University of Zurich[UZH] | ETH zürich

Institute of Informatics – Institute of Neuroinformatics

ROBOTICS & PERCEPTION GROUP

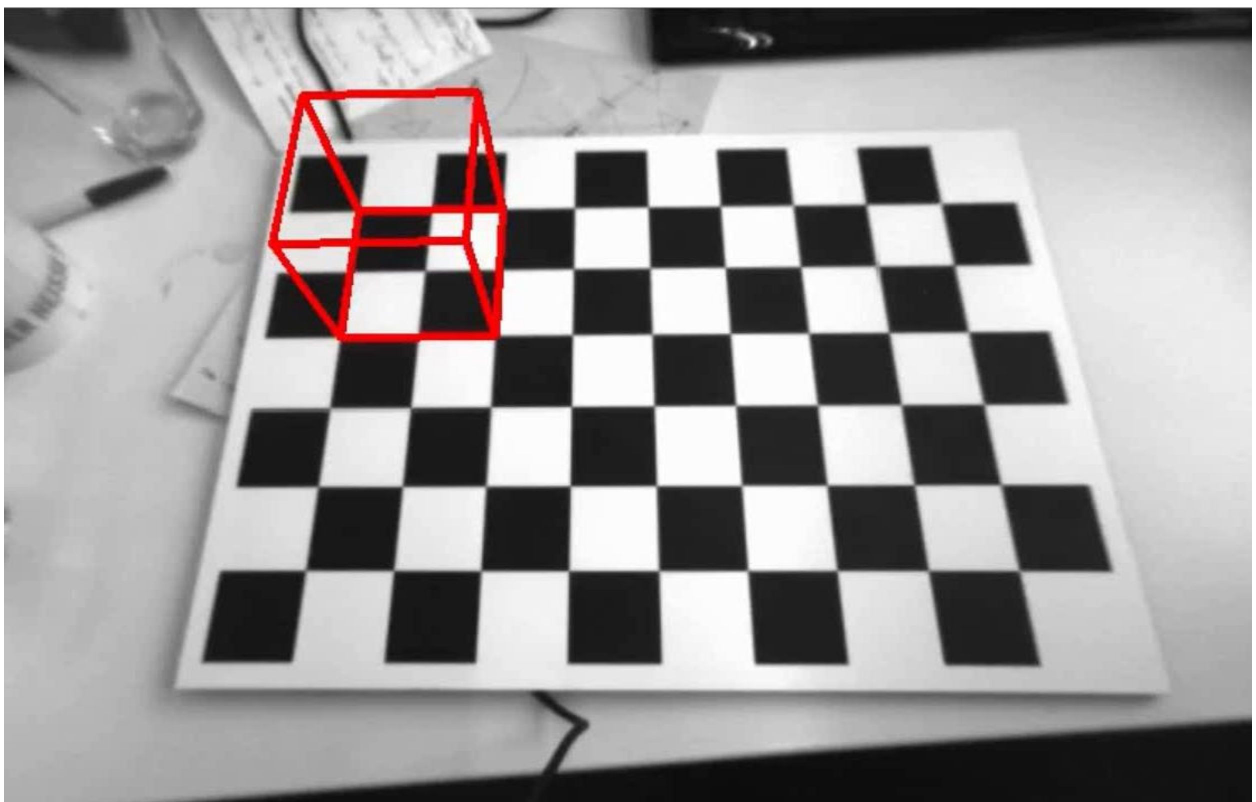
Lecture 02
Image Formation 1

Davide Scaramuzza

http://rpg.ifi.uzh.ch

Lab Exercise 1 - Today afternoon

- Room ETH HG E 1.1 from 13:15 to 15:00
- Work description: implement an augmented reality wireframe cube
  - Practice the perspective projection

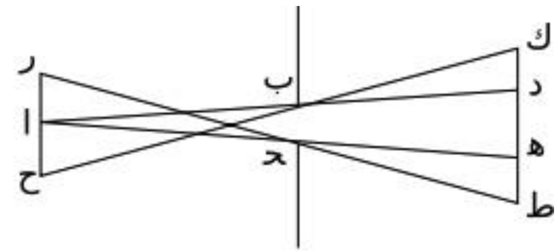


Outline of this lecture

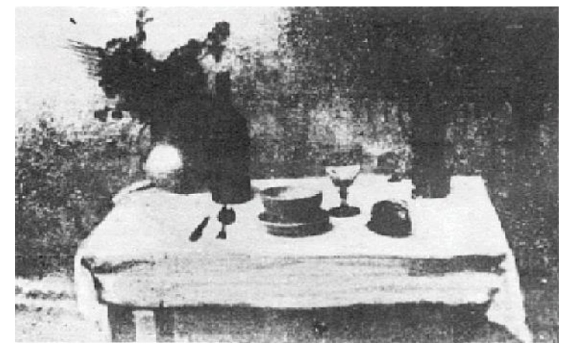
- Image Formation
- Other camera parameters
- Digital camera
- Perspective camera model
- Lens distortion

Historical context

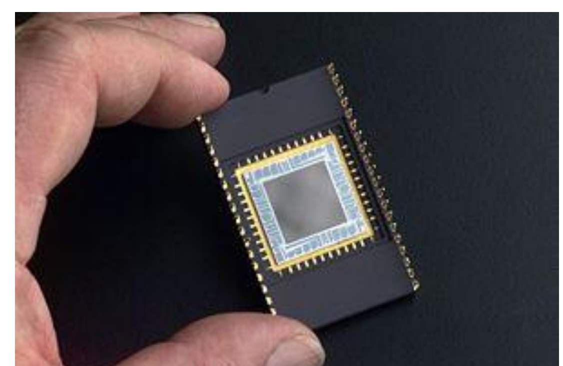
- **Pinhole model:** Mozi (470-390 BCE), Aristotle (384-322 BCE)
- **Principles of optics (including lenses):** Alhacen (965-1039)
- **Camera obscura**: Leonardo da Vinci (1452-1519), Johann Zahn (1631-1707)
- **First photo**: Joseph Nicephore Niepce (1822)
- **Daguerréotypes** (1839)
- **Photographic film** (Eastman, 1888, founder of Kodak)
- **Cinema** (Lumière Brothers, 1895)
- **Color Photography** (Lumière Brothers, 1908)
- **Television** (Baird, Farnsworth, Zworykin, 1920s)
- **First consumer camera with CCD**: Sony Mavica (1981)
- **First fully digital camera**: Kodak DCS100 (1990)



Alhacen's notes



Niepce, "La Table Servie," 1822

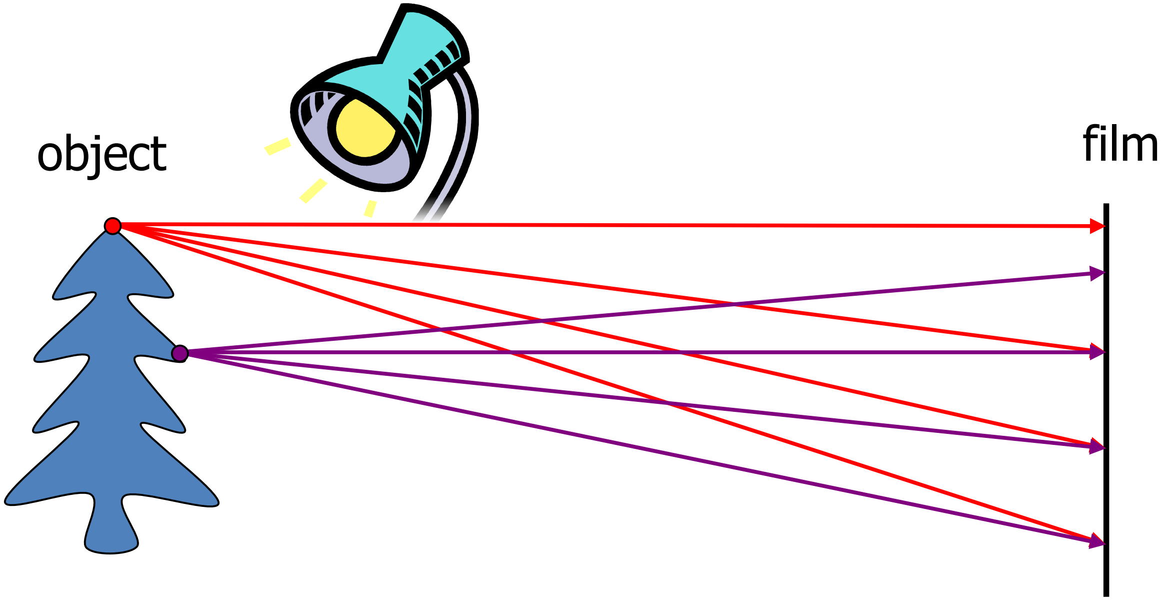


CCD chip

Image formation

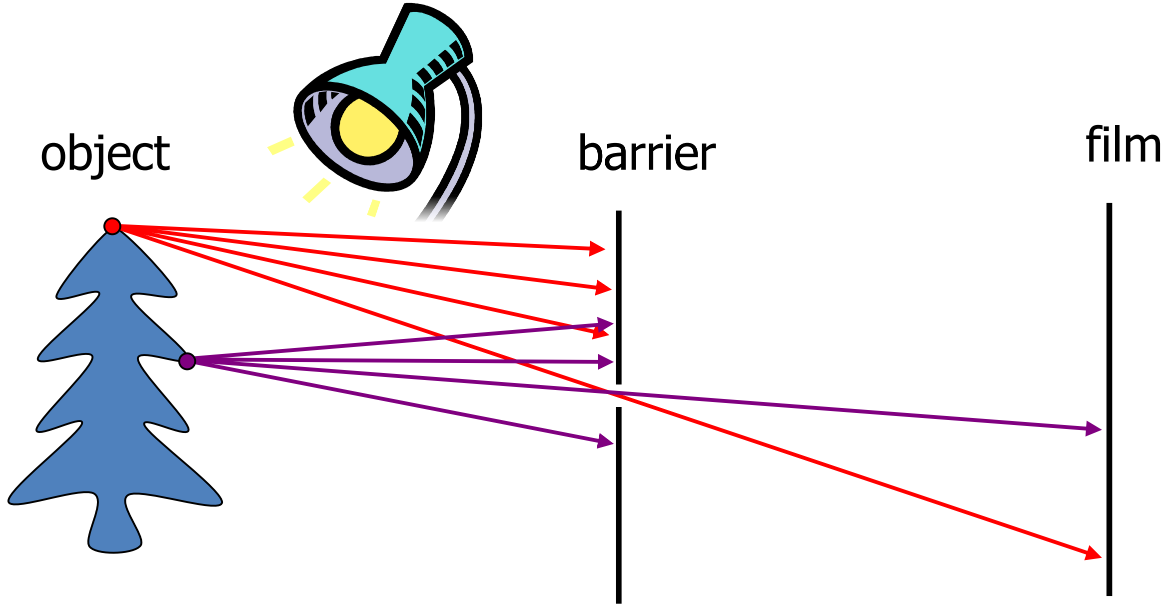
- How are objects in the world captured in an image?

How to form an image



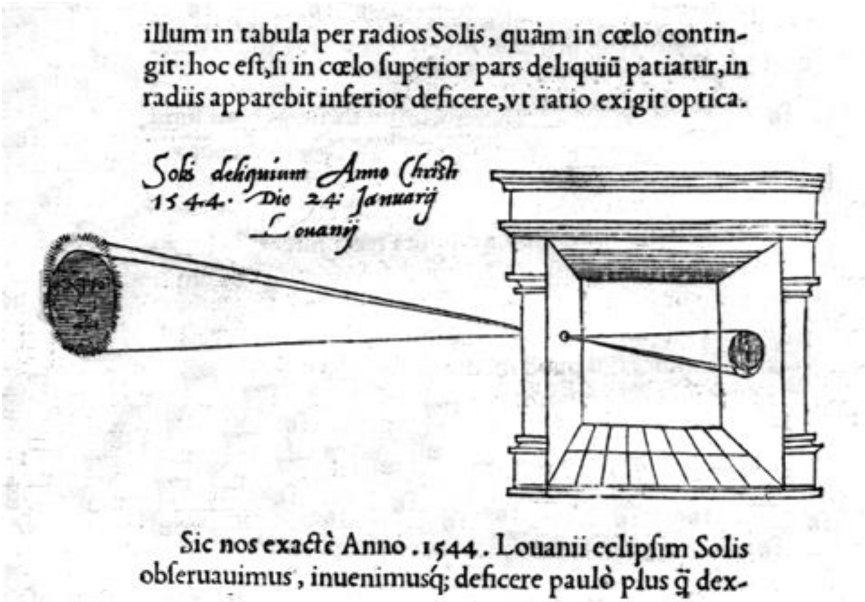
- Place a piece of film in front of an object
  ⇨ Do we get a reasonable image?

Pinhole camera



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening is known as the **aperture**

Camera obscura



In Latin, means ‘dark room’

- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)
- Image is inverted
- Depth of the room (box) is the effective focal length

"**Reinerus Gemma-Frisius**, observed an eclipse of the sun at Louvain on January 24, 1544, and later he used this illustration of the event in his book De Radio Astronomica et Geometrica, 1545. It is thought to be the first published illustration of a camera obscura..." Hammond, John H., The Camera Obscura, A Chronicle

Camera obscura at home

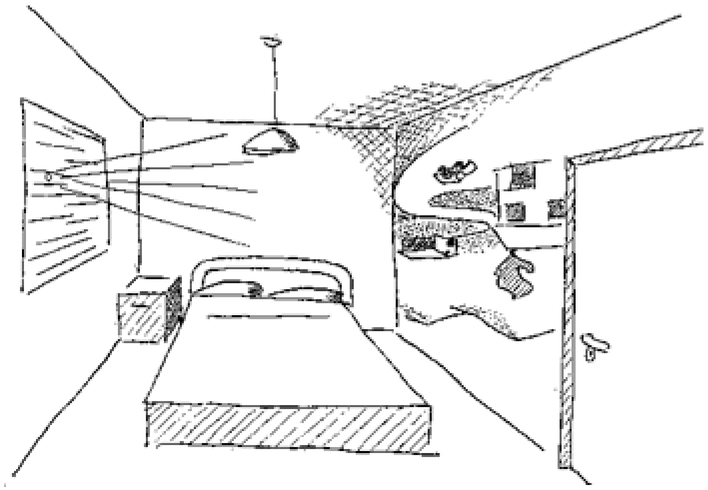


Figure 1 - A lens on the window creates the image of the external world on the opposite wall and you can see it every morning, when you wake up.

http://www.youtube.com/watch?v=B2aOs8RWntg

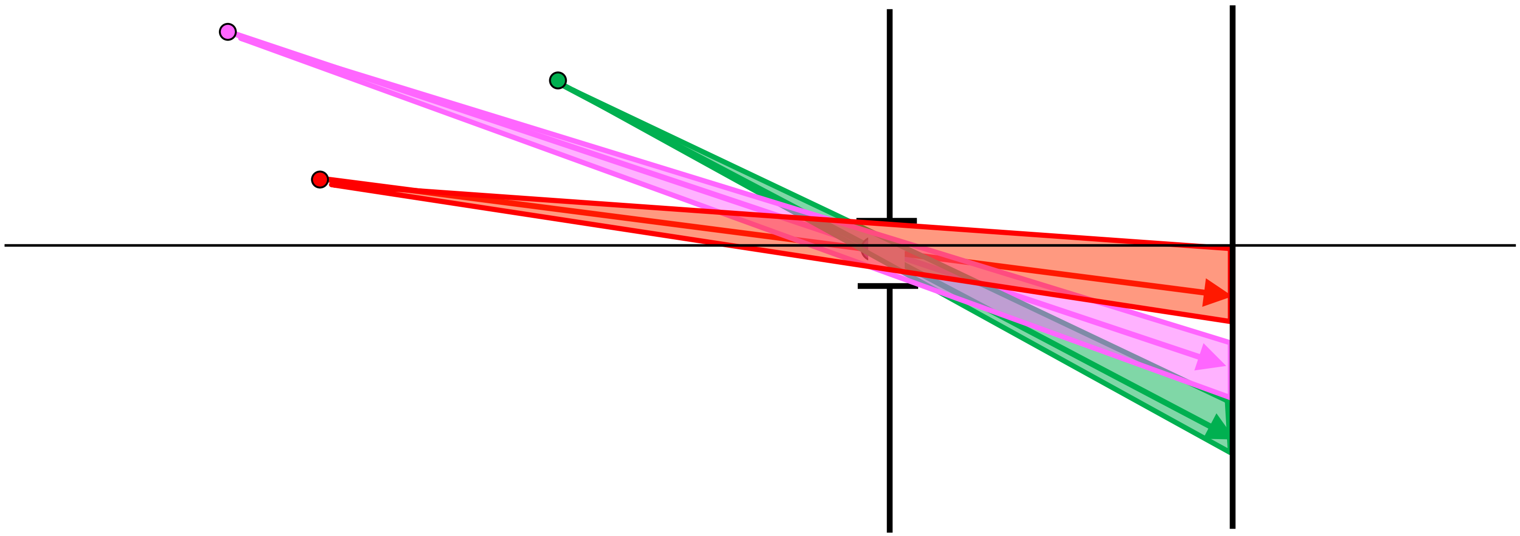
Sketch from http://www.funsci.com/fun3_en/sky/sky.htm

Home-made pinhole camera

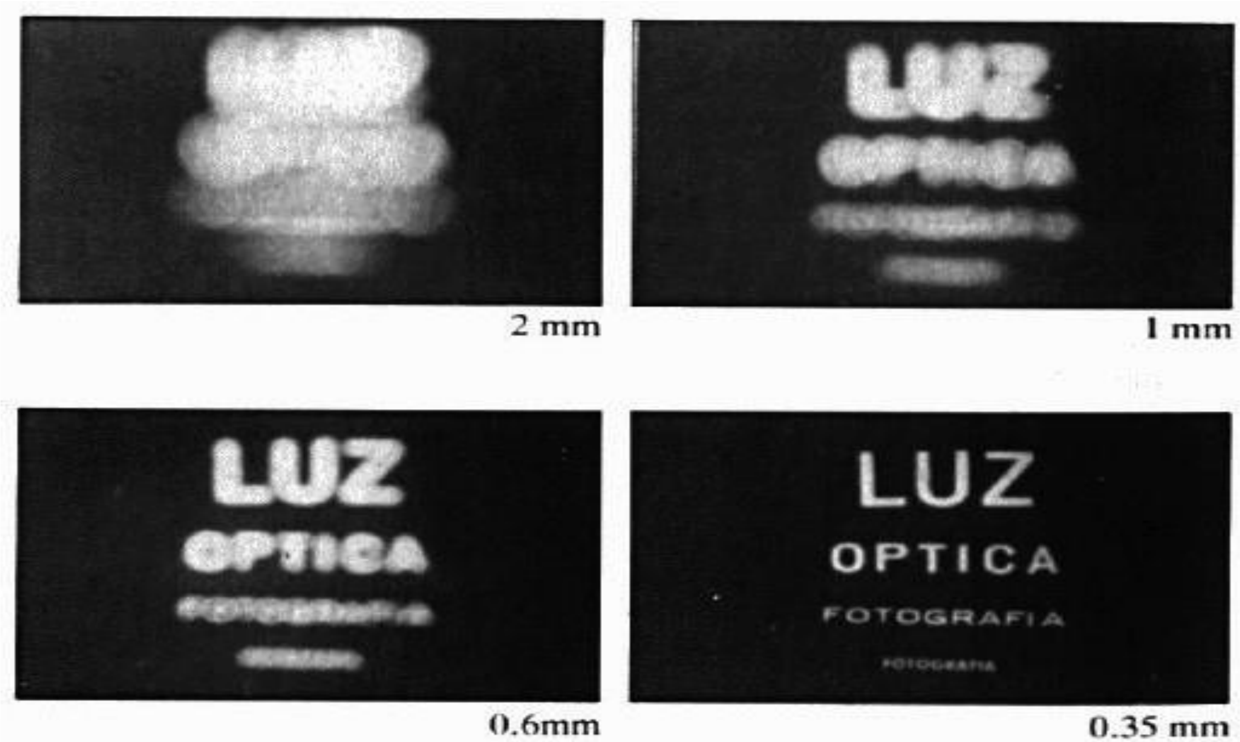
What can we do
to reduce the blur?

Effects of the Aperture Size

- *In an ideal pinhole*, only one ray of light reaches each point on the film ⇨ the image can be very dim
- Making the aperture bigger makes the image blurry

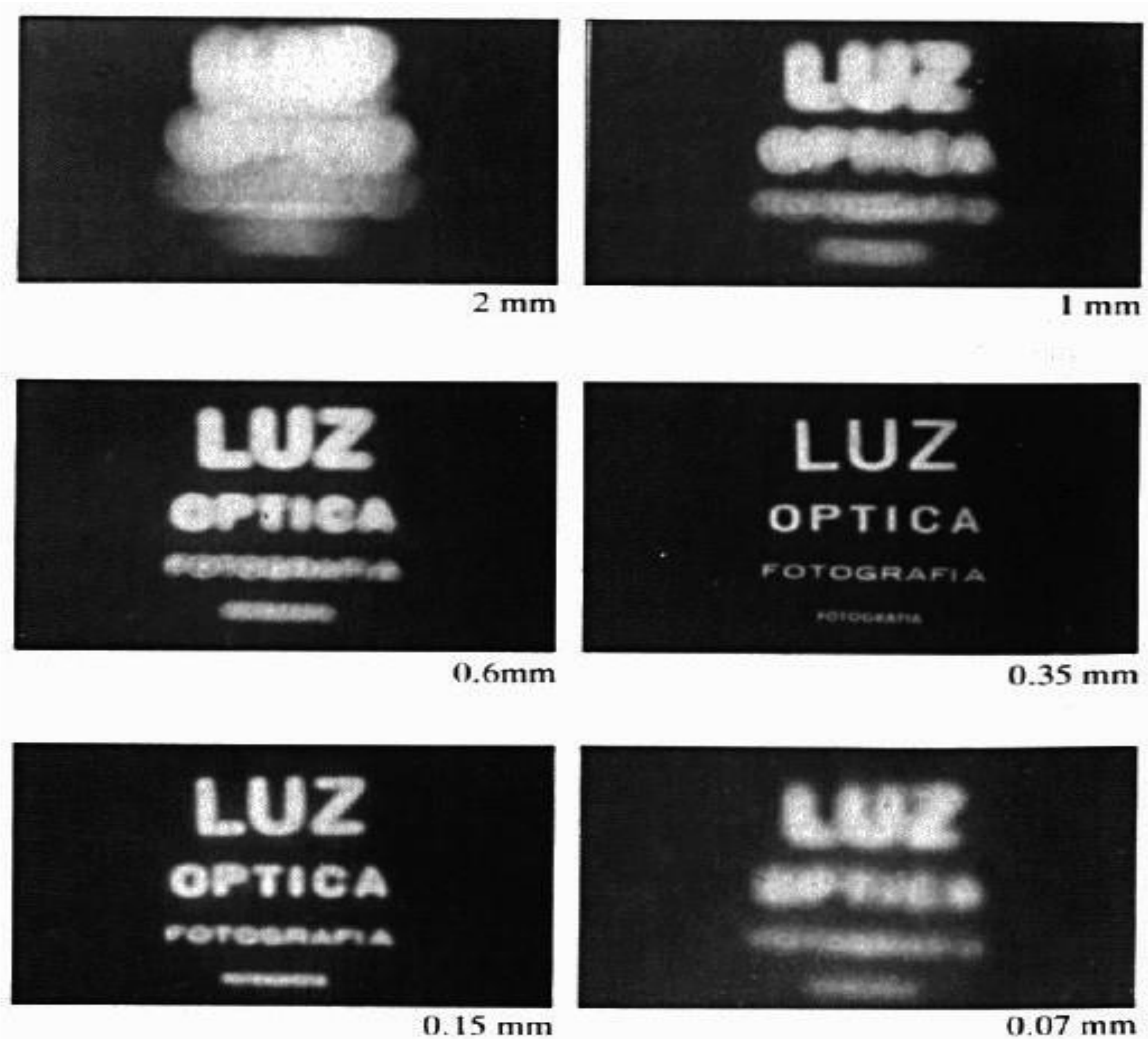


Shrinking the aperture



Why not make the aperture as small as possible?

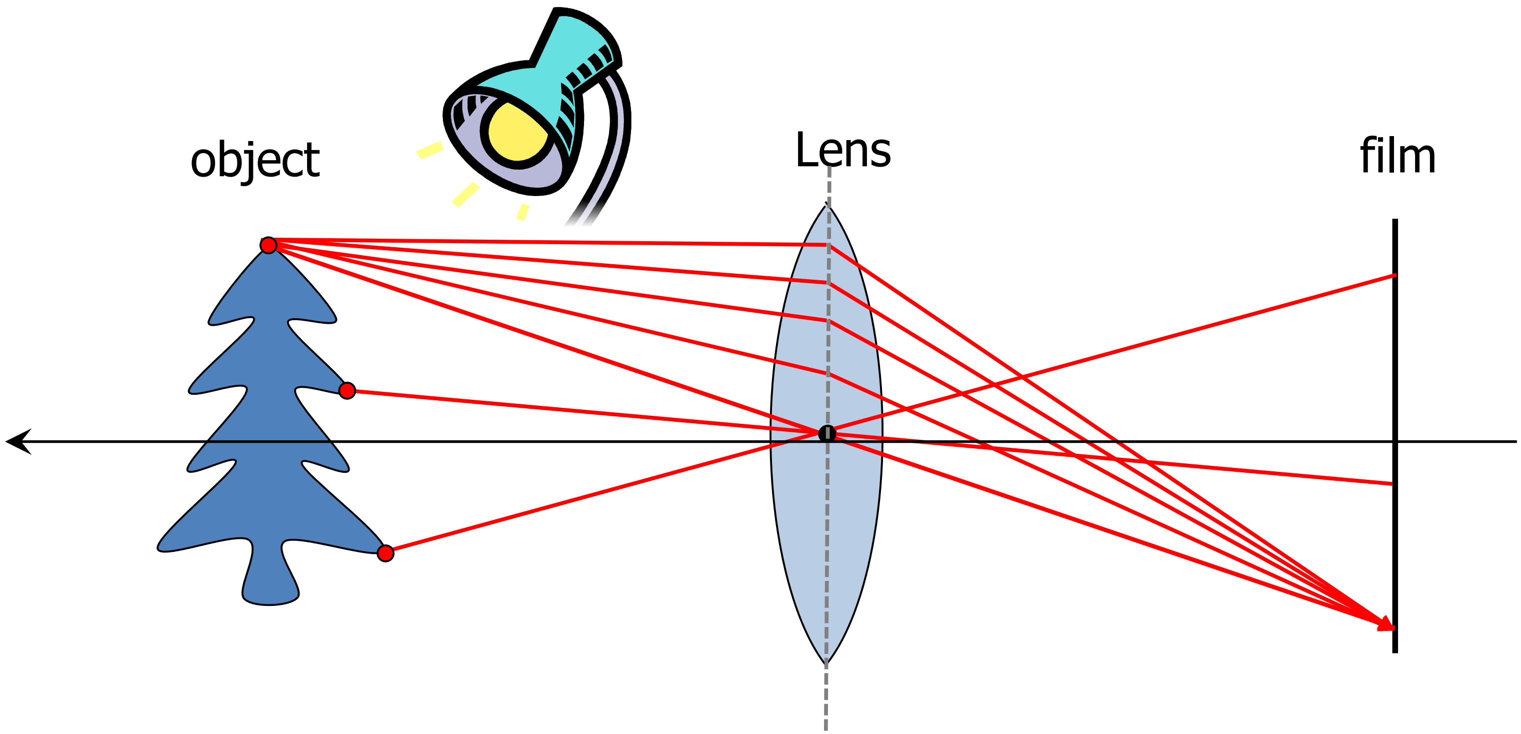
Shrinking the aperture



Why not make the aperture as small as possible?

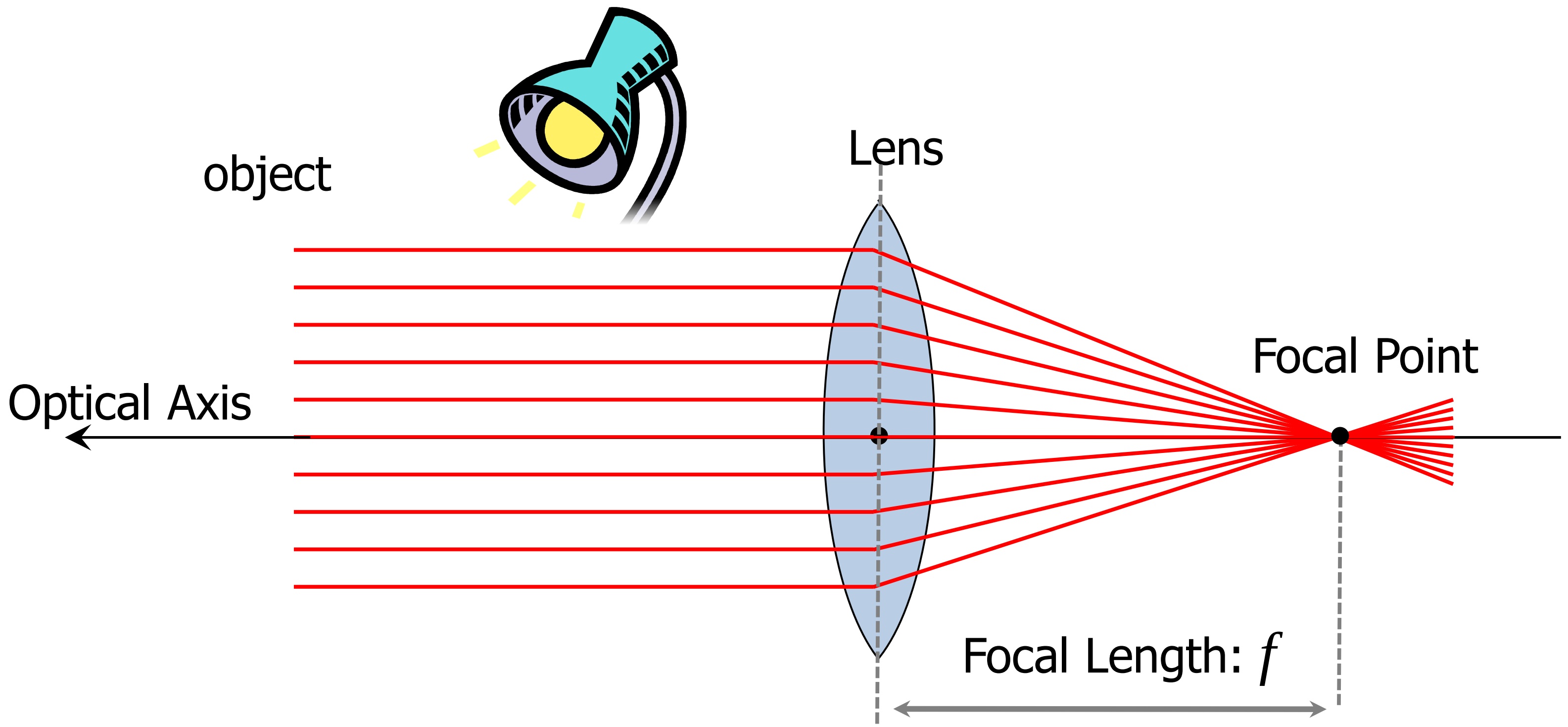
- Less light gets through (must increase the exposure)
- Diffraction effects…

Image formation using a converging lens



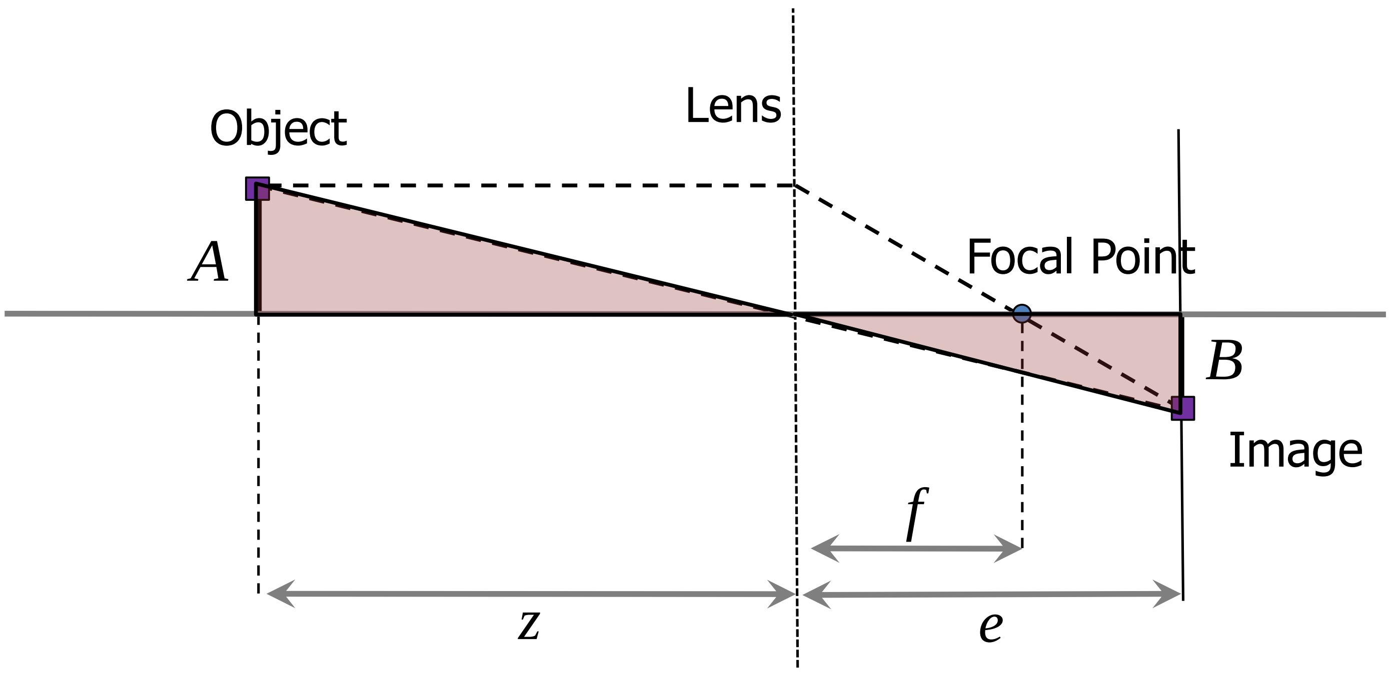
- A lens focuses light onto the film
- Rays passing through the the **Optical Center** are not deviated

Image formation using a converging lens



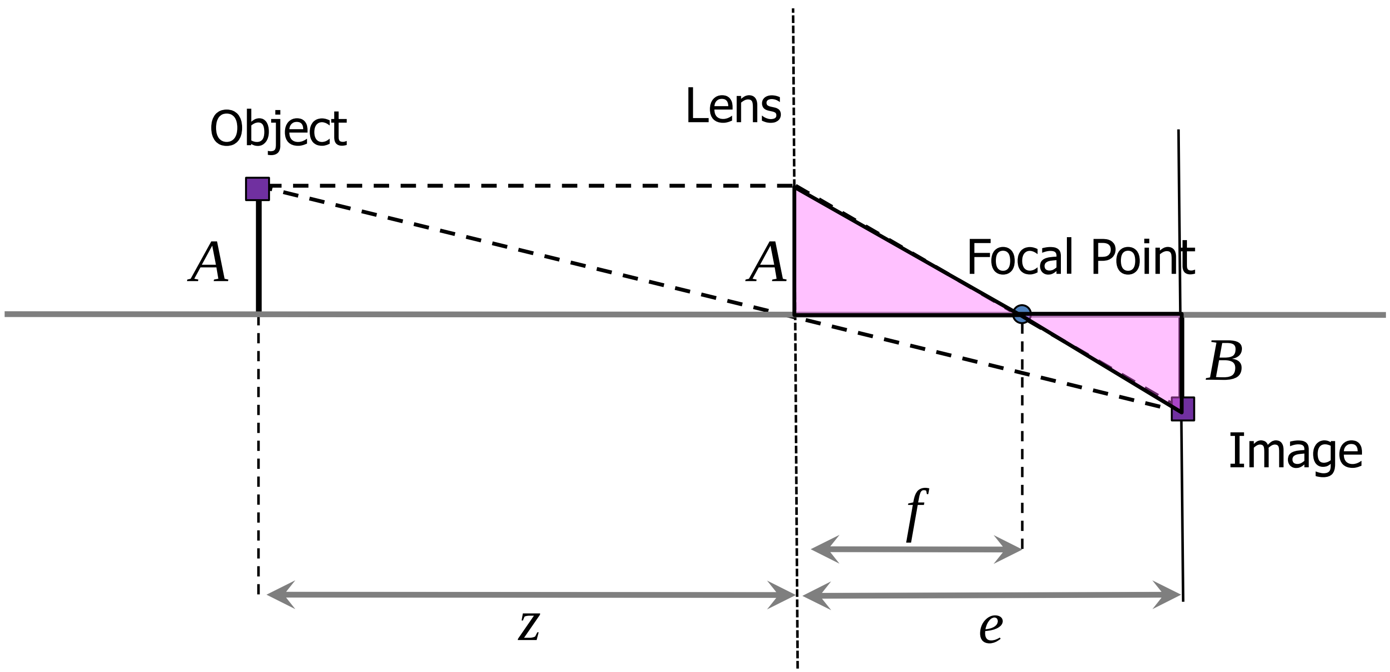
- All rays parallel to the **Optical Axis** converge at the **Focal Point**

Thin lens equation



Find a relationship between *f*, *z*, and *e*

Thin lens equation



- Similar Triangles:

$$\frac{B}{A} = \frac{e}{z}$$

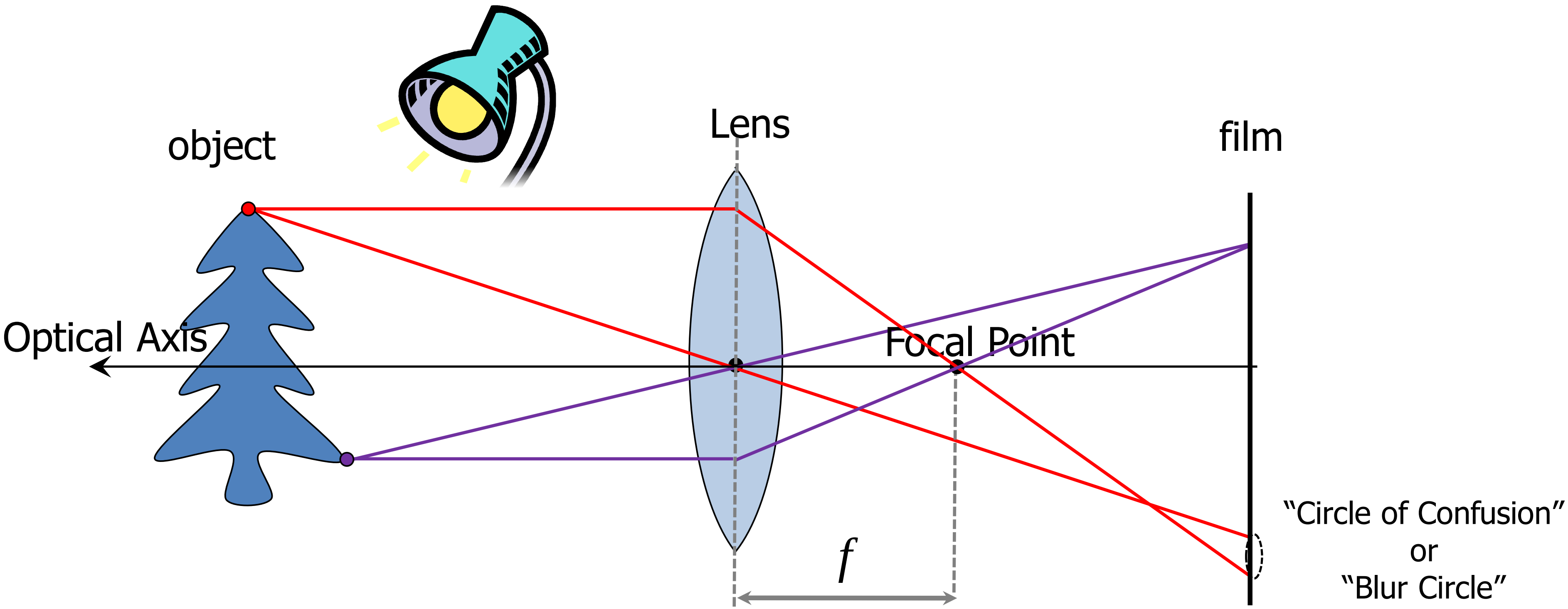
$$\frac{B}{A} = \frac{e-f}{f} = \frac{e}{f} - 1$$

$$\frac{e}{f} - 1 = \frac{e}{z} \Rightarrow \boxed{\frac{1}{f} = \frac{1}{z} + \frac{1}{e}}$$

"Thin lens equation"

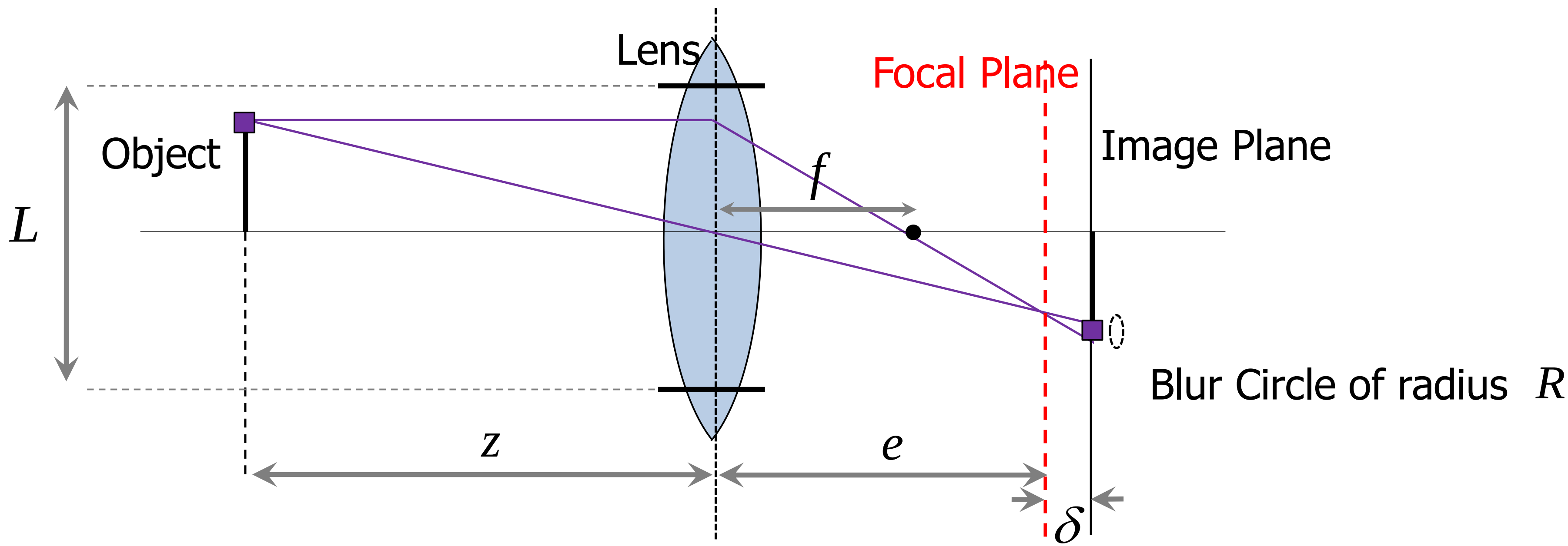
- Any object point satisfying this equation is in focus
- Can I use this to measure distances?

“In focus”



- For a fixed film distance from the lens, there is a specific distance between the object and the lens, at which the object appears “in focus” in the image
- Other points project to a “blur circle” in the image

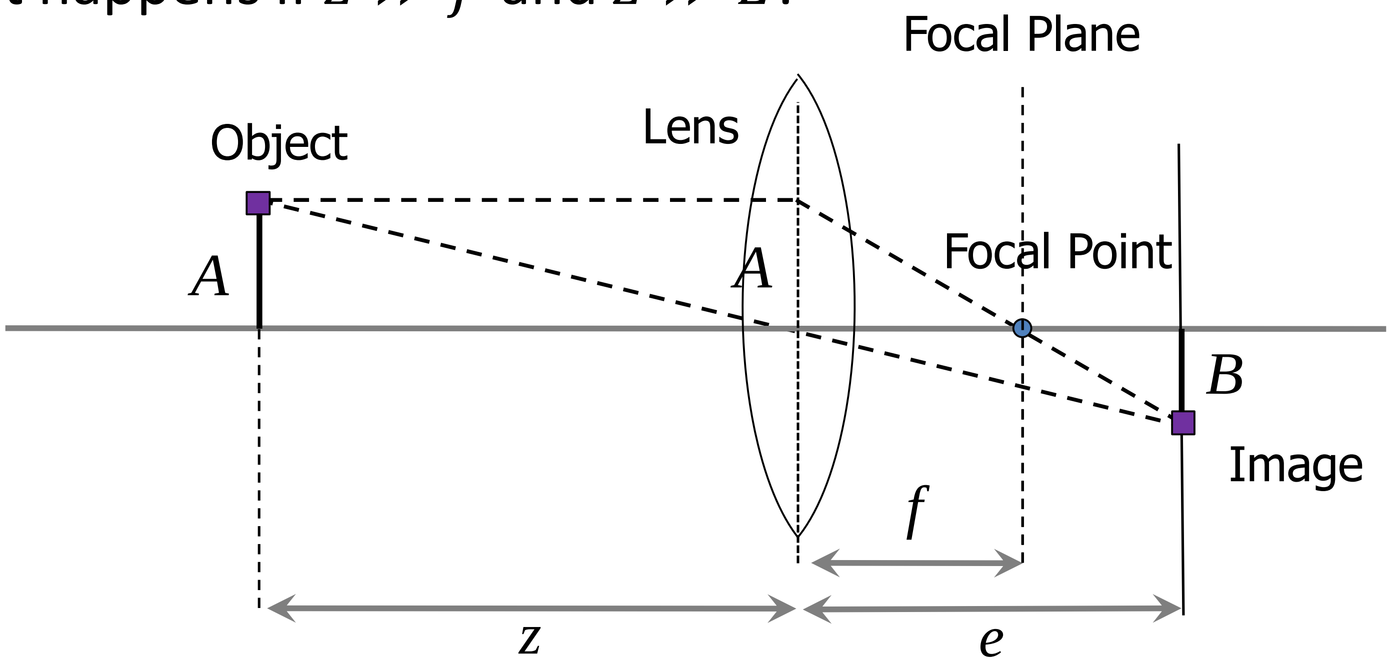
Blur Circle



- Object is out of focus ⇨ Blur Circle has radius: $R = \dfrac{L\delta}{2e}$
  - A small *L* (pinhole) gives a small *R* (Blur Circle)
  - To capture a 'good' image: adjust camera settings, such that R remains smaller than the image resolution

The Pin-hole approximation

- What happens if $z \gg f$ and $z \gg L$?

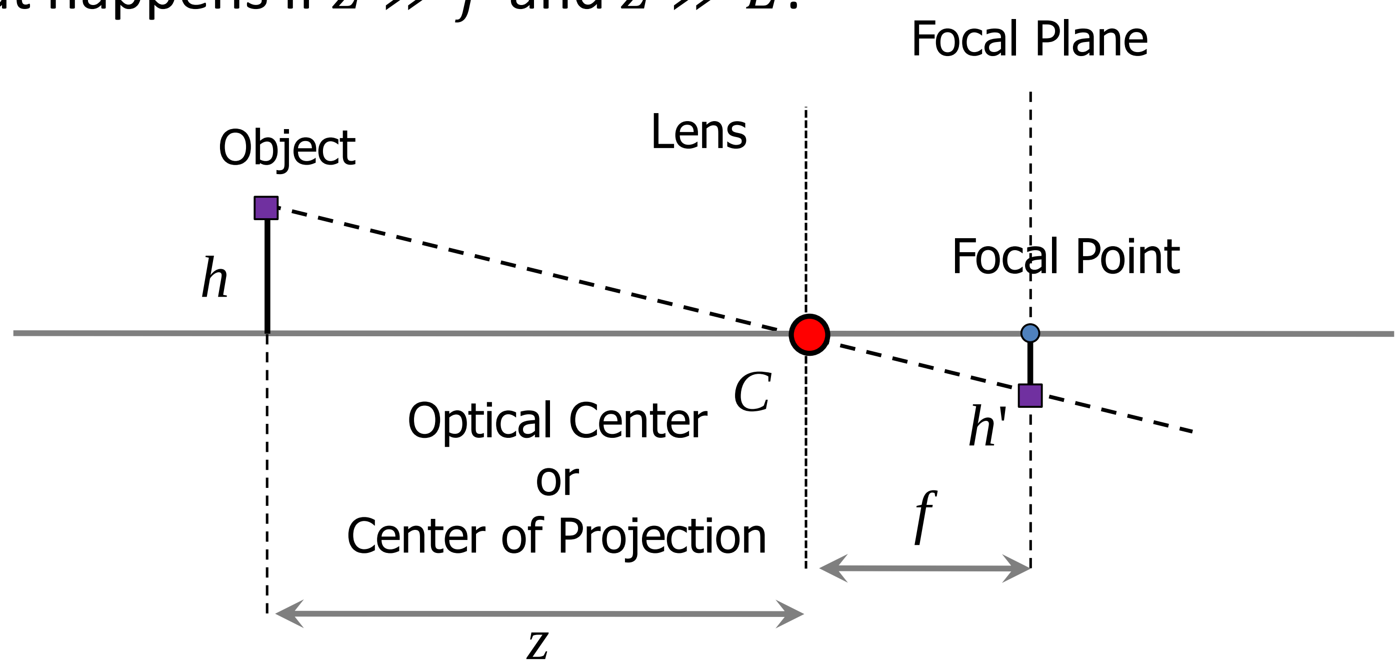


- We need to adjust the image plane such that objects at infinity are in focus. As the object gets far, the image plan gets closer to the focal plane

$$\frac{1}{f} = \underbrace{\boxed{\frac{1}{z}}}_{\cong 0} + \frac{1}{e} \Rightarrow \frac{1}{f} \approx \frac{1}{e} \Rightarrow f \approx e$$

The Pin-hole approximation

- What happens if $z \gg f$ and $z \gg L$?

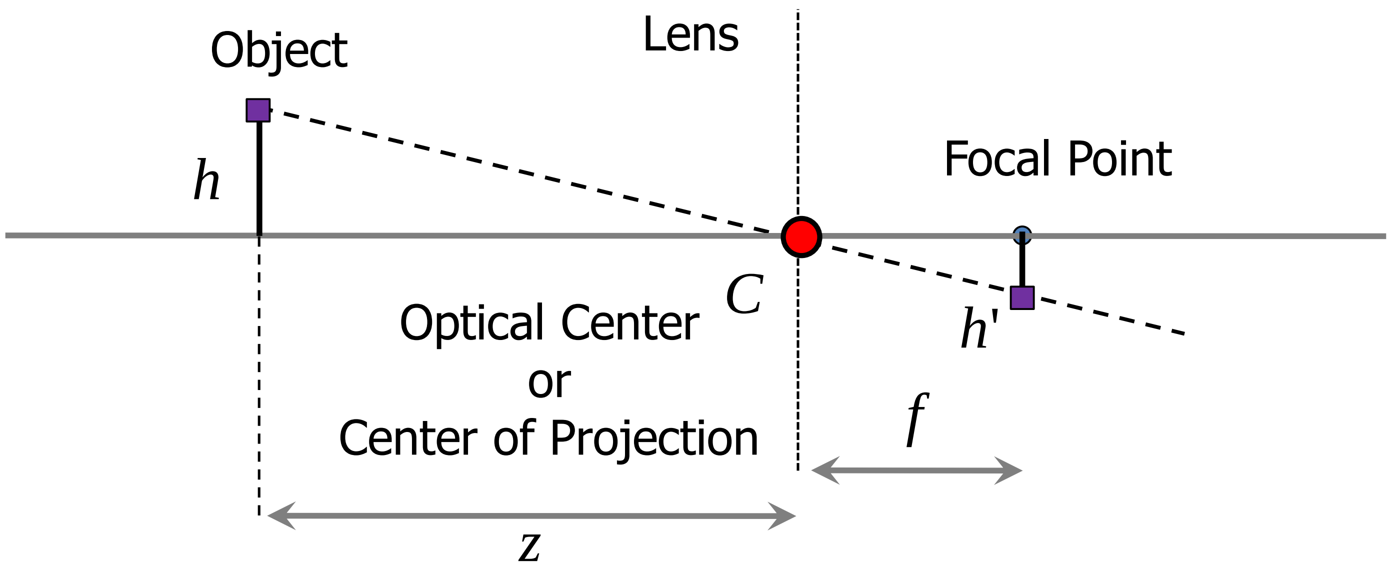


- We need to adjust the image plane such that objects at infinity are in focus. As the object gets far, the image plan gets closer to the focal plane

$$\frac{1}{f} = \underbrace{\boxed{\frac{1}{z}}}_{\cong 0} + \frac{1}{e} \quad \Rightarrow \frac{1}{f} \approx \frac{1}{e} \Rightarrow f \approx e$$

The Pin-hole approximation

- What happens if $z \gg f$ and $z \gg L$?



- This is known as Pinhole Approximation and the relation between the image and object becomes:

$$\frac{h'}{h} = \frac{f}{z} \Rightarrow h' = \frac{f}{z} h$$

- The dependence of the image of an object on its depth (i.e. distance from the camera) is known as **perspective**

Perspective effects

- Far away objects appear smaller

Perspective effects

Perspective and art

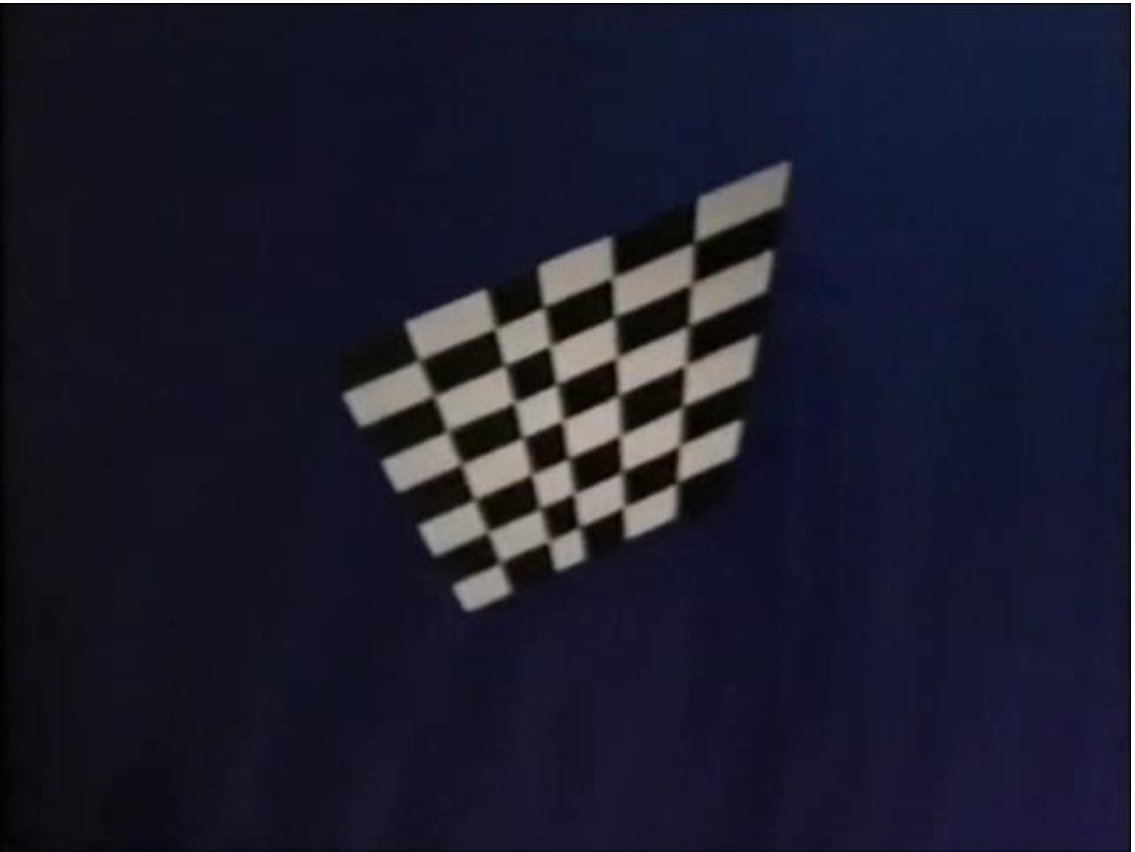
- Use of correct perspective projection indicated in 1st century BC frescoes
- During Renaissance time, artists developed systematic methods to determine perspective projection (around 1480-1515)

Raphael

Durer

Playing with Perspective

- Perspective gives us very strong depth cues
  ⇨ hence we can perceive a 3D scene by viewing its 2D representation (i.e. image)
- An example where perception of 3D scenes is misleading is the Ames room (check out the Ames room in the Technorama science museum in Winterthur)



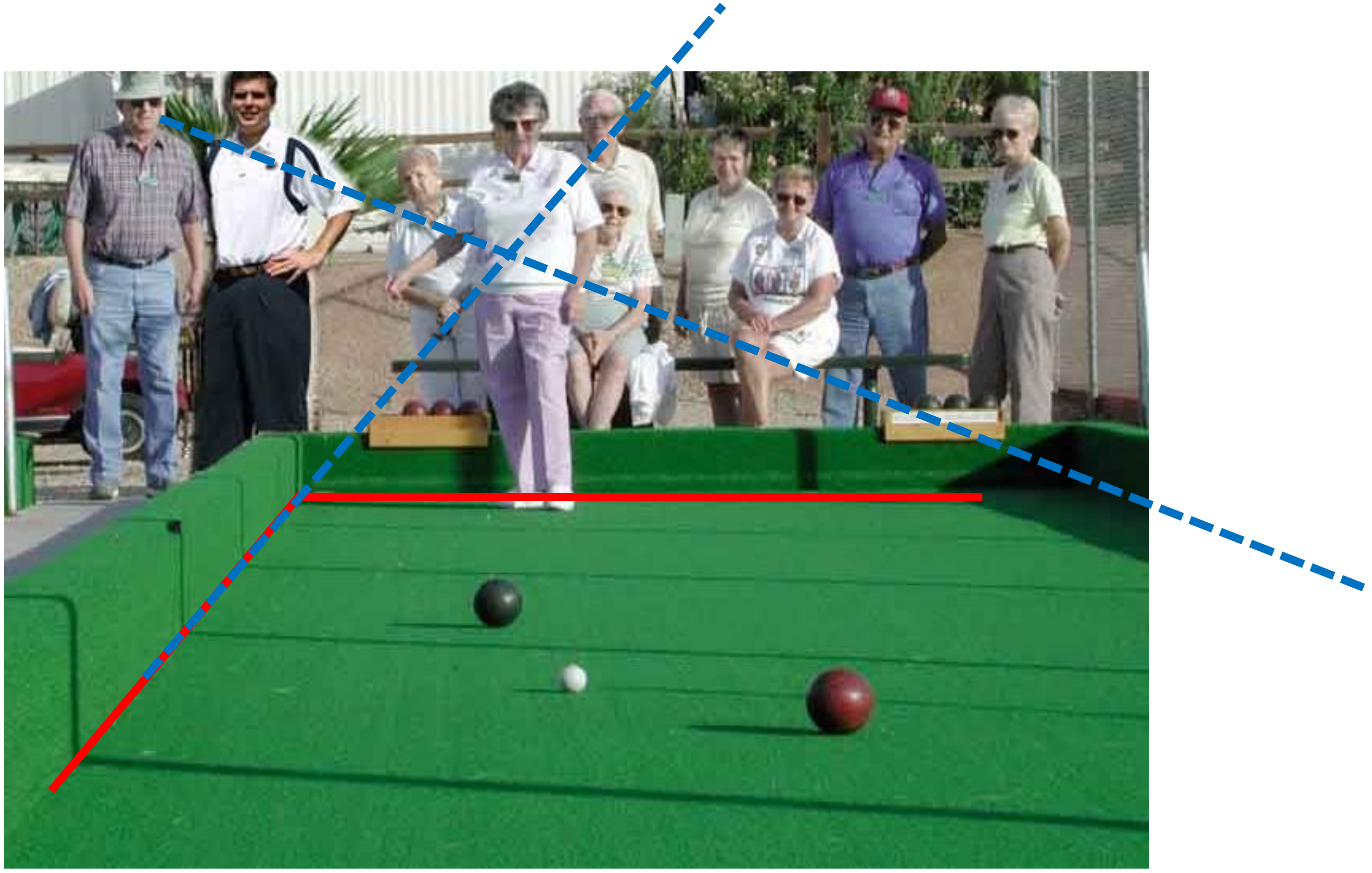
"Ames room"

A clip from "The computer that ate Hollywood" documentary. Dr. Vilayanur S. Ramachandran.

Perspective Projection

What is preserved?

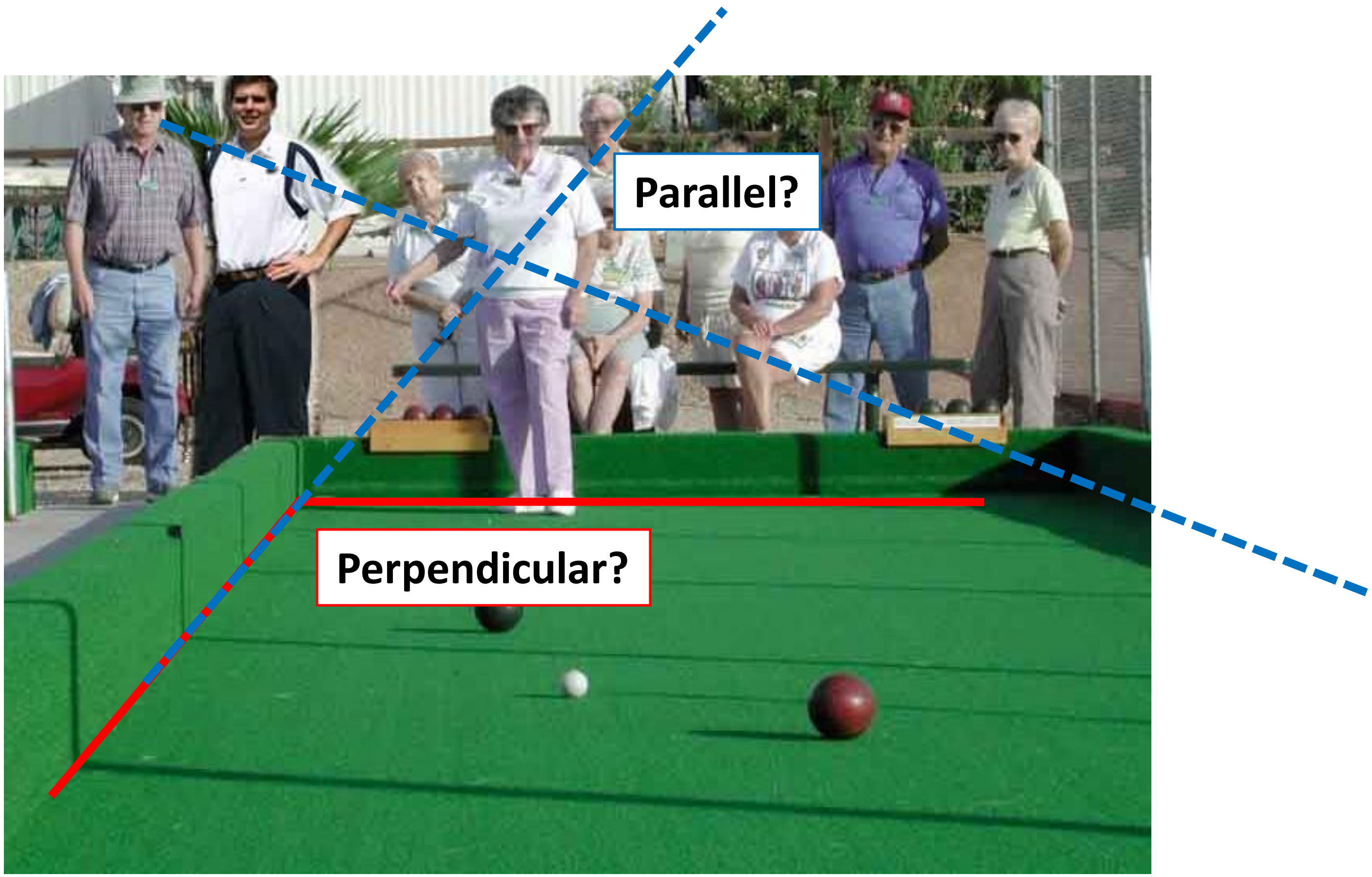
- Straight lines are still straight



Perspective Projection

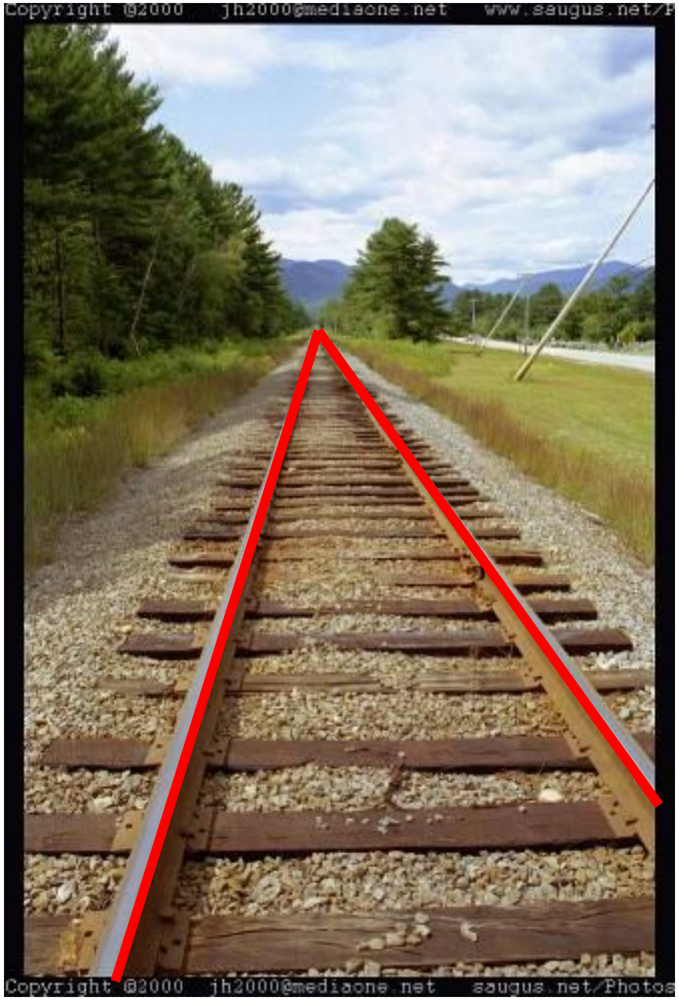
What is lost?

- Length
- Angles



Vanishing points and lines

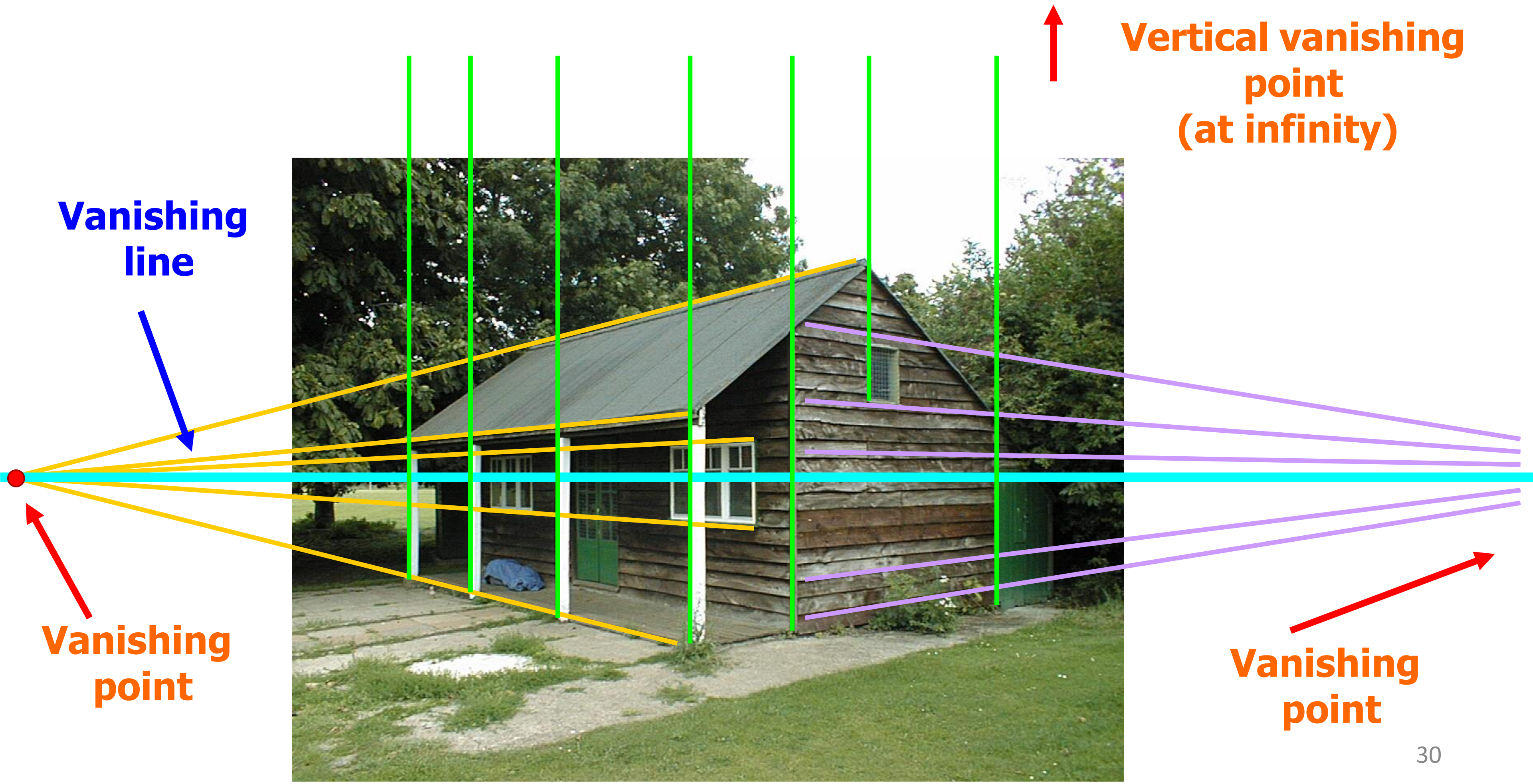
Parallel lines in the world intersect in the image at a “vanishing point”



Vanishing points and lines

Parallel lines in the world intersect in the image at a “vanishing point”

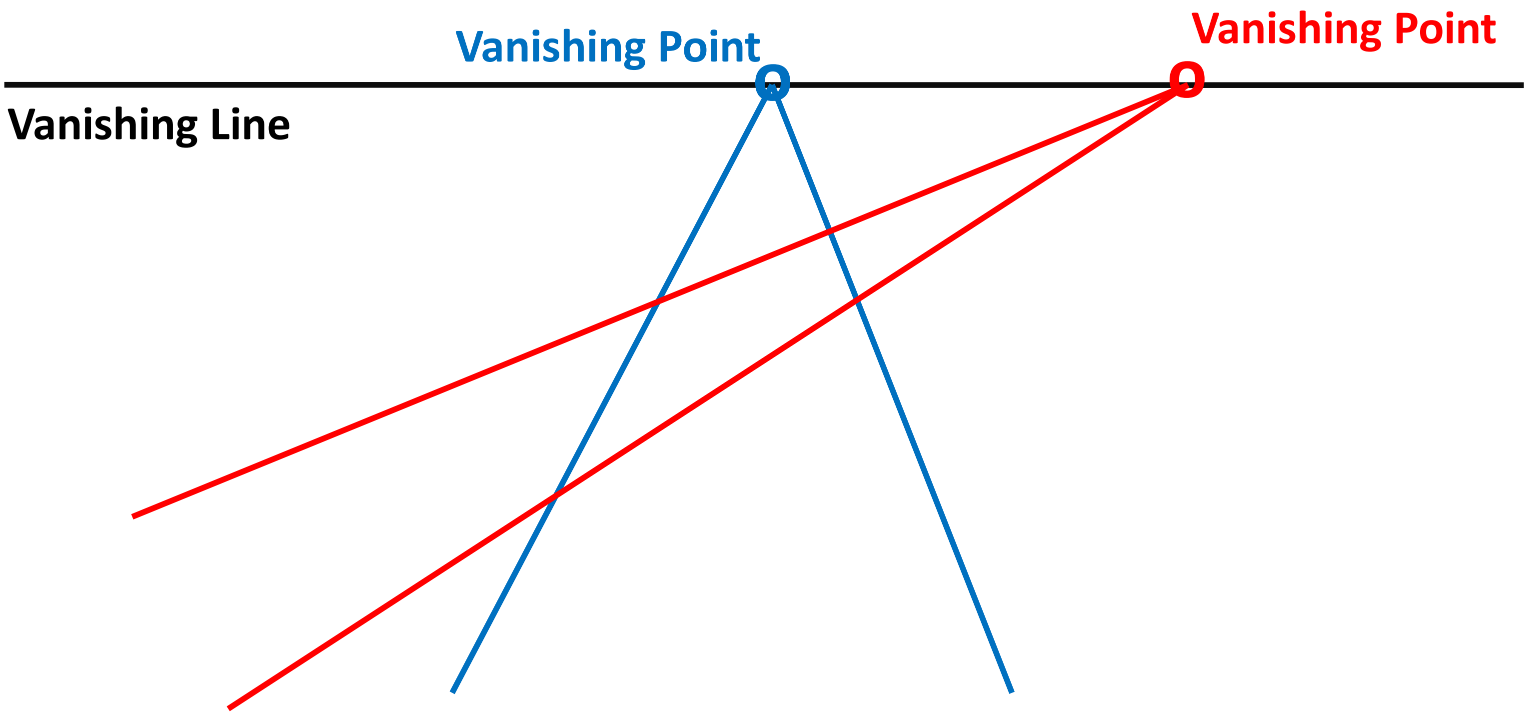
Parallel planes in the world intersect in the image at a “vanishing line”



Vanishing points and lines

Parallel lines in the world intersect in the image at a "vanishing point"

Parallel planes in the world intersect in the image at a "vanishing line"



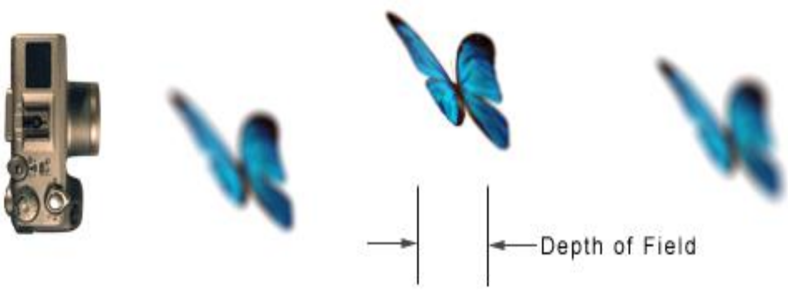
Outline of this lecture

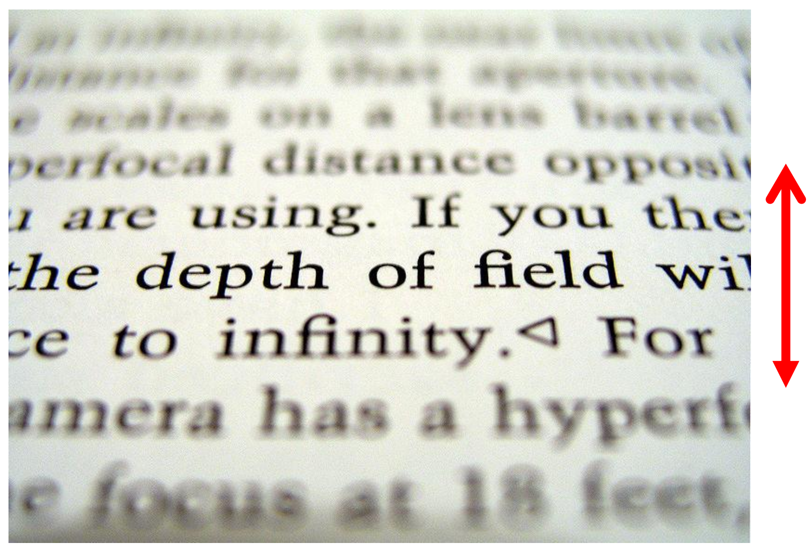
- Image Formation
- Other camera parameters
- Digital camera
- Perspective camera model
- Lens distortion

Focus and Depth of Field

- **Depth of Field (DOF) is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image.**

- Although a lens can precisely focus at only one distance at a time, the decrease in sharpness is gradual on each side of the focused distance, so that within the DOF, the unsharpness is imperceptible under normal viewing conditions

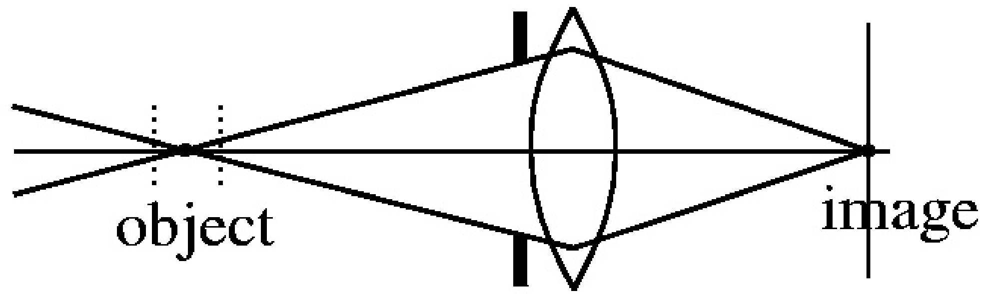


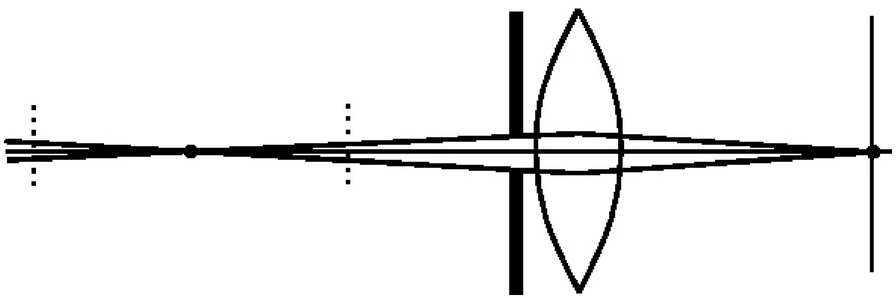


Depth of field

Focus and Depth of Field

- How does the aperture affect the depth of field?

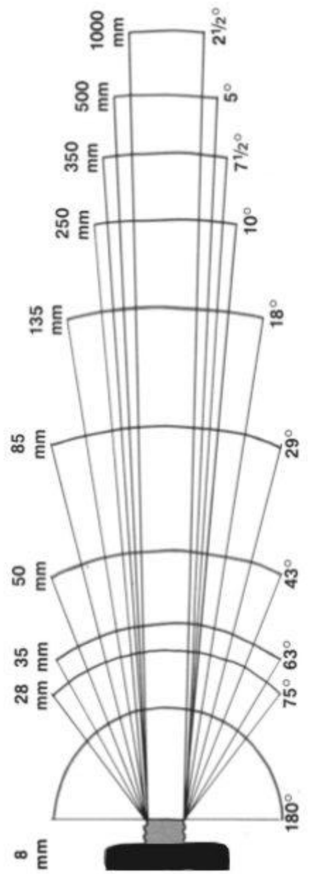




- A smaller aperture increases the DOF but reduces the amount of light into the camera

Field of View (FOV)

Angular measure of portion of 3D space seen by the camera



28 mm lens, 65.5° × 46.4°

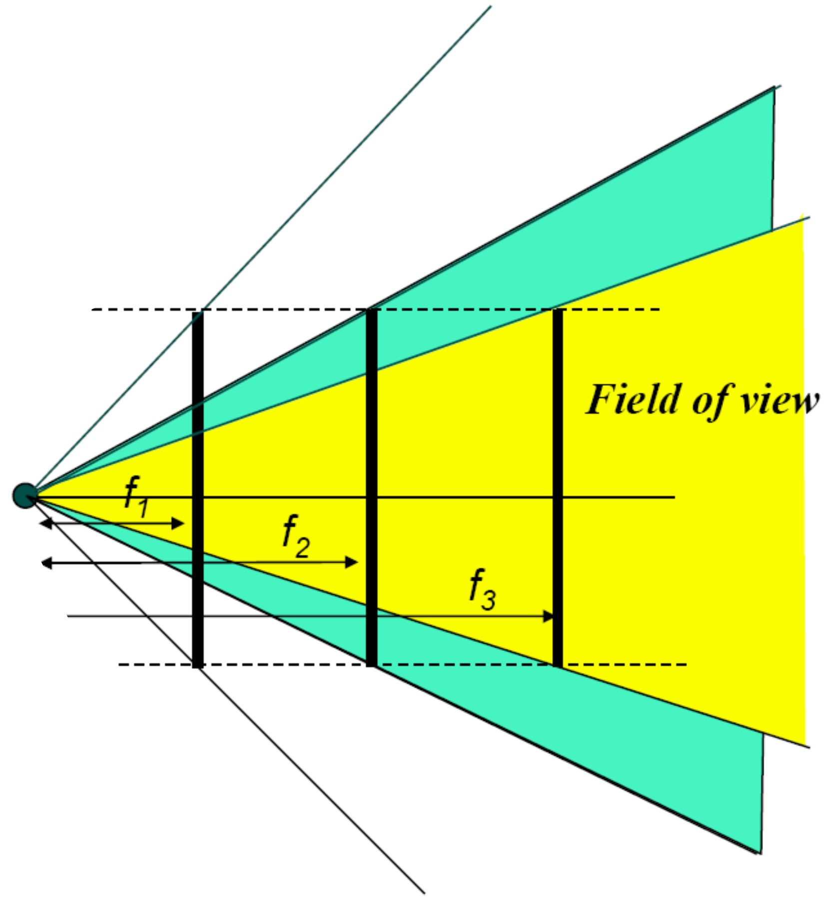
50 mm lens, 39.6° × 27.0°

70 mm lens, 28.9° × 19.5°

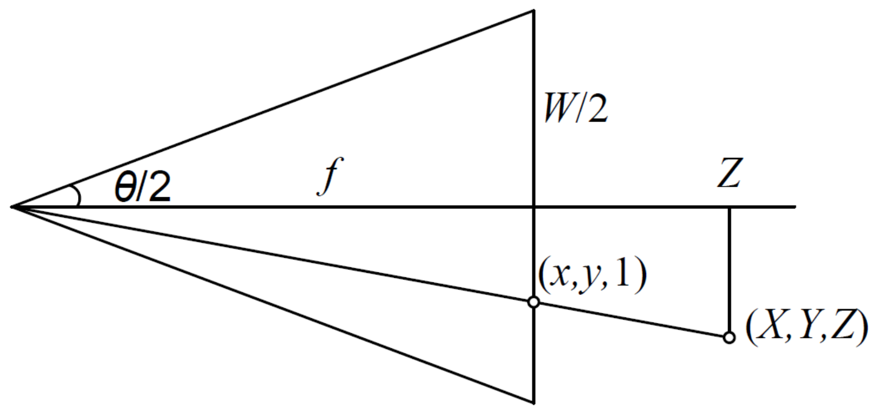
210 mm lens, 9.8° × 6.5°

Field of view depends on focal length

- As ***f*** gets smaller, image becomes more *wide angle*
  - more world points project onto the finite image plane
- As ***f*** gets larger, image becomes more *narrow angle*
  - smaller part of the world projects onto the finite image plane



Relation between field of view and focal length



$$\tan\frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}$$

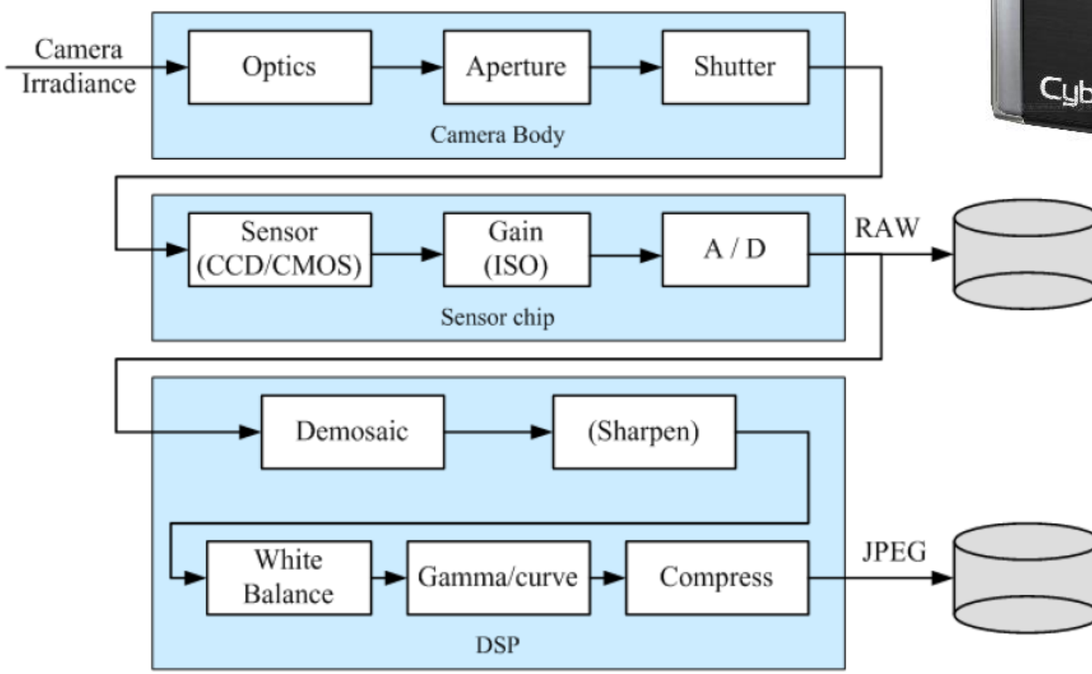
Smaller FOV = larger Focal Length

Outline of this lecture

- Image Formation
- Other camera parameters
- Digital camera
- Perspective camera model
- Lens distortion

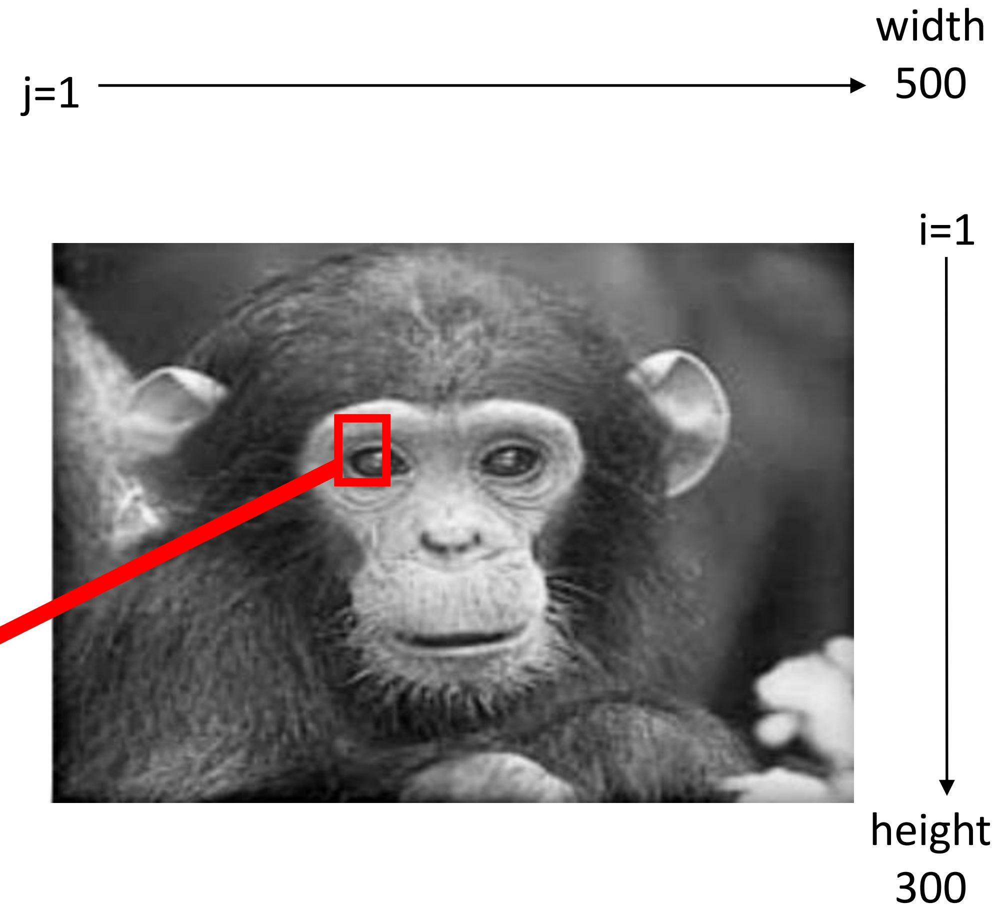
Digital cameras

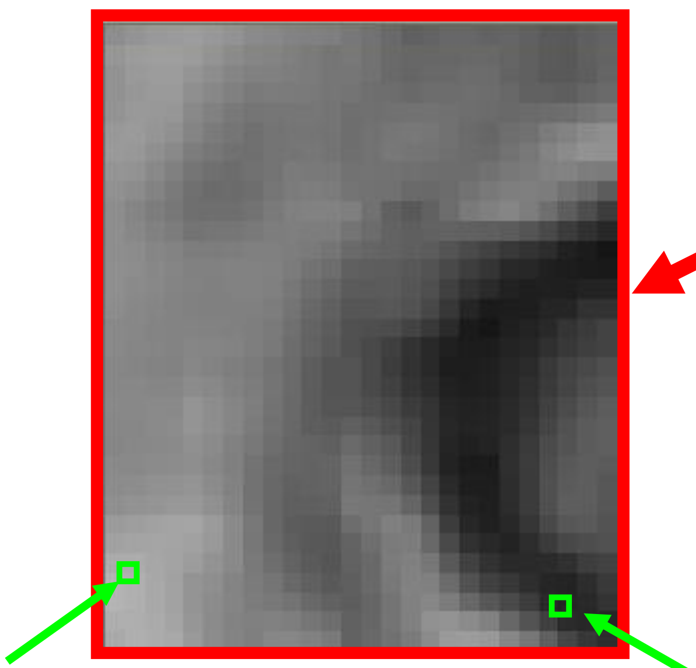
- The film is a array of CCD or CMOS light
  sensitive diodes that convert photons (light energy) into electrons



Digital images

Pixel Intensity with 8 bits ranges between [0,255]





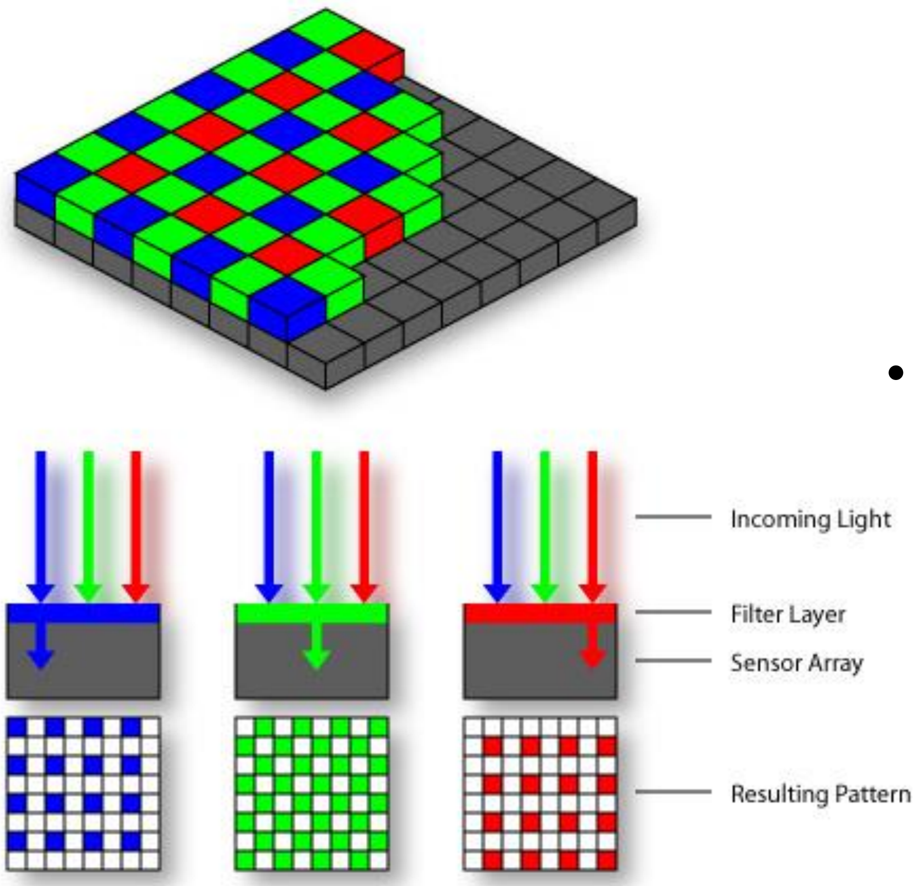
im[176][201] has value 164

im[194][203] has value 37

NB. Matlab coordinates: [rows, cols]; C/C++ [cols, rows]

Color sensing in digital cameras

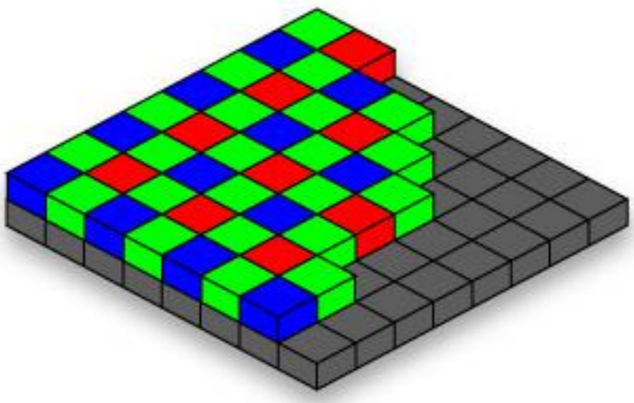
Bayer grid



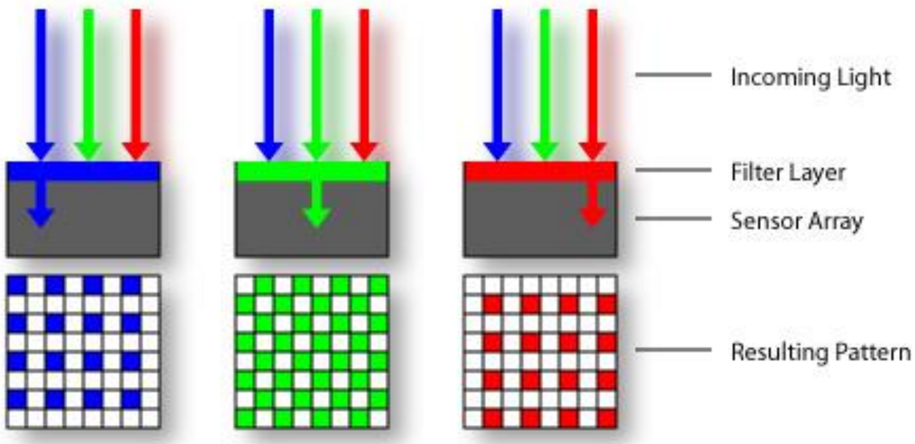
- The Bayer pattern (invented by Bayer in 1976, who worked at Kodak) places green filters over half of the sensors (in a checkerboard pattern), and red and blue filters over the remaining ones.
- This is because the luminance signal is mostly determined by green values and the human visual system is much more sensitive to high frequency detail in luminance than in chrominance.

Color sensing in digital cameras

Bayer grid



For each pixel, estimate missing color components from neighboring values (demosaicing)

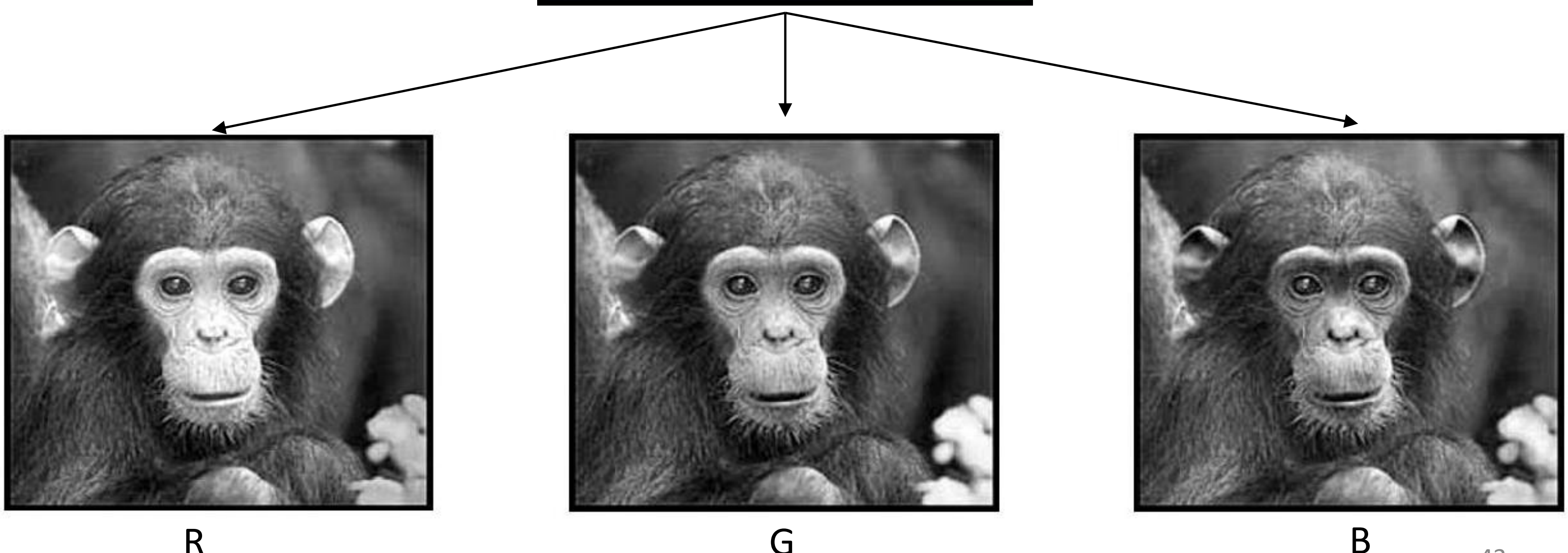


Foveon chip design (http://www.foveon.com) stacks the red, green, and blue sensors beneath each other but has not gained widespread adoption.

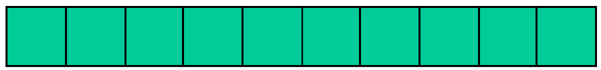
Color sensing in digital cameras

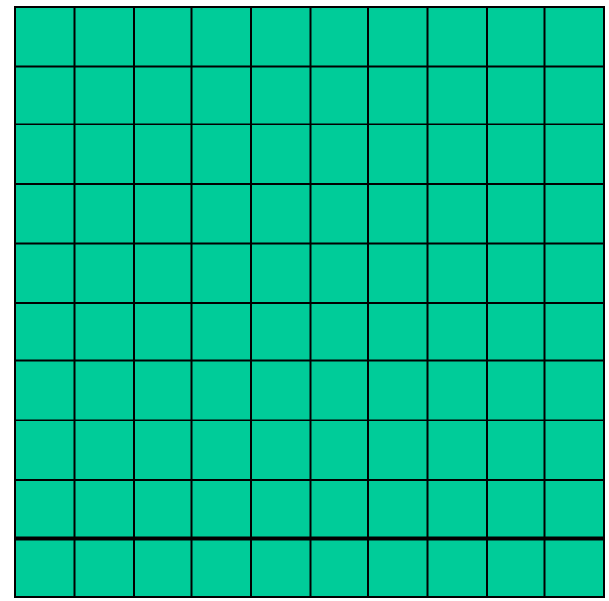
RGB color space

… but there are also many other color spaces… (e.g., YUV)



Rolling vs Global Shutter Camera





Rolling Shutter

- Pixels are exposed roll by roll
- Good for still or slow objects
- May distort image for moving objects

Global Shutter

- All pixels are exposed simultaneously
- Good for moving objects
- No image distortion

Rolling vs Global Shutter Camera

- Rolling Shutter cameras may distort moving objects
- Global Shutter cameras don’t have the problem

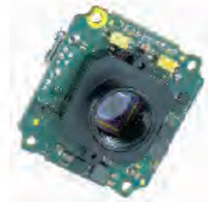
Rolling shutter

Global shutter

An example camera datasheet

mvBlueFOX-IGC / -MLC

Technical Details



Sensors

mvBlueFOX-IGC mvBlueFOX-MLC		Resolution (H x V pixels)	Sensor size (optical)	Pixel size (µm)	Frame rate	Sensor technology	Readout type	ADC resolution / output in bits	Sensor
-200w[1 2]	G/C	752 x 480	1/3"	6 x 6	90	CMOS	Global	10 → 10 / 8	Aptina MT9V
-202b	G/C	1280 x 960	1/3"	3.75 x 3.75	24.6	CMOS	Global	10 → 10 / 8	Aptina MT9M
-202d[1]	G/C	1280 x 960	1/3"	3.75 x 3.75	24.6	CMOS	Rolling	10 → 10 / 8	Aptina MT9M
-205[2]	G/C	2592 x 1944	1/2.5"	2.2 x 2.2	5.8	CMOS	Global Reset	10 → 10 / 8	Aptina MT9P

[1] High Dynamic Range (HDR) mode supported

[2] Software trigger supported

Sample: **mvBlueFOX-IGC200wG** means version with housing and 752 x 480 CMOS gray scale sensor.
mvBlueFOX-MLC200wG means single-board version without housing and with 752 x 480 CMOS gray scale sensor.

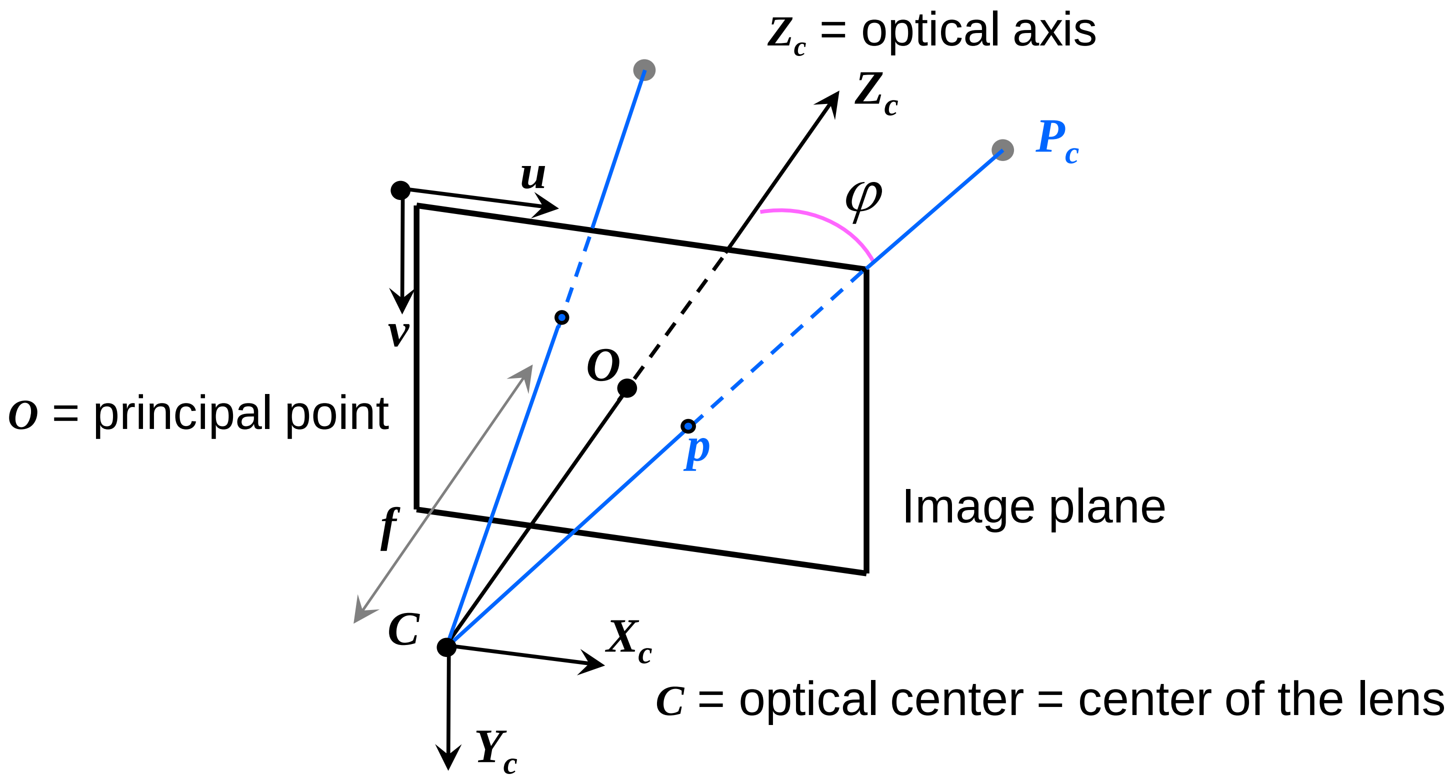
Hardware Features

Gray scale / Color	Gray scale (G) / Color (C)
Interface	USB 2.0 (up to 480 Mbit/s)
Image formats	Mono8, Mono10, BayerGR8, BayerGR10
Triggers	External hardware based (optional), software based (depending on the sensor) or free run
Size w/o lens (W x H x L) | Weight w/o lens	mvBlueFOX-IGC: 39.8 x 39.8 x 16.5 mm | approx. 10 g mvBlueFOX-MLC: 35 x 33 x 25 mm (without lens mount) | approx. 80 g
Permissible ambient temperature	Operation: 0 .. 45 °C / 30 to 80 % RH Storage: -20 .. 60 °C / 20 to 90 % RH
Lens mounts	Back focus adjustable C/CS-mount lens holder / C-mount, CS-mount or optional S-mount
Digital I/Os	mvBlueFOX-IGC (optional): 1 / 1 opto-isolated mvBlueFOX-MLC: 1 / 1 opto-isolated or 2 / 2 TTL compliant
Conformity	CE, FCC, RoHS
Driver	mvIMPACT Acquire SDK
Operating systems	Windows®, Linux® - 32 bit and 64 bit
Special features	Micro-PLC, automatic gain / exposure control, binning, screw lock connectors

Outline of this lecture

- Image Formation
- Other camera parameters
- Digital camera
- Perspective camera model
- Lens distortion

Perspective Camera

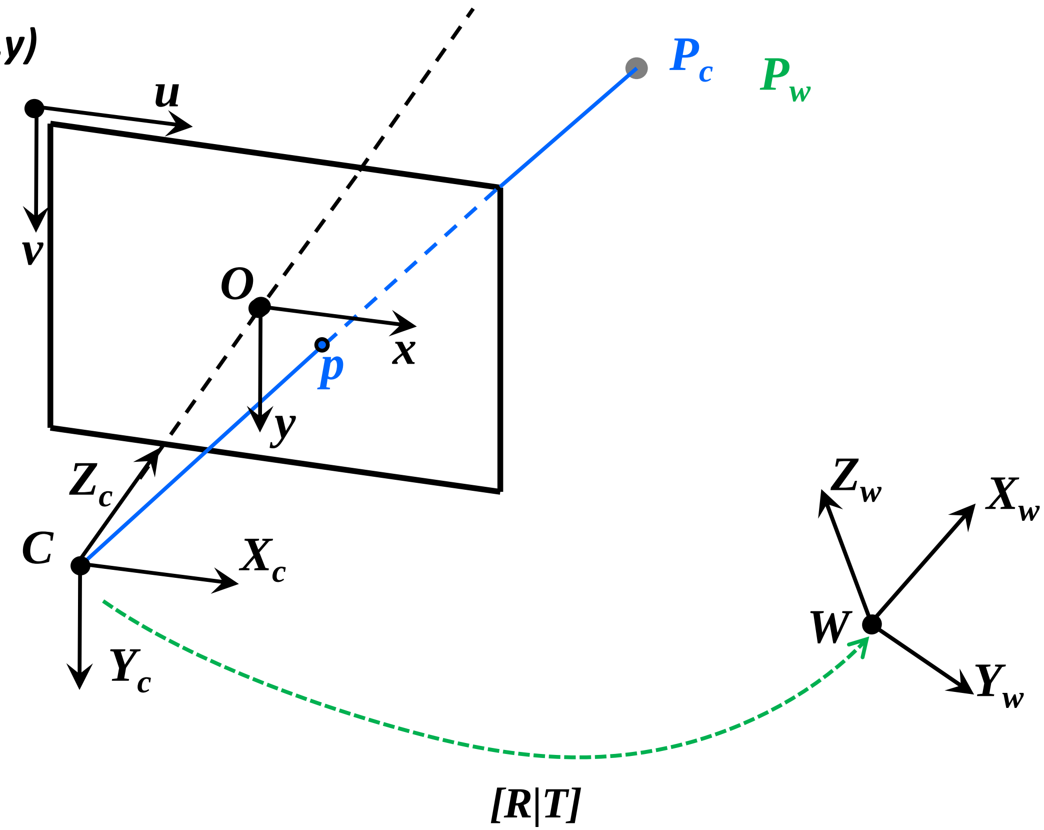


- For convenience, the image plane is usually represented in front of C such that the image preserves the same orientation (i.e. not flipped)
- Note: **a camera does not measure distances but angles!**
  ⇨ a camera is a "bearing sensor"

From World to Pixel coordinates

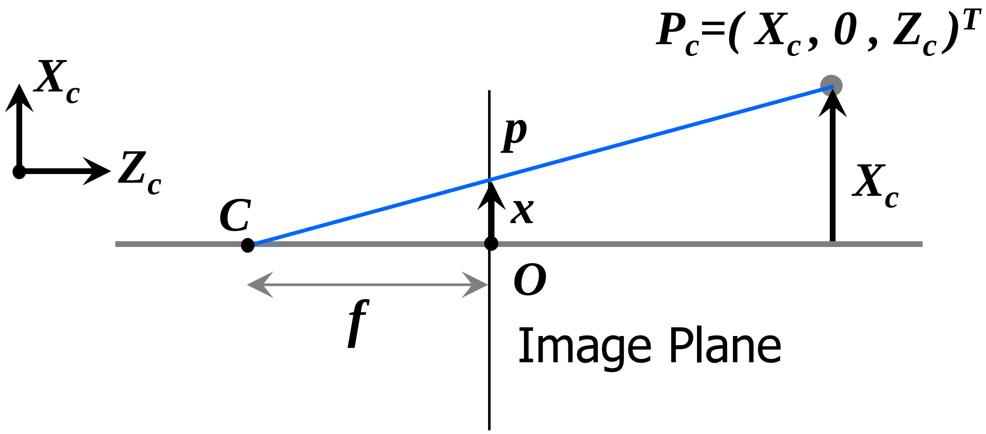
Goal: Find pixel coordinates ***(u,v)*** of point P_w in the world frame:

1. Convert world point P_w to camera point P_c through rigid body transform ***[R, T]***
2. Convert P_c to image-plane coordinates ***(x,y)***
3. Convert ***(x,y)*** to (discretized) pixel coordinates ***(u,v)***



Perspective Projection (1)

From the Camera frame to the image plane



- The Camera point $P_c = (X_c, 0, Z_c)^T$ projects to $p = (x, y)$ onto the image plane
- From similar triangles: $$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$
- Similarly, in the general case:

$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$

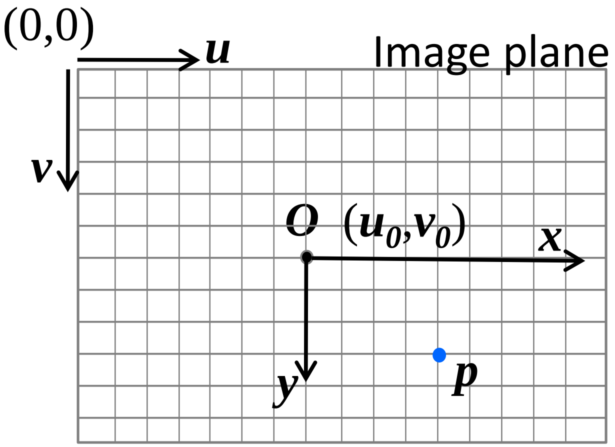
1. Convert P_c to image-plane coordinates (x,y)

2. Convert (x,y) to (discretised) pixel coordinates (u,v)

Perspective Projection (2)

From the Camera frame to pixel coordinates

- To convert ***p*** from the local image plane coords ***(x,y)*** to the pixel coords ***(u,v)***, we need to account for:
  - the pixel coords of the camera optical center $O = (u_0, v_0)$
  - Scale factors k_u, k_v for the pixel-size in both dimensions



So:

$$u = u_0 + k_u x \Rightarrow u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + k_v y \Rightarrow v = v_0 + \frac{k_v f Y_c}{Z_c}$$

- Use **Homogeneous Coordinates** for linear mapping from 3D to 2D, by introducing an extra element (scale):

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \quad \Rightarrow \quad \tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Perspective Projection (3)

- So:

$$u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + \frac{k_v f Y_c}{Z_c}$$

Expressed in matrix form and homogeneous coordinates:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Or alternatively

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Focal length in pixels

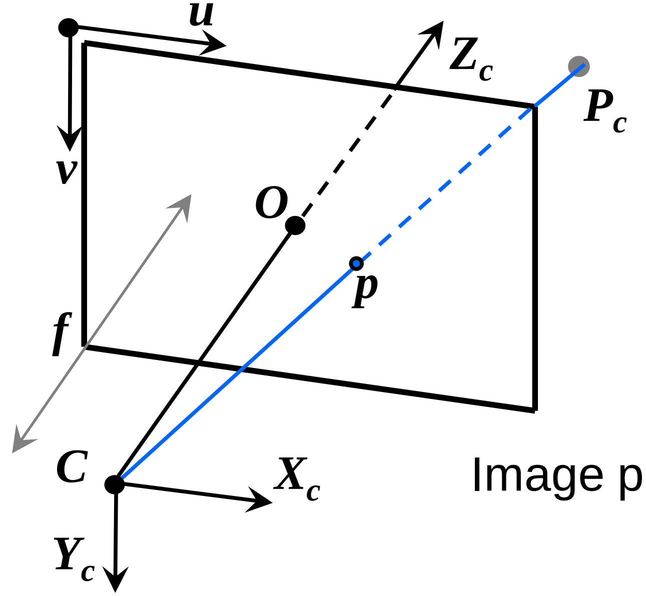


Image plane (CCD)

K is called "Calibration matrix" or "Matrix of Intrinsic Parameters"

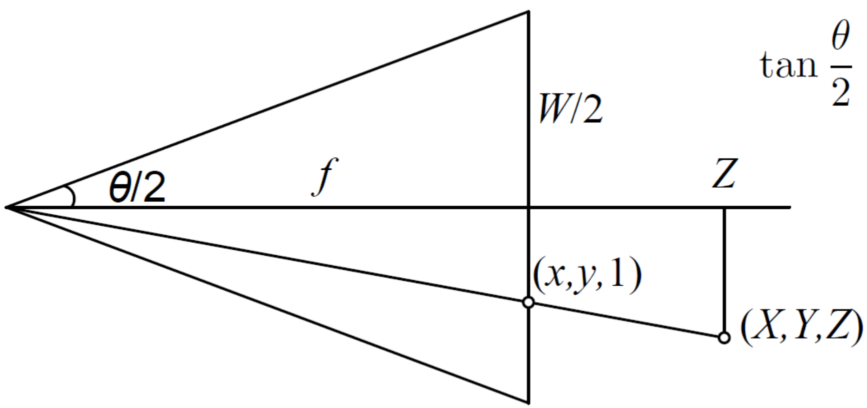
In the past it was common to assume a skew factor ($K_{12} \neq 0$) to account for possible skew in the pixel manufacturing process. However, the camera manufacturing process today is so good that we can safely assume $K_{12} = 0$ and $\alpha_u = \alpha_v$.

Exercise 1

- Determine the Intrinsic Parameter Matrix (K) for a digital camera with image size 640×480 pixels and horizontal field of view equal to $90°$
- Assume the principal point in the center of the image and square pixels
- What is the vertical field of view?

Exercise 1

- Determine the Intrinsic Parameter Matrix (K) for a digital camera with image size 640×480 pixels and horizontal field of view equal to $90°$
- Assume the principal point in the center of the image and square pixels



$$\tan\frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}$$

$$f = \frac{640}{2\tan\frac{\theta}{2}} = 320\,\boldsymbol{pixels}$$

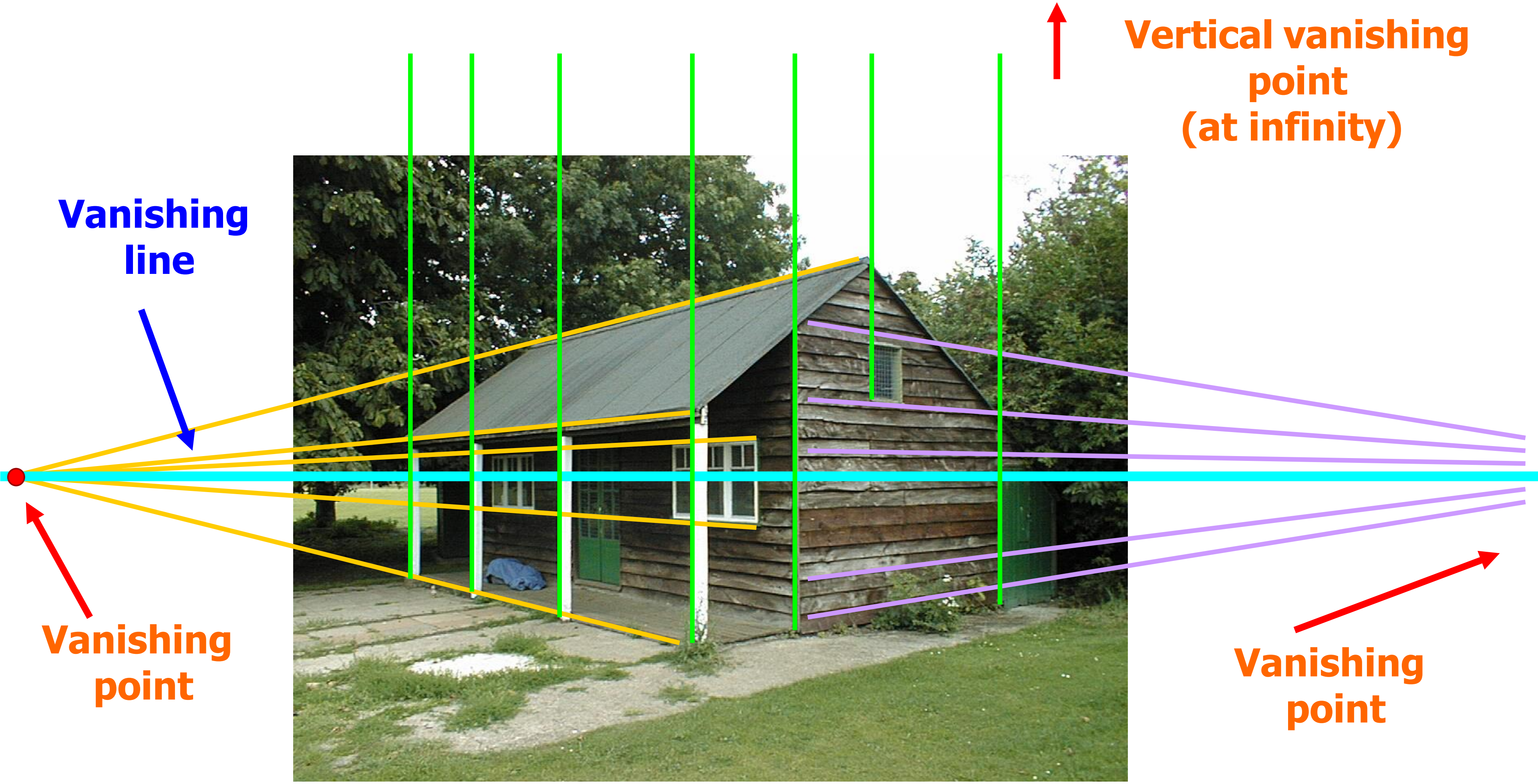
$$K = \begin{bmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

- What is the vertical field of view?

$$\theta_V = 2\tan^{-1}\frac{H}{2f} = 2\tan^{-1}\frac{480}{2\cdot 320} = 73.74°$$

Exercise 2

- Prove that world's parallel lines intersect at a vanishing point in the camera image



Slide from Efros, Photo from Criminisi

Exercise 2

- Prove that world's parallel lines intersect at a vanishing point in the camera image
- Let's consider the perspective projection equation in camera metric coordinates:

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

- Two parallel 3D lines have parametric equations:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + s \begin{bmatrix} l \\ m \\ n \end{bmatrix} \qquad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + s \begin{bmatrix} l \\ m \\ n \end{bmatrix}$$

- Now substitute this into the camera perspective projection equation and compute the limit for $s \to \infty$

$$\lim_{s\to\infty} f\,\frac{X_i + sl}{Z_i + sn} = f\,\frac{l}{n} = x_{VP}, \qquad \lim_{s\to\infty} f\,\frac{Y_i + sm}{Z_i + sn} = f\,\frac{m}{n} = y_{VP}$$

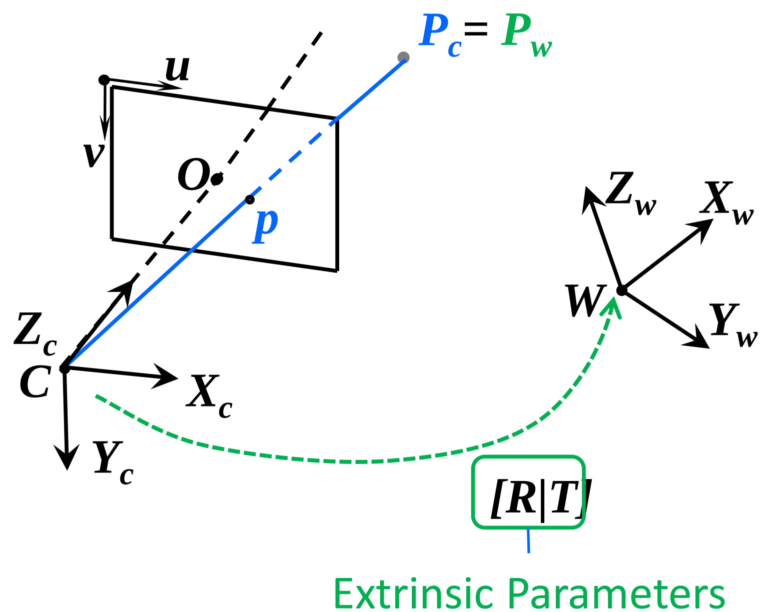
- The result solely depends on the direction vector of the line. These are the image coordinates of the vanishing point (VP).
- What is the intuitive interpretation of this?

Perspective Projection (4)

From the World frame to the Camera frame

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R & T \end{array}\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Projection Matrix (M)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boxed{K\left[R \middle| T\right]} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Perspective Projection Equation

Normalized image coordinates

In both computer vision and robotics, it is often convenient to use normalized image coordinates

- Let (u, v) be the pixel coordinates of an image point
- We define the normalized image coordinates $(\bar{x}, \bar{y})$

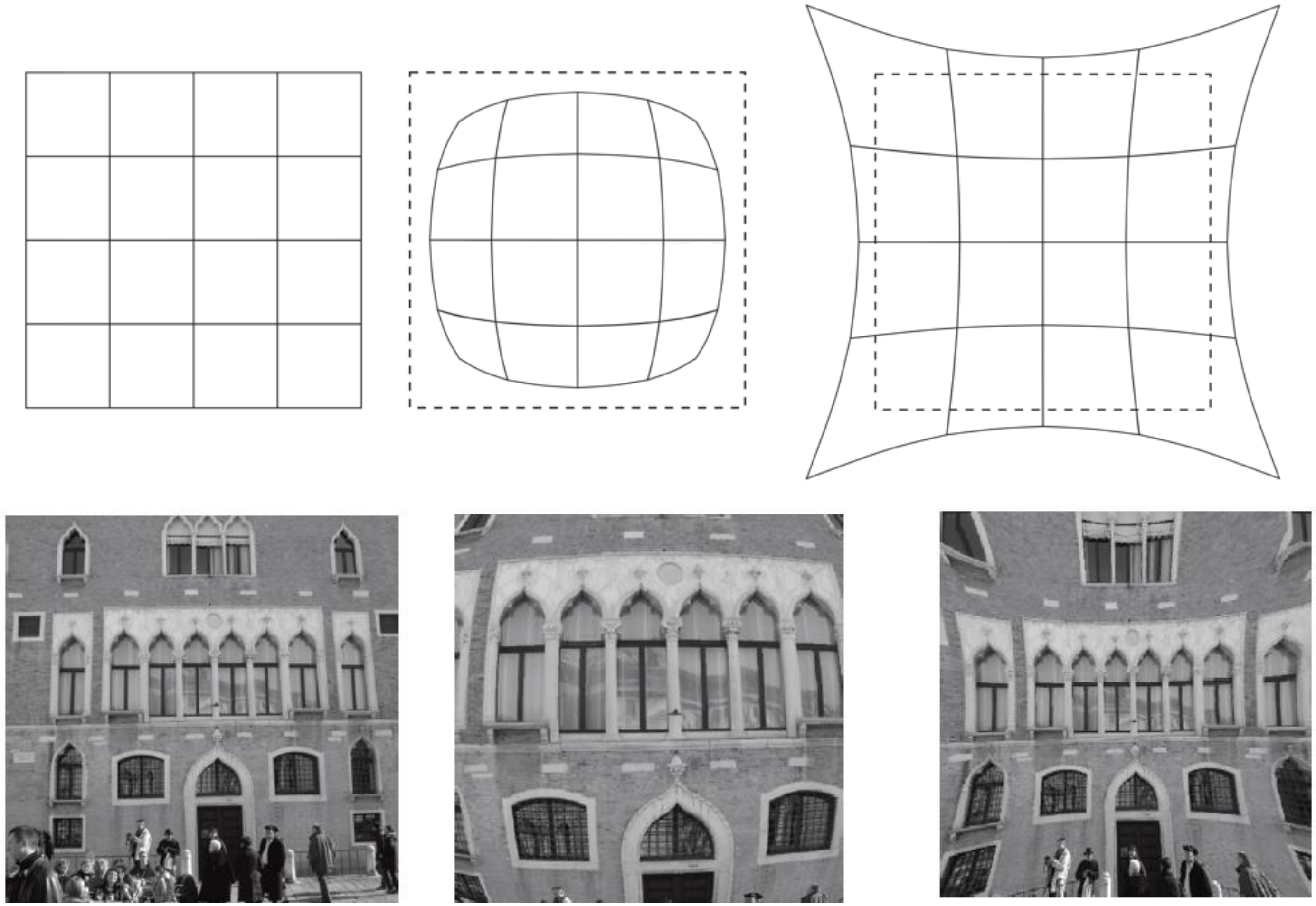
$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Normalized image coordinates can be interpreted as image coordinates on an virtual image plane with focal length equal to 1 meter

Outline of this lecture

- Image Formation
- Other camera parameters
- Digital camera
- Perspective camera model
- Lens distortion

Radial Distortion



No distortion Barrel distortion Pincushion

Radial Distortion

- The standard model of radial distortion is a transformation **from the ideal coordinates** (u, v) (i.e., **non-distorted**) **to the real, observed coordinates** (distorted) (u_d, v_d)

- For a given non distorted image point (u, v), the amount of distortion is a nonlinear function of it distance r from principal point. For most lenses, a simple quadratic model of distortion produces good results

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

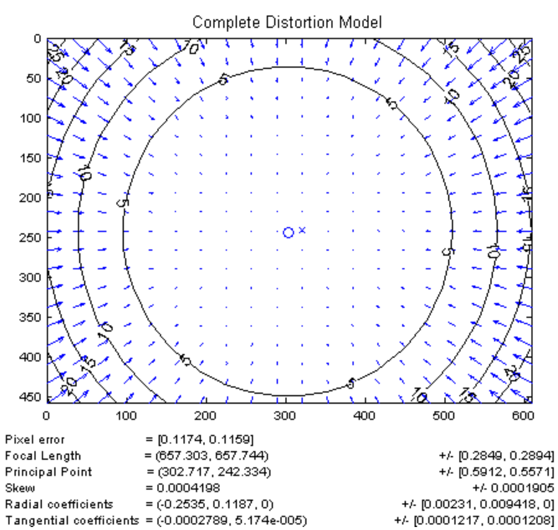
where $$r^2 = (u - u_0)^2 + (v - v_0)^2$$

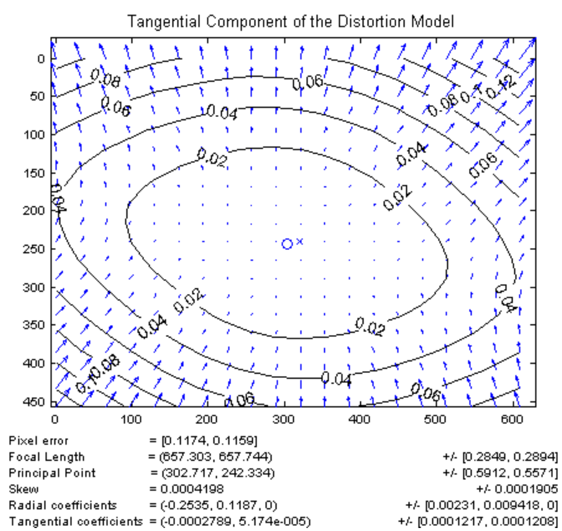
Radial & Tangential Distortion in the OpenCV and Matlab Camera Models

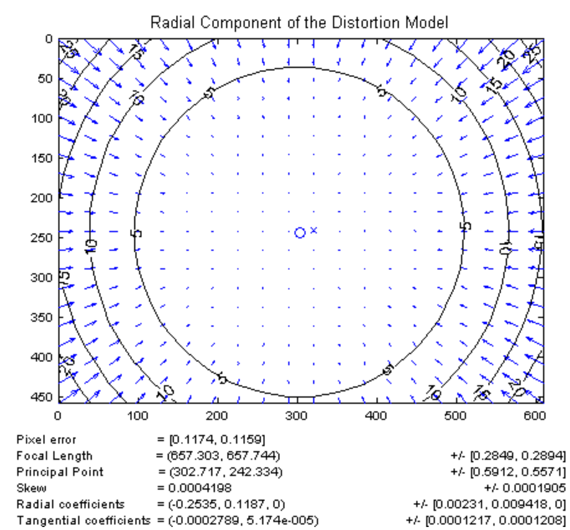
- **Radial Distortion:** Depending on the amount of distortion (an thus on the camera field of view), higher order terms can be introduced for the radial distortion
- **Tangential Distortion**: if the lens is misaligned (not perfectly orthogonal to the image sensor), a non radial distortion is introduced

This formula won't be asked at the exam

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \underbrace{(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)\begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix}}_{\text{Radial distortion}} + \underbrace{\begin{bmatrix} 2k_4(u-u_0)(v-v_0) + k_5(r^2 + 2(u-u_0)^2) \\ k_4(r^2 + 2(v-v_0)^2 + 2k_5(u-u_0)(v-v_0) \end{bmatrix}}_{\text{Tangential distortion}} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$







The **left figure** shows the impact of the **complete distortion model (radial + tangential)** on each pixel of the image. Each arrow represents the effective displacement of a pixel induced by the lens distortion. Observe that points at the corners of the image are displaced by as much as **25 pixels**. The **center figure** shows the impact of the **tangential component of distortion**. On this plot, the **maximum induced displacement is 0.14 pixel** (at the upper left corner of the image). Finally, the **right figure** shows the impact of the **radial component of distortion**. This plot is very similar to the full distortion plot, showing the tangential component could very well be discarded in the complete distortion model. On the three figures, the cross indicates the center of the image, and the circle the location of the principal point.

Summary: Perspective projection equations

- To recap, a 3D world point $P = (X_w, Y_w, Z_w)$ projects into the image point $p = (u, v)$

$$\tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{where} \quad K = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

and λ is the depth ($\lambda = Z_C$) of the scene point

- If we want to take into account the radial distortion, then the distorted coordinates (u_d, v_d) (in pixels) can be obtained as

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

where $$r^2 = (u - u_0)^2 + (v - v_0)^2$$

- See also the OpenCV documentation:

http://docs.opencv.org/2.4.13.3/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Summary (things to remember)

- Perspective Projection Equation
- Intrinsic and extrinsic parameters (K, R, t)
- Homogeneous coordinates
- Normalized image coordinates
- Image formation equations (including simple radial distortion)
- Chapter 4 of Autonomous Mobile Robot book: http://rpg.ifi.uzh.ch/docs/teaching/2018/Ch4_AMRobots.pdf

Understanding Check

Are you able to:

- Explain what a Blur Circle is?
- Derive the thin lens equation and perform the pinhole approximation?
- Define vanishing points and lines?
- Prove that parallel lines intersect at vanishing points?
- Explain how to build an Ames room?
- Derive a relation between the field of view and the focal length?
- Explain the perspective projection equation, including lens distortion and world to camera projection?