3D Vision 2

Week 8

Two-view Geometry: Homography Estimation

Two-view Geometry: Epipolar Geometry

Announcements

- Assignment 2 due 11:59pm **today**
  - This includes a one week extension that has already been applied
  - **Zero** marks if either report or code submitted late (unless extension)
    - Submit early; you can always resubmit an updated version later
    - Depending on your internet connection and load on the TurnItIn servers, uploading can sometimes be slow, so please factor this into your submission schedule
  - Submit your report (PDF) and code (ZIP file) **separately under the correct tab** in the submission box
  - Follow the instructions under Submission Requirements

Weekly Study Plan: Overview

Wk	Starting	Lecture	Lab	Assessment
1	19 Feb	Introduction	✗	
2	26 Feb	Low-level Vision 1	1	
3	4 Mar	Low-level Vision 2	1	
		Mid-level Vision 1		
4	11 Mar	Mid-level Vision 2	1	CLab1 report due Friday
		High-level Vision 1		
5	18 Mar	High-level Vision 2	2	
6	25 Mar	High-level Vision 3[1]	2	
	1 Apr	Teaching break	✗	
	8 Apr	Teaching break	✗	
7	15 Apr	3D Vision 1	2	CLab2 report due Friday
8	22 Apr	3D Vision 2	3	
9	29 Apr	3D Vision 3	3	
10	6 May	3D Vision 4	3	
		Mid-level Vision 3		
11	13 May	High-level Vision 4	✗	CLab3 report due Friday
12	20 May	Course Review	✗	

Weekly Study Plan: Part B

Wk	Starting	Lecture	By
7	15 Apr	3D vision: introduction, camera model, single-view geometry	Dylan
8	22 Apr	3D vision: camera calibration, two-view geometry (homography)	Dylan
9	29 Apr	3D vision: two-view geometry (epipolar geometry, triangulation, stereo)	Dylan
10	6 May	3D vision: multiple-view geometry	Weijian
		Mid-level vision: optical flow, shape-from-X	Dylan
11	13 May	High-level vision: self-supervised learning, detection, segmentation	Dylan
12	20 May	Course review	Dylan

Outline

1. Two-view Geometry: Homographies
2. Two-view Geometry: Homography Estimation
3. Two-view Geometry: Epipolar Geometry

Summary: Camera Calibration

- **Estimate:** $\mathbf{P}, \mathbf{K}, \mathbf{R}, \boldsymbol{C}$
- **Given:** 2D–3D point correspondences $\{\boldsymbol{x}_i, \boldsymbol{X}_i\}$
- **Use:** Direct Linear Transformation (DLT) algorithm to get $\mathbf{P}$
  - Assemble matrix $\mathbf{A}$, where

    for each (x_i, X_i) $$\mathbf{A}_i = \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \end{bmatrix}$$
  - Take SVD of $\mathbf{A}$, where $\mathbf{A} = \mathbf{U\Sigma V}^\top$; $\boldsymbol{p}$ is the last column of $\mathbf{V}$
  - Reshape vector $\boldsymbol{p}$ into matrix $\mathbf{P}$ $12 * 1 \rightarrow 3 * 4$
- **Decompose: $\mathbf{P} \rightarrow \boldsymbol{C}$**
  - $\mathbf{PC} = \mathbf{0} \Rightarrow \boldsymbol{C}$ is the right null space vector of $\mathbf{P}$
  - Take SVD, where $\mathbf{P} = \mathbf{U\Sigma V}^\top$; $\boldsymbol{C}$ is the last column of $\mathbf{V}$
- **Decompose: $\mathbf{P} \rightarrow \mathbf{K}, \mathbf{R}$**
  - $\mathbf{K}$ is upper triangular, $\mathbf{R}$ is orthogonal
  - Perform an RQ-decomposition of $\mathbf{KR}$ where $\mathbf{P} = [\mathbf{KR} | -\mathbf{KR}\boldsymbol{C}]$

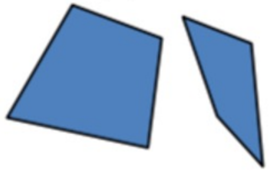
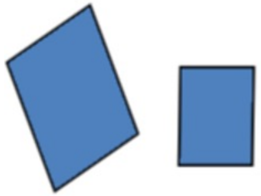
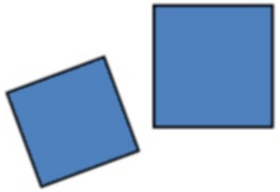
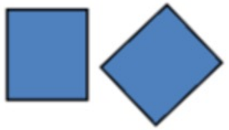
- $\overset{3*4}{\mathbf{P}} = \overset{3*3}{\mathbf{K}}[\overset{3*3}{\mathbf{R}} \mid -\mathbf{R}\overset{3*1}{\mathbf{C}}] = [\overset{3*3}{\mathbf{M}} \mid -\mathbf{M}\overset{3*1}{\mathbf{C}}]$

1. RQ decomposition of $\mathbf{M}$:
   - $(\overset{K}{\mathbf{R}_\Delta}, \overset{R}{\mathbf{Q}}) = \mathrm{RQ}(\mathbf{M})$
   - $\mathbf{K} = \mathbf{R}_\Delta$: upper triangular matrix (Δ is just to distinguish it from rotation)
   - $\mathbf{R} = \mathbf{Q}$: orthonormal matrix

Homographies

Two-view Geometry

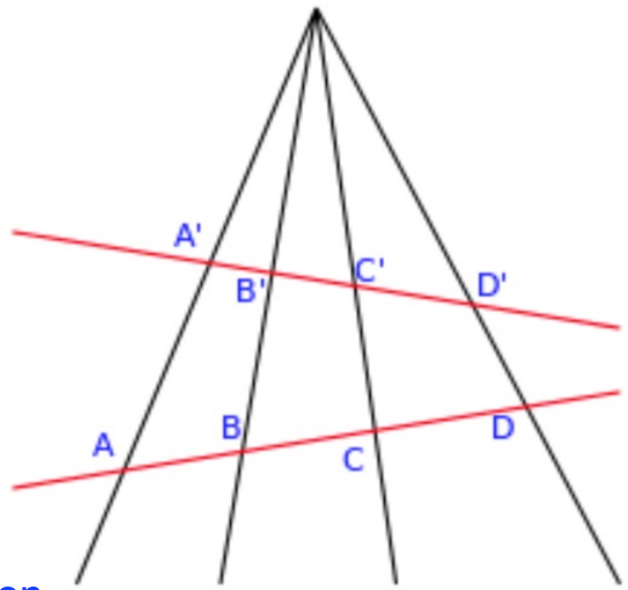
A Hierarchy of 2D Transformations

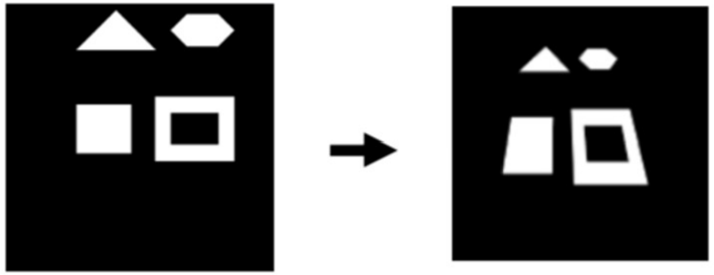
		Transformed Square	Invariants
Homography Projective 8 DoF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		• Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6 DoF	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Parallelism, ratio of areas, ratio of lengths on parallel lines (e.g., midpoints), linear combinations of vectors (centroids), line at infinity I_∞
Similarity 4 DoF	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Ratios of lengths, angles, circular points I,J
Euclidean 3 DoF	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Lengths, areas

Projective Transformation (Homography)

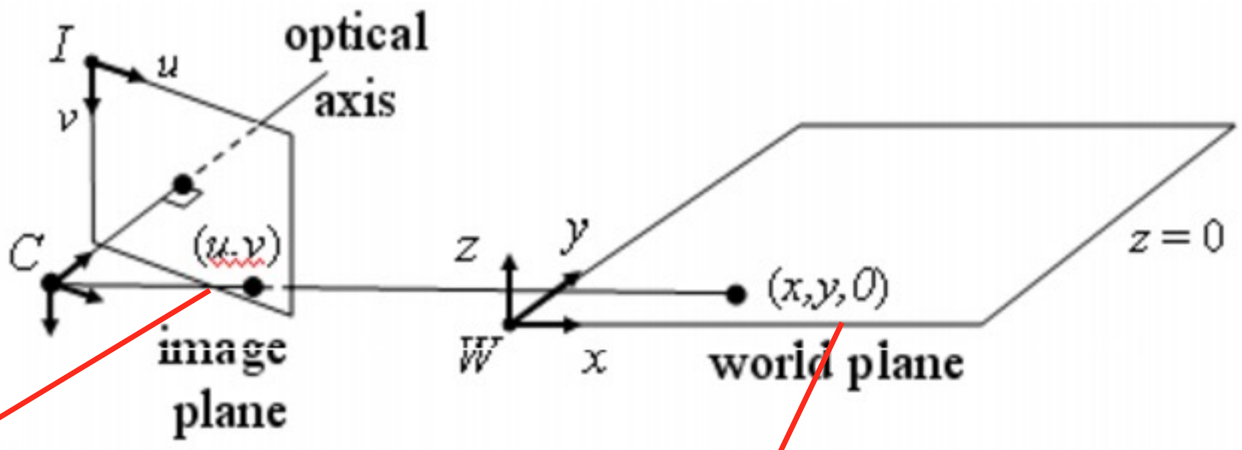
$$\begin{array}{c} \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \end{array} = \begin{array}{c} 3 * 3 \\ \begin{bmatrix} \mathbf{A} & \boldsymbol{t} \\ \boldsymbol{v}^\top & b \end{bmatrix} \end{array} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 8 DoF
- Preserves:
  - Cross ratio of 4 collinear points
    $$(A, B; C, D) = \frac{AC \cdot BD}{BC \cdot AD}$$
  - Collinearity points lying on the same straight line after trasnformation
- Example:
  - Points A, B, C, D and A', B', C', D' are related by a projective transformation, so their cross ratios are equal $(A, B; C, D) = (A', B'; C', D')$



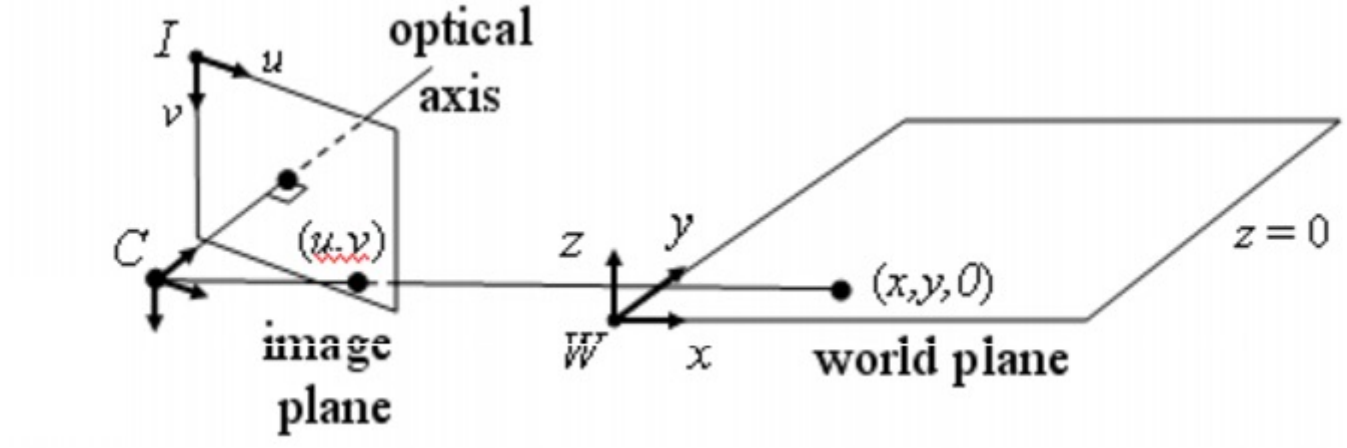


Homography Mapping from World Plane



$$\overset{\text{x\_img}}{\begin{bmatrix}u\\v\\1\end{bmatrix}} \propto \underbrace{\overset{3*3}{\begin{bmatrix}\alpha&\gamma&u_0\\0&\beta&v_0\\0&0&1\end{bmatrix}}}_{\substack{\mathbf{K}_{\{3\times3\}}\\ \text{intrinsic}}} \underbrace{\overset{3*4}{[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3 \quad \boldsymbol{t}]}}_{\substack{[\mathbf{R}_{\{3\times3\}} \quad \boldsymbol{t}_{\{3\times1\}}]\\ \text{extrinsic}}} \overset{\text{x\_world}}{\begin{bmatrix}x\\y\\0\\1\end{bmatrix}}$$

Homography Mapping from World Plane



$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}_{\{3\times3\}}} \underbrace{[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boxed{\boldsymbol{r}_3} \quad \boldsymbol{t}]}_{[\mathbf{R}_{\{3\times3\}} \quad \boldsymbol{t}_{\{3\times1\}}]} \begin{bmatrix} x \\ y \\ \boxed{0} \\ 1 \end{bmatrix} = \boxed{\underbrace{\mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}]}_{\mathbf{H}_{\{3\times3\}}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

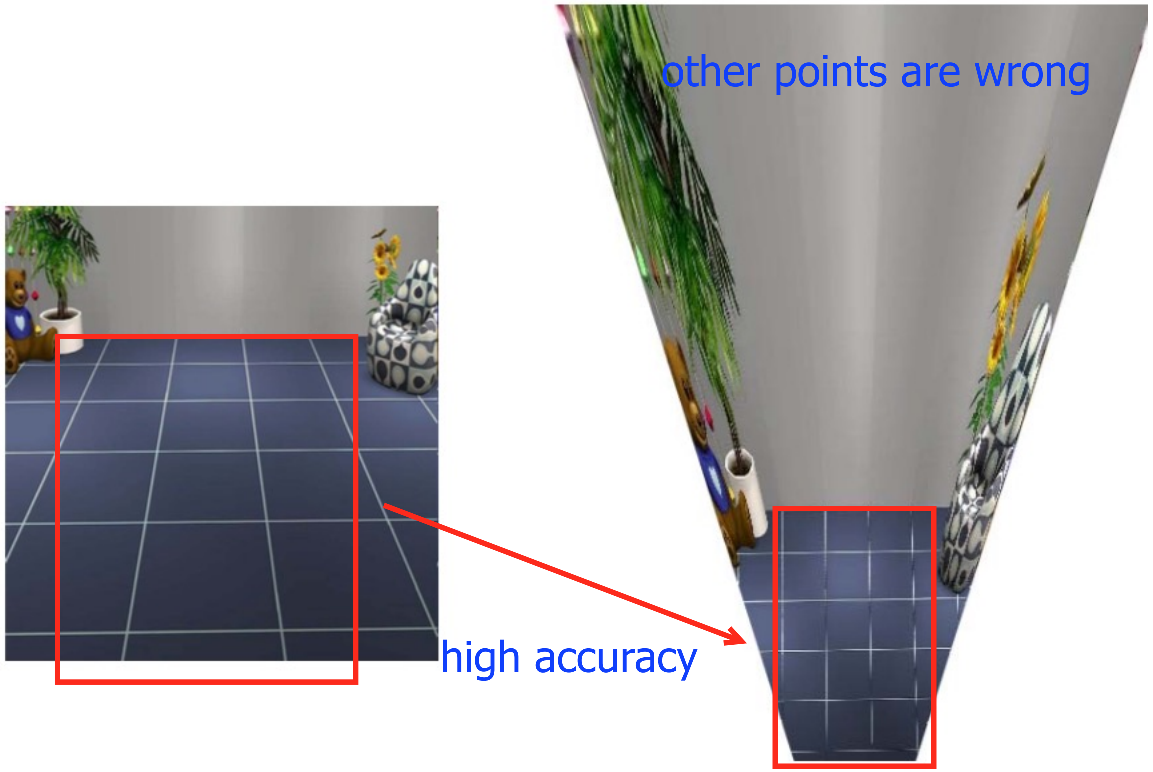
An Homography is a Planar Projectivity

- Given the homography matrix, one can transform an unfavourable view of a plane into a more desirable one
- While the chosen plane will be transformed with high accuracy, every other point in the image will suffer a wrong transformation

Left: Source, Right: Warped with homography

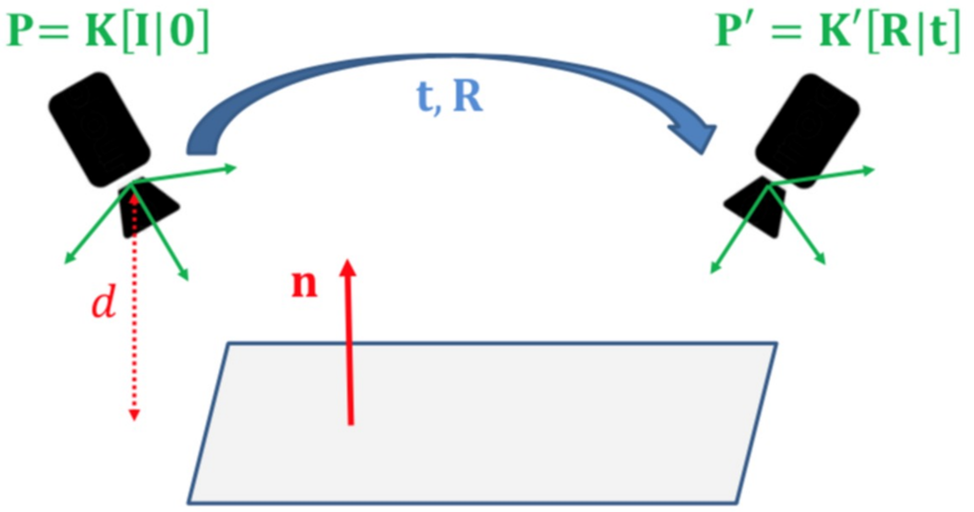
Left: Real view, Right: Artificially warped view

An Homography is a Planar Projectivity



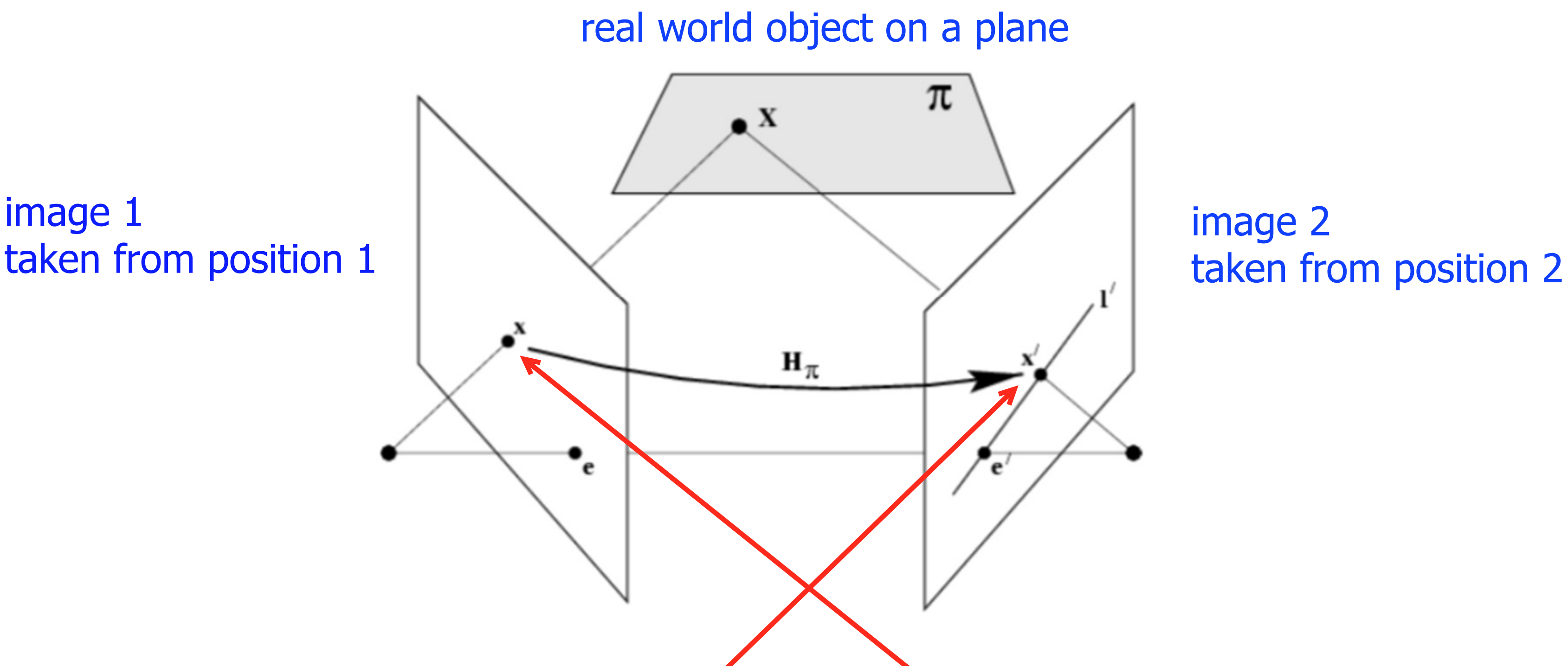
Planar Homography

- Transforms points to points
- Depends on:
  - Camera intrinsic parameters K
  - Relative motion parameters R, t
  - 3D plane parameters (normal vector and depth) d, n



$$\boxed{\mathbf{H} = \mathbf{K}'\left(\mathbf{R} - \frac{\boldsymbol{t}\boldsymbol{n}^{\top}}{d}\right)\mathbf{K}^{-1}}$$

Geometric Derivation



$$x' = \mathbf{H}_\pi x$$

Transformation of 2D Points and Lines

- Transforms points into points

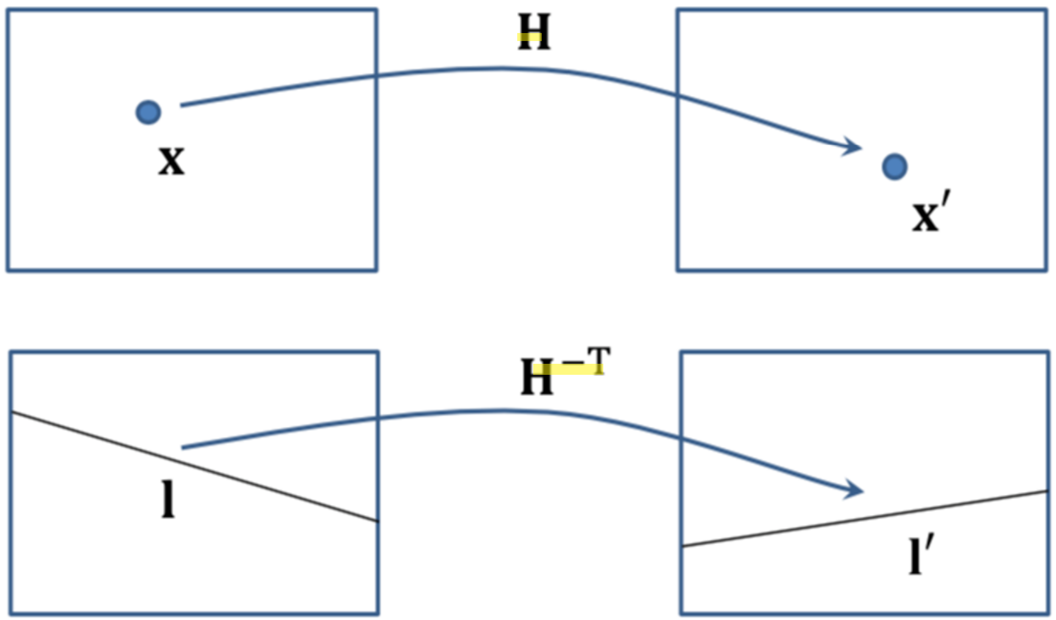
$$\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$$

- Also maps lines to lines

$$l' = \mathbf{H}^{-\top} l$$

- Proof:

$$\begin{aligned} l^\top \boldsymbol{x} &= l^\top (\mathbf{H}^{-1}\mathbf{H})\boldsymbol{x} \\ &= \boxed{(\mathbf{H}^{-\top} l)^\top} \mathbf{H}\boldsymbol{x} \\ &= \boxed{l'^\top} \boldsymbol{x}' \\ &= 0 \end{aligned}$$



Homography Estimation

Two-view Geometry

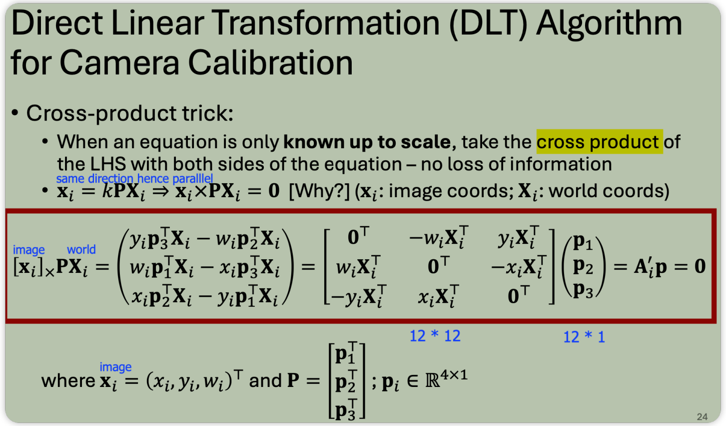
Direct Linear Transformation (DLT) Algorithm for Homography Estimation

$\boxed{\mathbf{x}_i' = k\mathbf{H}\mathbf{x}_i \Rightarrow \mathbf{x}_i' \times \mathbf{H}\mathbf{x}_i = \mathbf{0}}$ ($\mathbf{x}_i$: image 1 coord; $\mathbf{x}_i{}'$: image 2 coord)

$$\boxed{[\mathbf{x}_i']_\times \mathbf{H}\mathbf{x}_i = \begin{pmatrix} y_i'\mathbf{h}_3^\top \mathbf{x}_i - w_i'\mathbf{h}_2^\top \mathbf{x}_i \\ w_i'\mathbf{h}_1^\top \mathbf{X}_i - x_i'\mathbf{h}_3^\top \mathbf{x}_i \\ x_i'\mathbf{h}_2^\top \mathbf{X}_i - y_i'\mathbf{h}_1^\top x_i \end{pmatrix} = \begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \\ -y_i'\mathbf{x}_i^\top & x_i'\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{A}_i'\mathbf{h} = \mathbf{0}}$$

3*3

where $\mathbf{x}_i = (x_i, y_i, w_i)^\top$ and $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{bmatrix}$; $\mathbf{h}_i \in \mathbb{R}^{3\times 1}$


Direct Linear Transformation (DLT) Algorithm for Camera Calibration

- Cross-product trick:
  - When an equation is only **known up to scale**, take the cross product of the LHS with both sides of the equation – no loss of information
  - same direction hence paralllel
    $\mathbf{x}_i = k\mathbf{P}\mathbf{X}_i \Rightarrow \mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \mathbf{0}$ [Why?] ($\mathbf{x}_i$: image coords; $\mathbf{X}_i$: world coords)

image world

$$\boxed{[\mathbf{x}_i]_\times \mathbf{P}\mathbf{X}_i = \begin{pmatrix} y_i\mathbf{p}_3^\top \mathbf{X}_i - w_i\mathbf{p}_2^\top \mathbf{X}_i \\ w_i\mathbf{p}_1^\top \mathbf{X}_i - x_i\mathbf{p}_3^\top \mathbf{X}_i \\ x_i\mathbf{p}_2^\top \mathbf{X}_i - y_i\mathbf{p}_1^\top \mathbf{X}_i \end{pmatrix} = \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i\mathbf{X}_i^\top & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{A}_i'\mathbf{p} = \mathbf{0}}$$

12 * 12 12 * 1

image

where $\mathbf{x}_i = (x_i, y_i, w_i)^\top$ and $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix}$; $\mathbf{p}_i \in \mathbb{R}^{4\times 1}$

24

Direct Linear Transformation (DLT) Algorithm for Homography Estimation

- Equations are linear in $\mathbf{h}$
- Only 2 out of 3 equations are linearly independent, so pick two

$$\underbrace{\begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \end{bmatrix}}_{\mathbb{R}^{2\times 9}} \underbrace{\begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix}}_{\mathbb{R}^9} = \mathbf{A}_i\mathbf{h} = \mathbf{0}$$

- Holds for any homogeneous representation, e.g., $(x_i', y_i', 1)$
- Homography matrix has 8 DoF:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}_{\{3\times 3\}}} \underbrace{[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boxed{\boldsymbol{r}_3} \quad \boldsymbol{t}]}_{[\mathbf{R}_{\{3\times 3\}} \quad \boldsymbol{t}_{\{3\times 1\}}]} \begin{bmatrix} x \\ y \\ \boxed{0} \\ 1 \end{bmatrix} = \boxed{\underbrace{\mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}]}_{\mathbf{H}_{\{3\times 3\}}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  - 9 parameters defined up to scale
  - Linear solution requires at least 4 points (two DoF per point)

Direct Linear Transformation (DLT) Algorithm for Homography Estimation

- Using 4 points to solve for $\mathbf{H}$:

$$\mathbf{A}\mathbf{h} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \end{bmatrix} \mathbf{h} = \mathbf{0}$$

(2 * 4) * 9

  - $\mathbf{A} \in \mathbb{R}^{8\times 9}$ but $\mathrm{rank}(\mathbf{A}) = 8$
- How to solve?
  - The trivial solution $\mathbf{h} = \mathbf{0}$ is not interesting
  - Compute the 1D null-space (e.g., via SVD)
  - Fix norm of h afterwards (e.g., set $||\mathbf{h}|| = 1$)

Direct Linear Transformation (DLT) Algorithm for Homography Estimation

n ≥ 4

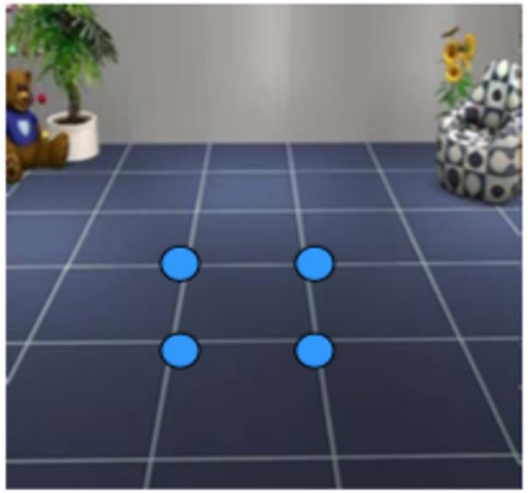
- Using n points to solve for $\mathbf{H}$: the over-determined case

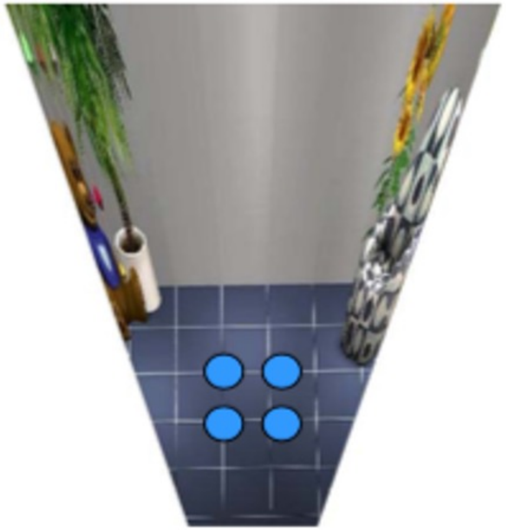
$$\mathbf{A}\mathbf{h} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} \mathbf{h} = \mathbf{0}$$

- How to solve?
  - No exact non-trivial solution due to inexact measurements (e.g., noise)
    - $\mathbf{A}\mathbf{h} = \mathbf{0}$ is not possible, so minimise $||\mathbf{A}\mathbf{h}||$ subject to $||\mathbf{h}|| = 1$

1. Take the singular value decomposition (SVD) of $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$
2. Take the rightmost column of $\mathbf{V}$
    - The right-singular vector of $\mathbf{A}$, corresponding to the smallest singular value (arranged in decreasing singular value order)

Computing $\mathbf{H}_p$





- At least 4 point correspondences needed (8 DoF)

Direct Linear Transformation (DLT) Algorithm for Homography Estimation

- Objective:
  - Given $n \geq 4$ 2D–2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the 3×3 homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i \approx \mathbf{H}\mathbf{x}_i$
- Algorithm:
  1. For each correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ compute $\mathbf{A}_\mathrm{i}$, taking only the first 2 rows
  2. Assemble the n $2{\times}9$ $\mathbf{A}_\mathrm{i}$ matrices into a single $2n{\times}9$ matrix $\mathbf{A}$
  3. Compute the SVD of $\mathbf{A}$: $\mathbf{U\Sigma V}^\top$
  4. Take the last column of $\mathbf{V}$ as the solution for $\mathbf{h}$
  5. Rearrange $\mathbf{h}$ to obtain $\mathbf{H}$ 9*1→3*3

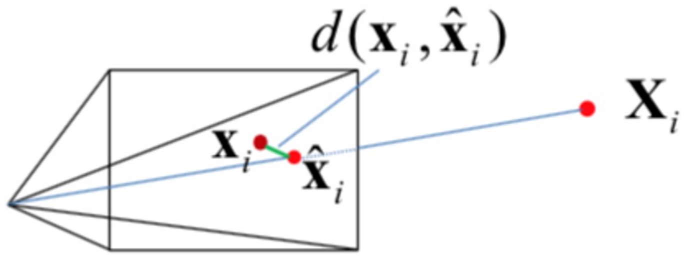
Normalised Direct Linear Transformation (DLT) Algorithm for Homography Estimation

- Objective:
  - Given $n \geq 4$ 2D–2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the 3×3 homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i \approx \mathbf{H}\mathbf{x}_i$
- Algorithm:
  1. Normalise 2D points: $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i,\ \tilde{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$
  2. Apply the DLT algorithm to $\{\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i\}$
  3. Denormalise the recovered solution $\widetilde{\mathbf{H}}$ using $\mathbf{H} = \mathbf{T}'^{-1}\widetilde{\mathbf{H}}\mathbf{T}$
- Example normalisation matrices:

$$\mathbf{T} = \begin{bmatrix} w+h & 0 & w/2 \\ 0 & w+h & h/2 \\ 0 & 0 & 1 \end{bmatrix}^{-1} ; \quad \mathbf{T}' = \begin{bmatrix} w'+h' & 0 & w'/2 \\ 0 & w'+h' & h'/2 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

Geometric Solvers for Homography Estimation

- So far, minimisation of an **algebraic error** criterion
    - Advantage: **linear solutions**
    - Disadvantage: no explicit geometric meaning
- Refinement:
    - Nonlinear minimisation of a **geometric error**
- (Same as for calibration)



Geometric Solvers for Homography Estimation

- Notation:
  - $\mathbf{x}$ measured coordinates
  - $\hat{\mathbf{x}}$ estimated coordinates
  - $d(\cdot,\cdot)$ Euclidean distance in image

pred

- Single-image error: $\hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}} \sum_i d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2$

sum of errors from transformations in both ways

- Symmetric transfer error: $\hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}} \sum_i d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2$

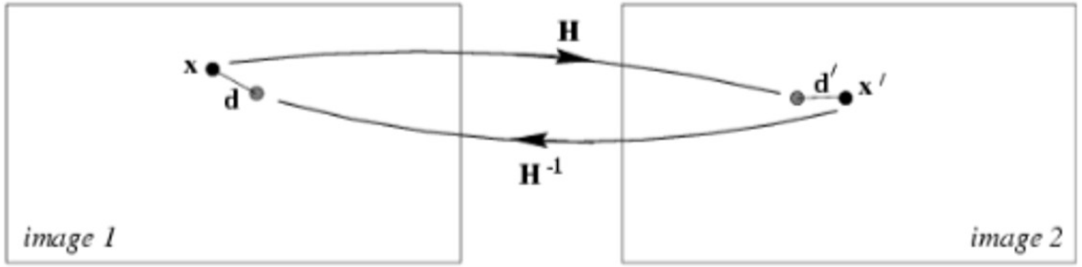
- Reprojection error:

$$\left(\hat{\mathbf{H}}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i\right) = \underset{\mathbf{H}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i}{\operatorname{argmin}} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

subject to $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$

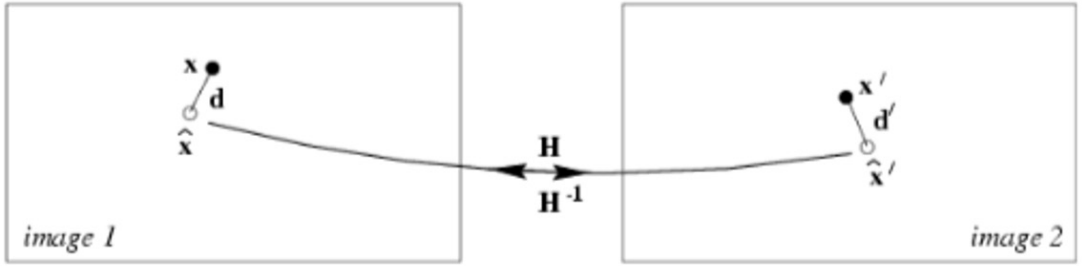
Geometric Solvers for Homography Estimation

symmetric
transfer error



$$d(\mathbf{x}, \mathbf{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathbf{H}\mathbf{x})^2$$

reprojection
error



$$d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

Iterative Refinement

Required to minimise geometric error:

(i) Often slower than DLT

(ii) Requires initialisation

(iii) No guaranteed convergence, local minima

(iv) Stopping criterion required

Initialisation

- Typically, use linear solution e.g. DLT
- If have outliers, use robust algorithm (e.g., M-estimator, or RANSAC)

Robust M-estimators

outliers contribute large residuals, where robust m-estimators can moderate its influence

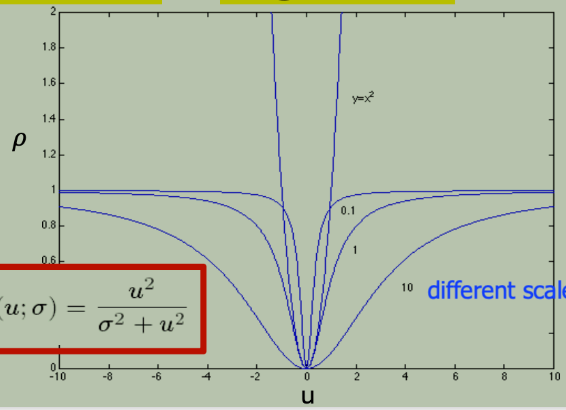
- General approach: find model parameters θ that minimise

$$\boxed{\sum_i \rho(r_i(x_i, \theta); \sigma)}$$

residual; scaler

where:
  - r_i is the residual/loss of the ith point w.r.t. model parameters θ
  - ρ is a robust function with scale parameter σ

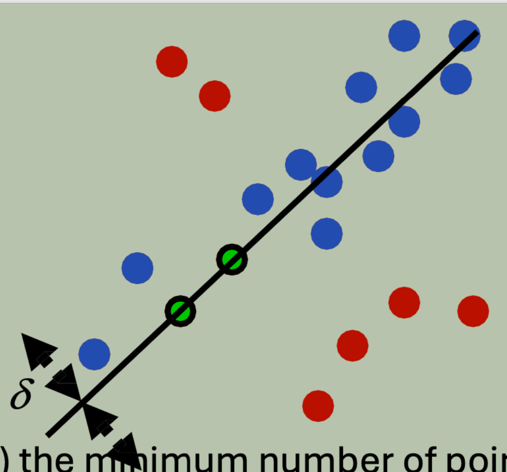
- Robust function behaves like squared distance for small values of the residual but saturates for larger values



$$\boxed{\rho(u;\sigma) = \frac{u^2}{\sigma^2 + u^2}}$$

RANSAC

$N_I = 14$



Algorithm:

1. **Sample** (randomly) the minimum number of points required to fit the model (n=2)
2. **Solve** for model parameters
3. **Score** by the fraction of inliers within a preset threshold

Repeat 1–3 until the best model is found

Illustration by Savarese

Iterative Methods

Many iterative algorithms exist

- Newton's method
- Levenberg-Marquardt

- Powell's method
- Simplex method

"Gold Standard" Algorithm for Homography Estimation

- Objective:
  - Given $n \geq 4$ 2D–2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the 3×3 homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i \approx \mathbf{H}\mathbf{x}_i$
- Algorithm:
  1. Initialisation: compute an initial estimate using normalised DLT or RANSAC
  2. Geometric minimisation of reprojection error:
     1. Minimise using Levenberg–Marquardt over the 9 entries of $\mathbf{H}$
     2. Compute initial estimate for optimal $\{\mathbf{x}_i\}$
     3. Minimise cost $\sum d(\mathbf{x}_\mathrm{i}, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_\mathrm{i}, \hat{\mathbf{x}}'_\mathrm{i})^2$ over $\{\mathbf{H}, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_\mathrm{n}\}$
  3. Denormalise the recovered solution

“Gold Standard” Algorithm for Homography Estimation

<u>Objective</u>

Compute homography between two images

<u>Algorithm</u>

(i) **Interest points:** Compute interest points in each image

(ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure

(iii) **RANSAC robust estimation:** Repeat for *n* samples

(a) Select 4 correspondences and compute **H** with DLT

(b) Calculate the distance $d_\perp$ for each putative match for each chosen correspondence

(c) Compute the number of inliers consistent with **H** ($d_\perp < t$) where d⊥ is lower than a threshold t

(d) Choose **H** with most inliers

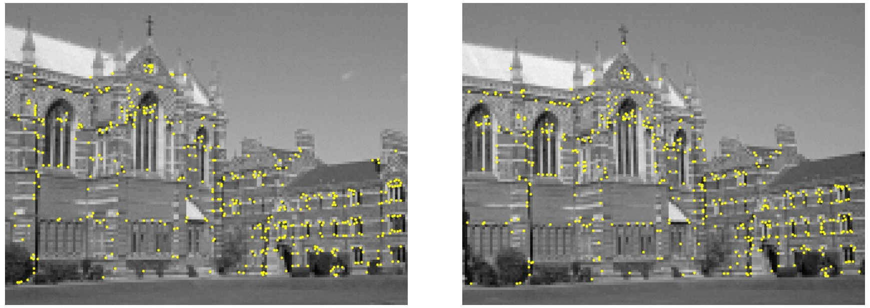
initialised from the last step

(iv) **Optimal estimation:** re-estimate **H** from all inliers by minimising geometric cost function with the Levenberg-Marquardt algorithm

(v) **Guided matching:** Determine more matches using prediction with computed **H**

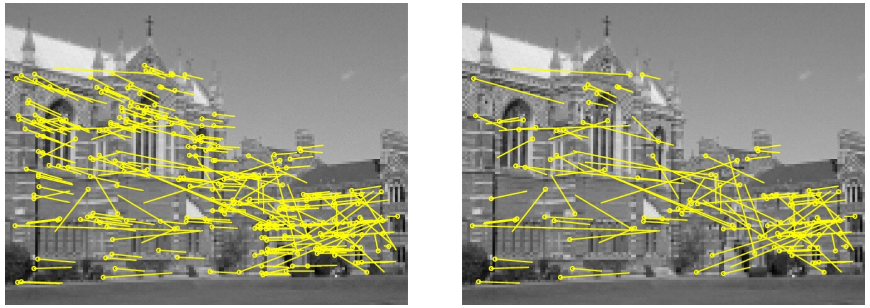
Optionally iterate last two steps until convergence

Example: Robust Computation



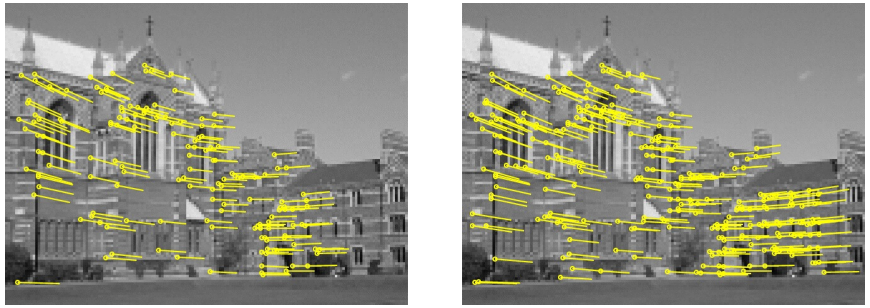
Interest points

(500/image)



Left: Putative correspondences (268)

Right: Outliers (117)



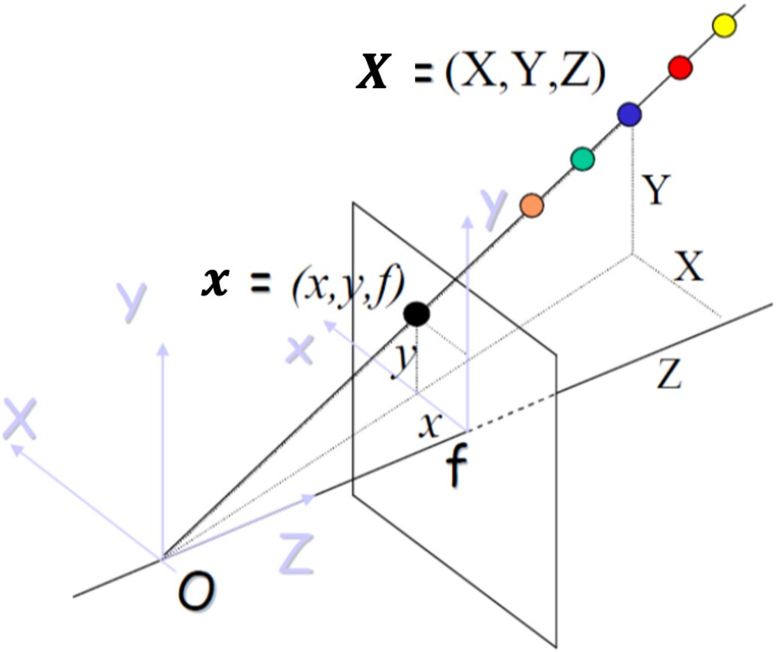
Left: Inliers (151) after RANSAC with DLT

Right: Final inliers (262) after Levenberg-Marquardt and guided matching

Epipolar Geometry

Two-view Geometry

Why Two Views?



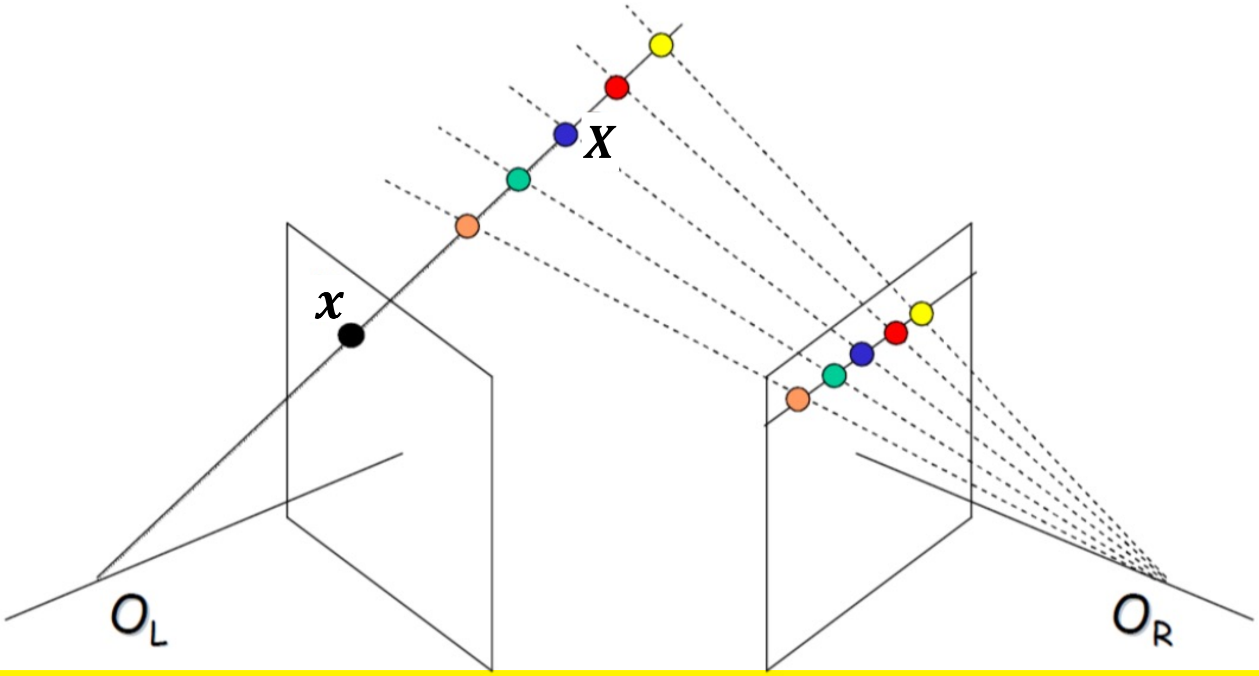
$$x = f\frac{X}{Z} = f\frac{kX}{kZ}$$

$$y = f\frac{Y}{Z} = f\frac{kY}{kZ}$$

- A fundamental *ambiguity*: any point on the ray OX has projection point $\boldsymbol{x}$ on the image plane
  - Note that here the camera center is (0,0,0) in the world coordinate system

Image Credit: Robert Collins

Why Two Views?

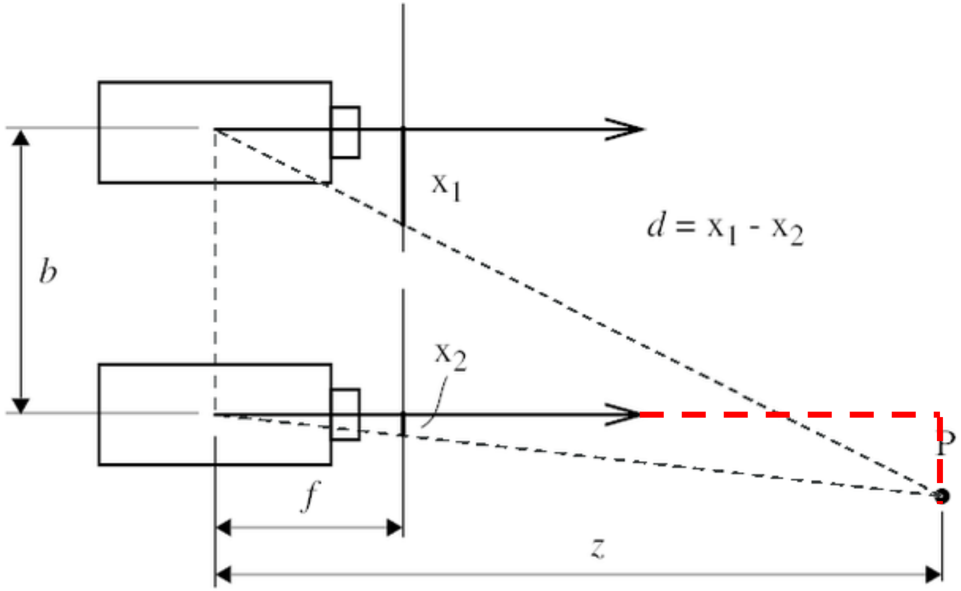


- A second camera can resolve the ambiguity, enabling measurement of depth via triangulation

Image Credit: Robert Collins

General Two-view Geometry

- Human stereopsis
- Stereograms
- Epipolar geometry and the epipolar constraint
  - Parallel optical axes
  - General case with calibrated cameras



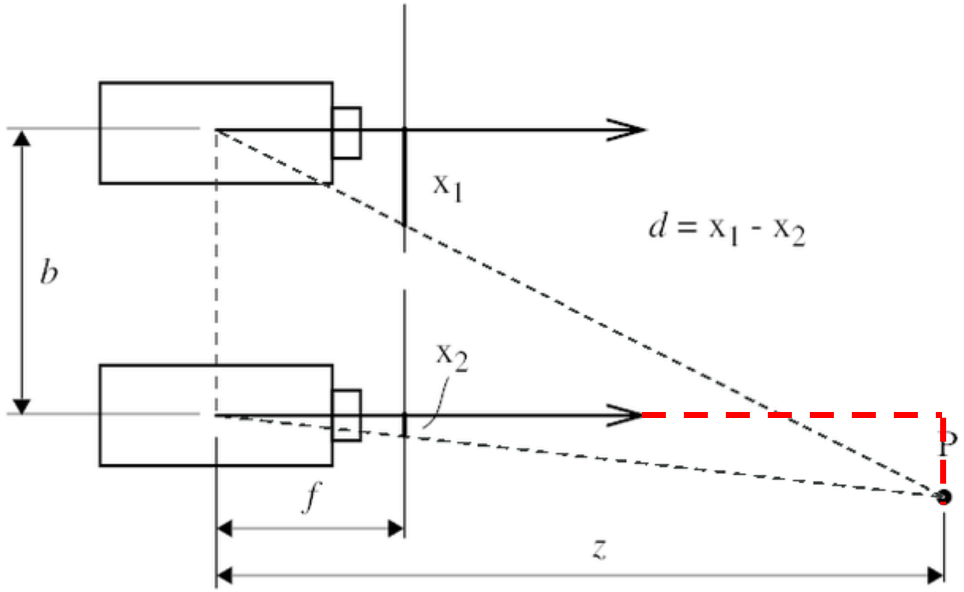
For similar triangles,

$$\frac{d}{b} = \frac{f}{z}$$

Parallel Optical Axes

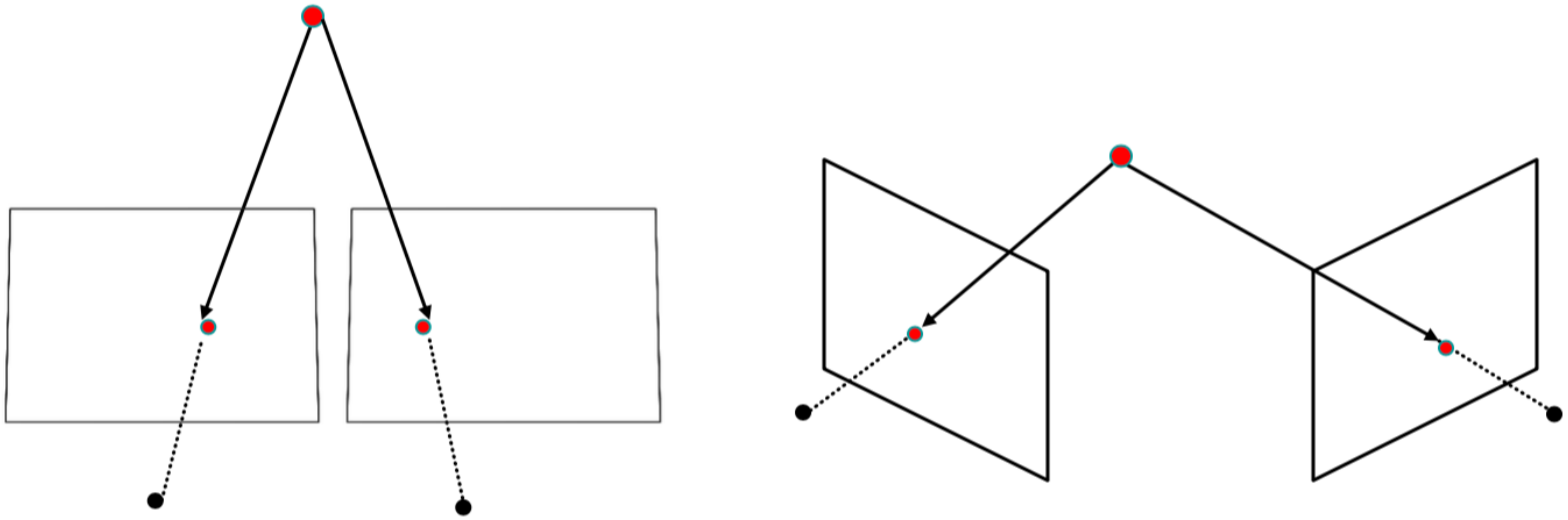
- Depth from disparity d

1. $\frac{z-f}{z} = \frac{b-x_1+x_2}{b}$ [similar △s]
2. $1 - \frac{f}{z} = 1 - \frac{d}{b}$
3. $z = \frac{bf}{d}$ [depth of 3D point]



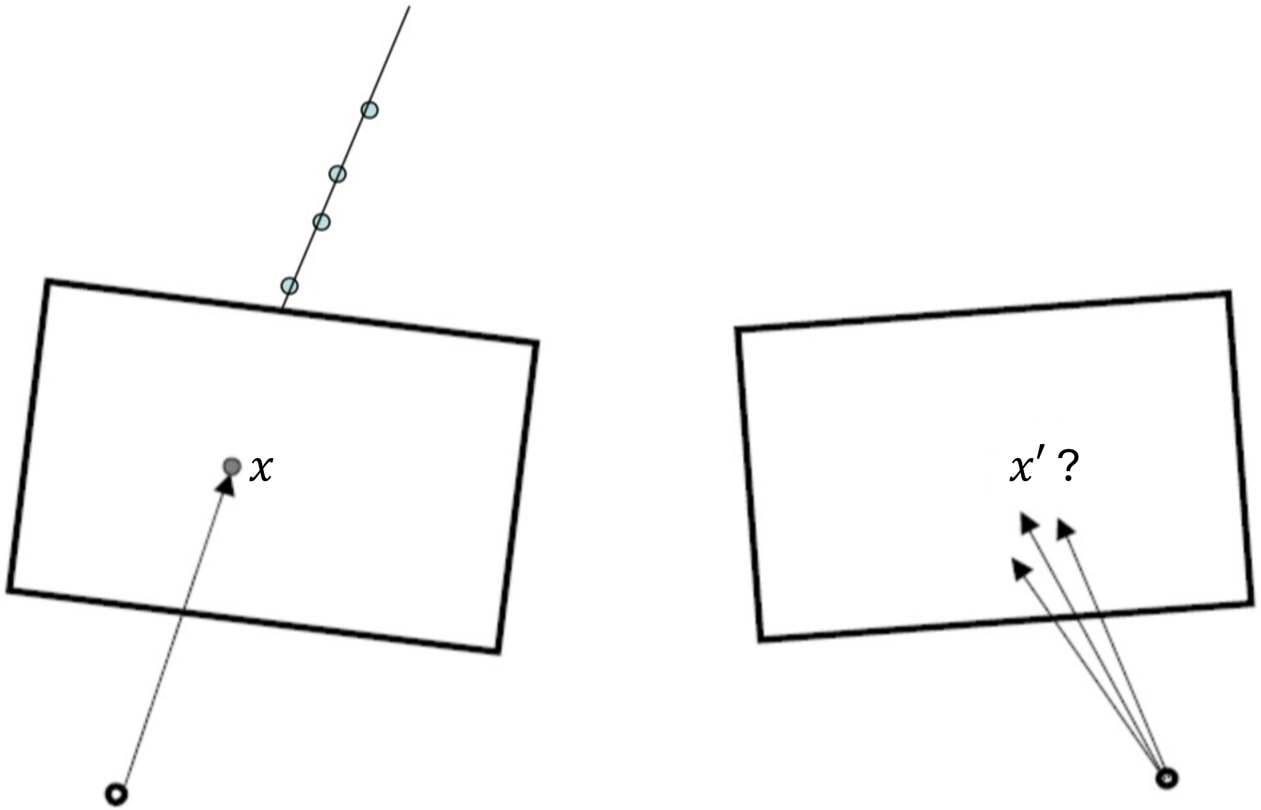
General Case with Calibrated Cameras

- The two cameras need not have parallel optical axes



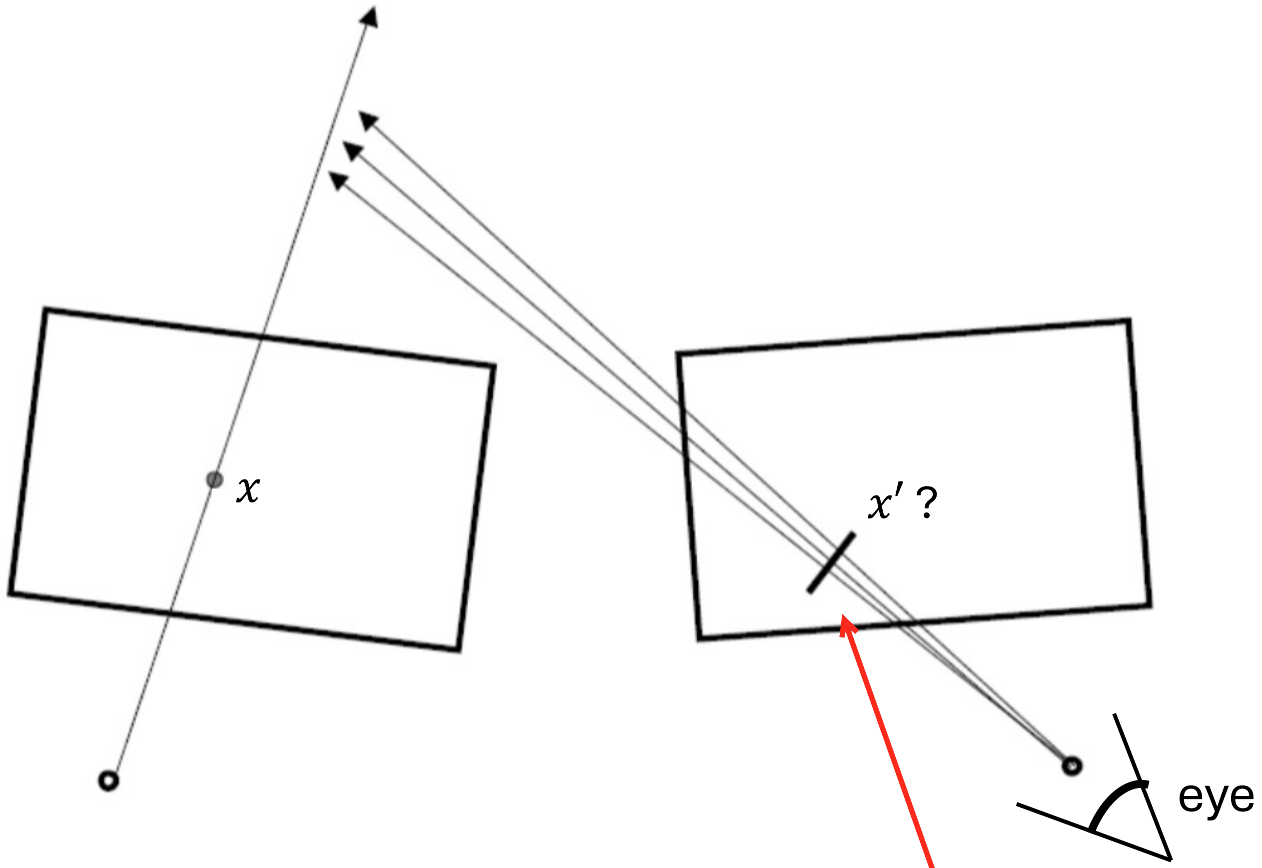
Parallel Vs. Non-parallel

Stereo Correspondence Constraints



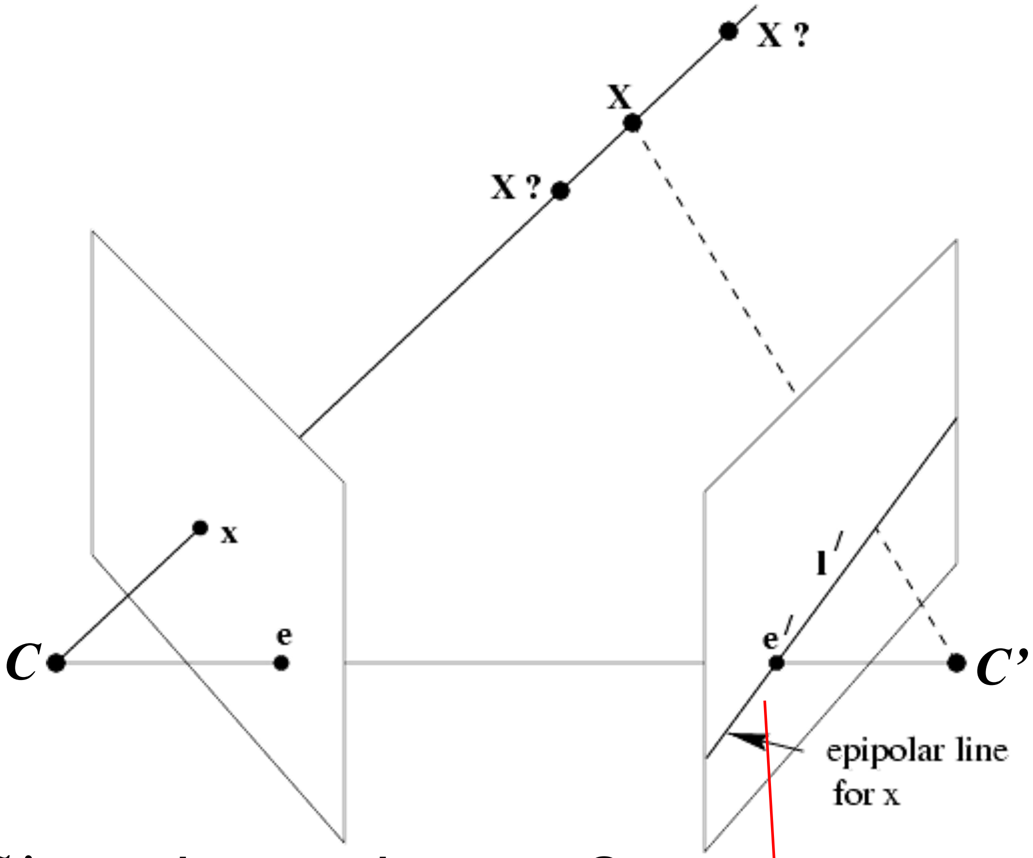
- Given x in the left image, where can the corresponding point x' be?

Stereo Correspondence Constraints



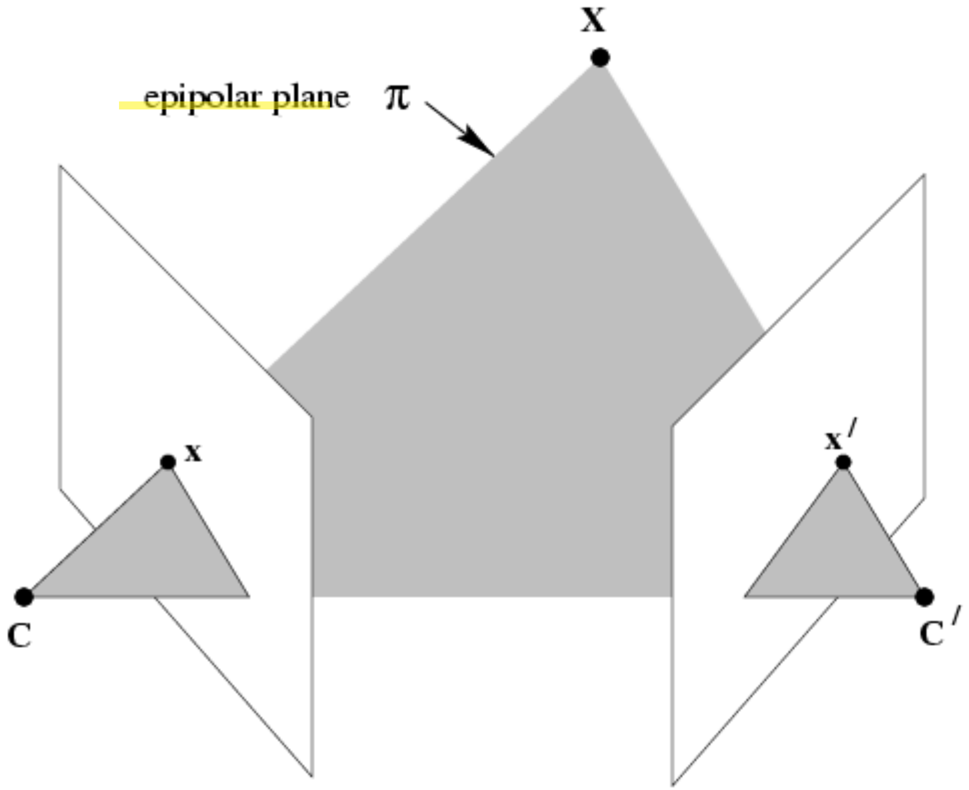
- If we, from the second image, view the ray going through x, this ray will form a **line** on the second image
- Point x' will be on this line in the second image

Epipolar Geometry



- What if only C, C', and x are known?
- We can obtain the epipolar line on the second image

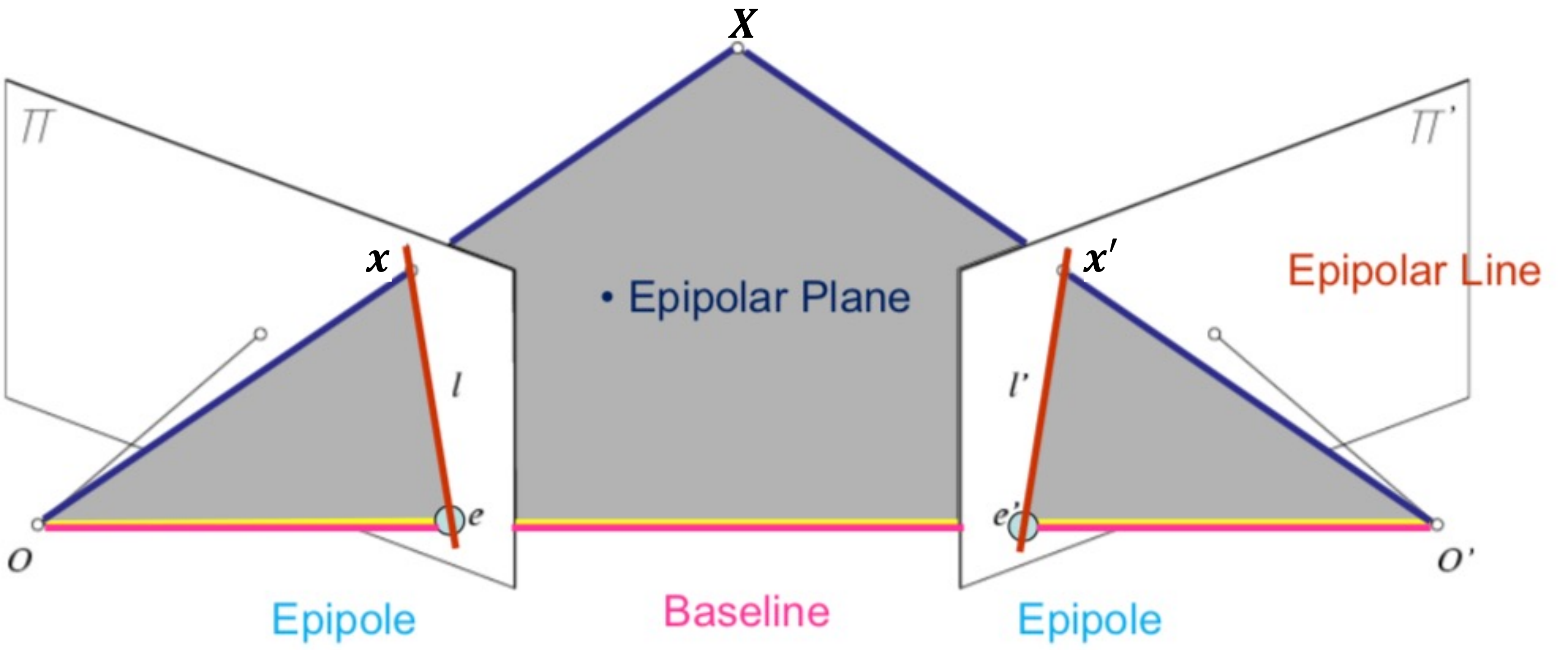
Epipolar Geometry



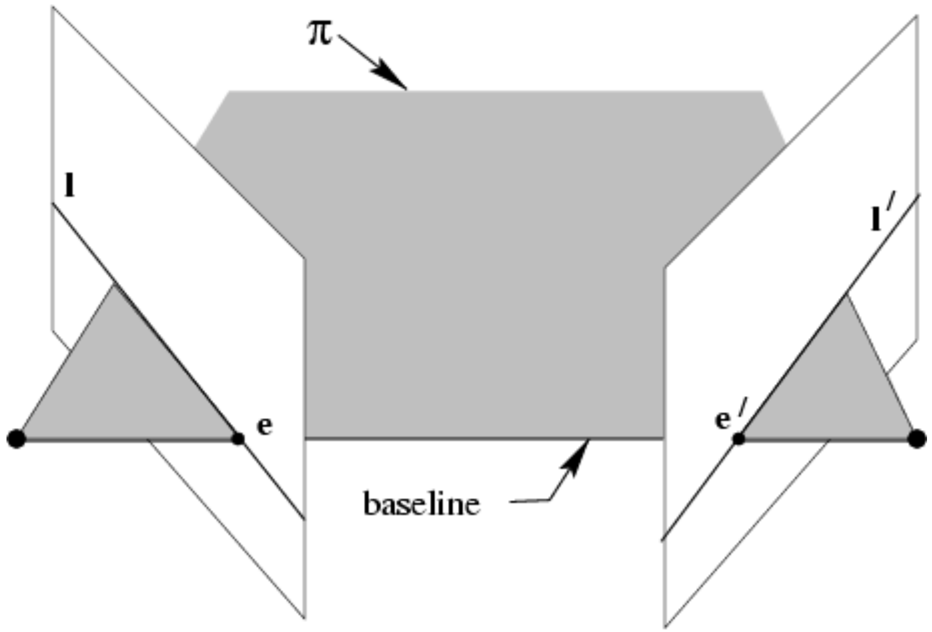
- C, C', x, x' and X are coplanar

Epipolar Geometry Terminology

- **Baseline:** line joining the camera centres
- **Epipole:** point of intersection of baseline with image plane
- **Epipolar plane:** plane containing baseline and world point
- **Epipolar line:** intersection of epipolar plane with the image plane



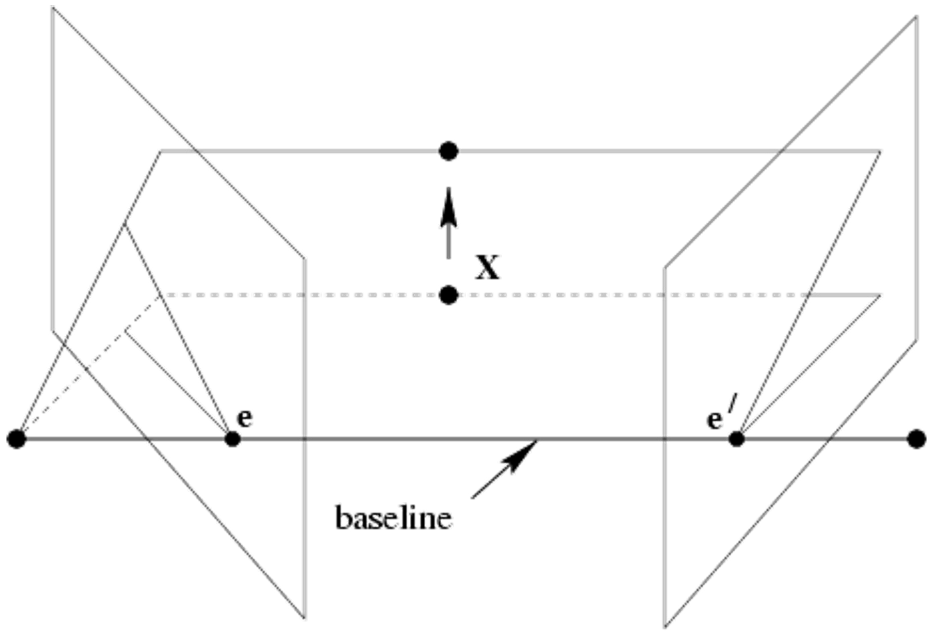
The Epipolar Plane



- All points on π project on *l* and *l'*

Epipolar Geometry

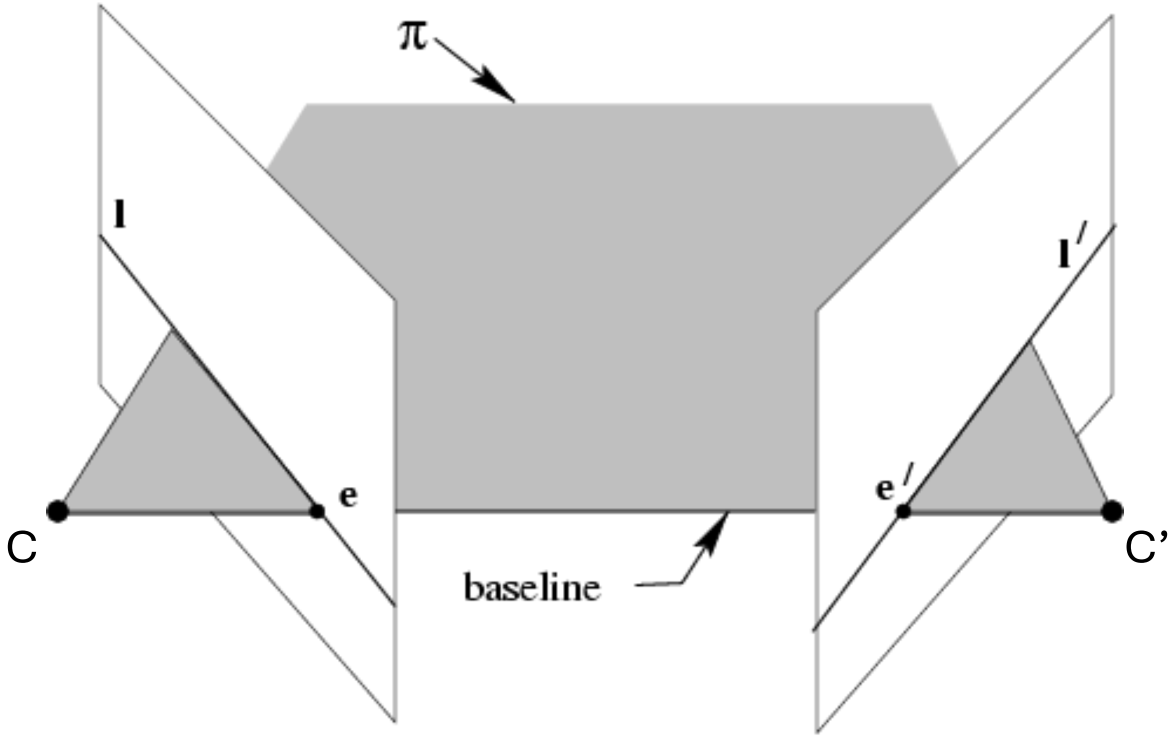
- All epipolar lines intersect at the epipole



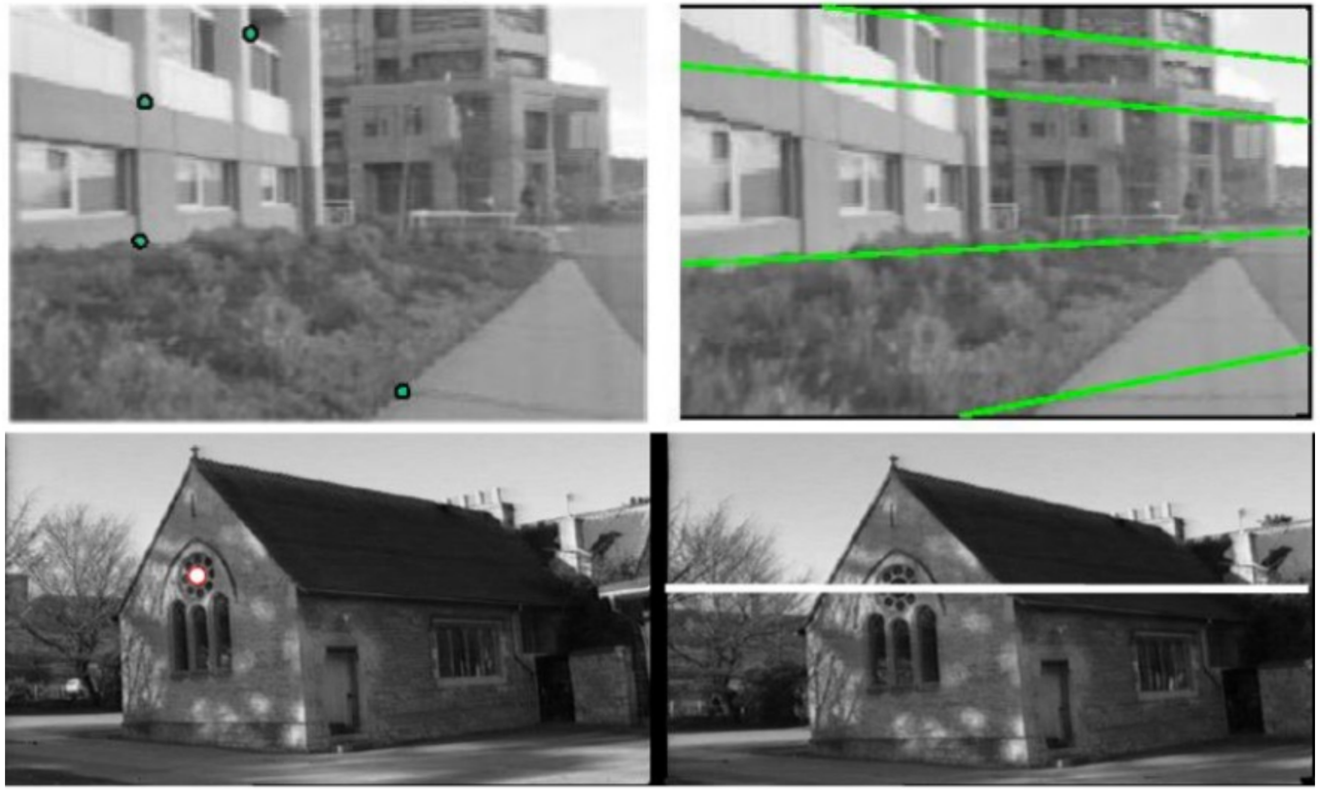
- Family of planes π and lines *l* and *l'* intersect at *e* and *e'*

Epipolar Geometry

- Epipole = intersection of baseline with image plane
  = projection of one optical centre in the other image

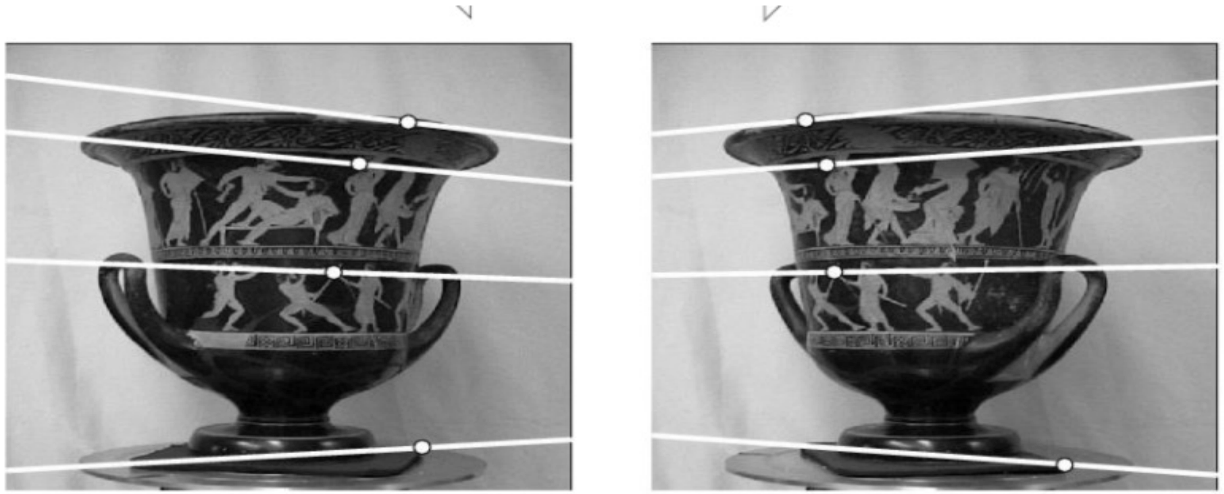


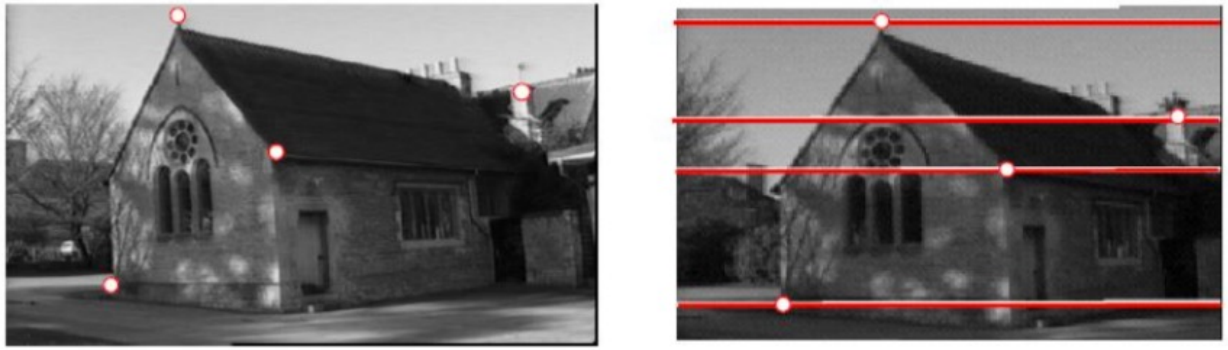
Examples



- Useful because it reduces the correspondence problem to a 1D search along an epipolar line

Examples

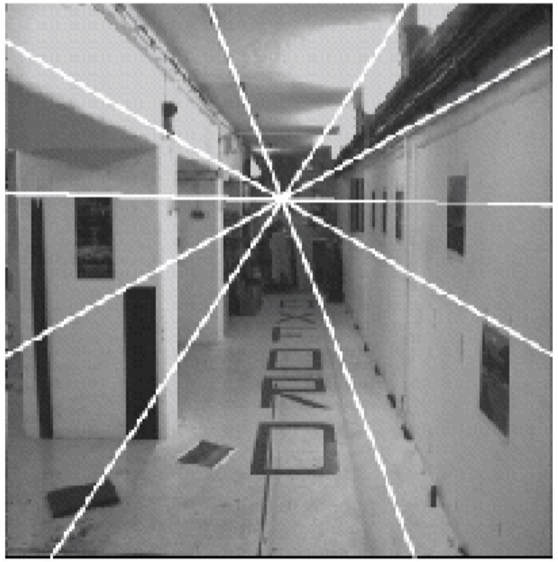


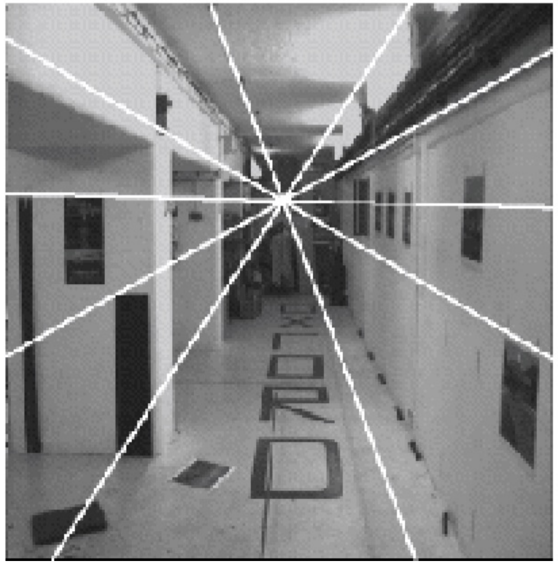


Examples

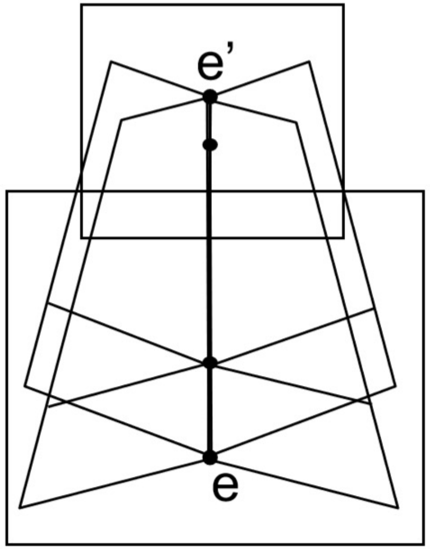
Image Credit: .gs8

Example: Forward Motion





- All epipolar lines go through the epipole, which is visible in the image plane in this case



- The epipole is the projection of an optical centre onto the other image

Epipolar Geometry: Summary

- The intrinsic geometry of 2 different views of the same 3D scene

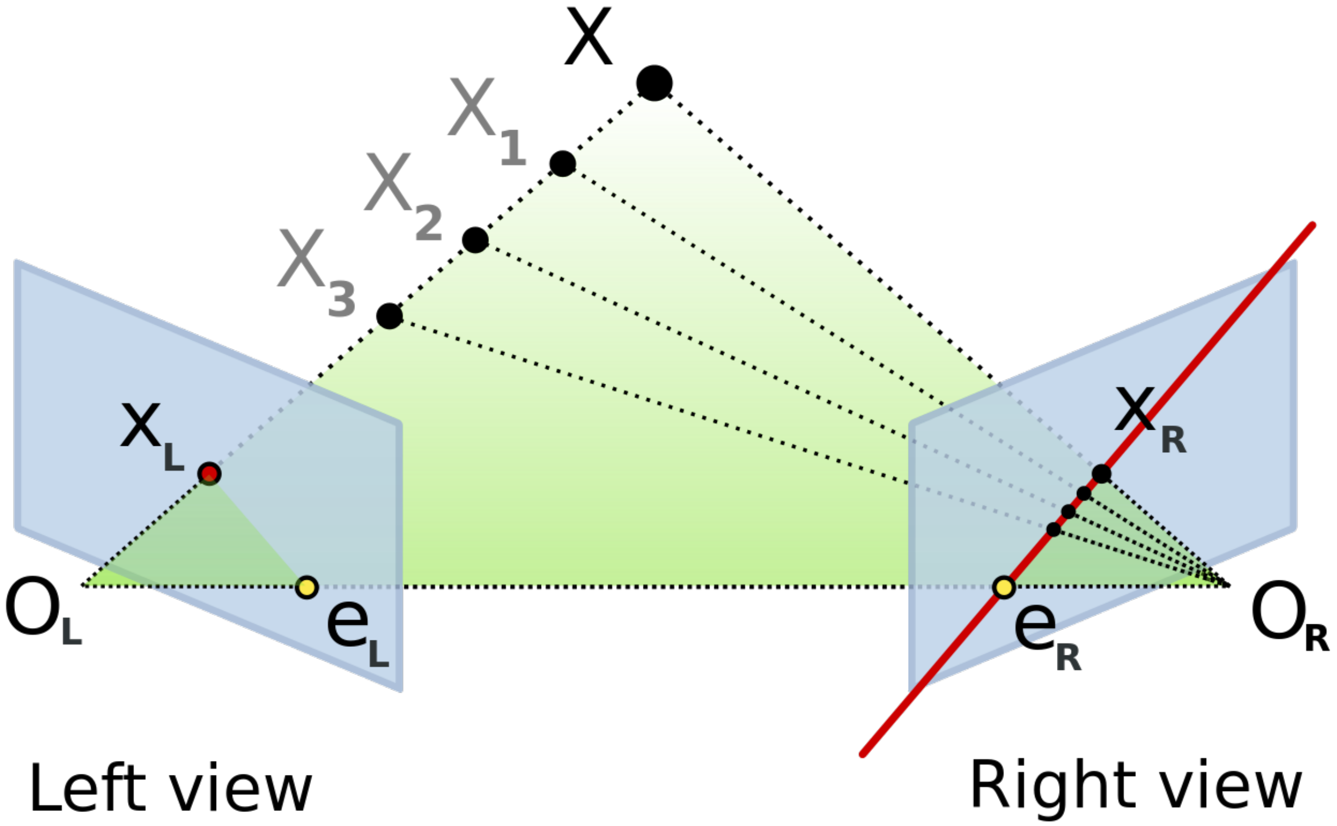


Image Credit: Arne Nordmann

Next Week

- Two-view geometry:
  - Epipolar geometry: Essential and Fundamental matrices
  - Triangulation
  - Stereo