

3D Vision 3

Week 9

Two-view Geometry: Epipolar Geometry

Two-view Geometry: Essential and Fundamental Matrices

Announcements

- **Tutorial 6:** available now on Wattle, runs across Weeks 9–10
- **Survey 2 – Week 8|9** is live on Wattle, please provide a response to improve the course for future years
- **Class Representatives:** another mechanism to provide feedback
 - Arjun Raj (u7526852@anu.edu.au) [COMP4528]
 - Ruxuan Li (u7763307@anu.edu.au) [ENGN4528]
 - Jiabao Han (u7765516@anu.edu.au) [COMP6528]
 - Xingchen Zhang (u7670173@anu.edu.au) [COMP6528]
- **Exam:** 9:00am–12:15pm Saturday 1 June 2024
 - 7-11 Barry Drive, First Floor Left Side

Weekly Study Plan: Overview

Wk	Starting	Lecture	Lab	Assessment
1	19 Feb	Introduction	X	
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	X	
	8 Apr	Teaching break	X	
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	X	CLab3 report due Friday
12	20 May	Course Review	X	

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

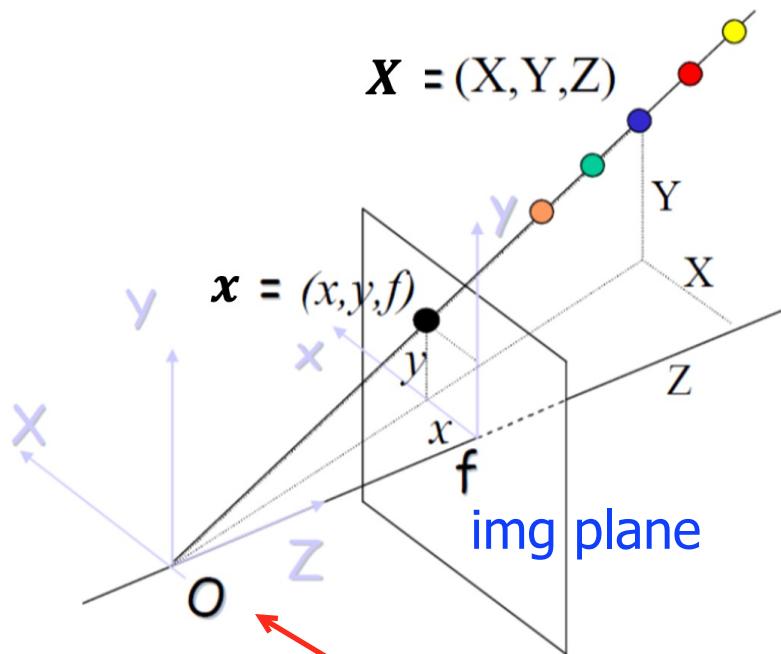
Outline

1. Two-view Geometry: Epipolar Geometry
2. Two-view Geometry: Essential and Fundamental Matrices
3. Two-view Geometry: Triangulation
4. Two-view Geometry: Stereo

Epipolar Geometry

Two-view Geometry

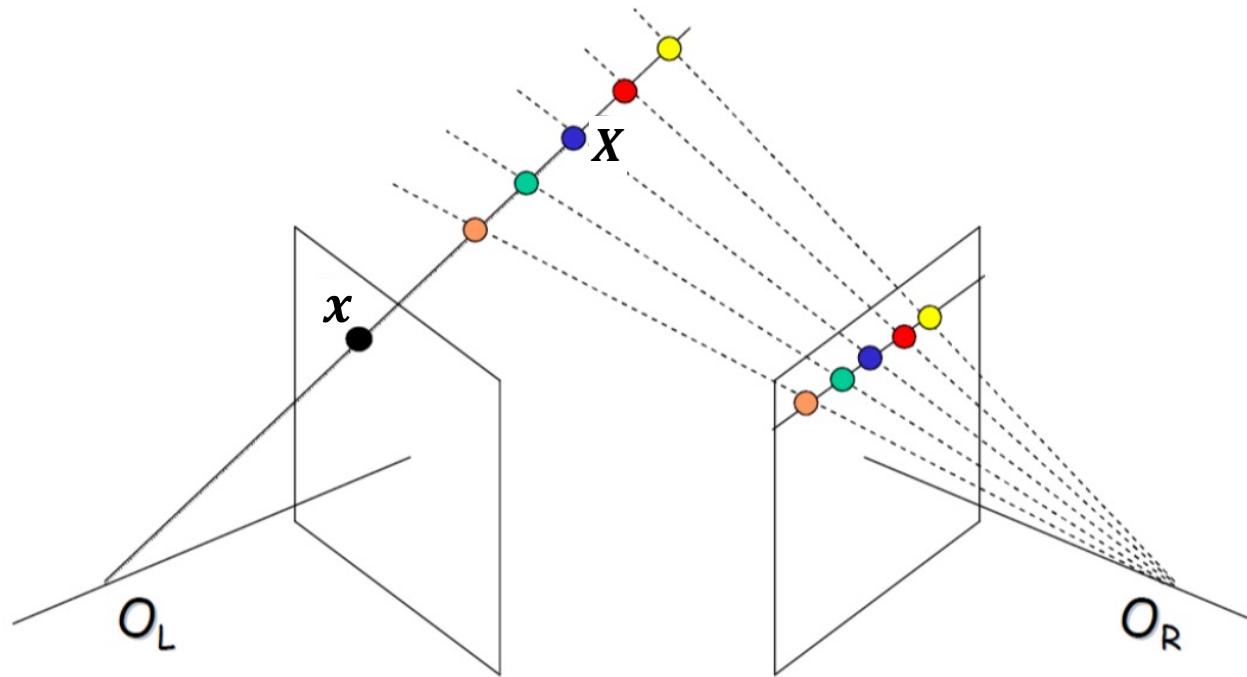
Why Two Views?



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

$$x = f \frac{X}{Z} = f \frac{kX}{kZ} \quad k \text{ represents a ray}$$
$$y = f \frac{Y}{Z} = f \frac{kY}{kZ}$$

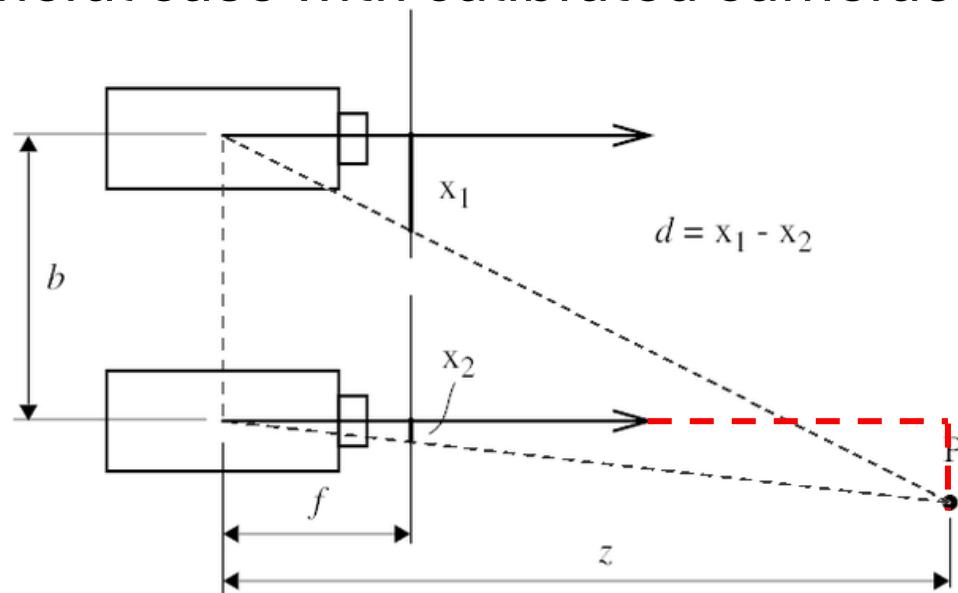
Why Two Views?



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

General Two-view Geometry

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



For similar triangles,

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

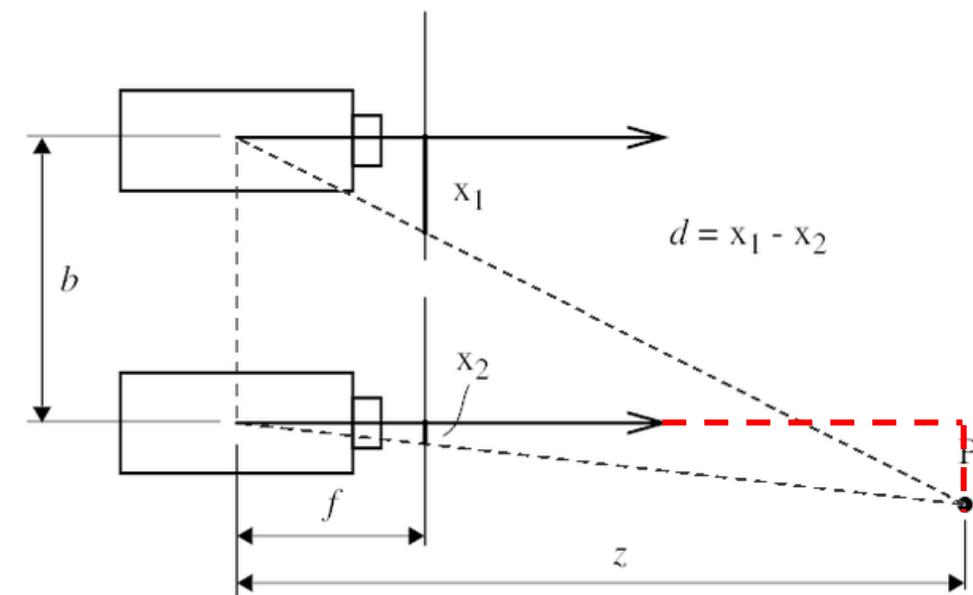
Parallel Optical Axes

- Depth from disparity d

$$1. \frac{z-f}{z} = \frac{b-x_1+x_2}{b} \text{ [similar } \triangle s]$$

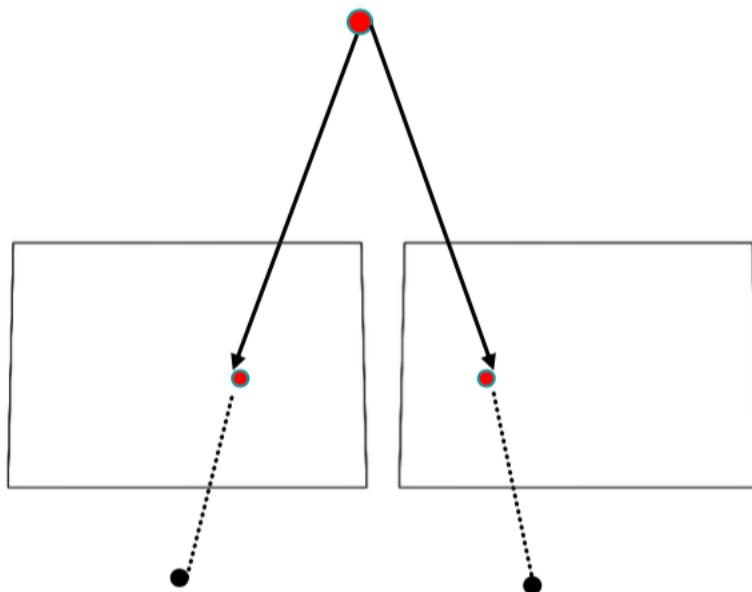
$$2. 1 - \frac{f}{z} = 1 - \frac{d}{b}$$

$$3. z = \frac{bf}{d} \text{ [depth of 3D point]}$$



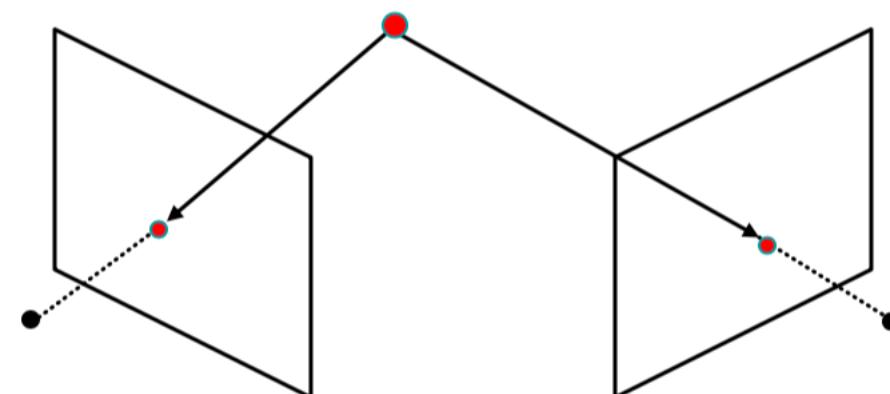
General Case with Calibrated Cameras

- The two cameras need not have parallel optical axes



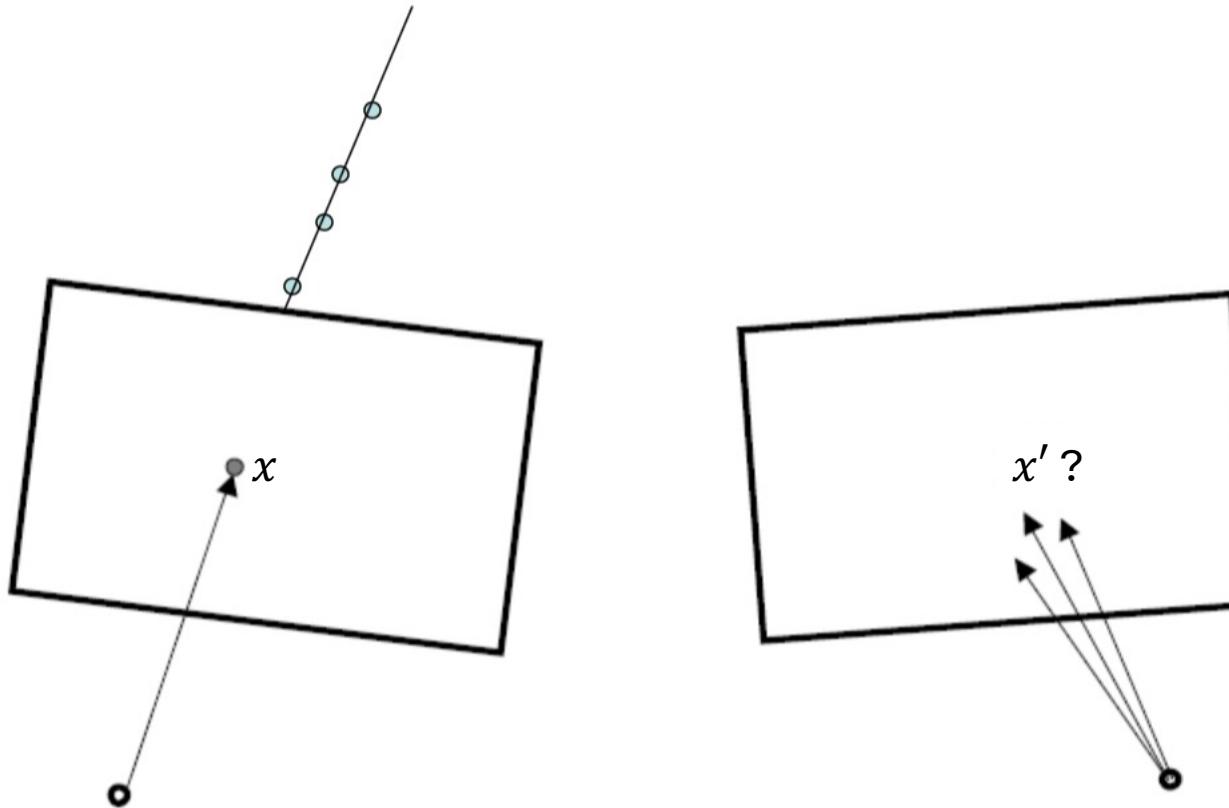
Parallel

Vs.



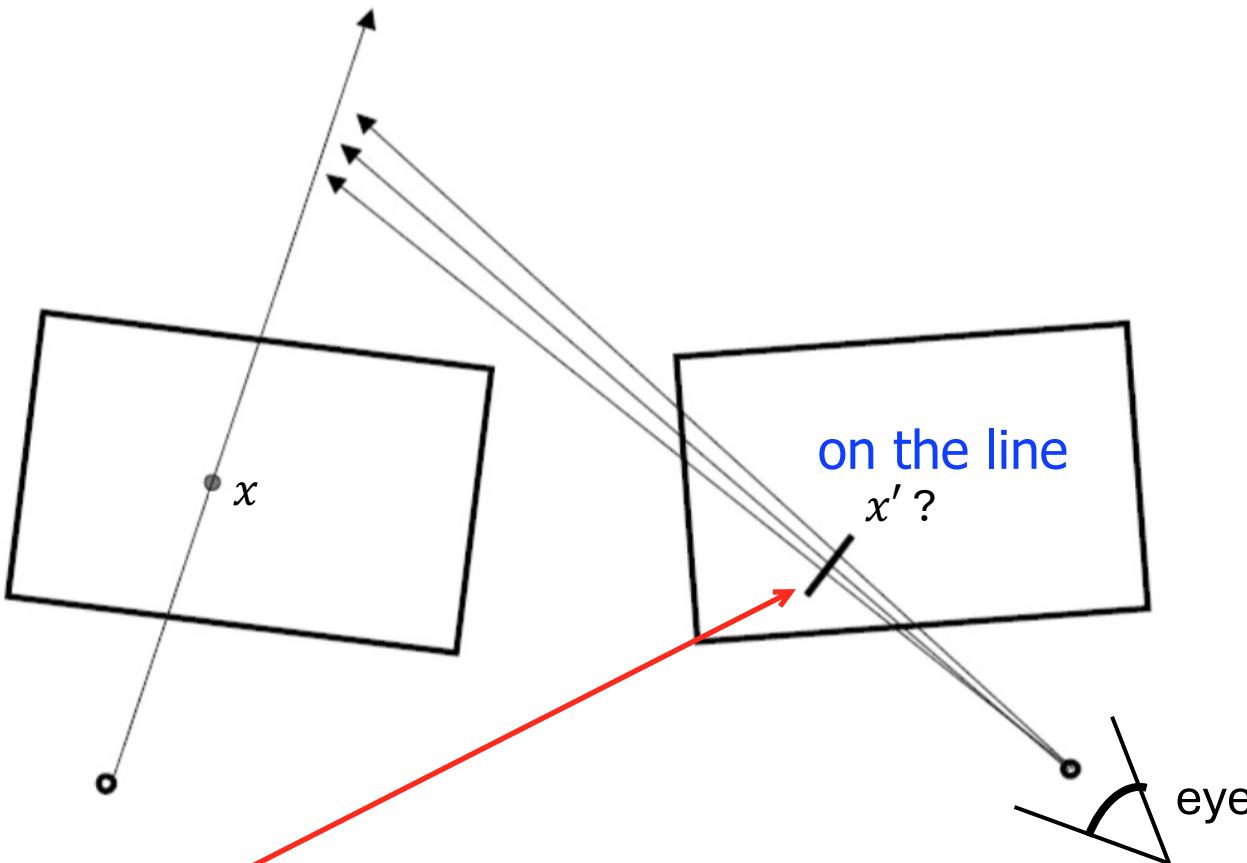
Non-parallel

Stereo Correspondence Constraints



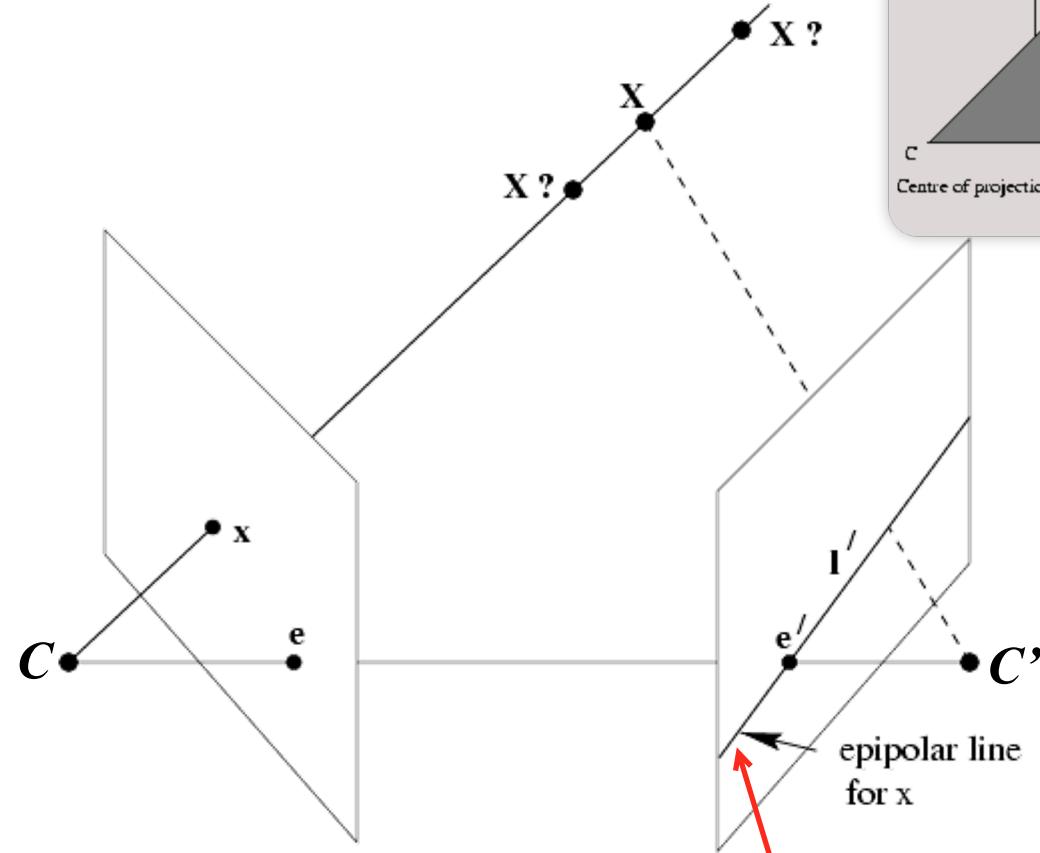
- Given x in the left image, where can the corresponding point x' be?

Stereo Correspondence Constraints

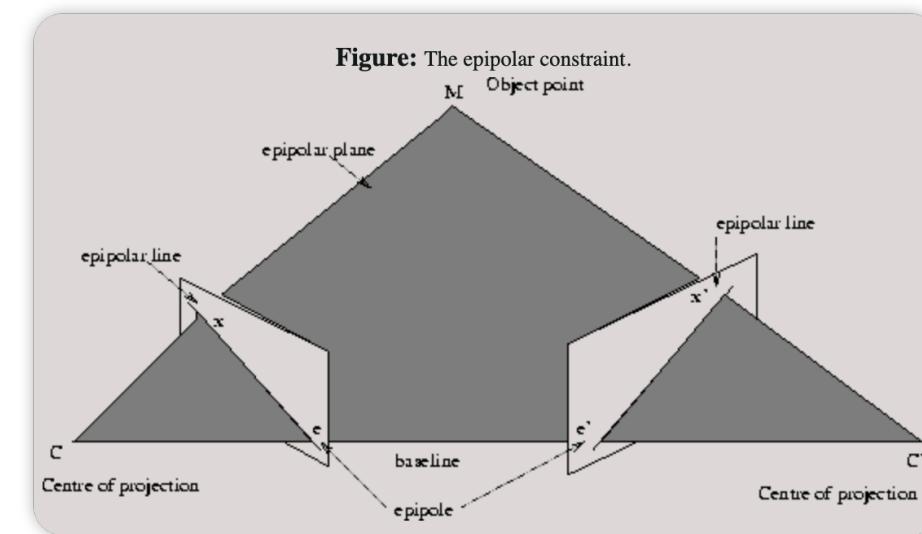


- If we, from the second image, view the ray going through x , this ray will form a **line** on the second image
- Point x' will be on this line in the second image

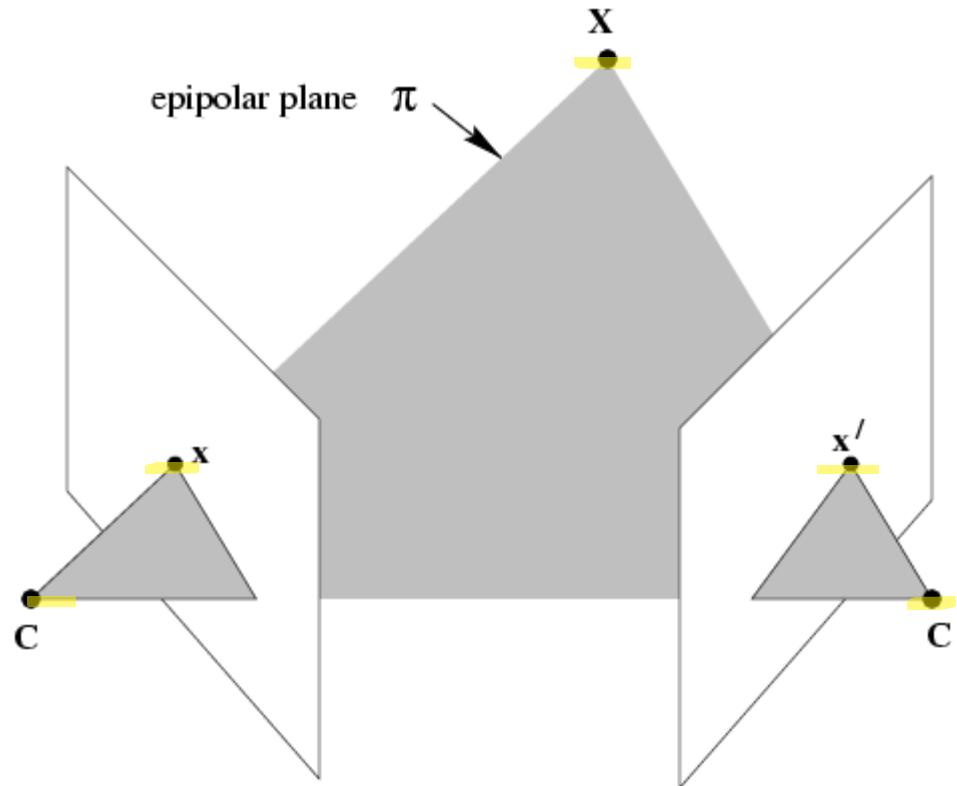
Epipolar Geometry



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



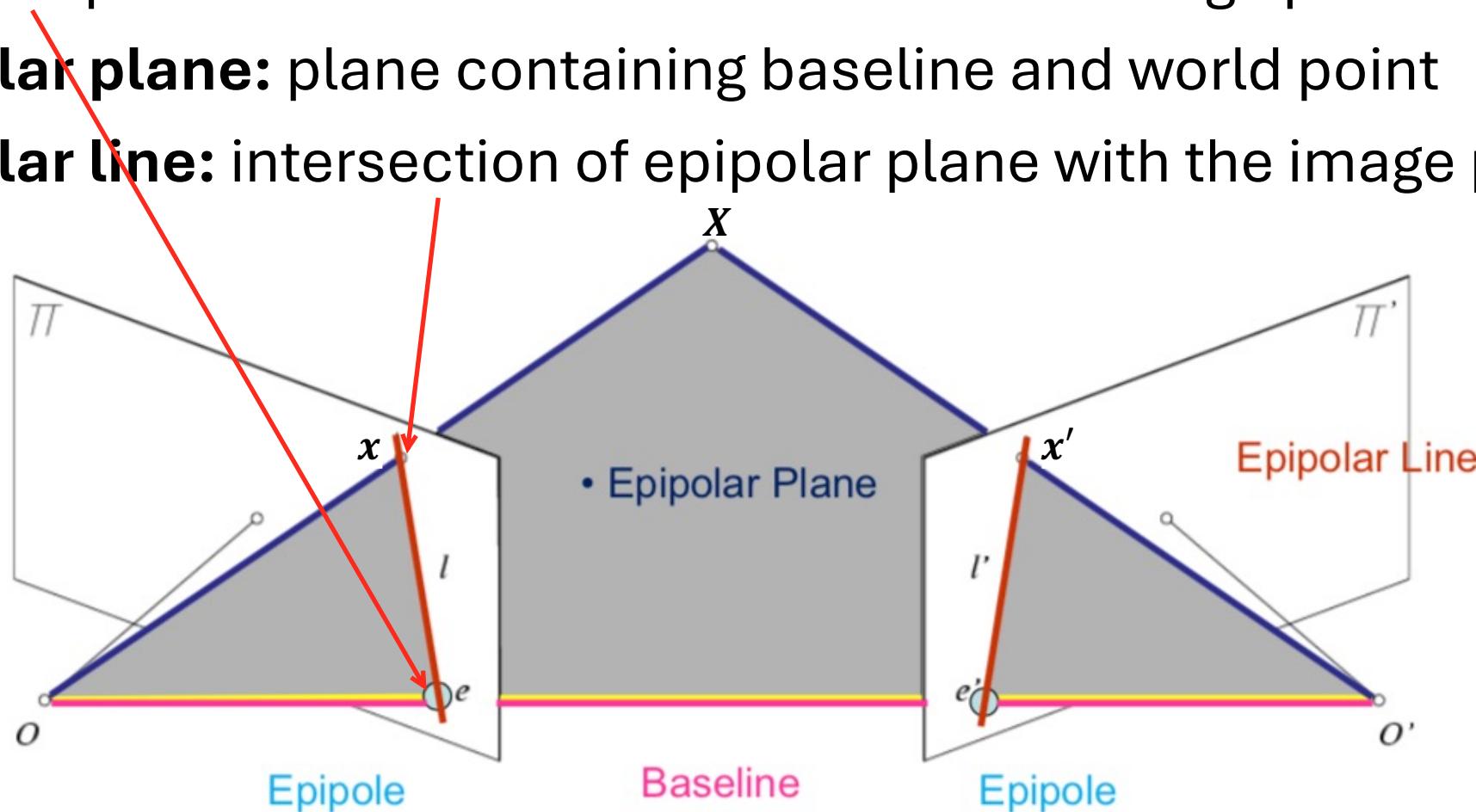
Epipolar Geometry



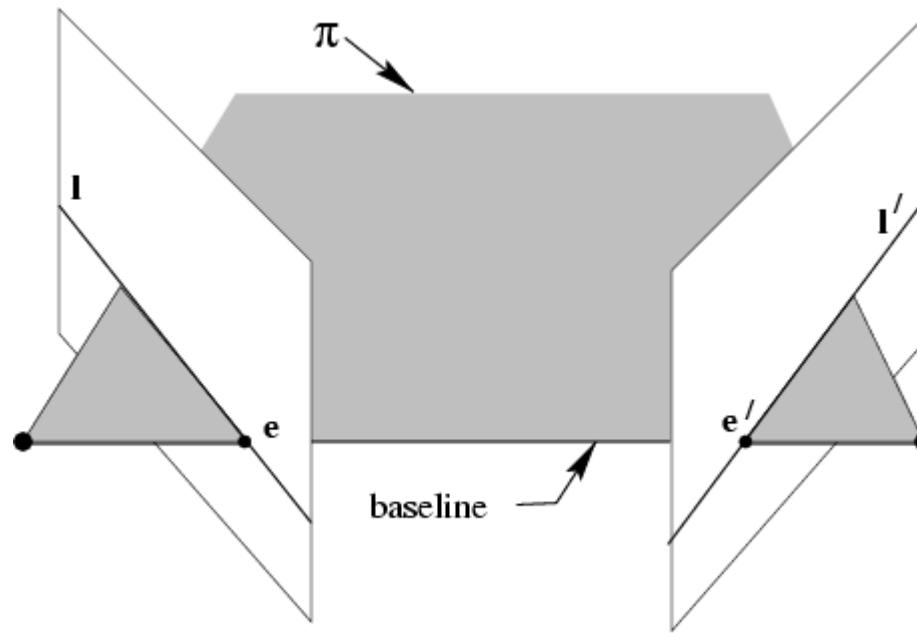
- C, C', x, x' and X are **coplanar**

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



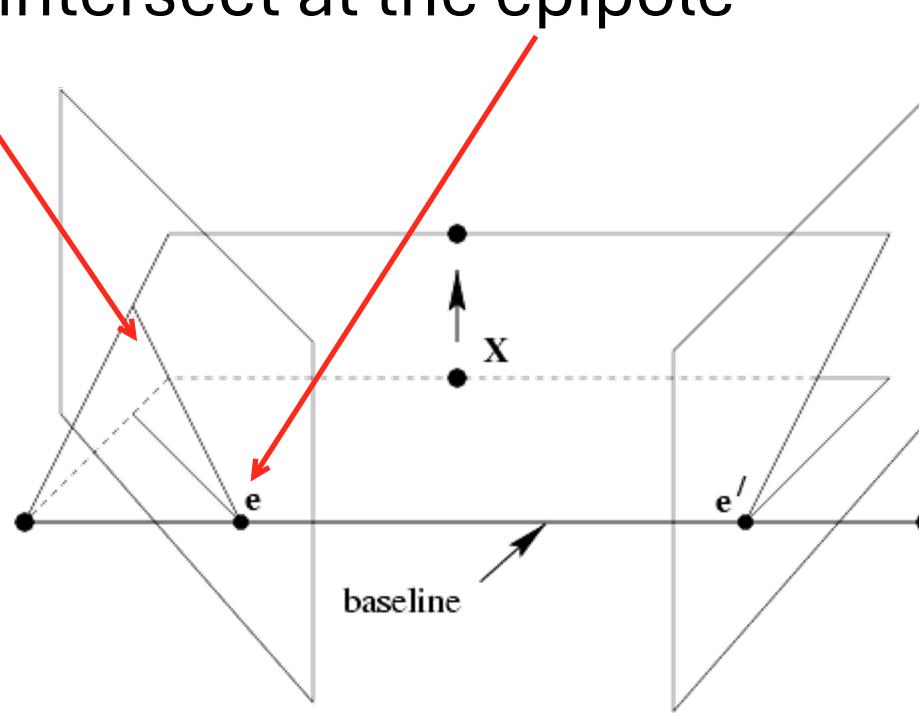
The Epipolar Plane



- All points on π project on l and l'

Epipolar Geometry

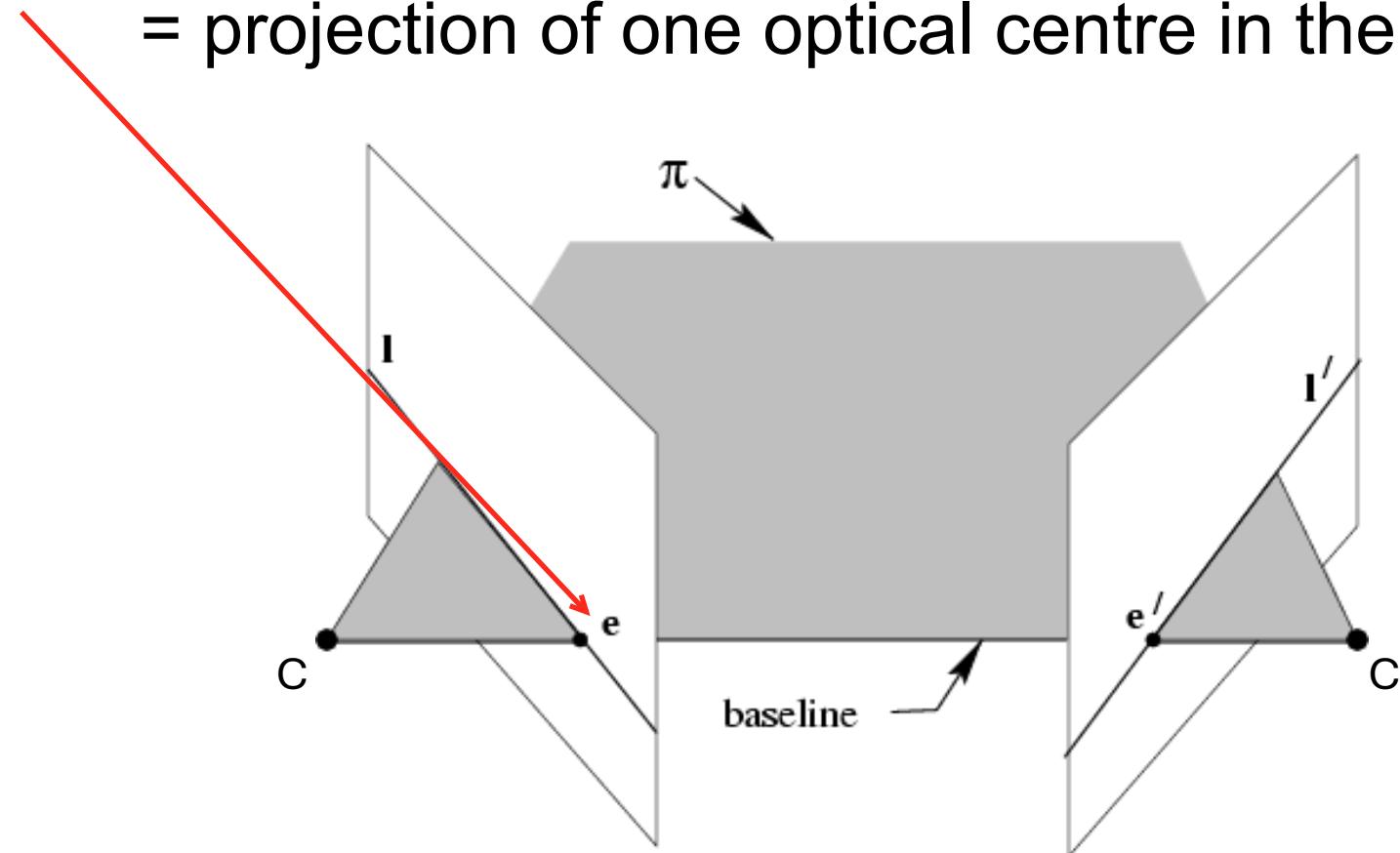
- All epipolar lines intersect at the epipole



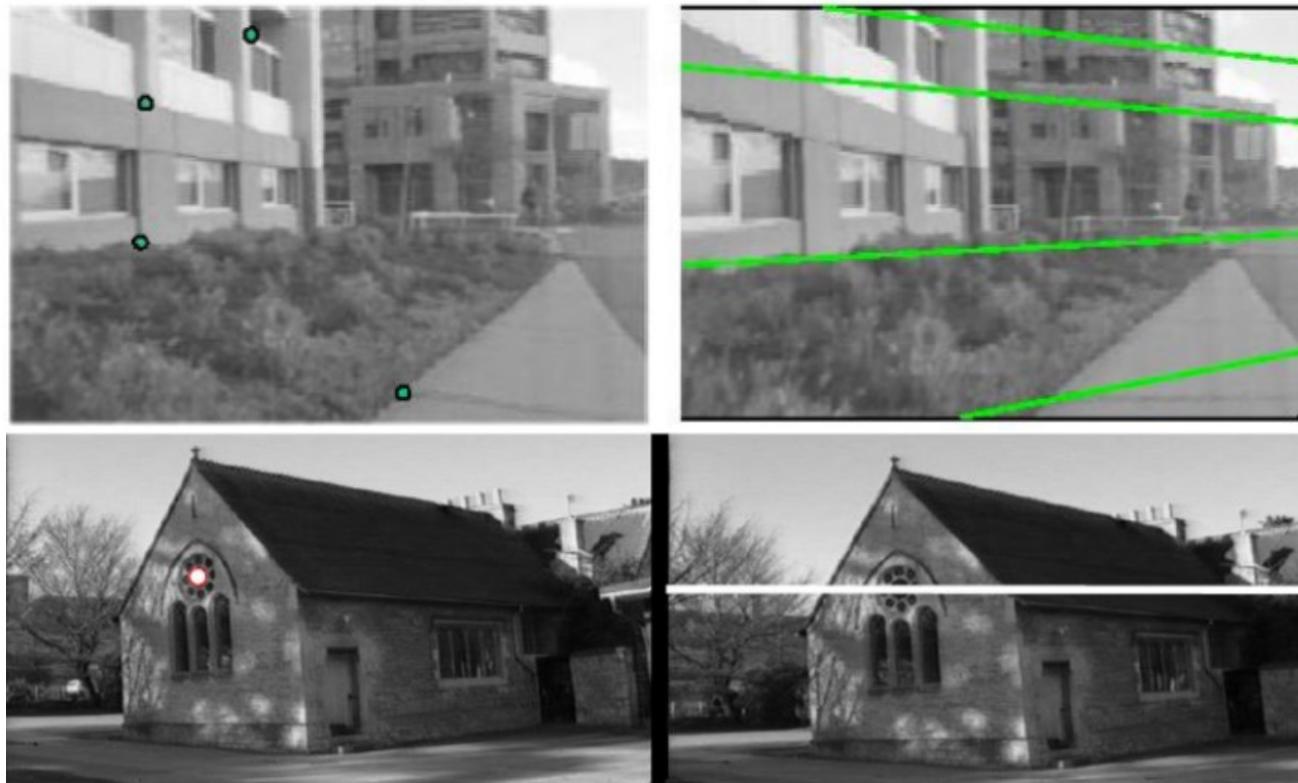
- Family of planes π and lines l and l' intersect at e and e'

Epipolar Geometry

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

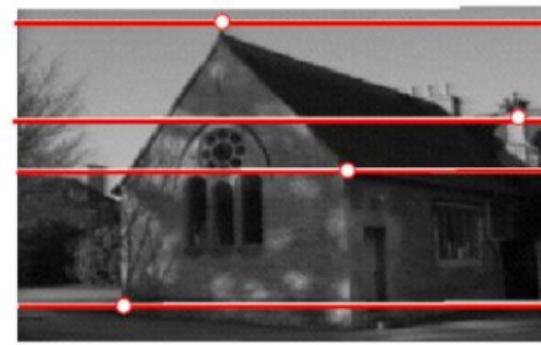
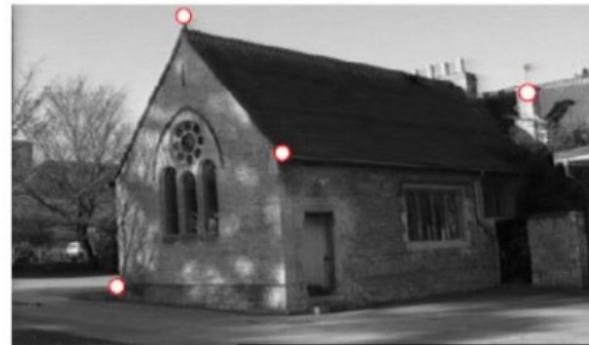


Examples



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

Examples

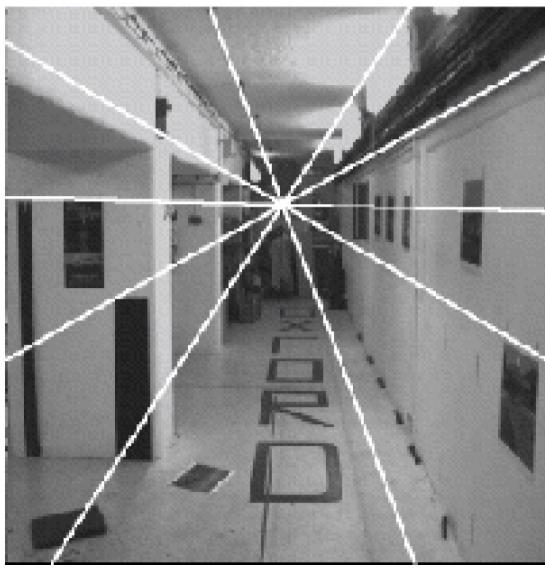
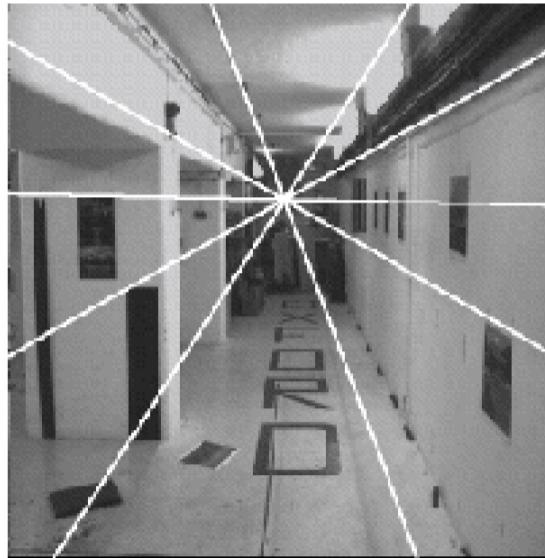


Examples

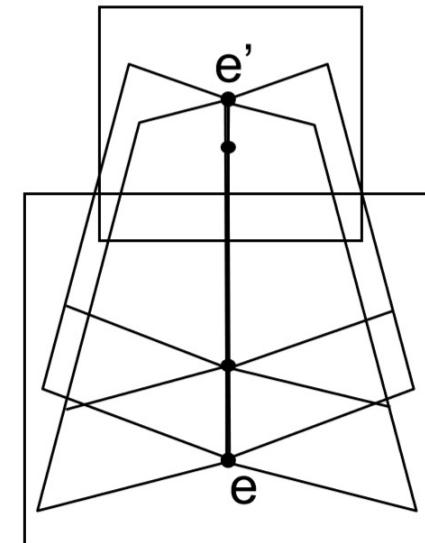


Image Credit: .gs8 22

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

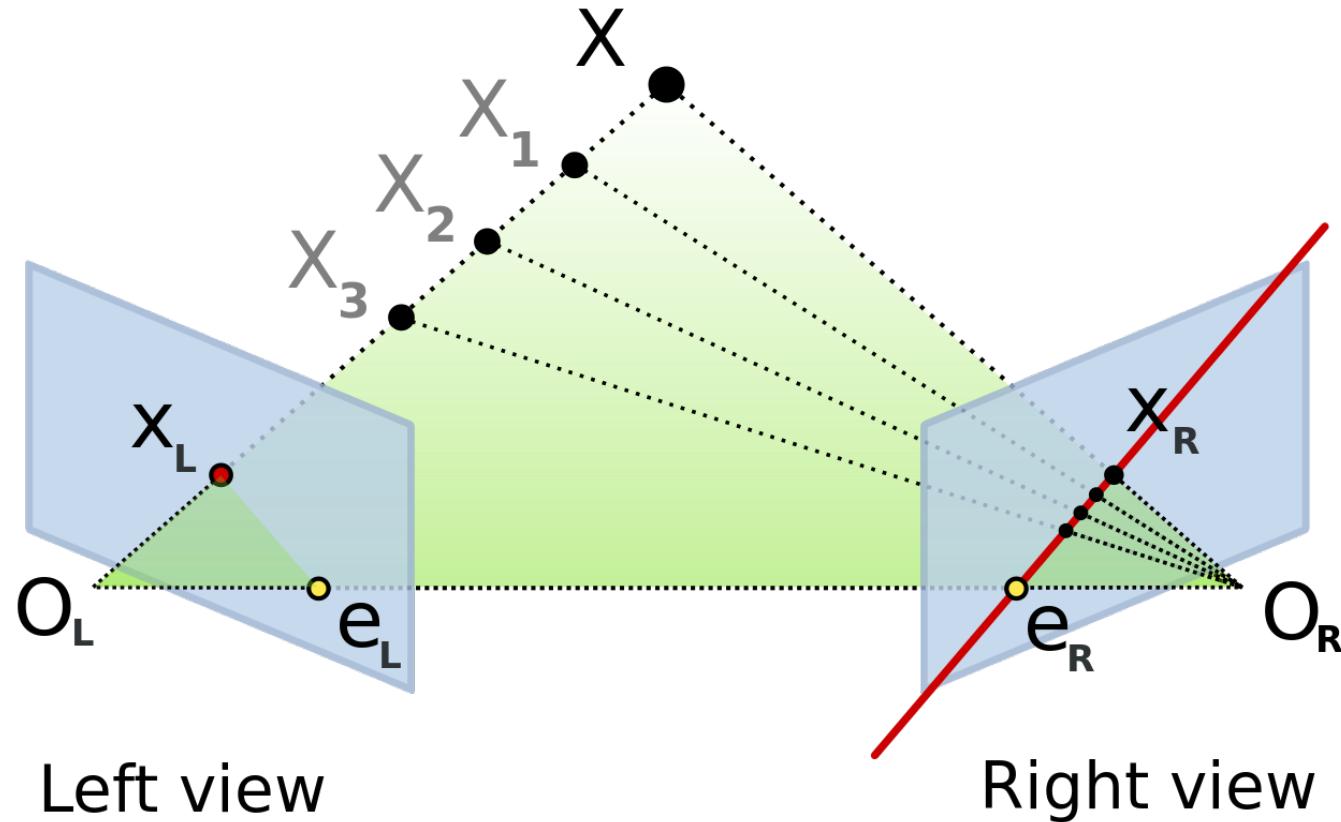


Image Credit: Arne Nordmann 24

Epipolar Lines: The Game

- https://shenshen.mit.edu/cv_book/figures/3d_scene_understanding/epipolar_game_play.png
- Answers: A-2, B-5, C-4, D-6, E-3, F-1

Essential and Fundamental Matrices

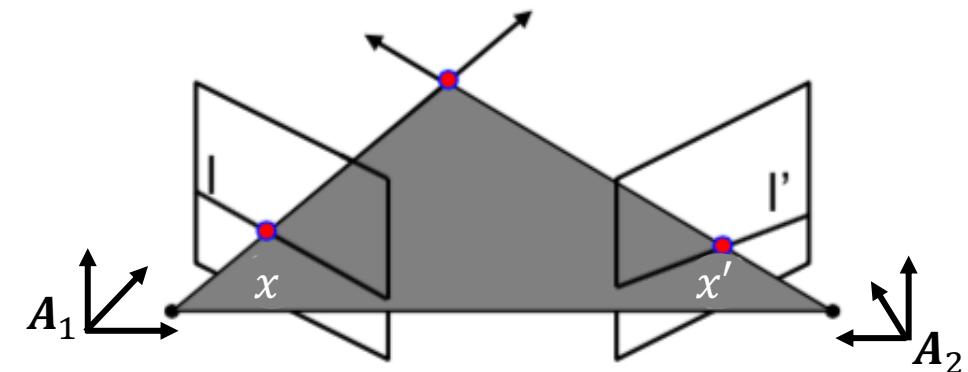
Two-view Geometry

The Essential Matrix E

E encapsulates the relative rotation (R) and translation (t) between two camera views

- Encodes the relationship between two images when the **intrinsics of both cameras are known**
 - If it's the same calibrated camera, moved in space, use E
- A point in one image is mapped to a line in the other image
 - The *epipolar line*

The Essential Matrix E



- The coordinates of x and x' are x_c and x'_c in the camera coordinate frames of camera 1 and camera 2, respectively [how? $x = Kx_c$]
K: intrinsics matrix; x: 2D img coord. xc: 3D cam coord.
- We can transform one coordinate to the other by the relative rotation and translation between the two cameras [why? See next]

$$x'_c = Rx_c + t \quad \text{R: rotation mat.; t: translation mat.}$$

- Take the **cross product** of both sides by t

$$t \times x'_c = t \times Rx_c + t \times t^0$$

$$t \times t = \begin{bmatrix} t_y t_z - t_z t_y \\ t_z t_x - t_x t_z \\ t_x t_y - t_y t_x \end{bmatrix}$$

- Take the **dot product** of both sides by x'_c^T

$$x'^T_c (t \times x'_c) = x'^T_c (t \times Rx_c)$$

0

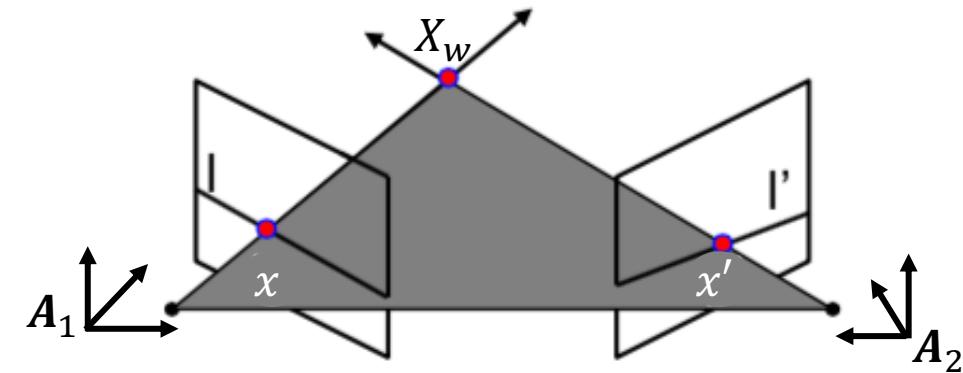
$(t \times x'_c)$ is \perp to both t and x'_c

- So we have:

$$0 = x'^T_c (t \times Rx_c) = x'^T_c [t]_x Rx_c$$

cross prod.

The Essential Matrix E



- We can transform one coordinate to the other by the **relative rotation** and **translation** between the **two cameras**

$$x_c = R_1[I \mid -C_1] \begin{bmatrix} X_w \\ 1 \end{bmatrix} \Rightarrow X_w = R_1^{-1}x_c + C_1 \quad \text{C1, C2: two cam centres}$$

$$x'_c = R_2[I \mid -C_2] \begin{bmatrix} X_w \\ 1 \end{bmatrix}$$

$$x'_c = R_2[I \mid -C_2] \begin{bmatrix} R_1^{-1}x_c + C_1 \\ 1 \end{bmatrix}$$

$$x'_c = R_2[R_1^{-1}x_c + C_1 - C_2]$$

$$\therefore x'_c = Rx_c + t, \text{ where}$$

$$R = R_2R_1^{-1} \text{ and } t = R_2(C_1 - C_2)$$

w.l.o.g. set $R_2 = I$ and $C_2 = 0$, then

$$R = R_1^{-1} \text{ and } t = C_1$$

(since we are free to choose
the world coordinate system)

The Essential Matrix E

$$x_c'^\top [t]_x R x_c = x_c'^\top E x_c = 0 \Rightarrow E \triangleq [t]_x R$$

- $[t]_x$: translation cross-product matrix in $\mathbb{R}^{3 \times 3}$ of rank 2

$$[t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

- R : rotation matrix in $\mathbb{R}^{3 \times 3}$
- x_c and x'_c : homogeneous point vectors in \mathbb{R}^3
- $E \triangleq [t]_x R$ in $\mathbb{R}^{3 \times 3}$ is defined as the **Essential matrix**
 - Relates the **camera coordinate** frames of camera 1 to camera 2

The Essential Matrix E: Properties

- $E \triangleq [t]_x R$
- Properties:
 - 5 DoF
 - Rank 2
 - Singular values: $\sigma_1 = \sigma_2$ and $\sigma_3 = 0$
- Constraints:
 - $\det(E) = 0$
 - $2EE^\top E - \text{tr}(EE^\top)E = 0$
- Estimation:
 - “The Five-Point Algorithm” [Nister 2004; Li & Hartley 2006]
 - “The Eight-Point Algorithm” (i.e., DLT) – see later slides

The Essential Matrix E: Recovering R and t

- $E \triangleq [t]_x R$

1. SVD: $E = U\Sigma V^T$ where $\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

2. Then,

$$\boxed{R = UW^{-1}V^T}$$
$$\boxed{[t]_x = UW\Sigma U^T}$$

where

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

3. To get all 4 solutions, solve for all combinations of $W \in \{W, W^{-1}\}$
4. Select solution where all of the 3D points are in front of both cameras

3. Compute Possible Rotations:

$$\mathbf{R}_1 = \mathbf{U}W\mathbf{V}^T, \quad \mathbf{R}_2 = \mathbf{U}W^T\mathbf{V}^T$$

4. Compute Possible Translations:

The translation vector \mathbf{t} is obtained from \mathbf{U} :

$$\mathbf{t} = \mathbf{U}[:, 2]$$

Here, $\mathbf{U}[:, 2]$ denotes the third column of \mathbf{U} .

5. Possible Solutions:

The possible solutions for (\mathbf{R}, \mathbf{t}) are:

$$(\mathbf{R}_1, \mathbf{t}), (\mathbf{R}_1, -\mathbf{t}), (\mathbf{R}_2, \mathbf{t}), (\mathbf{R}_2, -\mathbf{t})$$

6. Select the Correct Solution:

Use the cheirality condition to determine which combination results in points that are in front of both cameras.

The Fundamental Matrix F

- Encodes the relationship between two images when the intrinsics of both cameras are *not known*
- A point in one image is mapped to a line in the other image
 - The *epipolar line*

The Fundamental Matrix F

- We have:

$$\begin{aligned}x_c'^\top E x_c &= 0 \\x &= K x_c\end{aligned}$$

- So we have:

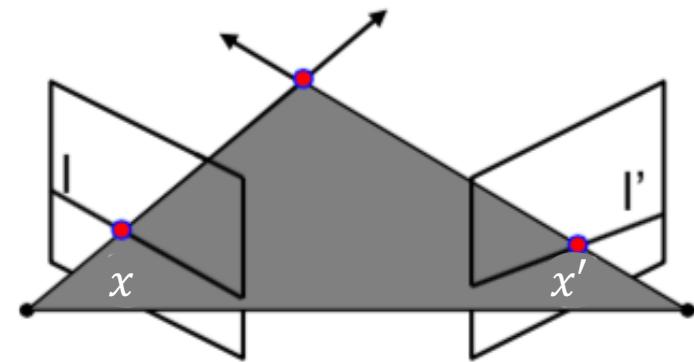
$$\begin{aligned}(K'^{-1}x')^\top E(K^{-1}x) &= 0 \\x'^\top \underbrace{K'^{-\top}EK^{-1}}_F x &= 0\end{aligned}$$

$$\begin{aligned}x'^\top F x &= 0 \\F &\triangleq K'^{-\top}EK^{-1} = K'^{-\top}[t]_x R K^{-1}\end{aligned}$$

K for cam 2; k' for cam 2

- F relates **pixel coordinates** of image 1 to those in image 2

The Fundamental Matrix F



- We have: $x'^\top Fx = 0$
- And $x'^\top l' = 0$
 - Since x' is on the epipolar line l'
- Therefore: $l' = Fx$
- Similarly: $l = F^\top x'$
- That is, for a given point x in the first image, we can find the epipolar line $l' = Fx$ in the second image
 - On which we can search for the corresponding point

2D line representation:

- In 2D, a line is represented as $ax + by + c = 0$ or equivalently,
- $$[a \quad b \quad c] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$
- We use $l = [a, b, c]^\top$
 - Orthogonal to vector (a, b)
 - Dot product of the line vector and any point on the line is 0

The Fundamental Matrix F: Epipole

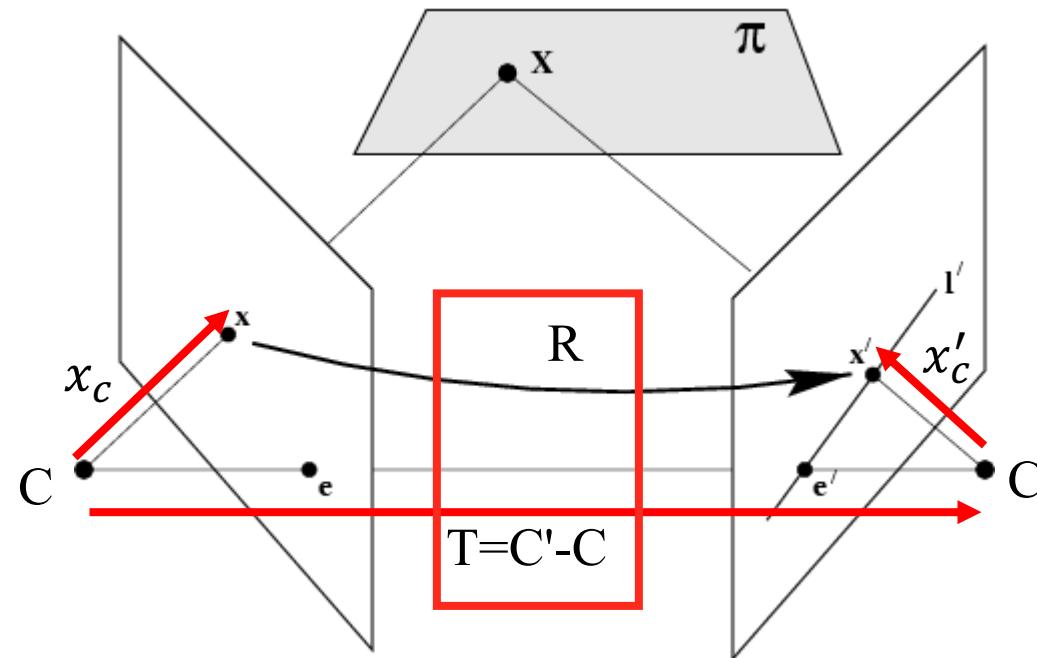
- $x'^\top F x = 0$
- We know that all epipolar lines (regardless of the point x') in the **first image** go through the epipole

- So the epipole e must satisfy

$$F e = 0$$

- How to solve?
 - SVD (again!) or eigendecomposition

F vs E



$$x'^T F x = 0 \quad \text{2D coord. of cam centre on image plane}$$
$$x_c'^T E x_c = 0 \quad \text{3D coord. of cam centre on cam coord.}$$

The Fundamental Matrix F: Properties

- F is the unique 3×3 rank 2 matrix that satisfies $x'^\top F x = 0$ for all $x' \leftrightarrow x$
 1. **Transpose:** if F is fundamental matrix for (x, x') , then F^\top is the fundamental matrix for (x', x)
 2. **Epipolar lines:** F maps from a point x to a line $l' = Fx$
 3. **Epipoles:** lie on all epipolar lines, thus $x'^\top F x = 0 \forall x \Rightarrow Fe = 0$

Direct Linear Transformation (DLT) Algorithm for Fundamental Matrix Estimation

- AKA “**The 8-Point Algorithm**”
- The Fundamental matrix F is defined by $x'^\top F x = 0$ for any pair of matches x and x' in two images
- Let $x = (u, v, 1)^\top$ and $x' = (u', v', 1)^\top$,
then each match gives a linear equation

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

where

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

The 8-Point Algorithm

$$\begin{bmatrix} u_1 u'_1 & v_1 u'_1 & u'_1 & u_1 v'_1 & v_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\ u_2 u'_2 & v_2 u'_2 & u'_2 & u_2 v'_2 & v_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\ \vdots & \vdots \\ u_n u'_n & v_n u'_n & u'_n & u_n v'_n & v_n v'_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

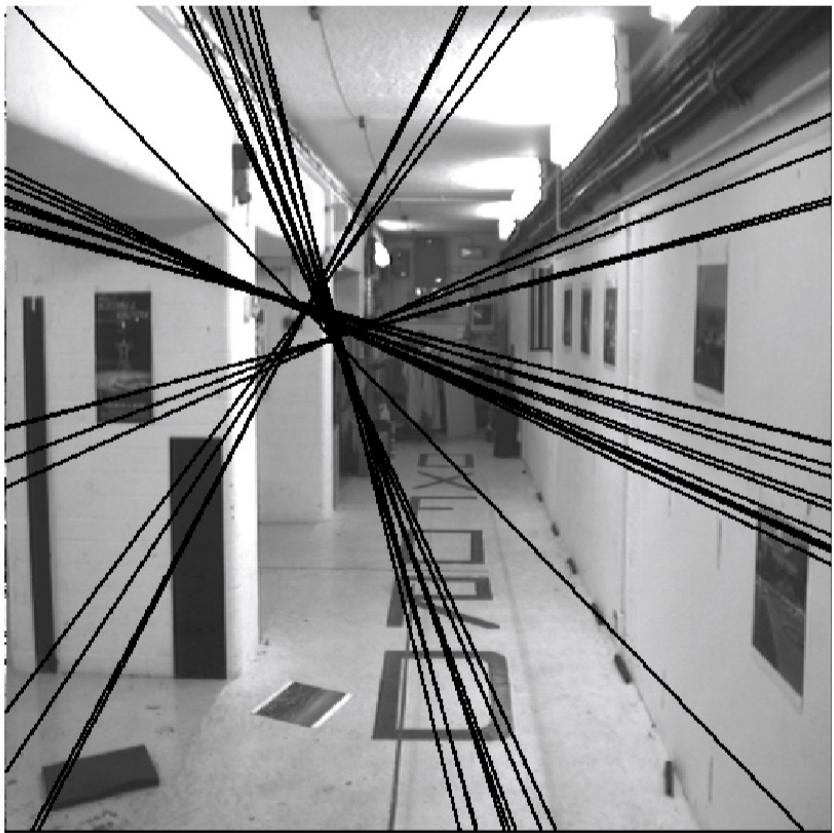
- Instead of solving $Af = 0$, we seek f to minimise $\|Af\|$: the least right-singular vector of A (the least eigenvector of $A^T A$)

The 8-Point Algorithm: Rank 2 Constraint

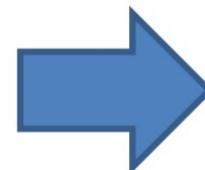
- But $\text{rank}(F) = 2$
- Therefore, replace **current estimate F'** with
$$\begin{aligned} F &= \underset{F'}{\text{minimise}} \|F - F'\| \\ &\text{subject to } \det(F) = 0 \end{aligned}$$
- How? **SVD**, again! Decompose $F' = U\Sigma V^\top$ and set $F = U\Sigma' V^\top$ where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \text{ and } \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The 8-Point Algorithm: Rank 2 Constraint



Impose
 $\text{rank}(F) = 2$



The 8-Point Algorithm

- **Advantages:**
 - Linear
 - Easy to implement
 - Fast
 - Can extract R and t decomposition
- **Disadvantages:**
 - Susceptible to noise
 - Optimises a non-physical algebraic quantity

The 8-Point Algorithm: Normalisation

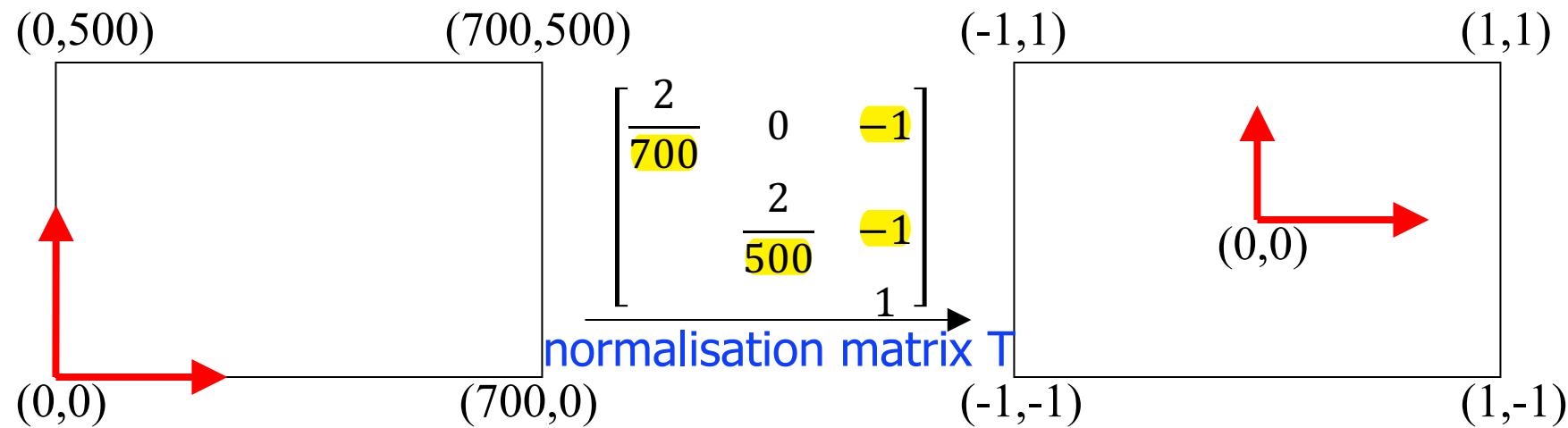
$$\begin{bmatrix} u_1 u'_1 & v_1 u'_1 & u'_1 & u_1 v'_1 & v_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\ u_2 u'_2 & v_2 u'_2 & u'_2 & u_2 v'_2 & v_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\ \vdots & \vdots \\ u_n u'_n & v_n u'_n & u'_n & u_n v'_n & v_n v'_n & v'_n & u_n & v_n & 1 \\ \sim 10000 & \sim 10000 & \sim 100 & \sim 10000 & \sim 10000 & \sim 100 & \sim 100 & \sim 100 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$



Orders of magnitude difference
between columns of data matrix
→ least-squares yields poor results

The Normalised 8-Point Algorithm

- Normalised least squares yields good results
- Transform image to $\sim[-1,1] \times [-1,1]$



The Normalised 8-Point Algorithm

1. Transform input by $\hat{x}_i = Tx_i, \hat{x}'_i = Tx'_i$ normalise input correspondents
2. Call the 8-point algorithm on \hat{x}_i, \hat{x}'_i to obtain \hat{F}
3. Then, $F = T'^\top \hat{F} T$ demormalise
 - Since $x'^\top F x = 0 \Rightarrow \hat{x}'^\top T'^\top F T \hat{x}' = 0 \Rightarrow \boxed{\hat{F} = T'^\top F T}$
 - Example normalisation matrices:

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

The Normalised 8-Point Algorithm

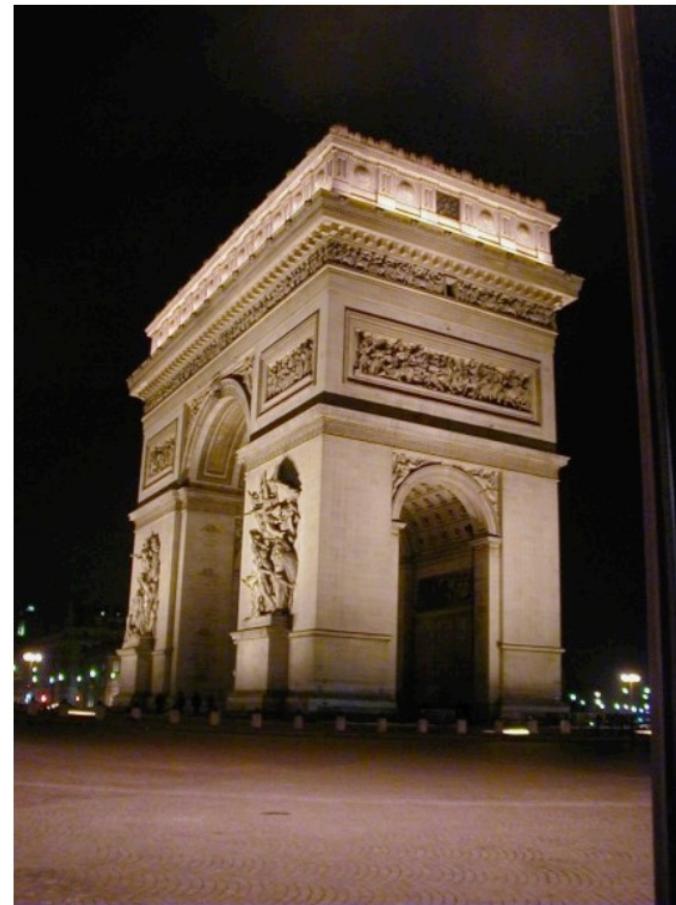
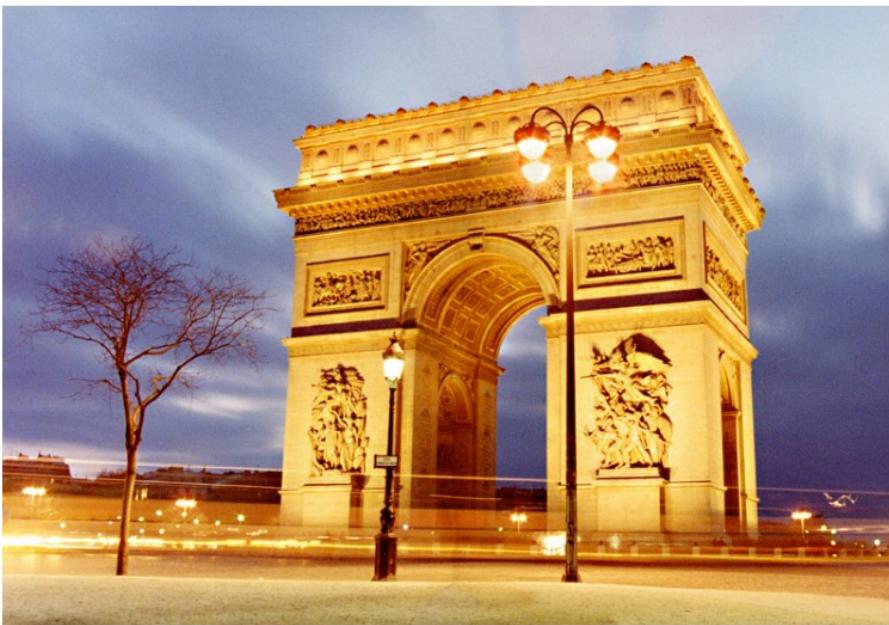
- Example normalisation matrices:

$$x'_i \leftarrow x_i - \bar{x}$$

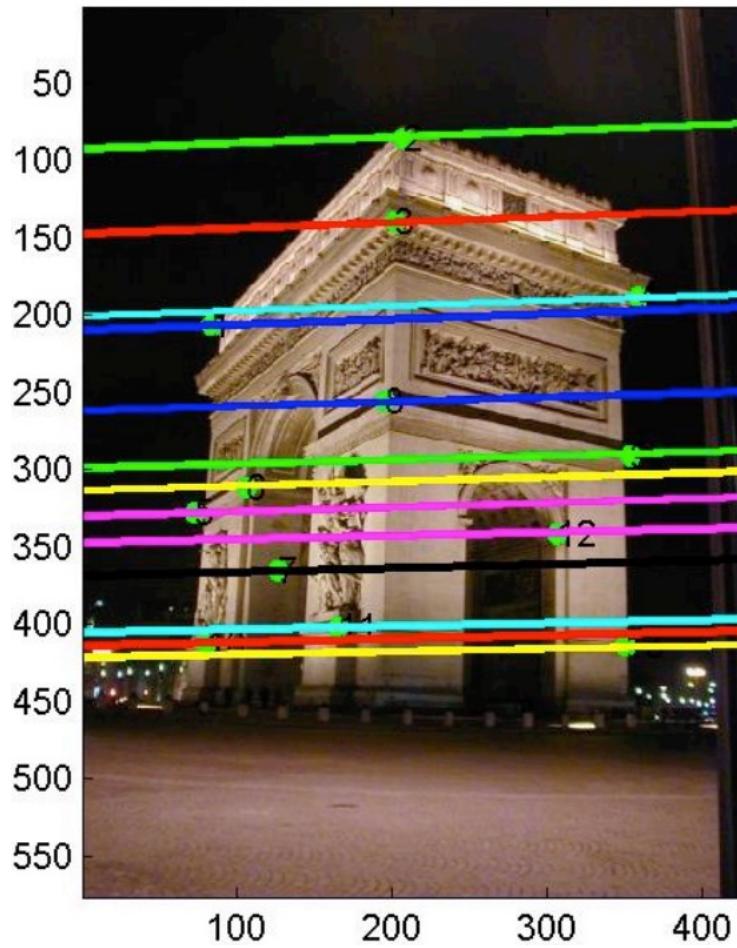
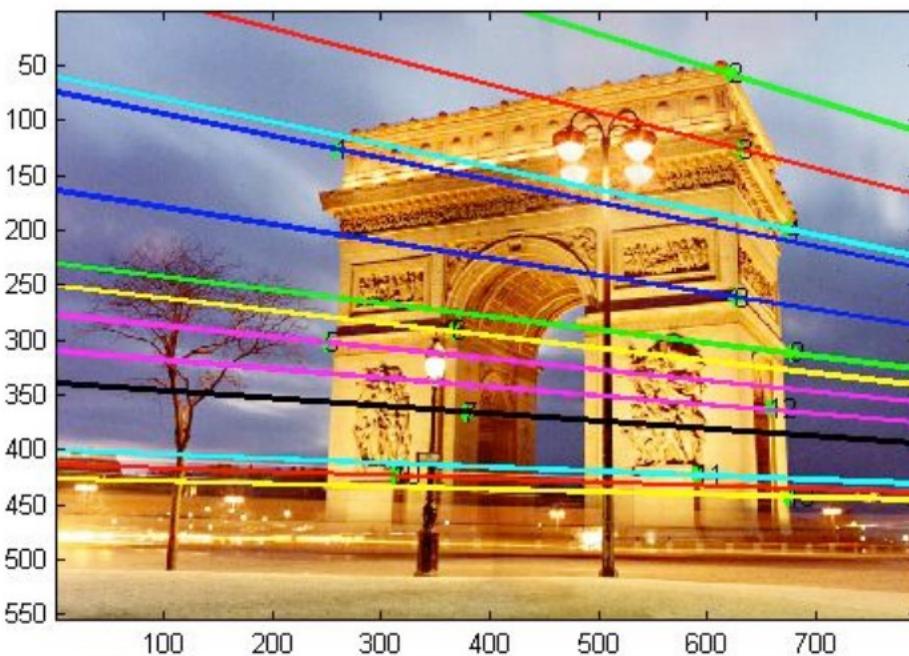
$$s \leftarrow \frac{\sqrt{2}}{\frac{1}{n} \sum_{i=1}^n \|x'_i\|}$$

$$T = \begin{bmatrix} s & 0 & -s\bar{x}_1 \\ 0 & s & -s\bar{x}_2 \\ 0 & 0 & 1 \end{bmatrix}$$

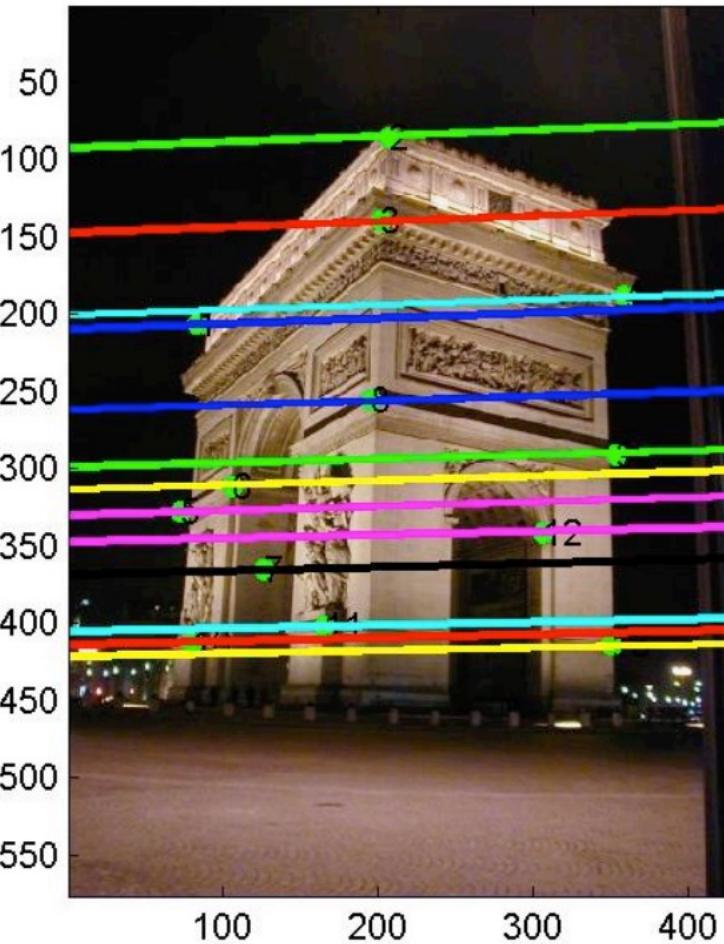
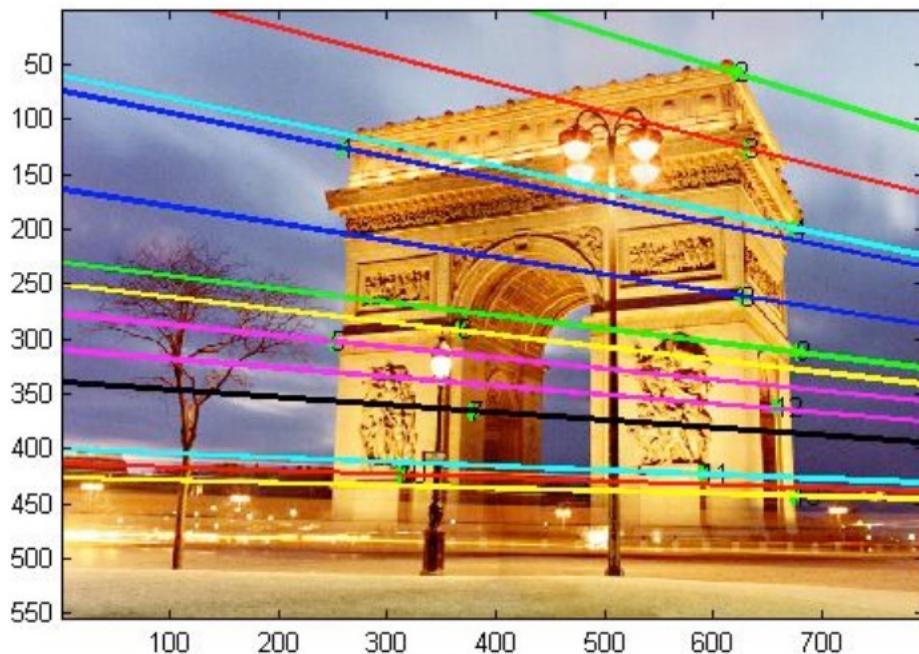
Example



Example



Example

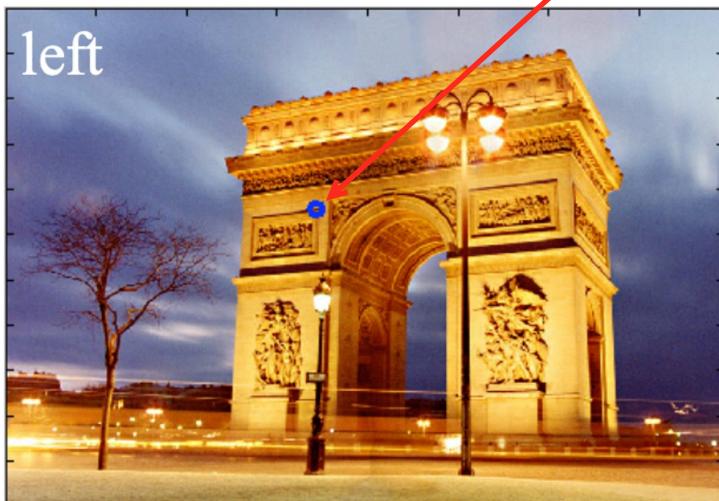


$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

Example

$$\begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix} \begin{pmatrix} 343.53 \\ 221.70 \\ 1.0 \end{pmatrix}$$

$I' = Fx$



$x = 343.5300 \ y = 221.7005$

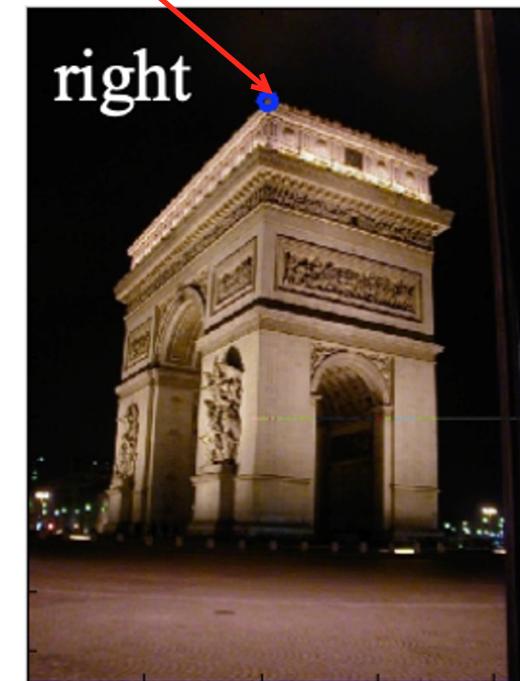
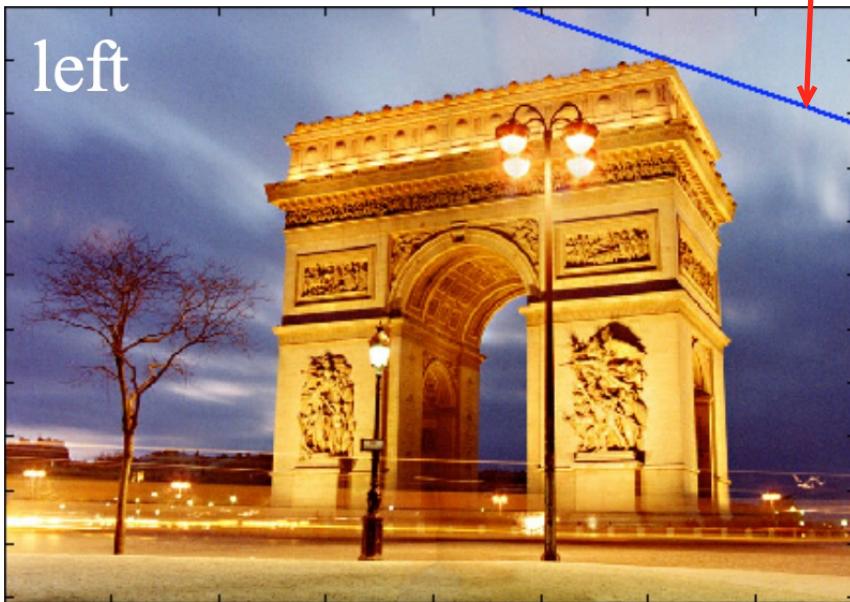
$$\begin{bmatrix} 1.3299 \\ 45.0195 \\ -1.1942 \times 10^4 \end{bmatrix}$$



Example

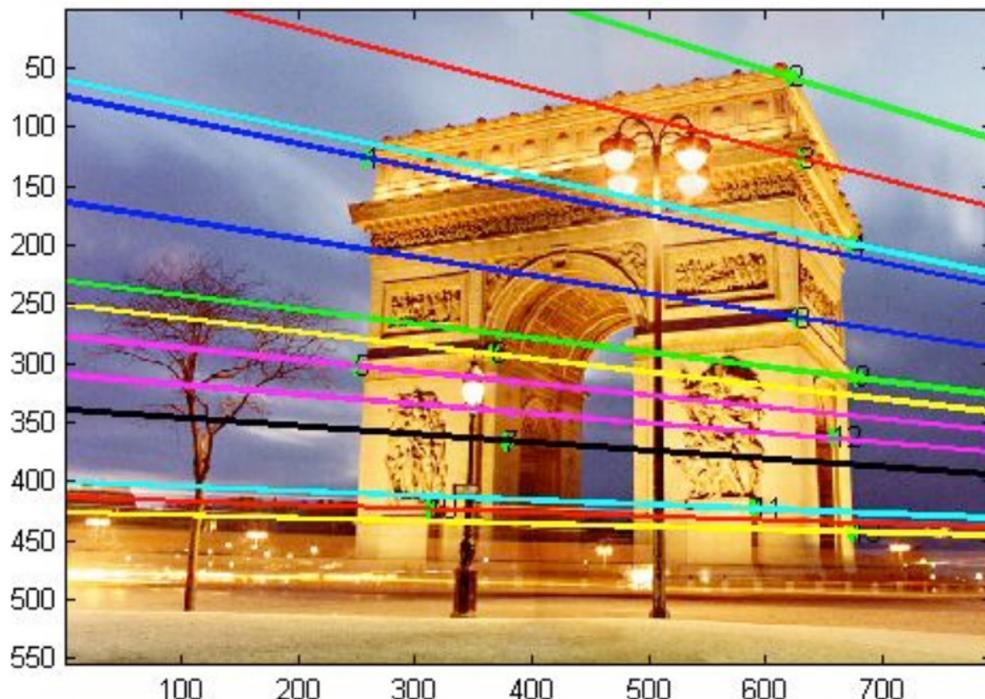
$$(205.5526 \quad 80.5 \quad 1.0) \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

$l = F^T x'$



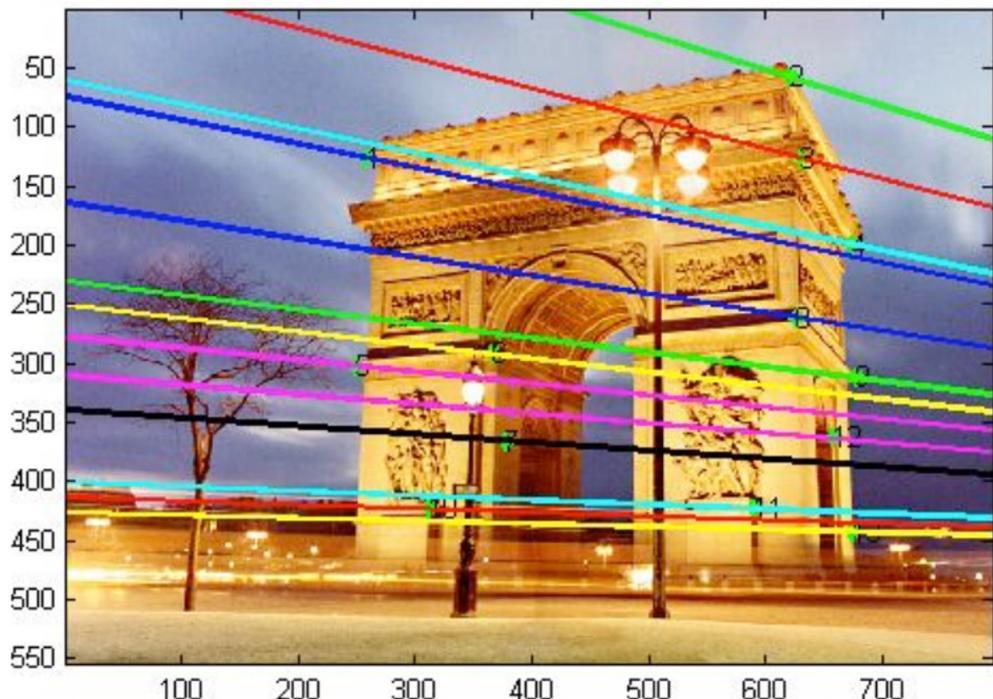
$x = 205.5526 \quad y = 80.5000$

Example



- Where is the epipole?
- $Fe = 0$
- Vector in the right null space of matrix F
- However, due to noise, F may not be singular. So instead, next best thing is eigenvector associated with smallest eigenvalue of $F^T F$

Example



- Eigendecomposition:

- $[u, d] = \text{eigs}(F' * F)$

$$\begin{array}{ccc} u = & & d = 1.0e8^* \\ \begin{matrix} -0.0013 & 0.2586 & -0.9660 \\ 0.0029 & -0.9660 & -0.2586 \\ 1.0000 & 0.0032 & -0.0005 \end{matrix} & \begin{matrix} -1.0000 & 0 & 0 \\ 0 & -0.0000 & 0 \\ 0 & 0 & -0.0000 \end{matrix} & \end{array}$$

- Eigenvector associated with smallest eigenvalue:

- $e = u(:, 3)$
- $e = (-0.9660 \quad -0.2586 \quad -0.0005)$

- $e = e/e(3)$ to get pixel cords
- $e = (1861.02 \quad 498.21 \quad 1)$

homo
to
2D img
coord.

Use RANSAC to Handle Outliers

1. **Repeat:**
 1. select minimal sample (8 pixel matches)
 2. compute solution(s) for F
 3. count the number of inliers

until $\Gamma(\# \text{inliers}, \# \text{samples}) > 95\%$ or bored

repeat until find a good sample ->
a good solution F
2. Recompute F using all inliers
use the solution F from the last step to
compute all inliers

The Fundamental Matrix F: Recovering P & P'

intrinsic matrix of both cams

- Since we can set the world coordinate system arbitrarily, set

$$\begin{aligned}P &= [I \mid 0] \\P' &= [[e']]_x F \mid e'\end{aligned}$$

where epipole can be found by solving

$$F^T e' = 0$$

via SVD/eigendecomposition
[HZ p. 256]

Essential & Fundamental Matrices: Summary

- Algebraic representations of epipolar geometry:
 - Projection matrices (given intrinsics + extrinsics)
 - **Essential matrix** (given intrinsics): $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$
 - **Fundamental matrix**: $\mathbf{F} = \mathbf{K}'^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = [\mathbf{e}']_{\times} \mathbf{K}' \mathbf{R} \mathbf{K}^{-1}$ [HZ p.244]
 - Using identity $[\mathbf{v}]_{\times} \mathbf{M} = \mathbf{M}^{-T} [\mathbf{M}^{-1} \mathbf{v}]_{\times}$ and $\mathbf{e}' = \mathbf{K}' \mathbf{t}$ (image of camera centre 1)
- Epipolar constraint for corresponding points $\{\mathbf{x}, \mathbf{x}'\}$:
$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$$
 - \mathbf{F} is rank 2 and is known only up to scale \rightarrow 7 DoF
 - What is $\mathbf{F} \mathbf{x}$? [the epipolar lines on the other image](#)
- **Estimation**: DLT (again!); assemble a matrix \mathbf{A} , compute the SVD, enforce rank 2 (with another SVD); or use a nonlinear solver

Next Lecture

- Two-view geometry: Triangulation
- Two-view geometry: Stereo