Class Website

https://redirect.cs.umbc.edu/courses/graduate/691cv/

Lecture Slides will be uploaded *after* the lecture
(usually in 1-2 days)

Access to Google Chat

We will wait until the Waitlist Deadline (Friday) ☺

After that, the TA will add you.

Sign-Up for Scribing →

- All students are required to scribe at least **twice** during the semester.
  - You can sign-up for a preferred week
- Scribing = high-quality detailed notes during the lectures in that week, typeset using Overleaf/LaTeX
  - Template is on the class website. Hand-drawn figures are allowed.
- *Notes for Monday lectures are due before class next Monday*
- *Notes for Wednesday lectures are due before class next Wednesday*

Email your notes as PDF, with subject: "[Scribing Submission] <lecture-date>" to gokhale@umbc.edu AND ssaha2@umbc.edu

We may deviate a bit from the planned topics listed below.

to sign-up, enter your name below in an empty slot -- do not overwrite your classmates' entries!

Lecture Date	Day	Planned Topic	Notes Due	Scribe 1	Scribe 2	Scribe 3
29-Jan	M	Intro	5-Feb	Olivia Amaral		
5-Feb	M	Filtering I	12-Feb	Olivia Amaral		
12-Feb	M	Features I	19-Feb	Jabril Hall		
19-Feb	M	Features III	26-Feb			
26-Feb	M	no scribing				
4-Mar	M	ML for CV (NN)	11-Mar			
11-Mar	M	Pytorch Tutorial	18-Mar			
18-Mar	M	no scribing (Spring Break)				
25-Mar	M	Image Transformations	1-Apr			
1-Apr	M	no scribing (Midterm)				
8-Apr	M	Epipolar Geometry	15-Apr			
15-Apr	M	V&L	22-Apr			
22-Apr	M	Robustness	29-Apr			
29-Apr	M	no scribing (Guest Lecture I)				
6-May	M	no scribing (Guest Lecture III)				
13-May	M	no scribing (Project Presentations)				

Lecture Date	Day	Planned Topic	Notes Due	Scribe 1	Scribe 2	Scribe 3
31-Jan	W	Image Formation				
7-Feb	W	Filtering II				
14-Feb	W	Features II				
21-Feb	W	no scribing				
28-Feb	W	ML for CV				
6-Mar	W	ML for CV (GD)				
13-Mar	W	Object Detection				
20-Mar	W	no scribing (Spring Break)				
27-Mar	W	Homographies				
3-Apr	W	Camera Models				
10-Apr	W	Stereo				
17-Apr	W	Image Synthesis				
24-Apr	W	buffer				
1-May	W	no scribing (Guest Lecture II)				
8-May	W	no scribing (Project Presentations)				

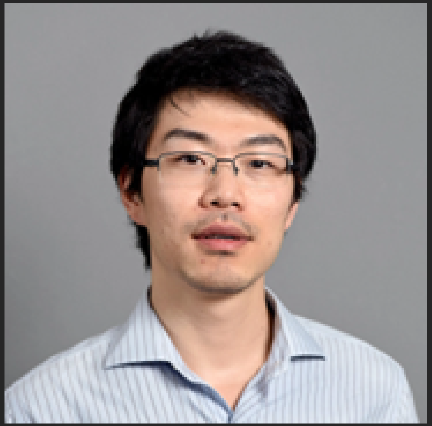
The LaTeX template has been released on the class website.

Please create an account on https://overleaf.com (it is free!)

For a tutorial on how to use LaTeX with Overleaf, visit:
https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
(this link is on the website)

PPR Seminar

Advances in Perception, Prediction, and Reasoning



Dr. Yezhou Yang

Associate Professor,
School of Computing & AI, Arizona State University

https://yezhouyang.engineering.asu.edu/

Visual Concept Learning Beyond Appearances: Modernizing a Couple of Classic Ideas

February 8, 2024 3:30 – 4:30 PM

ITE 325-B or Webex: https://umbc.webex.com/meet/gokhale

CMSC 491/691

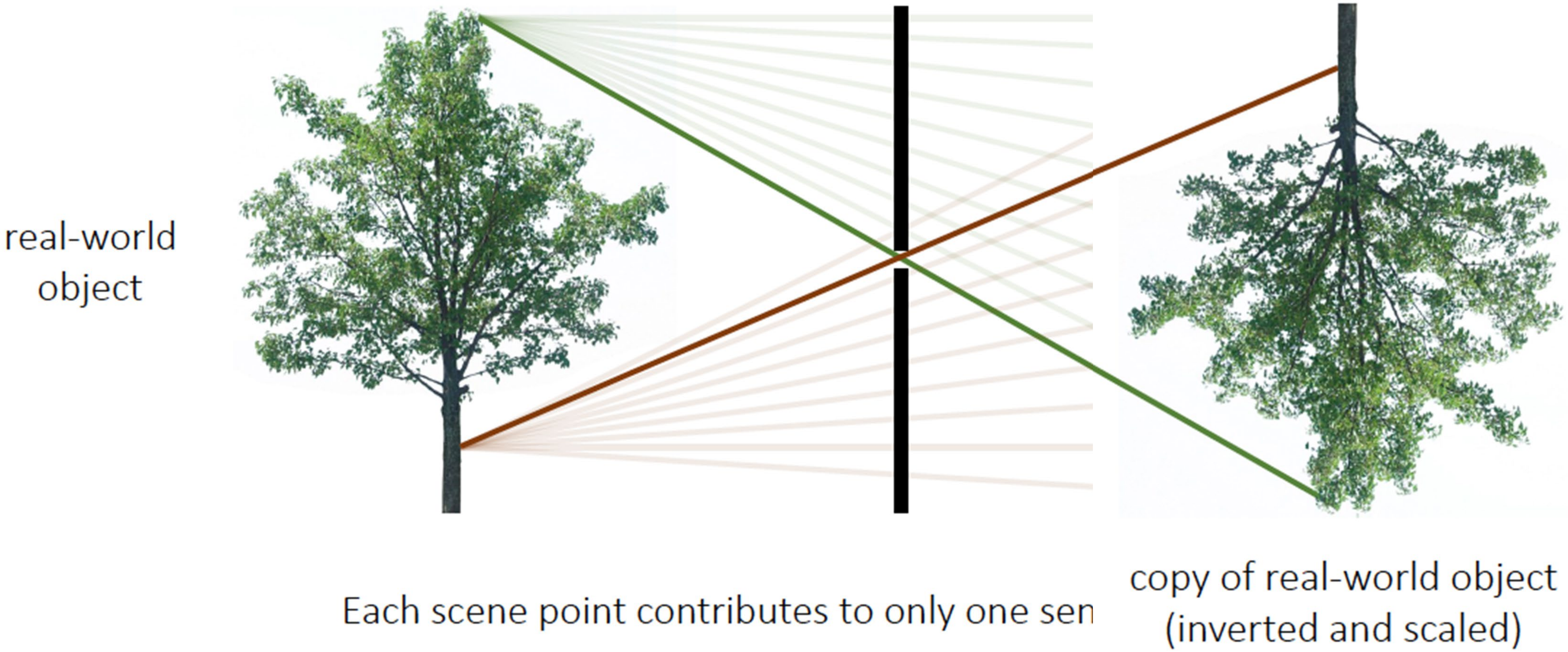
Lecture 2

Image Formation

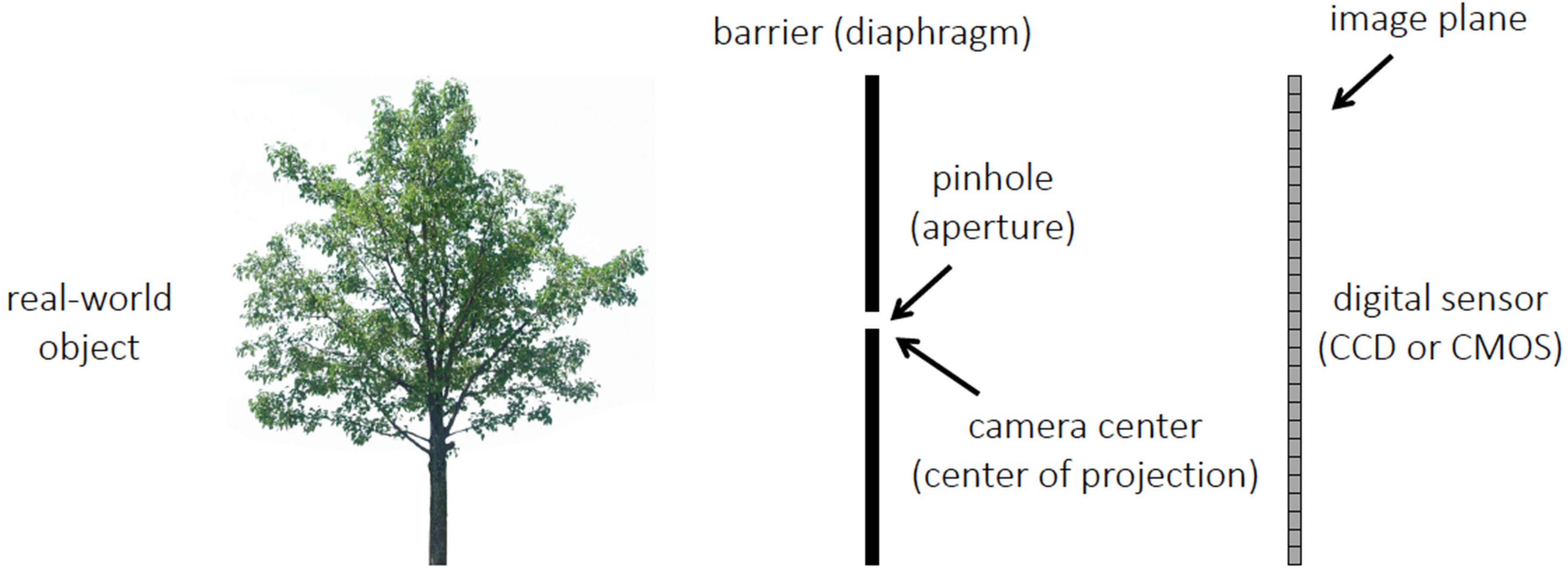
Some slides from Jayasuriya, Turaga, Szeliski, Brown

Recap

Pinhole imaging

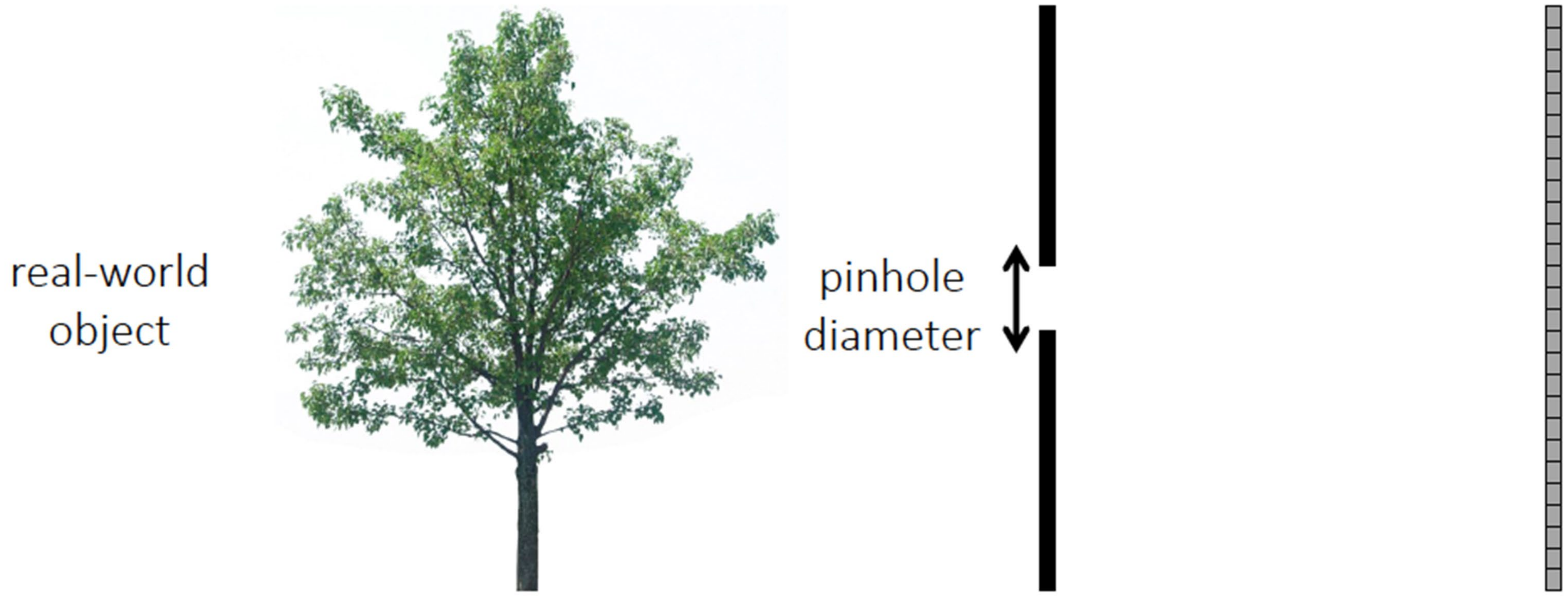


Recap Pinhole camera terms



Pinhole size

What happens as we change the pinhole diameter?

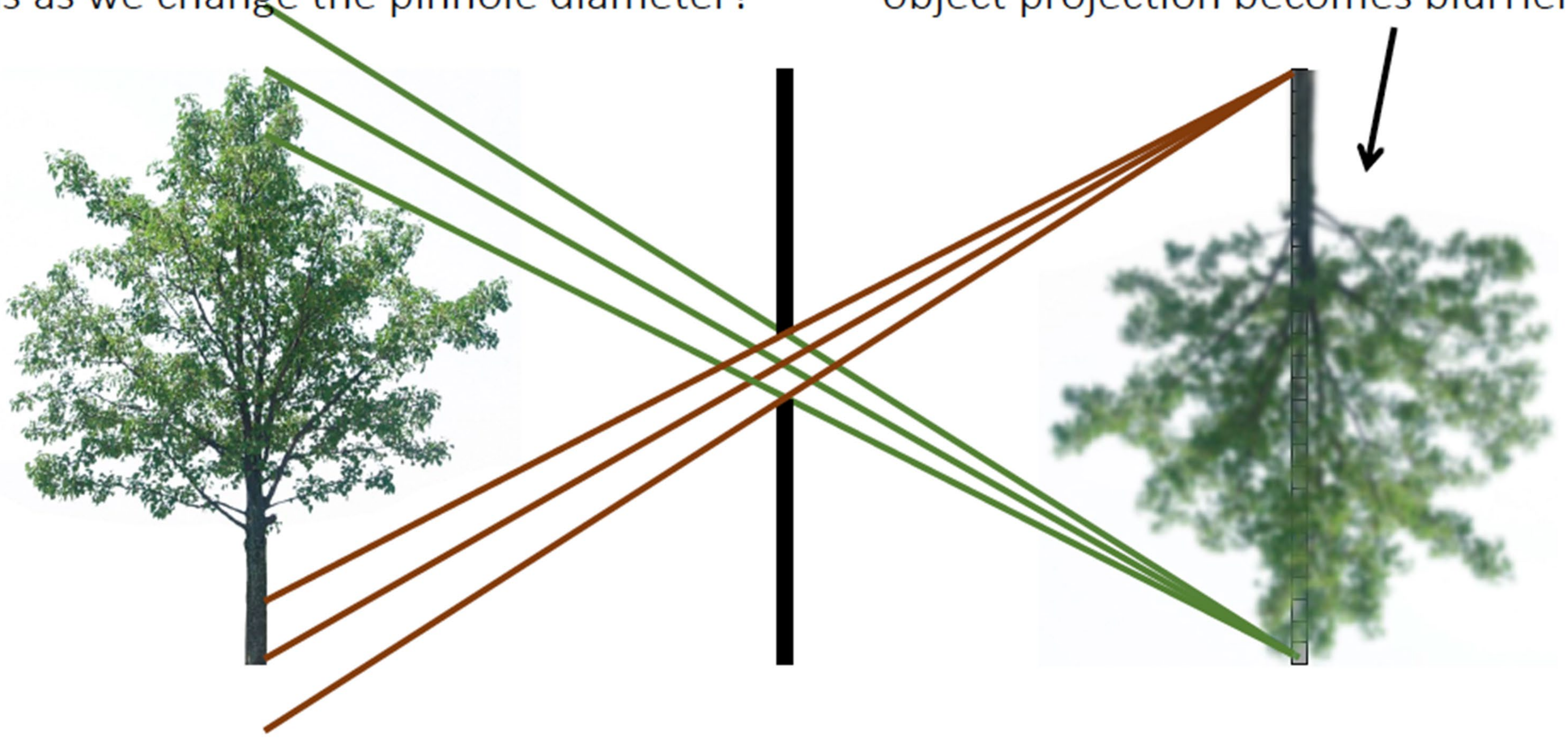


Pinhole size

What happens as we change the pinhole diameter?

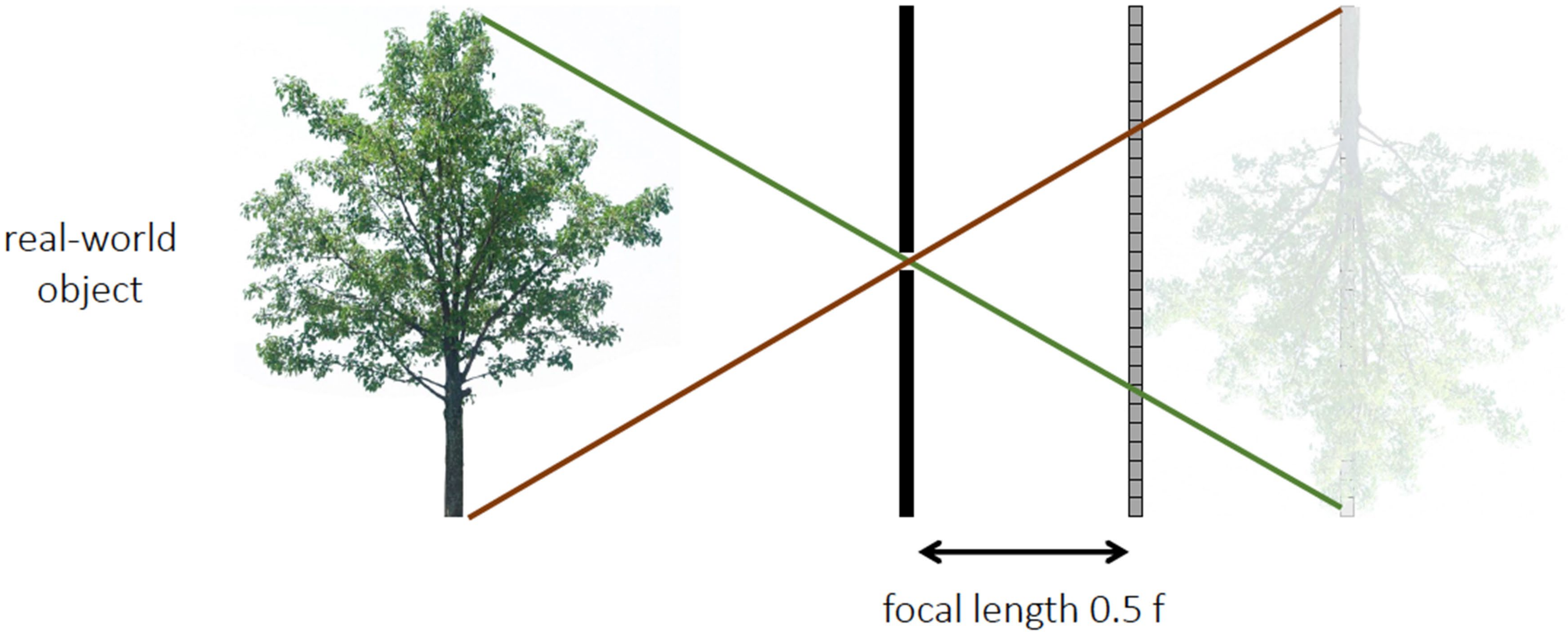
object projection becomes blurrier

real-world object



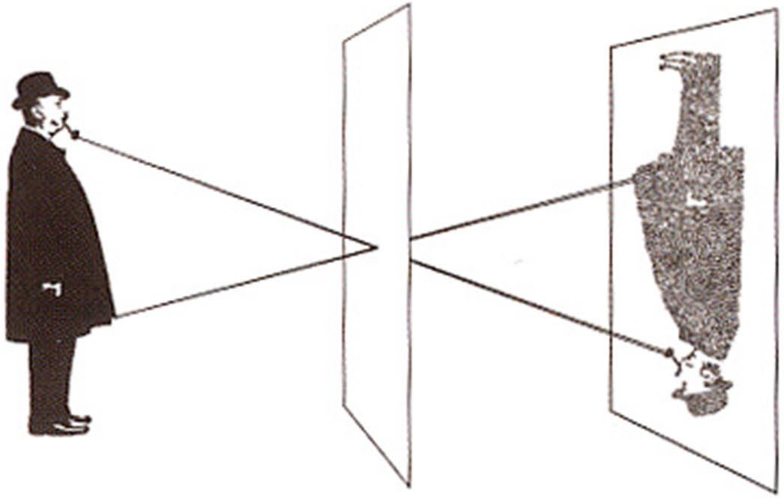
Focal length

What happens as we change the focal length?

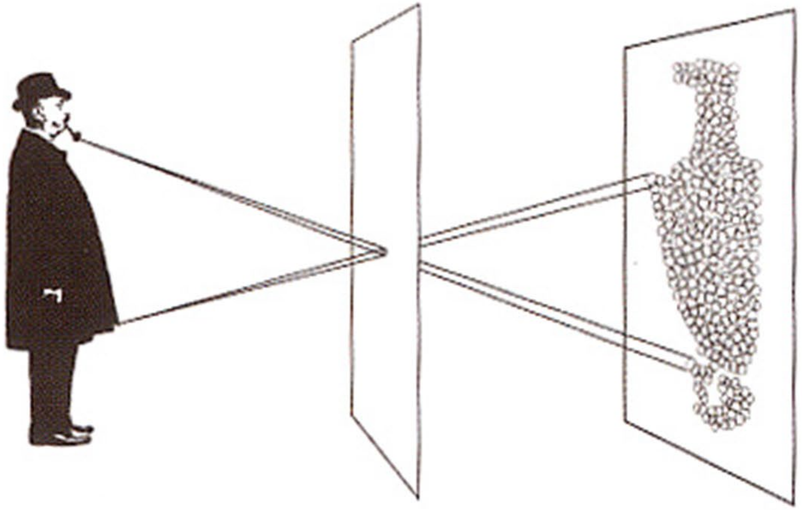


Magnification depends on focal length

Photograph made with small pinhole



Photograph made with larger pinhole



Recap

Problems with Pinholes

- Pinhole size (aperture) must be "very small" to obtain a clear image.
- However, as pinhole size is made smaller, less light is received by image plane.
- If pinhole is comparable to wavelength λ of incoming light, DIFFRACTION blurs the image!
- Sharpest image is obtained when:

pinhole diameter $d = 2\sqrt{f'\lambda}$

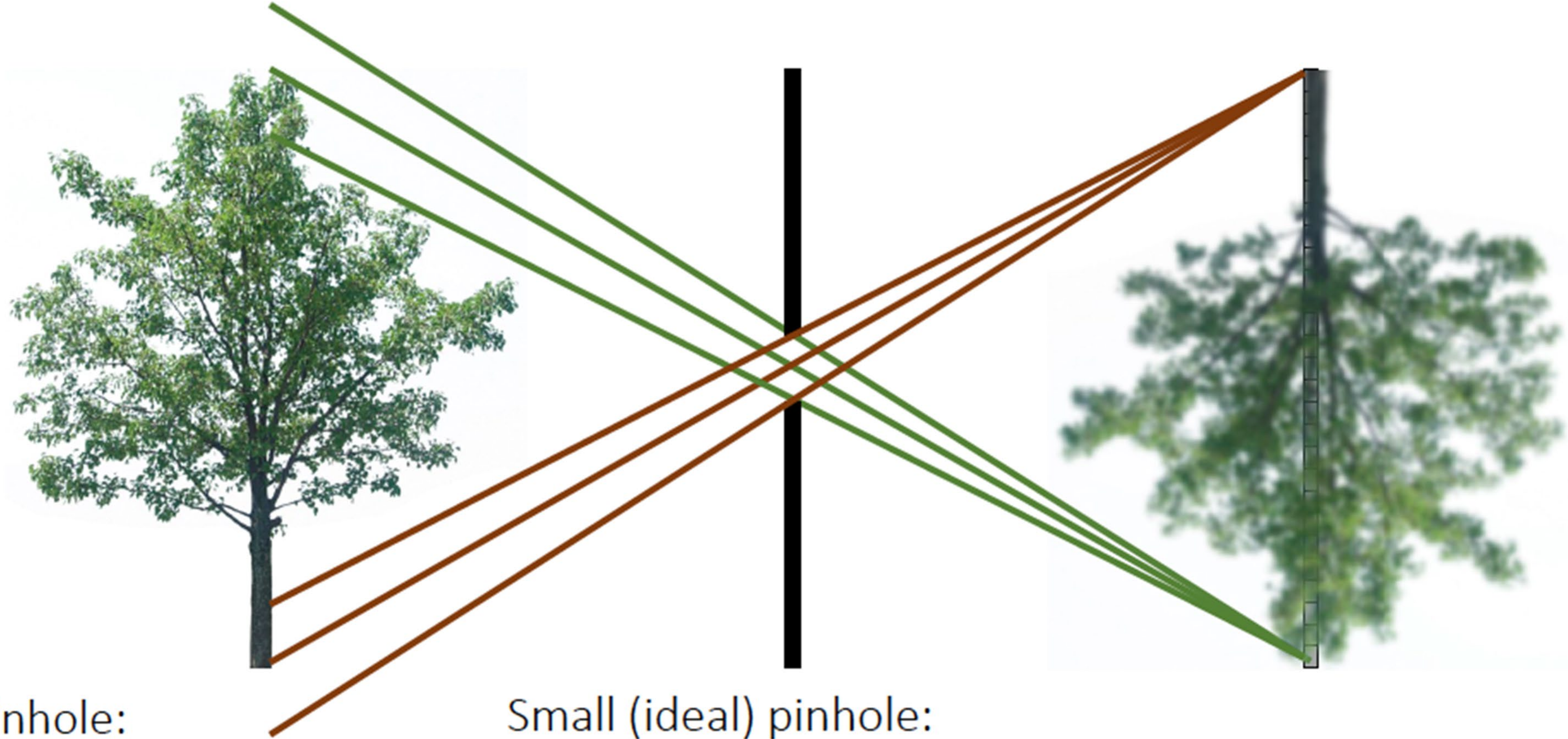
Example: If $f' = 50mm$,

$= 600nm$ (red),

$d = 0.36mm$

Fig. 5.96 The pinhole camera. Note the variation in image clarity as the hole diameter decreases. [Photos courtesy Dr. N. Joel, UNESCO.]

Pinhole camera

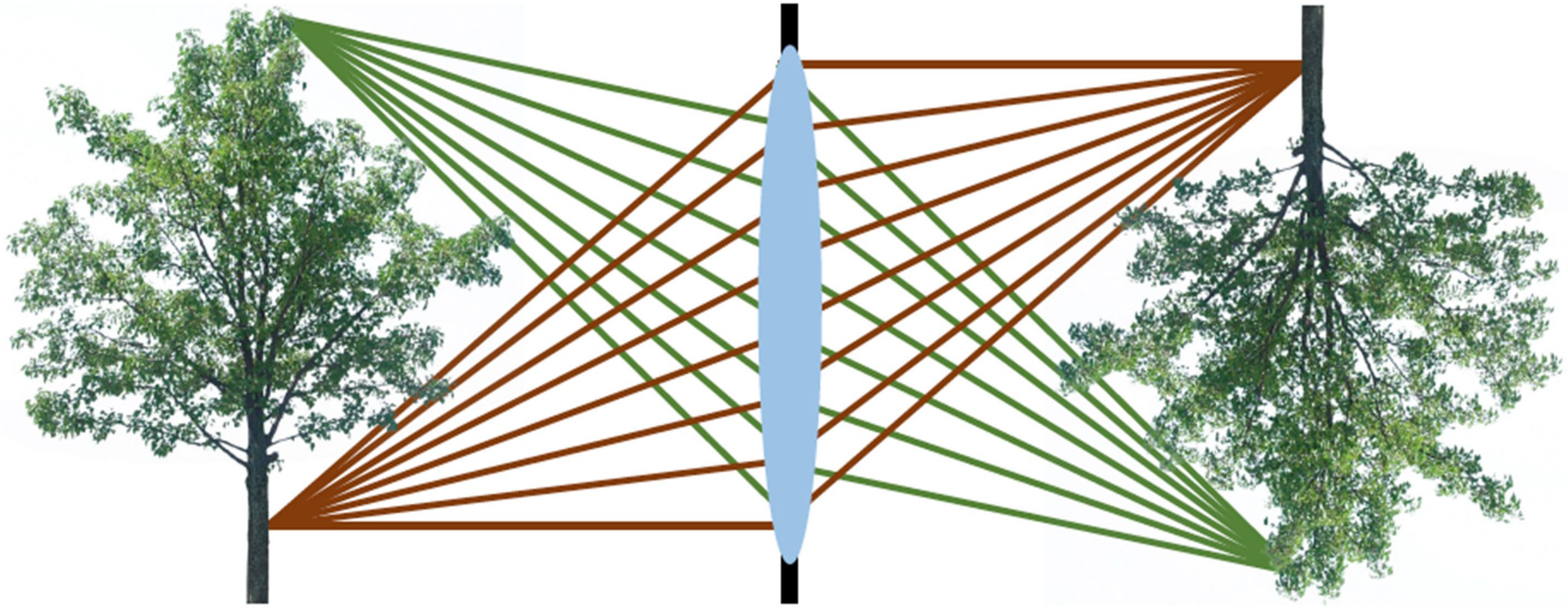


Large pinhole:
1. Image is blurry.
2. Signal-to-noise ratio is high.

Small (ideal) pinhole:
1. Image is sharp.
2. Signal-to-noise ratio is low.

Best of Both Worlds?

Almost, by using lenses

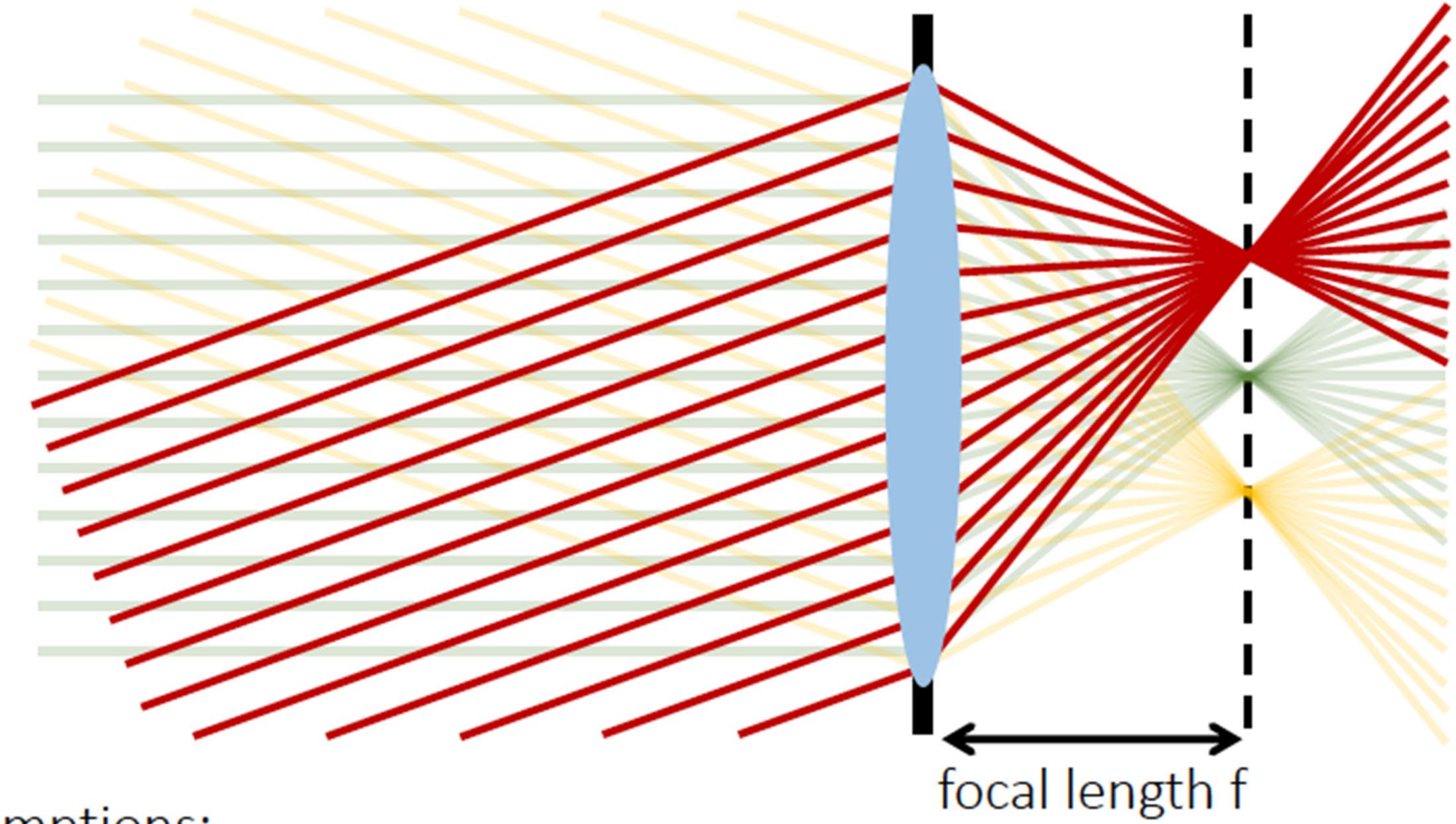


Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

Thin lens model

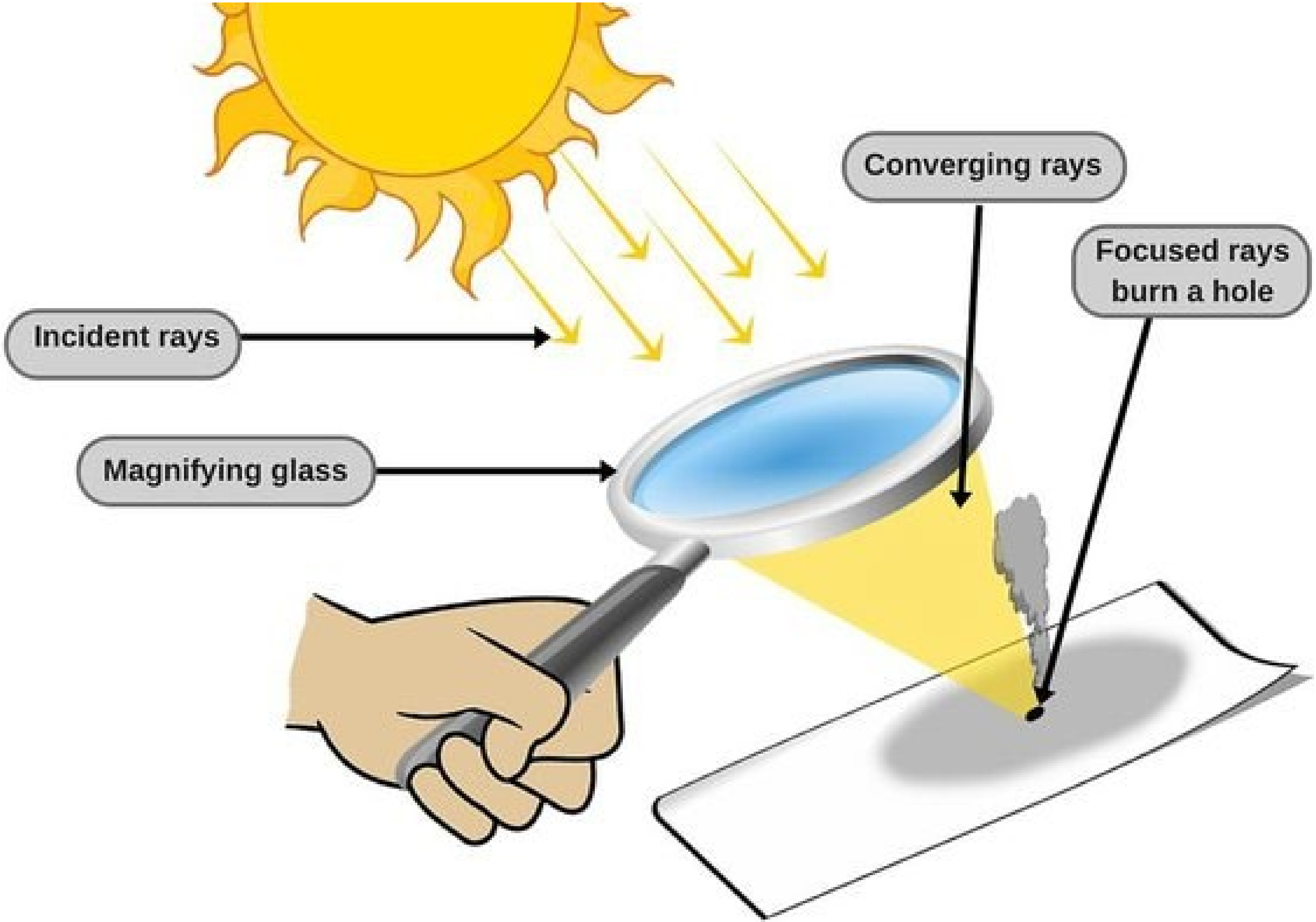
Simplification of geometric optics for well-designed lenses.

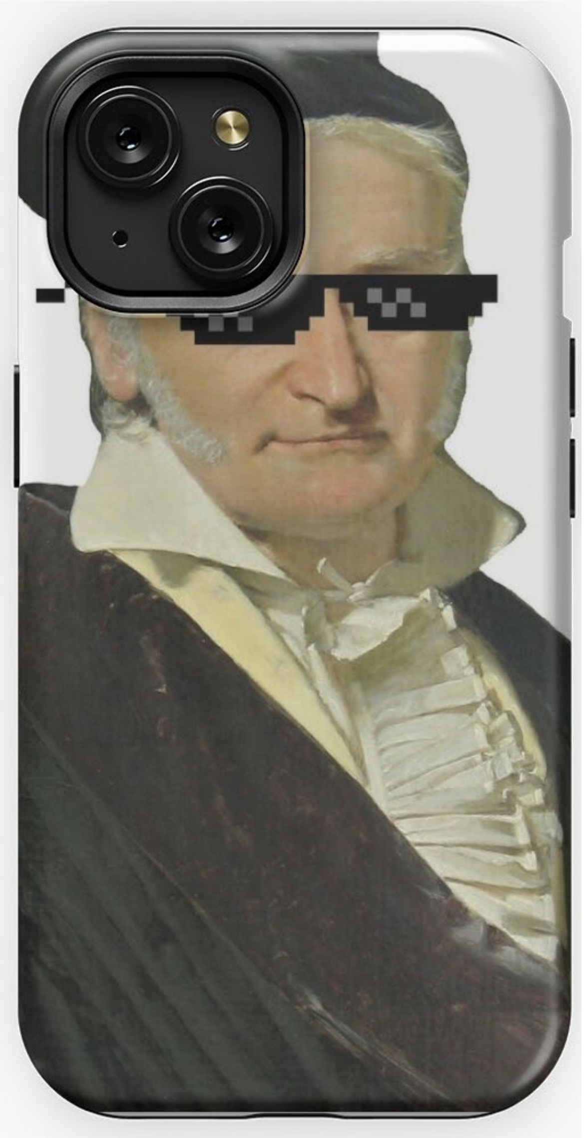


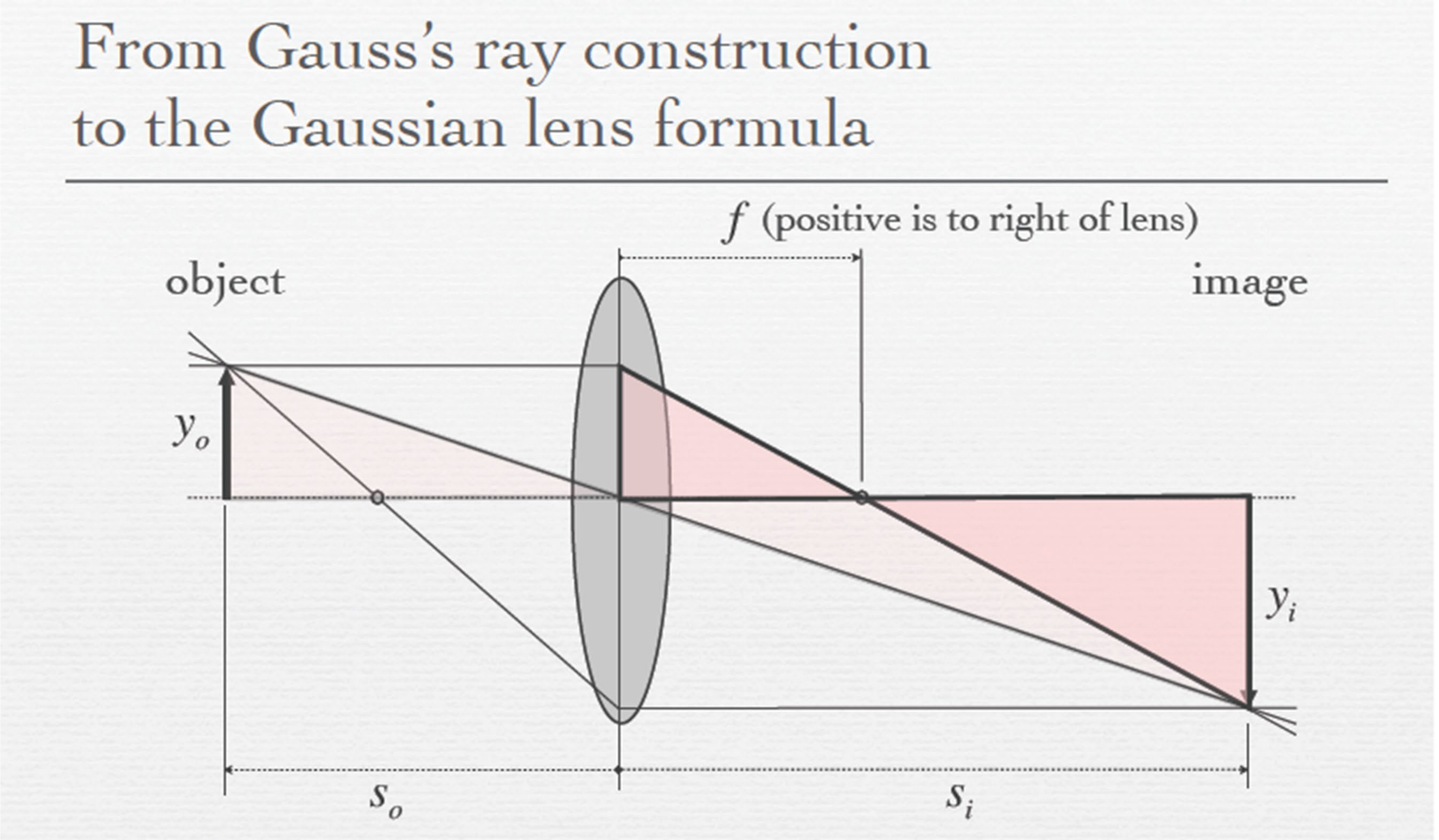
Two assumptions:

1. Rays passing through lens center are unaffected.
2. Parallel rays converge to a single point located on focal plane.

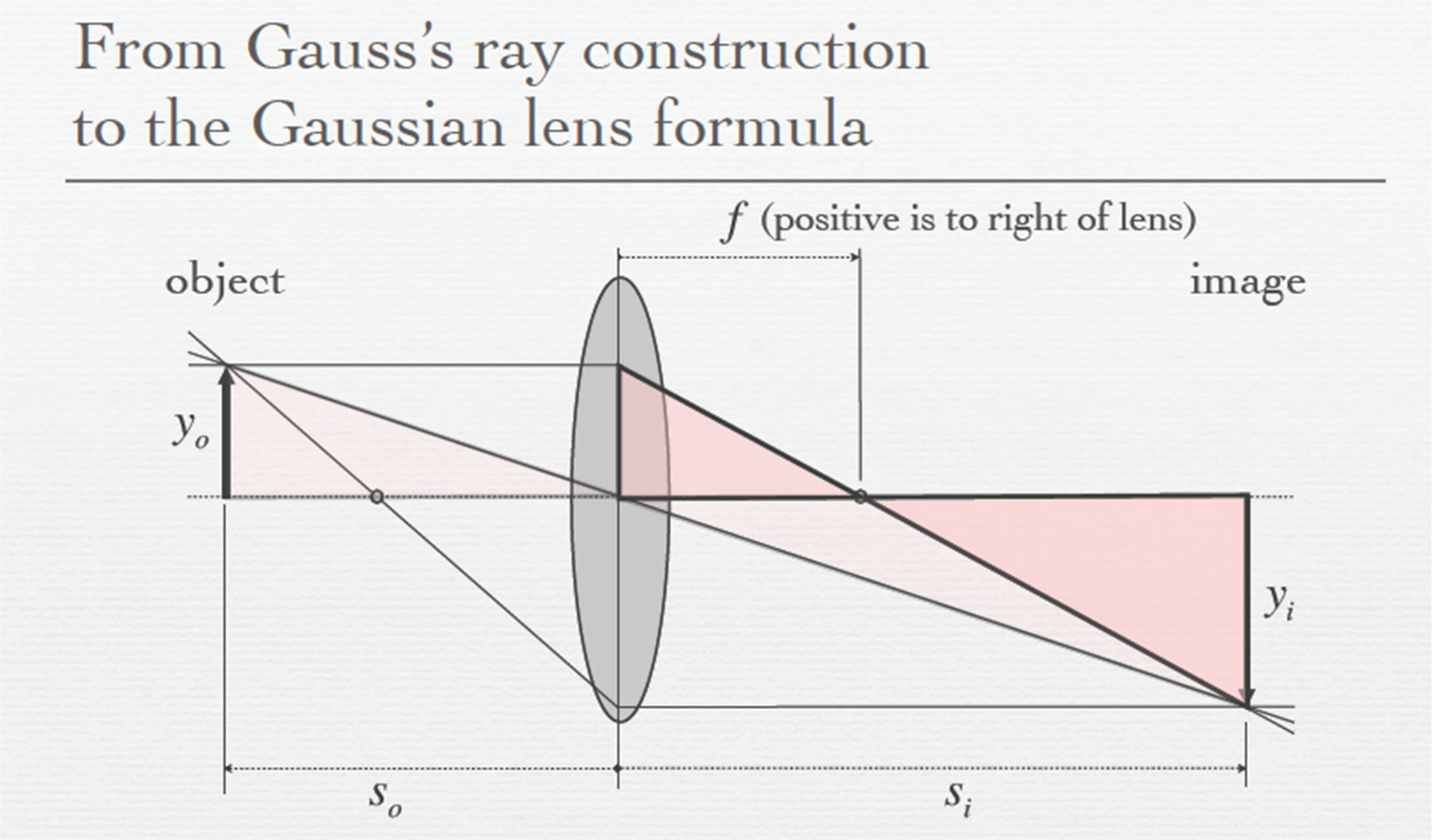
Can we verify the thin lens model?







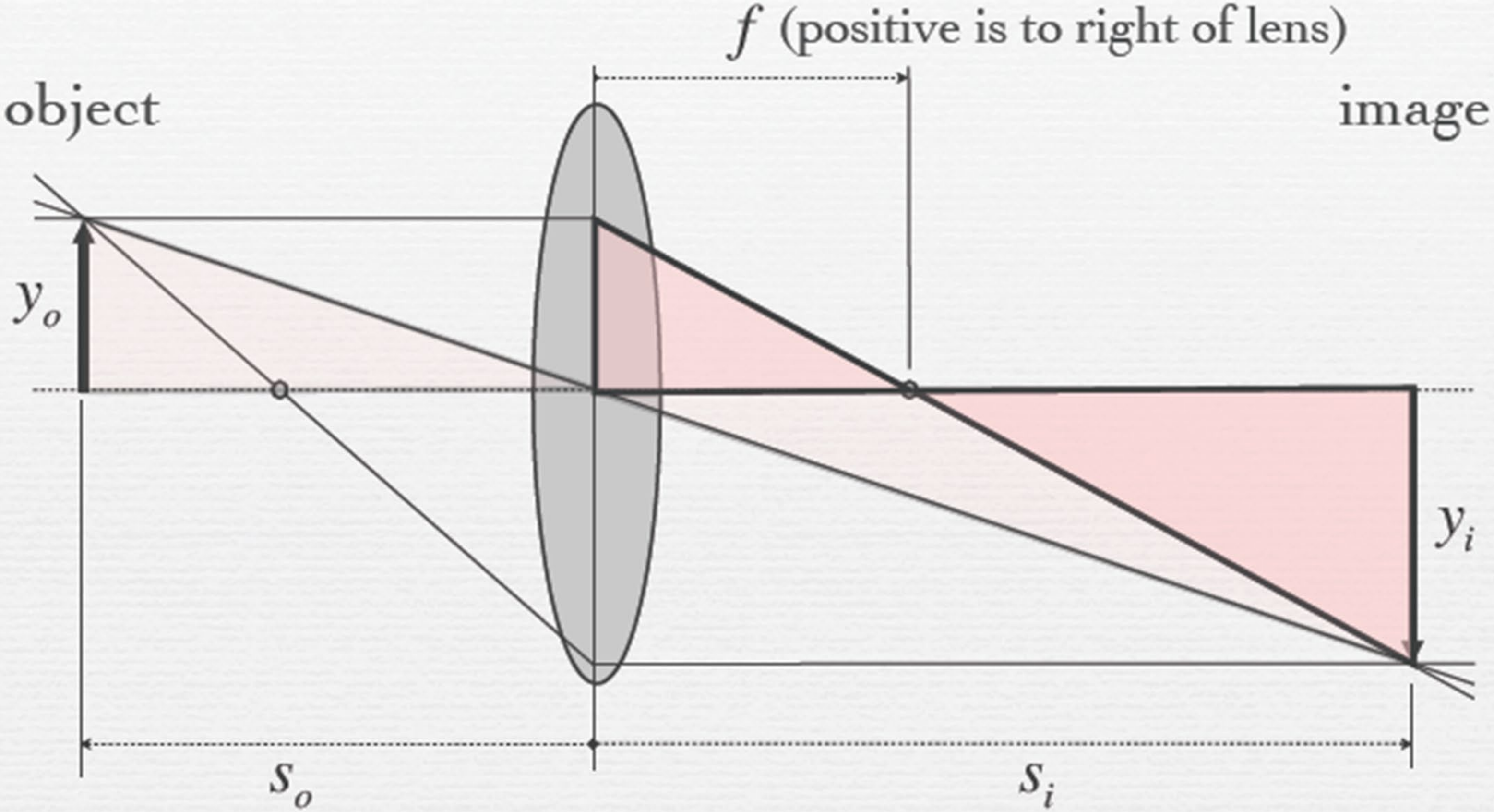
Exercise: Derive Relationship between s_o , s_i , f



Exercise: Derive Relationship between s_o , s_i , f

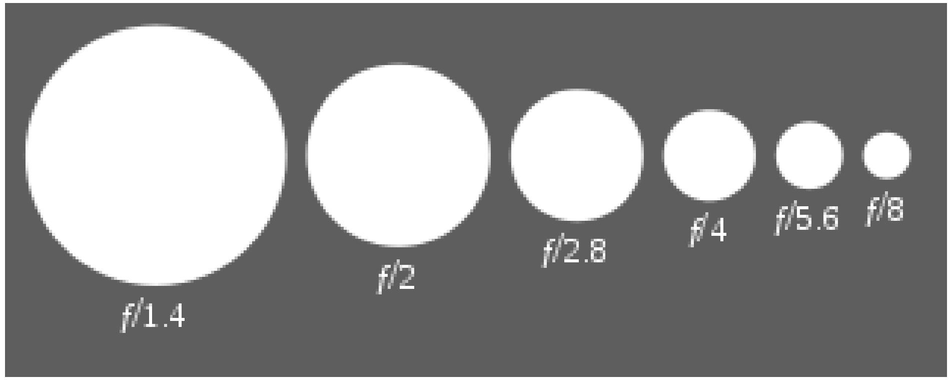
Hint: Similar Triangles

From Gauss's ray construction to the Gaussian lens formula



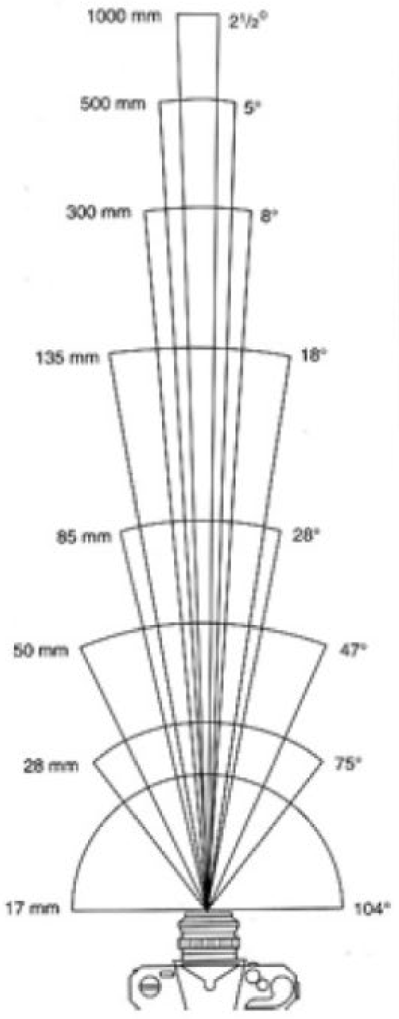
$$\frac{|y_i|}{y_o} = \frac{s_i}{s_o} \quad \text{and} \quad \frac{|y_i|}{y_o} = \frac{s_i - f}{f} \quad \ldots\ldots \quad \frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}$$

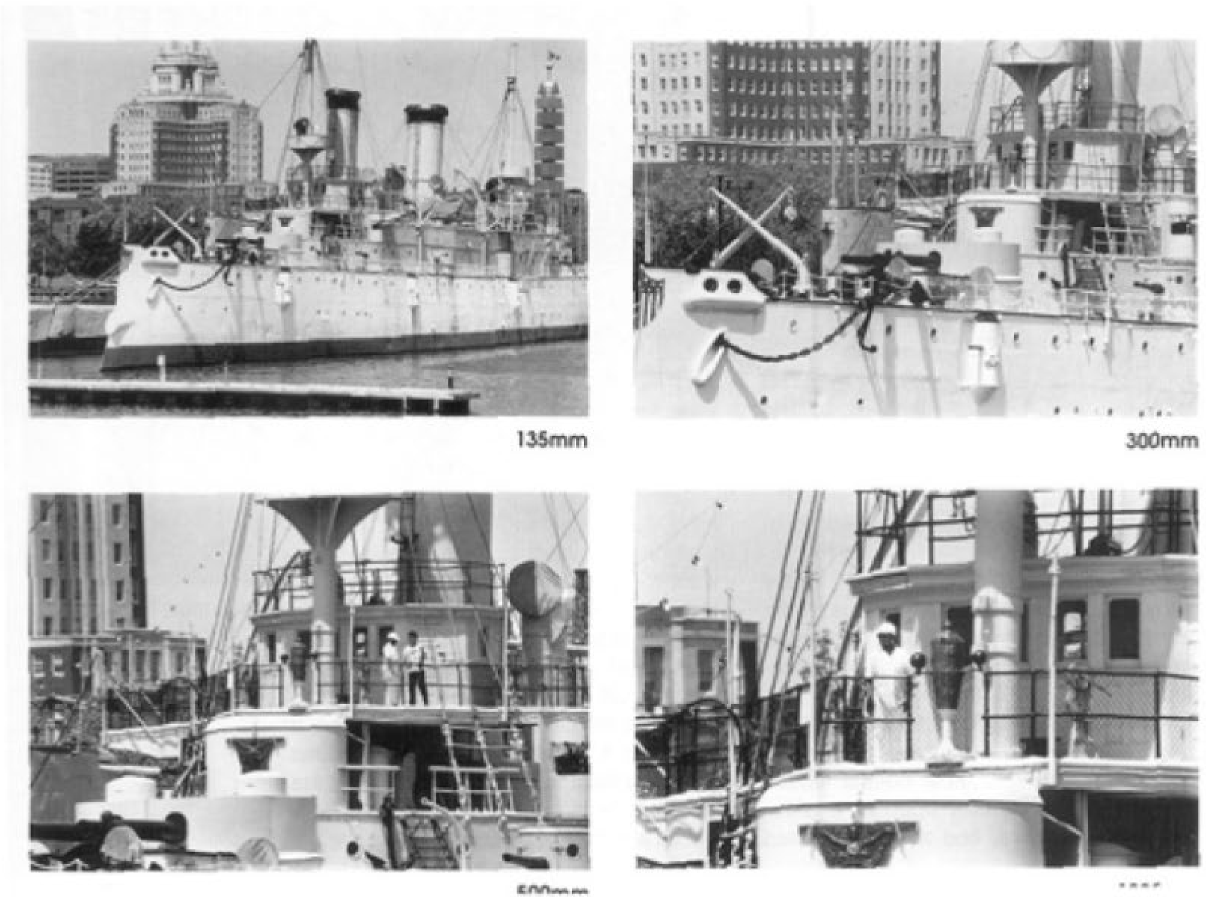
Depth of Field **(effect of varying aperture diameter)**



Smaller aperture: larger DoF

Field of View

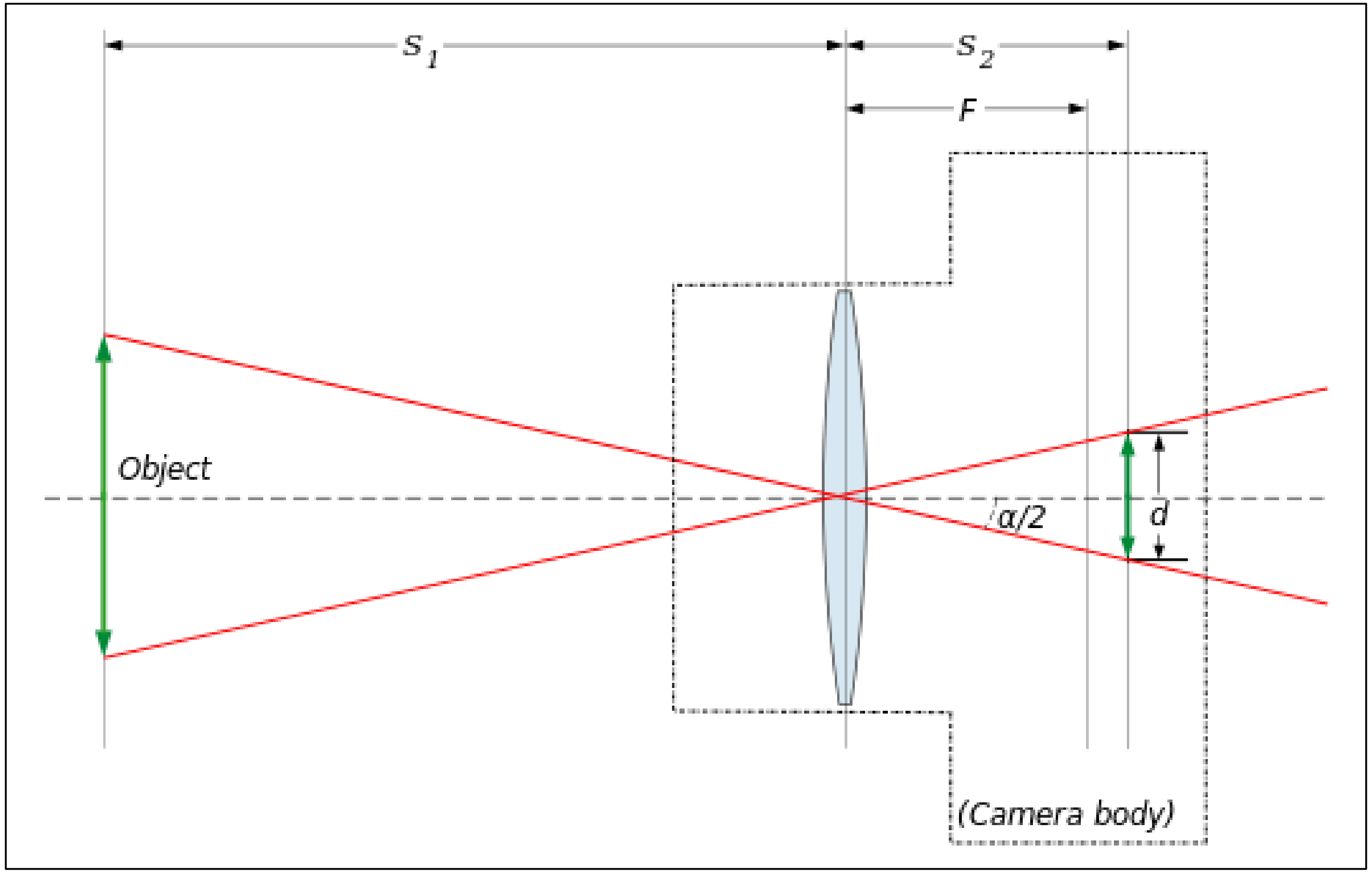




What does FOV depend on?

Slide by A. Efros

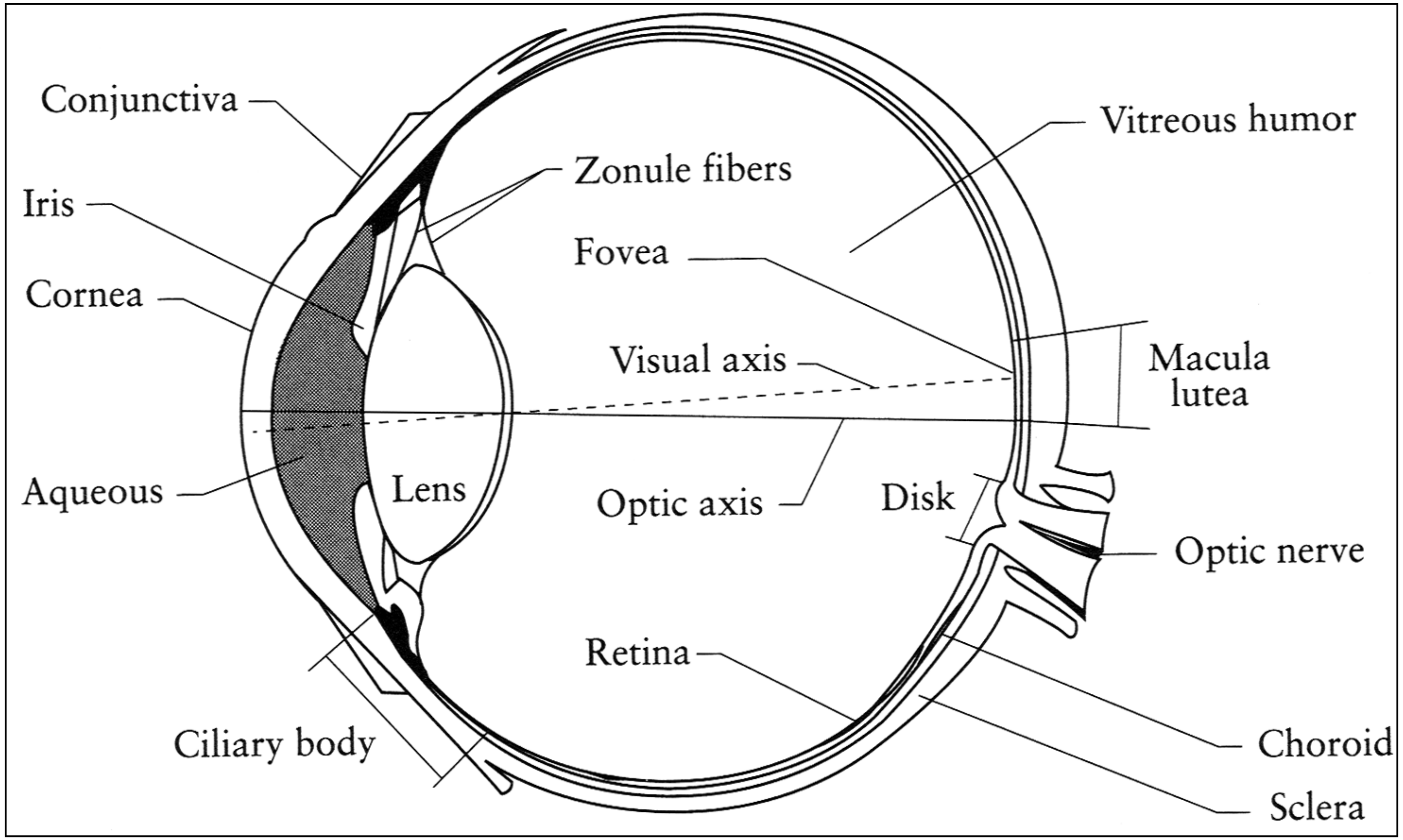
Field of View **(effect of varying focal length)**



Smaller f → larger DoF

$$\alpha = 2 \arctan \frac{d}{2f}$$

The Eye is a Camera



- Iris
  - colored annulus with radial muscles
- Pupil
  - the hole (aperture)
  - size is controlled by the iris
- What’s the “film”?


Disposable cameras are fun,
although it does seem wasteful
- YEAH, IT'S EITHER TAPING
OR CALLING.
And you don't ever
get to see your pictures.
ALSO, HOW DO CAMERAS WORK?

Digital Images

Subjective terms to describe color

Hue
Name of the color
(yellow, red, blue, green, . . .)

Value/Lightness/Brightness
How light or dark a color is.

Saturation/Chroma/Color Purity
How “strong” or “pure” a color is.

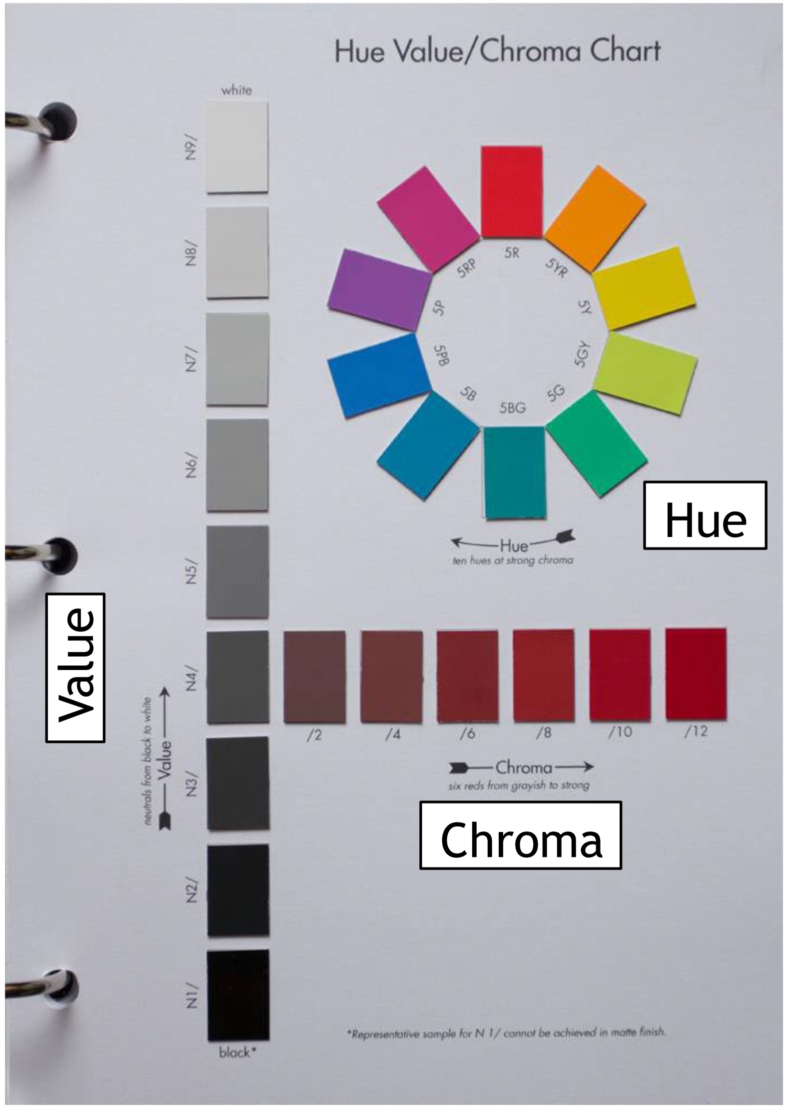
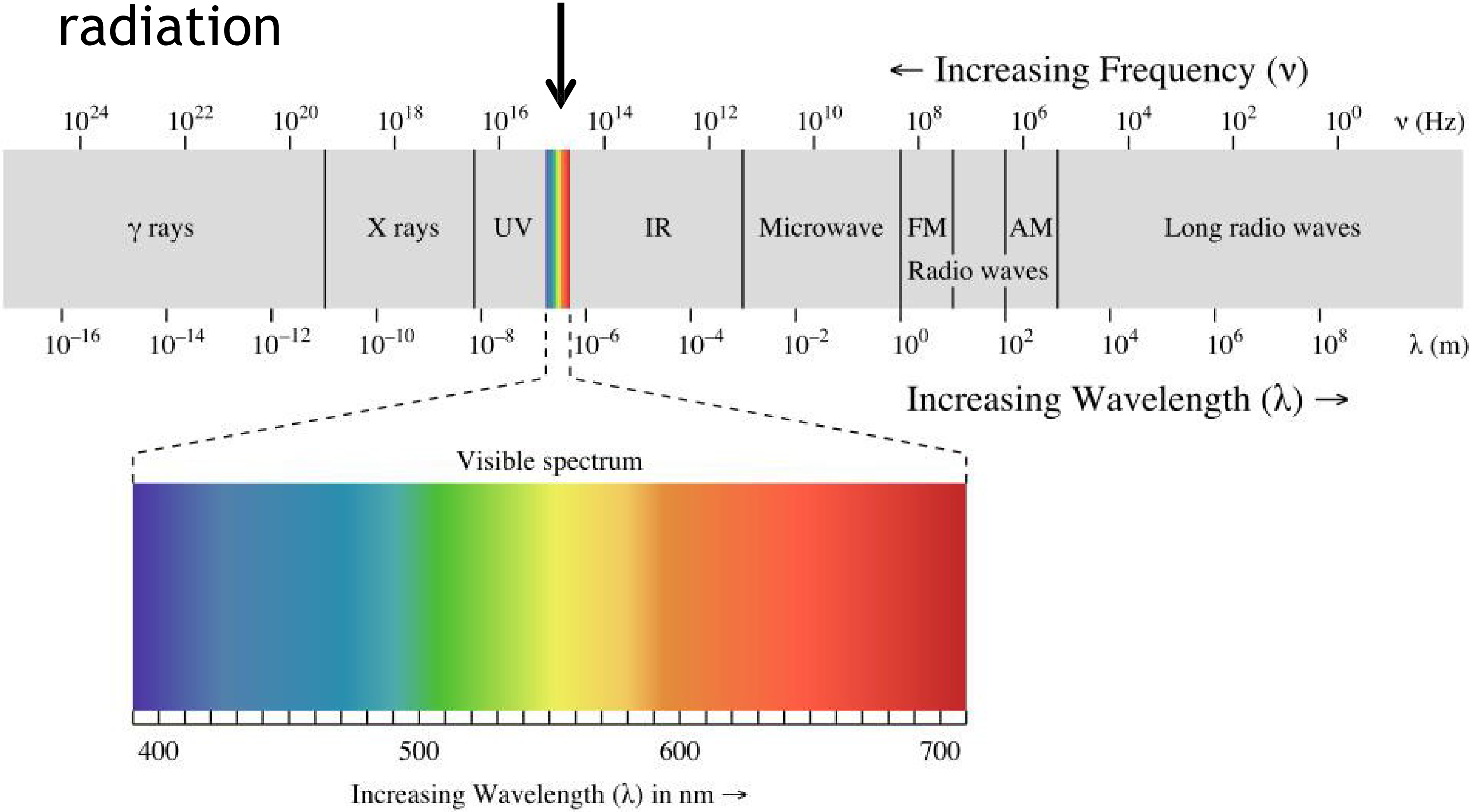


Image from Benjamin Salley
A page from a Munsell Student Color Set

Where do "color sensations" come from?

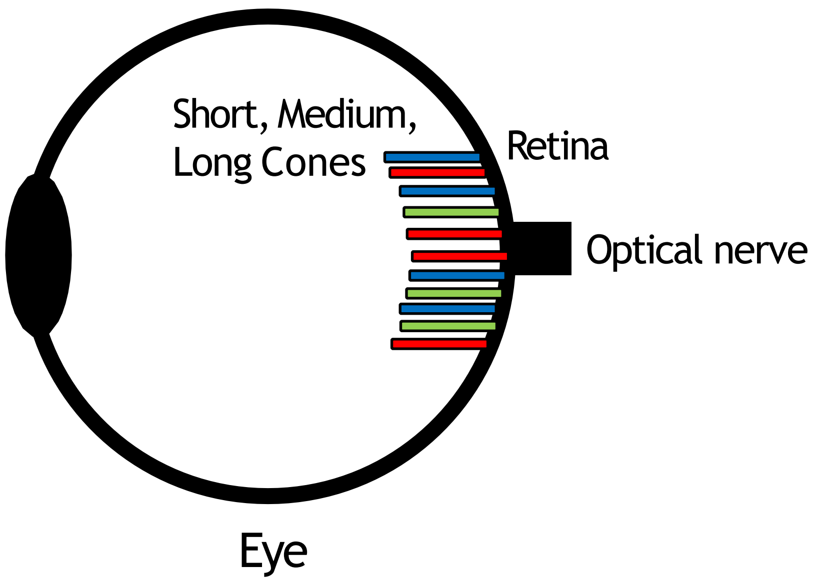
A *very* small range of electromagnetic radiation

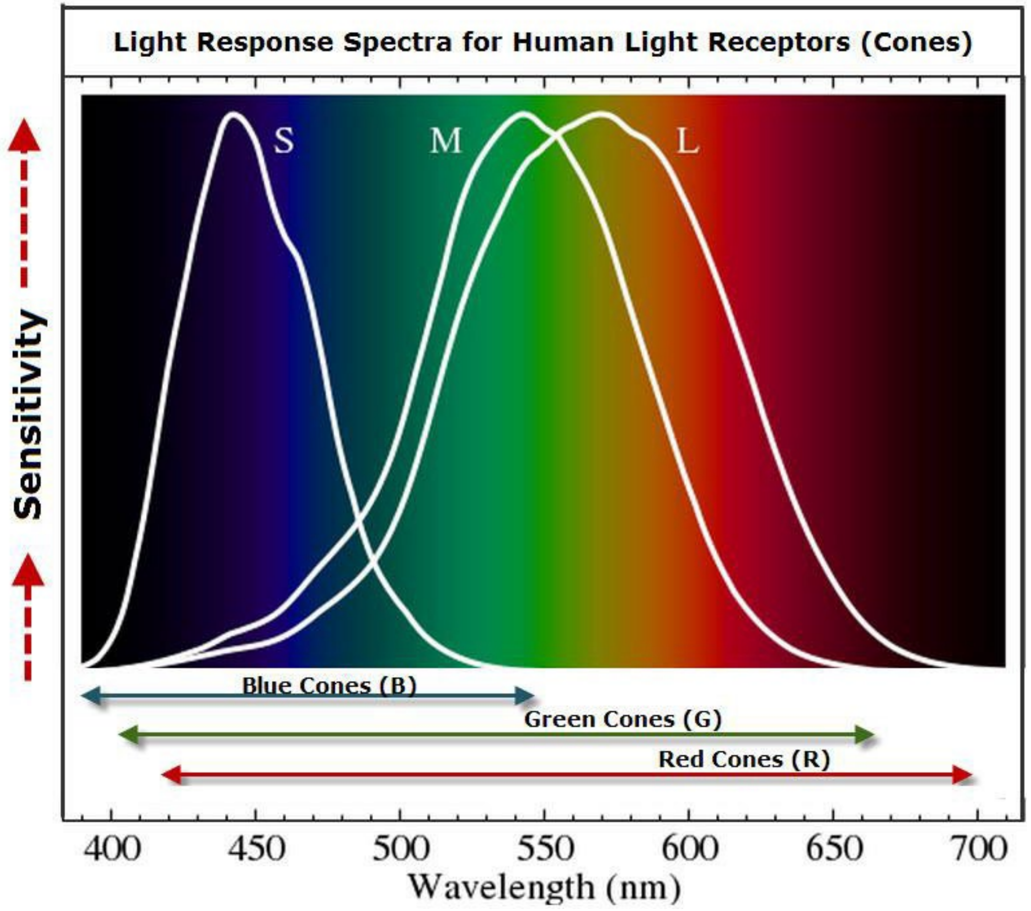


Generally, wavelengths from 380 to 720nm are visible to most individuals

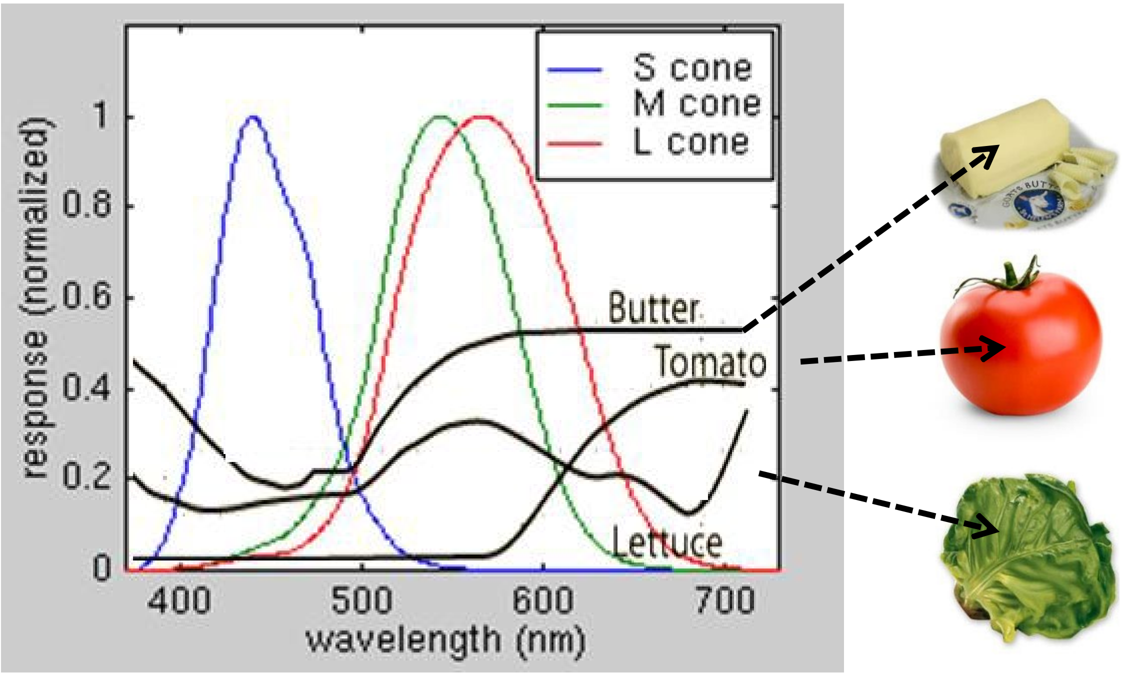
Biology of color sensations

- Our eye has three receptors (cone cells) that respond to visible light and give the sensation of color





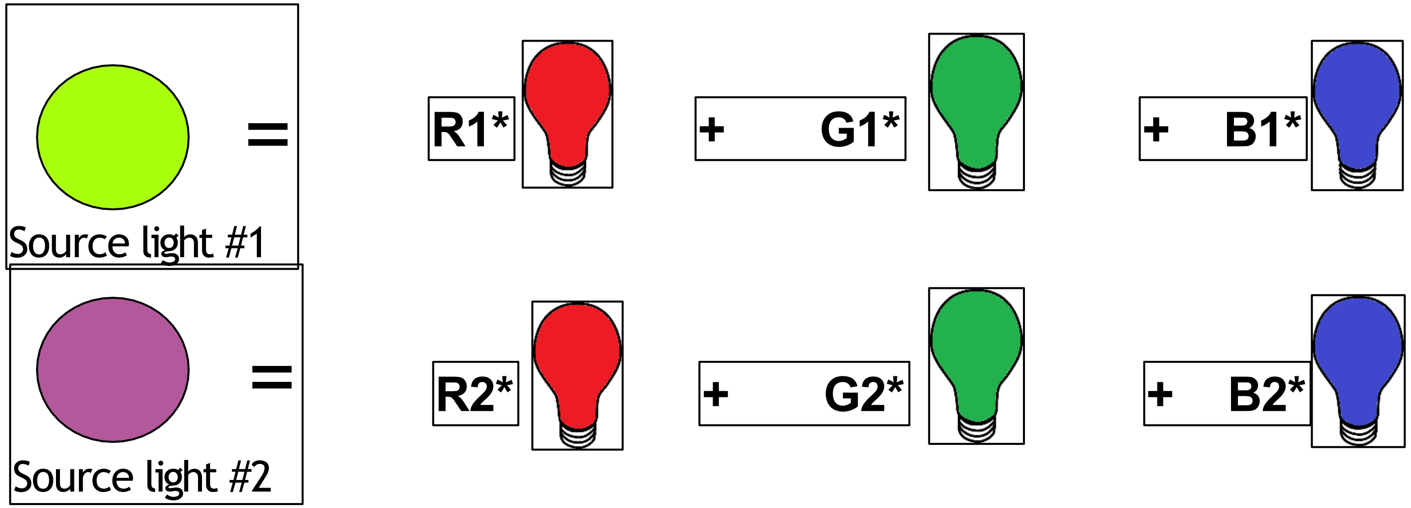
Spectral power distribution (SPD)



- We rarely see monochromatic light in real world scenes
- Instead, objects reflect a wide range of wavelengths.
- This can be described by a *spectral power distribution* (SPD)
- The SPD plot shows the relative amount of each wavelength reflected over the visible spectrum.

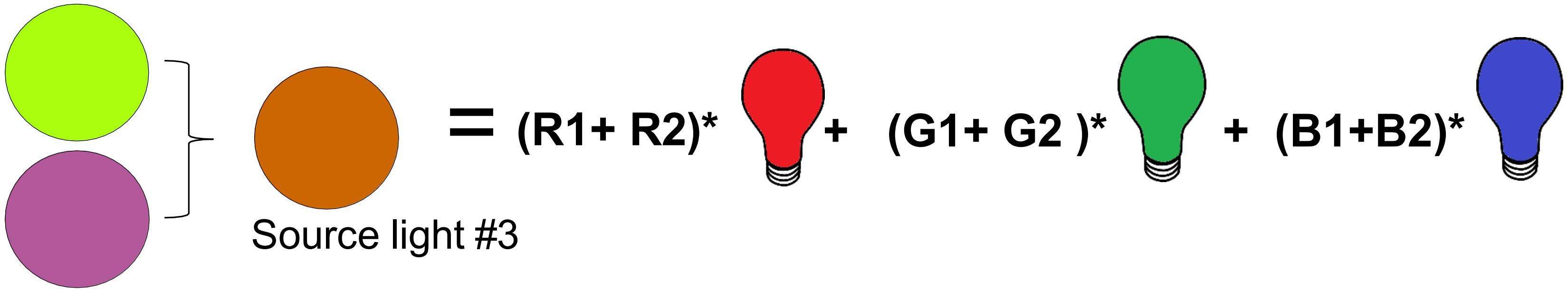
Tristimulus color theory (Grassman's Law)

Source color can be matched by **a linear combination** of three independent "primaries".



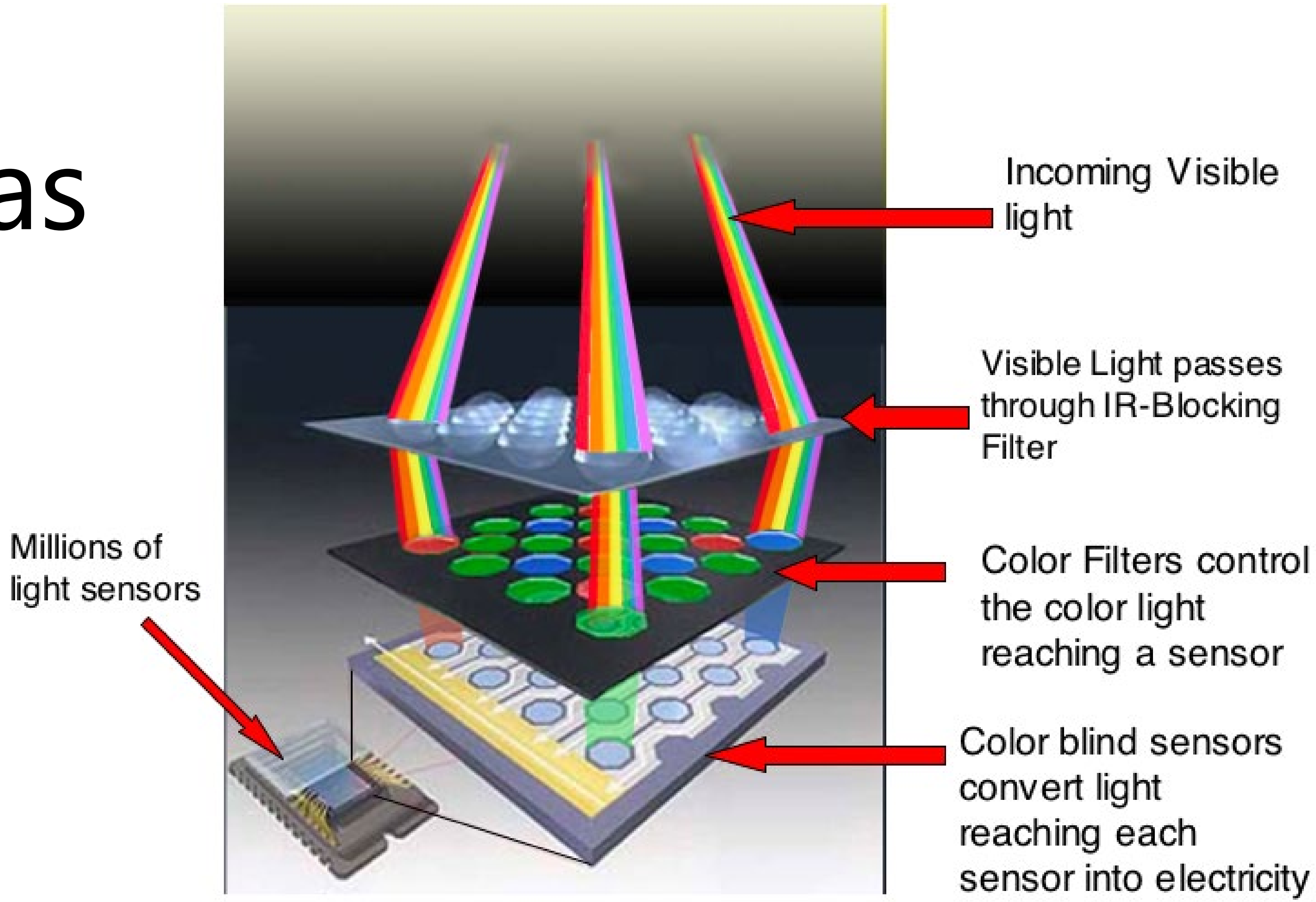
If we combined source lights 1 & 2 to get a new source light 3

The amount of each primary needed to match the new light #3 is the sum of the weights that matched lights sources #1 & #2.

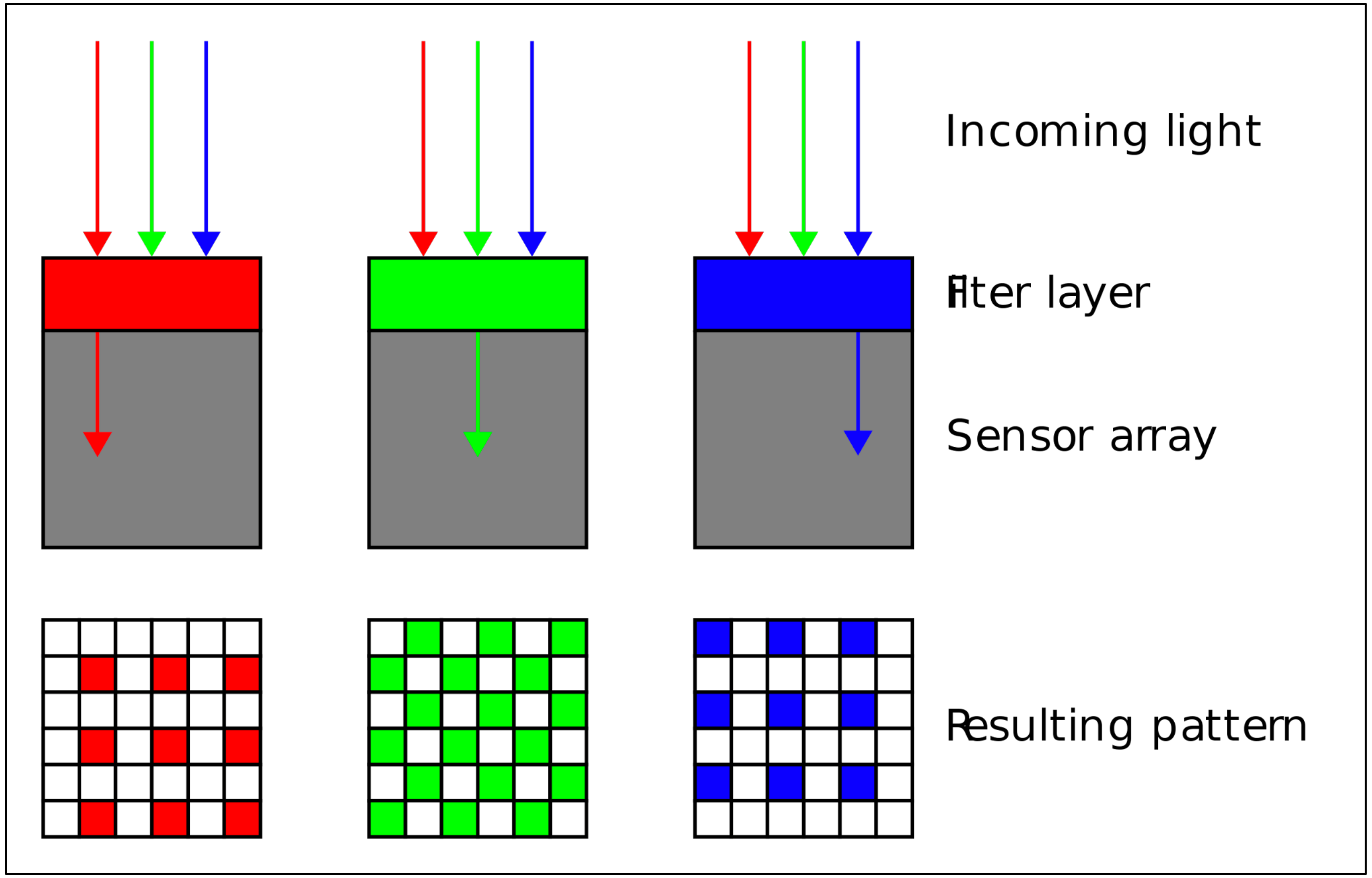


This may seem obvious now, but discovering that light obeys the laws of linear algebra was a huge and useful discovery.

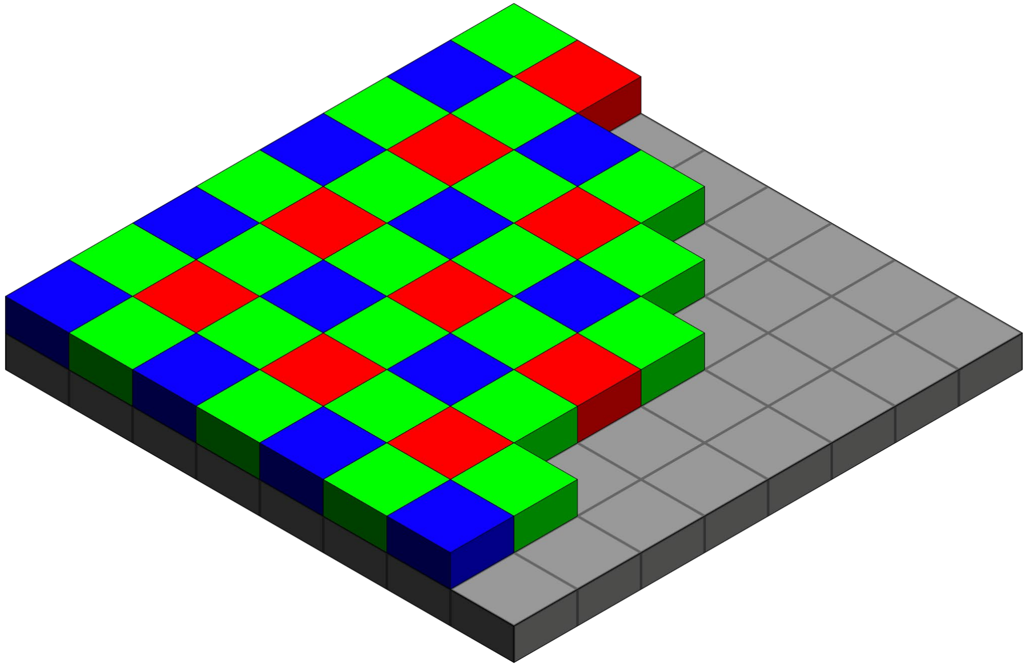
RGB in Cameras



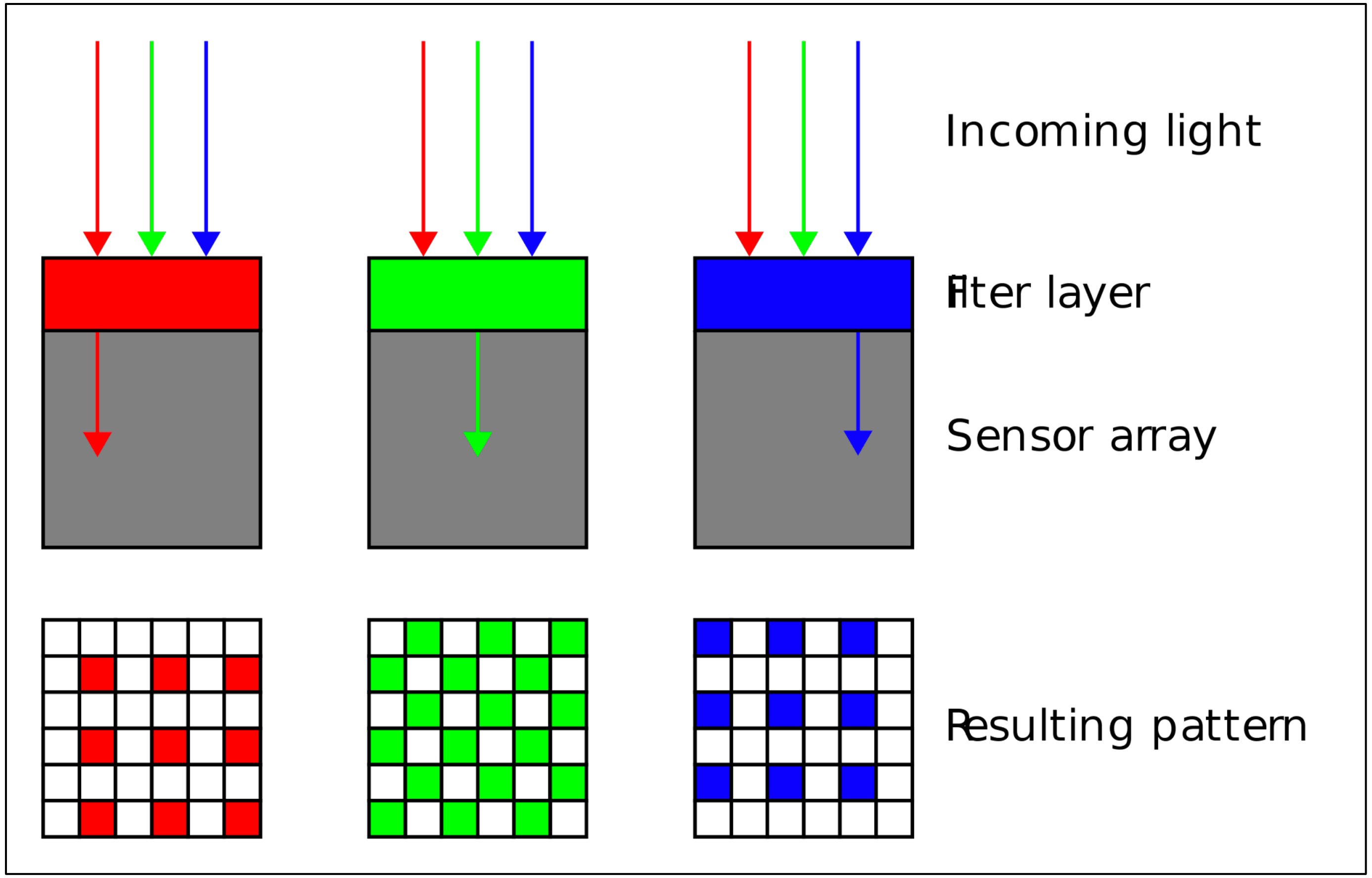
RGB in Cameras – Bayer Pattern



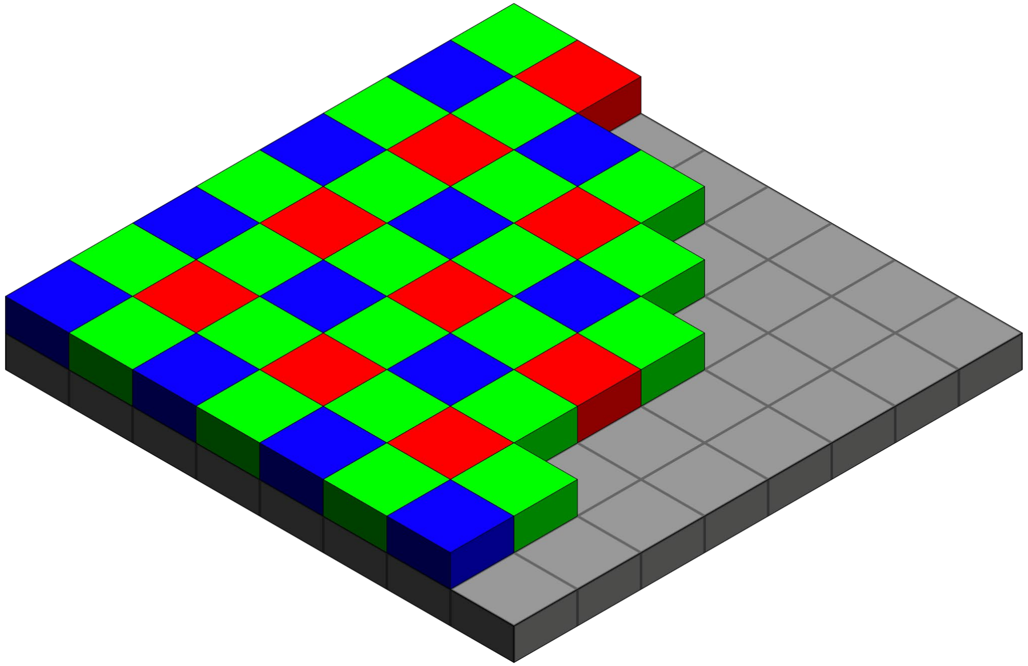
25% pixels see Red
25% pixels see Blue
50% pixels see Green



RGB in Cameras – Bayer Pattern



Then how do we get ***all colors*** at ***all pixels?***



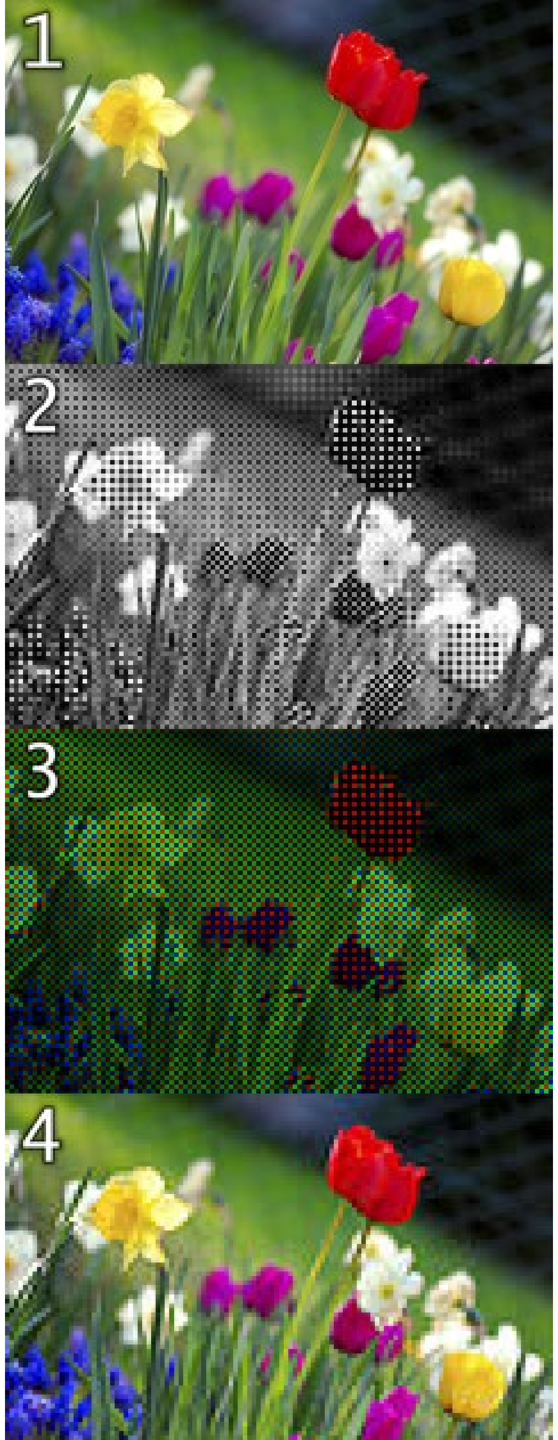
RGB in Cameras – Debayering / Demosaicing

Original (High Resolution)

Bayer (120x80) Intensities

Bayer Color-Coded

After Interpolation



How? → Interpolation !

Method 1: nearest-neighbor interpolation

- For each pixel, for the missing channel, assign the value of the closest pixel with that channel available

Method 2: Bi-Linear Interpolation

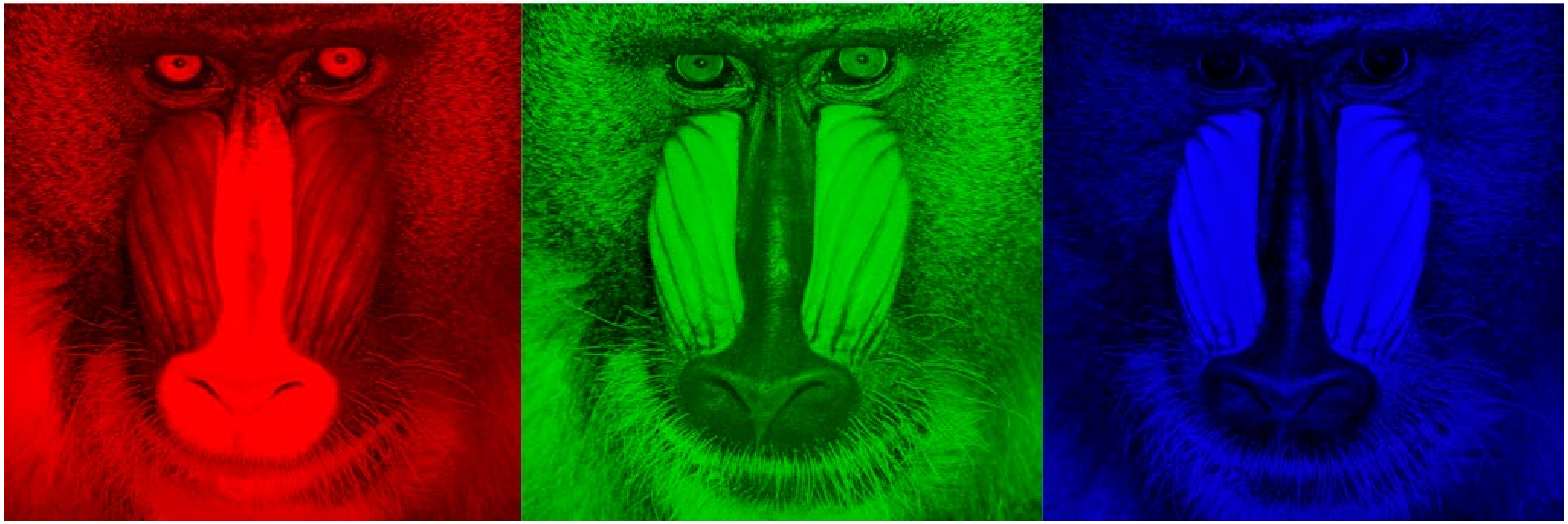
- Red-value of a non-red pixel
  = avg of 2 or 4 adjacent reds
- Similar for green and blue

More Advanced Methods ...

Finally !
Digital RGB images!

What we see

What the camera stores

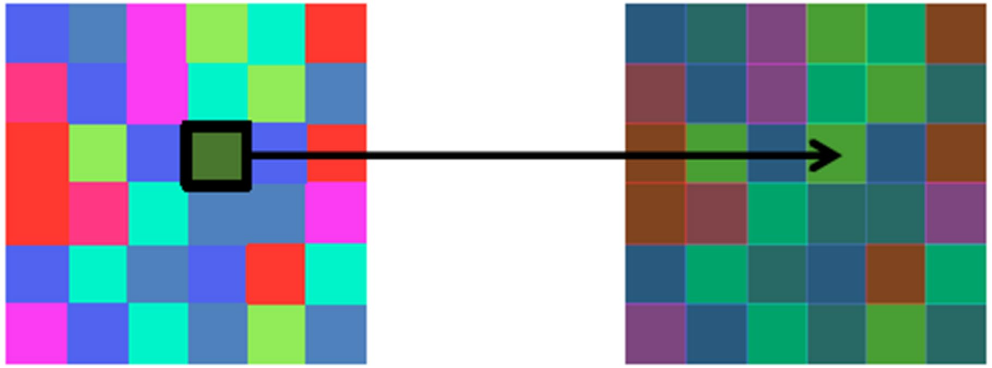


Computer Vision

"understanding" the visual world by processing (RGB) images

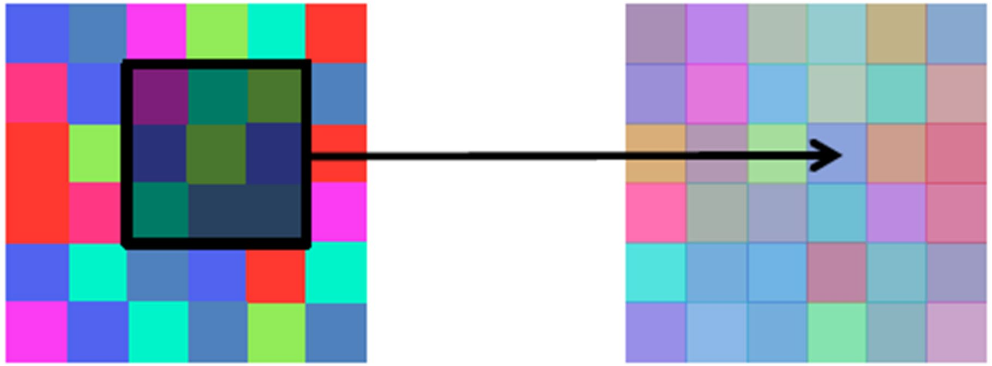
Point Processing vs Image Filtering

Point Operation



point processing

Neighborhood Operation



“filtering”

Examples of point processing

How would you implement these?

original

darken

lower contrast

non-linear lower contrast

How would you implement these?

Examples of point processing

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

non-linear lower contrast

$\left(\frac{x}{255}\right)^{1/3} \times 255$

How would you implement these?

Examples of point processing

original

$$x$$

darken

$$x - 128$$

lower contrast

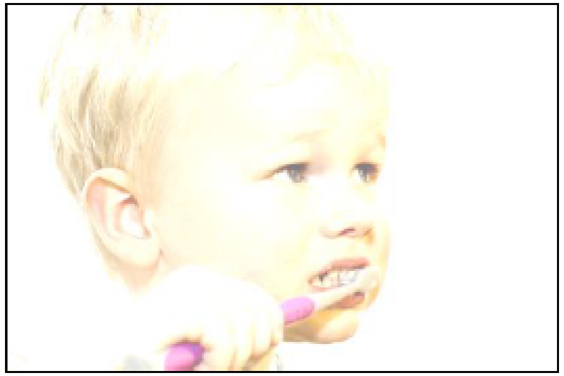
$$\frac{x}{2}$$

non-linear lower contrast

$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert

lighten



raise contrast

non-linear raise contrast

How would you implement these?

Examples of point processing

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

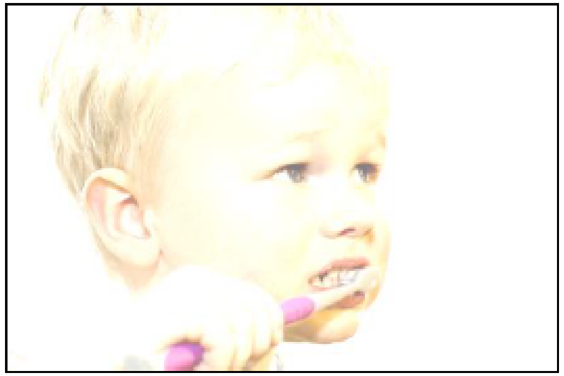
non-linear lower contrast

$\left(\frac{x}{255}\right)^{1/3} \times 255$

invert

$255 - x$

lighten



$x + 128$

raise contrast

$x \times 2$

non-linear raise contrast

$\left(\frac{x}{255}\right)^{2} \times 255$

Convolution

Convolution for 1D continuous signals

Definition of filtering as convolution:

notice the flip

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)dy$$

filtered signal

filter

input signal

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2)$$

Convolution for 1D ***discrete*** signals

Definition of filtering as convolution:

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2)$$

1D Convolution. Example

Suppose our input 1D image is:

$f =$	10	50	60	10	20	40	30

and our kernel is:

$g =$	1/3	1/3	1/3

Let's call the output image h. What is the value of $h(3)$?

1D Convolution. Example

Suppose our input 1D image is:

$$f = \boxed{10 \mid 50 \mid 60 \mid 10 \mid 20 \mid 40 \mid 30}$$

and our kernel is:

“Box” Filter that causes “Blur” or “Smoothing”

$$g = \boxed{1/3 \mid 1/3 \mid 1/3}$$

Let’s call the output image h. What is the value of $h(3)$?

$$h = \boxed{20 \mid 40 \mid 40 \mid 30 \mid 20 \mid 30 \mid 23.333}$$

Convolution for 2D discrete signals

Definition of filtering as convolution:

notice the flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x - i, y - j)$$

filtered image

filter

input image

Convolution for 2D discrete signals

Definition of filtering as convolution:

notice the flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i,j) I(x-i, y-j)$$

filtered image   filter   input image

If the filter $f(i,j)$ is non-zero only within $-1 \leq i, j \leq 1$, then

$$(f * g)(x, y) = \sum_{i,j=-1}^{1} f(i,j) I(x-i, y-j)$$

The kernel we saw earlier is the 3x3 matrix representation of $f(i,j)$.

3	5	2	8	1
9	7	5	4	3
2	0	6	1	6
6	3	7	9	2
1	4	9	5	1

Convolutional Filter

1	0	0
1	1	0
0	0	1

0	0	1
0	1	1
1	0	0

flipped

What's the output?

Convolution vs correlation

Definition of discrete 2D convolution:

notice the flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x - i, y - j)$$

Definition of discrete 2D correlation:

notice the lack of a flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x + i, y + j)$$

- Most of the time won't matter, because our kernels will be symmetric.
- Will be important when we discuss frequency-domain filtering

Image Convolution Examples

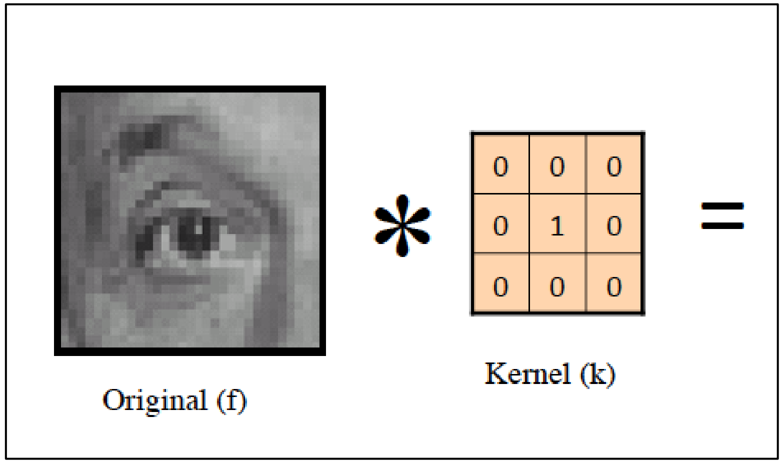


Image Convolution Examples

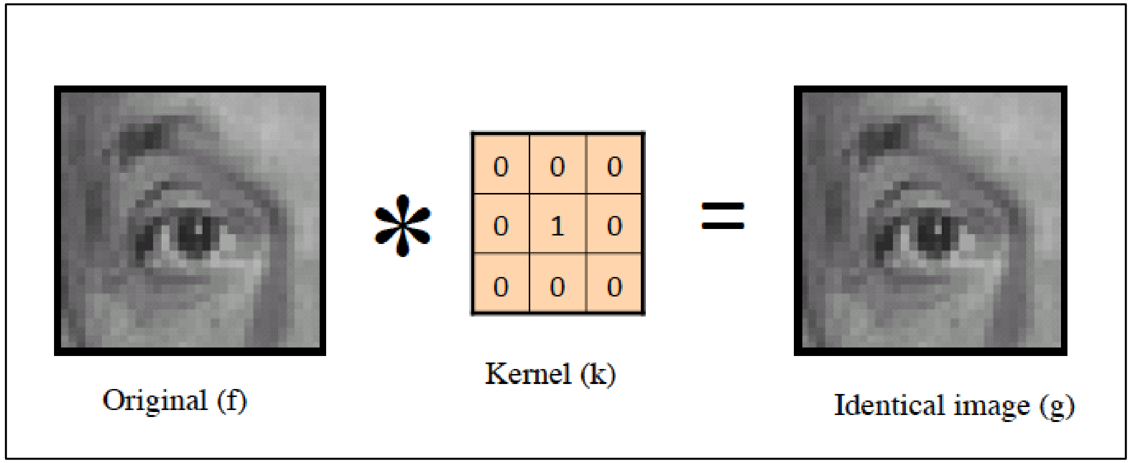
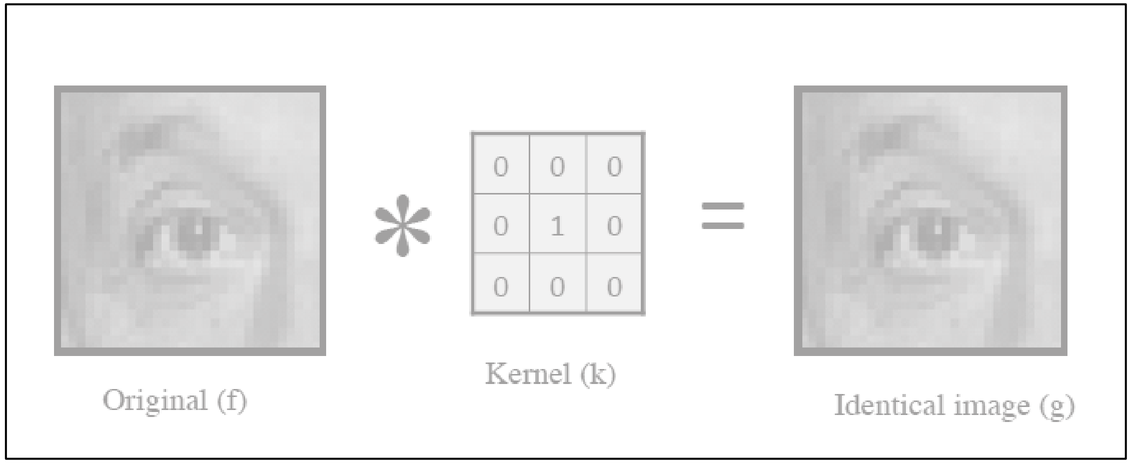


Image Convolution Examples



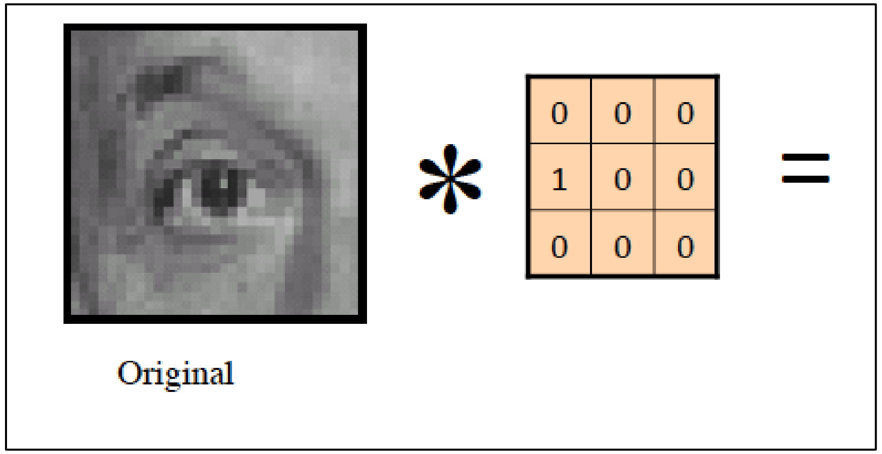
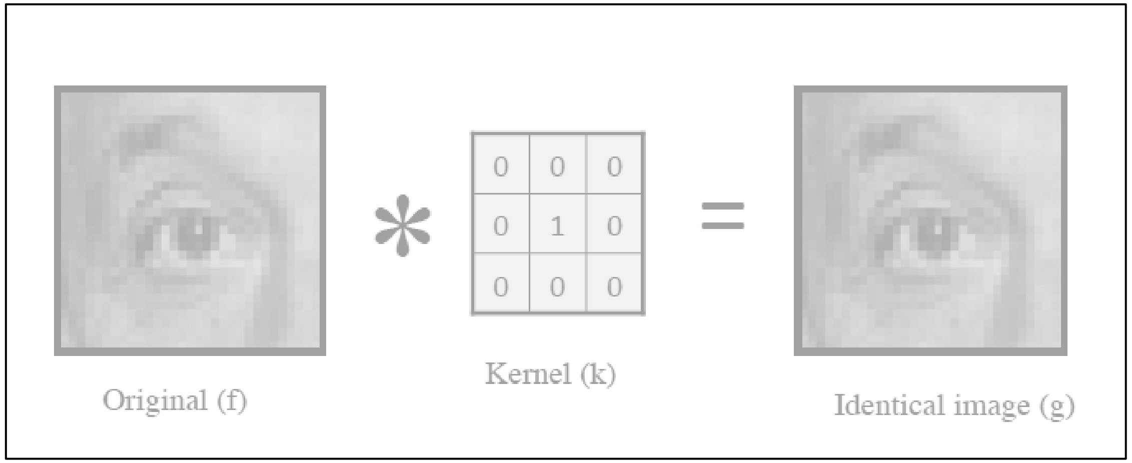


Image Convolution Examples



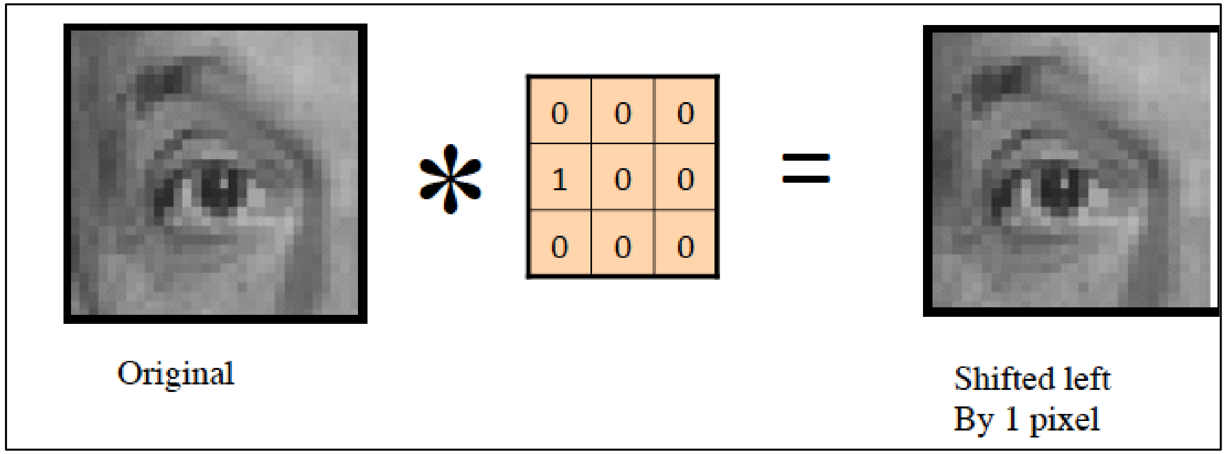
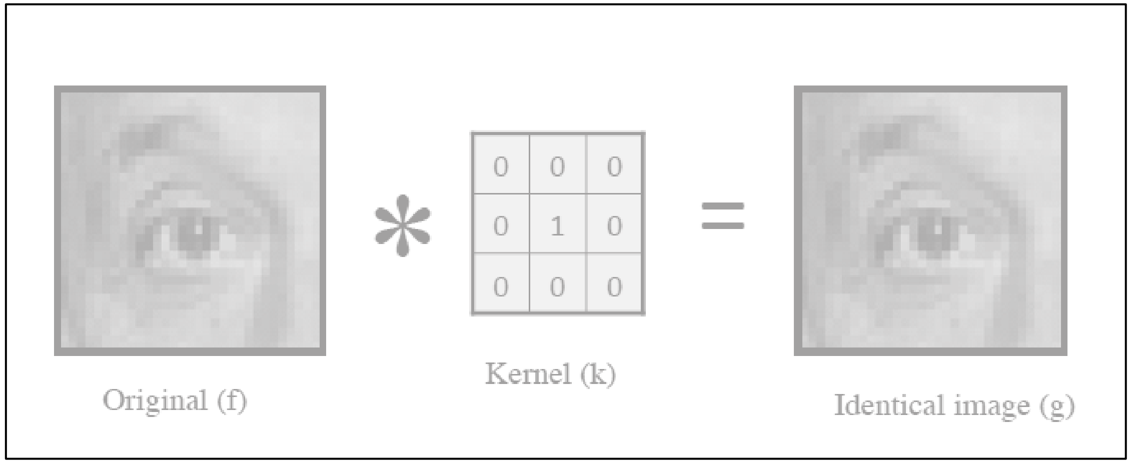
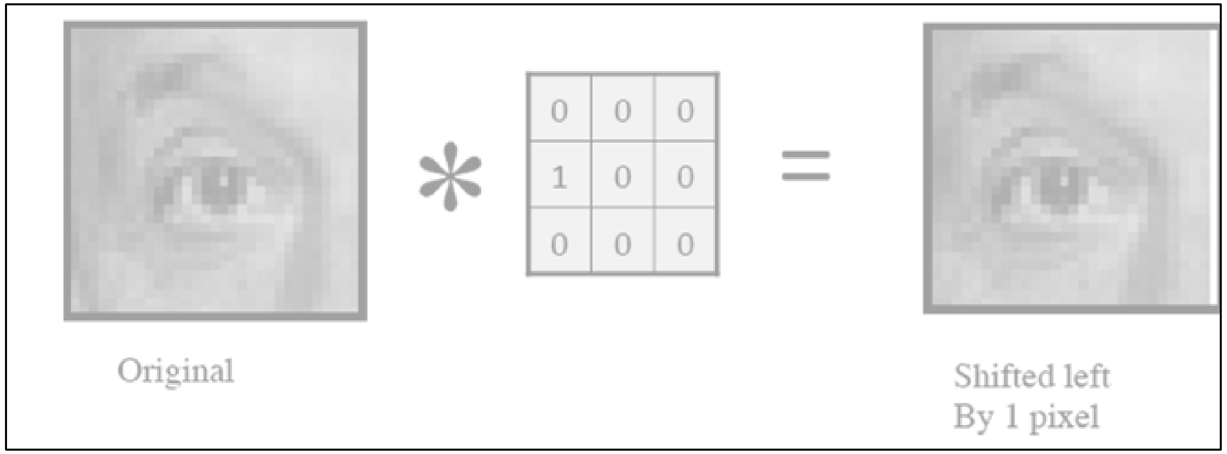


Image Convolution Examples





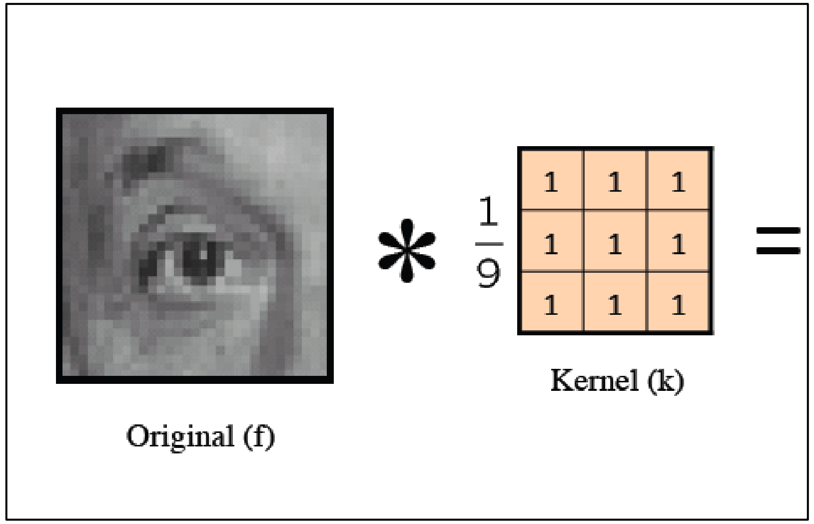
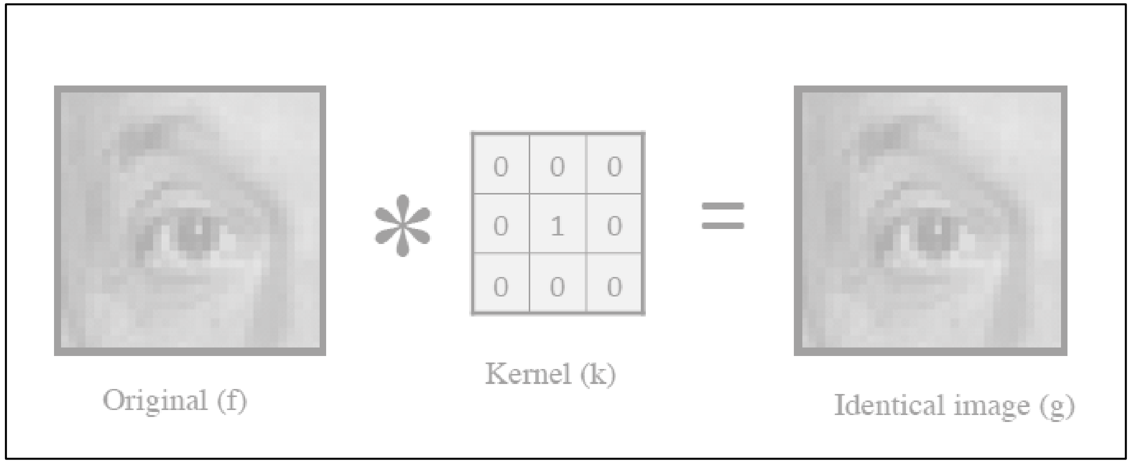
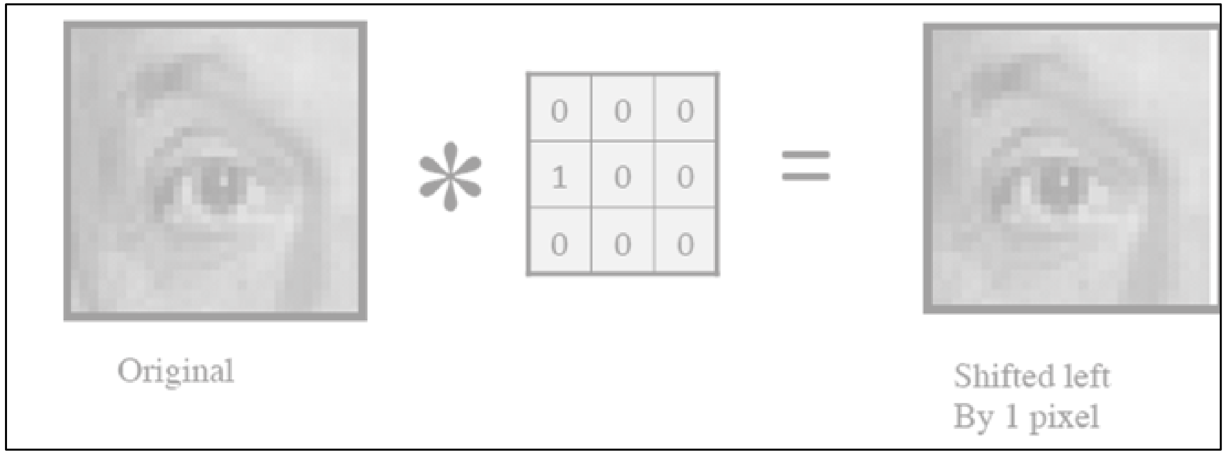


Image Convolution Examples





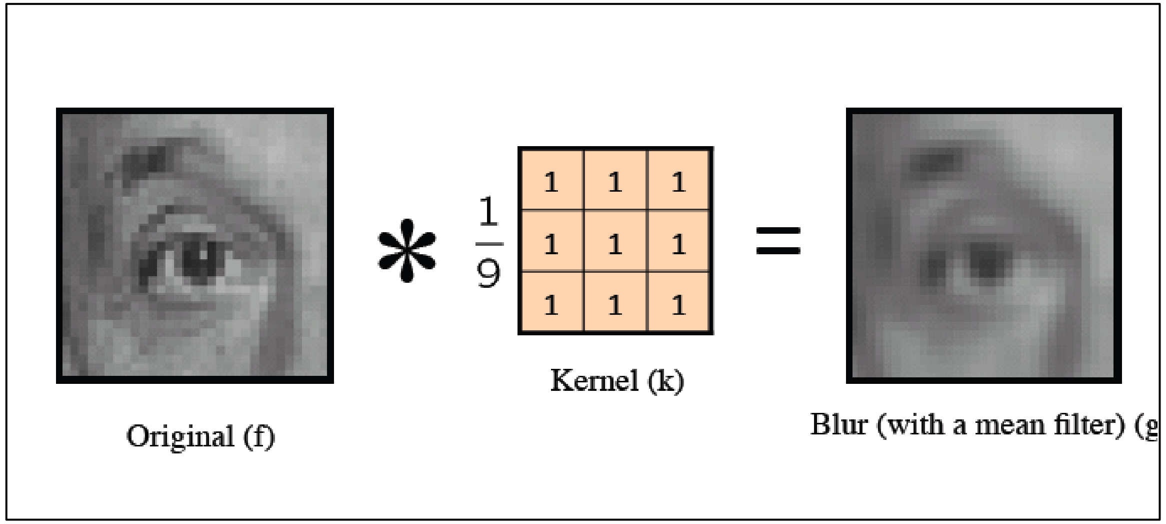
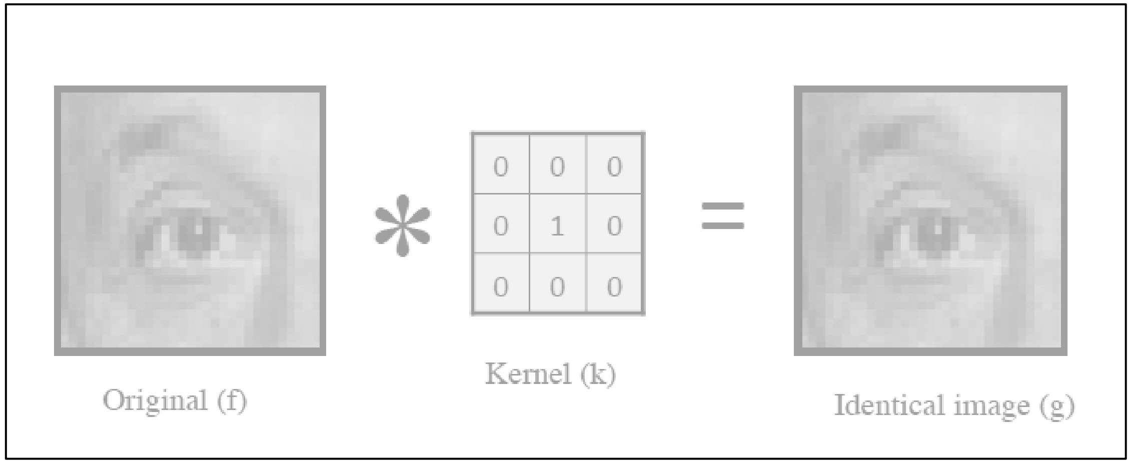
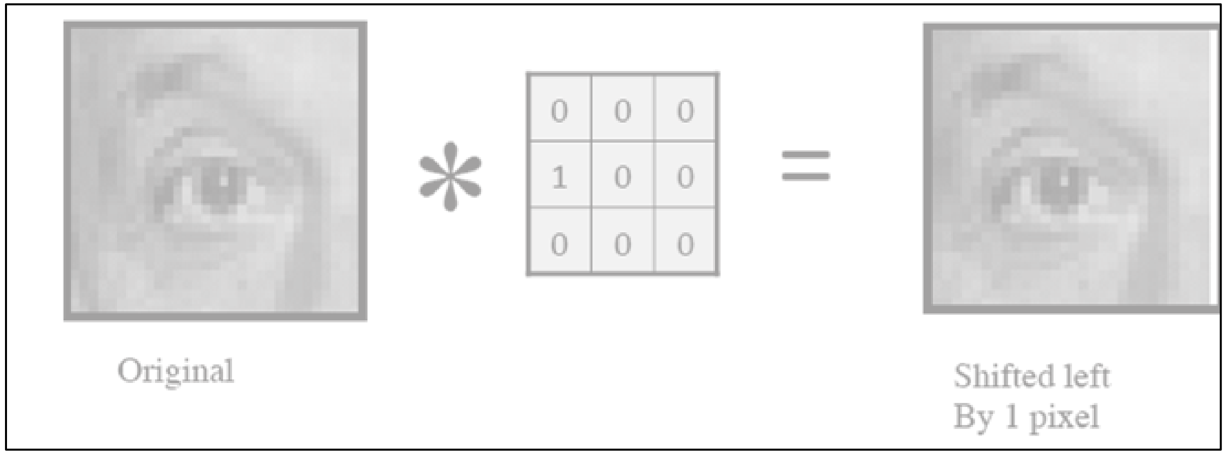
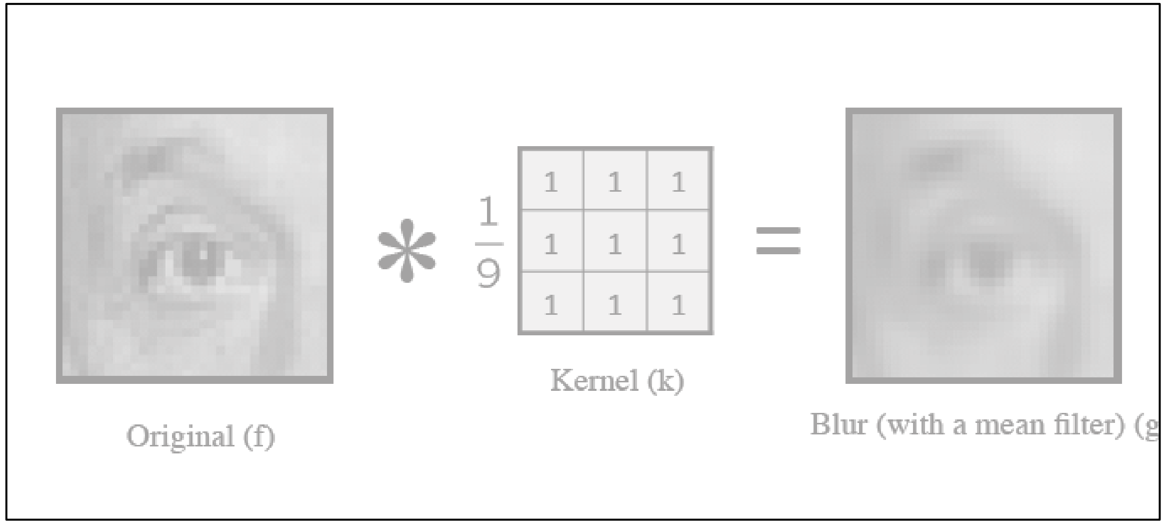


Image Convolution Examples







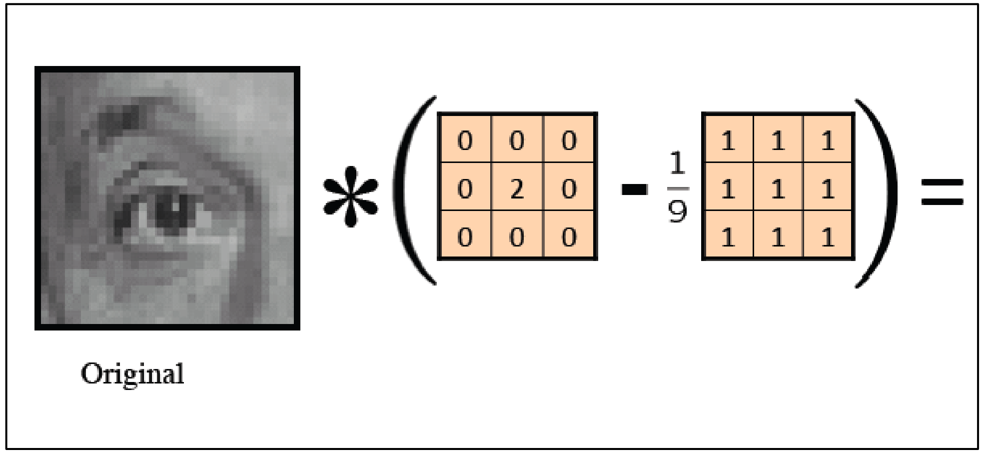
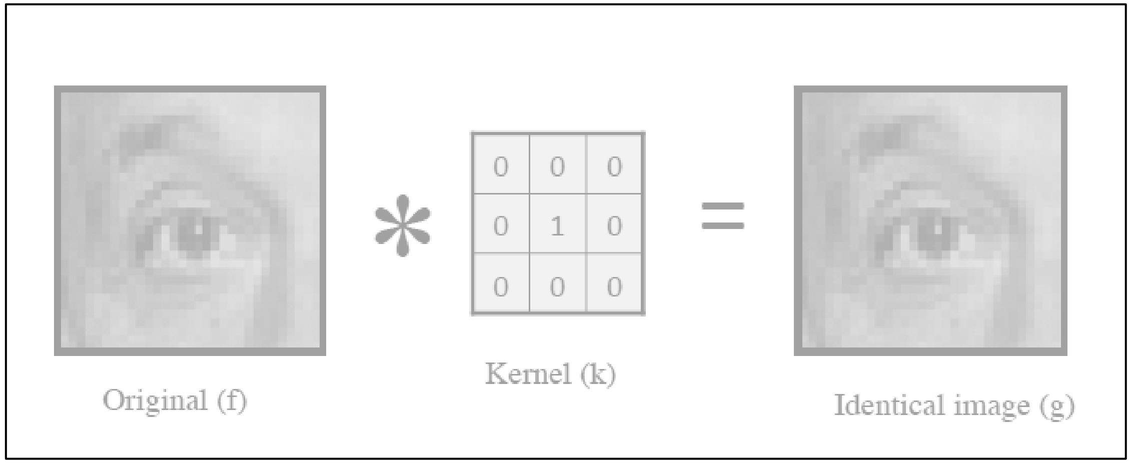
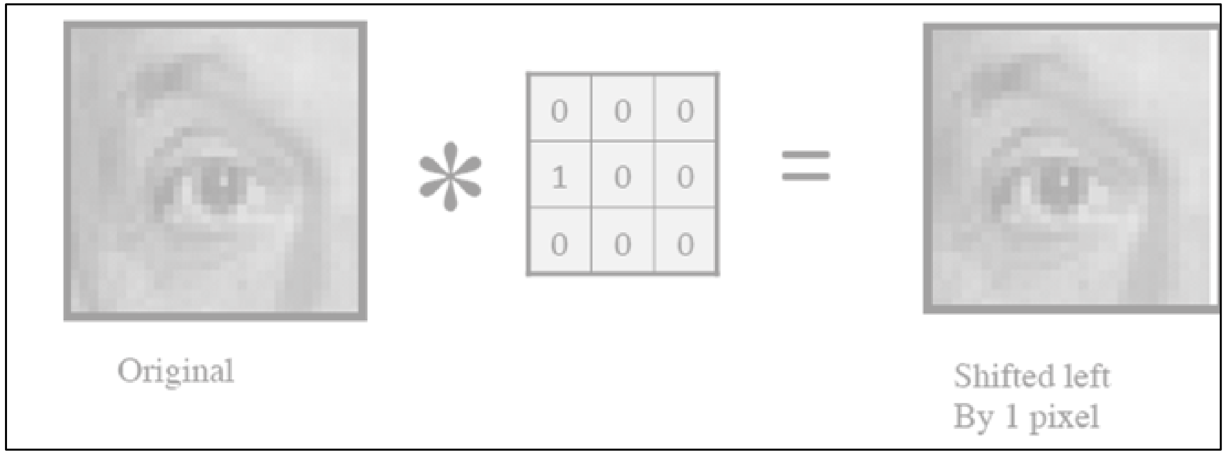
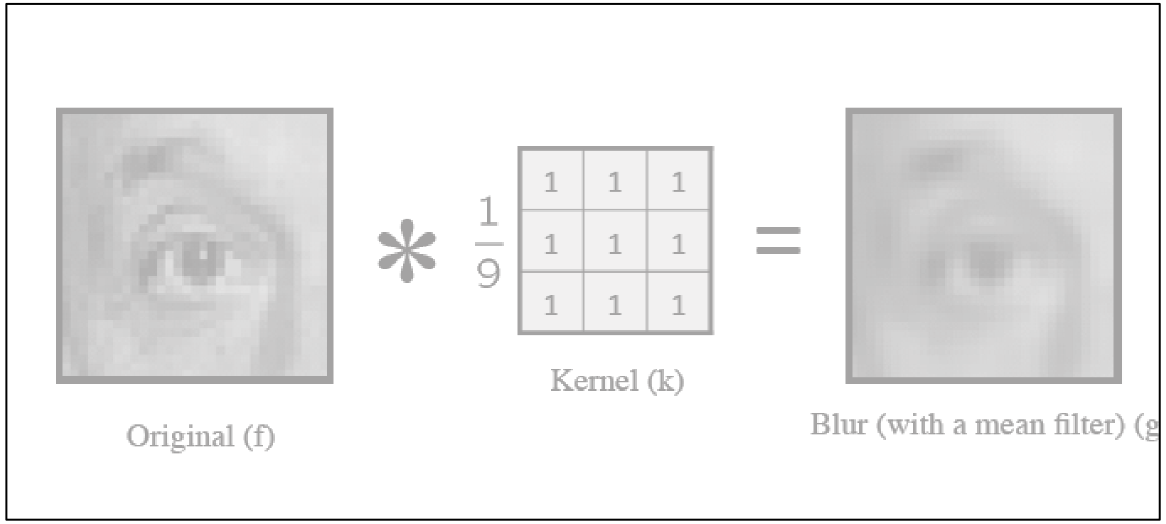
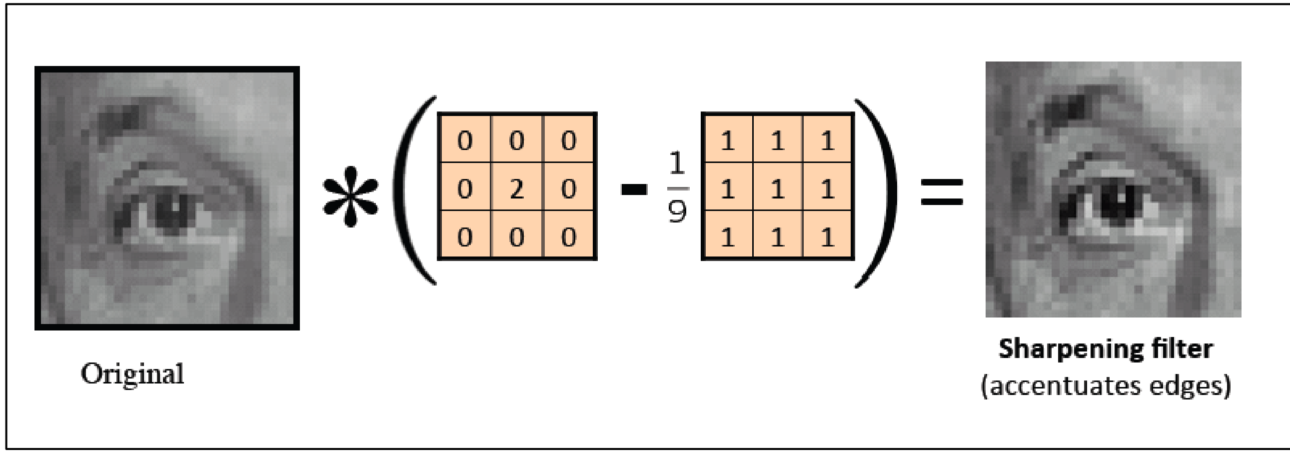


Image Convolution Examples





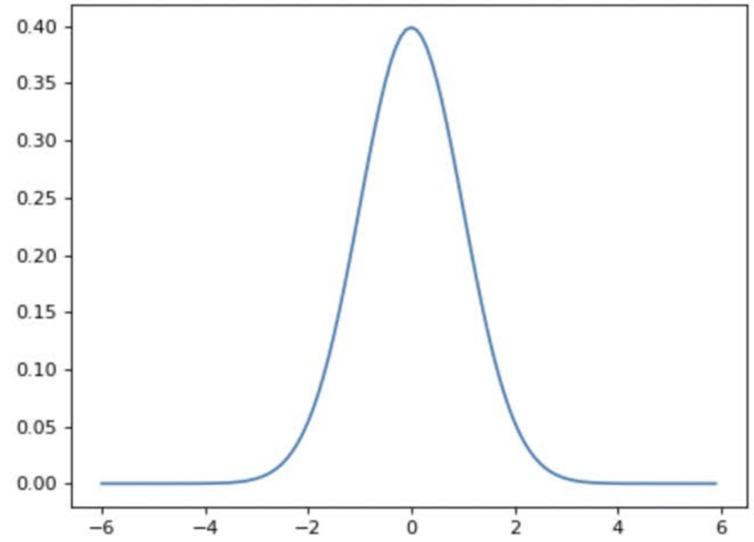


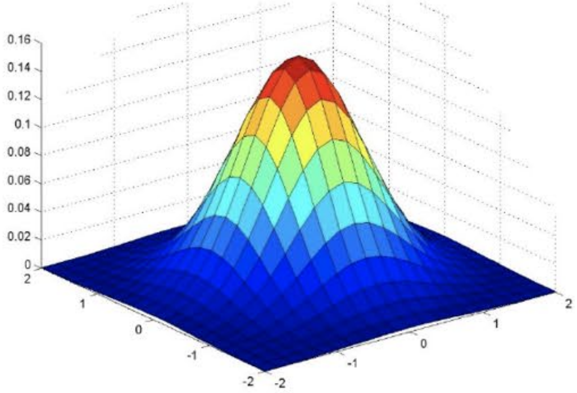


The Gaussian filter

$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

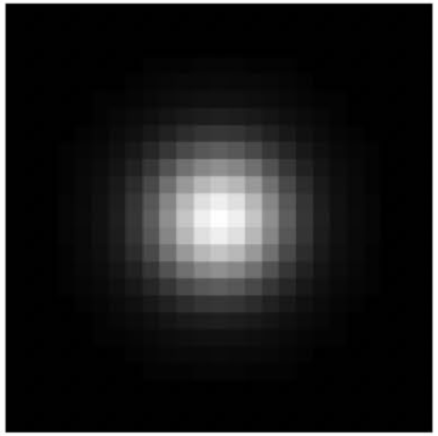
- named (like many other things) after Carl Friedrich Gauss





Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

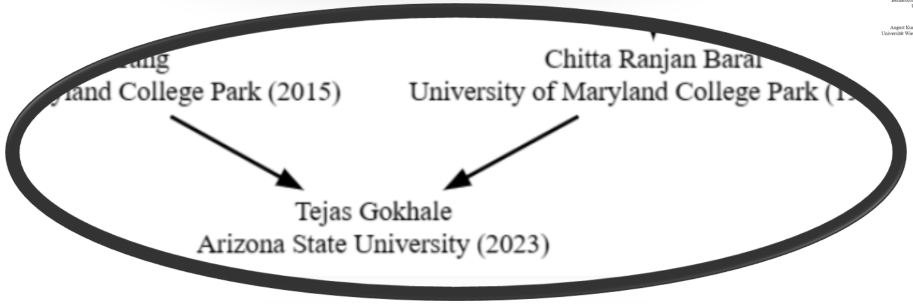


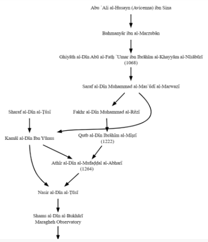
- weight falls off with distance from center pixel
- theoretically infinite, in practice truncated to some maximum distance

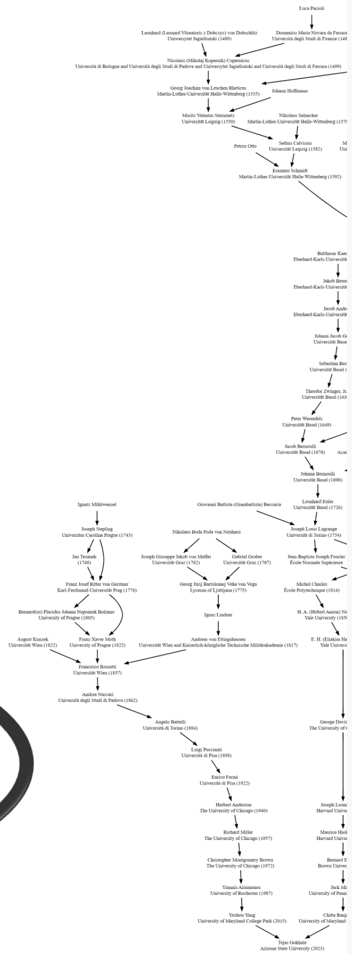
$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3x3 Gaussian kernel

If you do a CS PhD in US/UK/EU Gauss is your ancestor (in most cases)



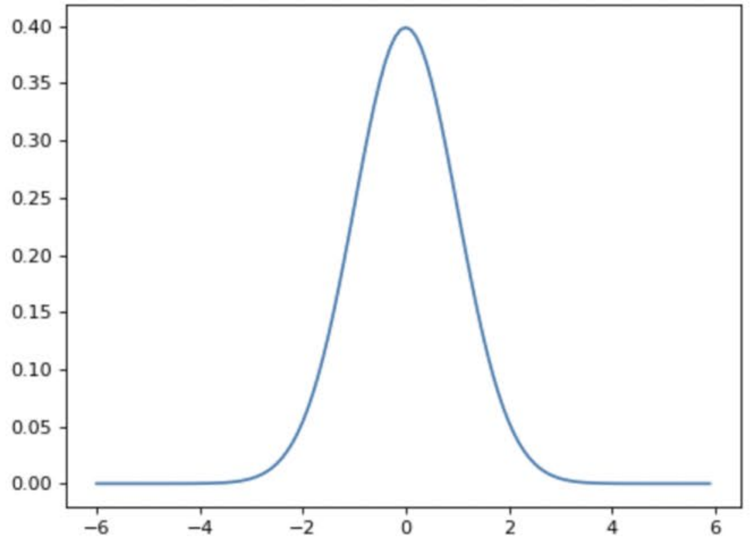


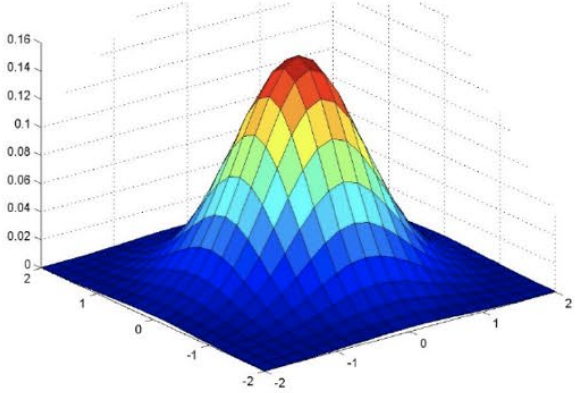


The Gaussian filter

$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

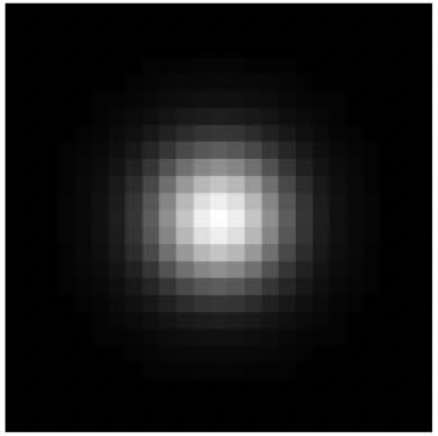
- named (like many other things) after Carl Friedrich Gauss





Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



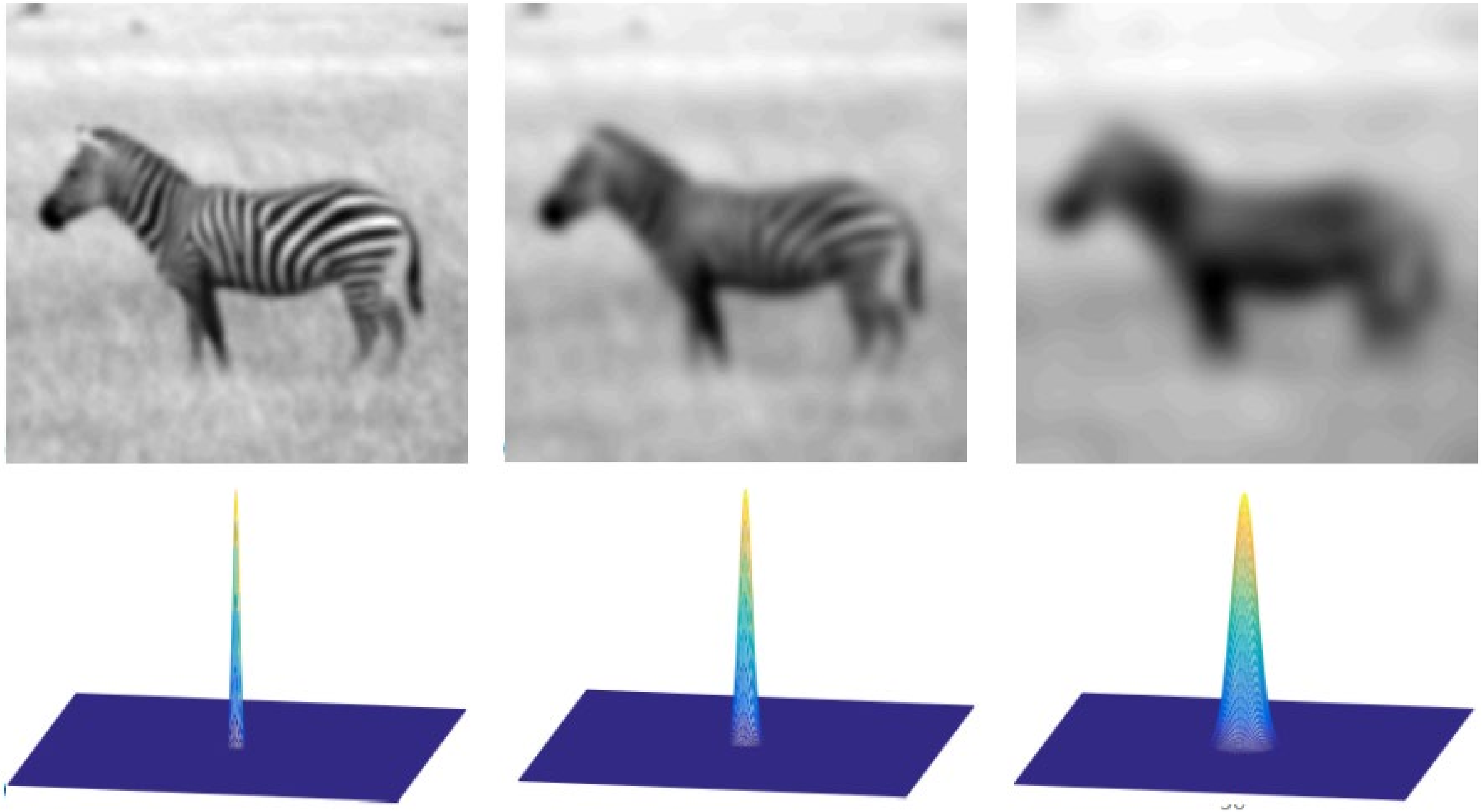
- weight falls off with distance from center pixel
- theoretically infinite, in practice truncated to some maximum distance

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3x3 Gaussian kernel

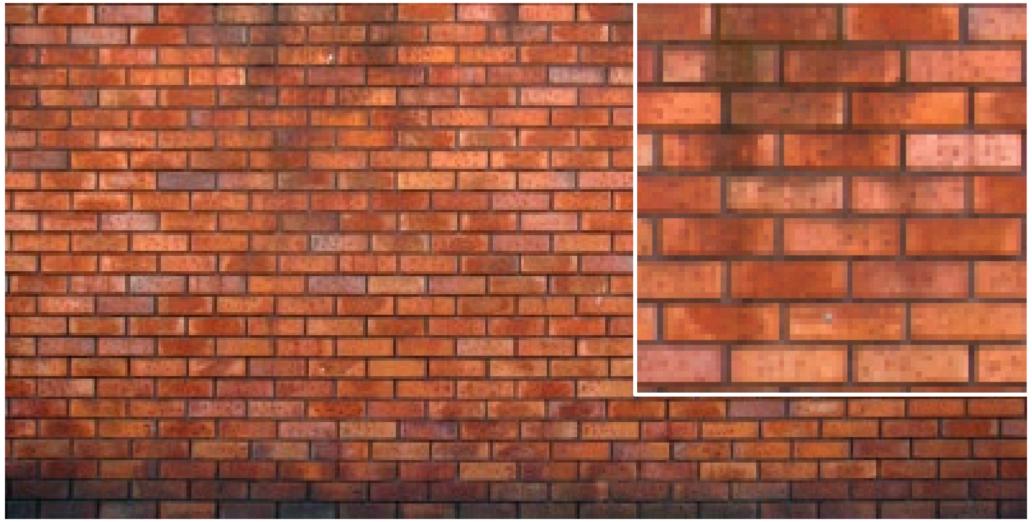
Gaussian filtering example

Scale



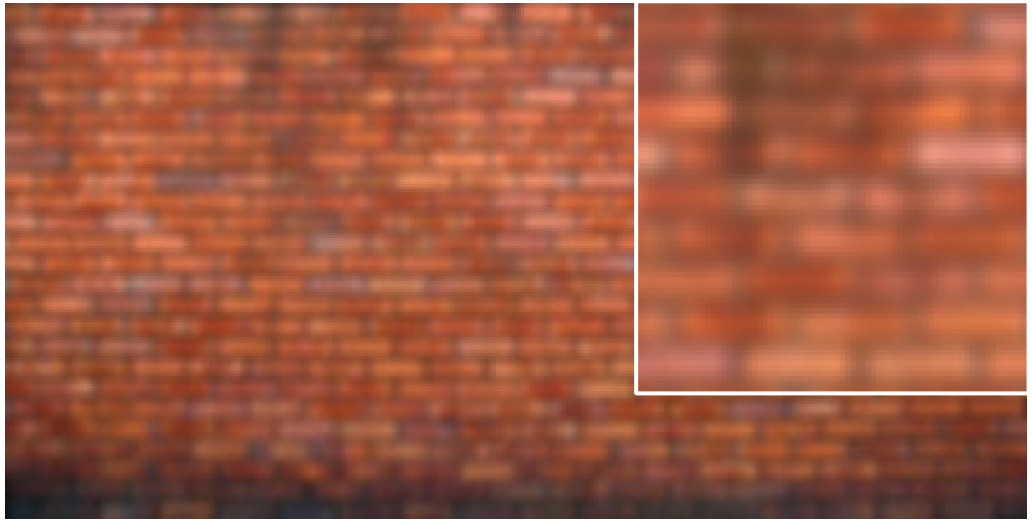
Slide from Antonio Torralba

Gaussian vs box filtering

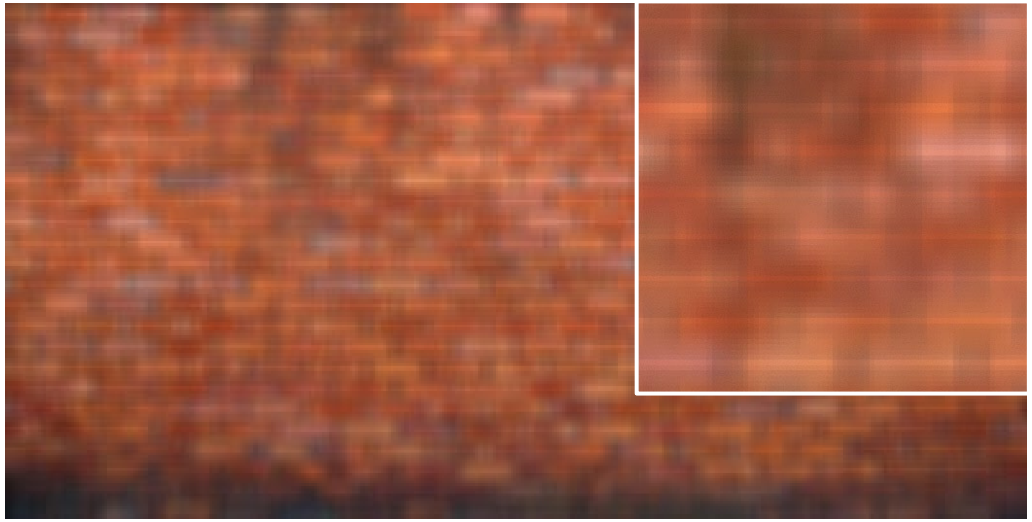


original

Which blur do you like better?

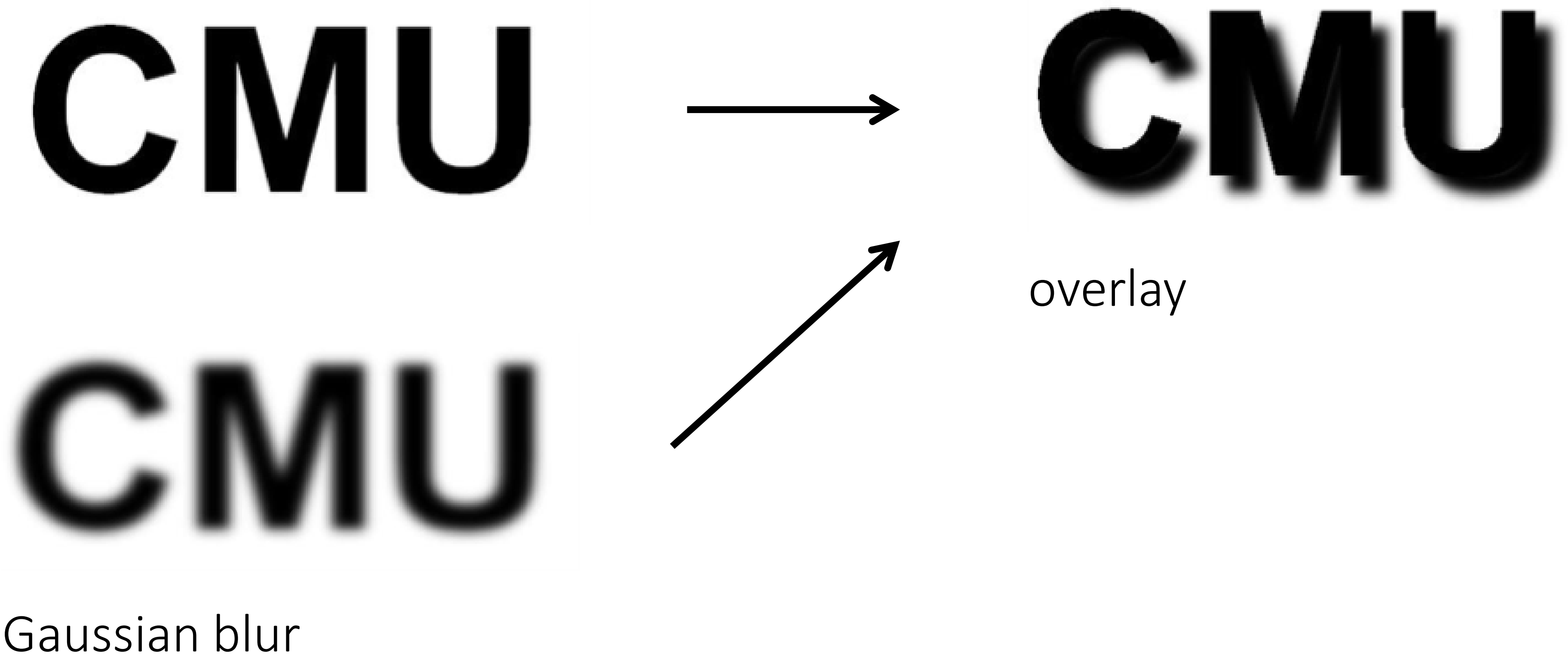


7x7 Gaussian



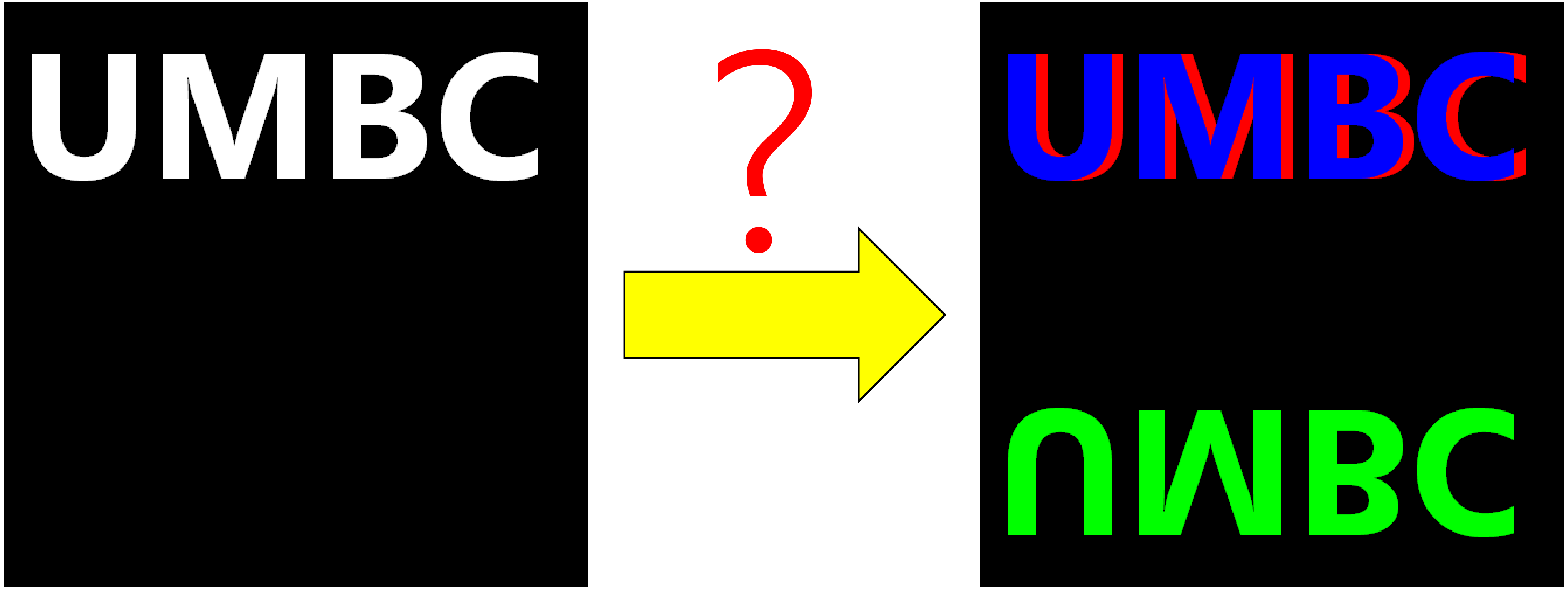
7x7 box

How would you create a soft shadow effect?



Quiz! (Bring Answers to Next Class)

Write an Equation to generate X_{out}
using X , appropriate filters, point operators, etc.



X $\quad$ X_{out}