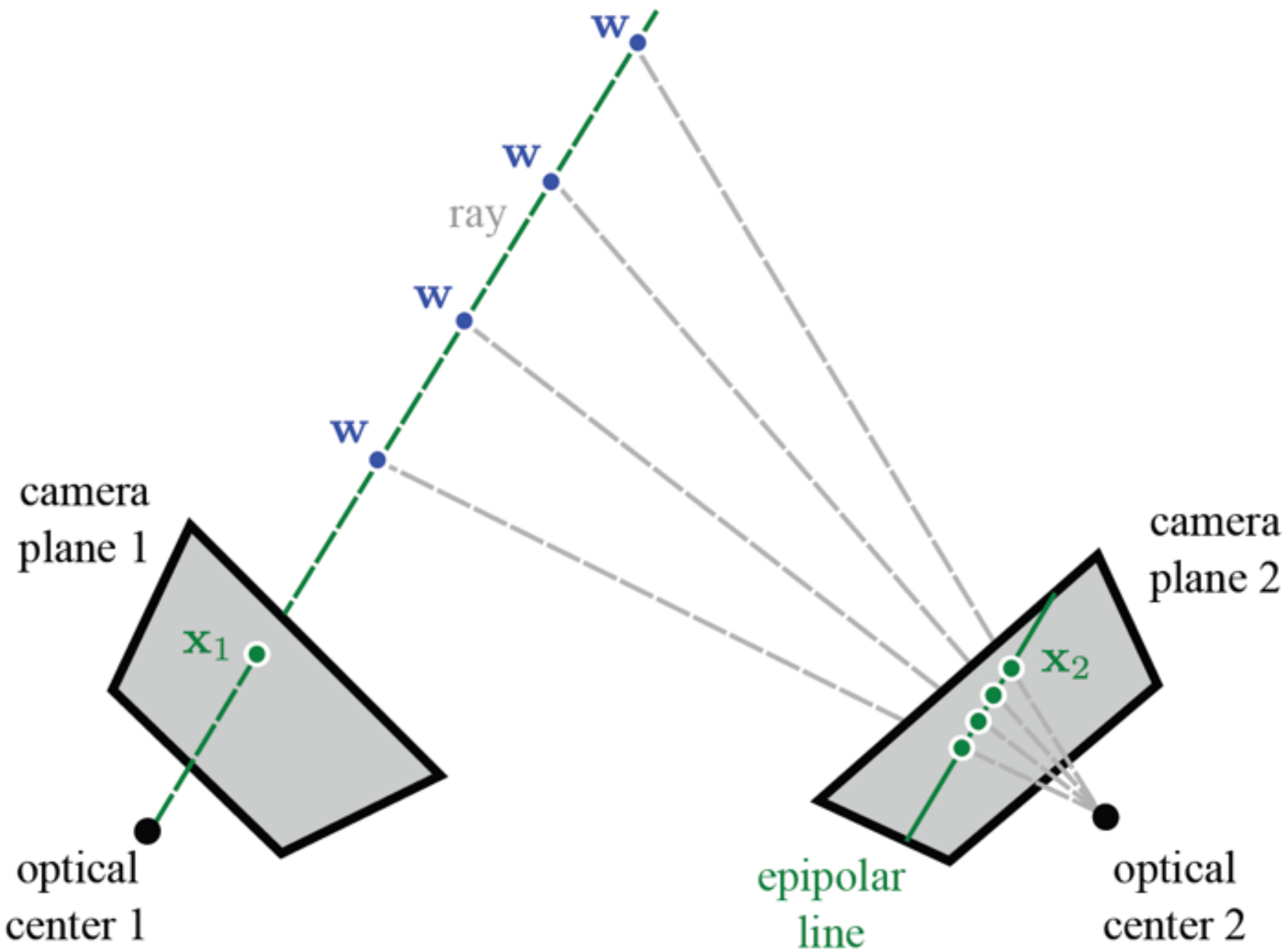
Structure from Motion



For simplicity, we'll start with simpler problem

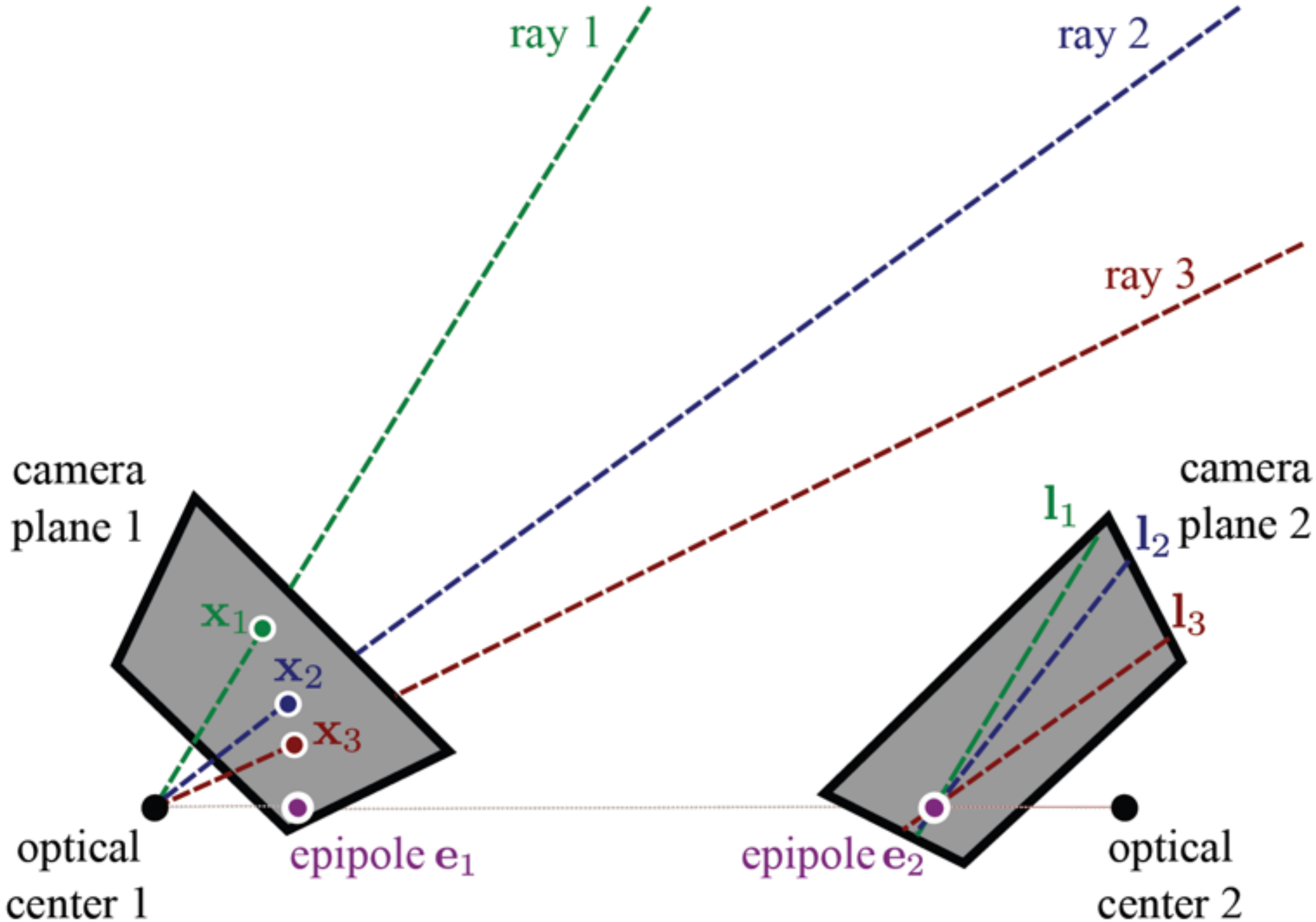
- Just J=2 images
- Known intrinsic matrix Λ

Epipolar lines



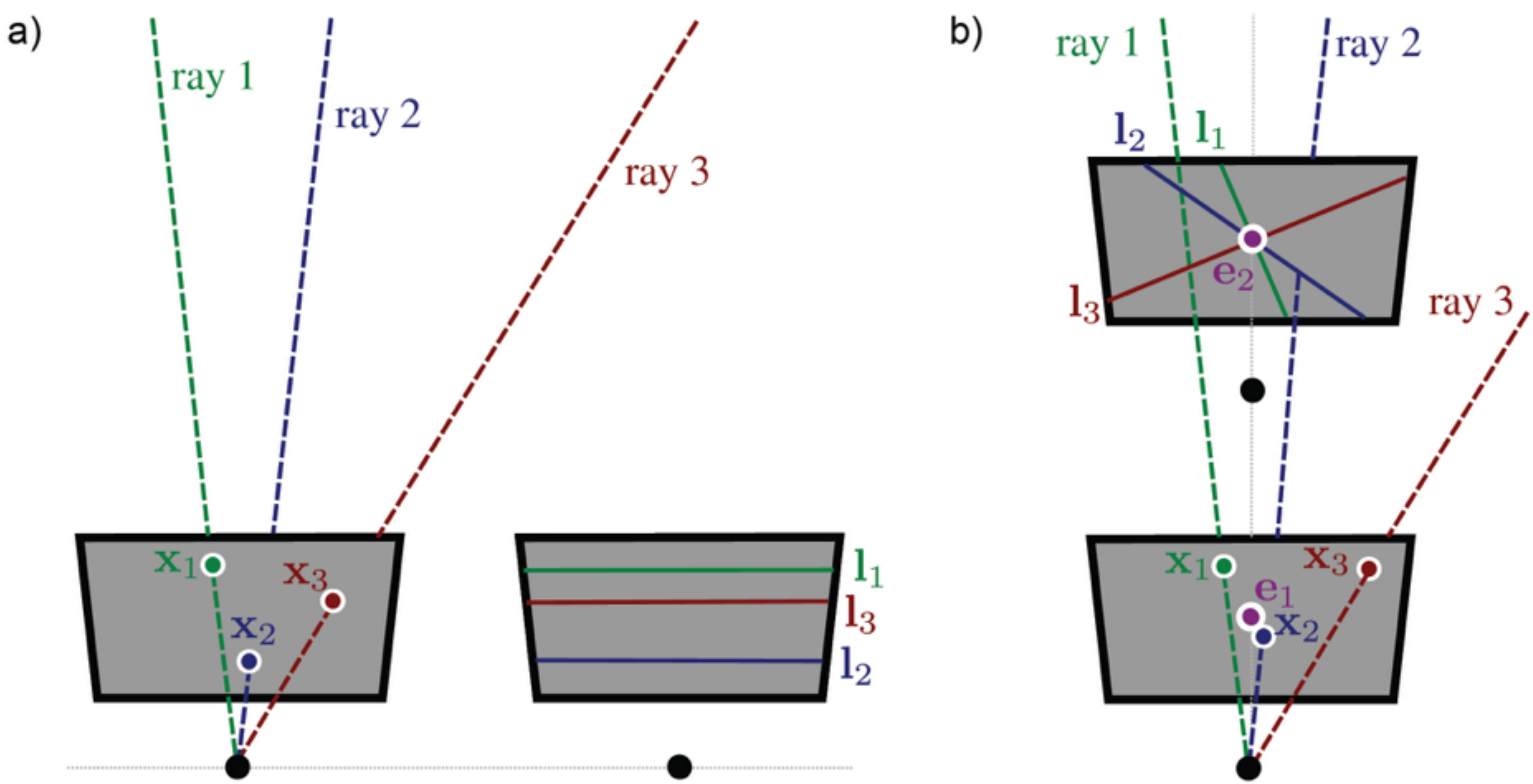
Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Epipole



Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Special configurations



Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Today

- Transformations between images
- Structure from Motion
- The Essential Matrix
- The Fundamental Matrix

Review - Cross Product

- Cross product between two 3D vectors is defined as,

$$\mathbf{c} = \mathbf{a} \times \mathbf{b}$$

- Operation is equivalent to the,

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

- Can be written in short as,

$$\mathbf{c} = [\mathbf{a}]_\times \mathbf{b}$$

Review - Cross Product

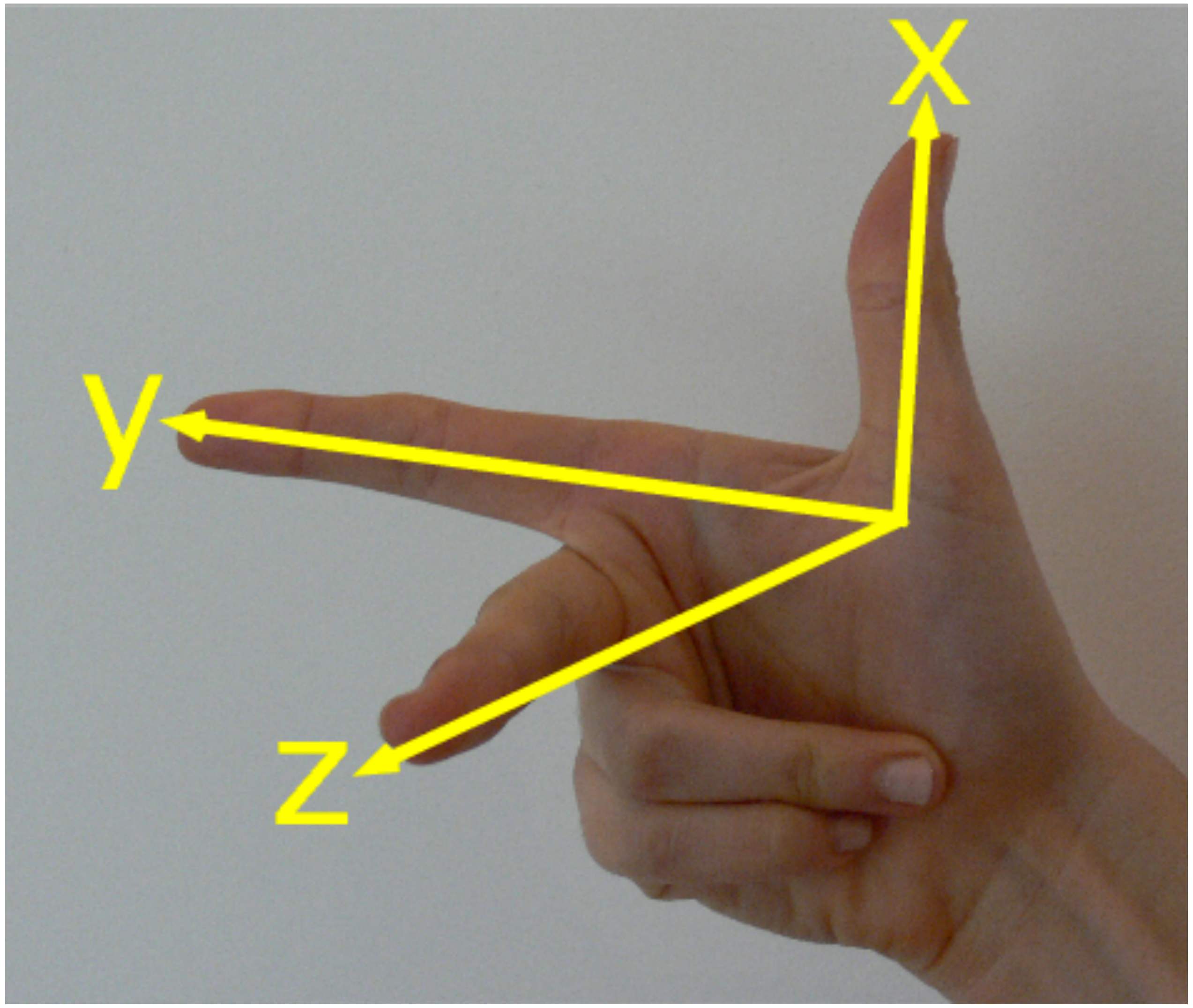
- In other words,

$$\mathbf{a}^T(\mathbf{a} \times \mathbf{b}) = \mathbf{b}^T(\mathbf{a} \times \mathbf{b}) = 0$$

Right-hand rule

- Given three orthonormal 3D vectors $(\mathbf{x}, \mathbf{y}, \mathbf{z})$

- Then, $\mathbf{z} = \mathbf{x} \times \mathbf{y}$



The Essential Matrix

First camera:

$$\lambda_1 \tilde{\mathbf{x}}_1 = \mathbf{w}$$

Second camera:

$$\lambda_2 \tilde{\mathbf{x}}_2 = \boldsymbol{\Omega}\mathbf{w} + \boldsymbol{\tau}$$

Substituting:

$$\lambda_2 \tilde{\mathbf{x}}_2 = \lambda_1 \boldsymbol{\Omega}\tilde{\mathbf{x}}_1 + \boldsymbol{\tau}$$

This is a mathematical relationship between the points in the two images, but it's not in the most convenient form.

The Essential Matrix

$$\lambda_2 \tilde{\mathbf{x}}_2 = \lambda_1 \boldsymbol{\Omega} \tilde{\mathbf{x}}_1 + \boldsymbol{\tau}$$

$$\lambda_2 \boldsymbol{\tau} \times \tilde{\mathbf{x}}_2 = \lambda_1 \boldsymbol{\tau} \times \boldsymbol{\Omega} \tilde{\mathbf{x}}_1$$

$$\tilde{\mathbf{x}}_2^T \boldsymbol{\tau} \times \boldsymbol{\Omega} \tilde{\mathbf{x}}_1 = 0$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

The Essential Matrix

$$\tilde{\mathbf{x}}_2^T \boldsymbol{\tau} \times \boldsymbol{\Omega} \tilde{\mathbf{x}}_1 = 0$$

The cross product term can be expressed as a matrix

$$\boldsymbol{\tau}_\times = \begin{bmatrix} 0 & -\tau_z & \tau_y \\ \tau_z & 0 & -\tau_x \\ -\tau_y & \tau_x & 0 \end{bmatrix}$$

Defining:

$$\mathbf{E} = \boldsymbol{\tau}_\times \boldsymbol{\Omega}$$

We now have the essential matrix relation

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

Properties of the Essential Matrix

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

- Rank 2: $\det[\mathbf{E}] = 0$
- 5 degrees of freedom
- Non-linear constraints between elements

$$2\mathbf{E}\mathbf{E}^T\mathbf{E} - \mathrm{trace}[\mathbf{E}\mathbf{E}^T]\mathbf{E} = 0$$

Recovering Epipolar Lines

Equation of a line: $ax + by + c = 0$

or $$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

or $$\mathbf{l}^T \tilde{\mathbf{x}} = 0$$

Recovering Epipolar Lines

Equation of a line: $\mathbf{l}^T\tilde{\mathbf{x}} = 0$

Now consider $\tilde{\mathbf{x}}_2^T\mathbf{E}\tilde{\mathbf{x}}_1 = 0$

This has the form $\mathbf{l}_1^T\tilde{\mathbf{x}}_1 = 0$ where $\mathbf{l}_1 = \tilde{\mathbf{x}}_2^T\mathbf{E}$

So the epipolar lines are

$$\begin{aligned}\mathbf{l}_1 &= \tilde{\mathbf{x}}_2^T\mathbf{E}\\ \mathbf{l}_2 &= \tilde{\mathbf{x}}_1^T\mathbf{E}^T\end{aligned}$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Recovering Epipolar Lines

Every epipolar line in image 1 passes through the epipole $\mathbf{e}_1$.

In other words $\tilde{\mathbf{x}}_2^T \mathbf{E} \tilde{\mathbf{e}}_1 = 0$ for ALL $\tilde{\mathbf{x}}_2^T$

This can only be true if $\mathbf{e}_1$ is in the nullspace of **E**.

$$\tilde{\mathbf{e}}_1 = \mathbf{null}[\mathbf{E}]$$

Similarly:

$$\tilde{\mathbf{e}}_2 = \mathbf{null}[\mathbf{E}^T]$$

We find the null spaces by computing $\mathbf{E} = \mathbf{ULV}^T$, and taking the last column of $\mathbf{V}$ and the last row of $\mathbf{U}$.

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Decomposition of E

Essential matrix: $\mathbf{E} = \boldsymbol{\tau}_{\times}\boldsymbol{\Omega}$

To recover translation and rotation use the matrix:

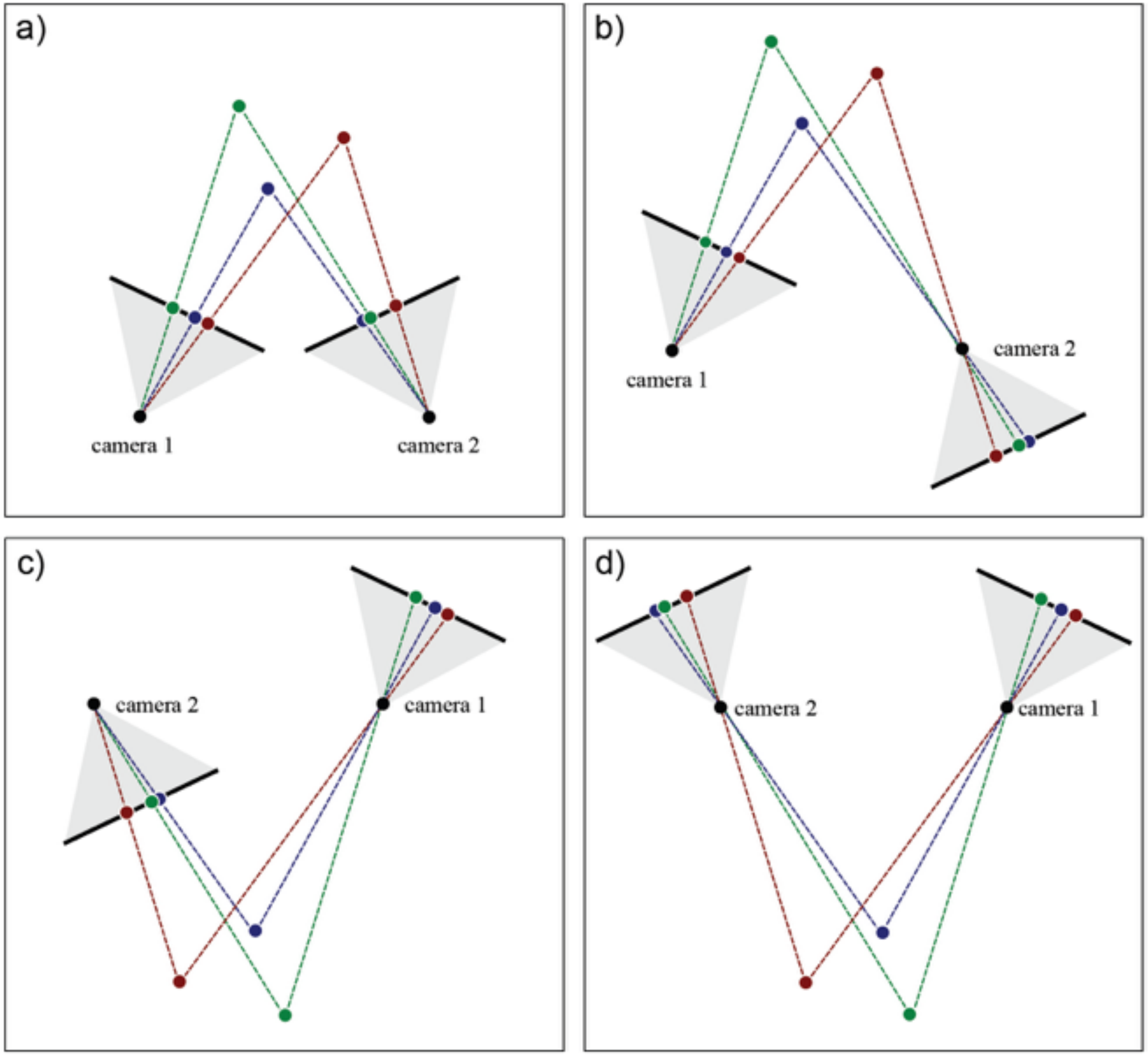
$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We take the SVD $\mathbf{E} = \mathbf{U}\mathbf{L}\mathbf{V}^T$ and then we set

$$\begin{aligned} \boldsymbol{\tau}_{\times} &= \mathbf{U}\mathbf{L}\mathbf{W}\mathbf{U}^T \\ \boldsymbol{\Omega} &= \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T \end{aligned}$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Four Interpretations



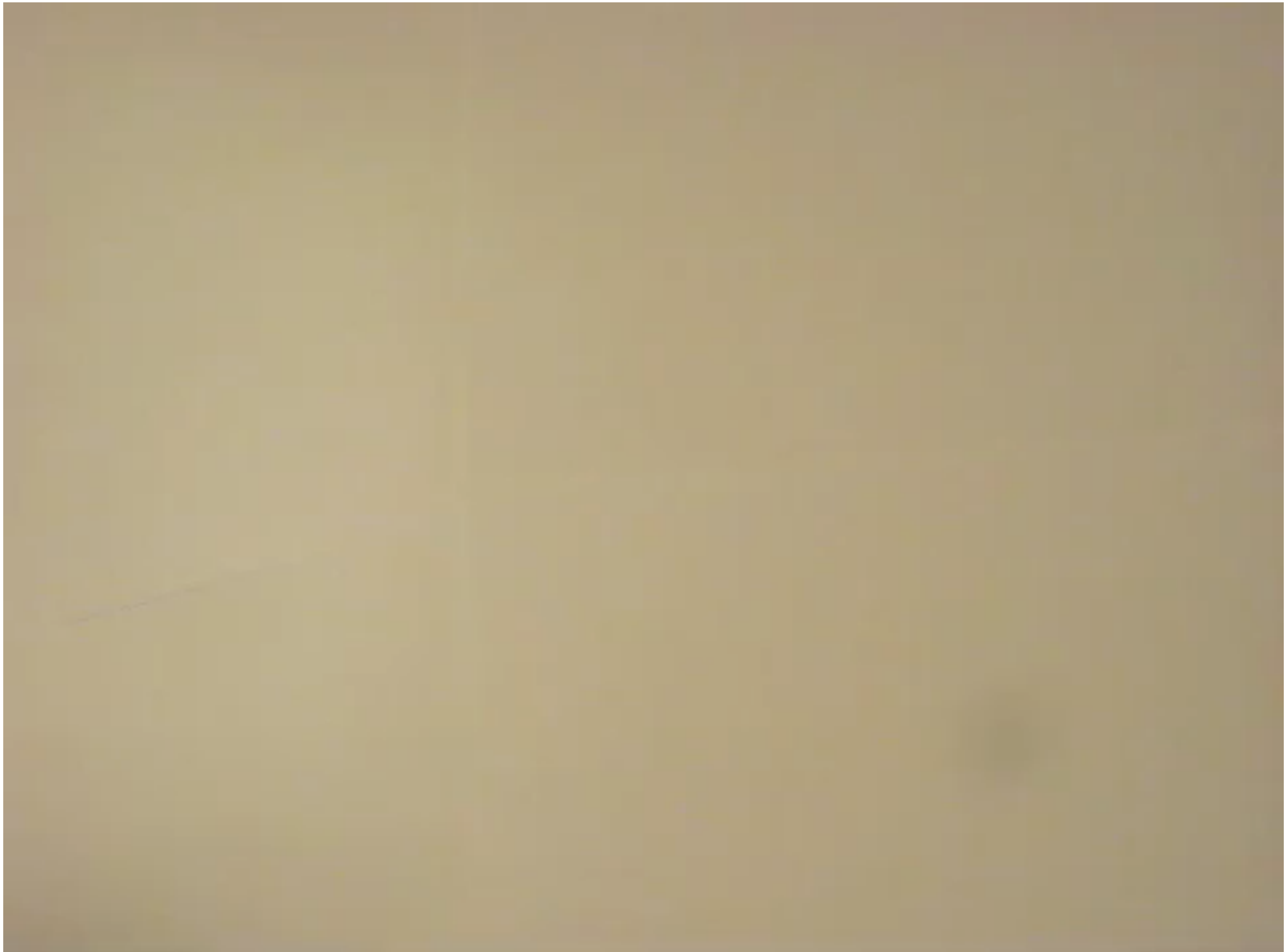
To get the different solutions, we mutliply τ by -1 and substitute

$$\mathbf{W} \text{ for } \mathbf{W}^{-1}$$

Today

- Transformations between images
- Structure from Motion
- The Essential Matrix
- The Fundamental Matrix

The Fundamental Matrix Song



Taken from Daniel Wedge's home page - http://danielwedge.com/fmatrix/

The Fundamental Matrix Song

Taken from Daniel Wedge's home page - http://danielwedge.com/fmatrix/

The Fundamental Matrix

Now consider two cameras that are not normalised

$$\begin{aligned} \lambda_1 \tilde{\mathbf{x}}_1 &= \boldsymbol{\Lambda}_1[\mathbf{I}, \mathbf{0}]\tilde{\mathbf{w}} \\ \lambda_2 \tilde{\mathbf{x}}_2 &= \boldsymbol{\Lambda}_2[\boldsymbol{\Omega}, \boldsymbol{\tau}]\tilde{\mathbf{w}} \end{aligned}$$

By a similar procedure to before, we get the relation

$$\tilde{\mathbf{x}}_2^T \boldsymbol{\Lambda}_2^{-T} \mathbf{E} \boldsymbol{\Lambda}_1^{-1} \tilde{\mathbf{x}}_1 = 0$$

or

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

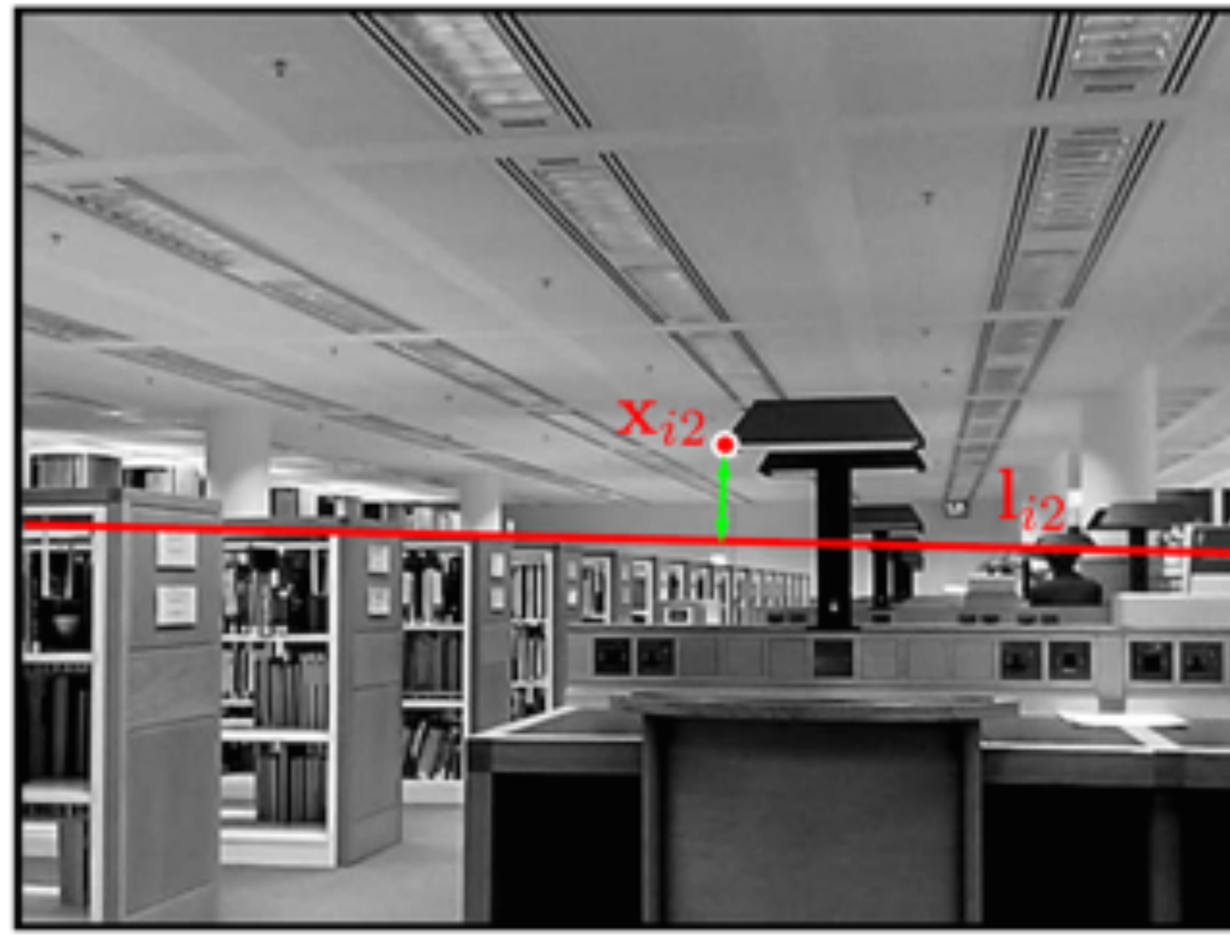
where

$$\mathbf{F} = \boldsymbol{\Lambda}_2^{-T} \mathbf{E} \boldsymbol{\Lambda}_1^{-1} = \boldsymbol{\Lambda}_2^{-T} \boldsymbol{\tau}_\times \boldsymbol{\Omega} \boldsymbol{\Lambda}_1^{-1}$$

Relation between essential and fundamental

$$\mathbf{E} = \boldsymbol{\Lambda}_2^T \mathbf{F} \boldsymbol{\Lambda}_1$$

Estimation of the Fundamental Matrix



Estimation of Fundamental Matrix

When the fundamental matrix is correct, the epipolar line induced by a point in the first image should pass through the matching point in the second image and vice-versa.

This suggests the criterion

$$\hat{\mathbf{F}} = \underset{\mathbf{F}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} \left((\text{dist}[\mathbf{x}_{i1}, \mathbf{l}_{i1}])^2 + (\text{dist}[\mathbf{x}_{i2}, \mathbf{l}_{i2}])^2 \right) \right]$$

If $\mathbf{l} = [a, b, c]$ and $\mathbf{x} = [x, y]^T$ $\text{dist}[\mathbf{x}, \mathbf{l}] = \dfrac{ax + by + c}{\sqrt{a^2 + b^2}}$

Unfortunately, there is no closed form solution for this quantity.

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

The 8 Point Algorithm

Approach:

- solve for fundamental matrix using homogeneous coordinates
- closed form solution (but to wrong problem!)
- Known as the 8 point algorithm

Start with fundamental matrix relation $\tilde{\mathbf{x}}_2^T \mathbf{F} \tilde{\mathbf{x}}_1 = 0$

Writing out in full:

$$\begin{bmatrix} x_{i2} & y_{i2} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_{i1} \\ y_{i1} \\ 1 \end{bmatrix}$$

or

$$x_{i2}x_{i1}f_{11}+x_{i2}y_{i1}f_{12}+x_{i2}f_{13}+y_{i2}x_{i1}f_{21}+y_{i2}y_{i1}f_{22}+y_{i2}f_{23}+x_{i1}f_{31}+y_{i1}f_{32}+f_{33}=0.$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

The 8 Point Algorithm

$$x_{i2}x_{i1}f_{11}+x_{i2}y_{i1}f_{12}+x_{i2}f_{13}+y_{i2}x_{i1}f_{21}+y_{i2}y_{i1}f_{22}+y_{i2}f_{23}+x_{i1}f_{31}+y_{i1}f_{32}+f_{33}=0.$$

Can be written as: $[x_{i2}x_{i1}, x_{i2}y_{i1}, x_{i2}, y_{i2}x_{i1}, y_{i2}y_{i1}, y_{i2}, x_{i1}, y_{i1}, 1]\mathbf{f}=0$

where $\mathbf{f} = [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^T$

Stacking together constraints from at least 8 pairs of points, we get the system of equations

$$\mathbf{Af} = \begin{bmatrix} x_{12}x_{11} & x_{12}y_{11} & x_{12} & y_{12}x_{11} & y_{12}y_{11} & y_{12} & x_{11} & y_{11} & 1 \\ x_{22}x_{21} & x_{22}y_{21} & x_{22} & y_{22}x_{21} & y_{22}y_{21} & y_{22} & x_{21} & y_{21} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{I2}x_{I1} & x_{I2}y_{I1} & x_{I2} & y_{I2}x_{I1} & y_{I2}y_{I1} & y_{I2} & x_{I1} & y_{I1} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}.$$

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

The 8 Point Algorithm

$$\mathbf{Af} = \begin{bmatrix} x_{12}x_{11} & x_{12}y_{11} & x_{12} & y_{12}x_{11} & y_{12}y_{11} & y_{12} & x_{11} & y_{11} & 1 \\ x_{22}x_{21} & x_{22}y_{21} & x_{22} & y_{22}x_{21} & y_{22}y_{21} & y_{22} & x_{21} & y_{21} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{I2}x_{I1} & x_{I2}y_{I1} & x_{I2} & y_{I2}x_{I1} & y_{I2}y_{I1} & y_{I2} & x_{I1} & y_{I1} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}.$$

Minimum direction problem of the form $\mathbf{Ab} = \mathbf{0}$
Find minimum of $|\mathbf{Ab}|^2$ subject to $|\mathbf{b}| = 1$.

To solve, compute the SVD $\mathbf{A} = \mathbf{ULV}^T$
and then set $\hat{\mathbf{b}}$ to the last column of $\mathbf{V}$.

Adapted from: Computer vision: models, learning and inference. Simon J.D. Prince

Fitting Concerns

- This procedure does not ensure that solution is rank 2. Solution: set last singular value to zero.
- Can be unreliable because of numerical problems to do with the data scaling – better to re-scale the data first
- Needs 8 points in general positions (cannot all be planar).
- Fails if there is not sufficient translation between the views
- Use this solution to start non-linear optimisation of true criterion (must ensure non-linear constraints obeyed).
- There is also a 7 point algorithm.

More to read…

- Prince et al.
  - Chapter 16, Sections 1-3.