Computer Vision

CSC-455

Muhammad Najam Dar

Today's Lecture

- Motion Analysis
- 3D Vision
- Triangulation Principle
- Stereoscopy

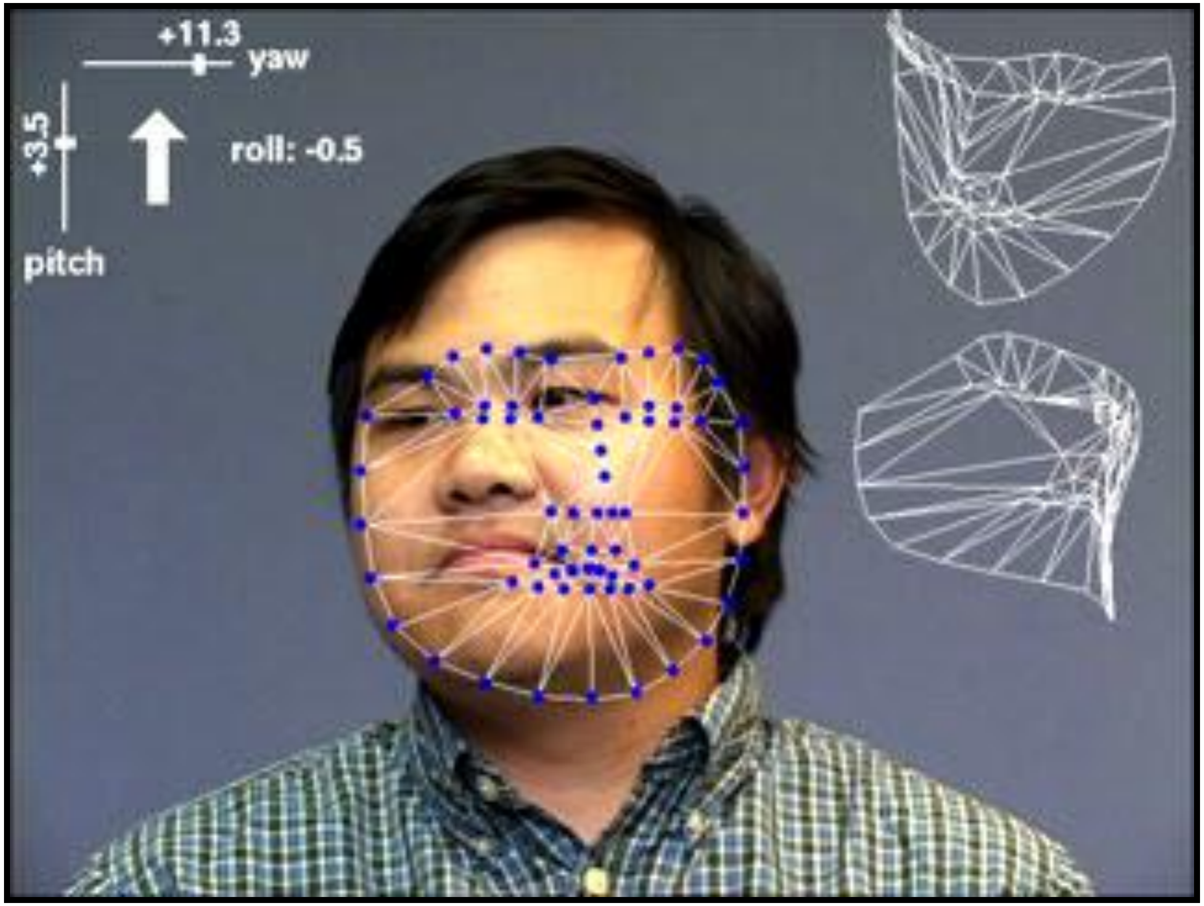
Optical Flow and Motion

We are interested in finding the movement of scene objects from time-varying images (videos).

Lots of uses

- Track object behavior
- Correct for camera jitter (stabilization)
- Align images
- 3D shape reconstruction
- Special effects

Face Tracking

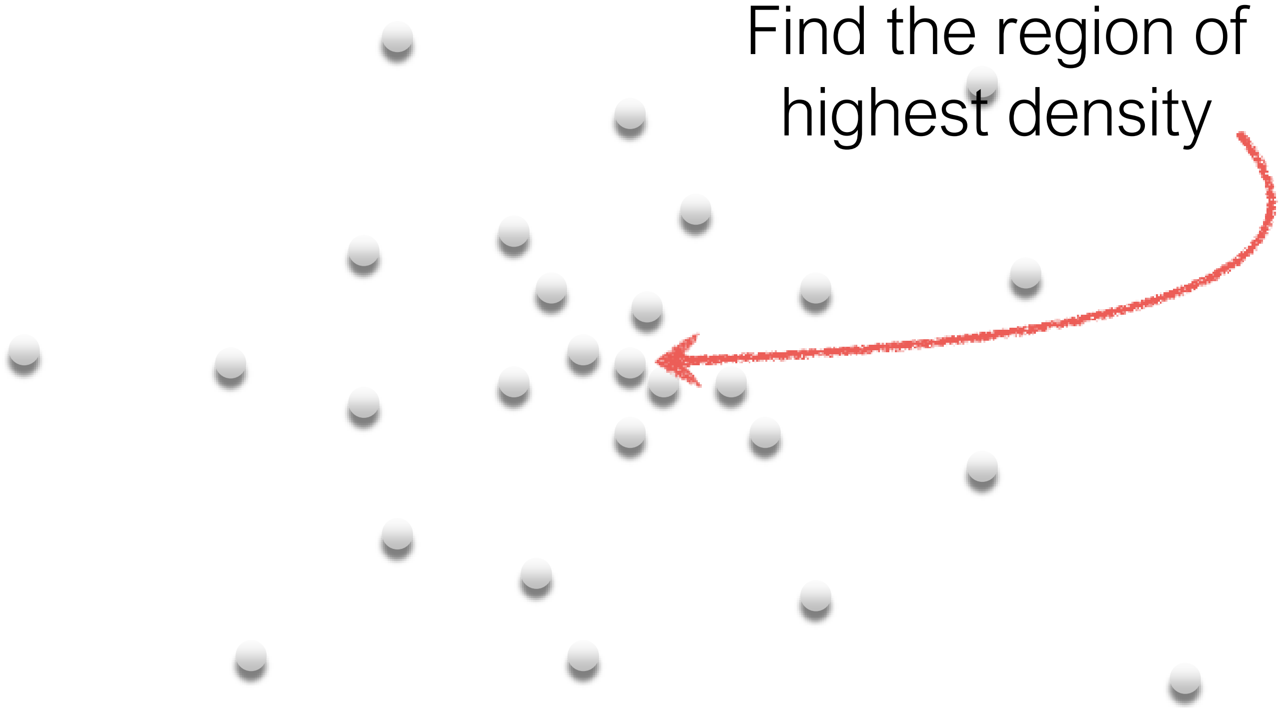



Nationale-Nederlanden
jackpot €5 milj.
ING

Mean Shift Algorithm

A ‘mode seeking’ algorithm

Fukunaga & Hostetler (1975)



Mean Shift Algorithm

A ‘mode seeking’ algorithm

Fukunaga & Hostetler (1975)

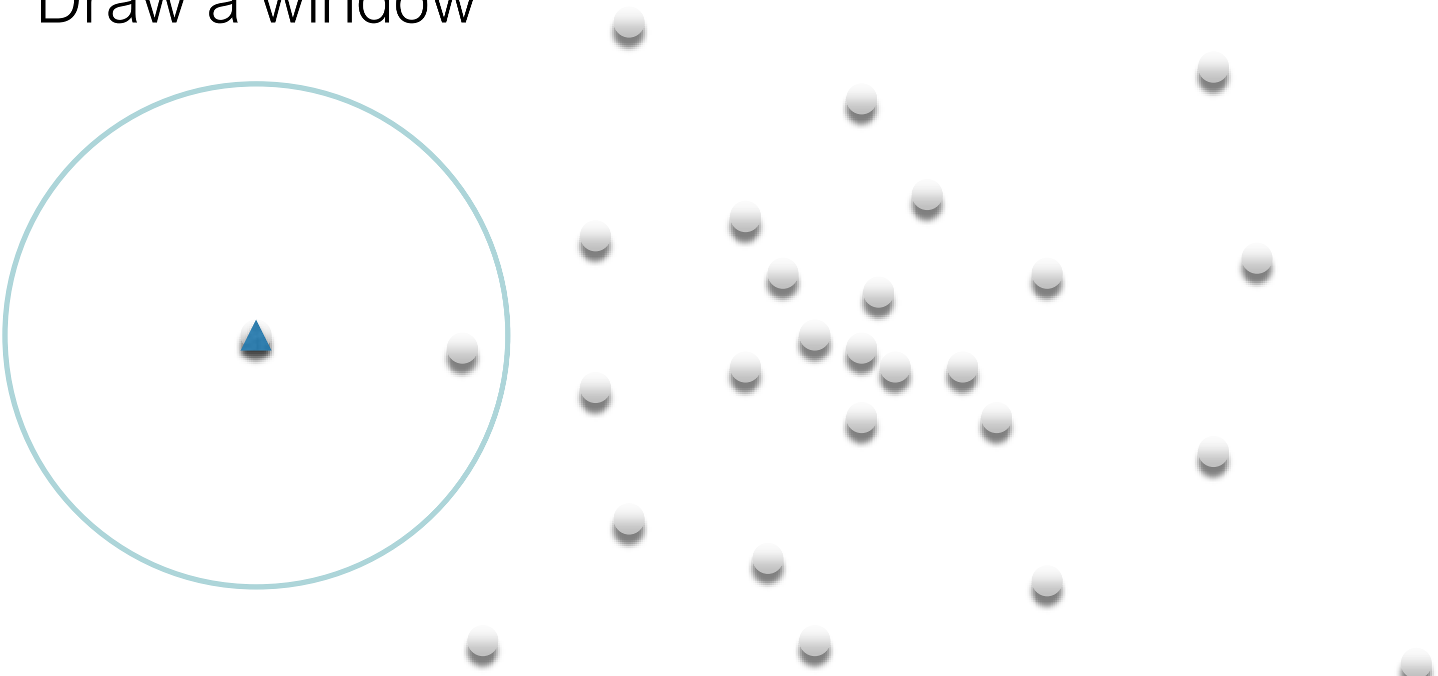
Pick a point

Mean Shift Algorithm

A 'mode seeking' algorithm

Fukunaga & Hostetler (1975)

Draw a window

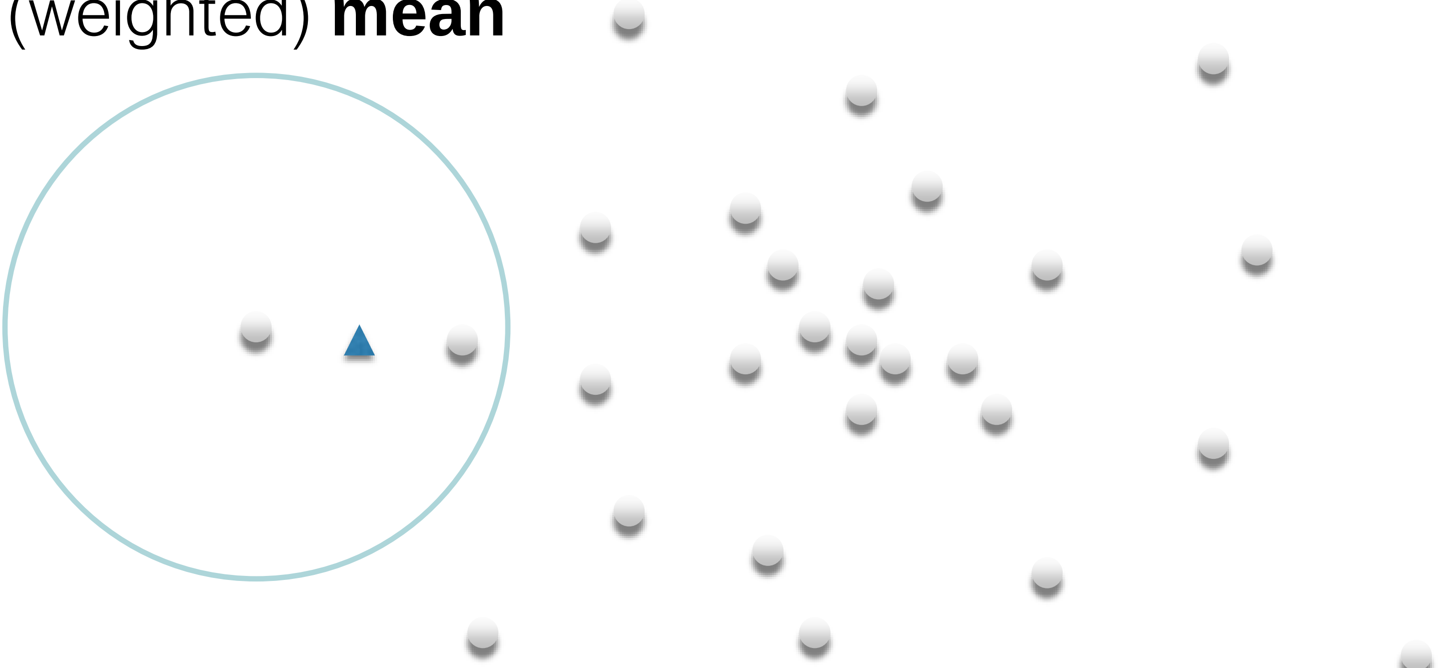


Mean Shift Algorithm

A 'mode seeking' algorithm

Fukunaga & Hostetler (1975)

Compute the (weighted) **mean**

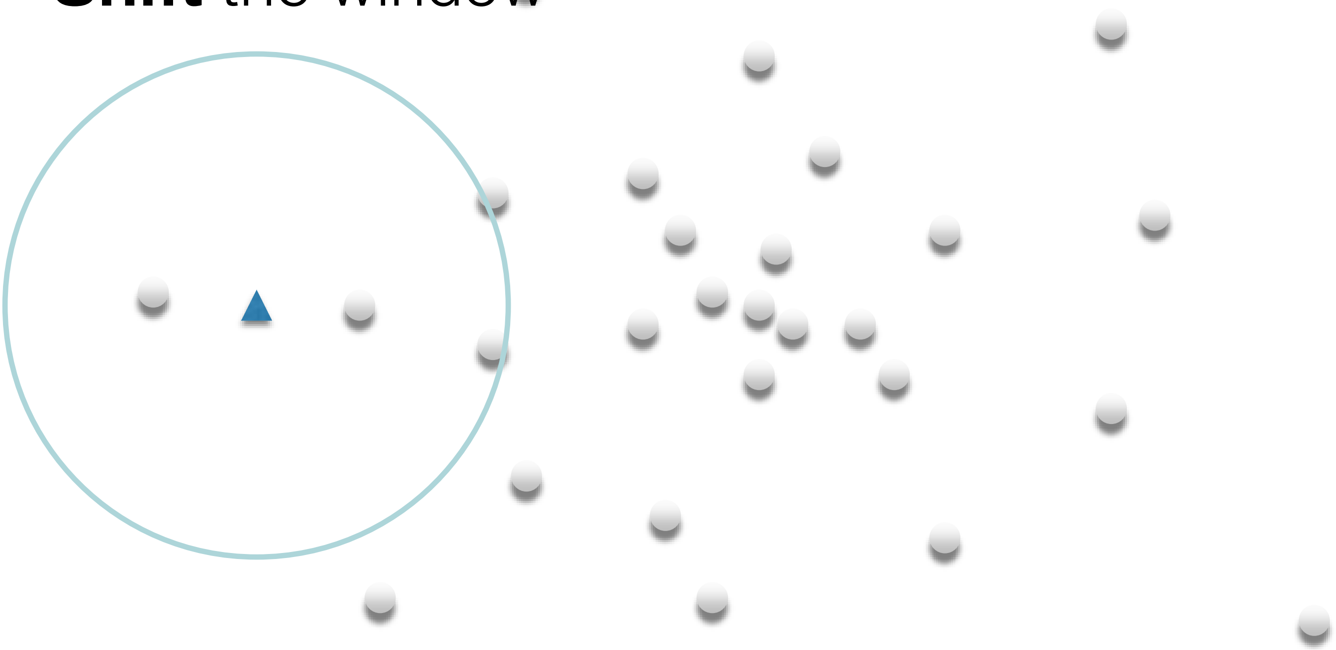


Mean Shift Algorithm

A 'mode seeking' algorithm

Fukunaga & Hostetler (1975)

Shift the window

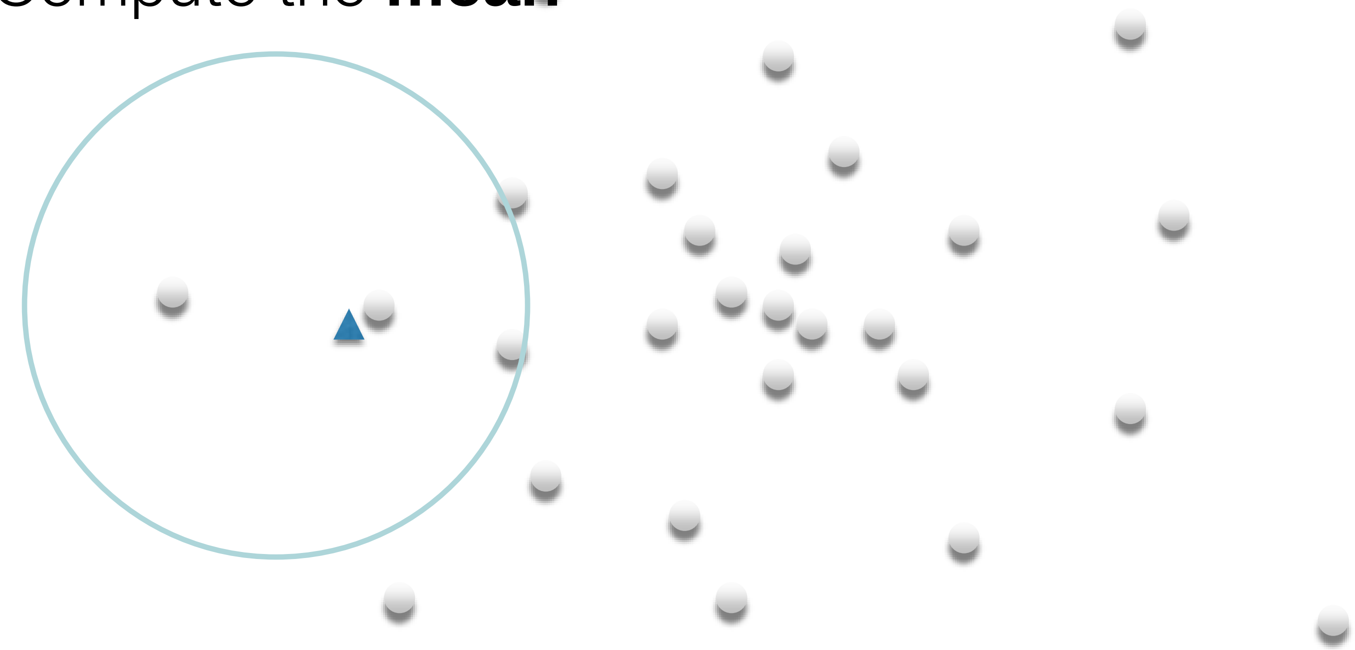


Mean Shift Algorithm

A 'mode seeking' algorithm

Fukunaga & Hostetler (1975)

Compute the **mean**

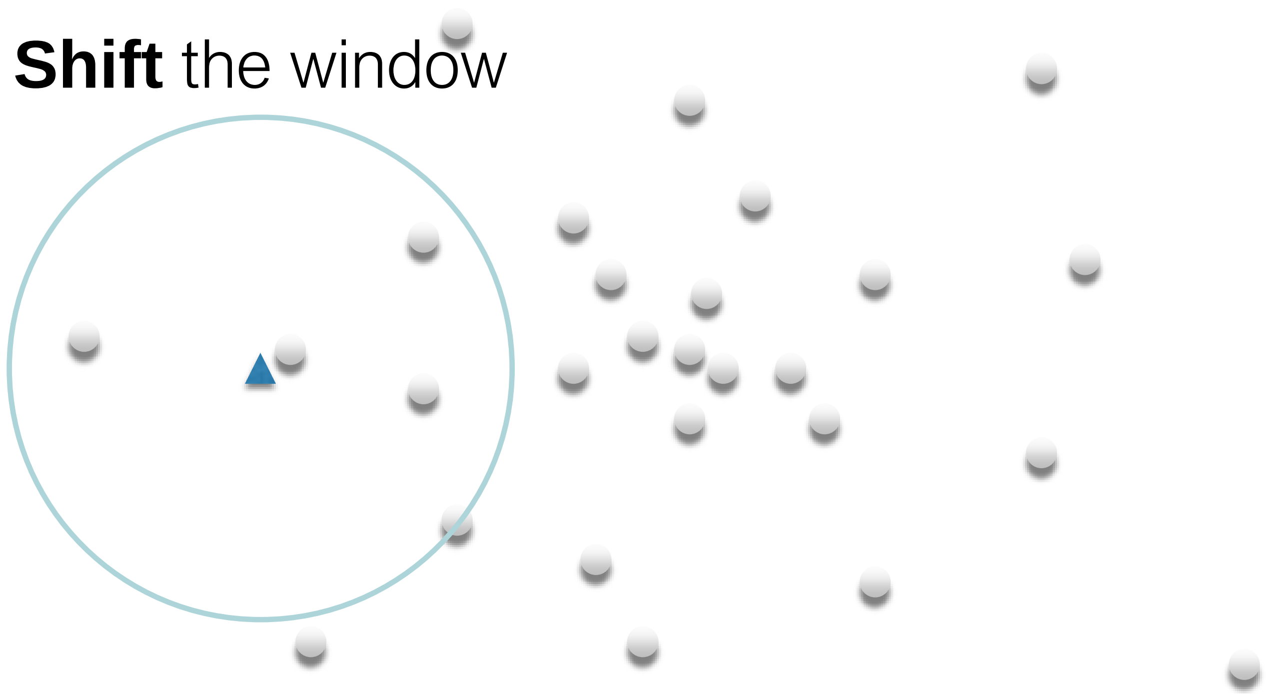


Mean Shift Algorithm

A ‘mode seeking’ algorithm

Fukunaga & Hostetler (1975)

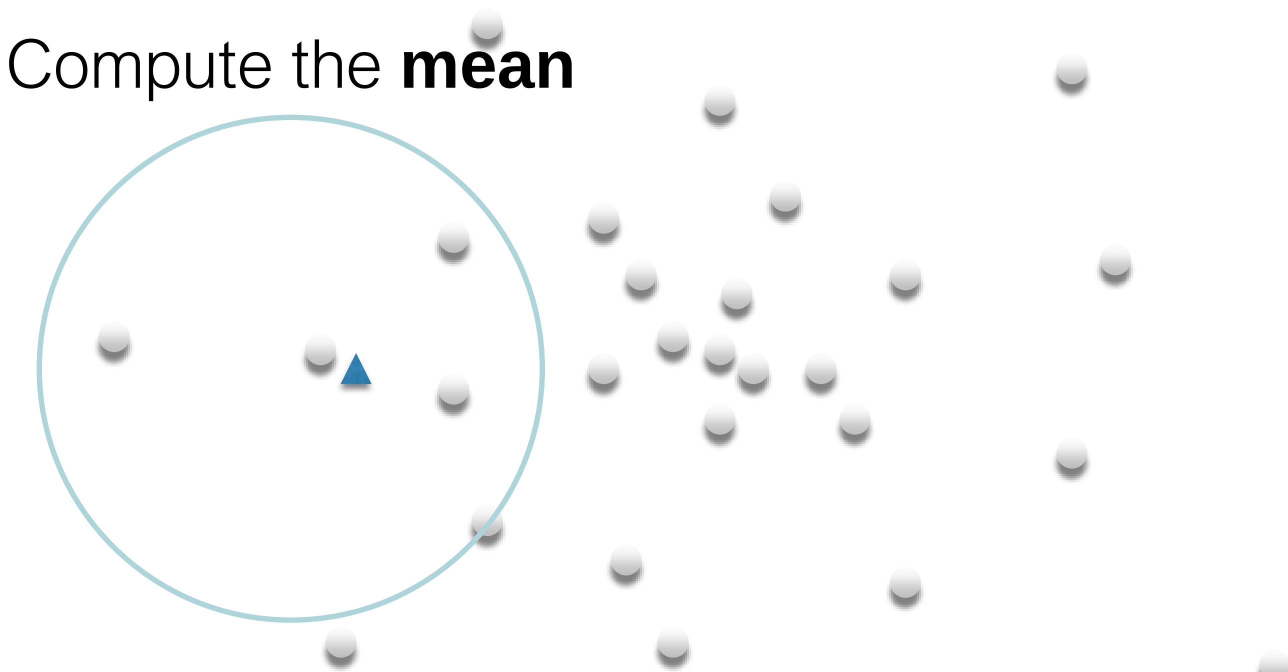
Shift the window



Mean Shift Algorithm

A ‘mode seeking’ algorithm

Fukunaga & Hostetler (1975)

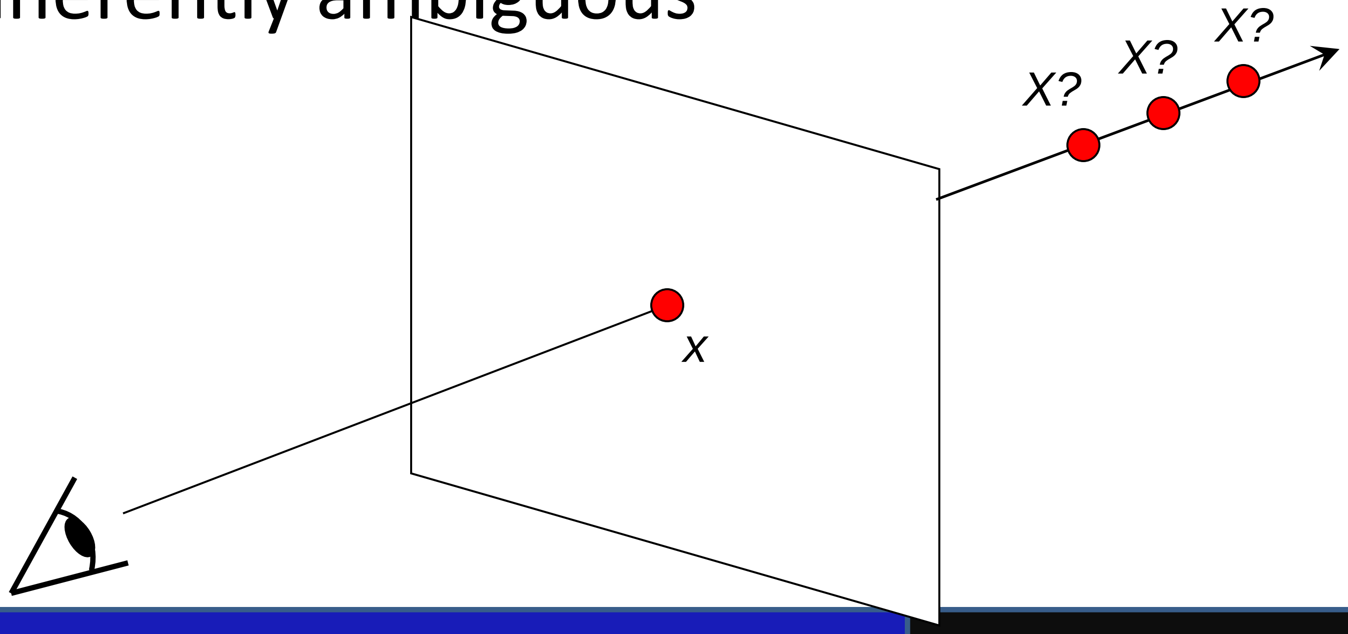


Today’s Lecture

- Motion Analysis
- 3D Vision
- Triangulation Principle
- Stereoscopy

Our goal: Recovery of 3D structure

- We will focus on perspective and motion
- We need *multi-view geometry* because recovery of structure from one image is inherently ambiguous



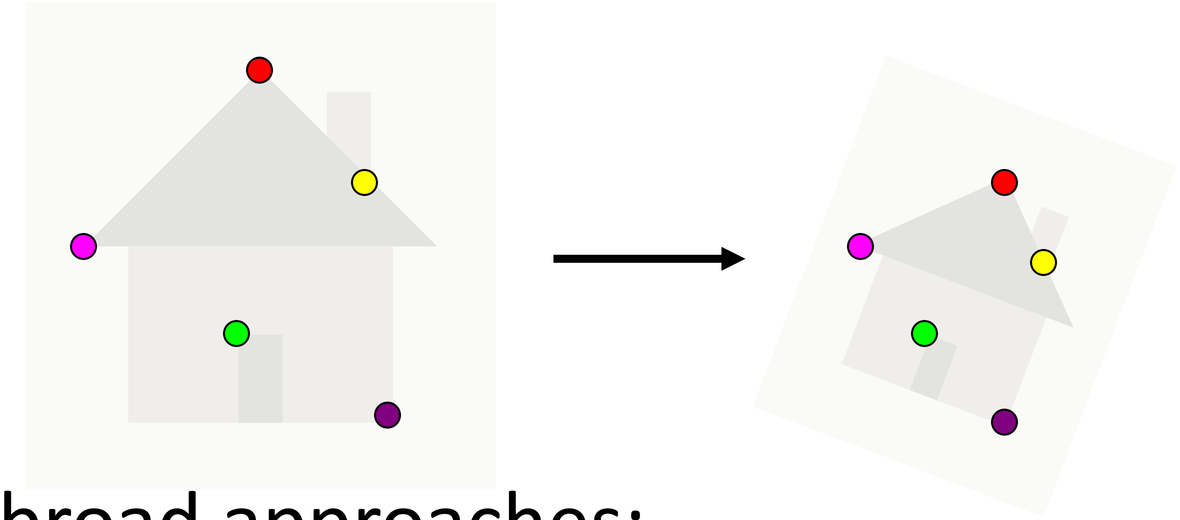
Our goal: Recovery of 3D structure

- We will focus on perspective and motion
- We need *multi-view geometry* because recovery of structure from one image is inherently ambiguous

Our goal: Recovery of 3D structure

- We will focus on perspective and motion
- We need *multi-view geometry* because recovery of structure from one image is inherently ambiguous

Image alignment

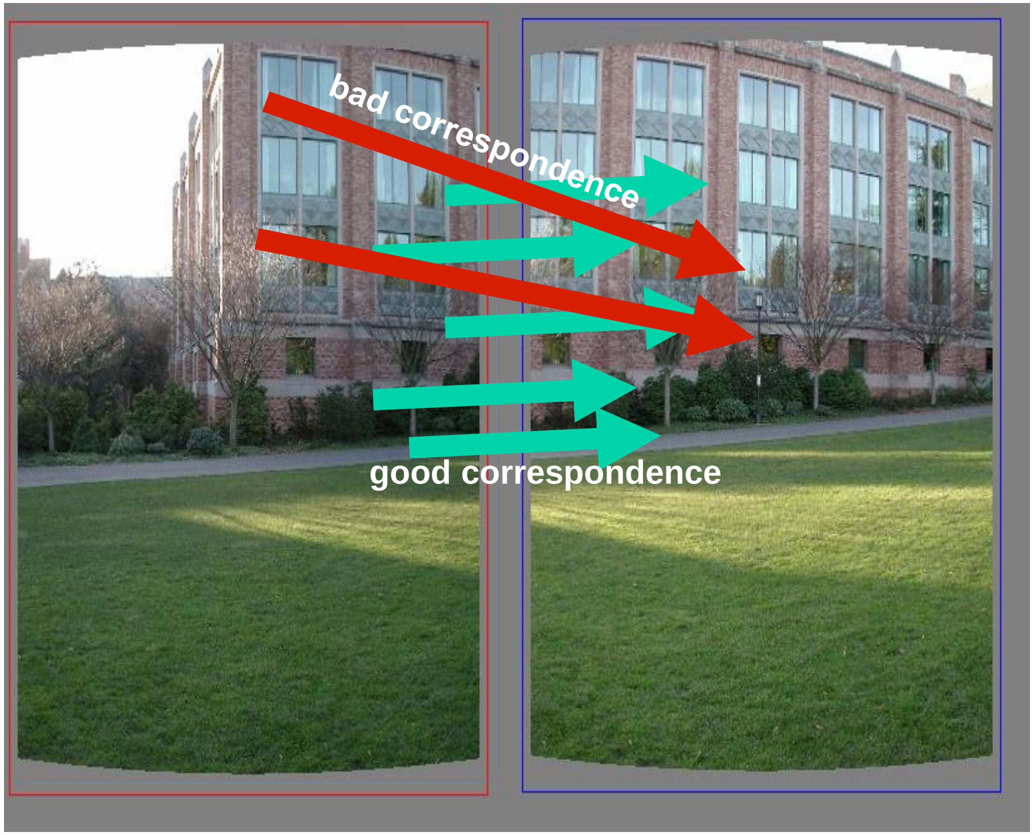


- Two broad approaches:
  - Direct (pixel-based) alignment
    - Search for alignment where most pixels agree
  - Feature-based alignment
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment

Given two images…

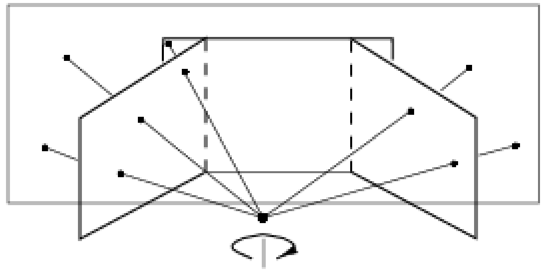
find matching features (e.g., SIFT) and a translation transform

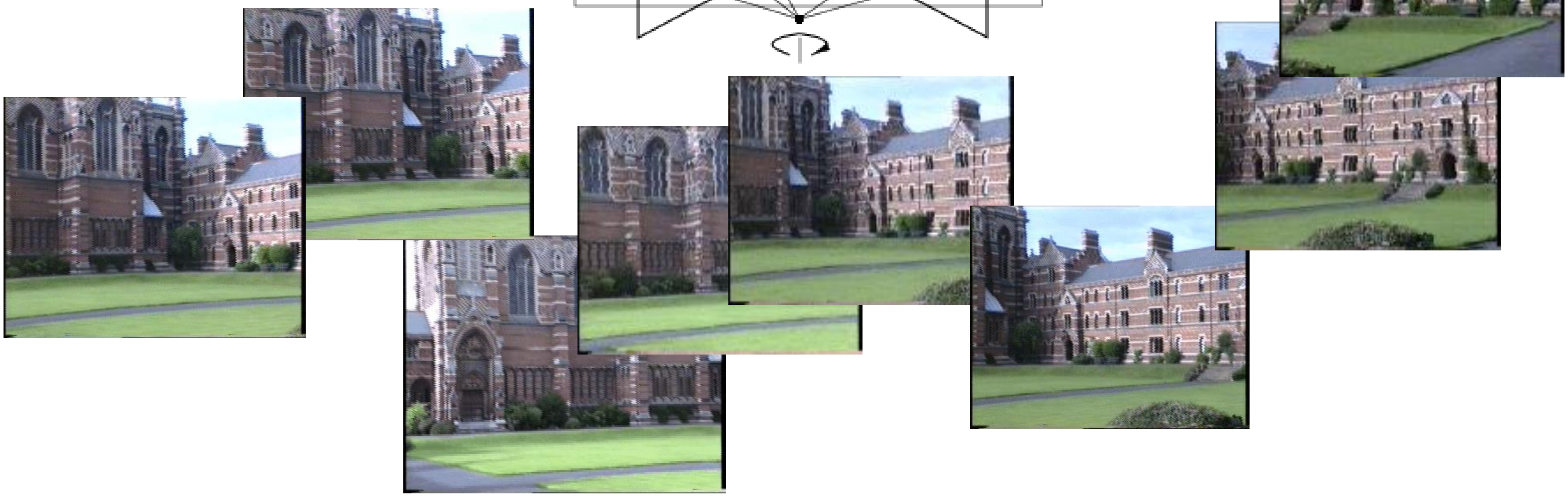
Matched points will usually contain bad correspondences



how should we estimate the transform?

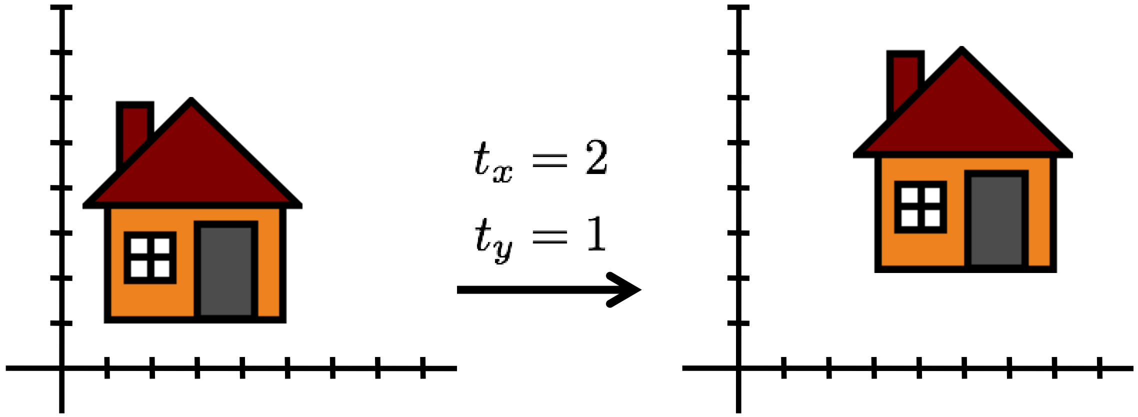
Application: Panorama stitching





2D translation using homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



2D transformations using homogeneous coordinates

Transformations as 3x3 matrices:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translation

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

Matrix composition

Transformations can be combined by matrix multiplication:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

p' = translation(t_x,t_y) rotation(θ) scale(s,s) **p**

Does the multiplication order matter?

Basic Transformations

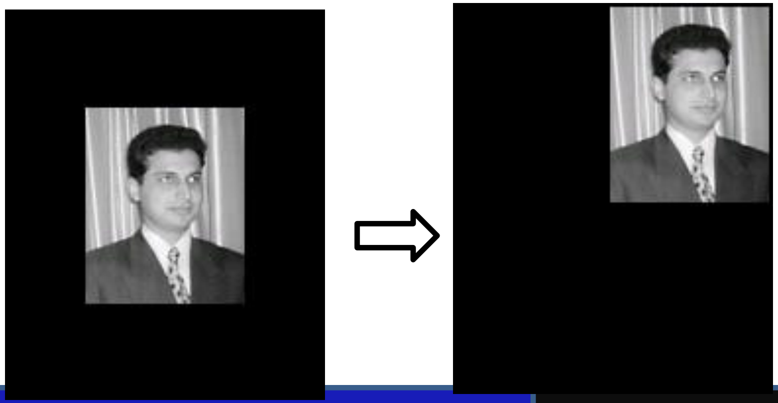
Translation: $(x' = x + x_0,\ \ y' = y + y_0,\ \ z' = z + z_0)$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2D)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(3D)



Basic Transformations

Scaling: $(x' = S_x x,\ \ y' = S_y y,\ \ z' = S_z z)$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2D)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

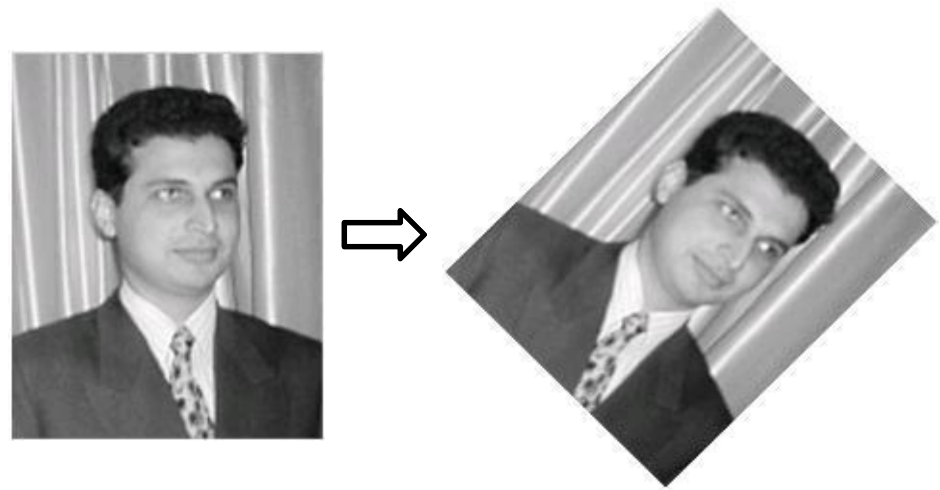
(3D)

Basic Transformations

Rotation (2D):

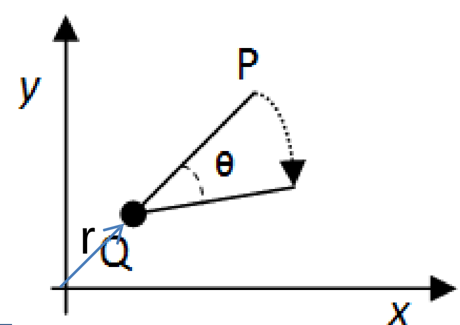
- around origin

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Cos\theta & Sin\theta & 0 \\ -Sin\theta & Cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



- around an arbitrary point (not origin)

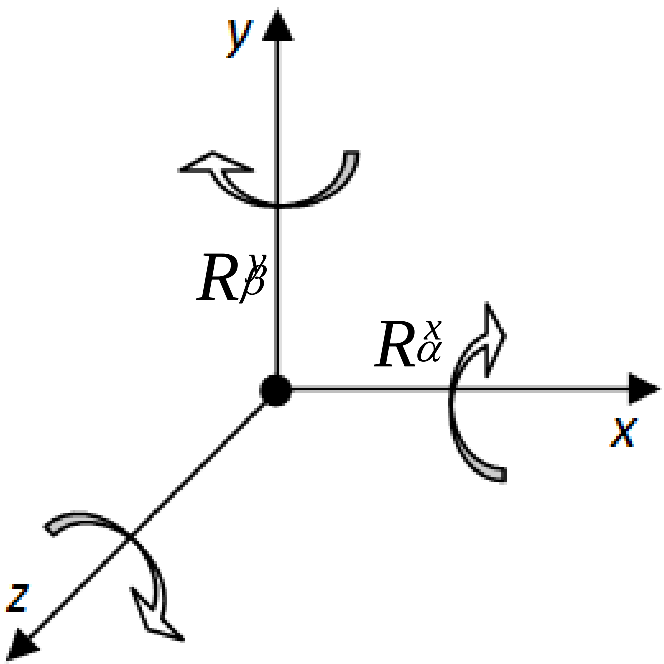
$$\Rightarrow T_r p(R_\theta \, T_{-r} p)$$



Rotation around point Q

Basic Transformations

Rotation (3D):



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{Cos}\,\alpha & \mathrm{Sin}\,\alpha & 0 \\ 0 & -\mathrm{Sin}\,\alpha & \mathrm{Cos}\,\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

around x-axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{Cos}\,\beta & 0 & -\mathrm{Sin}\,\beta & 0 \\ 0 & 1 & 0 & 0 \\ \mathrm{Sin}\,\beta & 0 & \mathrm{Cos}\,\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

around y-axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{Cos}\,\theta & \mathrm{Sin}\,\theta & 0 & 0 \\ -\mathrm{Sin}\,\theta & \mathrm{Cos}\,\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

around z-axis

Inverse Transformations

Inverse Translation:

- sign is changed

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2D)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(3D)

Inverse Transformations

Inverse Scaling:

- scaling values get inverted

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2D)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(3D)

Inverse Transformations

Inverse Rotation:

- angle sign changed

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Cos(-\theta) & Sin(-\theta) & 0 \\ -Sin(-\theta) & Cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2D)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Cos(-\beta) & 0 & -Sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ Sin(-\beta) & 0 & Cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

around y-axis

(3D)

Sample Questions

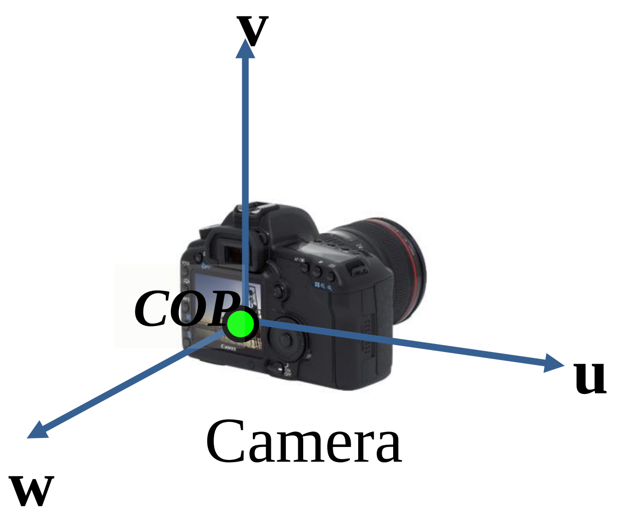
Q1. What is the rotation matrix for an object rotation of 30 deg around the z-axis, followed by 60 deg around the x-axis, and followed by a rotation of 90 deg around the y-axis. All rotations are counter clockwise.

Q2. Consider a 3D point $[2\ 1\ 2]^T$. What would be coordinates of the point after applying the following composite transformation:
(i) CCW rotation of 90 degrees around the x-axis,
(ii) translation by $dx = -2$, $dy = 1$, $dz = 1$, and
(iii) scaling by $sx = 1$, $sy = 2$ and $sz = 0.5$.

Practice Question

Q1. A unit cube with vertices at (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0) and (1,1,1) is scaled using the scale factors Sx=2, Sy=3 and Sz=4. What are the vertices of the transformed figure. Ans: (0,0,0), (0,0,4), (0,3,0), (0,3,4), (2,0,0), (2,0,4), (2,3,0) and (2,3,4)

A Tale of Two Coordinate Systems



Two important coordinate systems:

1. *World* coordinate system
2. *Camera* coordinate system

Perspective Transformation and Imaging Process

- Perspective Transformation is also called imaging transformation
- approximation of the image formation process
- projects 3D points onto a 2D camera plane

(x,y) ⇨ Camera coordinate system

(X,Y,Z) ⇨ World coordinate system (aligned with camera coordinate system)

The camera as a coordinate transformation

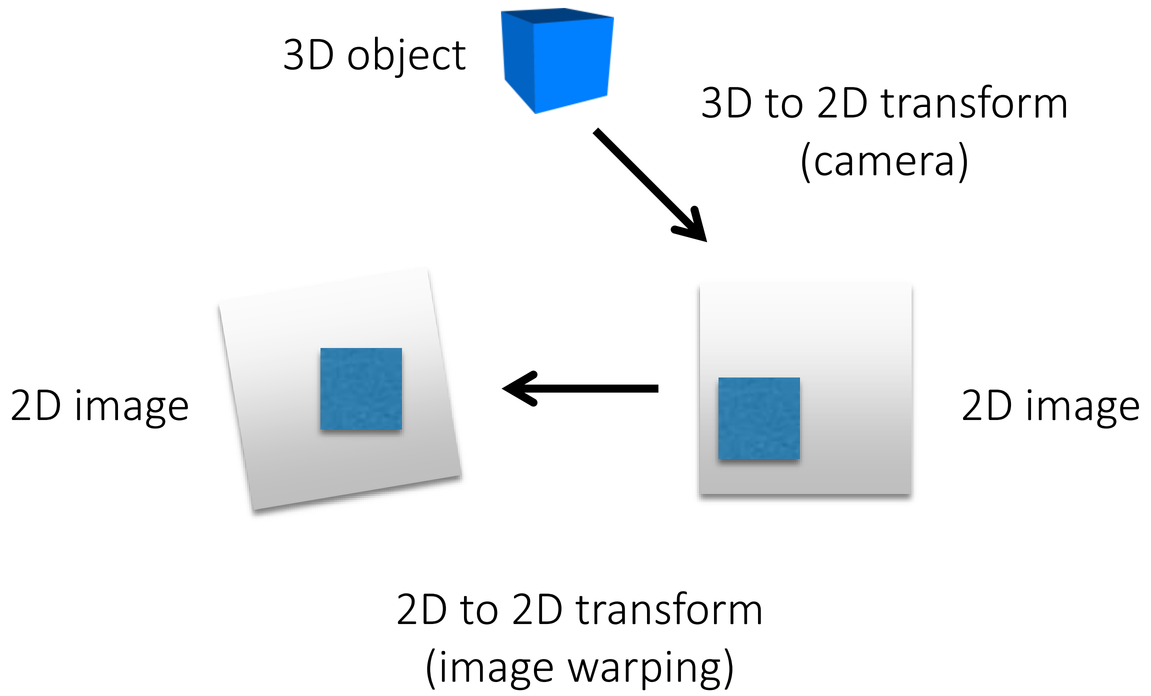
A camera is a mapping

from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

A camera is a mapping

from:

the 3D world

to:

a 2D image

homogeneous coordinates

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

2D image point

camera matrix

3D world point

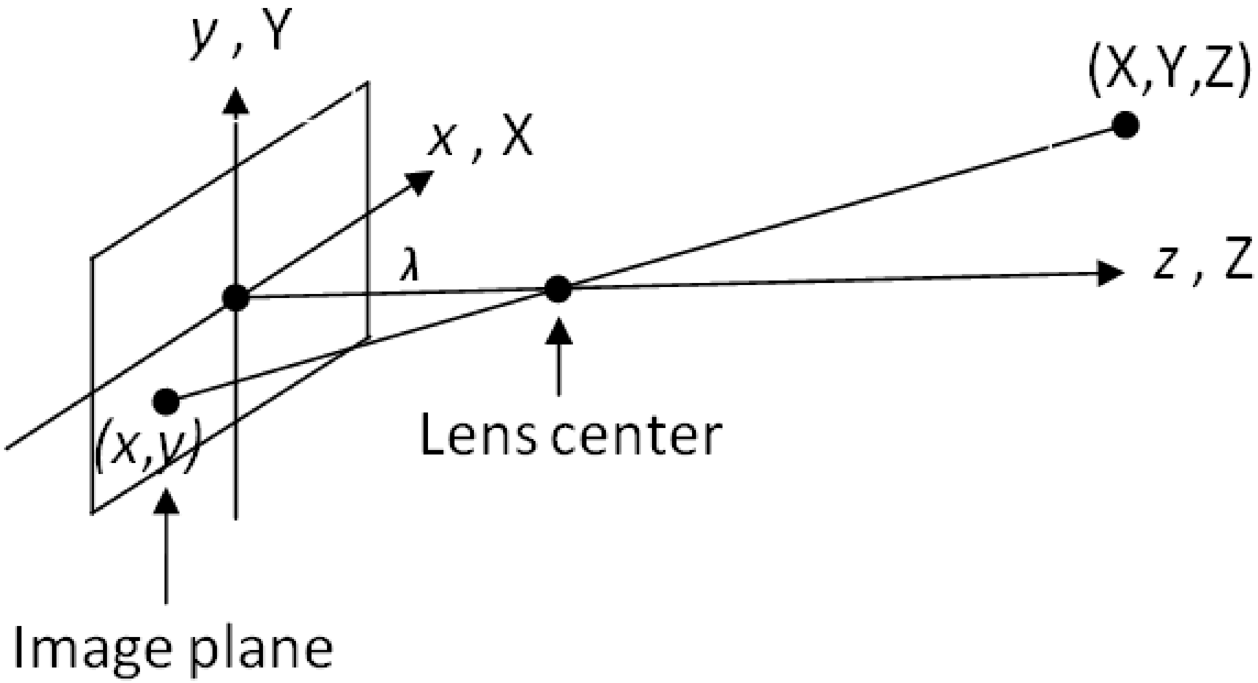
What are the dimensions of each variable?

The camera as a coordinate transformation

$$\boldsymbol{x} = \mathbf{PX}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image coordinates 3 x 1	camera matrix 3 x 4	homogeneous world coordinates 4 x 1



$$x = \lambda X/(\lambda - Z)$$

$$y = \lambda Y/(\lambda - Z)$$

where λ is the focal length

Cartesian Coordinate System

(Euclidean Geometry)

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\Longrightarrow$$

Homogeneous Coordinate System

(Projective Geometry)

$$W_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

$$\therefore \; W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} W_{h1}/W_{h4} \\ W_{h2}/W_{h4} \\ W_{h3}/W_{h4} \end{bmatrix}$$

Image (Camera) Homogeneous Coordinates:

$$\boxed{C_h = P\,W_h}$$ where P is perspective transform

for $P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix}$, $C_h = P\,W_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ (-kZ/\lambda) + k \end{bmatrix}$

$$C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} C_{h1}/C_{h4} \\ C_{h2}/C_{h4} \\ C_{h3}/C_{h4} \end{bmatrix}$$

$$\therefore \quad x = \lambda X/(\lambda - Z), \text{and}$$

$$y = \lambda Y/(\lambda - Z)$$

Inverse Perspective Transformation

- Maps an image point back to 3D:

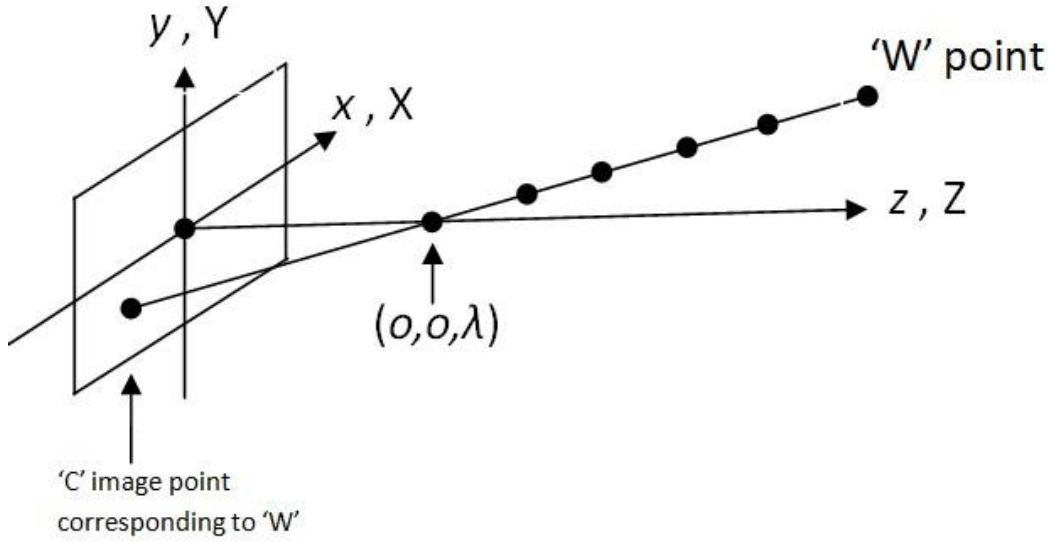
$$\boxed{W_h = P^{-1} C_h}\text{, where } P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\lambda & 1 \end{bmatrix},$$

- For an image point (x_0,y_0), the above inverse transformation ends up giving Z=0 for 3D point:

$$\Rightarrow W_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix}$$

Therefor e, $W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}$ is an unexpected result which gives Z=0 (many to one mapping problem)

- C=(x_0, y_0) is mapping of points on a straight line passing through (x_0, y_0, 0) and (0, 0, λ)



- Eqs of straight line in world coordinates:

$$X = \frac{x_0}{\lambda}(\lambda - Z), Y = \frac{y_0}{\lambda}(\lambda - Z)$$

- Inverse Perspective transformation formulated using 'z' component of 'C_h' as a free variable:

$$W_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix} = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ (kz/\lambda) + k \end{bmatrix}$$

$$\Rightarrow \quad W = \begin{bmatrix} \lambda x_0 / (\lambda + z) \\ \lambda y_0 / (\lambda + z) \\ \lambda z / (\lambda + z) \end{bmatrix}$$

which suggests that additional information is required to find 3D world point

- We can find out X and Y only and require additional information to find out Z so that point in 3D world coordinates is exactly known from x_0, y_0 in image plane

SO FAR:

Both coordinate systems were aligned

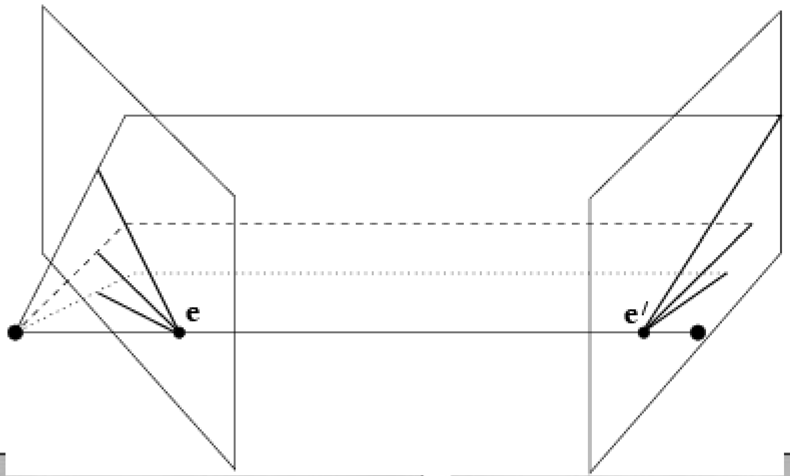
NEXT:

Imaging geometry where world coordinate system and camera coordinate system are not aligned

Today’s Lecture

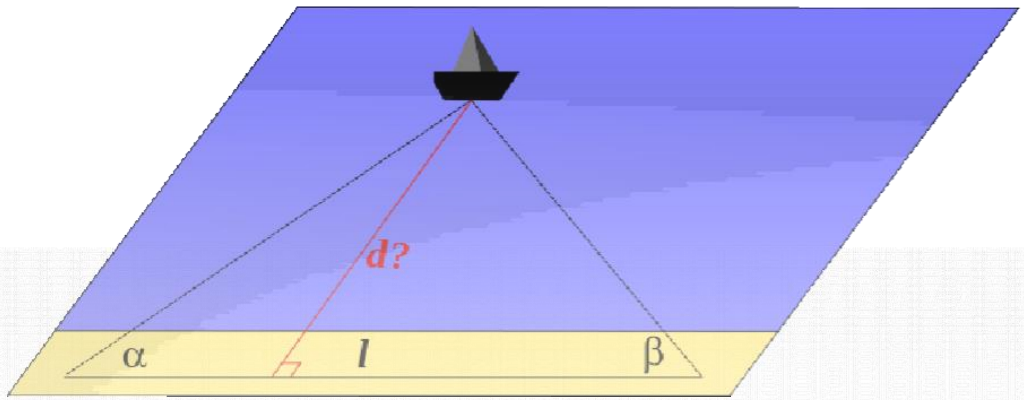
- Motion Analysis
- 3D Vision
- Triangulation Principle
- Stereoscopy

Example: Converging cameras



Triangulation[1]

- Process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline



Calculation

$$\ell = \frac{d}{\tan\alpha} + \frac{d}{\tan\beta}$$

Therefore

$$d = \ell \,/\, \left(\frac{1}{\tan\alpha} + \frac{1}{\tan\beta}\right)$$

Using the trigonometric identities tan α = sin α / cos α and sin(α + β) = sin α cos β + cos α sin β, this is equivalent to:

$$d = \frac{\ell \sin\alpha \sin\beta}{\sin(\alpha + \beta)}$$

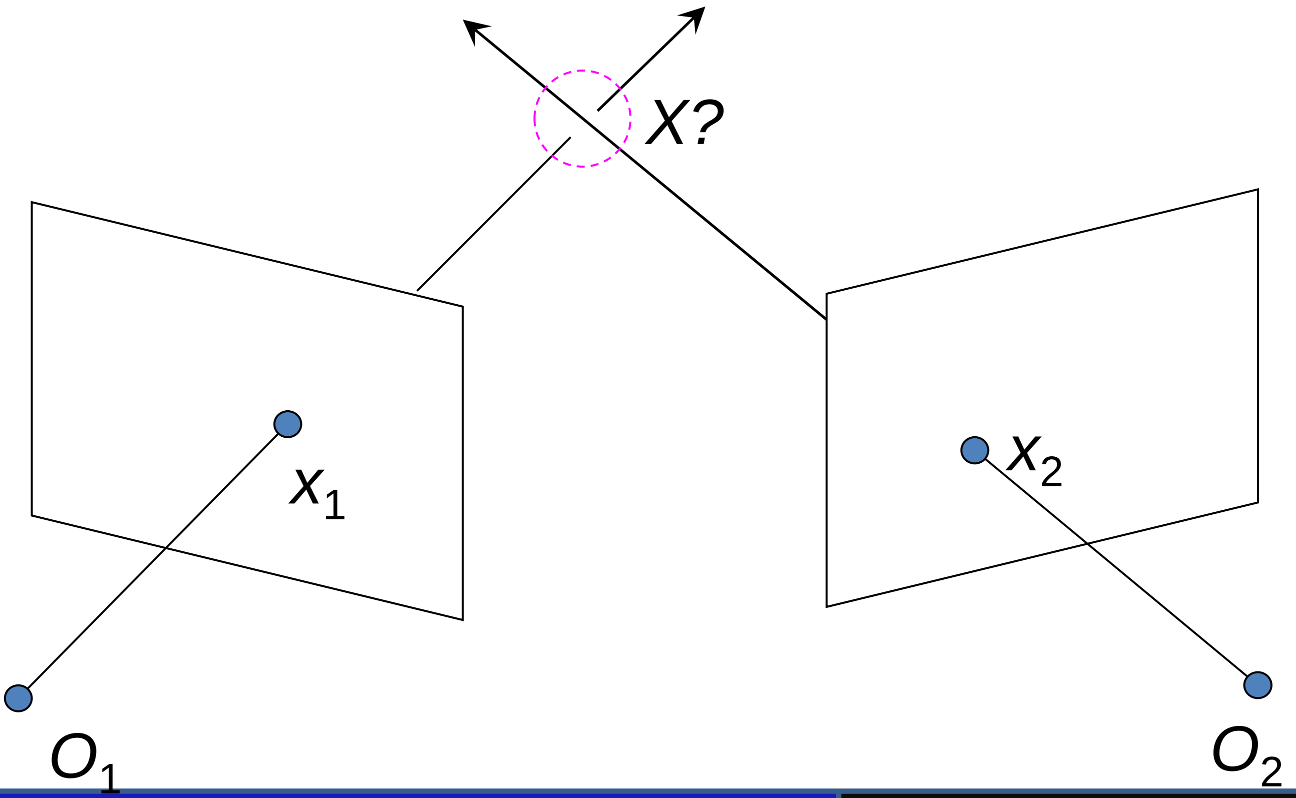
[1][http://en.wikipedia.org/wiki/Triangulation]

Multi View Geometry

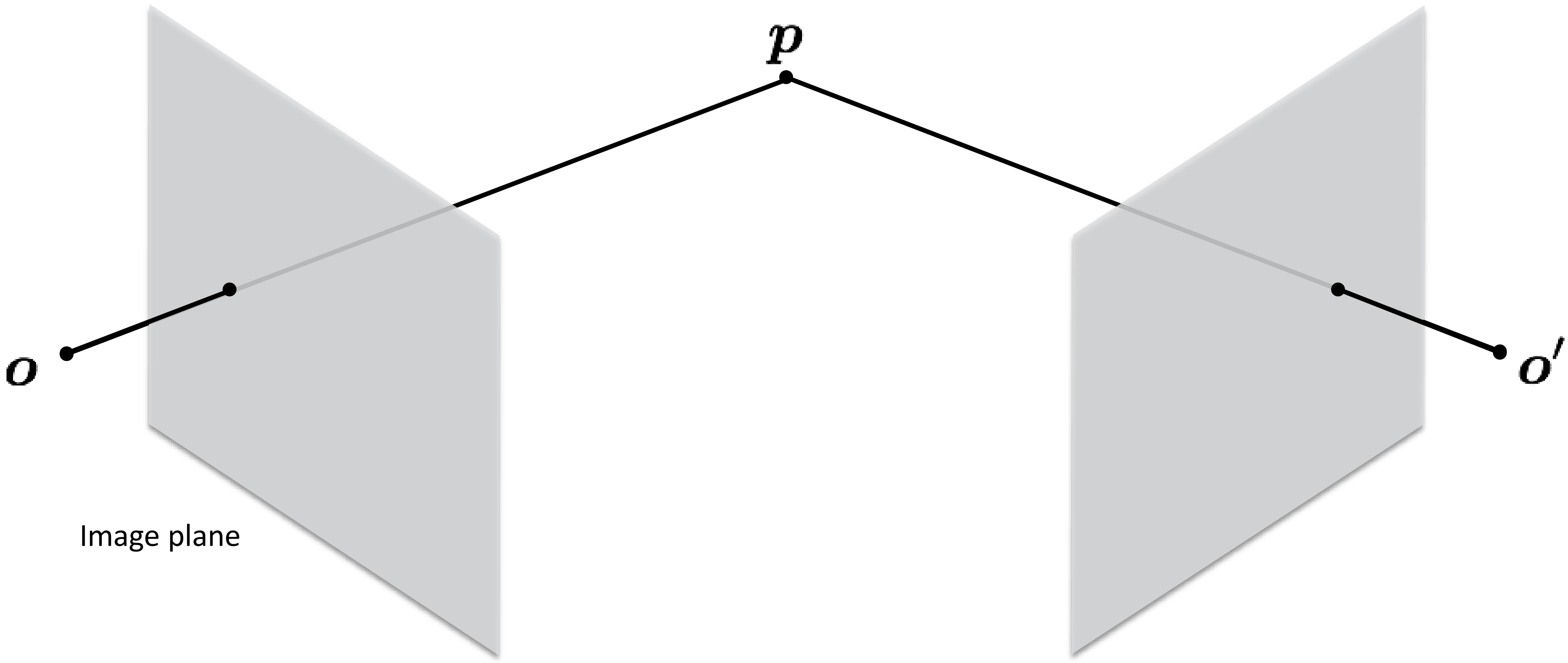
- Depth of a scene point along the corresponding projection ray is not directly accessible in a single image
- With at least two pictures, depth can be measured through **triangulation**
  - Most animals have two eyes, move their head when looking for friend or foe
  - Motivation for equipping an autonomous robot with a stereo or motion analysis system
- Binocular stereo vision— first image of any point must lie in the plane formed by its second image and the optical centers of the two cameras: *epipolar constraint* (can be represented by a 3x3 matrix)

Triangulation

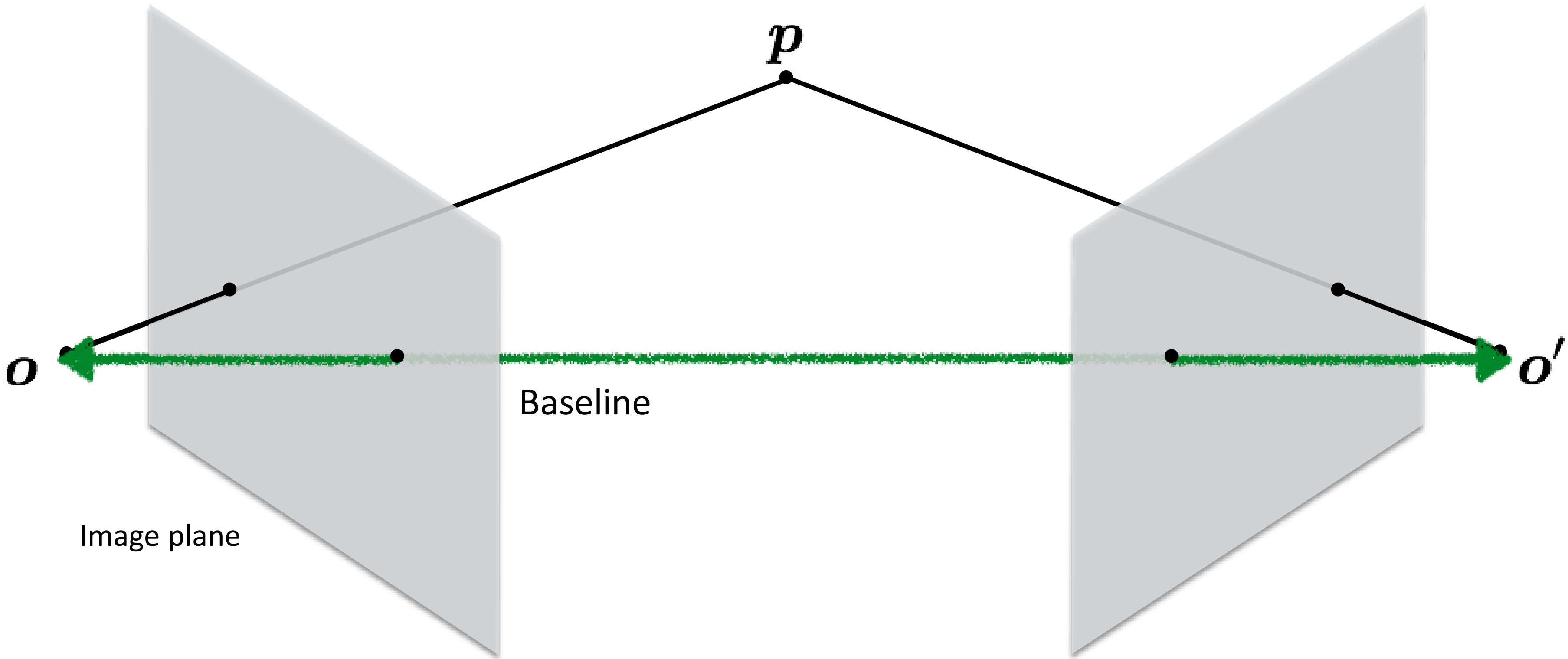
- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



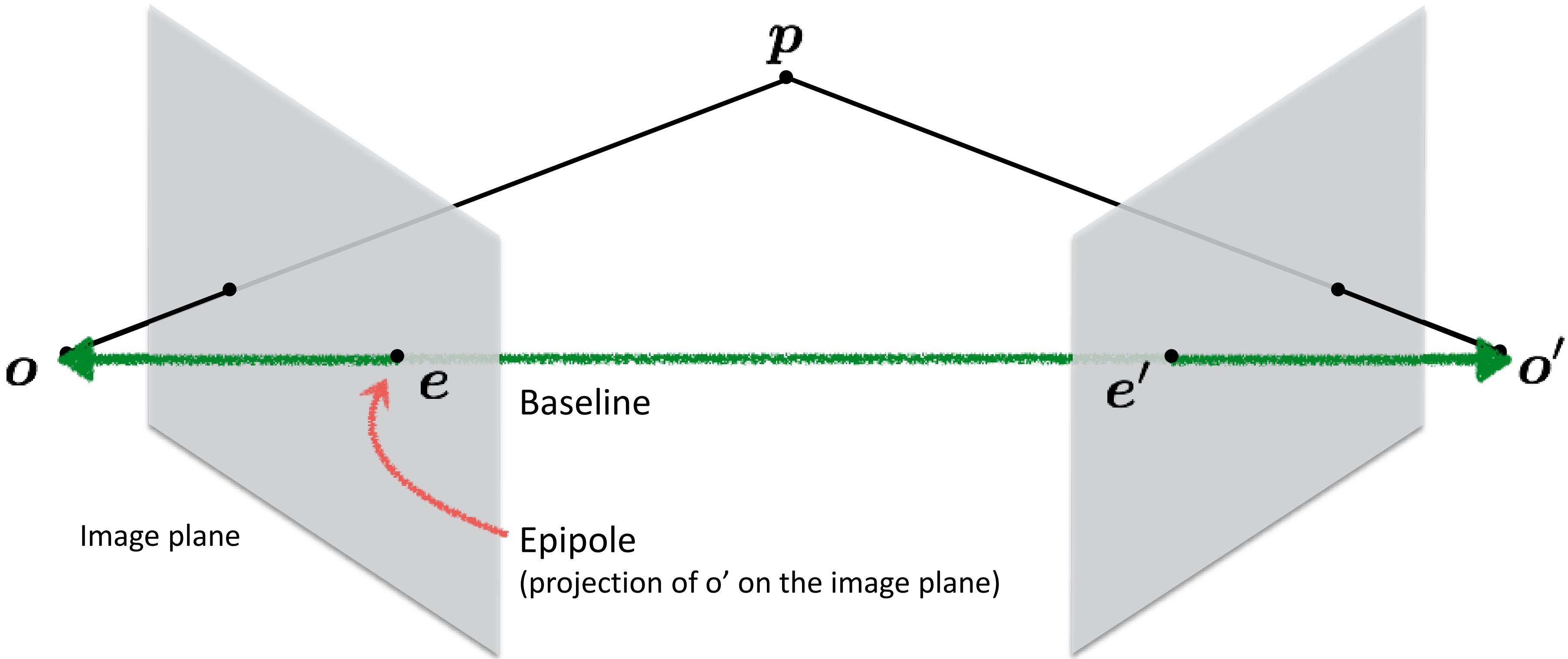
Epipolar geometry



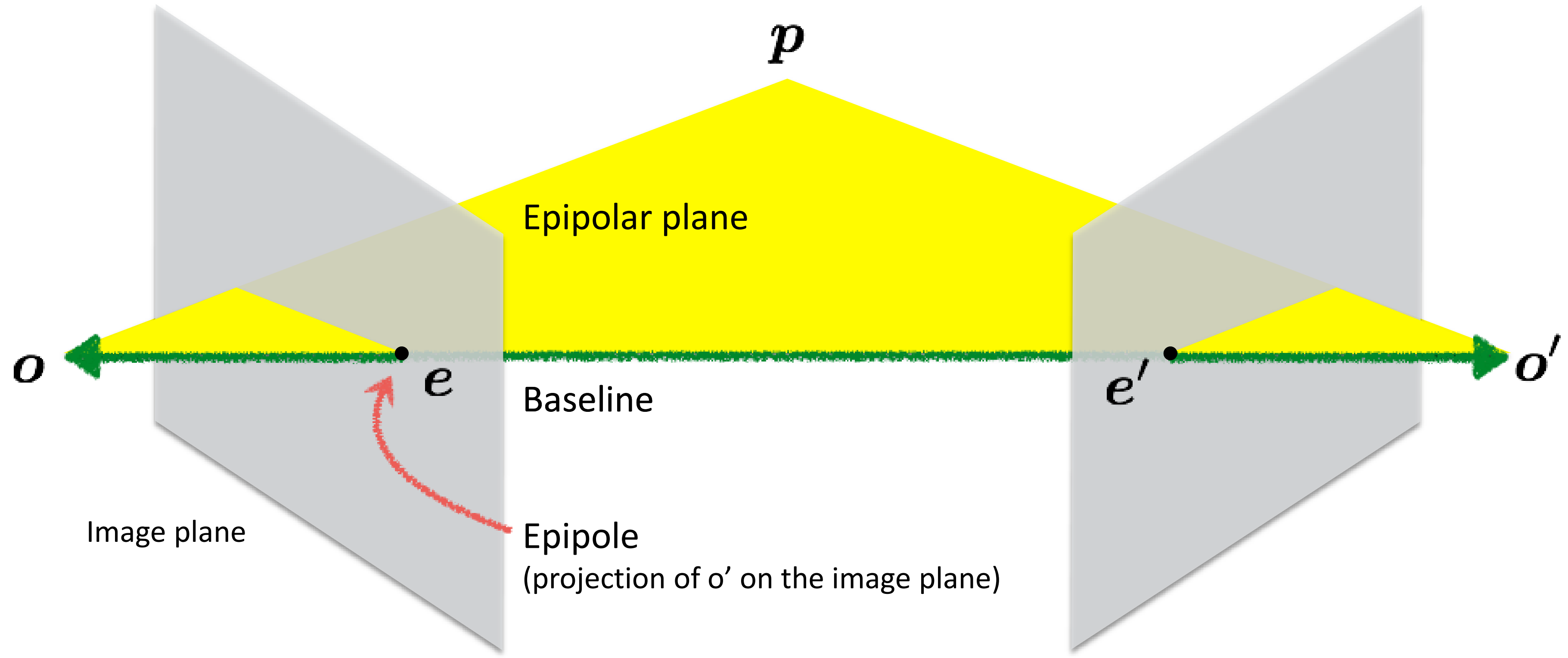
Epipolar geometry



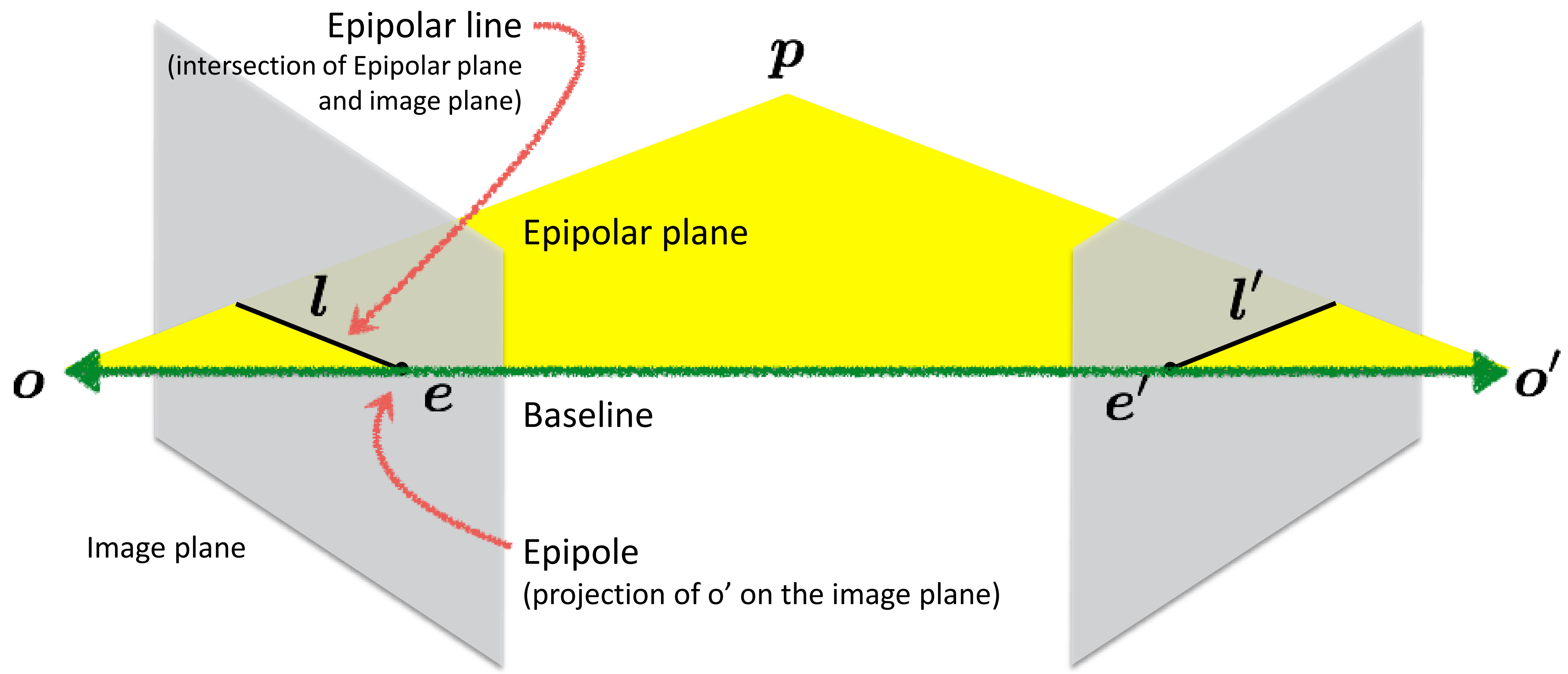
Epipolar geometry



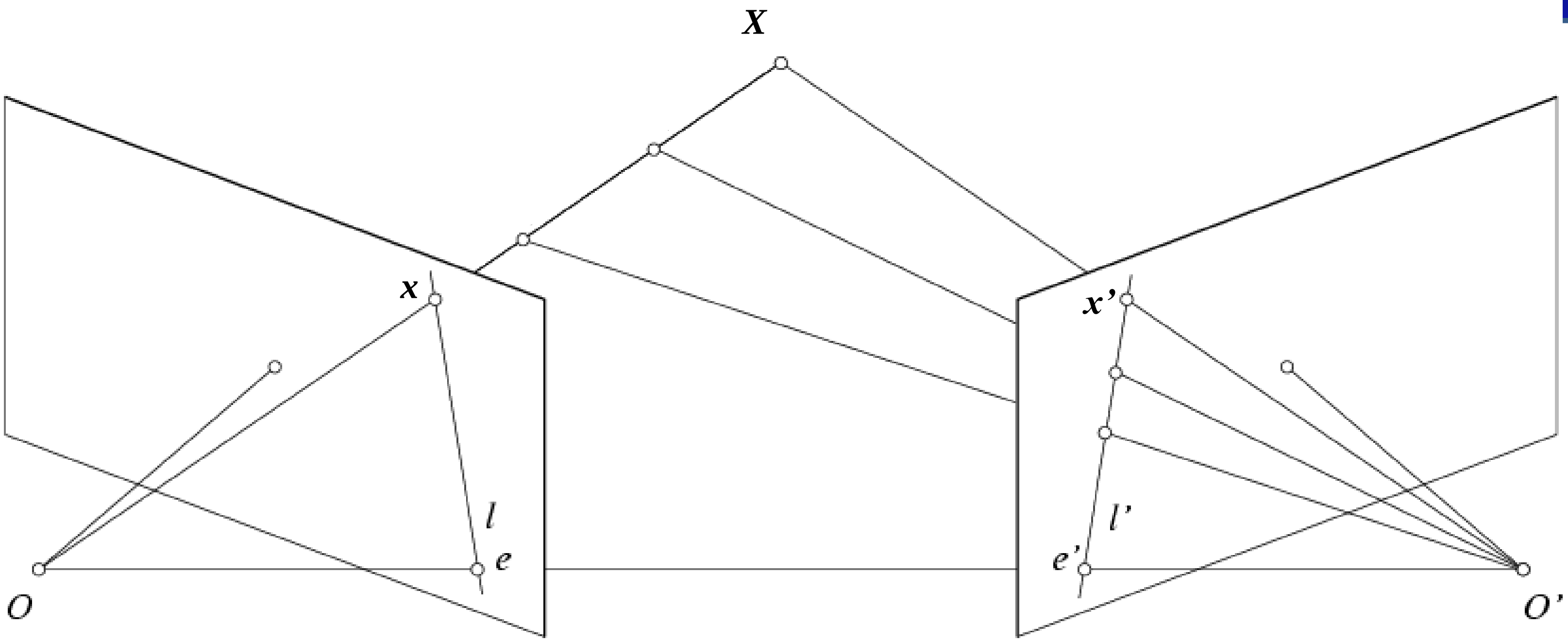
Epipolar geometry



Epipolar geometry

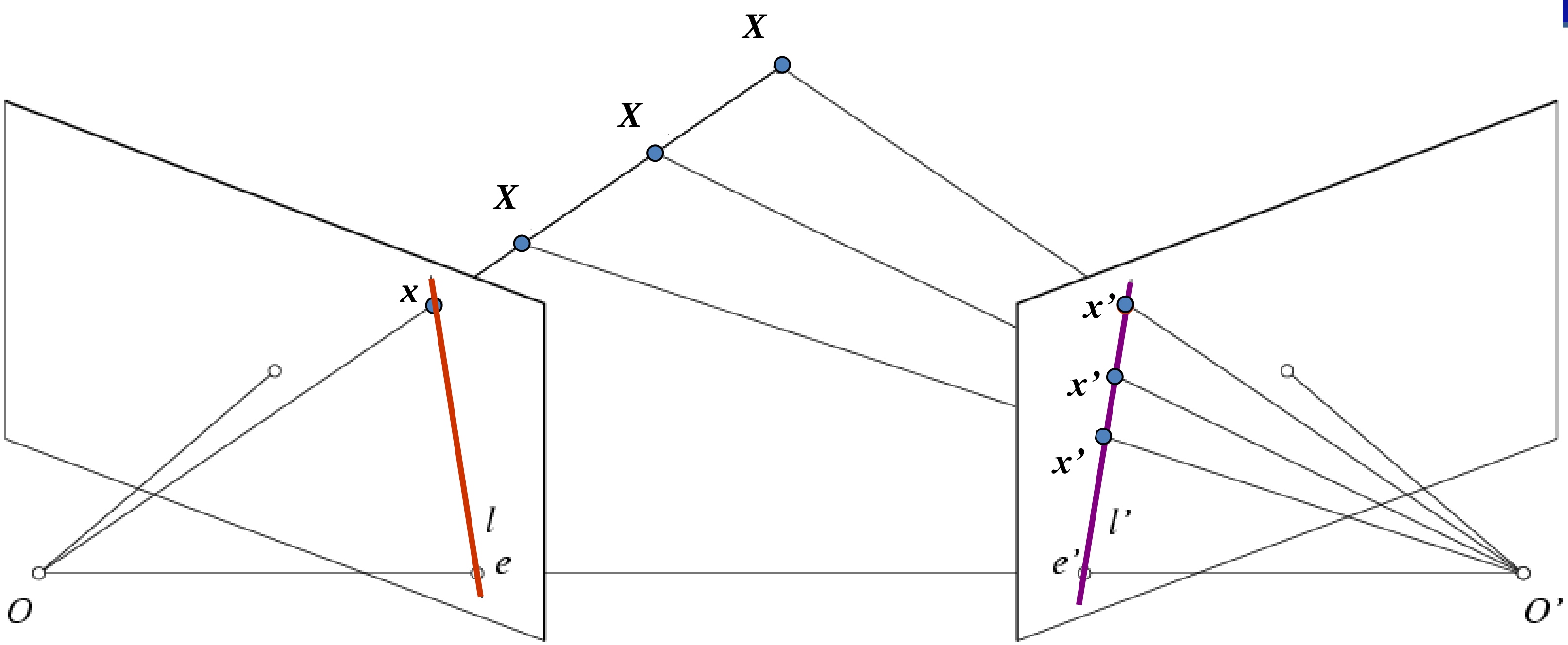


Epipolar constraint



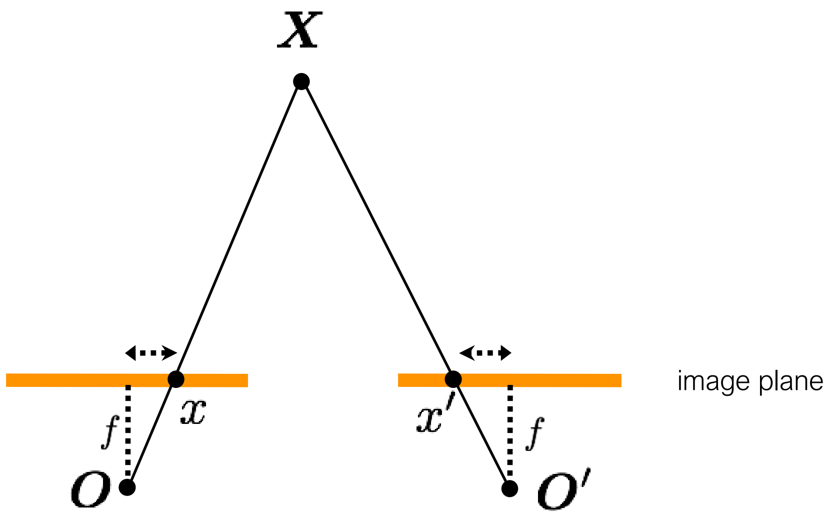
- If we observe a point x in one image, where can the corresponding point x' be in the other image?

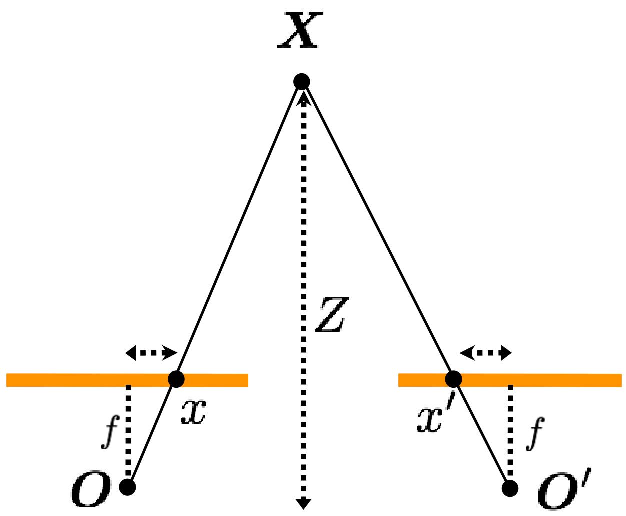
Epipolar constraint



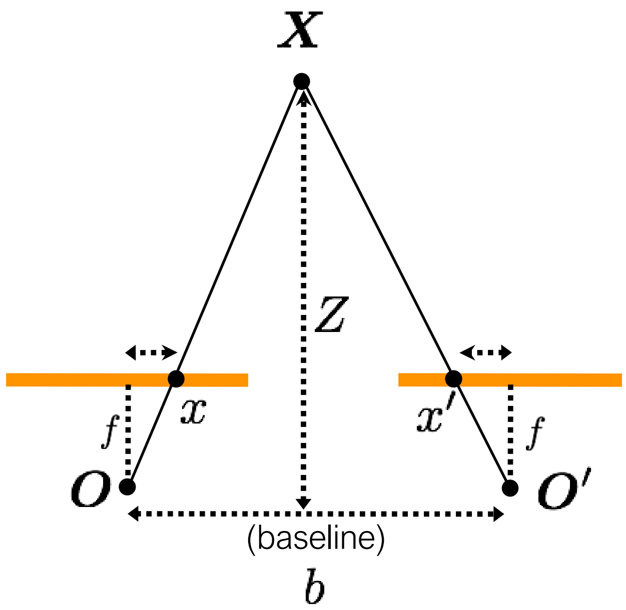
- Potential matches for *x* have to lie on the corresponding epipolar line *l'*.

- Potential matches for *x'* have to lie on the corresponding epipolar line *l*.


X
x
x'
f
f
O
O'
image plane

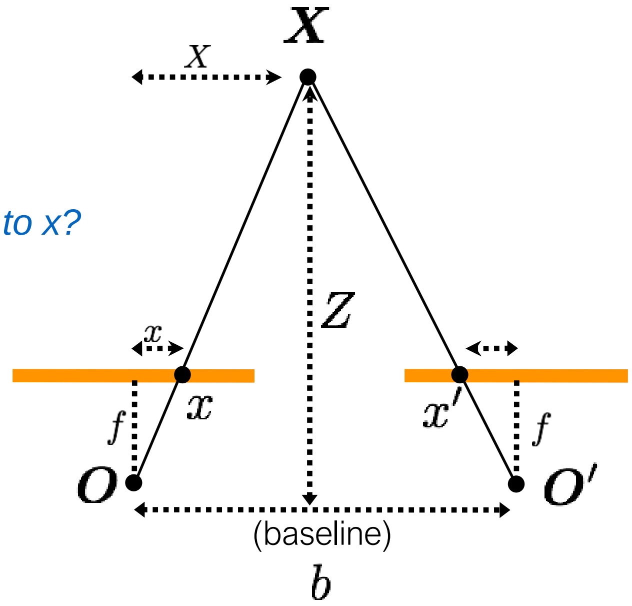


X
Z
x
x'
f
f
O
O'

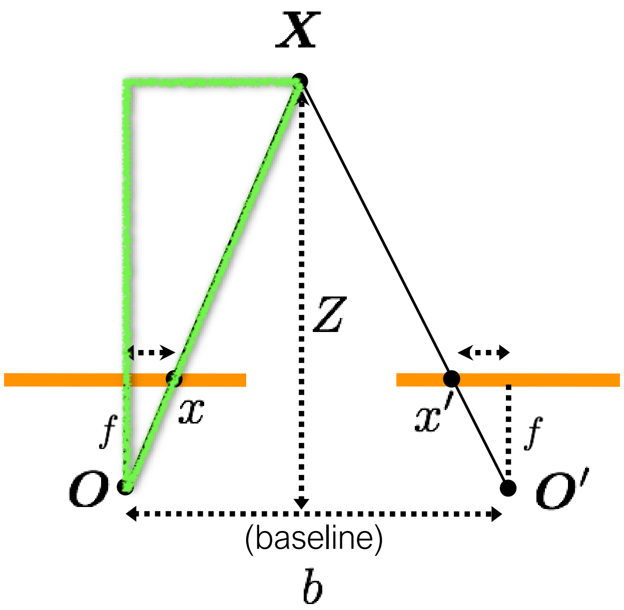


$\boldsymbol{X}$
Z
x
x'
f
f
$\boldsymbol{O}$
$\boldsymbol{O'}$
(baseline)
b

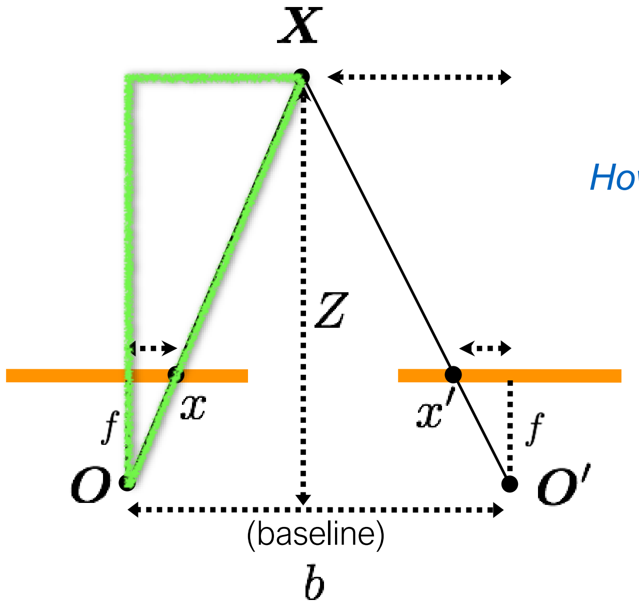
How is X related to x?



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

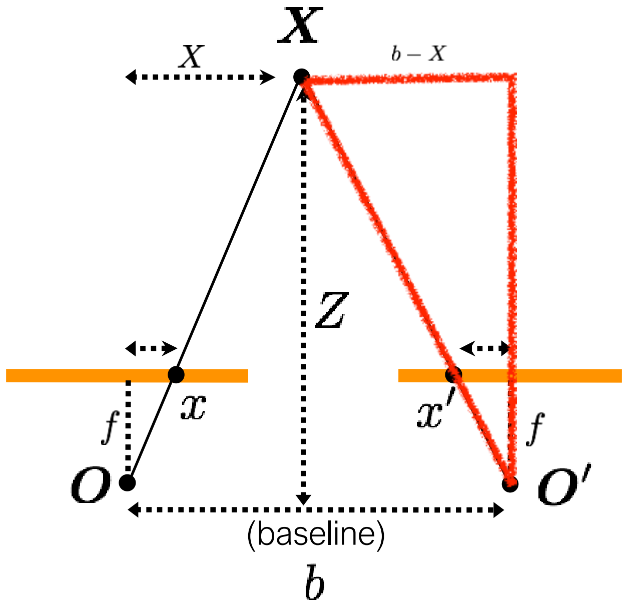


$$\frac{X}{Z} = \frac{x}{f}$$



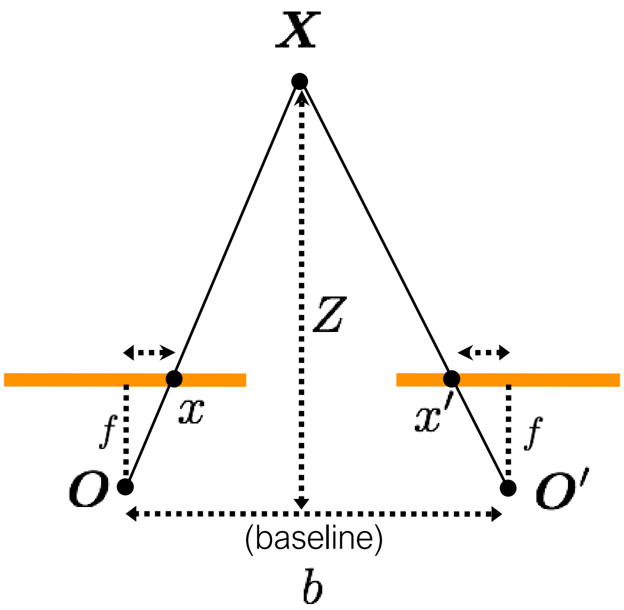
How is X related to x'?

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



$$\frac{b - X}{Z} = \frac{x'}{f}$$

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



$$\frac{b-X}{Z} = \frac{x'}{f}$$

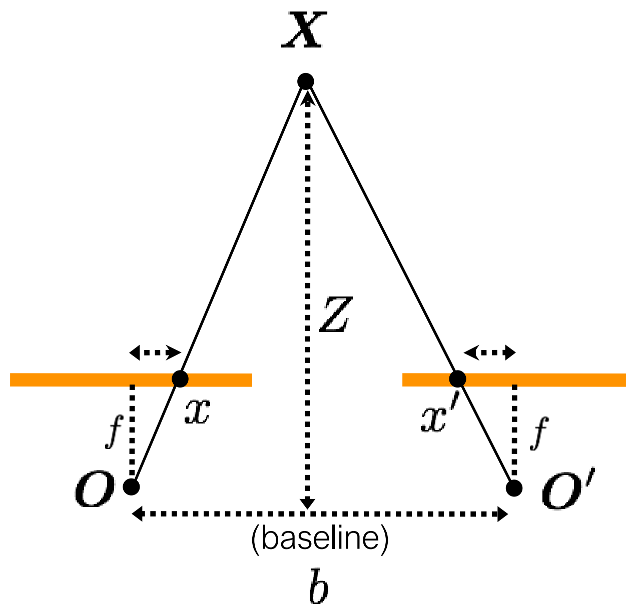
Disparity

$$d = x - x'$$

(wrt to camera origin of image plane)

$$= \frac{bf}{Z}$$

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



$$\frac{b - X}{Z} = \frac{x'}{f}$$

Disparity

$$d = x - x'$$
$$= \frac{bf}{Z}$$

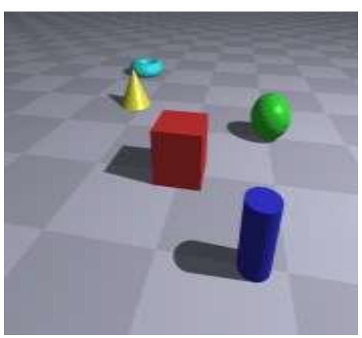
inversely proportional
to depth

Today's Lecture

- Motion Analysis
- 3D Vision
- Triangulation Principle
- Stereoscopy

Stereoscopy

- **Stereoscopy** (also called **stereoscopic** or **3-D imaging**) refers to a technique for creating or enhancing the illusion of depth in an image by presenting two offset images separately to the left and right eye of the viewer. Both of these 2-D offset images are then combined in the brain to give the perception of 3-D depth.



Stereoscopy

Stereoscopy Vs. Normal Image

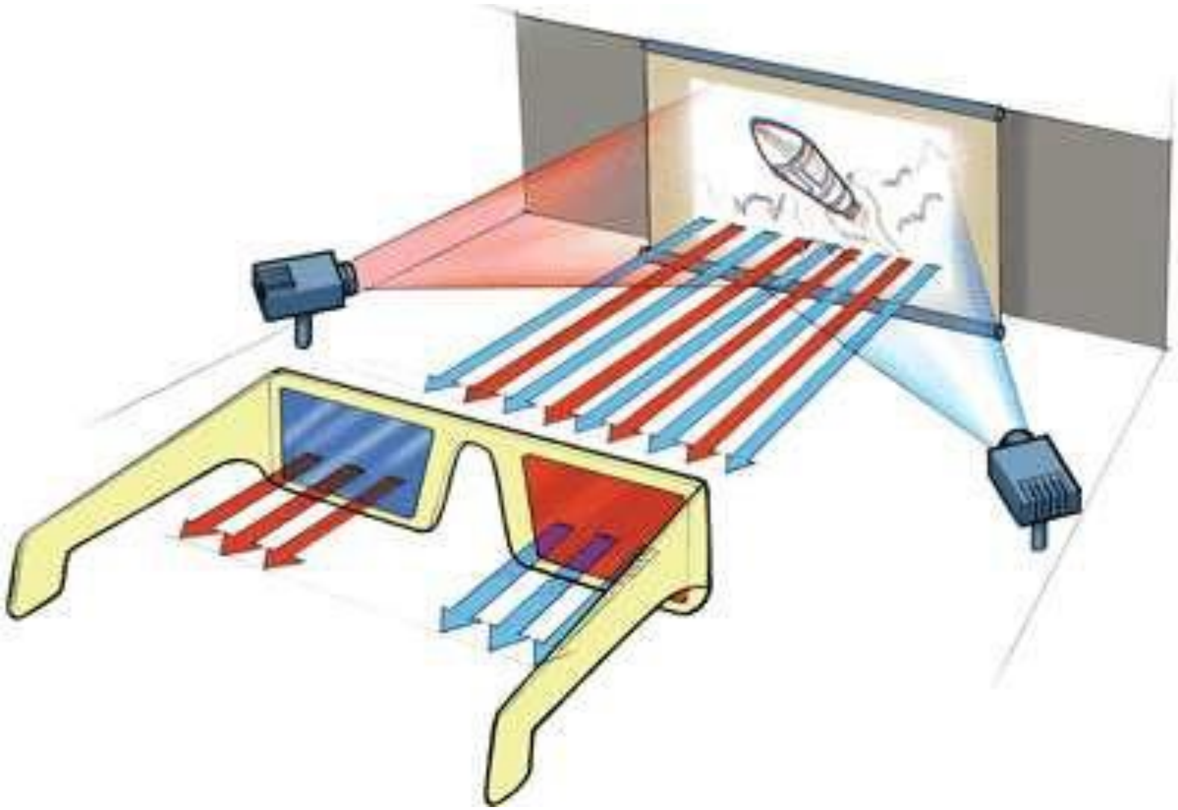
- In a normal image both of our eye sees the same picture

- But in a stereoscopic Image our two eyes sees two slightly different images, and that's how illusion of three-dimensional depth is created.

Modern 3D Technologies

- Modern 3D technology is divided into several procedure :
- **With lenses**:
  - Anaglyphic 3D (with passive red-cyan lenses)
  - Polarization 3D (with passive polarized lenses)
  - Alternate-frame sequencing (with active shutter lenses)
  - Head-mounted display (with a separate display positioned in front of each eye, and lenses used primarily to relax eye focus)
- **Without lenses**: Auto stereoscopic displays, sometimes referred to commercially as **Auto 3D**.

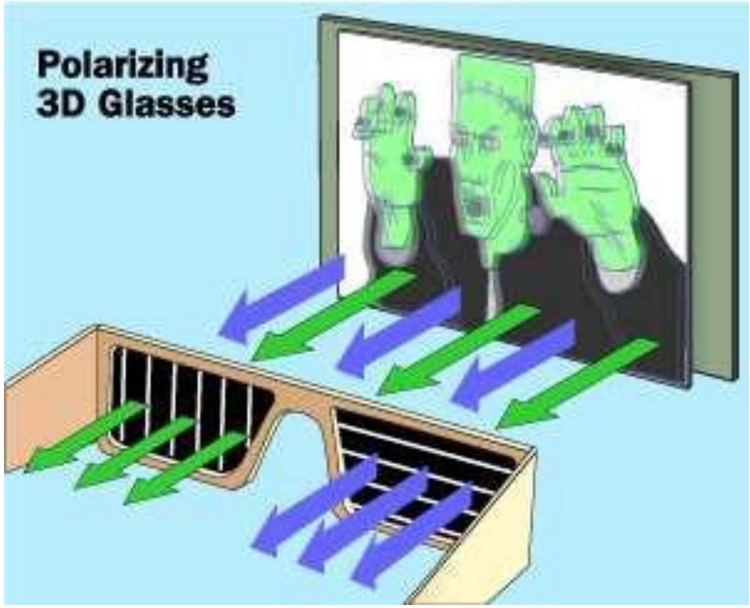
Anaglyph

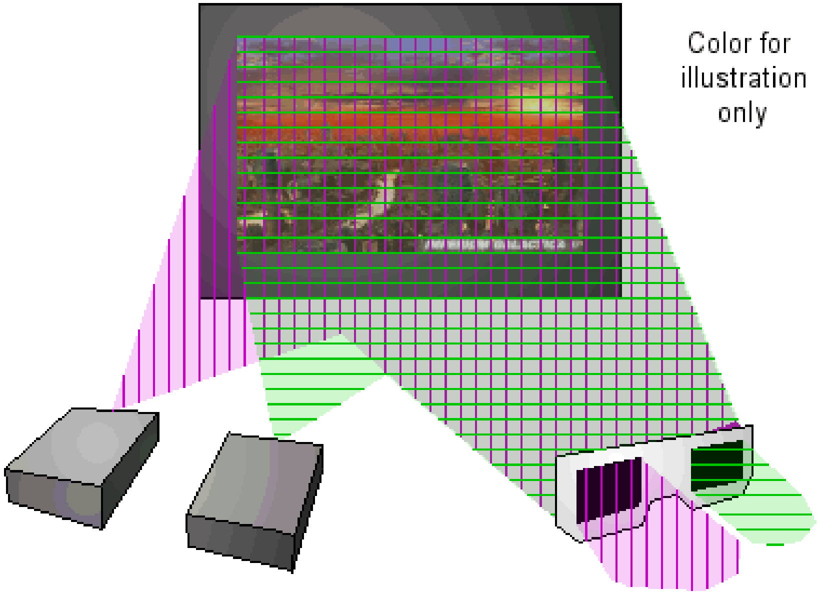


Disadvantage

- Color depth is absent due to colored filter glass.

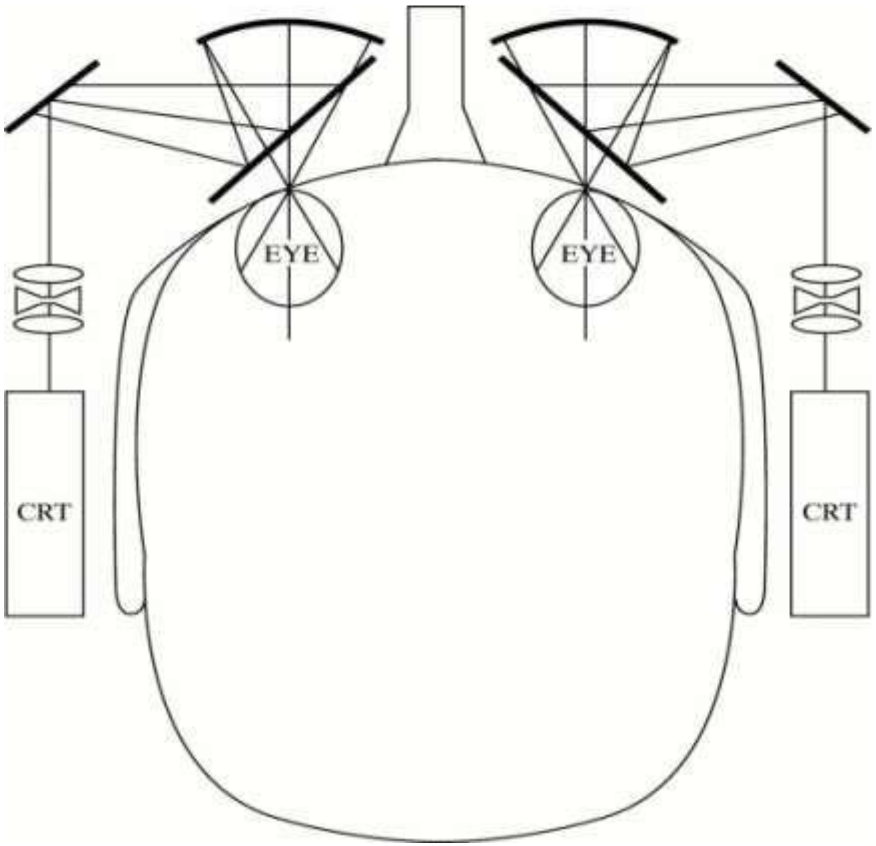
3D Polarization



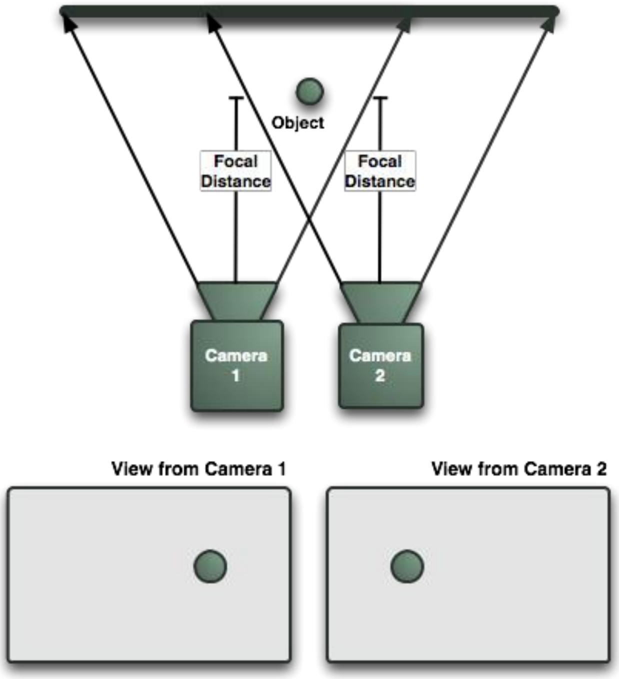


- Used in modern 3D Movie theater.

Head Mounted Display



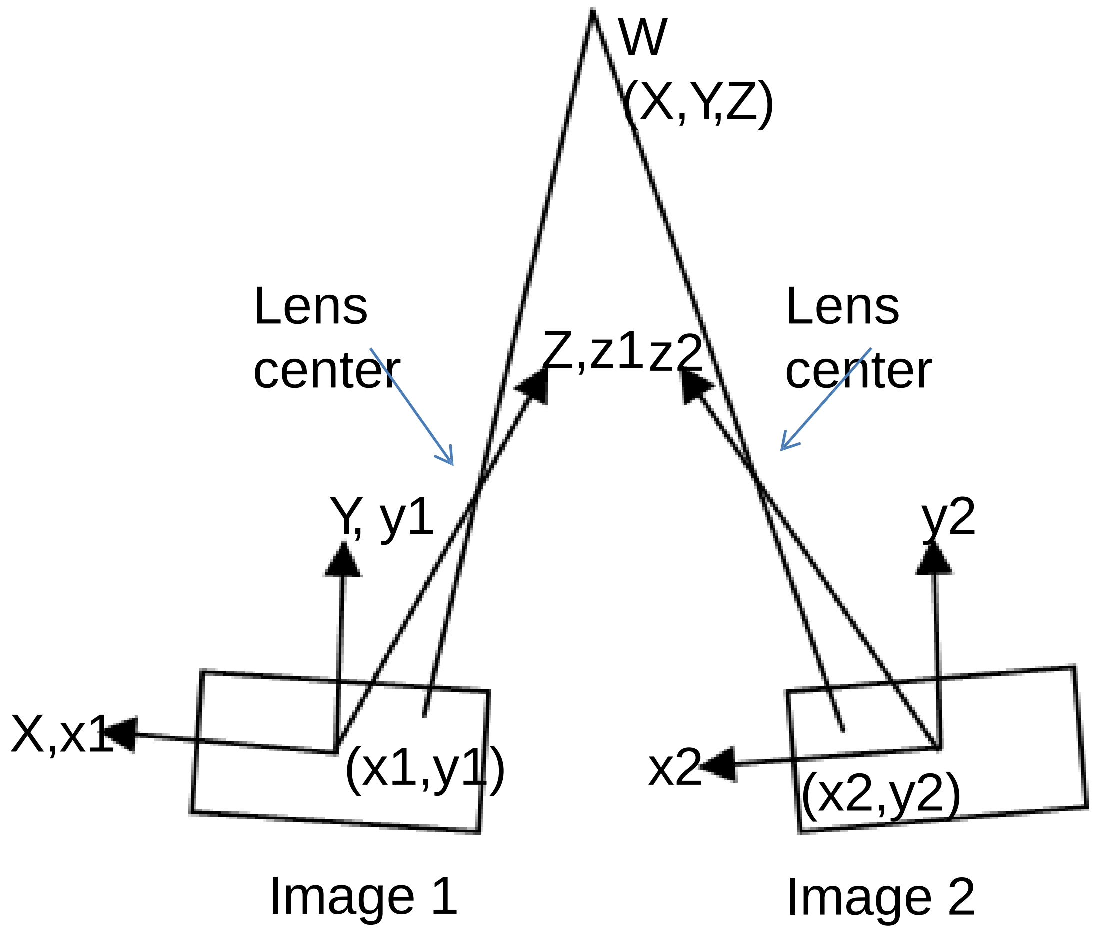
Stereoscopic Camera



Sample 3D Photograph

Stereo Imaging

To find all coordinates of 3D world point corresponding to an image point, we need another camera



Stereo Imaging Model

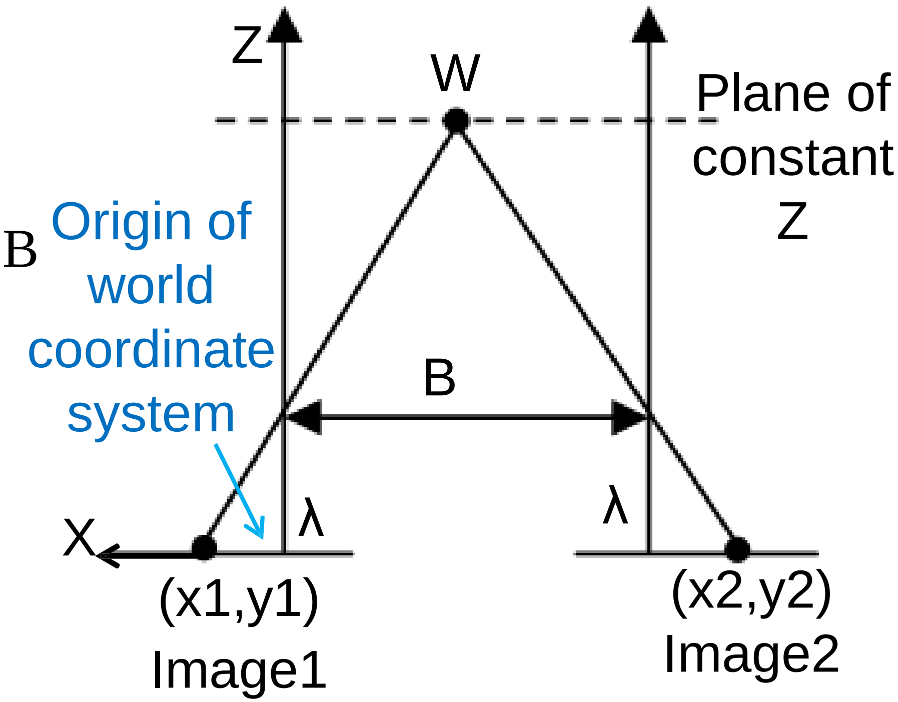
From Perspective Transform:

$$X_1 = \frac{x_1}{\lambda}(\lambda - Z), X_2 = \frac{x_2}{\lambda}(\lambda - Z)$$

$$X_2 = X_1 + B \Rightarrow \frac{x_2}{\lambda}(\lambda - Z) = \frac{x_1}{\lambda}(\lambda - Z) + B$$

$\Downarrow$

$$\boxed{Z = \lambda - \frac{\lambda B}{(x_2 - x_1)} \quad \& \quad X = \frac{x_1}{\lambda}(\lambda - Z) \quad Y = \frac{y_1}{\lambda}(\lambda - Z)}$$



Therefore, by having the knowledge of focal length (λ), displacement (B) and disparity (x2-x1), we can find out all coordinates of 3D world point

References

- Some Slide material has been taken from Dr. M. Usman Akram Computer Vision Lectures
- CSCI 1430: Introduction to Computer Vision by James Tompkin
- Statistical Pattern Recognition: A Review – A.K Jain et al., PAMI (22) 2000
- Pattern Recognition and Analysis Course – A.K. Jain, MSU
- *Pattern Classification"* by Duda et al., John Wiley & Sons.
- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Machine Vision: Automated Visual Inspection and Robot Vision", David Vernon, Prentice Hall, 1991
- www.eu.aibo.com/
- Advances in Human Computer Interaction, Shane Pinder, InTech, Austria, October 2008
- Computer Vision A modern Approach by Frosyth
- http://www.cs.cmu.edu/~16385/s18/
- Introduction to Machine Learning, Alphaydin
- Statistical Pattern Recognition: A Review – A.K Jain et al., PAMI (22) 2000
- Pattern Recognition and Analysis Course – A.K. Jain, MSU
- *Pattern Classification"* by Duda et al., John Wiley & Sons.
- http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html
- Some Material adopted from Dr. Adam Prugel-Bennett Dr. Andrew Ng and Dr. Aman ullah's Slides