Light, Camera and Shading

CS 543 / ECE 549 – Saurabh Gupta

Many slides adapted from S. Seitz, L. Lazebnik, D. Hoiem, D. Forsyth

Overview

- Cameras with lenses
  - Depth of field
  - Field of view
  - Lens aberrations
- Brightness of a pixel
  - Small taste of radiometry
  - In-camera transformation of light
  - Reflectance properties of surfaces
  - Lambertian reflection model
  - Shape from shading

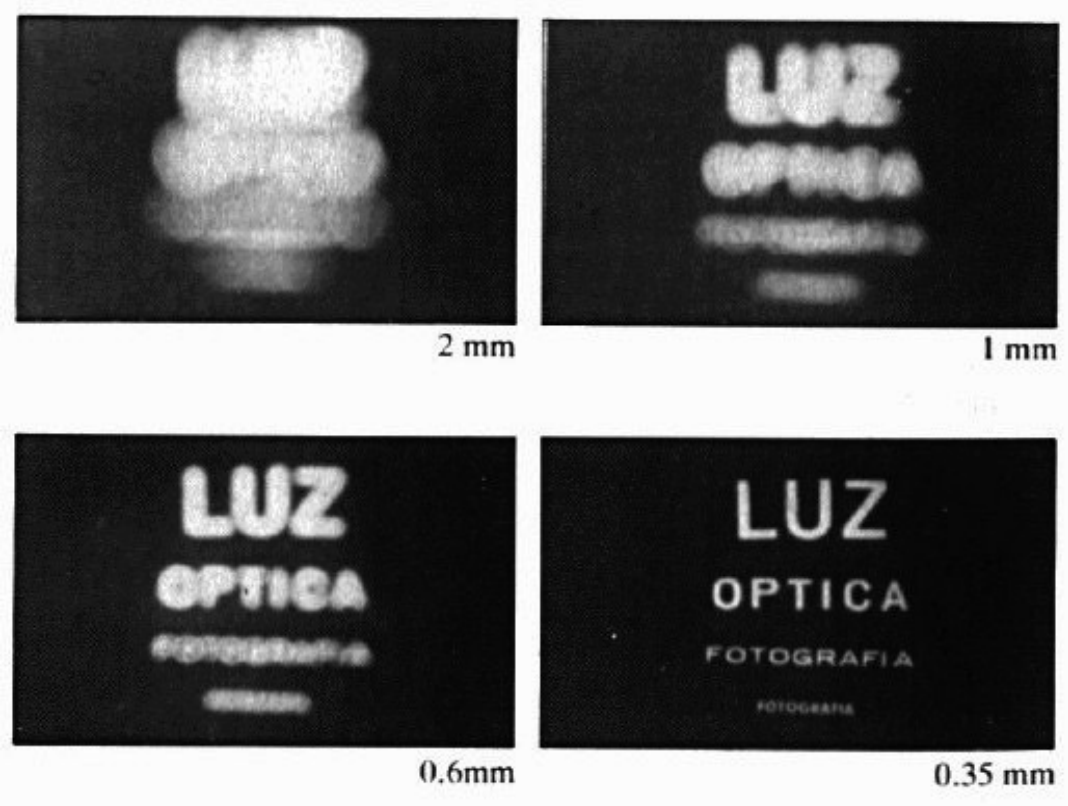
Building a Real Camera

Home-made pinhole camera

Slide by A. Efros

http://www.debevec.org/Pinhole/

Shrinking the aperture

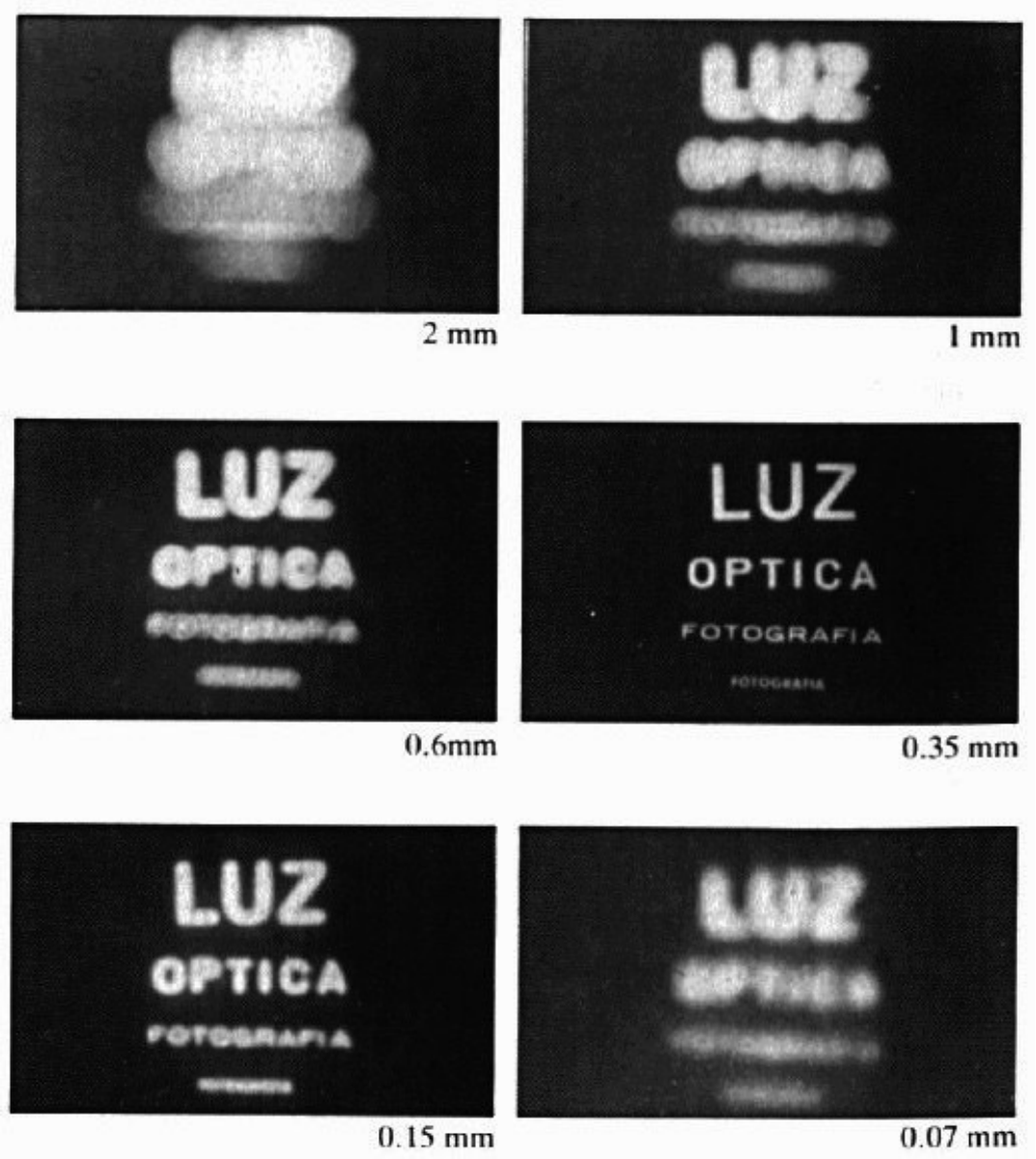


Why not make the aperture as small as possible?

- Less light gets through
- Diffraction effects…

Slide by Steve Seitz

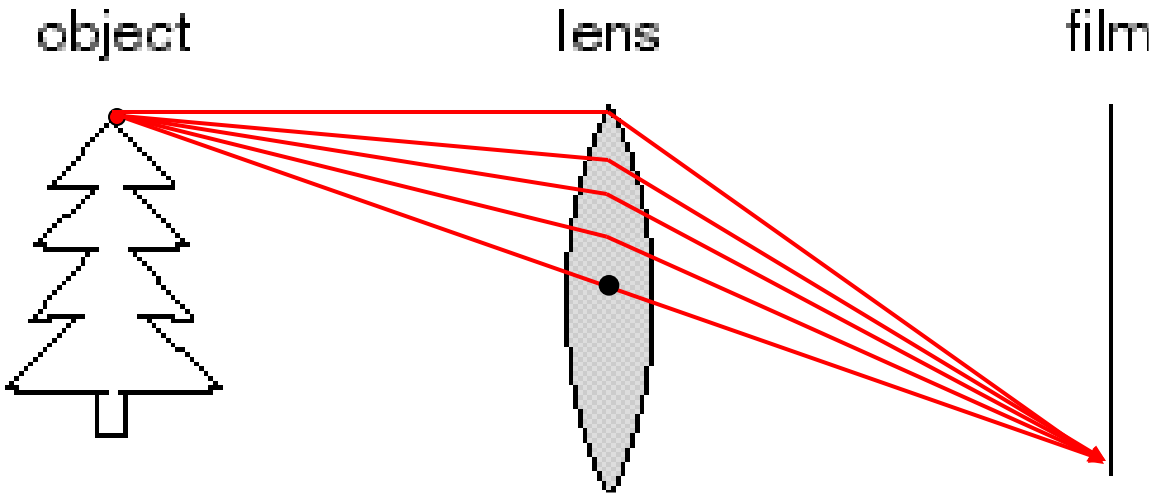
Shrinking the aperture



Slide by Steve Seitz

Adding a lens

Adding a lens

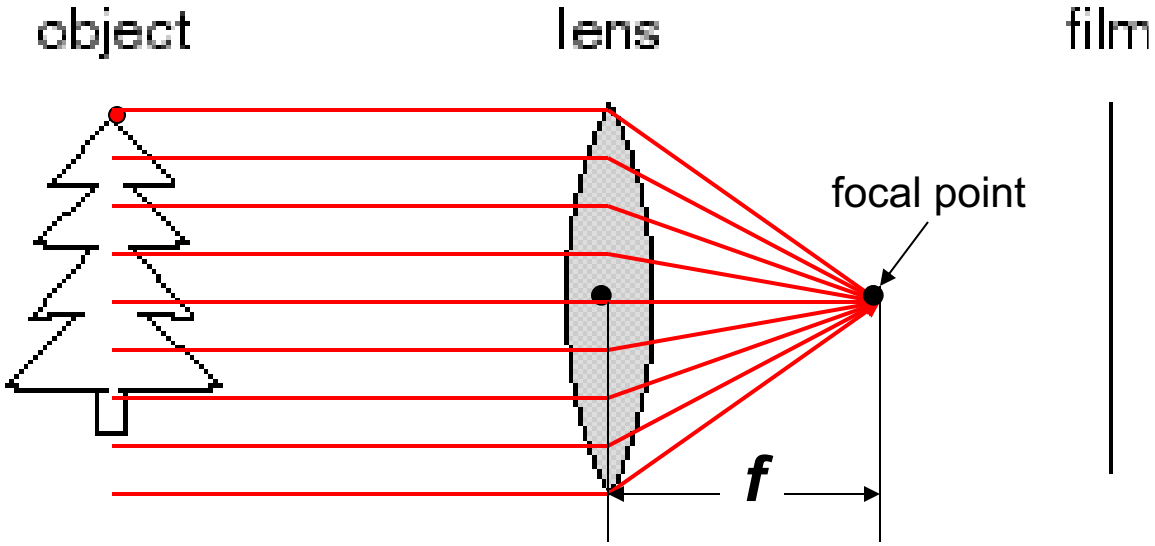


A lens focuses light onto the film

- Thin lens model:
  - Rays passing through the center are not deviated (pinhole projection model still holds)

Slide by Steve Seitz

Adding a lens



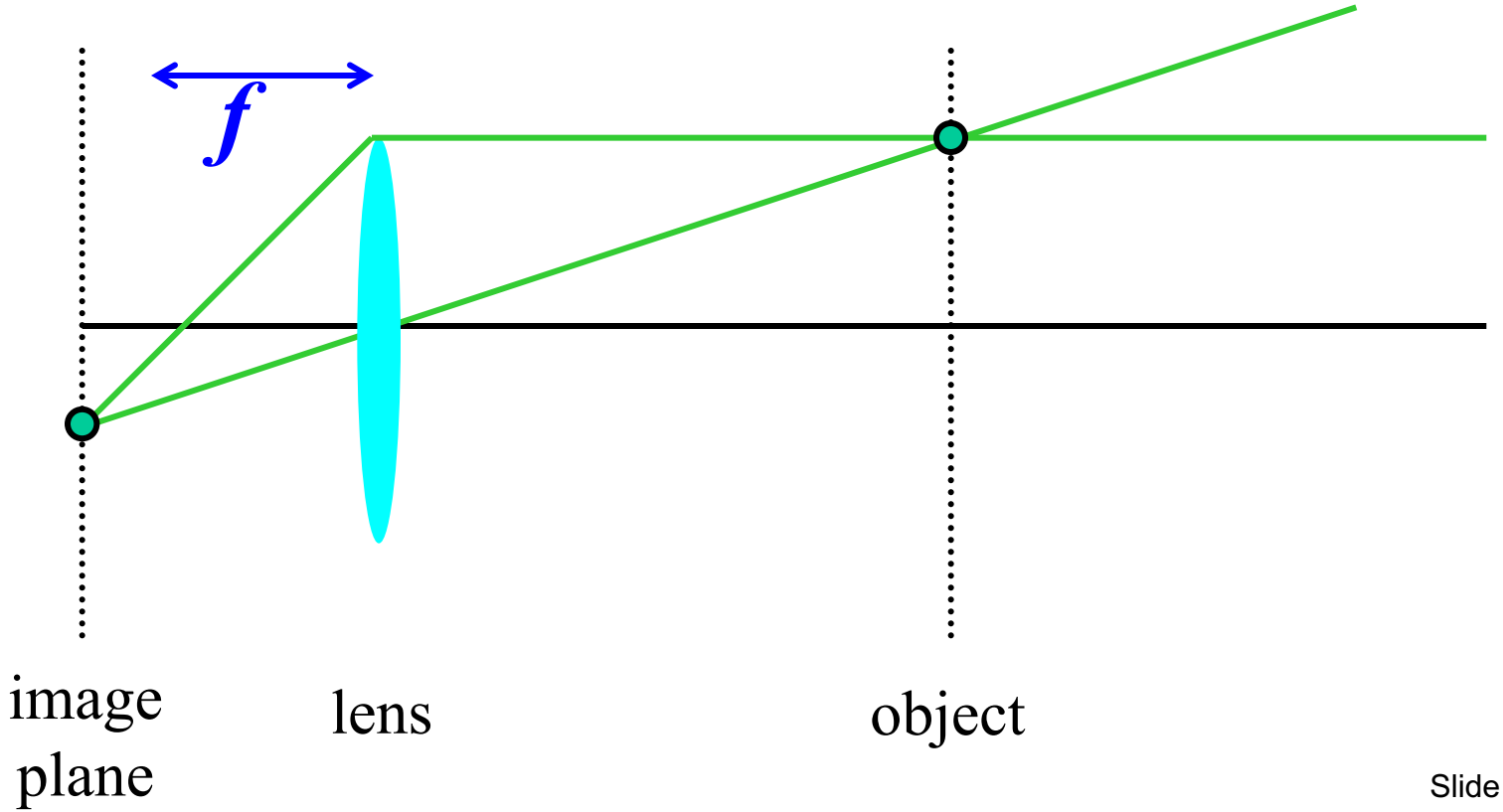
A lens focuses light onto the film

- Thin lens model:
  - Rays passing through the center are not deviated (pinhole projection model still holds)
  - All rays parallel to the optical axis pass through the *focal point*
  - All parallel rays converge to points on the *focal plane*

Slide by Steve Seitz

Thin lens formula

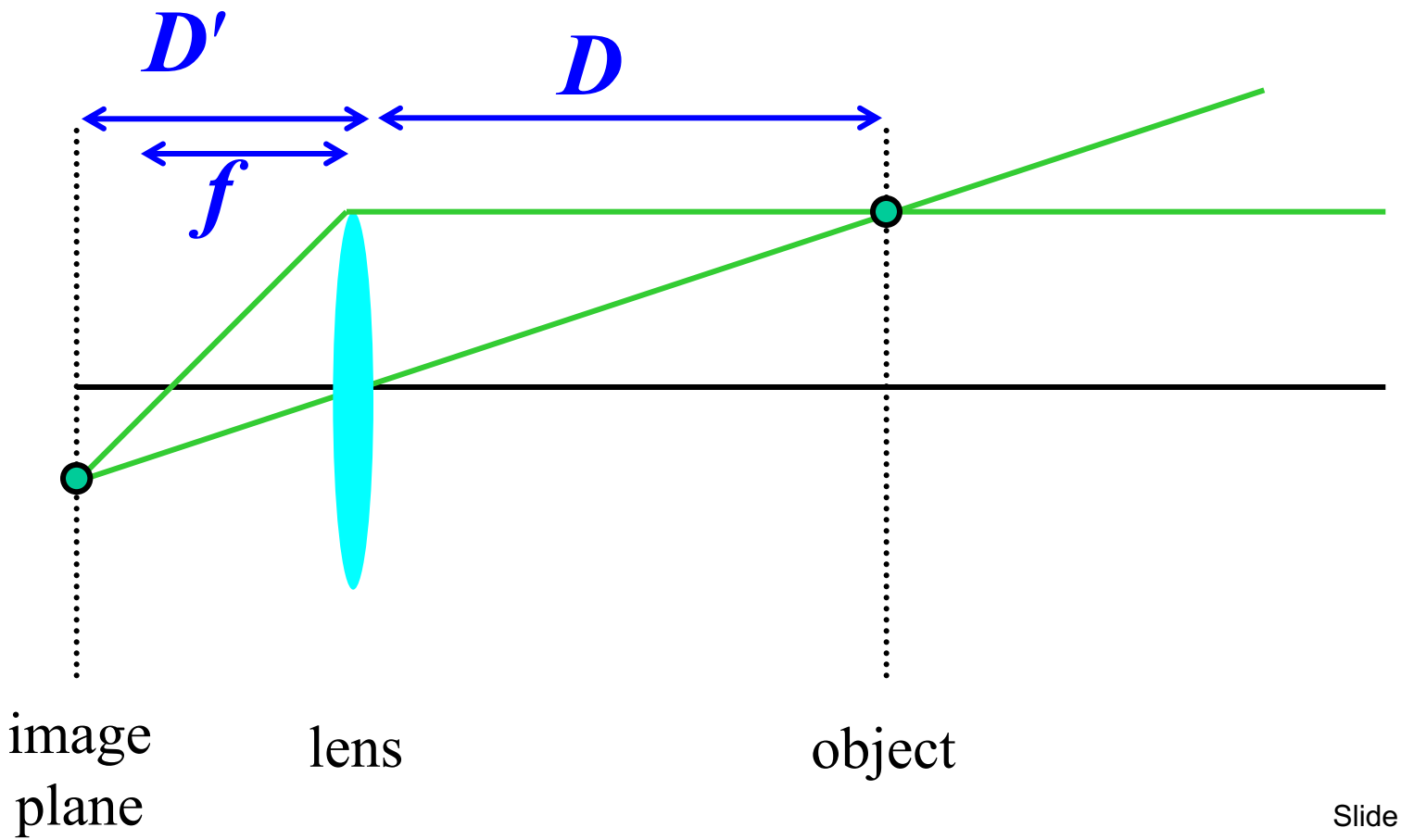
- Where does the lens focus the rays coming from a given point in the scene?



Slide by Frédo Durand

Thin lens formula

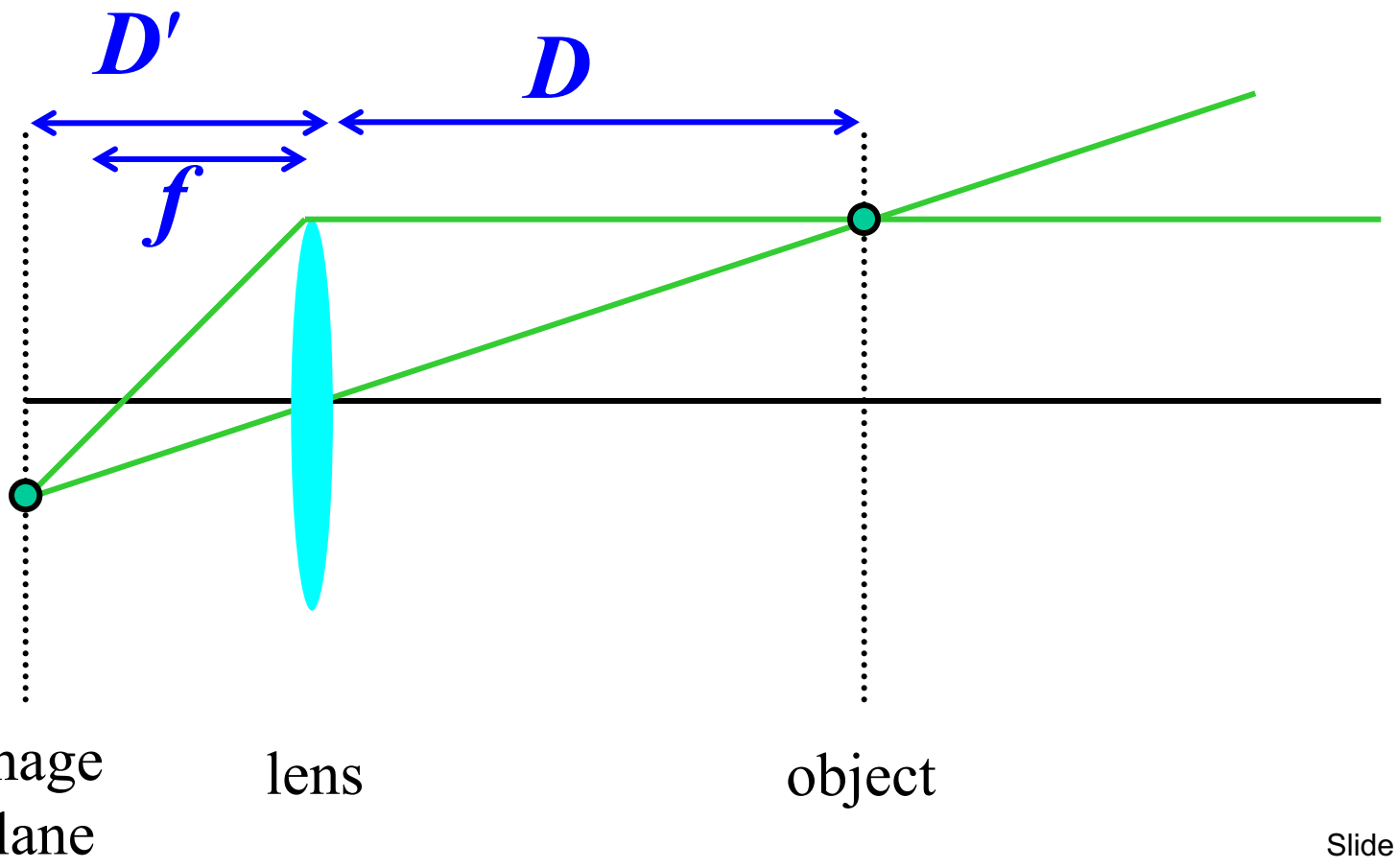
- What is the relation between the focal length (f), the distance of the object from the optical center (D), and the distance at which the object will be in focus (D')?



Slide by Frédo Durand

Thin lens formula

Similar triangles everywhere!

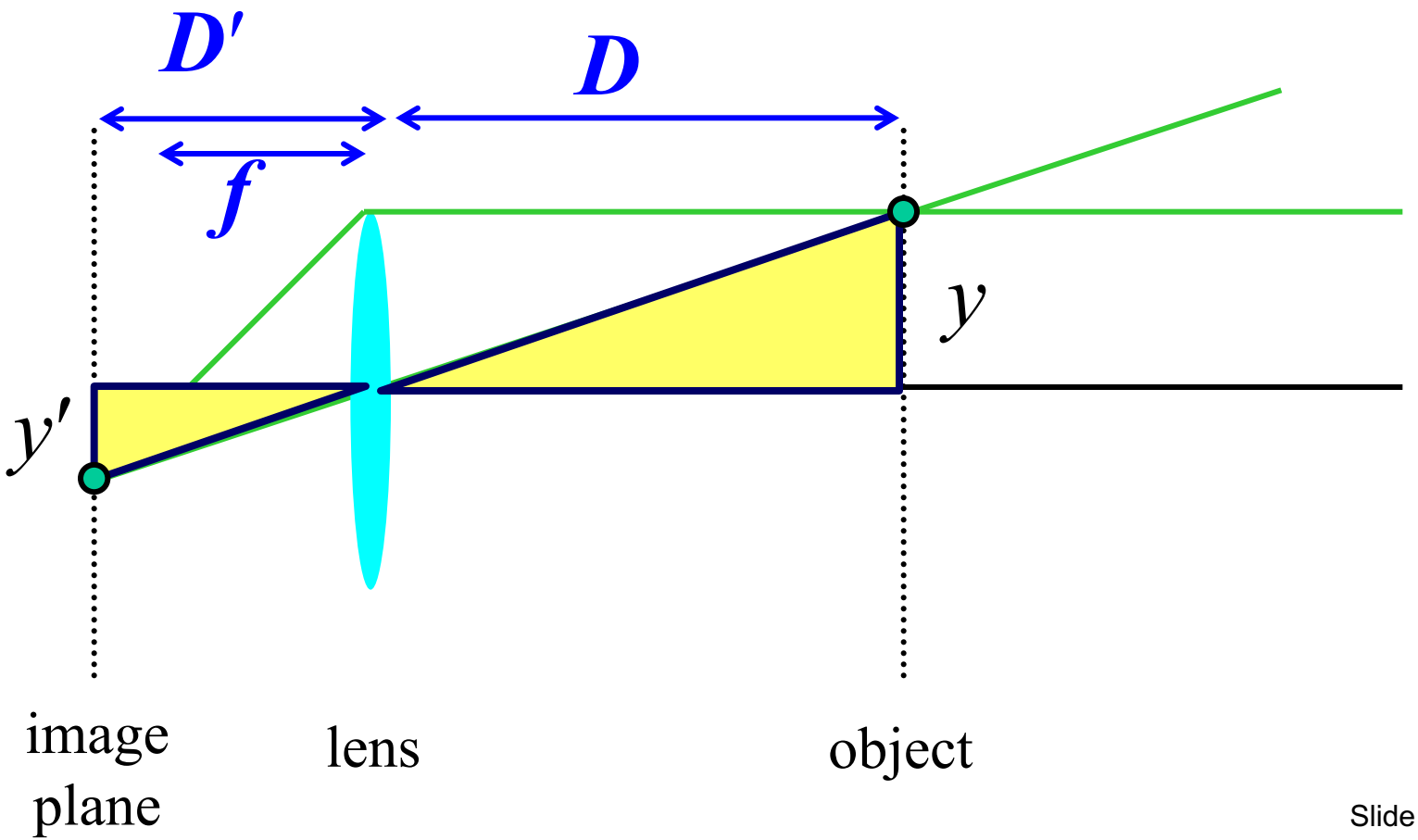


Slide by Frédo Durand

Thin lens formula

Similar triangles everywhere!

$$y'/y = D'/D$$



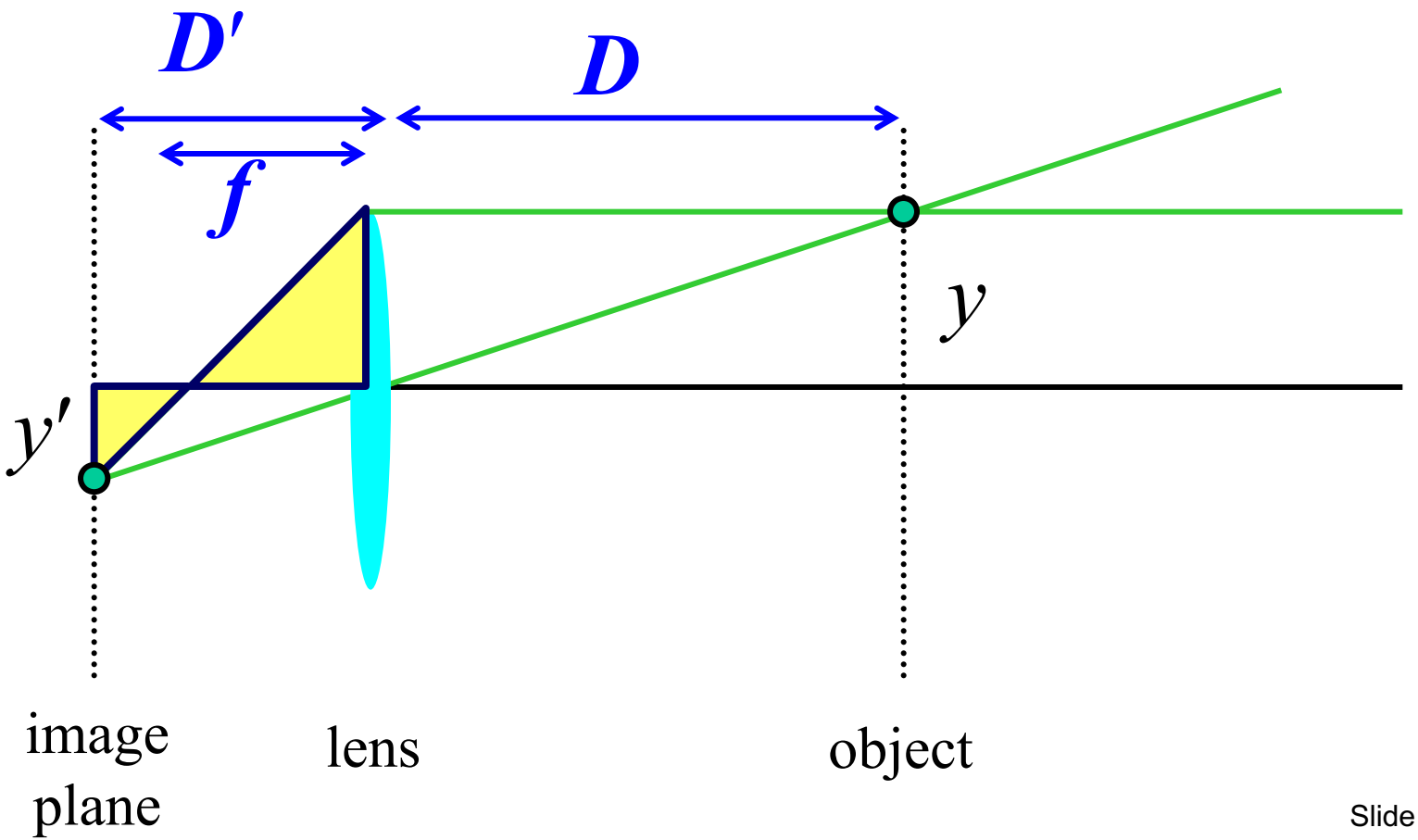
Slide by Frédo Durand

Thin lens formula

Similar triangles everywhere!

$$y'/y = D'/D$$

$$y'/y = (D'-f)/f$$



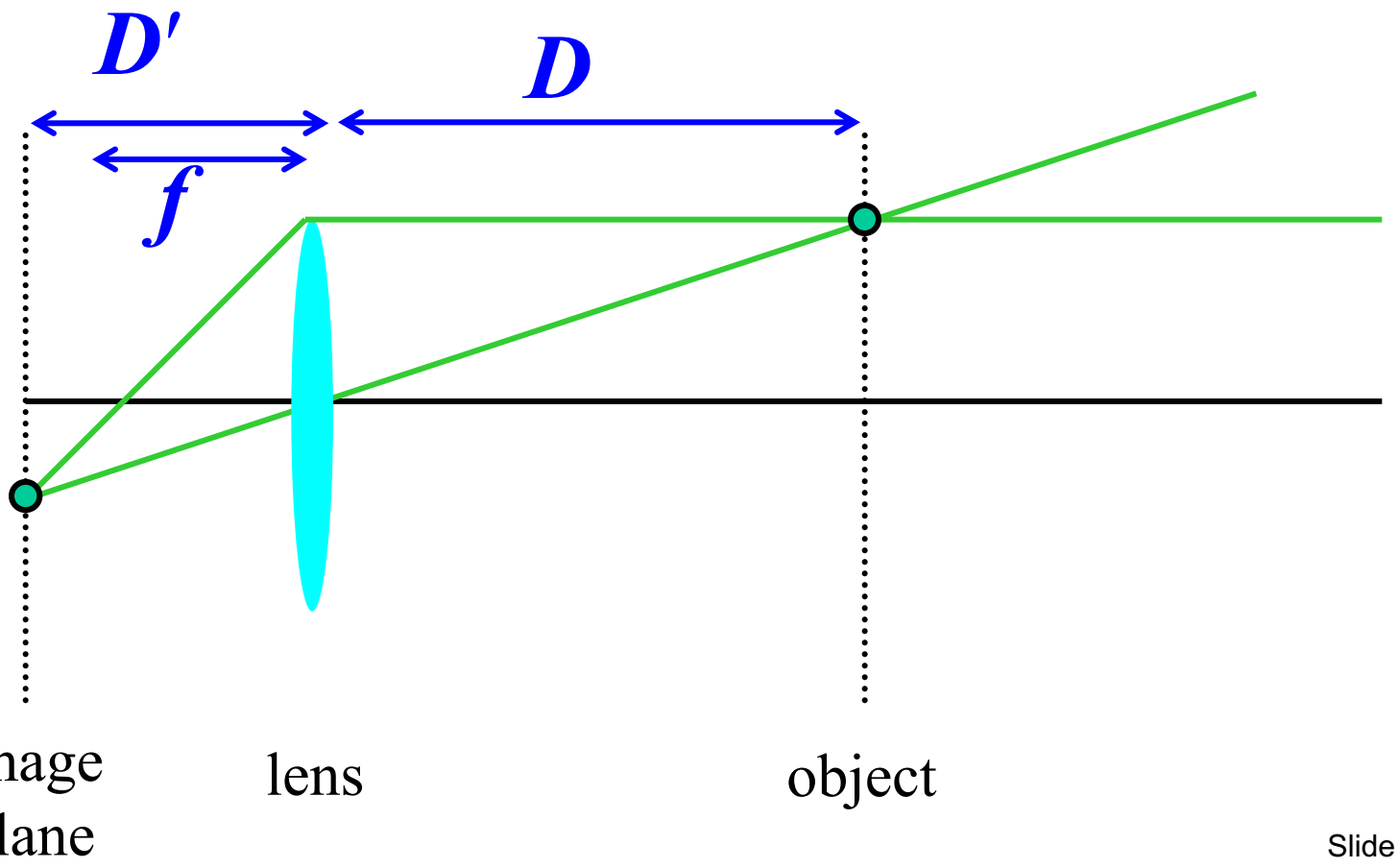
Slide by Frédo Durand

Thin lens formula

$$\frac{1}{D'} + \frac{1}{D} = \frac{1}{f}$$

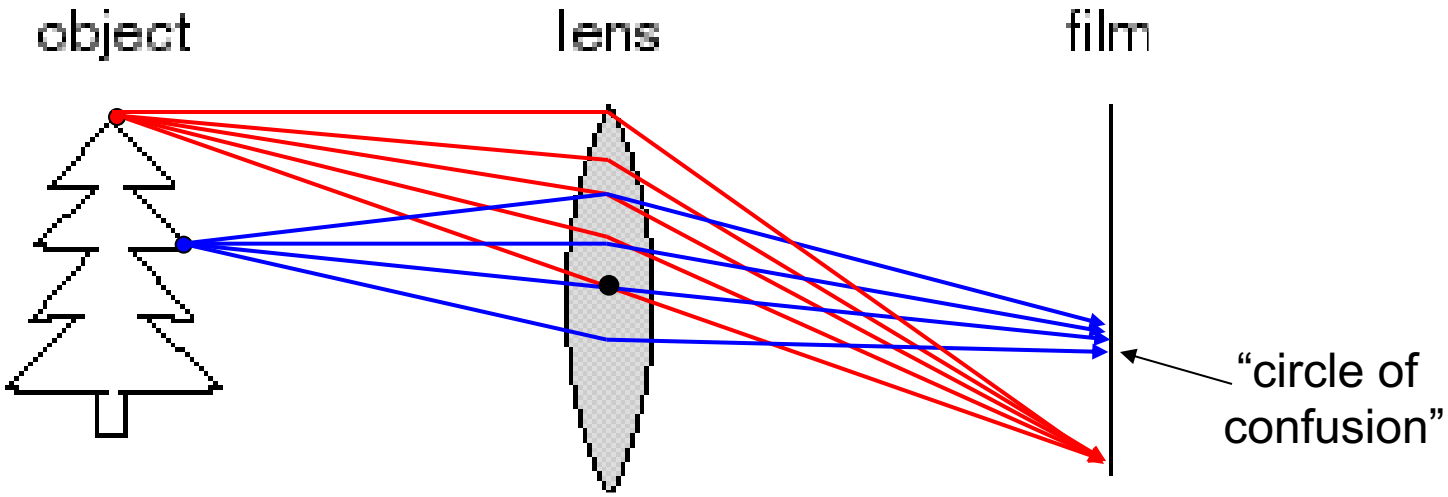
Any point satisfying the thin lens equation is in focus.

What happens when D is very large?



Slide by Frédo Durand

Depth of Field



For a fixed focal length, there is a specific distance at which objects are “in focus”

- Other points project to a “circle of confusion” in the image

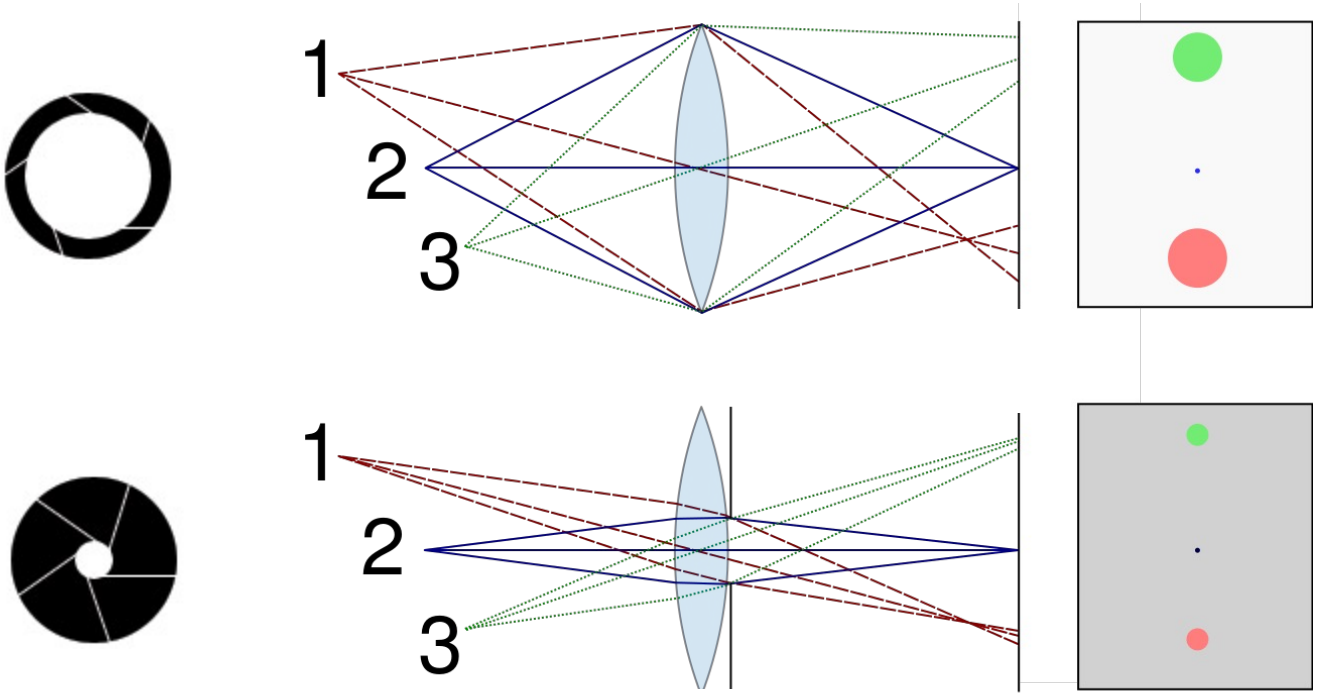
Slide by Steve Seitz

Depth of Field

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

Slide by A. Efros

Controlling depth of field



Changing the aperture size affects depth of field

- A smaller *aperture* increases the range in which the object is approximately in focus
- But small aperture reduces amount of light – need to increase *exposure*

http://en.wikipedia.org/wiki/File:Depth_of_field_illustration.svg

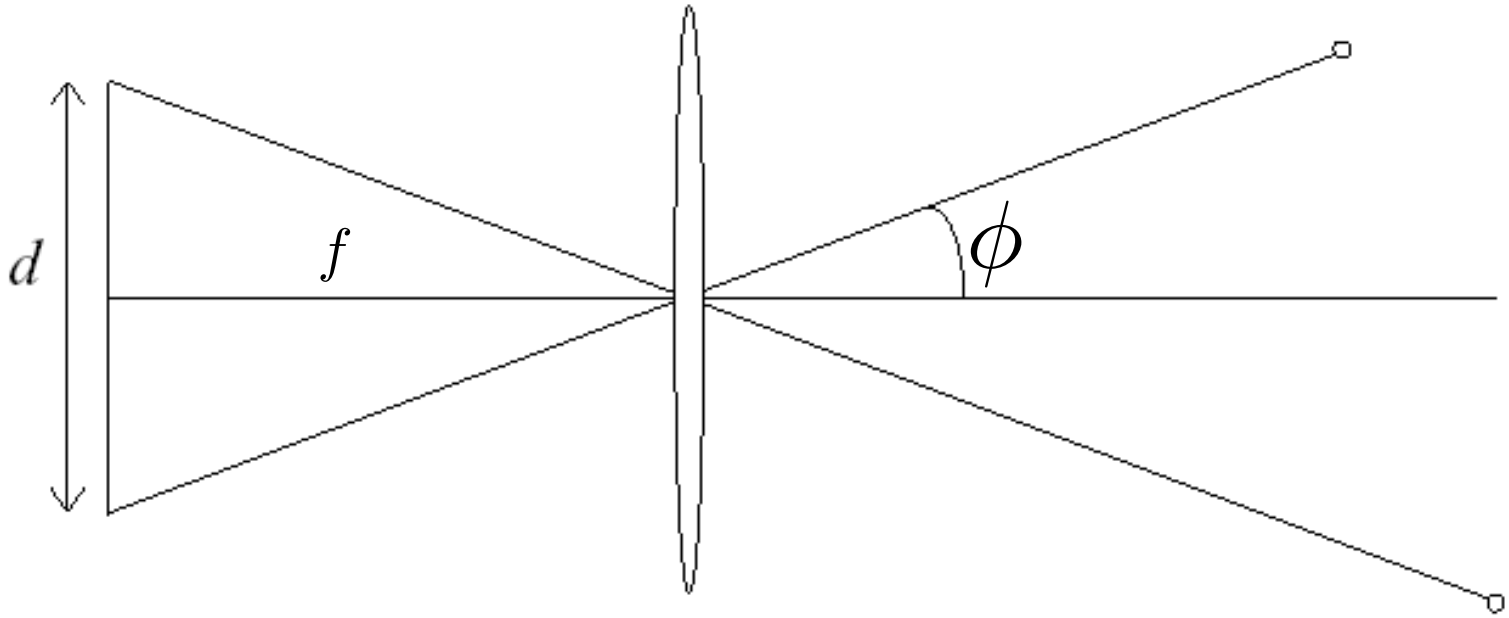
Varying the aperture

Large aperture = small DOF

Small aperture = large DOF

Slide by A. Efros

Field of View



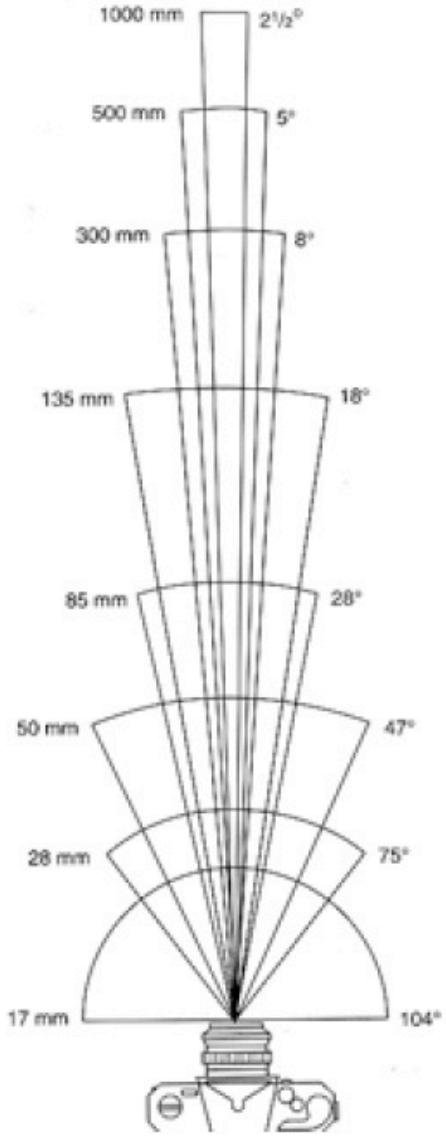
FOV depends on focal length and size of the camera retina

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

Larger focal length = smaller FOV

Slide by A. Efros

Field of View



17mm

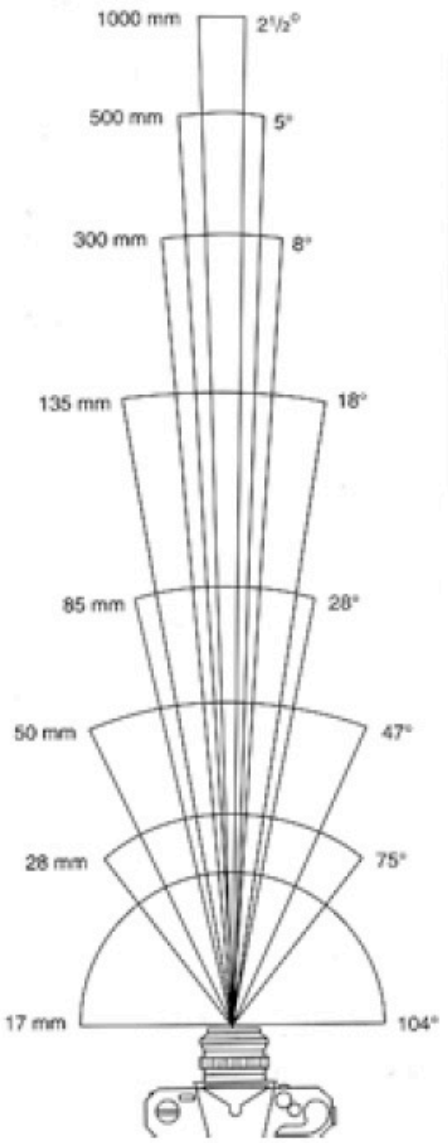
28mm

50mm

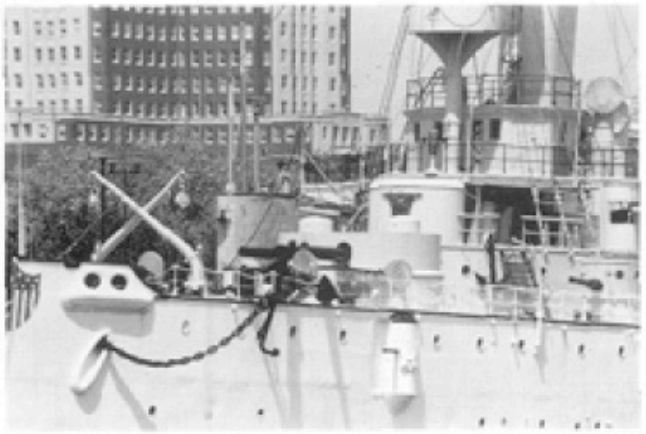
85mm

Slide by A. Efros

Field of View



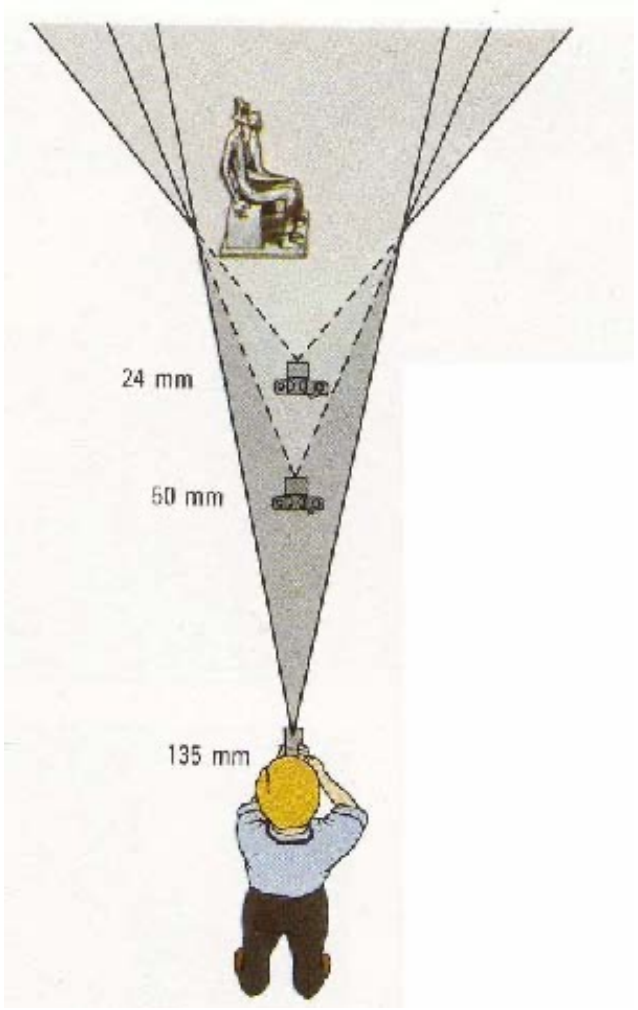
135mm



300mm

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

Same effect for faces

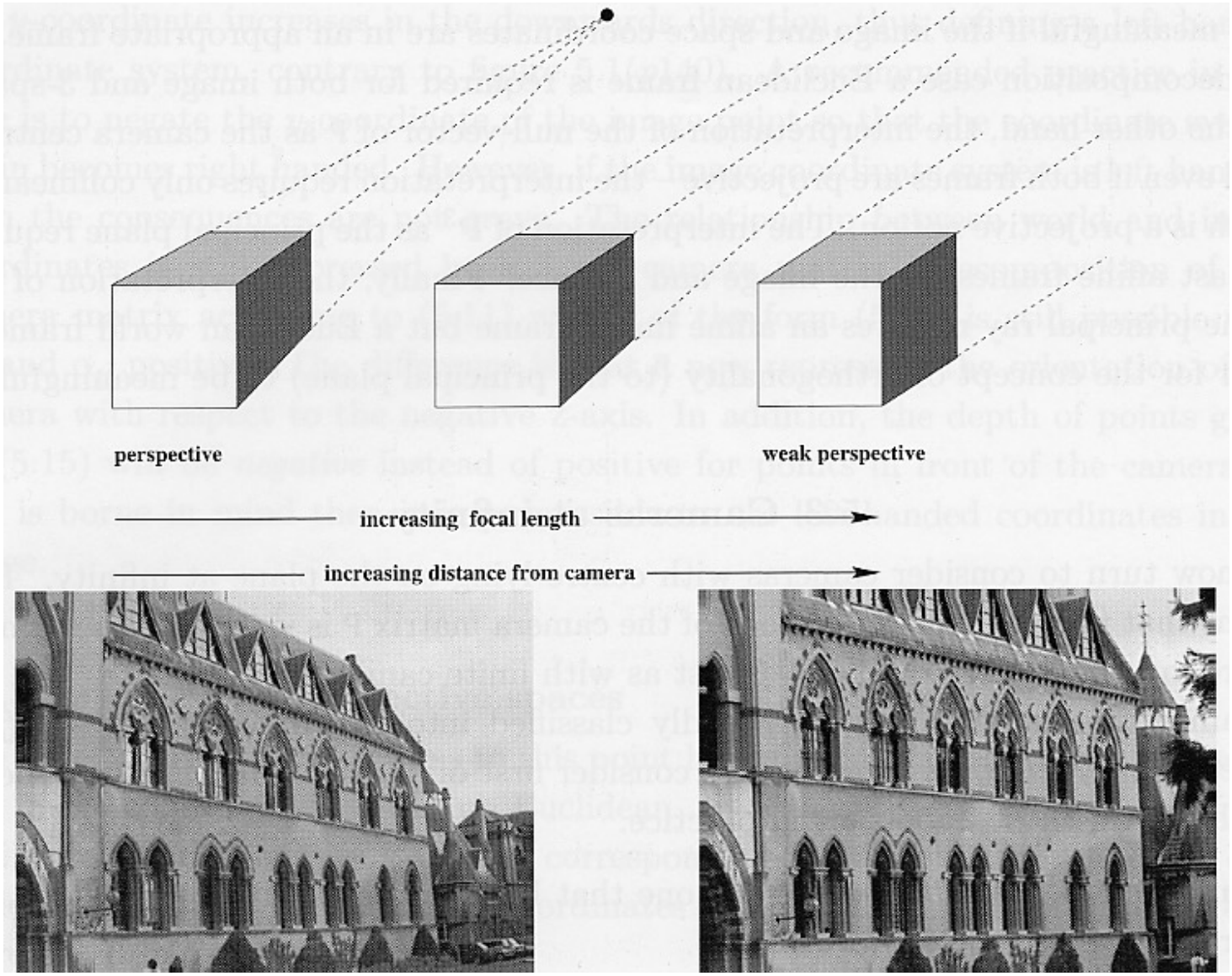
wide-angle

standard

telephoto

Source: F. Durand

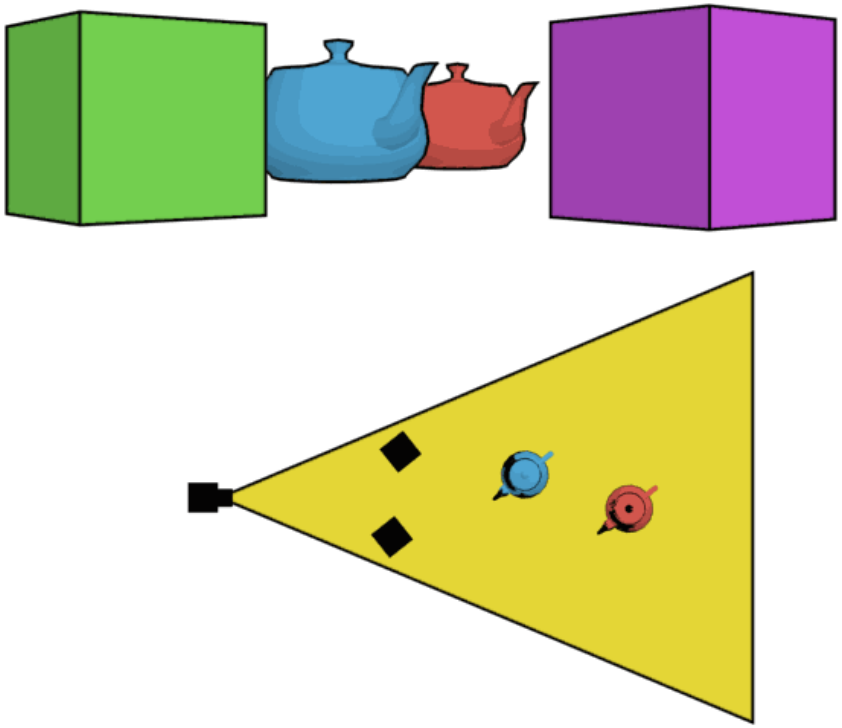
Approximating an orthographic camera



Source: Hartley & Zisserman

The dolly zoom

- Continuously adjusting the focal length while the camera moves away from (or towards) the subject



http://en.wikipedia.org/wiki/Dolly_zoom

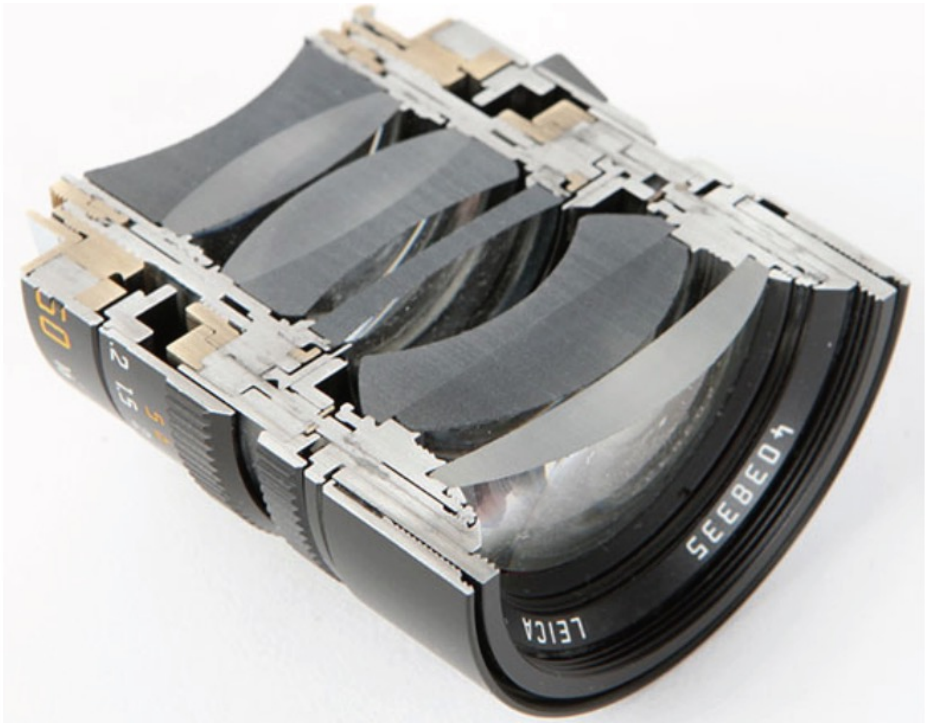
The dolly zoom

- Continuously adjusting the focal length while the camera moves away from (or towards) the subject
- “The Vertigo shot”

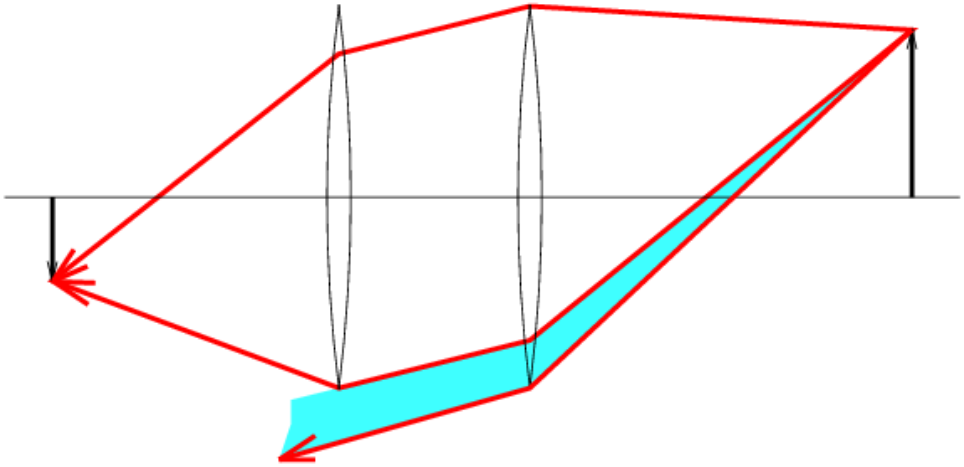
Example of dolly zoom from *Goodfellas* (YouTube)

Example of dolly zoom from *La Haine* (YouTube)

Real lenses

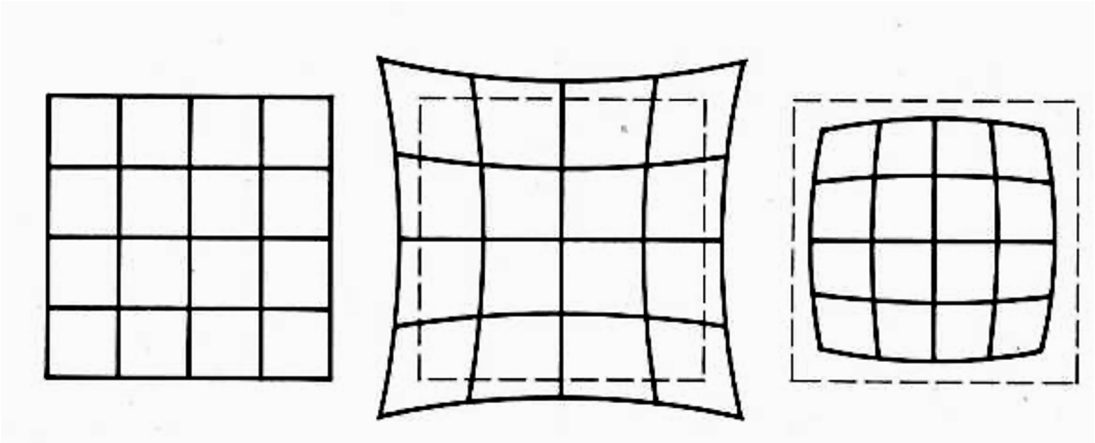


Lens flaws: Vignetting

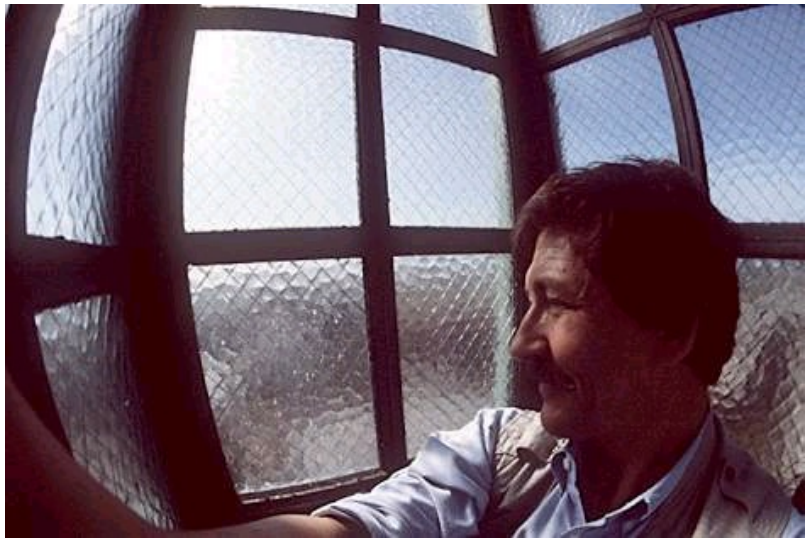


Radial Distortion

- Caused by imperfect lenses
- Deviations are most noticeable near the edge of the lens



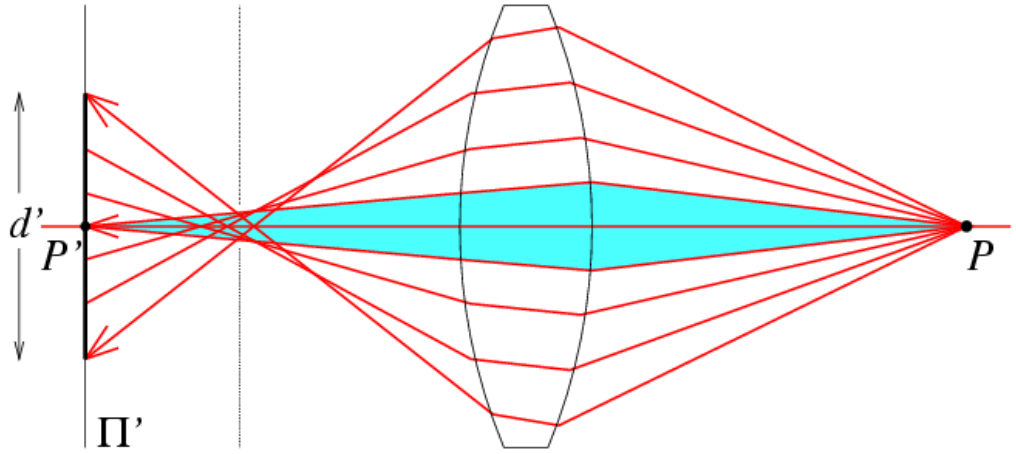
No distortion Pin cushion Barrel



Lens flaws: Spherical aberration

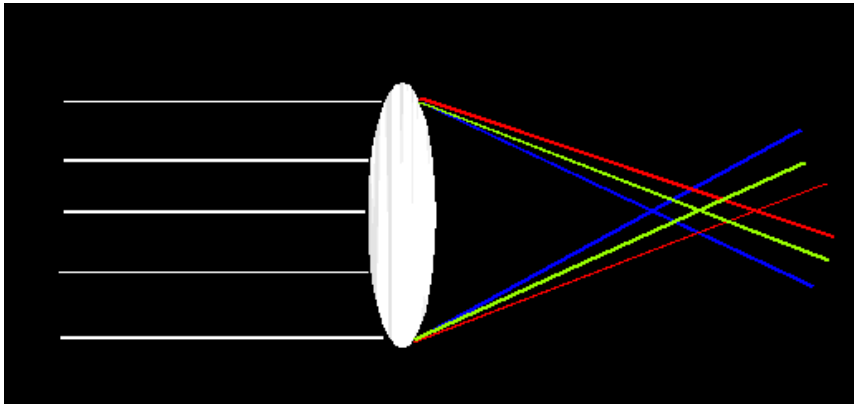
Spherical lenses don’t focus light perfectly

Rays farther from the optical axis focus closer



Lens Flaws: Chromatic Aberration

Lens has different refractive indices for different wavelengths: causes color fringing

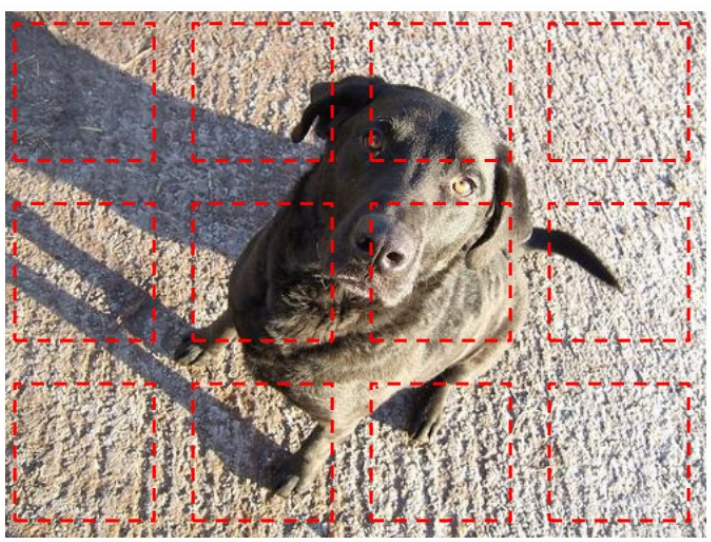


Near Lens Center

Near Lens Outer Edge

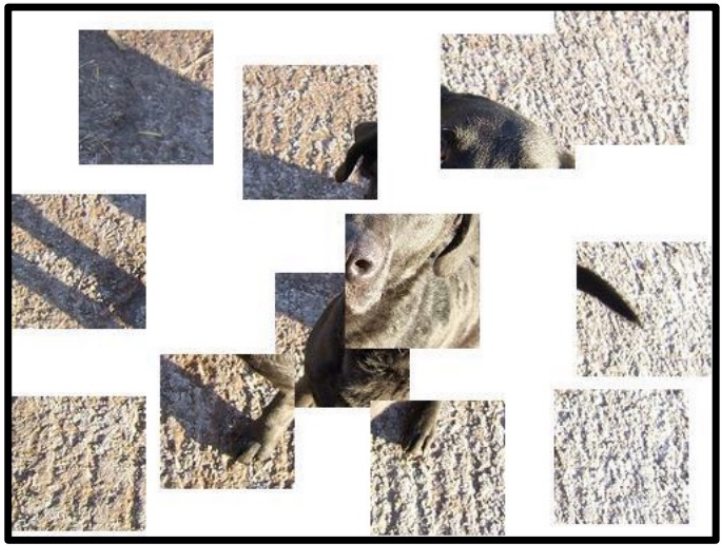
Lens Flaws: Chromatic Aberration

Researchers tried teaching a network about objects by forcing it to assemble jigsaws.



Initial layout, with sampled patches in red

Image layout is discarded



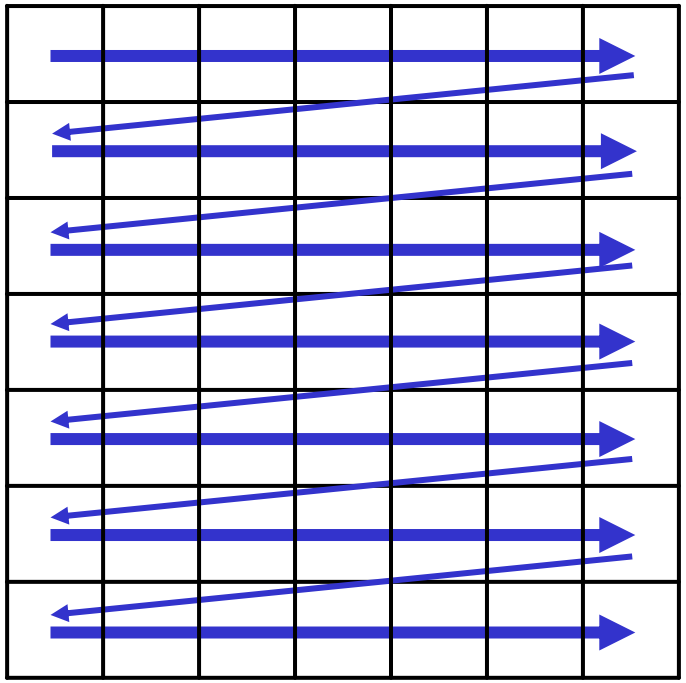
We can recover image layout automatically

Slide Credit: C. Doersch

Rolling Shutter

Rolling Shutter: pixels read in sequence

Can get global reading, but **$$$**

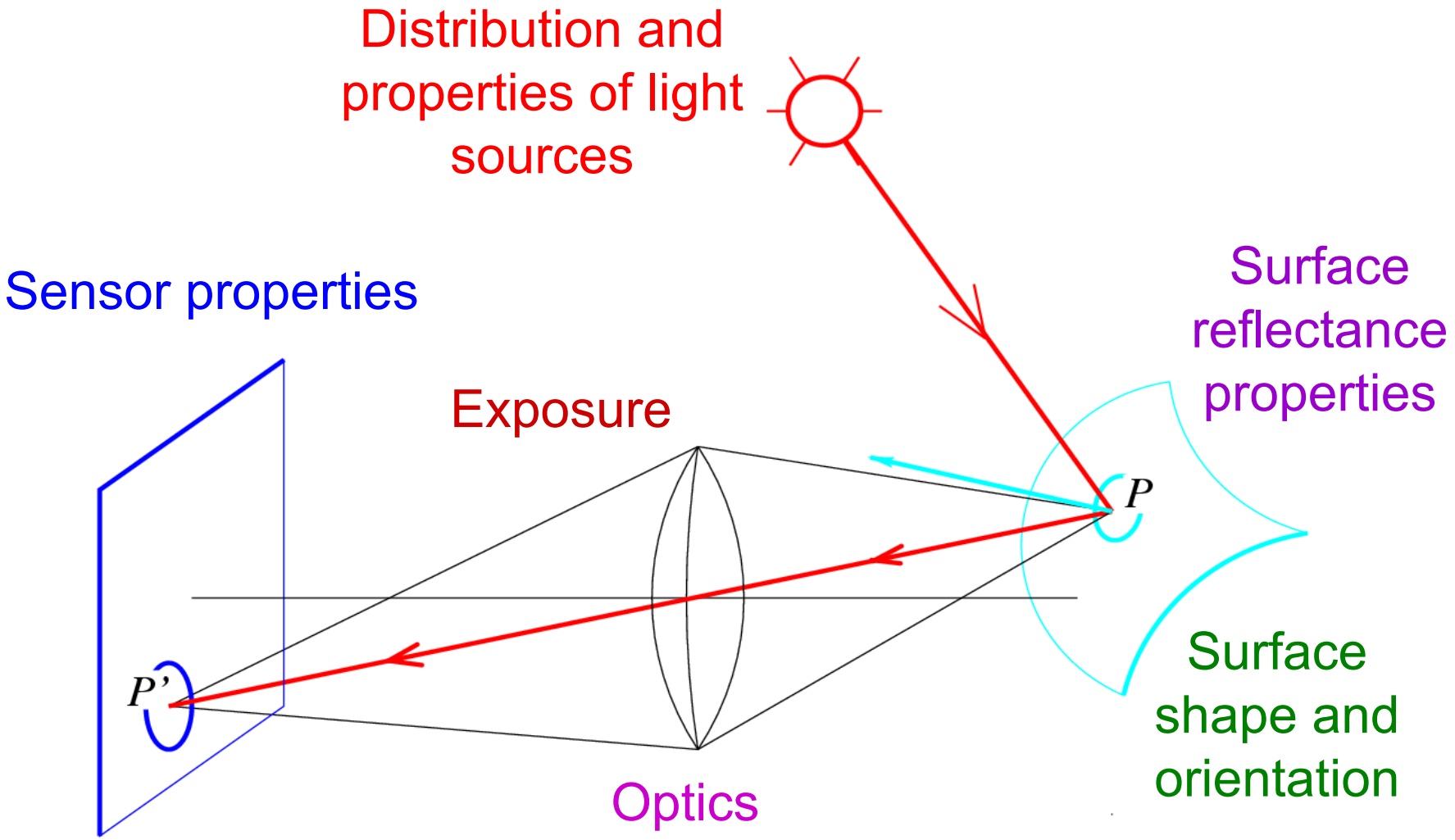


Overview

- Cameras with lenses
  - Depth of field
  - Field of view
  - Lens aberrations
- Brightness of a pixel
  - Small taste of radiometry
  - In-camera transformation of light
  - Reflectance properties of surfaces
  - Lambertian reflection model
  - Shape from shading

Image formation

What determines the brightness of an image pixel?



Slide by L. Fei-Fei

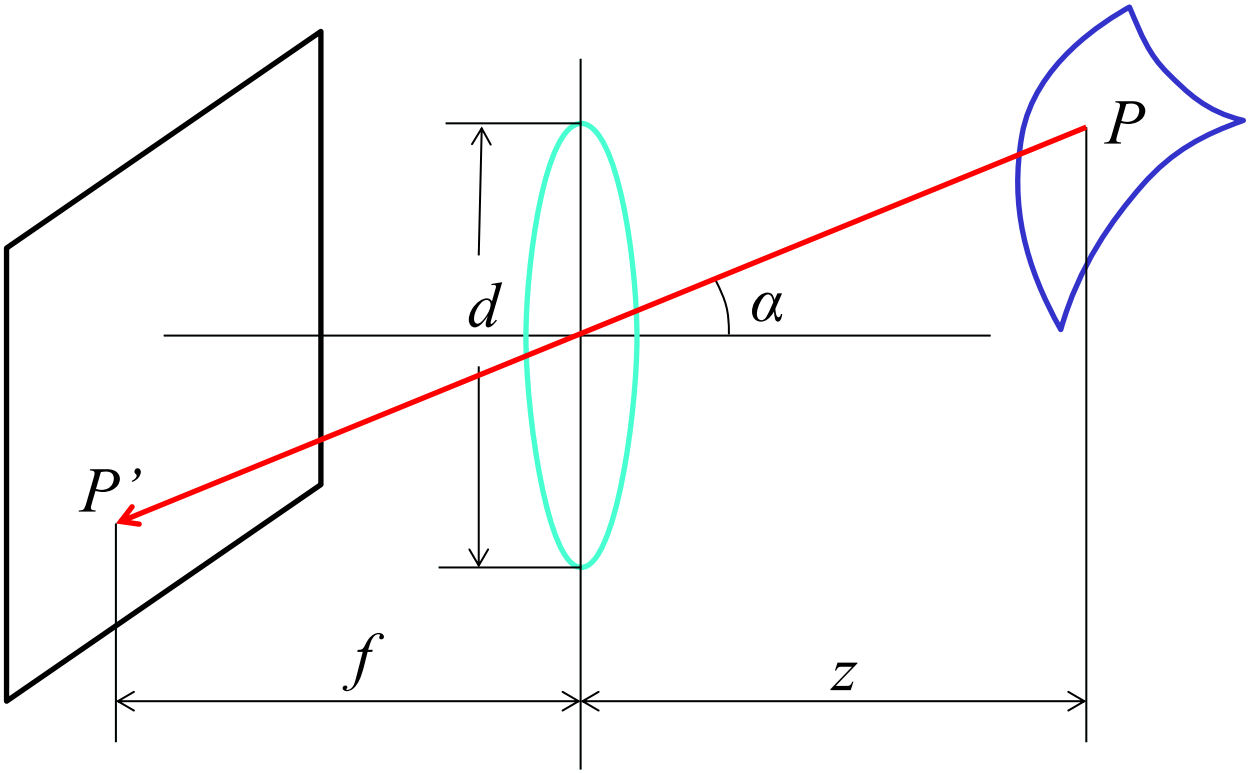
Fundamental radiometric relation

L: *Radiance* emitted from *P* toward *P*'

- Energy carried by a ray (Watts per sq. meter per steradian)

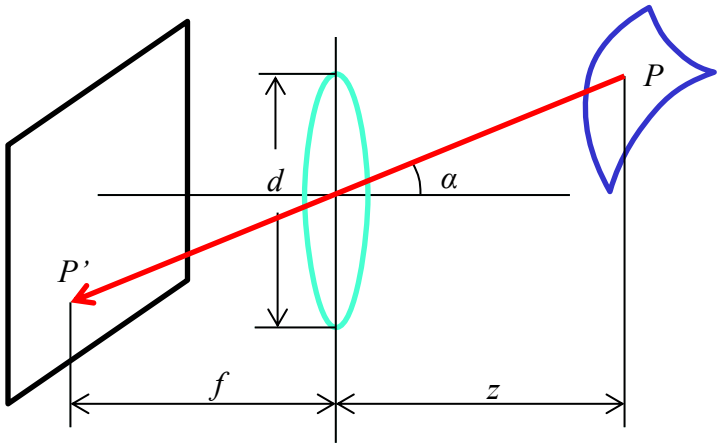
E: *Irradiance* falling on *P*' from the lens

- Energy arriving at a surface (Watts per sq. meter)



What is the relationship between *E* and *L*?

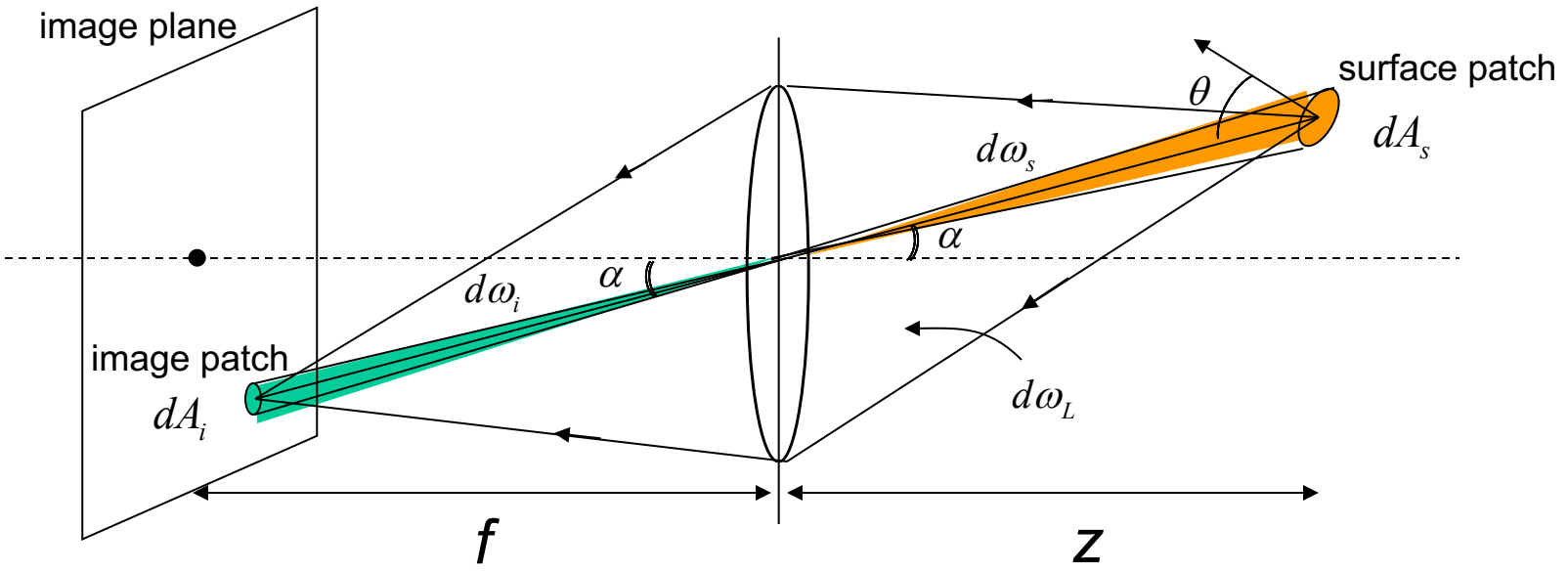
Fundamental radiometric relation



$$E = \left[\frac{\pi}{4} \left(\frac{d}{f} \right)^2 \cos^4 \alpha \right] L$$

- Image irradiance is linearly related to scene radiance
- Irradiance is proportional to the area of the lens and inversely proportional to the squared distance between the lens and the image plane
- The irradiance falls off as the angle between the viewing ray and the optical axis increases

Relation between Image Irradiance E and Scene Radiance L



- Solid angles of the double cone (orange and green):

$$d\omega_i = d\omega_s \qquad \frac{dA_i \cos\alpha}{(f/\cos\alpha)^2} = \frac{dA_s \cos\theta}{(z/\cos\alpha)^2}$$

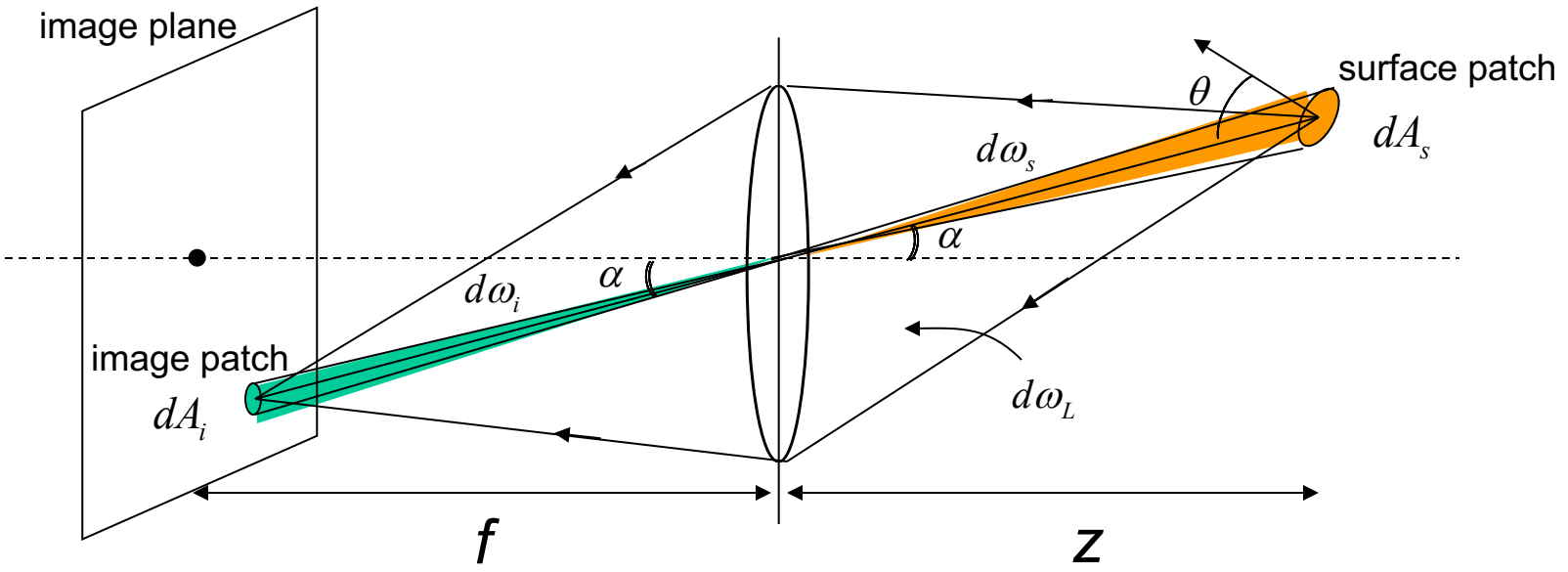
$$\boxed{\frac{dA_s}{dA_i} = \frac{\cos\alpha}{\cos\theta}\left(\frac{z}{f}\right)^2} \quad (1)$$

- Solid angle subtended by lens:

$$\boxed{d\omega_L = \frac{\pi d^2}{4}\frac{\cos\alpha}{(z/\cos\alpha)^2}} \quad (2)$$

Slide from S Narasimhan.

Relation between Image Irradiance E and Scene Radiance L



- Flux received by lens from dA_s = Flux projected onto image dA_i

$$L\,(dA_s\cos\theta)\,d\omega_L = E\,dA_i \longrightarrow (3)$$

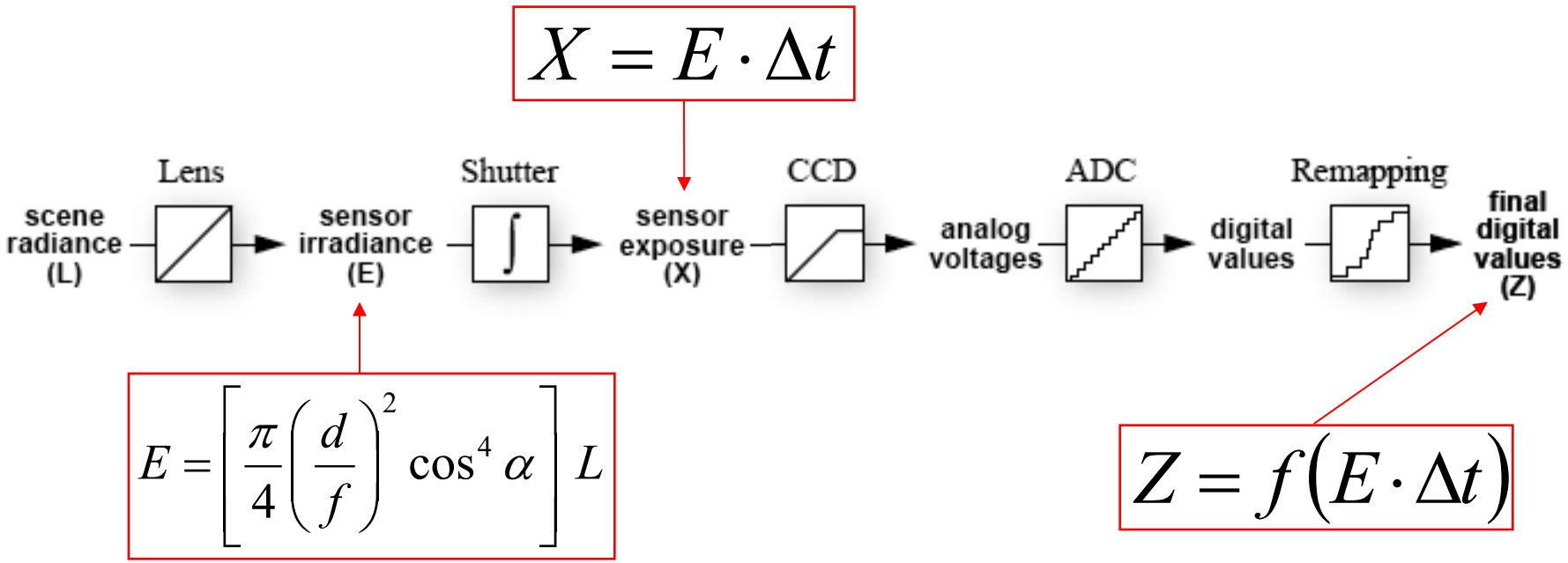
- From (1), (2), and (3):

$$E = L\,\frac{\pi}{4}\left(\frac{d}{f}\right)^2\cos\alpha^4$$

- Image irradiance is proportional to Scene Radiance!
- Small field of view → Effects of 4th power of cosine are small.

Slide from
S Narasimhan.

From light rays to pixel values

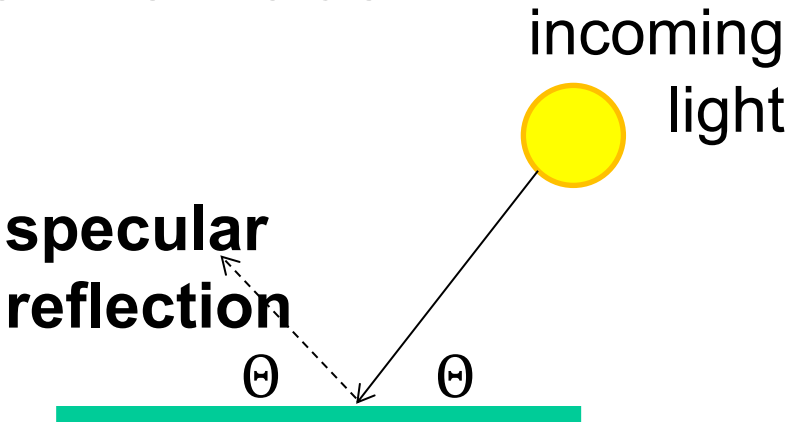


- Camera response function: the mapping f from irradiance to pixel values
  - Useful if we want to estimate material properties
  - Enables us to create *high dynamic range (HDR) images*
  - Classic reference: P. E. Debevec and J. Malik, *Recovering High Dynamic Range Radiance Maps from Photographs*, *SIGGRAPH 97*

Basic models of reflection

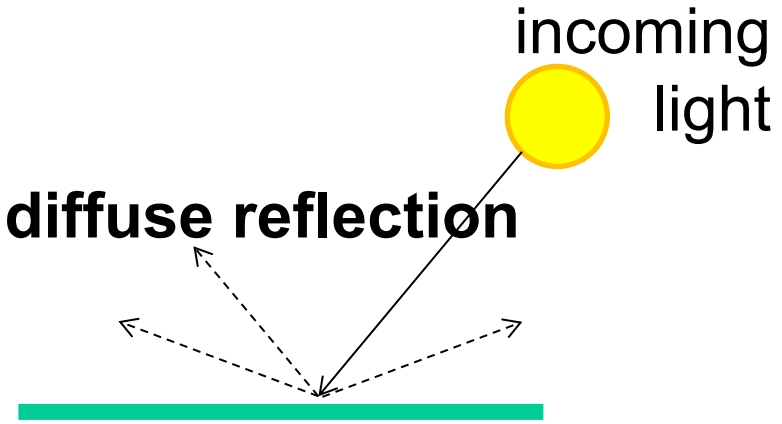
Specular: light bounces off at the incident angle

- E.g., mirror



Diffuse: light scatters in all directions

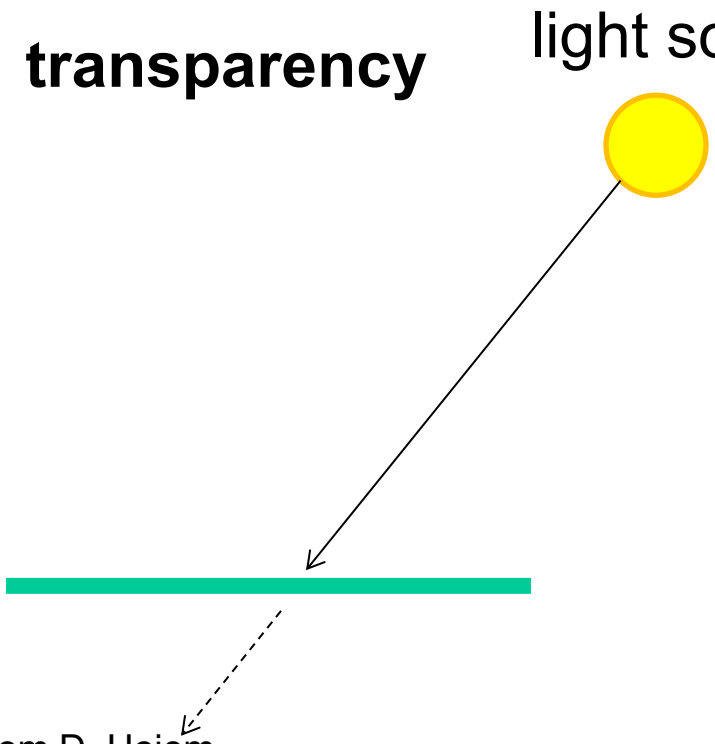
- E.g., brick, cloth, rough wood



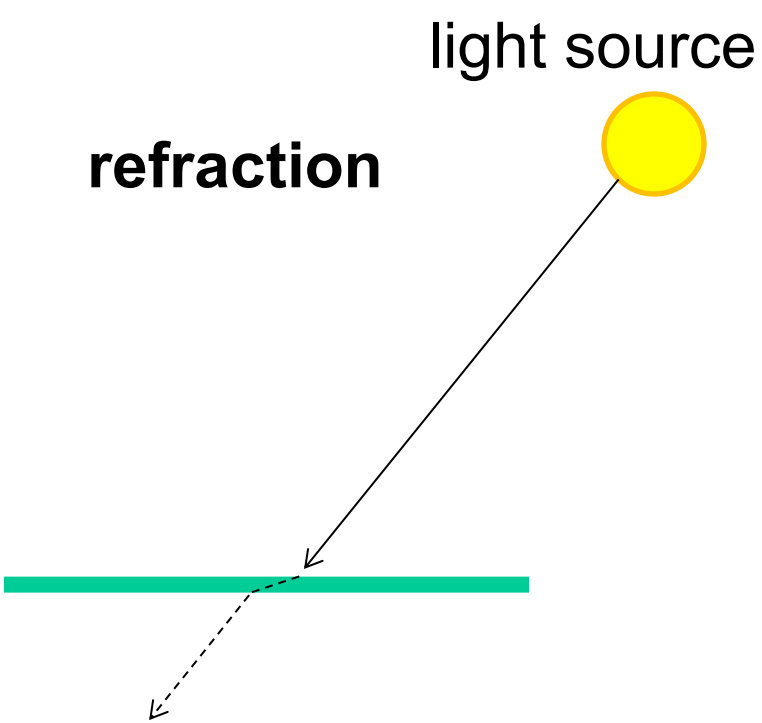
Slide from D. Hoiem

Other possible effects

transparency

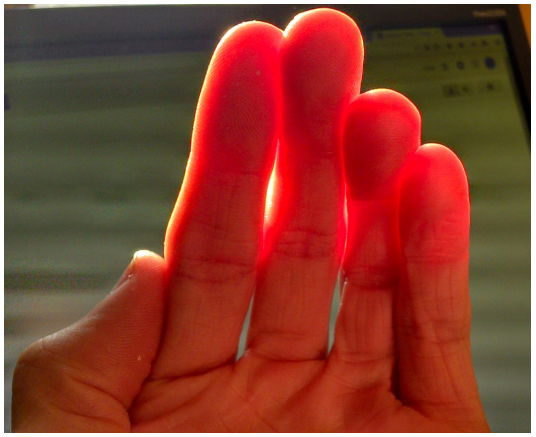


refraction

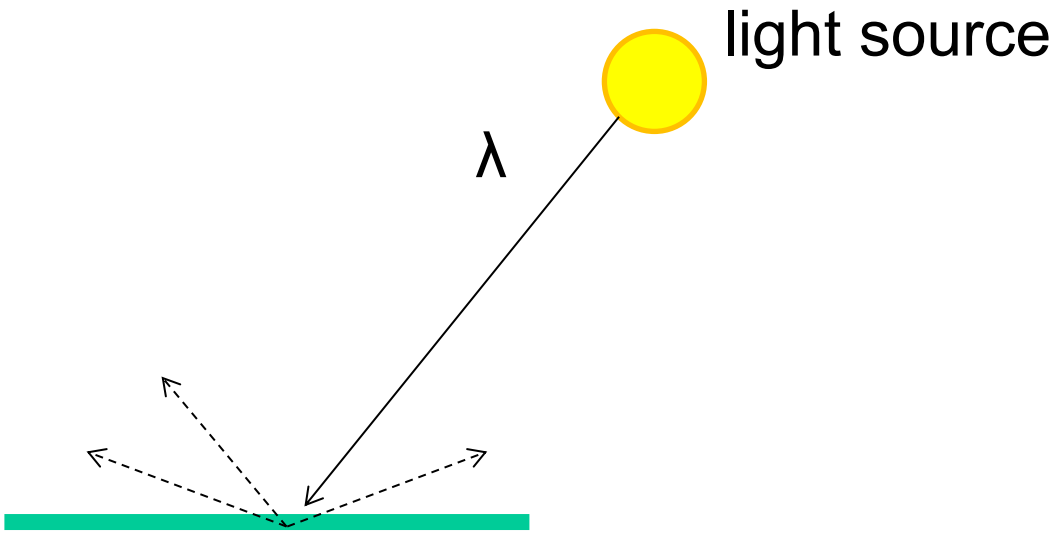


Slide from D. Hoiem

Other possible effects



subsurface scattering

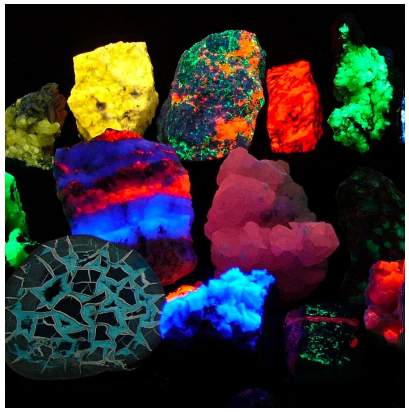


Slide from D. Hoiem

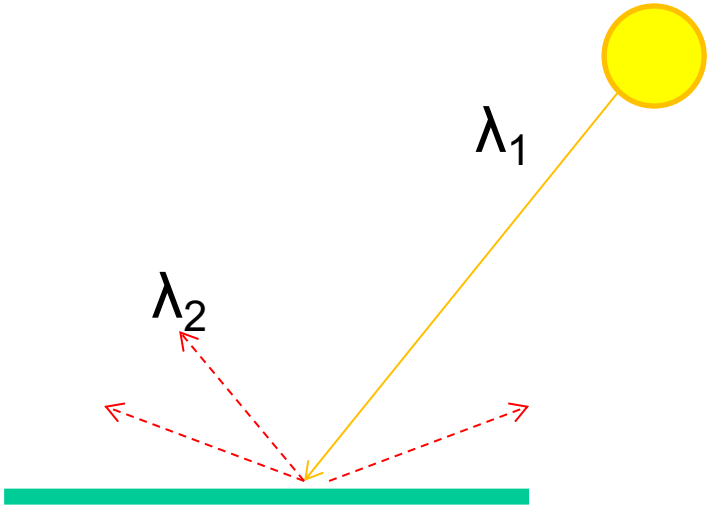
https://en.wikipedia.org/wiki/Subsurface_scattering#/media/File:Skin_Subsurface_Scattering.jpg

Other possible effects

fluorescence

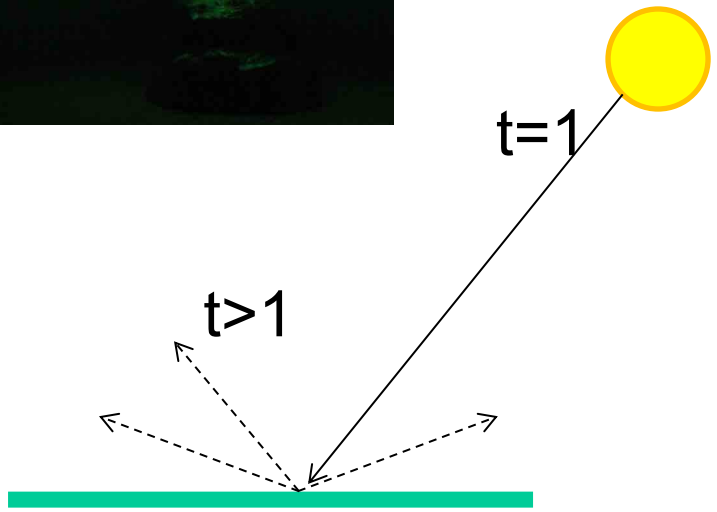


light source



phosphorescence

light source



https://en.wikipedia.org/wiki/Fluorescence#/media/File:Fluorescent_minerals_hg.jpg

Slide from D. Hoiem

Overview

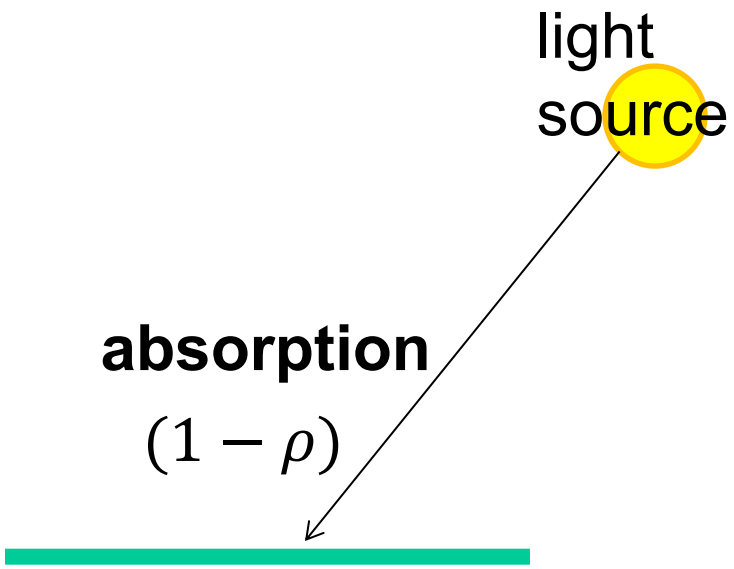
- Cameras with lenses
  - Depth of field
  - Field of view
  - Lens aberrations
- Brightness of a pixel
  - Small taste of radiometry
  - In-camera transformation of light
  - Reflectance properties of surfaces
  - Lambertian reflection model
  - Shape from shading

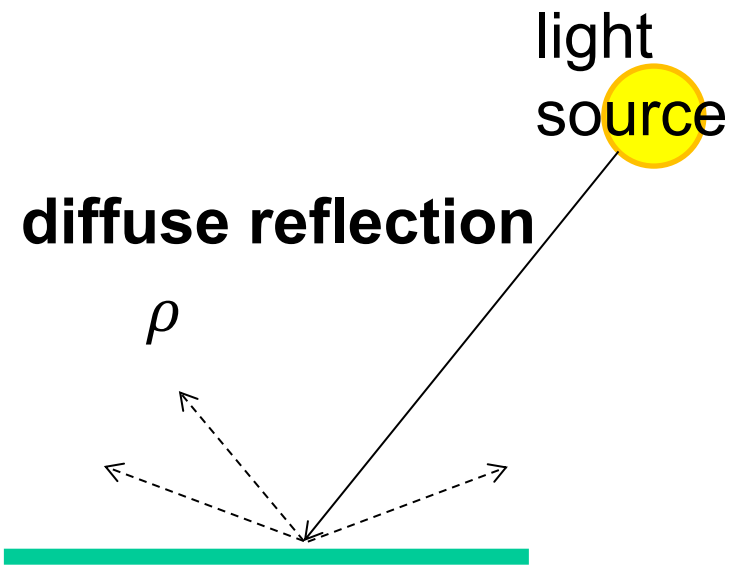
Lambertian reflectance model

Some light is absorbed (function of albedo ρ)

Remaining light is scattered (diffuse reflection)

Examples: soft cloth, concrete, matte paints





Slide from D. Hoiem

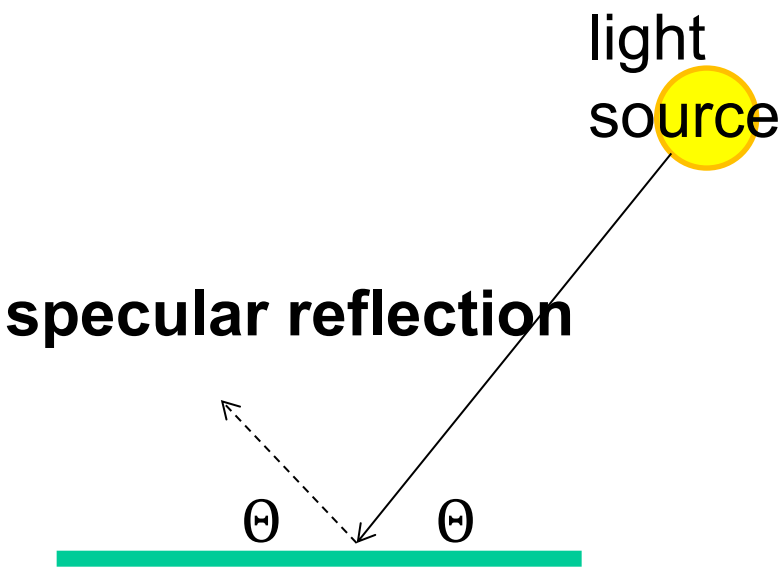
Specular Reflection

Reflected direction depends on light orientation and surface normal

- E.g., mirrors are fully specular

Flickr, by suzysputnik

Flickr, by piratejohnny



Slide from D. Hoiem

Most surfaces have both specular and diffuse components

Specularity = spot where specular reflection dominates (typically reflects light source)

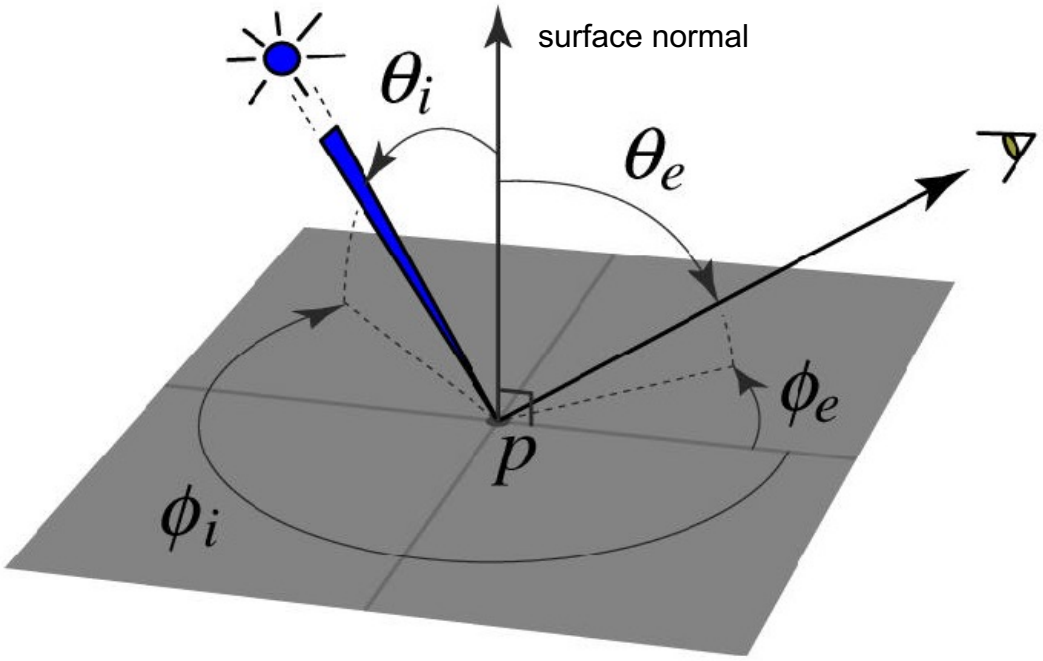
Photo: northcountryhardwoodfloors.com

Typically, specular component is small

Slide from D. Hoiem

BRDF: Bidirectional Reflectance Distribution Function

- Model of local reflection that tells how bright a surface appears when viewed from one direction when light falls on it from another
- Definition: ratio of the radiance in the emitted direction to irradiance in the incident direction



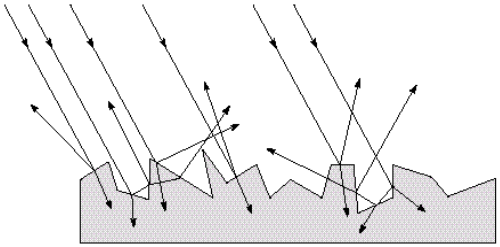
Slide credit: S. Savarese

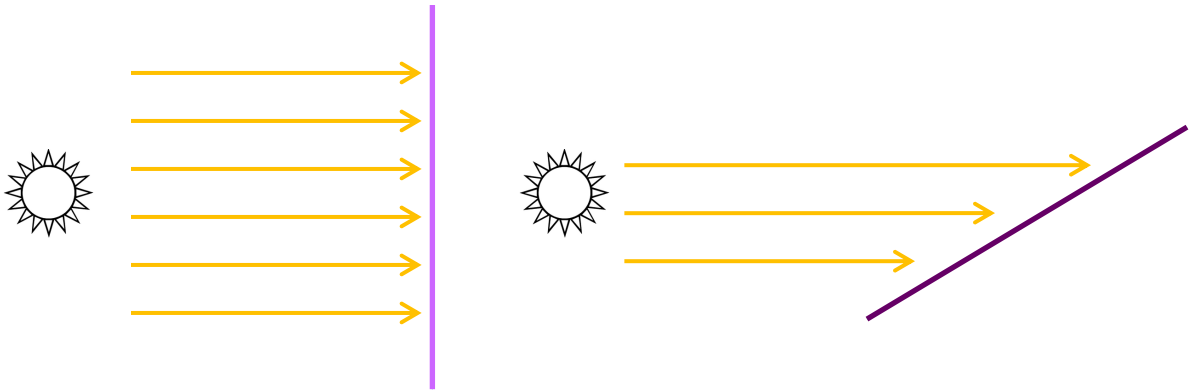
BRDFs can be incredibly complicated…

Diffuse reflection

- Light is reflected equally in all directions
  - Dull, matte surfaces like chalk or latex paint
  - Microfacets scatter incoming light randomly

- Brightness of the surface depends on the incidence of illumination

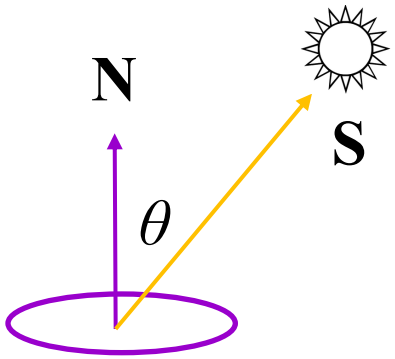




brighter

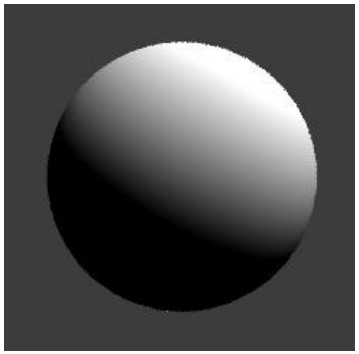
darker

Diffuse reflection: Lambert's law



$$B = \rho\, \mathbf{N} \cdot \mathbf{S}$$

$$= \rho \left\| \mathbf{S} \right\| \cos\theta$$

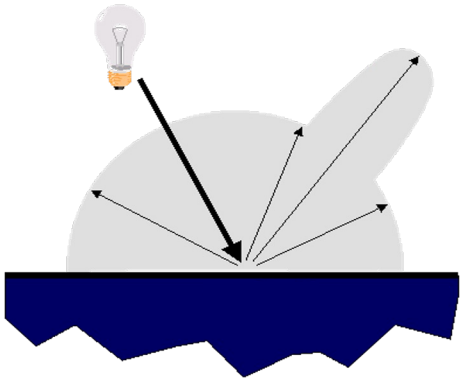


- *B*: radiosity (total power leaving the surface per unit area)
- ρ: albedo (fraction of incident irradiance reflected by the surface)
- *N*: unit normal
- *S*: source vector (magnitude proportional to intensity of the source)

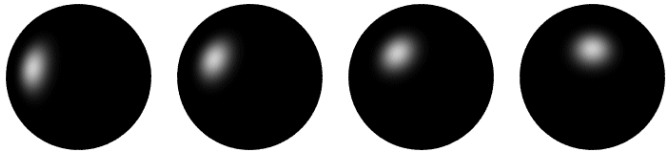
Slide from L. Lazebnik

Specular reflection

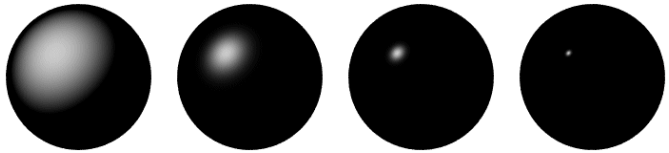
- Radiation arriving along a source direction leaves along the **specular direction** (source direction reflected about normal)
- On real surfaces, energy usually goes into a lobe of directions
- **Phong model**: reflected energy falls of with $\cos^n(\delta\theta)$

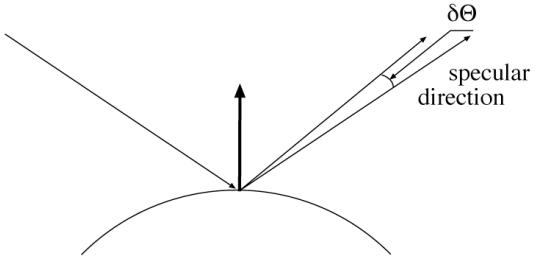


Moving the light source



Changing the exponent





Slide from L. Lazebnik

Specular reflection

Picture source

Slide from L. Lazebnik

Lambertian + Specular Model

- I(x) = Ambient Term + Diffuse Term + Specular Term

Photometric stereo (shape from shading)

- Can we reconstruct the shape of an object based on shading cues?

Luca della Robbia, *Cantoria*, 1438

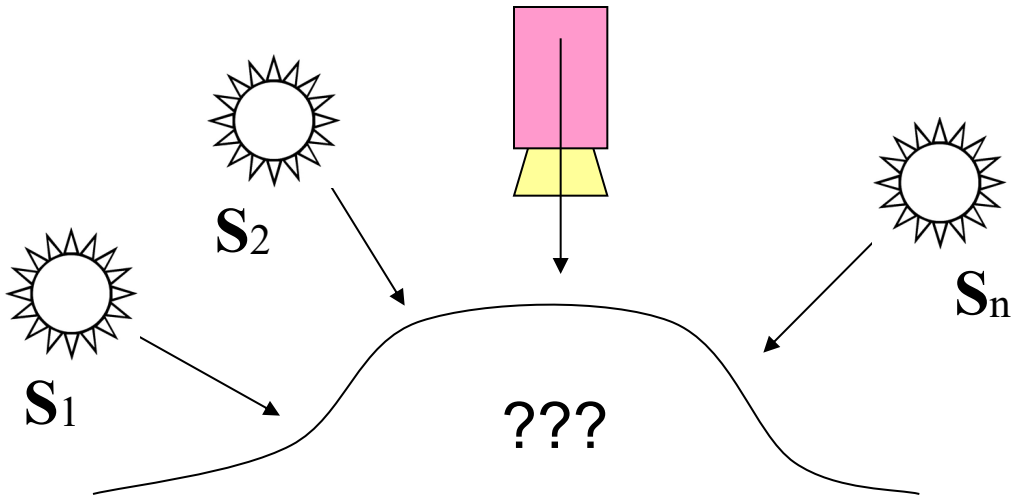
Slide from L. Lazebnik

Photometric stereo

Assume:

- A Lambertian object
- A *local shading model* (each point on a surface receives light only from sources visible at that point)
- A set of *known* light source directions
- A set of pictures of an object, obtained in exactly the same camera/object configuration but using different sources
- Orthographic projection

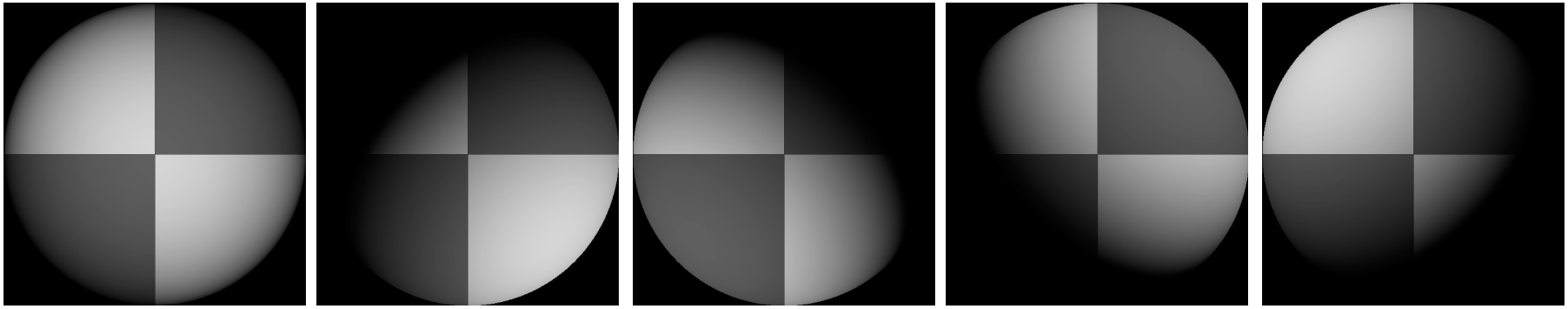
Goal: reconstruct object shape and albedo



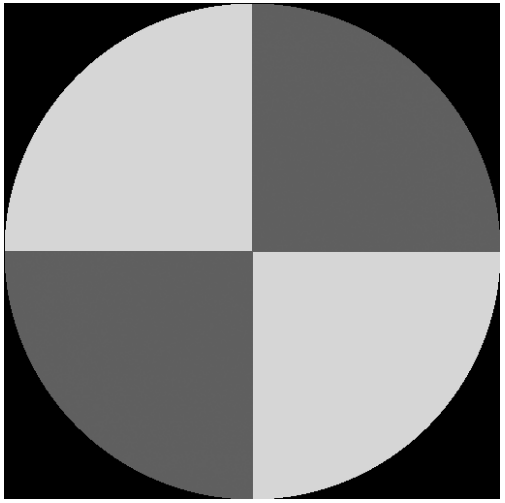
Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

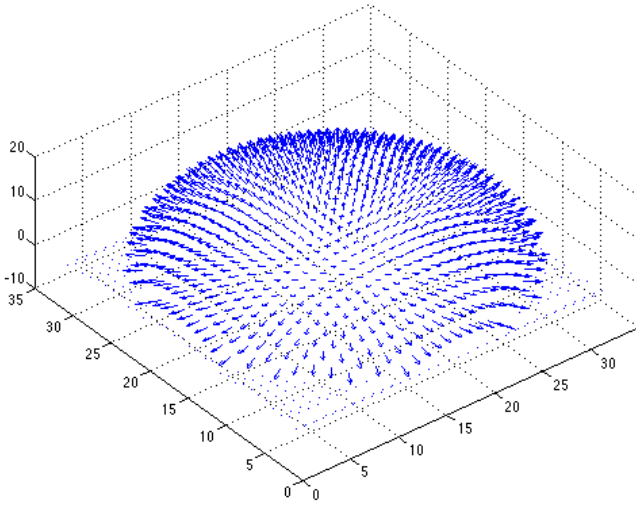
Example 1



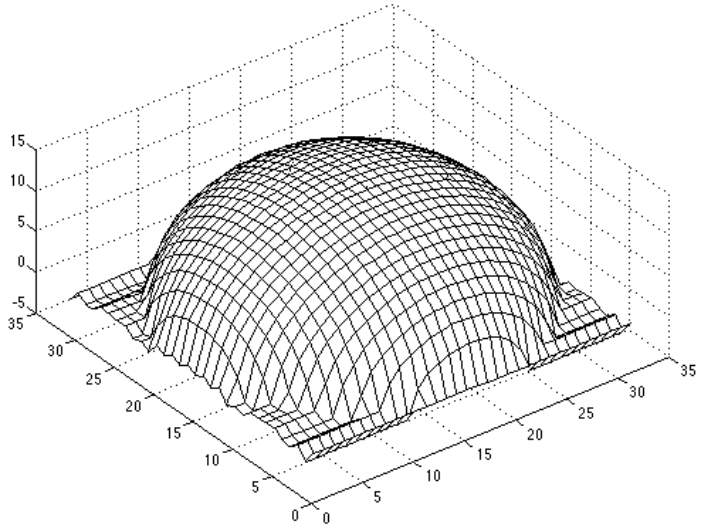
Recovered albedo



Recovered normal field



Recovered surface model



Slide from L. Lazebnik

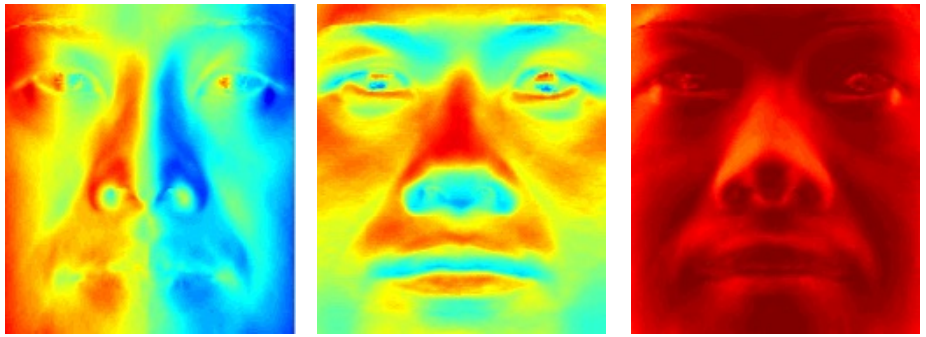
F&P 2nd ed., sec. 2.2.4

Example 2

Input

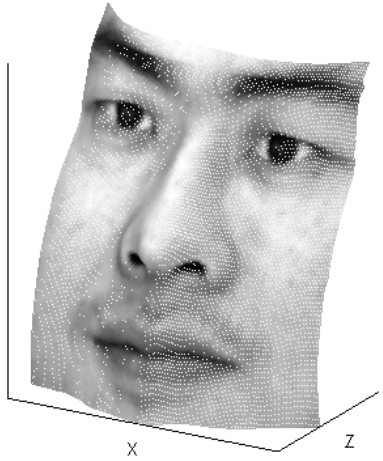
Recovered albedo

Recovered normal field



x y z

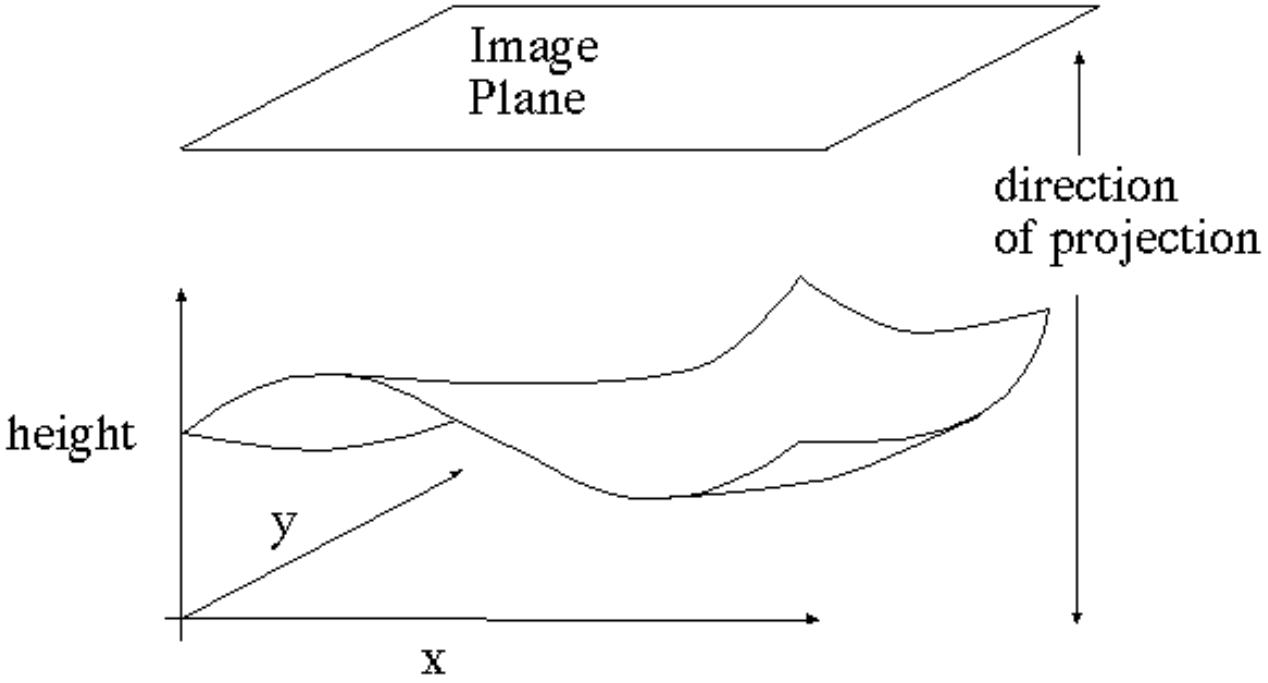
Recovered surface model



Slide from L. Lazebnik

Image model

- **Known:** source vectors $\mathbf{S}_j$ and pixel values $I_j(x,y)$
- **Unknown:** surface normal $\mathbf{N}(x,y)$ and albedo $\rho(x,y)$



Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Image model

- **Known:** source vectors $\mathbf{S}_j$ and pixel values $I_j(x,y)$
- **Unknown:** surface normal $\mathbf{N}(x,y)$ and albedo $\rho(x,y)$
- Assume that the response function of the camera is a linear scaling by a factor of k
- Lambert's law:

$$
\begin{aligned}
I_j(x,y) &= k\rho(x,y)\left(\mathbf{N}(x,y)\cdot\mathbf{S}_j\right) \\
&= \left(\rho(x,y)\mathbf{N}(x,y)\right)\cdot(k\mathbf{S}_j) \\
&= \mathbf{g}(x,y)\cdot\mathbf{V}_j
\end{aligned}
$$

Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Least squares problem

- For each pixel, set up a linear system:

$$\begin{bmatrix} I_1(x,y) \\ I_2(x,y) \\ \vdots \\ I_n(x,y) \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \vdots \\ \mathbf{V}_n^T \end{bmatrix} \mathbf{g}(x,y)$$

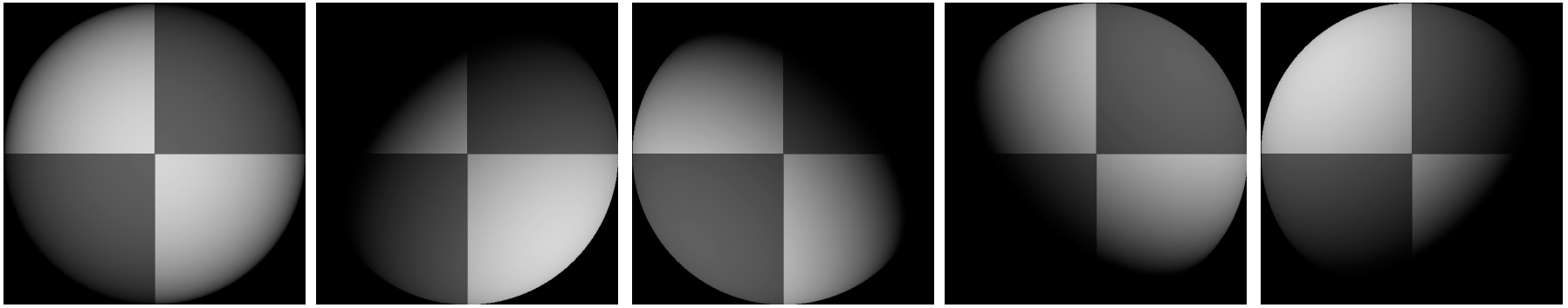
$(n \times 1)$ known — $(n \times 3)$ known — (3×1) unknown

- Obtain least-squares solution for $\mathbf{g}(x,y)$ (which we defined as $\mathbf{N}(x,y)\,\rho(x,y)$)
- Since $\mathbf{N}(x,y)$ is the unit normal, $\rho(x,y)$ is given by the magnitude of $\mathbf{g}(x,y)$
- Finally, $\mathbf{N}(x,y) = \mathbf{g}(x,y) / \rho(x,y)$

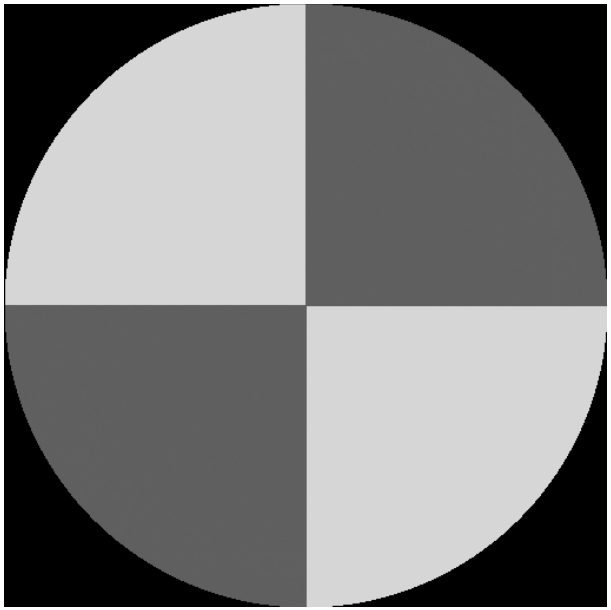
Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

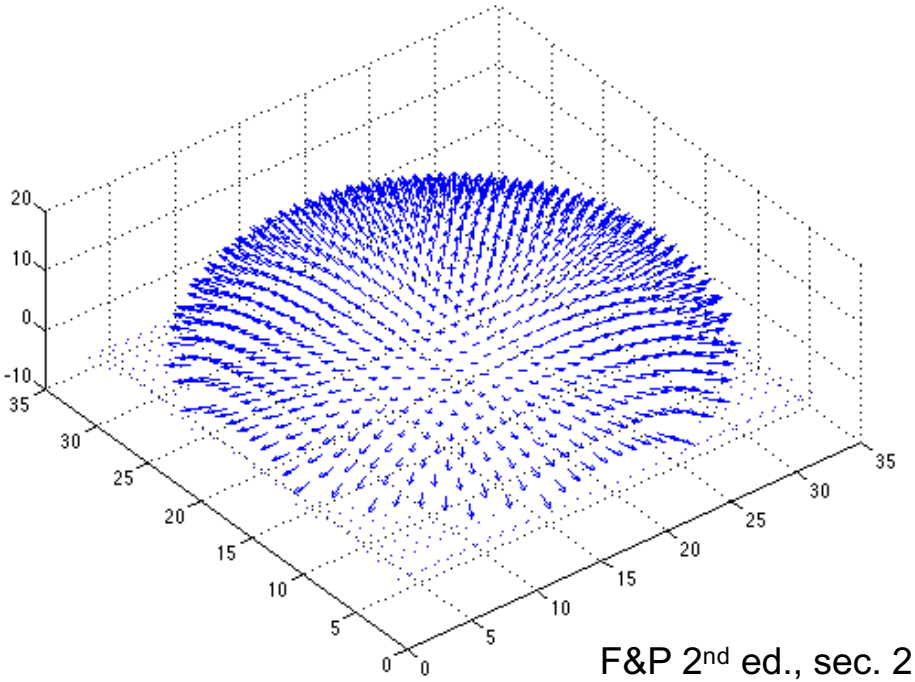
Synthetic example



Recovered albedo



Recovered normal field



Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Recovering a surface from normals

Recall the surface is written as

$$(x, y, f(x, y))$$

This means the normal has the form:

$$\mathbf{N}(x, y) = \frac{1}{\sqrt{f_x^2 + f_y^2 + 1}} \begin{pmatrix} f_x \\ f_y \\ 1 \end{pmatrix}$$

If we write the estimated vector *g* as

$$\mathbf{g}(x, y) = \begin{pmatrix} g_1(x, y) \\ g_2(x, y) \\ g_3(x, y) \end{pmatrix}$$

Then we obtain values for the partial derivatives of the surface:

$$f_x(x, y) = g_1(x, y) / g_3(x, y)$$
$$f_y(x, y) = g_2(x, y) / g_3(x, y)$$

Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Recovering a surface from normals

We can now recover the surface height at any point by integration along some path, e.g.

$$f(x,y) = \int_0^x f_x(s,0)\,ds + \int_0^y f_y(x,t)\,dt + C$$

(for robustness, should take integrals over many different paths and average the results)

Integrability: for the surface f to exist, the mixed second partial derivatives must be equal:

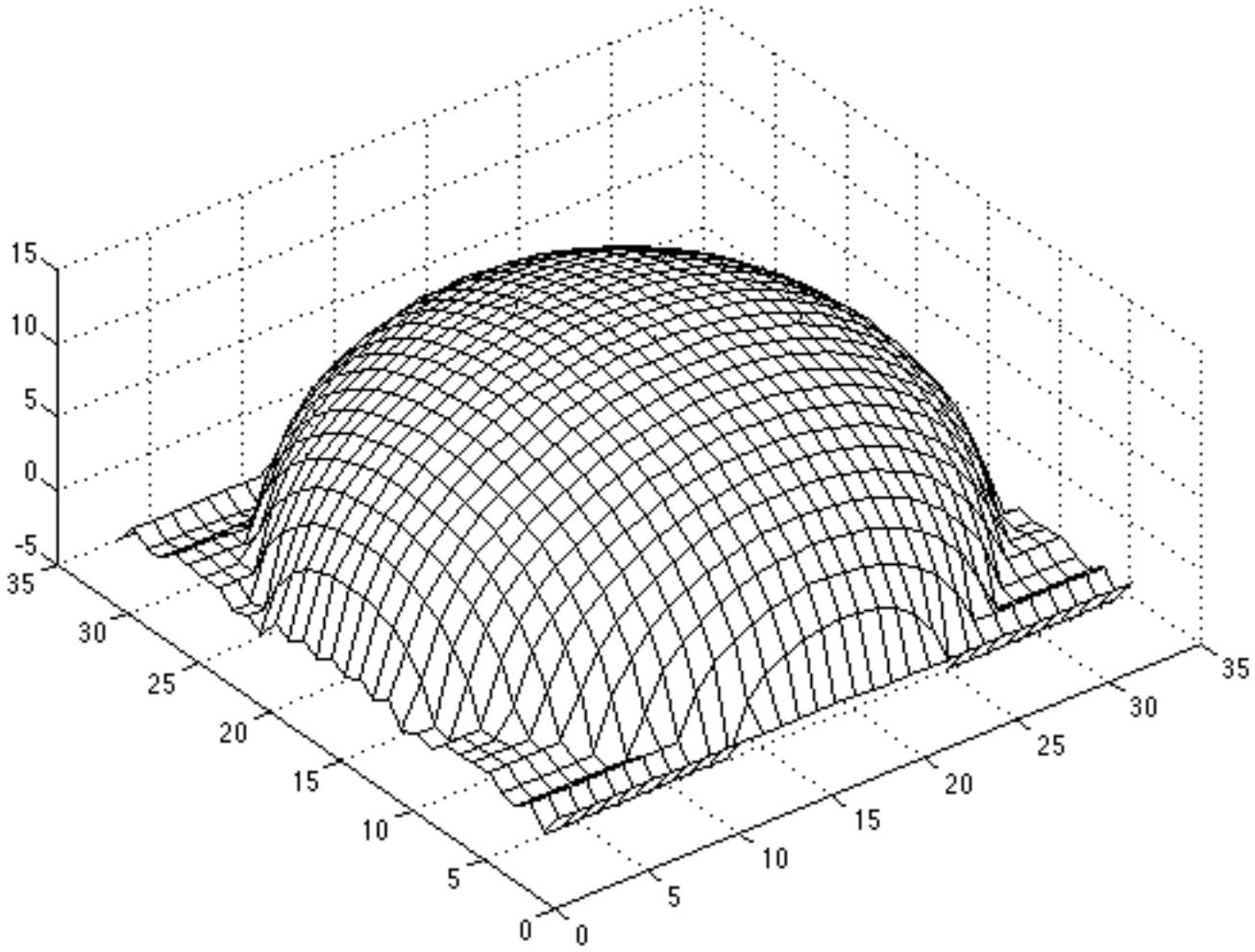
$$\frac{\partial}{\partial y}(g_1(x,y)/g_3(x,y)) = \frac{\partial}{\partial x}(g_2(x,y)/g_3(x,y))$$

(in practice, they should at least be similar)

Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Surface recovered by integration



Slide from L. Lazebnik

F&P 2nd ed., sec. 2.2.4

Limitations

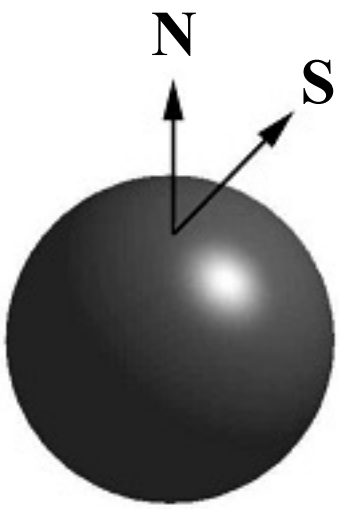
- Orthographic camera model
- Simplistic reflectance and lighting model
- No shadows
- No interreflections
- No missing data
- Integration is tricky

Slide from L. Lazebnik

Finding the direction of the light source

$$I(x,y) = \mathbf{N}(x,y) \cdot \mathbf{S}(x,y)$$

Full 3D case:

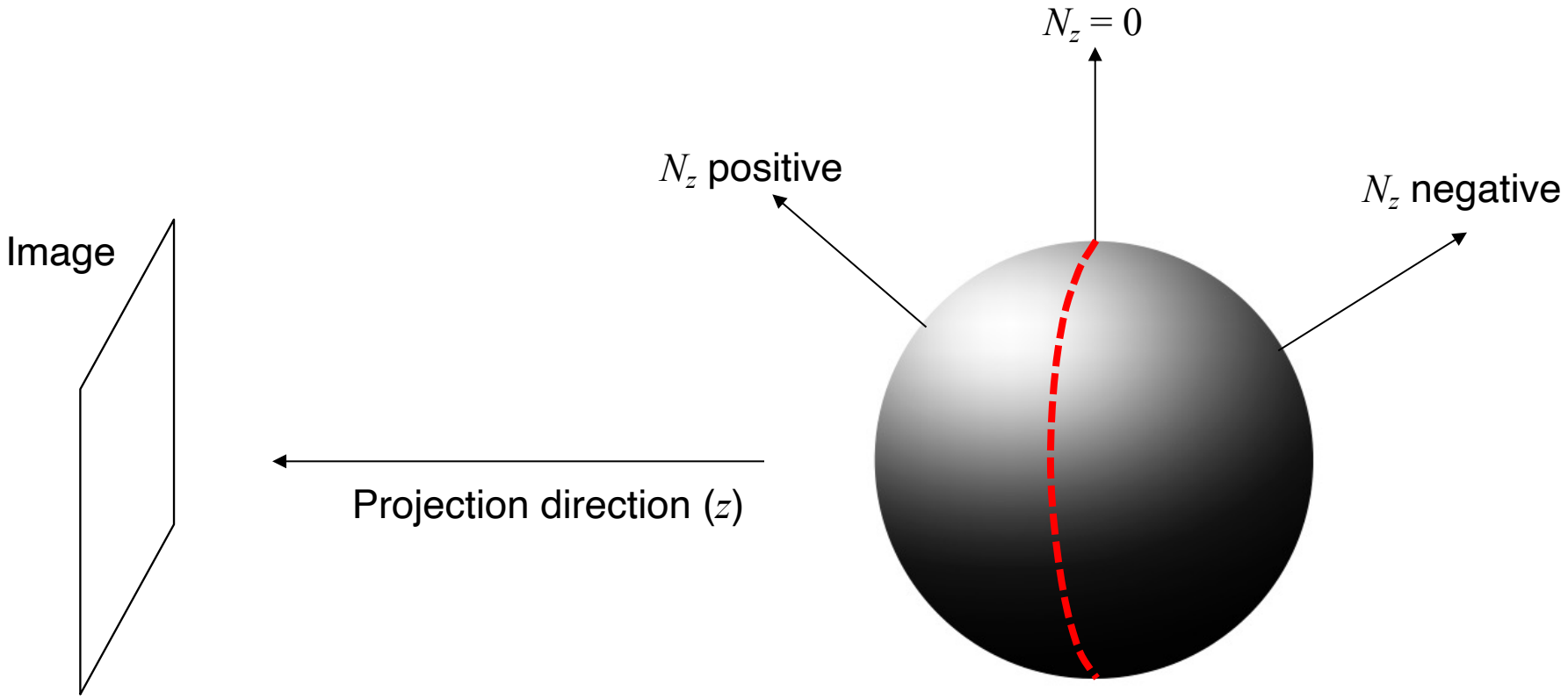


$$\begin{pmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) & N_z(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) & N_z(x_2,y_2) \\ \vdots & \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) & N_z(x_n,y_n) \end{pmatrix} \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} = \begin{pmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{pmatrix}$$

P. Nillius and J.-O. Eklundh, “Automatic estimation of the projected light source direction,” CVPR 2001

Finding the direction of the light source

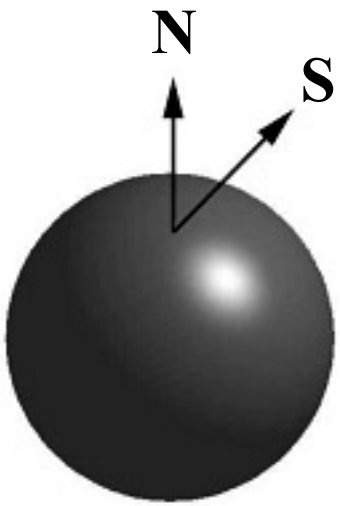
Consider points on the *occluding contour*:



P. Nillius and J.-O. Eklundh, “Automatic estimation of the projected light source direction,” CVPR 2001

Finding the direction of the light source

$$I(x,y) = \mathbf{N}(x,y) \cdot \mathbf{S}(x,y)$$



Full 3D case:

$$\begin{pmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) & N_z(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) & N_z(x_2,y_2) \\ \vdots & \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) & N_z(x_n,y_n) \end{pmatrix} \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} = \begin{pmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{pmatrix}$$

For points on the *occluding contour,* $N_z = 0$:

$$\begin{pmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) \\ \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) \end{pmatrix} \begin{pmatrix} S_x \\ S_y \end{pmatrix} = \begin{pmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{pmatrix}$$

P. Nillius and J.-O. Eklundh, “Automatic estimation of the projected light source direction,” CVPR 2001

Finding the direction of the light source

P. Nillius and J.-O. Eklundh, “Automatic estimation of the projected light source direction,” CVPR 2001

Application: Detecting composite photos

Real photo

Fake photo

M. K. Johnson and H. Farid, Exposing Digital Forgeries by Detecting Inconsistencies in Lighting, ACM Multimedia and Security Workshop, 2005.

Color

CS 543 / ECE 549 – Saurabh Gupta

Many slides adapted from S. Seitz, L. Lazebnik, J. Hayes, S. Palmer

While we wait, what color is the dress?

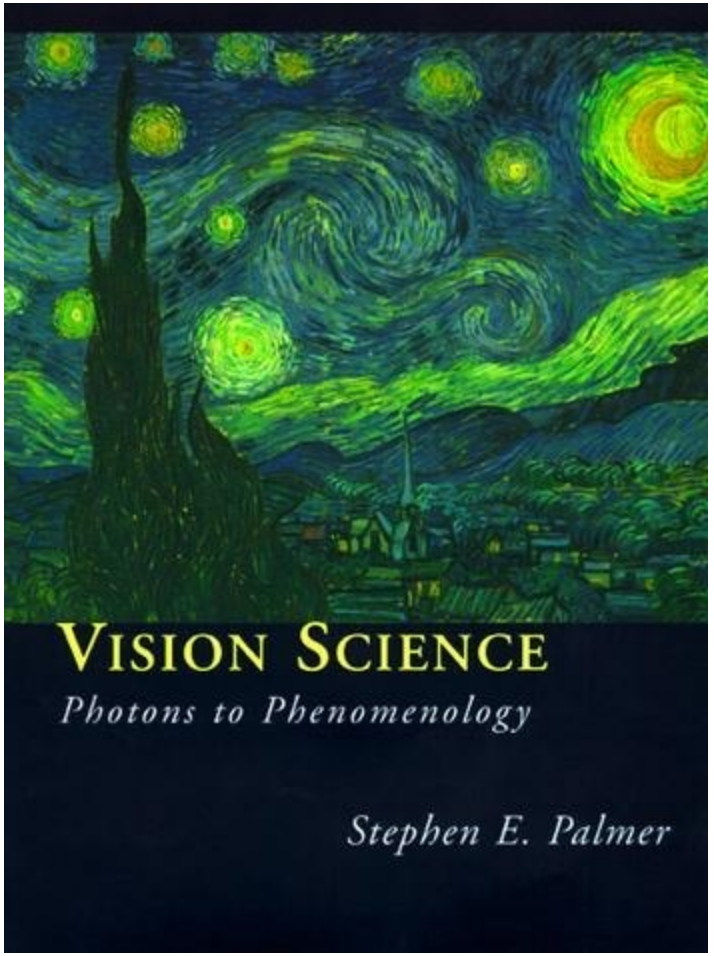
- A. White and gold
- B. Blue and Black

https://www.wired.com/2015/02/science-one-agrees-color-dress/

Slide by L. Lazebnik

What is color?

- Color is the result of interaction between physical light in the environment and our visual system
- Color is a psychological property of our visual experiences when we look at objects and lights, *not* a physical property of those objects or lights (S. Palmer, *Vision Science: Photons to Phenomenology*)

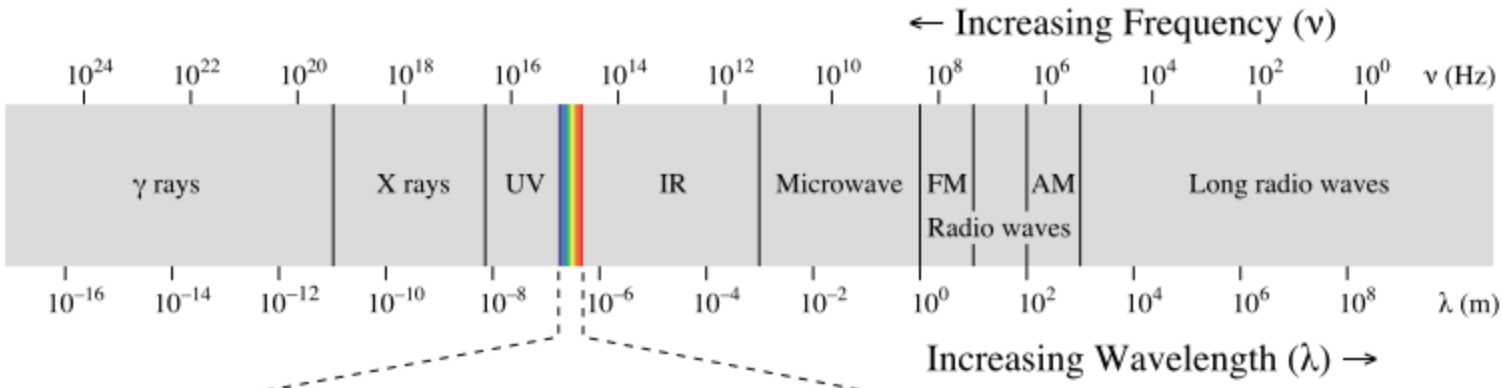


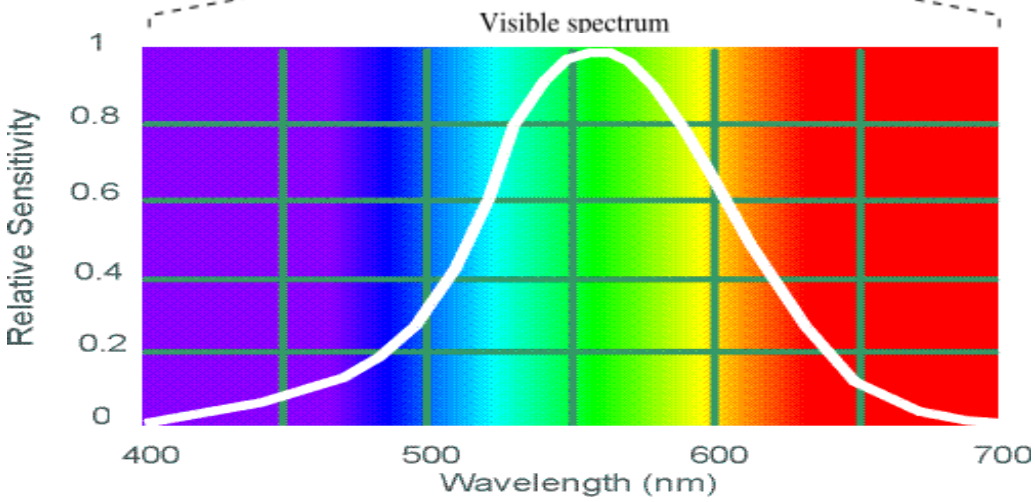
Slide by L. Lazebnik

Outline

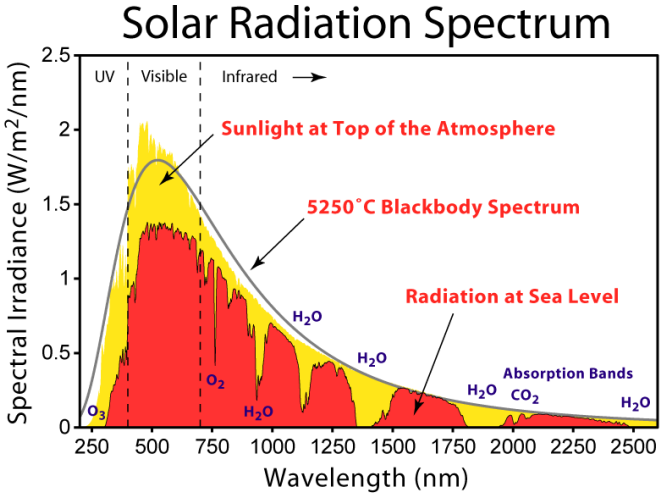
- Physical origin of color
- Spectra of sources and surfaces
- Physiology of color vision
- Trichromatic color theory
- Color spaces
- Color constancy, white balance

Electromagnetic spectrum





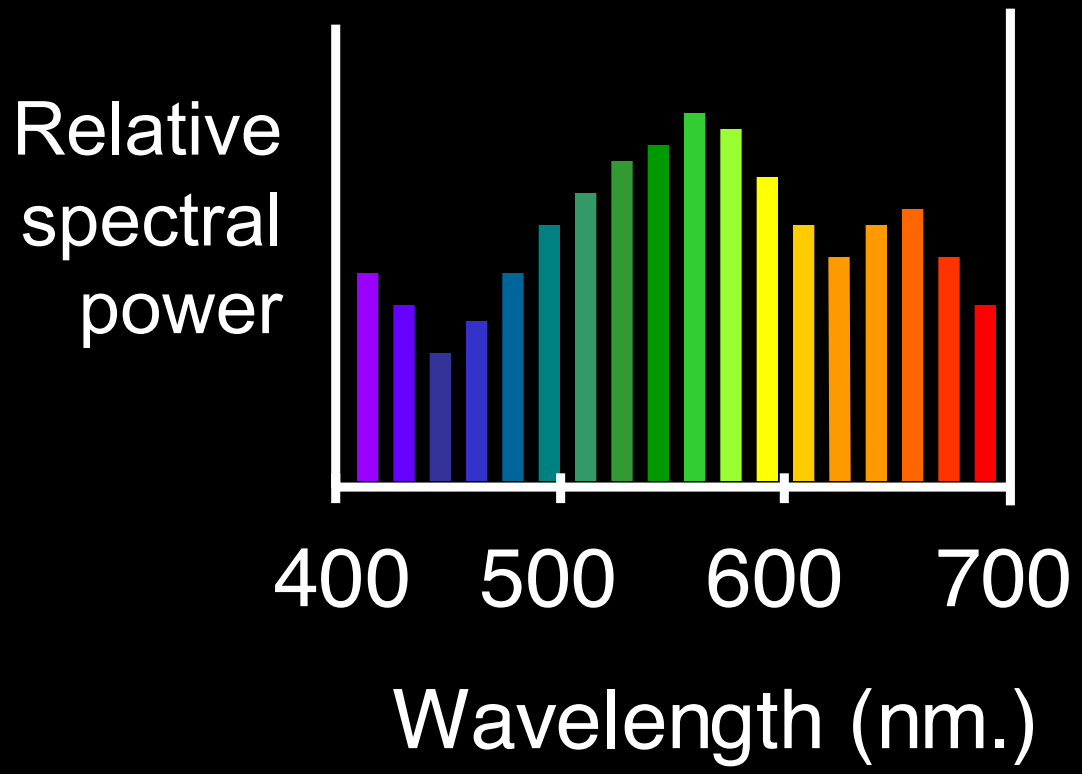
Human Luminance Sensitivity Function



Slide by L. Lazebnik

The Physics of Light

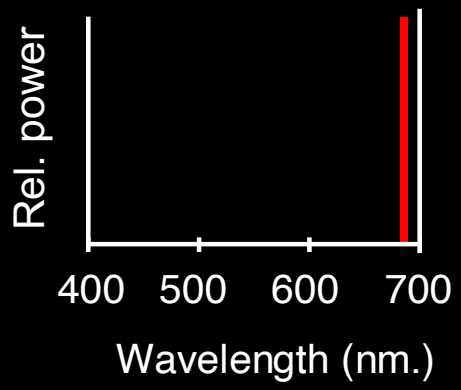
Any source of light can be completely described physically by its spectrum: the amount of energy emitted (per time unit) at each wavelength 400 - 700 nm.



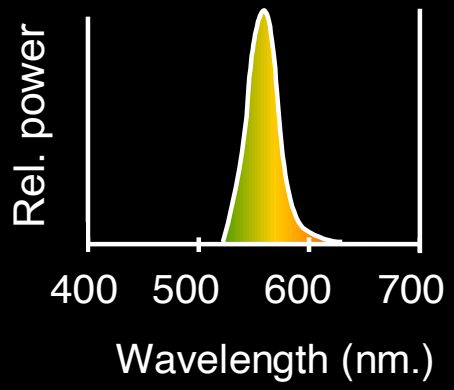
Spectra of Light Sources

Some examples of the spectra of light sources

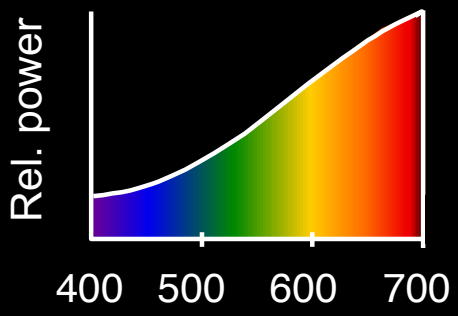
A. Ruby Laser



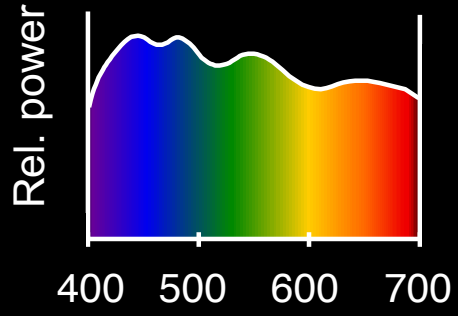
B. Gallium Phosphide Crystal



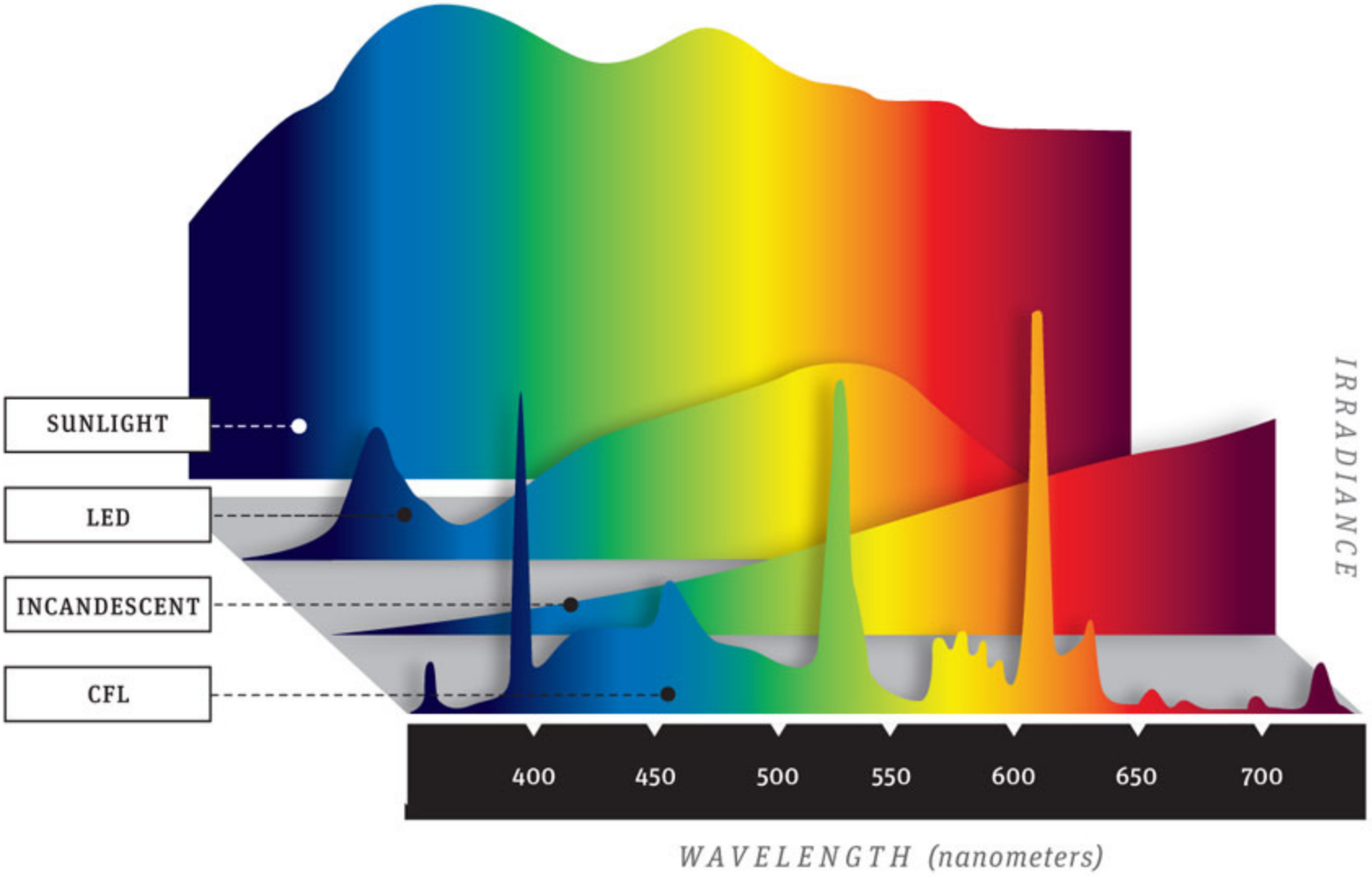
C. Tungsten Lightbulb



D. Normal Daylight



Spectra of light sources



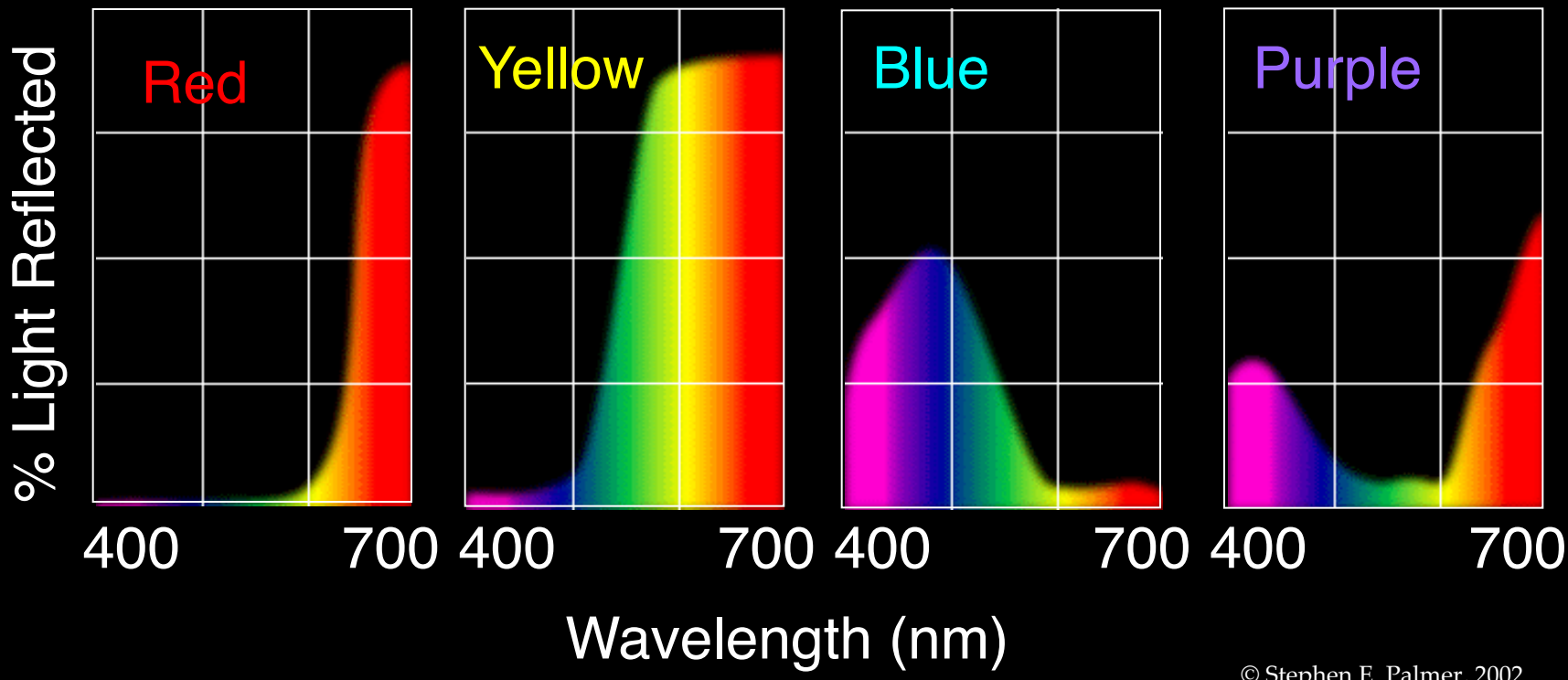
Source: Popular Mechanics

Slide by L. Lazebnik

Reflectance Spectra of Surfaces

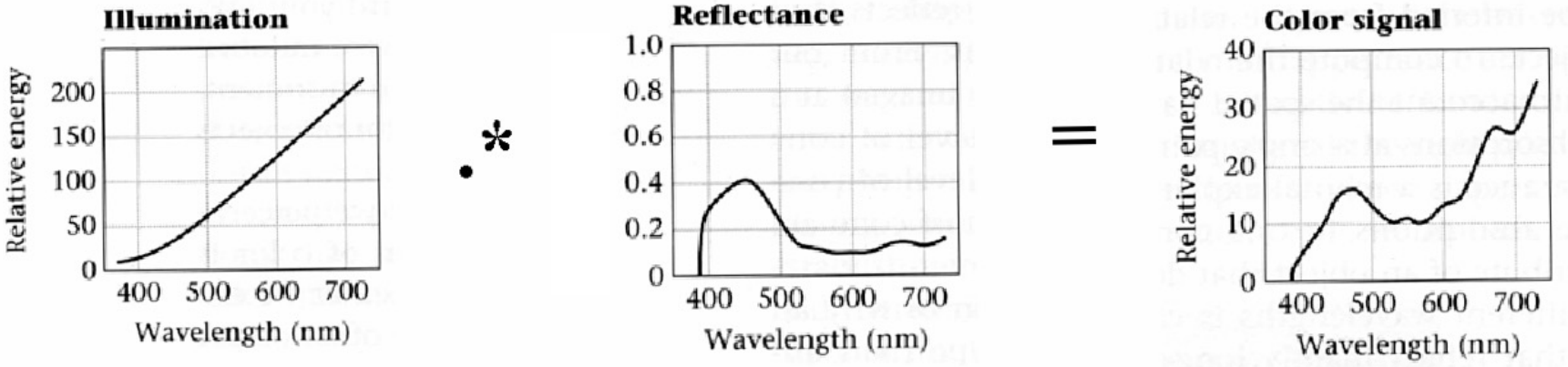
Some examples of the reflectance spectra of surfaces



Interaction of light and surfaces

- Reflected color is the result of interaction of light source spectrum with surface reflectance

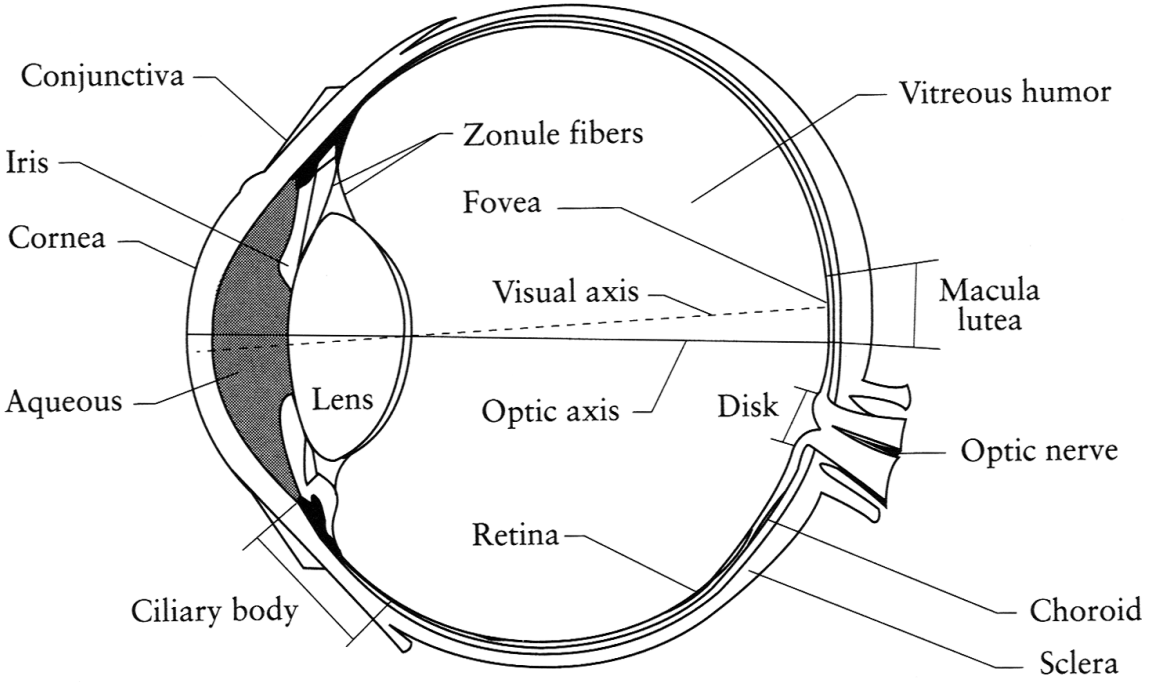


Slide by L. Lazebnik

So, what is the dimensionality of color?

- A. 3
- B. (255)^3
- C. 255
- D. Infinite

The Eye

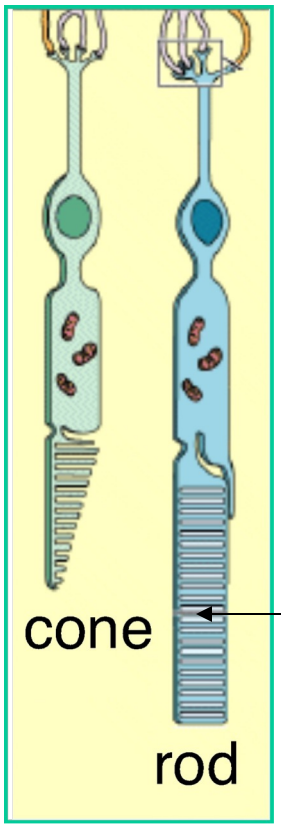


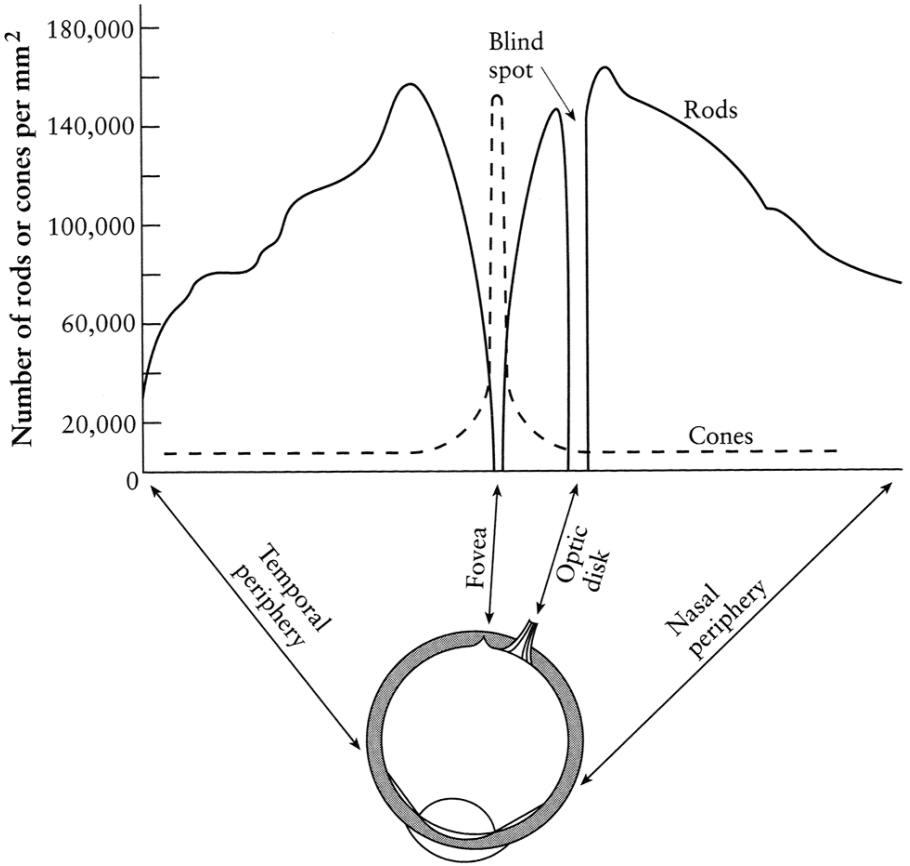
The human eye is a camera!

- **Lens** - changes shape by using ciliary muscles (to focus on objects at different distances)
- **Pupil** - the hole (aperture) whose size is controlled by the iris
- **Iris** - colored annulus with radial muscles
- **Retina** - photoreceptor cells

Slide by Steve Seitz

Rods and cones, fovea





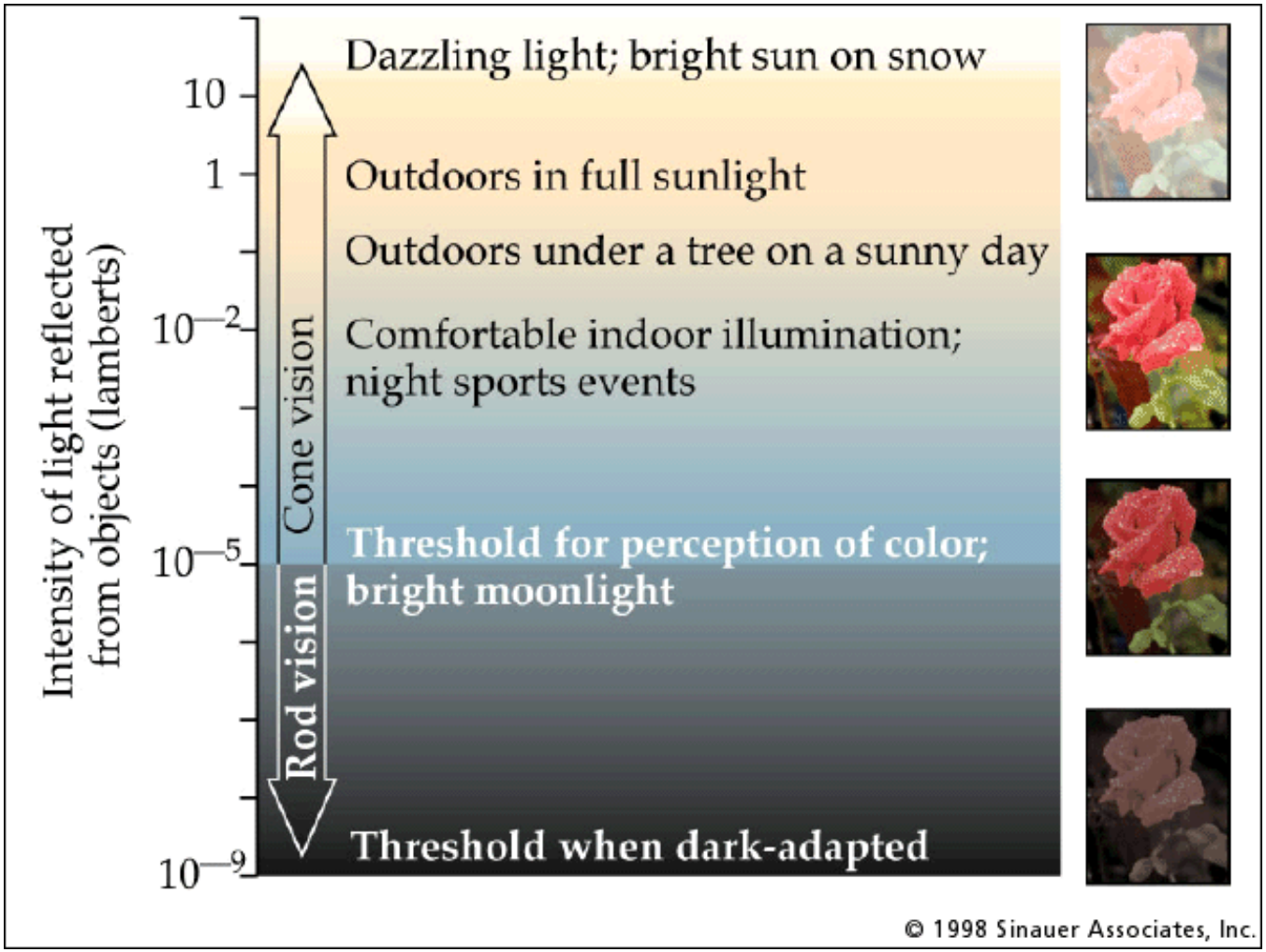
Rods are responsible for intensity, cones for color perception

Rods and cones are *non-uniformly* distributed on the retina

- **Fovea** - Small region (1 or 2°) at the center of the visual field containing the highest density of cones – *and no rods*

Slide by Steve Seitz

Rod / Cone sensitivity

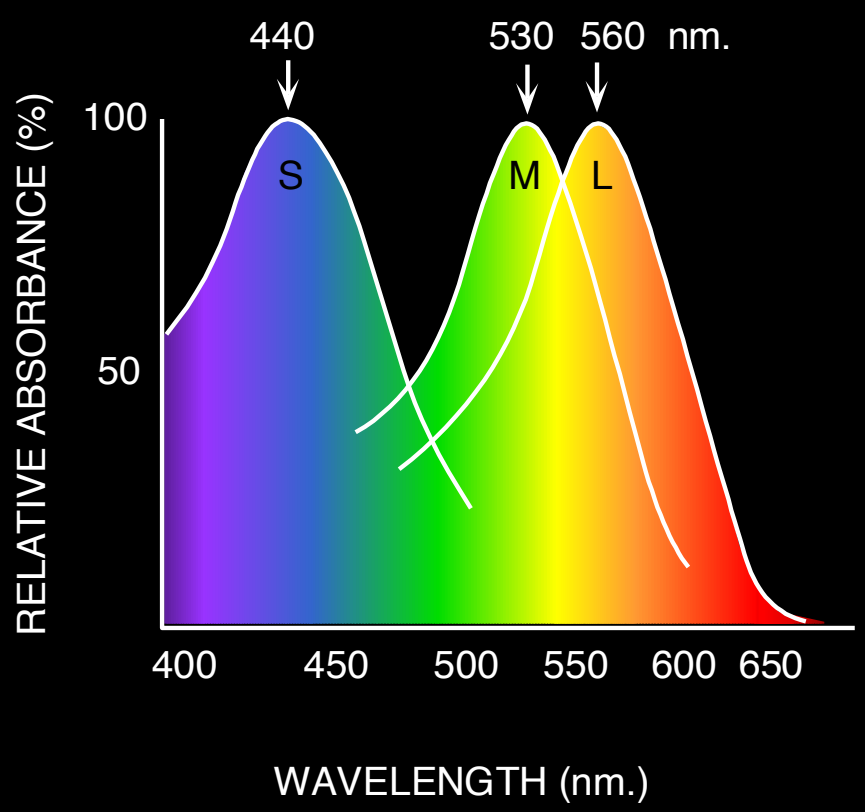


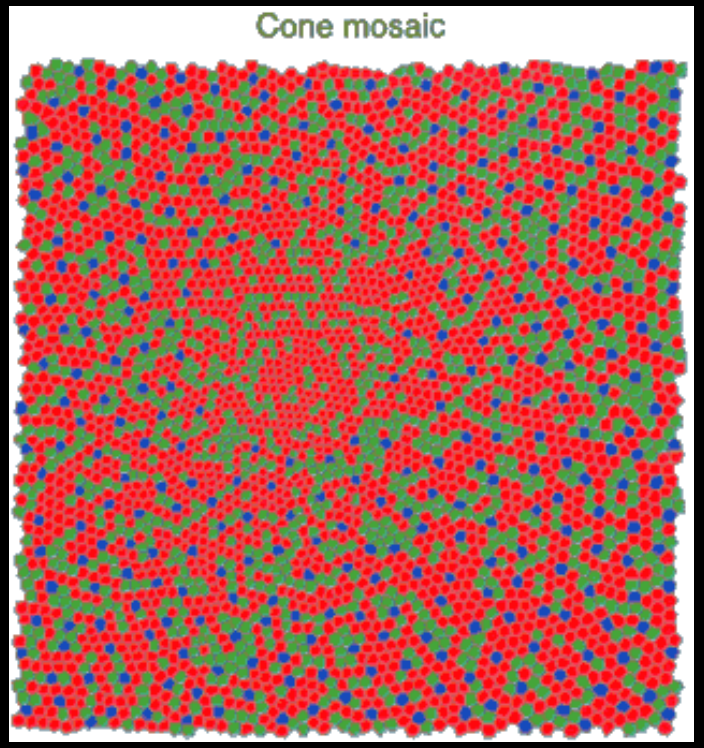
Why can't we read in the dark?

Slide by A. Efros

Physiology of Color Vision

Three kinds of cones:





- Ratio of L to M to S cones: approx. 10:5:1
- Almost no S cones in the center of the fovea

Physiology of Color Vision: Fun facts

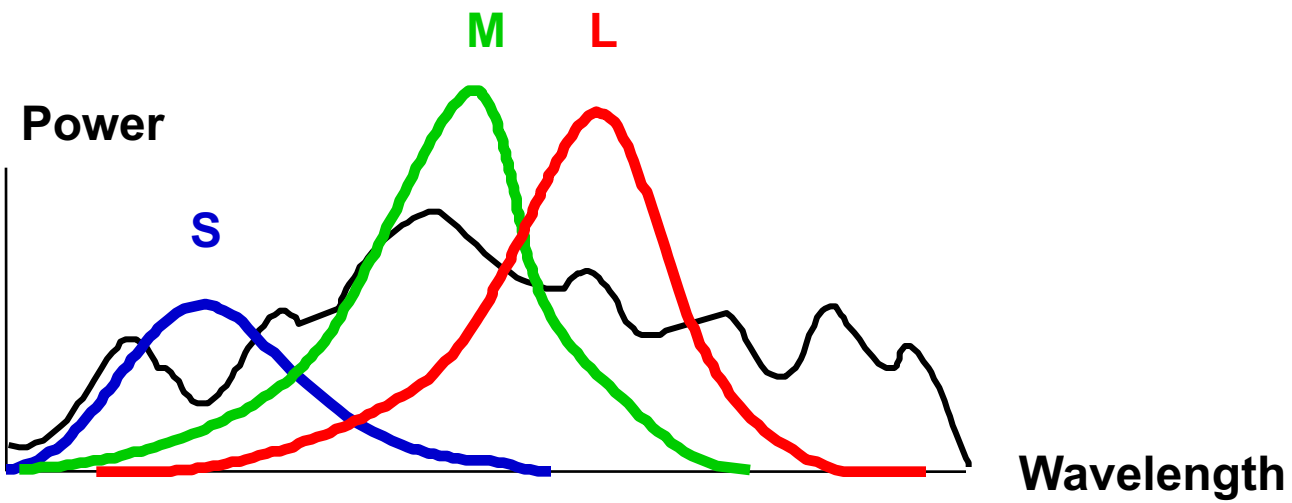
- "M" and "L" pigments are encoded on the X-chromosome
  - That's why men are more likely to be color blind
  - "L" gene has high variation, so some women may be *tetrachromatic*
- Some animals have one (night animals), two (e.g., dogs), four (fish, birds), five (pigeons, some reptiles/amphibians), or even 12 (mantis shrimp) types of cones

http://ngm.nationalgeographic.com/2016/02/evolution-of-eyes-text

http://en.wikipedia.org/wiki/Color_vision

Slide by D. Hoiem

Color perception



Rods and cones act as *filters* on the spectrum

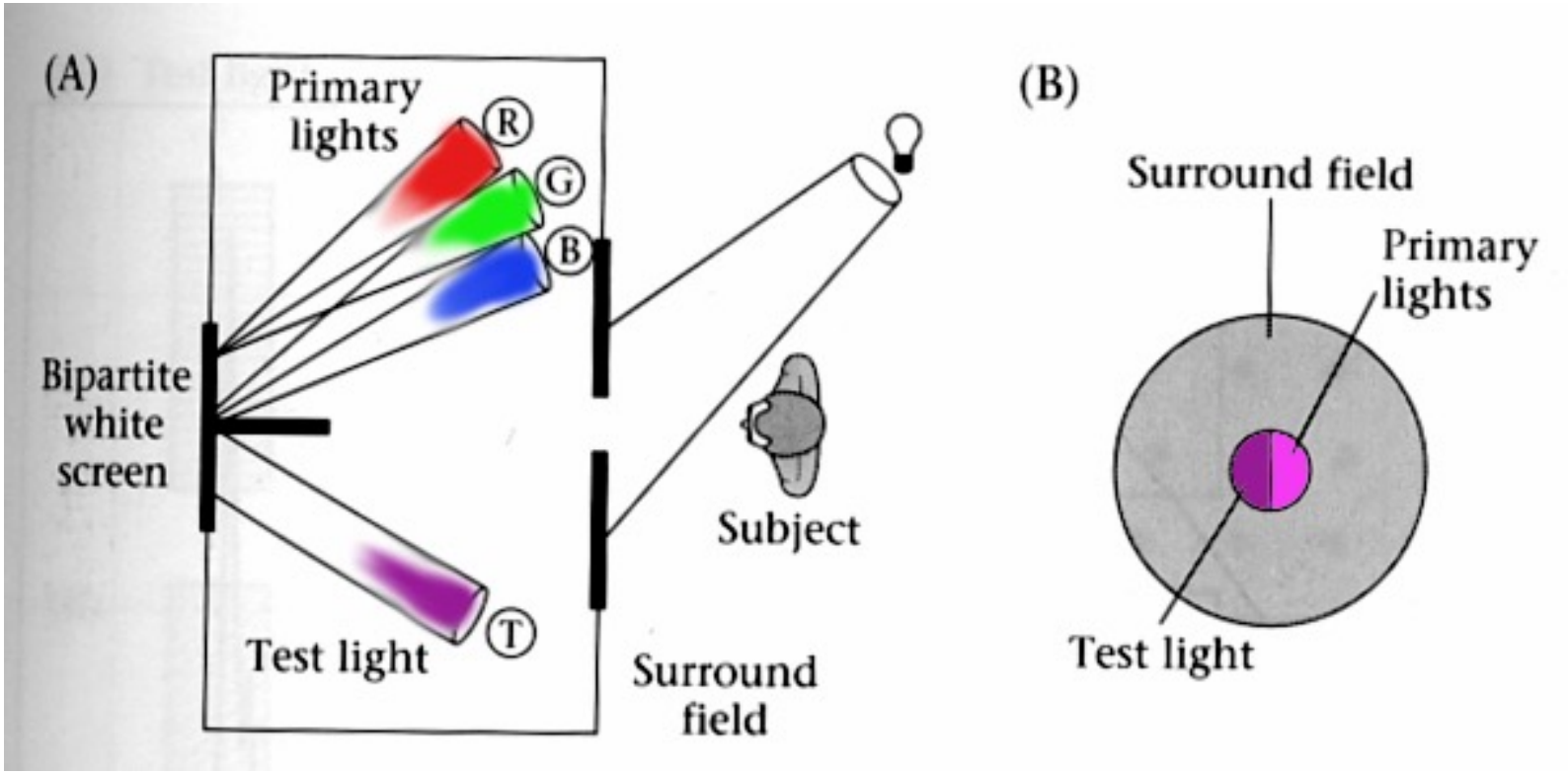
- To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths
  - Each cone yields one number

- How can we represent an entire spectrum with three numbers?
- We can't! Most of the information is lost
  - As a result, two different spectra may appear indistinguishable
    - » such spectra are known as **metamers**

Slide by Steve Seitz

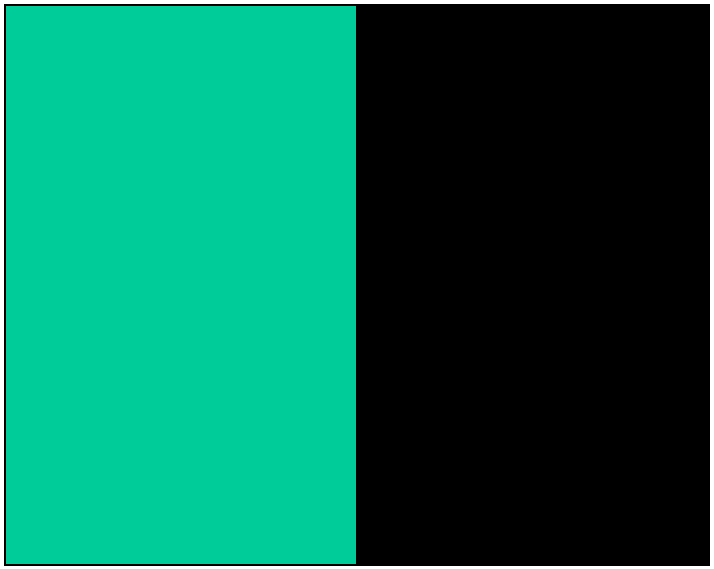
Color matching experiments

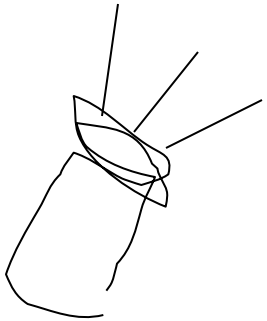
- We would like to understand which spectra produce the same color sensation in people under similar viewing conditions



Wandell, Foundations of Vision, 1995

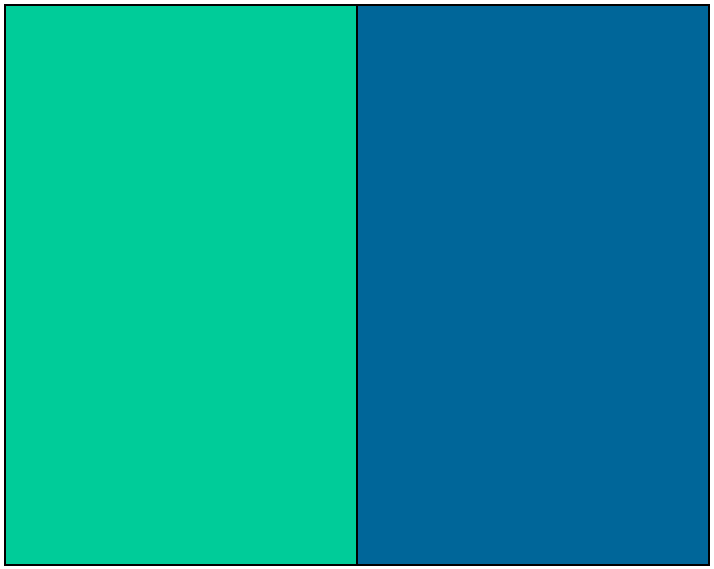
Color matching experiment 1

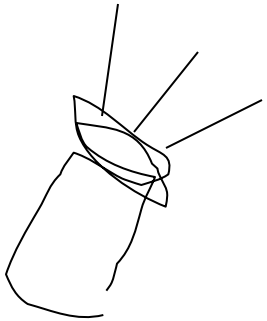


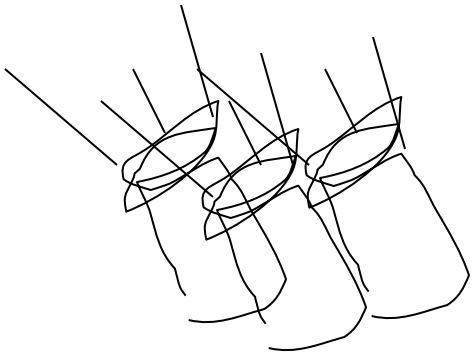


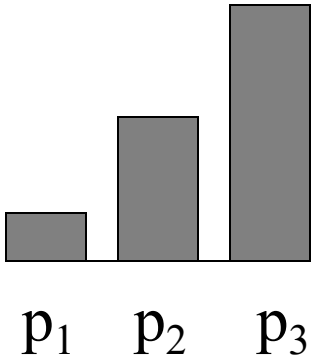
Source: W. Freeman

Color matching experiment 1



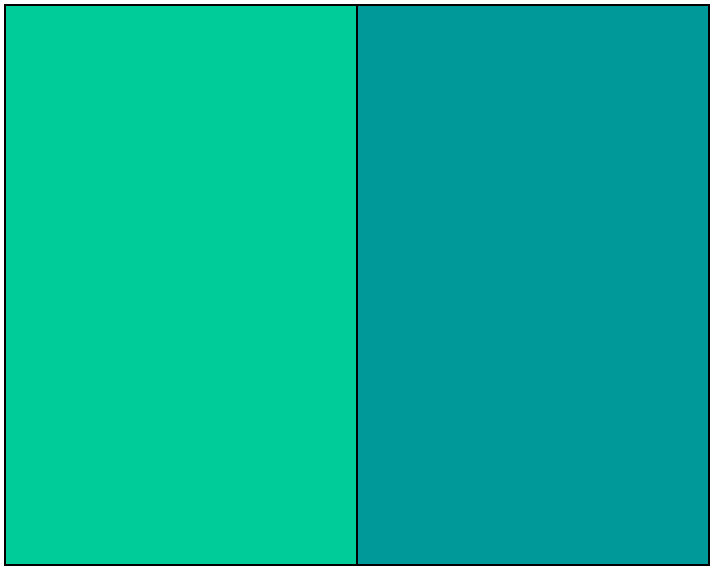


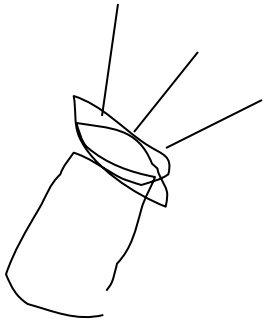


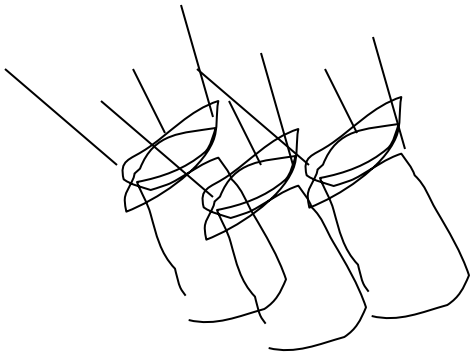


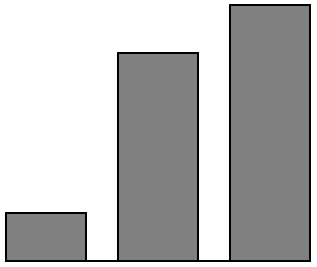
Source: W. Freeman

Color matching experiment 1



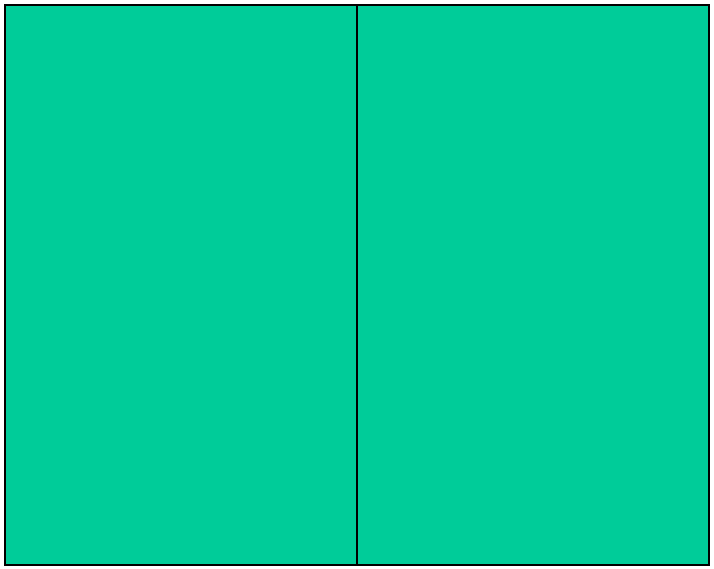




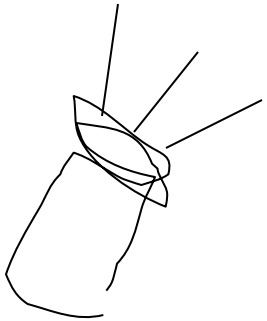


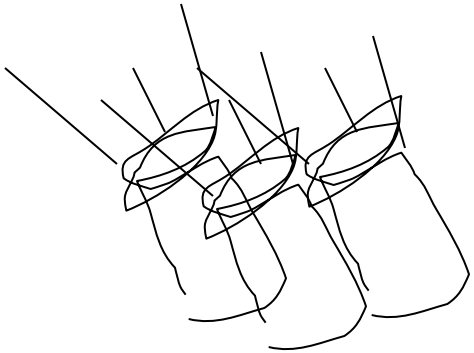
Source: W. Freeman

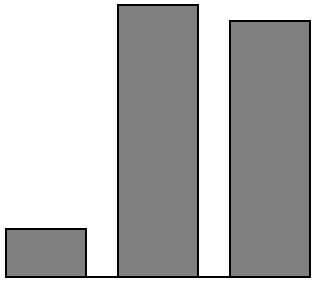
Color matching experiment 1



The primary color amounts needed for a match

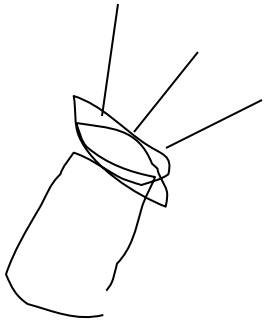






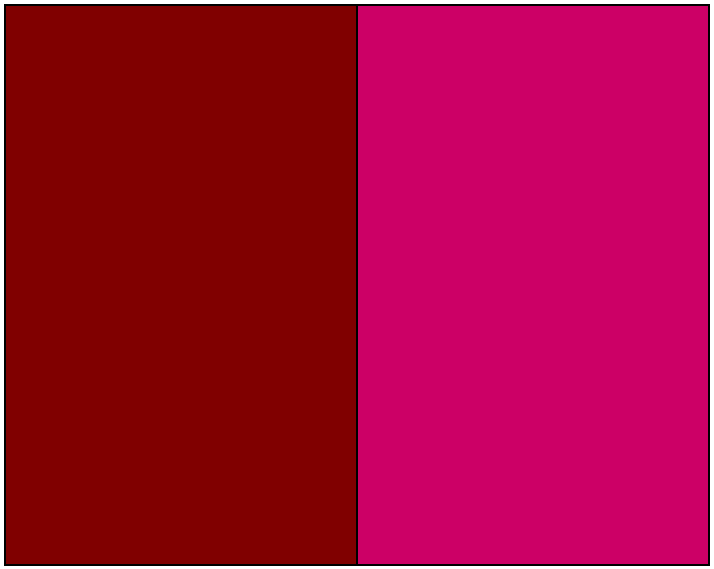
Source: W. Freeman

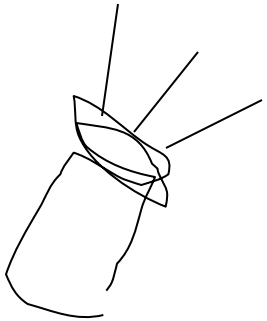
Color matching experiment 2

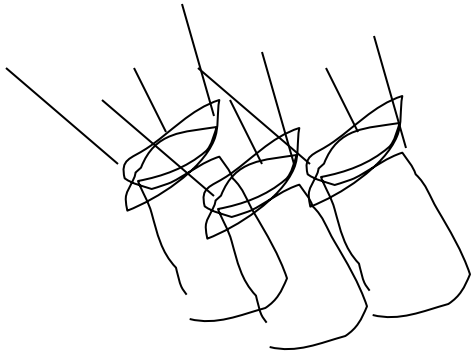


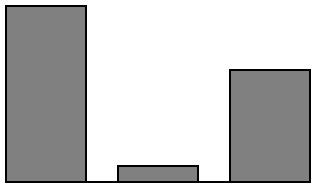
Source: W. Freeman

Color matching experiment 2



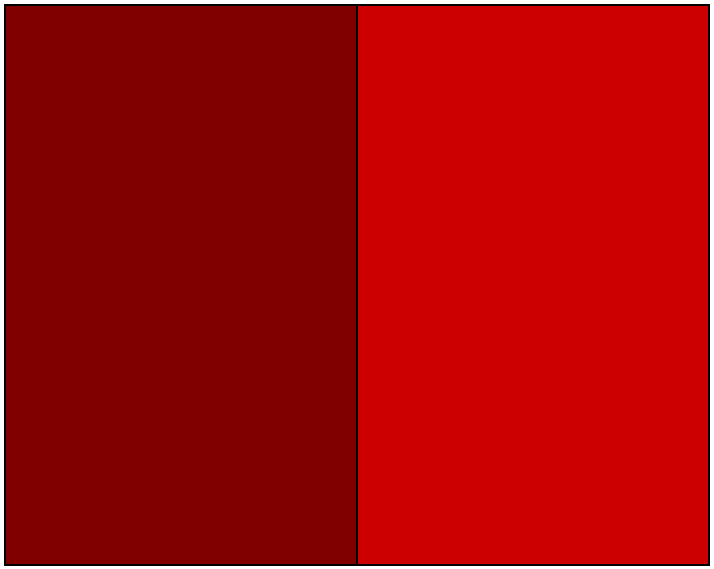


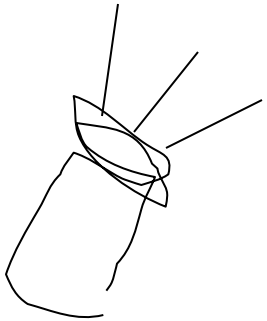


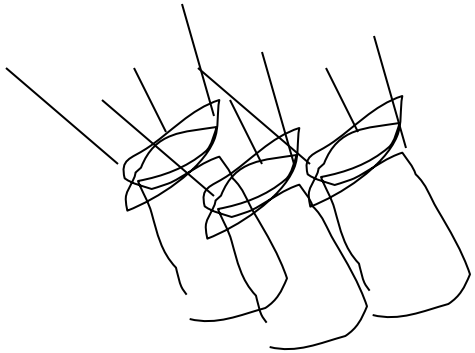


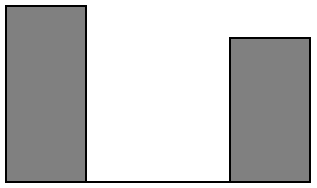
Source: W. Freeman

Color matching experiment 2





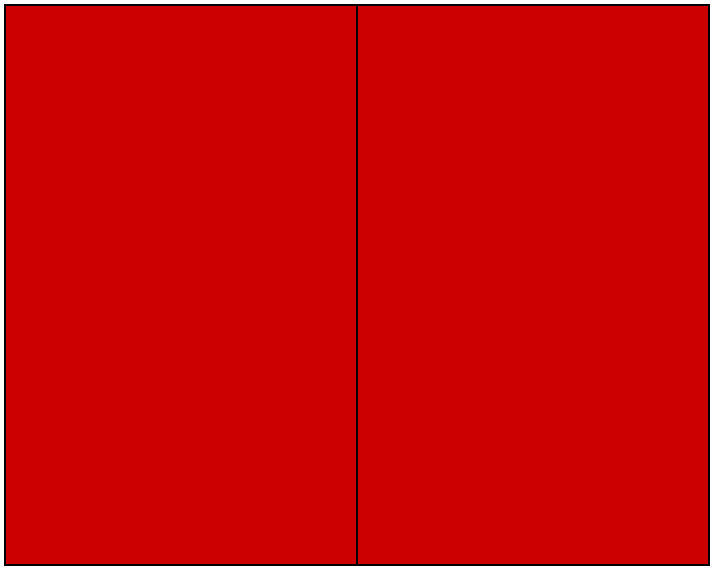




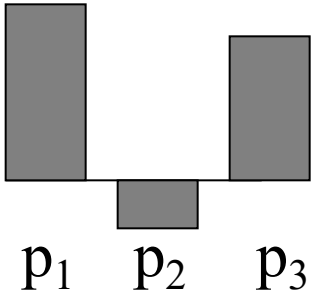
Source: W. Freeman

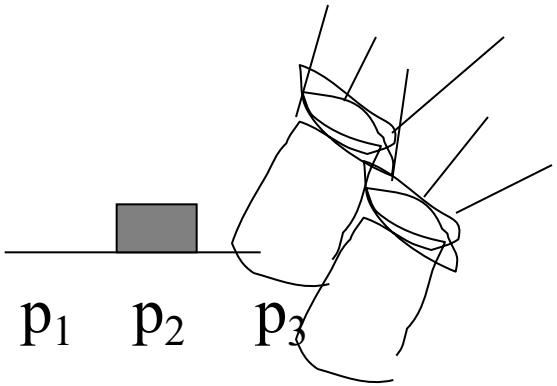
Color matching experiment 2

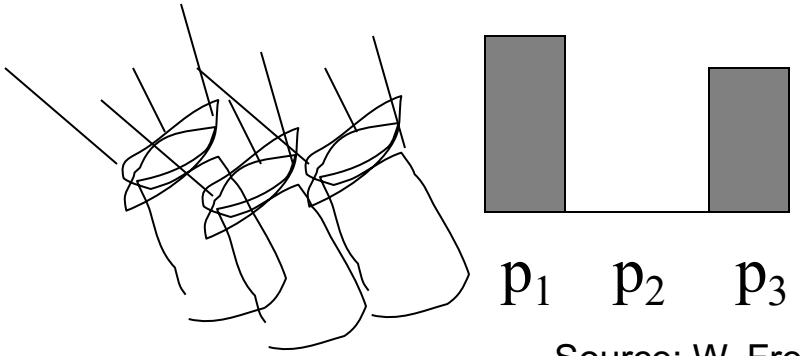
We say a "negative" amount of p_2 was needed to make the match, because we added it to the test color's side.



The primary color amounts needed for a match:







Source: W. Freeman

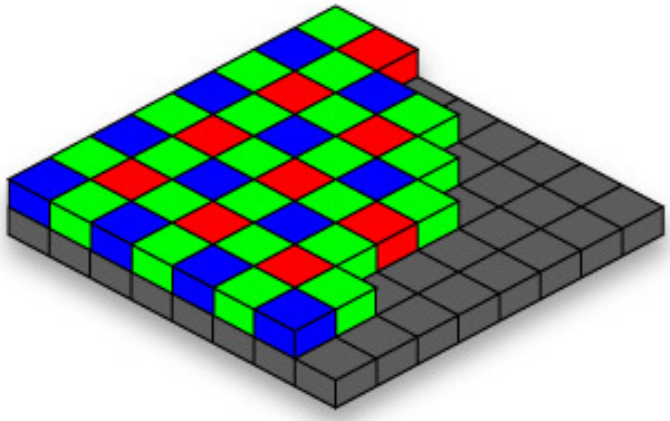
Trichromacy

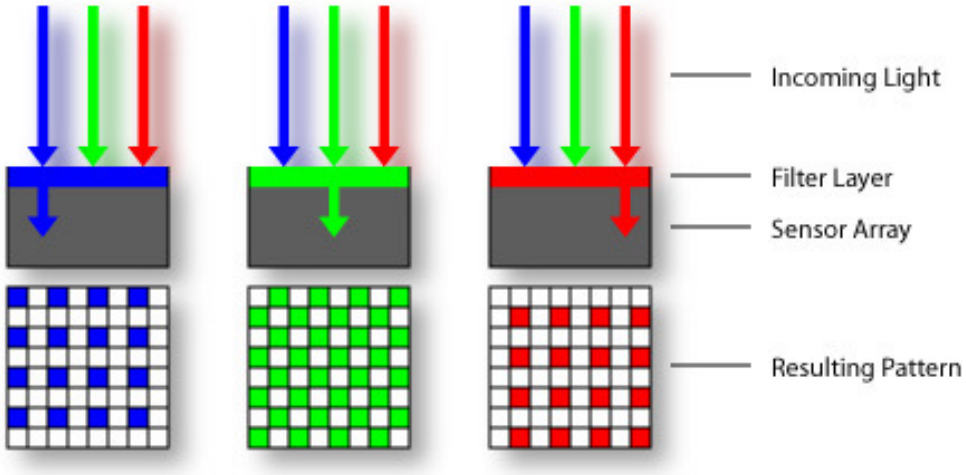
- In color matching experiments, most people can match any given light with three primaries
  - Primaries must be *independent*
- For the same light and same primaries, most people select the same weights
  - Exception: color blindness
- Trichromatic color theory
  - Three numbers seem to be sufficient for encoding color
  - Dates back to 18th century (Thomas Young)

https://en.wikipedia.org/wiki/Young_Helmholtz_theory

Slide by L. Lazebnik

Artificial Cones

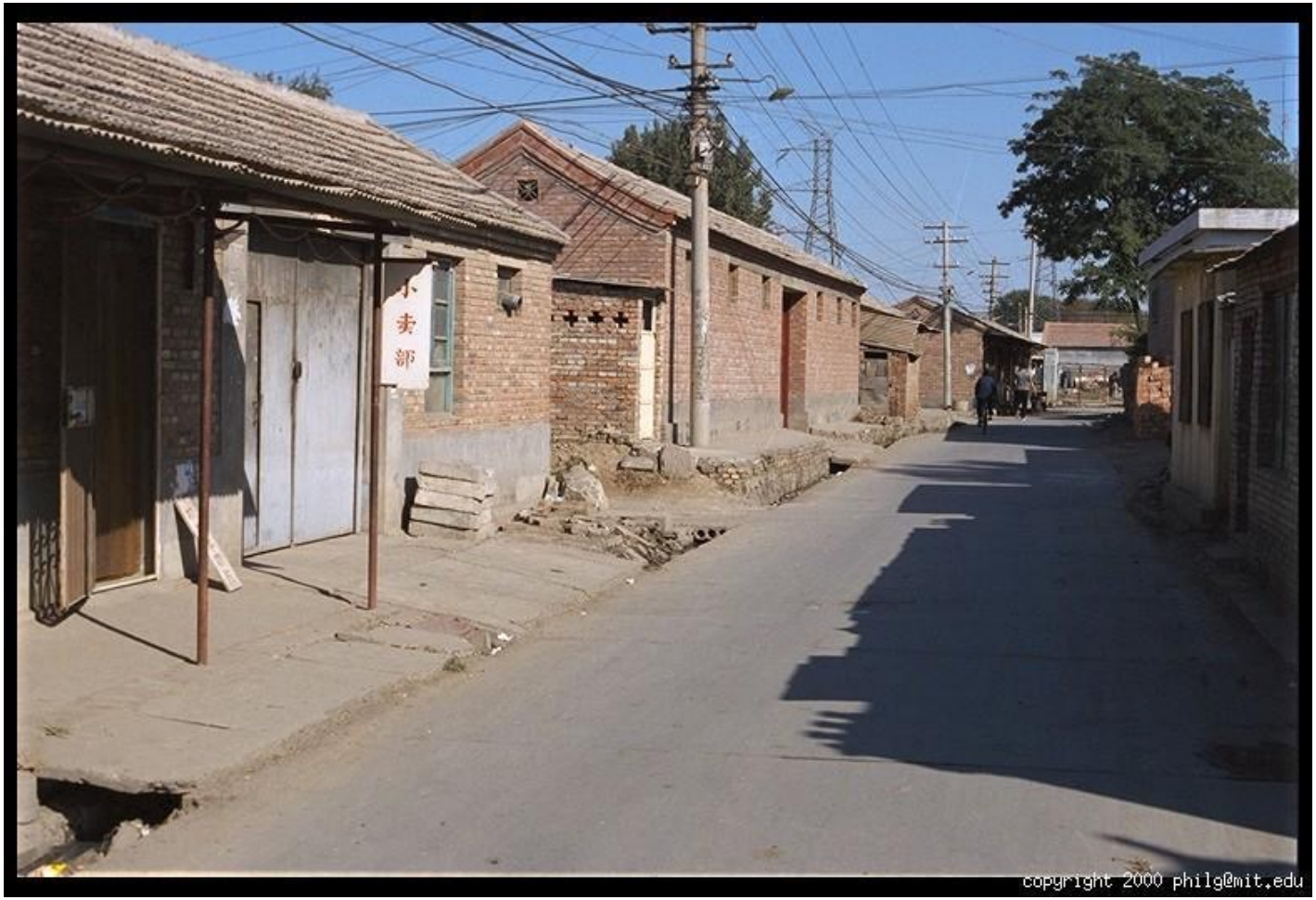




Estimate RGB at ‘G’ cells from neighboring values.

Slide by Steve Seitz

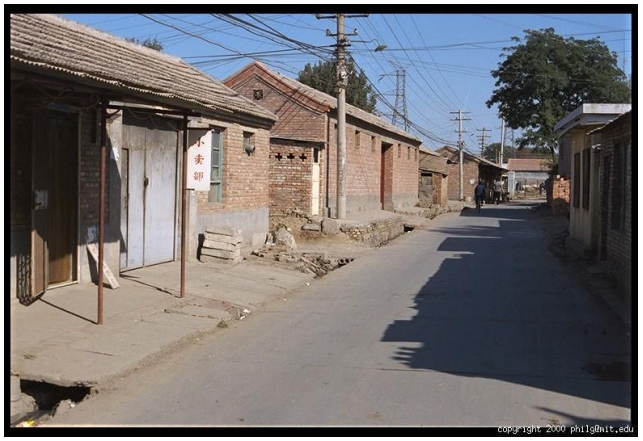
Color Image



Slide Credit: J. Hays

Color Image

Combined



Red

Green

Blue

Slide Credit: J. Hays

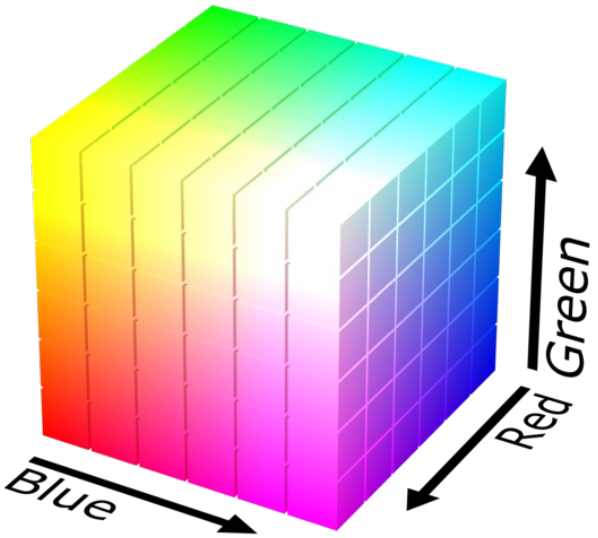
One Option: RGB

Pros

1. Simple
2. Common

Cons

1. Distances don't make sense
2. Correlated



Slide Credit: J. Hays, RGB cube: https://en.wikipedia.org/wiki/RGB_color_model

RGB

Photo credit: J. Hays

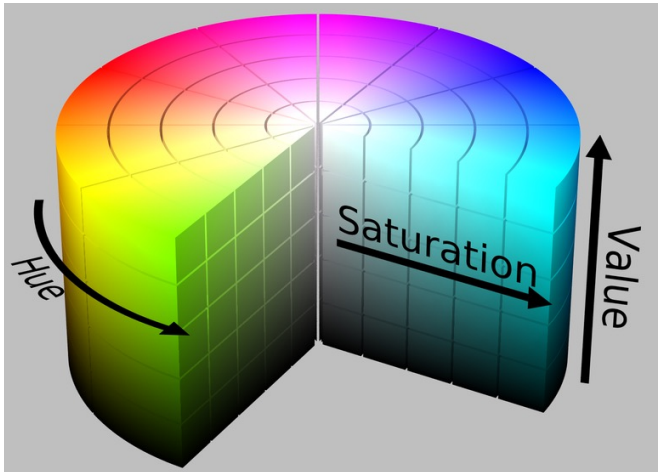
Another Option: HSV

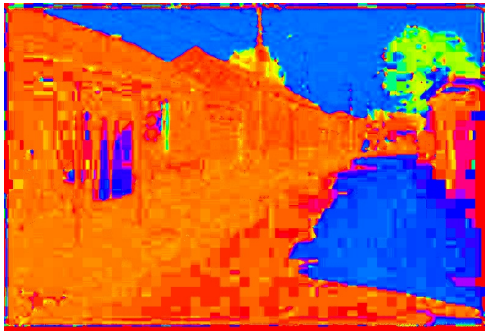
Pros

1. Intuitive for picking colors
2. Sort of common
3. Fast to convert

Cons

1. Not as perceptual





H
(S=1,V=1)

S
(H=1,V=1)

V
(H=1,S=0)

Slide Credit: J. Hays, HSV cylinder: https://en.wikipedia.org/wiki/HSL_and_HSV

HSV

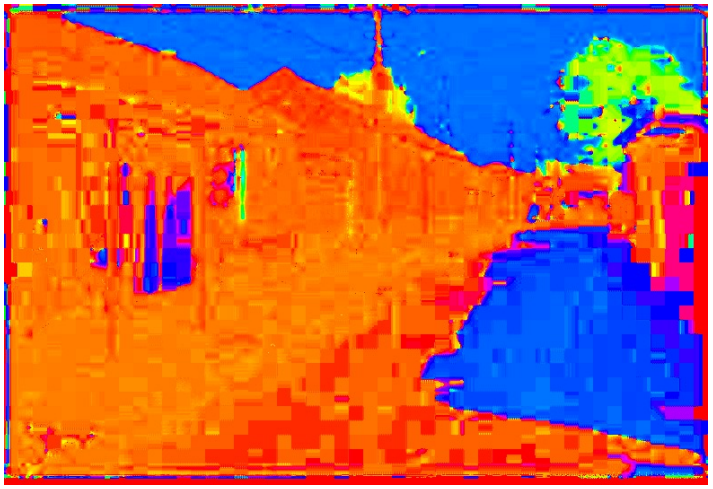


Photo credit: J. Hays

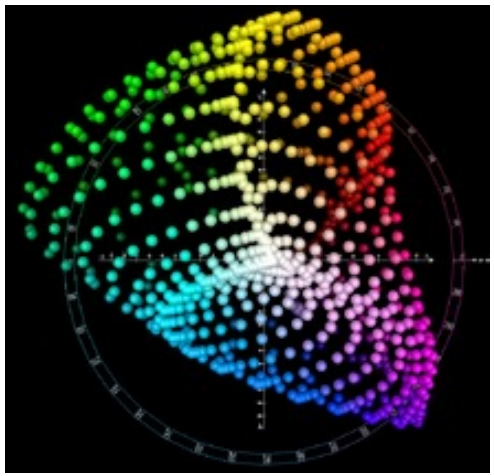
Another Option: Lab

Pros

1. Distances correspond with human judgment

Cons

1. Complex to calculate



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Slide Credit: J. Hays, Lab diagram cube: https://en.wikipedia.org/wiki/CIELAB_color_space

Lab

Photo credit: J. Hays

Why Are There So Many?

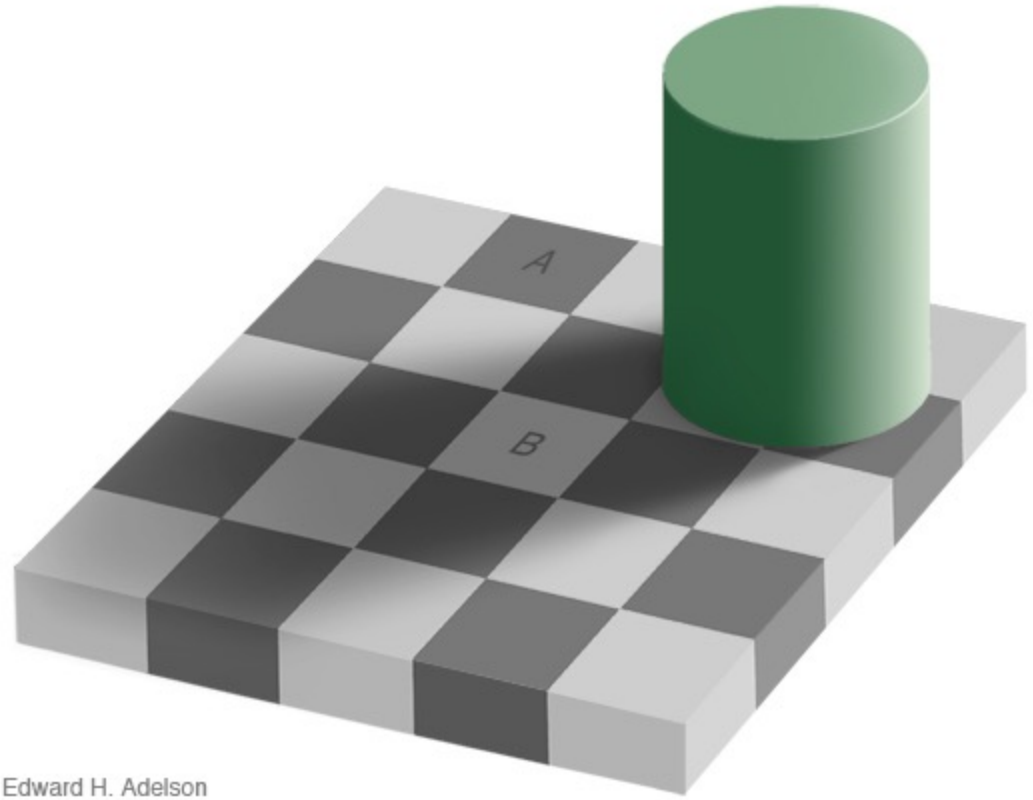
- Each serves different functions
  - RGB: sort of intuitive, standard, everywhere
  - HSV: good for picking, fast to compute
  - YCbCr/YUV: fast to compute, compresses well
  - Lab: the right(?) thing to do, but “slow” to compute
- Pick based on what you need and don’t sweat it: color really isn’t crucial

Color perception

- Color/lightness constancy
  - The ability of the human visual system to perceive the intrinsic reflectance properties of the surfaces despite changes in illumination conditions

Slide by L. Lazebnik

Checker shadow illusion

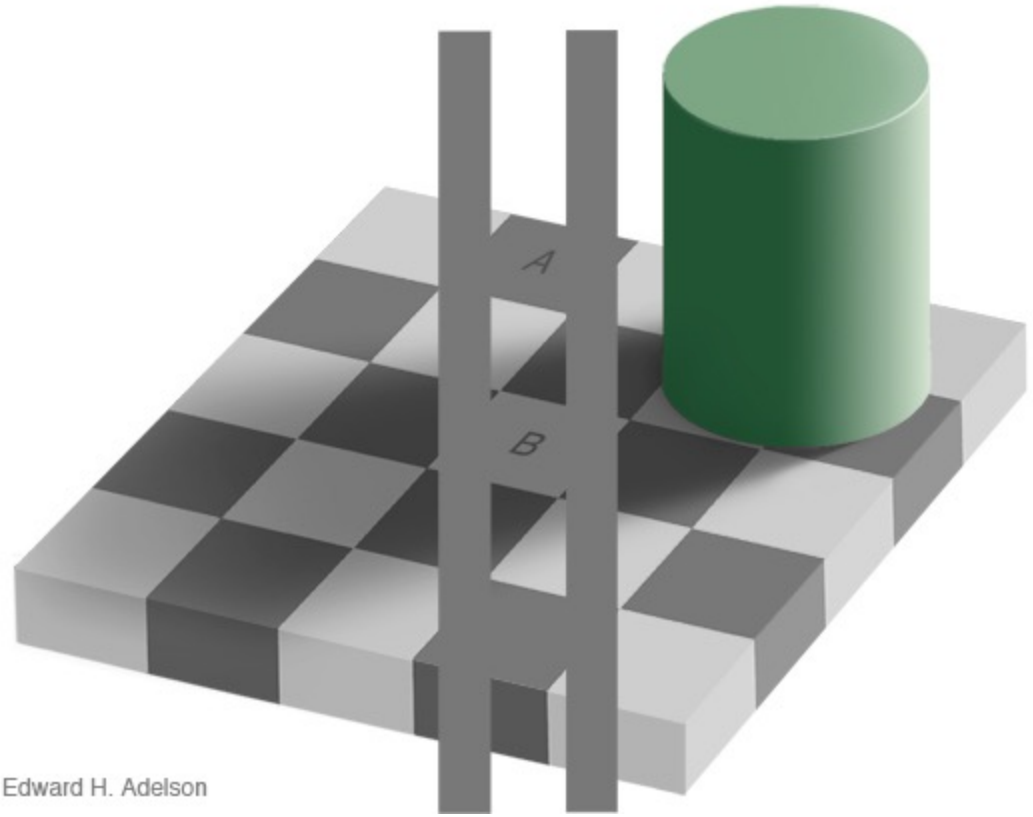


Edward H. Adelson

https://en.wikipedia.org/wiki/Checker_shadow_illusion

Slide by L. Lazebnik

Checker shadow illusion



- Possible explanations
  - Simultaneous contrast
  - Reflectance edges vs. illumination edges

https://en.wikipedia.org/wiki/Checker_shadow_illusion

Slide by L. Lazebnik

What color is the dress?

https://www.wired.com/2015/02/science-one-agrees-color-dress/

Slide by L. Lazebnik

This strawberry cake has no red pixels!

https://www.digitaltrends.com/photography/non-red-strawberries/

Slide by L. Lazebnik

White balance

- Analogous to color constancy mechanisms in human vision, cameras have mechanisms to adapt to the illumination in the environment so that neutral (white or gray) objects look neutral

Incorrect white balance

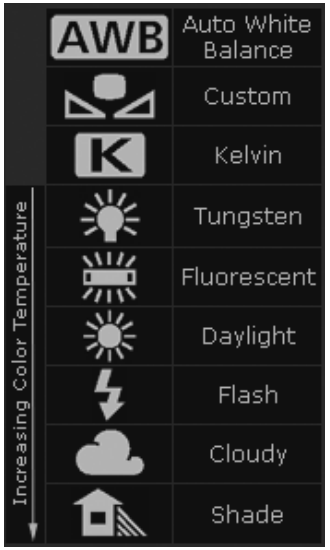
Correct white balance

http://www.cambridgeincolour.com/tutorials/white-balance.htm

Slide by L. Lazebnik

White balance

- Film cameras:
  - Different types of film or different filters for different illumination conditions
- Digital cameras:
  - Automatic white balance
  - White balance settings corresponding to several common illuminants
  - Custom white balance using a reference object



http://www.cambridgeincolour.com/tutorials/white-balance.htm

Slide by L. Lazebnik

White balance

- Von Kries adaptation: Multiply each channel by a *gain factor*

- Best way: gray card
  - Take a picture of a neutral object (white or gray)
  - If the object is recorded as r_w, g_w, b_w use weights $1/r_w$, $1/g_w$, $1/b_w$

Slide by L. Lazebnik

White balance

- Without gray cards: we need to “guess” which pixels correspond to white objects
- Gray world assumption
  - The image average r_{ave}, g_{ave}, b_{ave} is gray
  - Use weights $1/r_{ave}$, $1/g_{ave}$, $1/b_{ave}$
- Brightest pixel assumption
  - Highlights usually have the color of the light source
  - Use weights inversely proportional to the values of the brightest pixels
- Gamut mapping
  - Gamut: convex hull of all pixel colors in an image
  - Find the transformation that matches the gamut of the image to the gamut of a “typical” image under white light
- Use image statistics, learning techniques

Slide by L. Lazebnik

Mixed illumination

- When there are several types of illuminants in the scene, different reference points will yield different results

Reference: moon

Reference: stone

http://www.cambridgeincolour.com/tutorials/white-balance.htm

Slide by L. Lazebnik

Outline

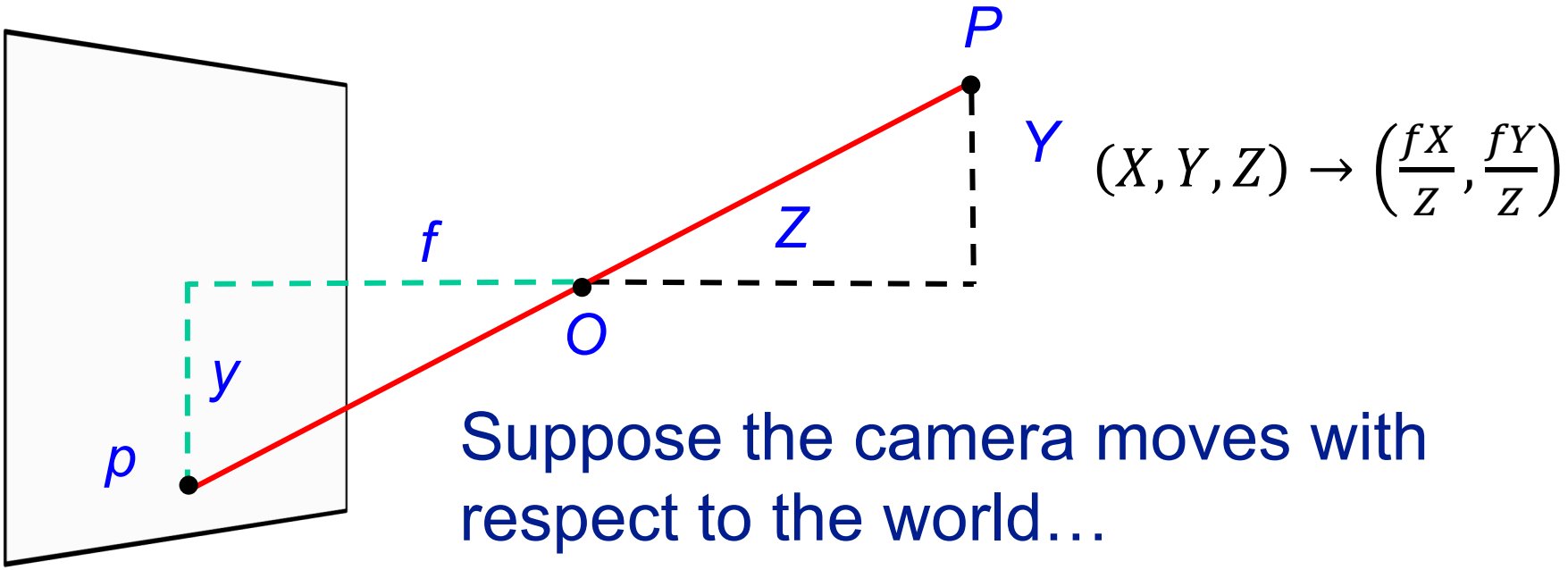
- Physical origin of color
- Spectra of sources and surfaces
- Physiology of color vision
- Trichromatic color theory
- Color spaces
- Color constancy, white balance

Dynamic Perspective

CS 543 / ECE 549 – Saurabh Gupta

Many slides adapted from J. Malik.

Perspective Projection



- Point P (X, Y, Z) in the world moves relative to the camera, its projection in the image (x, y) moves as well.
- This movement in the image plane is called optical flow.
- Suppose the image of the point (x, y) moves to $(x + \Delta x, y + \Delta y)$ in time Δt, then $\left(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}\right)$ are the two components of the optical flow.

Outline

- Relate optical flow to camera motion
- Special cases

How does a point X in the scene move?

- Assume that the camera moves with a translational velocity $t = (t_x, t_y, t_z)$ and angular velocity $\omega = \left(\omega_x, \omega_y, \omega_z\right)$.
- Linear velocity of point $P = (X, Y, Z)$ is given by $\dot{P} = -t\ - \omega \times P$.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} \omega_y Z - \omega_z Y \\ \omega_z X - \omega_x Z \\ \omega_x Y - \omega_y X \end{bmatrix}$$

Now, lets consider the effect of projection

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix} = -\begin{bmatrix}t_x\\ t_y\\ t_z\end{bmatrix} - \begin{bmatrix}\omega_y Z - \omega_z Y\\ \omega_z X - \omega_x Z\\ \omega_x Y - \omega_y X\end{bmatrix}$$

- Assume, $f = 1$, $x = \frac{X}{Z}, y = \frac{Y}{Z}$.
- $\dot{x} = \frac{\dot{X}Z - \dot{Z}X}{Z^2}$, $\dot{y} = \frac{\dot{Y}Z - \dot{Z}Y}{Z^2}$
- Substitute $\dot{X}, \dot{Y}, \dot{Z}$, from equation above:

$$\begin{bmatrix}u\\ v\end{bmatrix} = \begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \frac{1}{Z}\begin{bmatrix}-1 & 0 & x\\ 0 & -1 & y\end{bmatrix}\begin{bmatrix}t_x\\ t_y\\ t_z\end{bmatrix} + \begin{bmatrix}xy & -(1+x^2) & y\\ (1+y^2) & -xy & -x\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}$$

Dynamic Perspective Equations

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Translation Component

Rotation Component

Optical flow for pure rotation

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

- $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$

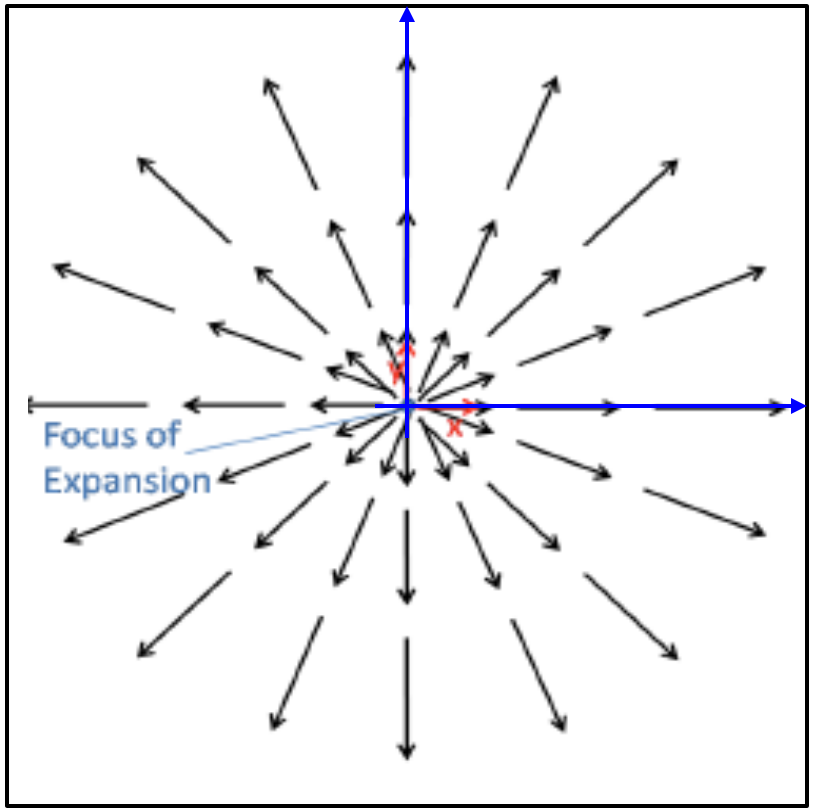
- We can determine ω from the flow field.

- Flow field is independent of $Z(x, y)$.

Optical flow for pure translation along Z-axis

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

- $\begin{bmatrix} u \\ v \end{bmatrix} = \frac{t_z}{Z(x,y)}\begin{bmatrix} x \\ y \end{bmatrix}$
- Optical flow vector is a scalar multiple of position vector.
- Scale factor ambiguity, if $t_z \rightarrow kt_z$, and $Z \rightarrow kZ$, optical flow remains unchanged.
- But, you can get time to collision, Z/t_z.




STOP


STOP

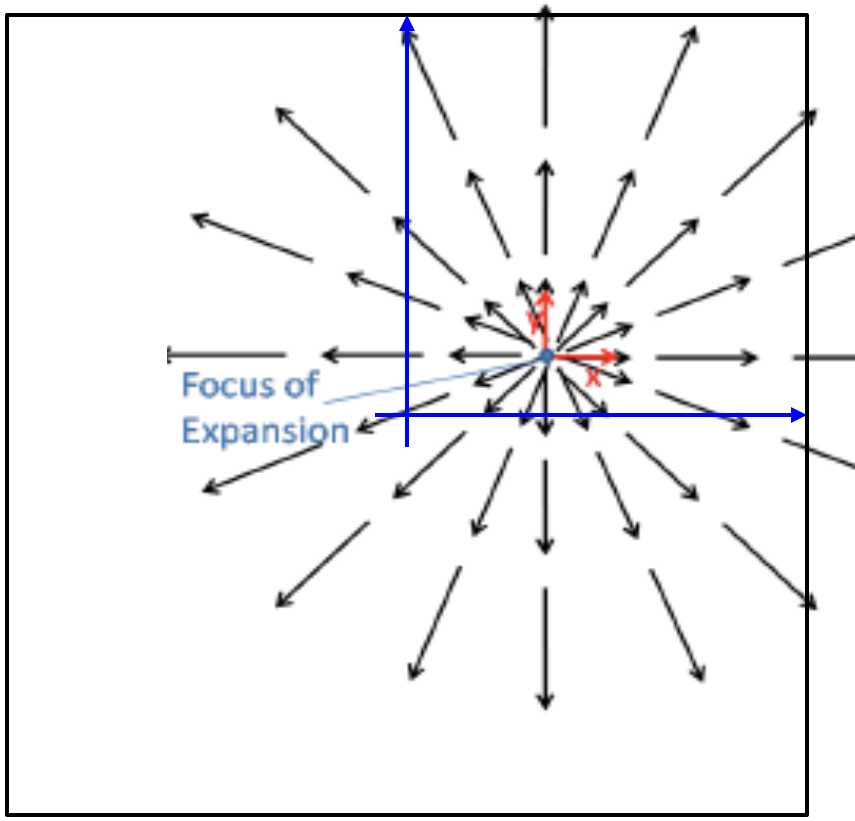

STOP


STOP

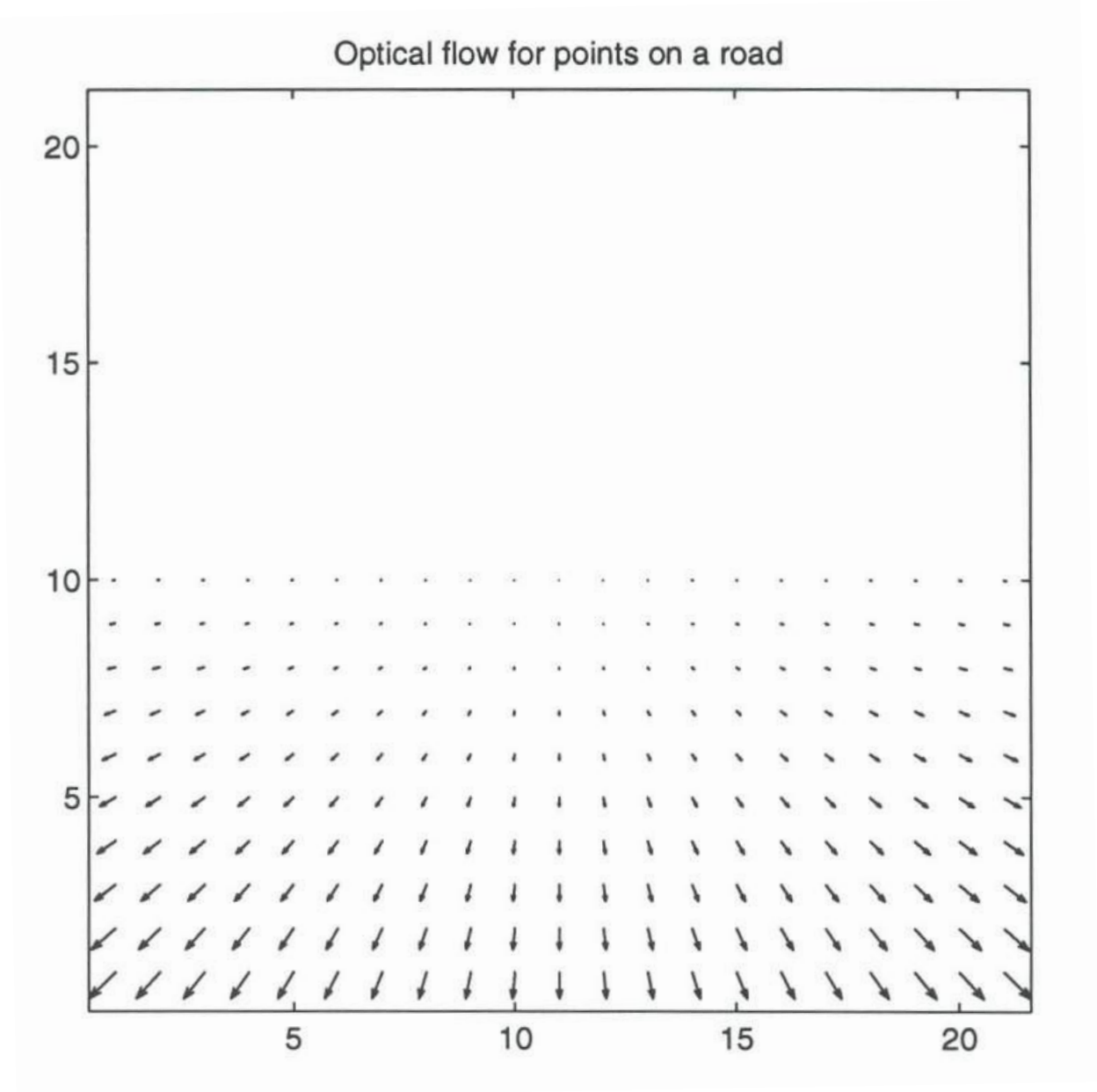
Optical flow for impure translation along Z-axis

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

- $u = \frac{-t_x + x t_z}{Z(x,y)}$, $\mathrm{v} = \frac{-t_y + y t_z}{Z(x,y)}$

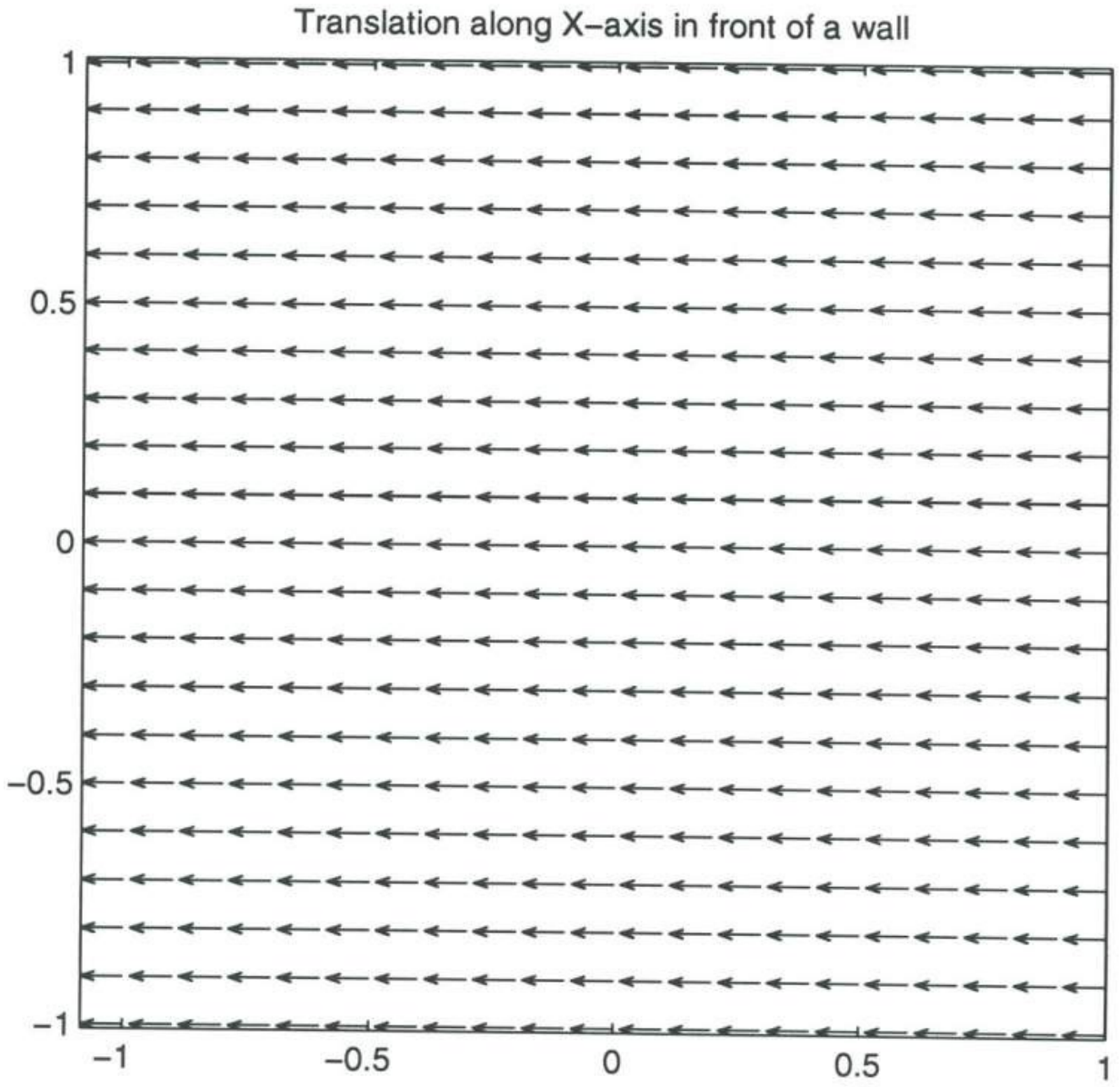


Optical flow for points on a road



Slide by J. Malik

Translating along X-axis in front of a wall



Slide by J. Malik

Estimating Optical Flow from Images

Aperture Problem

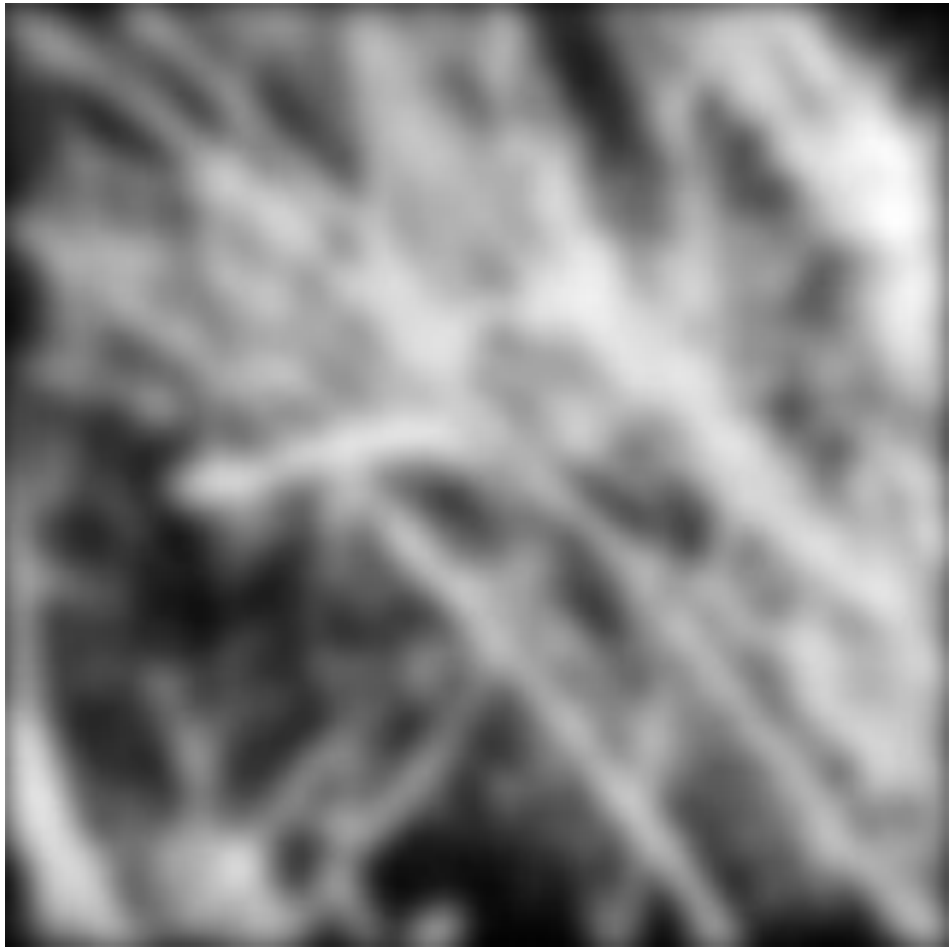
Recap

- Relate optical flow to camera motion

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

- Special cases
  - Pure rotation / pure translation / time to collision

Linear filtering



Slides from Lana Lazebnik

Motivation: Image denoising

- How can we reduce noise in a photograph?

Slide from Lana Lazebnik

Moving average

- Let's replace each pixel with a *weighted average* of its neighborhood
- The weights are called the *filter kernel*
- What are the weights for the average of a 3x3 neighborhood?

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

"box filter"

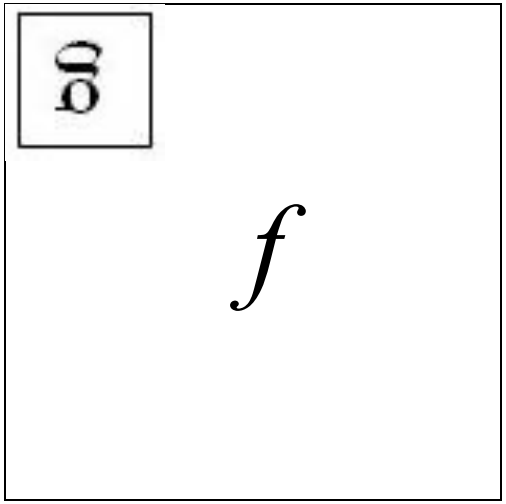
Source: D. Lowe

Defining convolution

- Let *f* be the image and *g* be the kernel. The output of convolving *f* with *g* is denoted *f* * *g*.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$

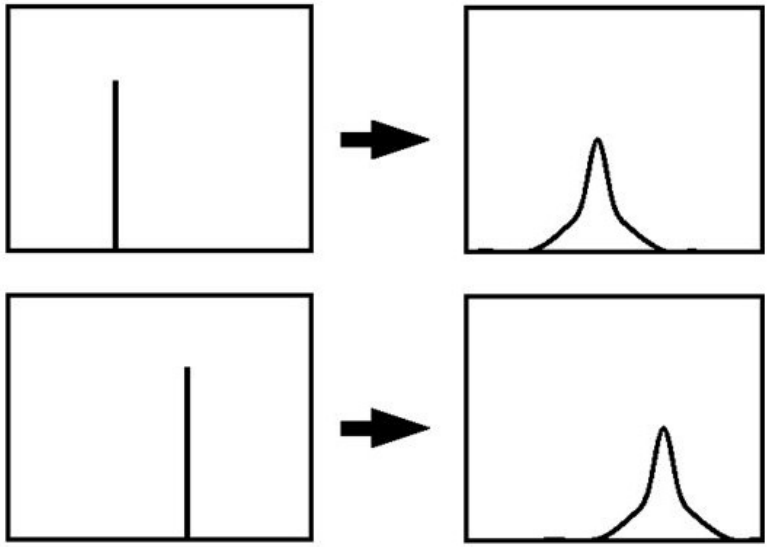
Convention:
kernel is “flipped”



Source: F. Durand

Key properties

- **Shift invariance:** same behavior regardless of pixel location:
  filter(shift(f)) = shift(filter(f))



- **Linearity:**
  filter(f_1 + f_2) =
  filter(f_1) + filter(f_2)

- Theoretical result: any linear shift-invariant operator can be represented as a convolution

Slide from Lana Lazebnik

Properties in more detail

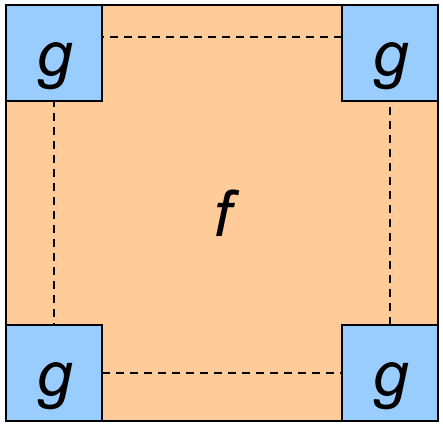
- Commutative: $a * b = b * a$
  - Conceptually no difference between filter and signal
- Associative: $a * (b * c) = (a * b) * c$
  - Often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - This is equivalent to applying one filter: $a * (b_1 * b_2 * b_3)$
- Distributes over addition: $a * (b + c) = (a * b) + (a * c)$
- Scalars factor out: $ka * b = a * kb = k (a * b)$
- Identity: unit impulse $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$, $a * e = a$

Slide from Lana Lazebnik

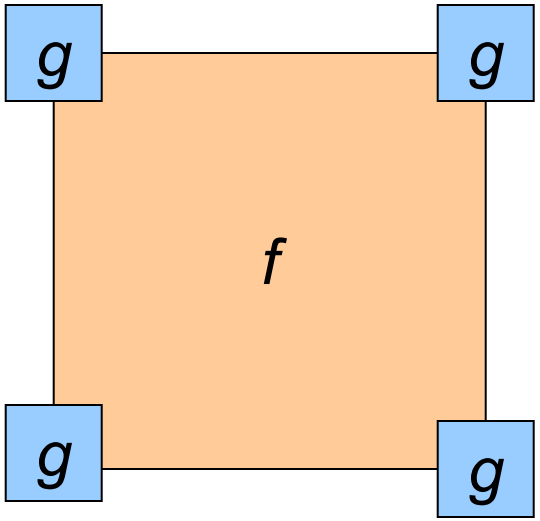
Dealing with edges

- If we convolve image *f* with filter *g*, what is the size of the output?

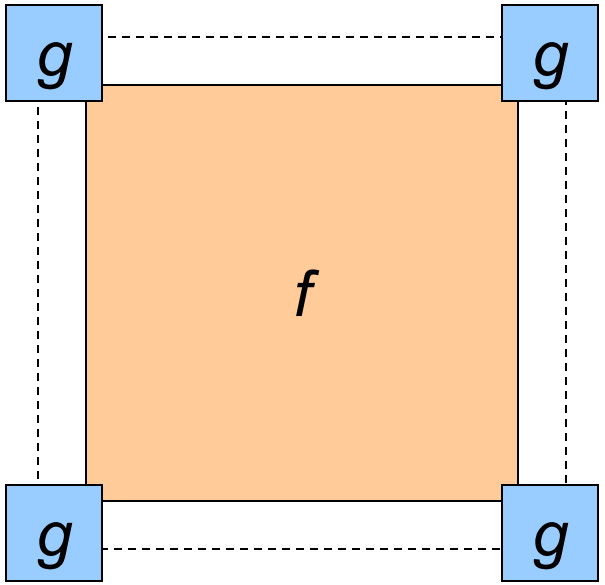
Output is smaller than input



Output is same size as input



Output is larger than input



Slide from Lana Lazebnik

Dealing with edges

- If the filter window falls off the edge of the image, we need to pad the image
  - Zero pad (or clip filter)
  - Wrap around
  - Copy edge
  - Reflect across edge

Source: S. Marschner

Practice with linear filters

Original

0	0	0
0	1	0
0	0	0

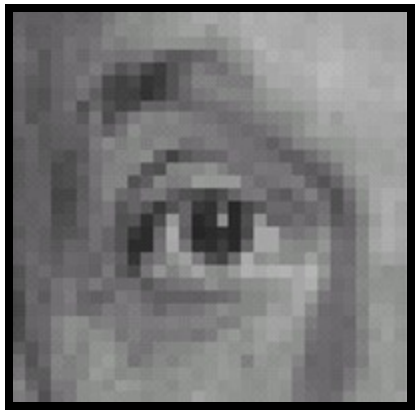
?

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	0	1
0	0	0

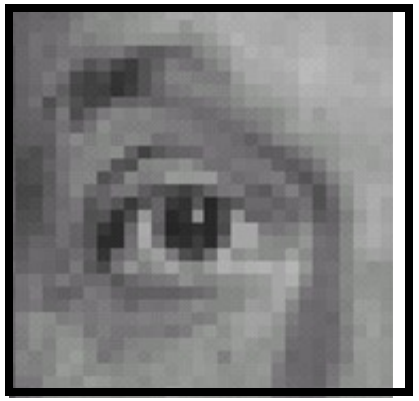
?

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	0	1
0	0	0



Shifted *left*
By 1 pixel

Source: D. Lowe

Practice with linear filters

Original

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

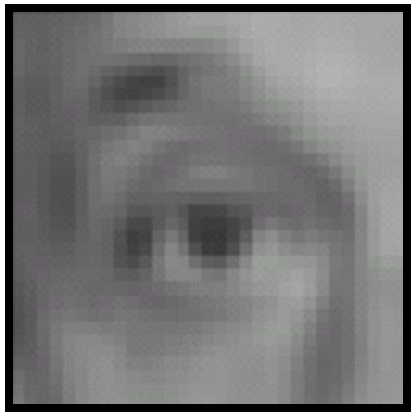
?

Source: D. Lowe

Practice with linear filters

Original

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Blur (with a box filter)

Source: D. Lowe

Practice with linear filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad ?$$

(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters

Original

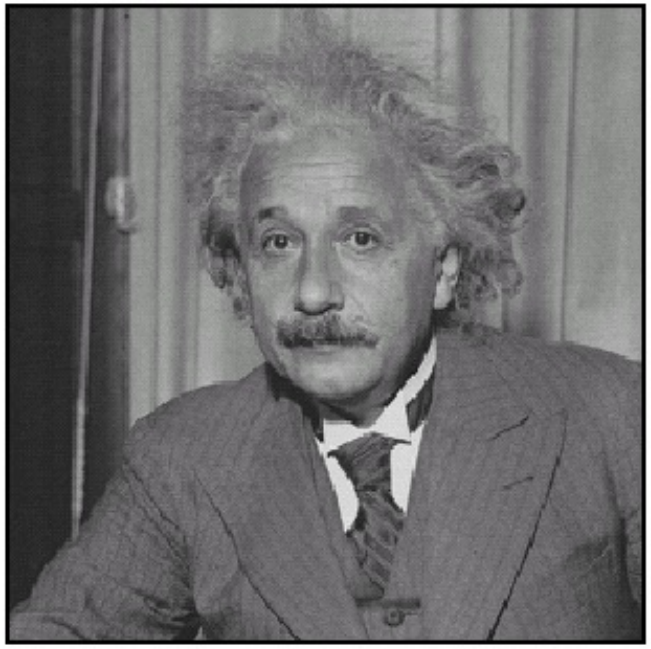
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

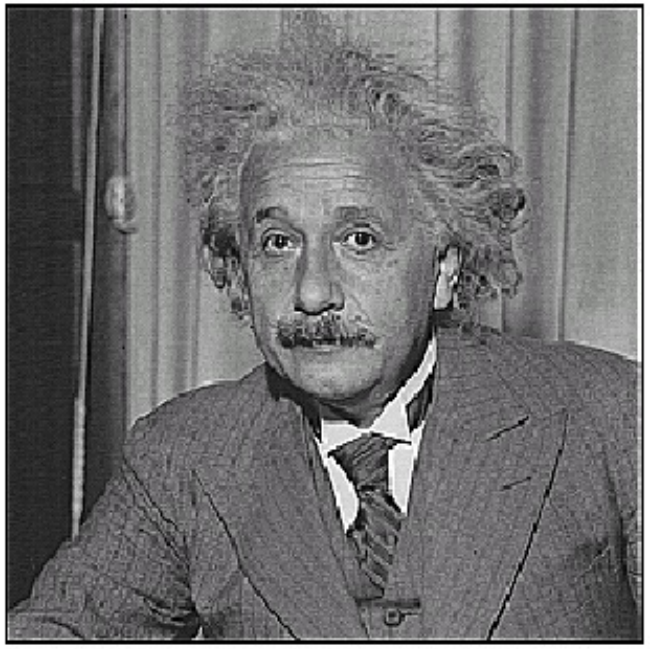
- Accentuates differences with local average

Source: D. Lowe

Sharpening



before



after

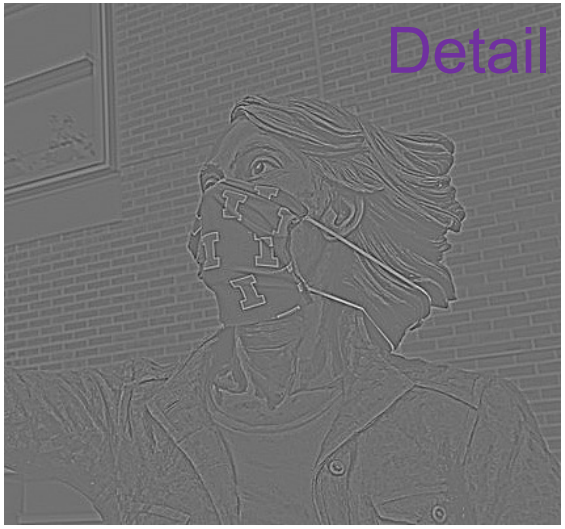
Source: D. Lowe

Sharpening

What does blurring take away?

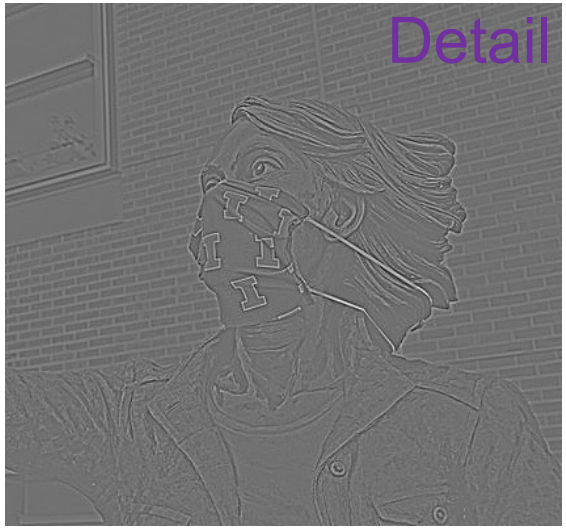
–

=



Let's add it back in.

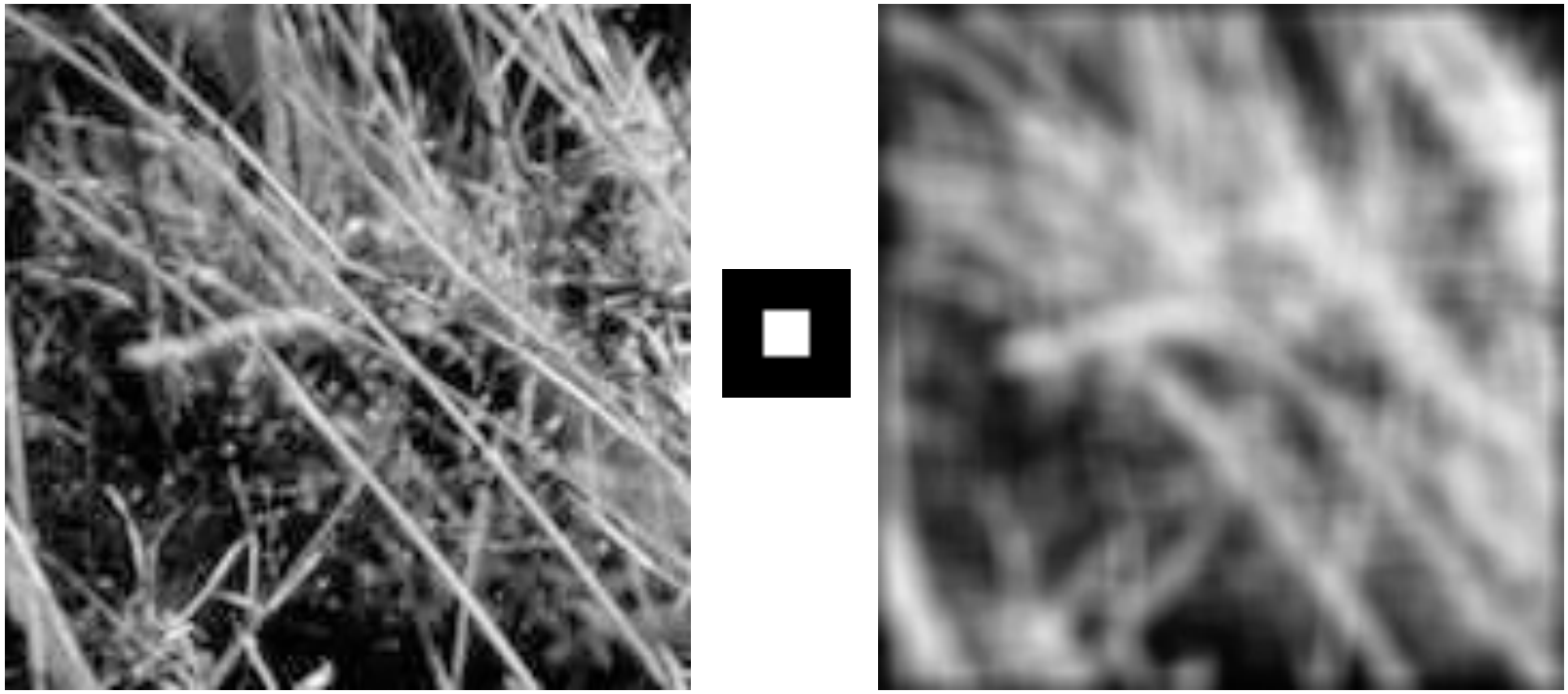
$+\alpha$



=

Smoothing with box filter revisited

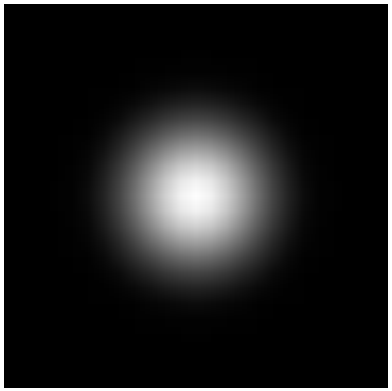
- What's wrong with this picture?
- What's the solution?



Source: D. Forsyth

Smoothing with box filter revisited

- What's wrong with this picture?
- What's the solution?
  - To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center

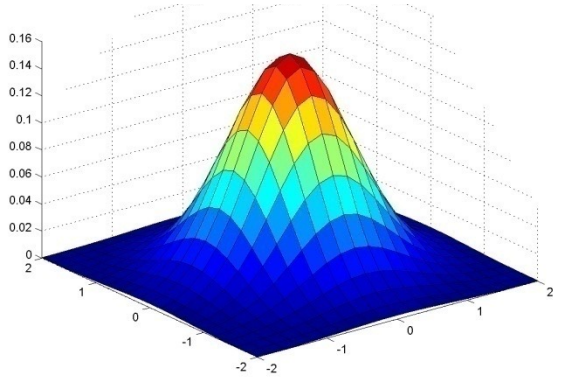


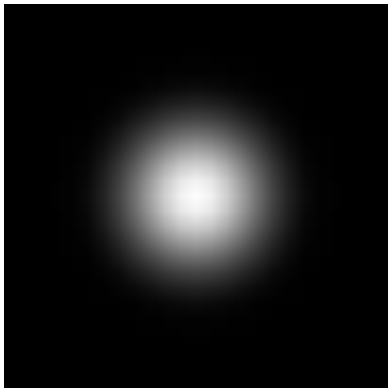
"fuzzy blob"

Slide from Lana Lazebnik

Gaussian Kernel

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





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

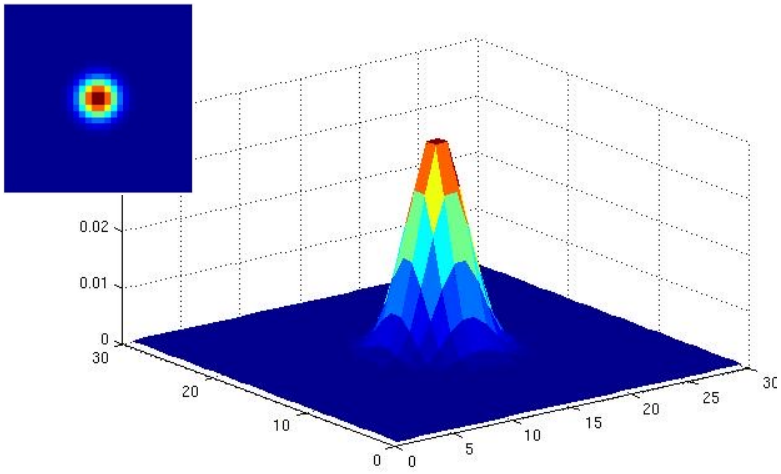
5 x 5, $\sigma = 1$

- Constant factor at front makes volume sum to 1 (can be ignored when computing the filter values, as we should renormalize weights to sum to 1 in any case)

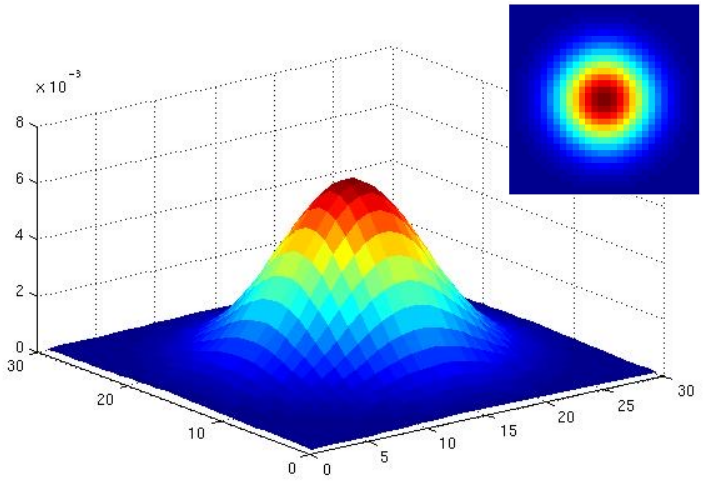
Source: C. Rasmussen

Gaussian Kernel

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



σ = 2 with 30 x 30 kernel



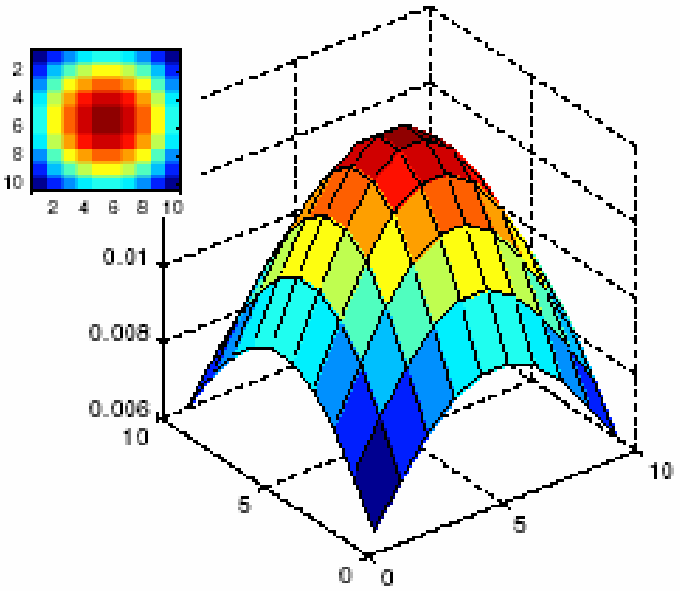
σ = 5 with 30 x 30 kernel

- Standard deviation σ: determines extent of smoothing

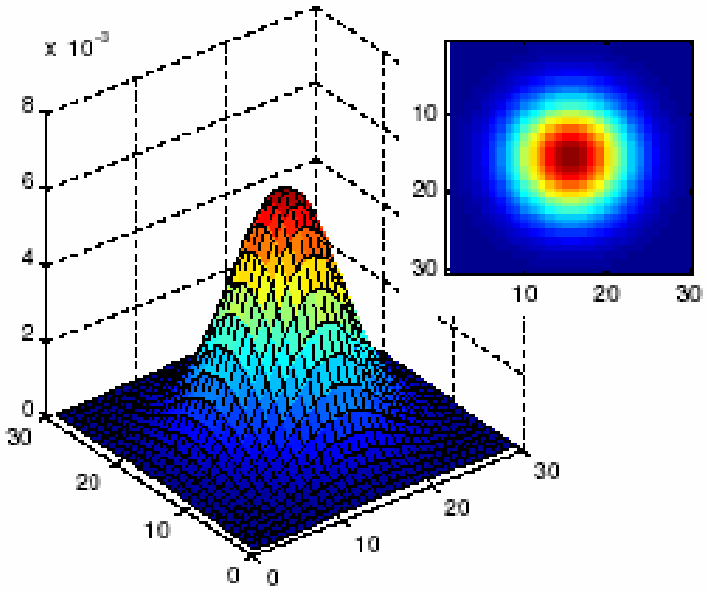
Source: K. Grauman

Choosing kernel width

- The Gaussian function has infinite support, but discrete filters use finite kernels



σ = 5 with 10x10 kernel

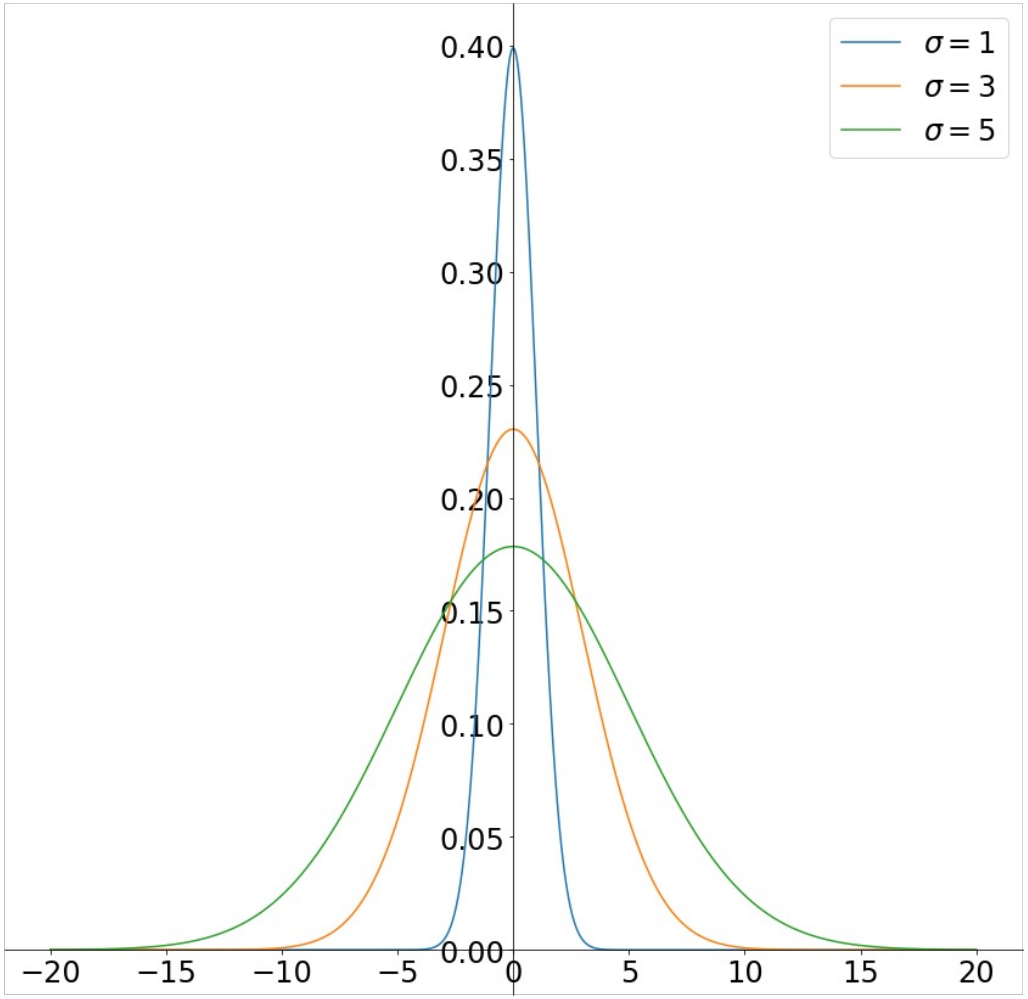


σ = 5 with 30x30 kernel

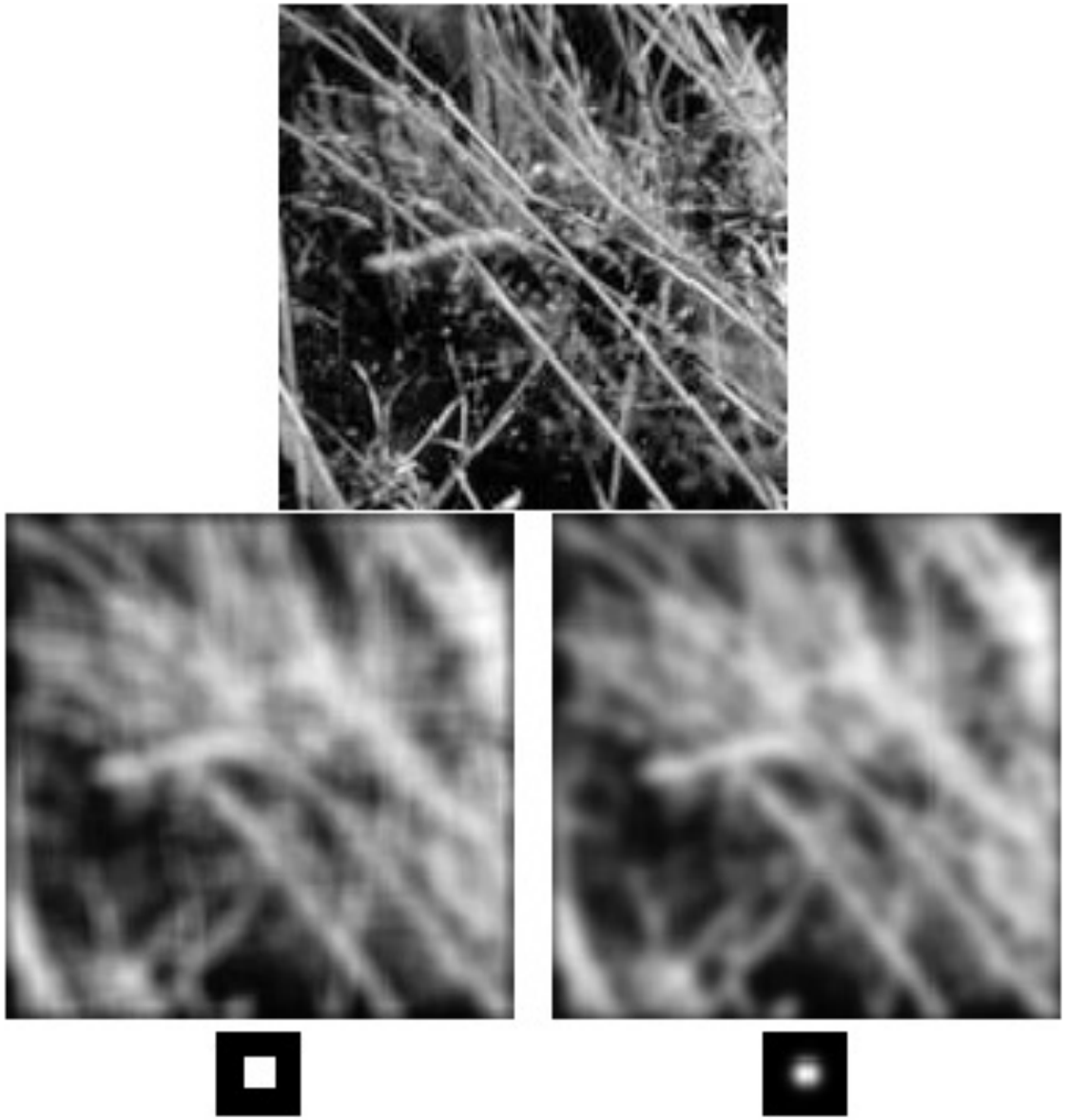
Source: K. Grauman

Choosing kernel width

- Rule of thumb: set filter half-width to about 3σ



Gaussian vs. box filtering



Slide from Lana Lazebnik

Gaussian filters

- Remove high-frequency components from the image (*low-pass filter*)
- Convolution with self is another Gaussian
  - So can smooth with small-σ kernel, repeat, and get same result as larger-σ kernel would have
  - Convolving two times with Gaussian kernel with std. dev. σ is same as convolving once with kernel with std. dev. $\sigma\sqrt{2}$
- *Separable* kernel
  - Factors into product of two 1D Gaussians
  - Discrete example:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Source: K. Grauman

Separability of the Gaussian filter

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

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}\right)$$

The 2D Gaussian can be expressed as the product of two functions, one a function of x and the other a function of y.

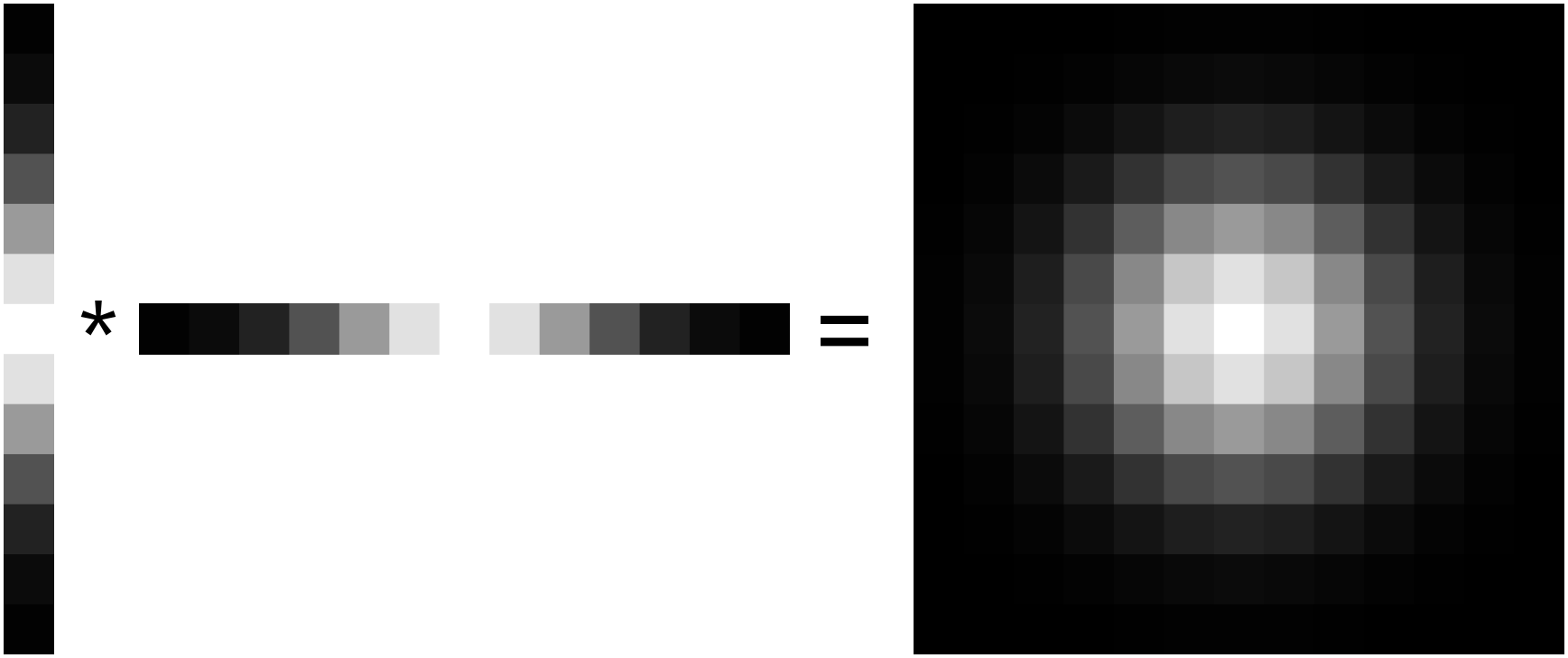
In this case the two functions are the (identical) 1D Gaussian.

Source: D. Lowe

Separability

1D Gaussian $*$ 1D Gaussian = 2D Gaussian

Image $*$ 2D Gauss = Image $*$ (1D Gauss $*$ 1D Gauss)
= (Image $*$ 1D Gauss) $*$ 1D Gauss



Why is separability useful?

- Separability means that a 2D convolution can be reduced to two 1D convolutions (one along rows and one along columns)
- What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
  - $O(n^2 m^2)$
- What if the kernel is separable?
  - $O(n^2 m)$

Slide from Lana Lazebnik

Noise

Original

Salt and pepper noise

Impulse noise

Gaussian noise

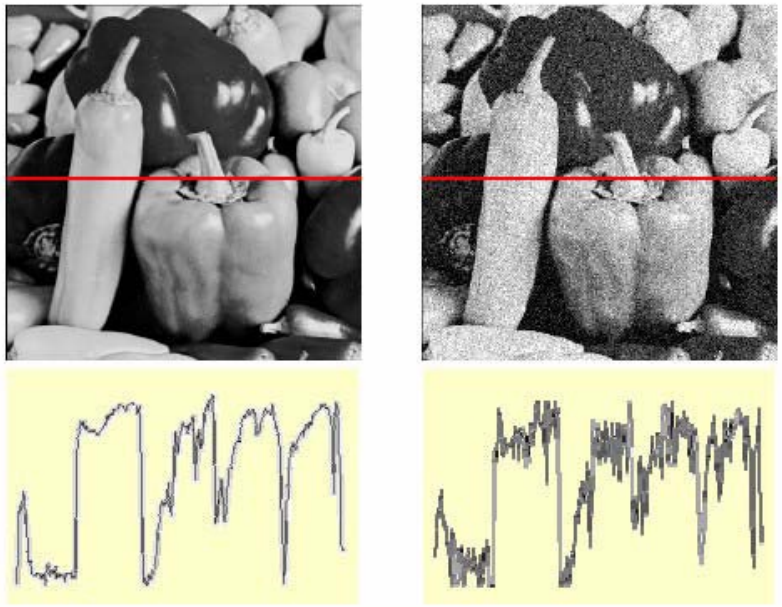
- **Salt and pepper noise**: contains random occurrences of black and white pixels
- **Impulse noise:** contains random occurrences of white pixels
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution

Source: S. Seitz

Gaussian noise

- Mathematical model: sum of many independent factors
- Good for small standard deviations
- Assumption: independent, zero-mean noise

Image
Noise

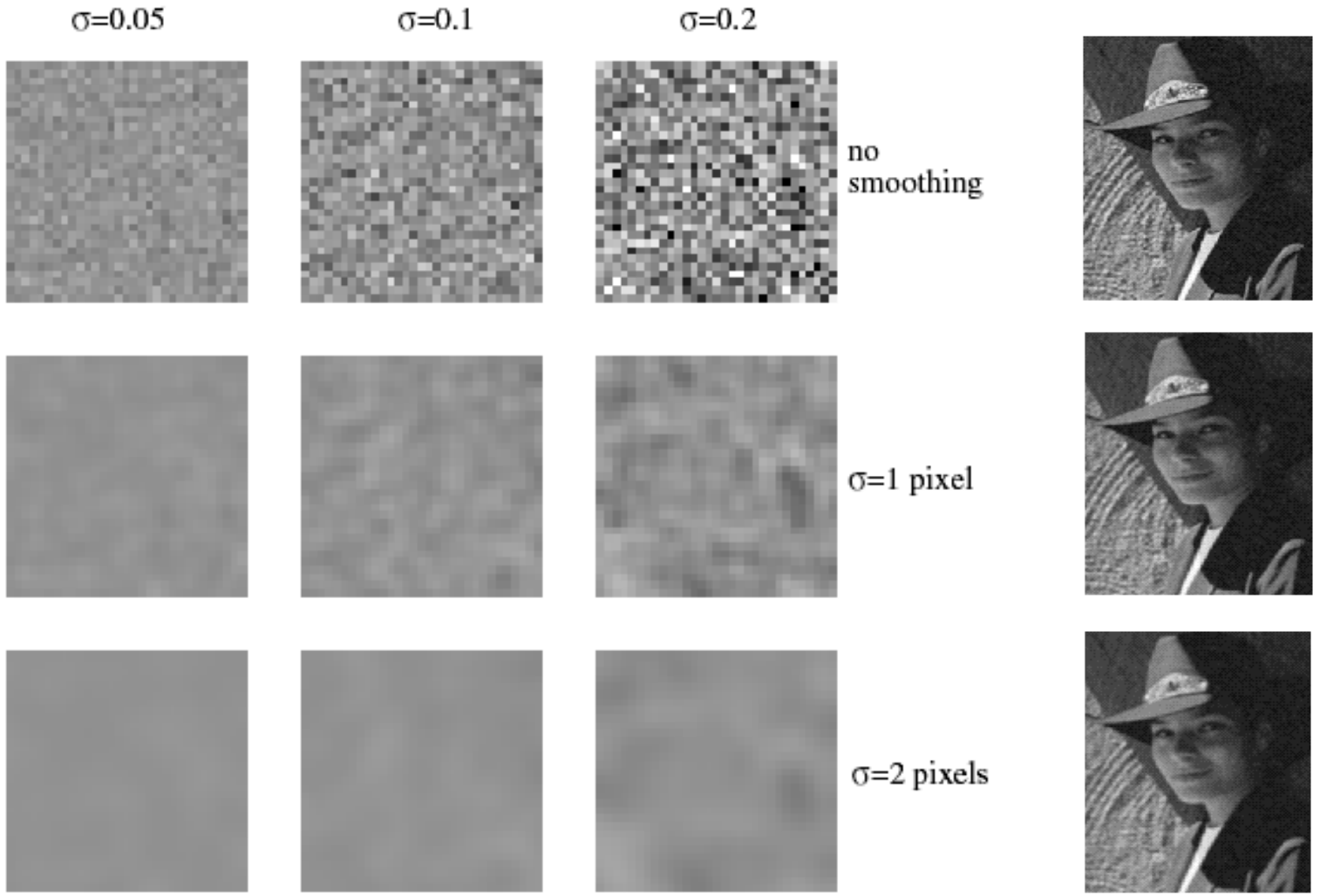


$$f(x,y) = \overbrace{\bar{f}(x,y)}^{\text{Ideal Image}} + \overbrace{\eta(x,y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
$\eta(x,y) \sim \mathcal{N}(\mu,\sigma)$

Source: M. Hebert

Reducing Gaussian noise



Smoothing with larger standard deviations suppresses noise, but also blurs the image

Slide from Lana Lazebnik

Reducing salt-and-pepper noise

3x3

5x5

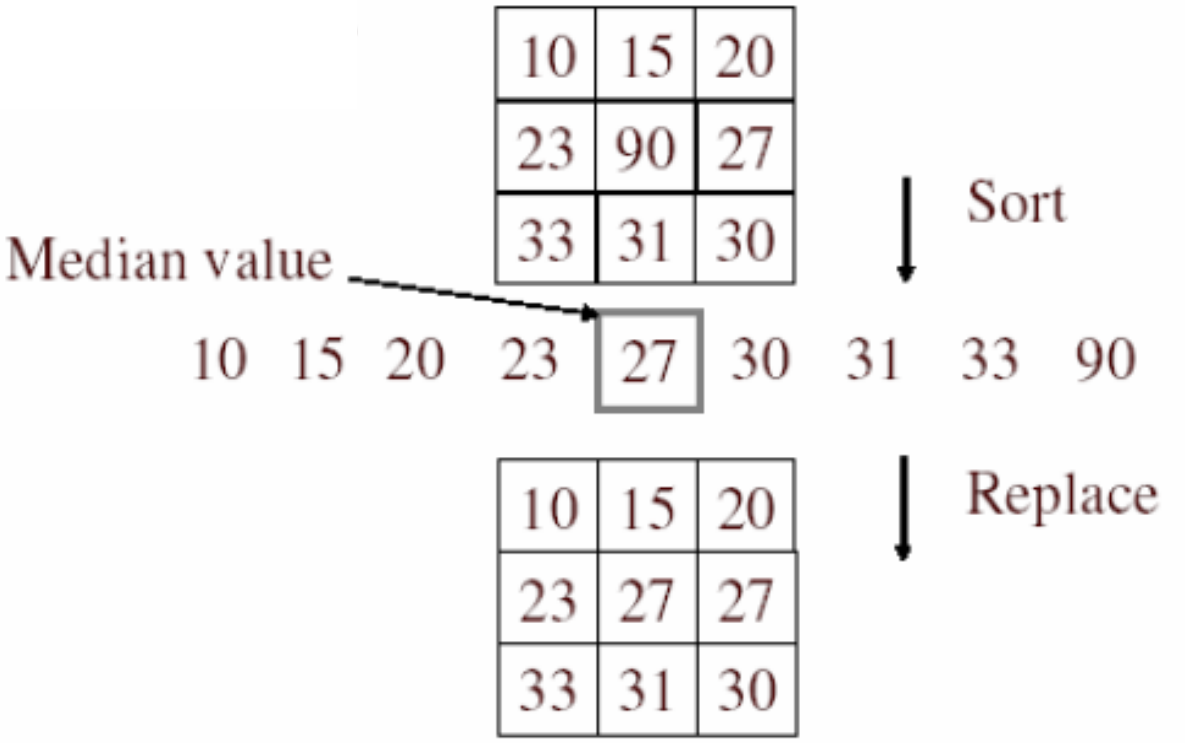
7x7

What's wrong with the results?

Slide from Lana Lazebnik

Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window



- Is median filtering linear?

Source: K. Grauman

Median filter

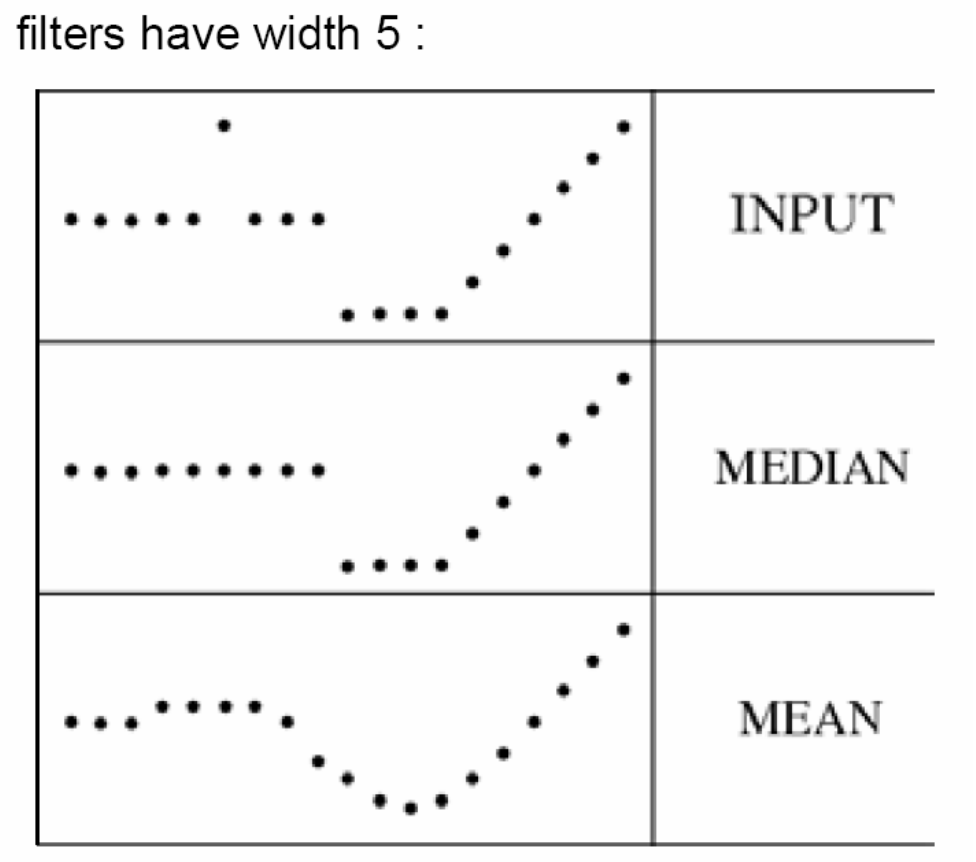
- Is median filtering linear?
- Let's try filtering

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Slide from Lana Lazebnik

Median filter

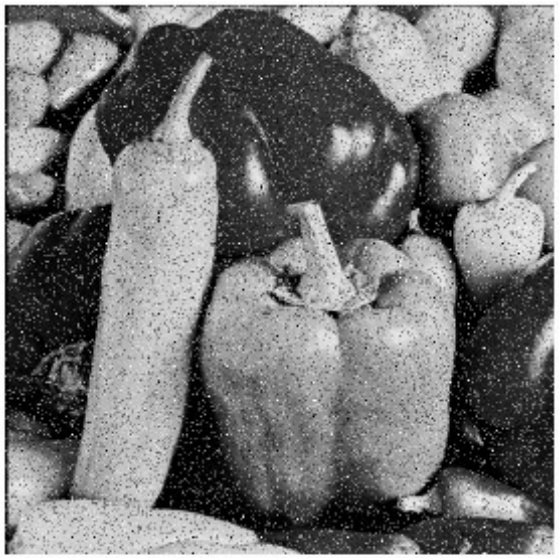
- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers

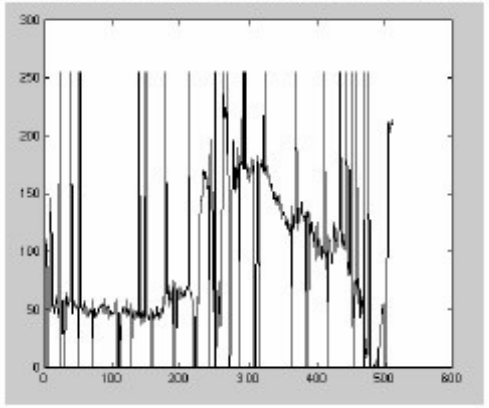


Source: K. Grauman

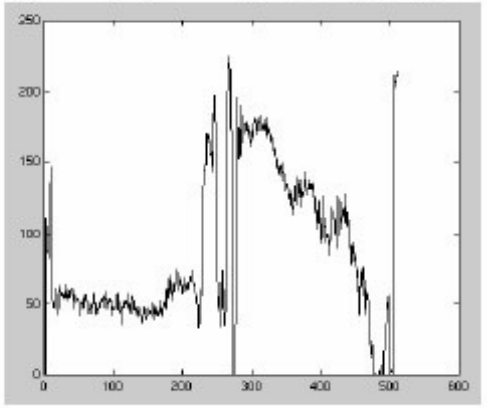
Median filter

Salt-and-pepper noise



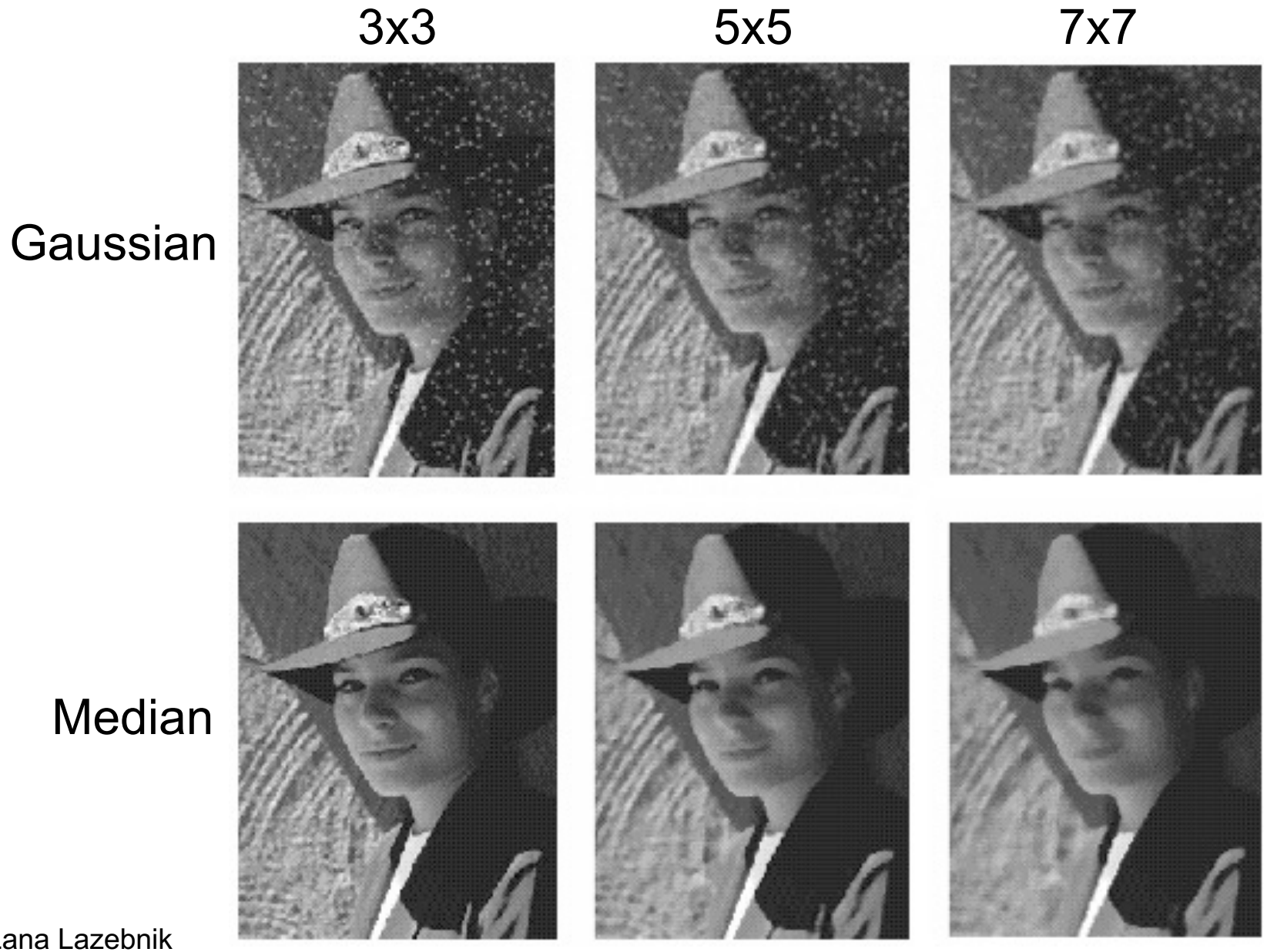


Median filtered



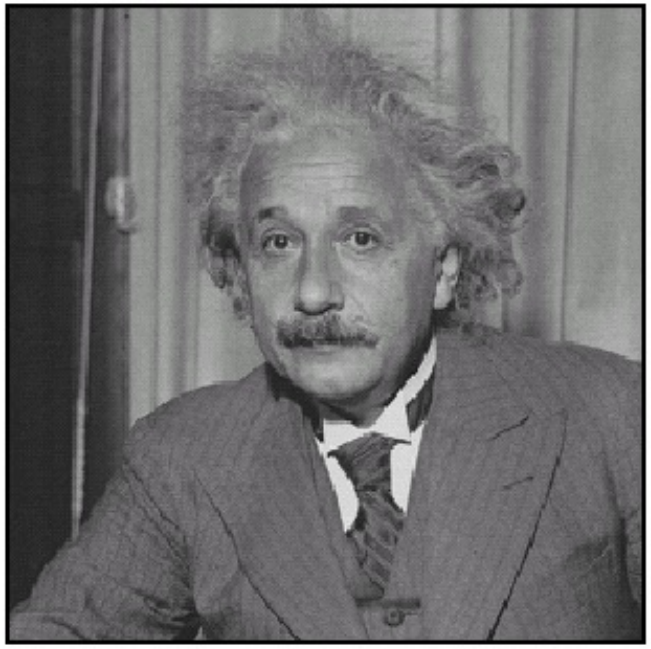
Source: M. Hebert

Gaussian vs. median filtering

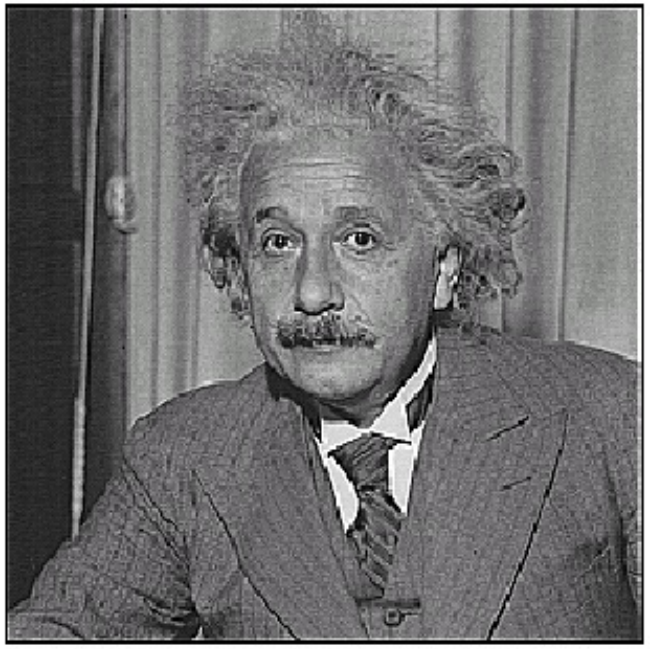


Slide from Lana Lazebnik

Sharpening revisited



before



after

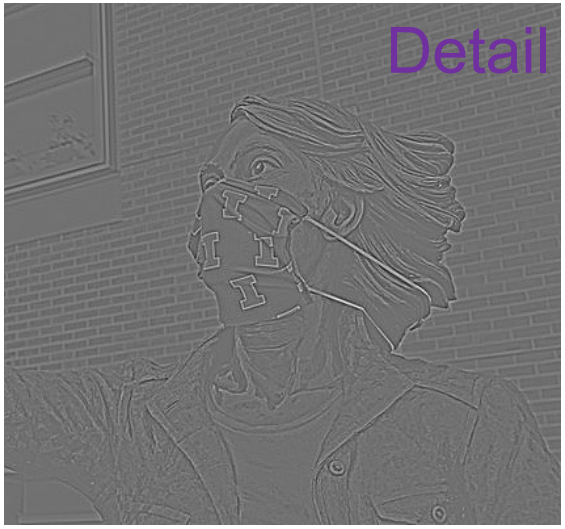
Source: D. Lowe

Sharpening

What does blurring take away?

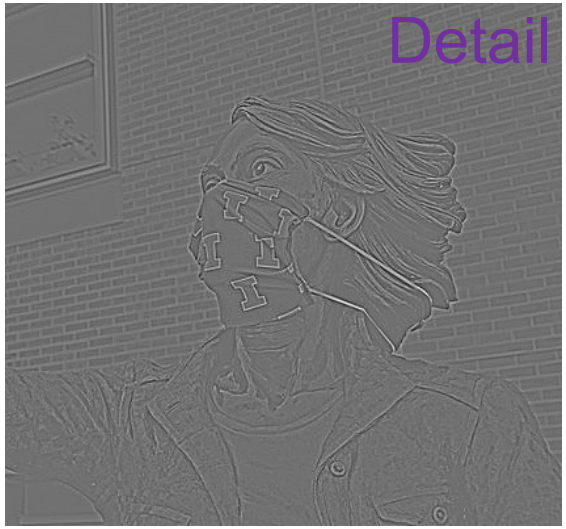
–

=



Let's add it back in.

$+\alpha$



=

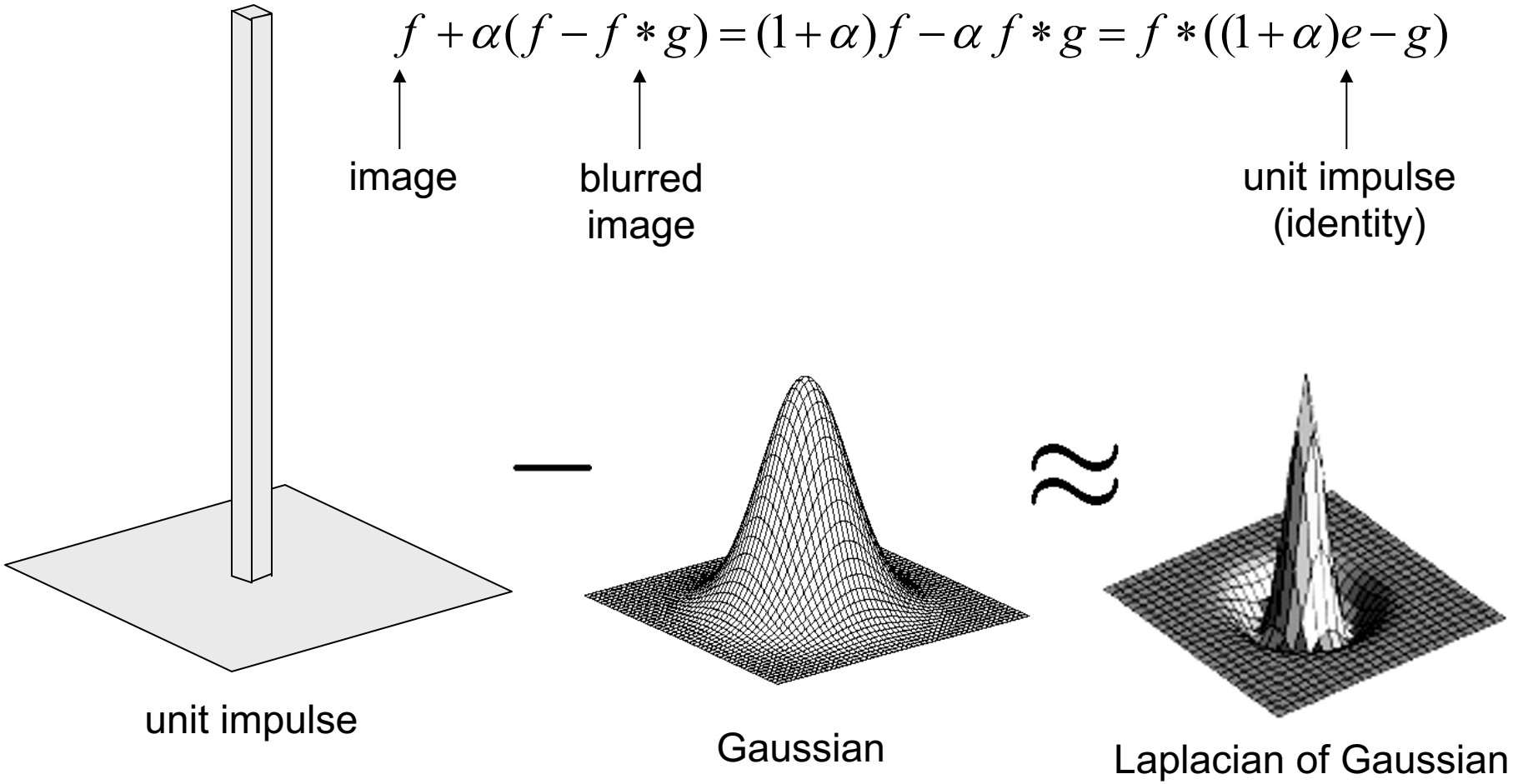
Unsharp mask filter

$$f + \alpha(f - f * g) = (1+\alpha)f - \alpha\, f * g = f * ((1+\alpha)e - g)$$

image

blurred image

unit impulse (identity)



unit impulse

Gaussian

Laplacian of Gaussian

Next Class: Frequency view of filtering

Slide from Lana Lazebnik

Guess the filter

Guess the filter

Review: Image filtering

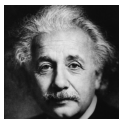
- Convolution
- Box vs. Gaussian filter
- Separability
- Median filter

Filtering (in Frequency Domain)

While we wait,

Who do you see in this picture?

A. Albert Einstein



B. Marilyn Monroe

C. None of these

CS 543 / ECE 549 – Saurabh Gupta

Many slides from Derek Hoiem.

Today's Class

- Fourier transforms
- Filtering in frequency domain
- Sampling
- Image Pyramids

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

Gaussian

Box filter

Thinking in terms of frequency

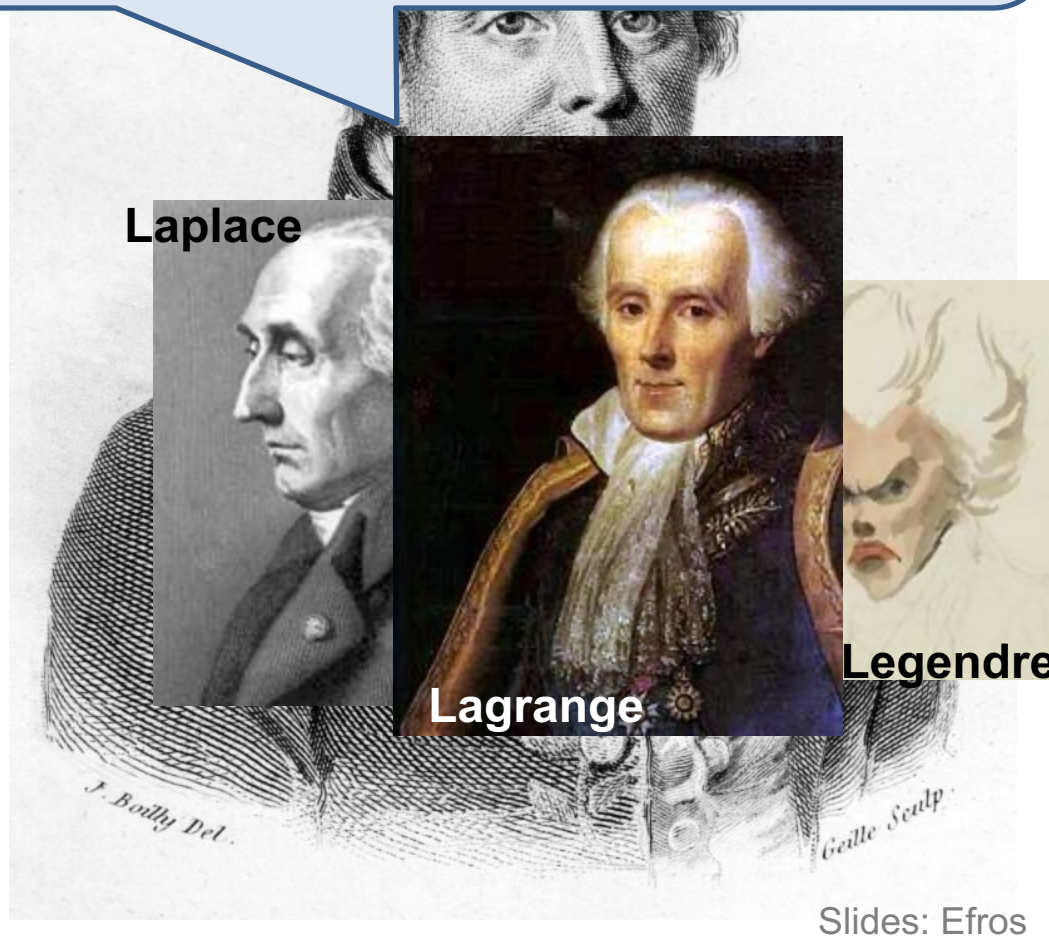
Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

Any *univariate function can* *rewritten as a weighted sum* *sines and cosines of differen* *frequencies.*

...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.

- Don't believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it's (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions



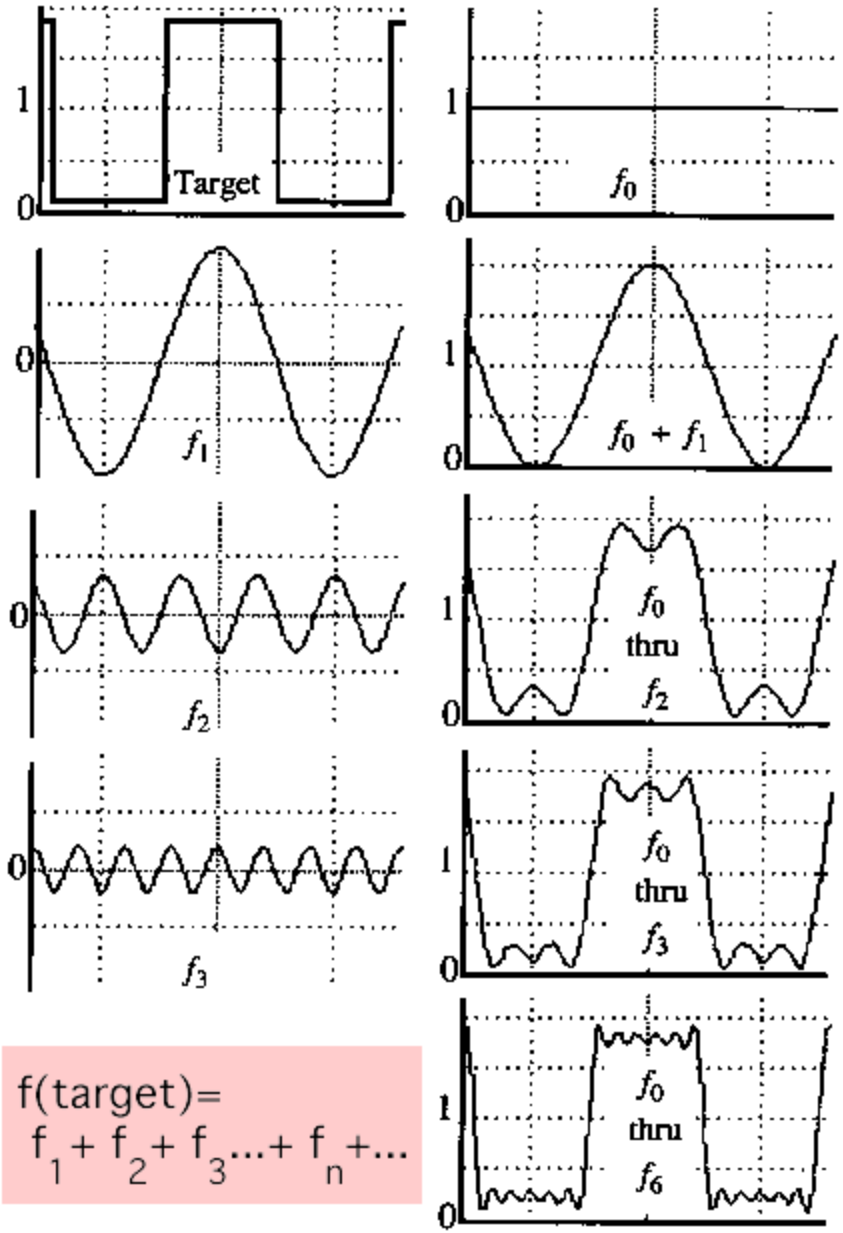
Slides: Efros

A sum of sines

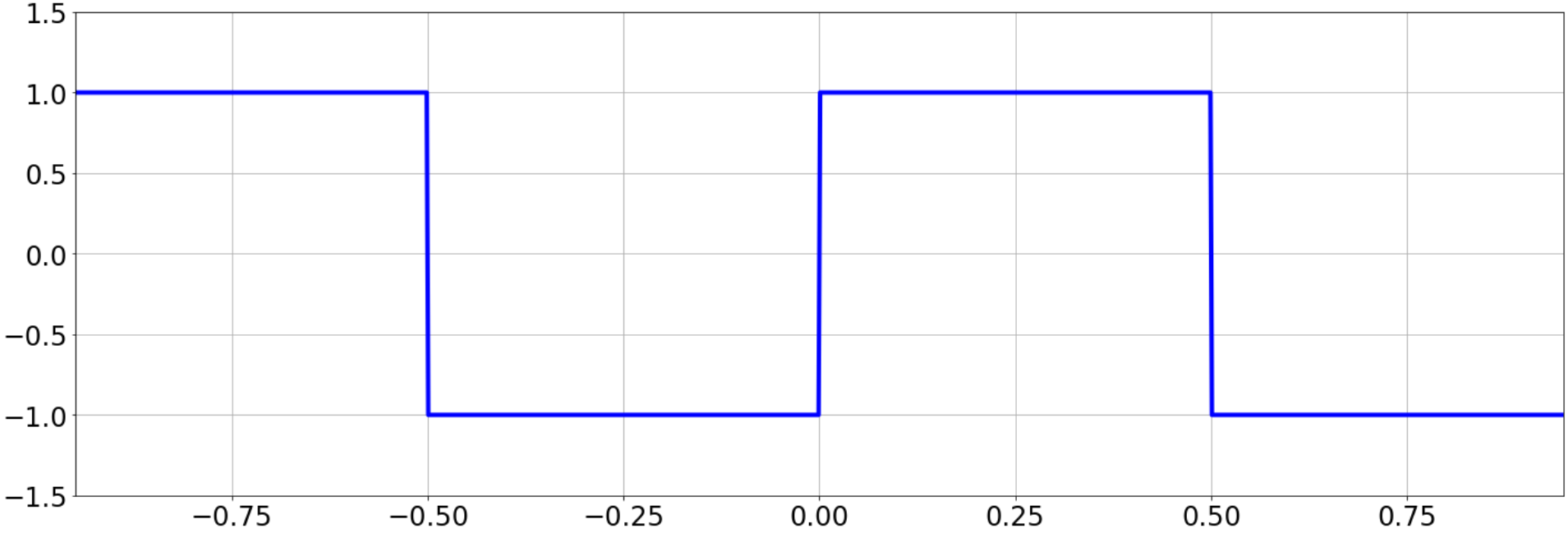
Our building block:

$$A\sin(\omega x + \phi)$$

Add enough of them to get any signal *f(x)* you want!



Frequency Spectra

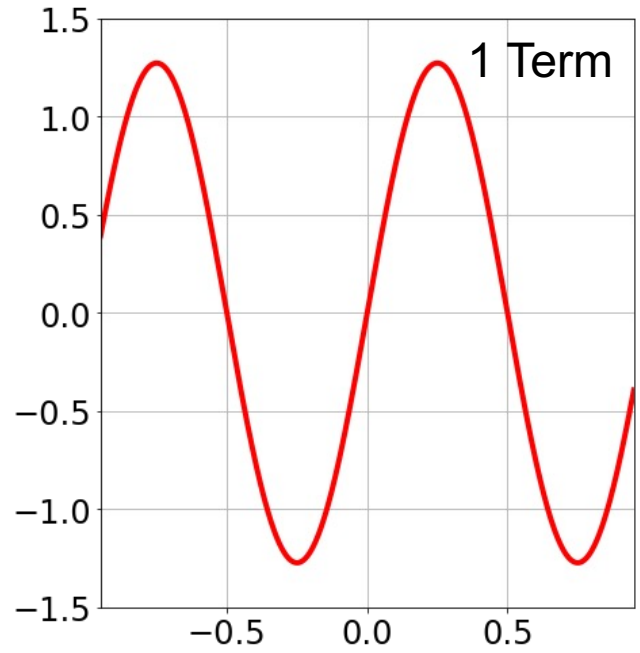


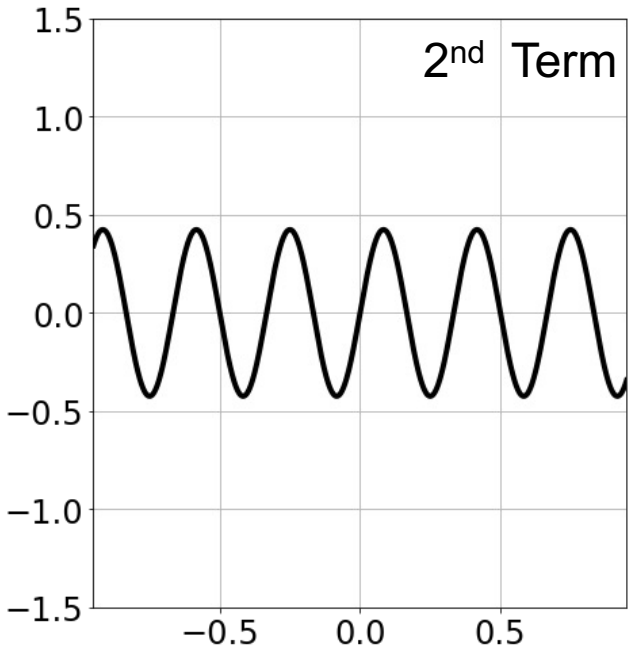
Square Wave:

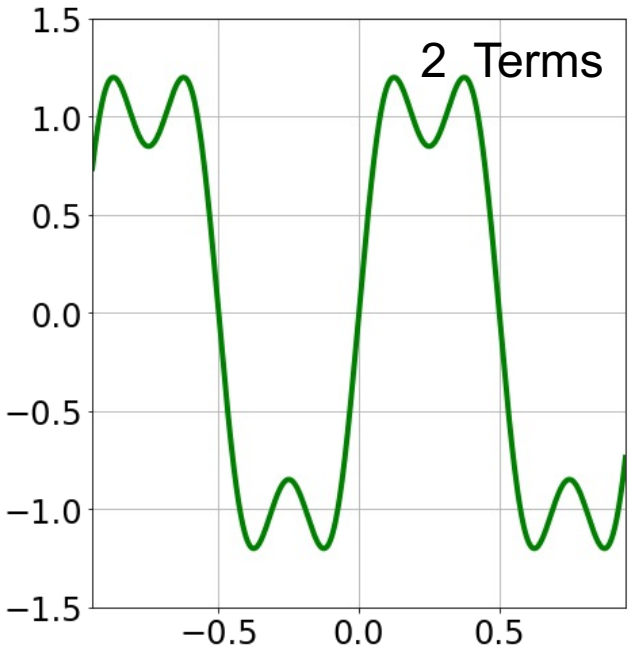
$$f(x) = \begin{cases} 1, & \text{if } \mathrm{frac}(x) < 0.5 \\ -1, & \text{otherwise} \end{cases}$$

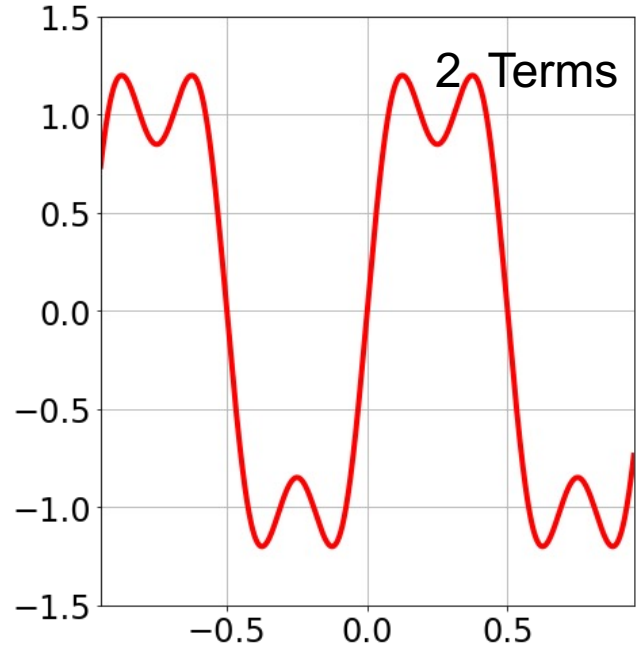
Fourier Transform:

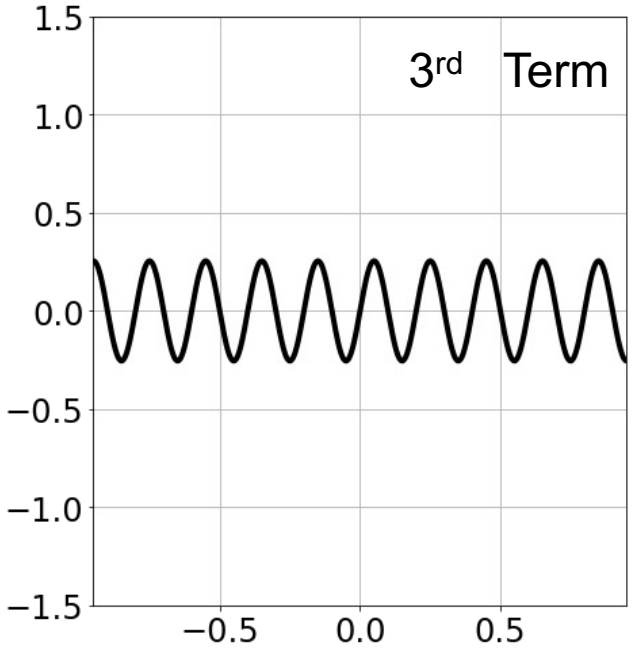
$$\frac{4}{\pi}\left(\frac{\sin(2\pi.1.x)}{1} + \frac{\sin(2\pi.3.x)}{3} + \frac{\sin(2\pi.5.x)}{5} + \ \ldots\right)$$

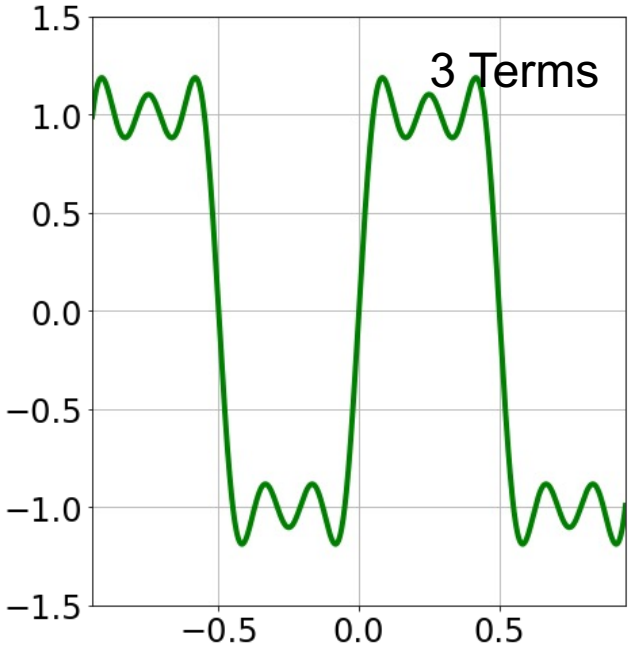

1 Term
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5

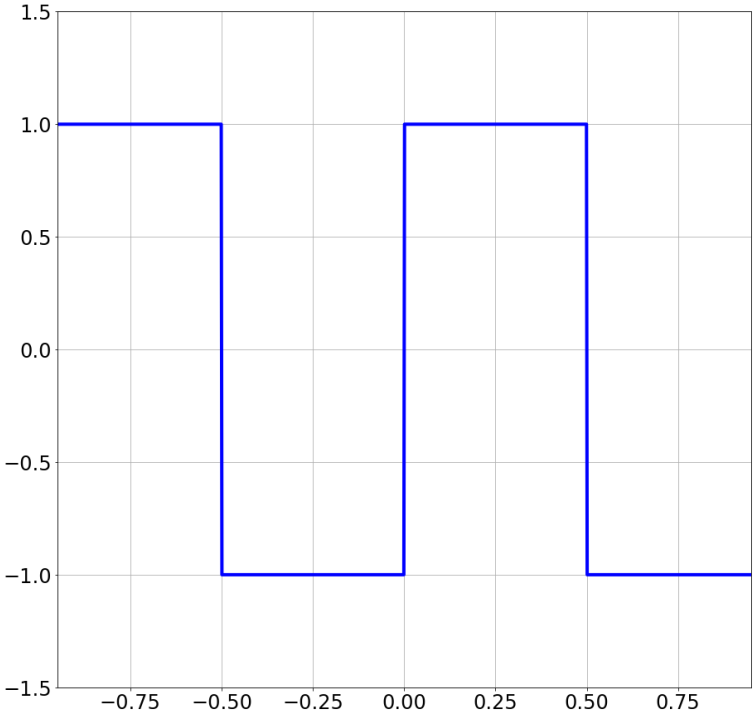

2nd Term
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5

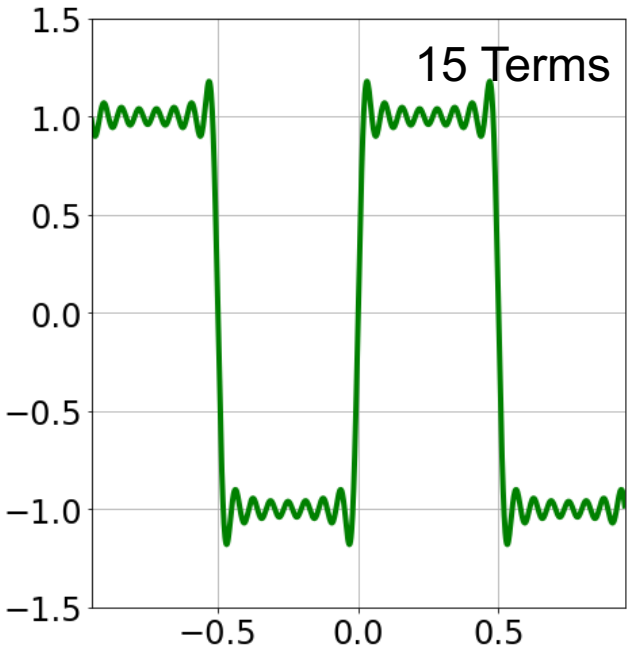

2 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5

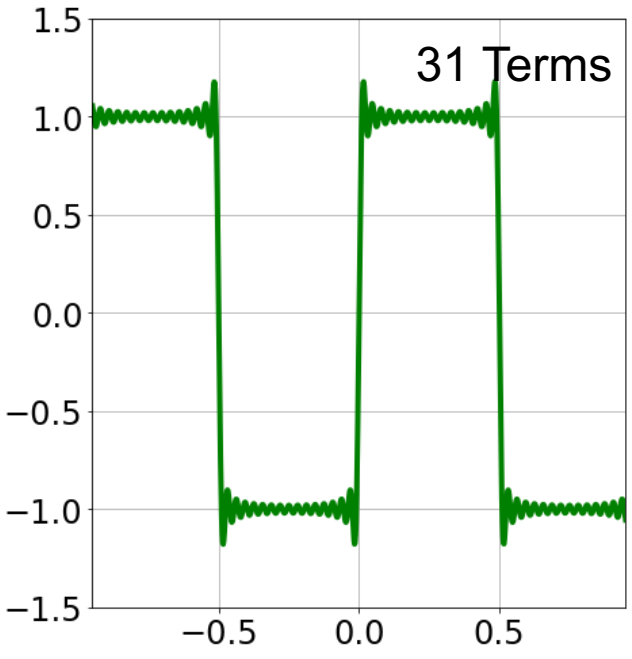

2 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5

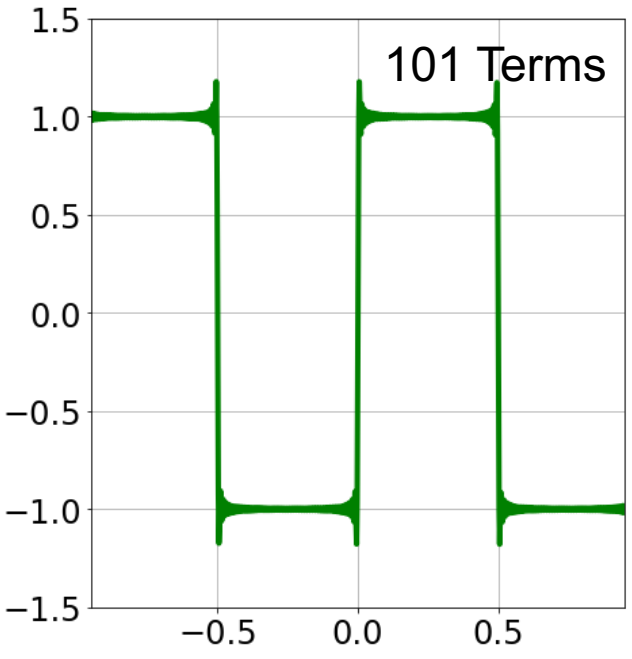

3rd Term
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5

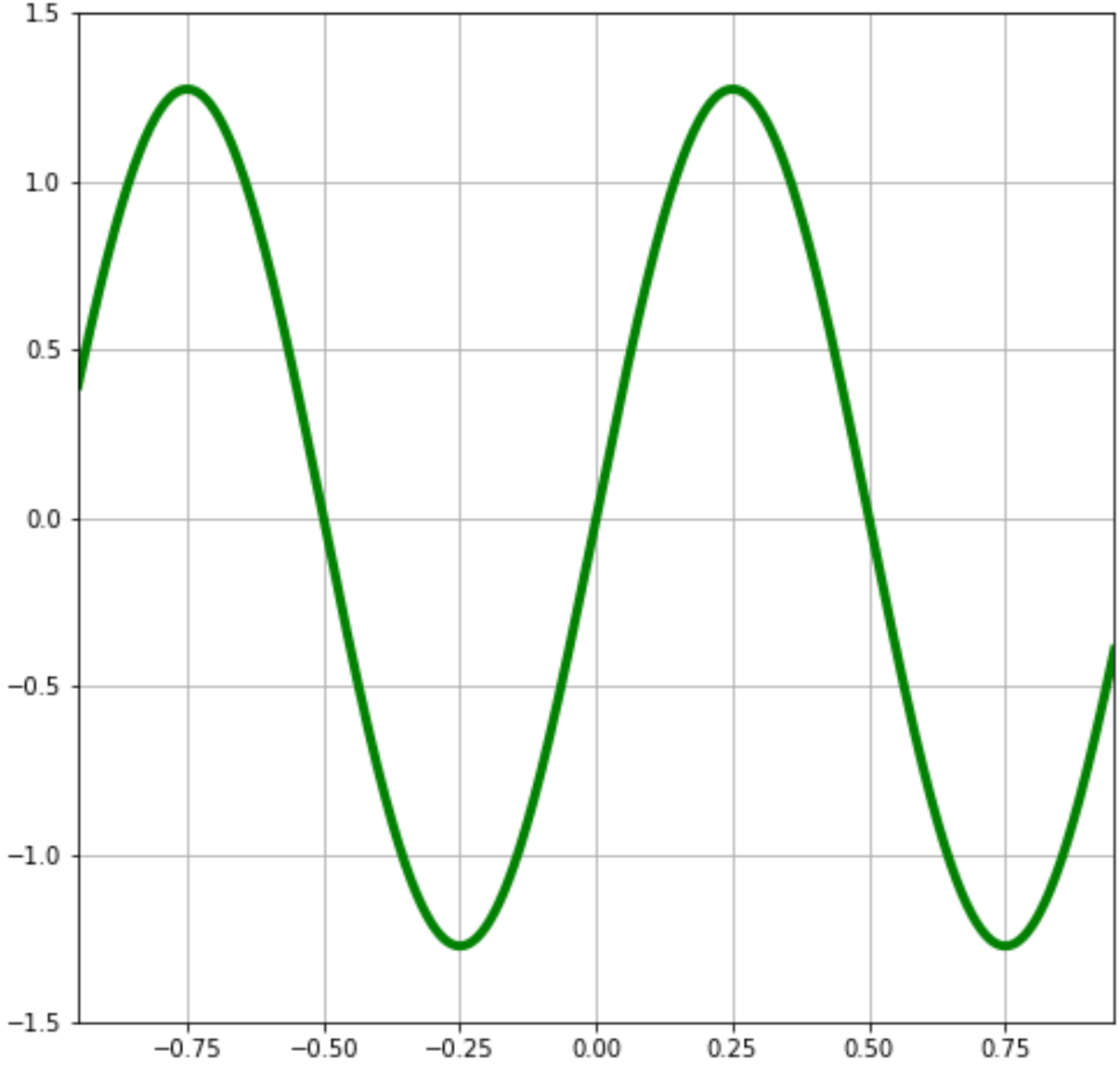

3 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75


15 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5


31 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5


101 Terms
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.5
0.0
0.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75

Fourier Transform

- Fourier transform stores the magnitude and phase at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of complex numbers
- Amplitude: $A = \sqrt{R(\omega)^2 + I(\omega)^2}$
- Phase: $\phi = tan^{-1} \frac{I(\omega)}{R(\omega)}$

Computing the Fourier Transform

- $H(\omega) = \mathcal{F}[h(x)]$
- Continuous:
  - $H(\omega) = \int_{-\infty}^{\infty} h(x) e^{-j\omega x} dx$
- Discrete:
  - $H(k) = \frac{1}{N} \sum_{x=0}^{N-1} h(x) e^{-j2\pi kx/N}$
- Euler's Formula:
  - $e^{jnx} = \cos(nx) + j \sin(nx)$

Properties of Fourier Transforms

- Linearity:
  - $\mathcal{F}[ax(t) + by(t)] = a\mathcal{F}[x(t)] + b\mathcal{F}[y(t)]$

- Fourier transform of a real signal is symmetric about the origin

- The energy of the signal is the same as the energy of its Fourier transform

See Szeliski Book (3.4)

The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$-\mathcal{F}[g * h] = \mathcal{F}[g]\mathcal{F}[h]$$

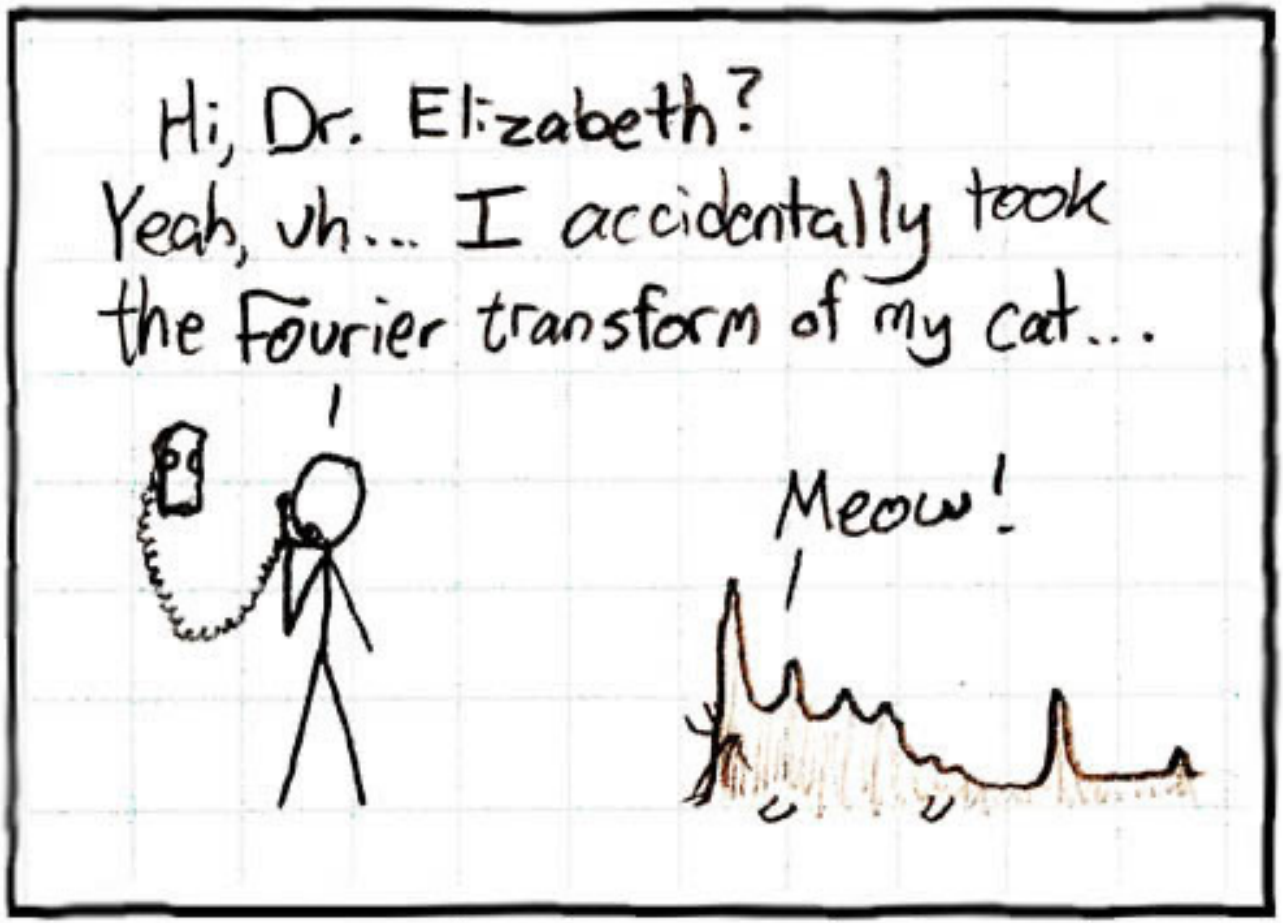
- The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms

$$-\mathcal{F}^{-1}[gh] = \mathcal{F}^{-1}[g] * \mathcal{F}^{-1}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

Other signals

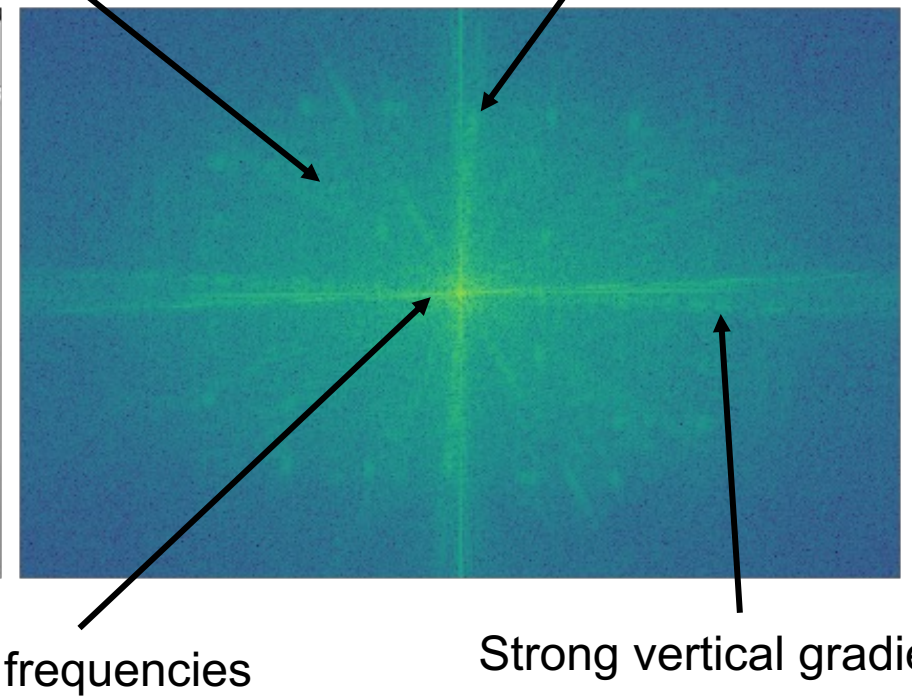
- We can also think of all kinds of other signals the same way



xkcd.com

Images

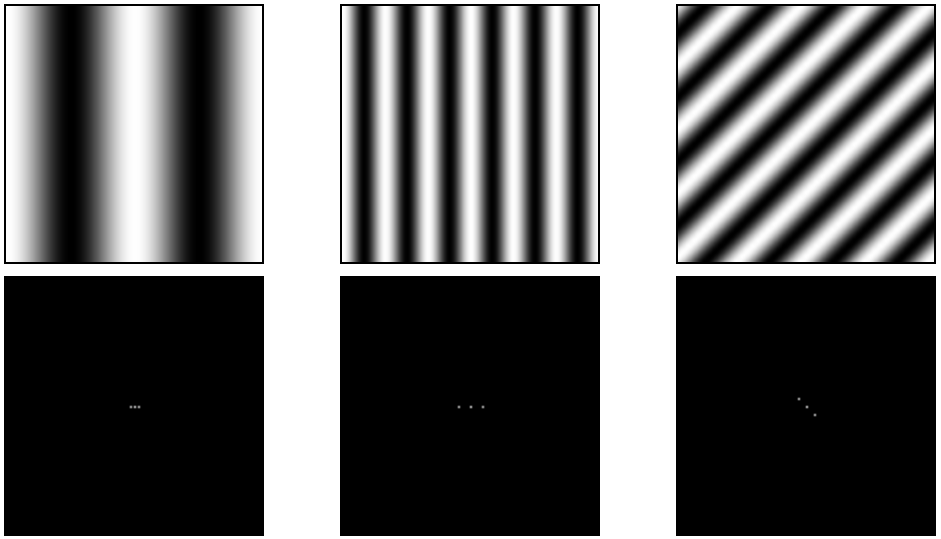
$$H(k,l) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} h(m,n)e^{-j2\pi\left(\frac{k}{M}m+\frac{l}{N}n\right)}$$



Fourier analysis in images

Intensity Image

Fourier Image



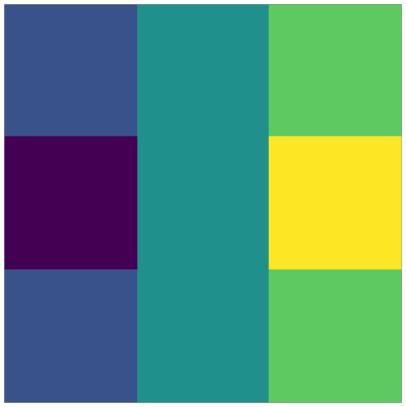
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Filtering in spatial domain

-1	0	1
-2	0	2
-1	0	1

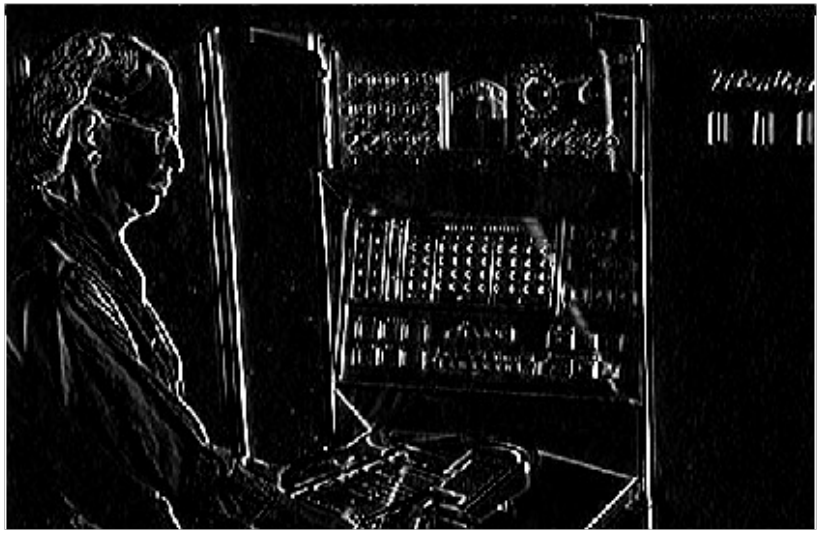
Original

*



=

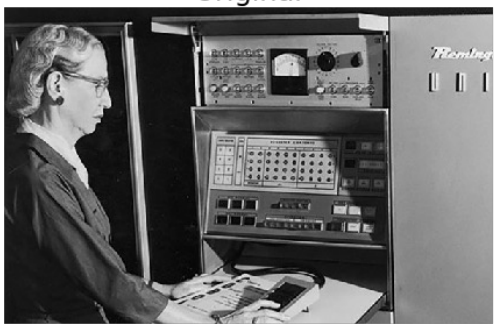
Filtered via Convolution



Filtering in frequency domain

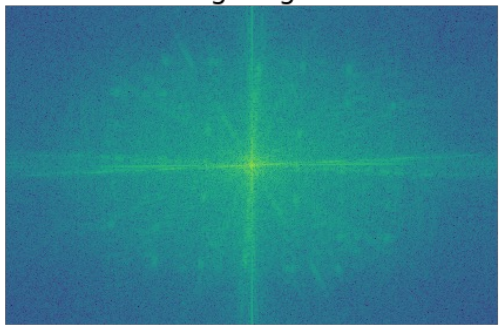
FFT

Original



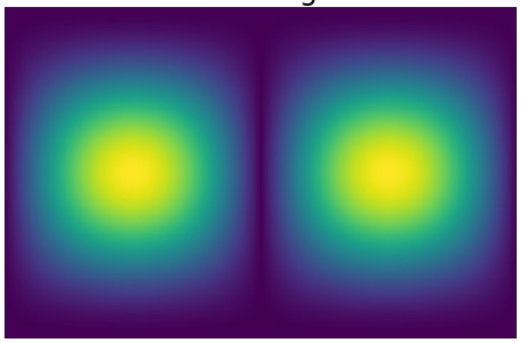
FFT

FFT Log Magnitude



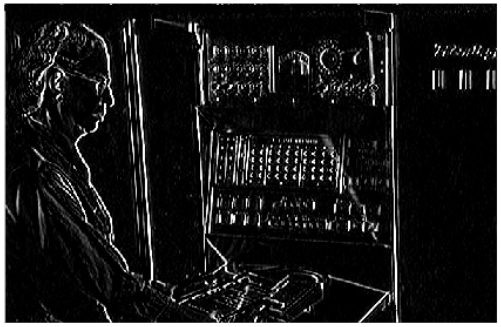
×

Filter FFT Magnitude



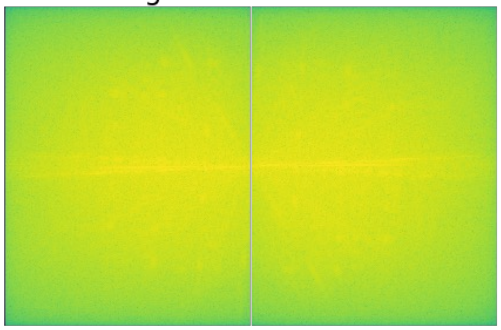
=

Filtered via FFT



Inverse FFT

Image FFT .* Filter FFT



Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

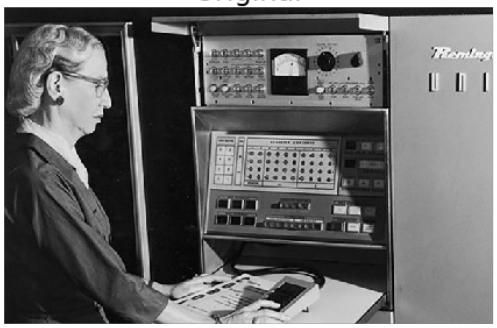
Gaussian

Box filter

Filtering in frequency domain (Box)

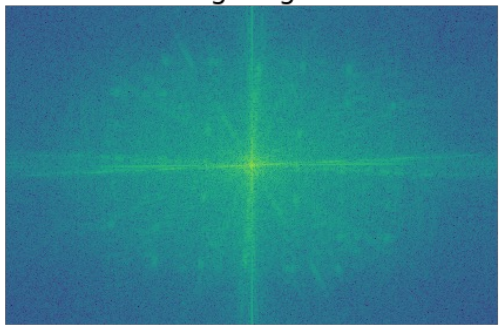
FFT

Original



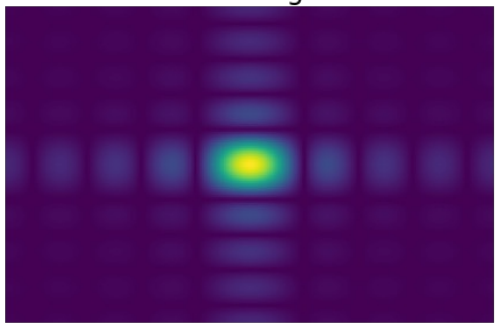
FFT

FFT Log Magnitude



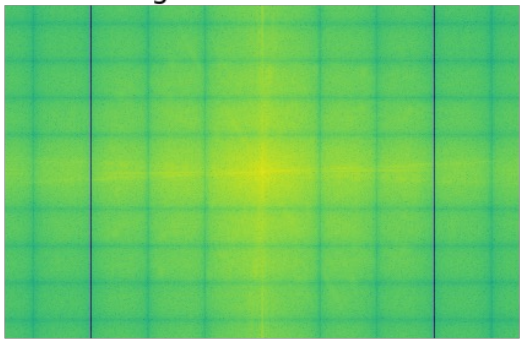
×

Filter FFT Magnitude



=

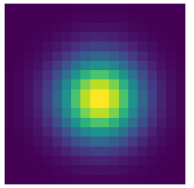
Image FFT .* Filter FFT



Inverse FFT

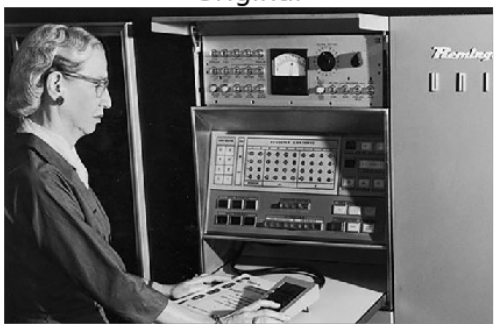
Filtered via FFT

Filtering in frequency domain (Gaussian)



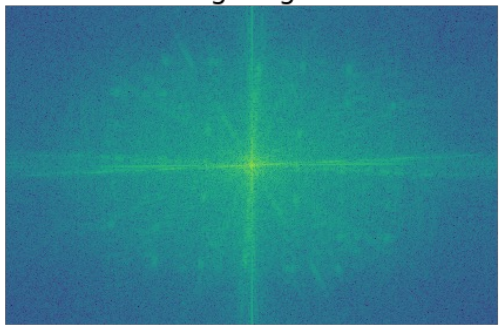
FFT

Original



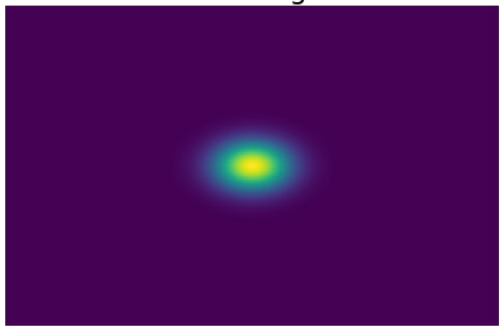
FFT

FFT Log Magnitude



×

Filter FFT Magnitude

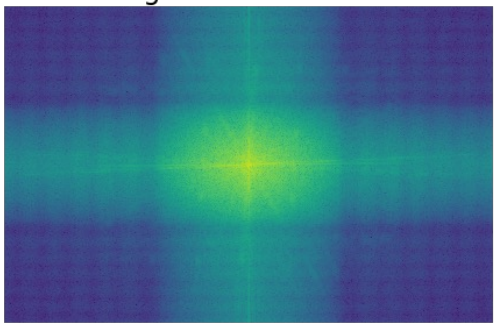


=

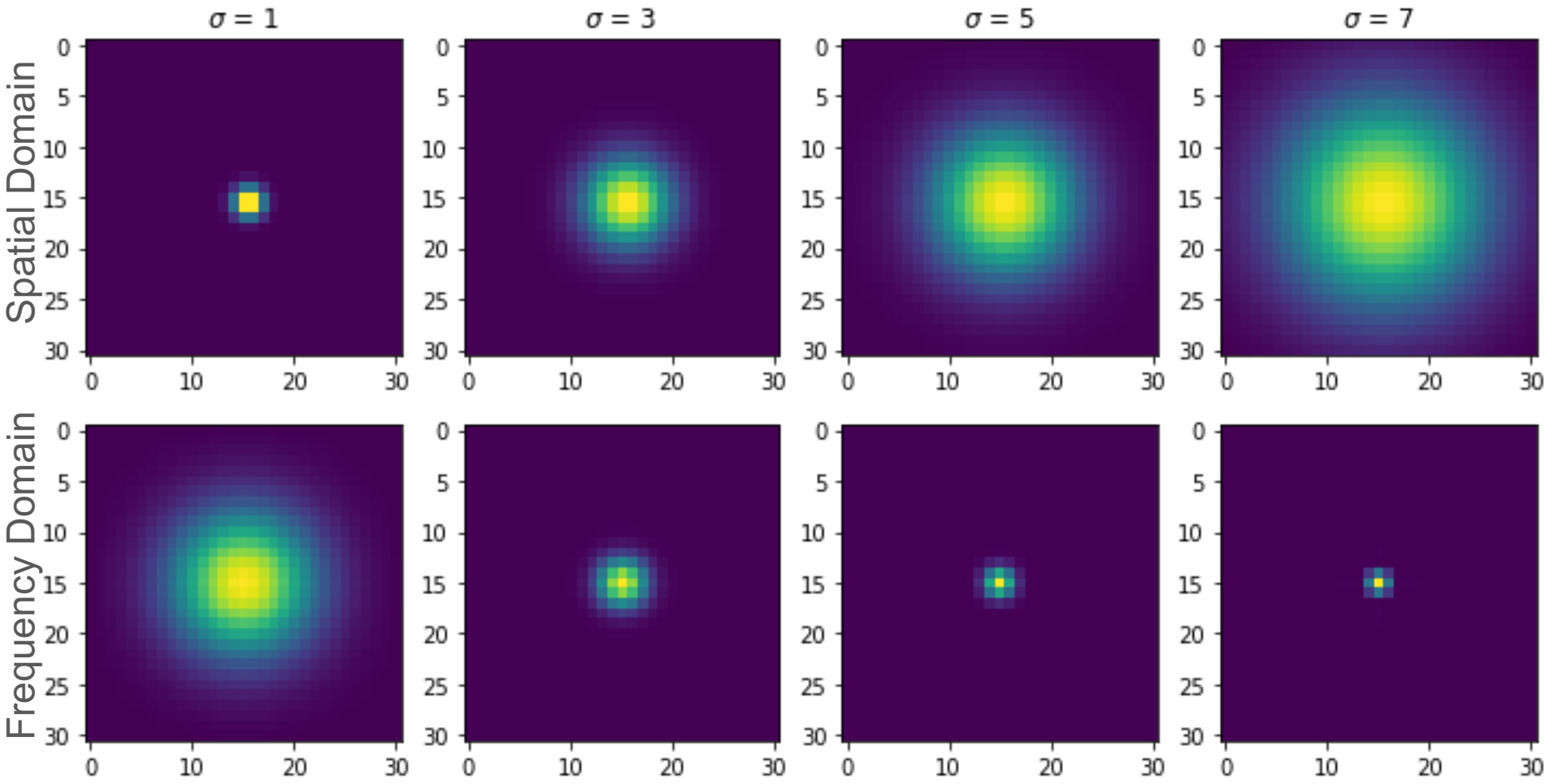
Filtered via FFT

Inverse FFT

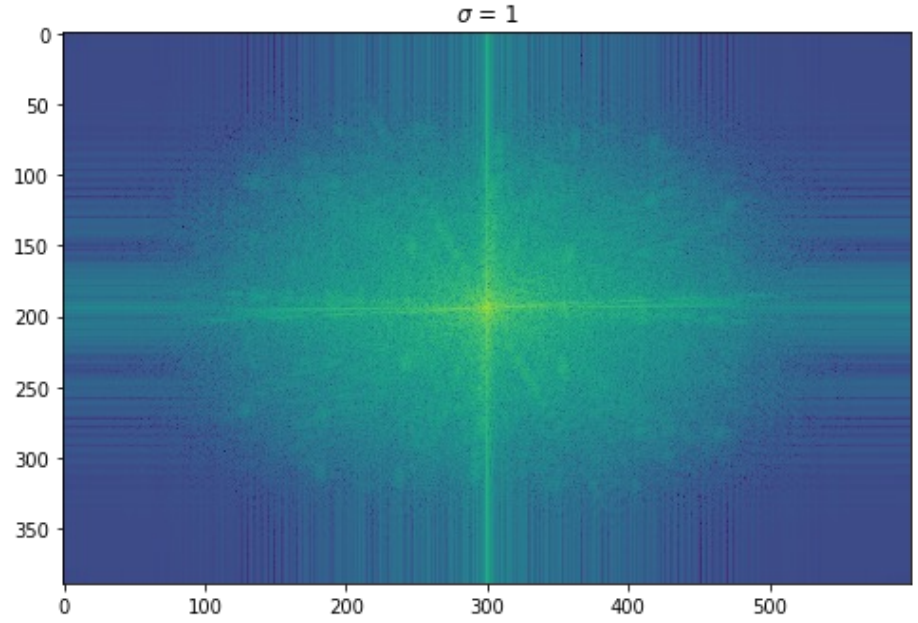
Image FFT .* Filter FFT

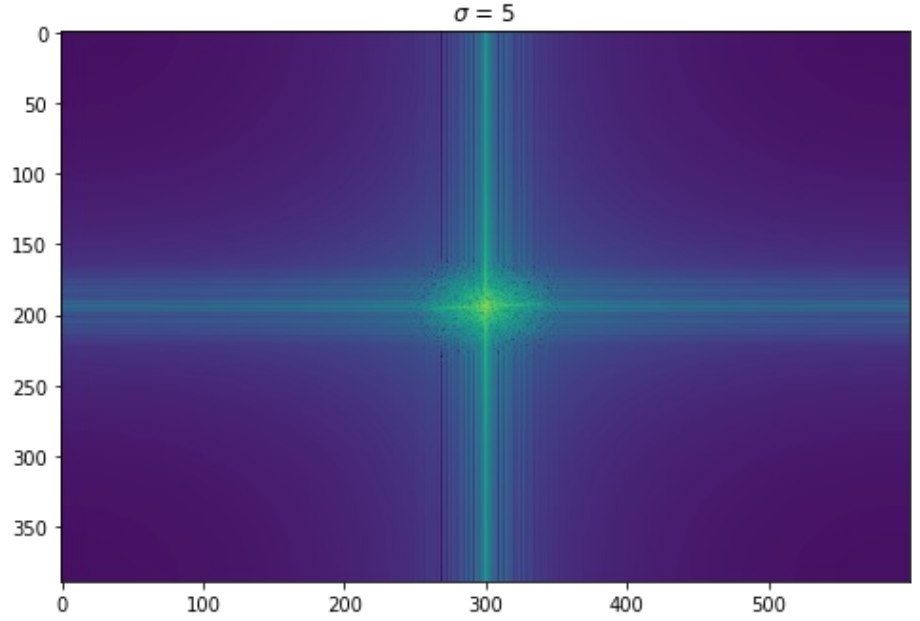


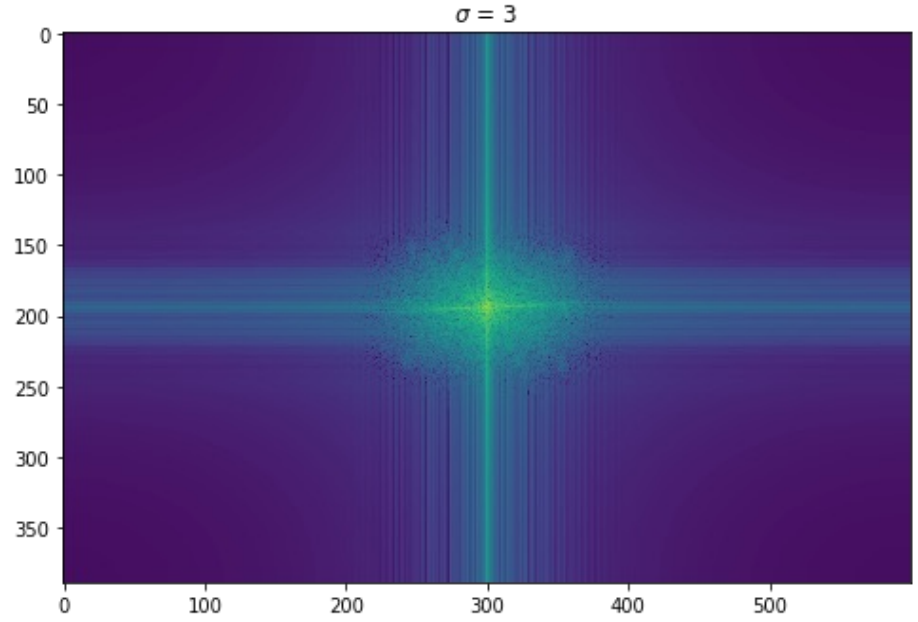
Filtering in frequency domain (Gaussian)

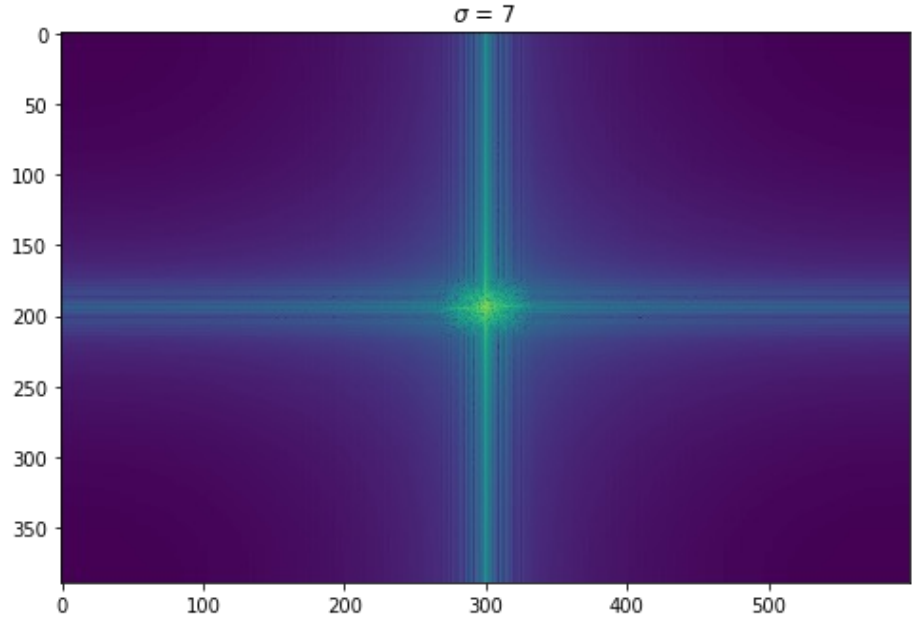


Filtering in frequency domain (Gaussian)

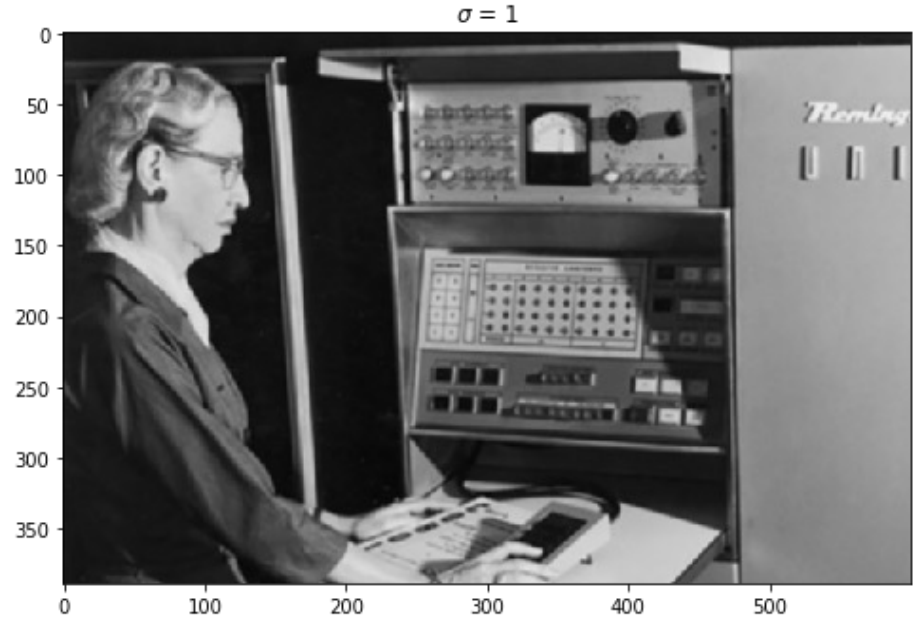


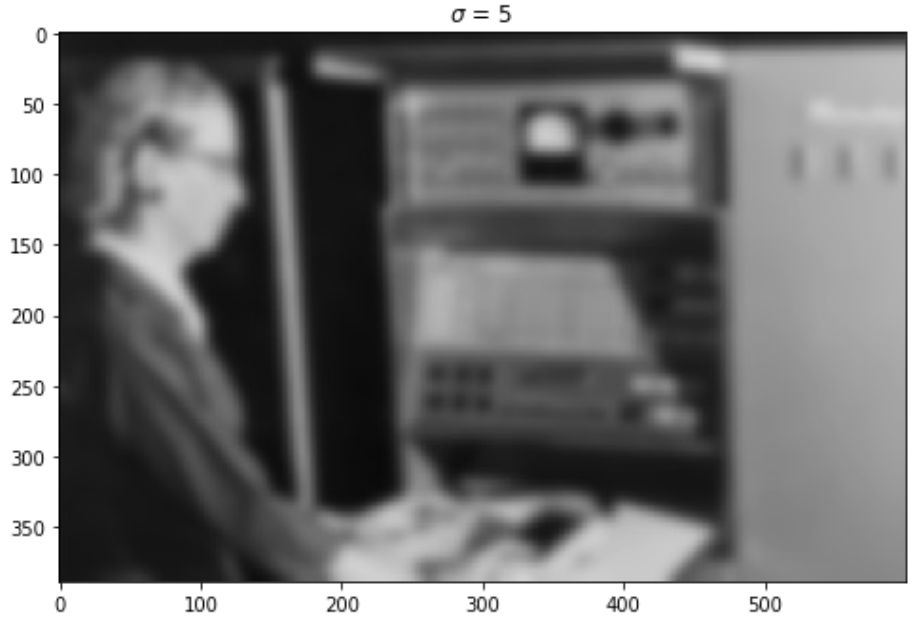


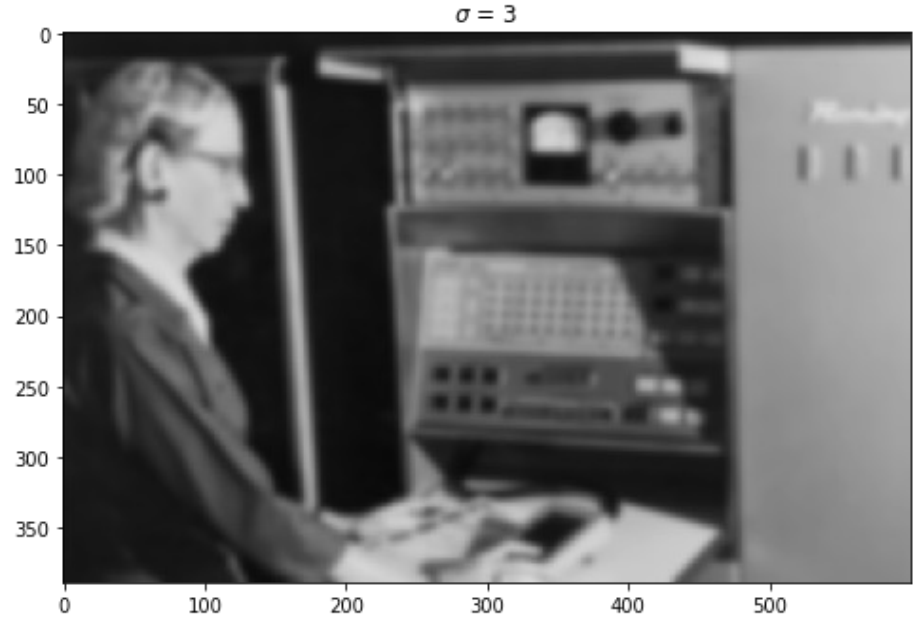


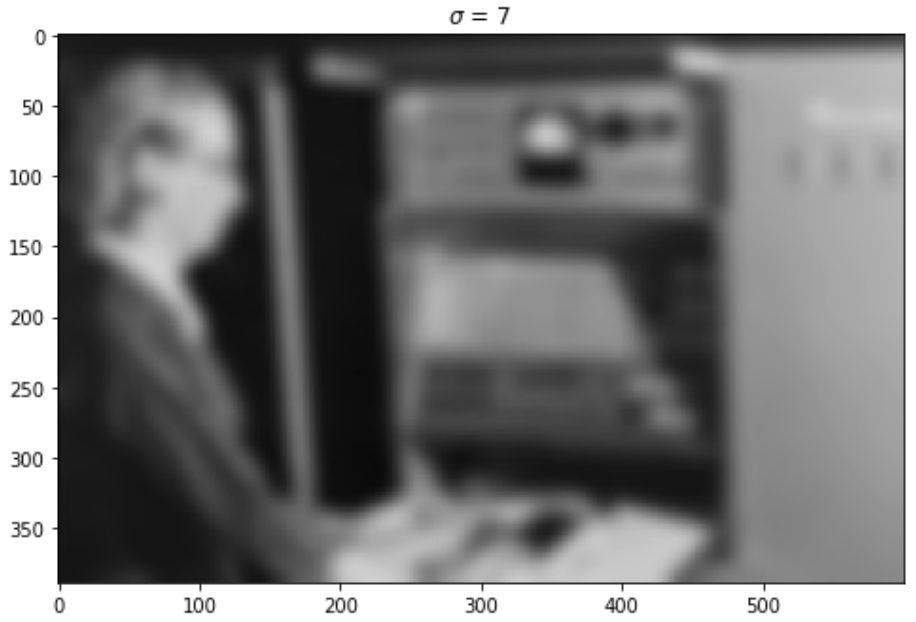


Filtering in frequency domain (Gaussian)









Sampling

Why does a lower resolution image still make sense to us? What do we lose?

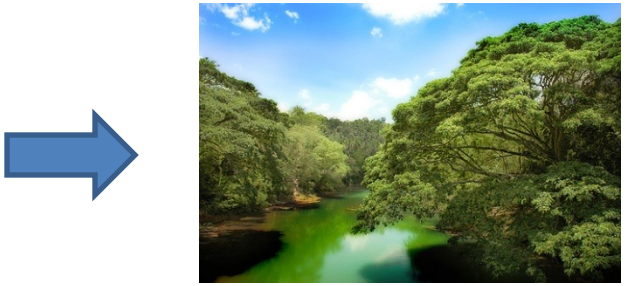
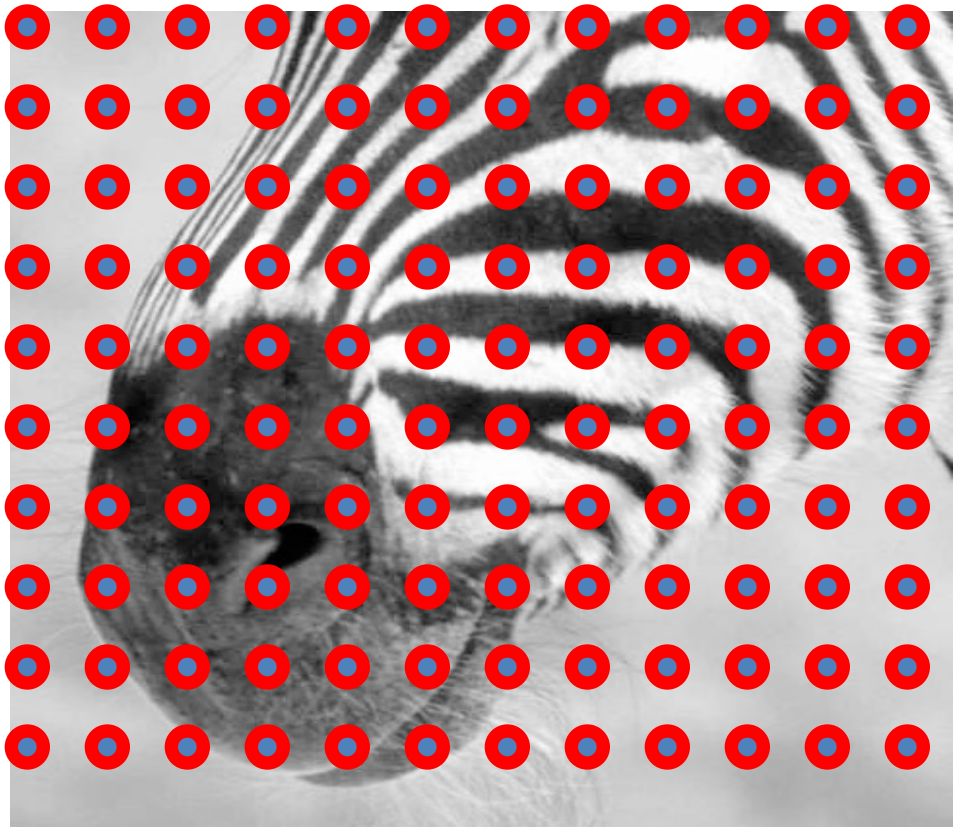


Image: http://www.flickr.com/photos/igorms/136916757/

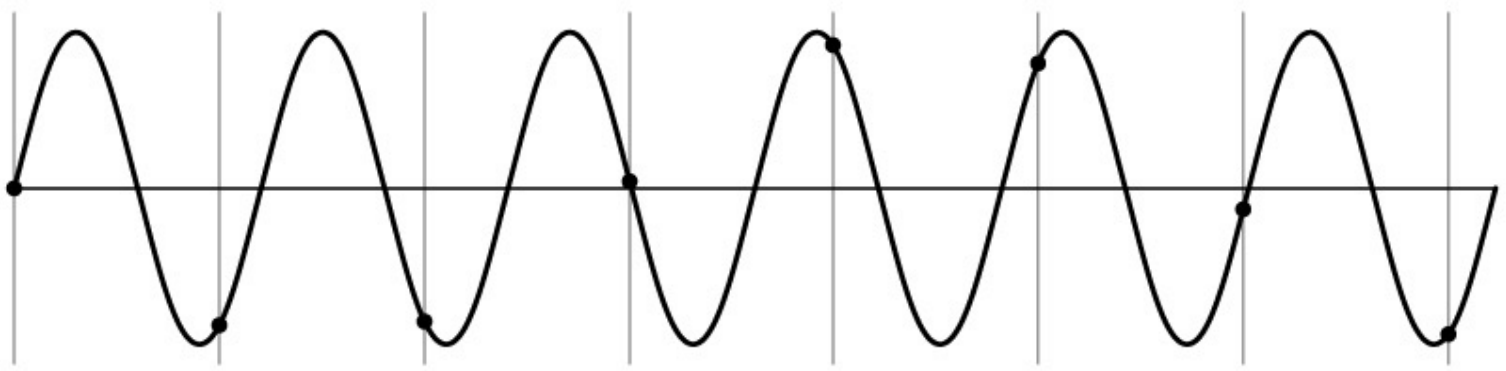
Subsampling by a factor of 2



Throw away every other row and column to create a 1/2 size image

Aliasing problem

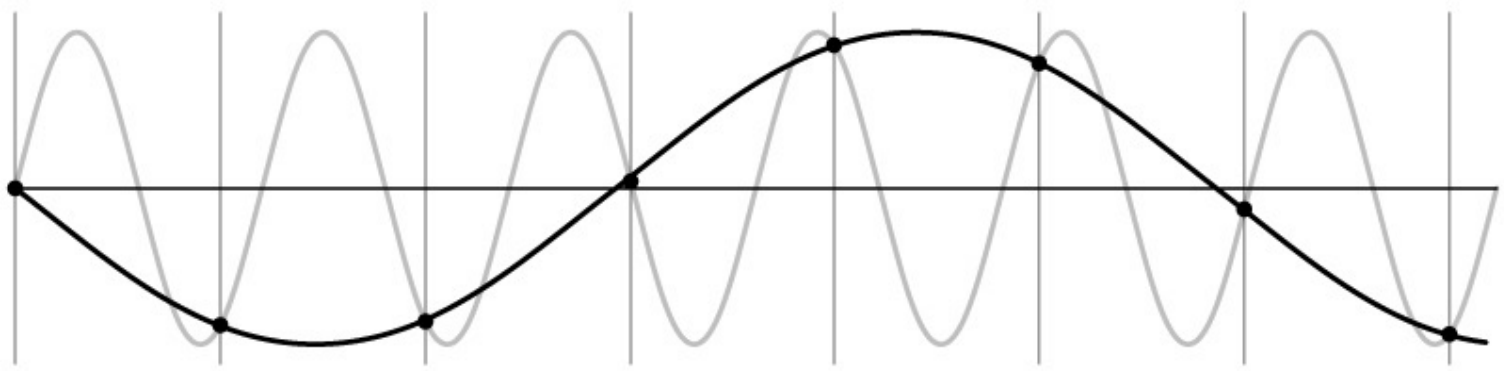
- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

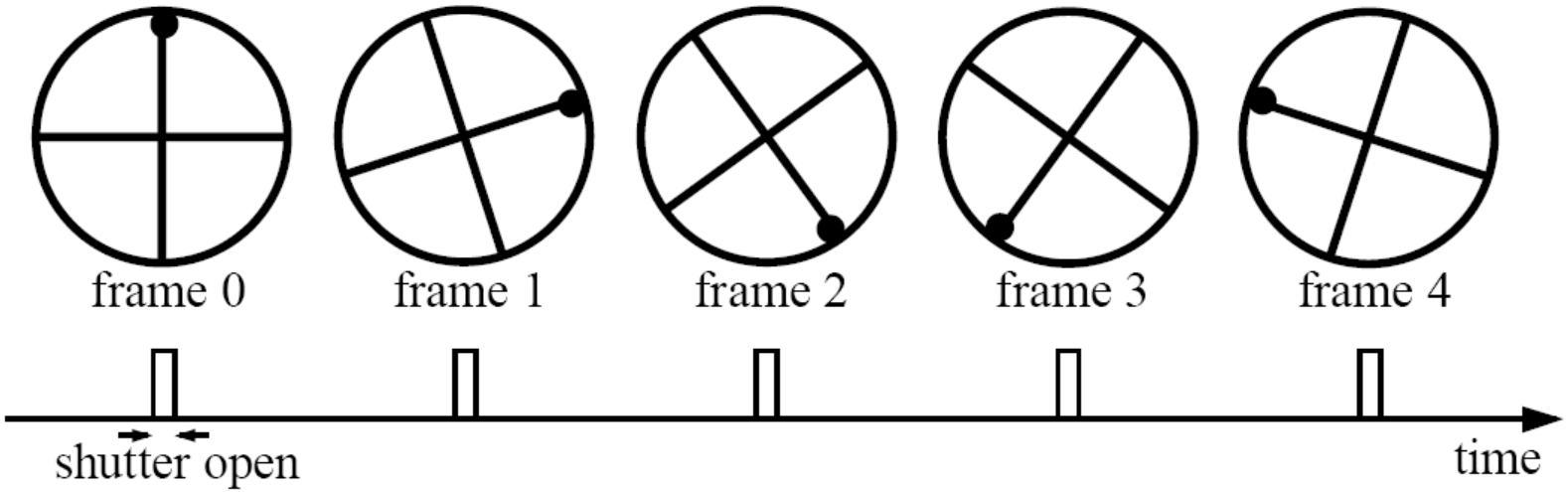
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies” See
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”

Source: D. Forsyth

Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

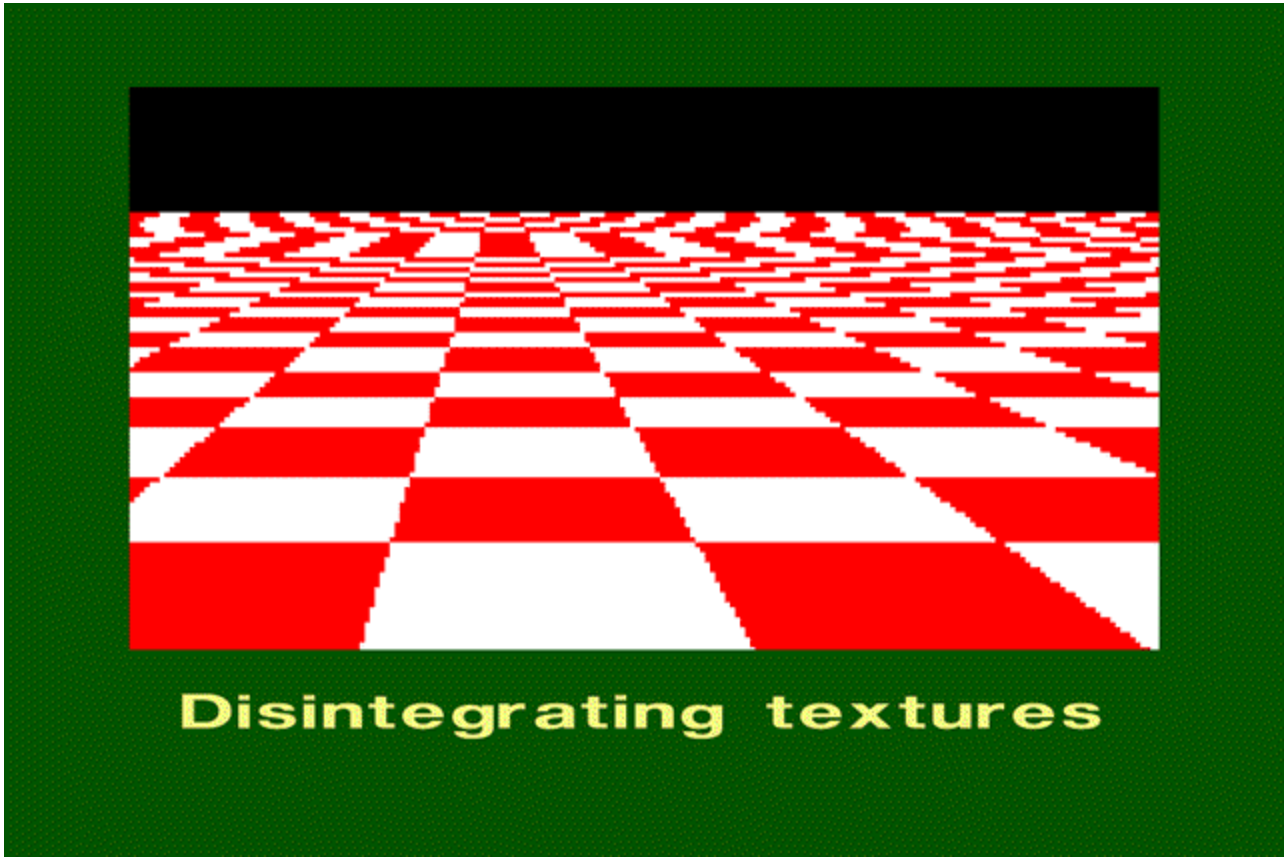
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

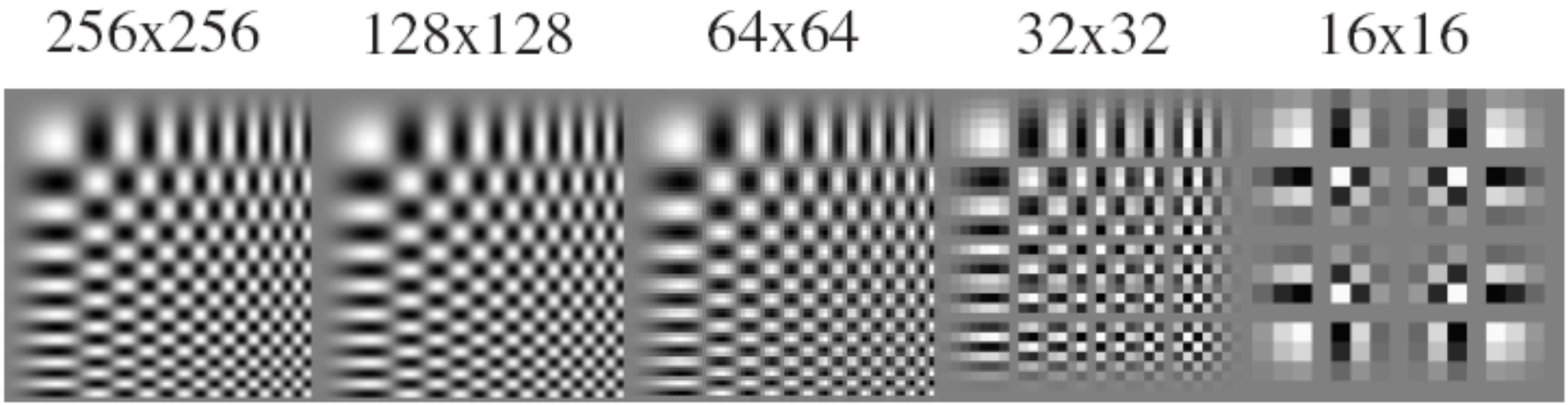
Slide by Steve Seitz

Aliasing in graphics



Source: A. Efros

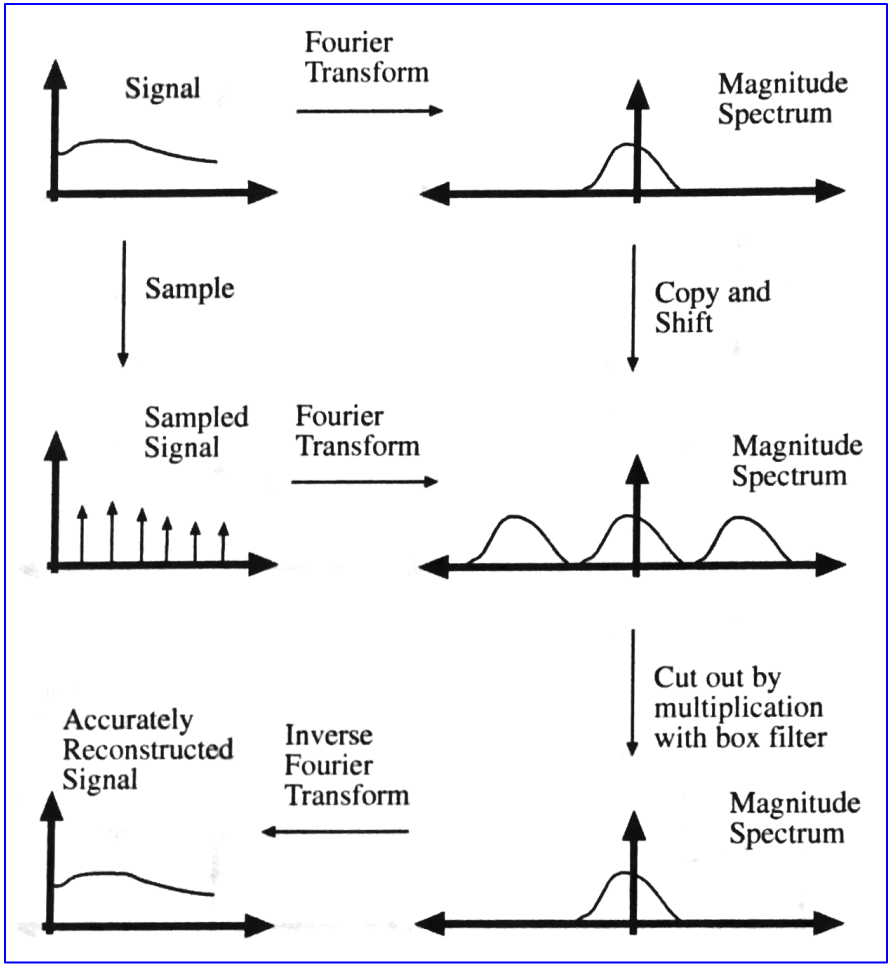
Sampling and aliasing



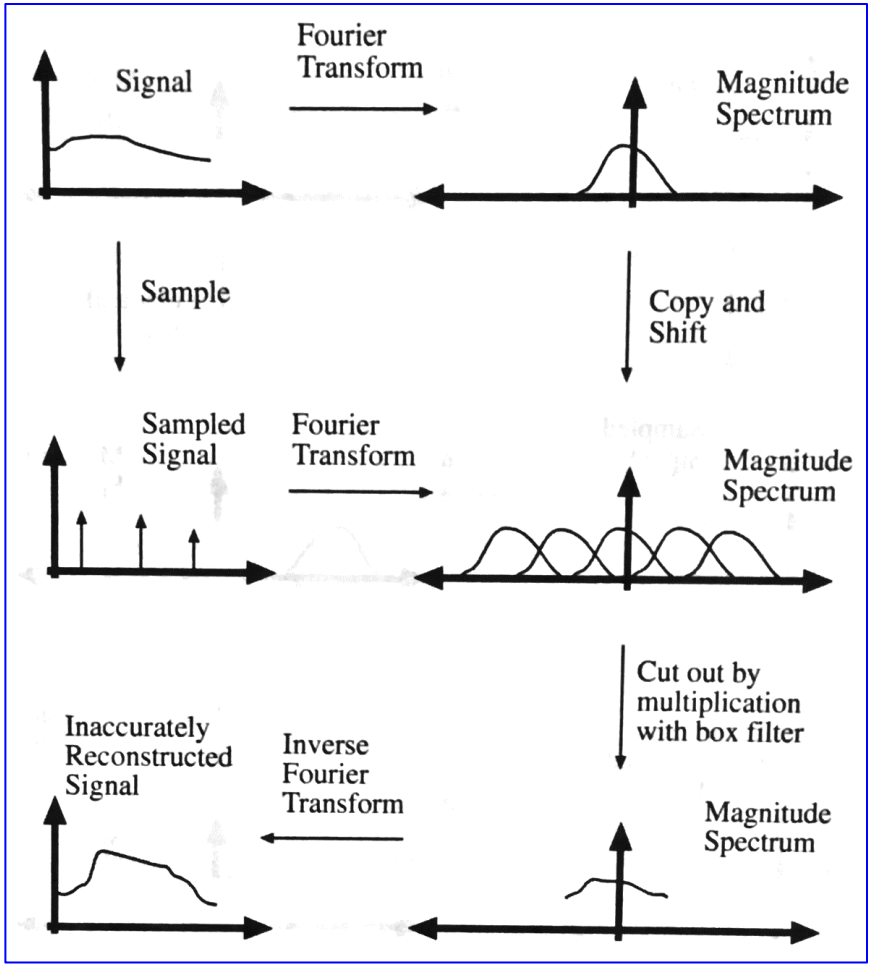
Slide from Derek Hoiem.

Aliasing in Frequency Domain

No Aliasing



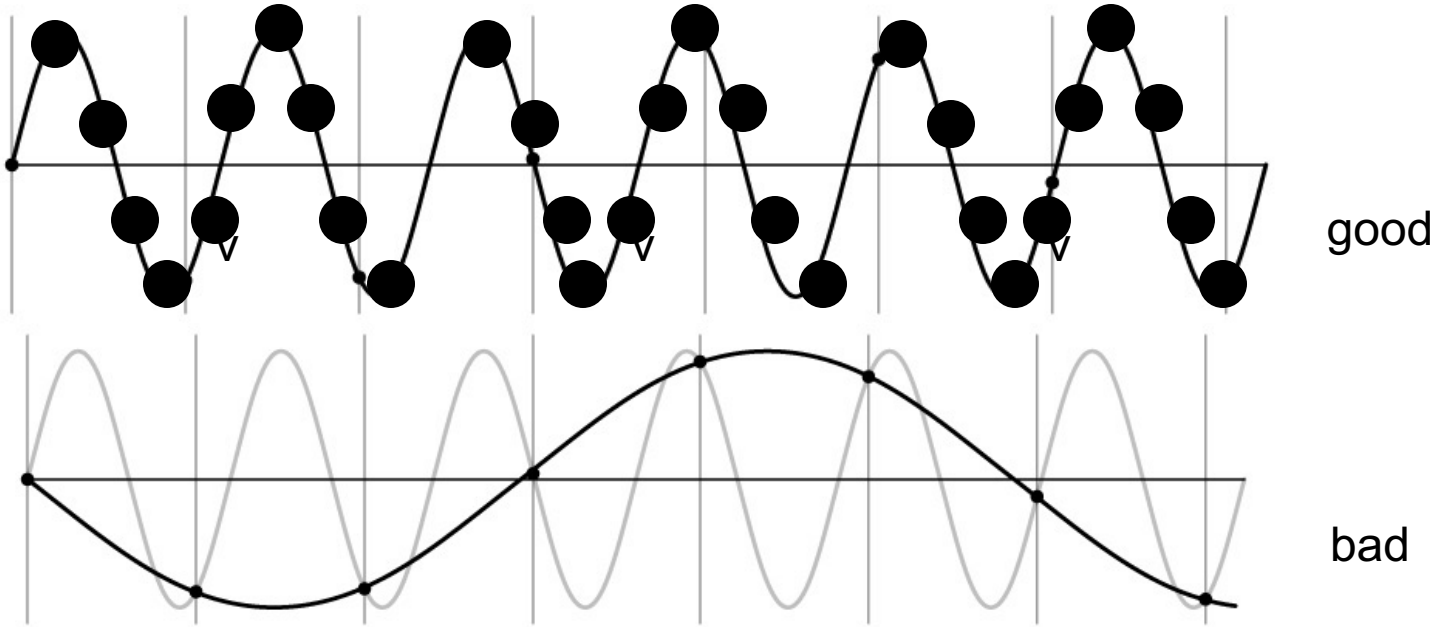
Aliasing



Source: Forsyth and Ponce

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



Slide from Derek Hoiem.

Anti-aliasing

Solutions:

- Sample more often

- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it's better than aliasing
  - Apply a smoothing filter

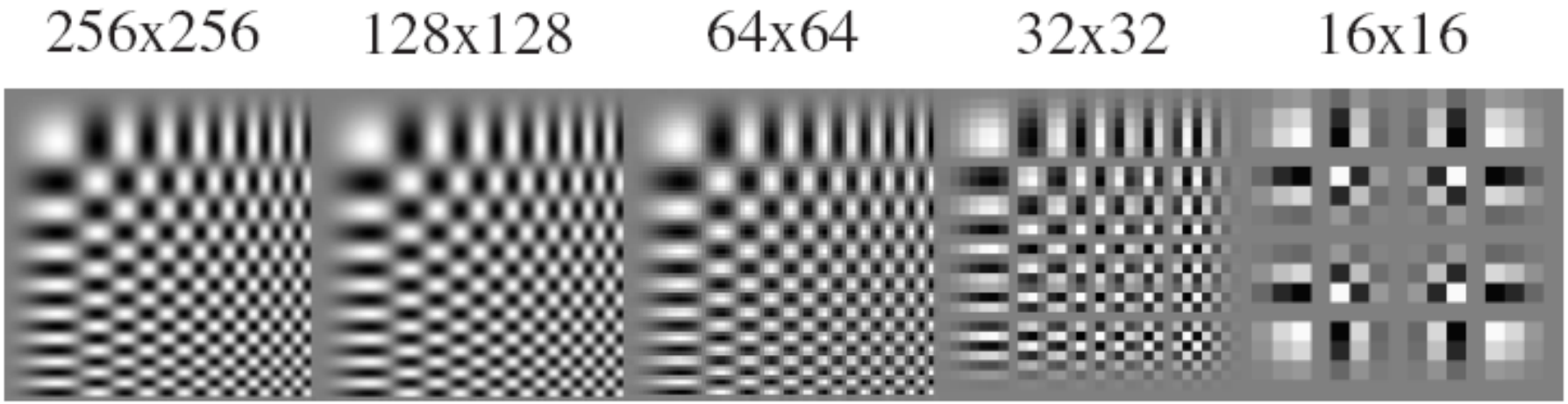
Slide from Derek Hoiem.

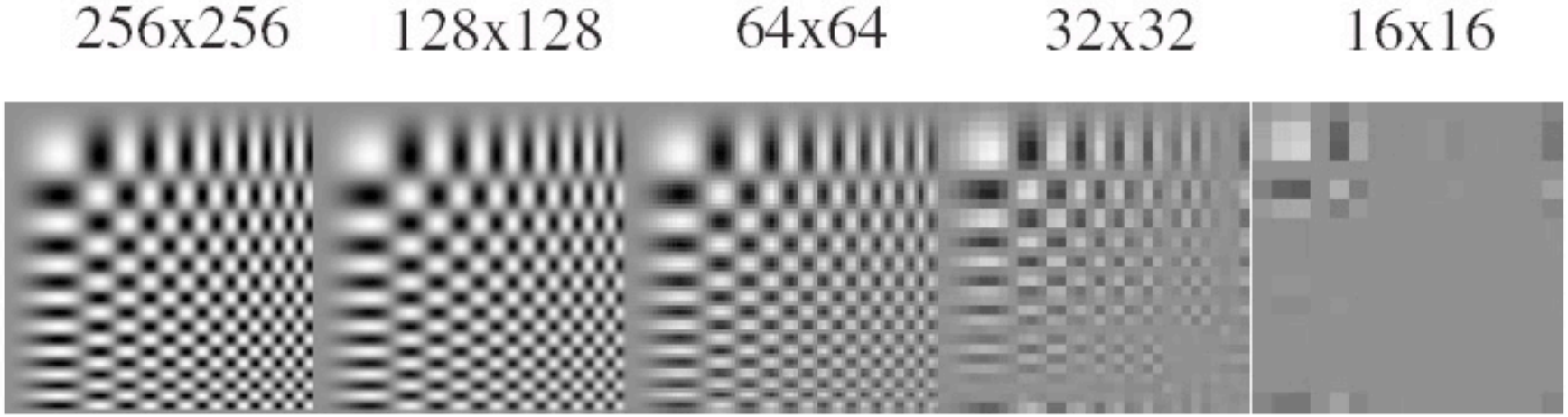
Algorithm for downsampling by factor of 2

1. Start with image(h, w)
2. Apply low-pass filter
3. Sample every other pixel

Slide from Derek Hoiem.

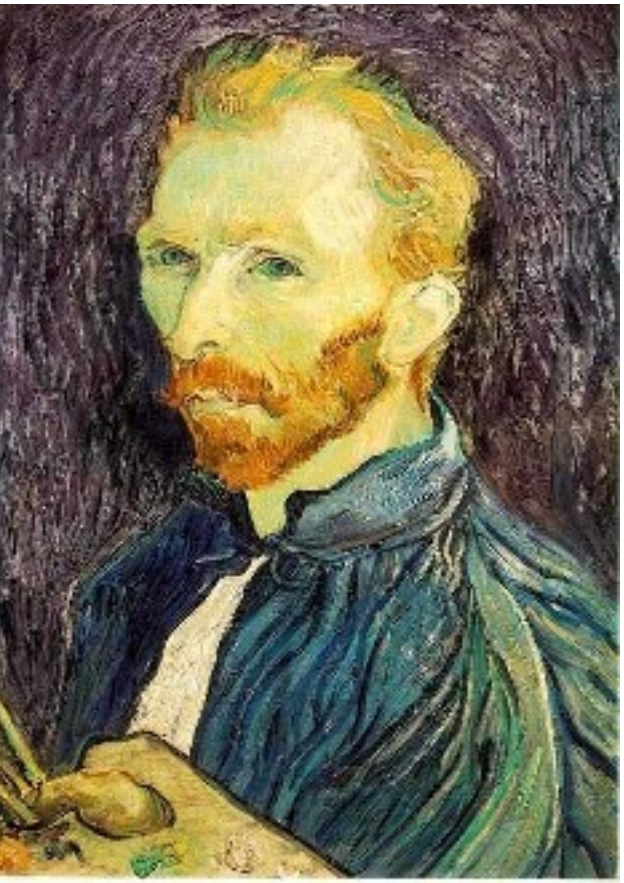
Anti-aliasing





Forsyth and Ponce 2002

Subsampling without pre-filtering



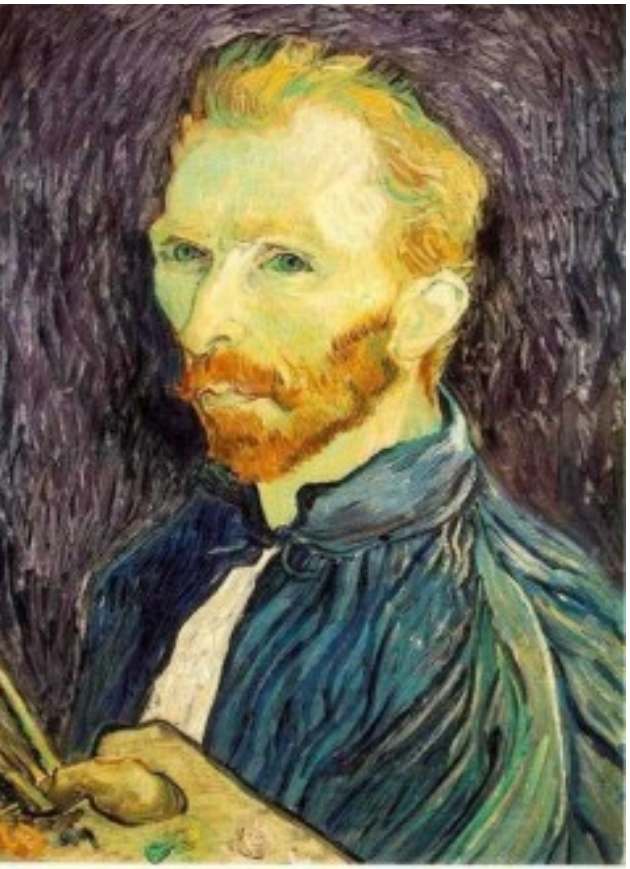
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide by Steve Seitz

Subsampling with Gaussian pre-filtering



Gaussian 1/2

G 1/4

G 1/8

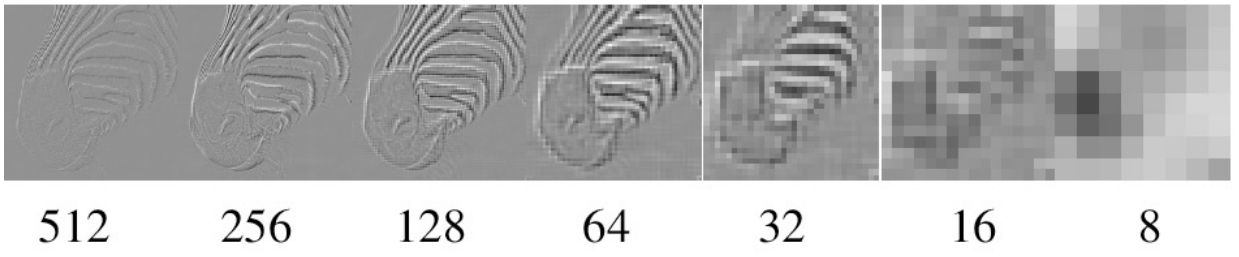
Slide by Steve Seitz

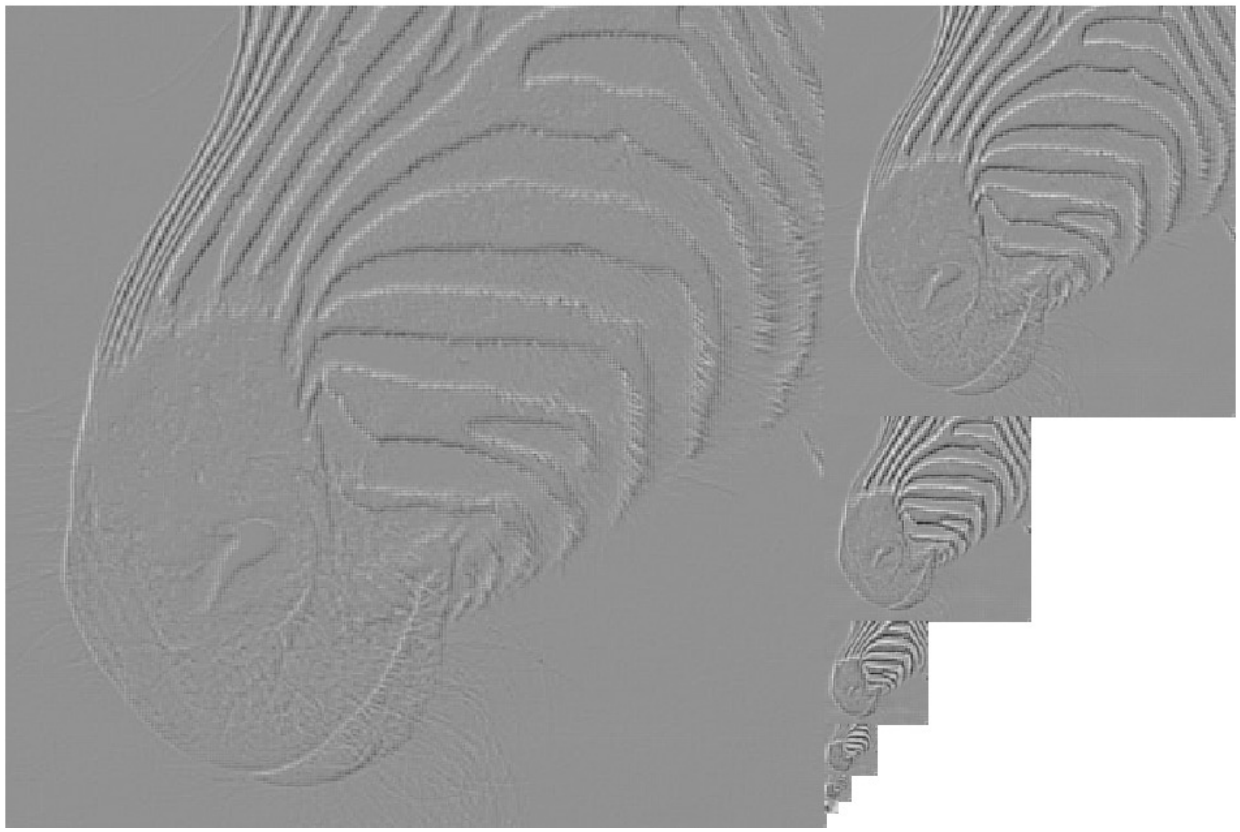
Gaussian pyramid (Repeated blurring and sampling)

512 256 128 64 32 16 8

Source: Forsyth

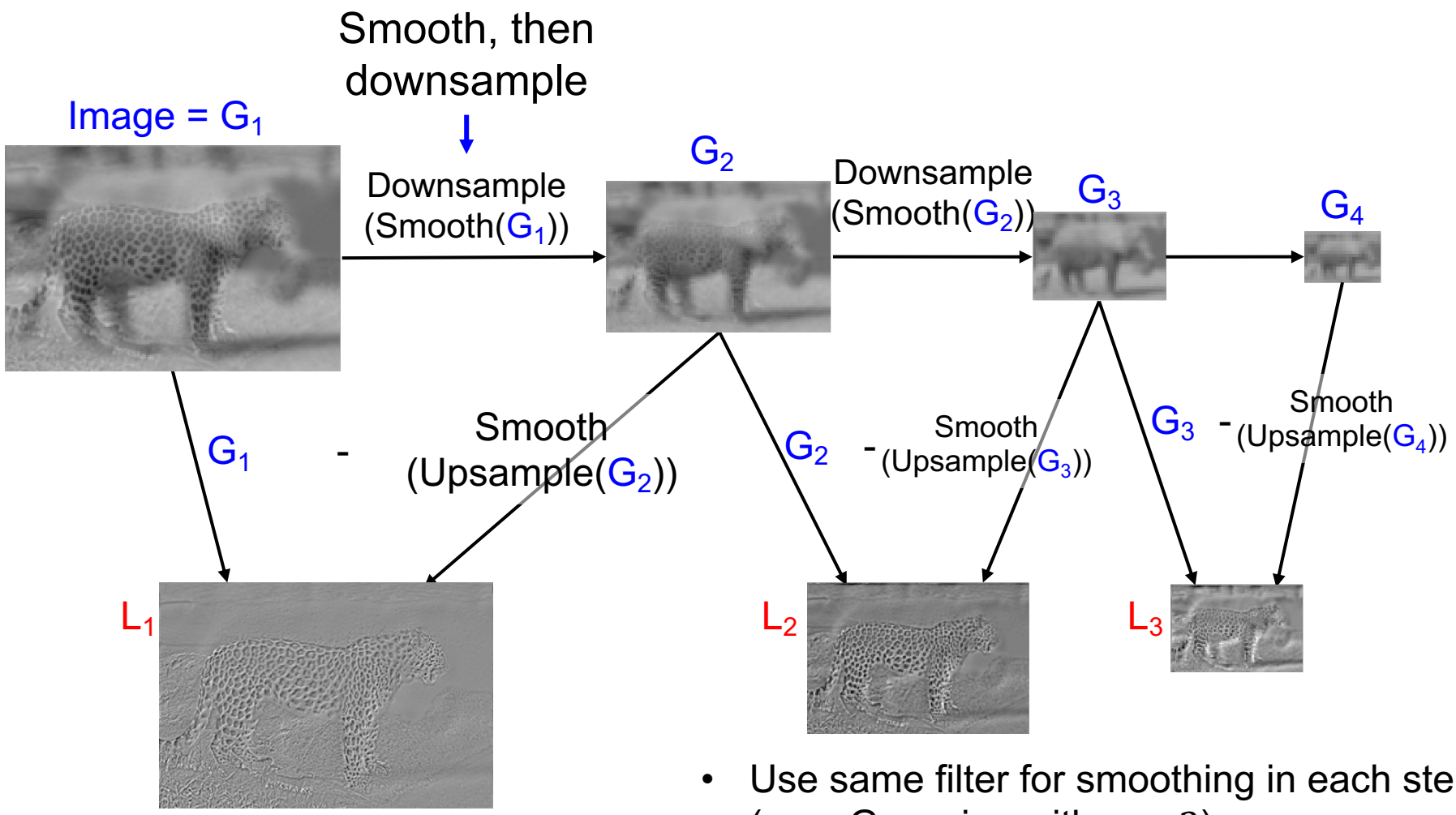
Laplacian pyramid





Source: Forsyth

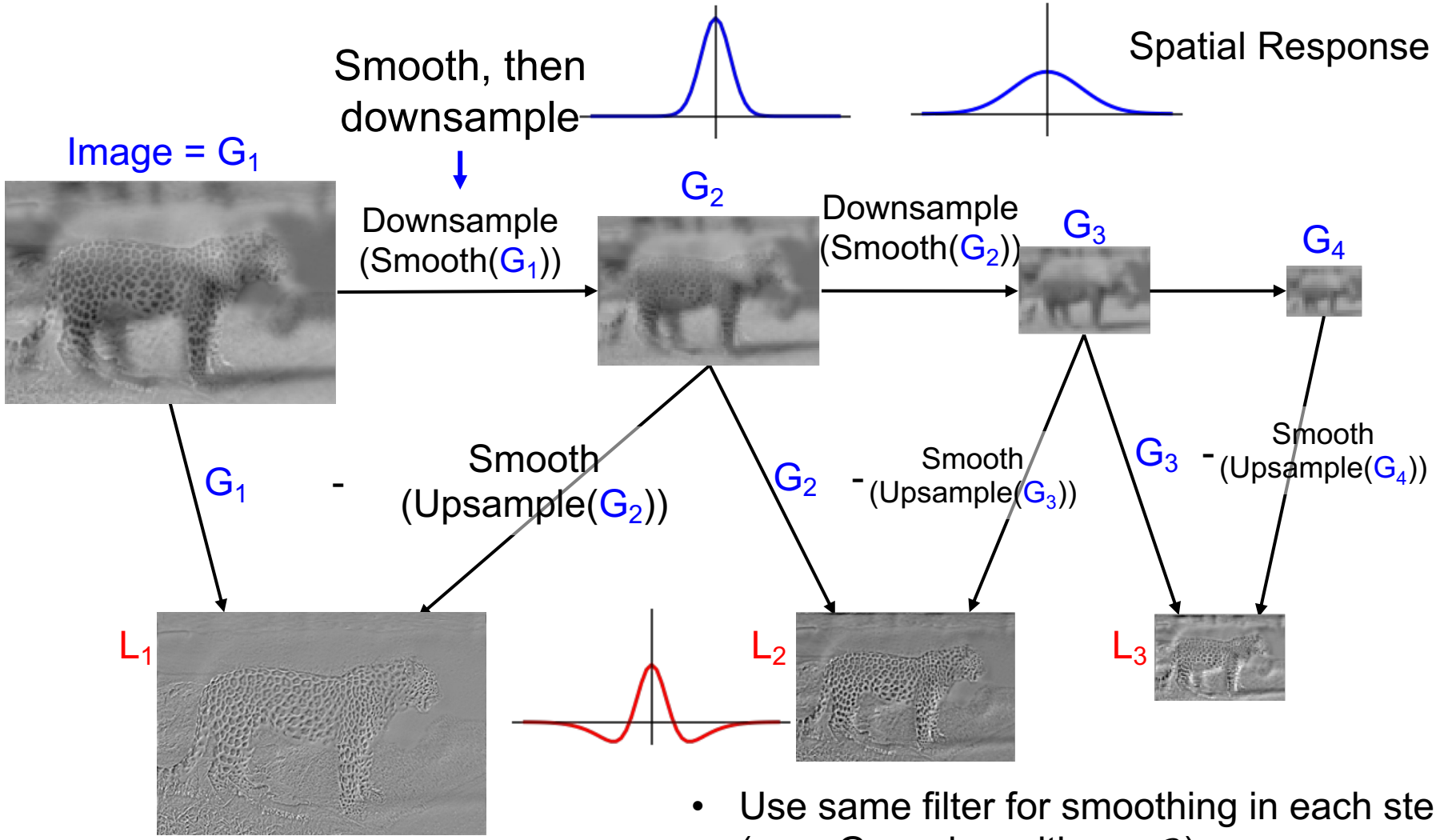
Creating the Difference of Gaussian Pyramid



- Use same filter for smoothing in each step (e.g., Gaussian with $\sigma = 2$)
- Downsample/upsample with "nearest" interpolation

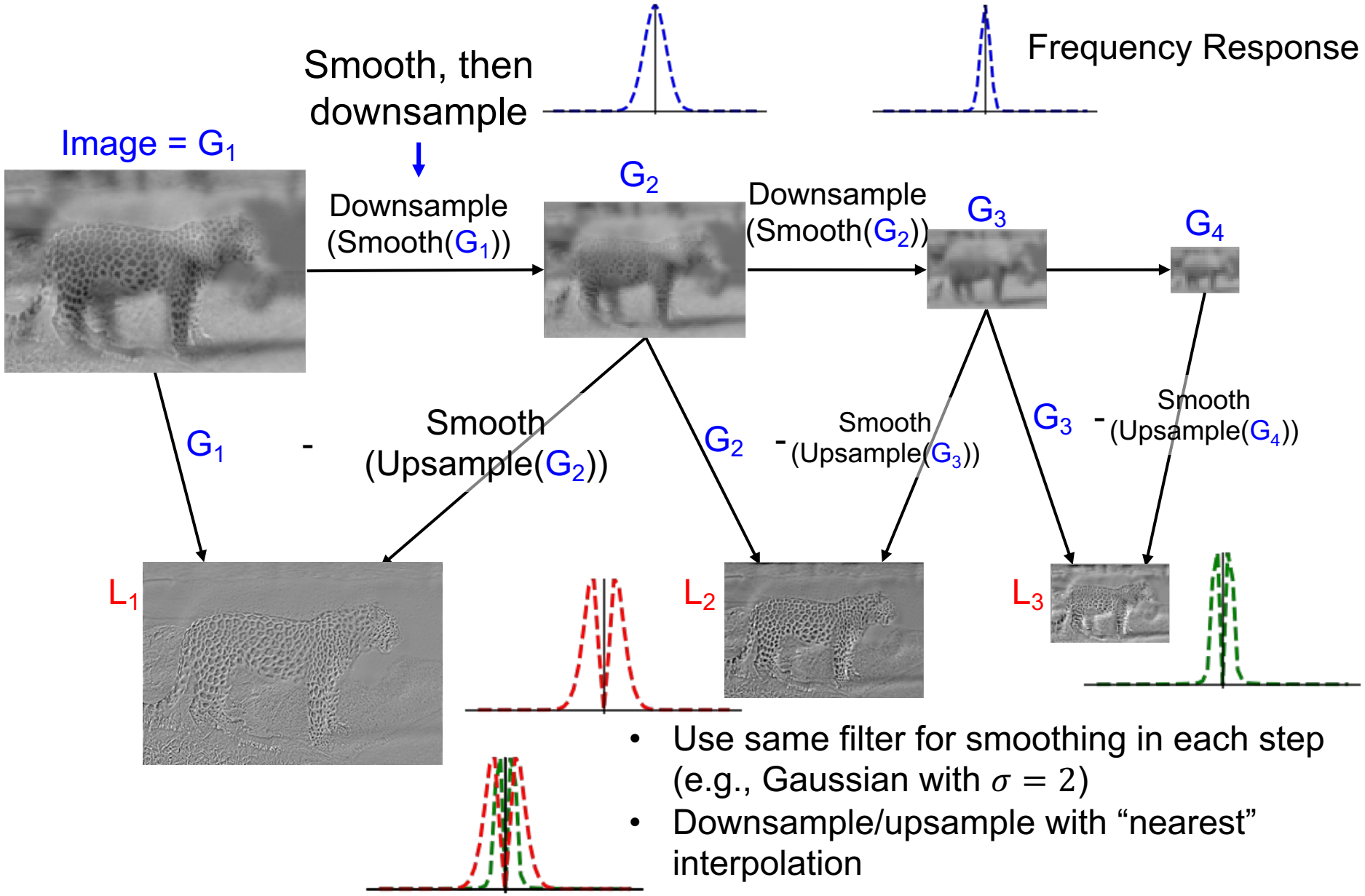
Leopard, Elephant image from Olivia and Torralba

Creating the Difference of Gaussian Pyramid



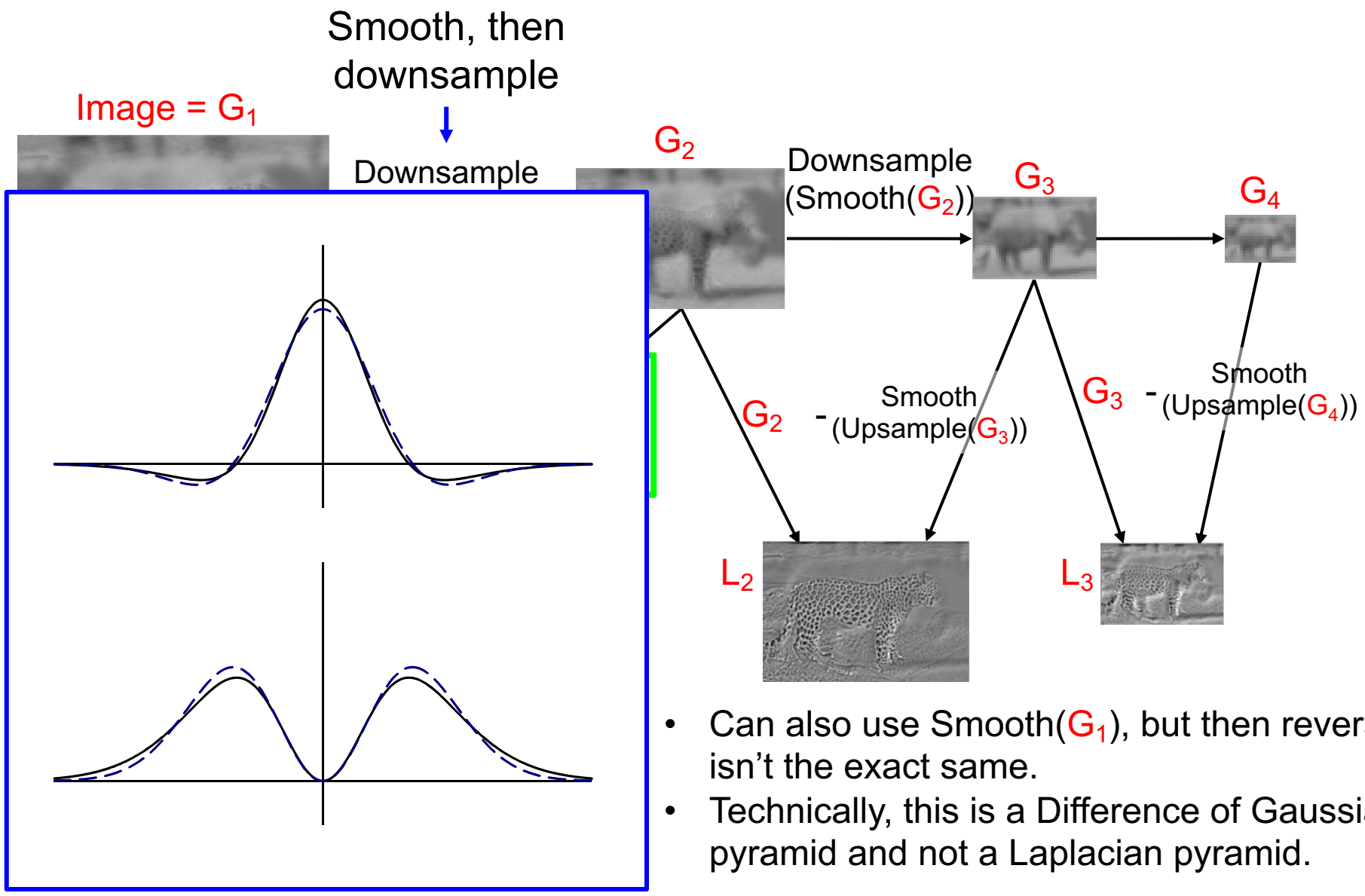
- Use same filter for smoothing in each step (e.g., Gaussian with $\sigma = 2$)
- Downsample/upsample with "nearest" interpolation

Creating the Difference of Gaussian Pyramid



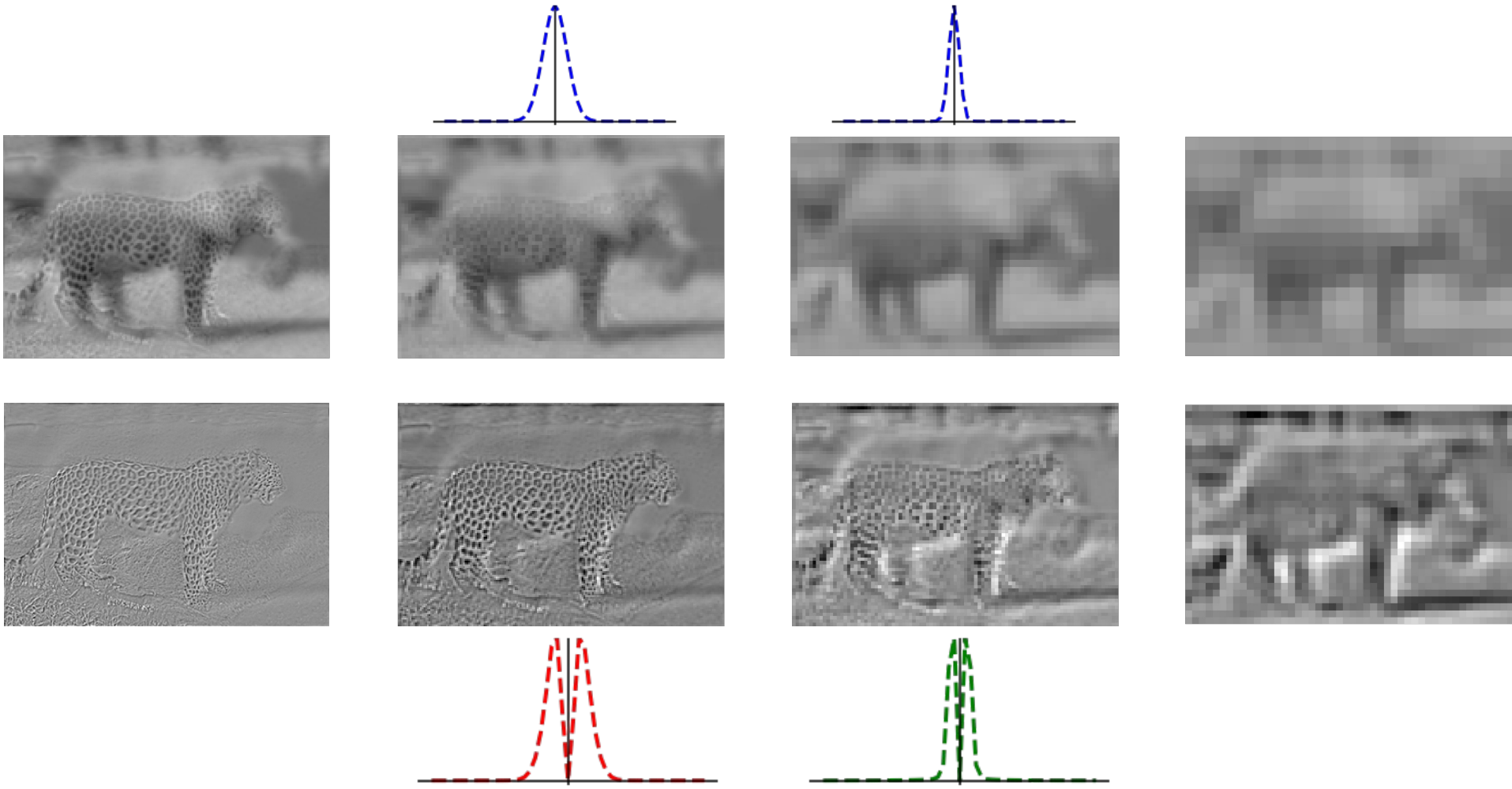
- Use same filter for smoothing in each step (e.g., Gaussian with $\sigma = 2$)
- Downsample/upsample with "nearest" interpolation

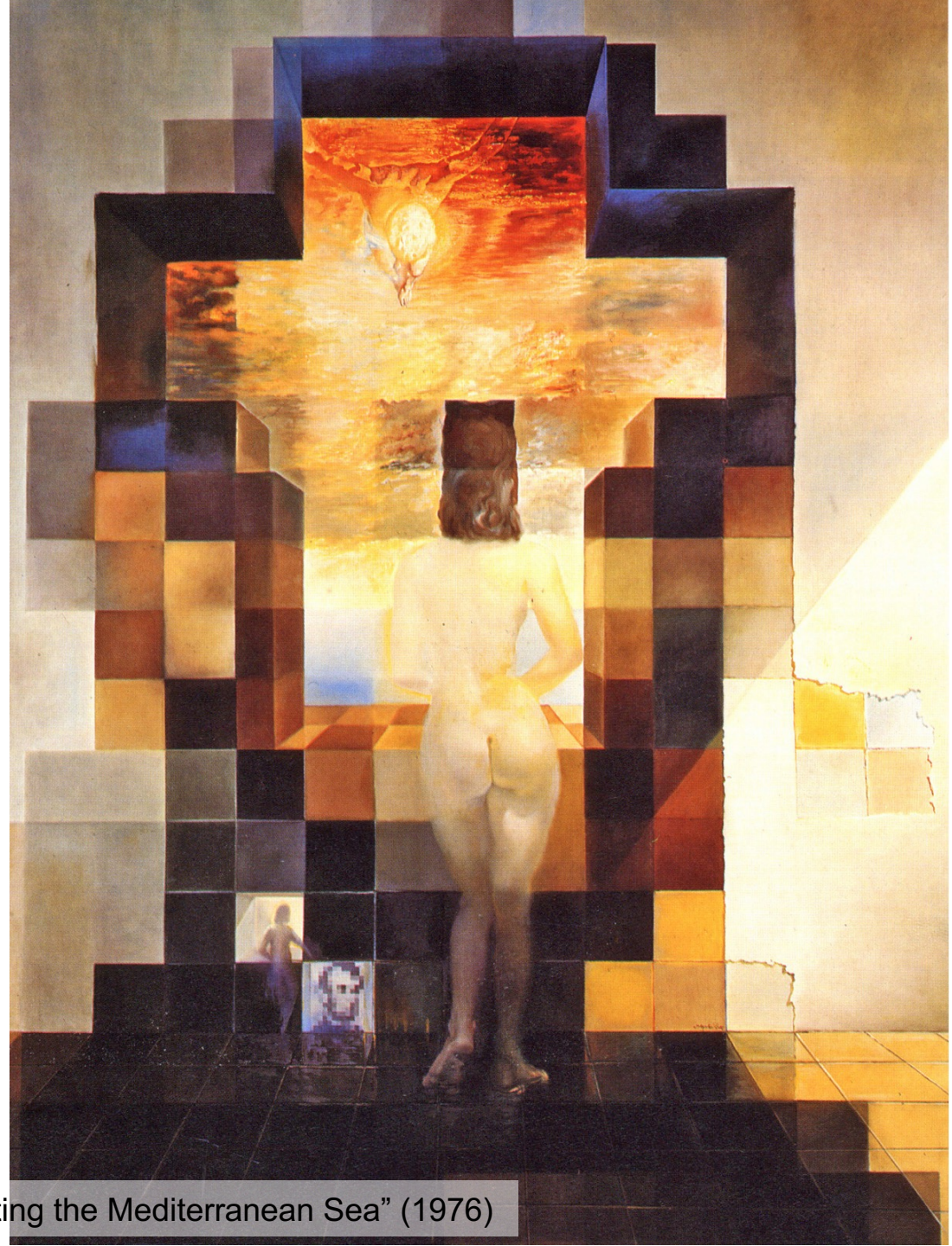
Creating the Difference of Gaussian Pyramid



- Can also use Smooth(G_1), but then reverse isn't the exact same.
- Technically, this is a Difference of Gaussian pyramid and not a Laplacian pyramid.

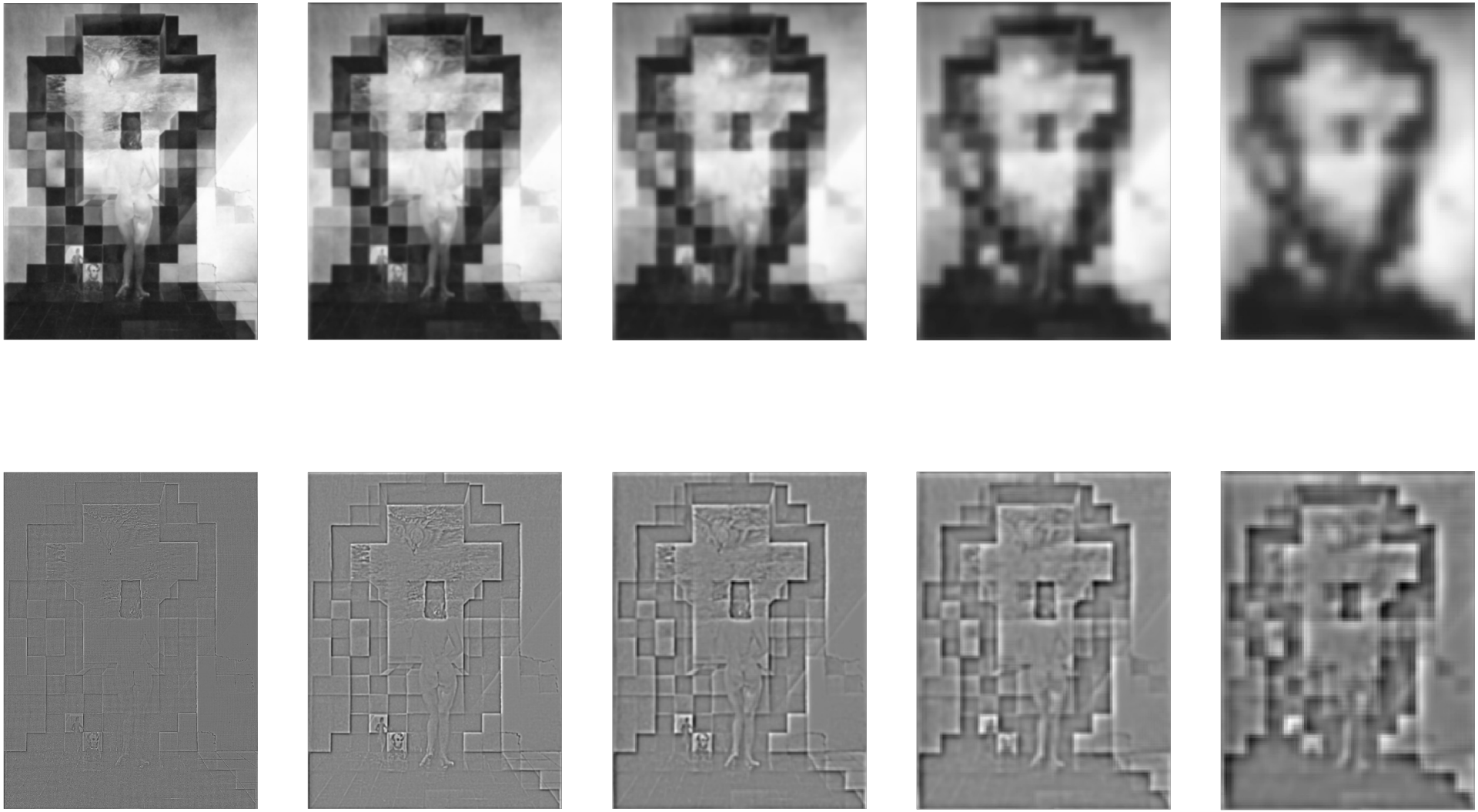
Images in a Difference of Gaussian Pyramid





Dali: “Gala Contemplating the Mediterranean Sea” (1976)

Images in a Difference of Gaussian Pyramid



Dali: “Gala Contemplating the Mediterranean Sea” (1976)

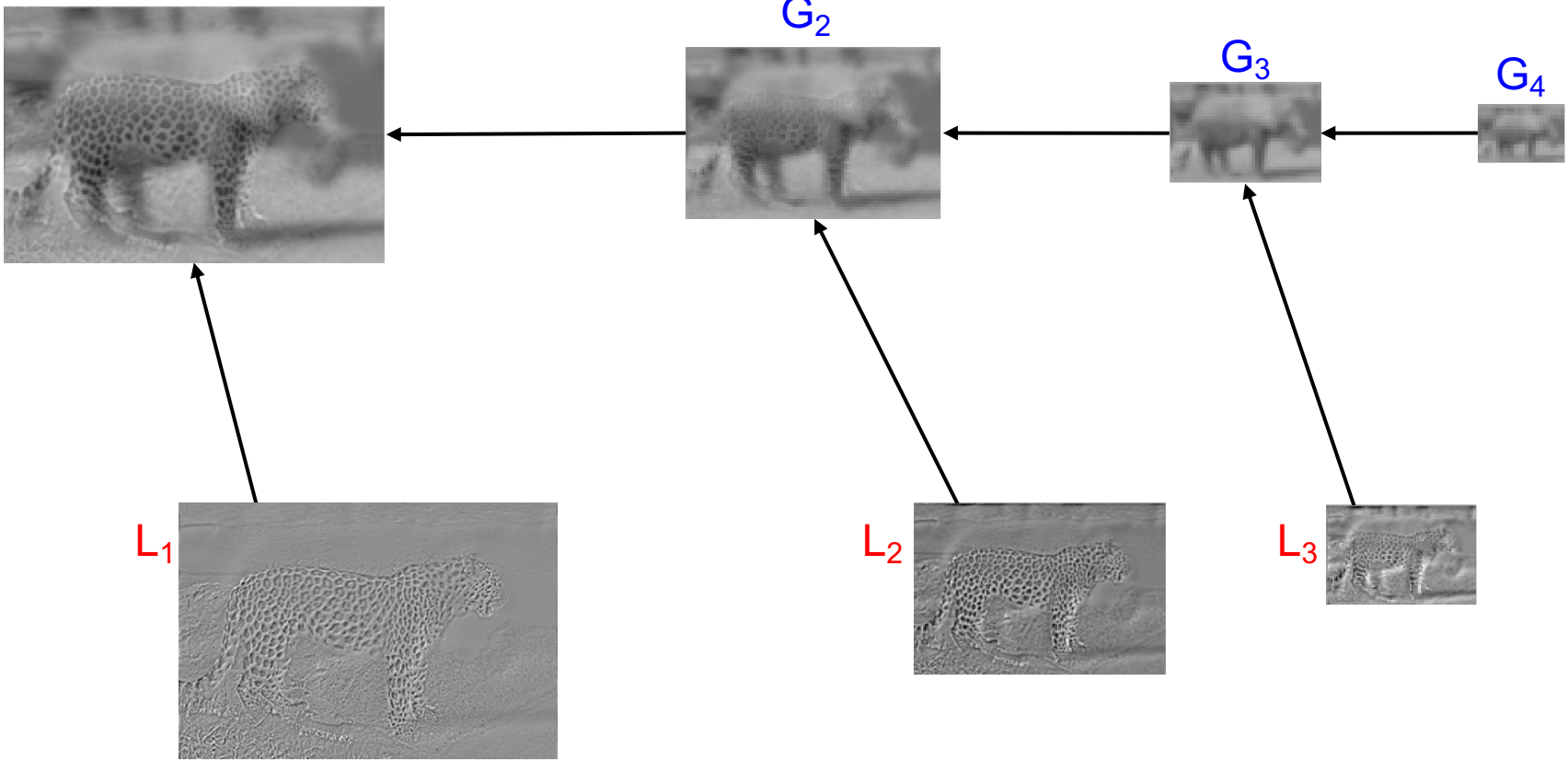
Reconstructing from Diff of Gauss Pyramid

$G_1 = L_1 +$ Smooth(Upsample(G_2))

Image = G_1

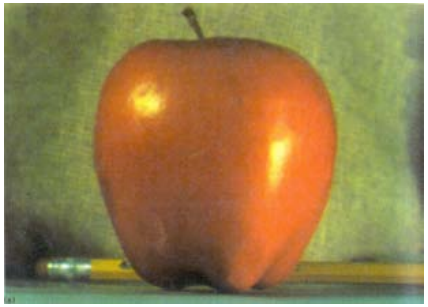
$G_2 = L_2 +$ Smooth(Upsample(G_3))

$G_3 = L_3 +$ Smooth(Upsample(G_4))



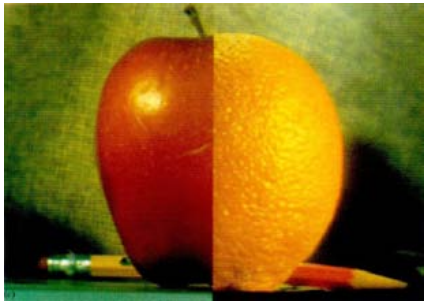
- Use same filter for smoothing as in deconstruction
- Upsample with “nearest” interpolation
- Reconstruction will be nearly lossless

Application: Image Blending

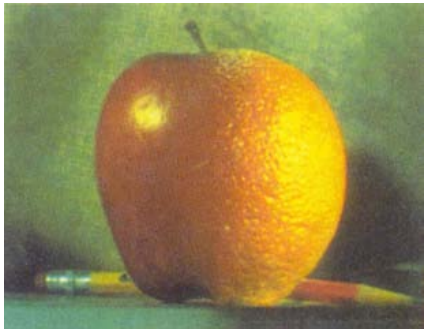


(a)

(b)

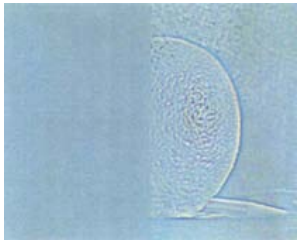


(c)



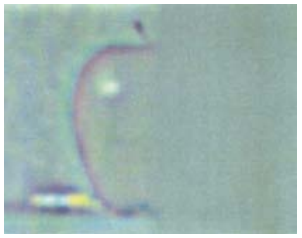
(d)

(a)

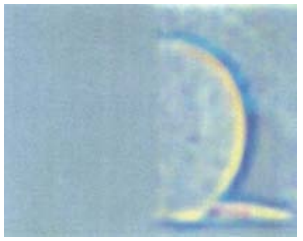


(b)

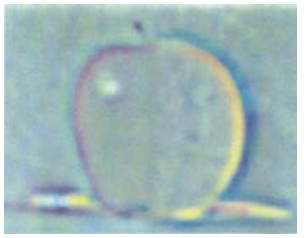
(c)



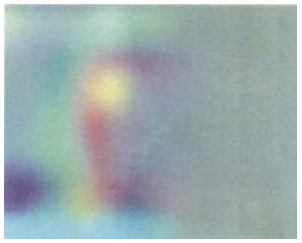
(d)



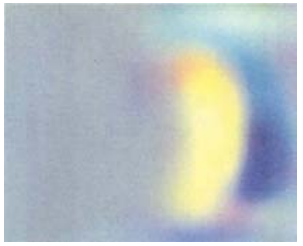
(e)



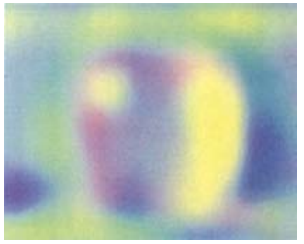
(f)



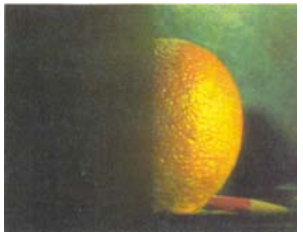
(g)



(h)

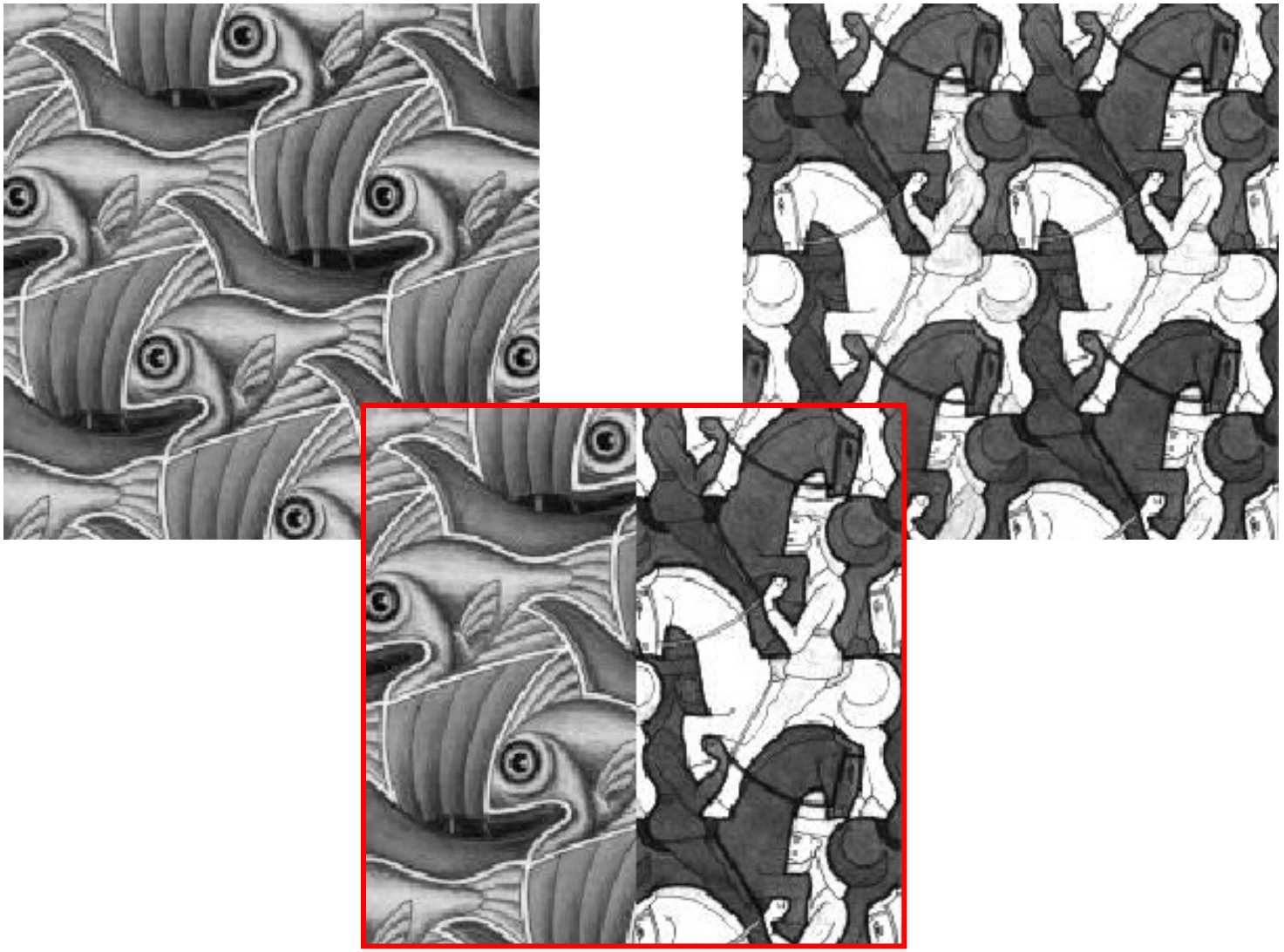


(i)

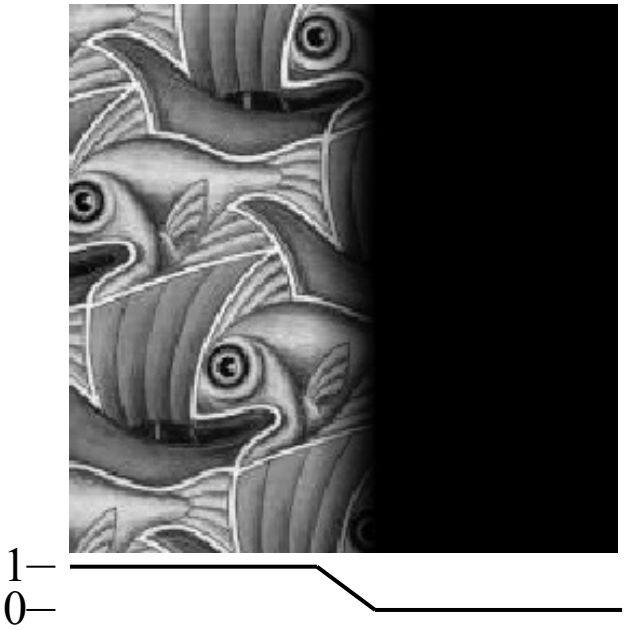


Laplacian pyramid blending (Burt and Adelson 1983b)

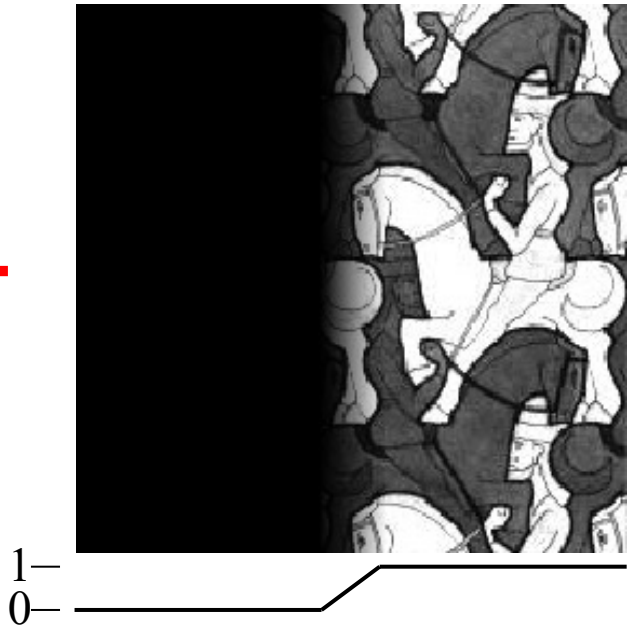
Blending



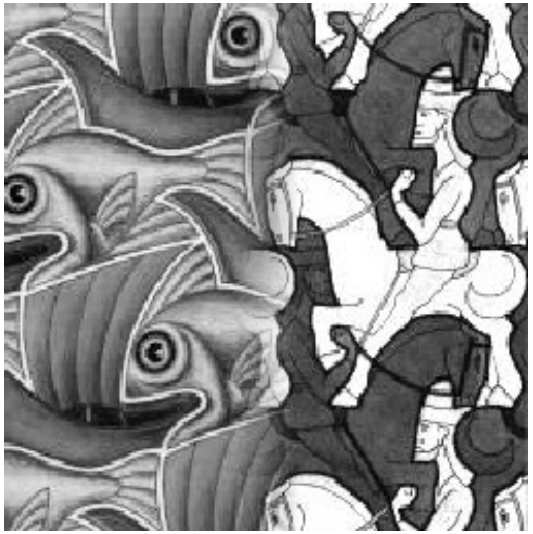
Alpha Blending / Feathering



+

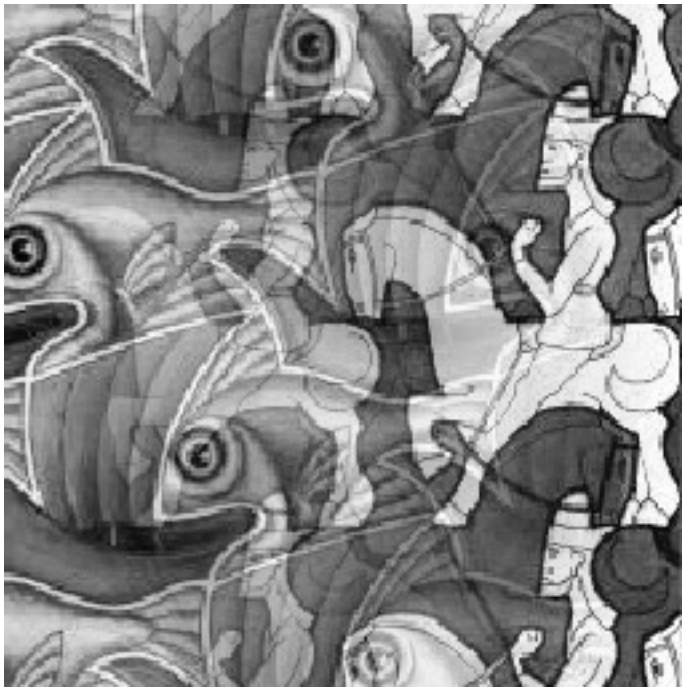


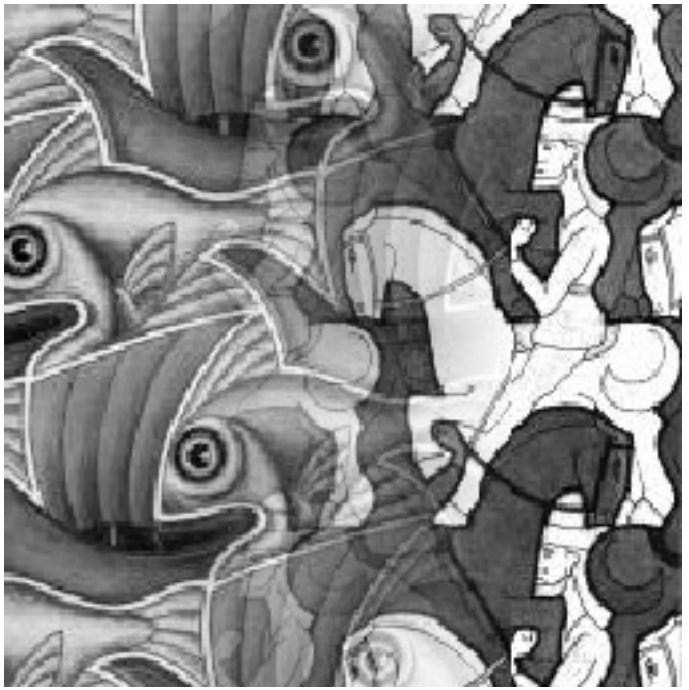
=

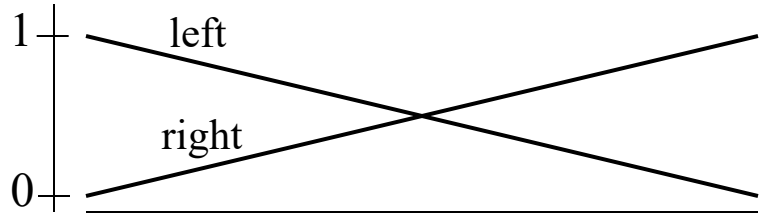


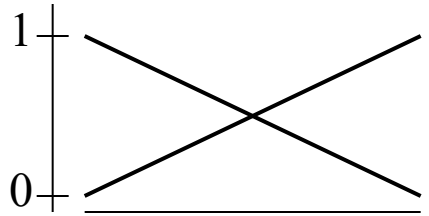
$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right}$$

Affect of Window Size

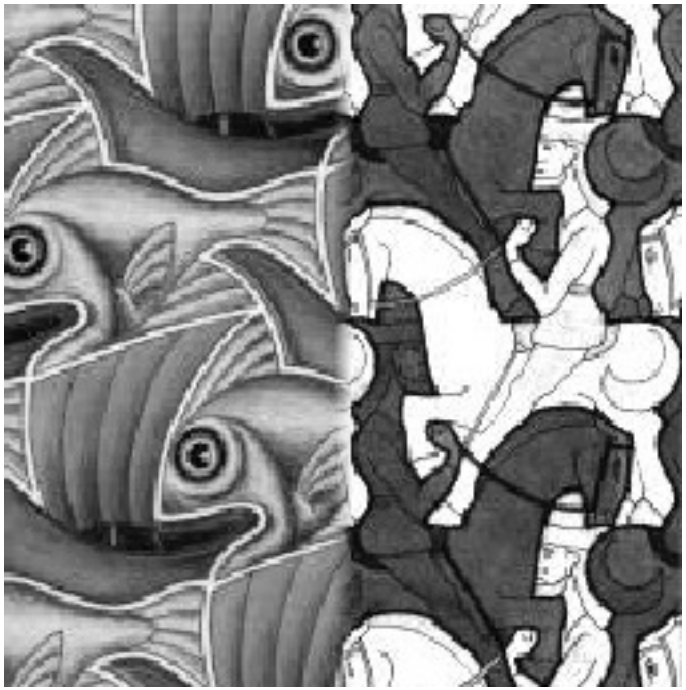


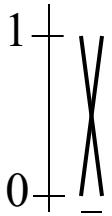


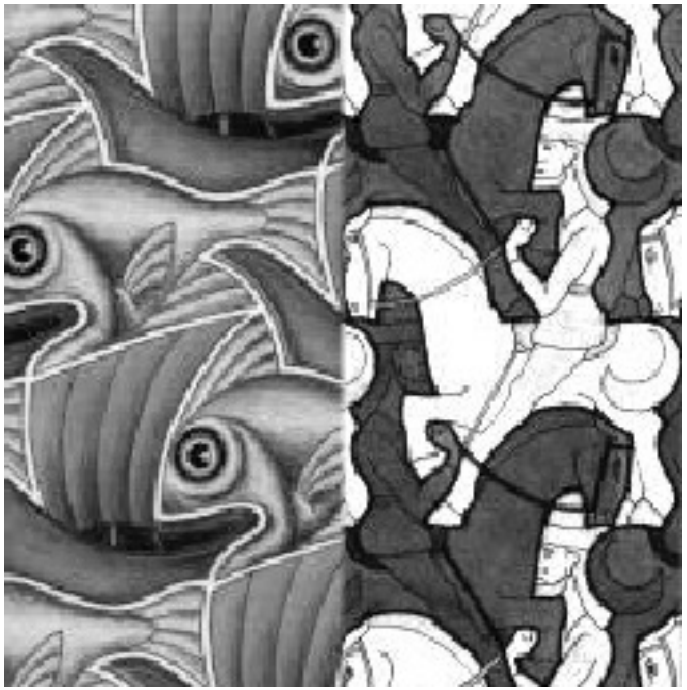


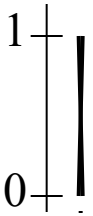


Affect of Window Size

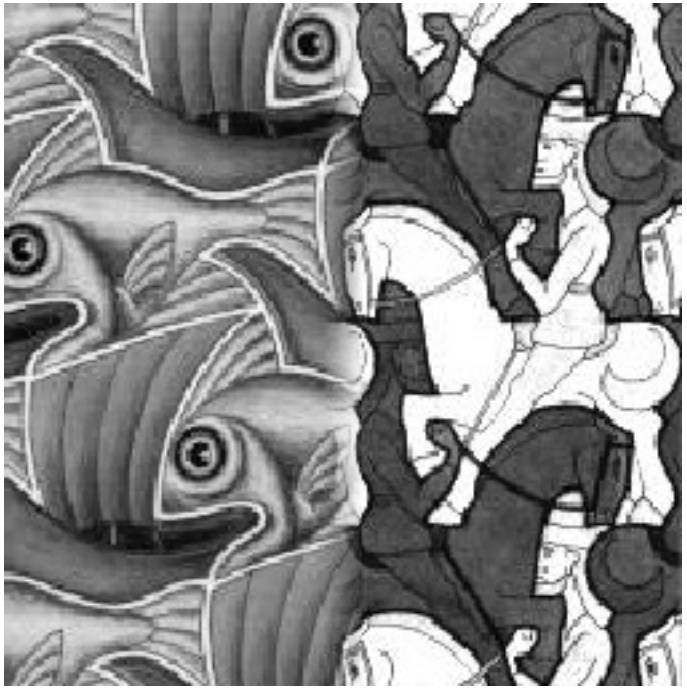


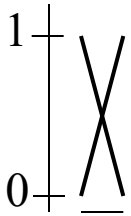






Good Window Size





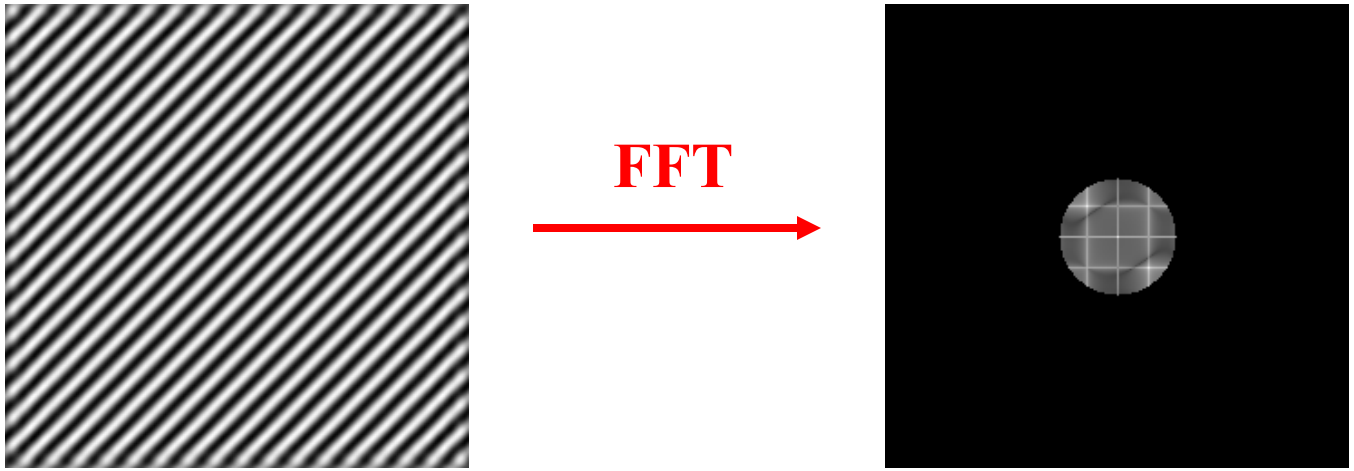
“Optimal” Window: smooth but not ghosted

What is the Optimal Window?

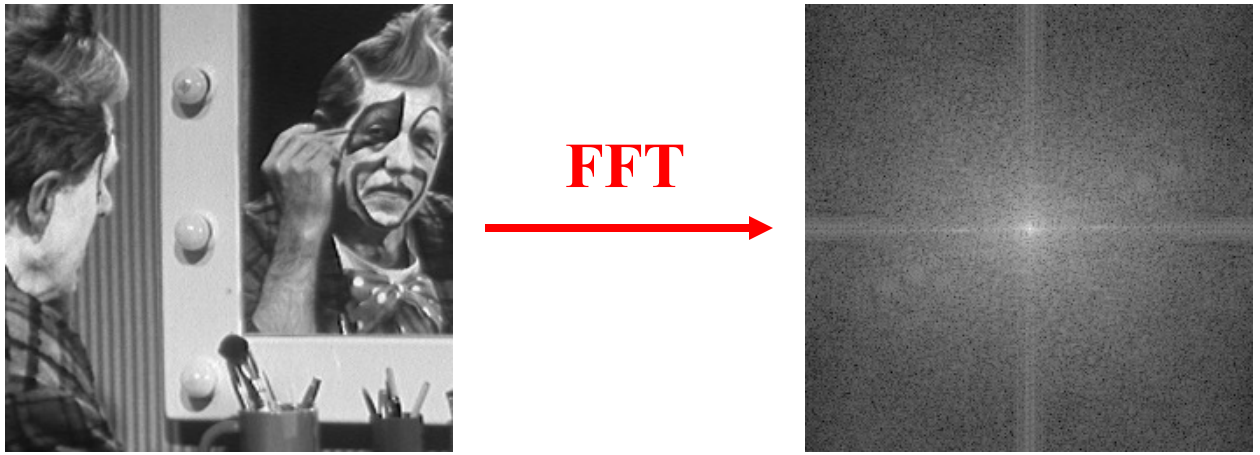
- To avoid seams
  - window = size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature

Natural to cast this in the *Fourier domain*

- largest frequency <= 2*size of smallest frequency
- image frequency content should occupy one “octave” (power of two)



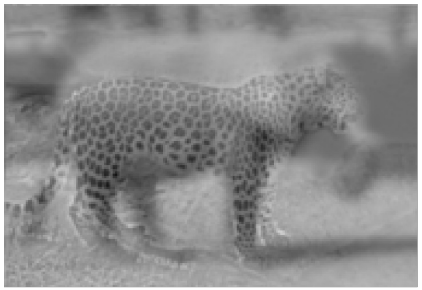
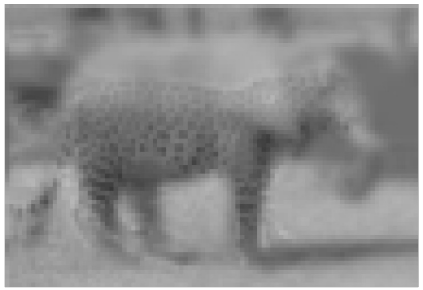
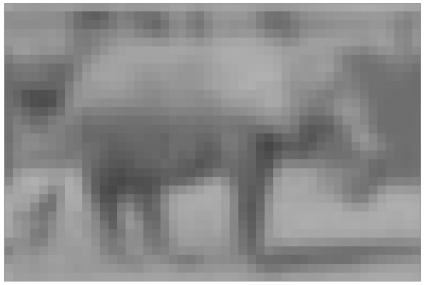
What if the Frequency Spread is Wide

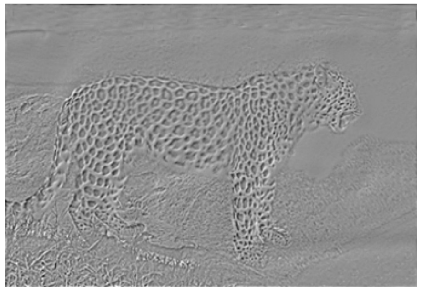


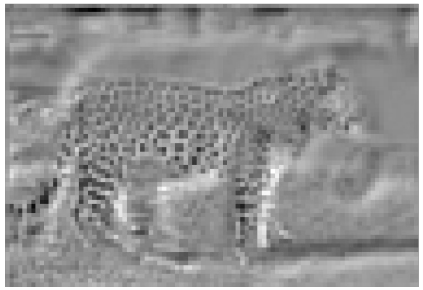
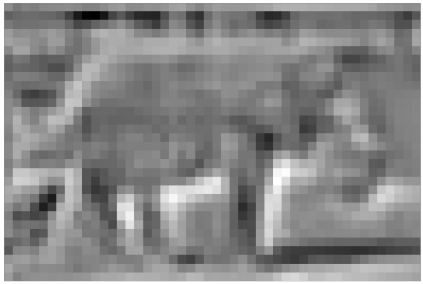
- Idea (Burt and Adelson)
  - Compute $F_{left} = FFT(I_{left})$, $F_{right} = FFT(I_{right})$
  - Decompose Fourier image into octaves (bands)
    - $F_{left} = F_{left}^{1} + F_{left}^{2} + \ldots$
  - Feather corresponding octaves F_{left}^{i} with F_{right}^{i}
    - Can compute inverse FFT and feather in spatial domain
  - Sum feathered octave images in frequency domain
- Better implemented in *spatial domain*

Octaves in the Spatial Domain

Lowpass Images

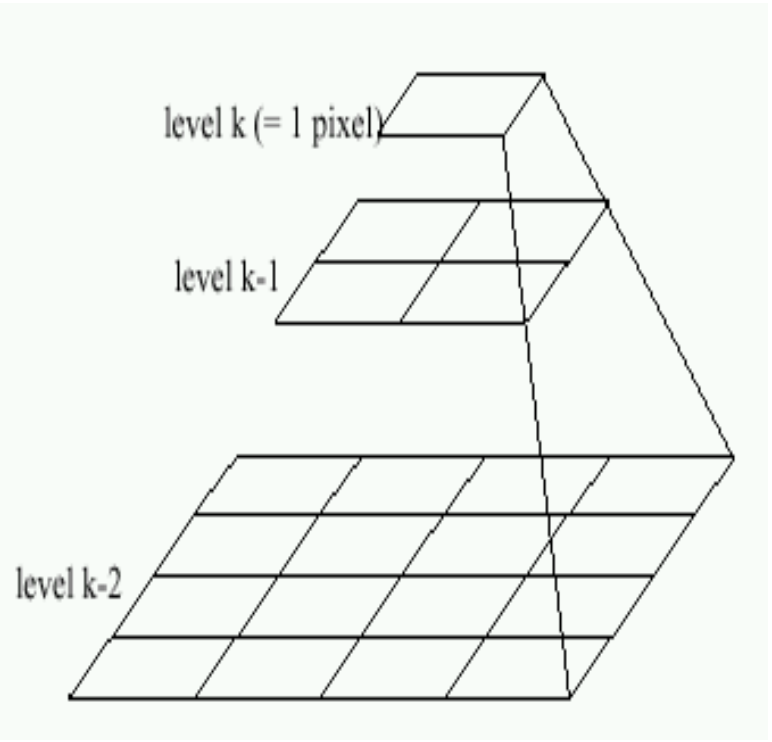




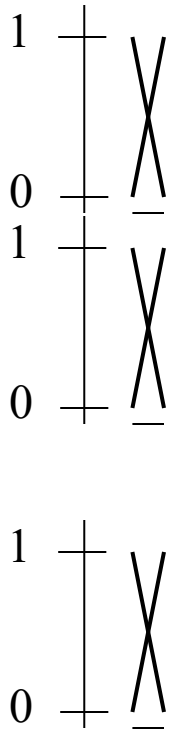



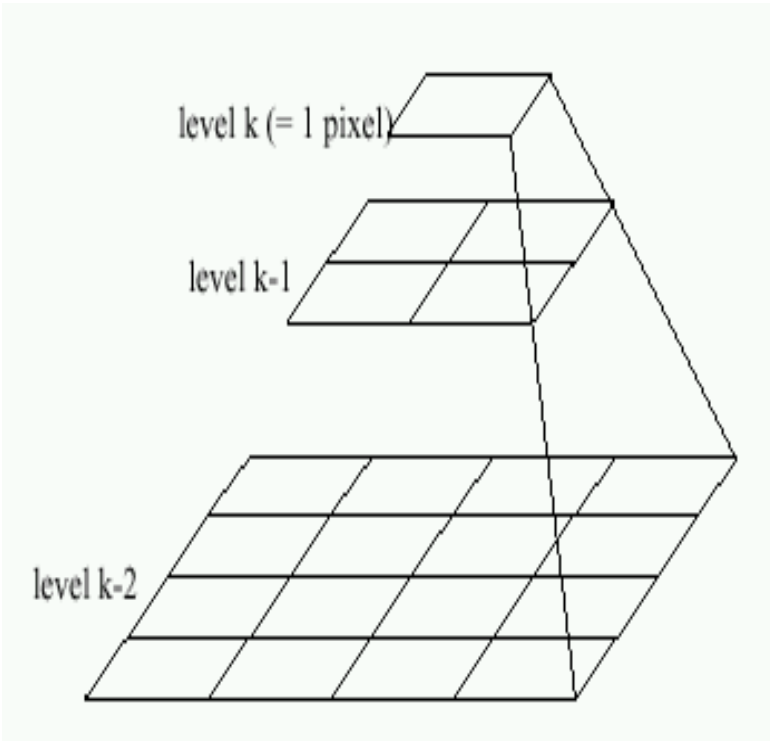



- Bandpass Images

Pyramid Blending





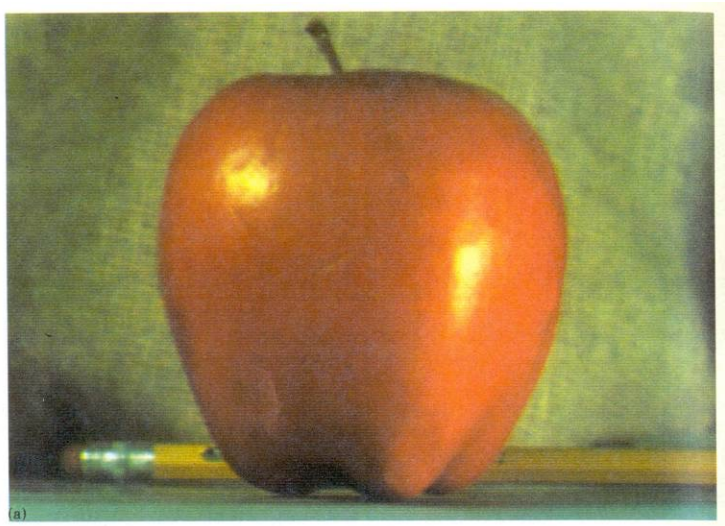


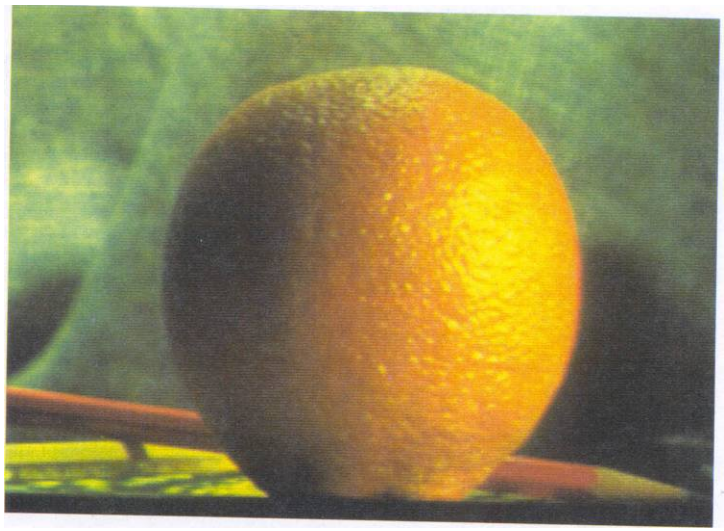
Left pyramid

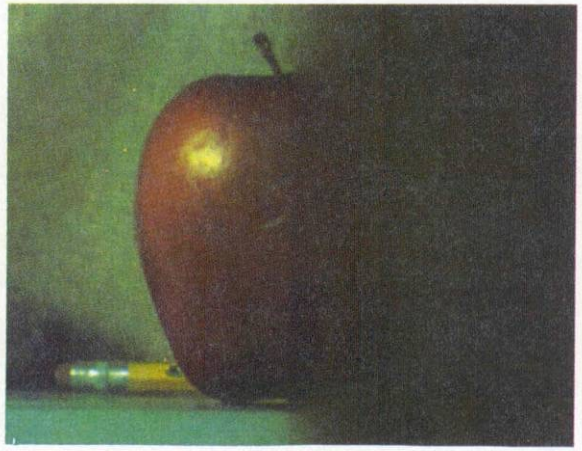
blend

Right pyramid

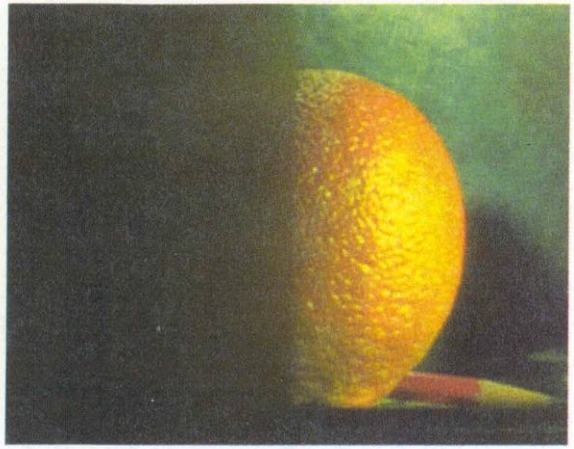
Pyramid Blending



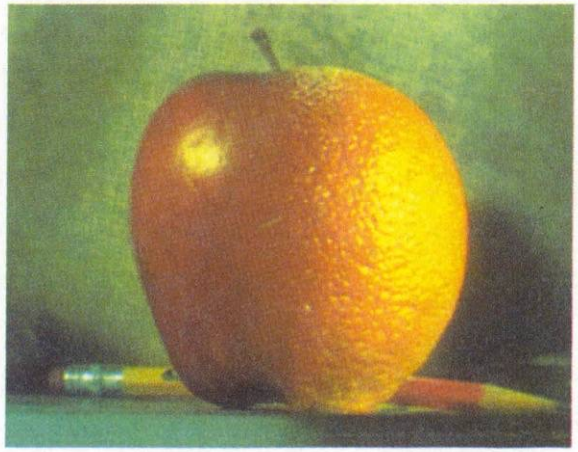




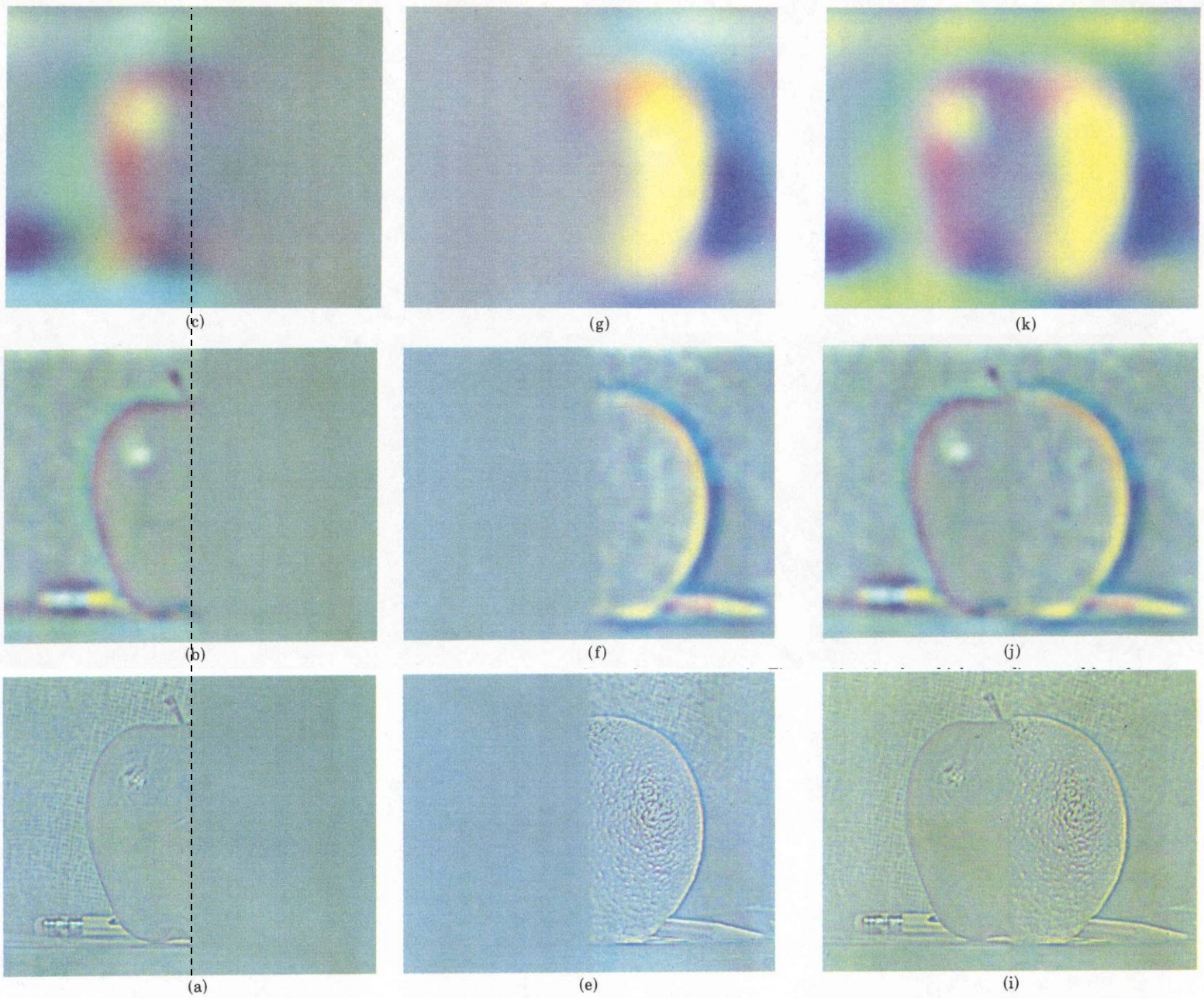
(d)



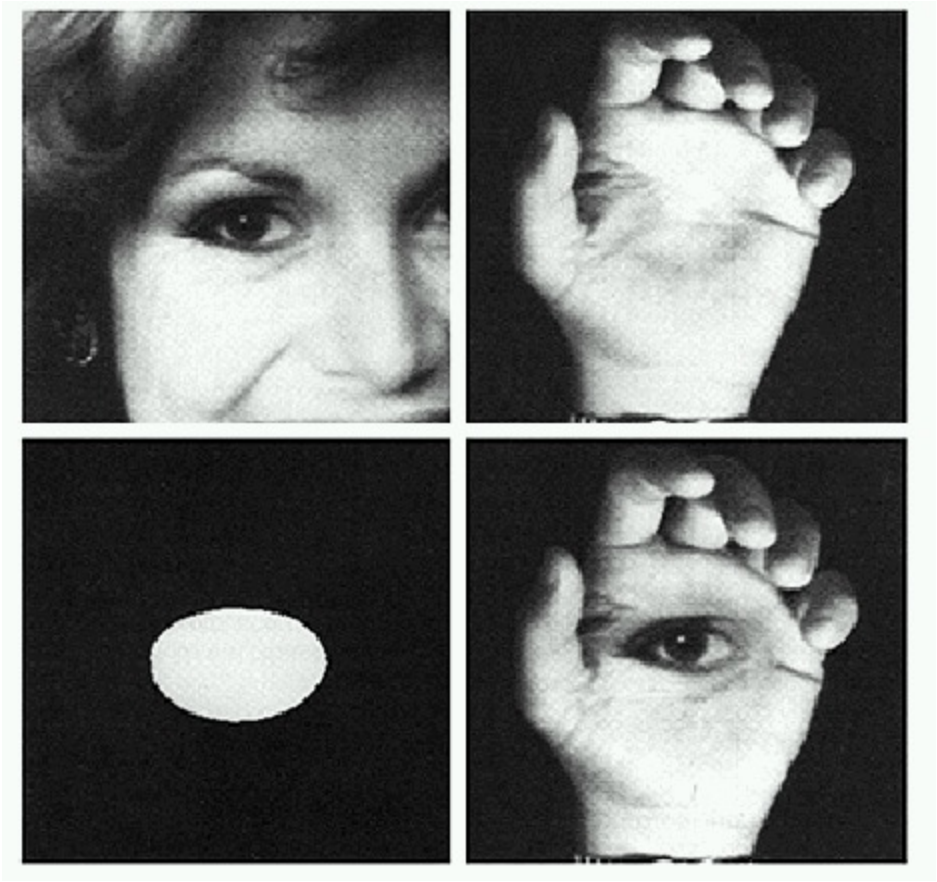
(h)



(l)


laplacian level 4
laplacian level 2
laplacian level 0
(c)
(g)
(k)
(b)
(f)
(j)
(a)
(e)
(i)
left pyramid
right pyramid
blended pyramid

Blending Regions



Laplacian Pyramid: Blending

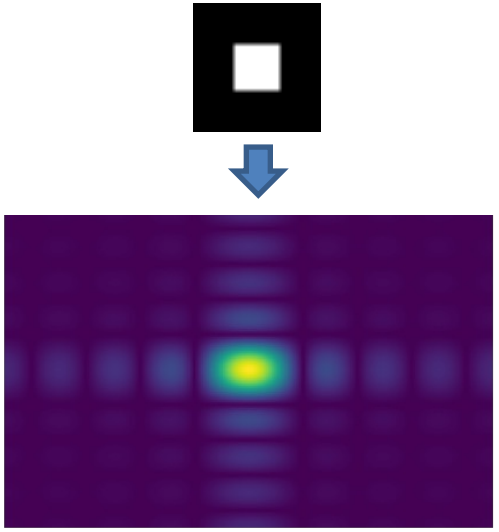
- General Approach:
  1. Build Laplacian pyramids *LA* and *LB* from images *A* and *B*
  2. Build a Gaussian pyramid *GR* from selected region *R*
  3. Form a combined pyramid *LS* from *LA* and *LB* using nodes of *GR* as weights:
     - *LS(i,j) = GR(I,j,)*LA(I,j) + (1-GR(I,j))*LB(I,j)*
  4. Collapse the *LS* pyramid to get the final blended image

Major uses of image pyramids

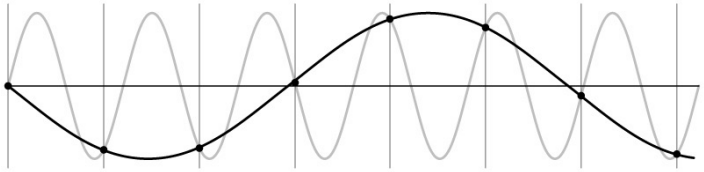
- Compression
- Object detection
  - Scale search
  - Features
- Detecting stable interest points
- Registration
  - Course-to-fine

Recap

- Sometimes it makes sense to think of filtering in the frequency domain
  - Fourier analysis



- Sampling and Aliasing



- Image Pyramids