

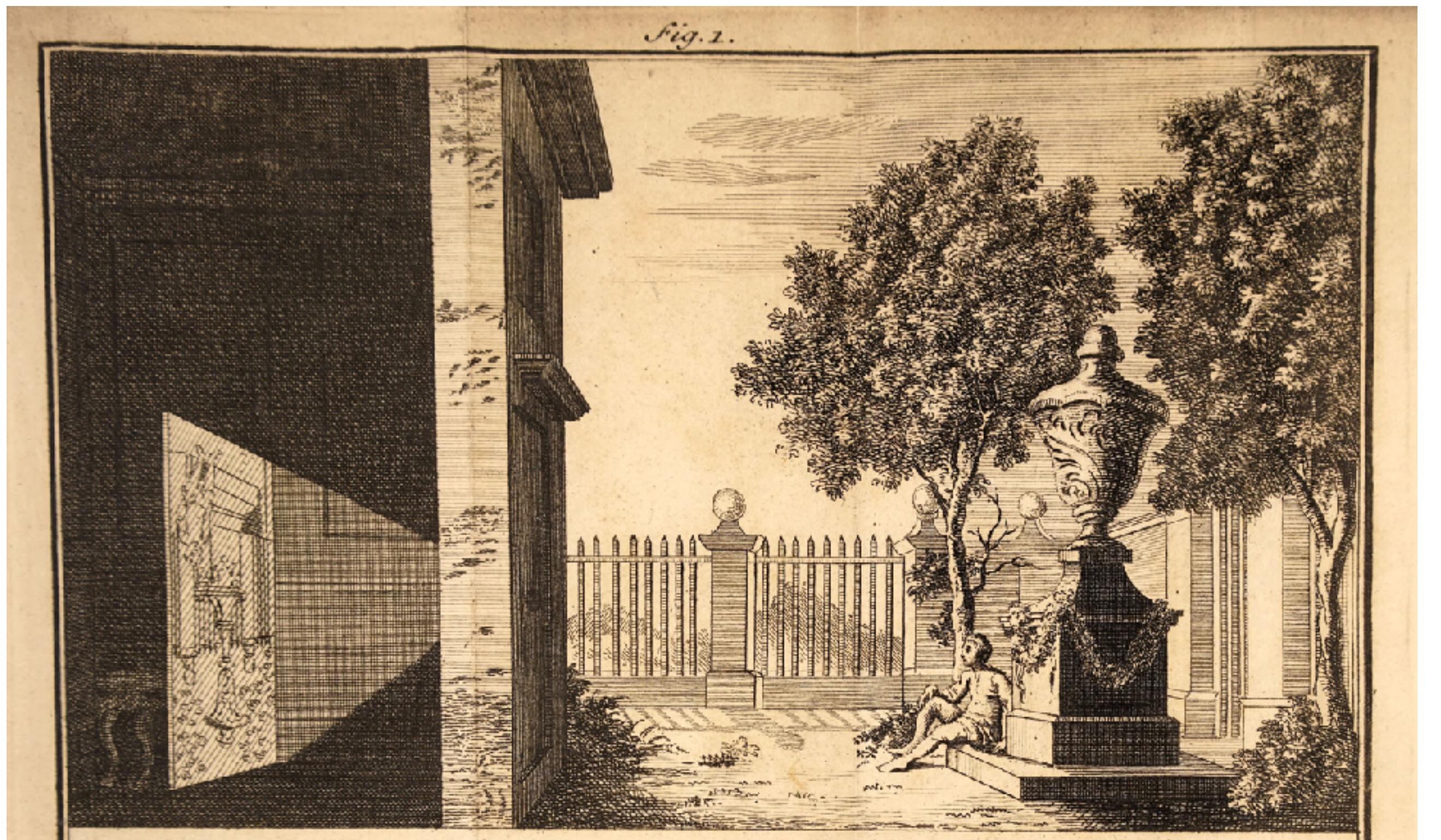
II: Multi-View Geometry

3D CV

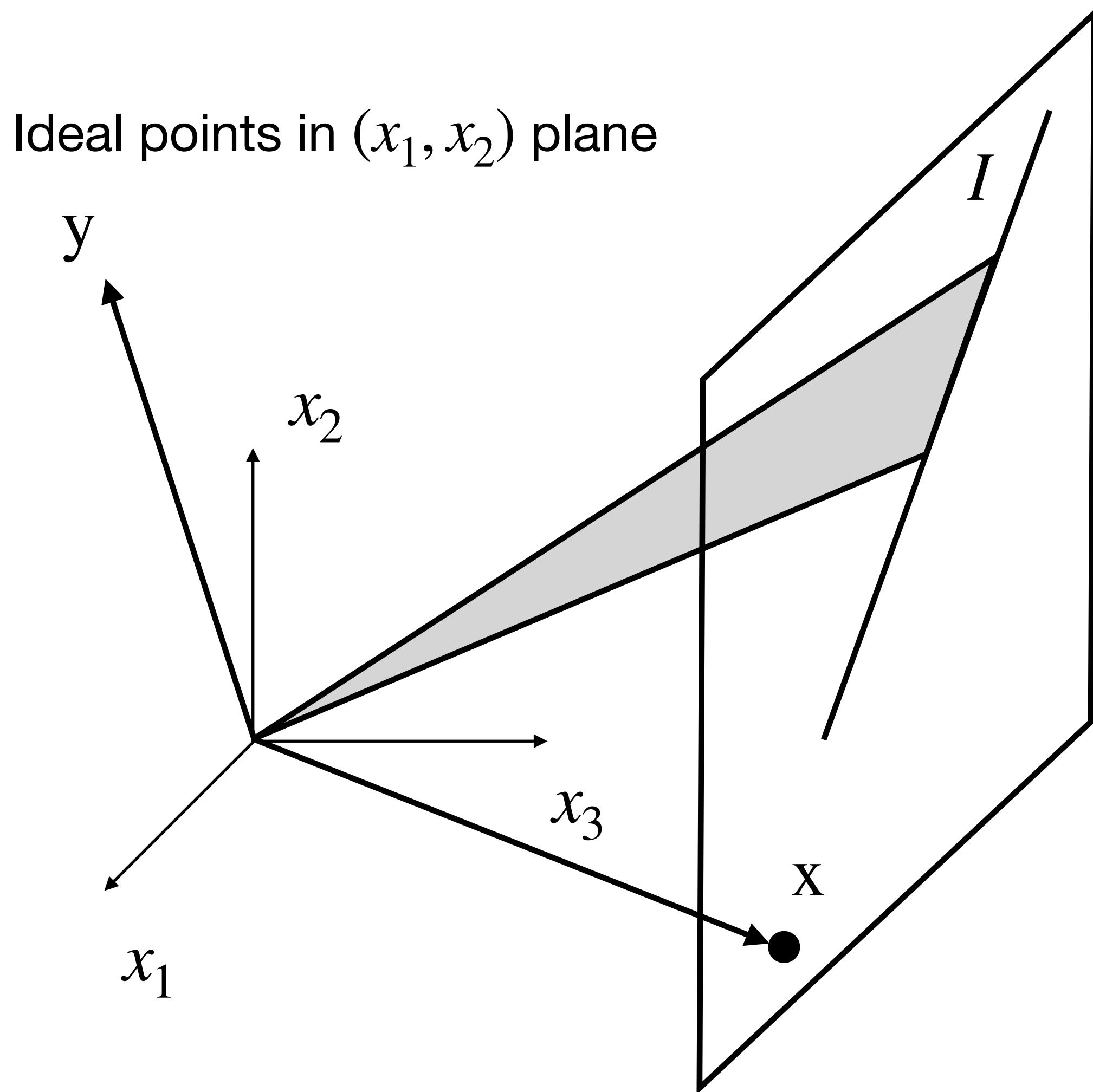
Kirill Struminsky

In the Previous Episode

- Image formation
- Homogeneous coordinates
- Projective transformations

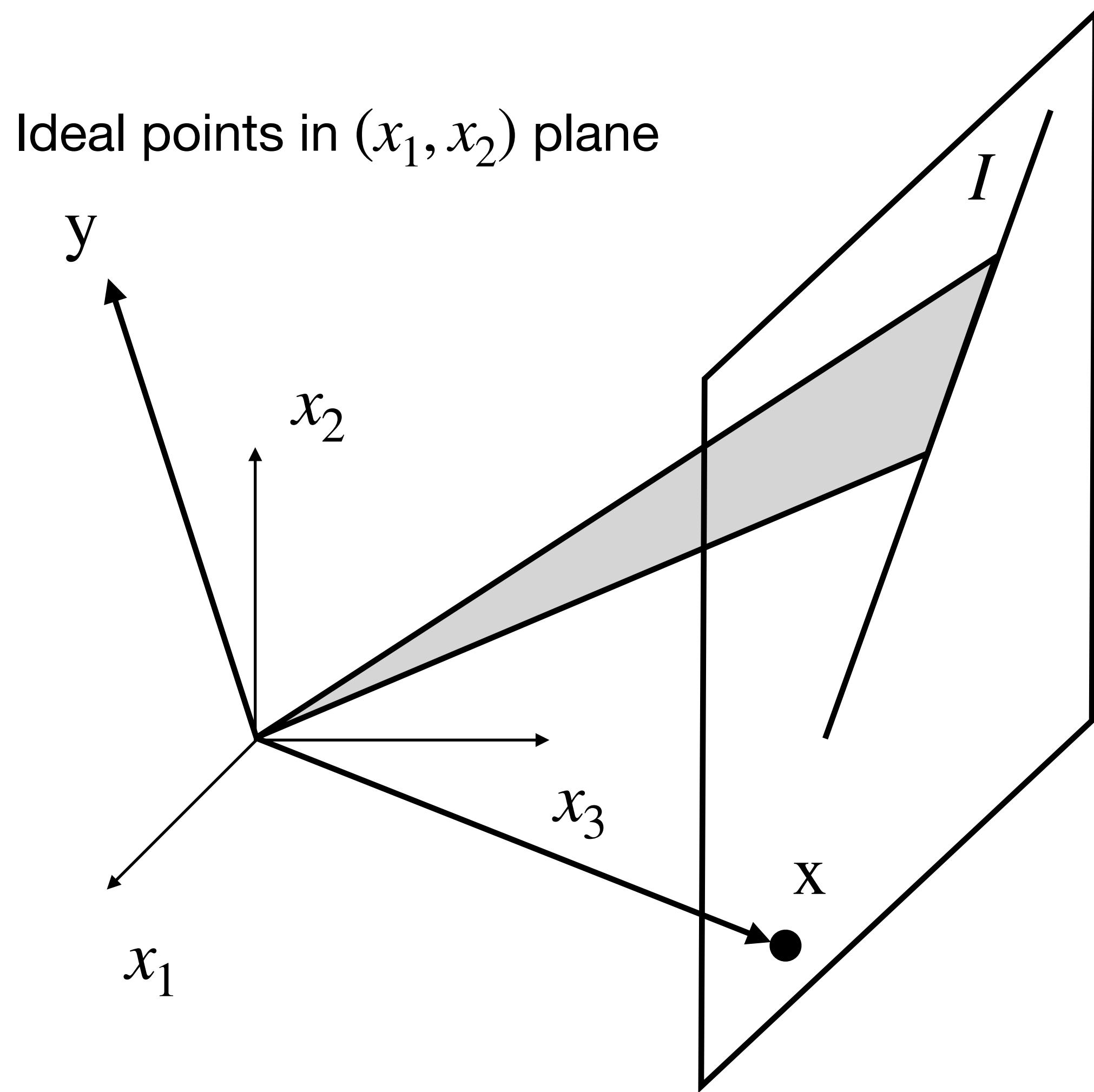


Homogeneous Coordinates for \mathbb{R}^n



- Map $x \in \mathbb{R}^n$ into \mathbb{R}^{n+1}
 $\tilde{x} = (x_1, \dots, x_n, 1)$
- Say $\tilde{x} \sim \tilde{y}$ if $\exists \lambda : x = \lambda y$
- Projective space \mathbb{P}^n is a set of equivalence classes in \mathbb{R}^{n+1} w.r.t. \sim
- Added ideal points $(w_1, \dots, w_n, 0)$

Homogeneous Coordinates for \mathbb{R}^2



- Point on the projective plane
 $\tilde{x} = (x_1, x_2, x_3)$
- Equation below defines proj. line for
 $I = (a, b, c)$
 $I \cdot \tilde{x} = ax_1 + bx_2 + cx_3 = 0$
- Lines I_1 and I_2 intersect at $I_1 \times I_2$
- Points \tilde{x} and \tilde{y} lie on a line $\tilde{x} \times \tilde{y}$

Projective Transformations

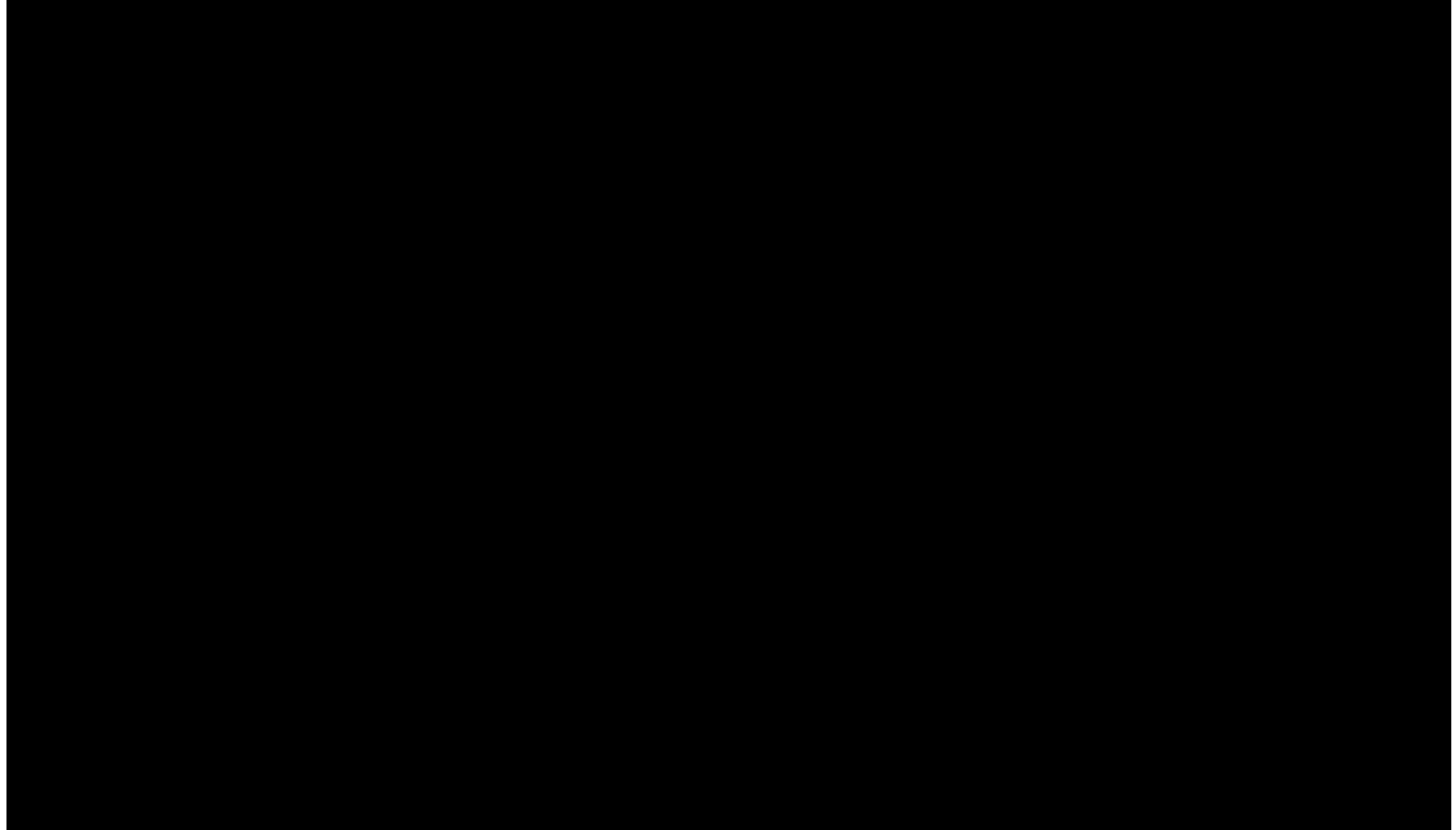
- In homogeneous coordinates, a 3×4 matrix defines a pinhole camera

$$H = [K | 0] C^{W2C} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -Rt \\ 0 & 1 \end{bmatrix}$$

- Matrix K defines ***intrinsic parameters*** of a camera
- Matrix C^{W2C} defines ***extrinsic parameters*** with $R \in SO(3)$ and $t \in \mathbb{R}^3$

Today's Lecture: Structure from Motion (SfM)





Structure From Motion is Still Relevant

- Fundamental 3D vision setup
- Has been under development for over 40 years
- ImageNet moment has not arrived (yet)
- Most advanced reconstruction techniques still rely on SfM for preprocessing



Not in Today's Lecture

Correspondences, Random Sample Consensus, Incremental Initialisations

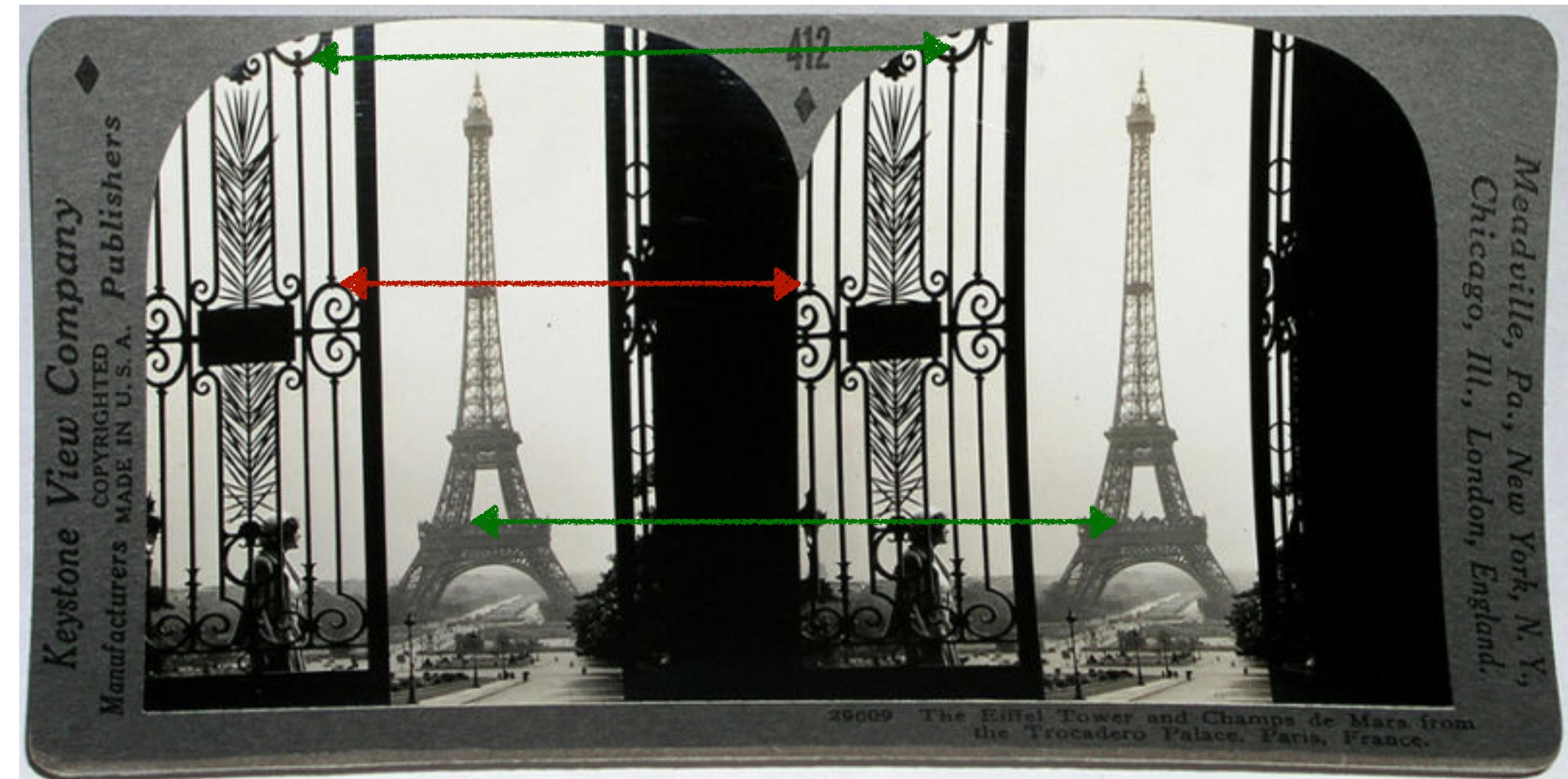
- Correspondences

$$x_l \leftrightarrow x_r$$

- For some x and projections H_l and H_r

$$x_l = H_l x$$

$$x_r = H_r x$$



Sic Parvis Magna

A More Detailed Plan

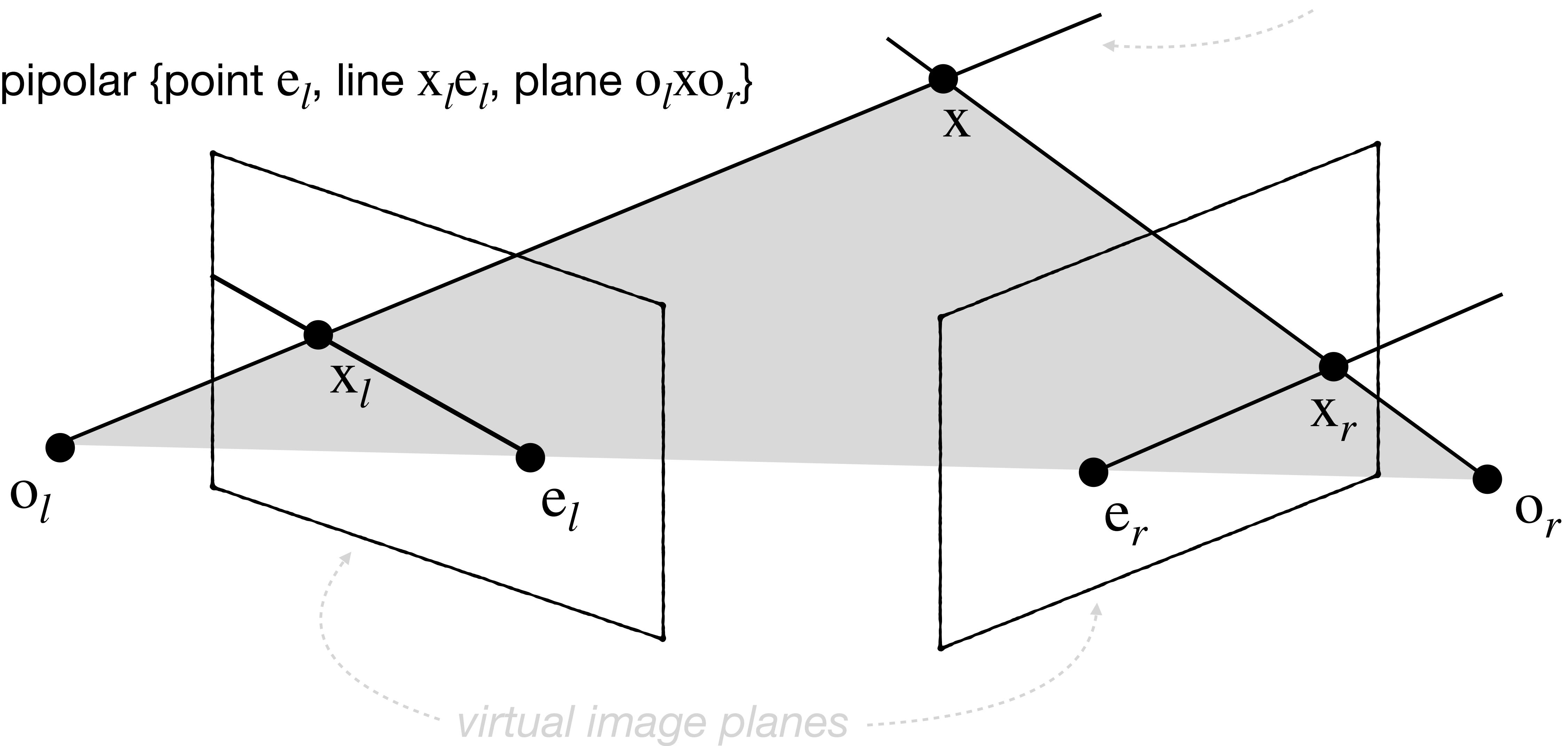
- SfM simultaneously recovers
 - Scene structure as a sparse point cloud
 - Positions of *multiple* cameras
- We will consider key basic components for a stereo-pair
 - How does the image change as we change location? ***Epipolar geometry***
 - Can we infer camera positions from images? ***Fundamental matrix***
 - How one reconstruct 3D structure given camera positions? ***Triangulation***

Epipolar Geometry

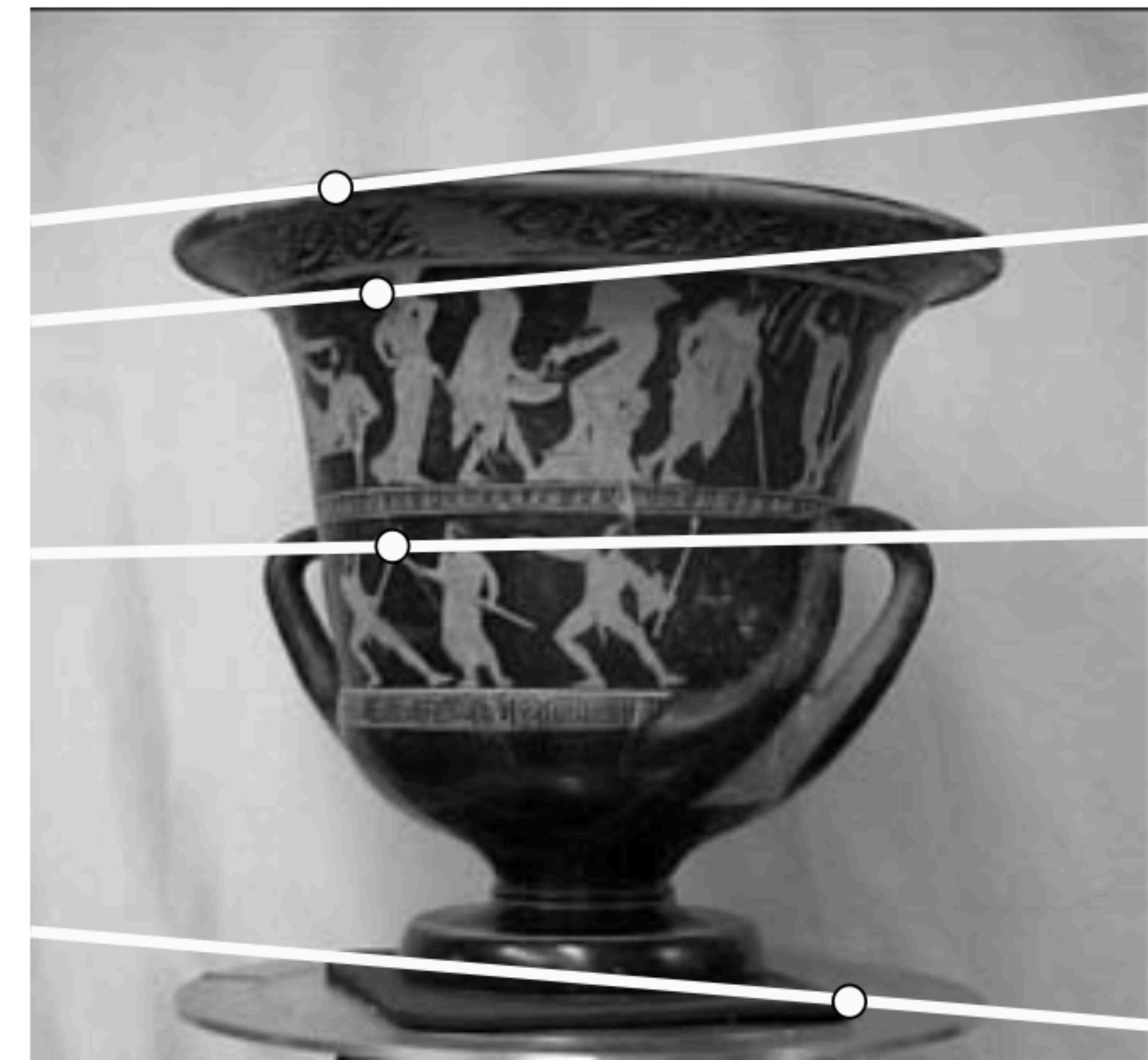
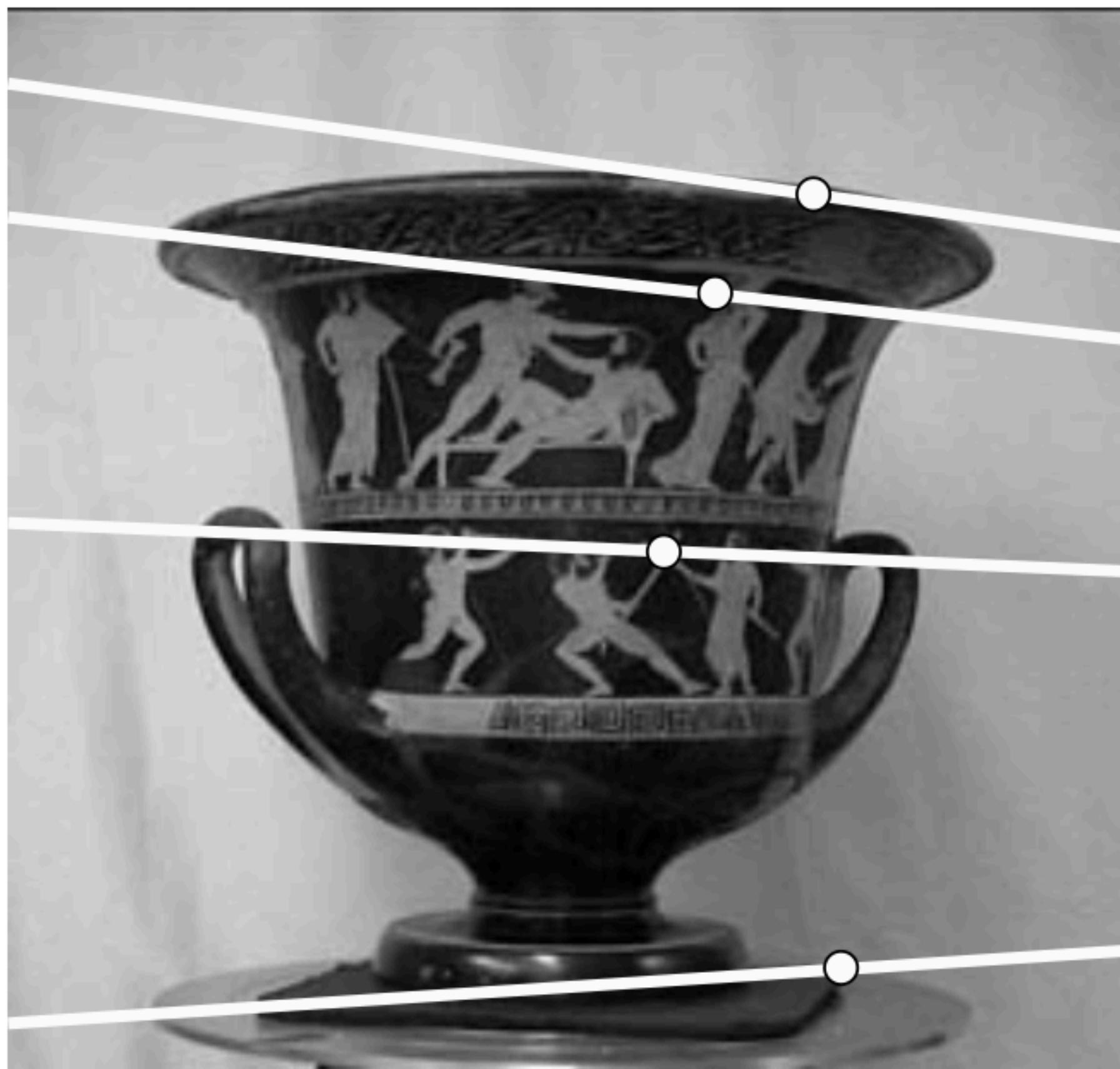
Epipolar Geometry

How does the image changes as we move?

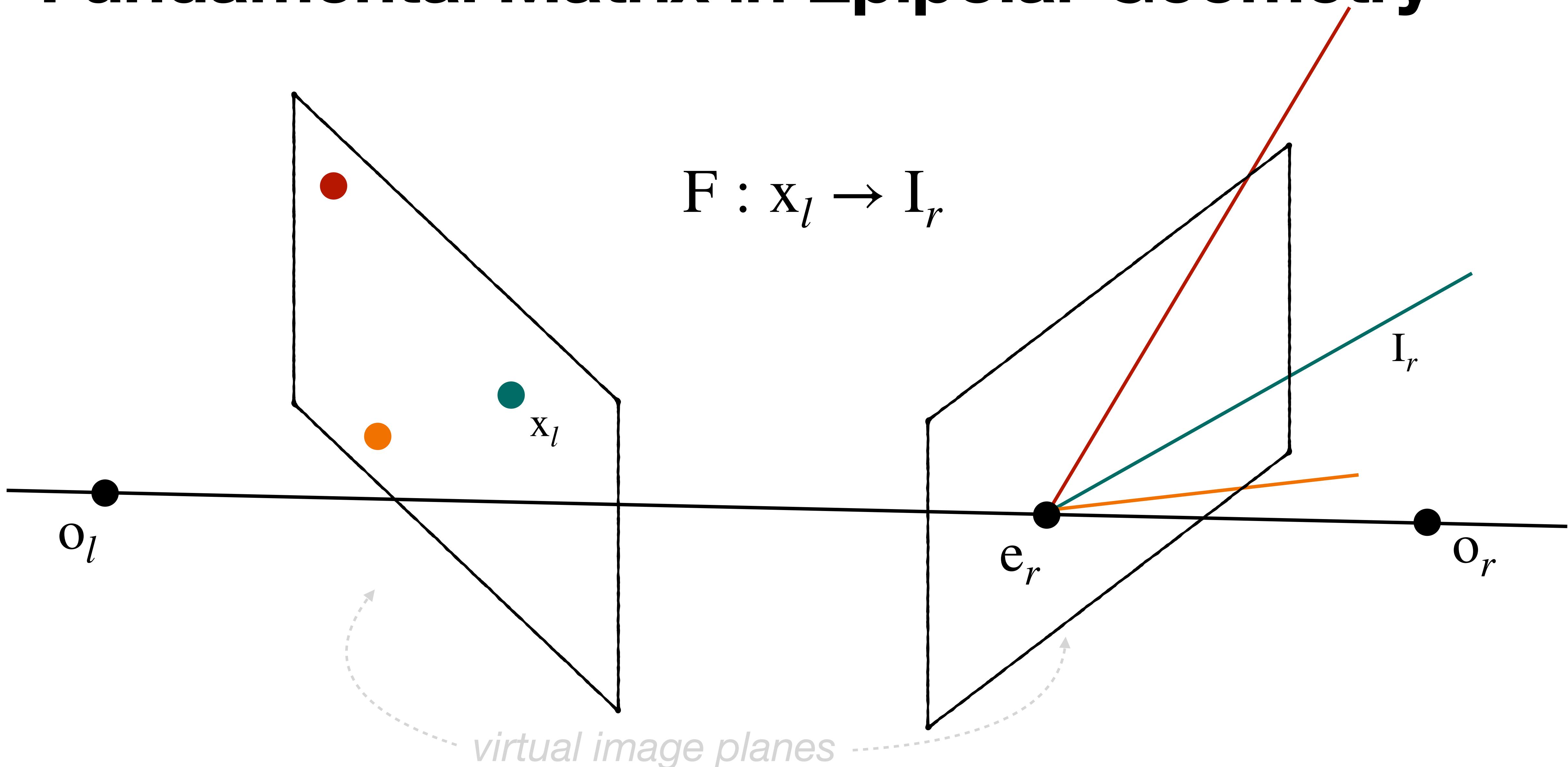
epipolar {point e_l , line $x_l e_l$, plane $o_l x o_r$ }



Epipolar Lines in Real Life



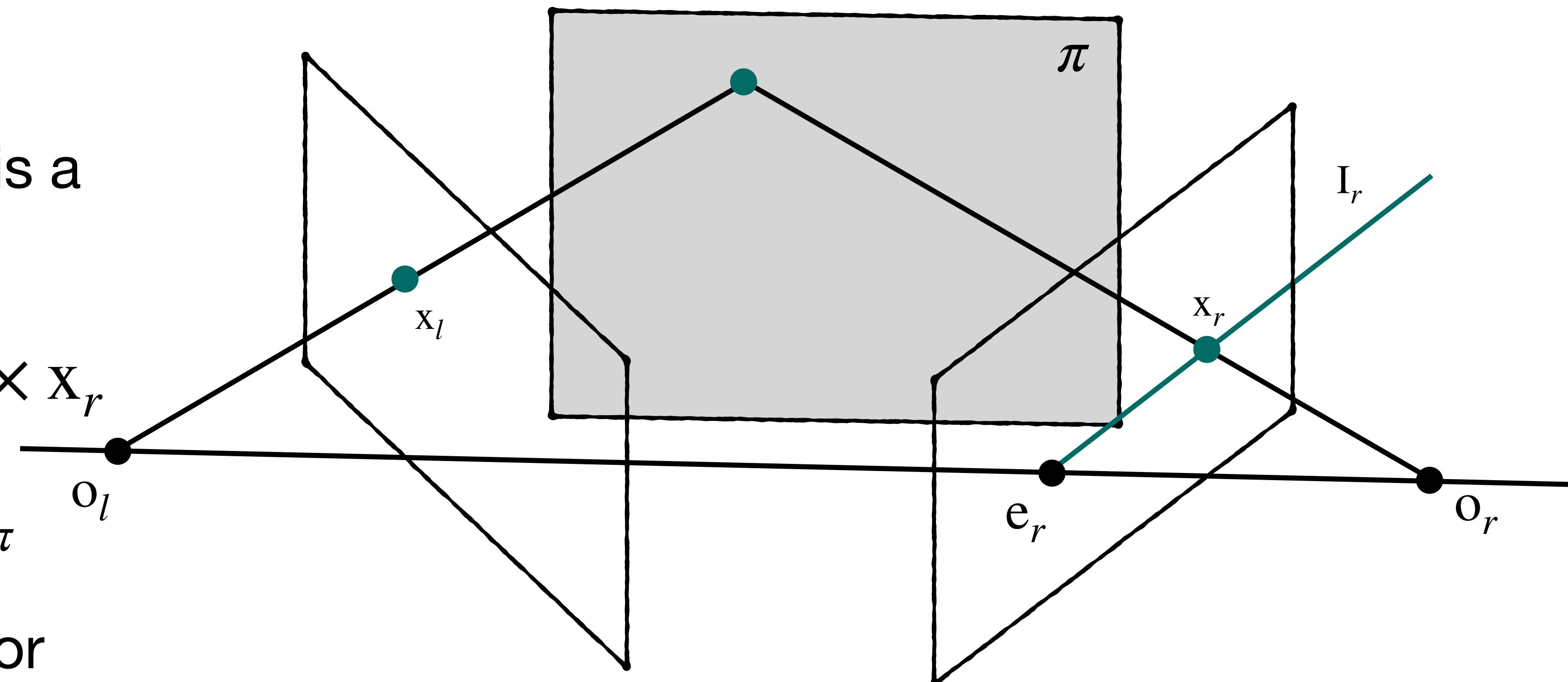
Fundamental Matrix in Epipolar Geometry



Fundamental Matrix

Fundamental Matrix: Geometric Viewpoint

- Step 1: map x_l to x_r
 - Reprojection through π is a homography H_π
- Step 2: map x_r to $I_r = e_r \times x_r$
- Composition: $F = [e_r]_x H_\pi$
 - $[e_r]_x$ denotes a matrix for vector product



Fundamental Matrix: Algebraic Viewpoint

Ray back-projection

- Ray $o_l x_l$ is a solution to $H_l x = x_l$

- Consider pseudo-inverse H_l^+

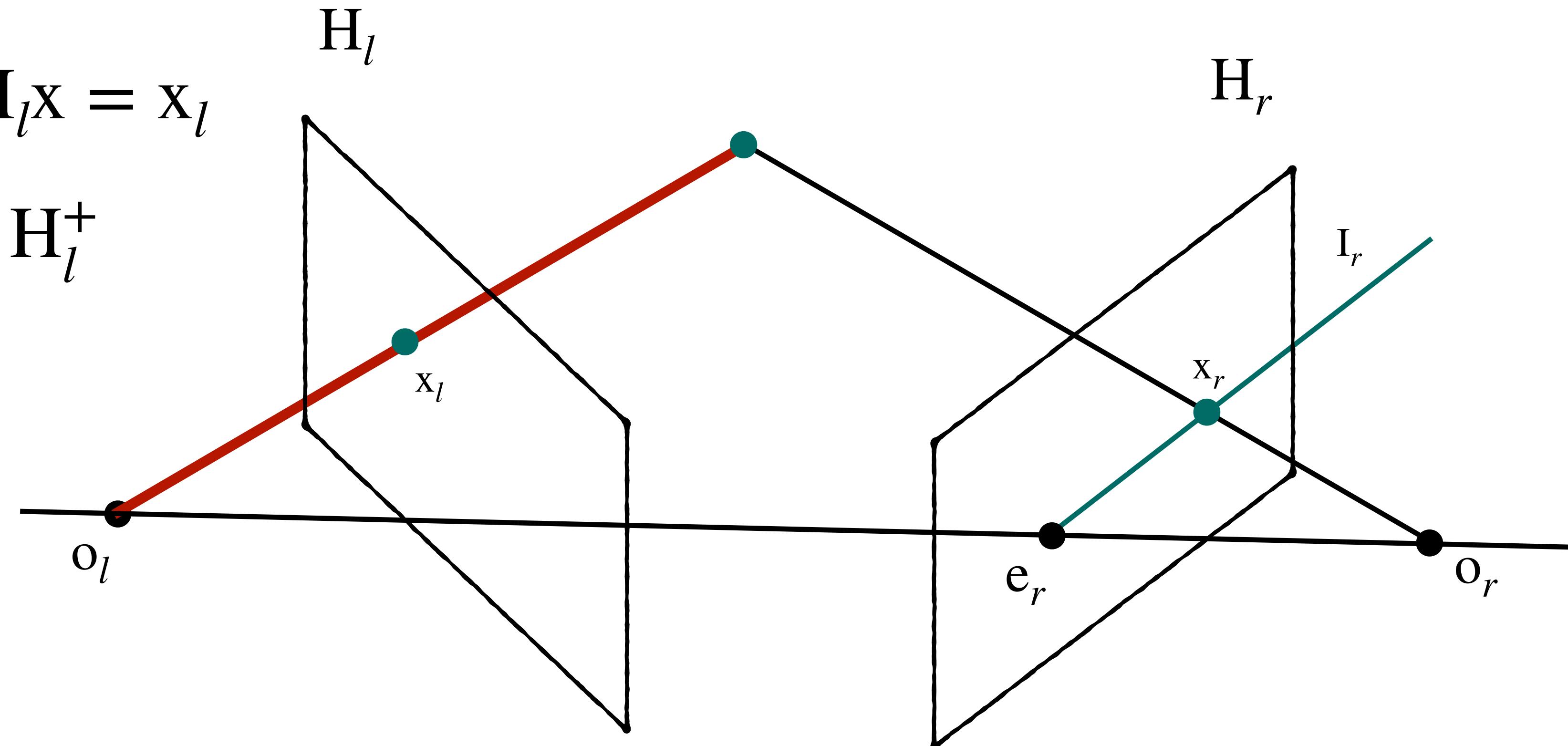
- $H_l H_l^+ = I$

- Explicit parameterisation:

- $x = H_l^+ x_l + \lambda o_l$

- Camera center solves $H_l o_l = 0$

- Pseudo-inverse: $H_l (H_l^+ x_l) = x_l$



Fundamental Matrix: Algebraic Viewpoint

Epipolar Line Construction

- Back-projected ray:

- $\bullet \quad x = H_l^+ x_l + \lambda o_l$

- Consider two points:

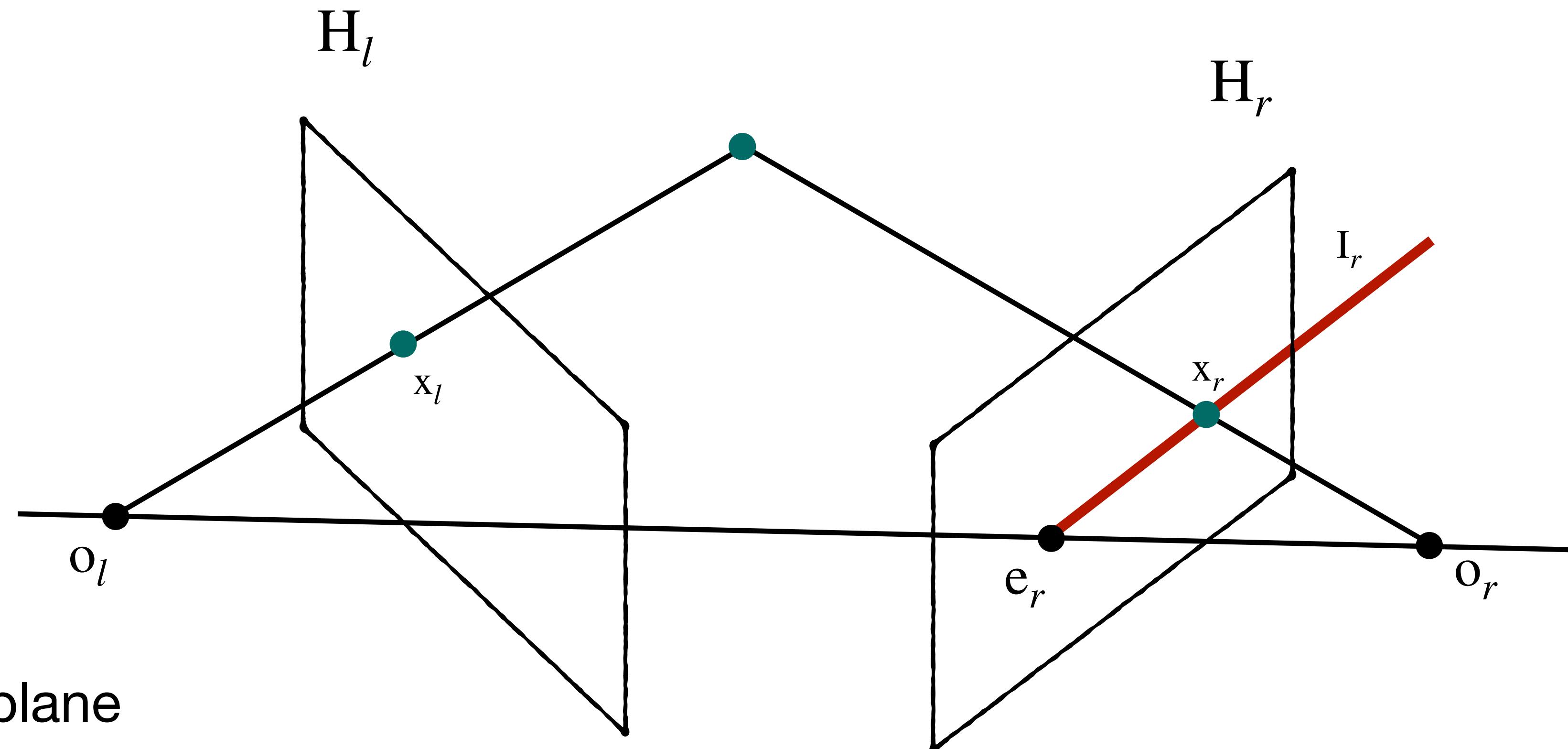
- $\bullet \quad \lambda = 0 : H_l^+ x_l$

- $\bullet \quad \lambda = \infty : o_l$

- Project them onto the right plane

- $\bullet \quad e_r = H_r o_l, \quad H_r H_l^+ x_l$

- $\bullet \quad \text{Construct the line } I_r = [e_r]_x H_r H_l^+ x_l = F x_l$



Fundamental Matrix: Calibrated Cameras

- Consider $H_l = [K_l | 0] \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$ and $H_r = [K_r | 0] \begin{bmatrix} R & -Rt \\ 0 & 1 \end{bmatrix}$
 - Matrix $R \in SO(3)$ and vector $t \in \mathbb{R}^3$ parameterise relative positions of cameras
 - Compute fundamental matrix

- We have $H_l^+ = \begin{bmatrix} K_l^{-1} \\ 0^T \end{bmatrix}$ and $o_l = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

$$F = [e_r]_x H_r H_l^+ = [K_r t]_x K_r R K_l^{-1} = \dots = K_r^{-1} [t]_x R K_l^{-1}$$

Fundamental Matrix: Relative Positions

- Consider $H_l = [K_l | 0] \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$ and $H_r = [K_r | 0] \begin{bmatrix} R & -Rt \\ 0 & 1 \end{bmatrix}$
- Fundamental matrix encodes R and t :

$$F = K_r^{-1} [t]_x R K_l^{-1}$$

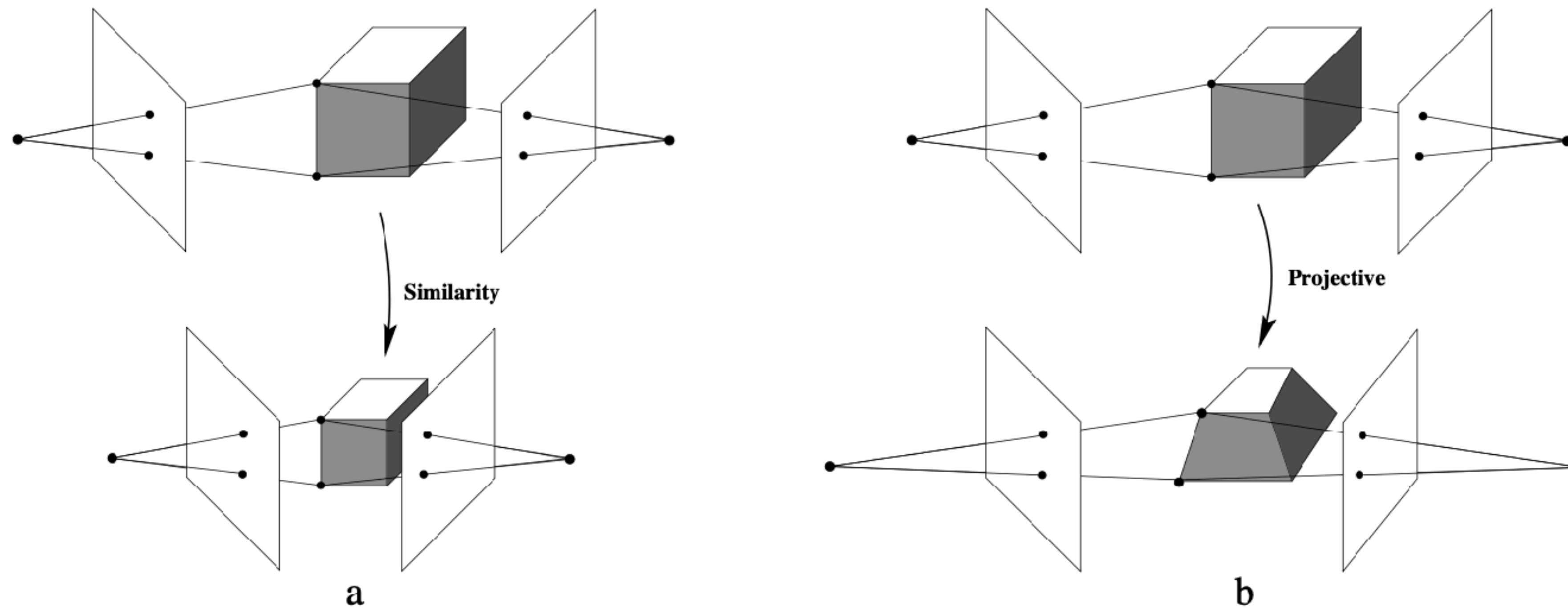
- Usually, we know K_r and K_l in advance
- Matrix $E = [t]_x R$ is known as **essential matrix**
- One can recover t and R from E (*up to scale*)

Estimation of Fundamental Matrix

- Note that $\mathbf{x}_r^T \mathbf{F} \mathbf{x}_l = 0$ for any corresponding pair $\mathbf{x}_l \leftrightarrow \mathbf{x}_r$
- Equation $\mathbf{x}_r^T \mathbf{F} \mathbf{x}_l = 0$ is a linear equation in \mathbf{F}
- To recover \mathbf{F} (*up to scale factor*) we need 8 point correspondences
- ***Eight-point algorithm:***
 - finds \mathbf{F} given point correspondences for a stereo-pair
- *How many points do we actually need? Hint: $\mathbf{F} = \mathbf{K}_r^{-1}[\mathbf{t}]_\times \mathbf{R} \mathbf{K}_l^{-1}$*

Structure Recovery

Reconstruction Ambiguity



Structure Recovery

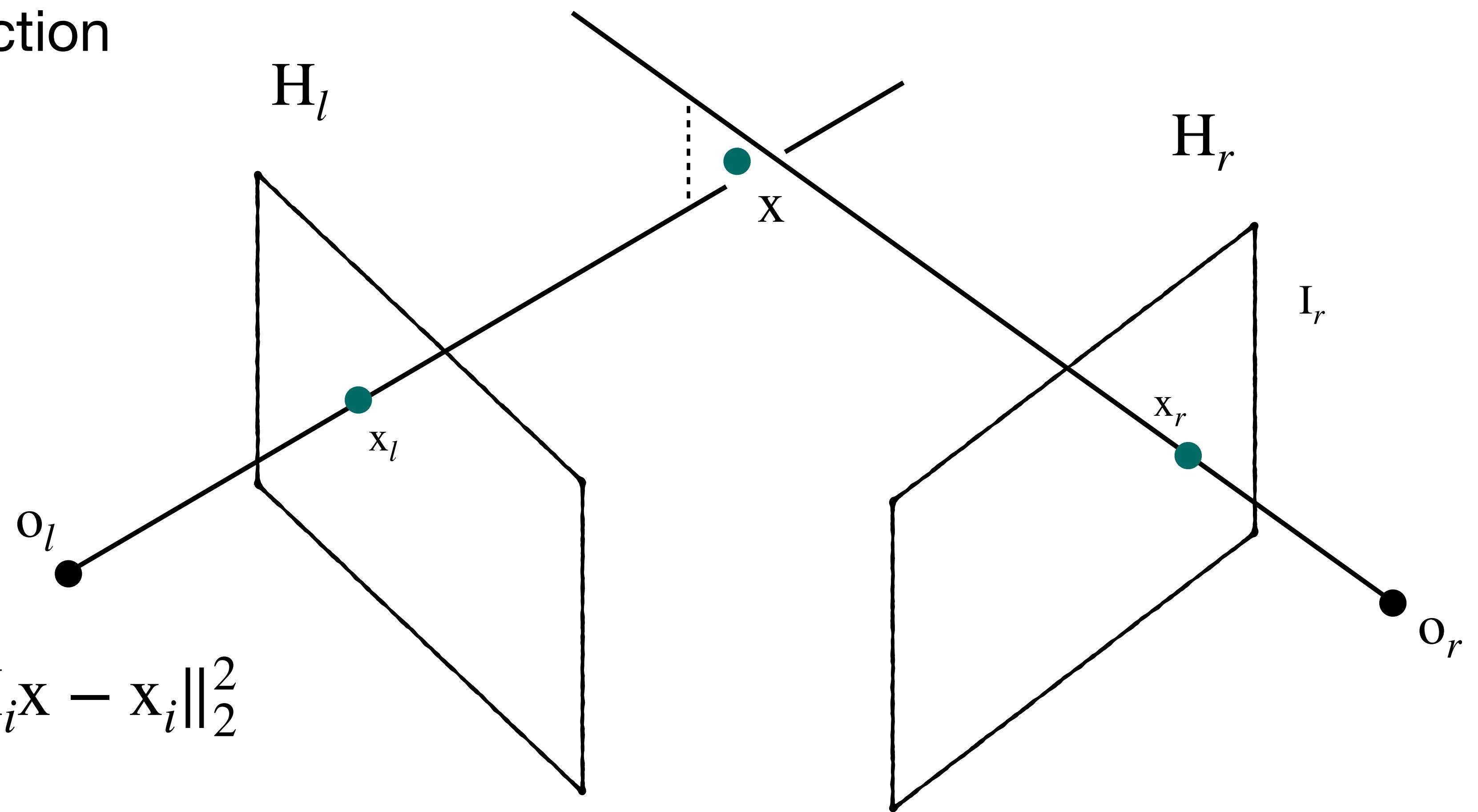
- Naive: find the intersection

- $H_l^+ x_l + \lambda o_l$

- $H_r^+ x_r + \mu o_l$

- More practical:

$$x = \arg \min \sum_{i \in \{l, r\}} \|H_i x - x_i\|_2^2$$

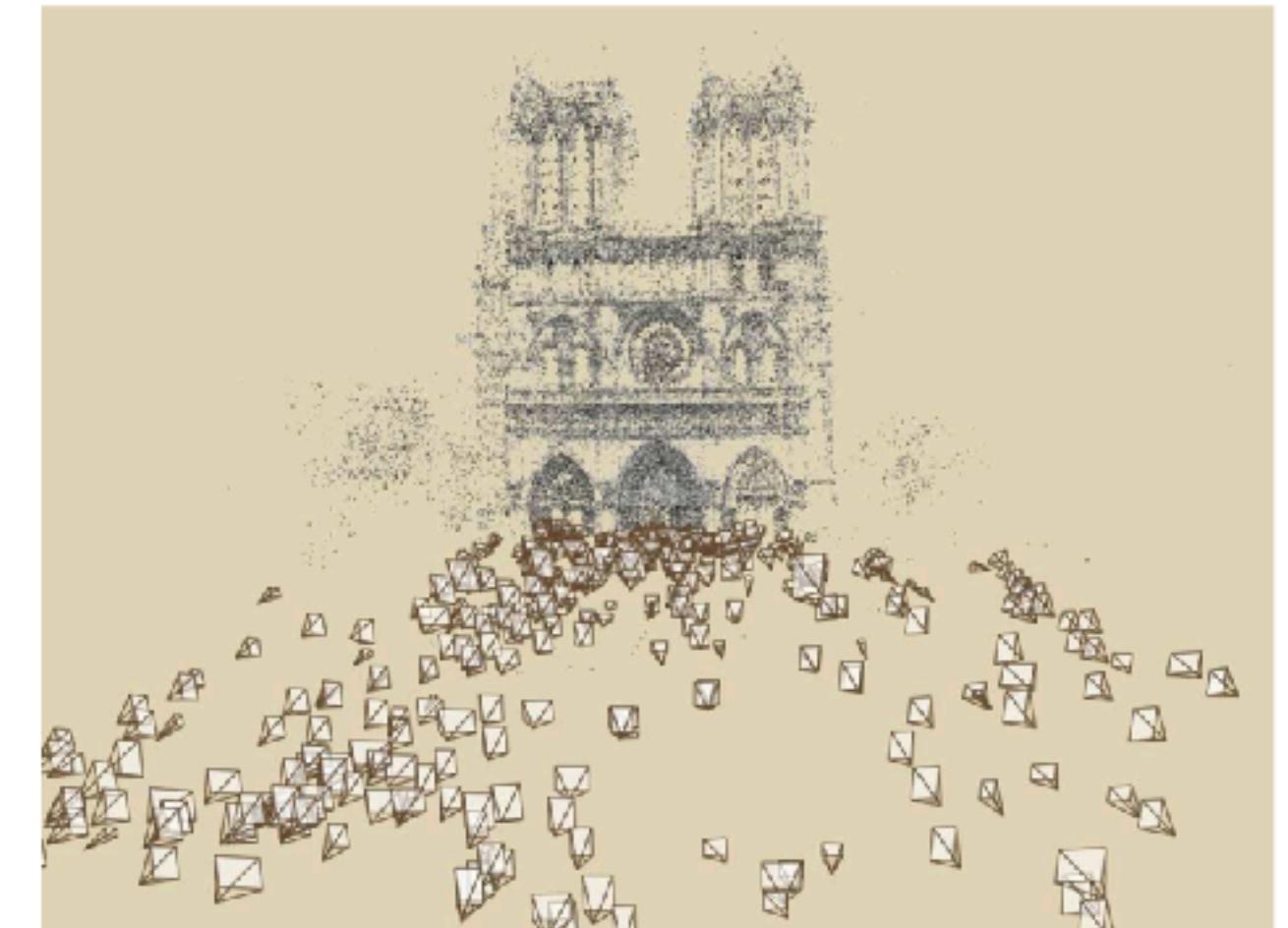
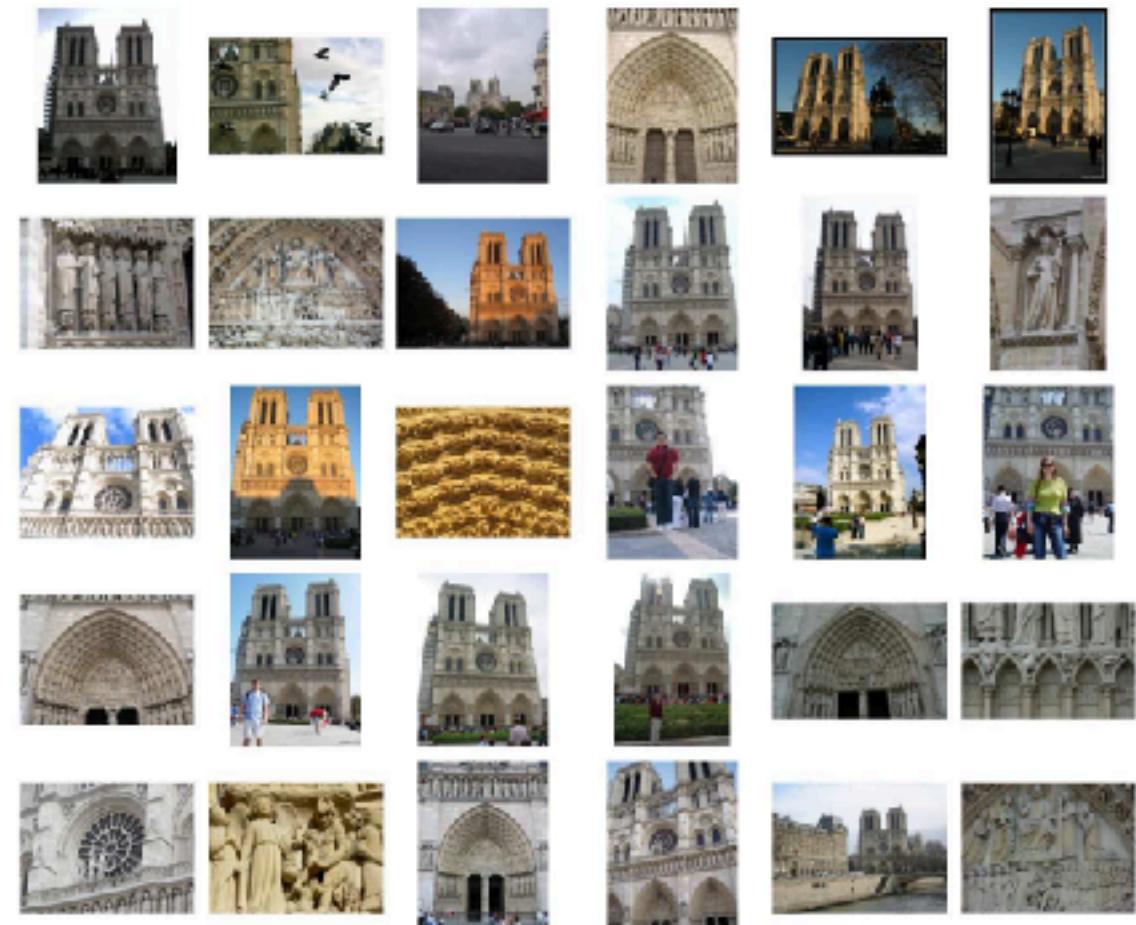


Bundle Adjustment

Bundle Adjustment

Goal

- Bundle Adjustment is a step in Structure From Motion Pipeline
- Input: images and point correspondences
- Aims to
 1. estimate camera parameters
 2. recover structure
- Does (1) and (2) simultaneously



Bundle Adjustment

Problem Formulation (with a slight shift in notation)

- Input: $\mathcal{X}_s = \{\mathbf{x}_{ip}^s\}$
 - i is the camera index
 - p is the point cloud index
- Parameters:
 - Cameras $\Pi = \{\pi_i\}$, $\pi_i = (\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i)$
 - Point cloud $\mathcal{X}_p = \{\mathbf{x}_p^w\}$

Bundle Adjustment

Problem Formulation (with a slight shift in notation)

- Input: $\mathcal{X}_s = \{\mathbf{x}_{ip}^s\}$
- Parameters: cameras $\Pi = \{\pi_i\}$ and point cloud $\mathcal{X}_p = \{\mathbf{x}_p^w\}$
- Objective: reprojection error

$$\Pi^*, \mathcal{X}_w^* = \arg \min_{\Pi, \mathcal{X}_w} \sum_i \sum_p w_{ip} \|\pi_i(\mathbf{x}_p^w) - \mathbf{x}_{ip}^s\|_2^2$$

where π_i denotes the projection of i -th screen and w_{ip} indicates visibility

Bundle Adjustment

In Practice

- Objective: reprojection error

$$\Pi^*, \mathcal{X}_w^* = \arg \min_{\Pi, \mathcal{X}_w} \sum_i \sum_p w_{ip} \|\pi_i(\mathbf{x}_p^w) - \mathbf{x}_{ip}^s\|_2^2$$

- Does not converge with an arbitrary initialisation
- Modern solution construct initialisation incrementally
 - Consider pair of images
 - Recover relative positions, approximate structure with triangulation etc.

Key takeaways

- *Fundamental matrix*
 - Helps inferring relative camera positions
 - Can be estimated based on point correspondences
- *Triangulation* recovers sparse structure given relative camera positions & correspondences
- Next lecture: depth estimation

