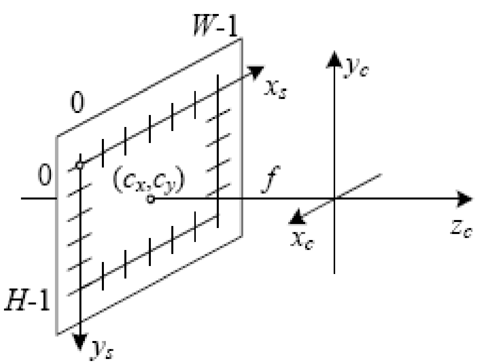
计算机视觉

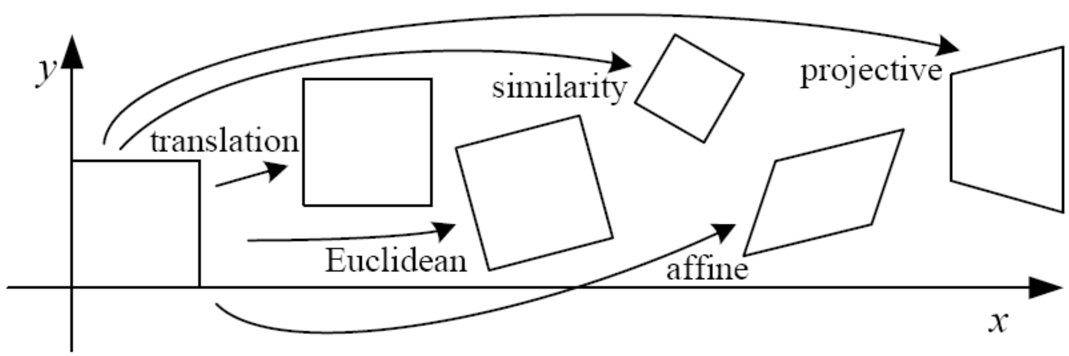
Computer Vision

Lecture 2: Geometric Image Models

张 超

信息科学技术学院 智能科学系





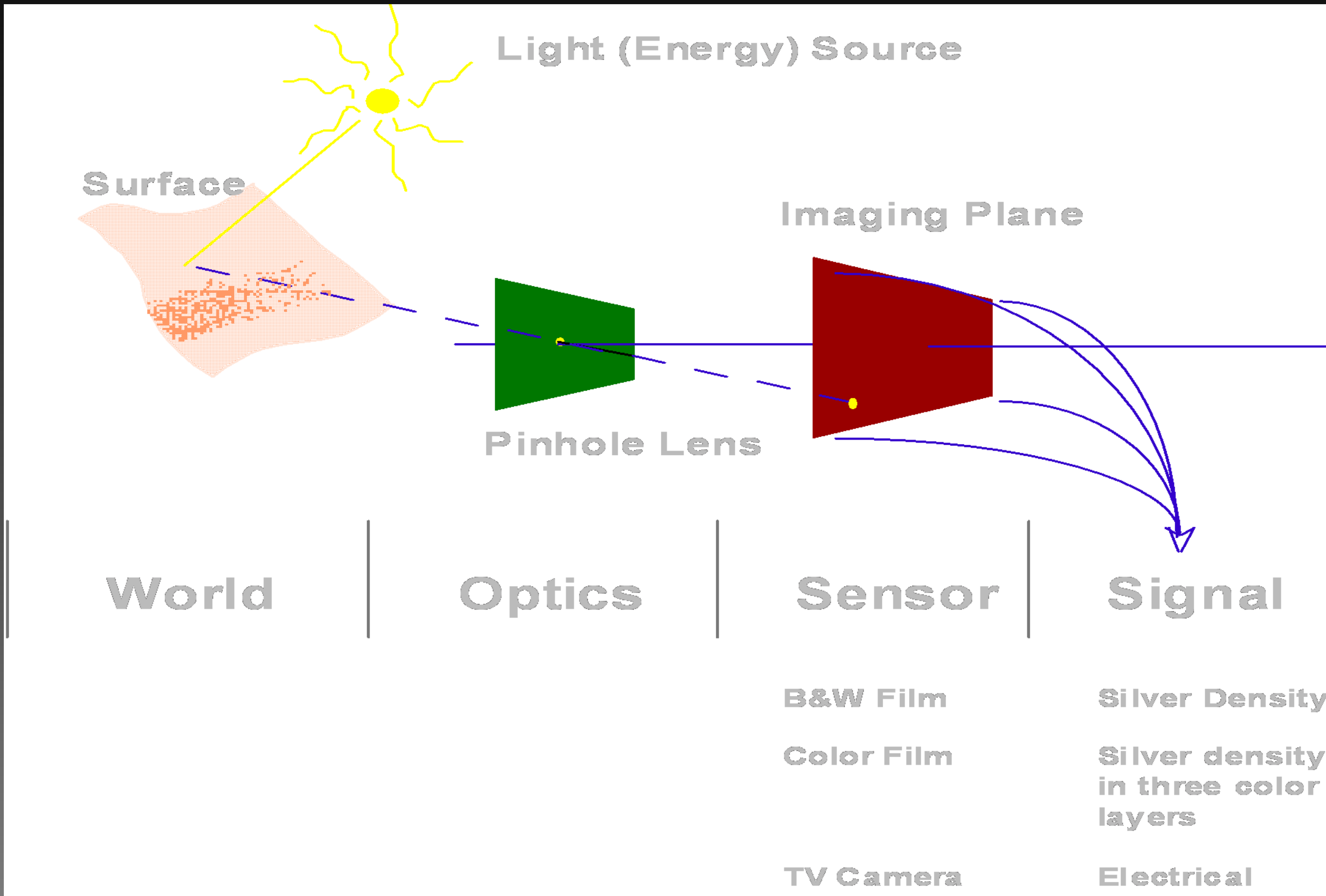
Outline for Today

- Introduction
- Pinhole Camera Model
  - Perspective Projection
  - Affine Projection
- Homogeneous Coordinates
- Camera Parameters
  - Intrinsic Parameters
  - Extrinsic Parameters

Where are we?

- Imaging: pixels, features, …
- Scenes: geometry, material, lighting
- Recognition: people, objects, …

Image Formation



Steps

World → Optics → Sensor →
Signal → Digitizer →
Digital Representation

World	reality
Optics	focus “light” from world on sensor
Sensor	converts “light” to “electrical energy”
Signal	representation of incident light as continuous electrical energy
Digitizer	converts continuous signal to discrete signal
Digital Rep.	final representation of reality in computer memory

Image Formation Process

- Light conditions
- Surface properties
- Camera optics
- Scene geometry

Factors in Image Formation

- **Geometry**
  - concerned with the relationship between points in the three-dimensional world and their images
- **Radiometry（辐射度学）**
  - concerned with the relationship between the amount of light radiating from a surface and the amount incident at its image
- **Photometry（光度学）**
  - concerned with ways of measuring the intensity of light
- **Digitization**
  - concerned with ways of converting continuous signals (in both space and time) to digital approximations

Image Formation

- Vision infers world properties from images.
- So we need to understand and describe how images depend on these properties.

Lecture Assumptions

- **Typical imaging scenario:**
  - visible light
  - ideal lenses
  - standard sensor (e.g. TV camera)
  - opaque objects
- **Goal**

To create 'digital' images which can be processed to recover some of the characteristics of the 3D world which was imaged.

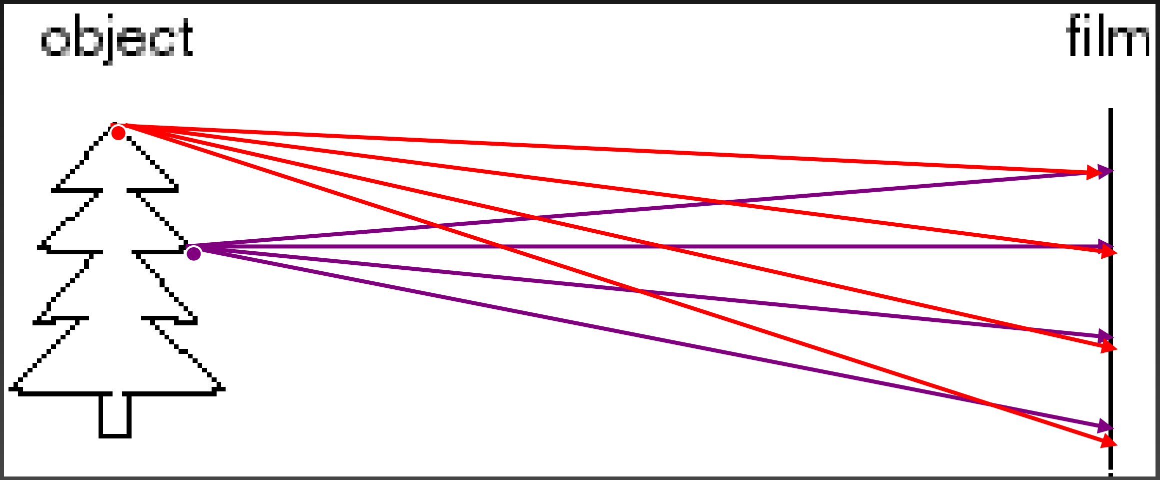
Geometry

- **Geometry describes the projection of:**

three-dimensional
(3D) world

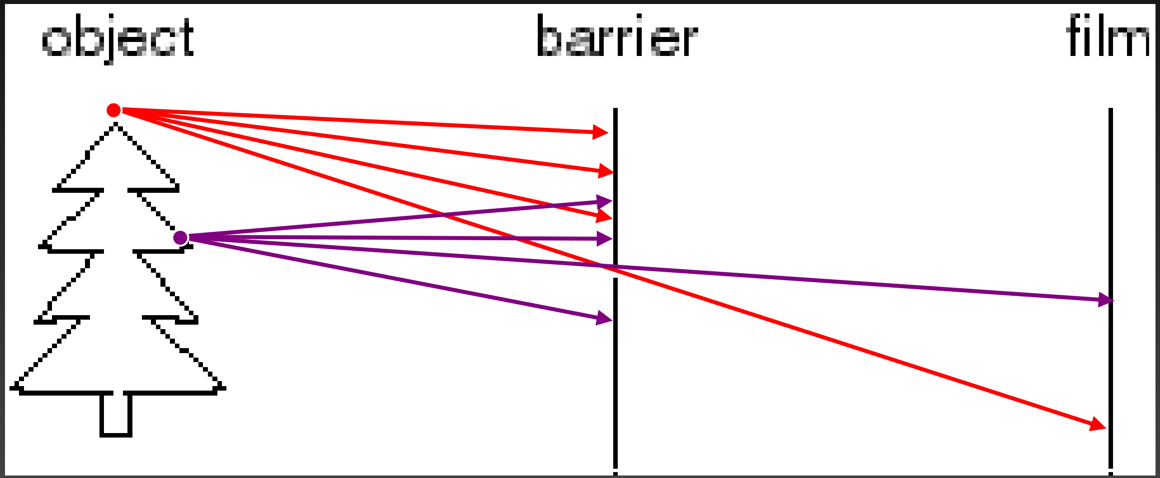
two-dimensional
(2D) image plane.

Image formation



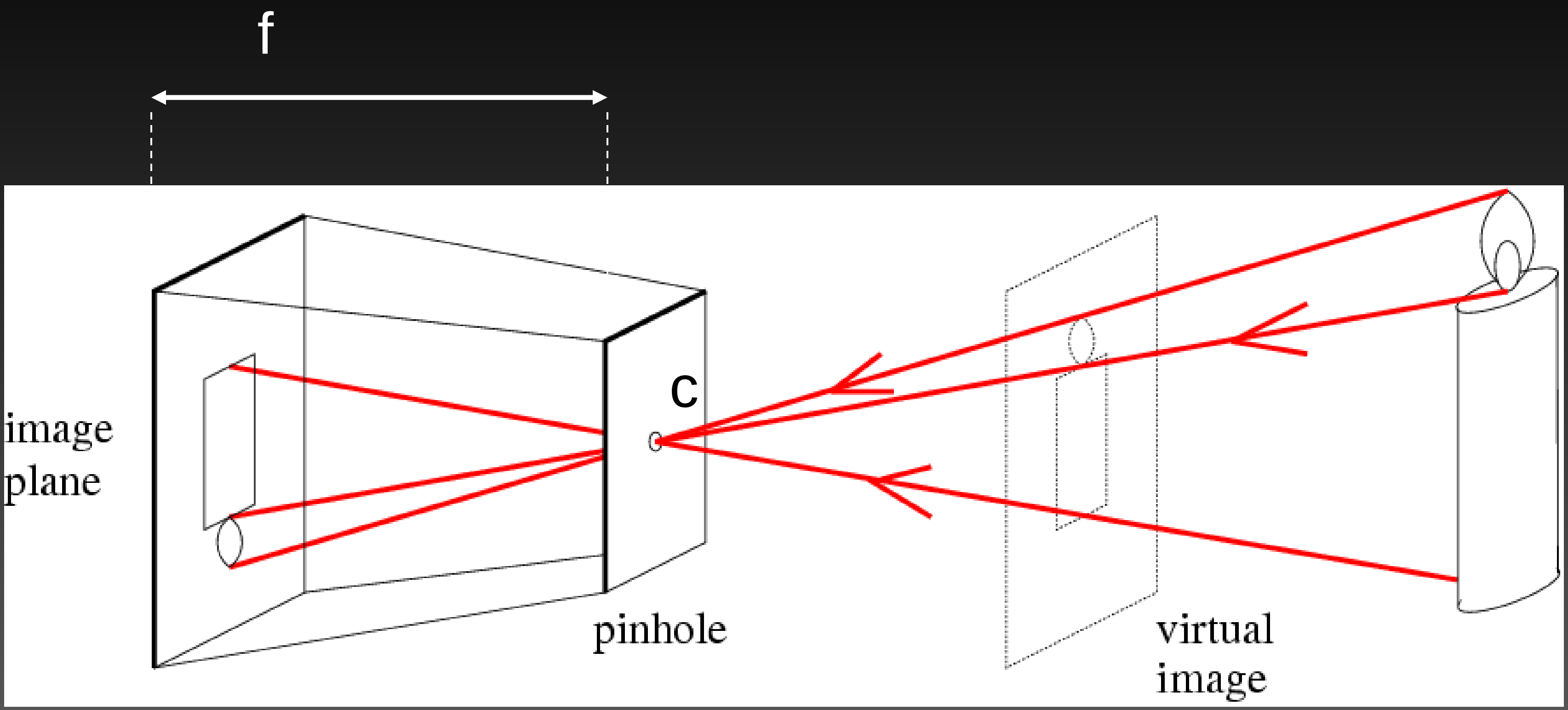
- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

Pinhole camera（小孔摄象机）



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the aperture
  - How does this transform the image?
    - it gets inverted

Pinhole camera



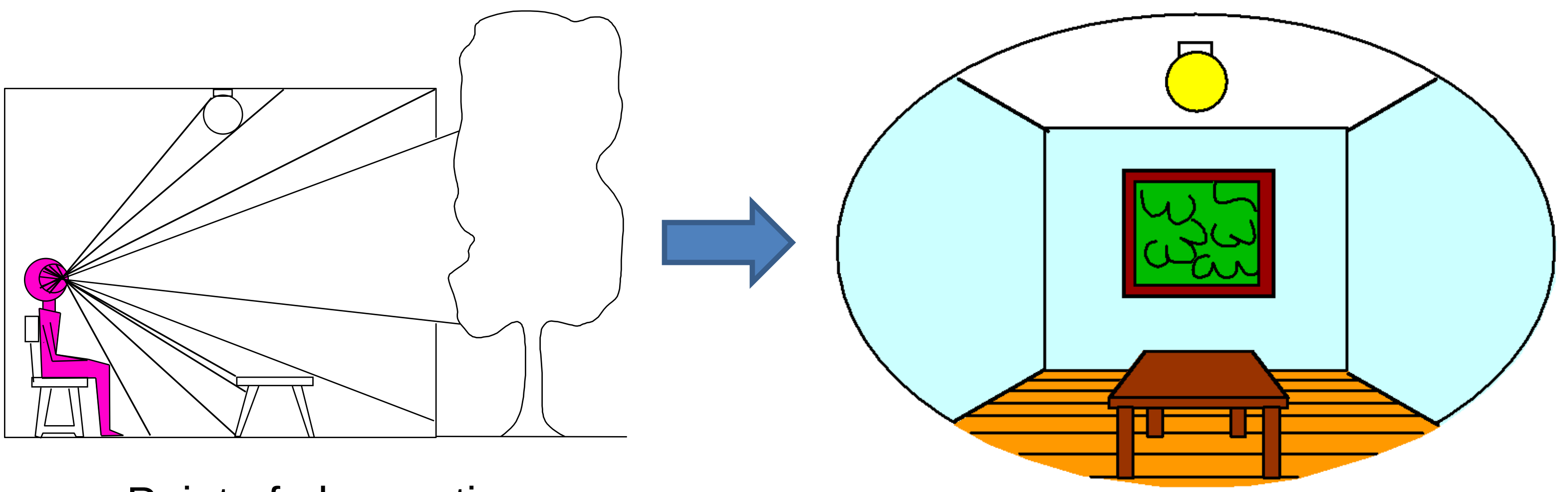
f = focal length

c = center of the camera

Figure from Forsyth

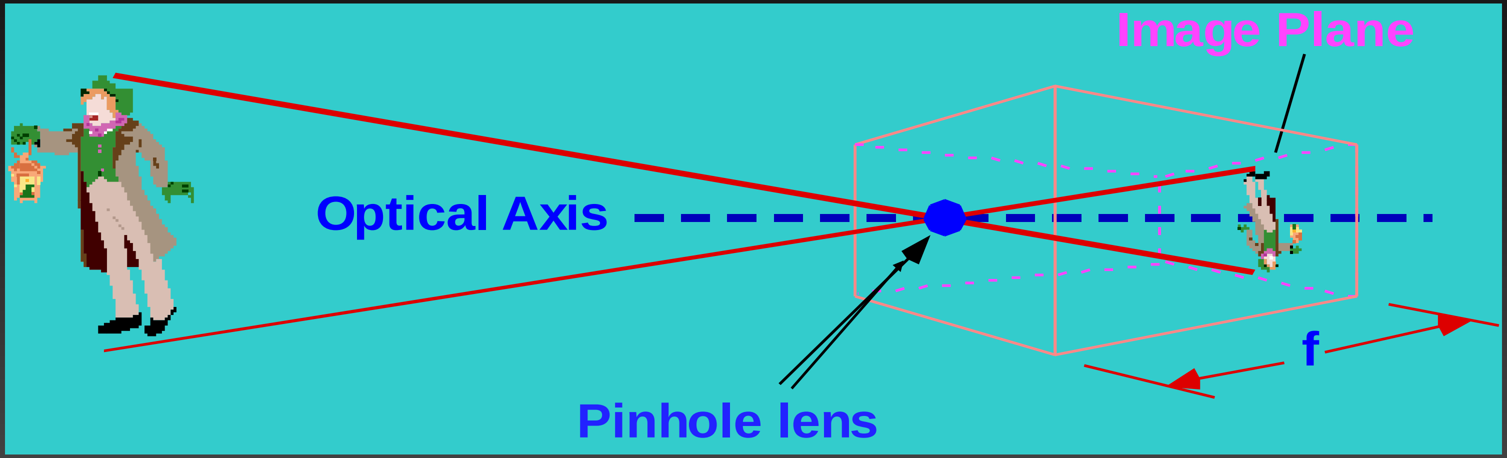
Dimensionality Reduction Machine (3D to 2D)

3D world *2D image*



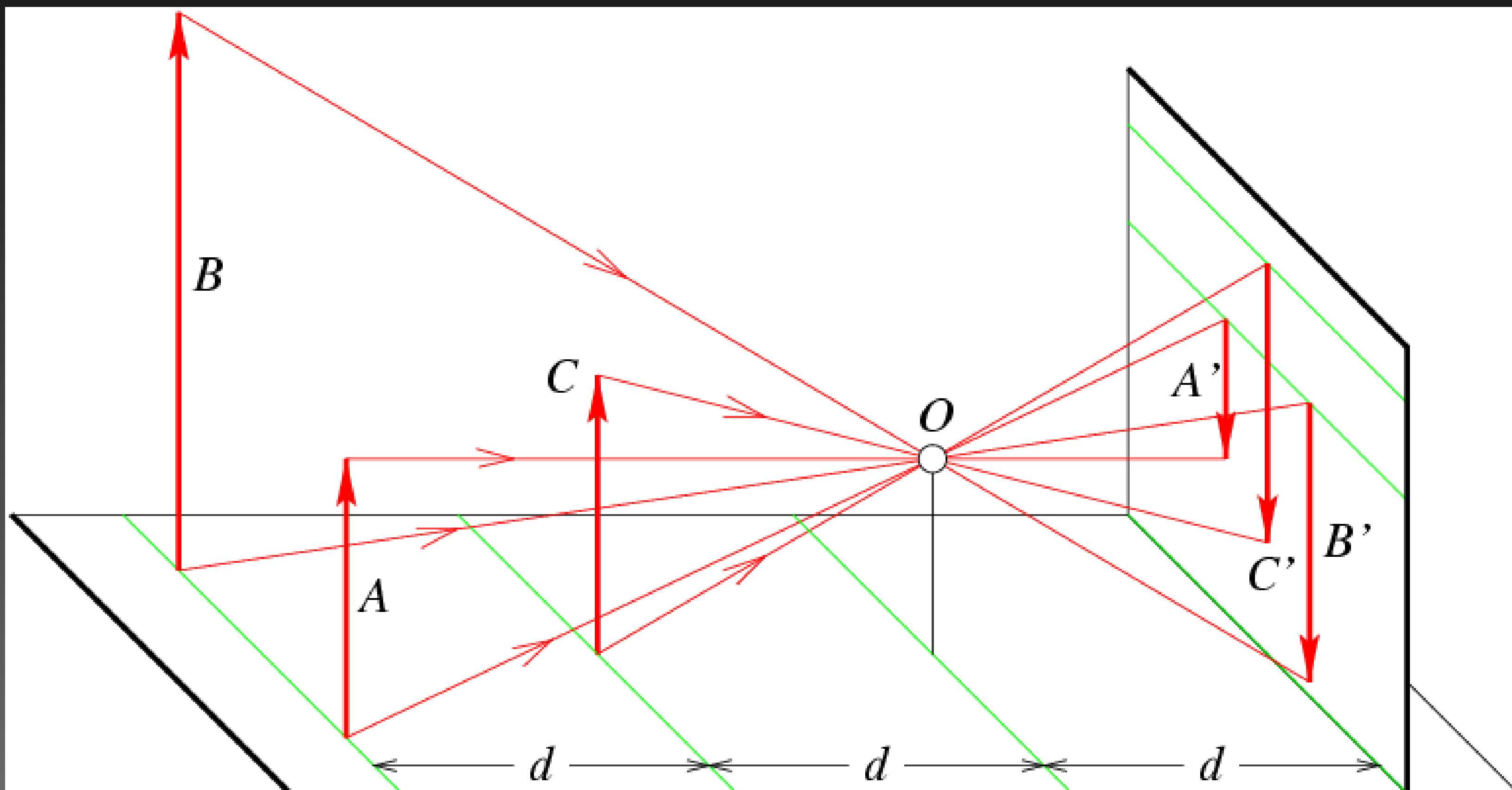
Point of observation

Pinhole Camera Model



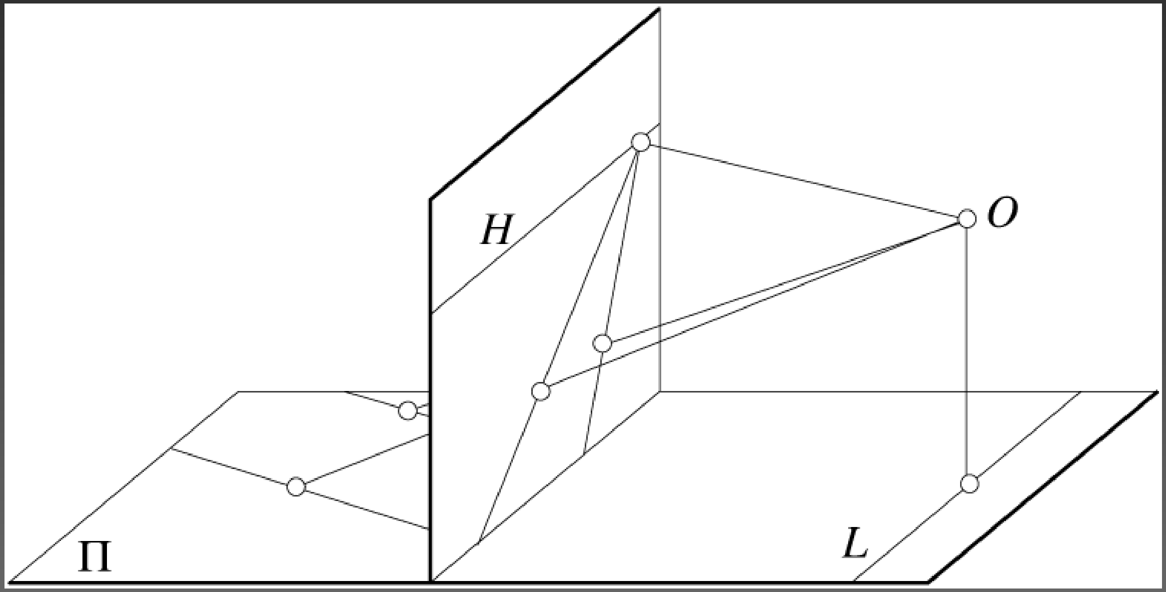
- World projected to 2D Image
  - Image inverted
  - Size reduced
  - Image is dim
  - No direct depth information
- Known as perspective projection（透视投影）

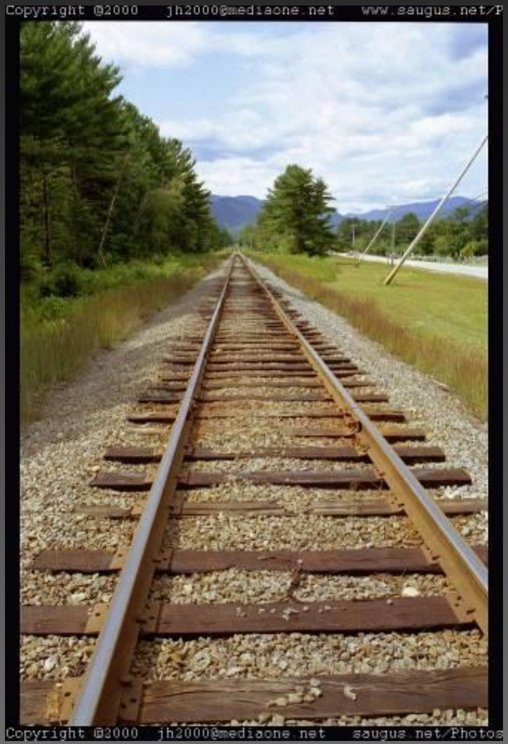
Distant objects are smaller



Parallel lines meet

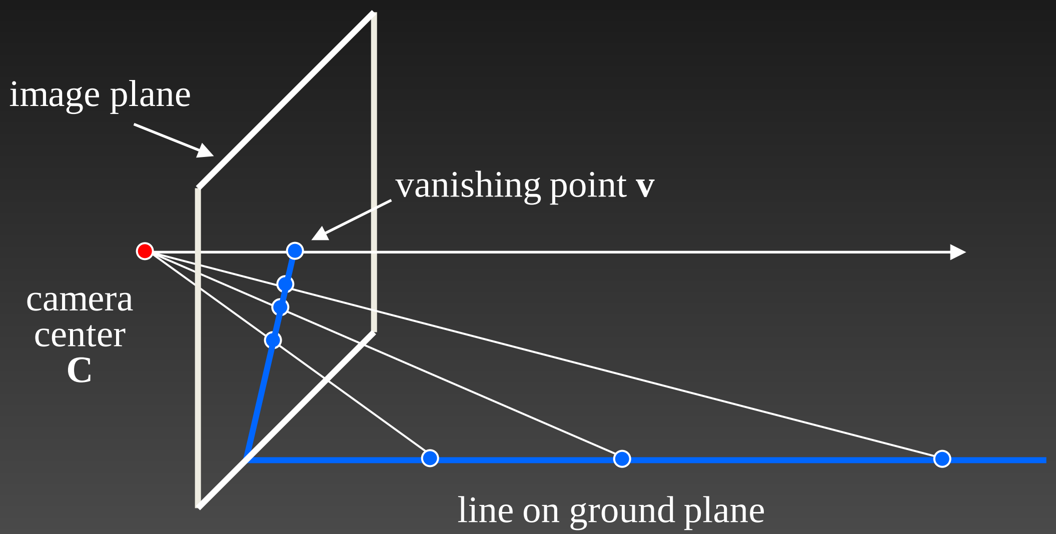
Common to draw image plane *in front* of the focal point.
Moving the image plane merely scales the image.





(Forsyth & Ponce)

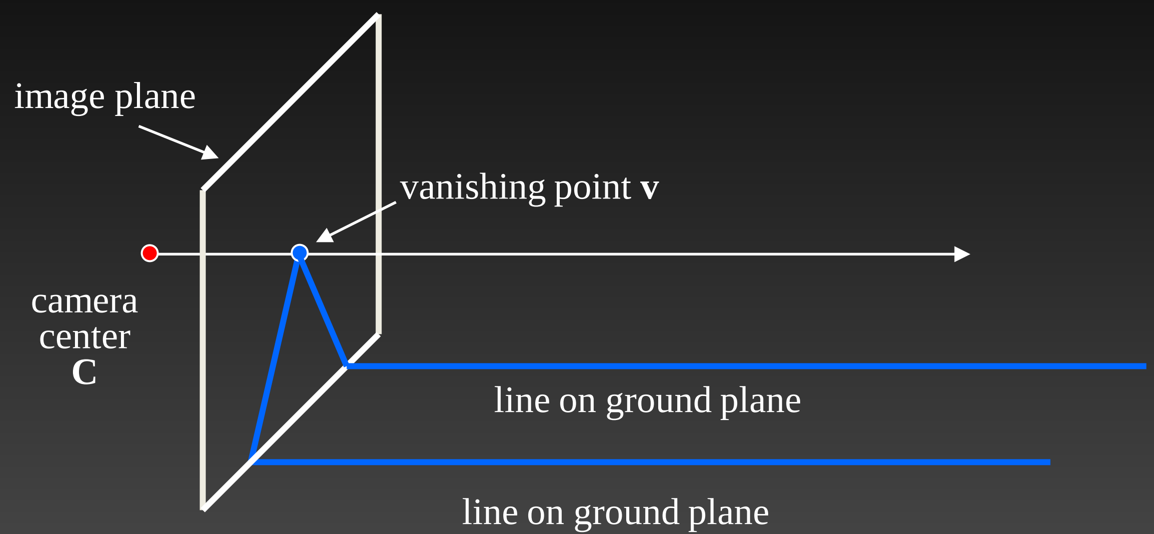
Vanishing points（消失点）



Vanishing point

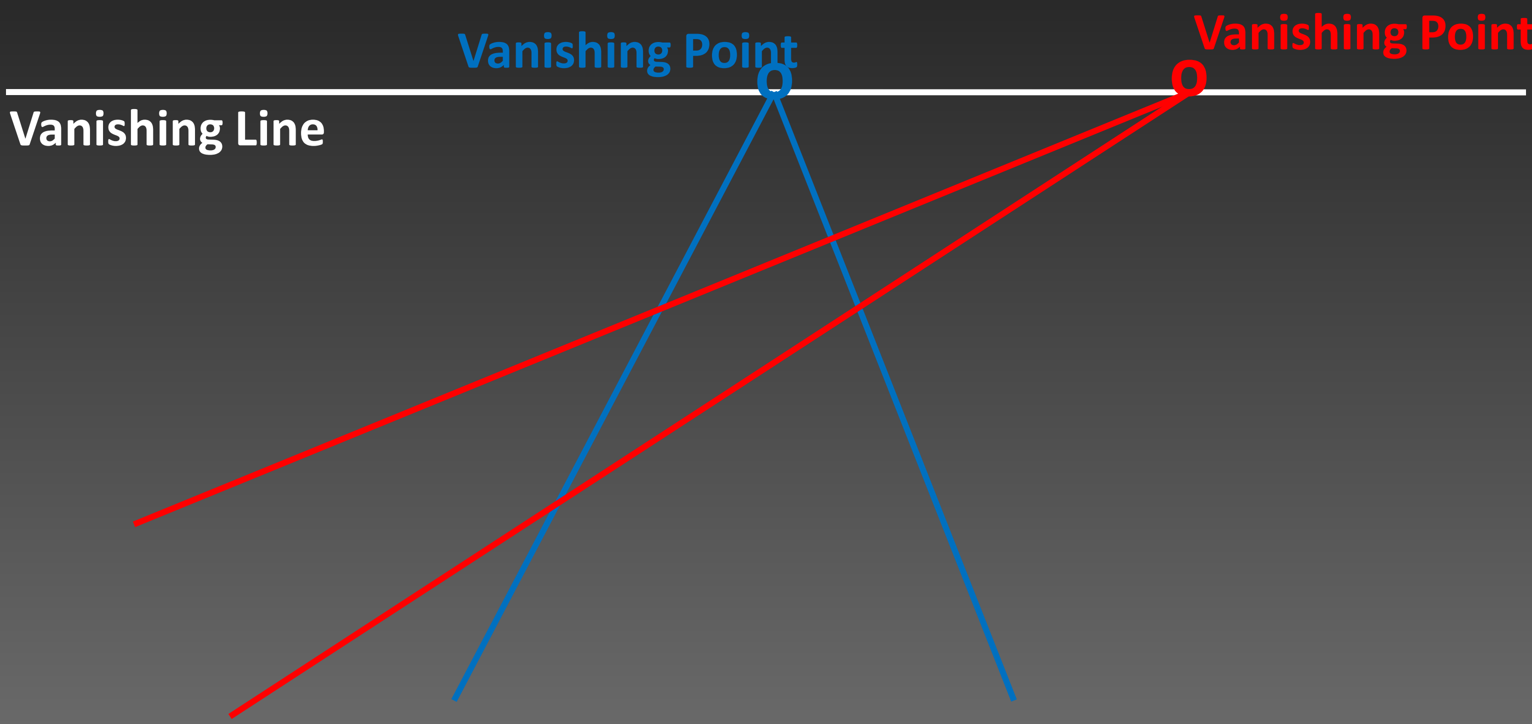
- projection of a point at infinity

Vanishing points

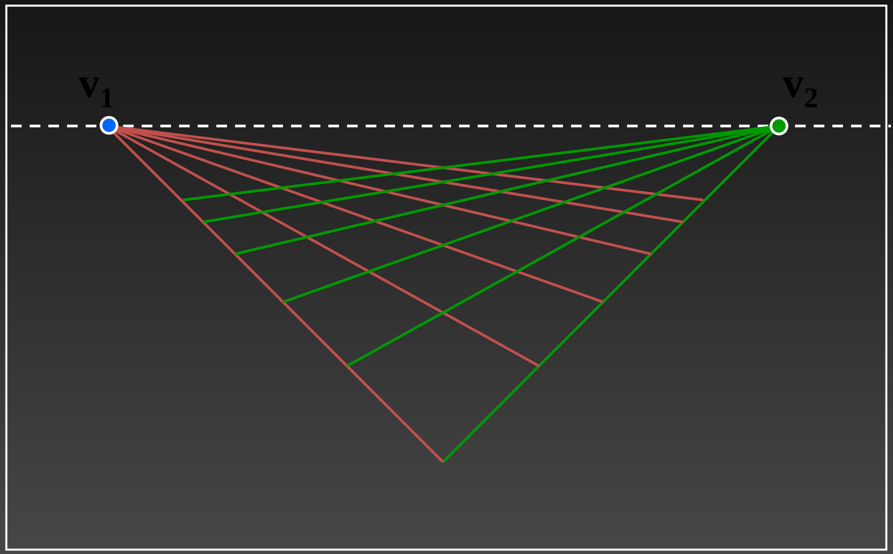


- **Properties**
  - Any two parallel lines have the same vanishing point v
  - The ray from C through v is parallel to the lines
  - An image may have more than one vanishing point
    - in fact every pixel is a potential vanishing point

Vanishing points and lines

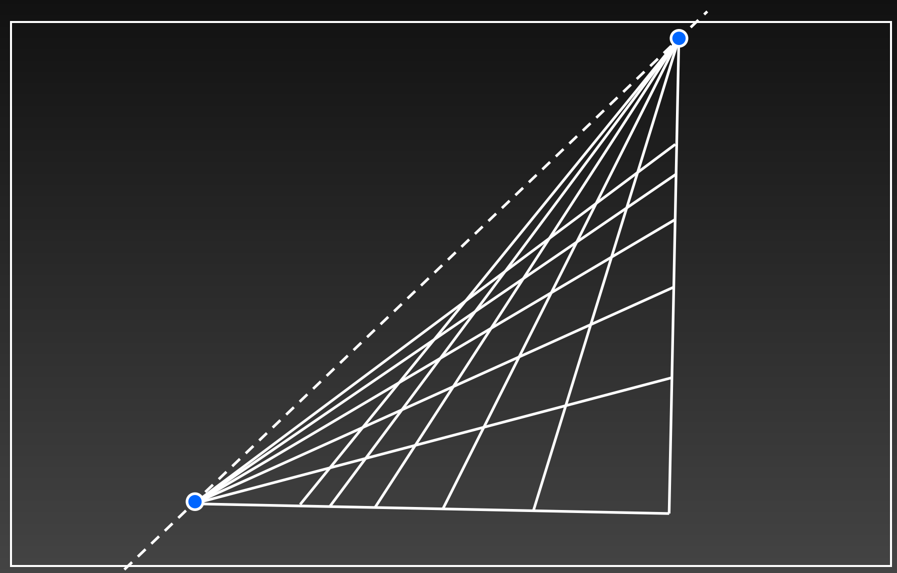


Vanishing lines



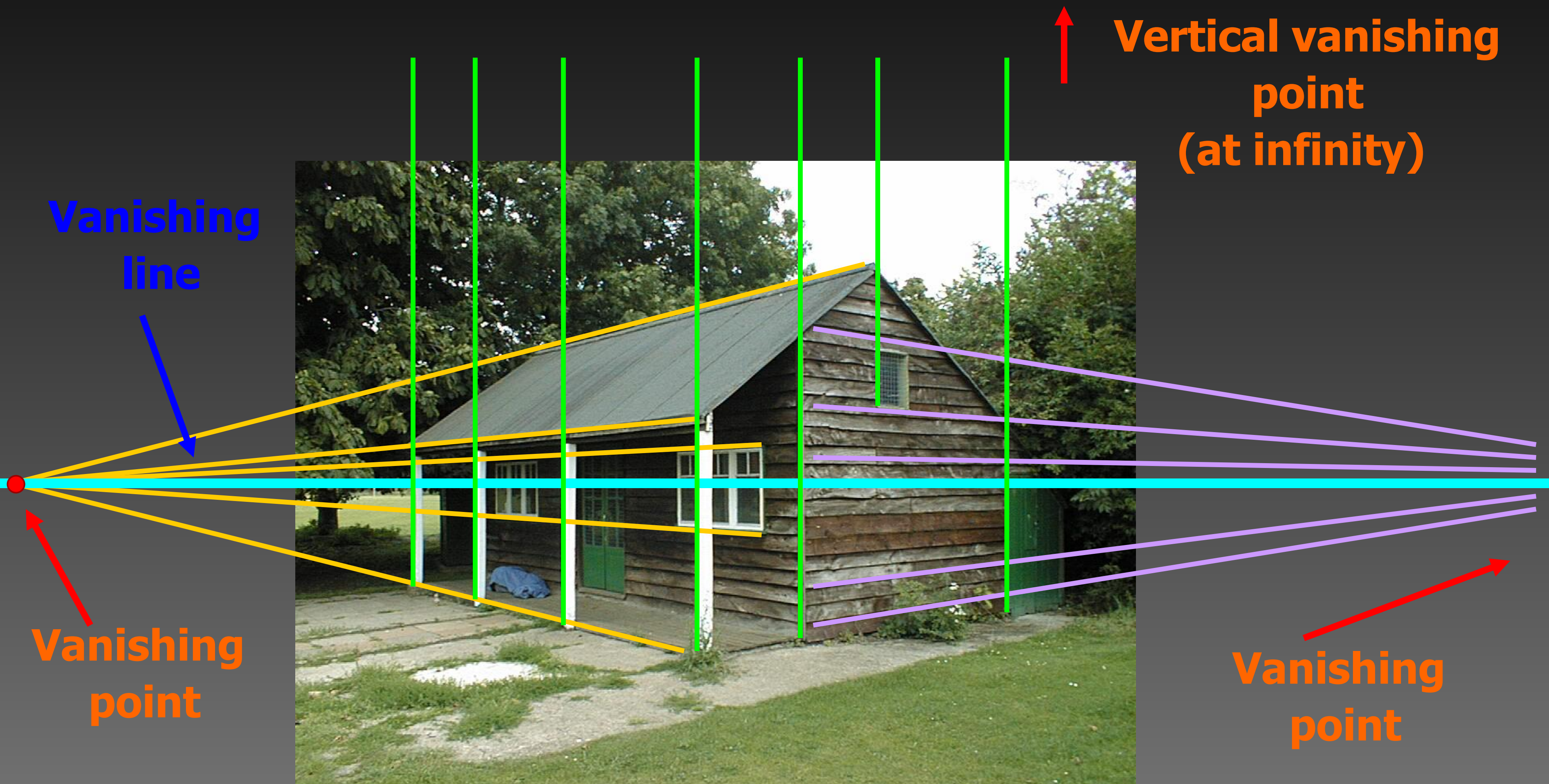
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of vanishing points from lines on the same plane is the *vanishing line*
    - For the ground plane, this is called the *horizon*

Vanishing lines



- Multiple Vanishing Points
  - Different planes define different vanishing lines

Vanishing points and lines



Slide from Efros, Photo from Criminisi

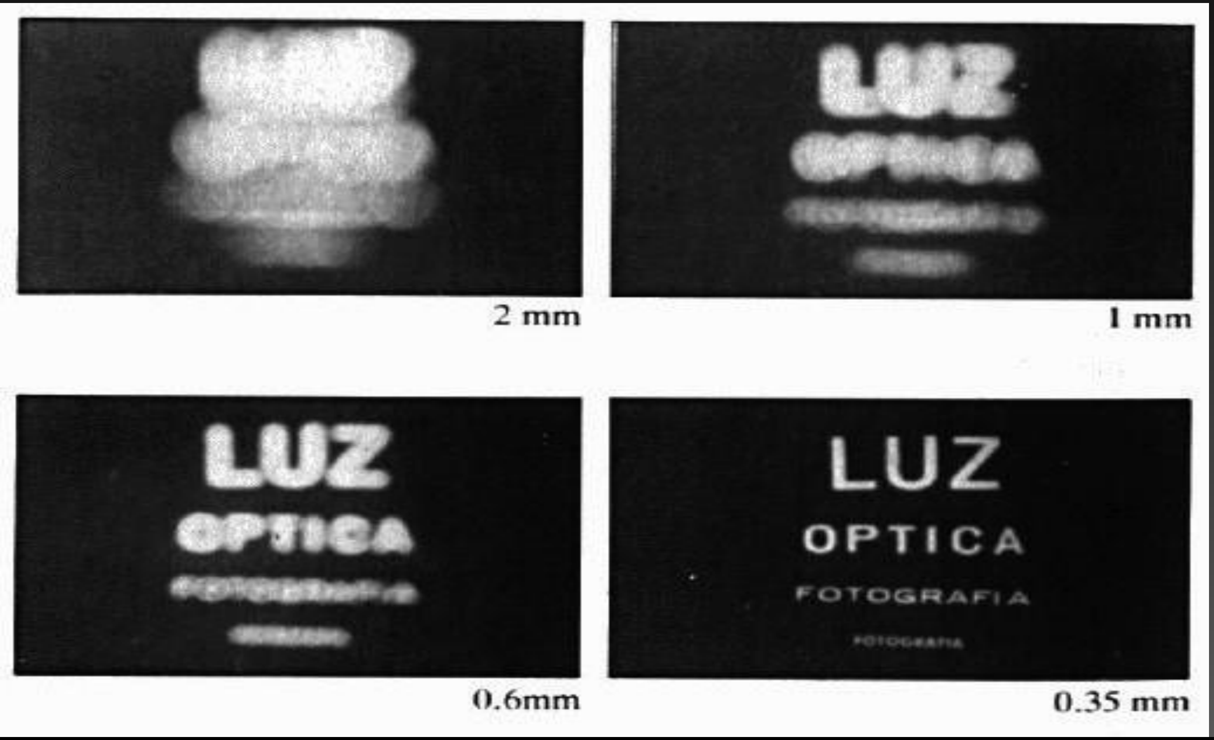
Properties of Projection

- Points project to points
- Lines project to lines
- Planes project to the whole image
- Circles project into ellipses
- Angles are not preserved
- Farther objects appear smaller
- Degenerate cases
  - Line through focal point projects to a point.
  - Plane through focal point projects to line
  - Plane perpendicular to image plane projects to part of the image (with horizon).

Pinhole Camera Limitations

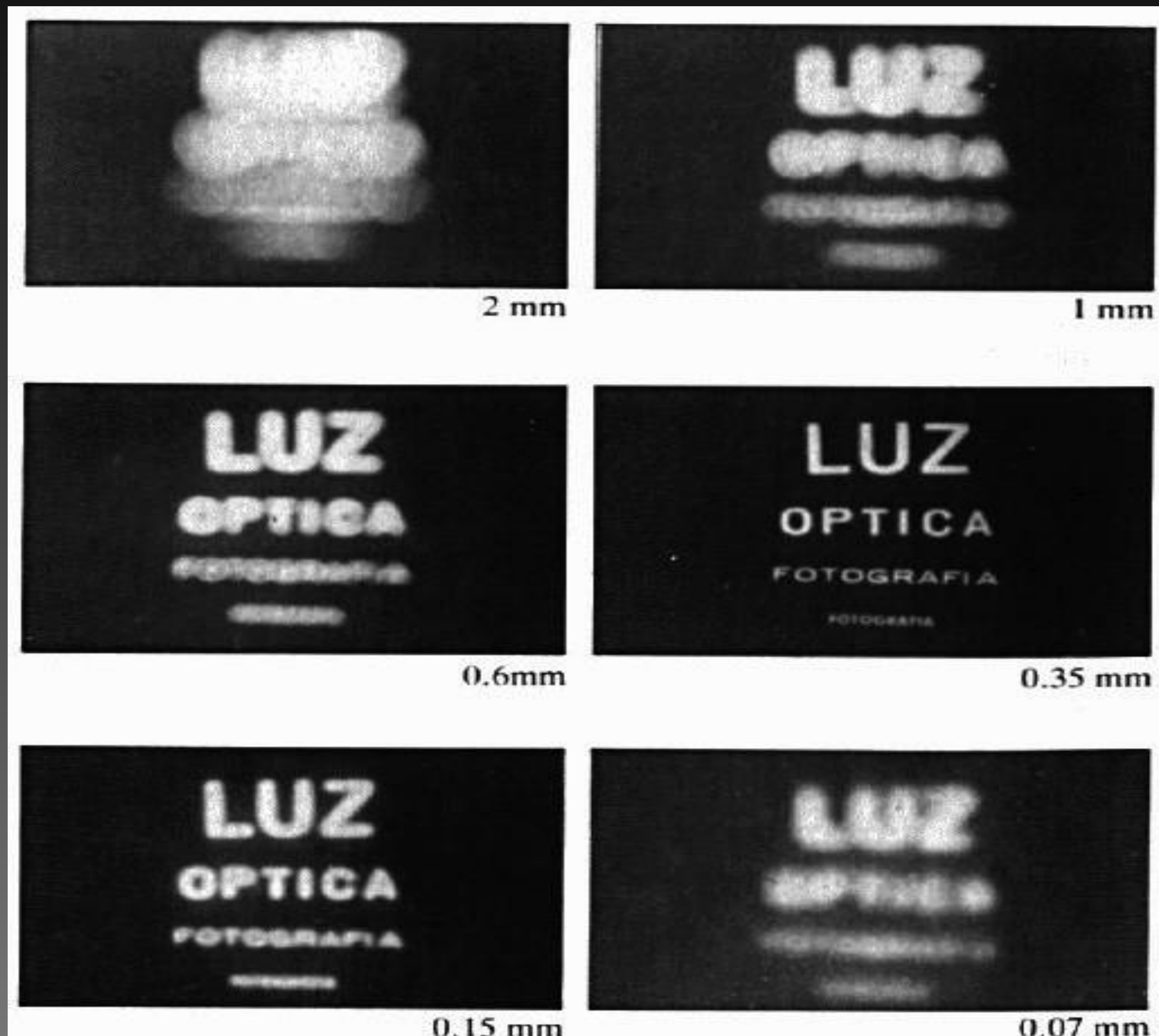
- Aperture too big: blurry image
- Aperture too small: requires long exposure or high intensity
- Aperture much too small: diffraction through pinhole $\Rightarrow$ blurry image

How does the aperture size affect the image?



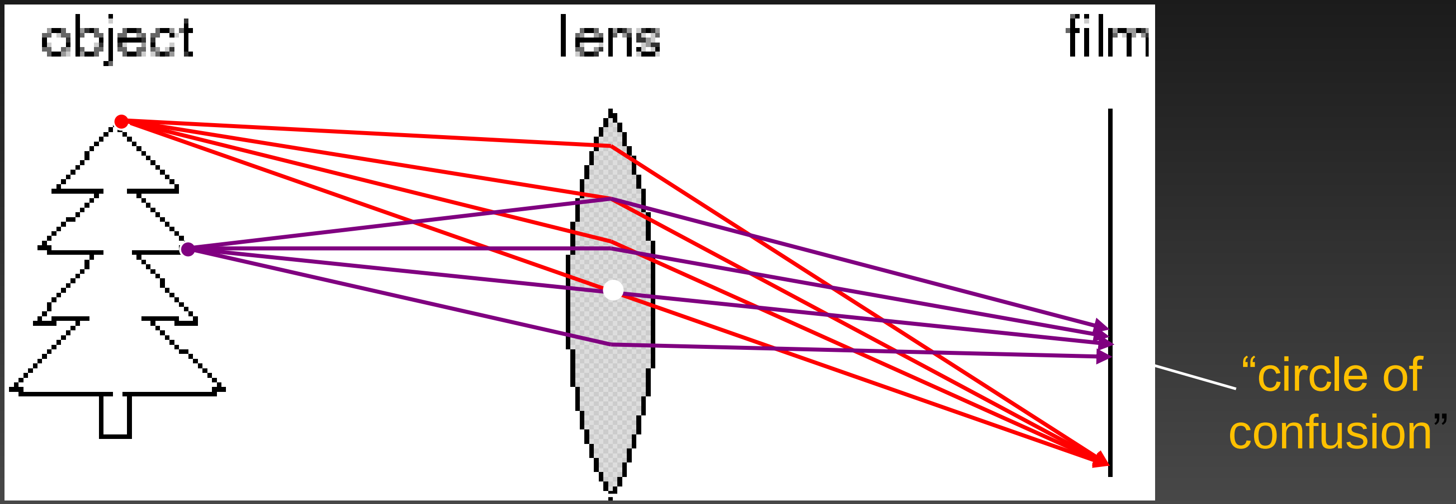
- **Why not make the aperture as small as possible?**
  - Less light gets through
  - *Diffraction（衍射）* effects...

Shrinking the aperture



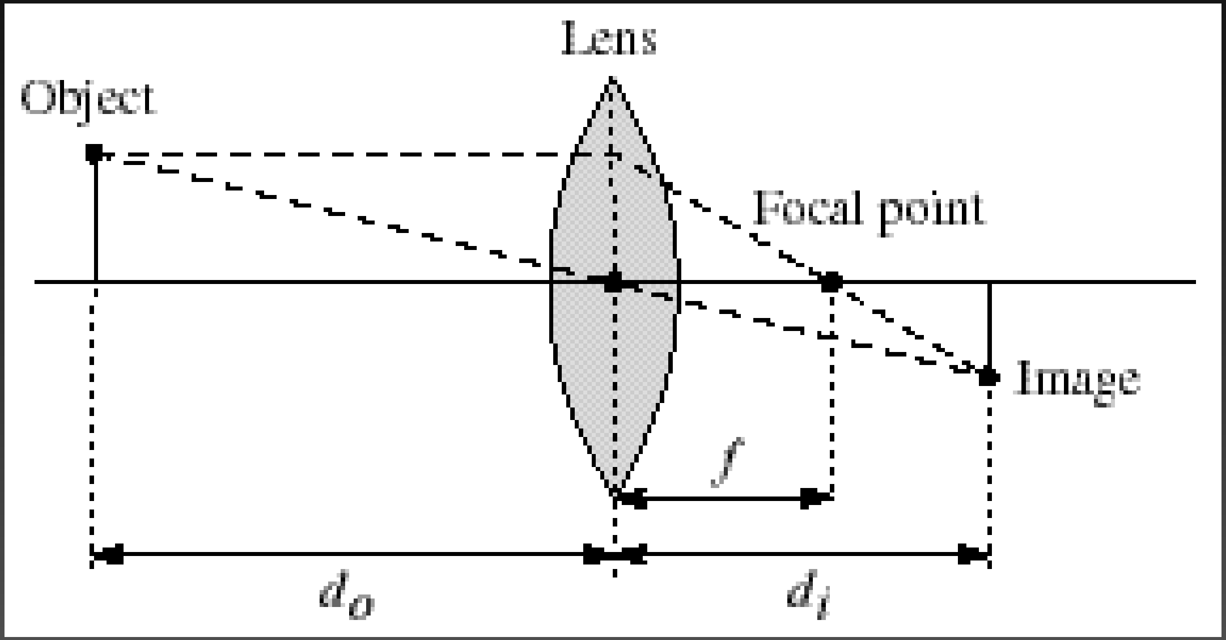
Building a real camera

Adding a lens



- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
    - other points project to a “circle of confusion” in the image
  - Changing the shape of the lens changes this distance

Thin lenses

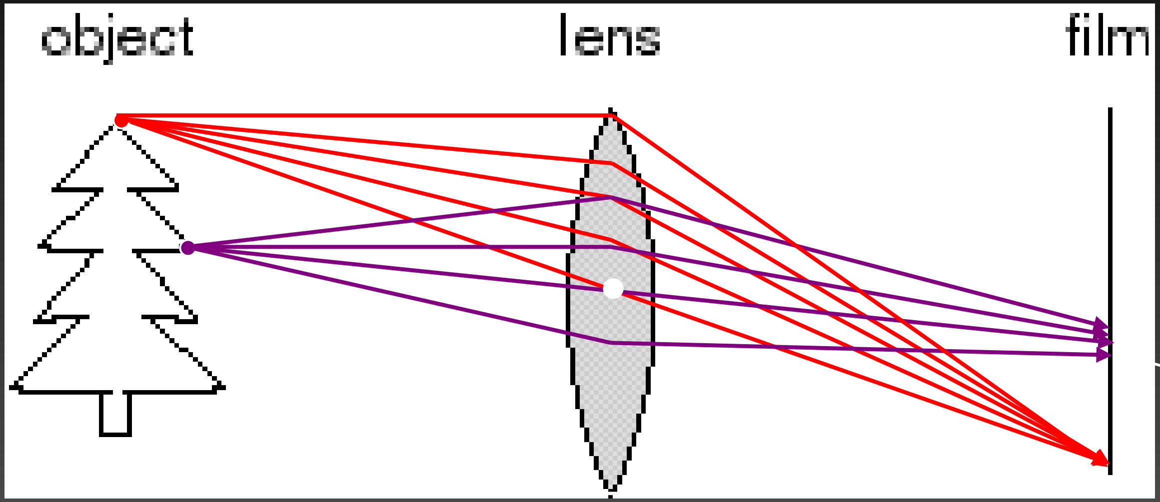


- **Thin lens equation:**

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

  - **Any object point satisfying this equation is in focus**
  - **What is the shape of the focus region?**
  - **How can we change the focus region?**

Adding a lens

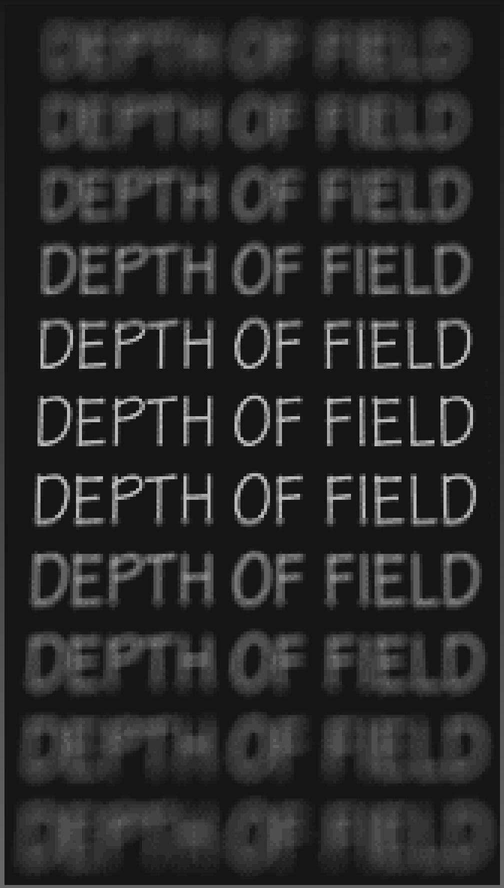


"circle of confusion"

- A lens focuses light onto the film
  - There is a specific distance at which objects are "in focus"
    - other points project to a "circle of confusion" in the image

Slide by Steve Seitz

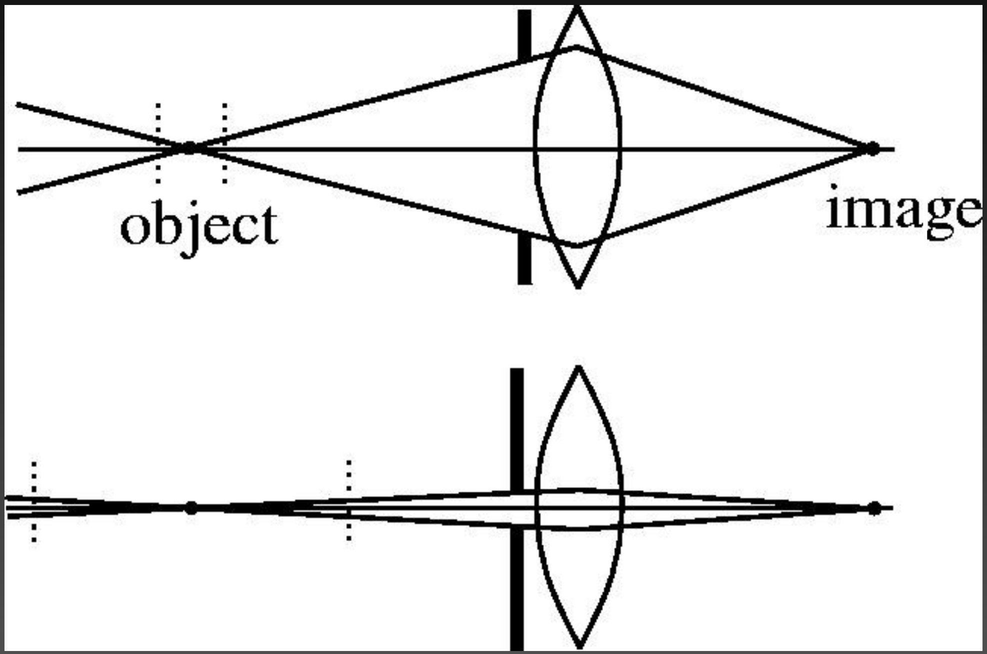
Depth of Field



http://www.cambridgeincolour.com/tutorials/depth-of-field.htm

Slide by A. Efros

Depth of field



f / 5.6

f / 32

- **Changing the aperture size affects depth of field**
  - **A smaller aperture increases the range in which the object is approximately in focus**

Flower images from Wikipedia http://en.wikipedia.org/wiki/Depth_of_field

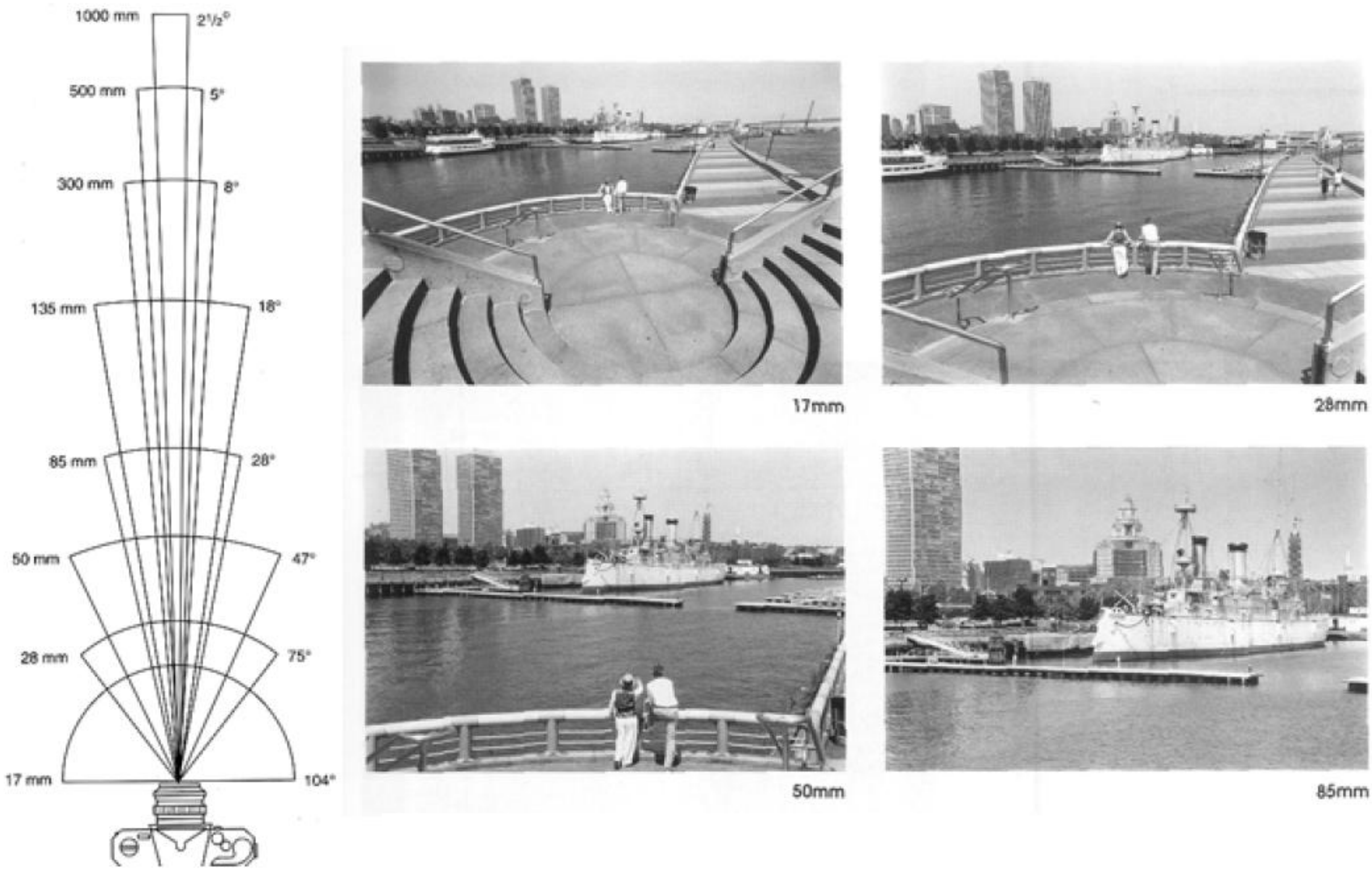
Varying the aperture

Large aperture = small DOF

Small aperture = large DOF

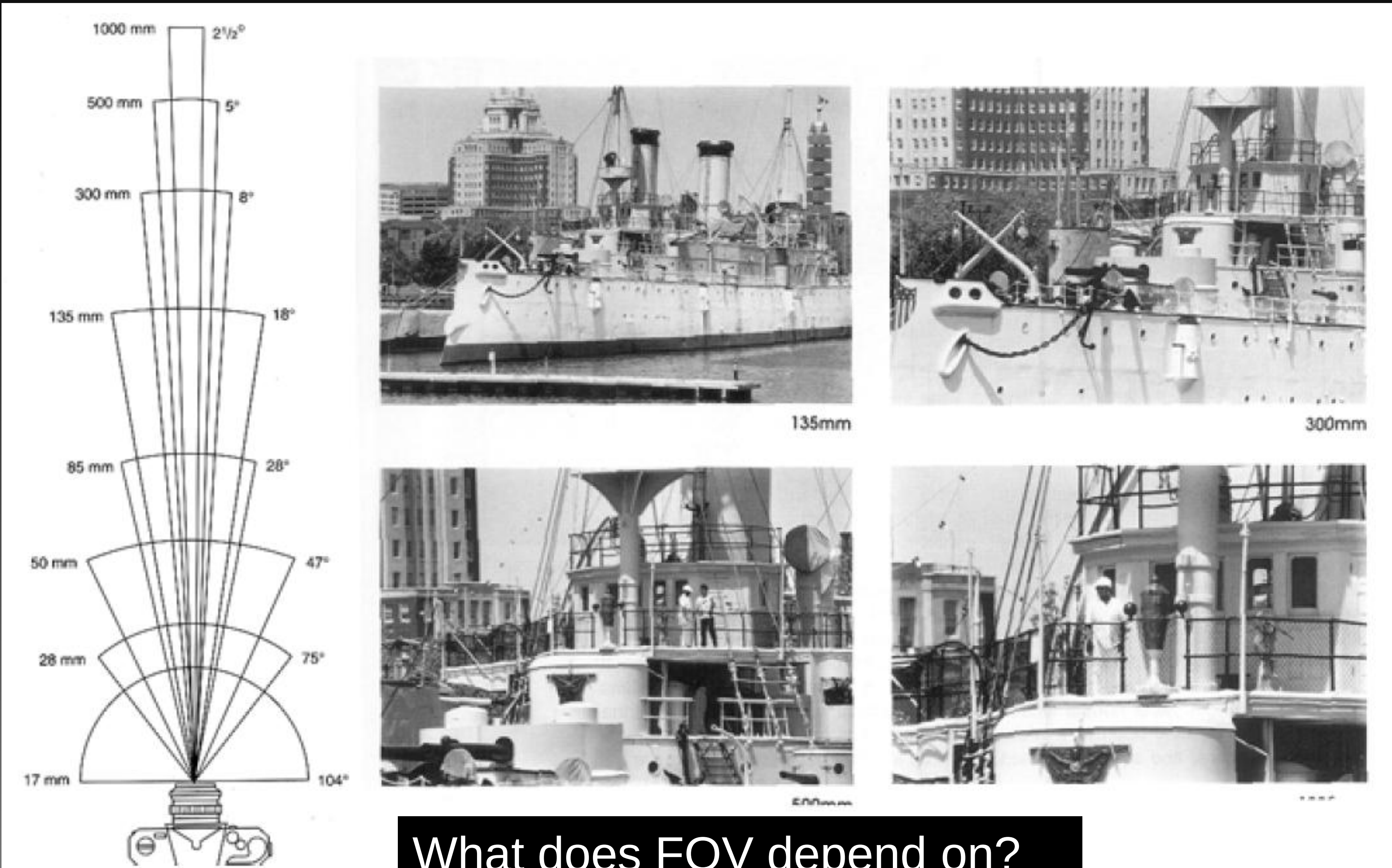
Slide by A. Efros

Field of View



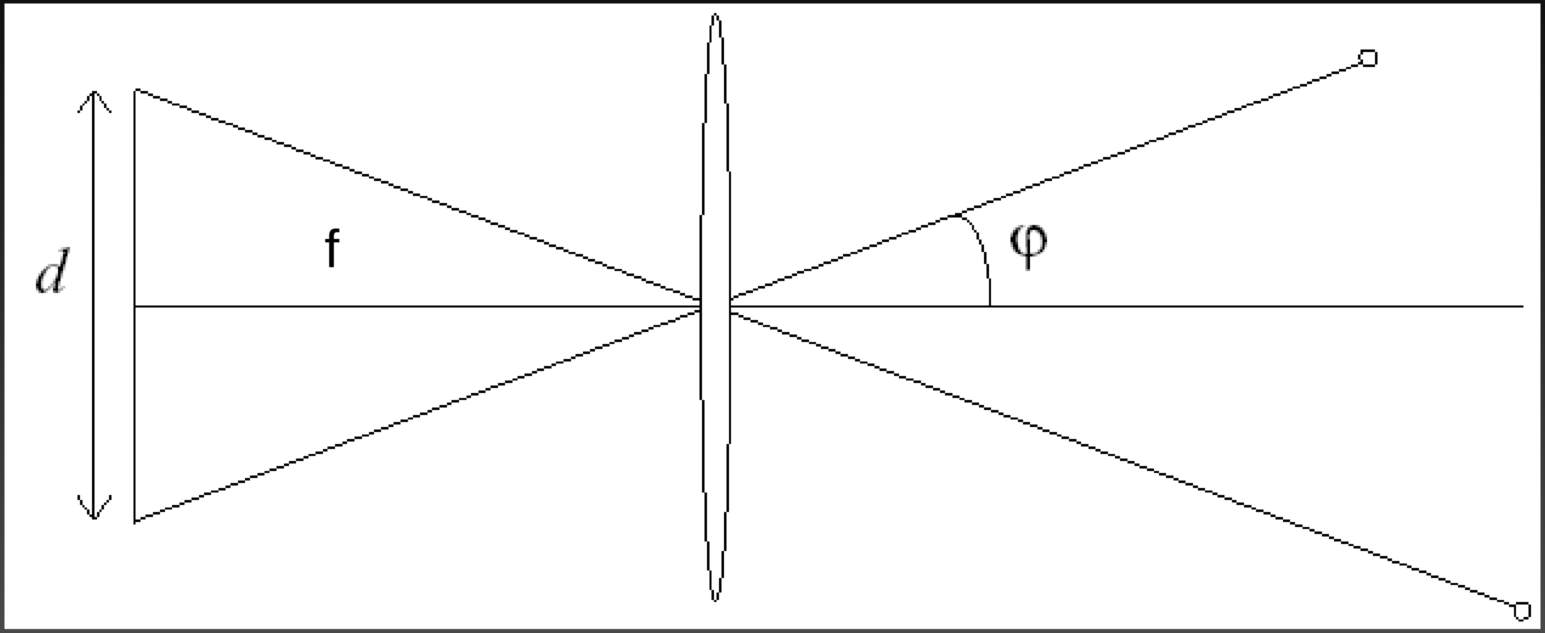
Slide by A. Efros

Field of View



What does FOV depend on?

Field of View



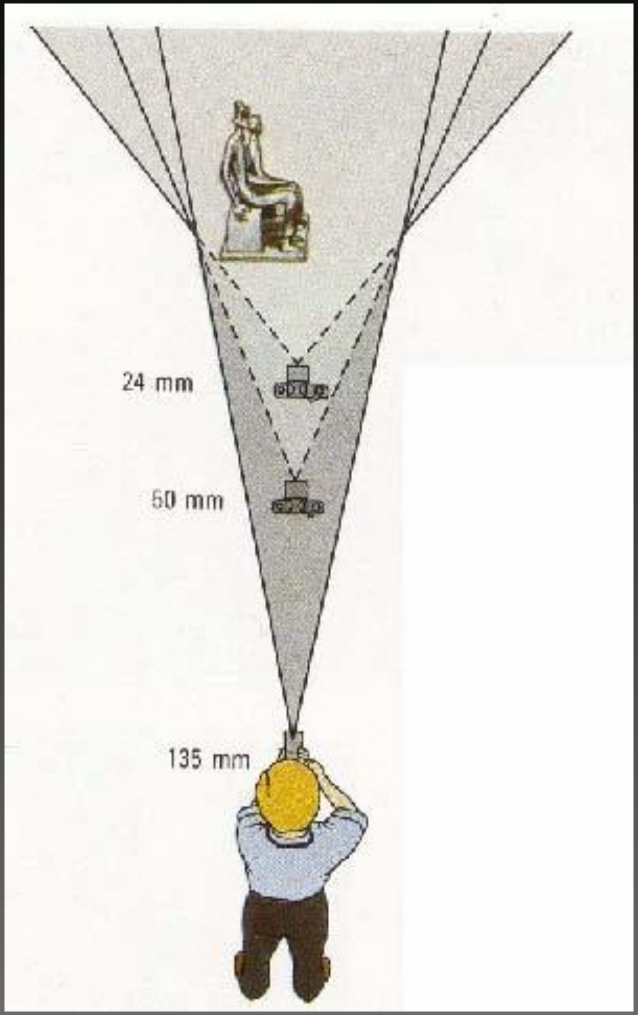
FOV depends on focal length and size of the camera retina

$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

Smaller FOV = larger Focal Length

Slide by A. Efros

Field of View / Focal Length

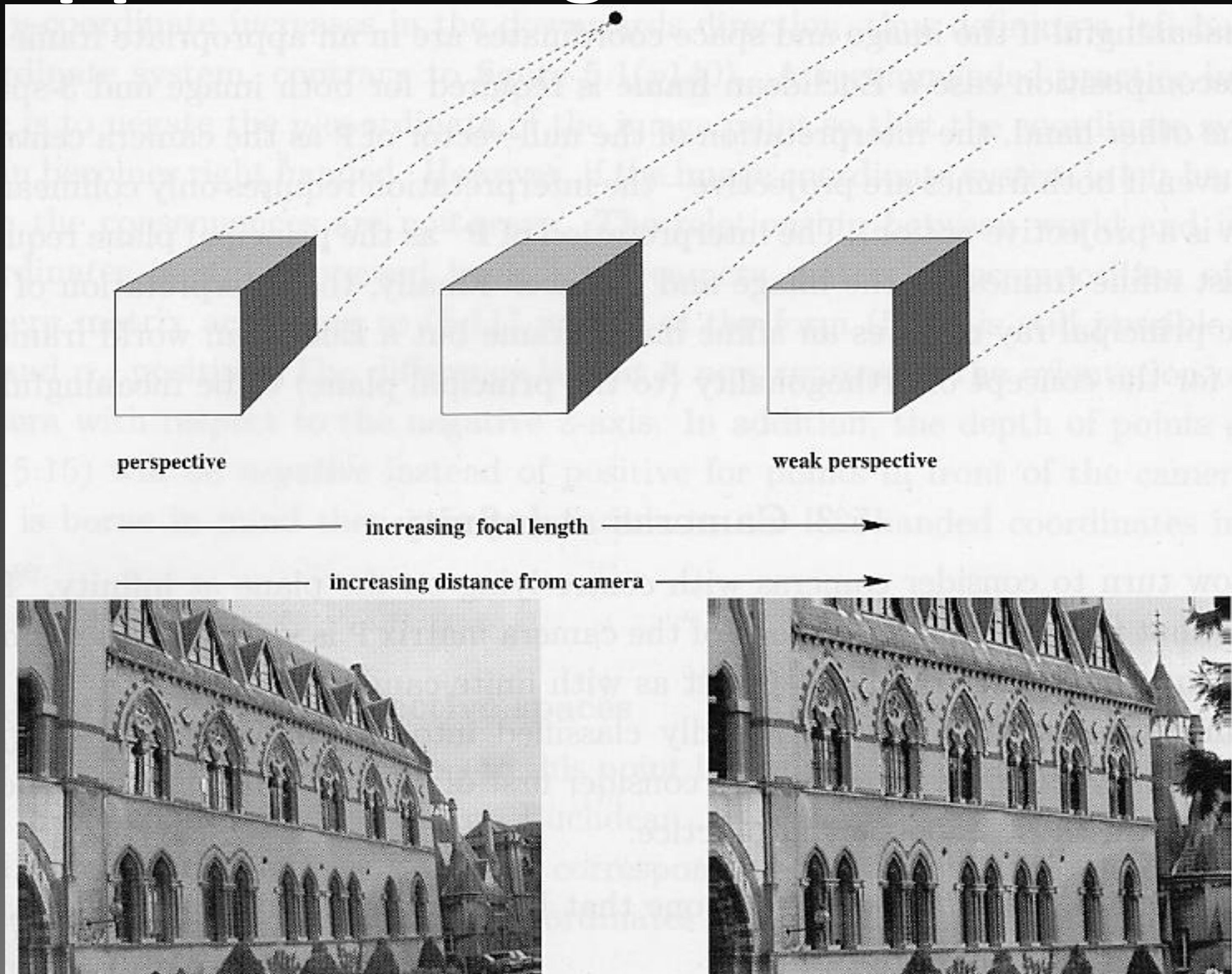


Large FOV, small f
Camera close to car

Small FOV, large f
Camera far from the car

Sources: A. Efros, F. Durand

Approximating an affine camera

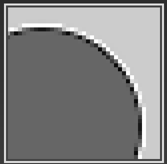


Source: Hartley & Zisserman

Digital camera artifacts

- **Noise**
  - low light is where you most notice noise
  - light sensitivity (ISO) / noise tradeoff
  - stuck pixels

- **In-camera processing**
  - oversharpening can produce halos

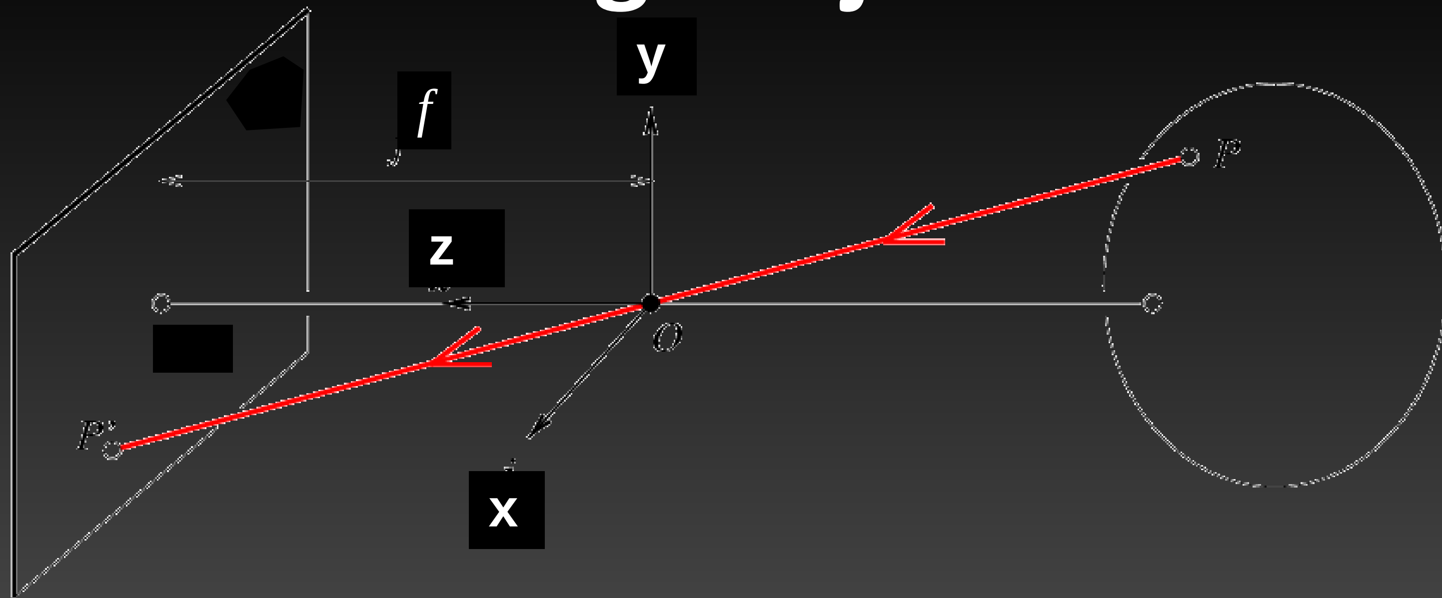


- **Compression**
  - JPEG artifacts, blocking
- **Blooming**
  - charge overflowing into neighboring pixels
- **Color artifacts**
  - purple fringing from microlenses,
  - white balance

Slide by Steve Seitz

Pinhole camera image

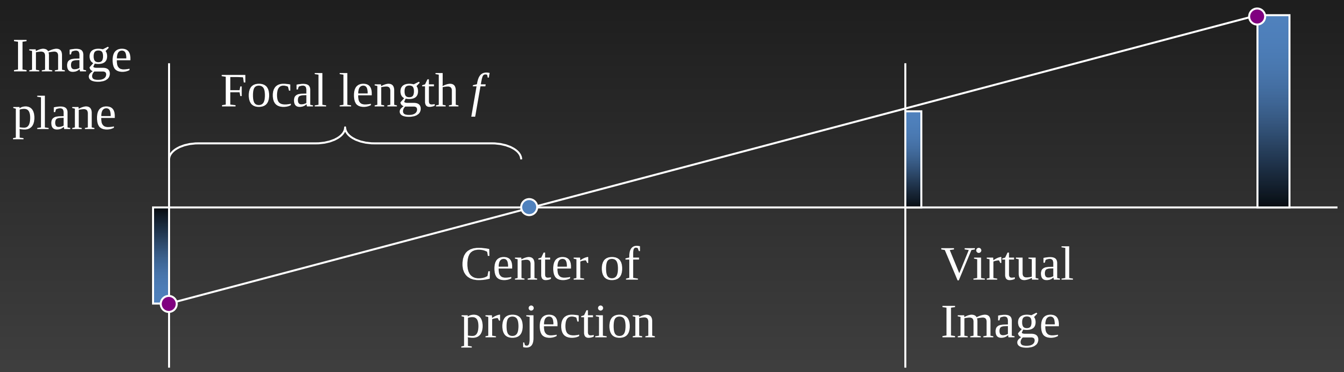
Modeling Projection



- **The coordinate system**
  - The optical center (*O*) is at the origin
  - The image plane is parallel to xy-plane (perpendicular to z axis)

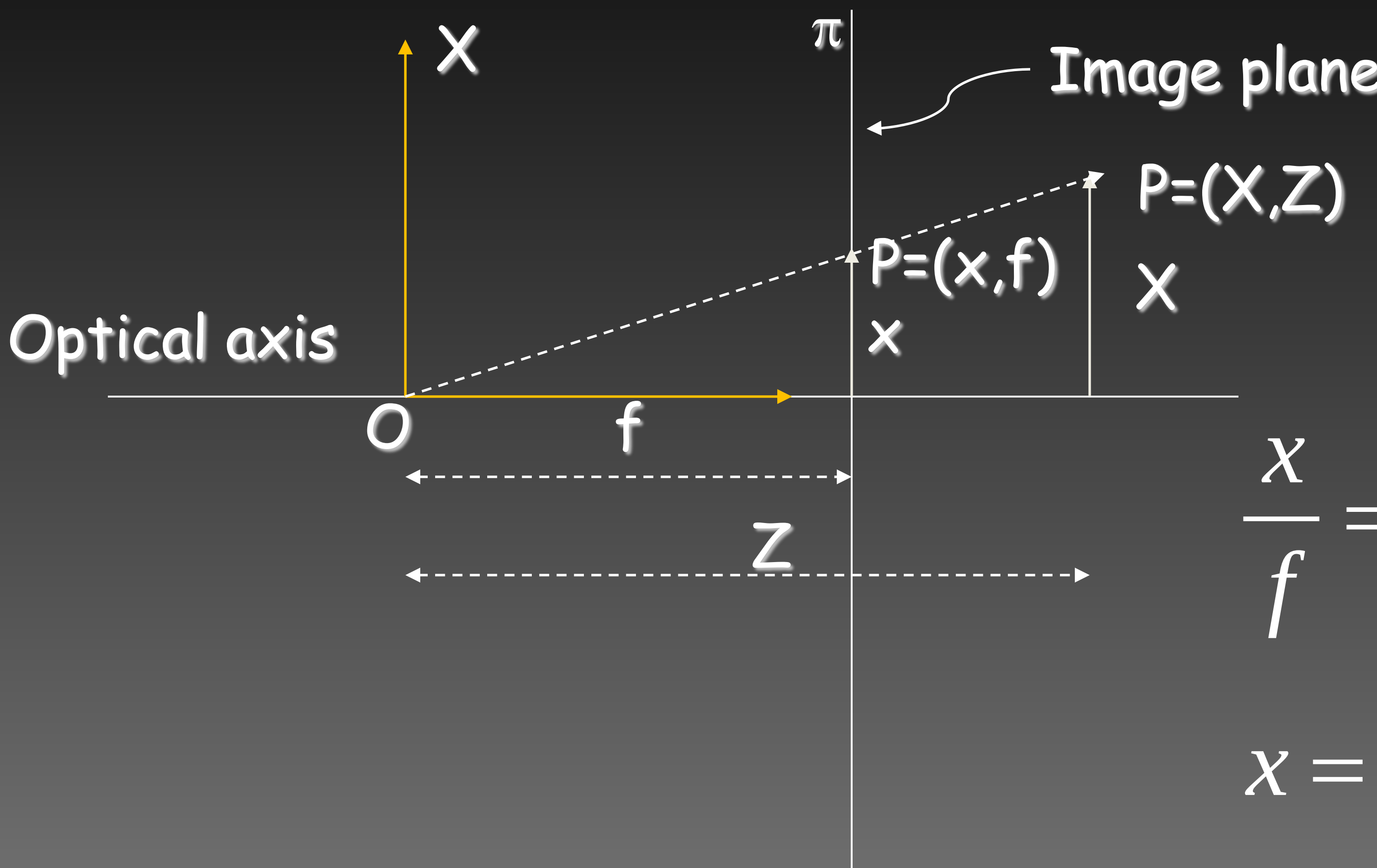
Source: J. Ponce, S. Seitz

Pinhole Camera Model



When we take a picture, we lose a dimension of information:
(X, Y, Z) 3-D coordinates of a point
(x, y) 2-D projection (image coordinates)
f focal length of the camera

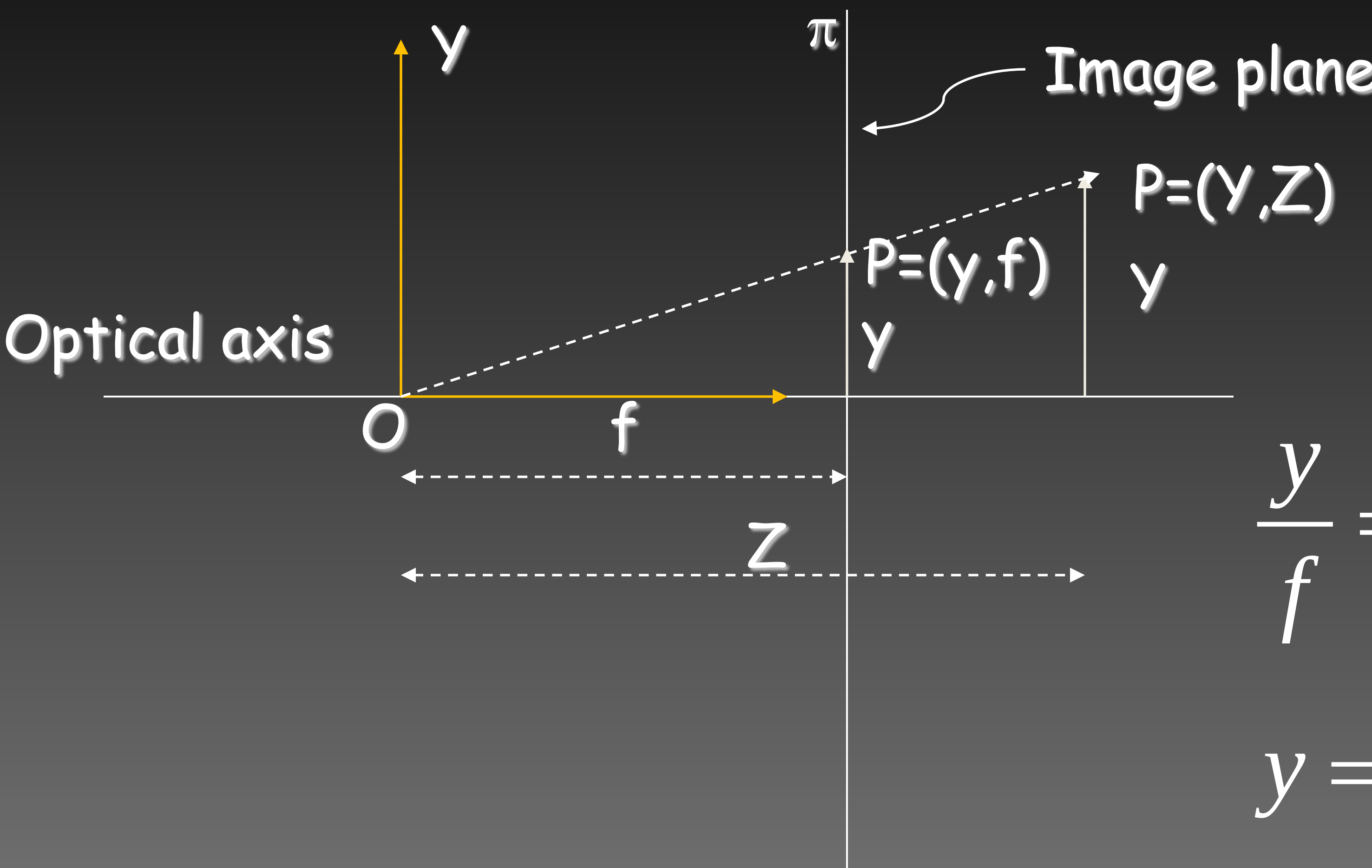
Pinhole Camera Model



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

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

Pinhole Camera Model



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

$$y = f\,\frac{Y}{Z}$$

Perspective Equations

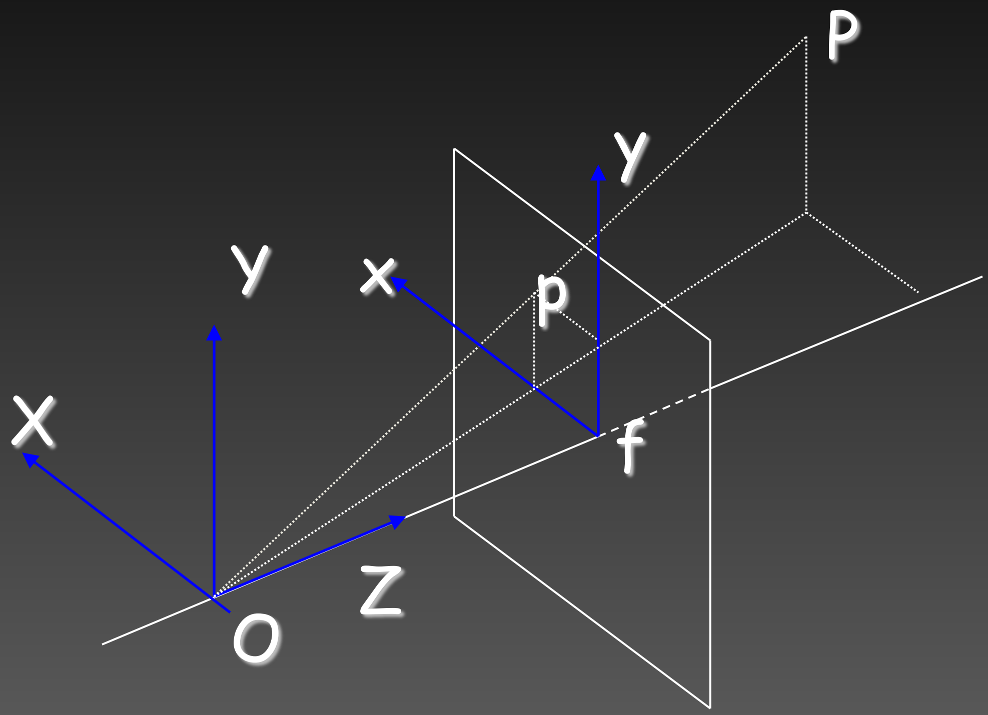
- Given point P(X,Y,Z) in the 3D world
- The two equations:

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

- transform world coordinates (X,Y,Z) into image coordinates (x,y)

Pinhole Camera Model

(Camera Coordinates)



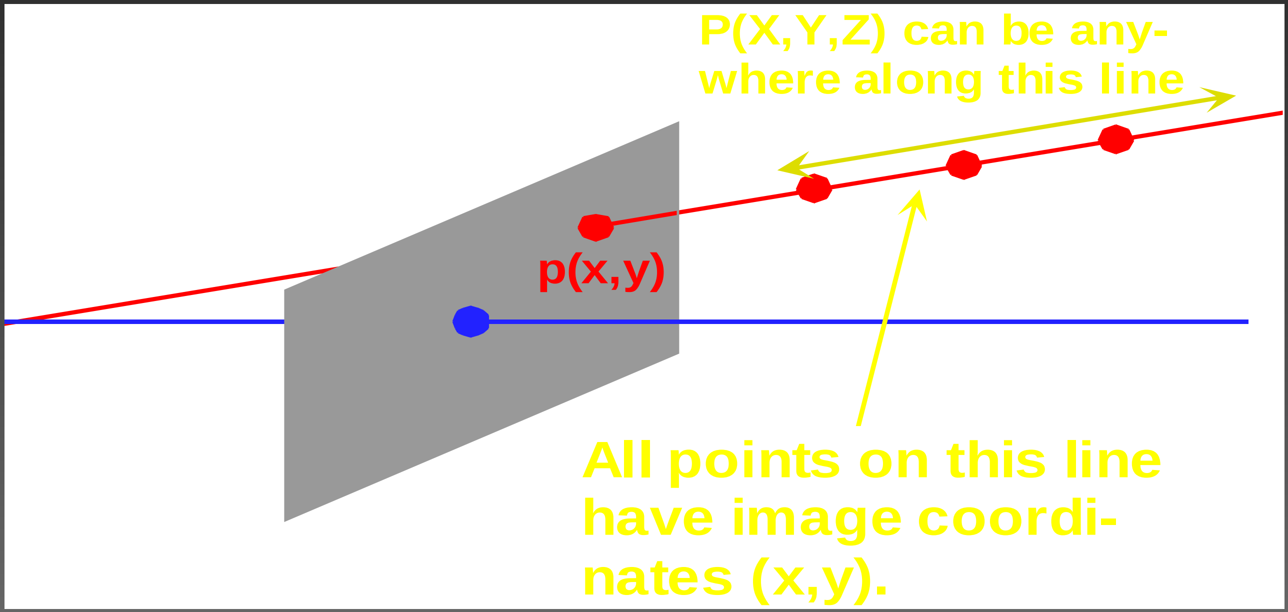
$$x = f\,\frac{X}{Z}$$

$$y = f\,\frac{Y}{Z}$$

- Non-linear equations
- Any point on the ray OP has image p !!

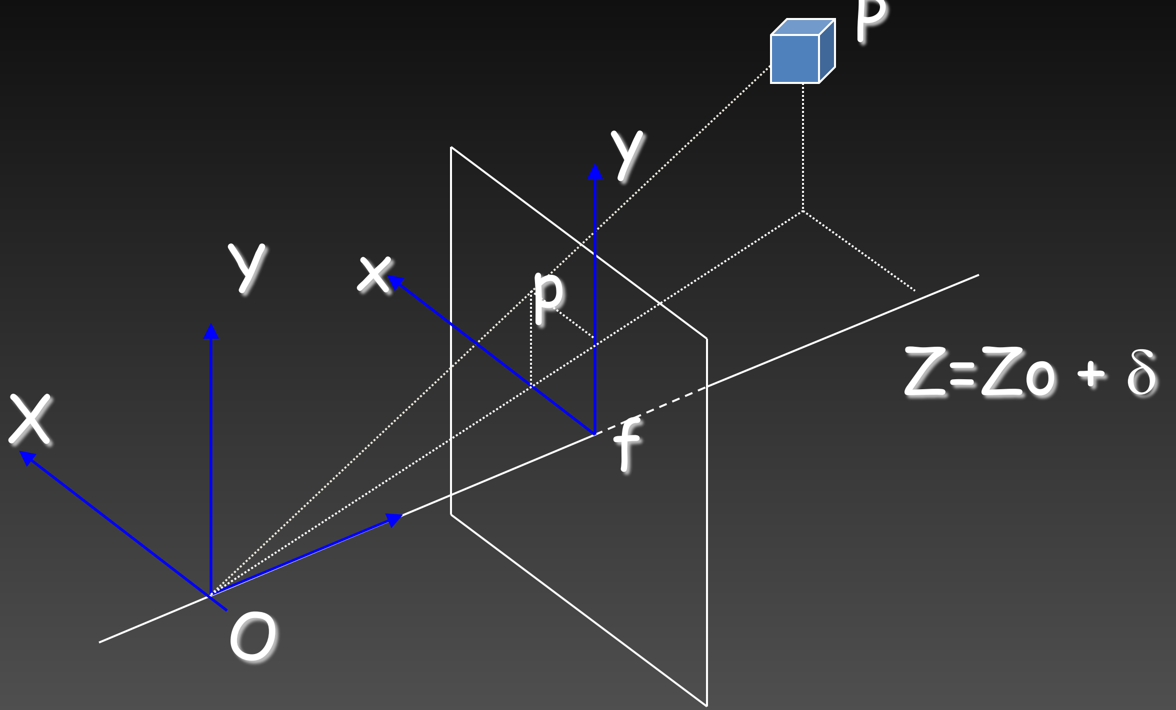
Reverse Projection

- **Given a center of projection and image coordinates of a point, it is not possible to recover the 3D depth of the point from a single image.**



In general, at least two images of the same point taken from two different locations are required to recover depth.

Weak Perspective Model

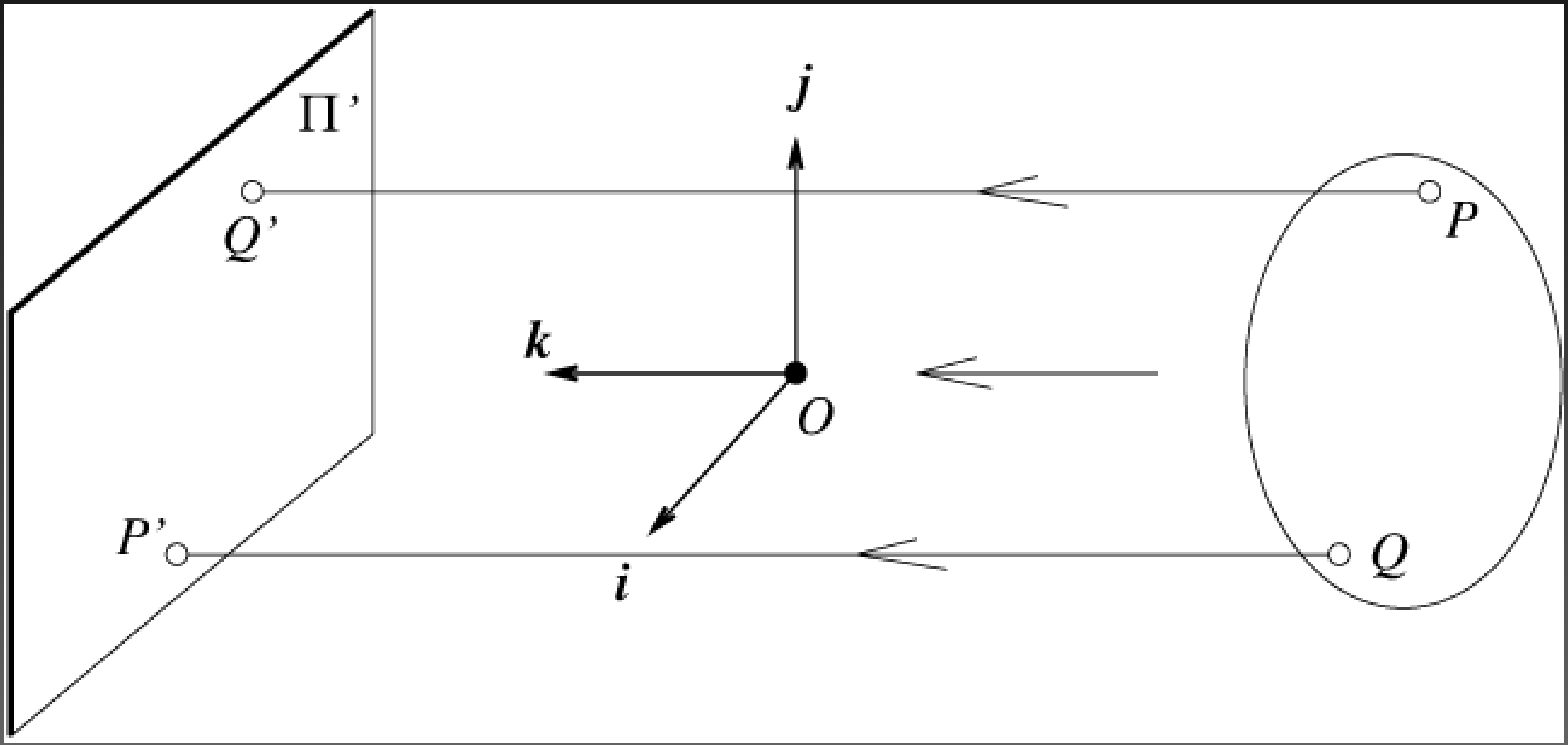


- Object depth δ << Camera distance Zo
- Linear equations !!

$$x \approx f \frac{X}{Z_o}$$

$$y \approx f \frac{Y}{Z_o}$$

Orthographic projection



the viewpoint (sensor) is located at infinity relative to the objects being imaged

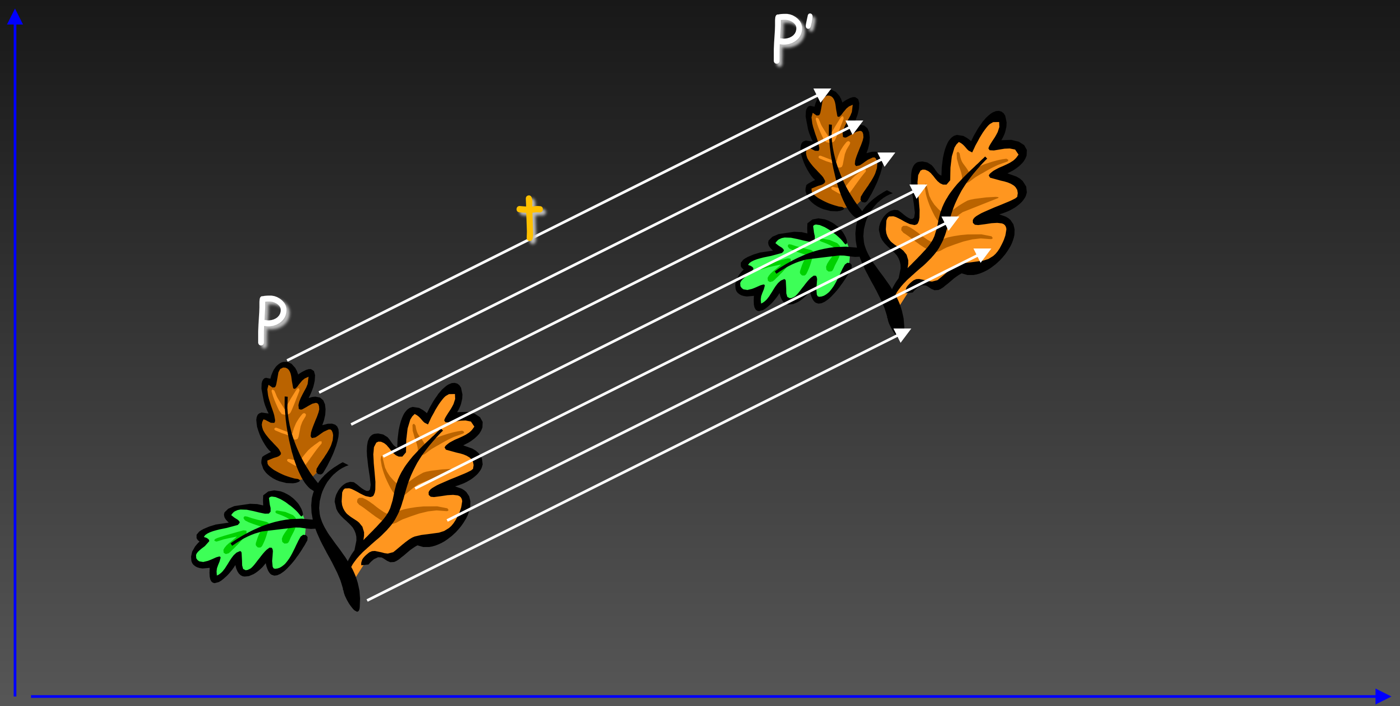
Pros and Cons of These Models

- Weak perspective much simpler math.
  - Accurate when object is small and distant.
  - Most useful for recognition.
- Pinhole perspective much more accurate for scenes.
  - Used in structure from motion.
- When accuracy really matters, must model real cameras.

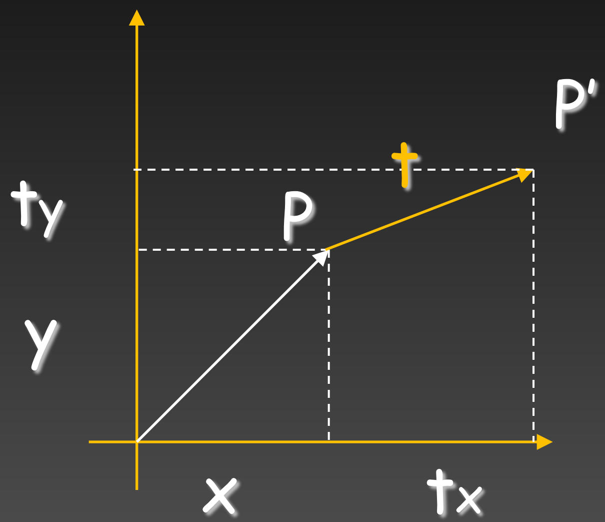
- **Homogeneous Coordinates:**

A general View

2D Translation



2D Translation Equation

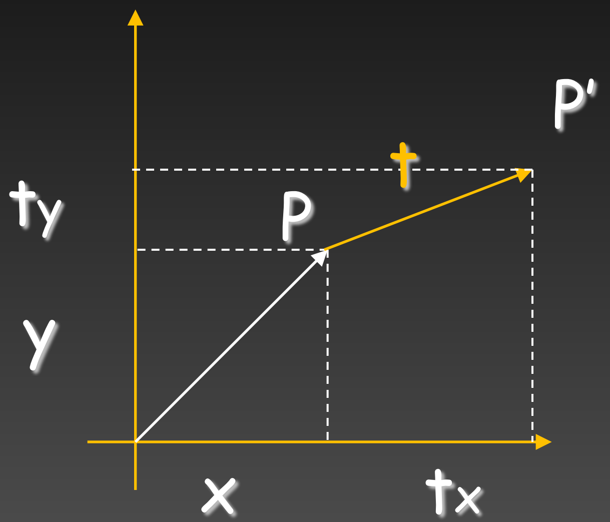


$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' = (x + t_x, y + t_y) = \mathbf{P} + \mathbf{t}$$

2D Translation using Matrices



$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & \overset{t}{\boxed{\begin{matrix} t_x \\ t_y \end{matrix}}} \end{bmatrix} \cdot \begin{bmatrix} \overset{P}{\boxed{\begin{matrix} x \\ y \end{matrix}}} \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- **Multiply the coordinates by a non-zero scalar and add an extra coordinate equal to that scalar. For example,**

$$(x, y) \rightarrow (x \cdot z, y \cdot z, z) \quad z \neq 0$$

$$(x, y, z) \rightarrow (x \cdot w, y \cdot w, z \cdot w, w) \quad w \neq 0$$

- NOTE: If the scalar is 1, there is no need for the multiplication!

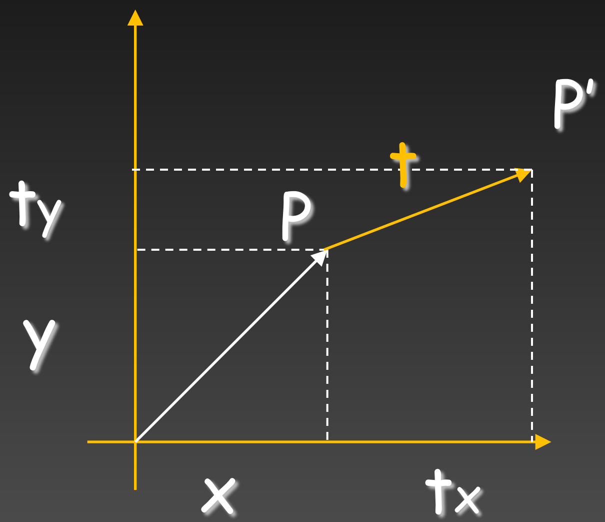
Back to Cartesian Coordinates:

- Divide by the last coordinate and eliminate it. For example,

$$(x, y, z) \quad z \neq 0 \rightarrow (x/z, y/z)$$

$$(x, y, z, w) \quad w \neq 0 \rightarrow (x/w, y/w, z/w)$$

2D Translation using Homogeneous Coordinates



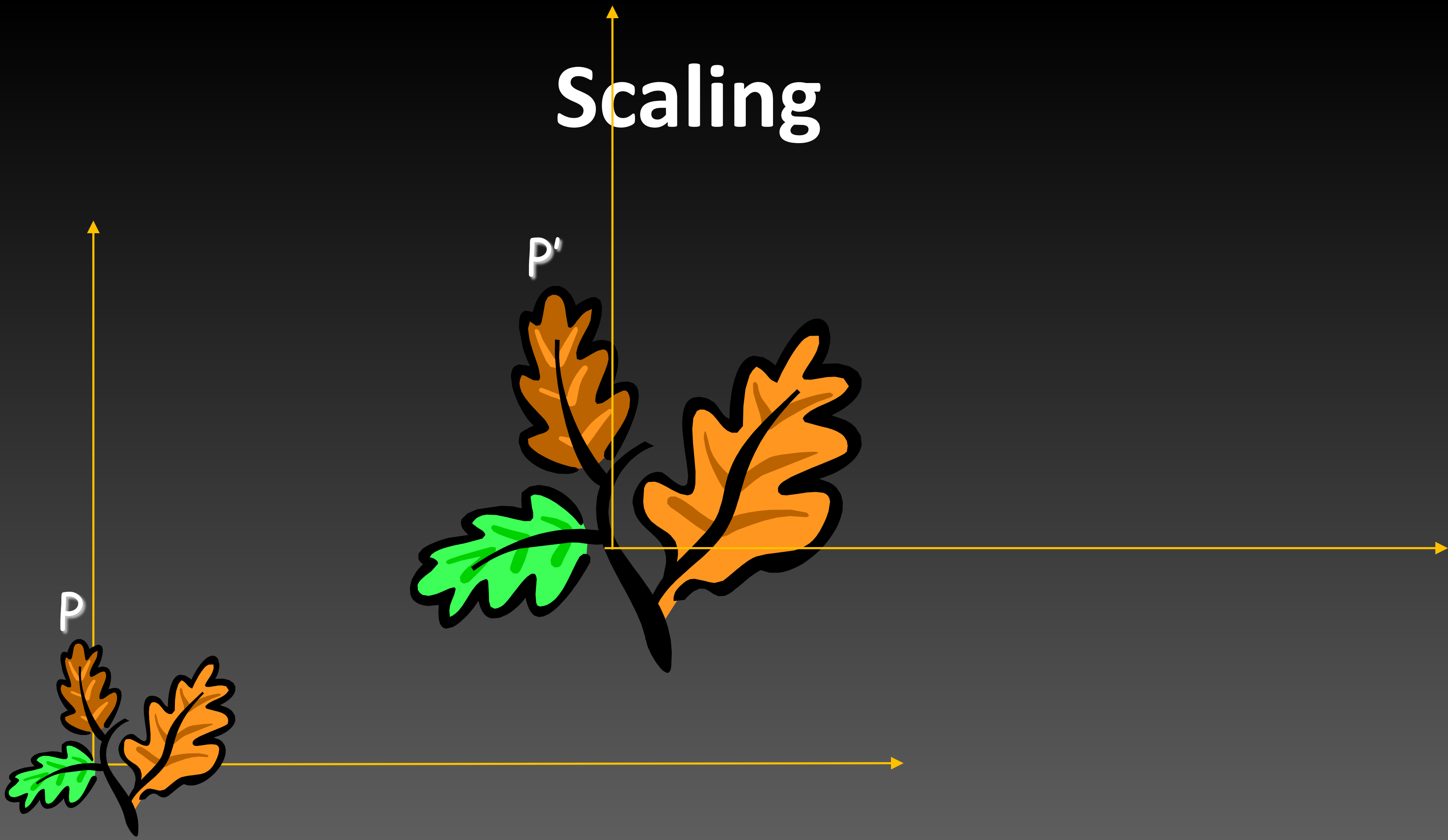
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

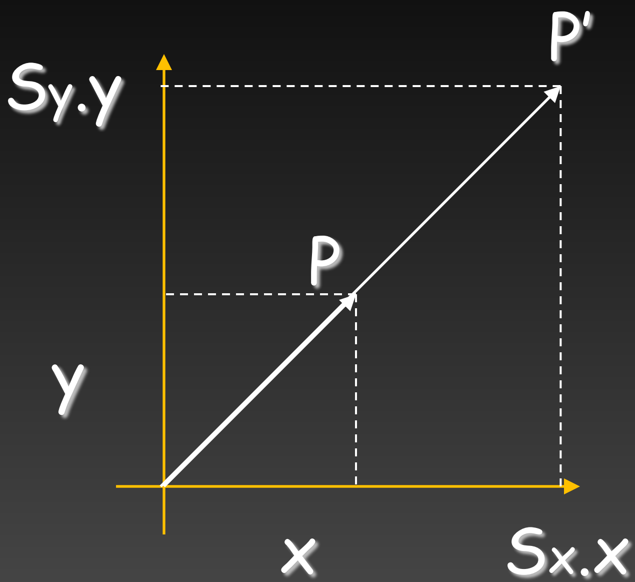
$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \cdot \overset{\mathbf{P}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P}$$

Scaling



Scaling Equation



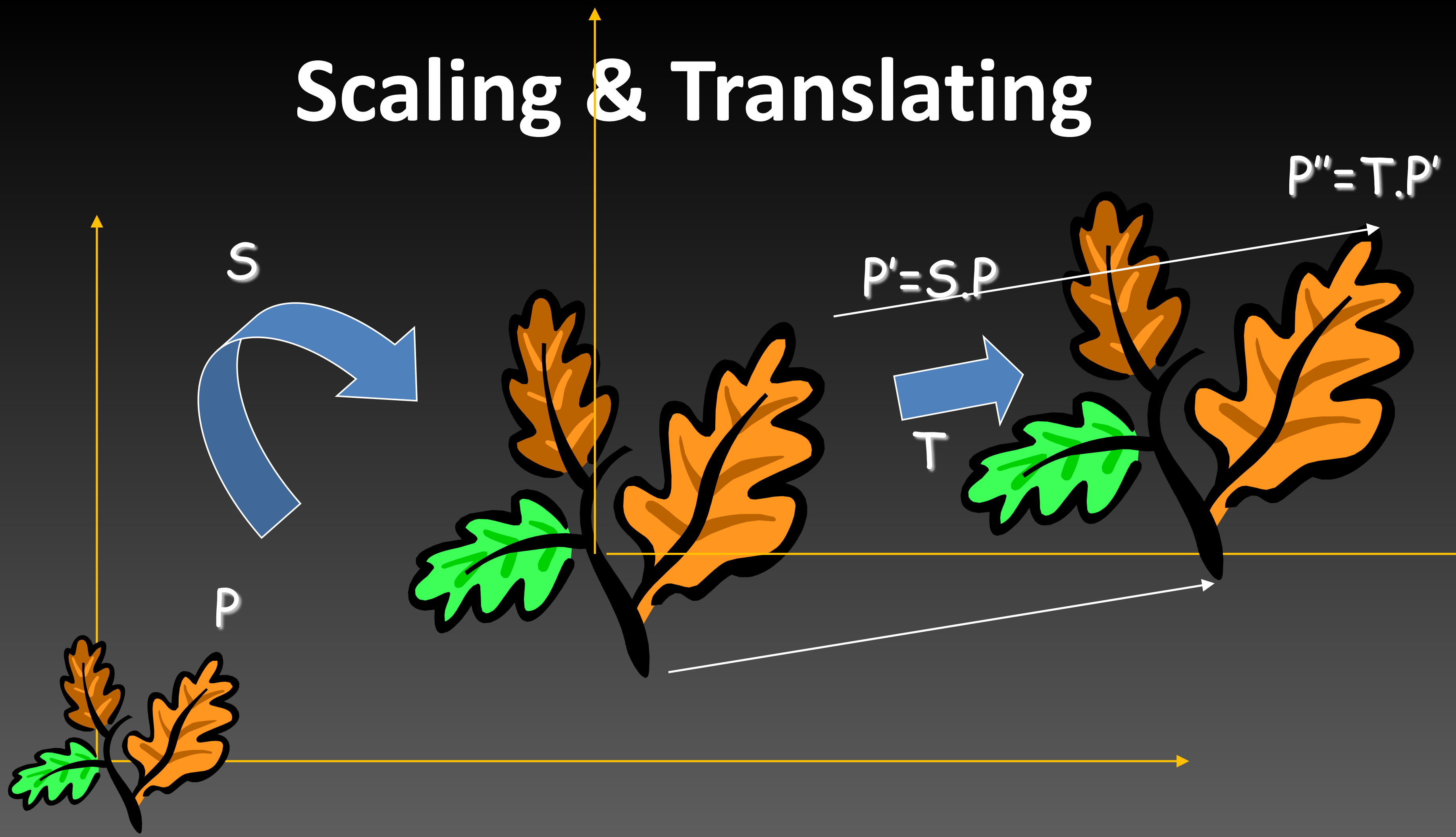
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

Scaling & Translating



P''=T.P'=T.(S.P)=(T.S).P

Scaling & Translating

P''=T.P'=T.(S.P)=(T.S).P

$$\mathbf{P''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

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

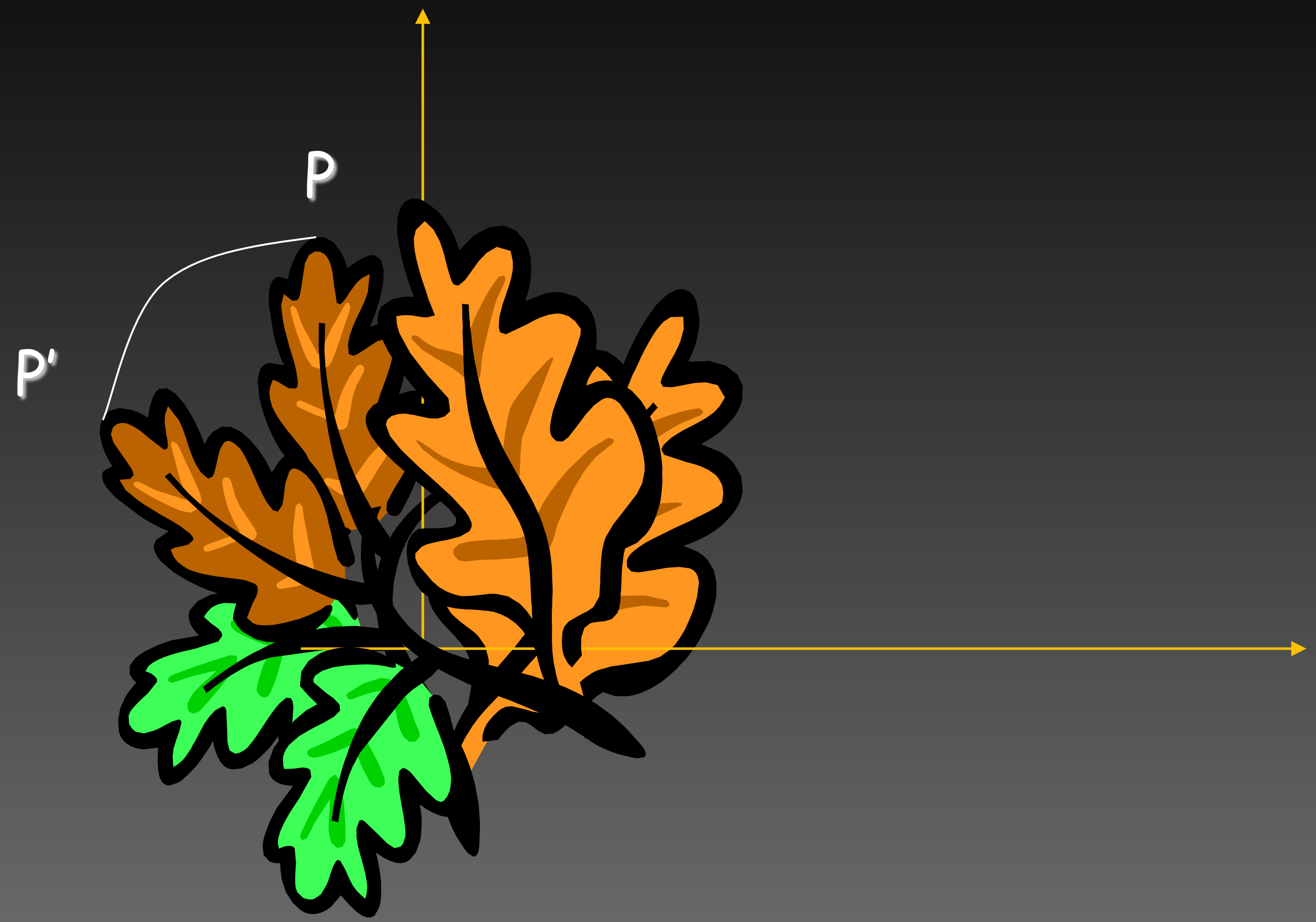
Translating & Scaling ≠ Scaling & Translating

P''=S.P'=S.(T.P)=(S.T).P

$$\mathbf{P''} = \mathbf{S}\cdot\mathbf{T}\cdot\mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

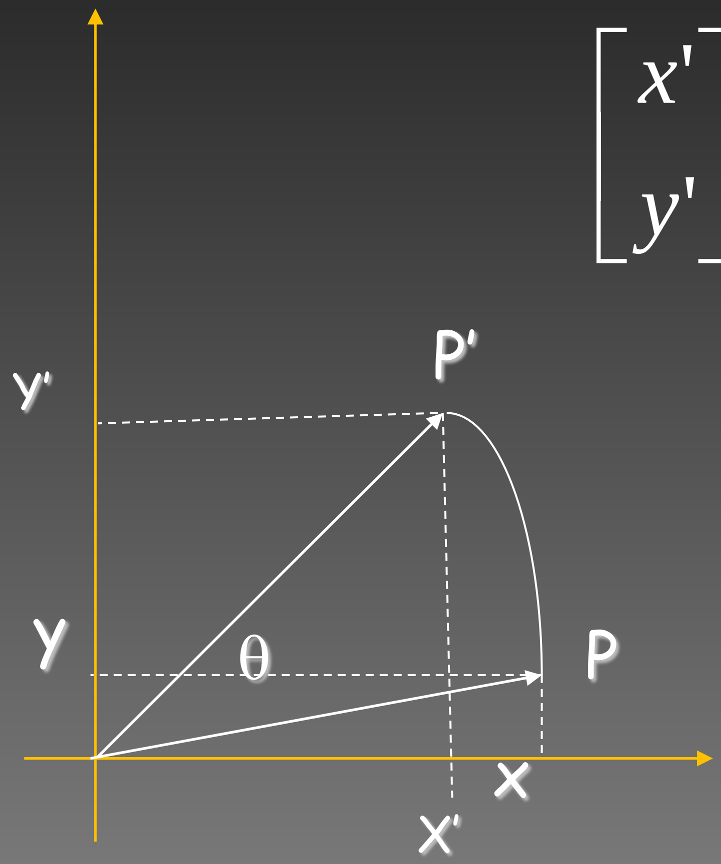
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Rotation



Rotation Equations

Counter-clockwise rotation by an angle θ



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P'} = \mathbf{R.P}$$

Degrees of Freedom

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

R is 2x2 $\Rightarrow$ 4 elements

BUT! There is only 1 degree of freedom: θ

The 4 elements must satisfy the following constraints:

$$\mathbf{R} \cdot \mathbf{R}^{\mathbf{T}} = \mathbf{R}^{\mathbf{T}} \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

Scaling, Translating & Rotating

Order matters!

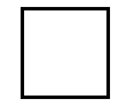
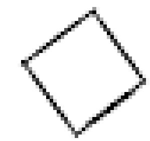
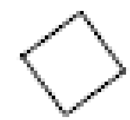
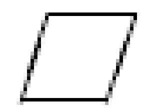
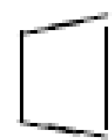
$P' = S.P$

$P''=T.P'=(T.S).P$

$P'''=R.P''=R.(T.S).P=(R.T.S).P$

$R.T.S \neq R.S.T \neq T.S.R \ldots$

Summary of planar geometry

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	straight lines	

3D Space: 4-element vector

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \text{ where } w \neq 0 \text{ and typically } w = 1$$

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \sin\phi\cos\psi\sin\theta - \cos\phi\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta & \sin\phi\cos\theta \\ \cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi & \cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$

Summary of 3D geometry

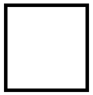
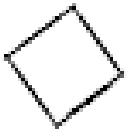
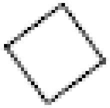
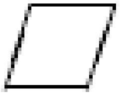
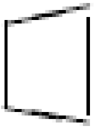
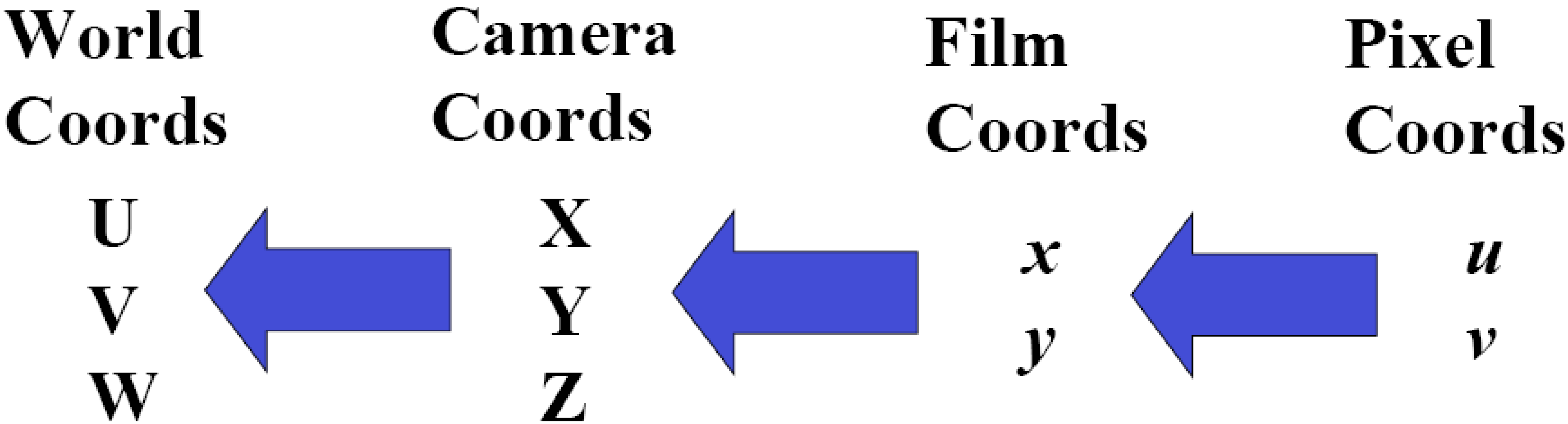
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{3\times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{3\times 4}$	6	lengths	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{3\times 4}$	7	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{3\times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{4\times 4}$	15	straight lines	

Image Formation Coordinate Transformation

World Coords		Camera Coords		Film Coords		Pixel Coords
U	$\Rightarrow$	X	$\Rightarrow$	x	$\Rightarrow$	u
V		Y		y		v
W		Z				

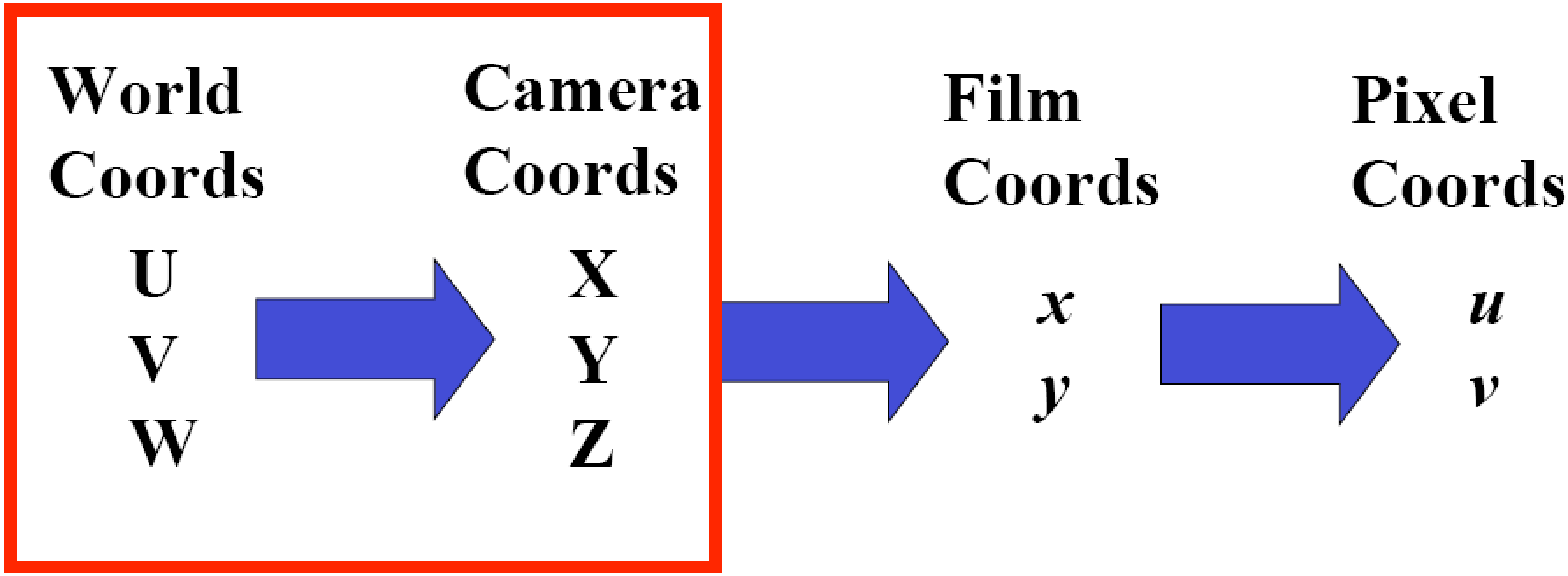
We want a mathematical model to describe how 3D World points get projected into 2D Pixel coordinates.

Image Formation Coordinate Transformation



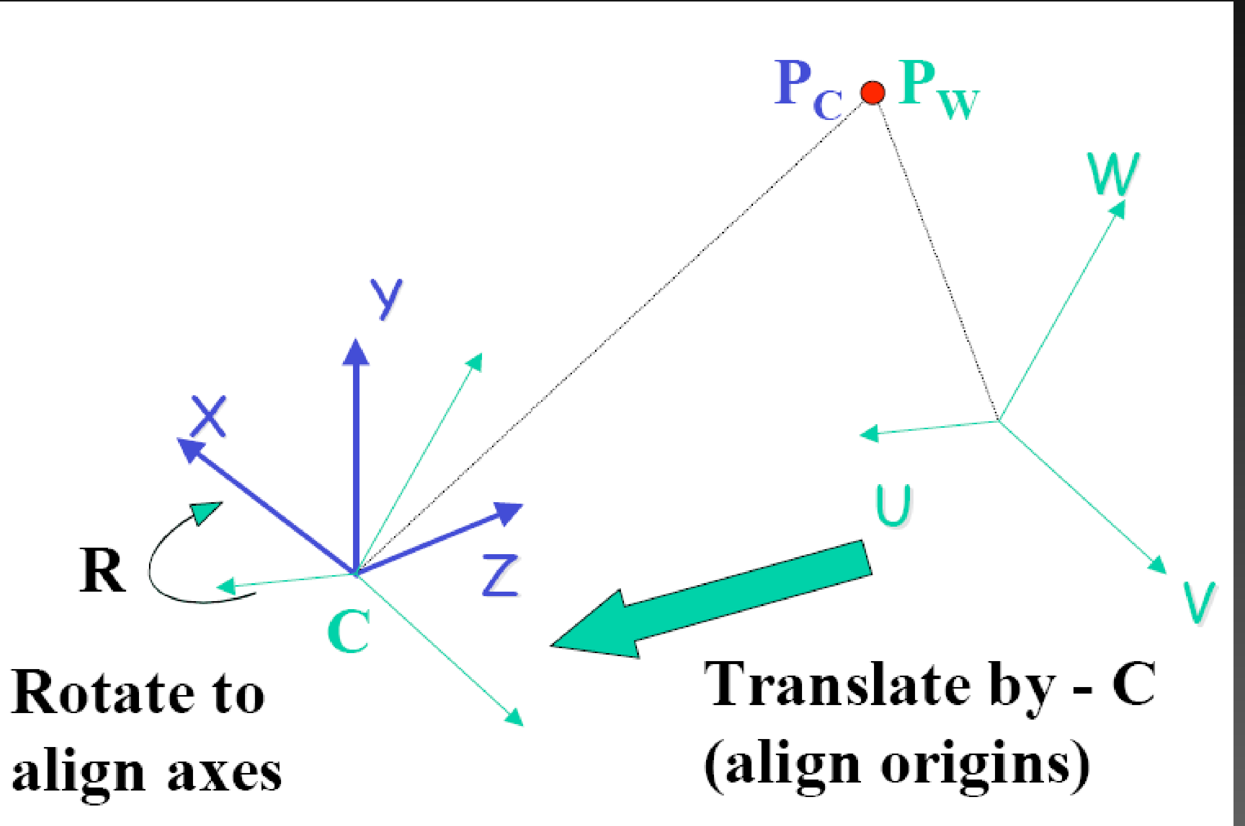
Soon, we will talk about backward projection to recover 3D scene structure from images (via stereo or motion)

World to Camera Transformation



Rigid Transformation (rotation+translation) between world and camera coordinate systems

World to Camera Transformation



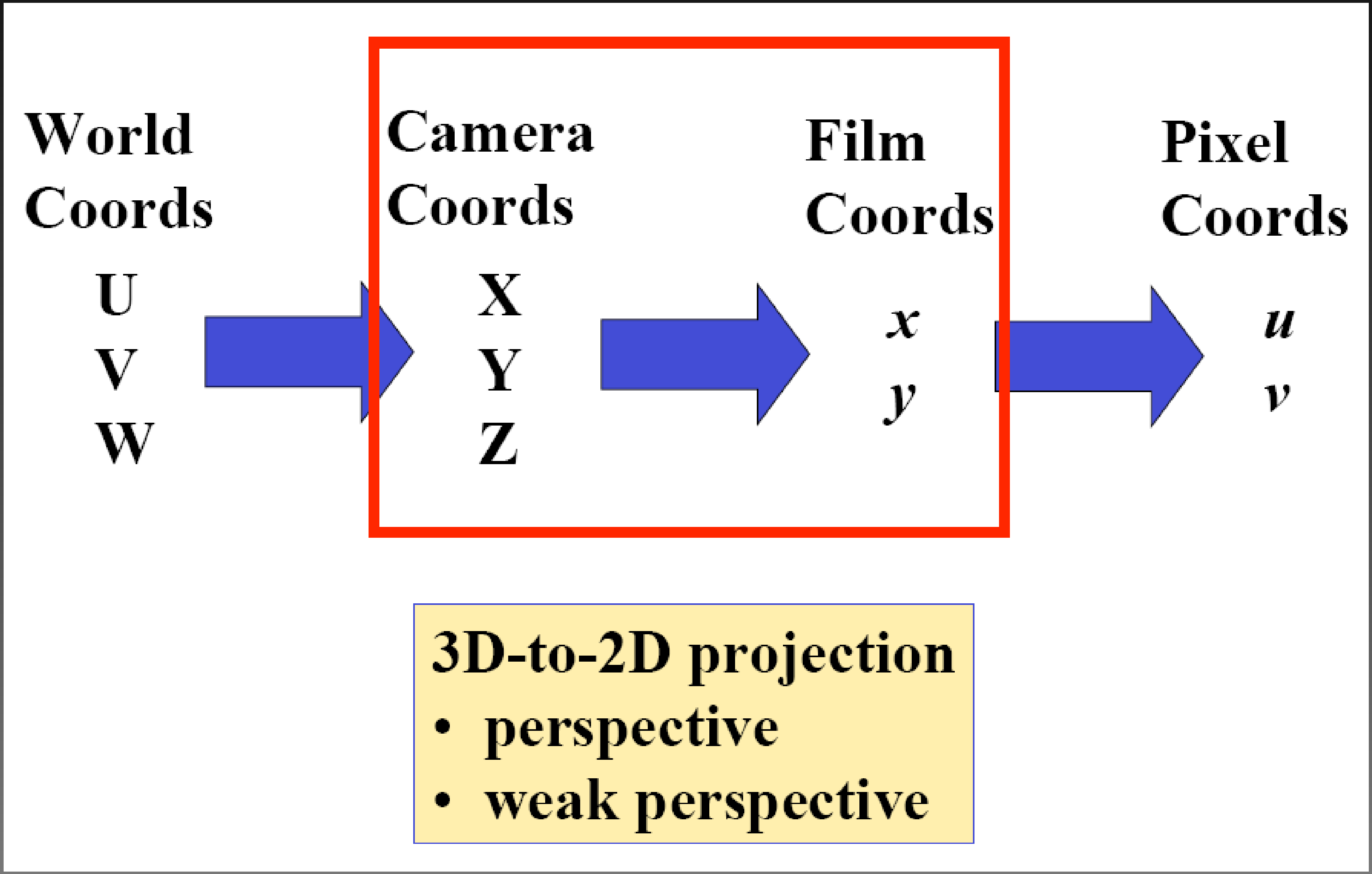
$$P_C = R\,(P_W - C)$$

World to Camera Transformation

$$\mathbf{P_C = R\,(\,P_W - C\,)}$$

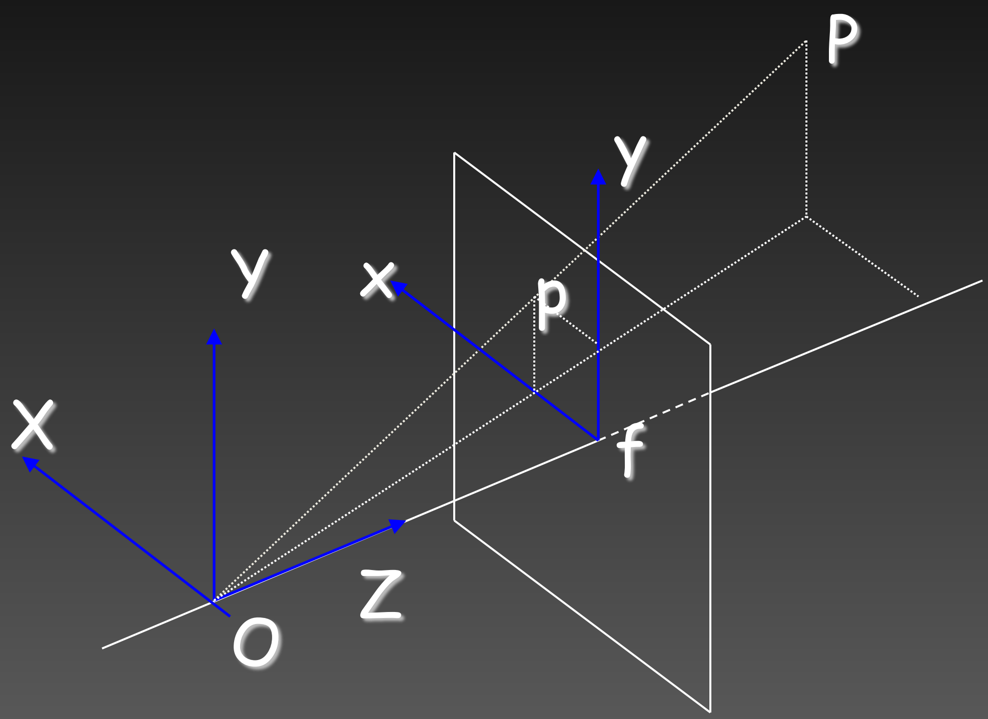
$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

$$P_C = M_{ext} \cdot P_W$$


World Coords
U
V
W
Camera Coords
X
Y
Z
Film Coords
x
y
Pixel Coords
u
v
3D-to-2D projection
• perspective
• weak perspective

Pinhole Camera Model

(Camera Coordinates)



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

$$y = f\,\frac{Y}{Z}$$

- Non-linear equations
- Any point on the ray OP has image p !!

Perspective Matrix Equation

(Camera Coordinates)

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

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

Using homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

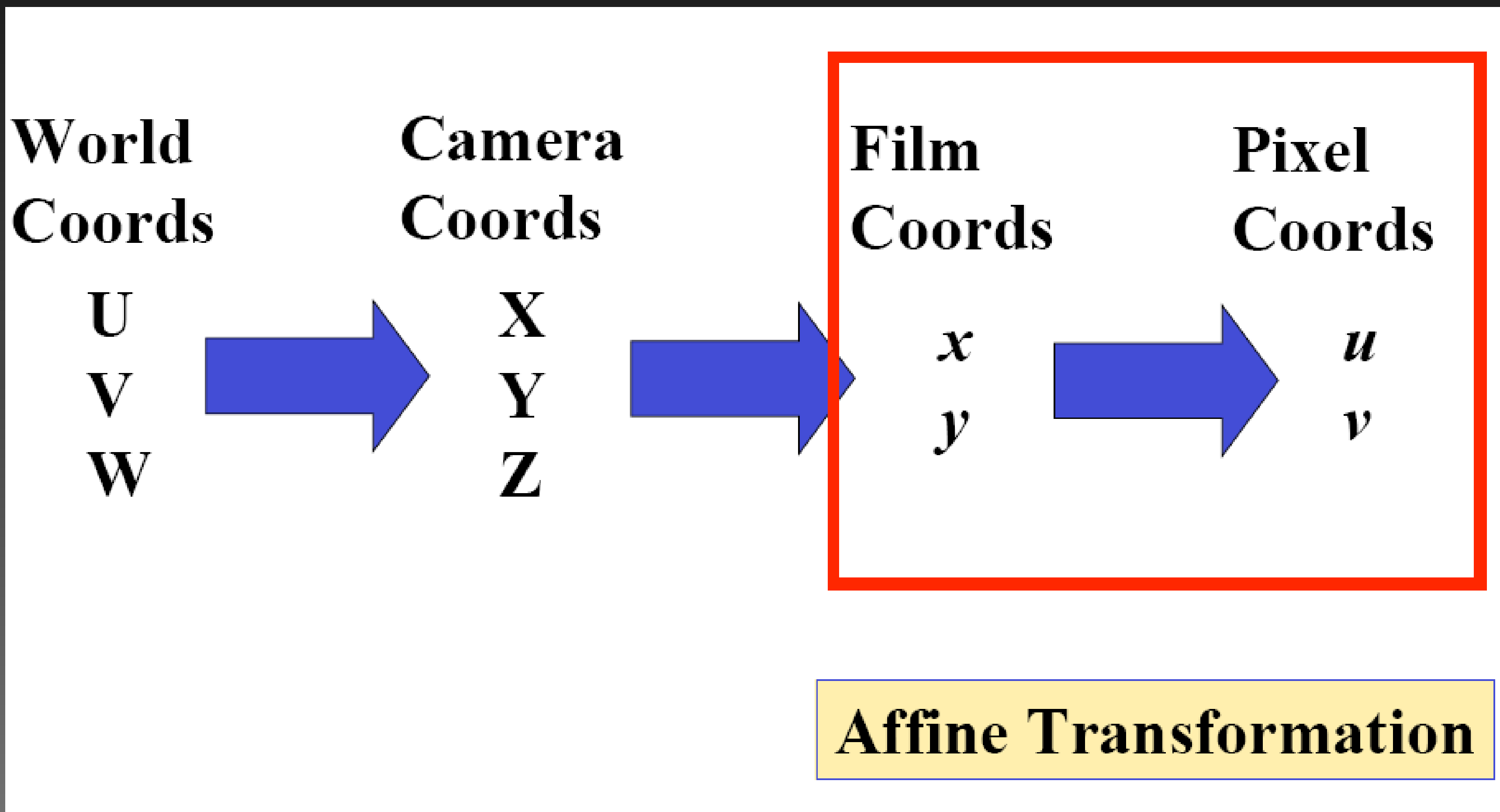
$$x = \frac{x'}{z'} \quad y = \frac{y'}{z'}$$

Perspective Matrix Equation

(Camera Coordinates)

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

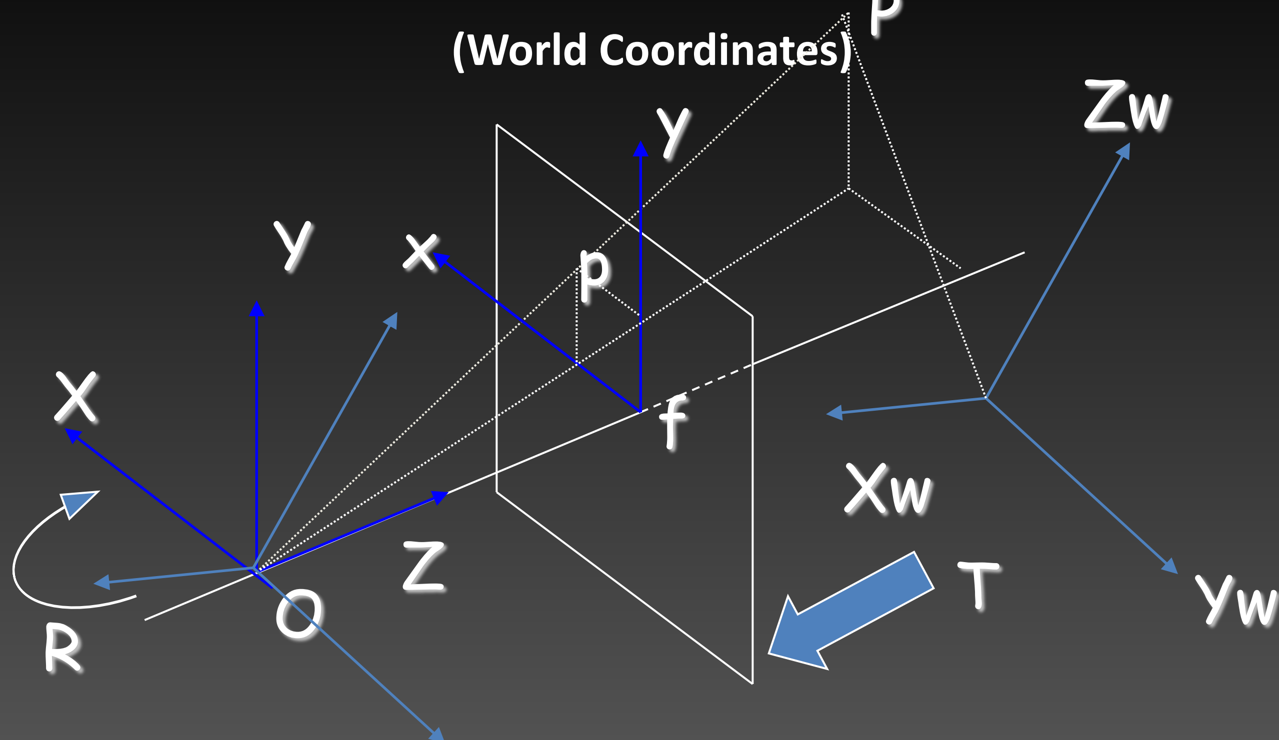
$$p = M_{\mathrm{int}} \cdot P_{\mathrm{C}}$$


World Coords
U
V
W
Camera Coords
X
Y
Z
Film Coords
x
y
Pixel Coords
u
v
Affine Transformation

Camera Parameters

- **Extrinsic parameters(外部参数)**
  - Location and orientation of the camera reference frame with respect to a known world reference frame
- **Intrinsic parameters（内部参数）**
  - Link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame

Pinhole Camera Model



$$P = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

$$p = M_{\text{int}} P = M_{\text{int}} M_{\text{ext}} \cdot P_w$$

Putting it all together:

- Extrinsic parameters (R, T):

$$P = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

- Intrinsic parameter (f):

$$p = M_{\text{int}} P = M_{\text{int}} M_{\text{ext}} \cdot P_w$$

$$p = M \cdot P_w$$

Extrinsic parameters

- Any set of geometric parameters that identify uniquely the transformation between the unknown camera reference frame and a known reference frame (world reference frame)
  - A 3-D translation vector, T
  - A 3x3 rotation matrix, R
  - R and T both require 3 parameters. These correspond to the 6 extrinsic parameters needed for camera calibration

$$P = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

Extrinsic parameters

$$P = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

$X^c = r_{11} X^w + r_{12} Y^w + r_{13} Z^w + T_x$

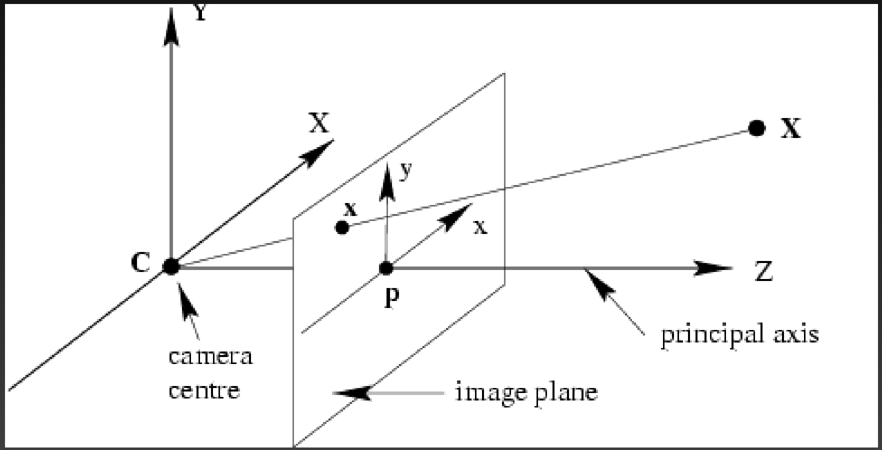
$Y^c = r_{21} X^w + r_{22} Y^w + r_{23} Z^w + T_y$

$Z^c = r_{31} X^w + r_{32} Y^w + r_{33} Z^w + T_z$

More intrinsic parameters:

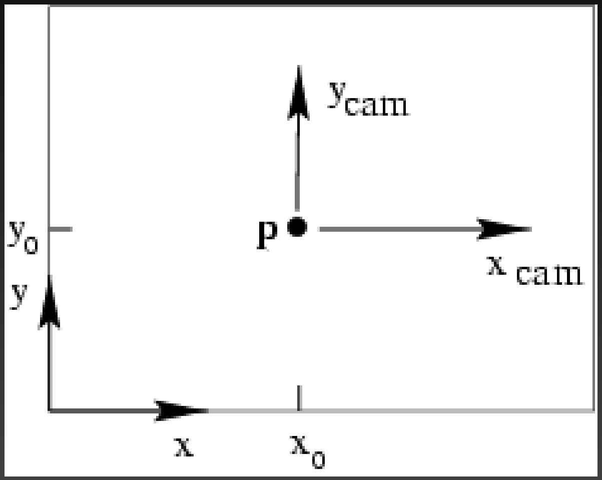
- The CCD sensor is made of a rectangular grid nxm of photosensors（光传感器）.
- Each photosensor generates an analog signal that is digitized by a frame grabber into an array of NxM pixels.

Camera coordinate system



- Principal axis: line from the camera center perpendicular to the image plane
- Normalized (camera) coordinate system: camera center is at the origin and the principal axis is the z-axis
- Principal point (p): point where principal axis intersects the image plane (origin of normalized coordinate system)

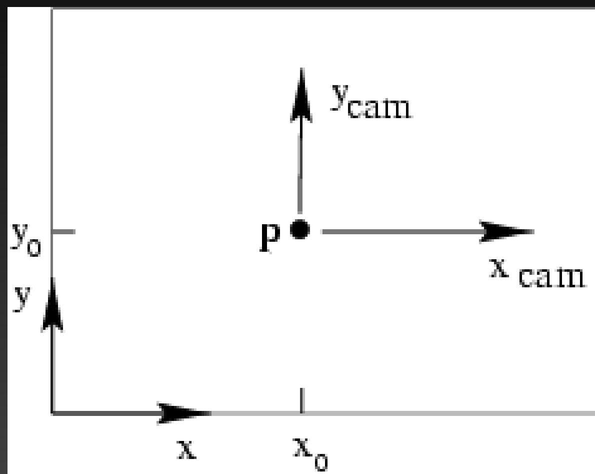
Principal point offset



principal point: (o_x, o_y)

- **Camera coordinate system: origin is at the prinicipal point**
- **Image coordinate system: origin is in the corner**

Principal point offset

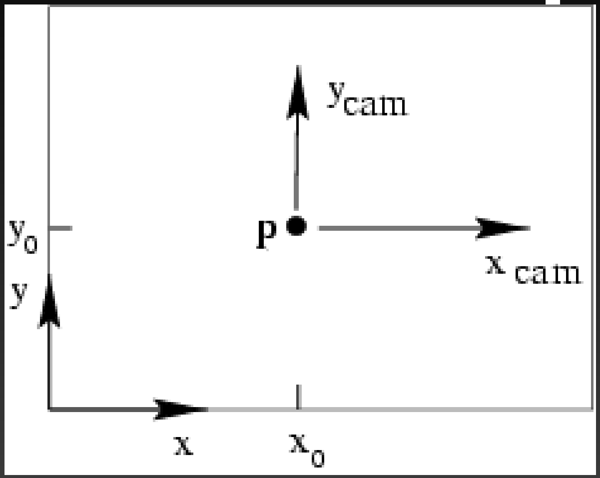


principal point: (o_x, o_y)

$$(X, Y, Z) \mapsto (f\,X/Z + o_x,\ f\,Y/Z + o_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\,X + Zo_x \\ f\,Y + Zo_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & o_x & 0 \\ & f & o_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal point offset



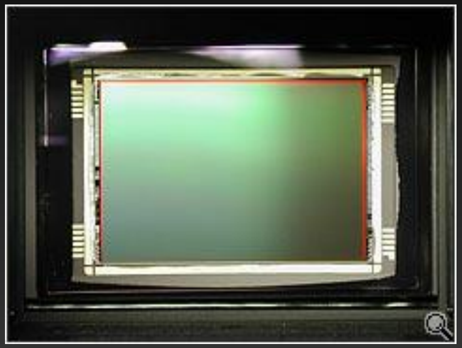
principal point: (o_x, o_y)

$$\begin{pmatrix} fX + Zo_x \\ fY + Zo_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & o_x \\ & f & o_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$K = \begin{bmatrix} f & & o_x \\ & f & o_y \\ & & 1 \end{bmatrix}$$ calibration matrix

$$\mathrm{P} = \mathrm{K}[\mathrm{I} \mid 0]$$

Pixel coordinates

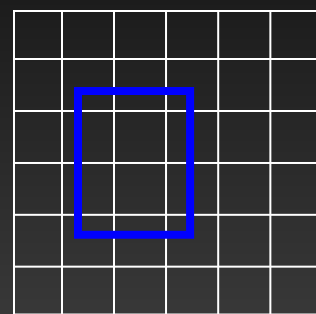


Pixel size: $\dfrac{1}{s_x} \times \dfrac{1}{s_y}$

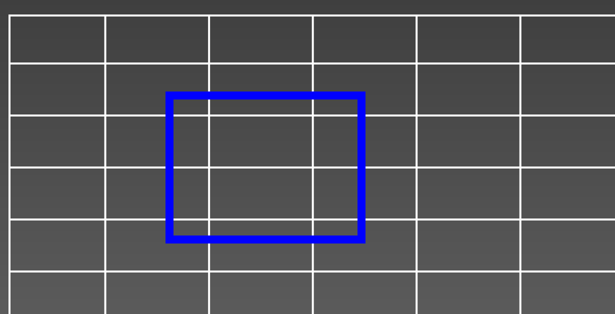
- ***s_x* pixels per meter in horizontal direction, s_y pixels per meter in vertical direction**

$$K = \underbrace{\begin{bmatrix} s_x & & \\ & s_y & \\ & & 1 \end{bmatrix}}_{\text{pixels/m}} \underbrace{\begin{bmatrix} f & & o_x \\ & f & o_y \\ & & 1 \end{bmatrix}}_{\text{m}} = \underbrace{\begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}}_{\text{pixels}}$$

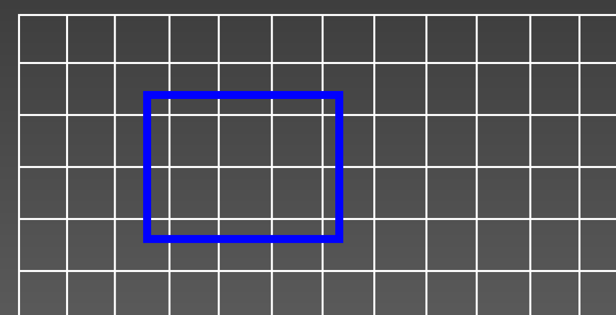
Intrinsic Parameters



NxN pixels Imaged Grid



nxn CCD elements
n:m aspect ratio



mxn CCD elements
n:n aspect ratio

Effective Sizes: s_x and s_y

In practice, we will assume that there is a
1-1 correspondence between CCD elements and pixels.

$$x = f\frac{X}{Z} = (x_{im} - o_x)s_x$$

$$y = f\frac{Y}{Z} = (y_{im} - o_y)s_y$$

Where o_x and o_y are the coordinates of the image center

A more complete Mint

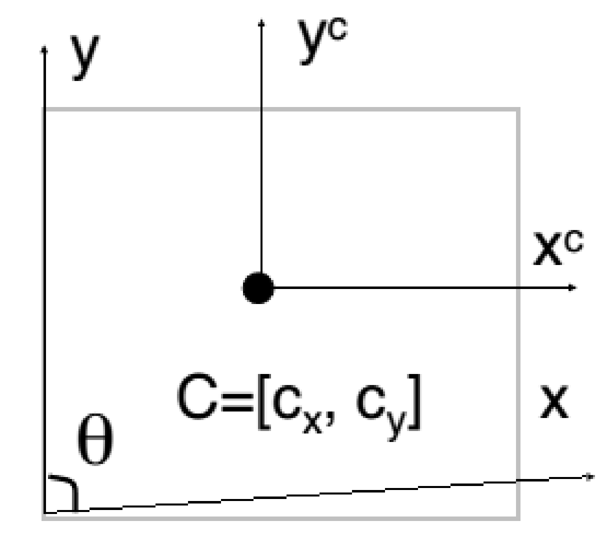
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f/s_x & 0 & -o_x & 0 \\ 0 & f/s_y & -o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$p = M_{\text{int}} \cdot P$$

Camera Skewness

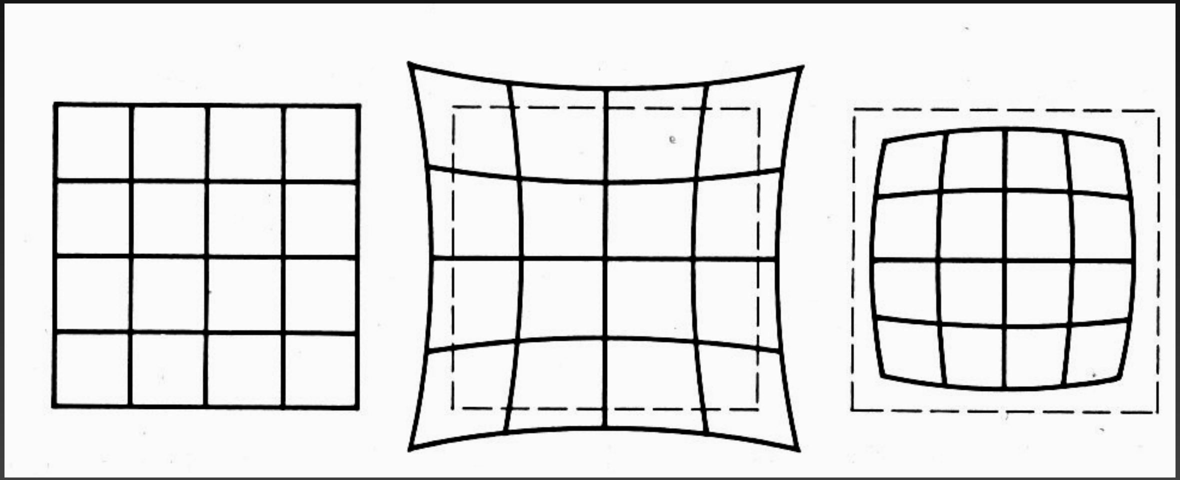
The two image axes are not perpendicular

The angle θ between the two axes is slightly larger or smaller than 90 degrees



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f/s_x & -f\cot\theta/s_x & -o_x & 0 \\ 0 & f/(s_y \sin\theta) & -o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Distortion



No distortion   Pin cushion   Barrel

- **Radial distortion of the image**
  - **Caused by imperfect lenses**
  - **Deviations are most noticeable for rays that pass through the edge of the lens**

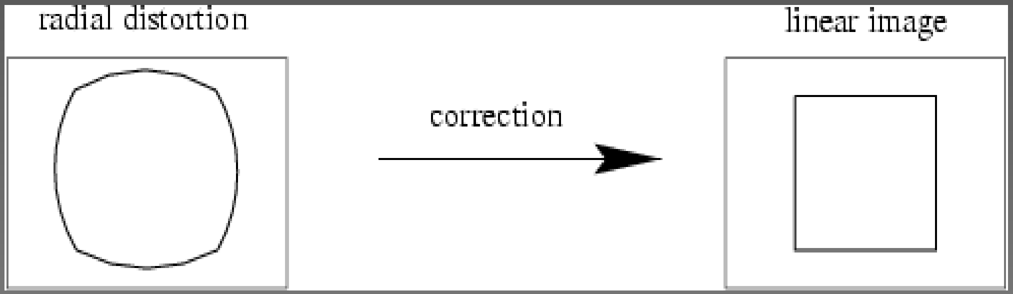
Radial Distortion(径向畸变)

- **Radial distortion can not be represented by matrix**

  $$x = x_d (1 + k_1 r^2 + k_2 r^4)$$

  $$y = y_d (1 + k_1 r^2 + k_2 r^4)$$

- **(x_d , y_d) the coordinates of the distorted point, and $r^2 = x_d^2 + y_d^2$, k_1 is first-order radial distortion coefficient**



Intrinsic parameters

- Defined as:
  - Focal length, f
  - The location of the image center in pixel coordinate, (o_x, o_y),
  - The effective size in the horizontal and vertical direction, (s_x, s_y),
  - If required, camera skewness θ， the radical distortion coefficient, k1

Putting It All Together: A Camera Model

Scale to pixel size

Camera orientation

Translate to image center

Perspective projection

Camera location

3D point (homogeneous coords)

$$\mathbf{T}_{img}\mathbf{S}_{pix}\mathbf{P}_{cam}\mathbf{R}_{cam}\mathbf{T}_{cam}\vec{x}$$

Then perform homogeneous divide, and get (u,v) coords

Putting It All Together: A Camera Model

Intrinsics

Extrinsics

$$\mathbf{T}_{img}\mathbf{S}_{pix}\mathbf{P}_{cam}\mathbf{R}_{cam}\mathbf{T}_{cam}\vec{x}$$

Putting It All Together: A Camera Model

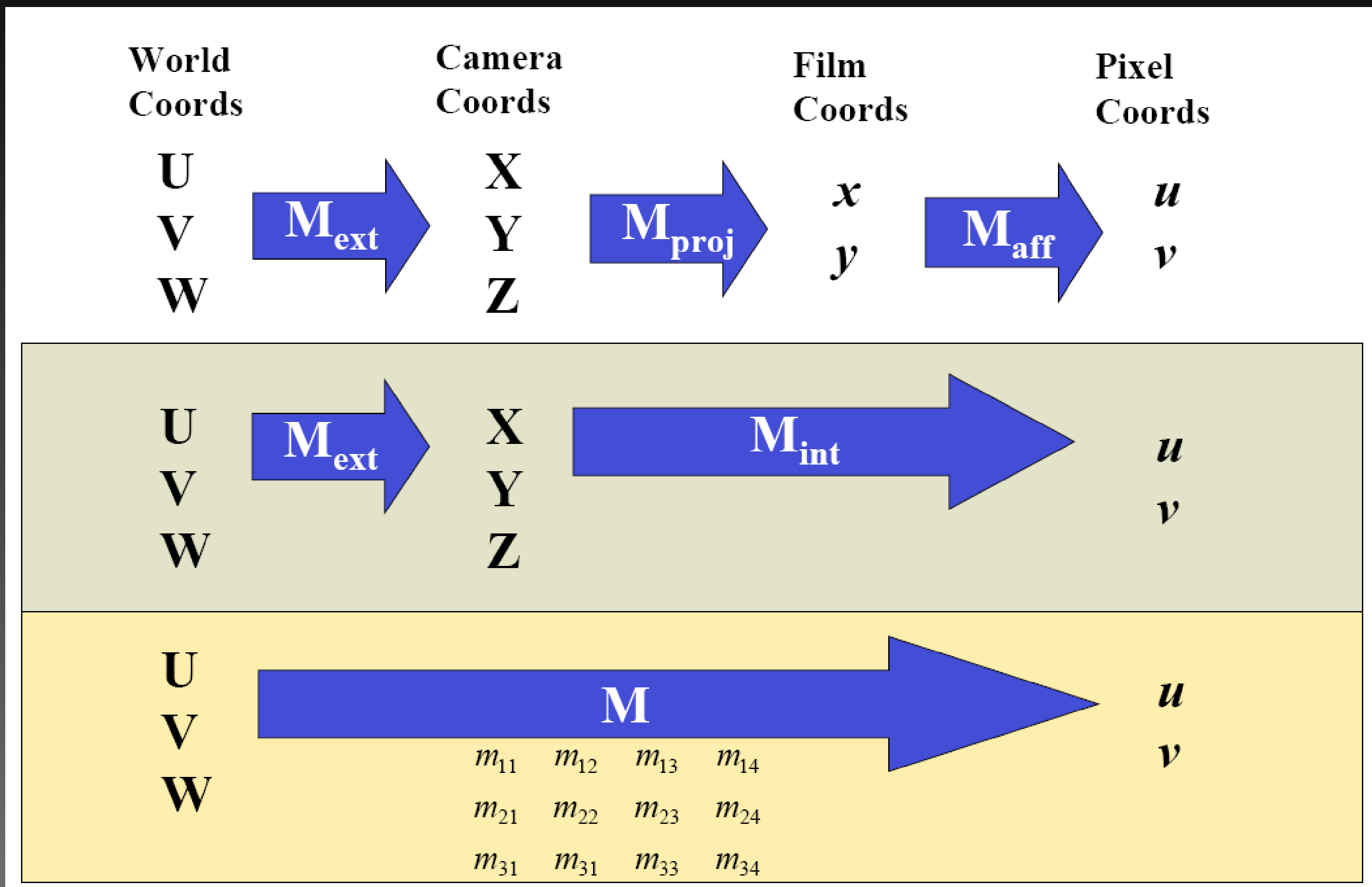
Camera coordinates

Normalized device coordinates

Image coordinates
Pixel coordinates

World coordinates

$$\mathbf{T}_{img}\mathbf{S}_{pix}\mathbf{P}_{cam}\mathbf{R}_{cam}\mathbf{T}_{cam}\vec{x}$$


World Coords
Camera Coords
Film Coords
Pixel Coords
U V W
M_{ext}
X Y Z
M_{proj}
x y
M_{aff}
u v
U V W
M_{ext}
X Y Z
M_{int}
u v
U V W
M
u v
m_{11} m_{12} m_{13} m_{14}
m_{21} m_{22} m_{23} m_{24}
m_{31} m_{31} m_{33} m_{34}

Summary

- Pinhole camera model
- Perspective projection, weak perspective model
- Homogeneous coordinates
- Camera parameters: extrinsic and intrinsic parameters.

Readings

- Forsyth & Ponce: Ch 1.1 and 1.2
- Szeliski: Chapter 2 (2.1: 2.1.5)