Stereo and Geometric Stereo

CS635

Daniel G. Aliaga
Department of Computer Science
Purdue University

- Thanks to S. Narasimhan @ CMU
  for some of the slides

Problem Statement

- How to create (realistic) 3D models of existing objects and scenes in the world?
  - Object vs. scene
  - Shape vs. color vs. material properties
  - Automatic vs. manual
  - And many more factors…

3D Shape Reconstruction from Photos

- http://carlos-hernandez.org/cvpr2010/




Fundamental Approaches

- Manual modeling
  - CAD, Sketchup, 3D Studio Max, MS Paint
- Point Clouds
  - LIDAR, Laser, Kinect
- Photographs
  - “photogrammetry and remote sensing”
  - Single Photograph
  - Stereo Reconstruction (2 photos)
  - Multi-view Reconstruction
    - narrow (video?) or wide baseline

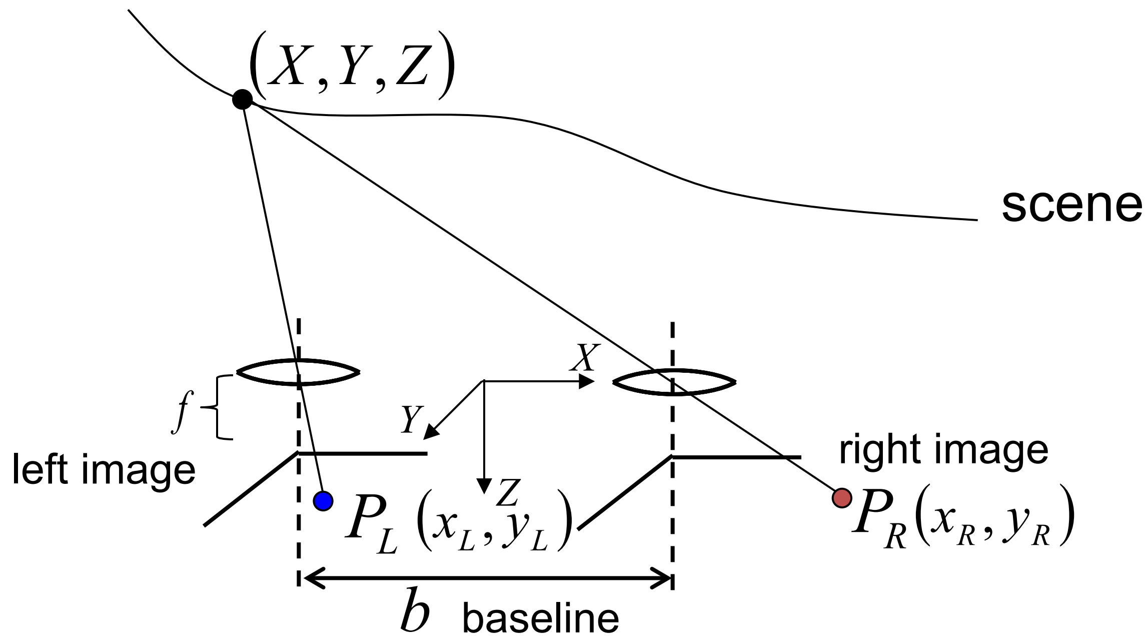
Fundamental Approaches

- Manual modeling
  - CAD, Sketchup, 3D Studio Max, MS Paint
- Point Clouds
  - LIDAR, Laser, Kinect
- Photographs
  - "photogrammetry and remote sensing"
  - Single Photograph
  - **Stereo Reconstruction (2 photos)**
  - Multi-view Reconstruction
    - narrow (video?) or wide baseline

Definitions

- Camera geometry (=*motion*)
  - Given corresponded points on ≥2 views, what are the poses of the cameras?
- Correspondence geometry (=*correspondence*)
  - Given a point in one view, what are the constraints of its position in another view?
- Scene geometry (=*structure*)
  - Given corresponded points on ≥2 views and the camera poses, what is the 3D location of the points?

Stereo Rig

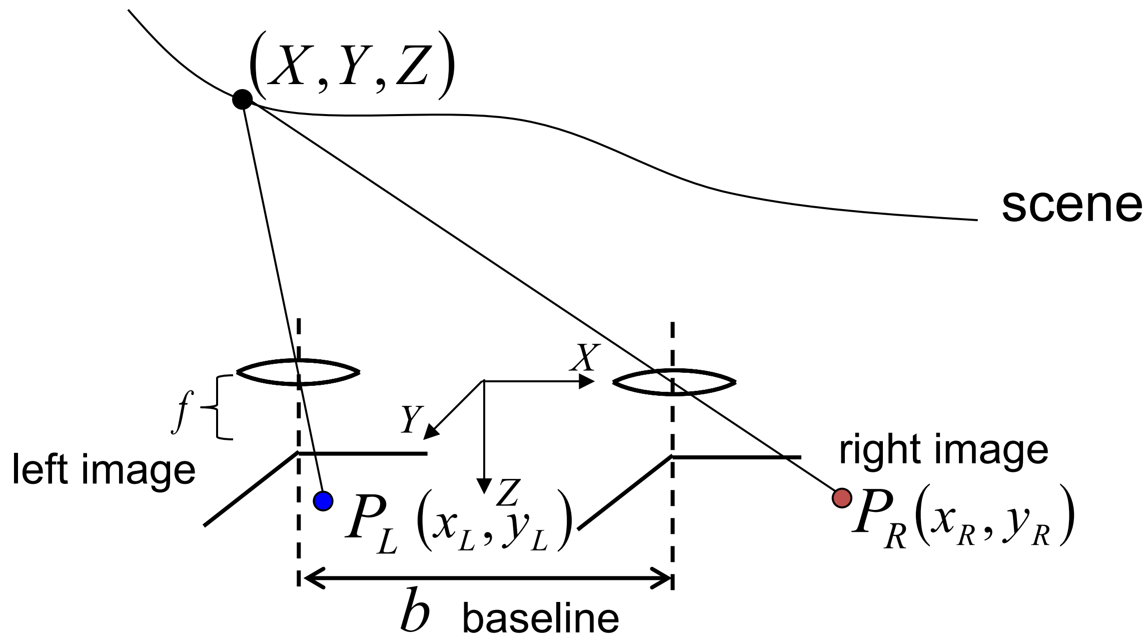


Assume that we know P_L corresponds to P_R

Using perspective projection (defined using coordinate system shown)

What is relationship between x_L and X?

Stereo Rig

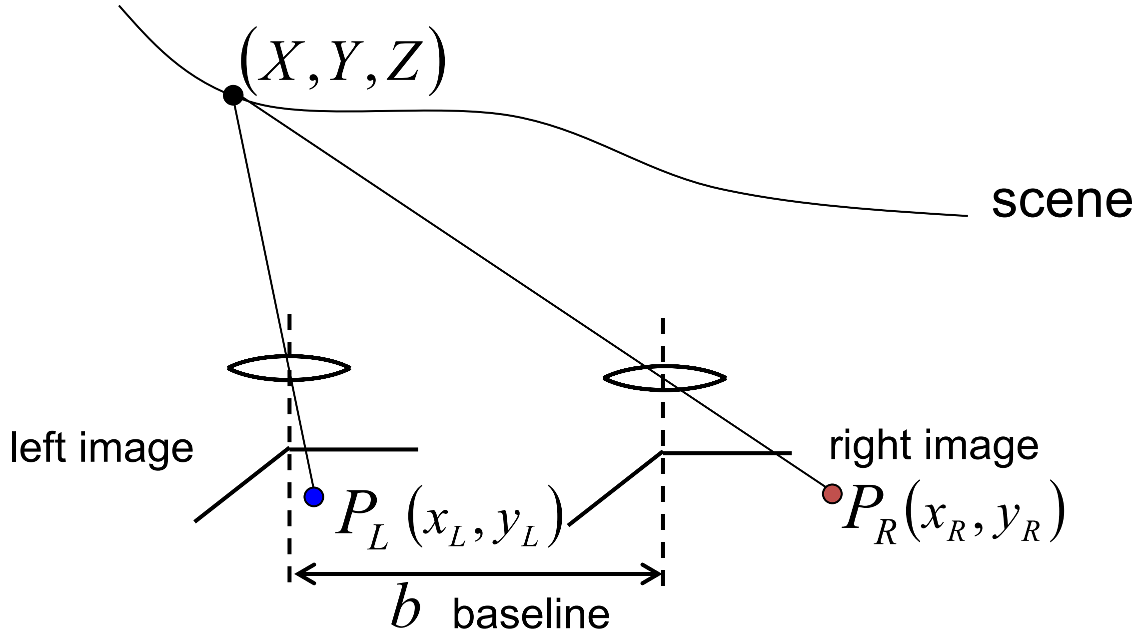


Assume that we know P_L corresponds to P_R

Using perspective projection (defined using coordinate system shown)

$$\Rightarrow \quad \frac{x_L}{f} = \frac{X + b/2}{Z} \qquad \frac{x_R}{f} = \frac{X - b/2}{Z} \qquad \frac{y_L}{f} = \frac{y_R}{f} = \frac{Y}{Z}$$

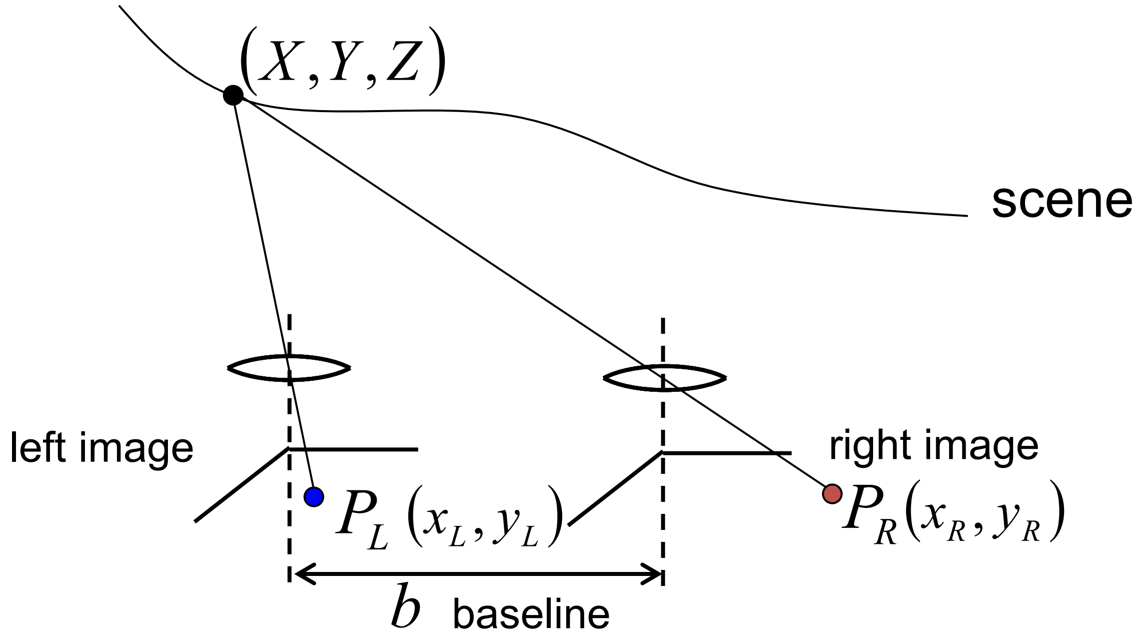
Stereo Rig



$$\frac{x_L}{f} = \frac{X + b/2}{Z} \qquad \frac{x_R}{f} = \frac{X - b/2}{Z} \qquad \frac{y_L}{f} = \frac{y_R}{f} = \frac{Y}{Z}$$

$$\Rightarrow \quad X = \frac{b(x_L + x_R)}{2(x_L - x_R)} \qquad Y = \frac{b(y_L + y_R)}{2(x_L - x_R)} \qquad Z = \frac{bf}{(x_L - x_R)}$$

Stereo: Disparity and Depth



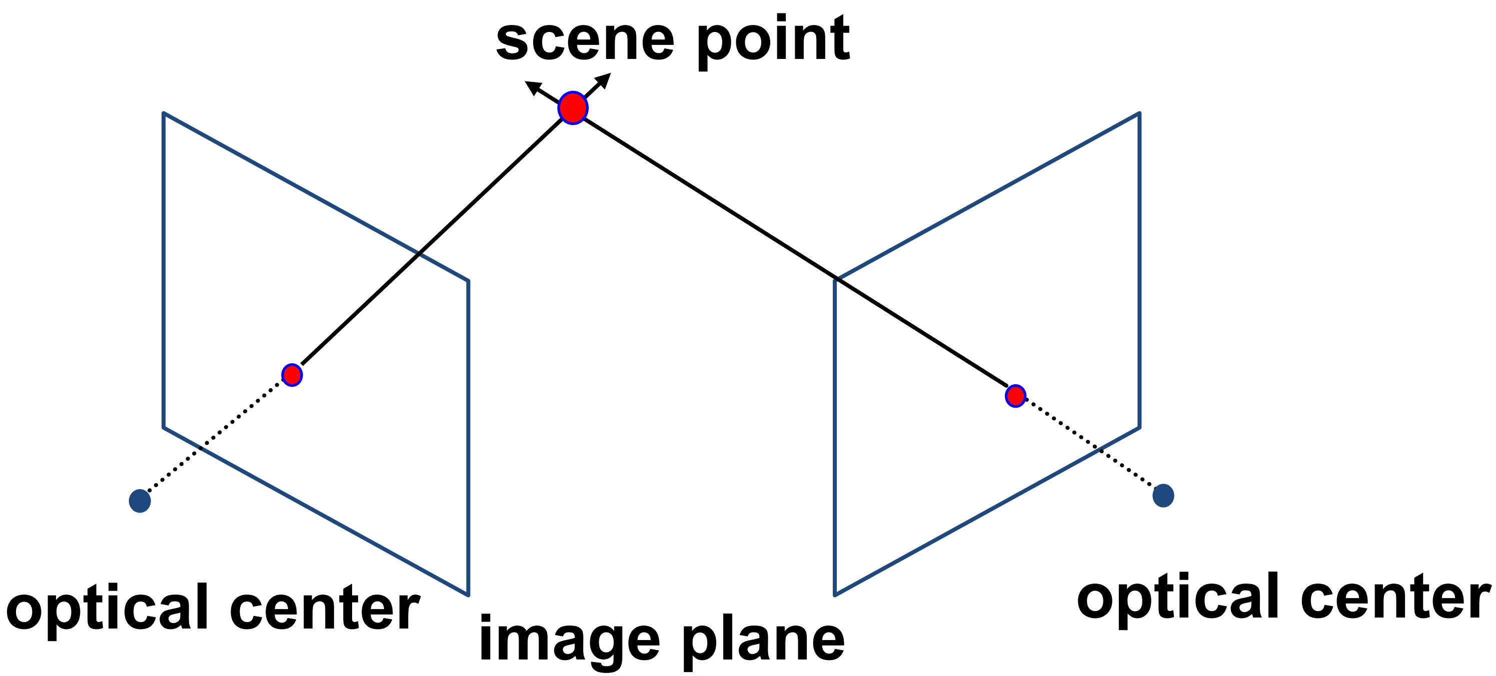
$$\frac{x_L}{f} = \frac{X + b/2}{Z} \qquad \frac{x_R}{f} = \frac{X - b/2}{Z} \qquad \frac{y_L}{f} = \frac{y_R}{f} = \frac{Y}{Z}$$

$$X = \frac{b(x_L + x_R)}{2(x_L - x_R)} \qquad Y = \frac{b(y_L + y_R)}{2(x_L - x_R)} \qquad Z = \frac{bf}{(x_L - x_R)}$$

$\Rightarrow\ d = x_L - x_R$ is the **disparity** between corresponding left and right image points

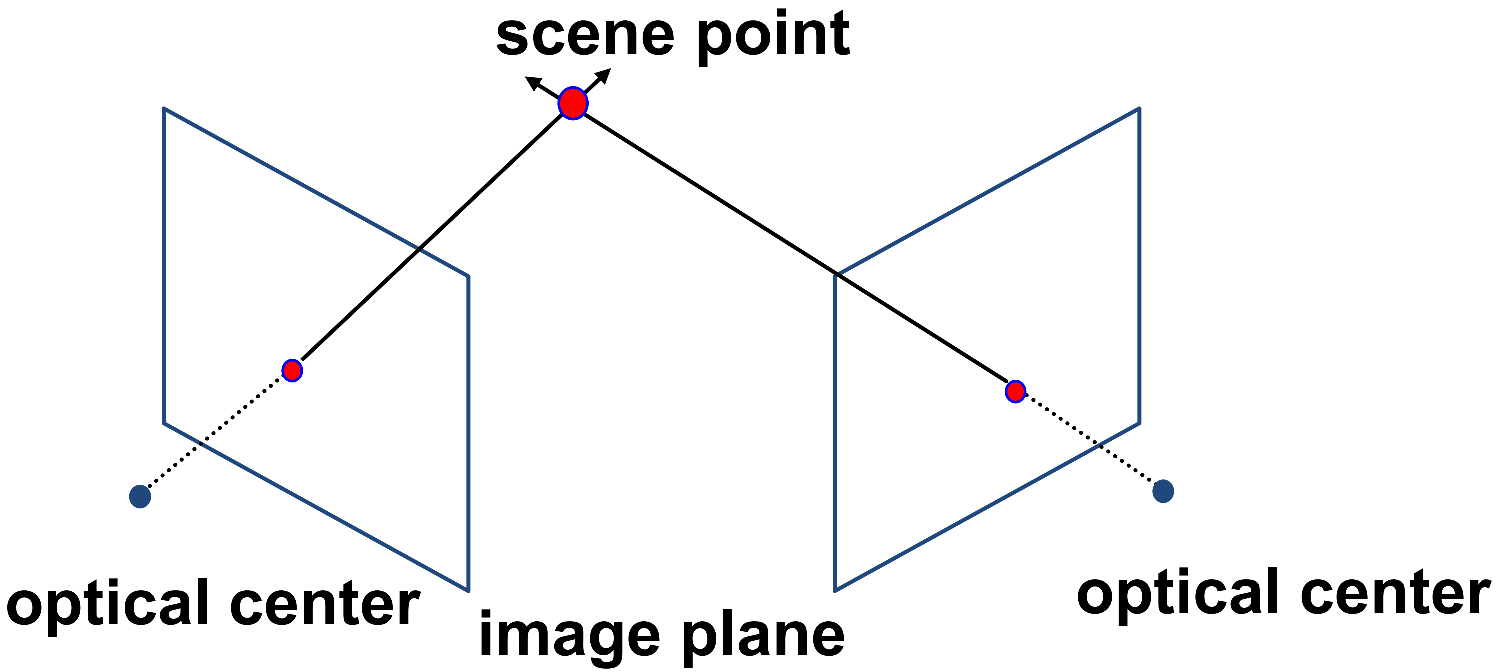
- inversely proportional to depth
- disparity increases with baseline **b**

Stereo: Ray Triangulation



(Ray) Triangulation: compute reconstruction as intersection of two rays

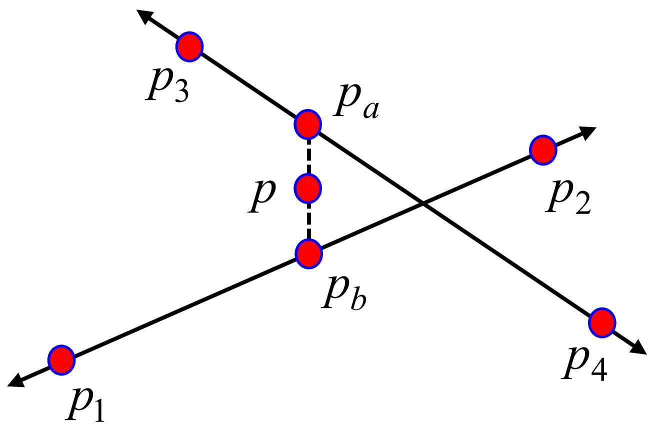
Stereo: Ray Triangulation



Do two lines intersect in 3D?

If so, how do you compute their intersection?

Stereo: Ray Triangulation



Equations for the intersection:

$$(p_1 - p_2)\cdot(p_a - p_b) = 0$$

$$(p_3 - p_4)\cdot(p_a - p_b) = 0$$

$$p_b = p_1 + s(p_2 - p_1)$$

$$p_a = p_3 + t(p_4 - p_3)$$

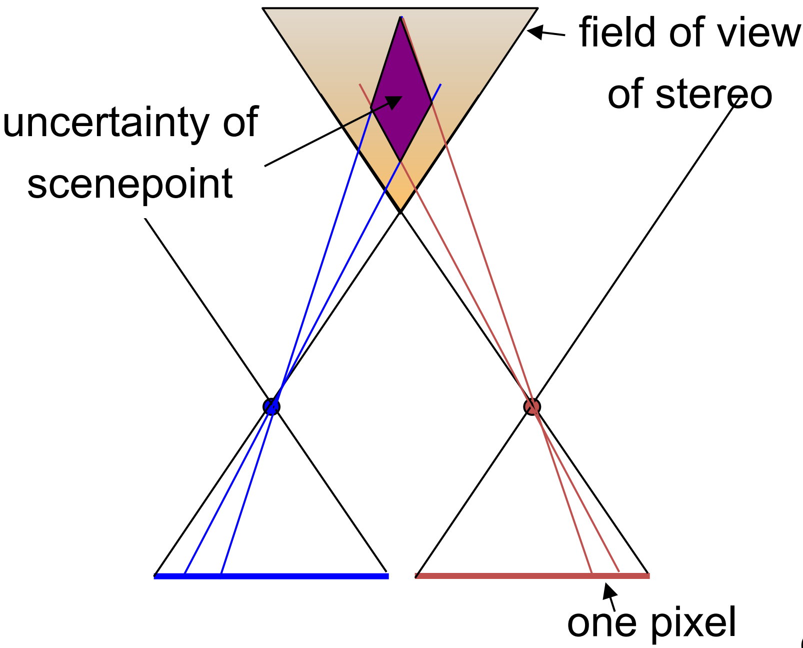
Solve for *s* and *t*, compute *p*:

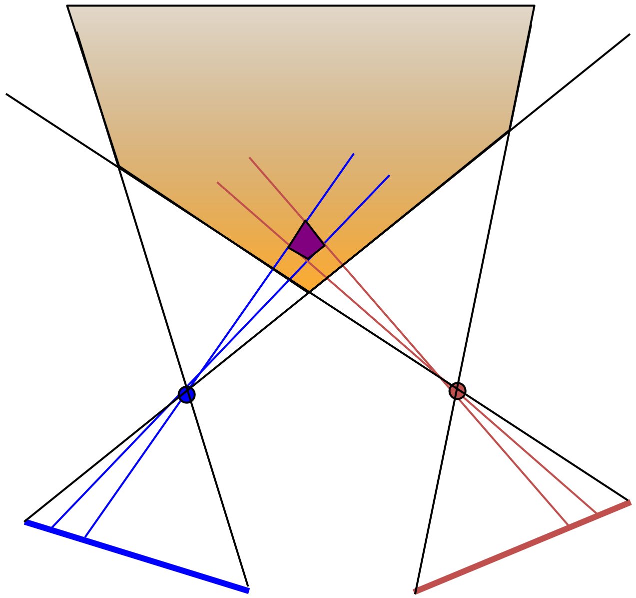
$$s = \ldots$$

$$t = \ldots$$

$$p = 0.5(p_a + p_b)$$

Stereo: Vergence

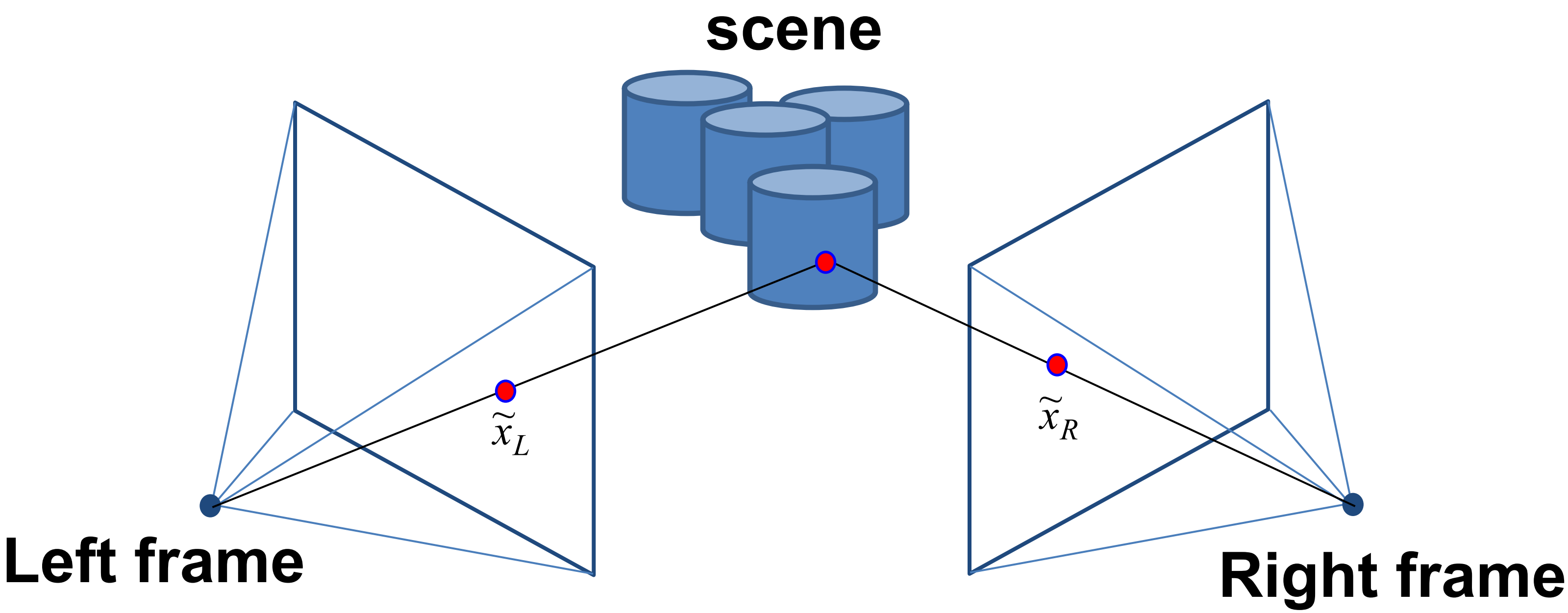




optical axes of the two cameras need not be parallel

1. Field of view **decreases** with **increase** in baseline and vergence
2. Accuracy **increases** with **increase** in baseline and vergence

Camera Geometry



- We need to transform "left frame" to "right frame" – includes a rotation and translation:

$$\widetilde{x}_R = R\,\widetilde{x}_L + t_{LR}$$

Camera Geometry

- In matrix notation, we can write $\tilde{x}_R = R\,\tilde{x}_L + t_{LR}$ as:

$$\tilde{x}_L = \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \quad \tilde{x}_R = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad t_{LR} = \begin{bmatrix} r_{14} \\ r_{24} \\ r_{34} \end{bmatrix}$$

Camera Geometry

- In matrix notation, we can write $\widetilde{x}_R = R\,\widetilde{x}_L + t_{LR}$ as:

$$\boxed{\begin{aligned} r_{11}\, x_L + r_{12}\, y_L + r_{13}\, z_L + r_{14} &= x_R \\ r_{21}\, x_L + r_{22}\, y_L + r_{23}\, z_L + r_{24} &= y_R \\ r_{31}\, x_L + r_{32}\, y_L + r_{33}\, z_L + r_{34} &= z_R \end{aligned}}$$

Camera Geometry: Orthonormality Constraints

$$R^T R = I$$

(a) Rows of R are perpendicular vectors

$$r_{11}\, r_{21} + r_{12}\, r_{22} + r_{13}\, r_{23} = 0$$

$$r_{21}\, r_{31} + r_{22}\, r_{32} + r_{23}\, r_{33} = 0$$

$$r_{11}\, r_{31} + r_{12}\, r_{32} + r_{13}\, r_{33} = 0$$

(b) Each row of R is a unit vector

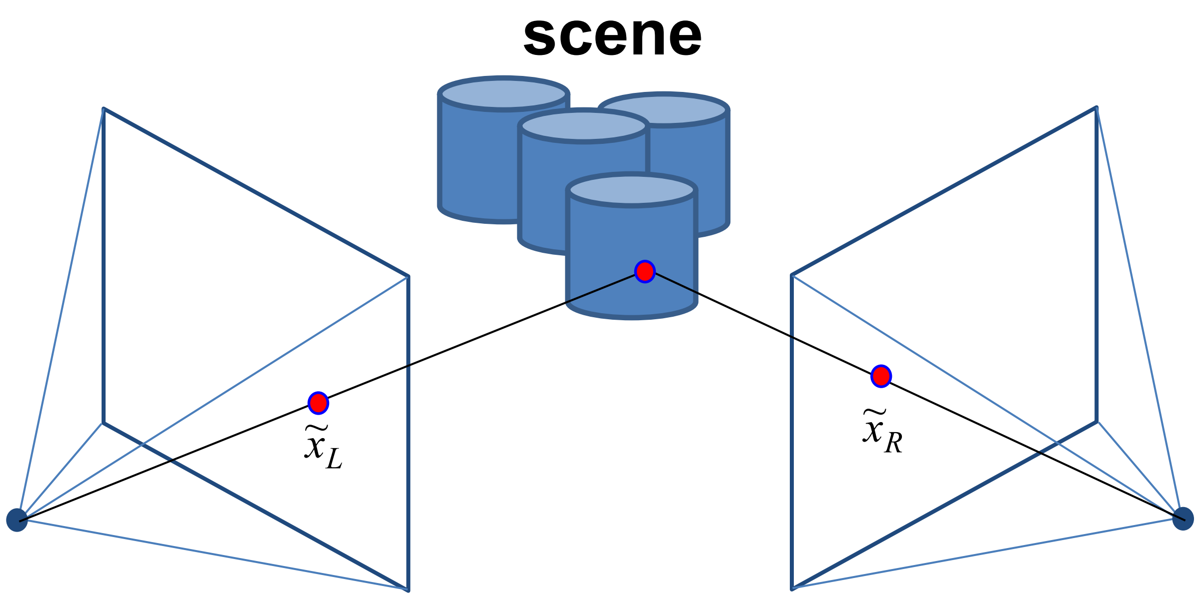
$$r_{11}{}^2 + r_{12}{}^2 + r_{13}{}^2 = 1$$

$$r_{21}{}^2 + r_{22}{}^2 + r_{23}{}^2 = 1$$

$$r_{31}{}^2 + r_{32}{}^2 + r_{33}{}^2 = 1$$

NOTE: Constraints are NON-LINEAR!

Camera Geometry: Problem Definition

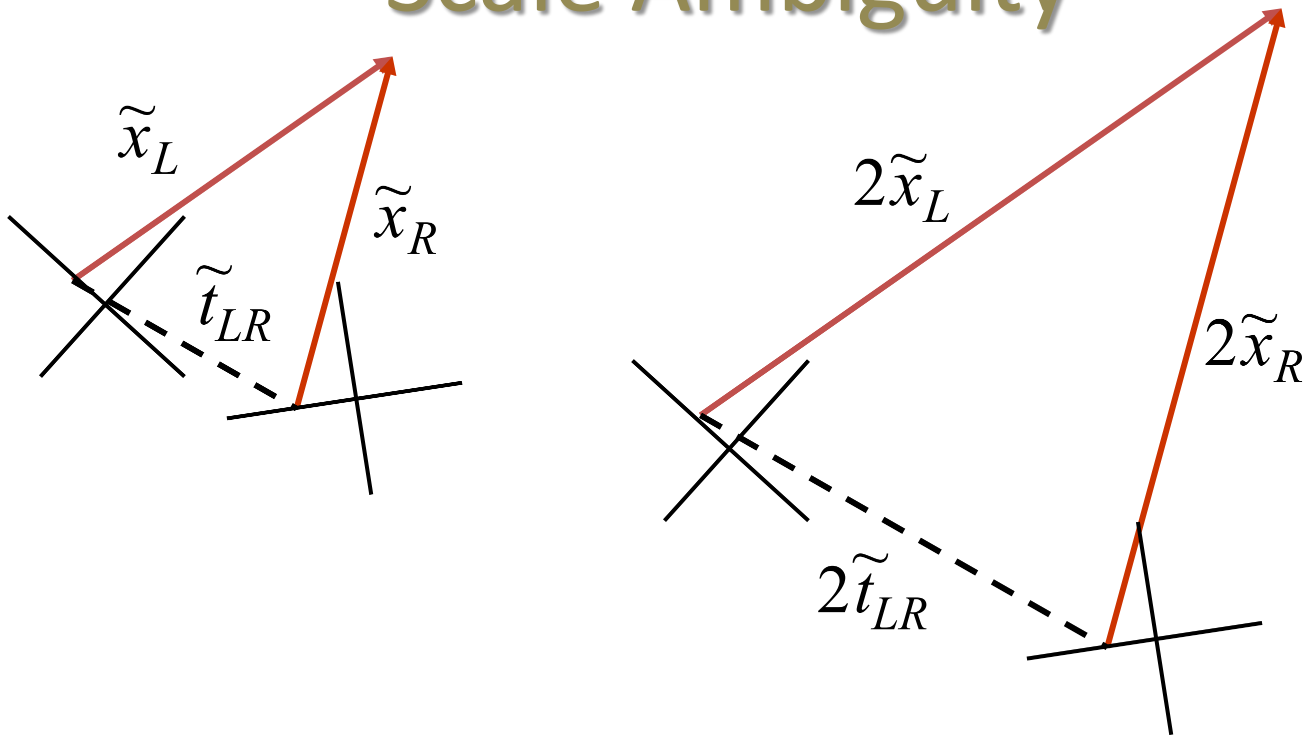


Problem:

Given $\tilde{x}_L$ $\tilde{x}_R$'s

Find R t_{LR} $\Rightarrow$ $(r_{11}, r_{12}, \ldots, r_{34})$ subject to (nonlinear) constraints

Camera Geometry: Scale Ambiguity



Problem: same image coords can be generated by doubling $\tilde{x}_L$ $\tilde{x}_R$ $\tilde{t}_{LR}$

thus, we can find $\tilde{t}_{LR}$ only up to a scale factor!

Partial

Solution: fix scale by using constraint: $\tilde{t}_{LR} \cdot \tilde{t}_{LR} = 1$ (1 additional equation)

Camera Geometry: How many scene points are needed?

Each scene point gives 3 equations:

$$r_{11}\, x_L + r_{12}\, y_L + r_{13}\, z_L + r_{14} = x_R$$
$$r_{21}\, x_L + r_{22}\, y_L + r_{23}\, z_L + r_{24} = y_R$$
$$r_{31}\, x_L + r_{32}\, y_L + r_{33}\, z_L + r_{34} = z_R$$

and 6+1 additional equations from orthonormality of rotation matrix constraints and scale constraint.

Thus, for *n* scene points, we have ($3n + 6 + 1$) equations and 12 unknowns

What is the minimum value for *n*?

Camera Geometry: Solving an Over-determined System

- Generally, more than 3 points are used to find the 12 unknowns

- Formulate error for scene point *i* as:

$$e_i = (R\,\tilde{x}_L + t_{LR}) - \tilde{x}_R$$

- Find $R \;\&\; t_{LR}$ that minimize:

$$E = \sum_{i=1}^{N} |\, e_i \,|^2 \;+\; [\lambda_1\,(R^T R - I) + \lambda_2(t_{LR} \cdot t_{LR} - 1)]$$

Camera Geometry: A Linear Estimation

Assume a near correct rotation is known. Then an orthogonal rotation matrix looks like:

$$R = \begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix}$$

where $\boldsymbol{\omega}$ is the 3D rotation axis and its length is the amount by which to rotate

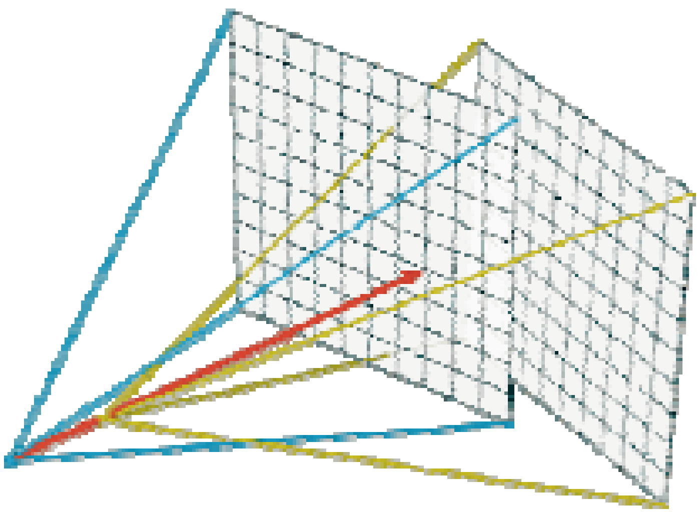
- Using this matrix, iteratively and linearly solve for ω's and t_{LR}:

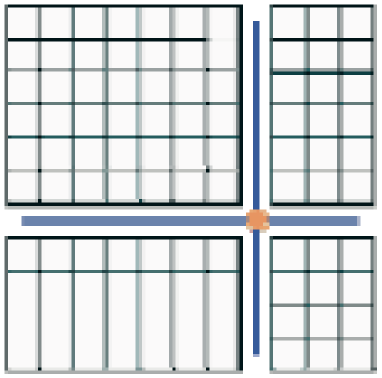
$$(R\,\tilde{x}_L + t_{LR}) - \tilde{x}_R = 0$$

- Limitations:
  - 1. ignores normality/scale (fix by re-scaling each iteration)
  - 2. assumes good initial guess
- How many equations/scene-points are needed?
  - 6 unknowns, 3 equations per scene point, so ≥ 2 points

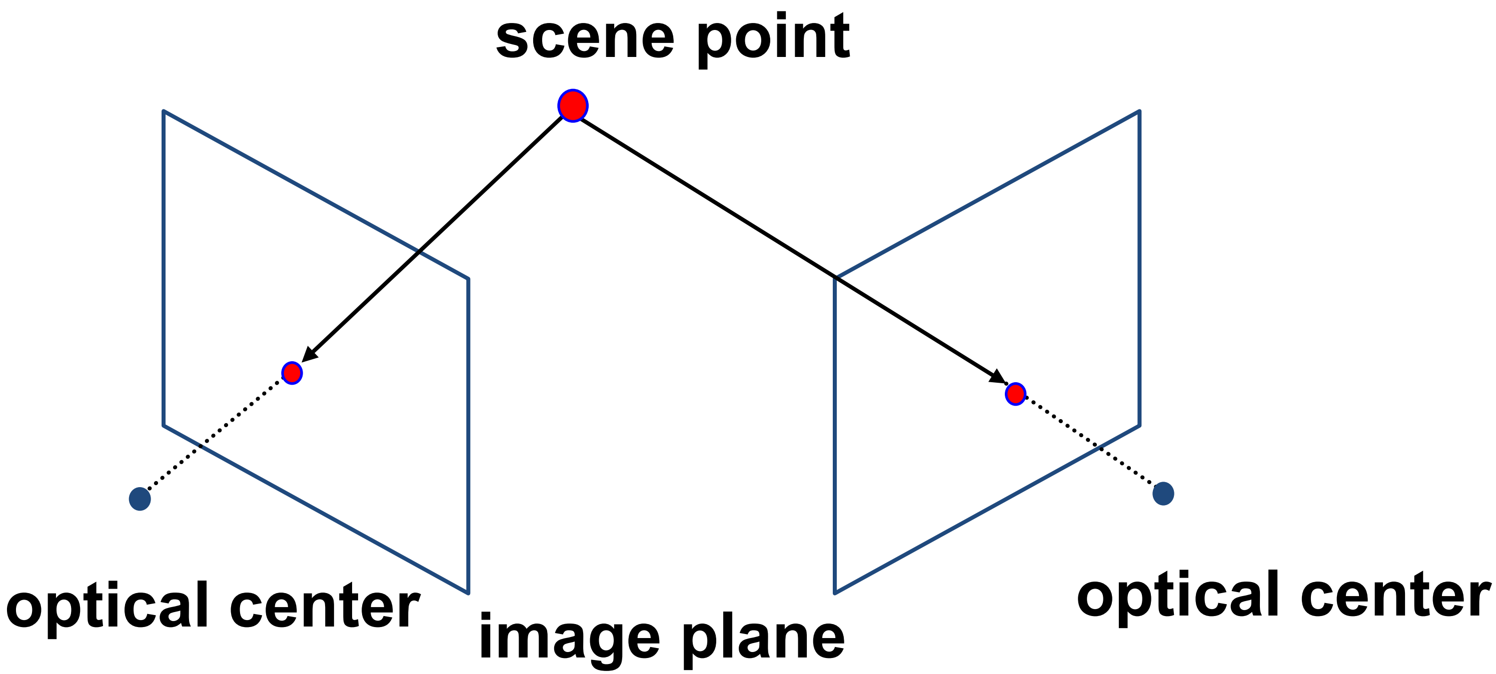
Occlusion Compatible Rendering Order

- Project the new viewpoint onto the original image and divide the image into 1, 2 or 4 "sheets"
- Instantaneous epipole

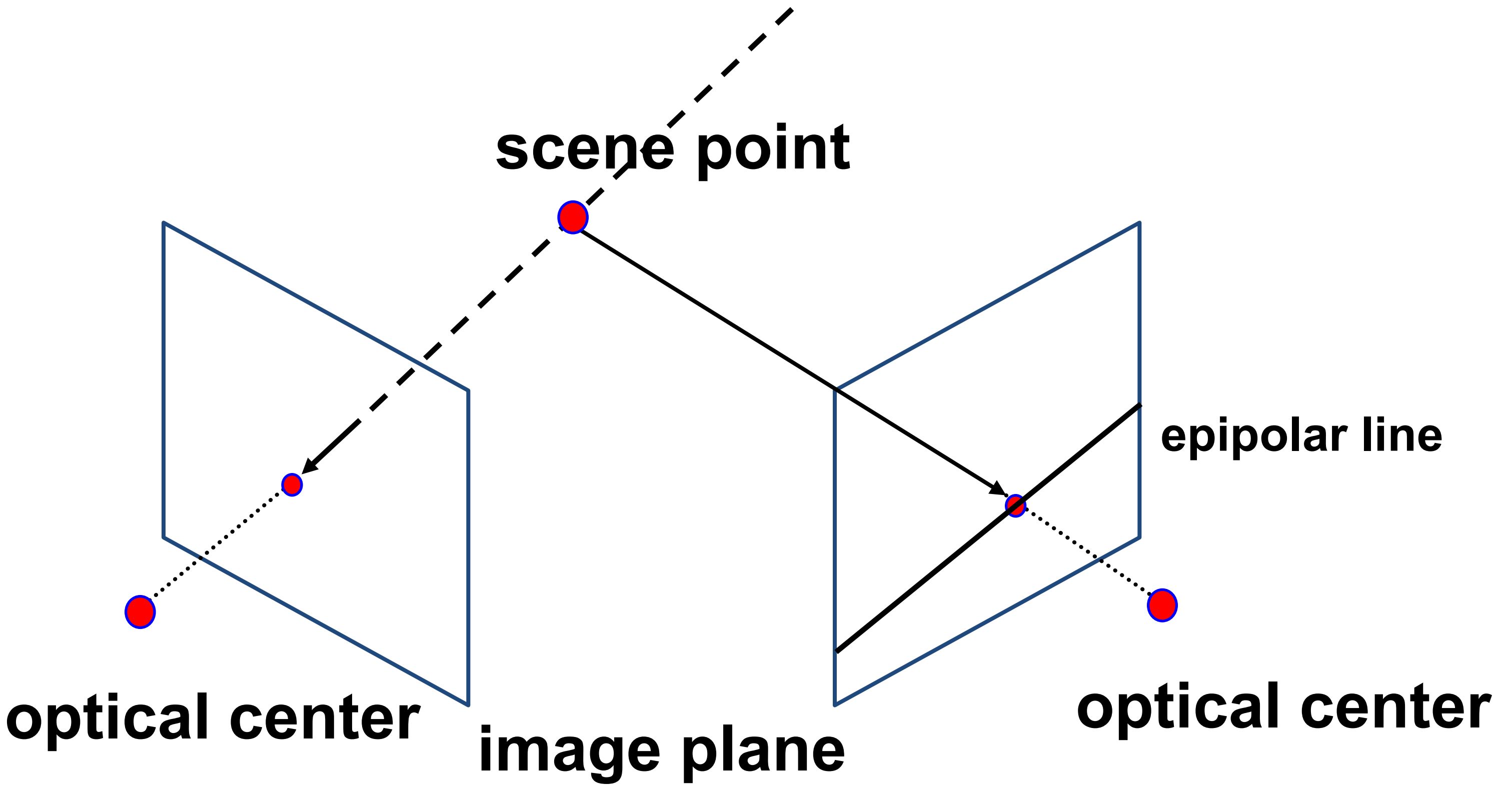




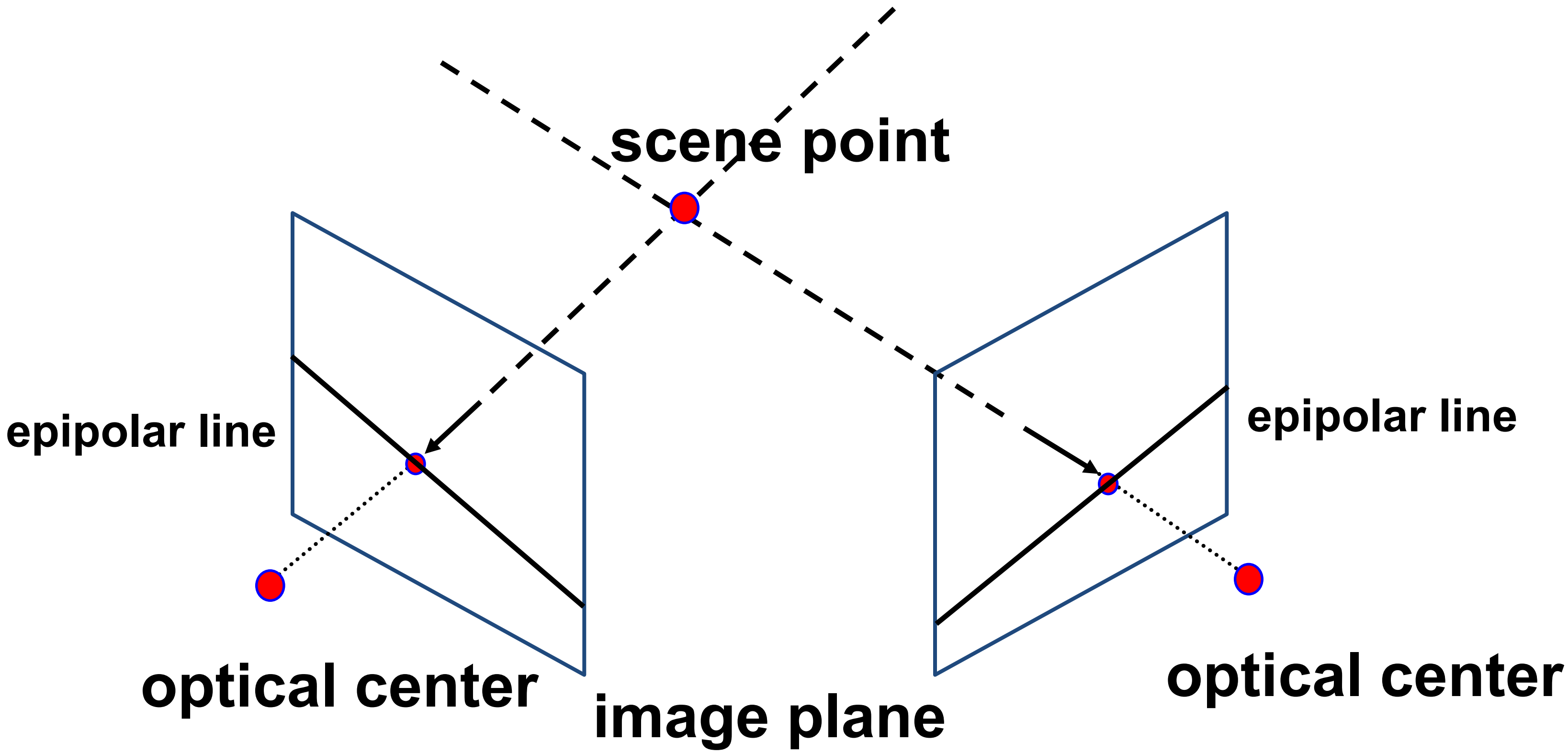
Correspondence



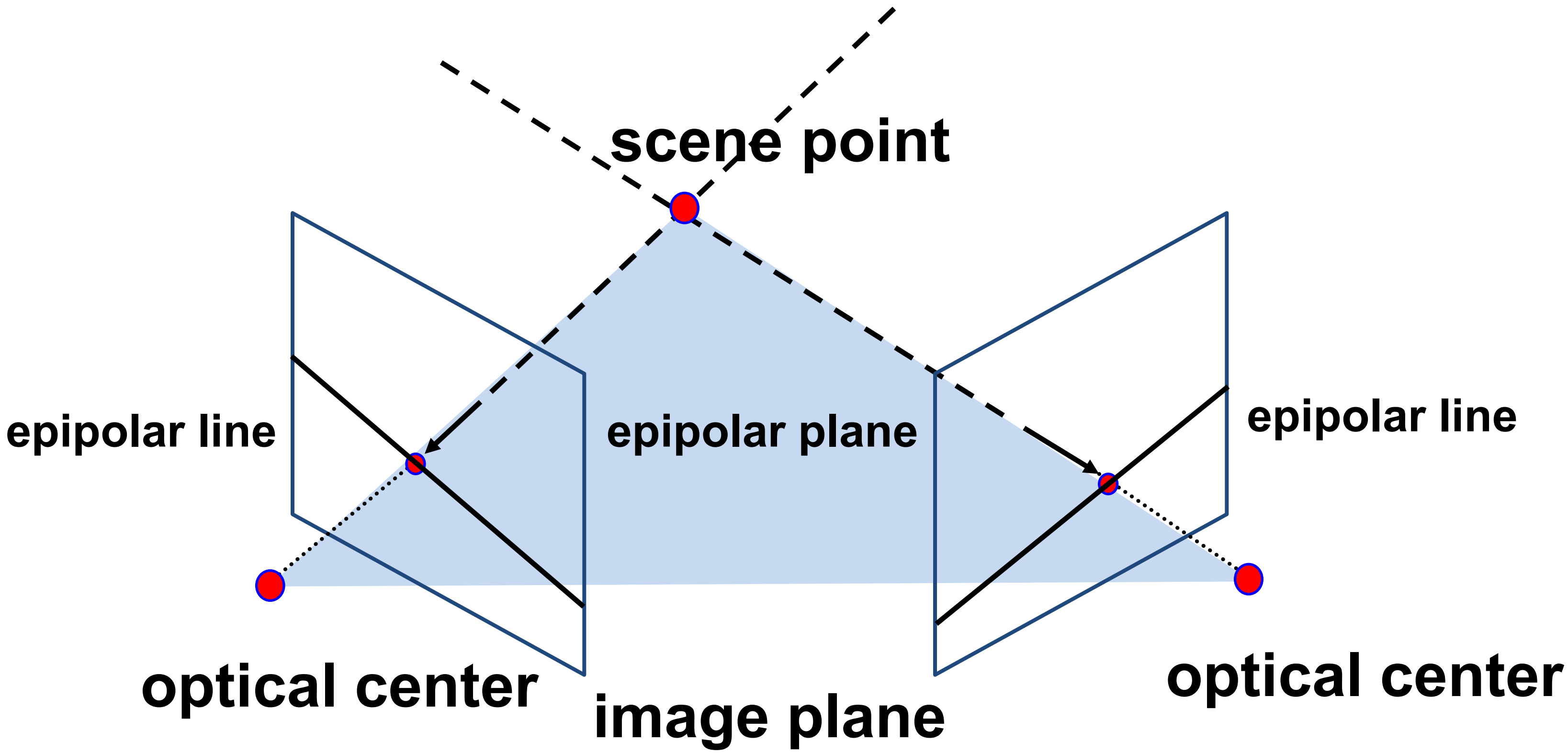
Correspondence



Correspondence

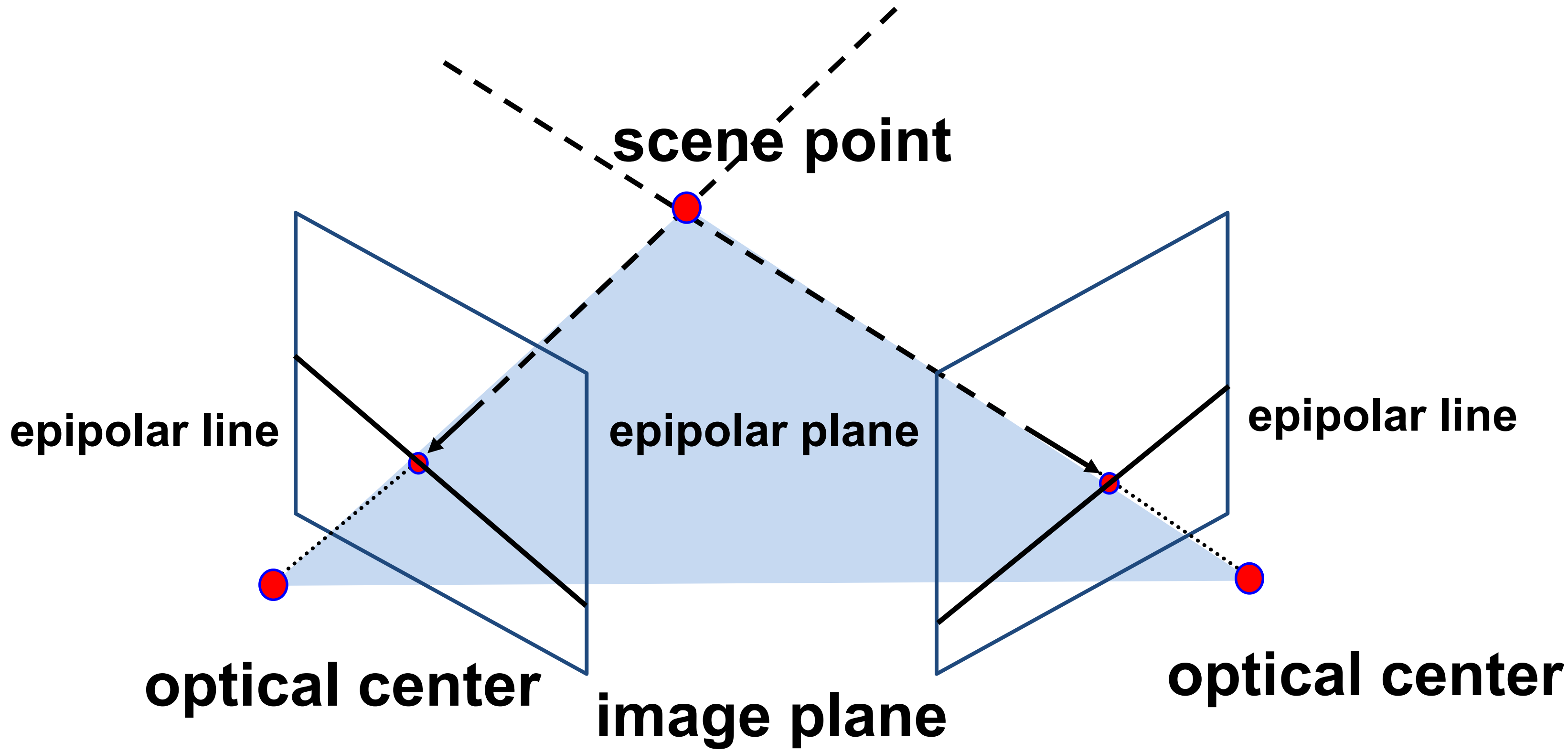


Epipolar Geometry



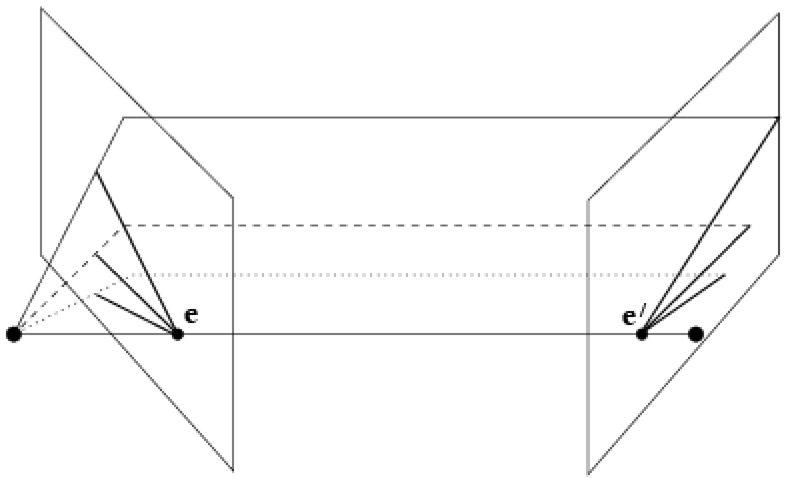
Epipolar Constraint: reduces correspondence problem to 1D search along *conjugate epipolar lines*

Epipolar Geometry

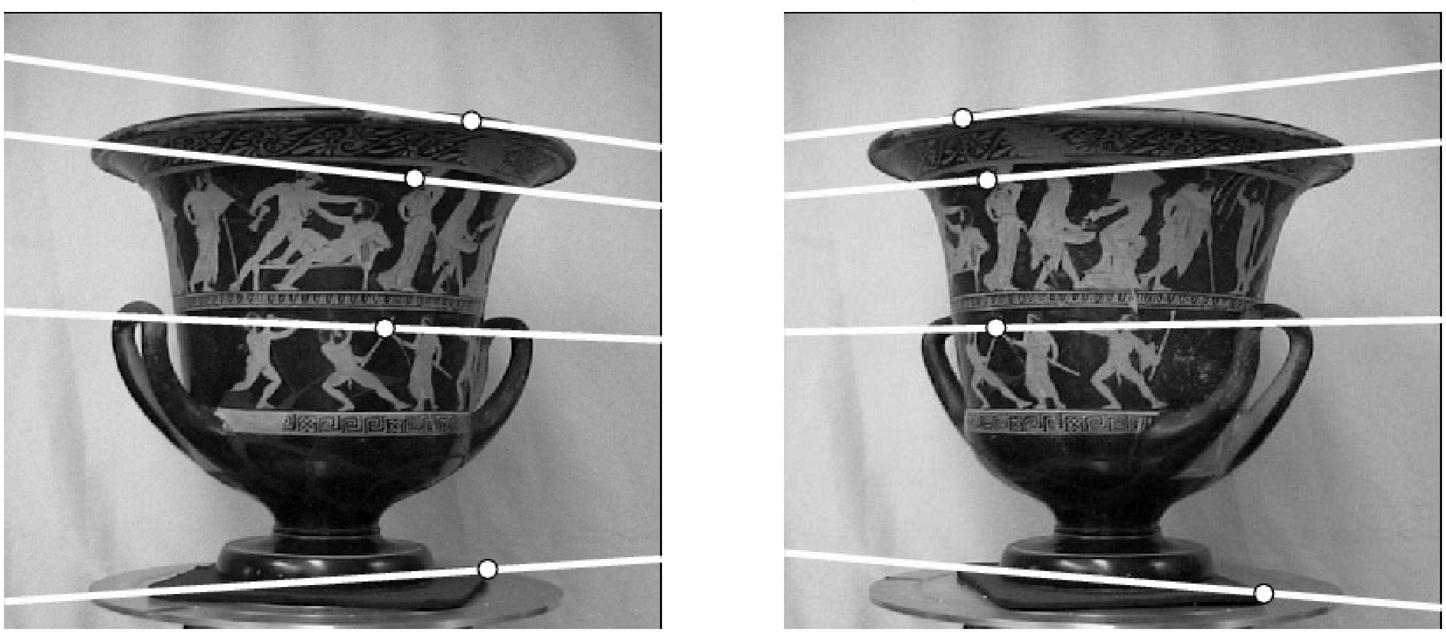


Epipolar Constraint: can be expressed using the *fundamental matrix* F

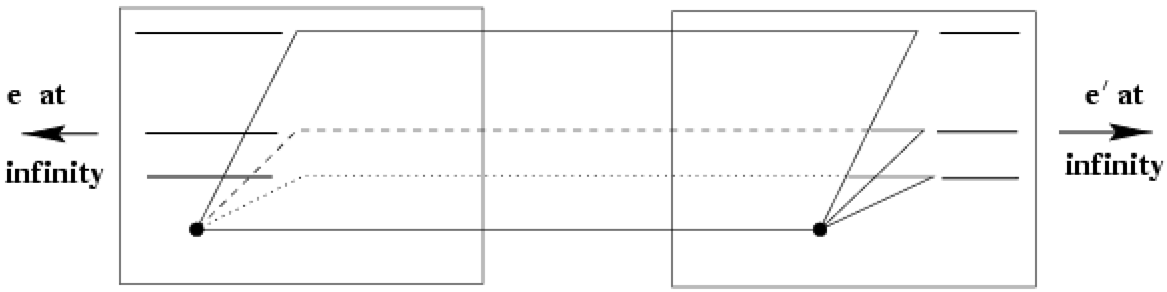
Epipolar Geometry



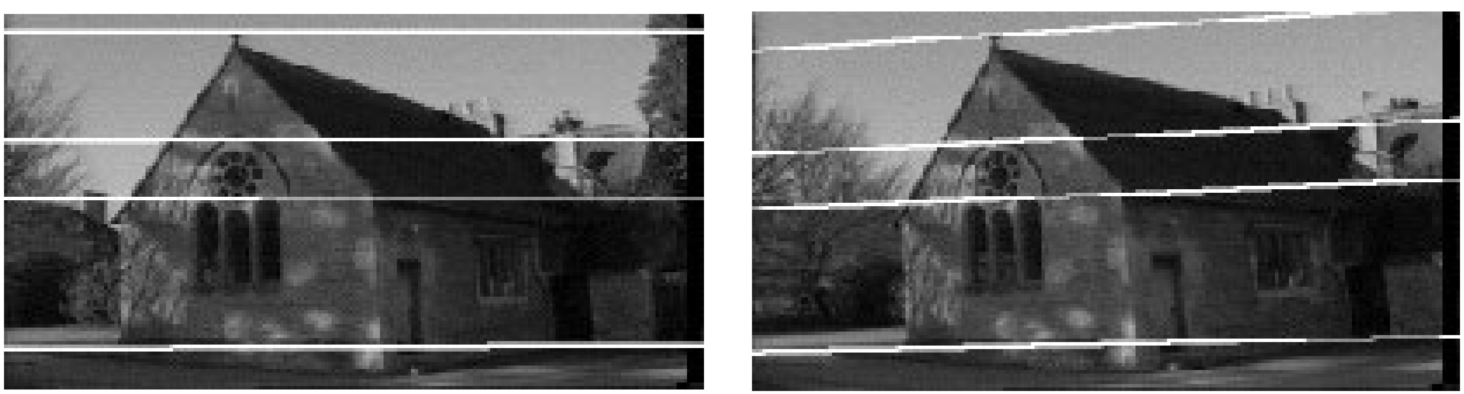
converging cameras



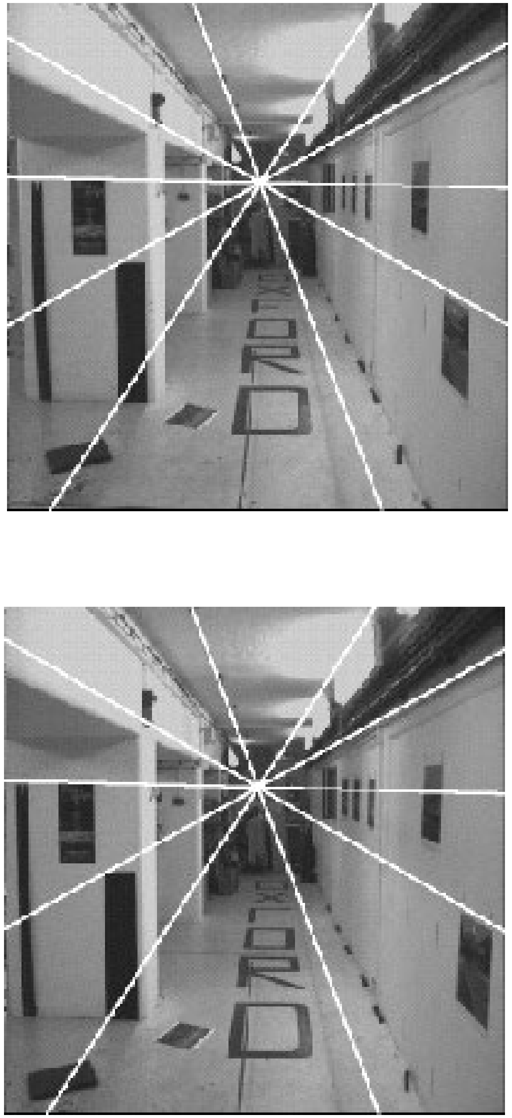
Epipolar Geometry



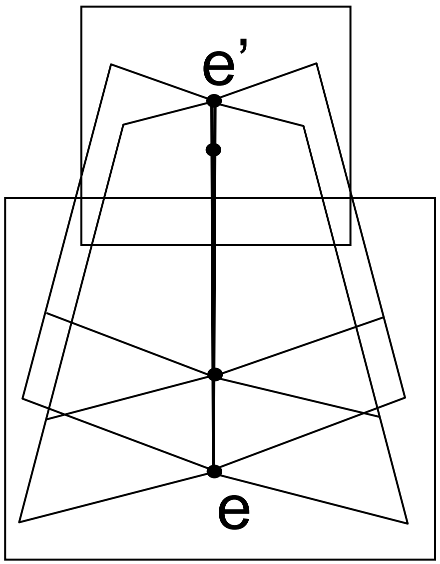
motion parallel with image plane



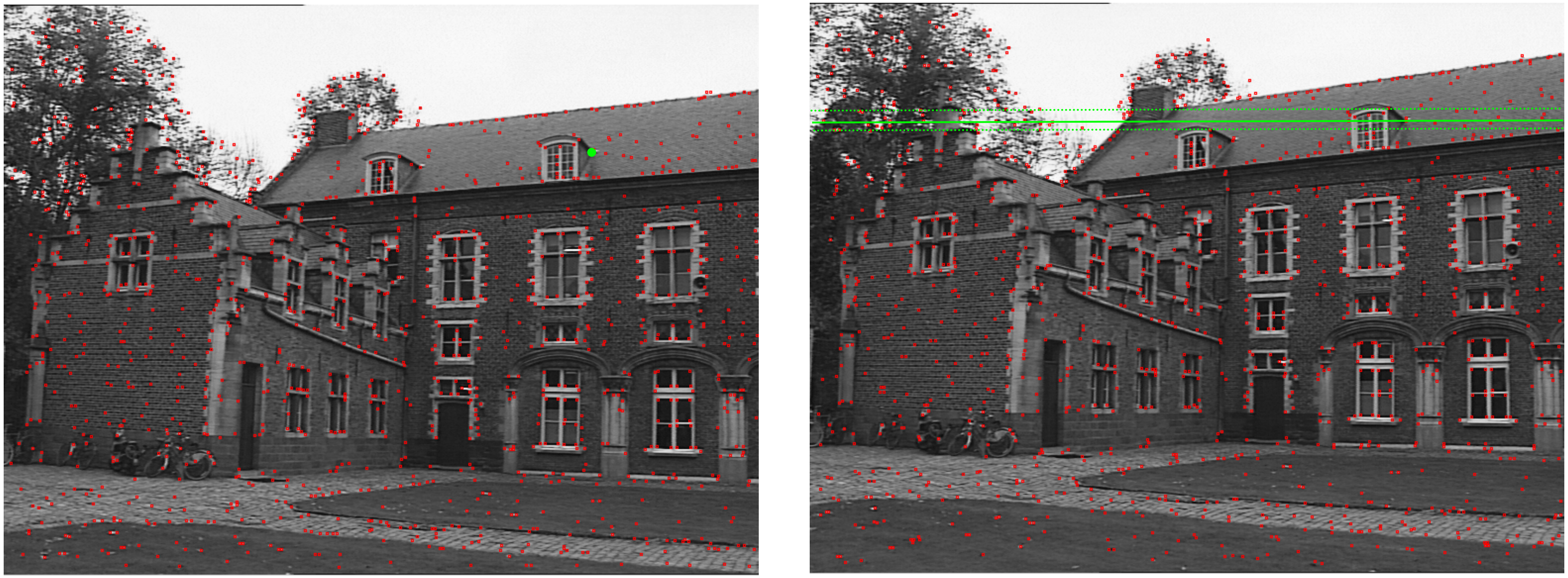
Epipolar Geometry



Forward motion

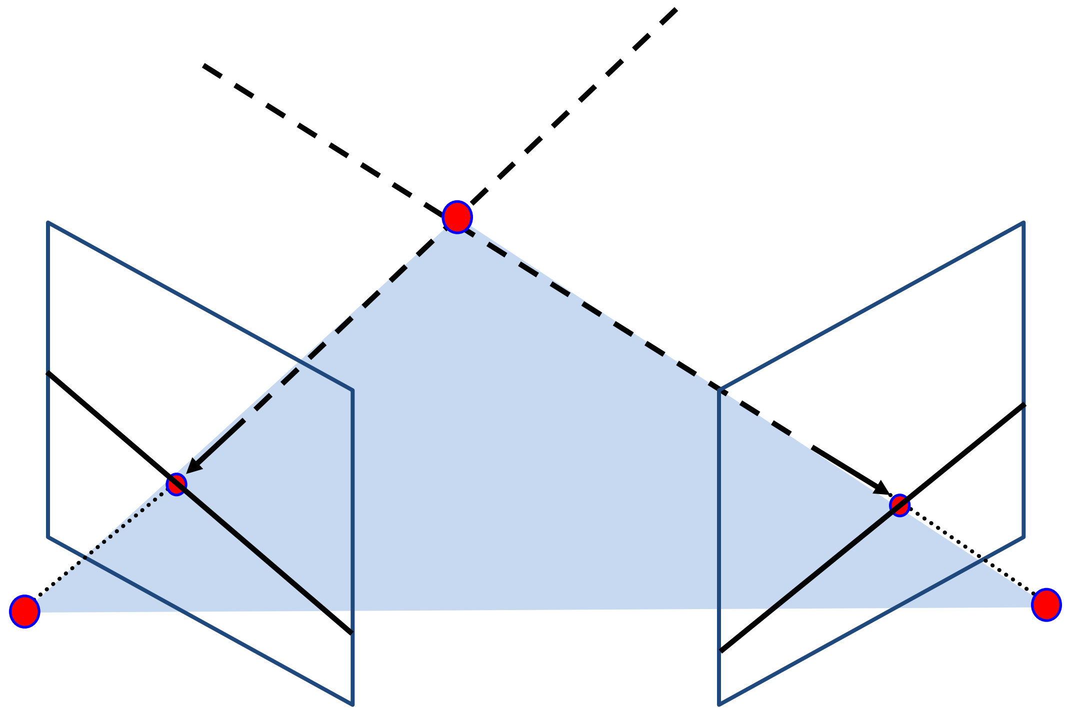


Epipolar Geometry



Correspondence reduced to looking in a small neighborhood of a line…

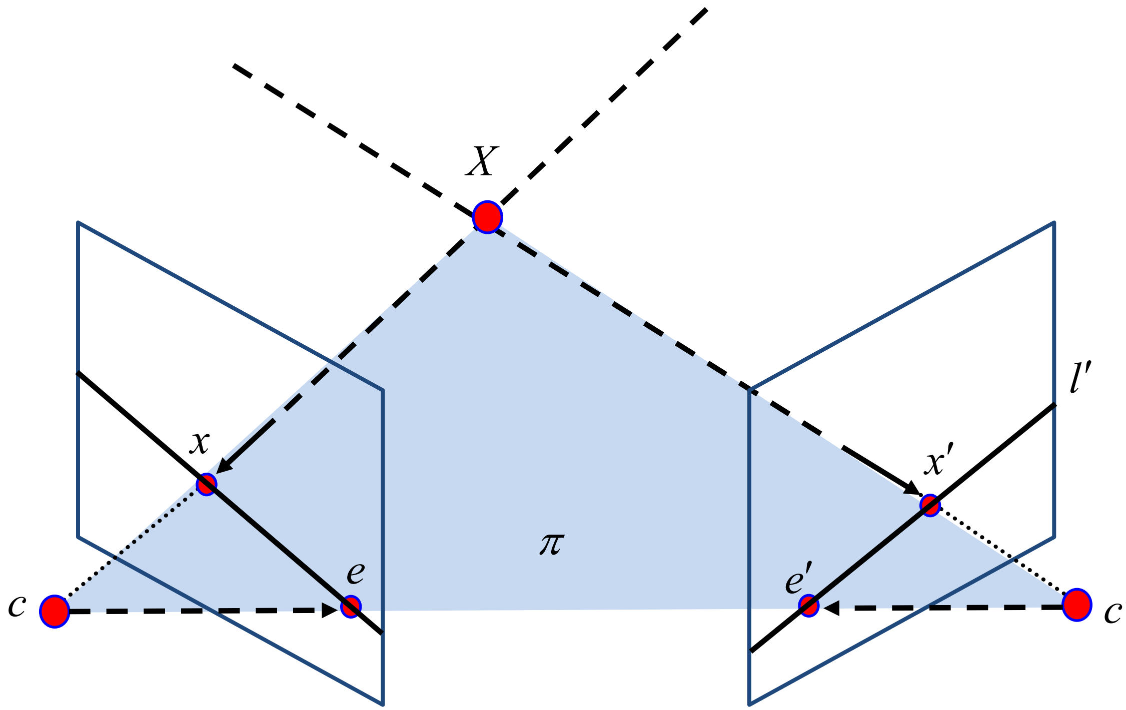
Fundamental Matrix



How to compute the fundamental matrix?

1. geometric explanation…
2. algebraic explanation…

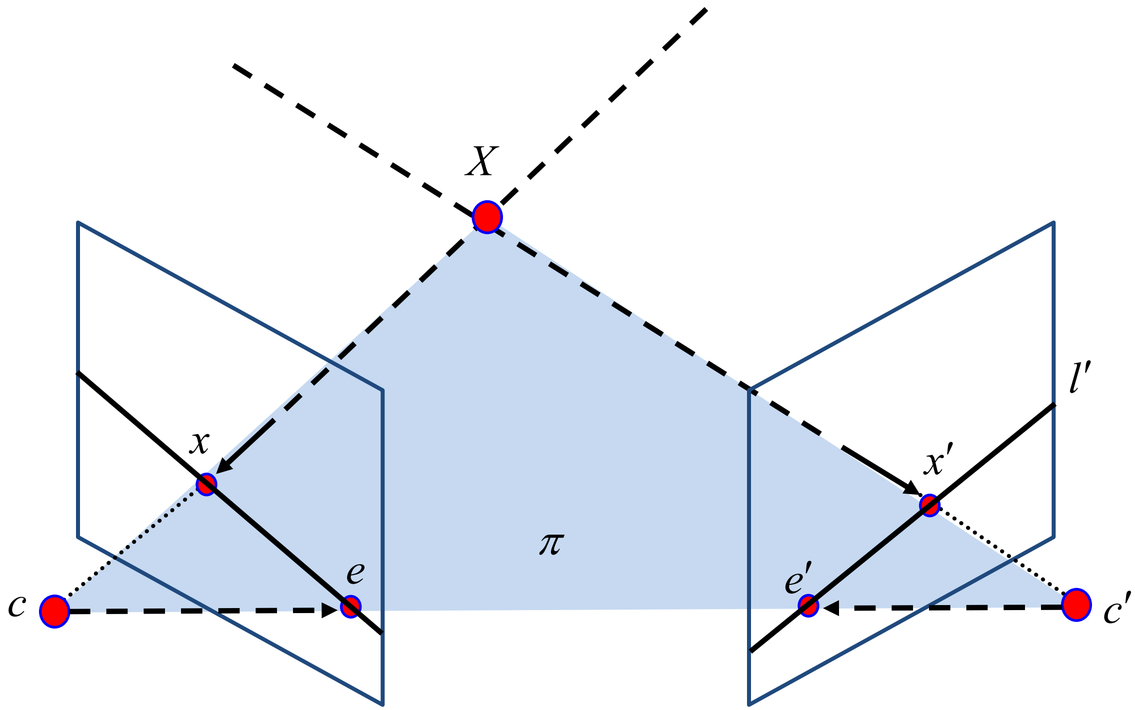
Fundamental Matrix: Geometric Exp.



Thus, there is a mapping $x \rightarrow l'$

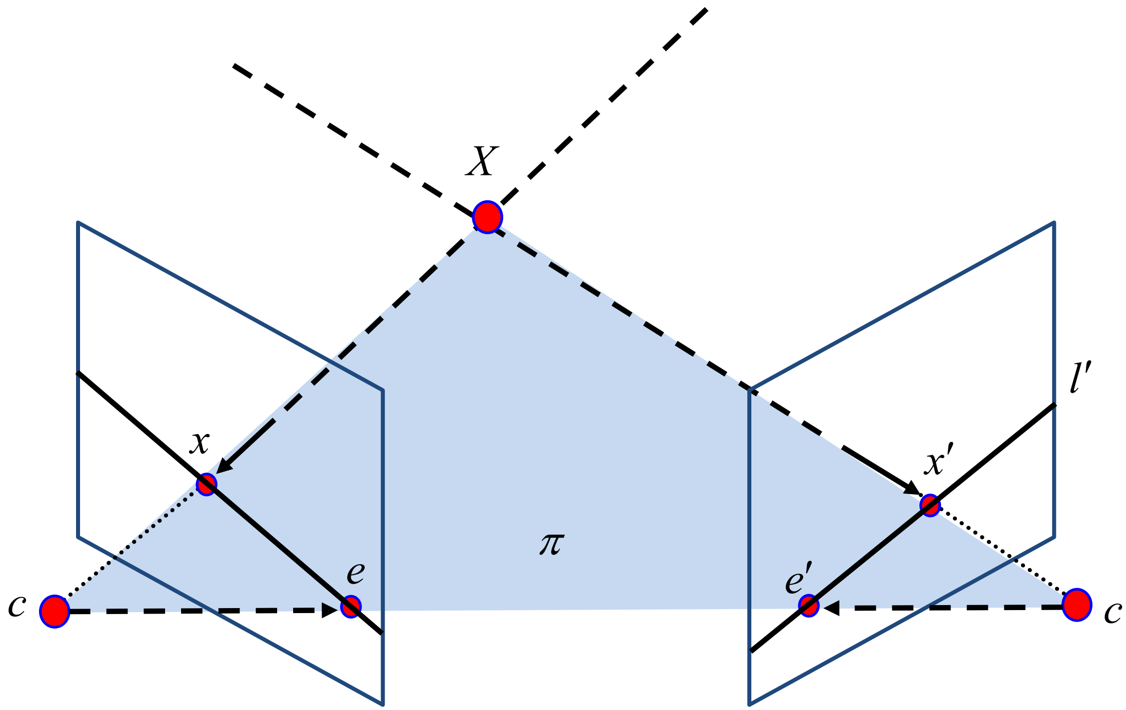
$\uparrow$ point $\uparrow$ line

Fundamental Matrix: Geometric Exp.



How do you map a point to a line?

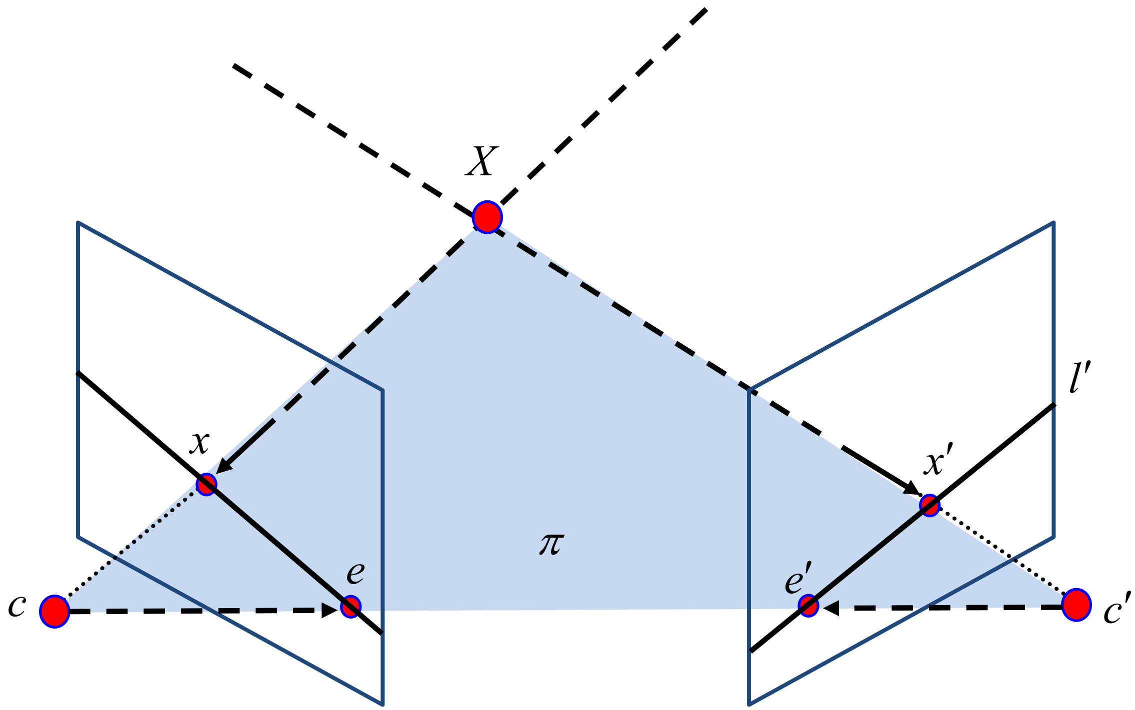
Fundamental Matrix: Geometric Exp.



Idea:

- We know (x')'s are in a plane
- Define a line by its "perpendicular", then we can use dot product; e.g., $x' \cdot l' = 0$ or $(x' - c') \cdot l' = 0$

Fundamental Matrix: Geometric Exp.



What is a definition of l' as perpendicular to the pictured epipolar line?

$$l' = (e' - c') \times (x' - c') \longrightarrow l' = e' \times x'$$

(assume all in canonical frame of the right-side camera)

Fundamental Matrix: Geometric Exp.

$$l' = e' \times x'$$

Cross product can be expressed using matrix notation:

$$e' \times x' = \begin{bmatrix} 0 & -e'_z & e'_y \\ e'_z & 0 & -e'_x \\ -e'_y & e'_x & 0 \end{bmatrix} \begin{bmatrix} x'_x \\ x'_y \\ x'_z \end{bmatrix}$$

$$e' \times x' = [e']_\times x'$$

$$\boxed{l' = [e']_\times x'}$$

Fundamental Matrix: Geometric Exp.

How do you compute x' ?

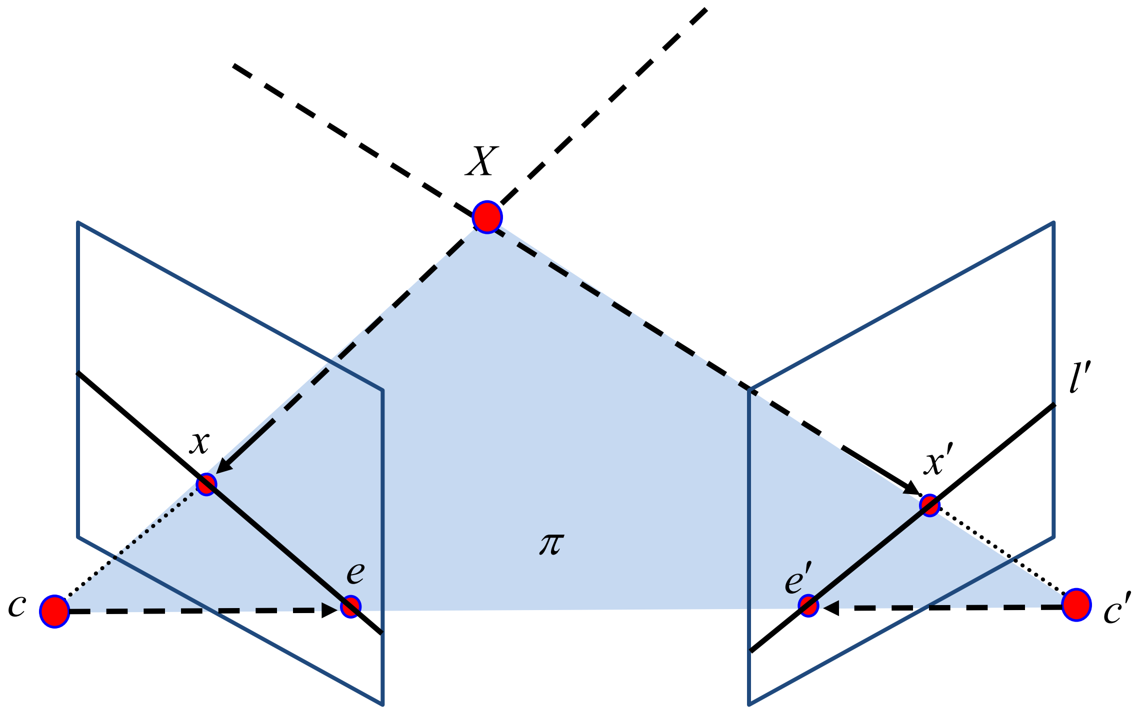
Use a homography (or projective transformation) to map x to x'

(Homography: maps points in a plane to another plane)

$$x = \begin{bmatrix} x_x \\ x_y \\ 1 \end{bmatrix}, x' = \begin{bmatrix} w'x'_x \\ w'x'_y \\ w' \end{bmatrix}, H = \begin{bmatrix} . & . & . \\ . & . & . \\ . & . & . \end{bmatrix}$$

$$\boxed{x' = Hx}$$

Fundamental Matrix: Geometric Exp.

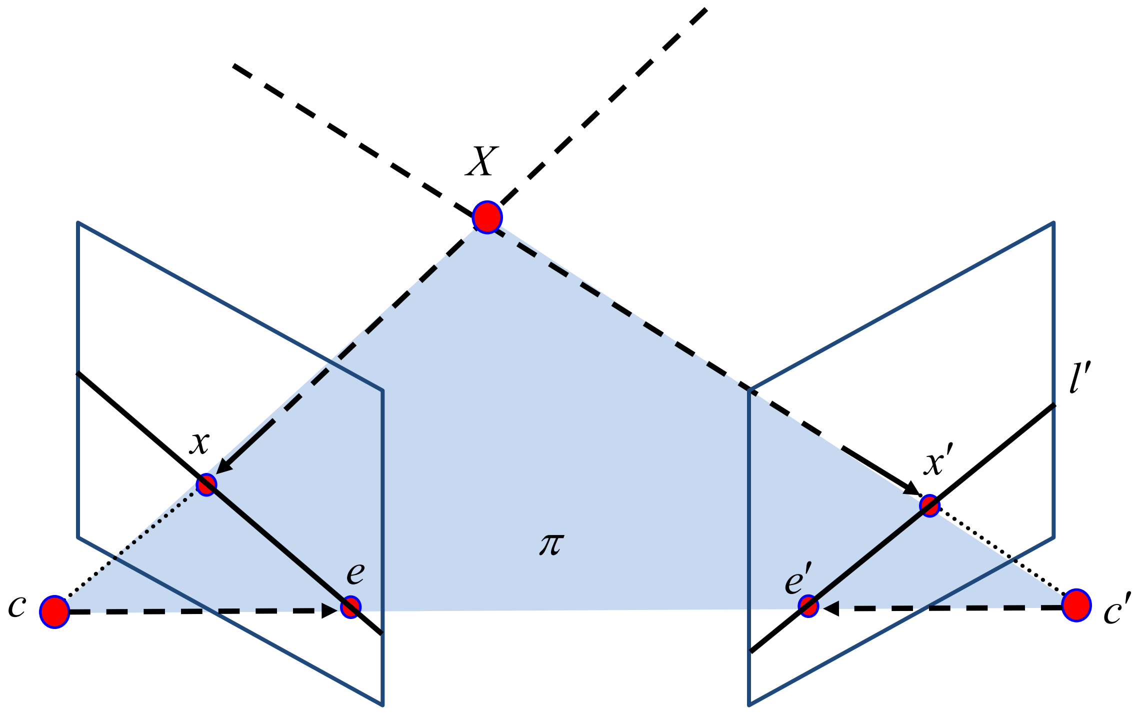


$$\left.\begin{array}{l} l' = [e']_\times x' \\ x' = Hx \end{array}\right\} \quad l' = [e']_\times Hx \Rightarrow F = [e']_\times H \Rightarrow \boxed{x'^T F x = 0}$$

Want $x' \cdot l' = 0$ …

Epipolar Constraint

Fundamental Matrix: Algebraic Exp.



$$x = \begin{bmatrix} R & t \end{bmatrix} X$$

$$x = PX$$

$$x' = P'X$$

Fundamental Matrix: Algebraic Exp.

$x = PX \qquad X' = ?$

$X(t) = P^{+}x + tc$ where P^{+} is the pseudoinverse of P

Why pseudoinverse?

Since P not square, pseudoinverse means $PP^{+} = I$ but solved as an optimization

Recall $l' = [e']_{\times} x'$

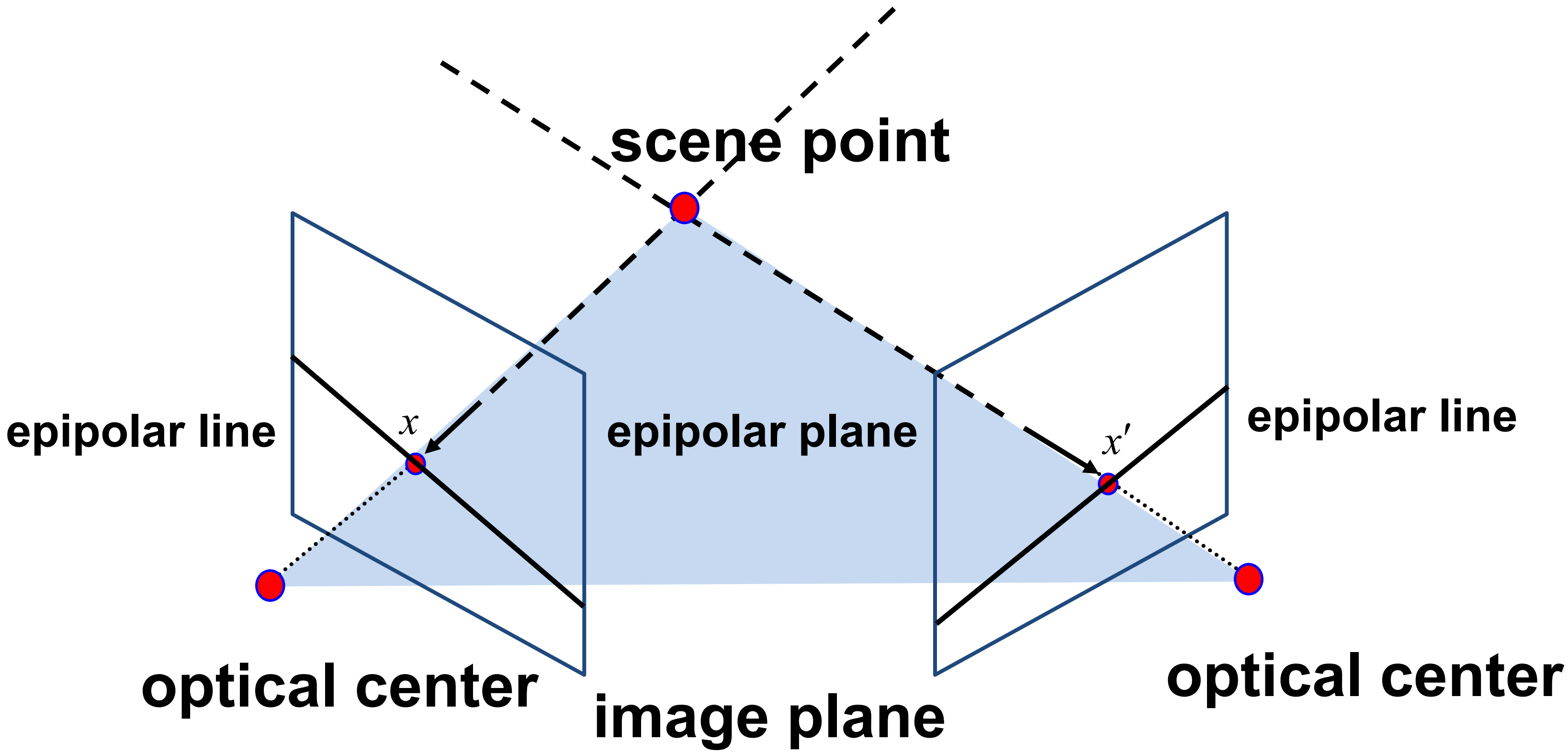
What is x' in terms of x?

(Let's assume $t = 0$ which means X in on the image plane)

$x' = P'P^{+}x \Rightarrow F = [e']_{\times} P'P^{+} \Rightarrow \boxed{x'^{T} F x = 0}$

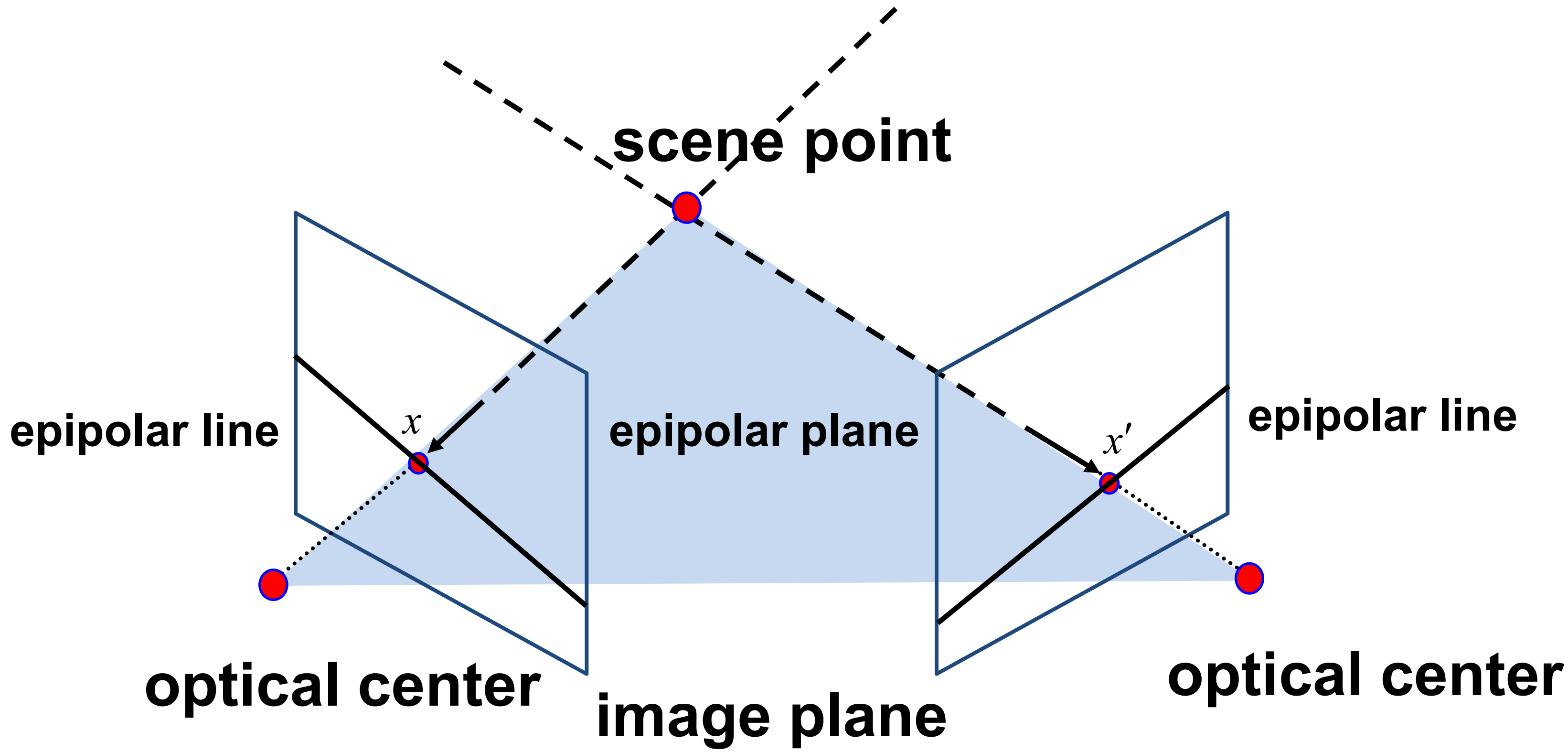
Epipolar Constraint

Correspondence: Epipolar Geometry



Epipolar constraint reduces correspondence problem to 1D search along *conjugate epipolar lines*

Correspondence: Epipolar Geometry

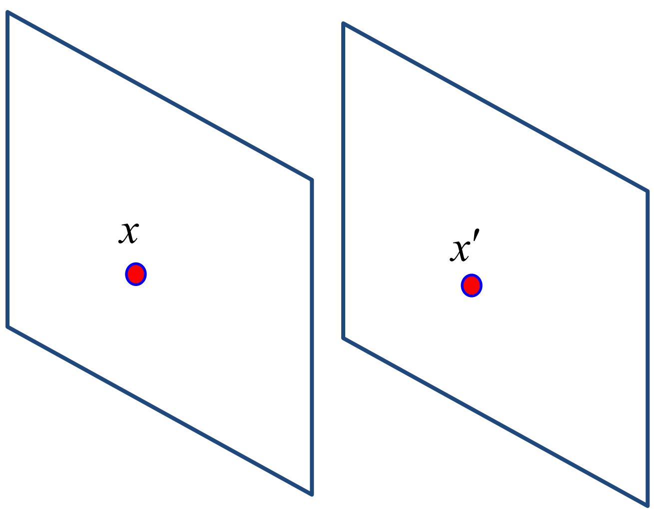


Epipolar constraint can be expressed as $x'^T F x = 0$

Fundamental matrix

Correspondence: Epipolar Geometry

Interesting case: what happens if camera motion is pure translation?



$$P = [I \mid 0] \quad P' = [I \mid t]$$

$$F = [e']_\times \qquad (H = I)$$

If motion parallel to x-axis…

$$e' = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

Thus the desire to do image rectification

$$F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$ implies horizontal epipolar line…

Correspondence: Epipolar Geometry

Thus for rectified images, correspondence is reduced to looking in a small neighborhood of a line…

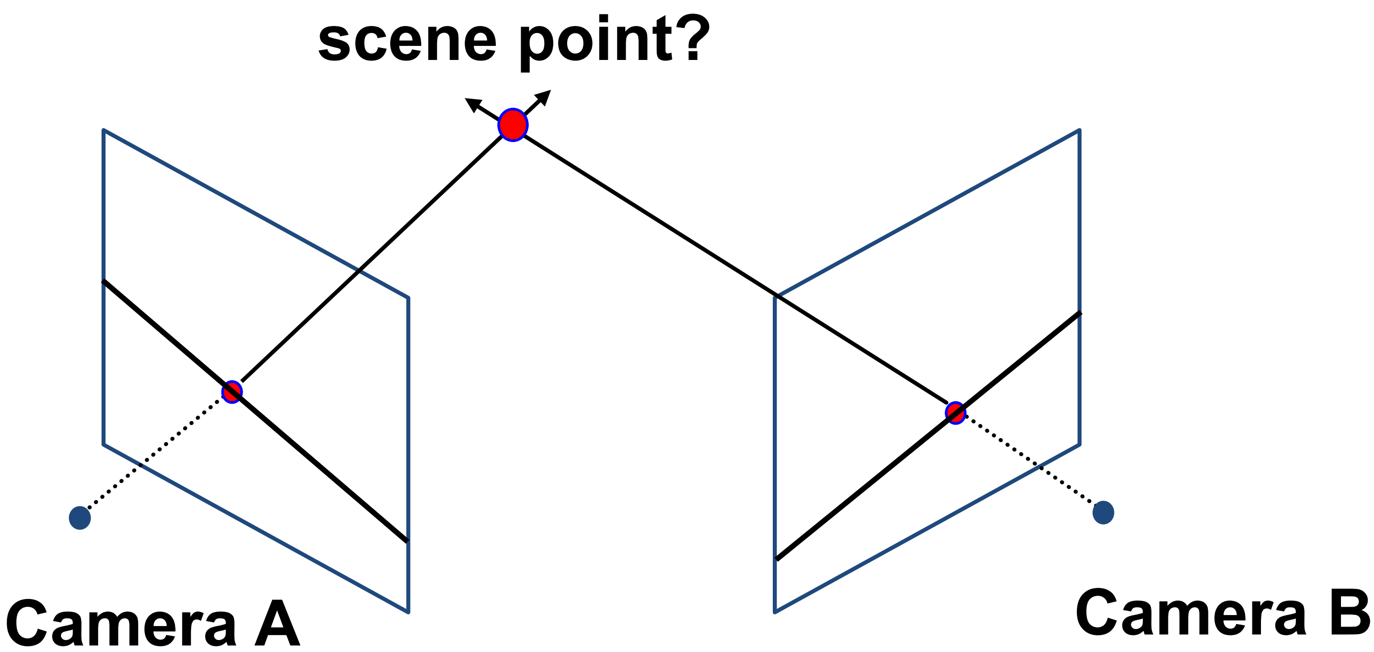
Essential Matrix

- Similar to the fundamental matrix but includes the intrinsic calibration matrix, thus the equation is in terms of the normalized image coordinates, e.g.:

$$x'^{T} E x' = 0 \quad \text{and} \quad E = K'^{T} F K$$

essential matrix

Scene Geometry

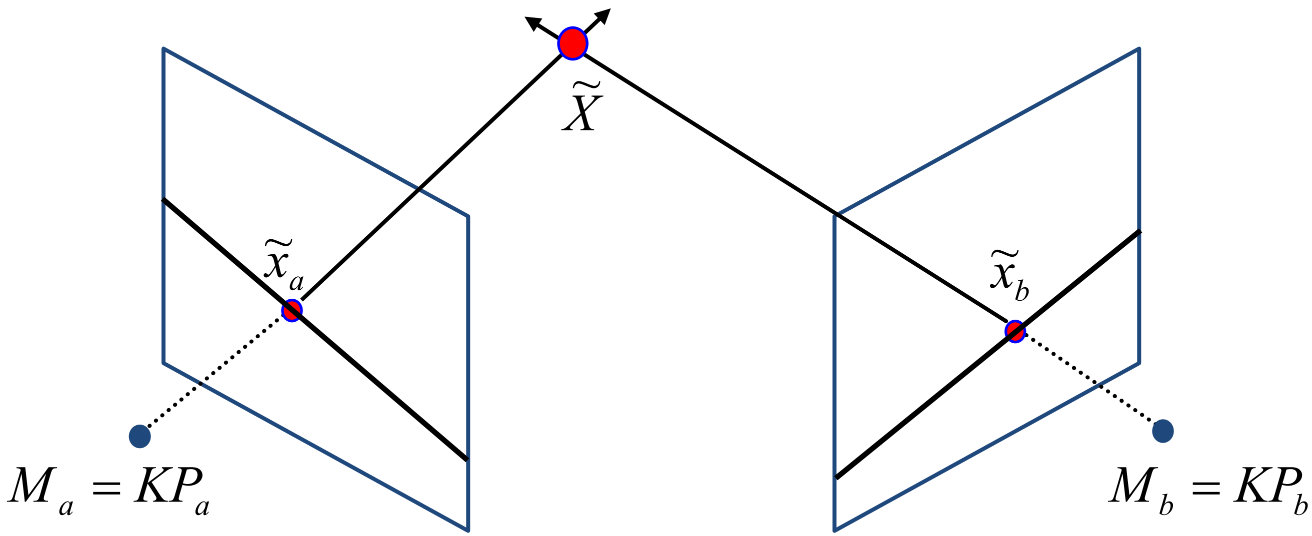


Camera geometry known

Correspondence and epipolar geometry known

What is the location of the scene point (scene geometry)?

Scene Geometry: Linear Formulation



$$\widetilde{x}_a = M_a \widetilde{X} \quad \text{or} \quad \widetilde{x}_b = M_b \widetilde{X}$$

Problem?

Assumes we know $\widetilde{x} = \begin{bmatrix} x' & y' & w' \end{bmatrix}^T$

But what is the value for w'?

Scene Geometry: Linear Formulation

$\tilde{x} = M\tilde{X}$ where $\tilde{x} = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$

Recall $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$ where x and y are the observed projections

Let $\tilde{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$, thus $s = ?w'$

Hence? $sx = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$

$sy = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$

Scene Geometry: Linear Formulation

Given $\begin{aligned} sx &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ sy &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \\ s &= m_{31}X + m_{32}Y + m_{33}Z + m_{34} \end{aligned}$ and N cameras

For a scene point, how many unknowns? 3+N

For a scene point, how many camera views needed? 3N≥3+N

In general, one scene point observed in at least two views is sufficient…

Scene Geometry: Linear Formulation

$$\begin{bmatrix} \; \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ s_a \\ s_b \end{bmatrix} = \begin{bmatrix} \; \end{bmatrix}$$

$$\left(\begin{aligned} sx &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ sy &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \\ s &= m_{31}X + m_{32}Y + m_{33}Z + m_{34} \end{aligned} \right) \text{x2}$$

Scene Geometry: Linear Formulation

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & -x & 0 \\ m_{21} & m_{22} & m_{23} & -y & 0 \\ m_{31} & m_{32} & m_{33} & -1 & 0 \\ m'_{11} & m'_{12} & m'_{13} & 0 & -x' \\ m'_{21} & m'_{22} & m'_{23} & 0 & -y' \\ m'_{31} & m'_{32} & m'_{33} & 0 & -1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ s \\ s' \end{bmatrix} = \begin{bmatrix} -m_{14} \\ -m_{24} \\ -m_{34} \\ -m'_{14} \\ -m'_{24} \\ -m'_{34} \end{bmatrix}$$

Cameras *M* and *M'*

$$\left(\begin{aligned} sx &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ sy &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \\ s &= m_{31}X + m_{32}Y + m_{33}Z + m_{34} \end{aligned} \right) \text{x2}$$

Scene Geometry: Nonlinear Form.

- Remember "Bundle Adjustment"
  - Given initial guesses, use nonlinear least squares to refine/compute the calibration parameters
  - Simple but good convergence depends on accuracy of initial guess

Scene Geometry: Nonlinear Form.

Recall

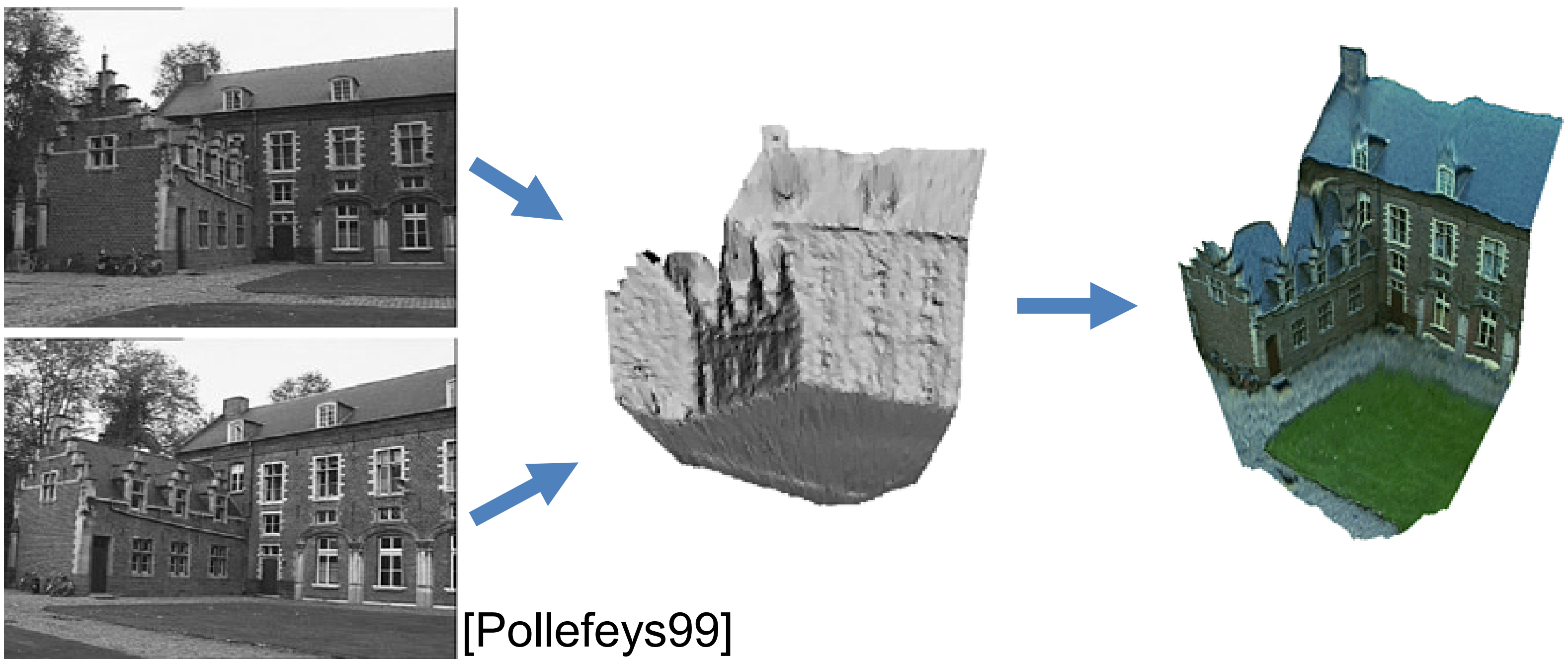
$$E = \frac{1}{mn}\sum_{ij}\left[(x_{ij} - \frac{m_{i1}\cdot\widetilde{X}_j}{m_{i3}\cdot\widetilde{X}_j})^2 + (y_{ij} - \frac{m_{i2}\cdot\widetilde{X}_j}{m_{i3}\cdot\widetilde{X}_j})^2\right]$$

Goal is $E \to 0$

For scene geometry, $\widetilde{X}$ are the unknowns…

Example Result

- Using dense feature-based stereo



[Pollefeys99]