Digitale Medientechnologien

ACV – Applied Computer Vision

Bachelor Medientechnik & Creative Computing

Matthias Zeppelzauer
matthias.zeppelzauer@fhstp.ac.at

Djordje Slijepcevic
djordje.slijepcevic@fhstp.ac.at

Credits

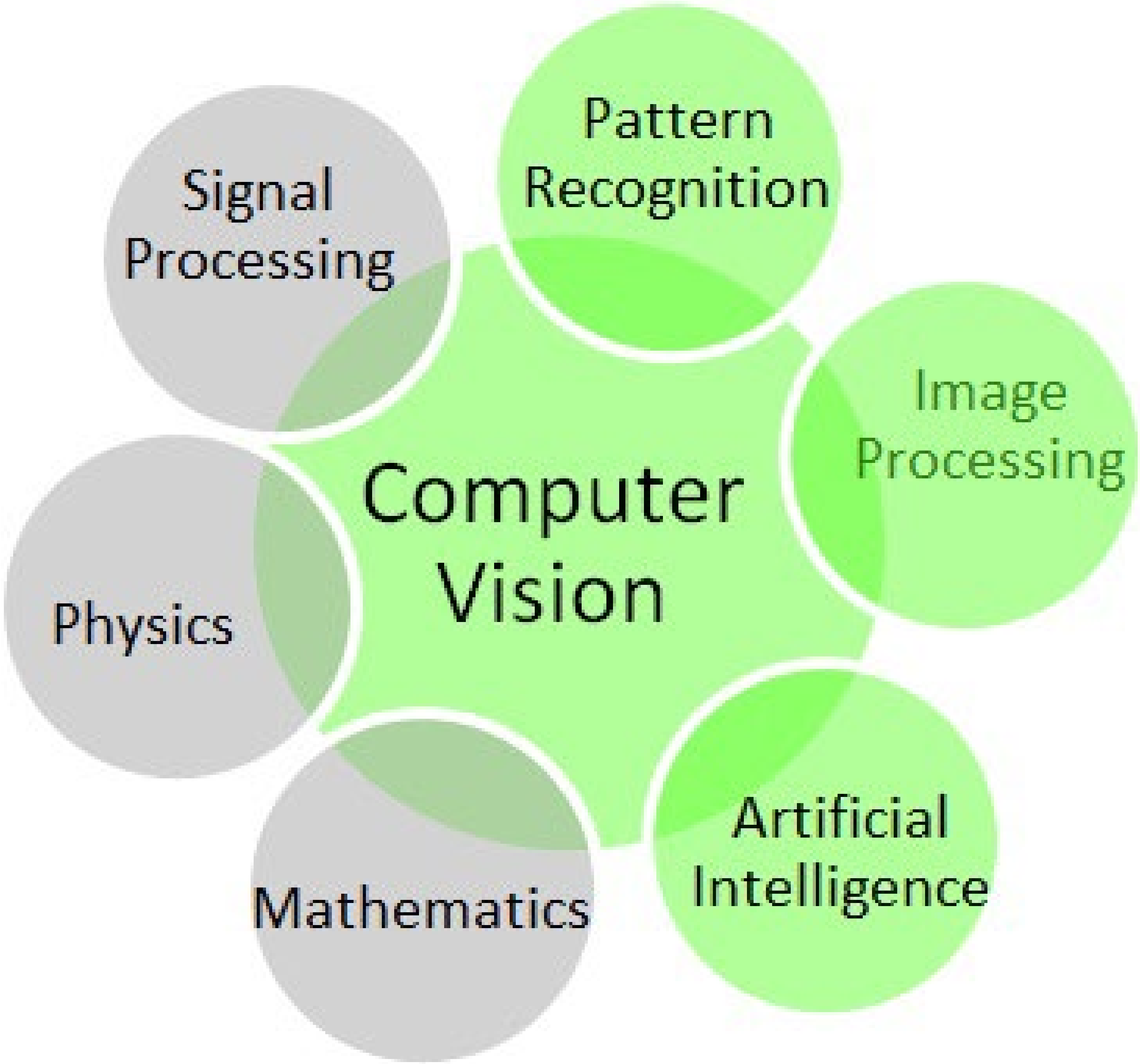
- Antonio Torralba
- Alexei Efros
- Leon Sigal
- Kristen Grauman
- Steve Seitz
- James Hays

Today

- The challenges of CV: Why is CV hard?
- Image Formation: how are images generated?
- Image Filtering: applying basic filters to images smoothing, sharpening

The Field of Computer Vision

Related areas to Computer Vision



Why is Computer Vision Hard?

9 Challenges in Computer Vision

Challenge 1: view point variation

Michelangelo 1475-1564

slide by Fei Fei, Fergus & Torralba

Challenge 2: illumination

slide credit: S. Ullman

Challenge 3: occlusion

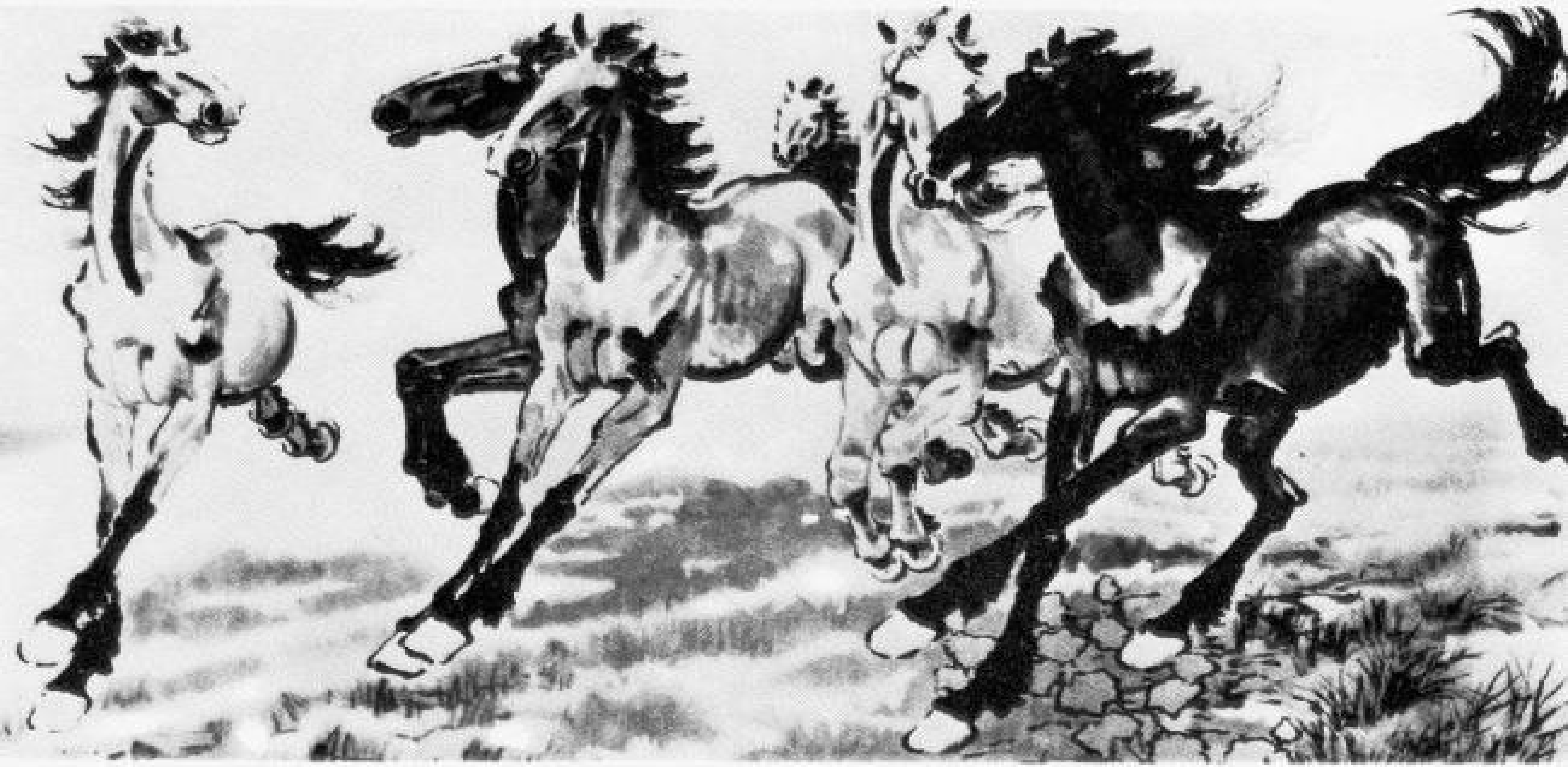
Magritte, 1957

slide by Fei Fei, Fergus & Torralba

Challenge 4: scale

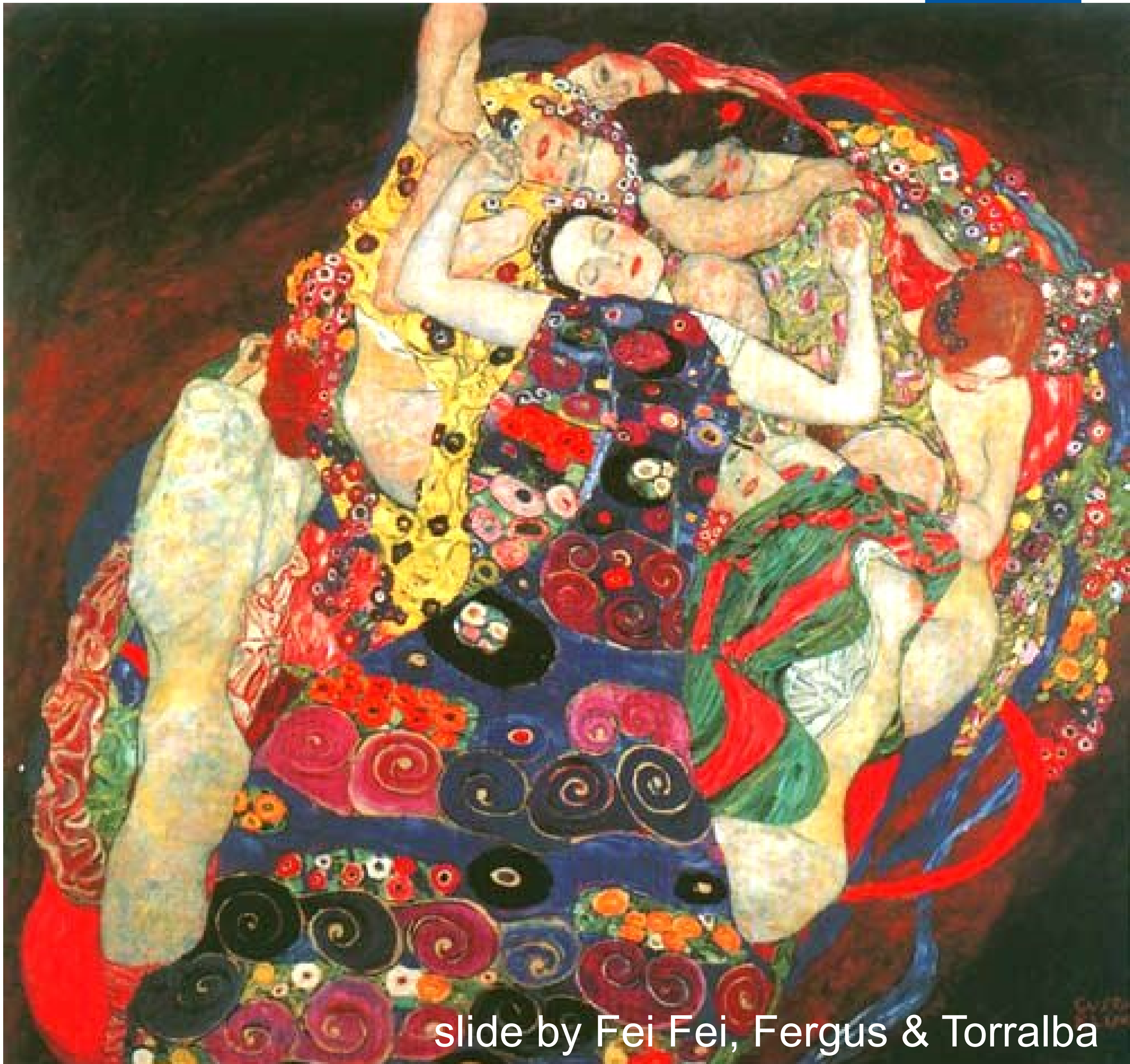
slide by Fei Fei, Fergus & Torralba

Challenge 5: deformation



Xu, Beihong 1943

Challenge 6: background clutter

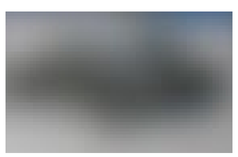


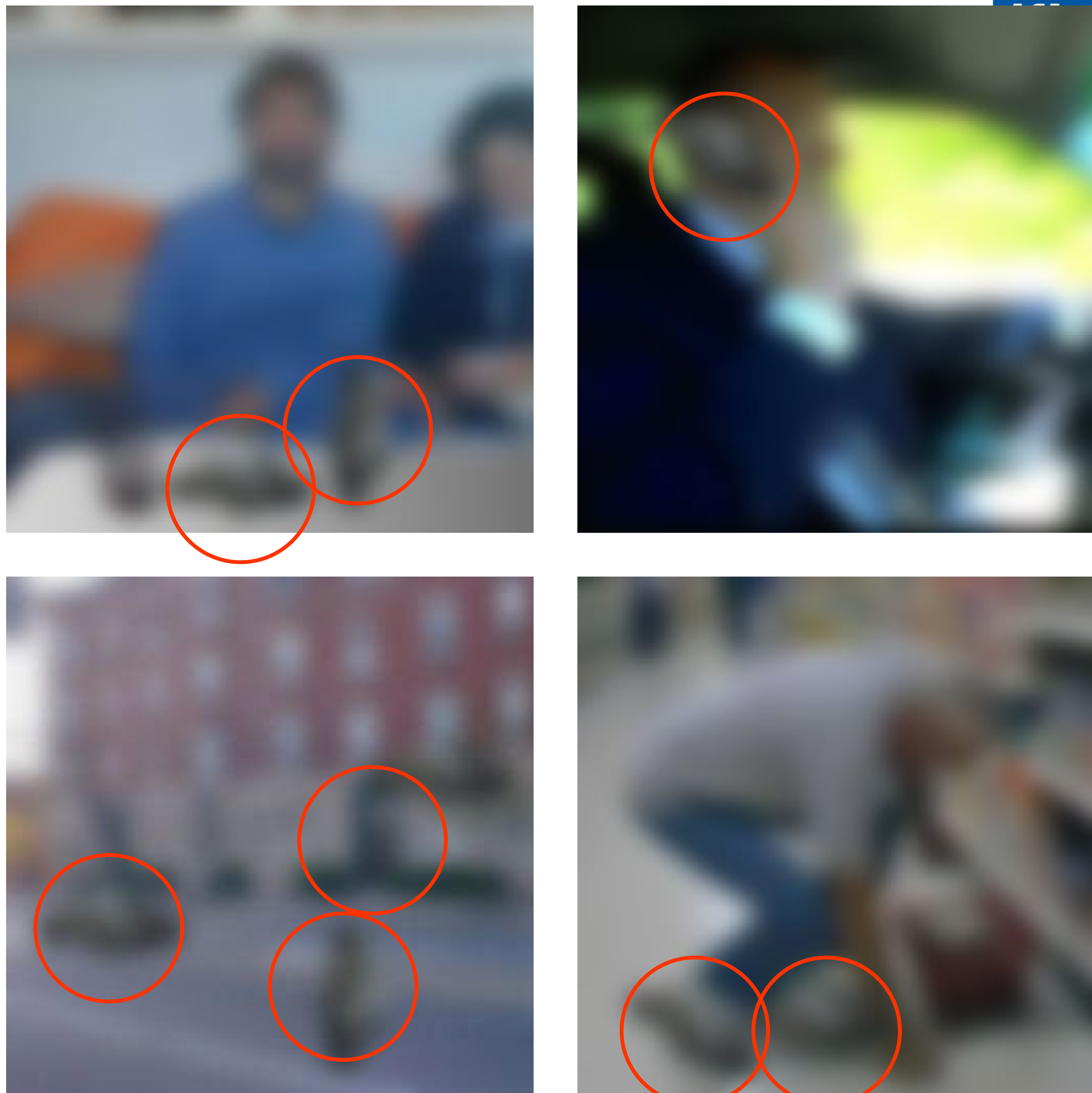
Klimt, 1913

Challenge 7: object intra-class variation

slide by Fei-Fei, Fergus & Torralba

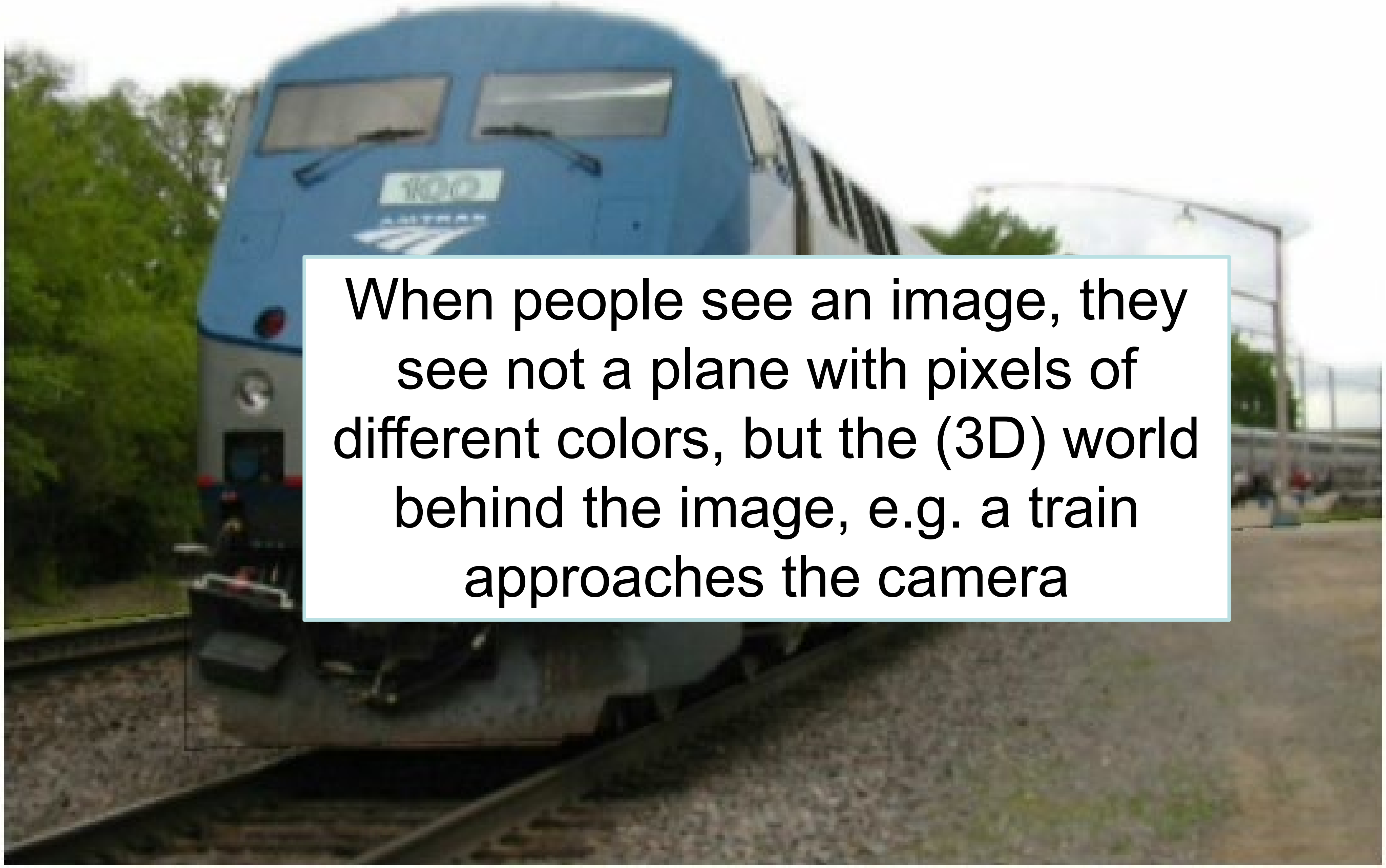
Challenge 8: local ambiguity





slide by Fei-Fei, Fergus & Torralba

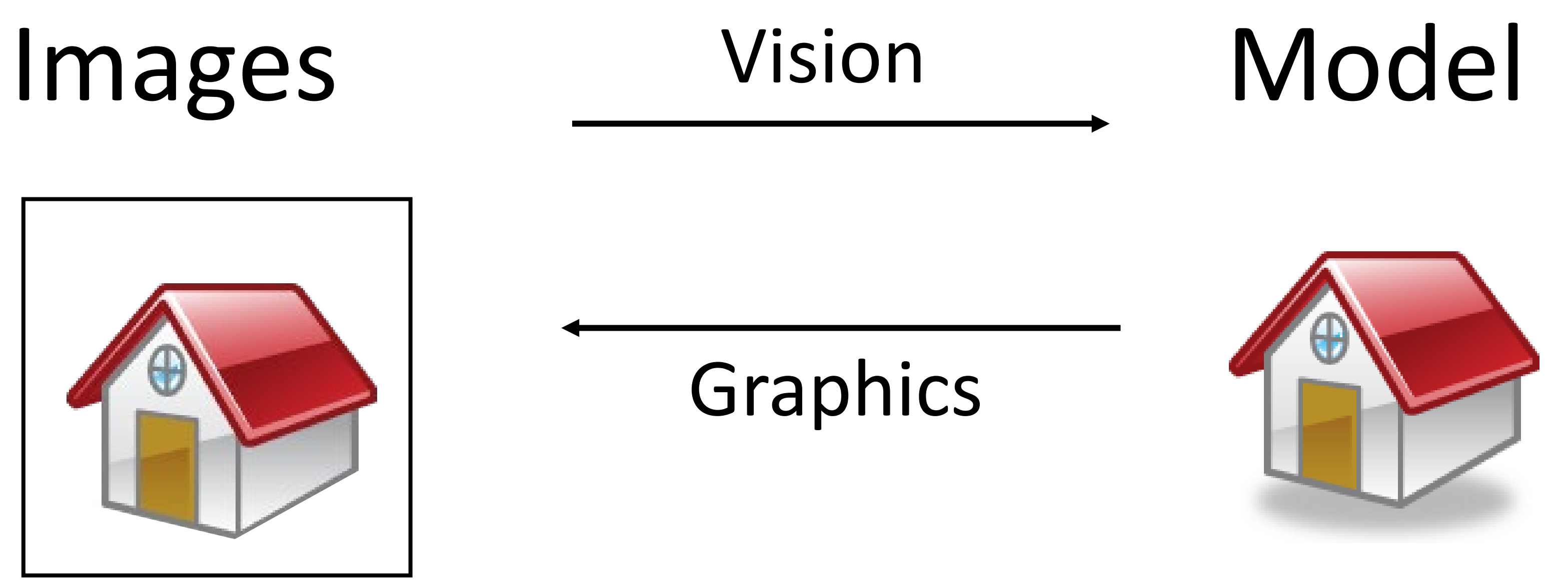
Challenge 9: the world behind the image



When people see an image, they see not a plane with pixels of different colors, but the (3D) world behind the image, e.g. a train approaches the camera

Again: Why is CV hard?

- Vision and graphics



Inverse problems: analysis and synthesis.

Inverse Problems

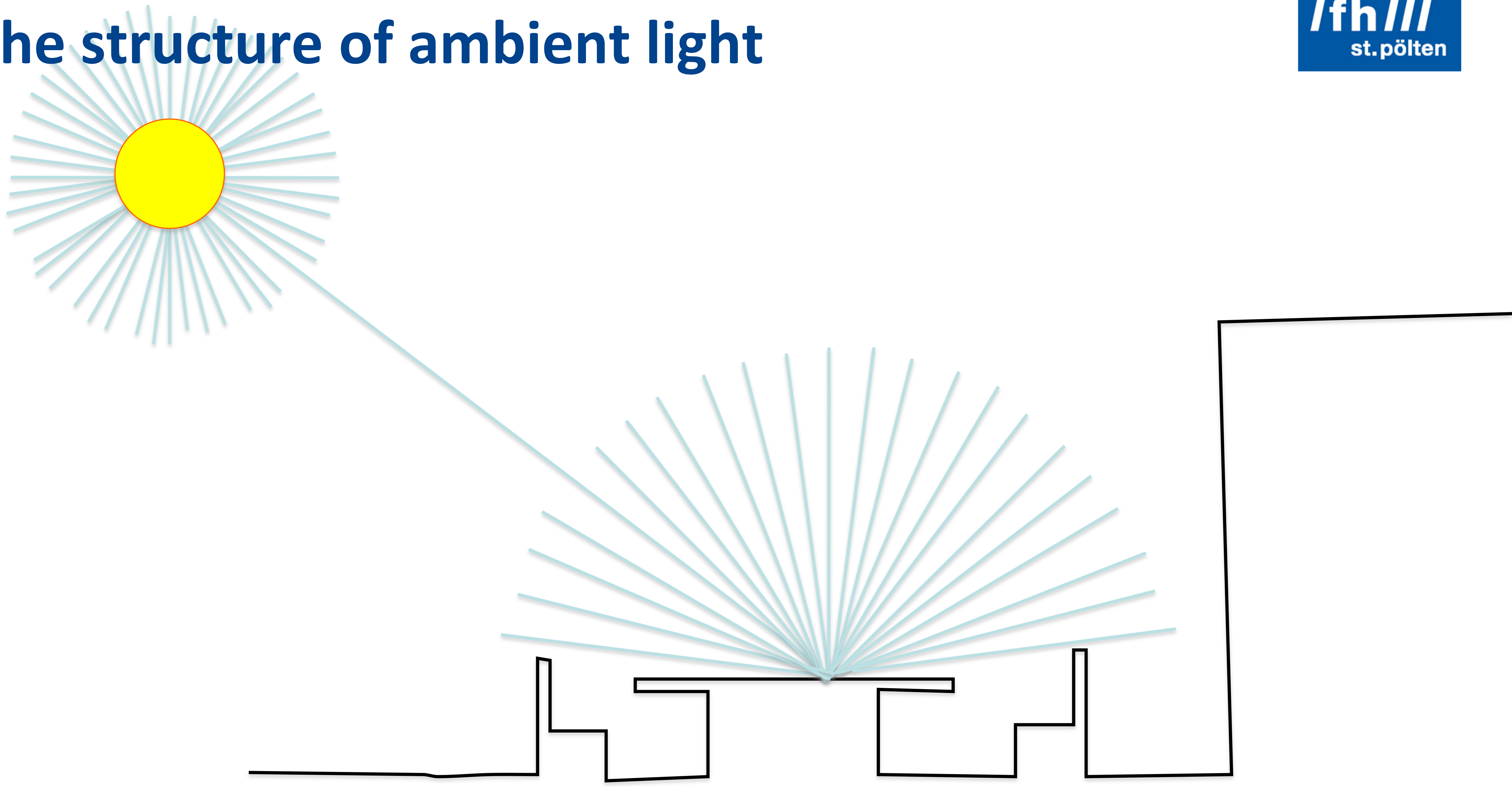
- The real world much more complex than what we can measure in images.
- Imaging is a mapping from 3D to 2D. We always loose information here!
- It is impossible to invert the imaging process, i.e. generate a complete 3D model out of an image.
- We can only find approximations for the original data

In this course, we will:

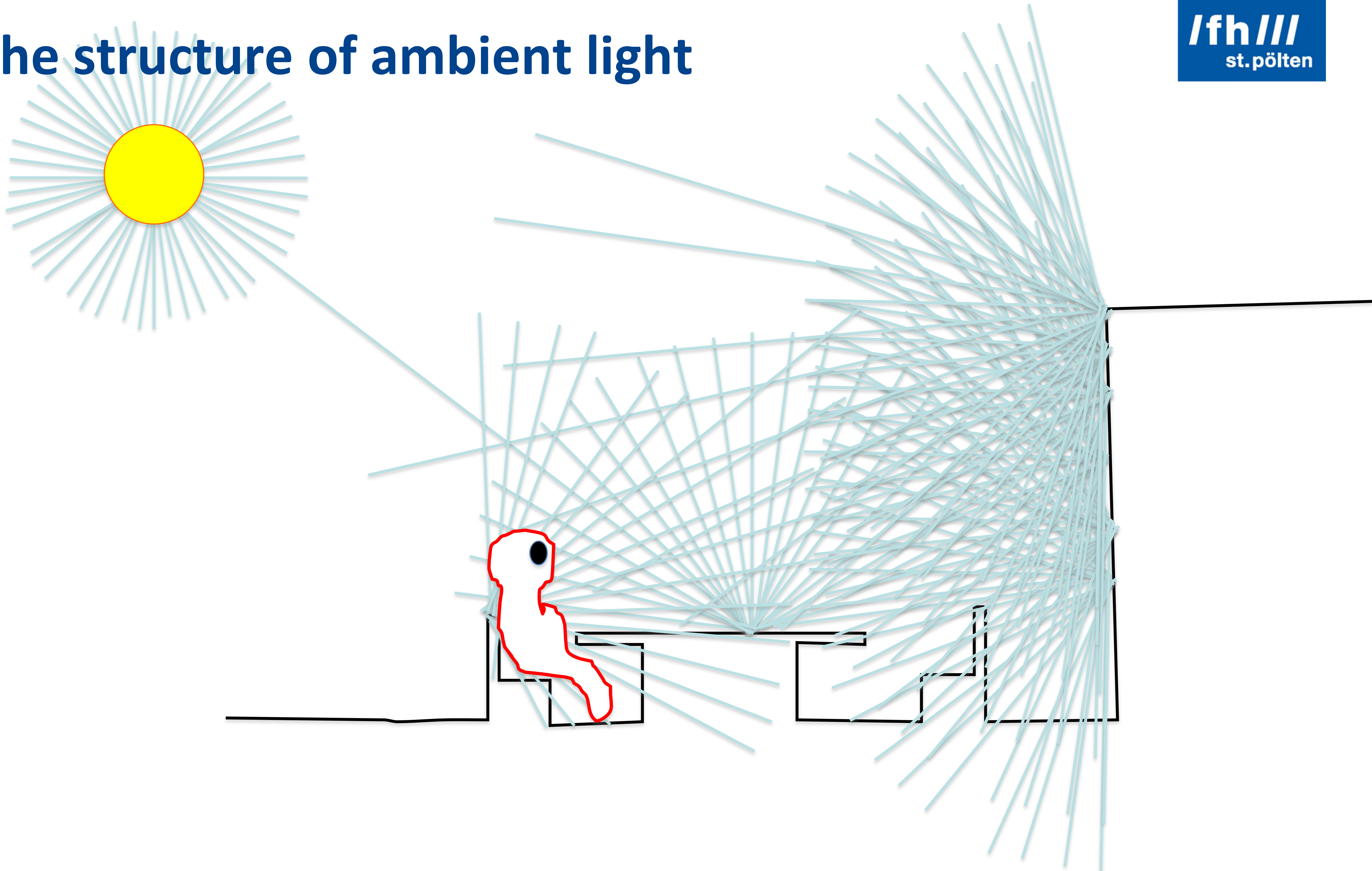
Take a few baby steps…

Image Formation: how are images generated?

The structure of ambient light

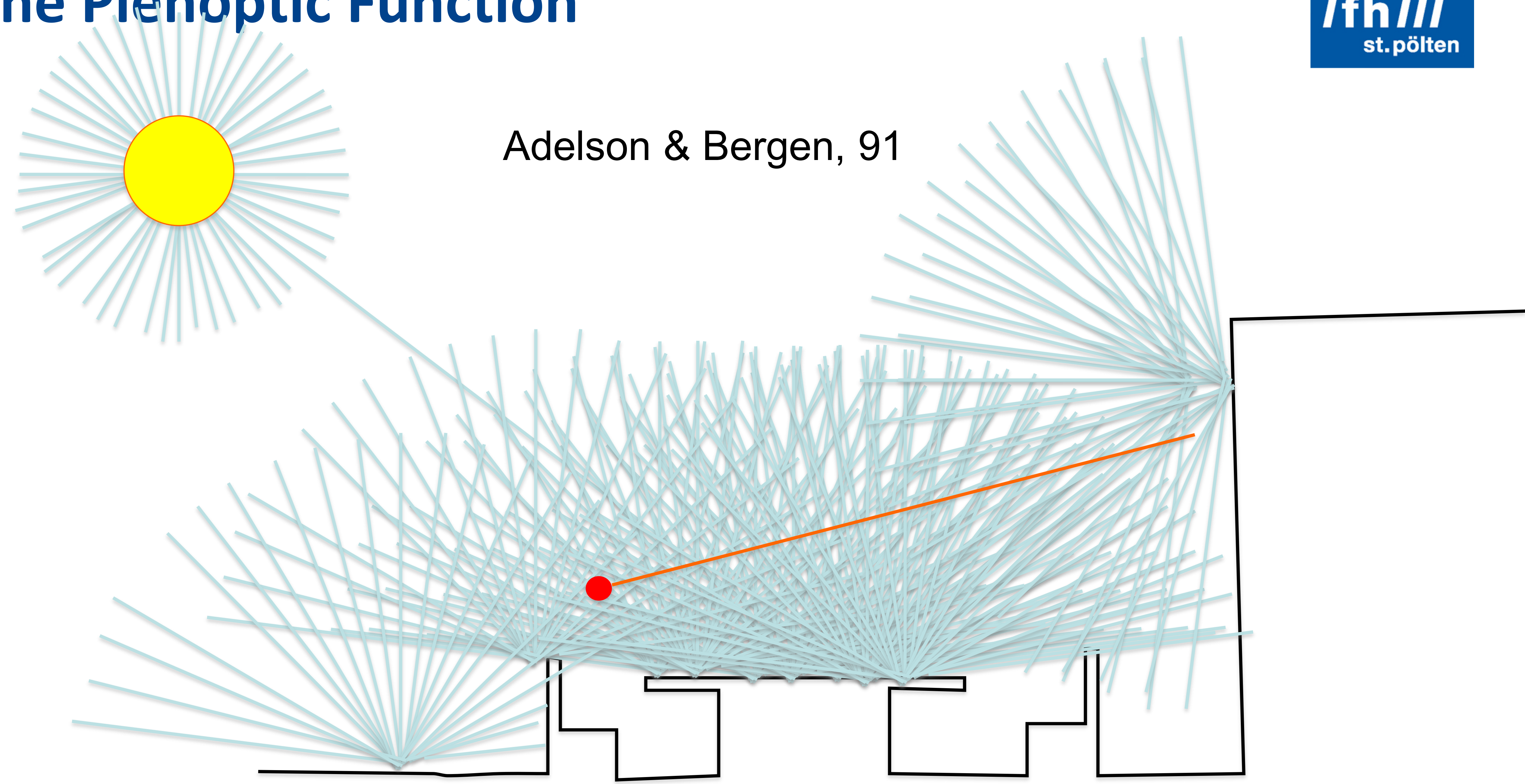


The structure of ambient light



The Plenoptic Function

Adelson & Bergen, 91



The intensity P can be parameterized as:

$$P\ (\theta, \phi, \lambda, t, X, Y, Z)$$

“The complete set of all convergence points constitutes the permanent possibilities of vision.” Gibson

Measuring the Plenoptic function

"The significance of the plenoptic function is this: The world is made of 3D objects, but these objects do not communicate their properties directly to an observer. Rather, the objects fill the space around them with the pattern of light rays that constitutes the plenoptic function, and the observer takes samples from this function." Adelson, & Bergen 91.

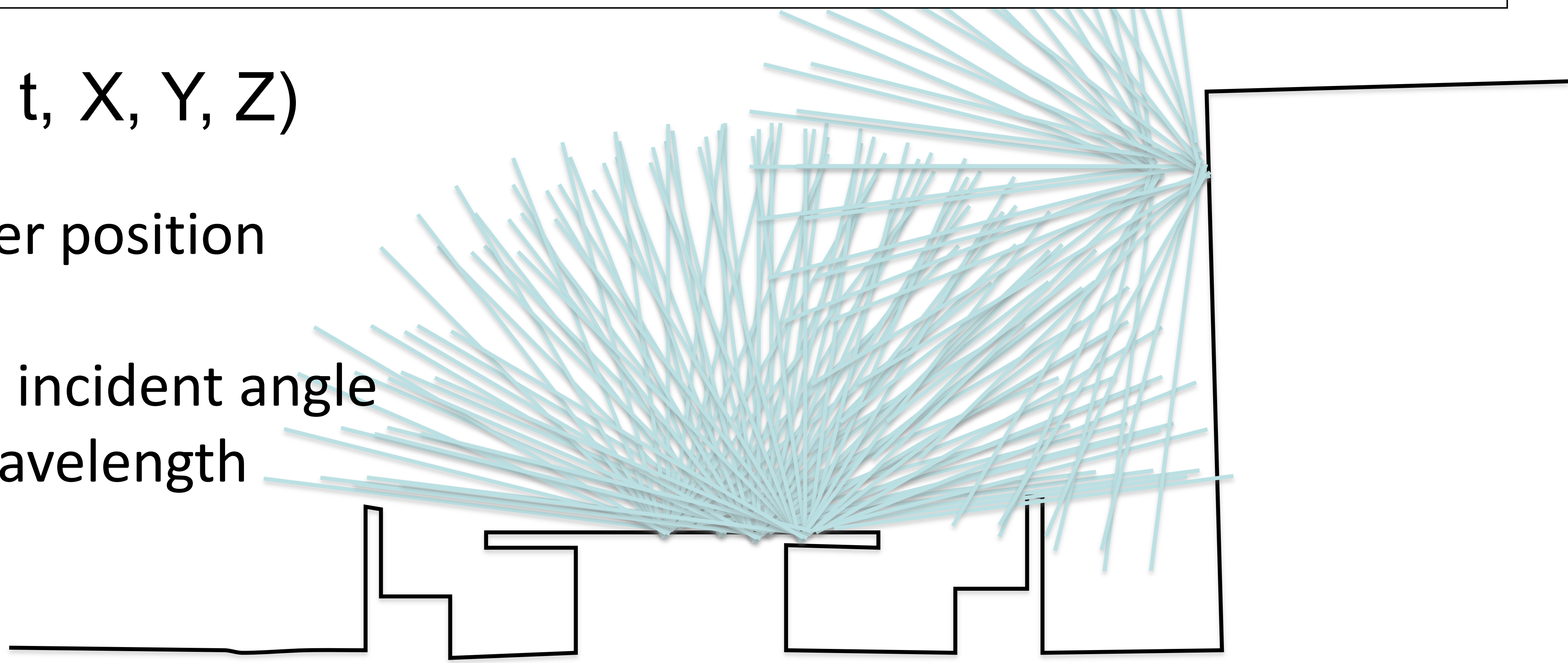
$P(\theta, \phi, \lambda, t, X, Y, Z)$

X,Y,Z: viewer position

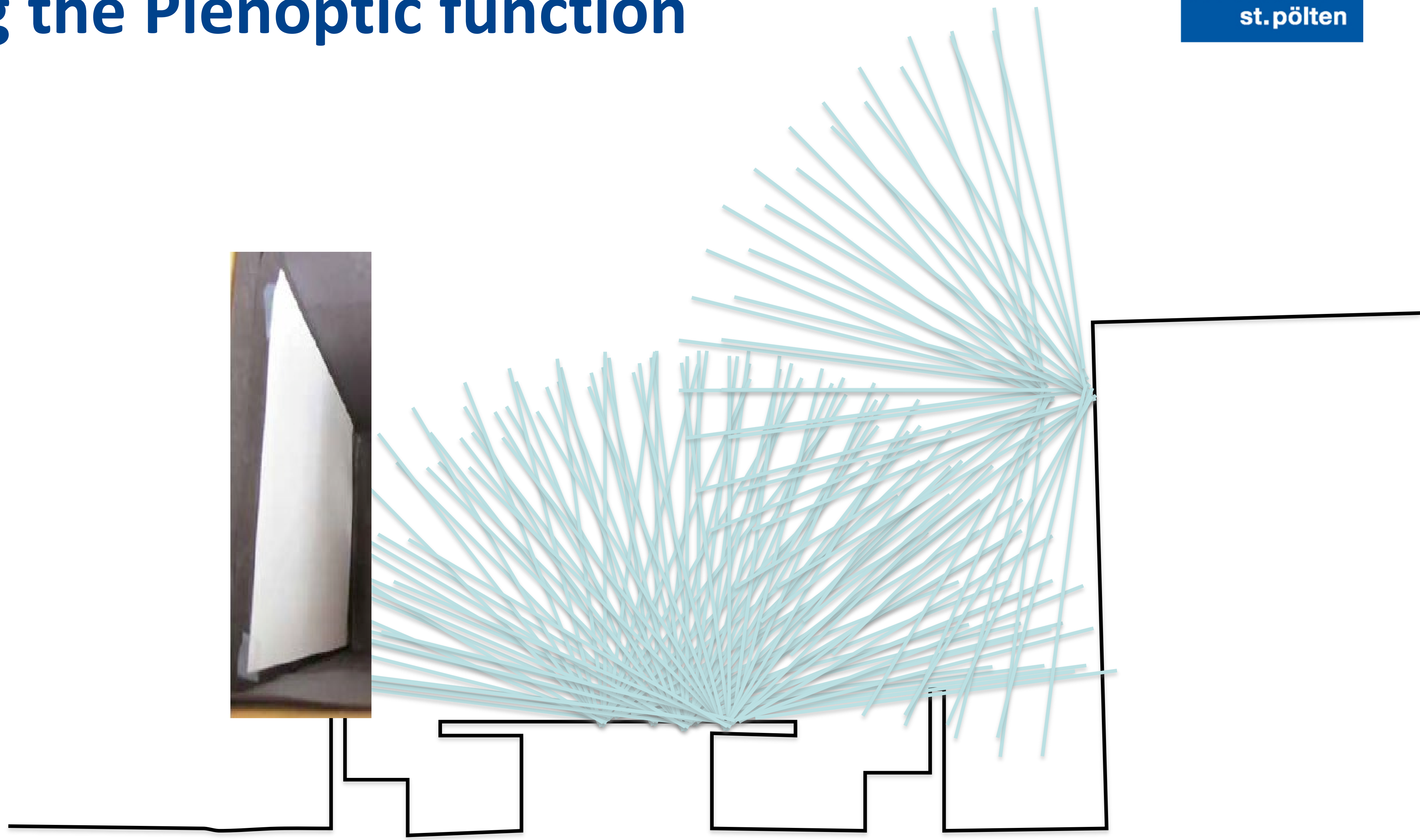
t: time

Phi, Theta: incident angle

lambda: wavelength

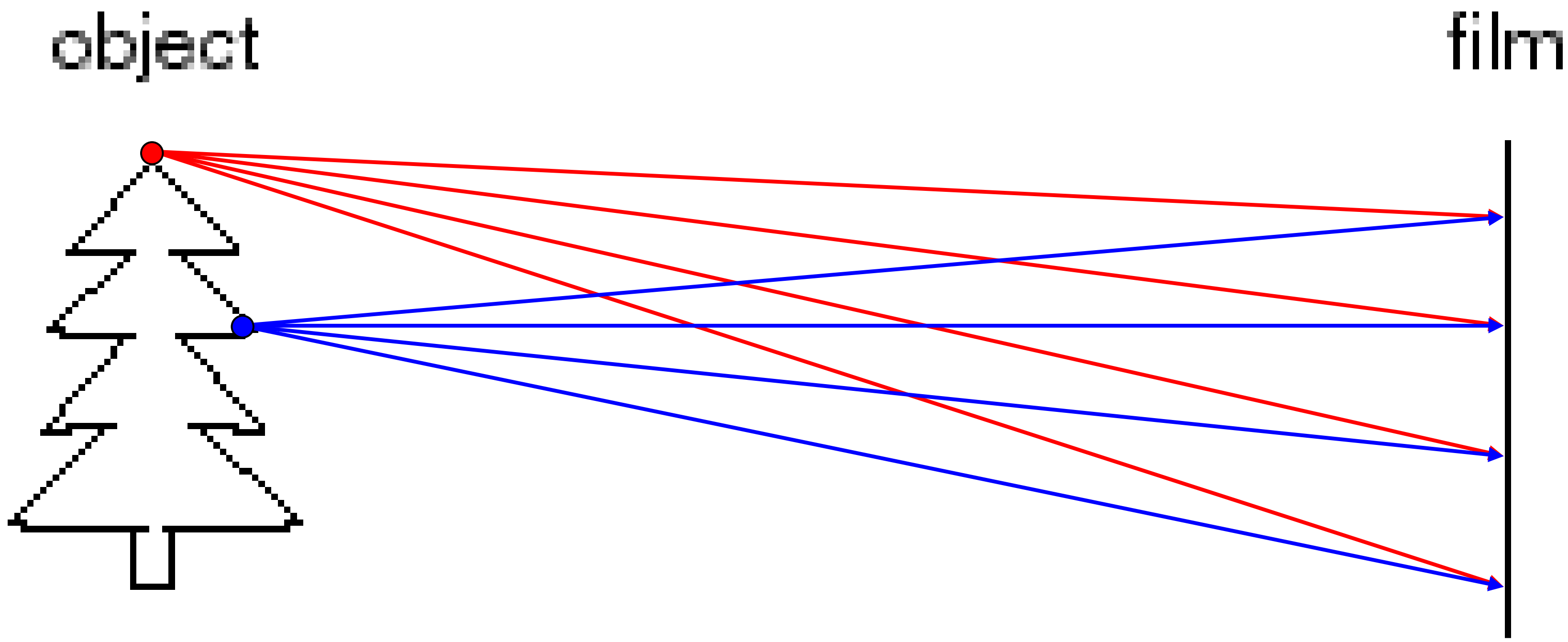


Measuring the Plenoptic function



Why is there no picture appearing on the paper (film)?

Image formation

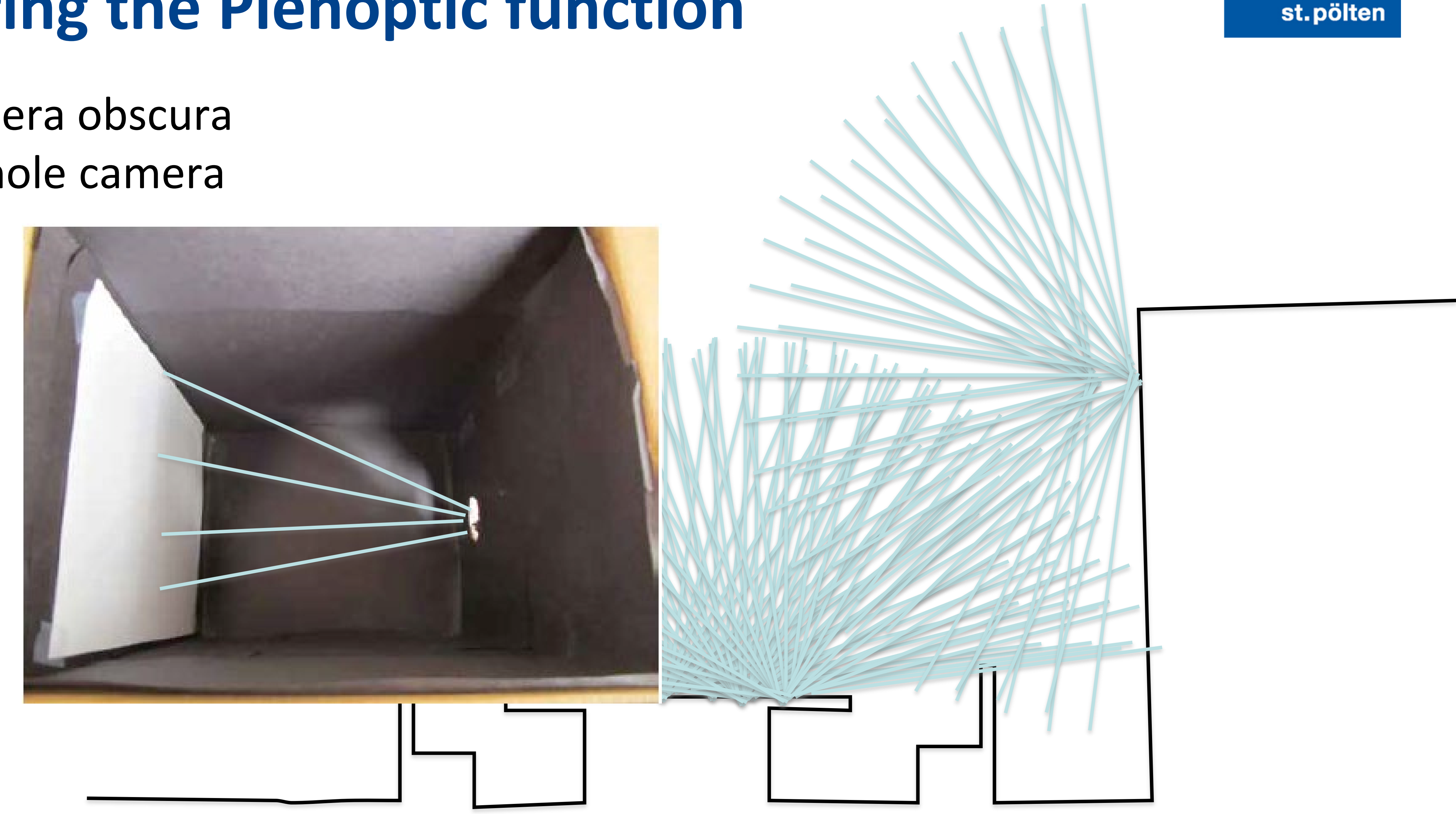


- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

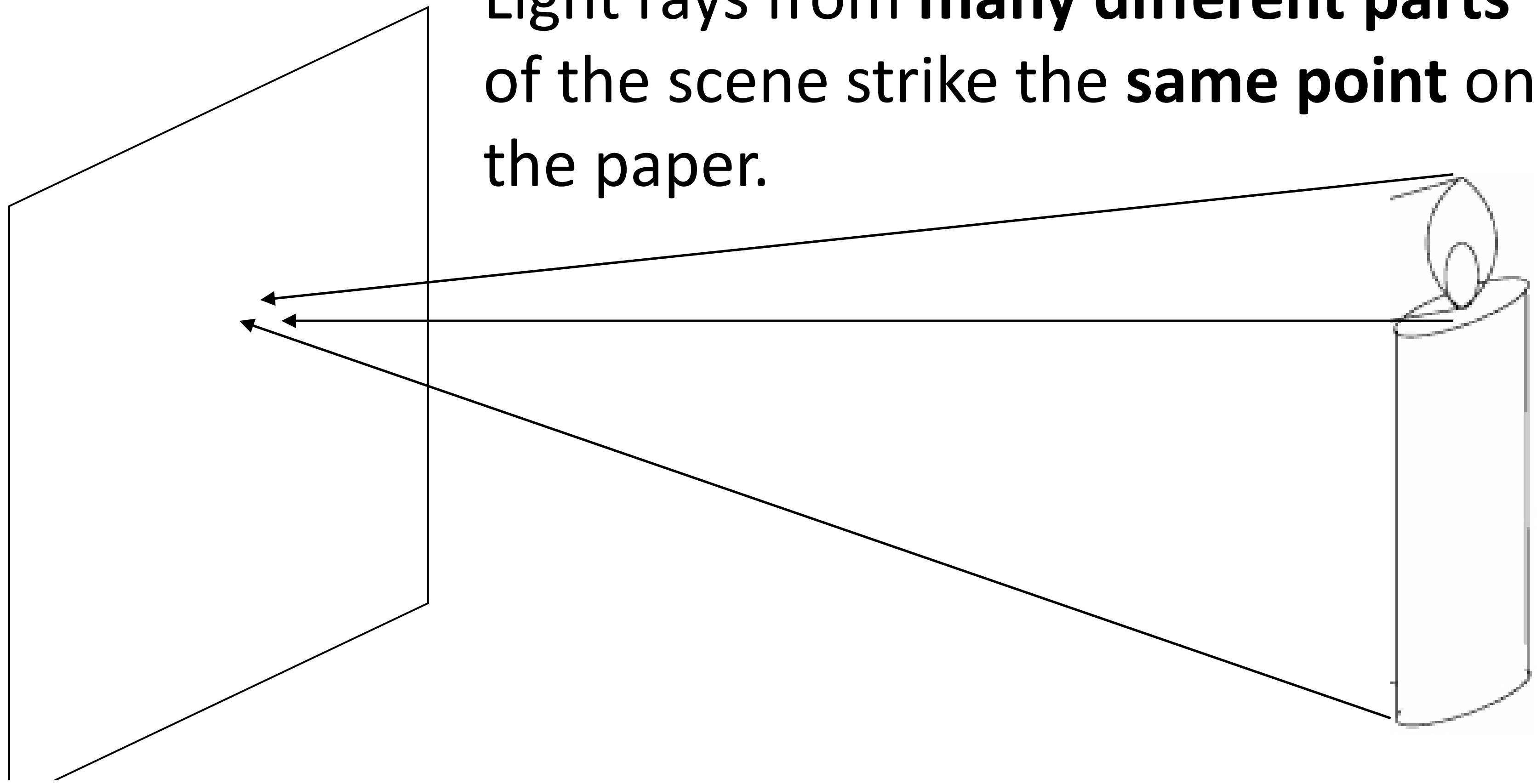
Slide by Steve Seitz

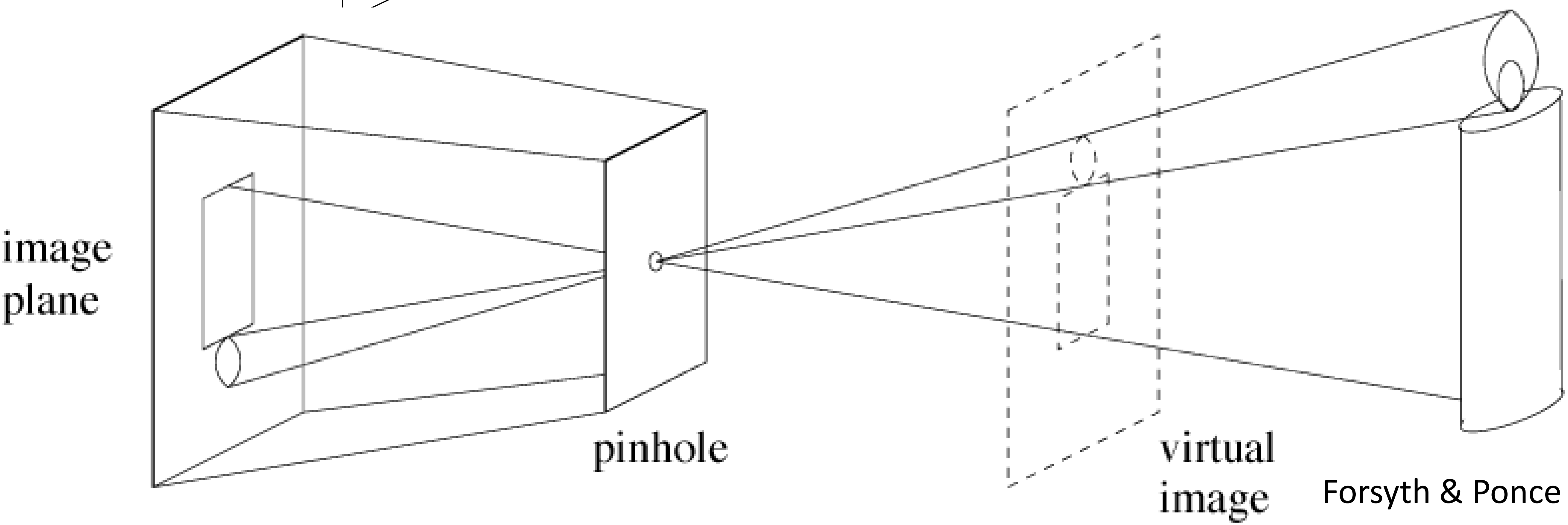
Measuring the Plenoptic function

- The camera obscura
  The pinhole camera



Light rays from **many different parts** of the scene strike the **same point** on the paper.

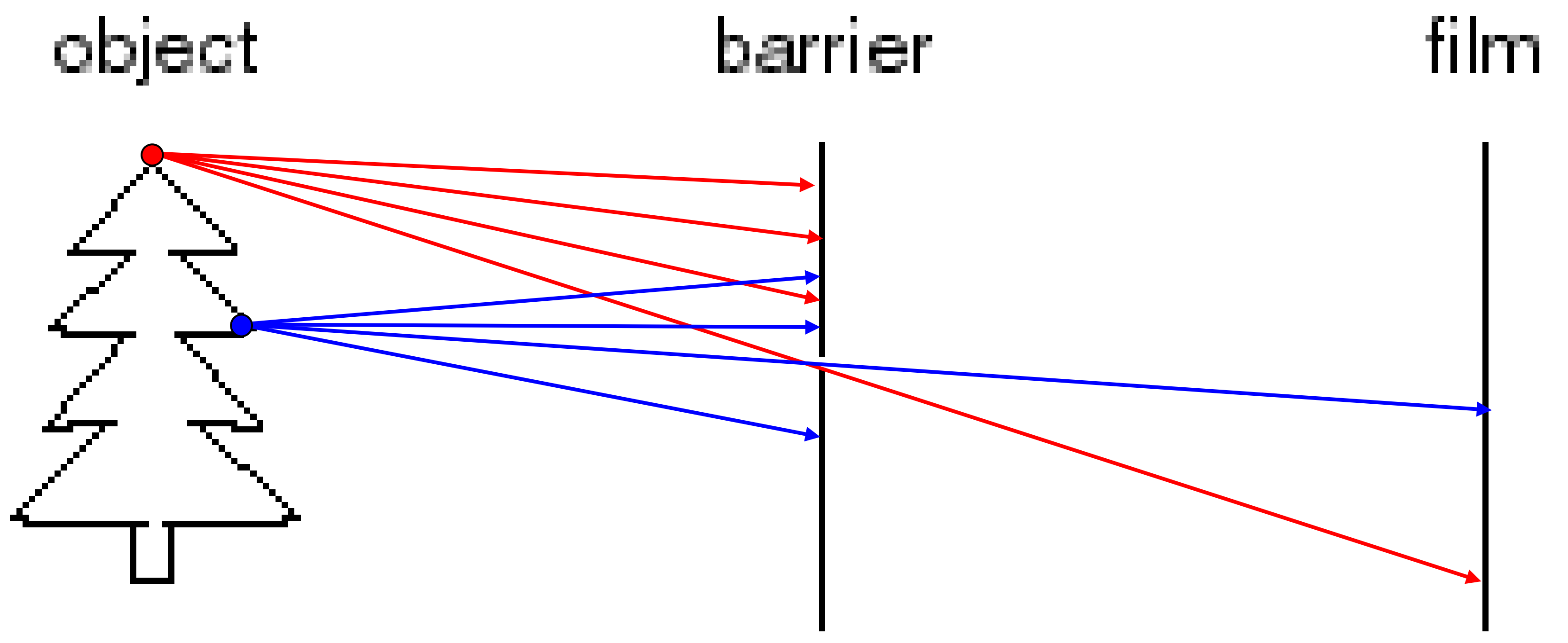




Forsyth & Ponce

The pinhole camera only allows rays from **one point** in the scene **to strike each point** of the paper.

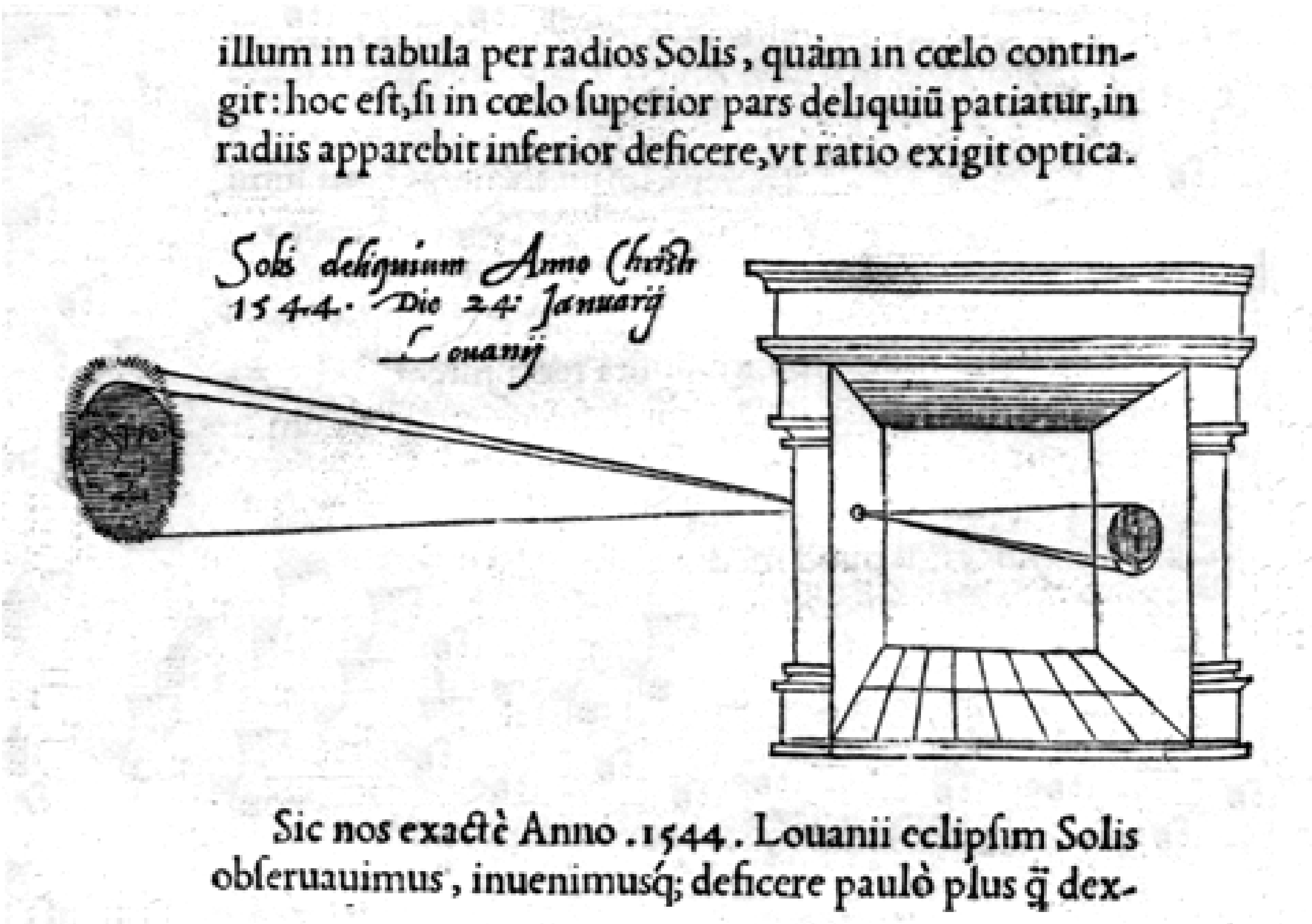
Pinhole camera



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

Slide by Steve Seitz

Camera obscura



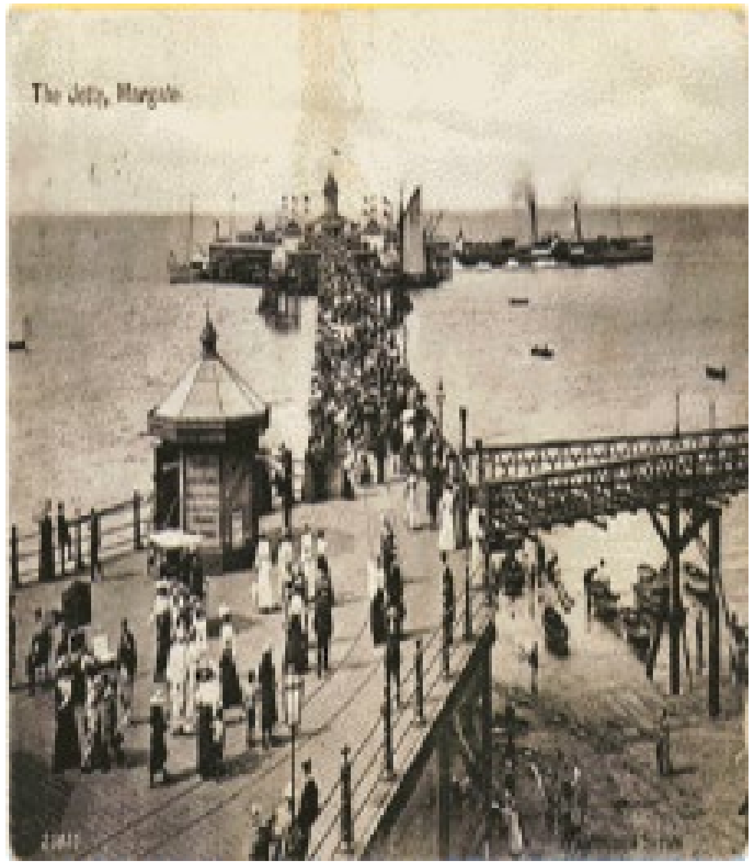
In Latin, means ‘dark room’

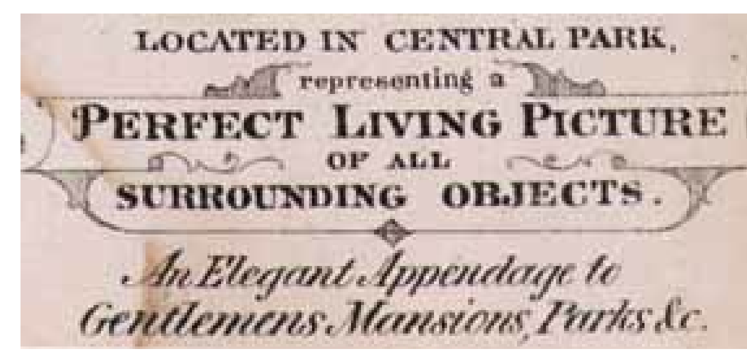
"**Reinerus Gemma-Frisius**, observed an eclipse of the sun at Louvain on January 24, 1544, and later he used this illustration of the event in his book De Radio Astronomica et Geometrica, 1545. It is thought to be the first published illustration of a camera obscura..."

Hammond, John H., The Camera Obscura, A Chronicle

http://www.acmi.net.au/AIC/CAMERA_OBSCURA.html

Camera obscura





Jetty at Margate England, 1898.

An attraction in the late 19th century

Around 1870s

http://brightbytes.com/cosite/collection2.html

Adapted from R. Duraiswami

Camera obscura at home

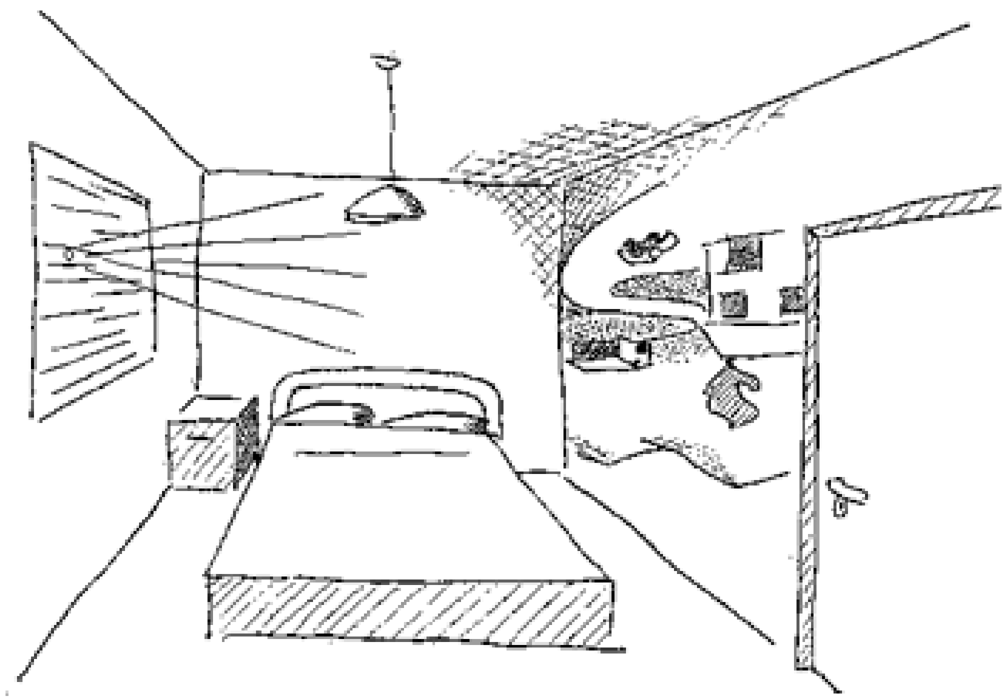
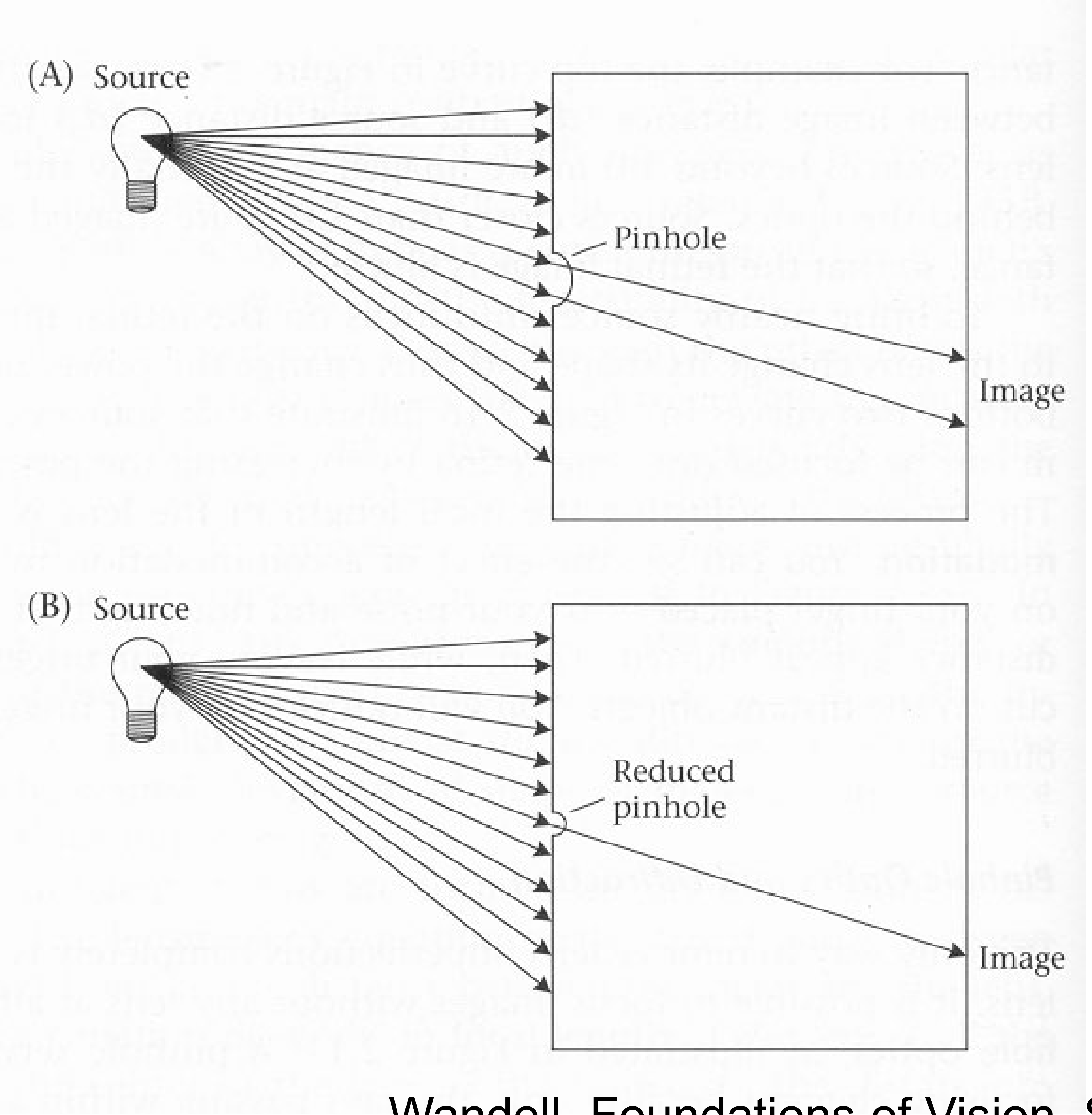


Figure 1 - A lens on the window creates the image of the external world on the opposite wall and you can see it every morning, when you wake up.

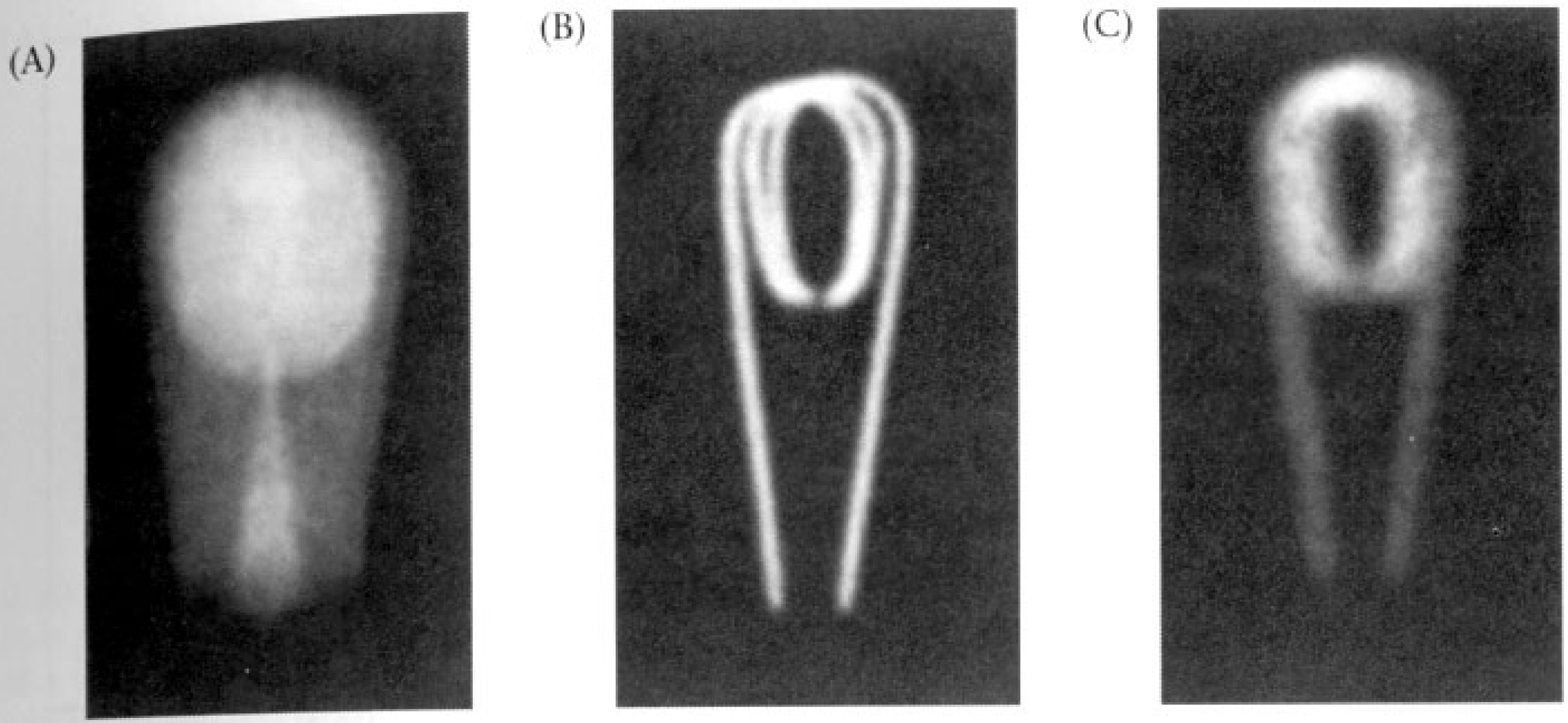
Sketch from http://www.funsci.com/fun3_en/sky/sky.htm

https://learn.zoner.com/experiment-making-a-room-sized-camera-obscura/

Effect of pinhole size



Wandell, Foundations of Vision, Sinauer, 1995



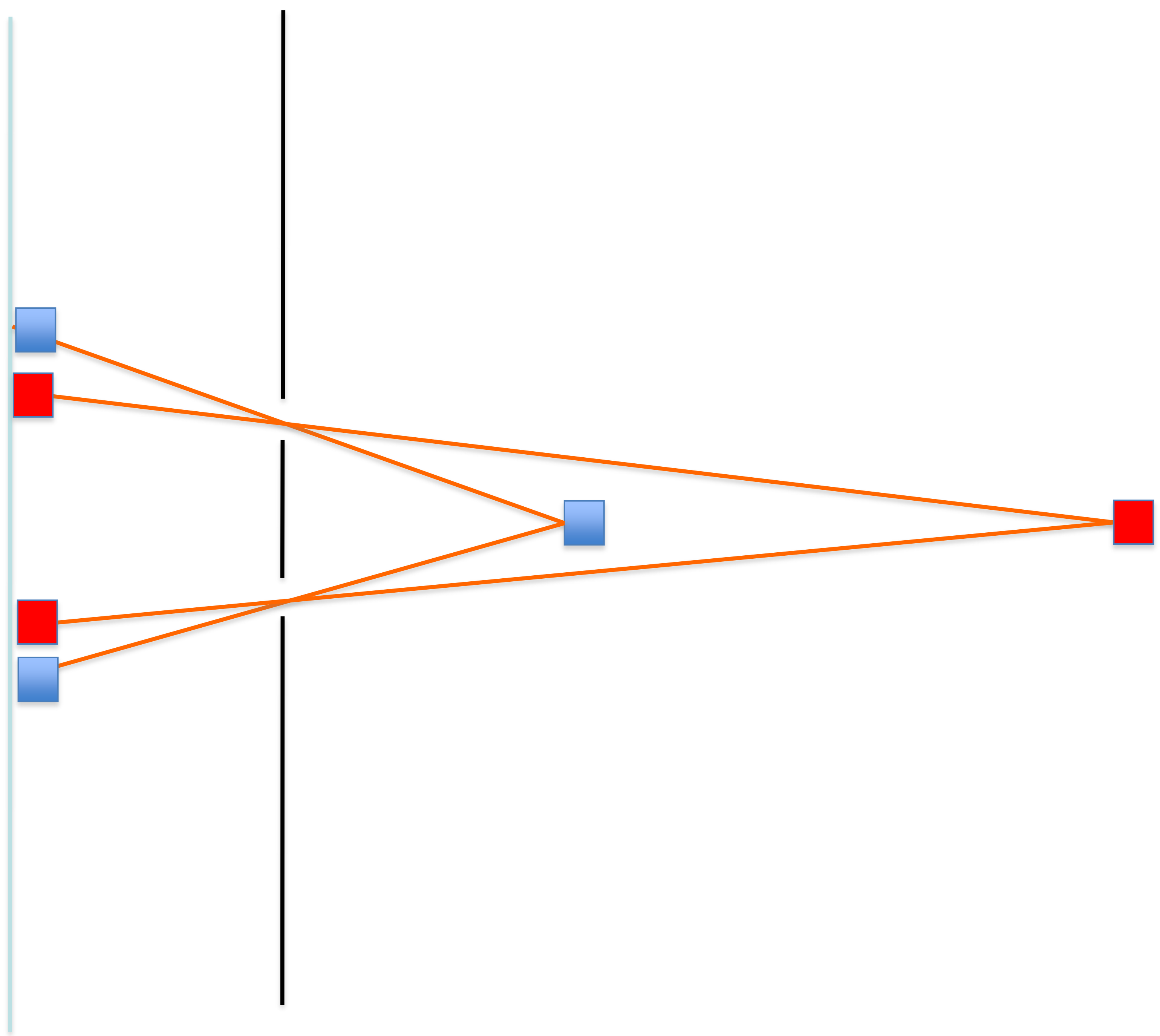
2.18 DIFFRACTION LIMITS THE QUALITY OF PINHOLE OPTICS. These three images of a bulb filament were made using pinholes with decreasing size. (A) When the pinhole is relatively large, the image rays are not properly converged, and the image is blurred. (B) Reducing the size of the pinhole improves the focus. (C) Reducing the size of the pinhole further worsens the focus, due to diffraction. From Ruechardt, 1958.

Wandell, Foundations of Vision, Sinauer, 1995

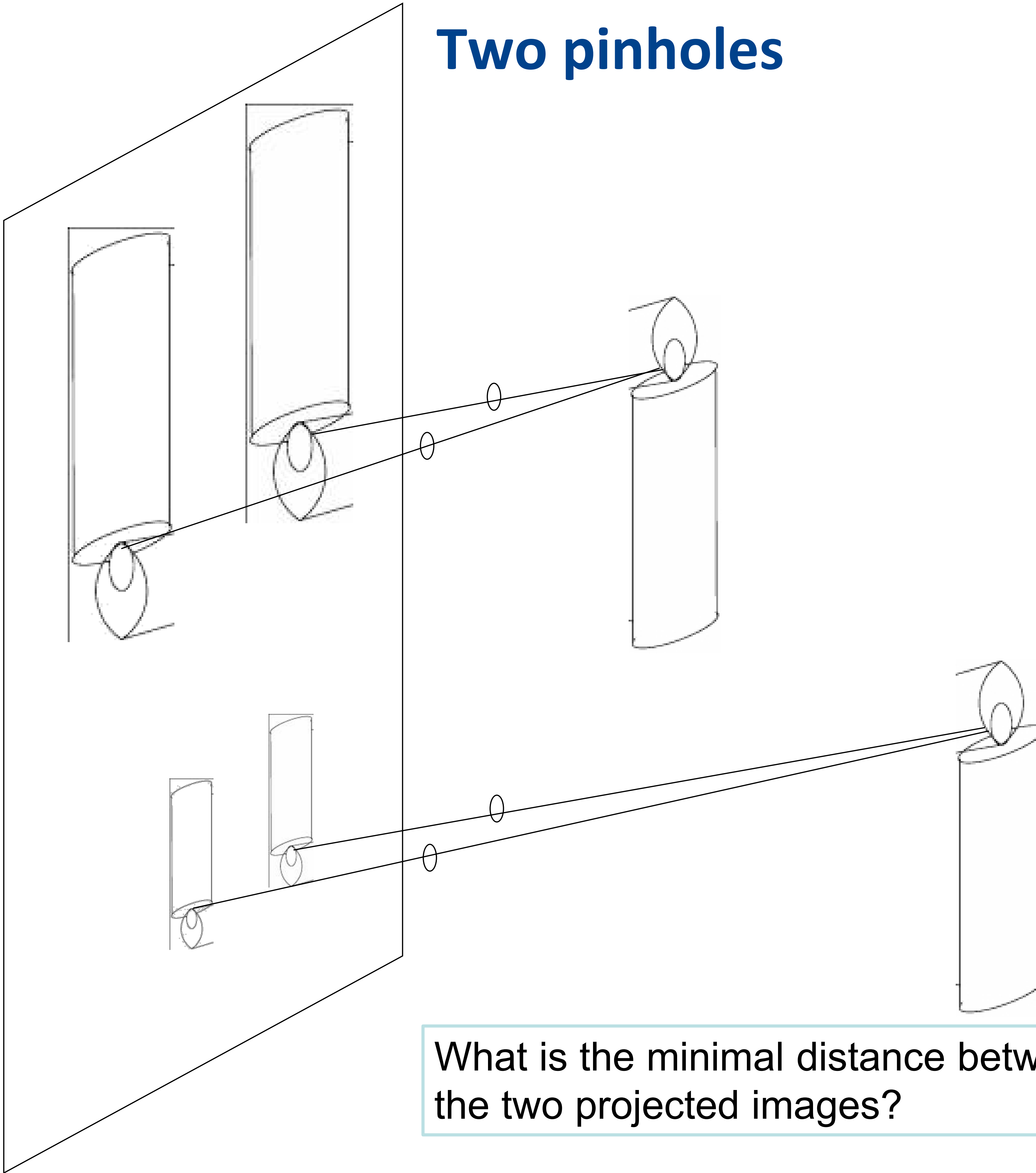
Playing with pinholes

e.g., Matchbox Pinhole Camera:
https://www.youtube.com/watch?v=espROi9g2pg

Two pinholes

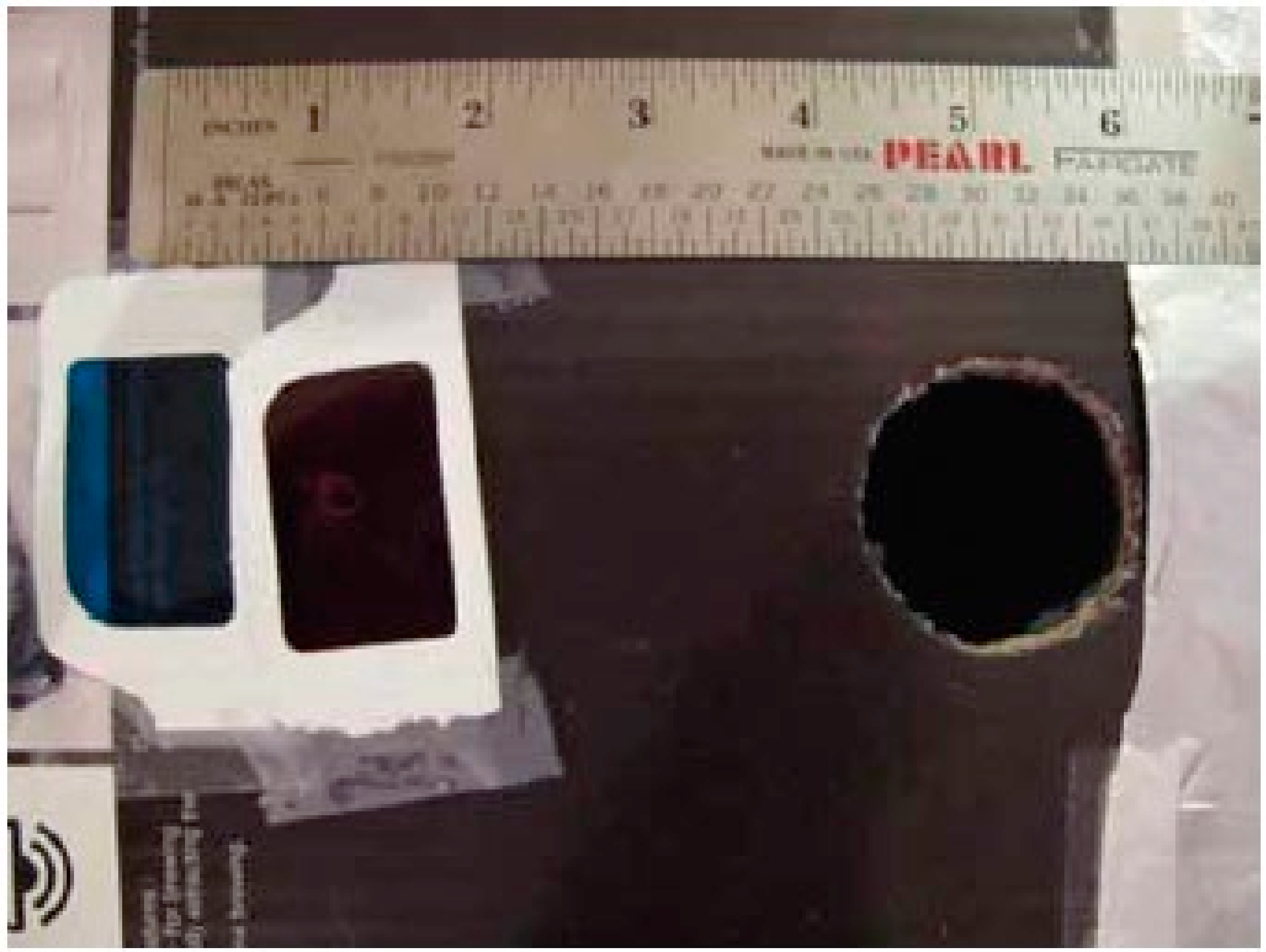


Two pinholes

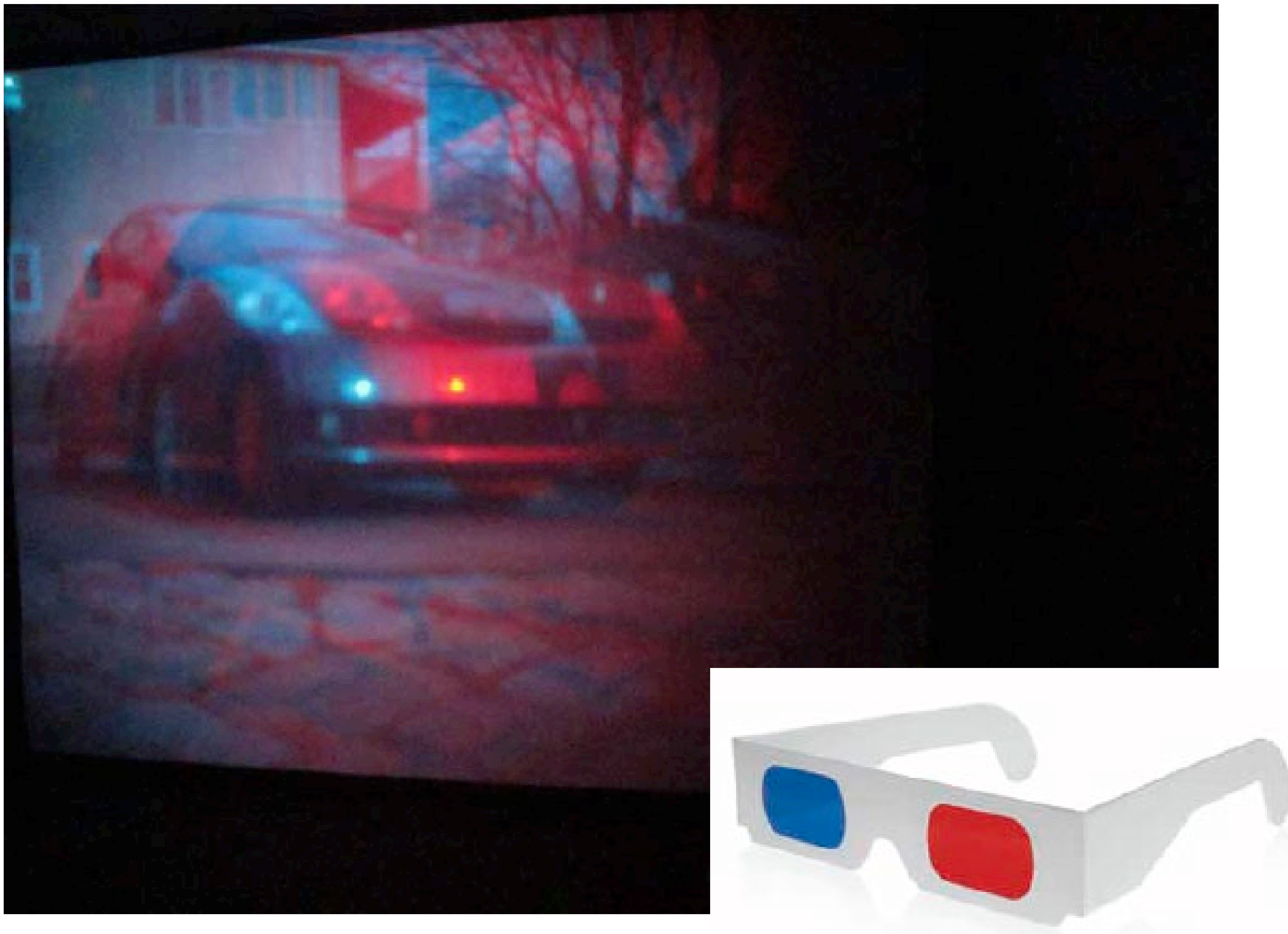


What is the minimal distance between the two projected images?

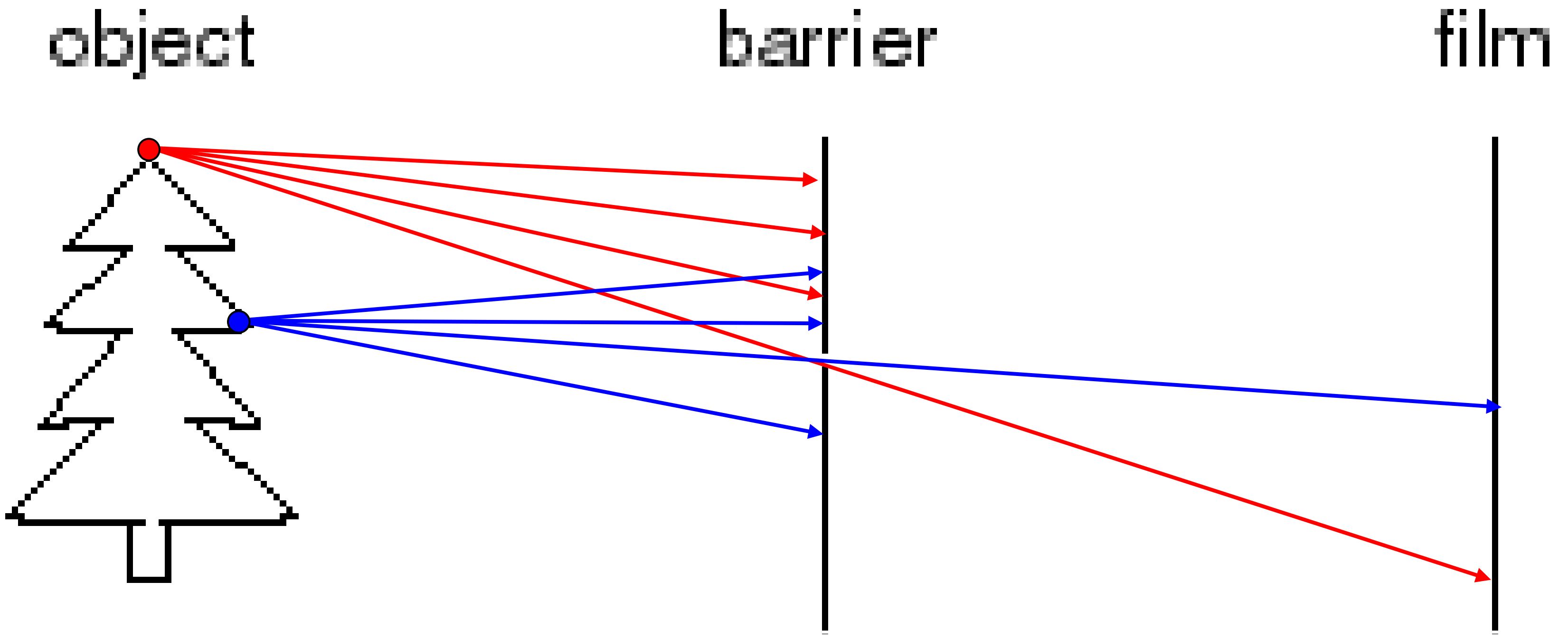
Anaglyph pinhole camera



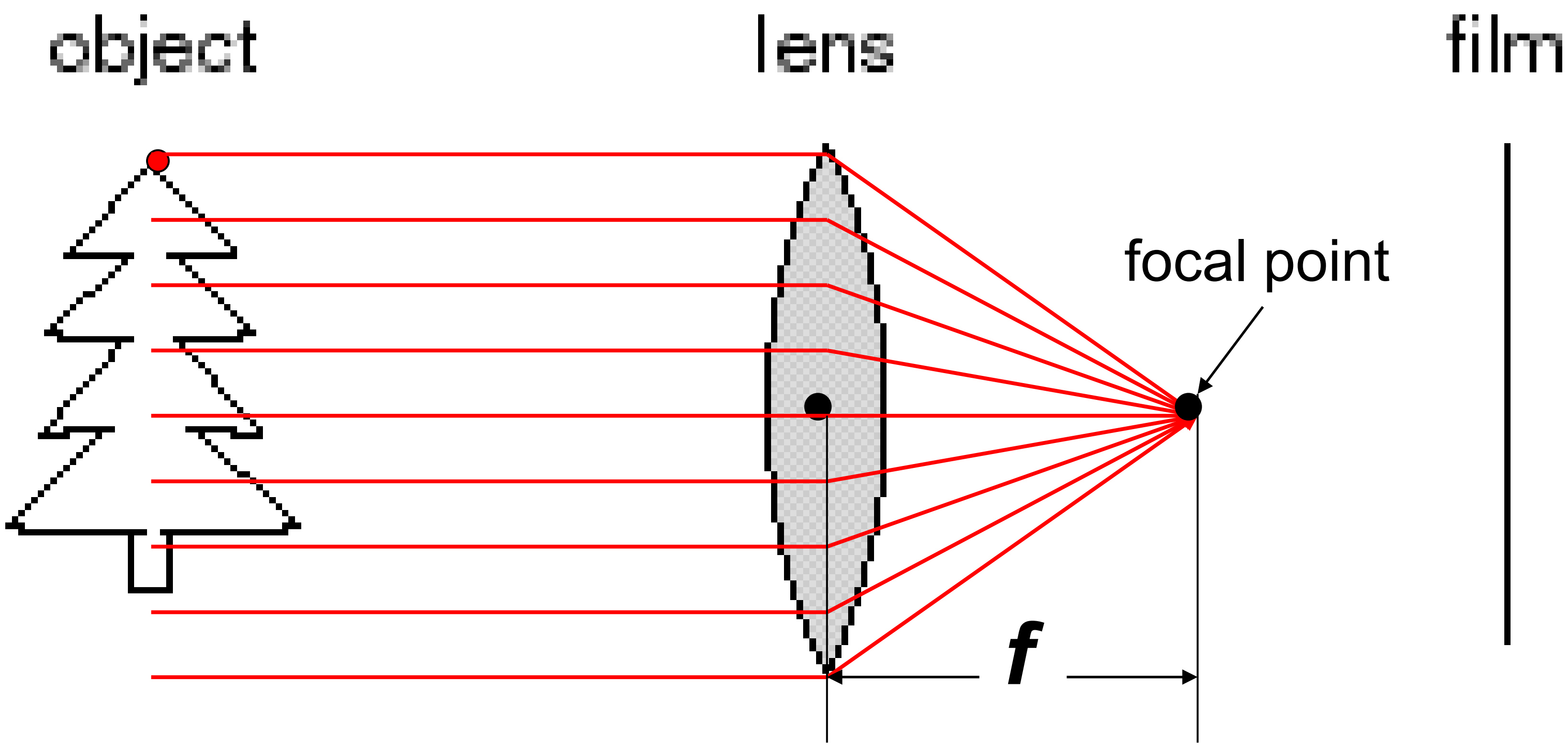
Anaglyph pinhole camera



From Pinhole to Camera…



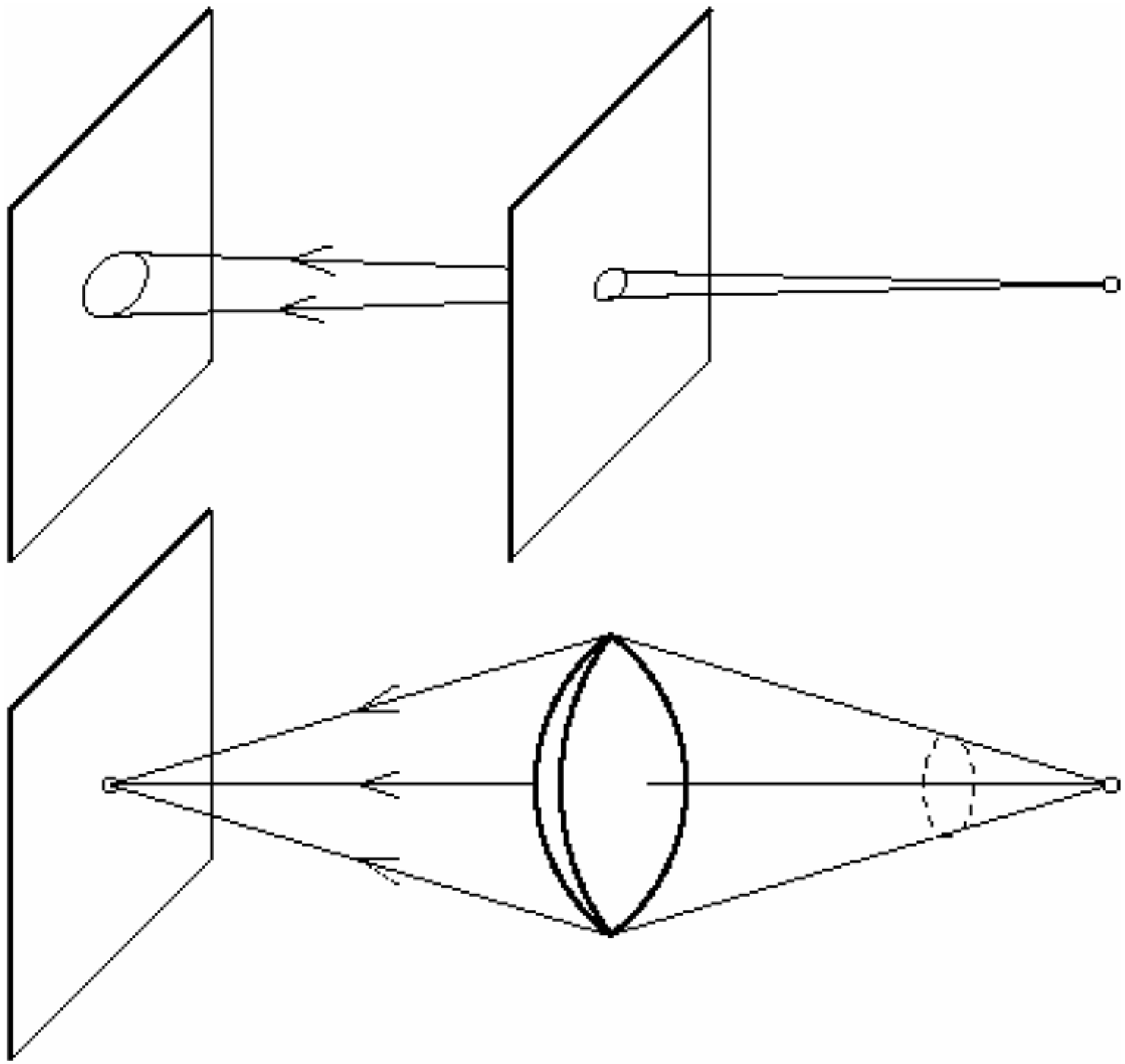
Adding a lens



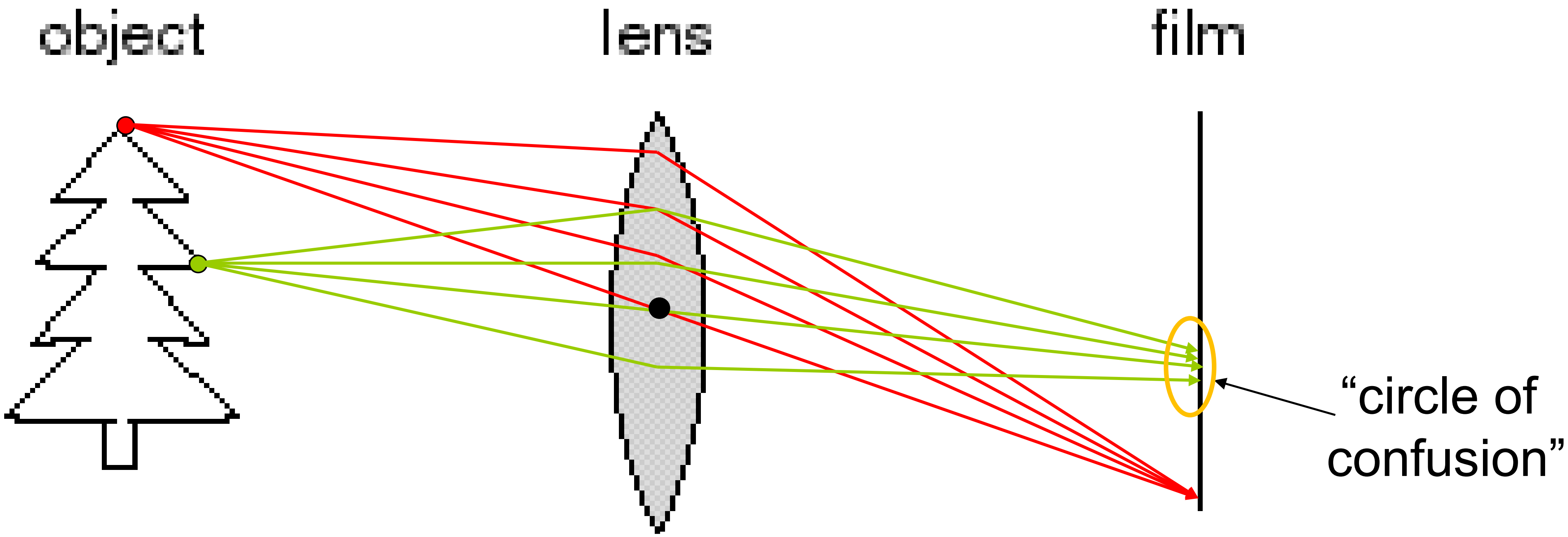
- A lens focuses light onto the film
  - Rays passing through the center are not deviated
  - All parallel rays converge to one point on a plane located at the *focal length f*
  - Lenses make pinhole model practical!

Slide by Steve Seitz

Pinhole vs. Lens



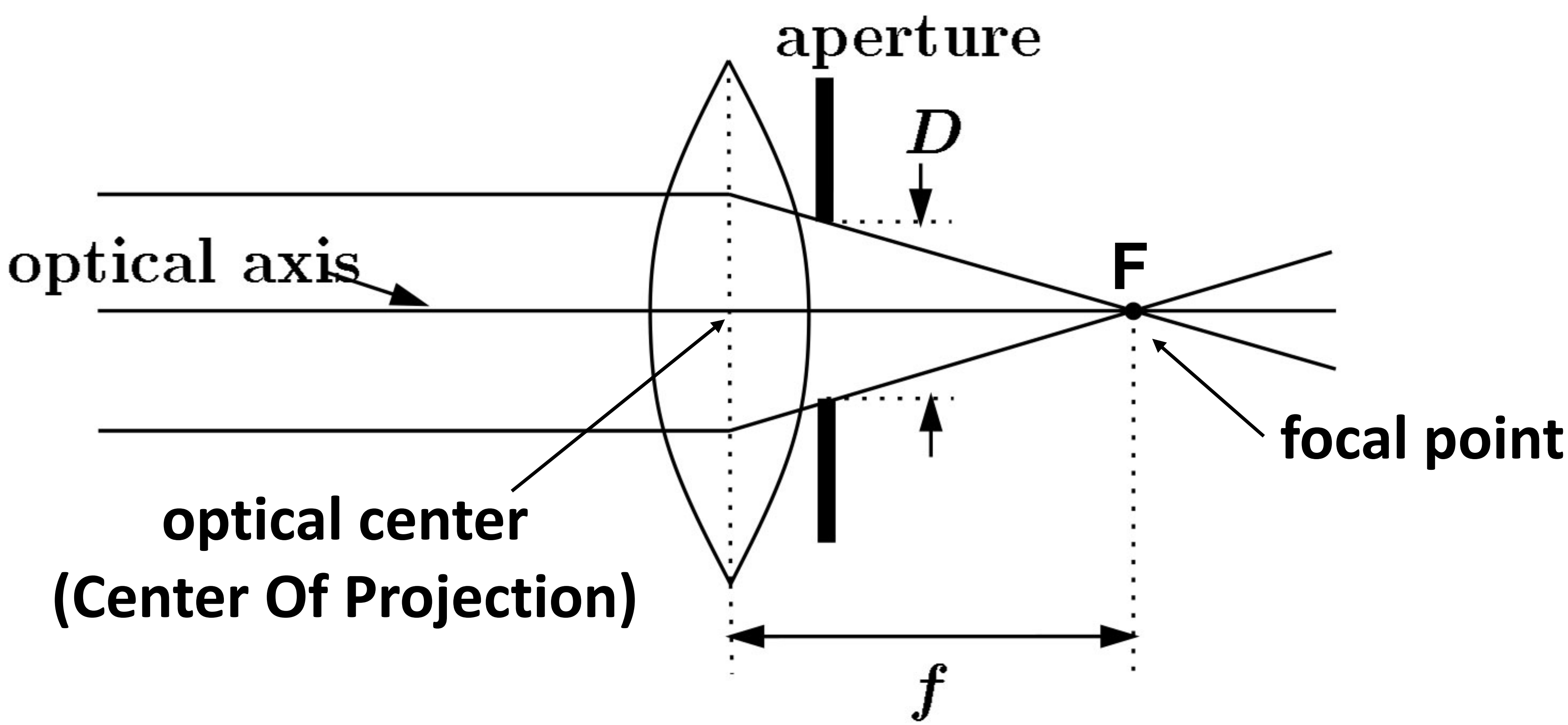
Adding a lens



- 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

Slide by James Hays

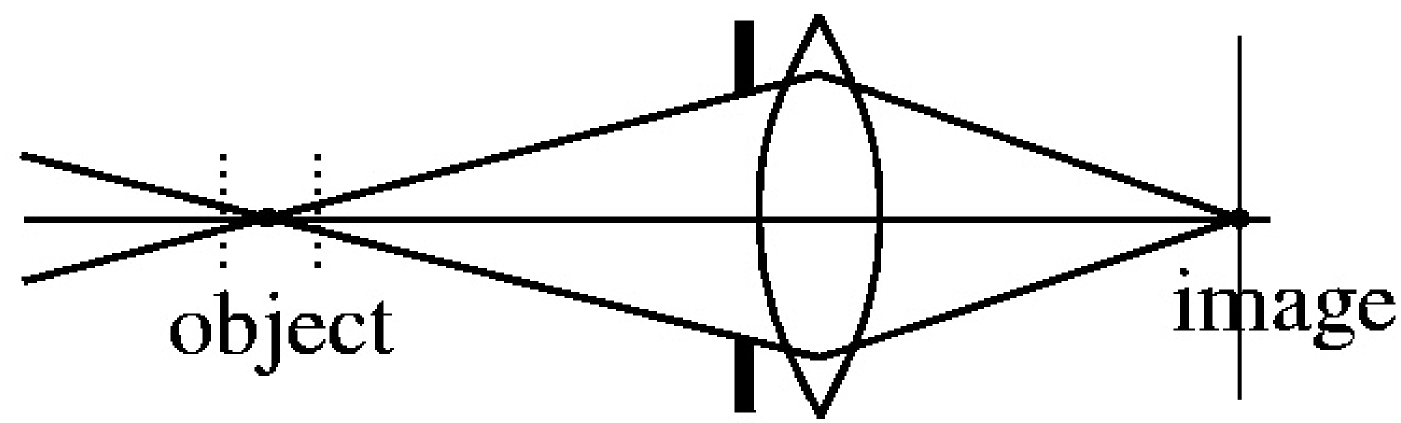
Cameras with lenses

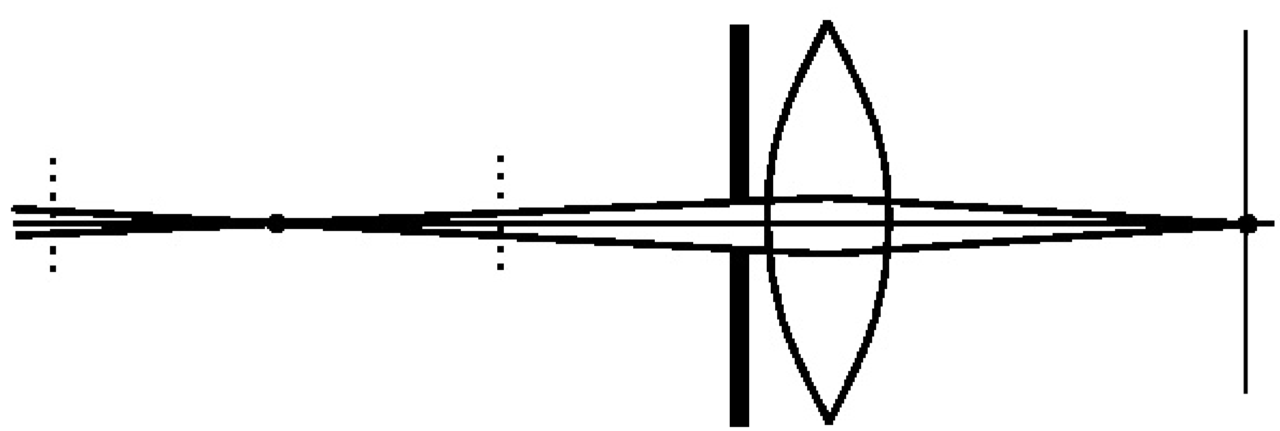


- A lens focuses parallel rays onto a single focal point
- Gather more light, while keeping focus;
- Make pinhole perspective projection practical

Focus and depth of field

- How does the aperture affect the depth of field?





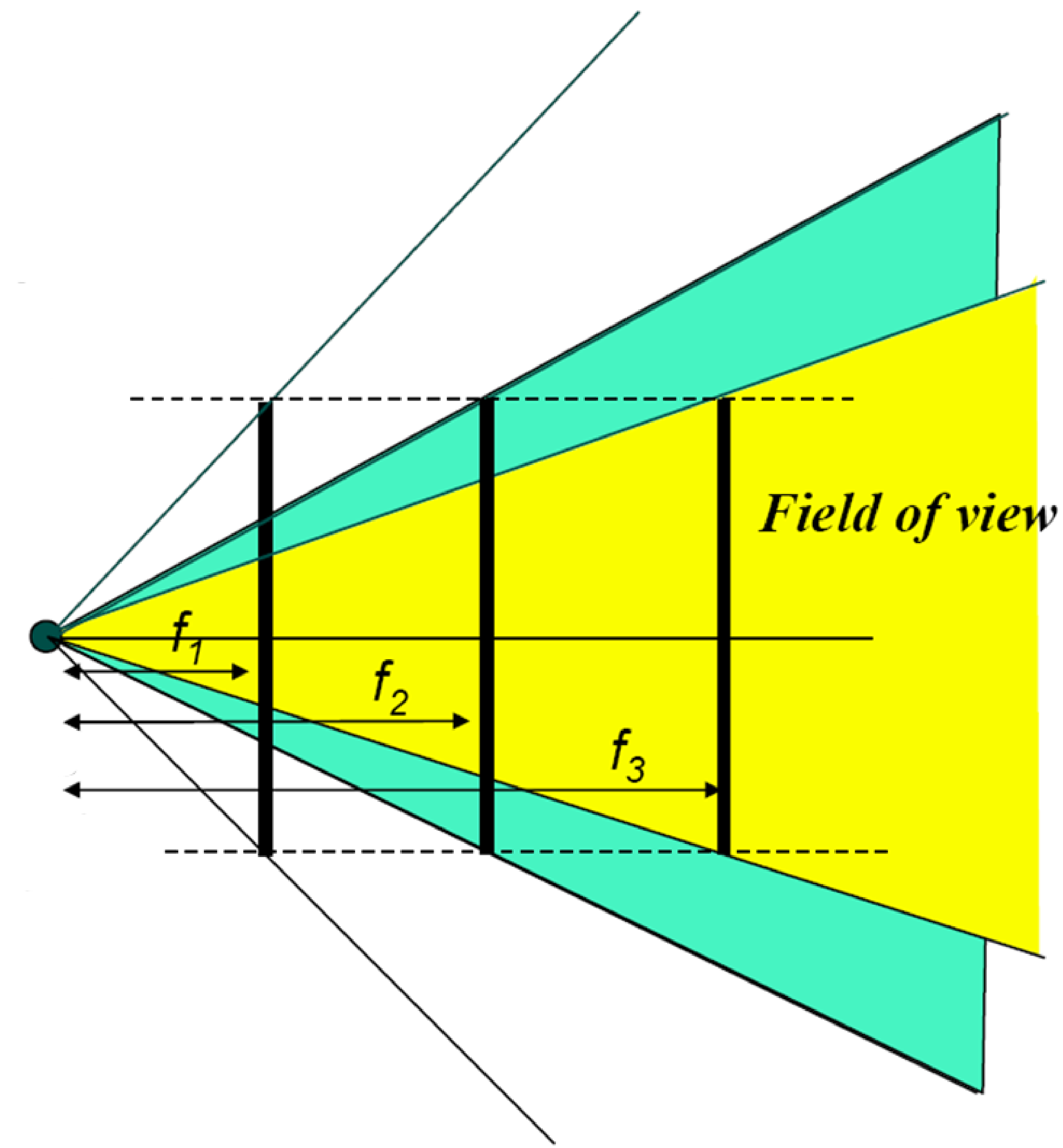
- A smaller aperture increases the range in which the object is approximately in focus

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

Slide from S. Seitz

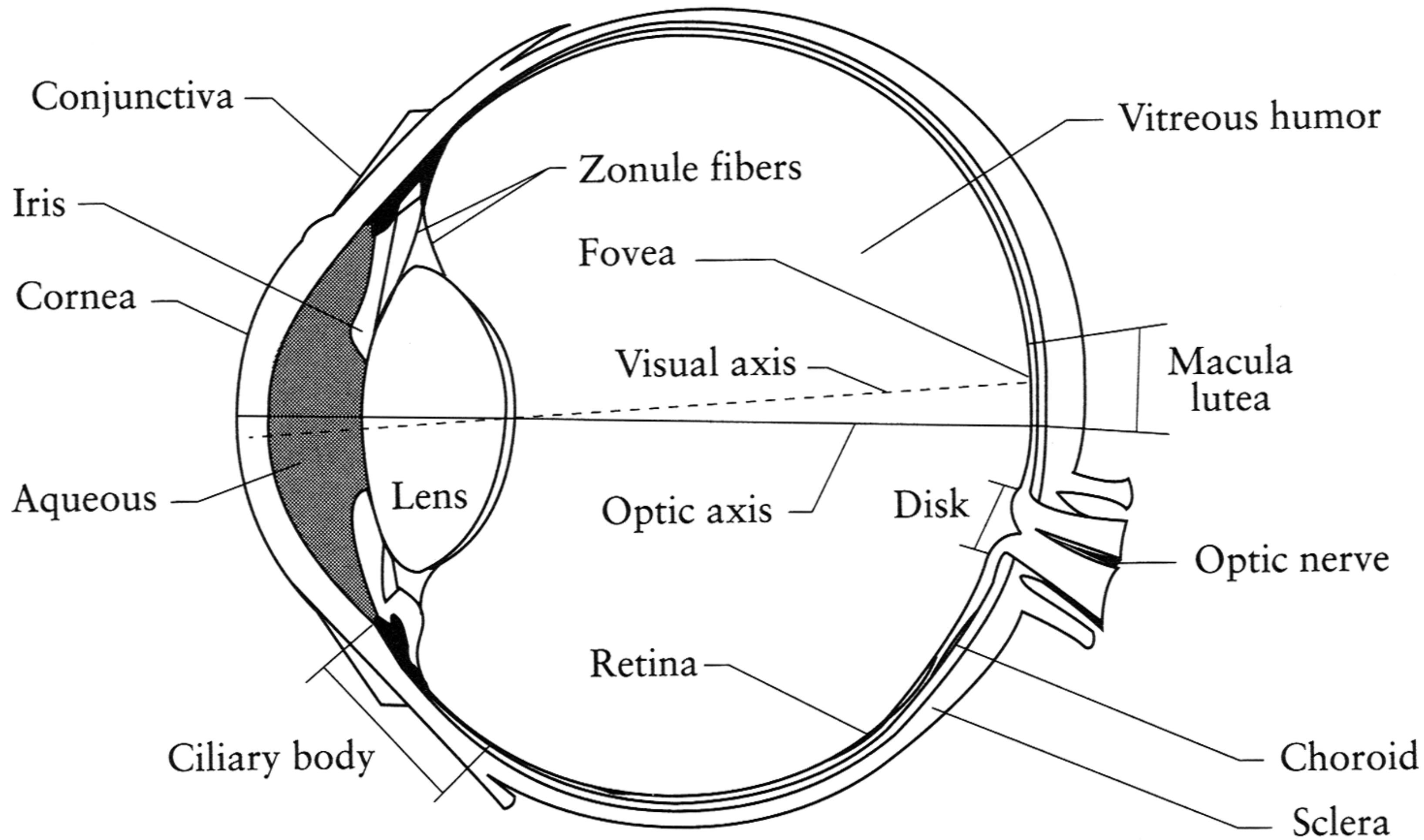
Field of view depends on focal length

- 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 *telescopic*
  - smaller part of the world projects onto the finite image plane



from R. Duraiswami

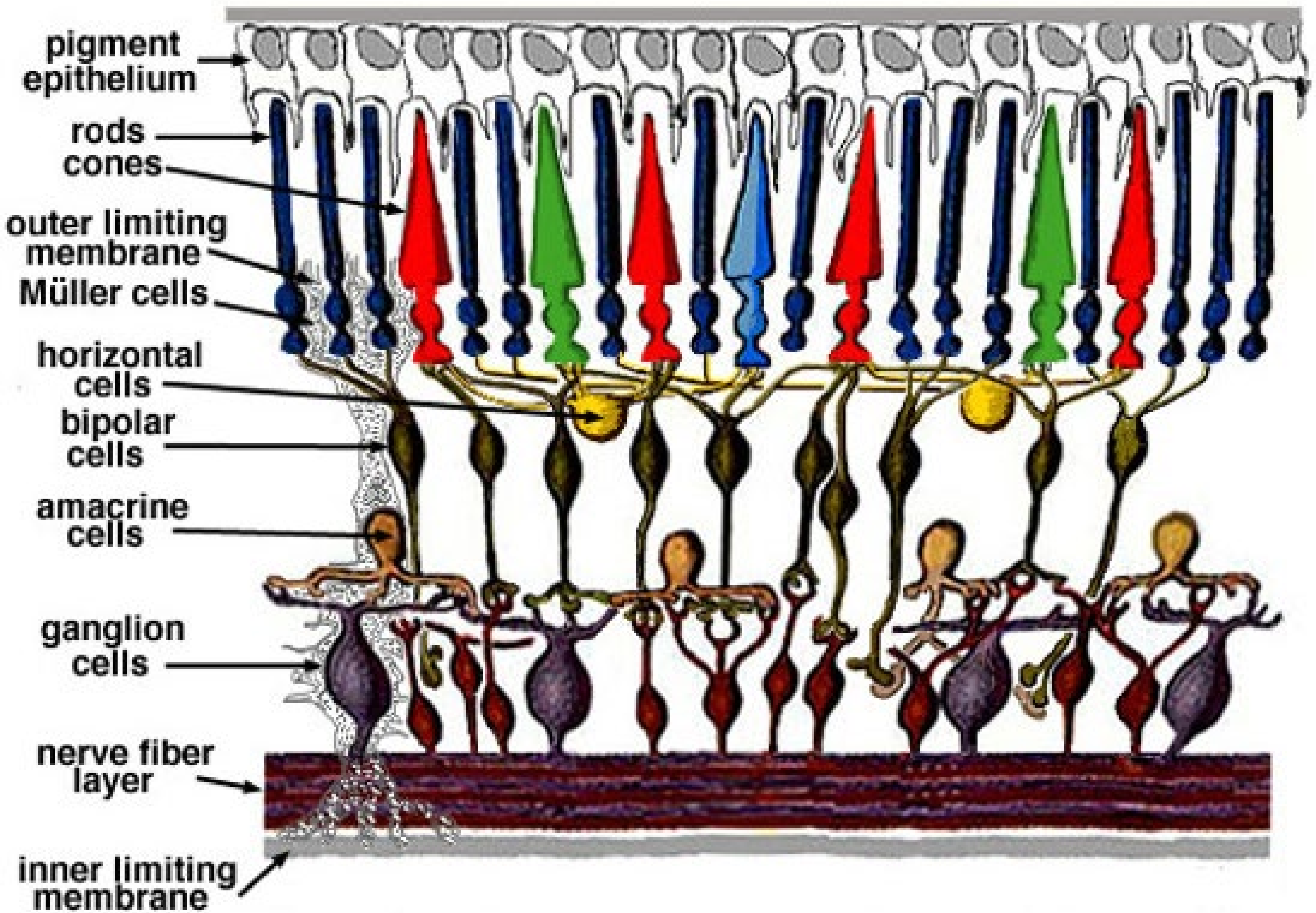
The Human Eye



- The human eye is a camera!
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    – photoreceptor cells (rods and cones) in the **retina**

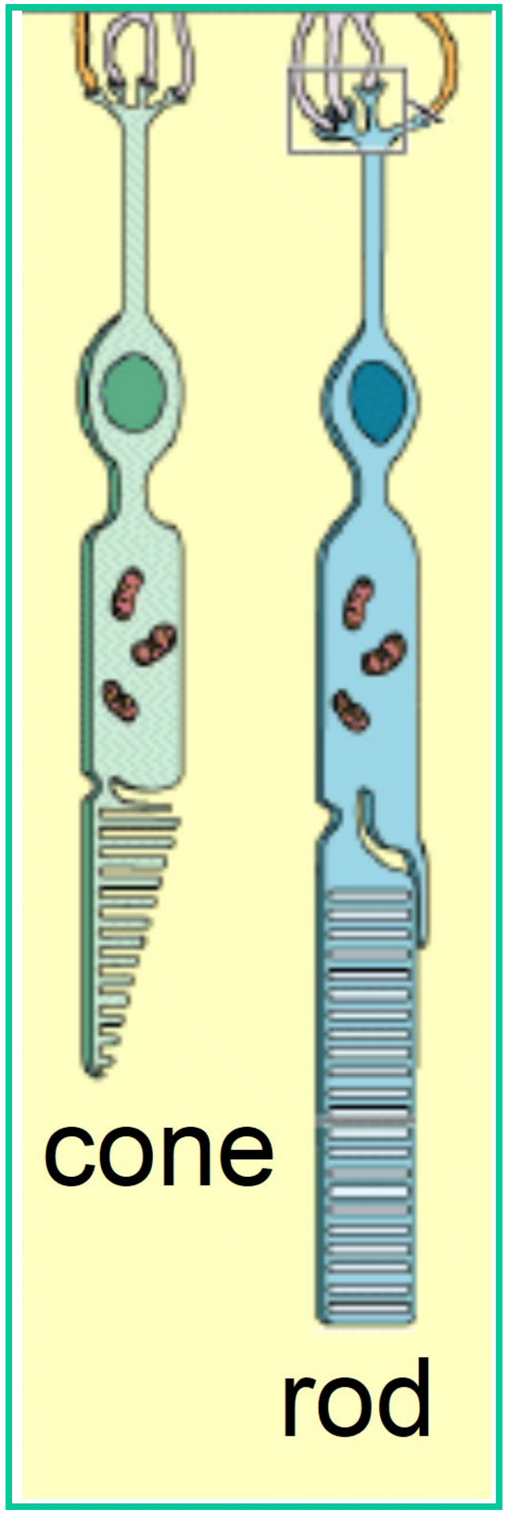
Slide by Steve Seitz

Our Sensor: The Retina

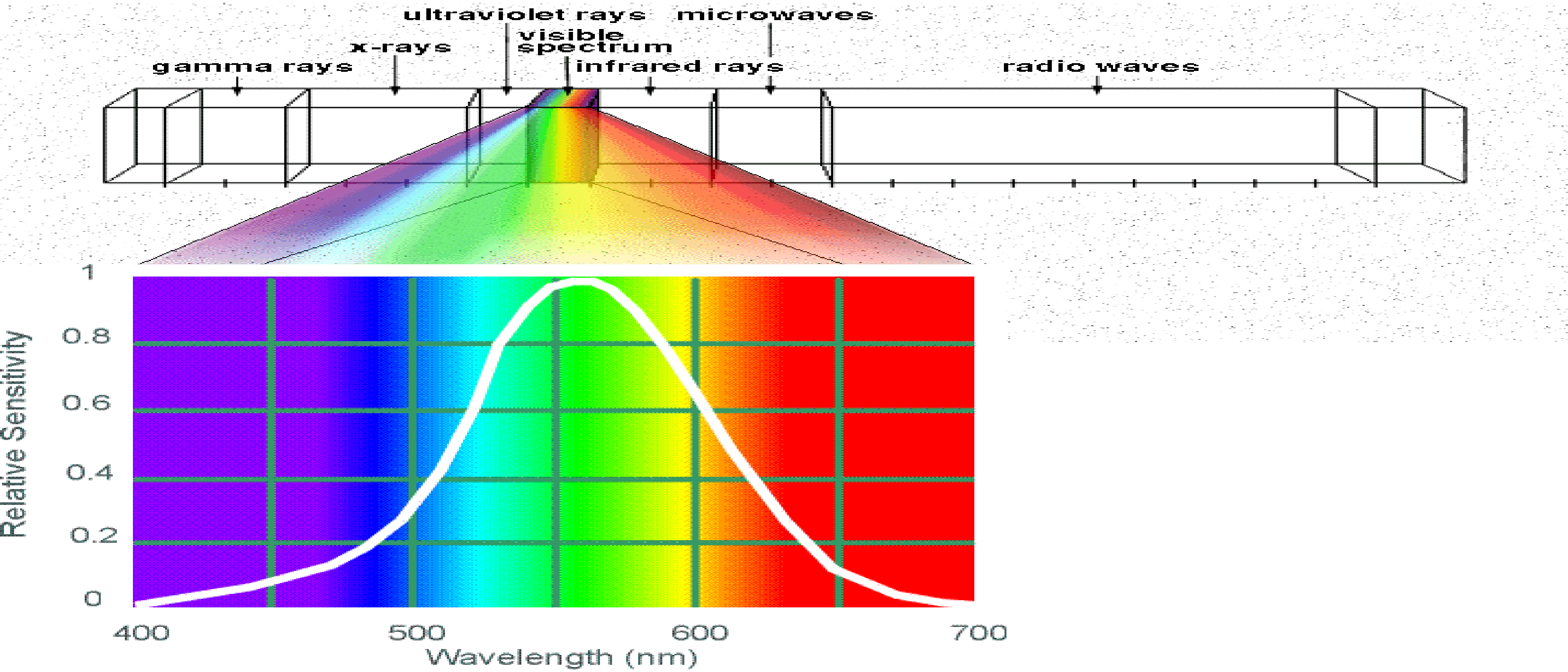


Two types of light-sensitive receptors

- **Cones**
  - cone-shaped
  - less sensitive
  - operate in high light
  - color vision
- **Rods**
  - rod-shaped
  - highly sensitive
  - operate at night
  - gray-scale vision



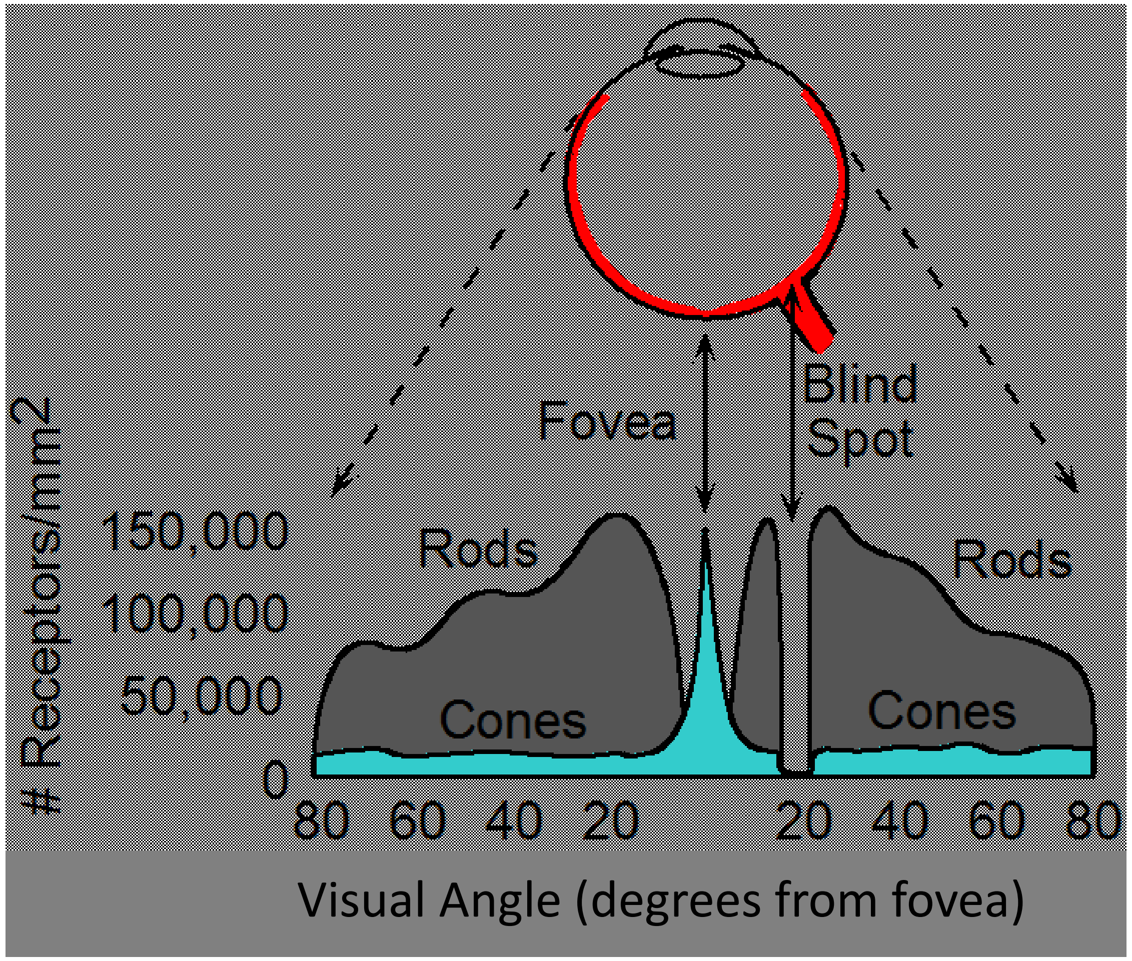
Electromagnetic Spectrum



Human Luminance Sensitivity Function

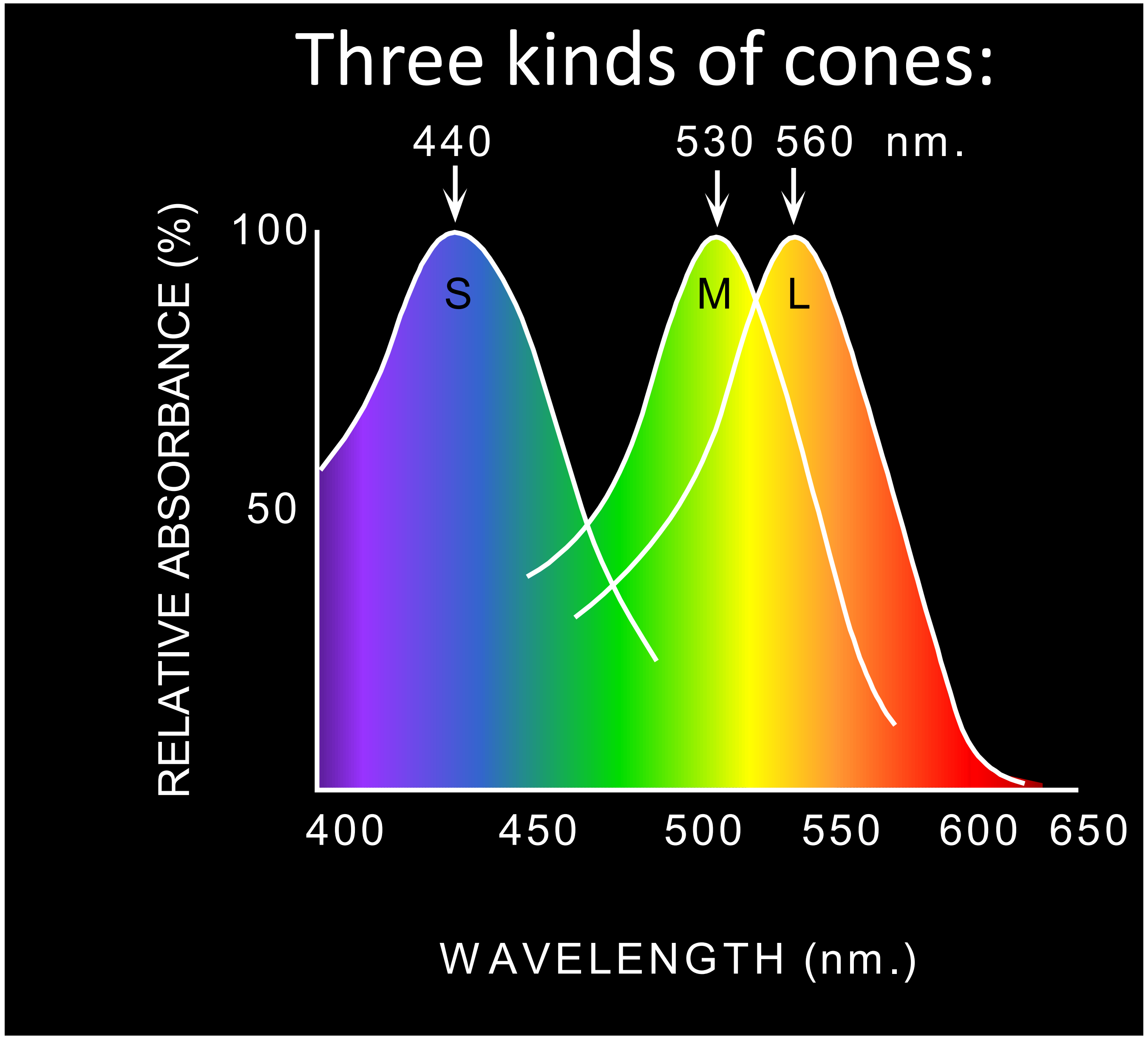
http://www.yorku.ca/eye/photopik.htm

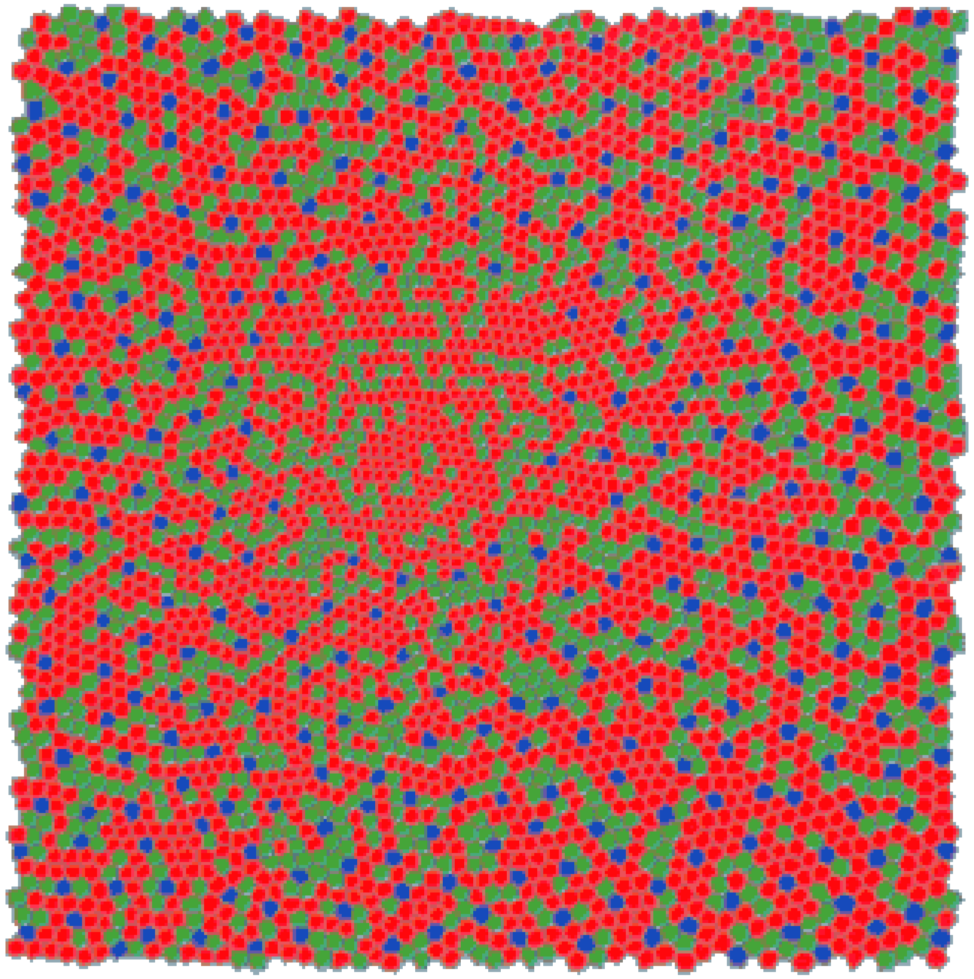
Distribution of Rods and Cones



- Night Sky: where are there more stars off-center?
- Averted vision: http://en.wikipedia.org/wiki/Averted_vision

Physiology of Color Vision





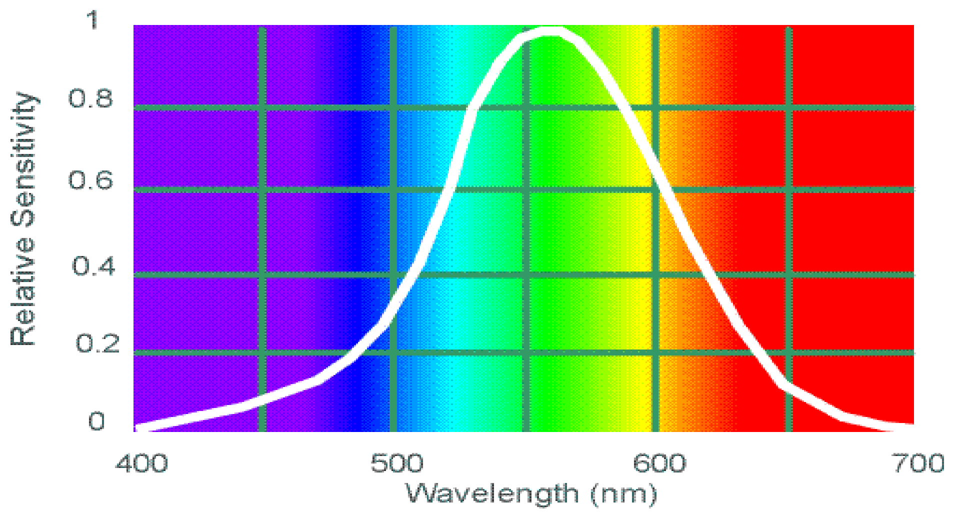
Digital Image Formation

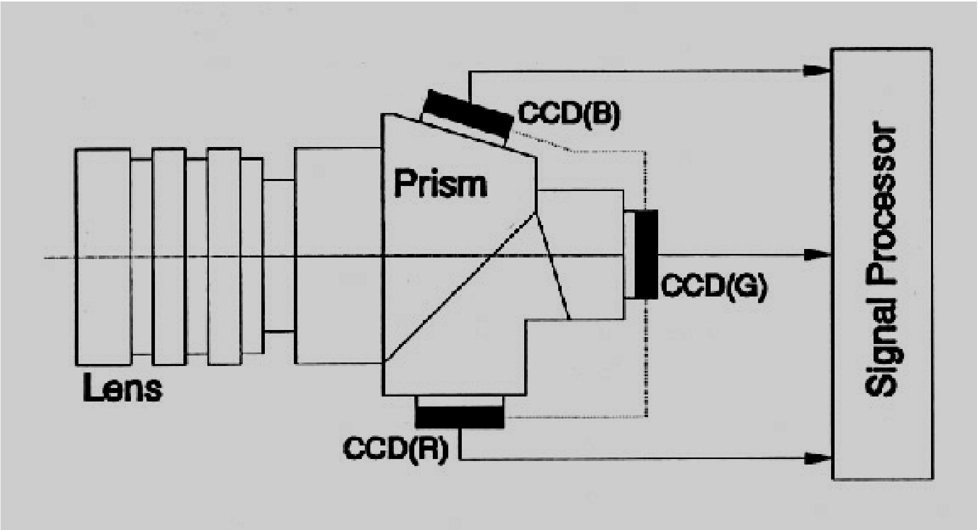
A digital camera replaces film with a sensor array

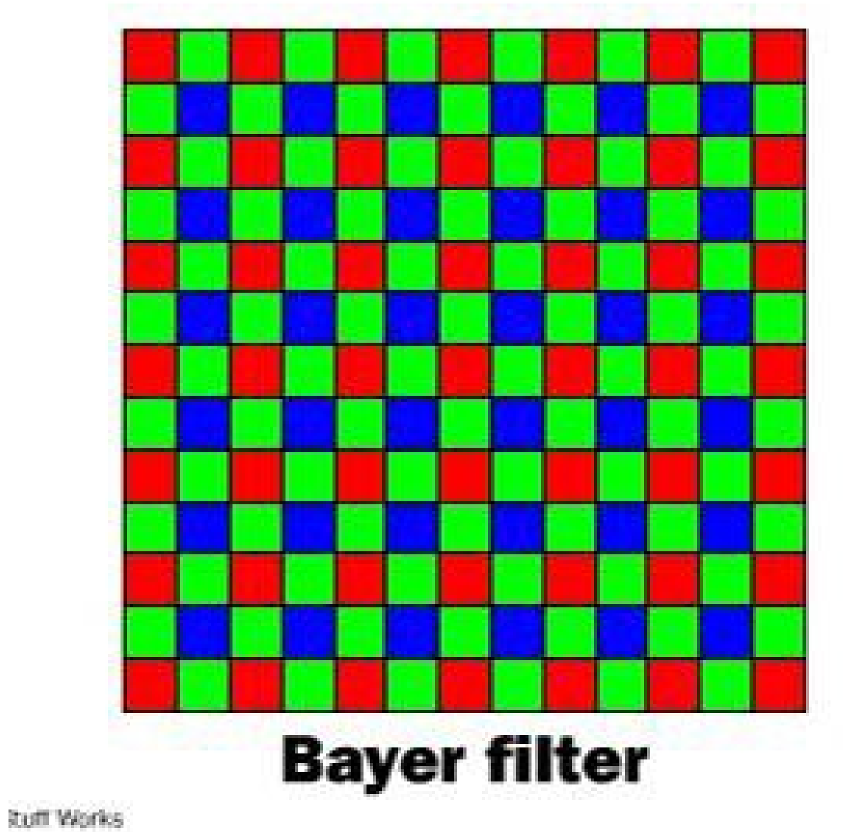
- Each cell in the array is light-sensitive diode that converts photons to electrons
- http://electronics.howstuffworks.com/digital-camera.htm

Slide by Steve Seitz

Digital Image Formation



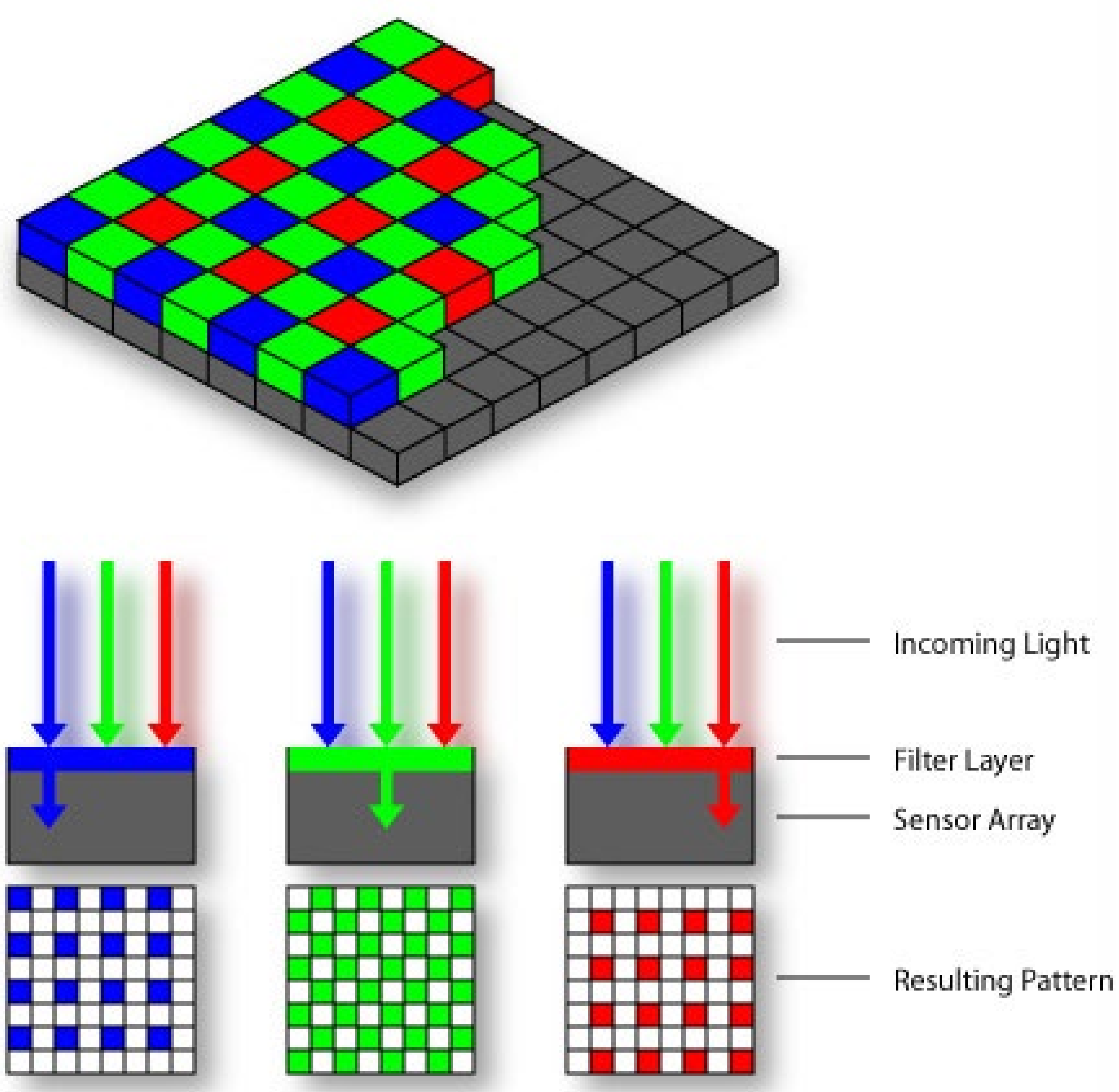


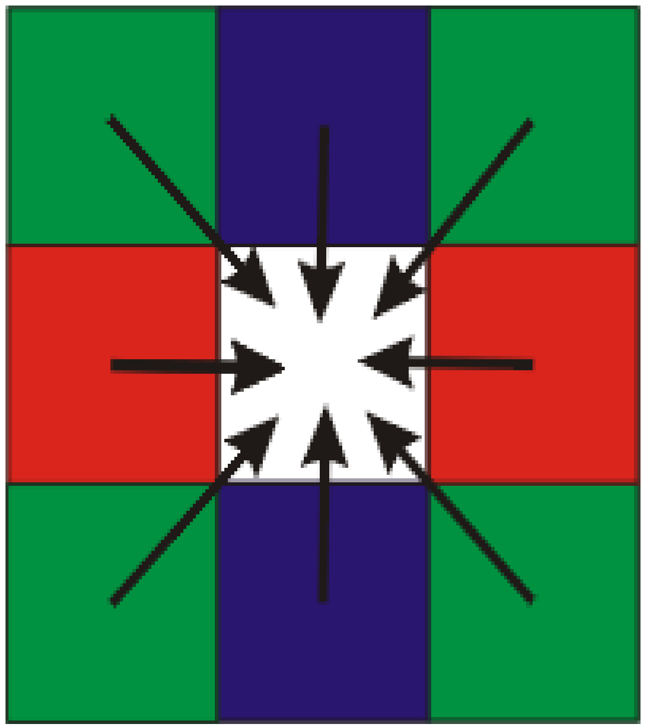


- Why is there more green?

http://www.cooldictionary.com/words/Bayer-filter.wikipedia

Practical Color Sensing: Bayer Grid

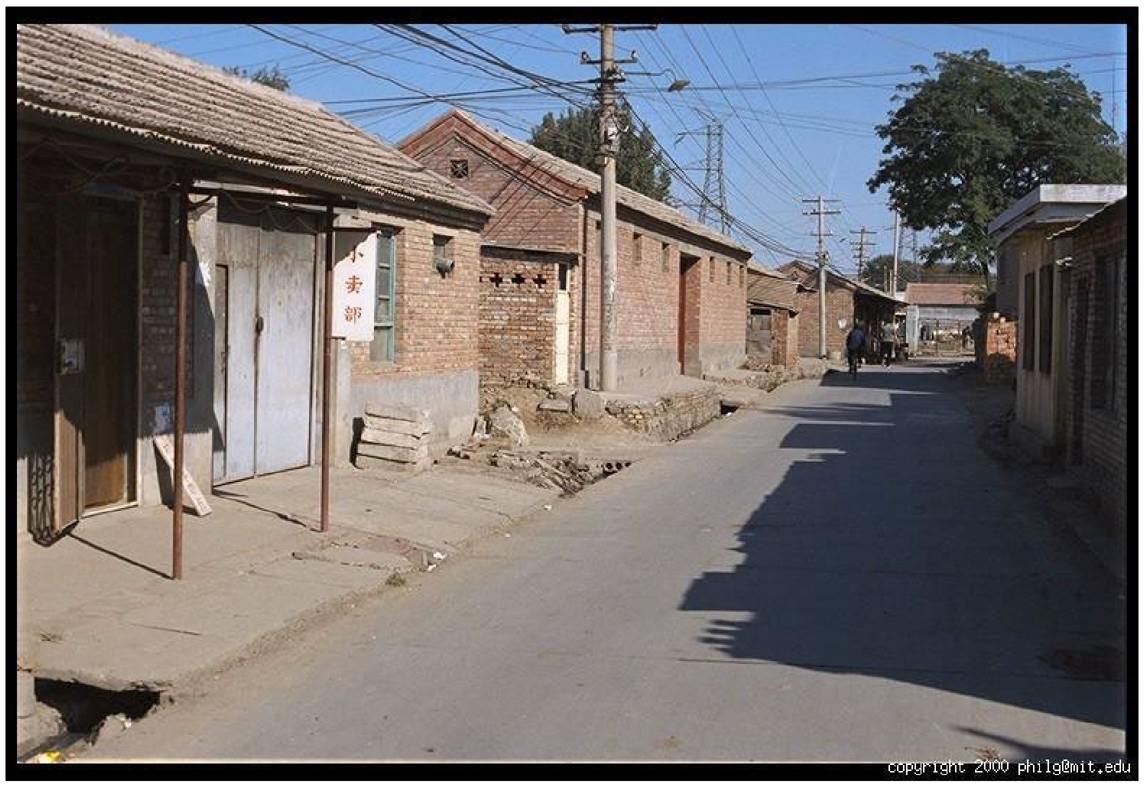




- Estimate RGB at ‘G’ cells from neighboring values

Slide by Steve Seitz

Color Image



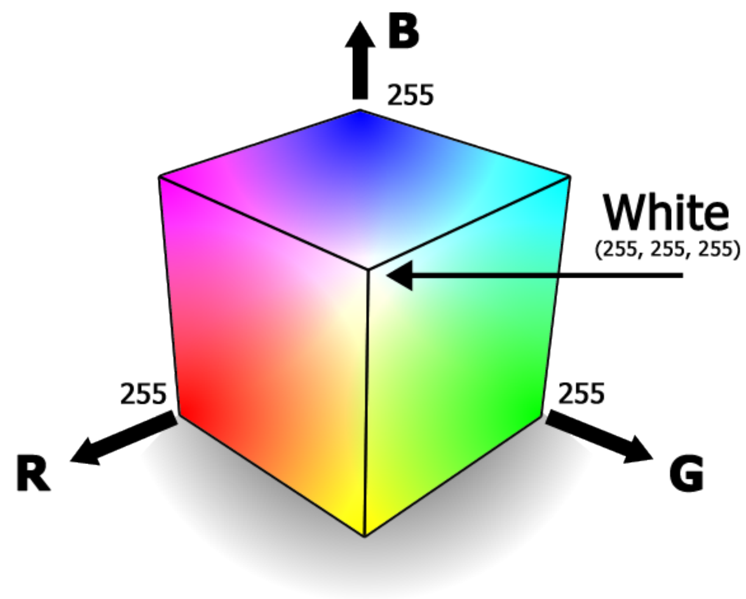
R

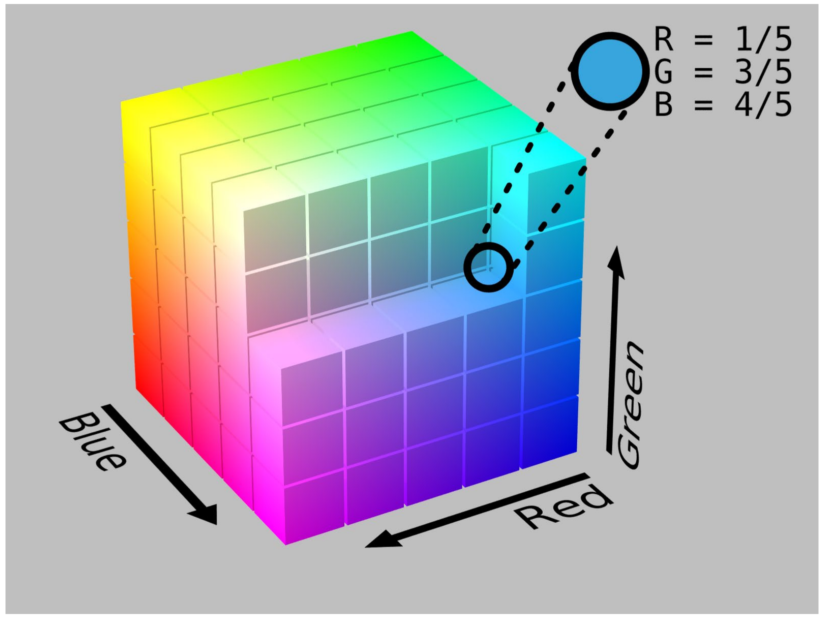
G

B

How to represent color

- Basic concept: Colorspace
- Colorspace is a metric space that contains (almost) all possible colors
- Why almost?
- Many colorspaces exist (HSV, RGB, LAB, LUV…)
- Most frequent color space: RGB

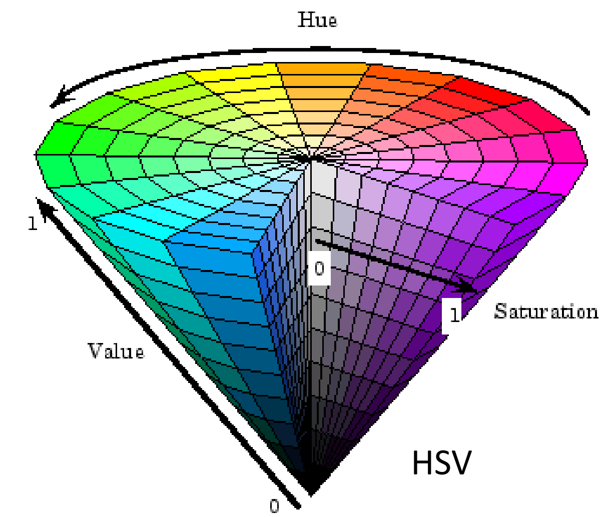




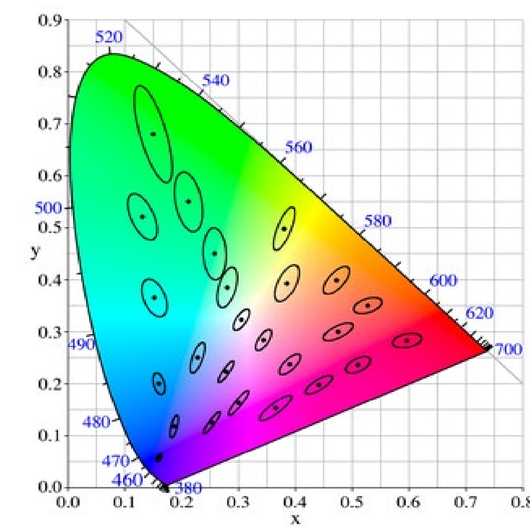
Colorspaces

- RGB:
  - Linear color space (non-uniform color space)
  - Good for devices (e.g., phosphors for monitor), but not for perception
  - Other example: HSV, CIE XYZ
  - But: similar colors not always next to each other
- Non-linear color spaces (uniform color spaces)
  - Enable measuring similarity between colors, i.e. similar colors are next to each other and distance corresponds to similarity.
  - E.g. CIE LAB / LUV color spaces: non-linear transforms of the CIE XYZ color space
  - Conversion simple between RGB and LAB/LUV (e.g. Matlab: lab2luv())

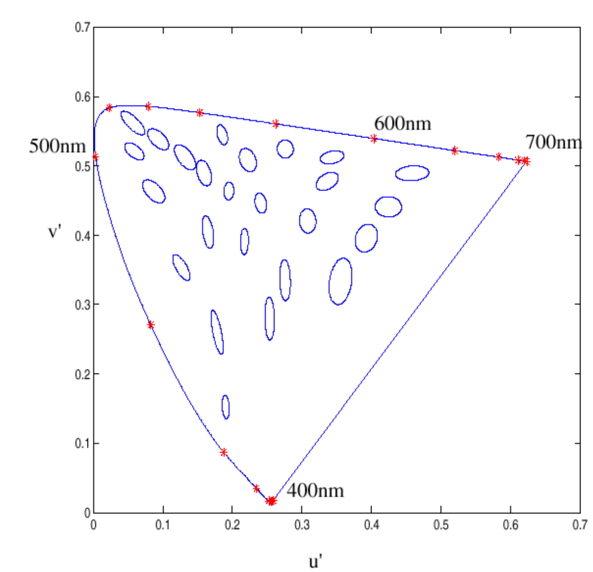
```
from skimage import io, color
rgb = io.imread(filename)
lab = color.rgb2lab(rgb)
```



HSV



McAdam ellipses:
Just noticeable differences in color



CIE LUV

What is more important?

- Color or Intensity?

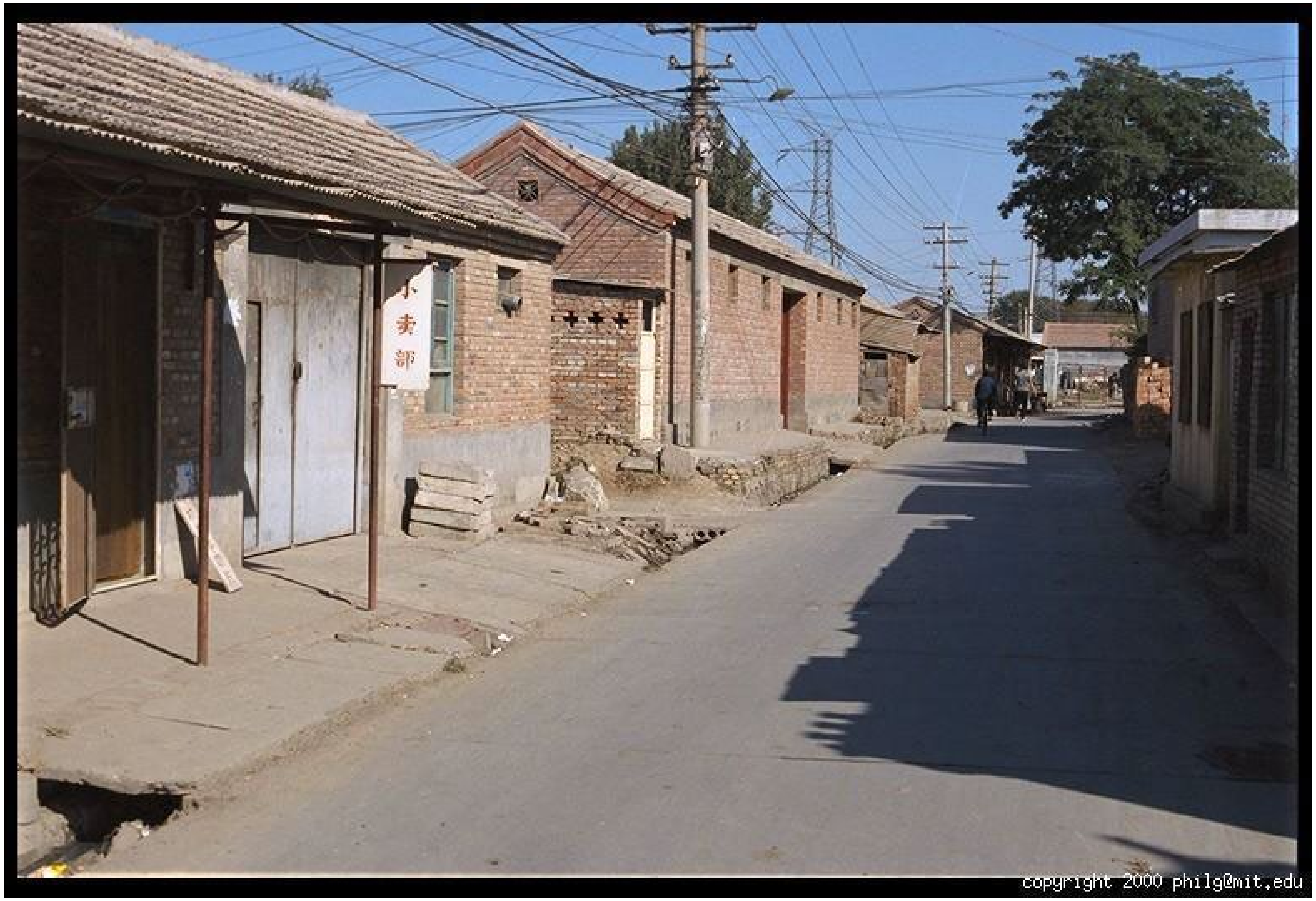
The color part..

Only color shown – constant intensity

The intensity part..

Only intensity shown – constant color

Most information in intensity!

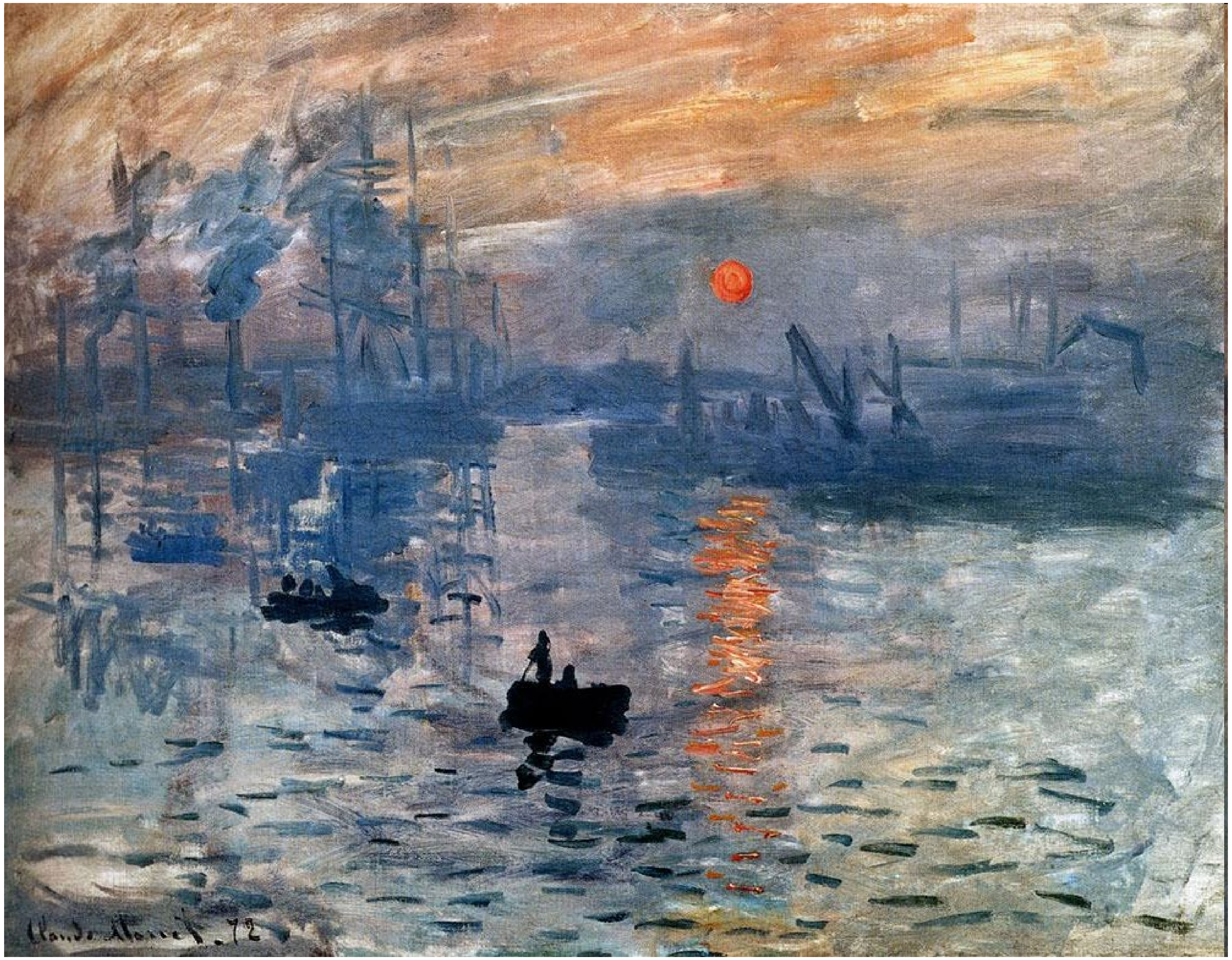


Original image

But: we cannot neglect color completely

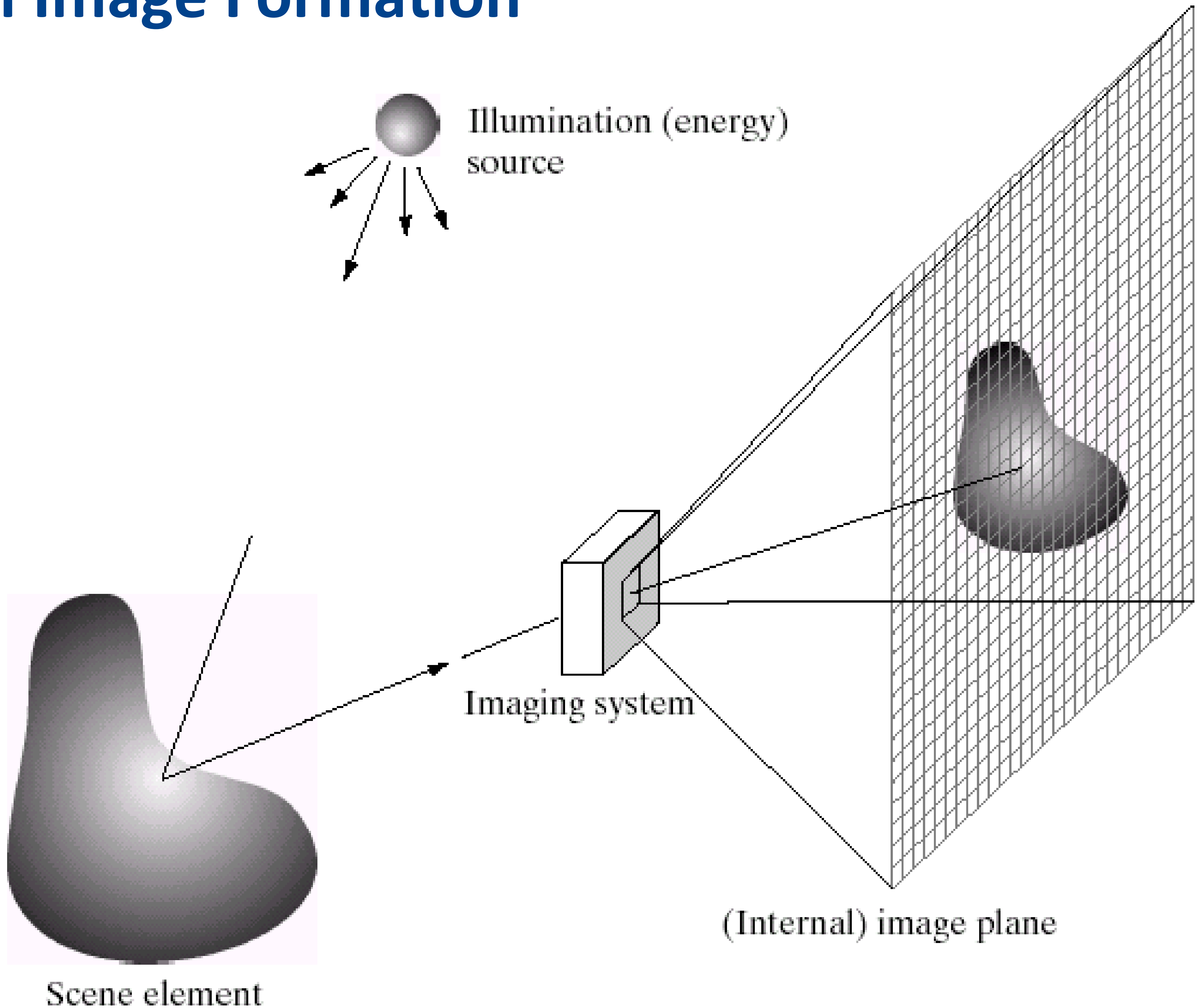
Monet, Sunrise (Impression)

But: we cannot neglect color completely



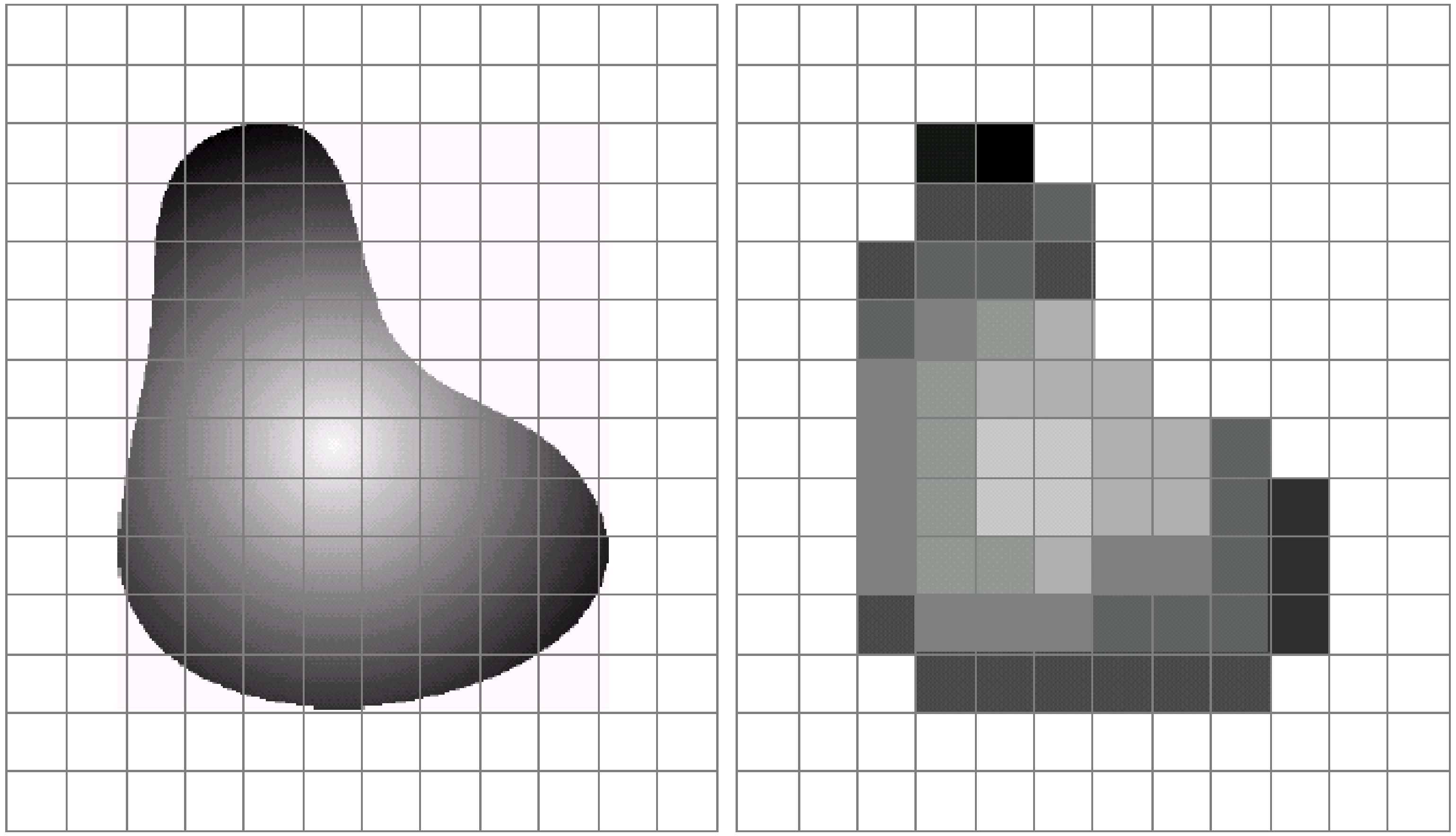
Monet, Sunrise (Impression)

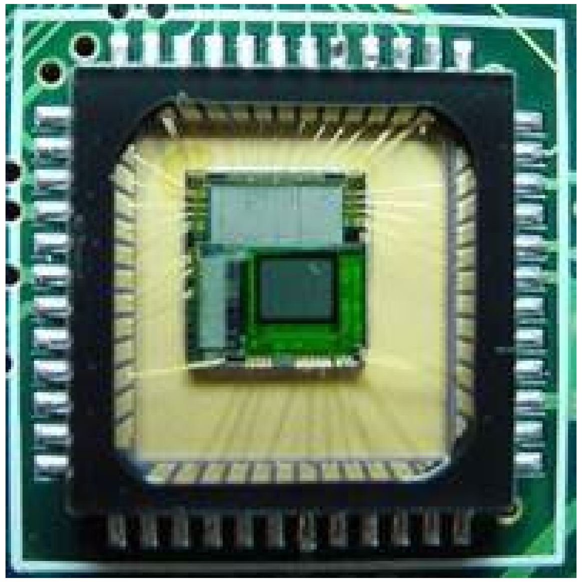
Digital Image Formation



Slide credit: Derek Hoiem

Digital images





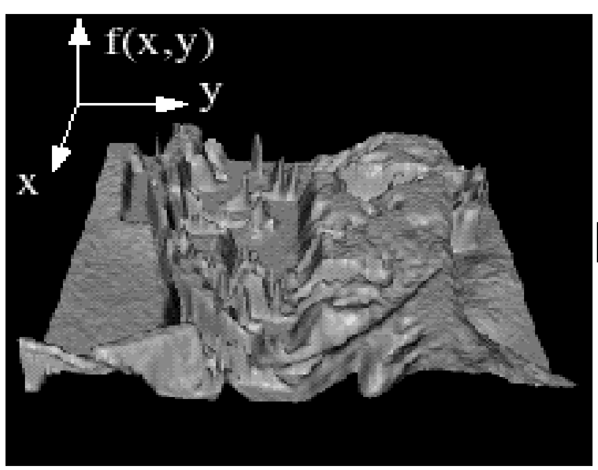
a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Slide credit: Derek Hoiem

Digital images

- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)
- Image thus represented as a matrix of integer values.

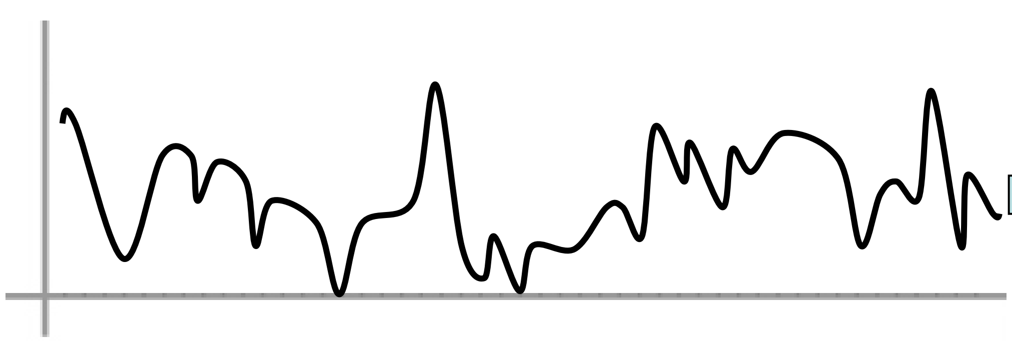


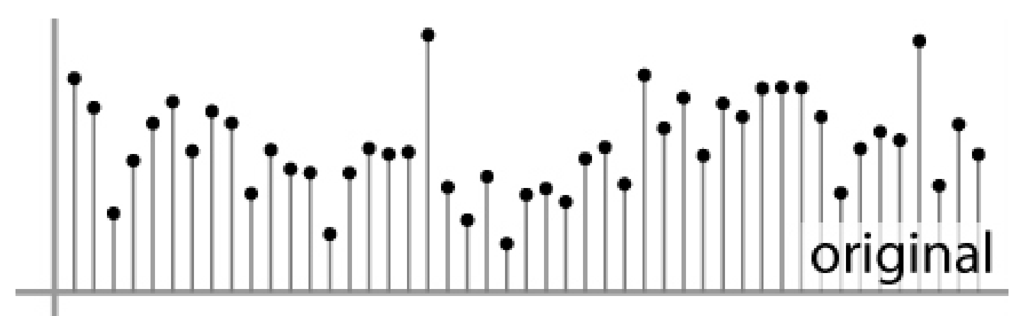
j

i

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

2D





1D

Adapted from S. Seitz

Images in Python

- Images represented as a matrix
- Suppose we have a NxM RGB image called "im"
  - im[0,0,0] = top-left pixel value in R-channel
  - im[y, x, b] = y pixels down, x pixels to right in the b^{th} channel
  - im[N-1, M-1, 2] = bottom-right pixel in B-channel
- im = scipy.imread(filename) returns a uint8 image (values 0 to 255)
- Convert to double format (values 0 to 1) with im = im.astype(np.float32);

column →

row ↓

R

0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

G

									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

B

									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

Slide credit: Derek Hoiem

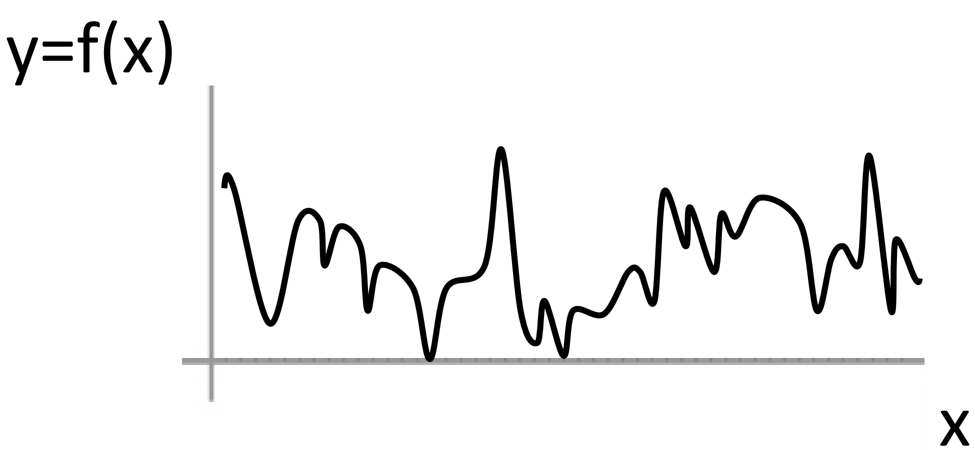
Images are Functions

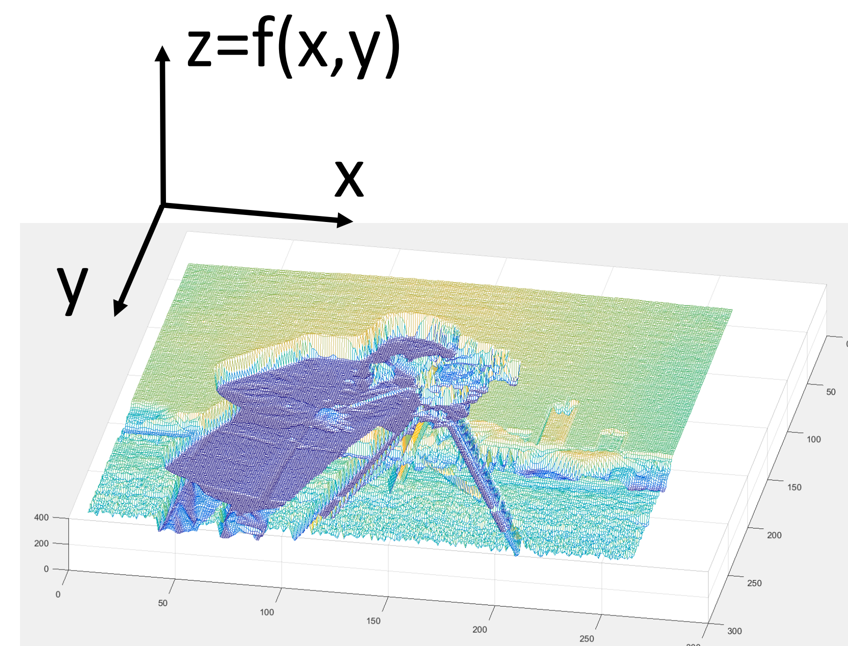
- To calculate with images we need a *formal concept* of an image
- In a mathematical sense an image is a *function* f in two variables: x and y

$$image = f(x, y)$$

- This is a function in 2D which can be represented by a 2D *matrix*!
- 1D function: y=f(x)

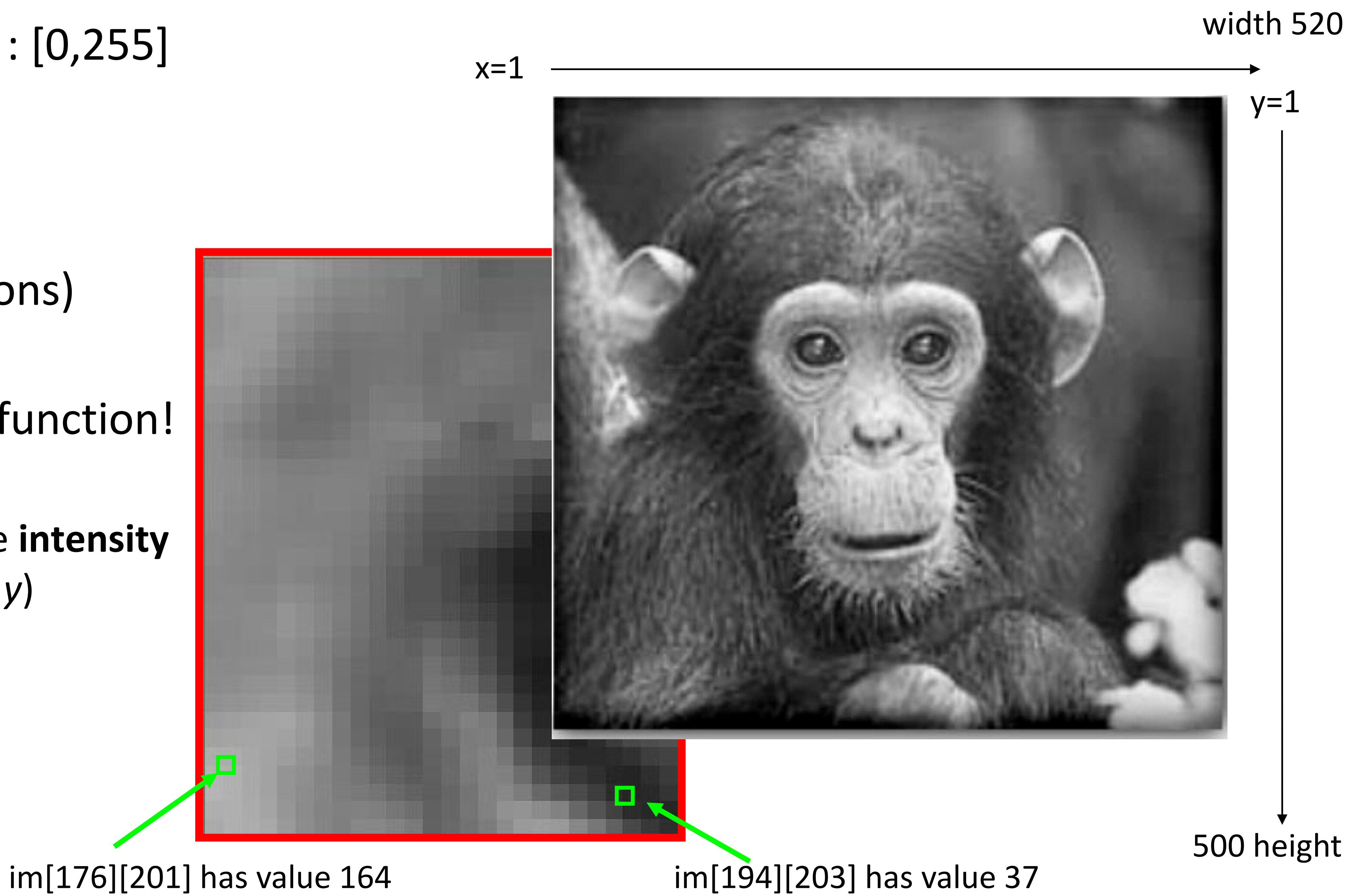
2D function: z=f(x,y)





Accessing the function z=f(x,y)

- Intensity (z) : [0,255]
- x,y: spatial coordinates (pixel positions)
- → Discrete function!
- *f(x,y)* gives the **intensity** at position (*x, y*)



Credit: Kristen Grauman

Color images

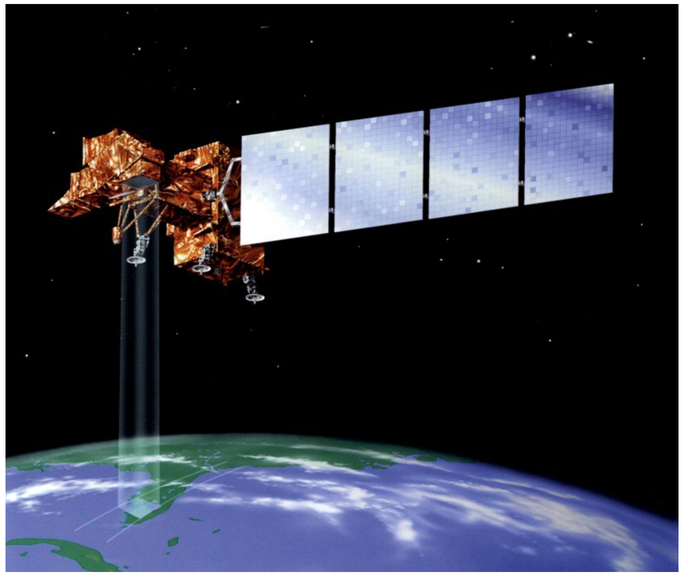
- A color image is just three functions pasted together. We can write this as a “vector-valued” function:
- For color images *f(x,y)* gives the **color** at position (*x, y*)
- Color can be encoded e.g. with red (r), green (g), and blue (b) component

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

- At each pixel position (x,y) we get three values: (r,g,b)
- A color image can be considered as a stack of 3 gray value images (matrices) or an image with 3 layers

Other types of images

- Example: multispectral images - hundreds of channels
- f(x,y) yields a vector of many values (e.g. for different IR band)
- Example Landsat:
  - Spatial resolution 30x30cm
  - 12 channels, see https://de.wikipedia.org/wiki/Landsat



Source: http://patingtoci24.soup.io/post/390611507/Download-multispectral-imaging

Back to grayscale images

- As images are functions, we can calculate with them..

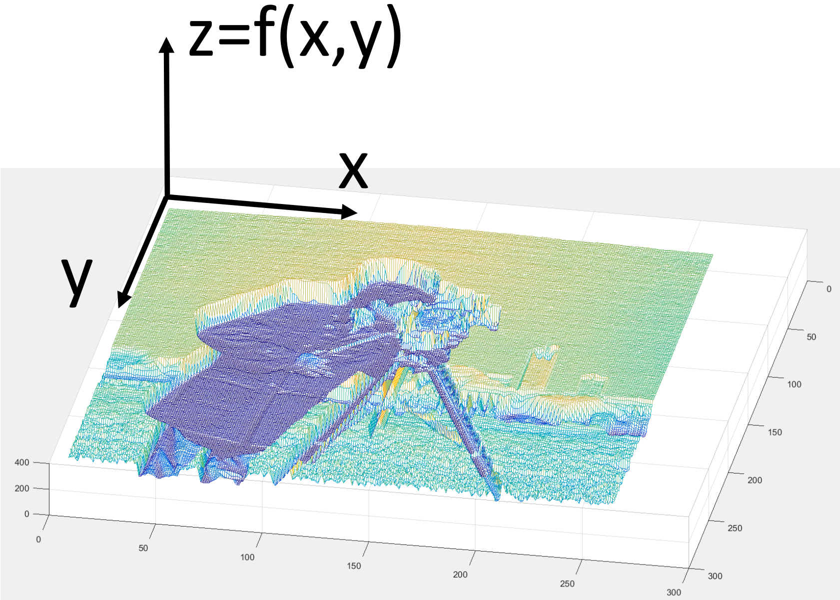
$$image = f(x, y)$$

- Example 1: how can we add two images together? What is the result?

$$f_1(x, y) + f_2(x, y) = ?$$

- Example 2: what happens if we add or multiply the image with a number?

$$3 * f(x, y) =? \qquad 3 + f(x, y) =?$$



Roadmap

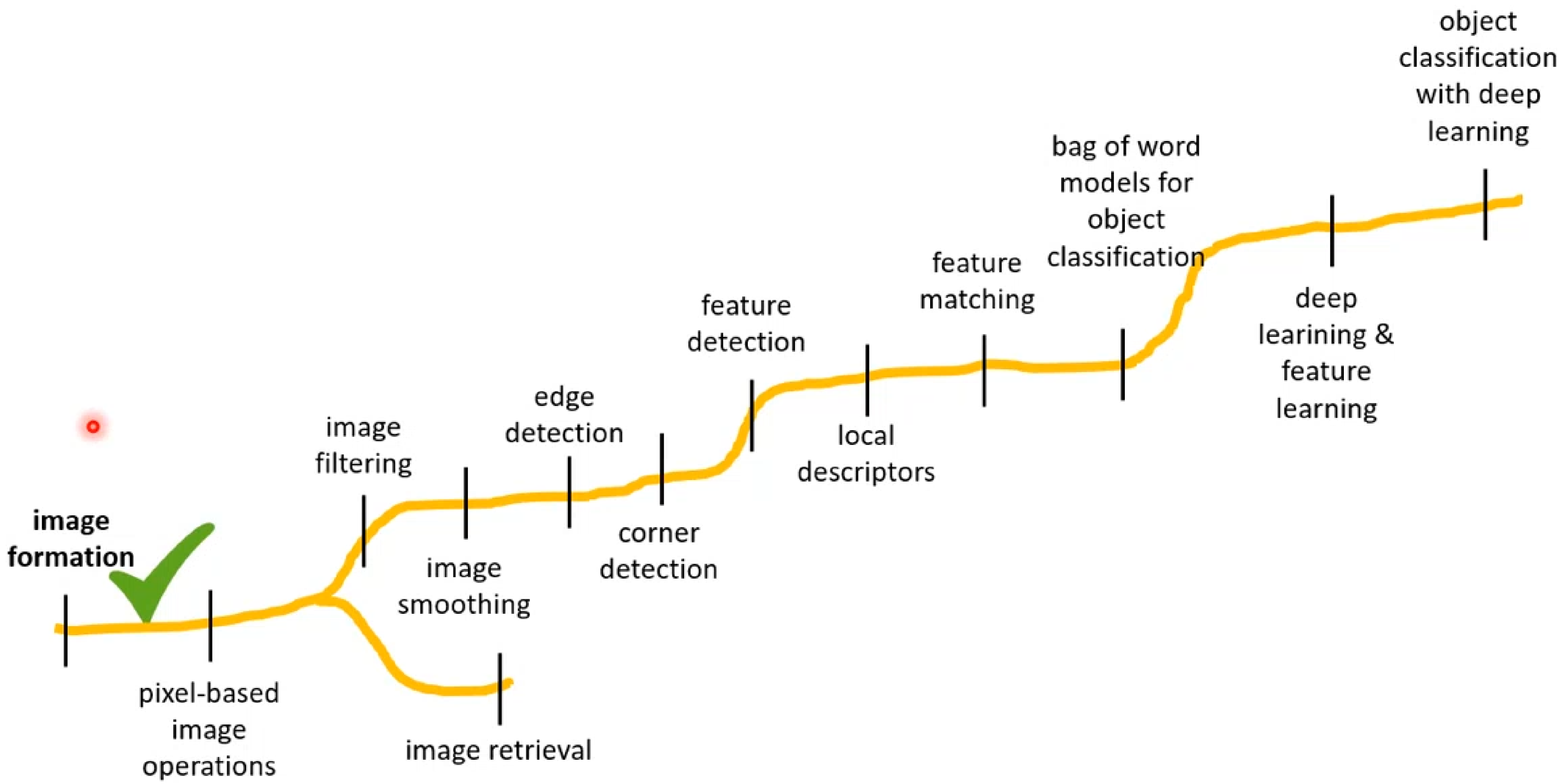


image formation
pixel-based image operations
image filtering
image smoothing
image retrieval
edge detection
corner detection
feature detection
local descriptors
feature matching
bag of word models for object classification
deep learining & feature learning
object classification with deep learning