

---

# Geometric Model of Camera

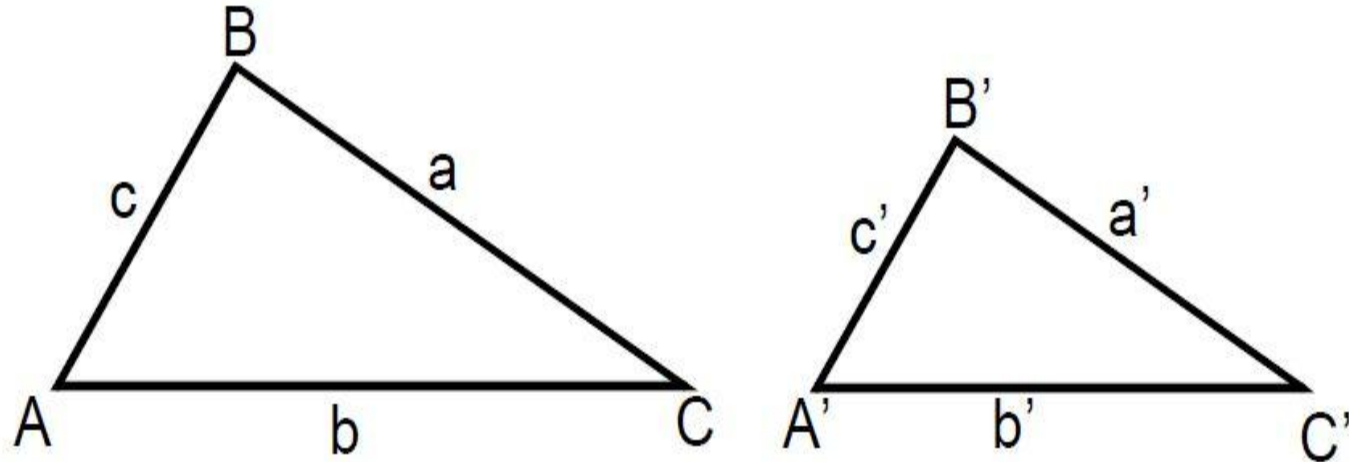
Dr. Gerhard Roth

COMP 4900D

Winter 2012

# Similar Triangles

---



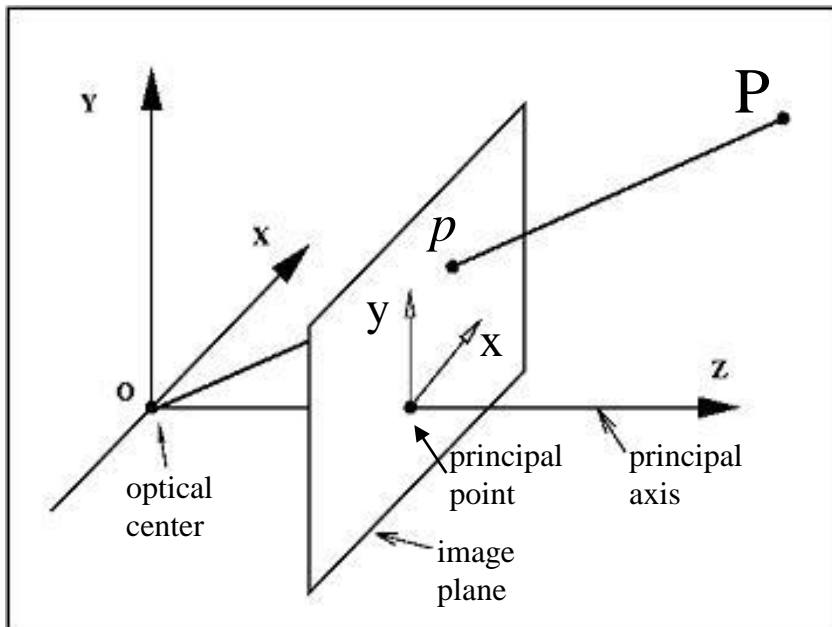
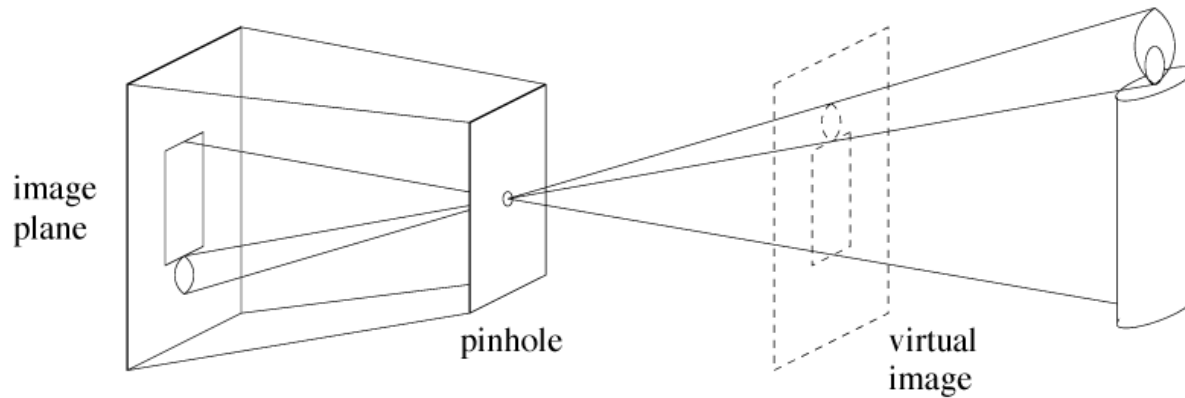
**property (i):** corresponding angles are equal  
( $A = A'$  and  $B = B'$  and  $C = C'$ )

**property (ii):** corresponding sides have proportional lengths

$$\left( \frac{a}{a'} = \frac{b}{b'} = \frac{c}{c'} \right)$$

# Geometric Model of Camera

## Perspective projection

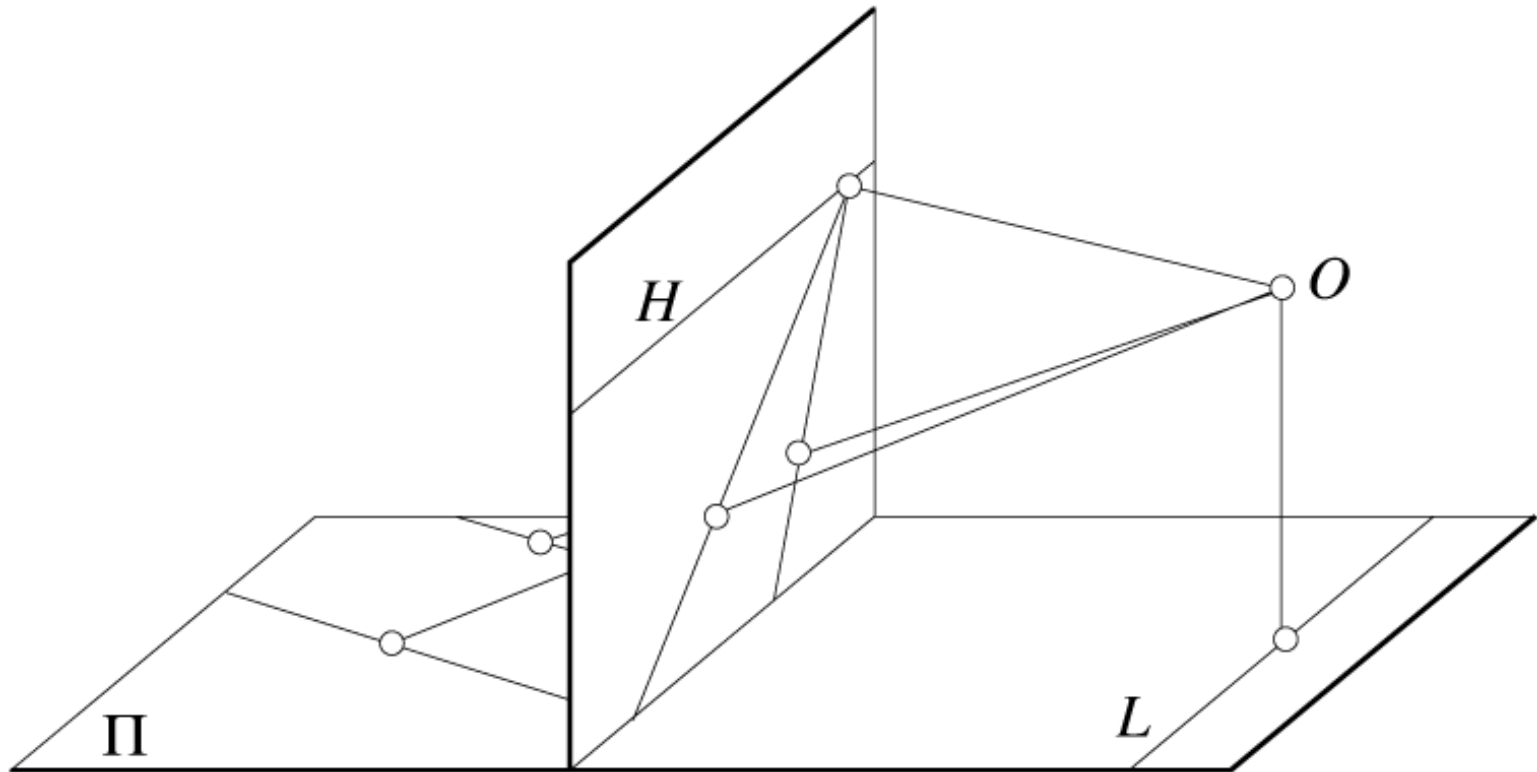


$$P(X,Y,Z) \rightarrow p(x,y)$$

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

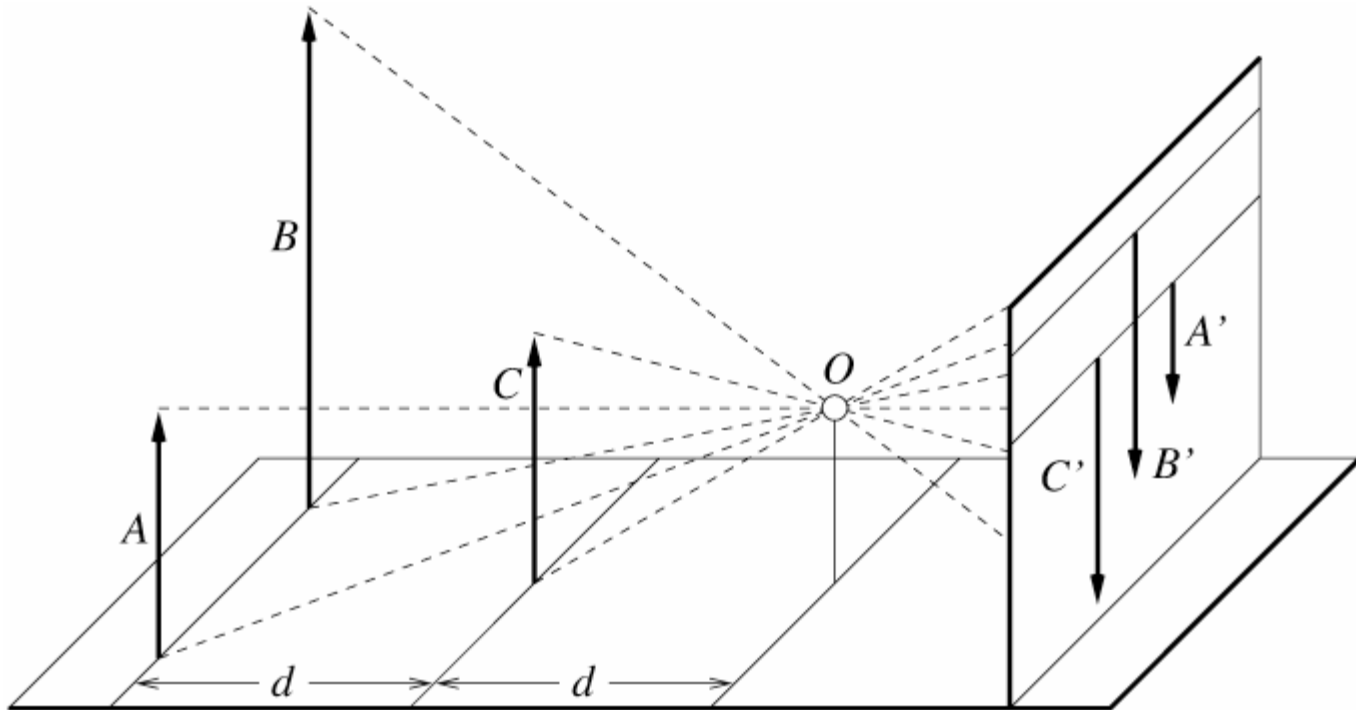
# Parallel lines aren't...

---



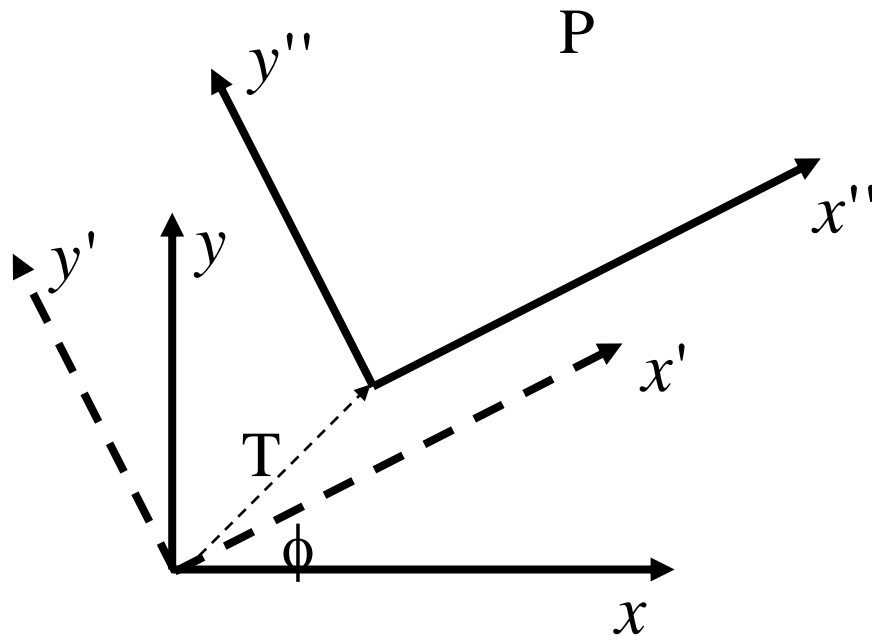
# Lengths can't be trusted...

---



# Coordinate Transformation – 2D

---



Rotation and Translation

$$p' = \begin{bmatrix} p_x' \\ p_y' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

$$p'' = \begin{bmatrix} p_x' \\ p_y' \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

# Homogeneous Coordinates

---

Go one dimensional higher:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

$w$  is an arbitrary non-zero scalar, usually we choose 1.

From homogeneous coordinates to Cartesian coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 / x_3 \\ x_2 / x_3 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 / x_4 \\ x_2 / x_4 \\ x_3 / x_4 \end{bmatrix}$$

# 2D Transformation with Homogeneous Coordinates

---

2D coordinate transformation:

$$p'' = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

2D coordinate transformation using homogeneous coordinates:

$$\begin{bmatrix} p_x'' \\ p_y'' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & T_x \\ -\sin \phi & \cos \phi & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$



# 3D Rotation Matrix

---

Rotate around each coordinate axis:

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad R_2(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad R_3(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Combine the three rotations:

$$R = R_1 R_2 R_3$$

3D rotation matrix has three parameters,  
no matter how it is specified.

# Rotation Matrices

---

- Both 2d and 3d rotation matrices have two characteristics
- They are orthogonal (also called orthonormal)

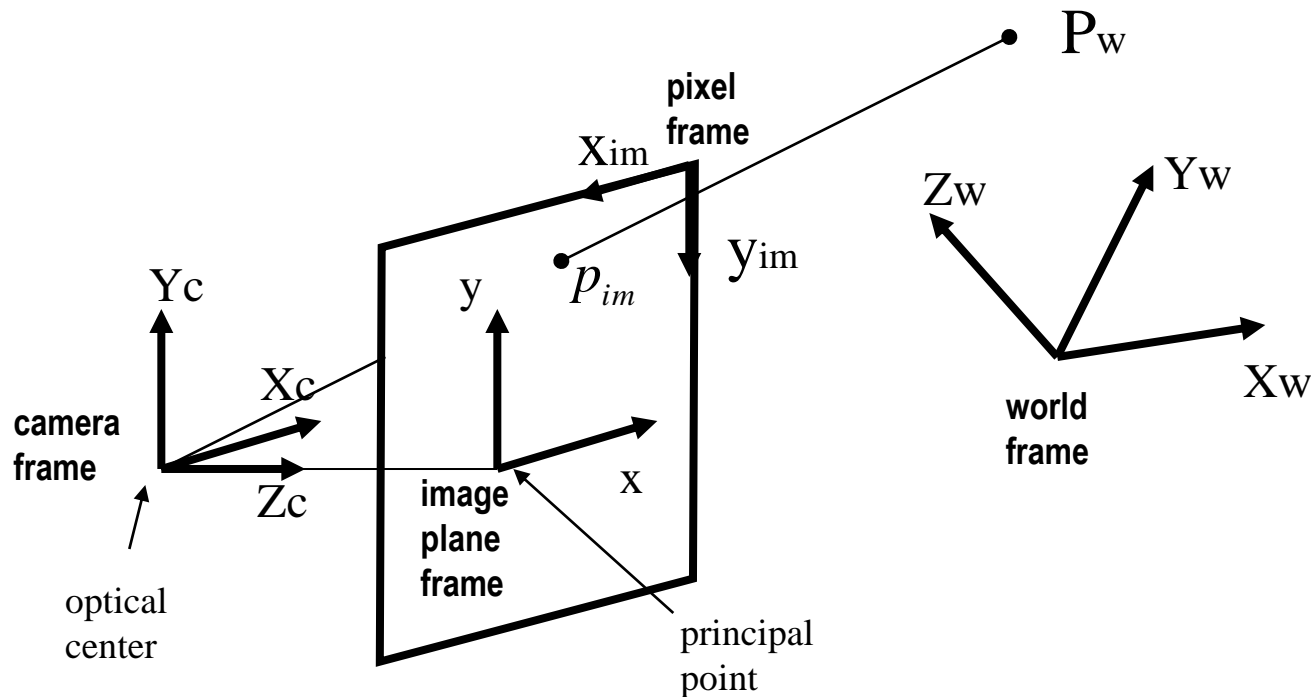
$$R^T R = I \qquad R^T = R^{-1}$$

- Their determinant is 1
- Matrix below is orthogonal but not a rotation matrix because the determinate is not 1

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \longleftarrow \text{this is a reflection matrix}$$

# Four Coordinate Frames

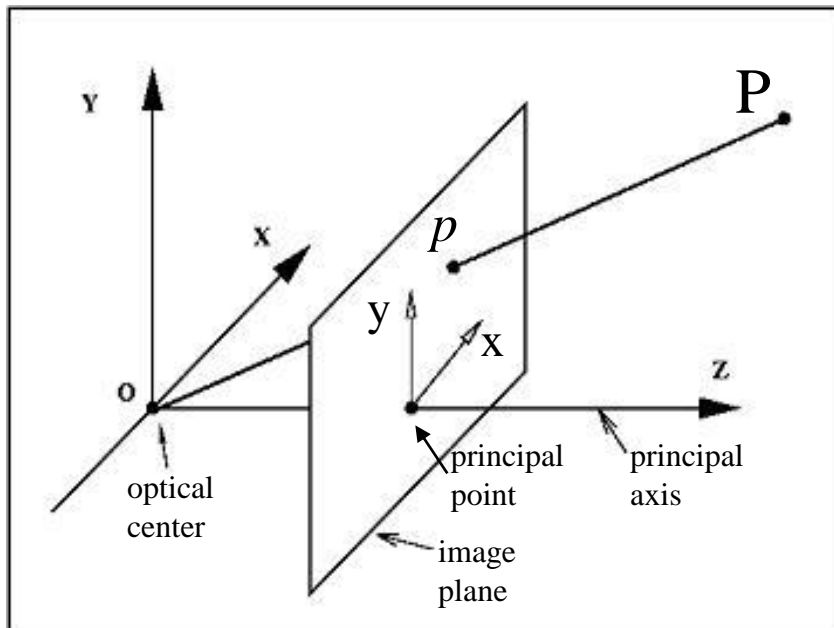
---



Camera model: 
$$p_{im} = \begin{bmatrix} \text{transformation} \\ \text{matrix} \end{bmatrix} P_w$$

# Perspective Projection

---



$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

These are *nonlinear*.

Using homogenous coordinate, we have a *linear* relation:

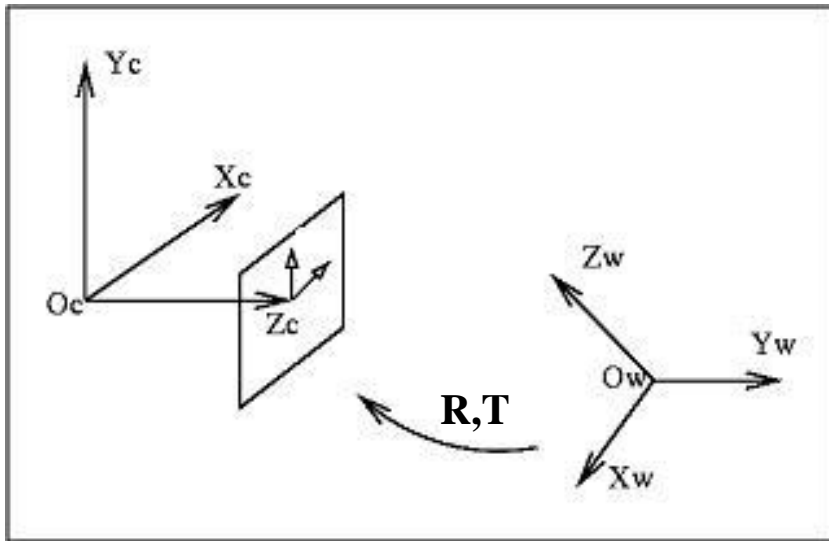
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = u / w \quad y = v / w$$

# World to Camera Coordinate

---

Transformation between the camera and world coordinates. Here we rotate, then translate, to go from world to camera co-ordinates which is opposite of book, but is simpler and is the way in which OpenCV routines do it:



$$\mathbf{X}_c = \mathbf{R}\mathbf{X}_w + \mathbf{T}$$

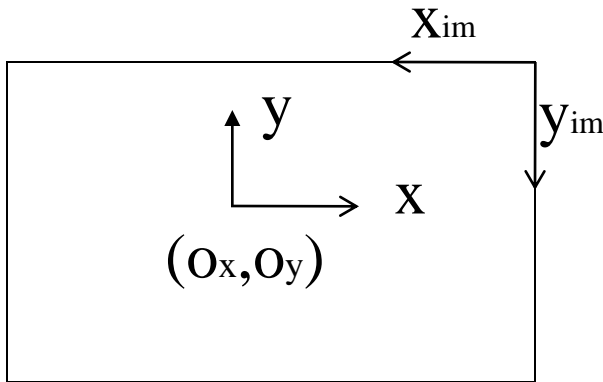
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

# Camera Coordinates to Pixel Coordinates

---

$$x = (o_x - x_{im})s_x \quad y = (o_y - y_{im})s_y$$

$s_x, s_y$  : pixel sizes in millimeters per pixel



$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Camera co-ordinates  $x$ , and  $y$  are in millimetres

Image co-ordinates  $x_{im}$ ,  $y_{im}$ , are in pixels

Center of projection  $o_x$ ,  $o_y$  is in pixels

# Put All Together – World to Pixel

---

$$\begin{aligned}
 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
 &= \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
 \end{aligned}$$

$$x_{im} = x_1 / x_3 \quad y_{im} = x_2 / x_3$$

# Camera Parameters

---

- Extrinsic parameters define the location and orientation of the camera reference frame with respect to a world reference frame
  - Depend on the external world, so they are extrinsic
- Intrinsic parameters link pixel co-ordinates in the image with the corresponding co-ordinates in the camera reference frame
  - An intrinsic characteristic of the camera
- Image co-ordinates are in pixels
- Camera co-ordinates are in millimetres



# Intrinsic Camera Parameters

---

$$K = \begin{bmatrix} -f / s_x & 0 & o_x \\ 0 & -f / s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

K is a 3x3 upper triangular matrix, called the **Camera Calibration Matrix**.

There are five intrinsic parameters:

- (a) The pixel sizes in x and y directions  $s_x, s_y$  in millimeters/pixel
- (b) The focal length  $f$  in millimeters
- (c) The principal point  $(o_x, o_y)$  in pixels, which is the point where the optic axis intersects the image plane.

# Camera intrinsic parameters

---

- Can write three of these parameters differently by letting  $f/s_x = f_x$  and  $f/s_y = f_y$ 
  - Then intrinsic parameters are  $ox, oy, f_x, f_y$
  - The units of these parameters are pixels!
- In practice pixels are square ( $s_x = s_y$ ) so that means  $f_x$  should equal  $f_y$  for most cameras
  - However, every explicit camera calibration process (using calibration objects) introduces some small errors
  - These calibration errors make  $f_x$  not exactly equal to  $f_y$
- So in OpenCV the intrinsic camera parameters are the four following  $ox, oy, f_x, f_y$ 
  - However  $f_x$  is usually very close to  $f_y$  and if this is not the case then there is a problem

# Extrinsic Parameters and Proj. Matrix

---

$$p_{im} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = K \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$[R|T]$  defines the **extrinsic parameters**.

The 3x4 matrix  $M = K[R|T]$  is called the **projection matrix**.

It takes 3d points in the world co-ordinate system and maps them to the appropriate image co-ordinates in pixels

# Using the projection matrix - example

---

$$\begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \mathbf{P}_{14} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \mathbf{P}_{24} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \mathbf{P}_{34} \end{bmatrix} = \begin{bmatrix} 1000 & 0 & 320 & -19600 \\ 0 & 1000 & 240 & -27200 \\ 0 & 0 & 1 & -30 \end{bmatrix}$$

$$u = \frac{u'}{w'} \quad v = \frac{v'}{w'}$$

- Where would the point (20,50,200) project to in the image?

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{matrix} \text{Projection matrix} \\ \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \mathbf{P}_{14} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \mathbf{P}_{24} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \mathbf{P}_{34} \end{bmatrix} \end{matrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1000 & 0 & 320 & -19600 \\ 0 & 1000 & 240 & -27200 \\ 0 & 0 & 1 & -30 \end{bmatrix} \begin{bmatrix} 20 \\ 50 \\ 200 \\ 1 \end{bmatrix}$$

$$u = \frac{u'}{w'} \quad v = \frac{v'}{w'}$$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} 64400 \\ 70800 \\ 170 \end{bmatrix}$$

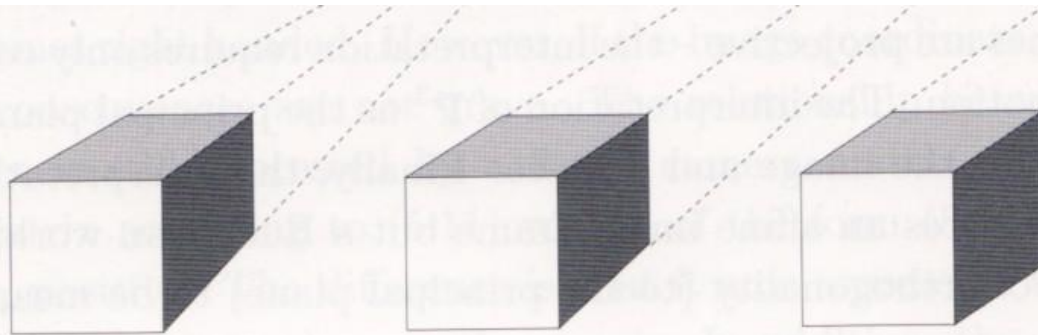
$$u = u'/w' = 64400/170 = 378.8$$

$$v = v'/w' = 70800/170 = 416.5$$

- World point (20,50,200) project to (379,414)

# Effect of change in focal length

Small  $f$  is wide angle, large  $f$  is telescopic

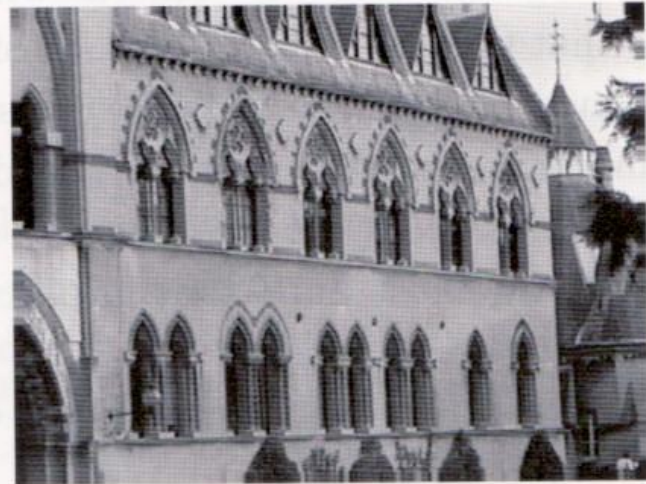
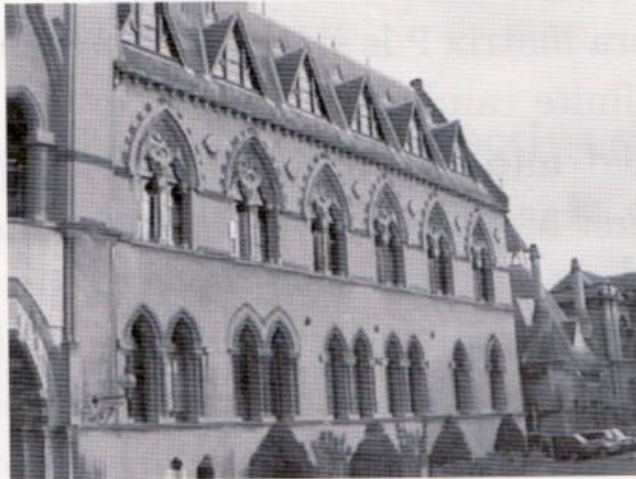


perspective

weak perspective

increasing focal length

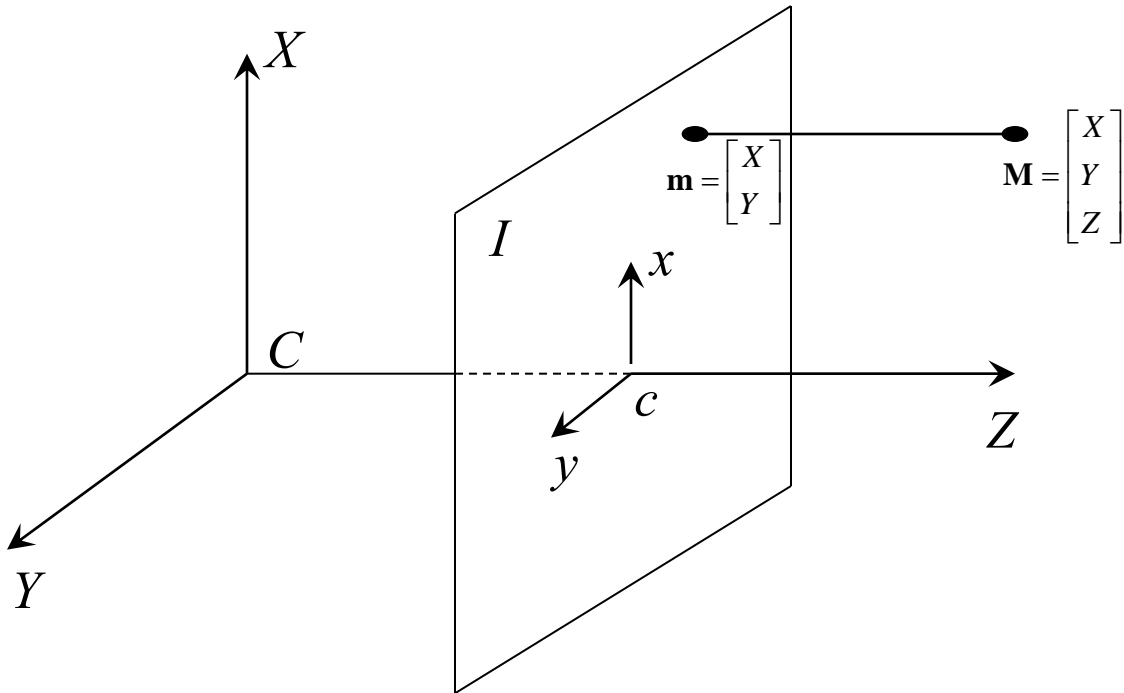
increasing distance from camera



# Orthographic Projection

---

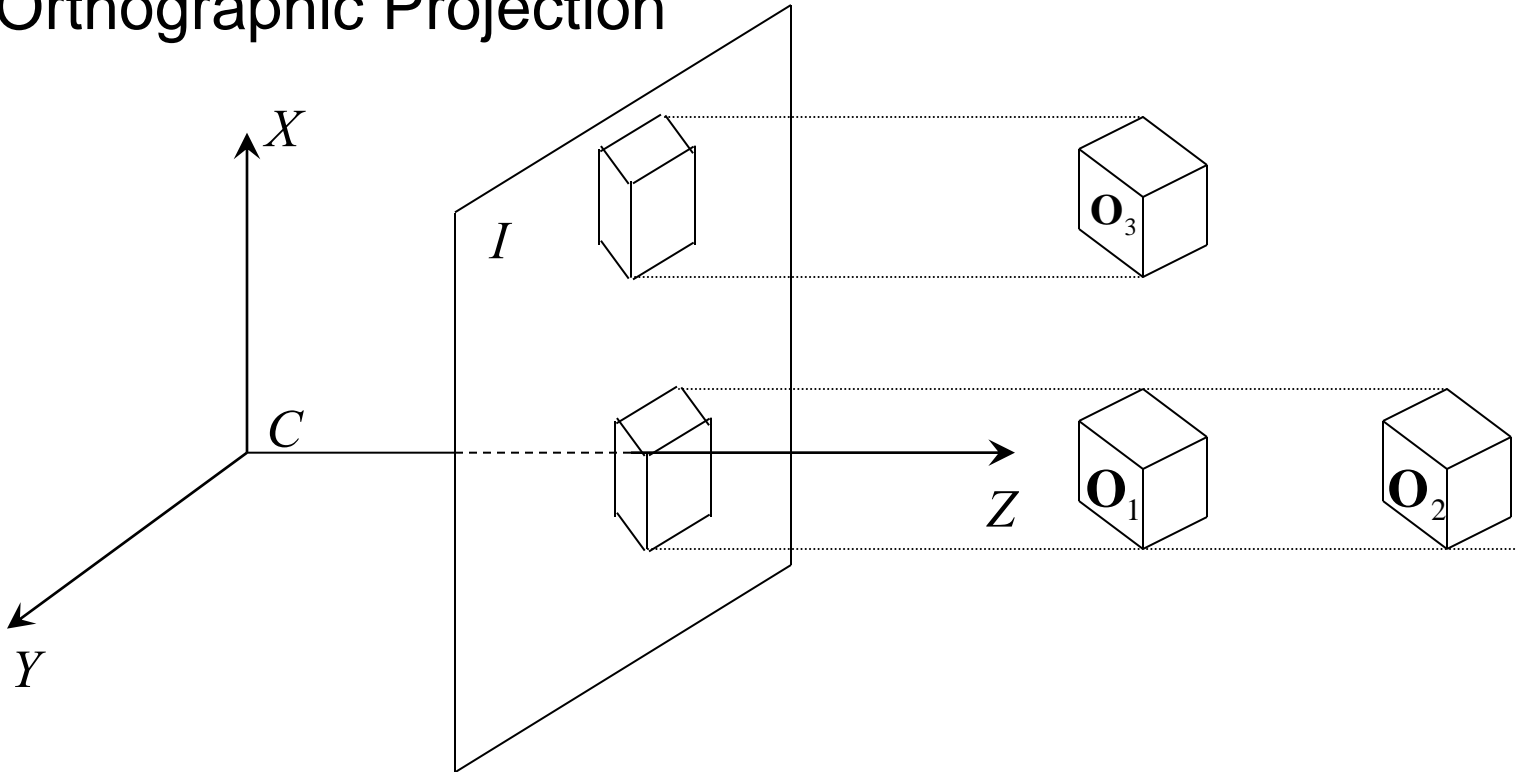
## Orthographic Projection



# Orthographic Projection

---

## Orthographic Projection



# Weak Perspective Model

---

Assume the relative distance between any two points along the principal axis is much smaller ( $1/20^{\text{th}}$  at most) than the average distance  $\bar{Z}$

Then the camera projection can be approximated as:

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

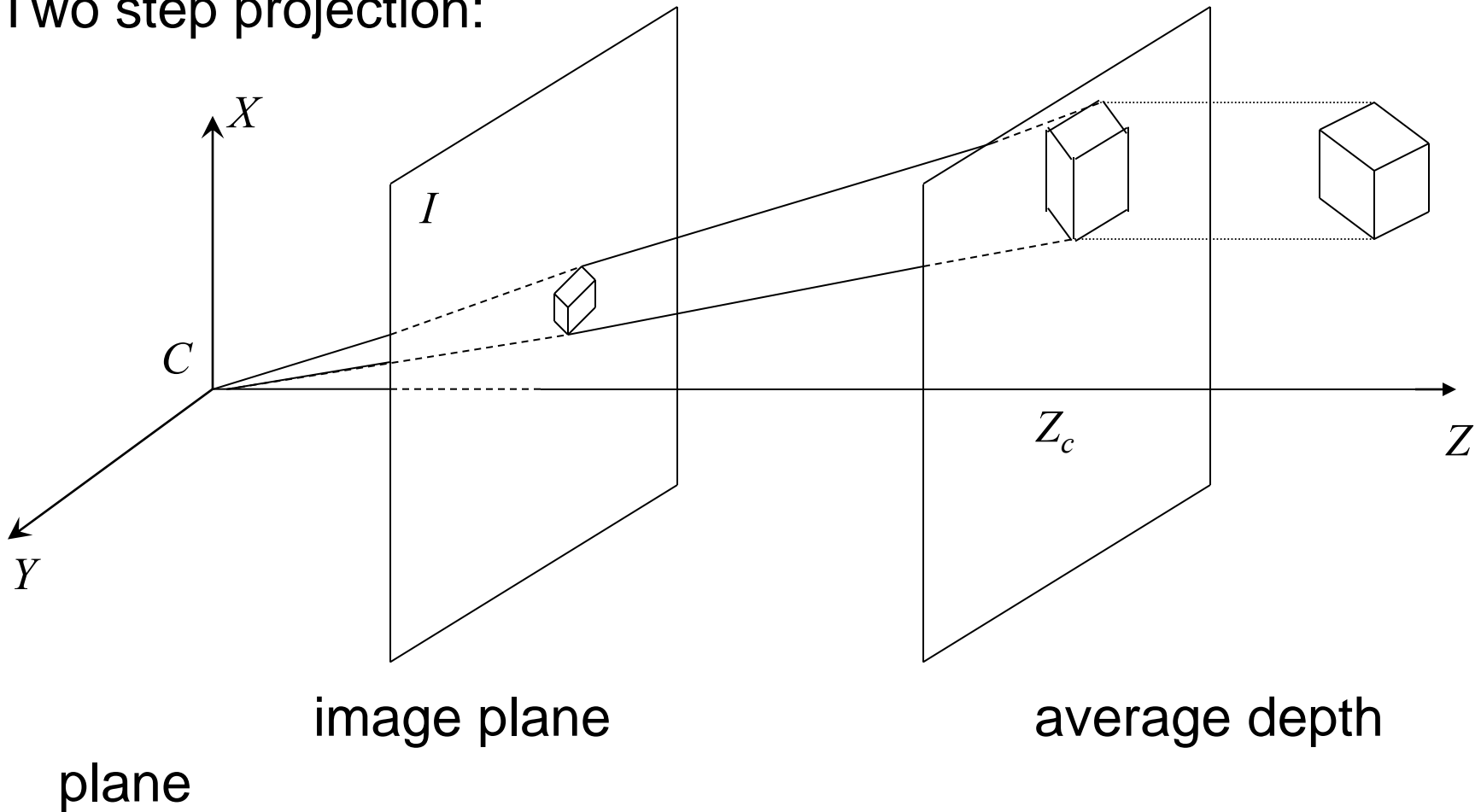
$$y = f \frac{Y}{Z} \approx \frac{f}{\bar{Z}} Y$$

This is the **weak-perspective** camera model.  
Sometimes called scaled orthography.



# Weak Perspective Projection

Two step projection:

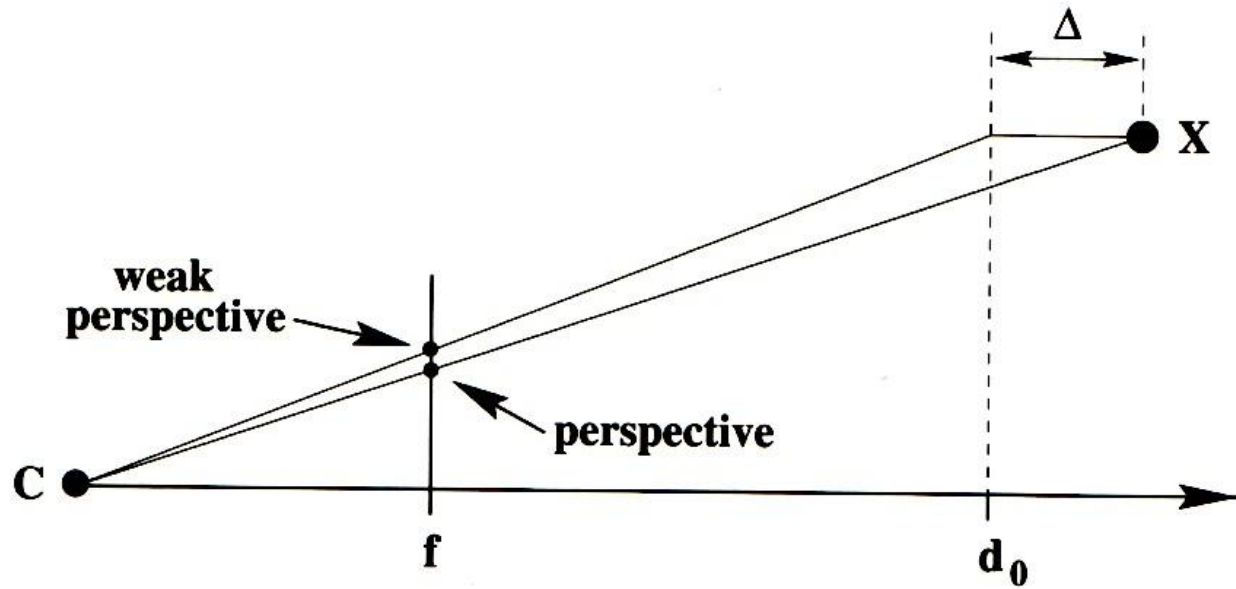


# Weak Perspective

---

158

5 Camera Models



# Impact of different projections

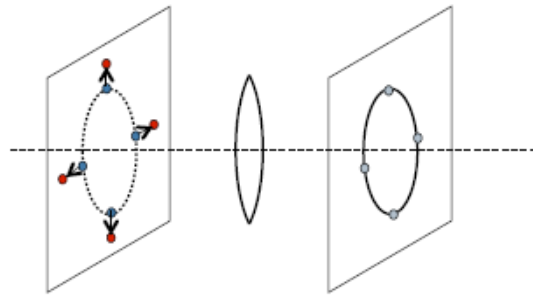
---

- Perspective projections
  - Parallel lines in world are not parallel in the image
  - Object projection gets smaller with distance from camera
- Weak perspective projection
  - Parallel lines in the world are parallel in the image
  - Object projection gets smaller with distance from camera
- Orthographic projection
  - Parallel lines in the world are parallel in the image
  - Object projection is unchanged with distance from camera

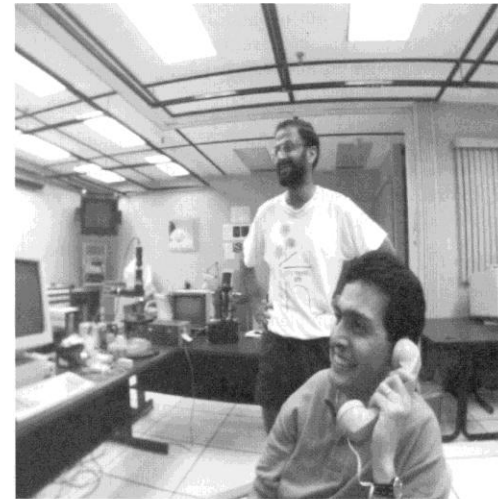
# Image distortion due to optics

---

- Radial distortion which depends on radius  $r$ , distance of each point from center of image
- $r^2 = (x - o_x)^2 + (y - o_y)^2$



Radial distortion



$$x_{\text{corrected}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{\text{corrected}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Correction uses three parameters,  $k_1, k_2, k_3$

# Correcting Radial Distortions

---



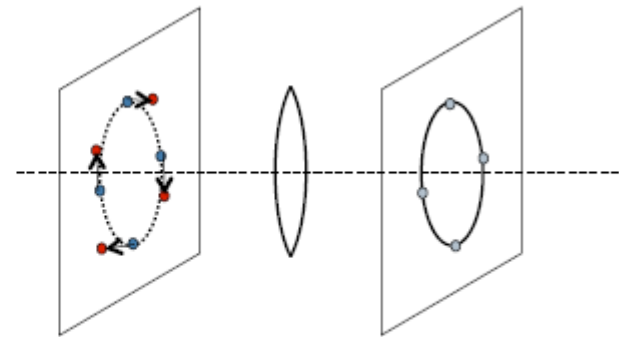
# Tangential Distortion

---

- Lens not exactly parallel to the image plane

$$x_{\text{corrected}} = x + [2p_1y + p_2(r^2 + 2x^2)]$$

$$y_{\text{corrected}} = y + [p_1(r^2 + 2y^2) + 2p_2x]$$



Tangential distortion

- Correction uses two parameter  $p_1$ ,  $p_2$
- Both types of distortion are removed (image is un-distorted) and only then does standard calibration matrix  $K$  apply to the image
- Camera calibration computes both  $K$  and these five distortion parameters

# How to find the camera parameters

---

- Can use the EXIF tag for any digital image
  - Has focal length  $f$  in millimeters but not the pixel size
  - But you can get the pixel size from the camera manual
  - There are only a finite number of different pixels sizes because number of sensing element sizes is limited
  - If there is not a lot of image distortion due to optics then this approach is sufficient (only linear calibration)
- Can perform explicit camera calibration
  - Put a calibration pattern in front of the camera
  - Take a number of different pictures of this pattern
  - Now run the calibration algorithm (different types)
  - Result is intrinsic camera parameters (linear and non-linear) and the extrinsic camera parameters of all the images