CS 558: Computer Vision
2nd Set of Notes

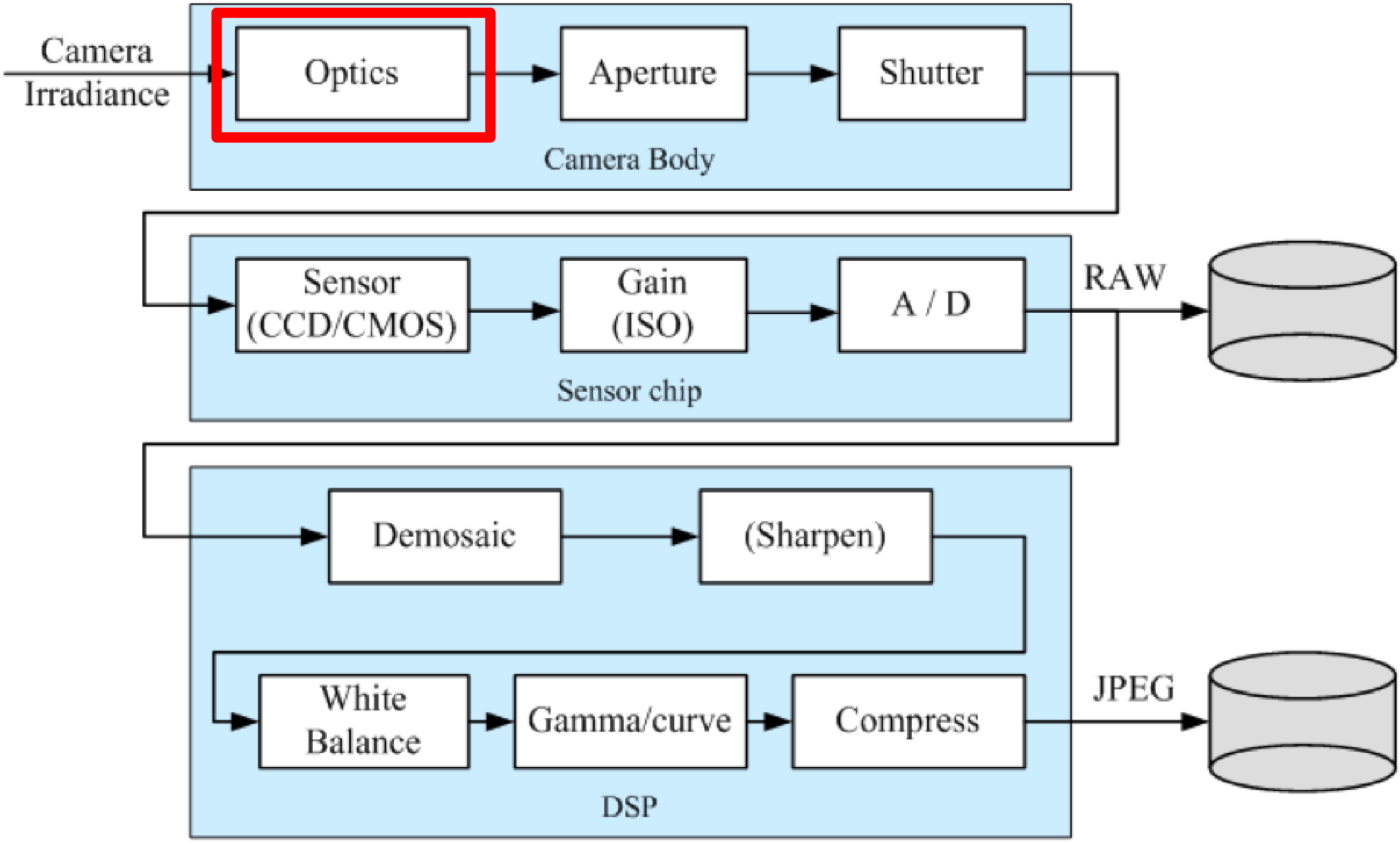
Instructor: Philippos Mordohai
Webpage: www.cs.stevens.edu/~mordohai
E-mail: Philippos.Mordohai@stevens.edu
Office: Lieb 215

Overview

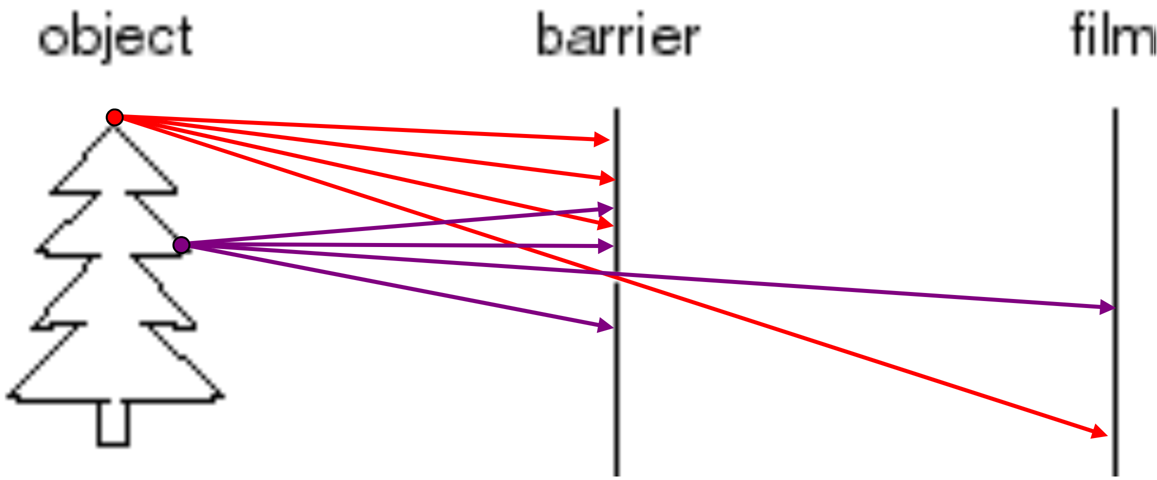
- Brief summary of optics and aperture
- Camera body: shutter
- The sensor
  - Based on slides by G. Doretto

- Light and Shading
- Linear filters
  - Based on slides by D. Hoiem

Camera Body: Optics



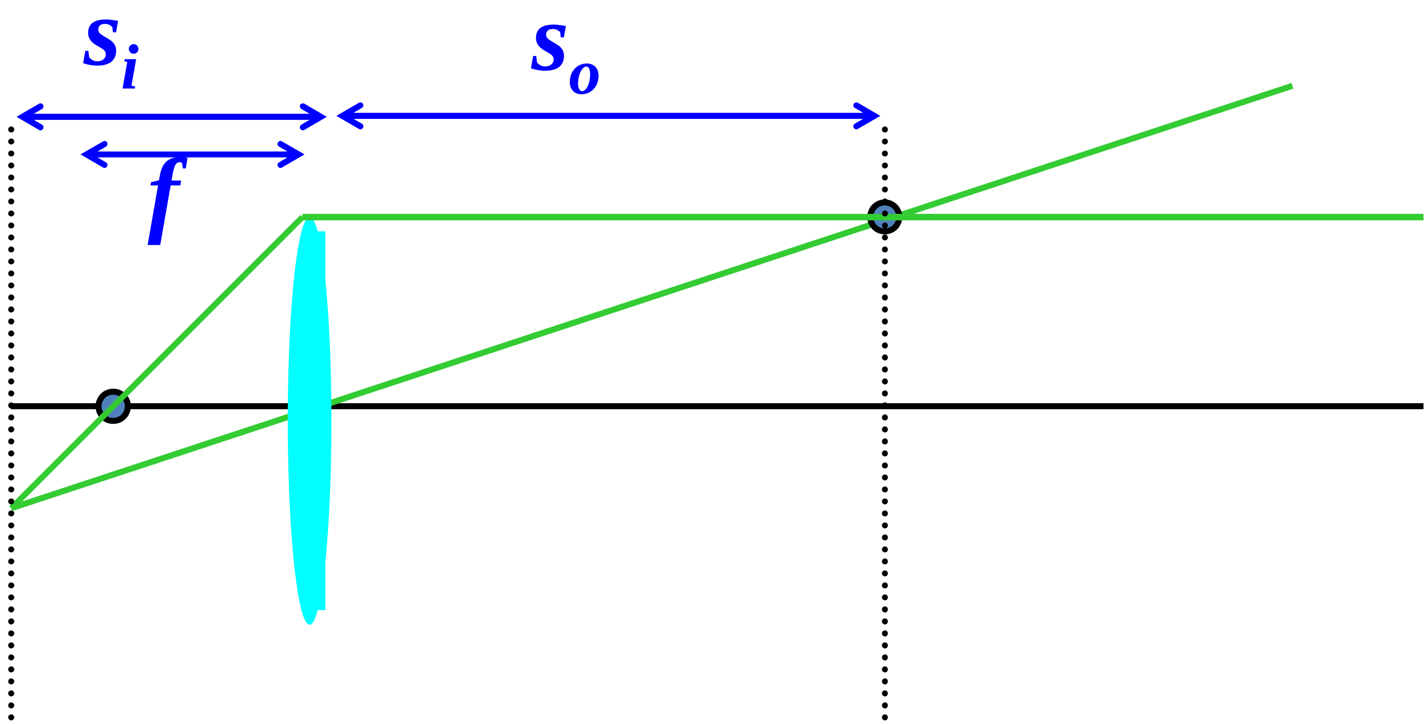
Pinhole camera



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

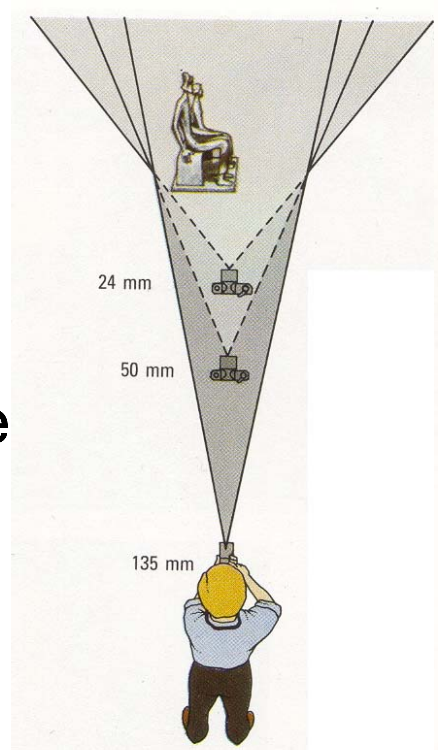
Thin lens formula

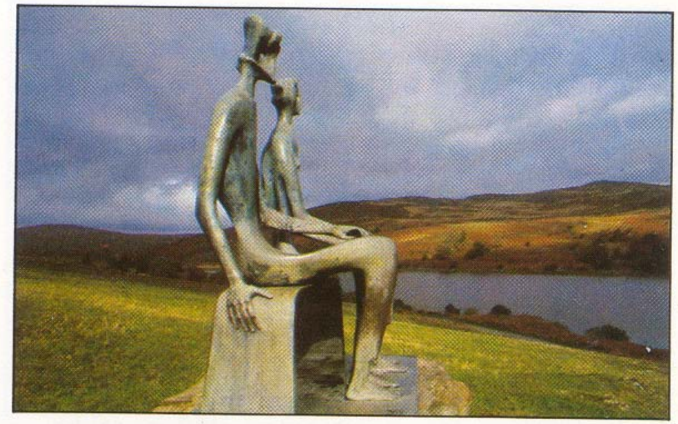
$$\boxed{\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}}$$



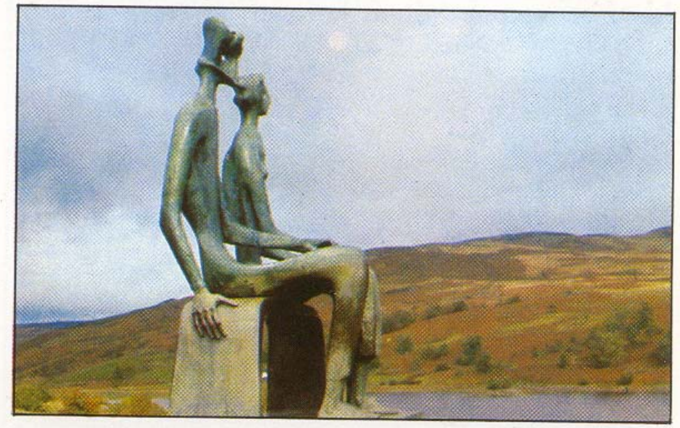
Changing the focal length vs. changing the viewpoint

- Telephoto makes it easier to select background (a small change in viewpoint is a big change in background.
  – changing the focal length lets us move back from a subject, while maintaining its size on the image
  – but moving back changes perspective relationships





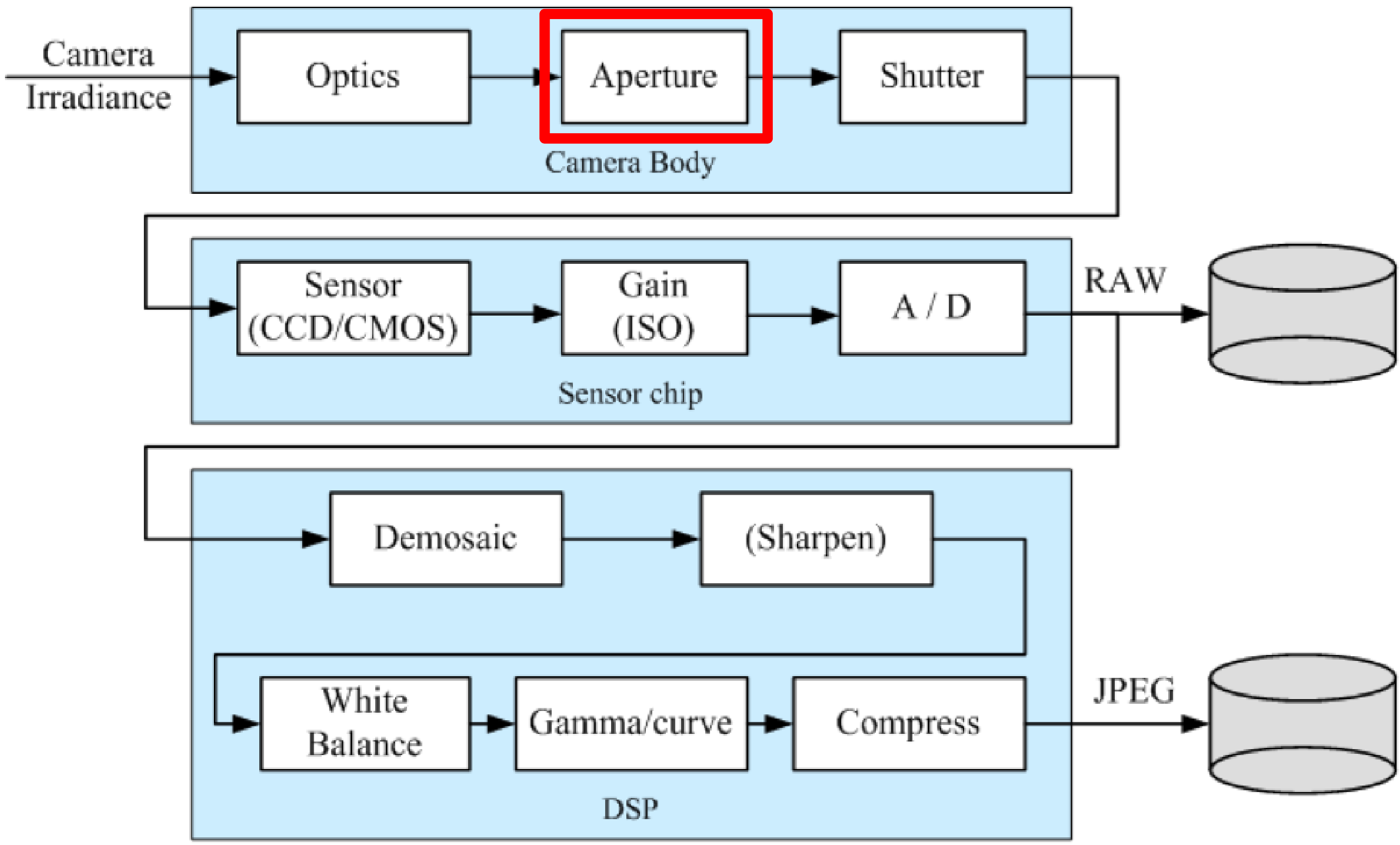
Grand-angulaire 24 mm



Normal 50 mm

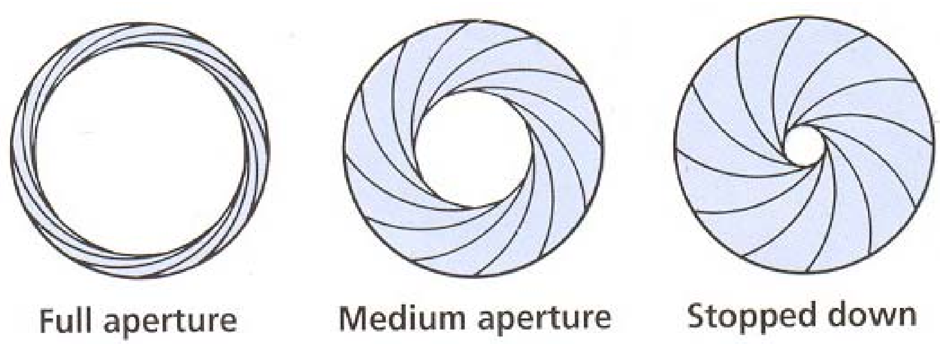
Longue focale 135 mm

Camera Body: Aperture



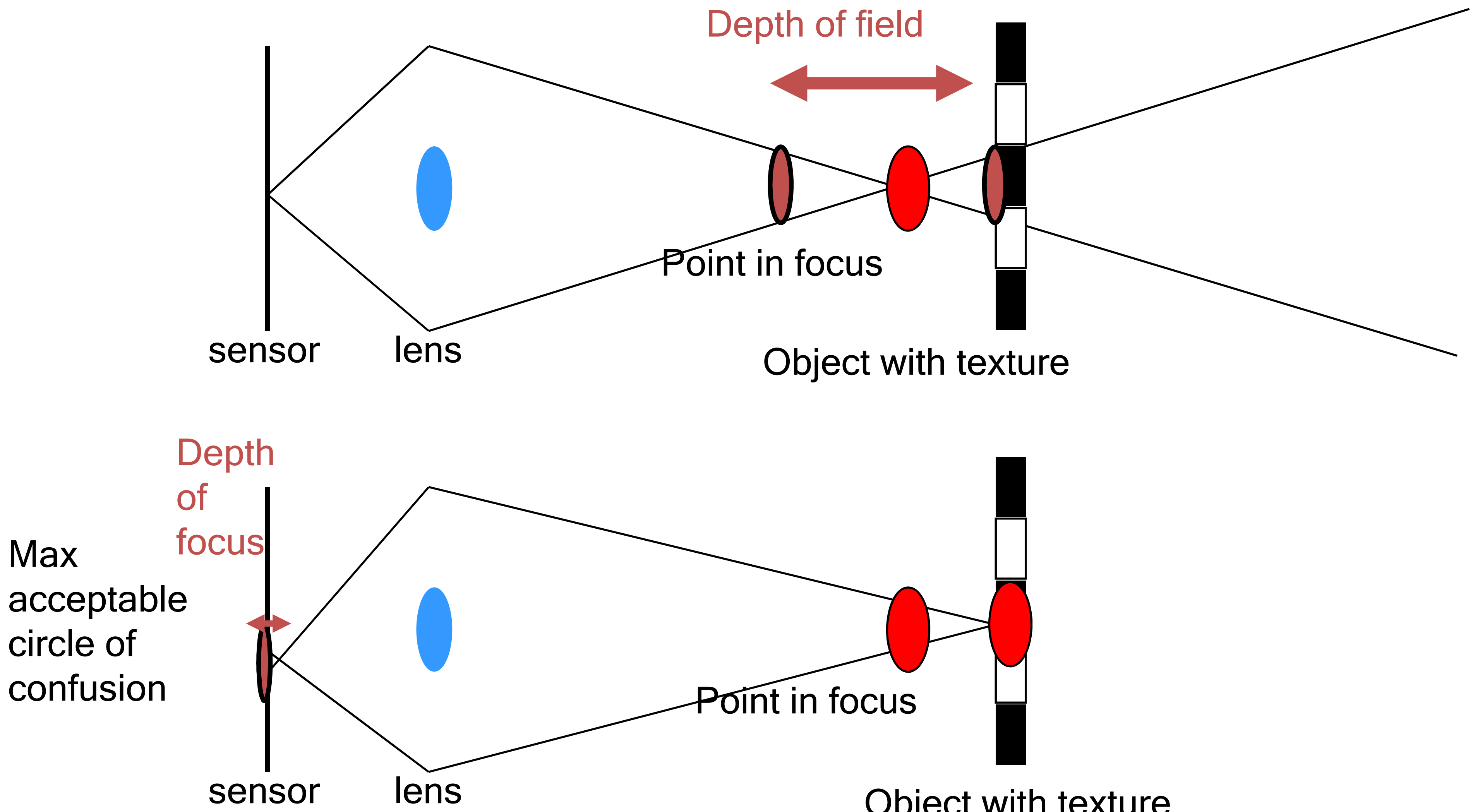
Aperture

- Diameter of the lens opening (controlled by diaphragm)
- Expressed as a fraction of focal length, in ***f-number*** ***N***
  - f/2.0 on a 50mm lens means that the aperture is 25mm
  - f/2.0 on a 100mm lens means that the aperture is 50mm
- Disconcerting: small f-number = big aperture
- What happens to the area of the aperture when going from f/2.0 to f/4.0?
- Typical f-numbers are (each of them counts as one f/stop) f/2.0, f/2.8, f/4, f/5.6, f/8, f/11, f/16, f/22, f/32
  - See the pattern?

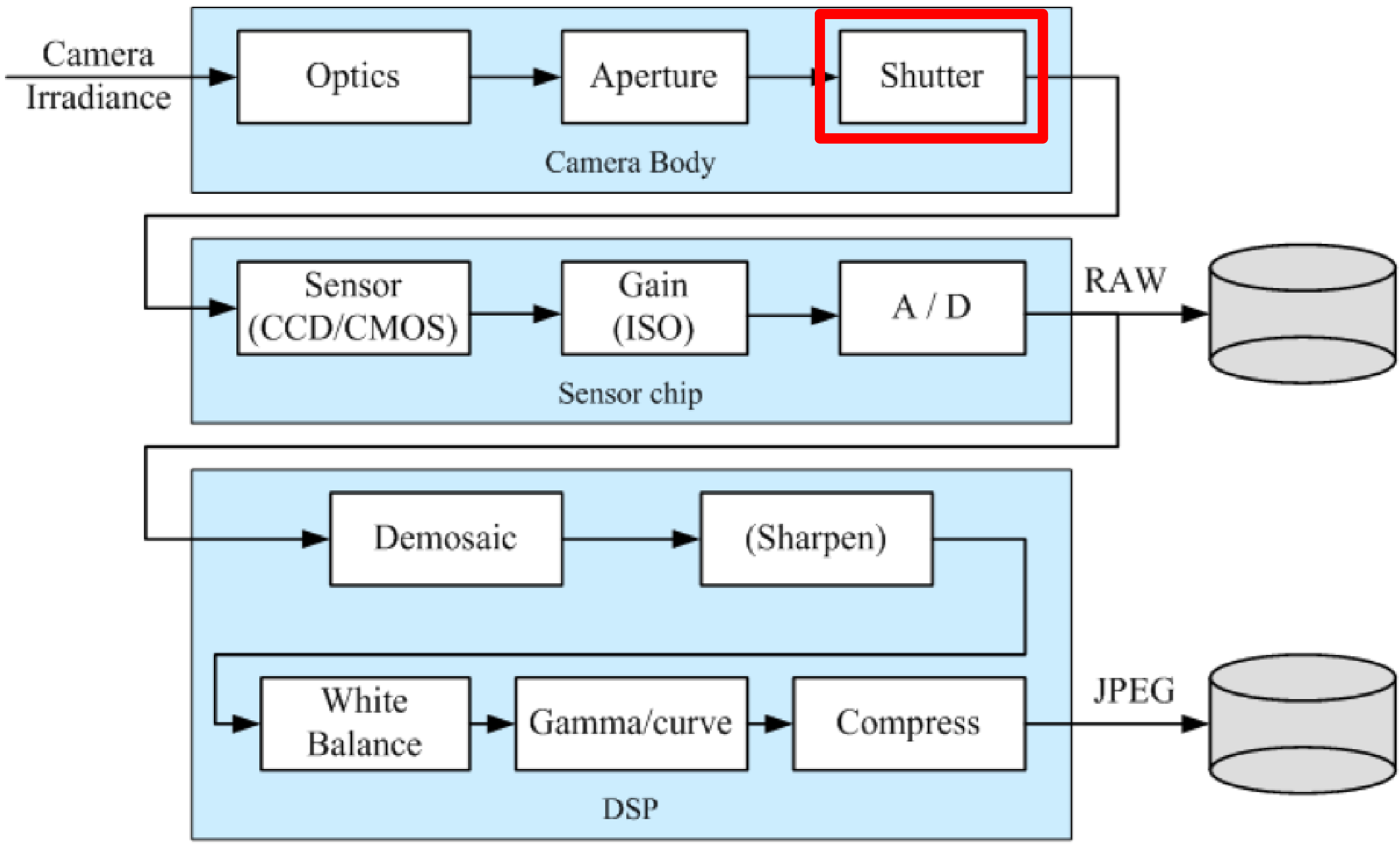


Depth of field

- We allow for some tolerance



Camera Body: Shutter

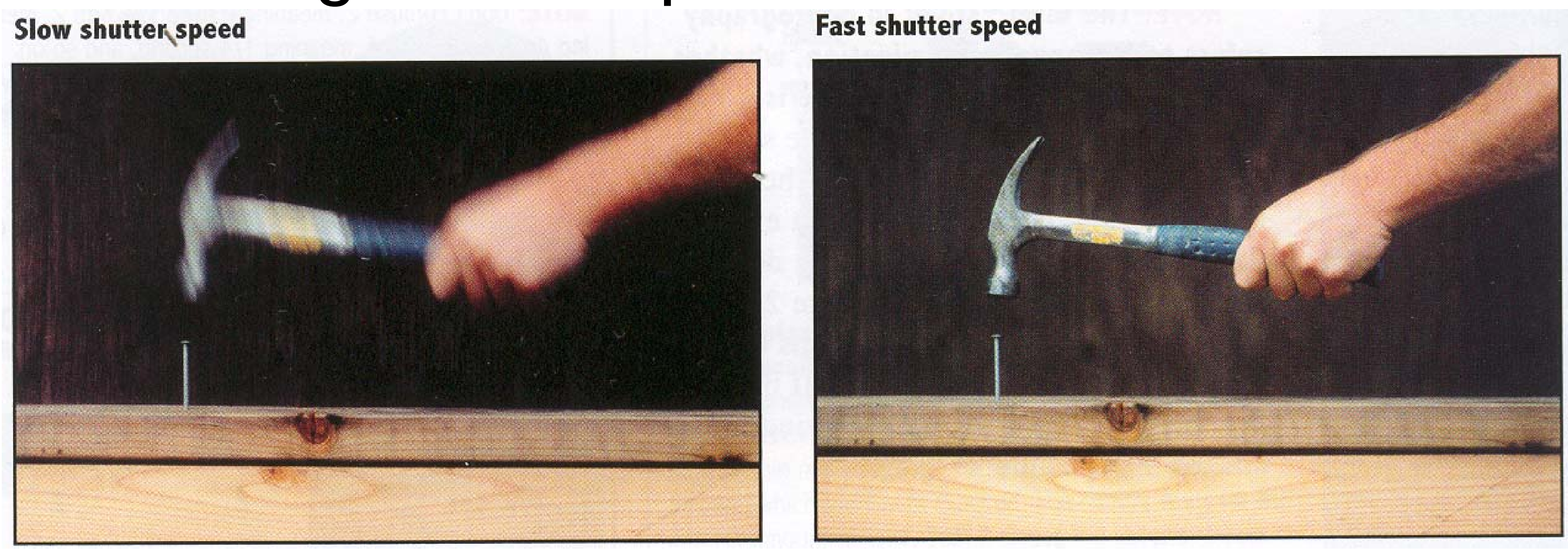


Shutter speed

- Controls how long the film/sensor is exposed
- Pretty much linear effect on exposure
- Usually in fraction of a second:
  - 1/30, 1/60, 1/125, 1/250, 1/500
  - Get the pattern ?
- On a normal lens, normal humans can hand-hold down to 1/60

Main effect of shutter speed

- Motion blur
- Halving shutter speed doubles motion blur

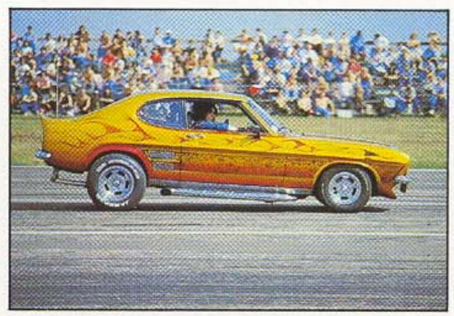


From Photography, London et al.

Effect of shutter speed

- Freezing motion

Walking people | Running people | Car | Fast train



1/125 | 1/250 | 1/500 | 1/1000

Exposure

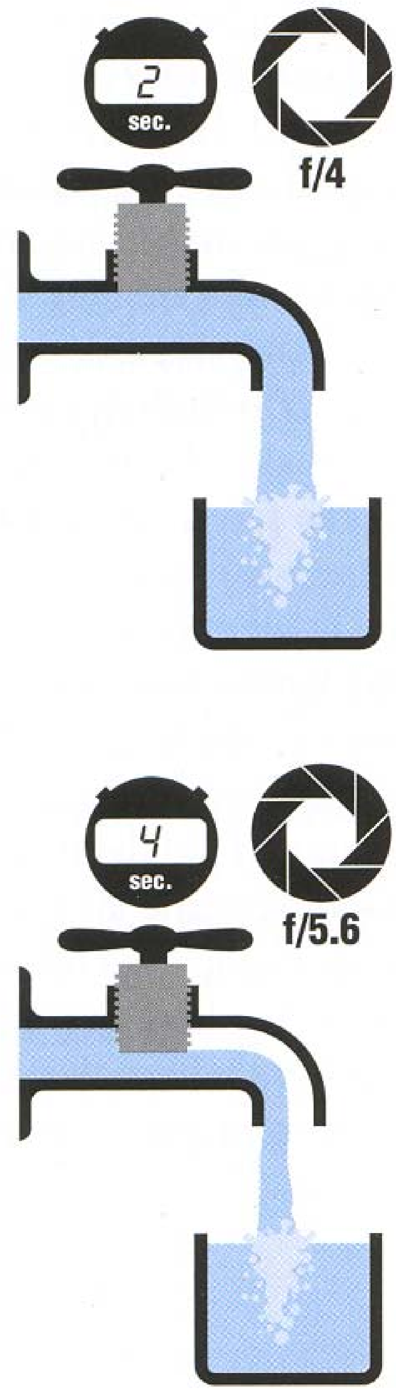
- Two main parameters:
  - Aperture (in f number)
  - Shutter speed (in fraction of a second)
- Exposure = irradiance x time

$$\boxed{H = E \times T}$$

- Irradiance (E)
  - controlled by aperture
- Exposure time (T)
  - controlled by shutter

Reciprocity

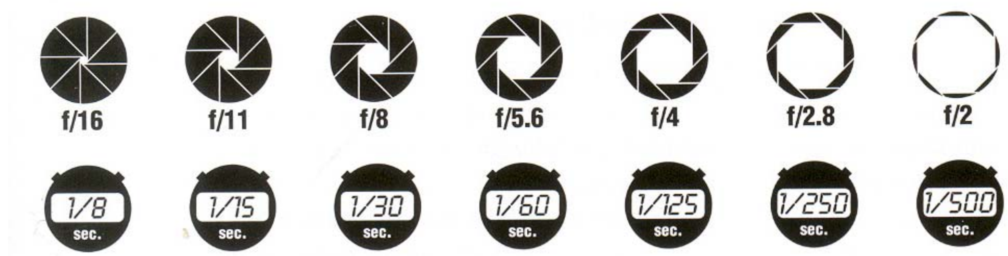
- Reciprocity
  - The same exposure is obtained with an exposure twice as long and an aperture *area* half as big
  - – Hence square root of two progression of f stops vs. power of two progression of shutter speed



From Photography, London et al.

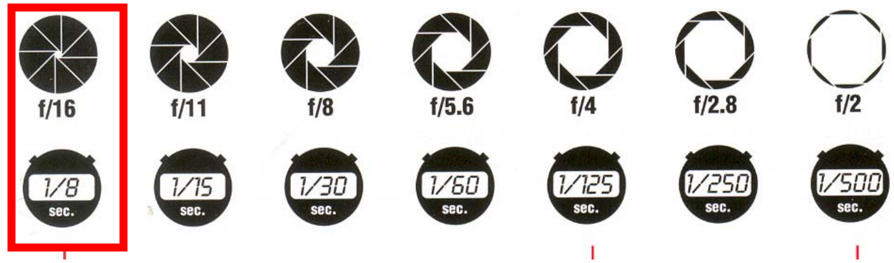
Reciprocity

- Assume we know how much light we need
- We have the choice of an infinity of shutter speed/aperture pairs



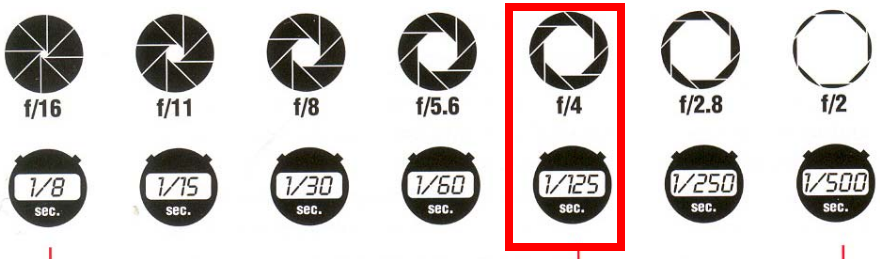
- What will guide our choice of a shutter speed?
  - Freeze motion vs. motion blur, camera shake
- What will guide our choice of an aperture?
  - Depth of field, distortion reduction, diffraction limit
- Often we must compromise
  - Open more to enable faster speed (but shallow DoF)

Small aperture (deep depth of field), slow shutter speed (motion blurred). *In th scene, a small aperture (f/16) produced great depth of field; the nearest paving stones as well as the farthest trees are sharp. But to admit enough light, a slow shutter speed (1/8 sec) was needed; it was too slow to show moving pigeons shar It also meant that a tripod had to be used to hold the camera steady.*



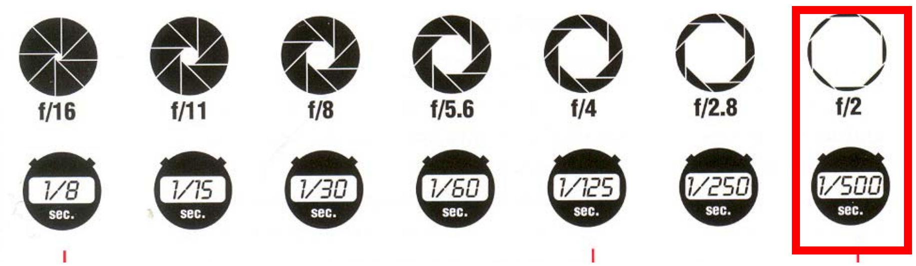
From Photography, London et al.

Medium aperture (moderate depth of field), medium shutter speed (some motion sharp). *A medium aperture (f/4) and shutter speed (1/125 sec) sacrifice some background detail to produce recognizable images of the birds. But the exposure is still too long to show the motion of the birds' wings sharply.*



From Photography, London et al.

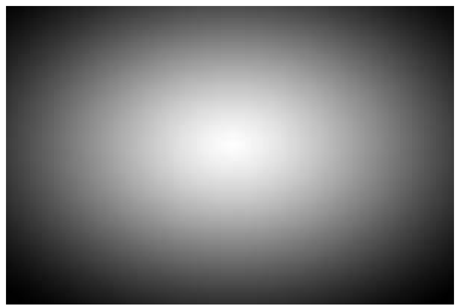
Large aperture (shallow depth of field), fast shutter speed (motion sharp). *A fa*
shutter speed (1/500 sec) stops the motion of the pigeons so completely that the
flapping wings are frozen. But the wide aperture (f/2) needed gives so little dept
of field that the background is now out of focus.



From Photography, London et al.

Metering

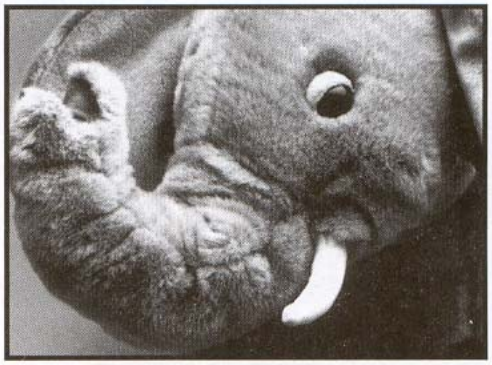
- Photosensitive sensors measure scene luminance
- Usually TTL (through the lens)
- Simple version: center-weighted average



- Assumption? Failure cases?
  - Usually assumes that a scene is 18% gray
  - Problem with dark and bright scenes

White polar bear given exposure suggested by meter

White polar bear given 2 stops more exposure



Gray elephant given exposure suggested by meter

Black gorilla given exposure suggested by meter

Black gorilla given 2 stops less exposure

From Photography, London et al.

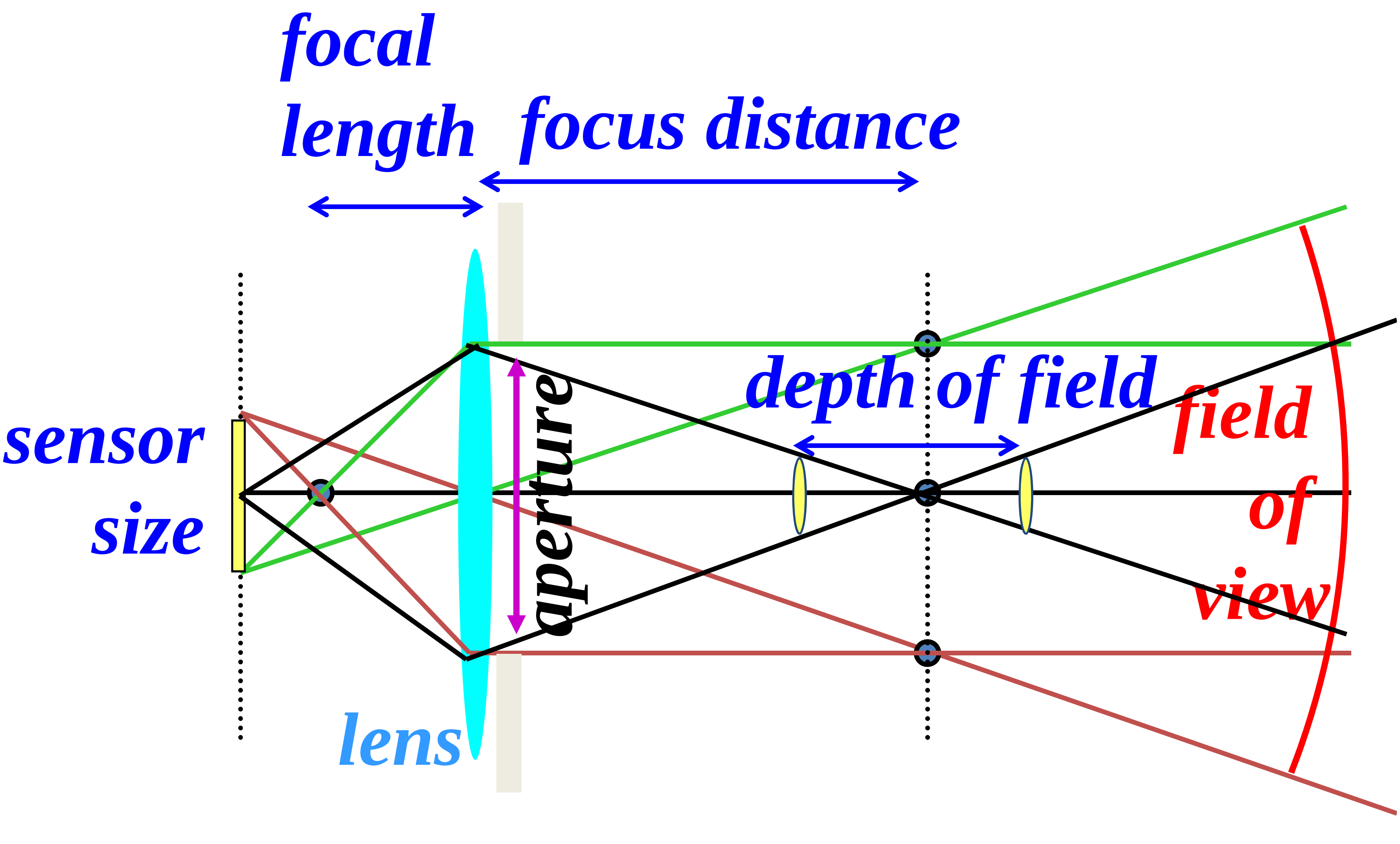
Exposure & Metering

- The camera metering system measures how bright the scene is
- In Aperture priority mode, the photographer sets the aperture, the camera sets the shutter speed
- In Shutter-speed priority mode, the photographers sets the shutter speed and the camera deduces the aperture
  - In both cases, reciprocity is exploited
- In Program mode, the camera decides both exposure and shutter speed (middle value more or less)
- In Manual, the user decides everything (but can get feedback)

Pros and cons of various modes

- Aperture priority
  – Direct depth of field control
  – Cons: can require impossible shutter speed (e.g. with f/1.4 for a bright scene)
- Shutter speed priority
  – Direct motion blur control
  – Cons: can require impossible aperture (e.g. when requesting a 1/1000 speed for a dark scene)
    - Note that aperture is somewhat more restricted
- Program
  – Almost no control, but no need for neurons
- Manual
  – Full control, but takes more time and thinking

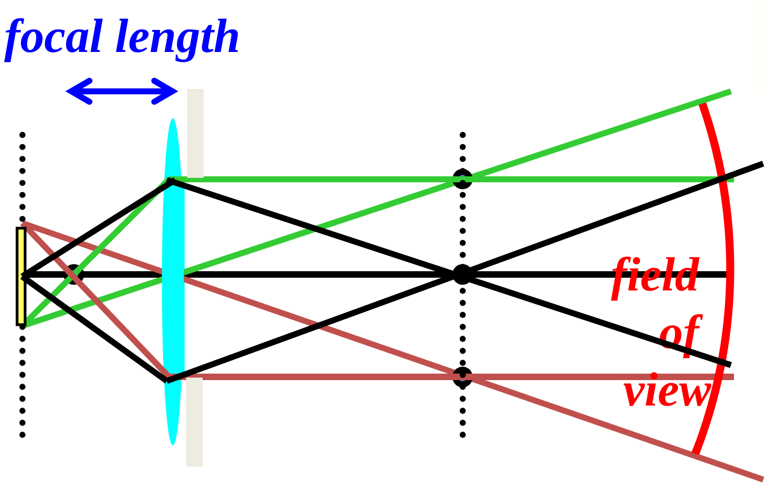
Recap

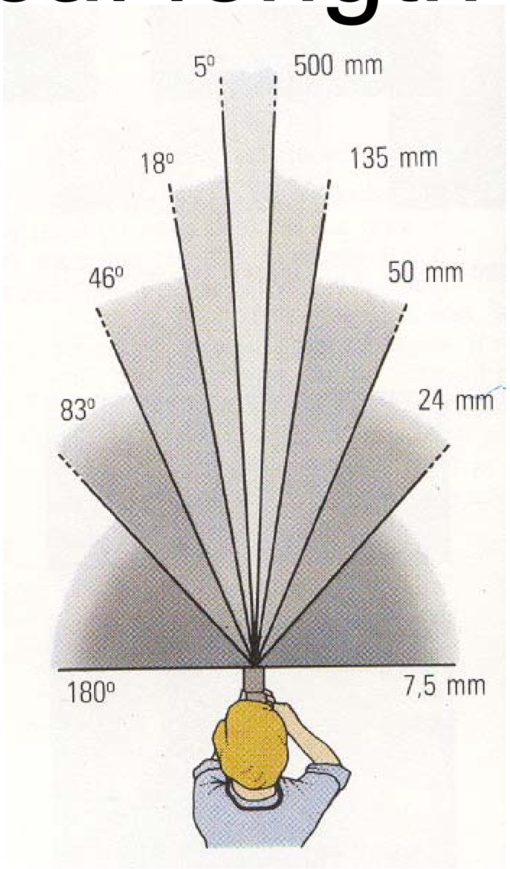


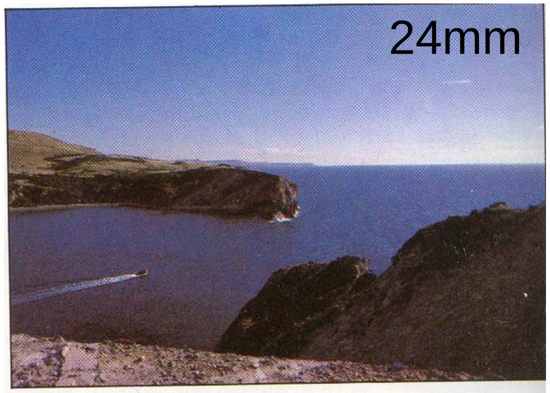
focal
length
focus distance
sensor
size
aperture
depth of field
field
of
view
lens

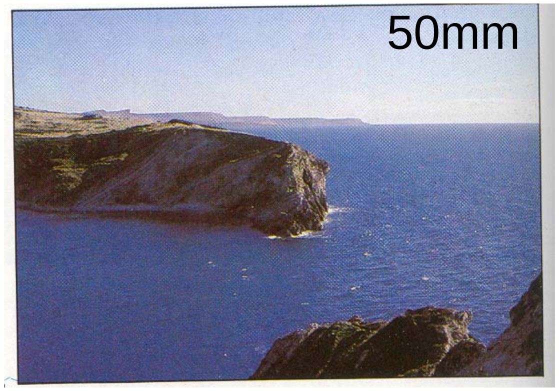
Focal length

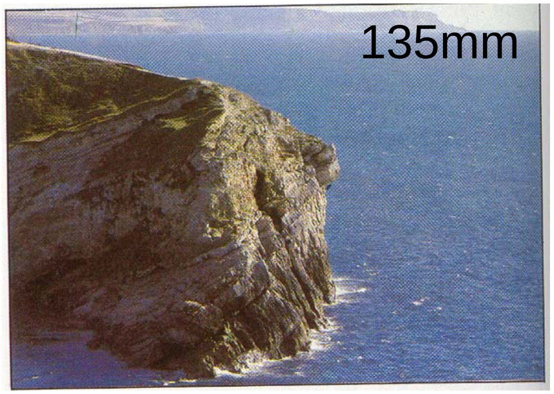
- <30mm: wide angle
- 50mm: standard
- >100mm telephoto
- Affected by sensor size (crop factor)









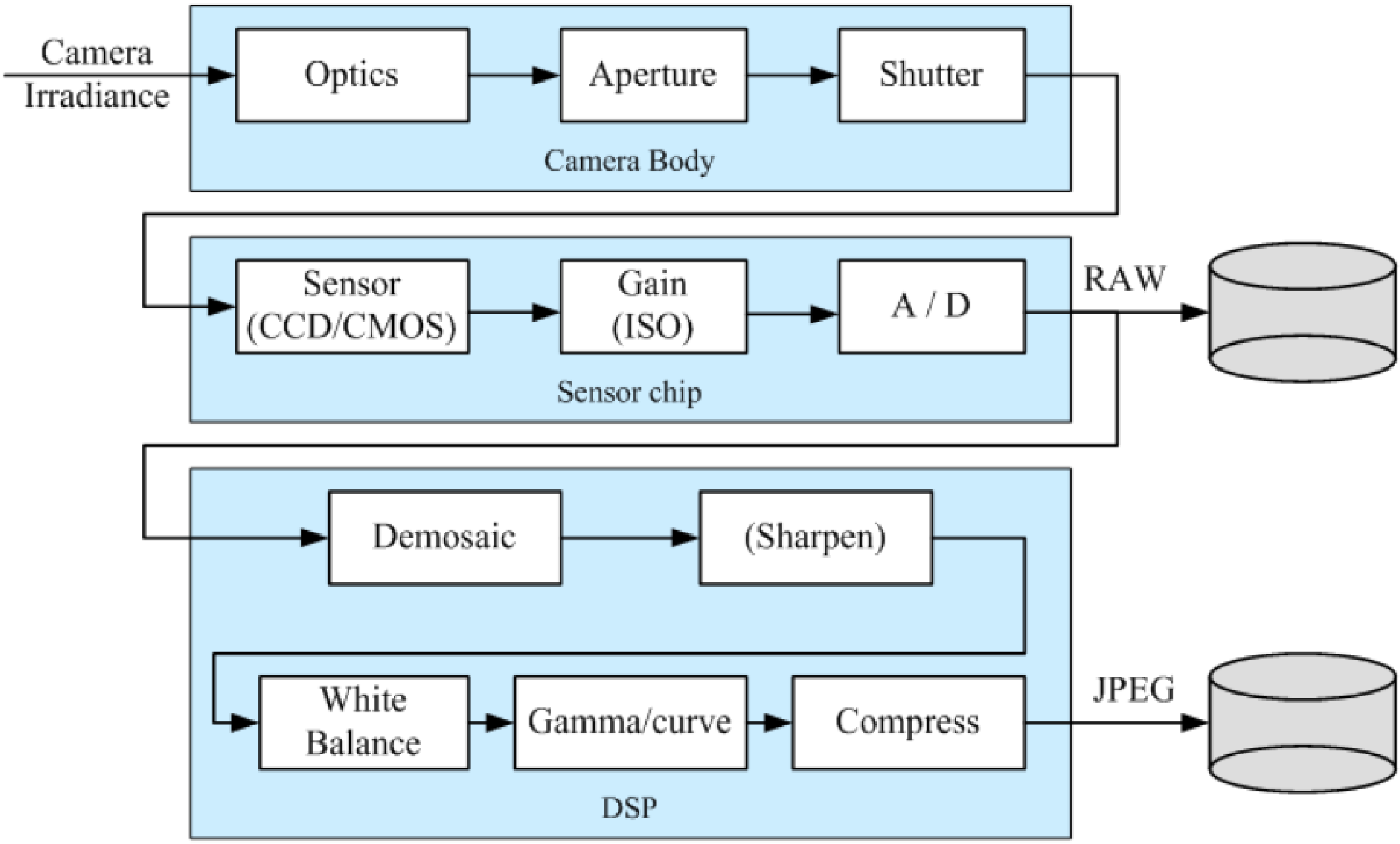


Exposure

- Aperture (f number)
  - Expressed as ratio between focal length and aperture diameter: diameter = f / <f number>
  - f/2.0, f/2.8, f/4.0, f/5.6, f/8.0, f/11, f/16 (factor of sqrt (2))
  - Small f number means large aperture
  - Main effect: depth of field
  - A good standard lens has max aperture f/1.8.
    A cheap zoom has max aperture f/3.5
- Shutter speed
  - In fraction of a second
  - 1/30, 1/60, 1/125, 1/250, 1/500 (factor of 2)
  - Main effect: motion blur
- Sensitivity
  - Gain applied to sensor
  - In ISO, bigger number, more sensitive (100, 200, 400, 800, 1600)
  - Main effect: sensor noise

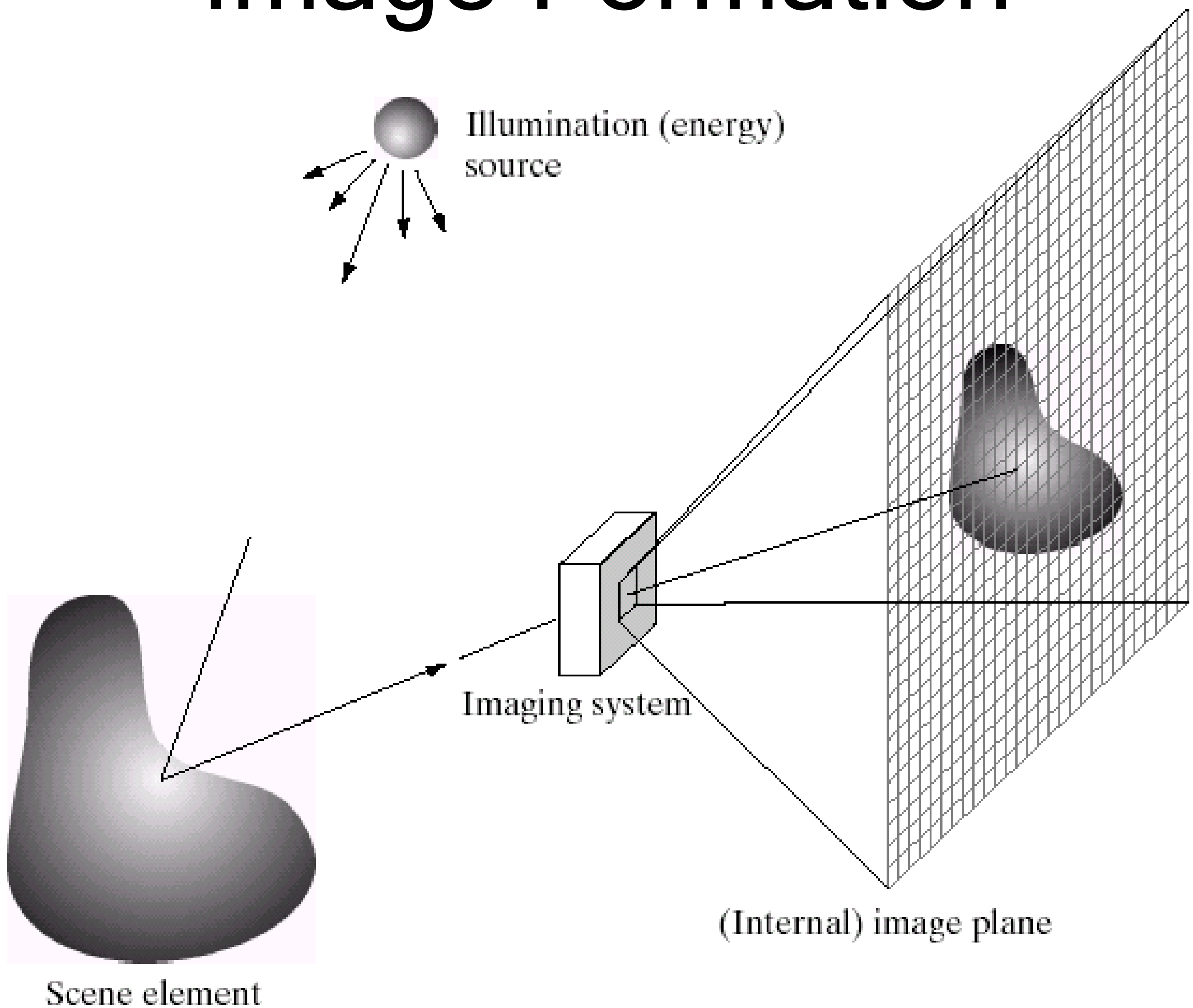
Reciprocity between these three numbers:
for a given exposure, one has two degrees of freedom.

Sensor Chip



Camera
Irradiance
Optics
Aperture
Shutter
Camera Body
Sensor
(CCD/CMOS)
Gain
(ISO)
A / D
Sensor chip
RAW
Demosaic
(Sharpen)
White
Balance
Gamma/curve
Compress
DSP
JPEG

Image Formation

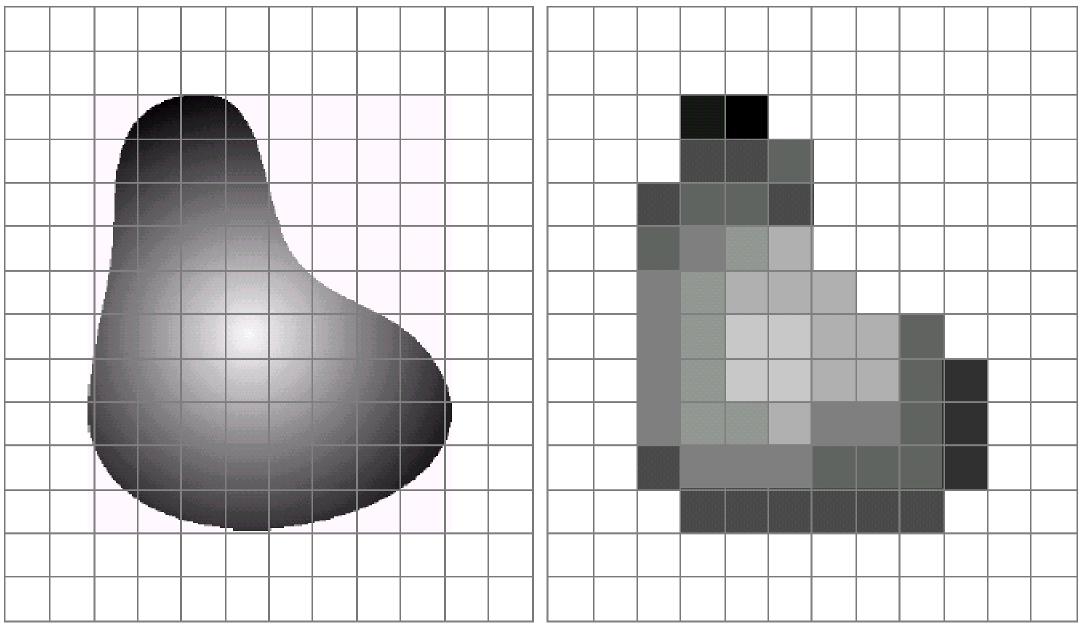


Digital camera

A digital camera replaces film with a sensor array

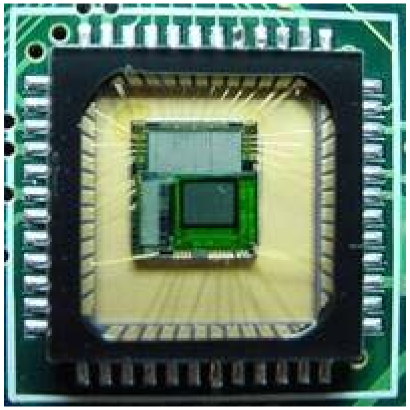
- Each cell in the array is light-sensitive diode that converts photons to electrons
- Two common types: Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS)

Sensor Array



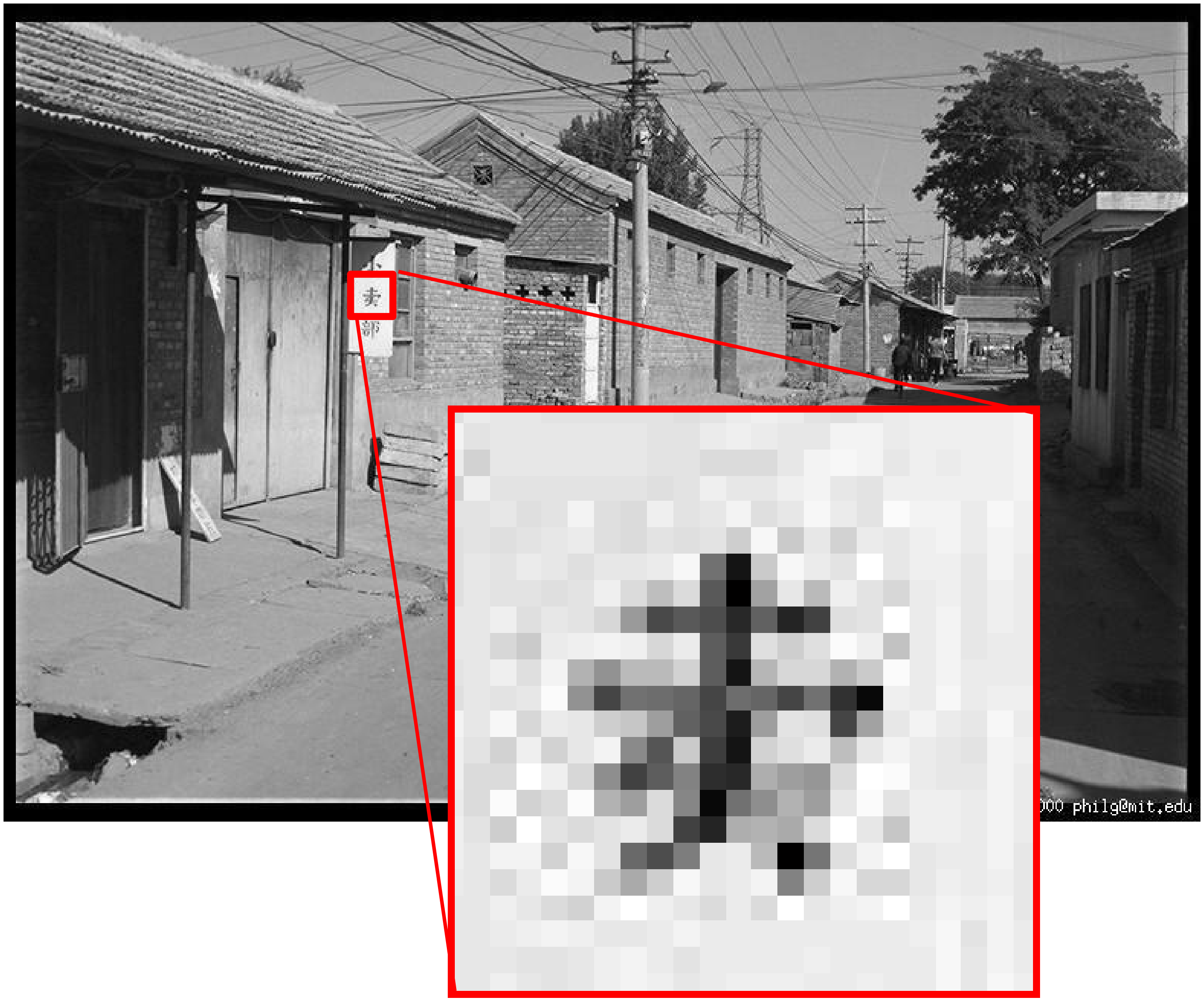
a b

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

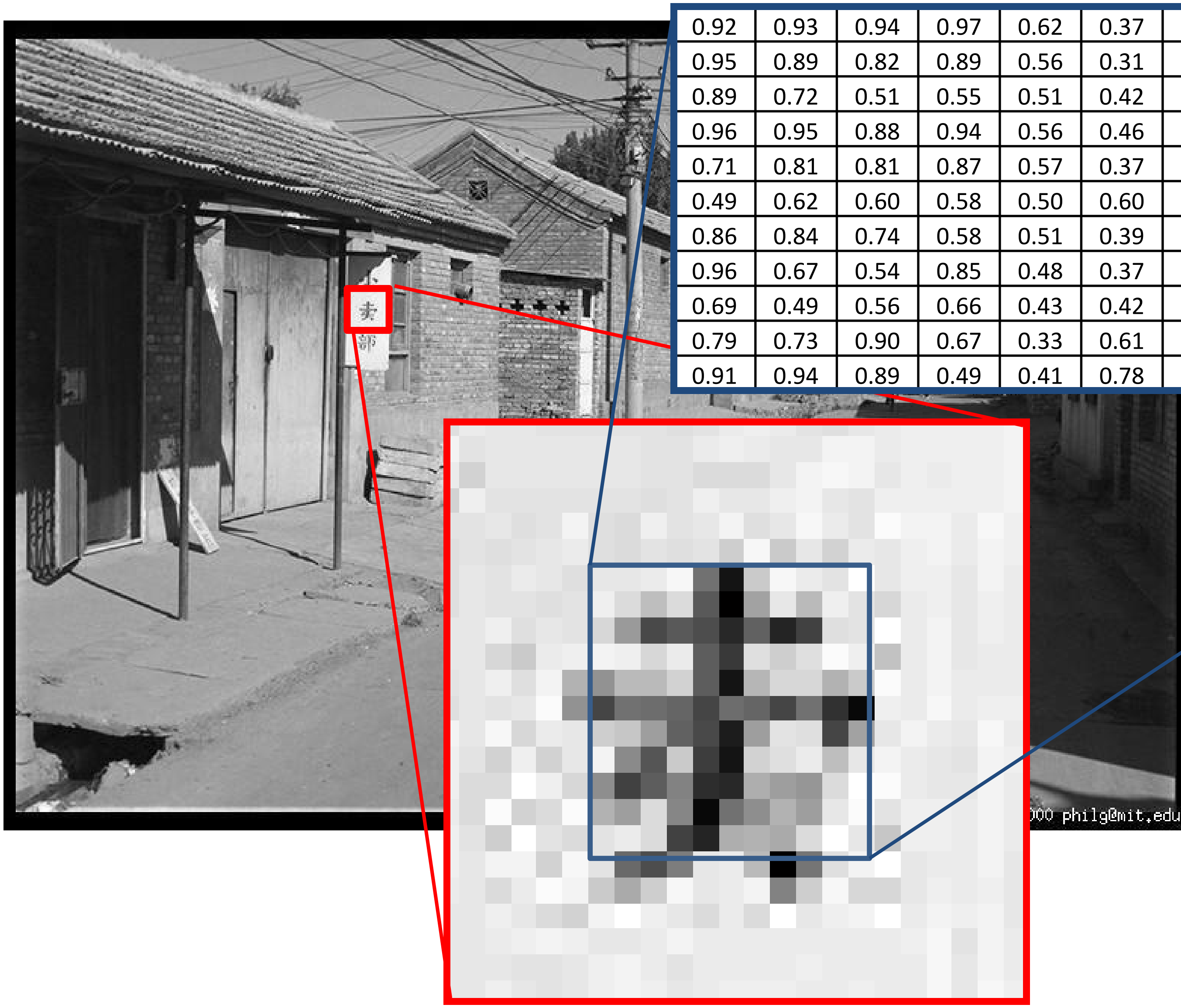


CMOS sensor

The raster image (pixel matrix)

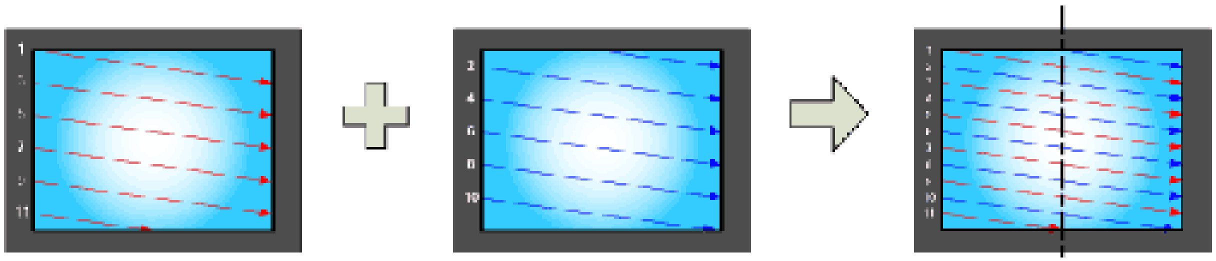


The raster image (pixel matrix)



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

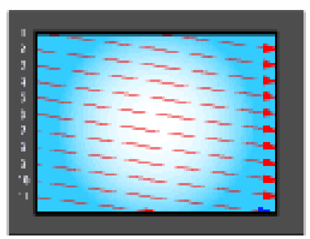
Interlace vs. progressive scan



1st field: Odd field

2nd field: Even field

One complete frame
using interlaced scanning



One complete frame
using progressive scanning

http://www.axis.com/products/video/camera/progressive_scan.htm

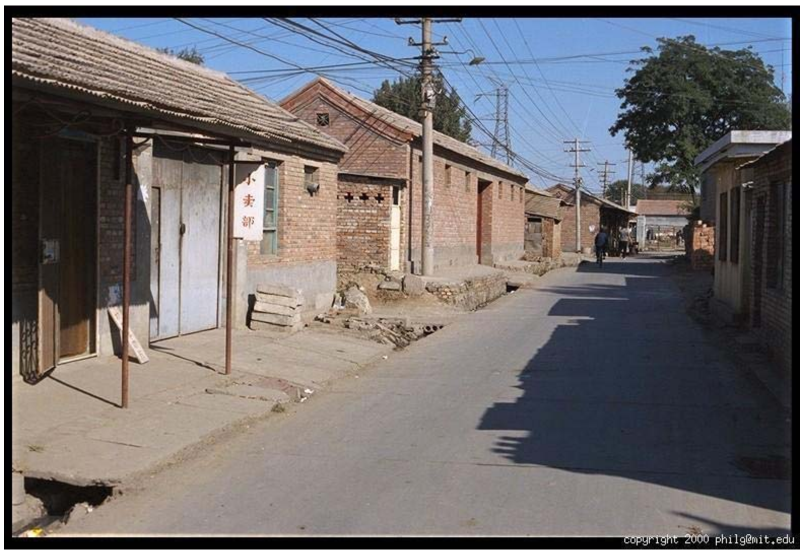
Progressive scan

http://www.axis.com/products/video/camera/progressive_scan.htm

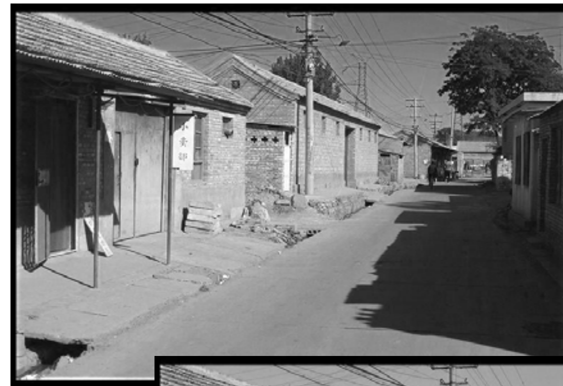
Interlace

http://www.axis.com/products/video/camera/progressive_scan.htm

Color Image



R

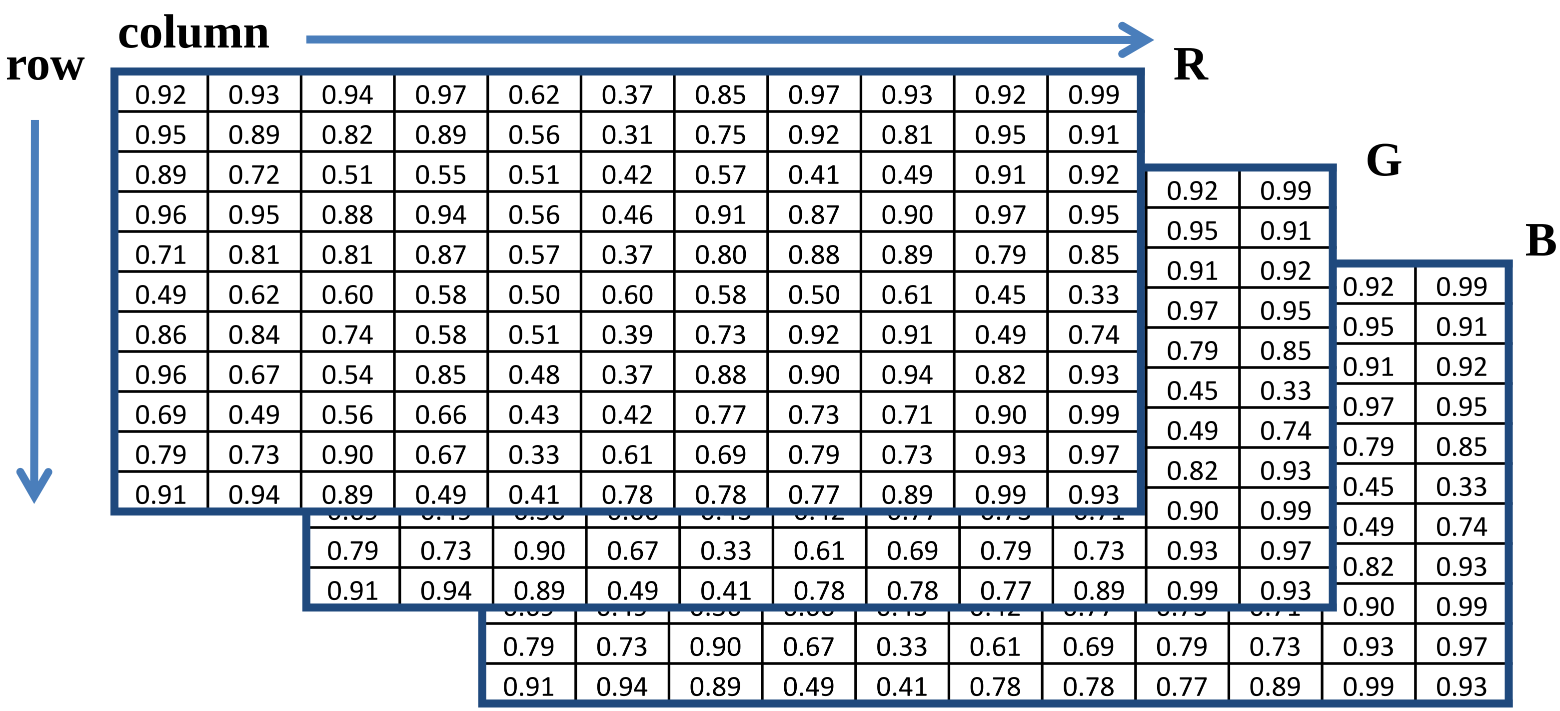


G

B

Images in Matlab

- Images represented as a matrix
- Suppose we have a NxM RGB image called "im"
  - im(1,1,1) = top-left pixel value in R-channel
  - im(y, x, b) = y pixels down, x pixels to right in the b^{th} channel
  - im(N, M, 3) = bottom-right pixel in B-channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - Convert to double format (values 0 to 1) with im2double



CCD color sampling

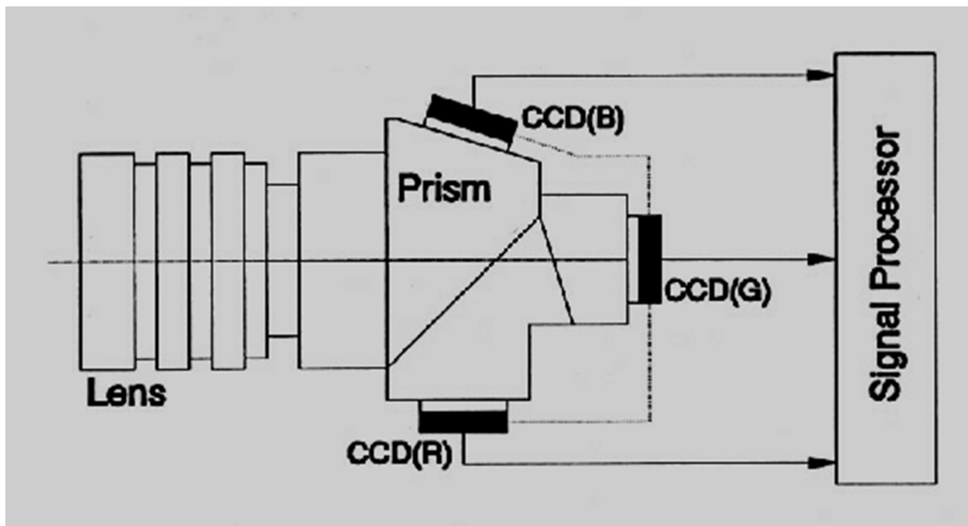
- Problem: a photosite can record only one number
- We need 3 numbers for color

Some approaches to color sensing

- Scan 3 times (temporal multiplexing)
  - Drum scanners
  - Flat-bed scanners
  - Russian photographs from 1800's
- Use 3 detectors
  - High-end 3-tube or 3-ccd video cameras
- Use spatially offset color samples (spatial multiplexing)
  - Single-chip CCD color cameras
  - Human eye

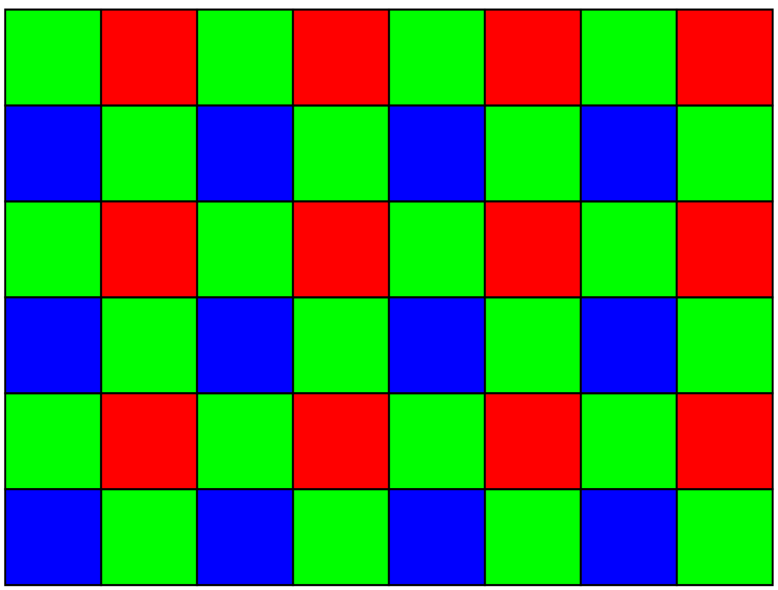
3 CCD Sensor

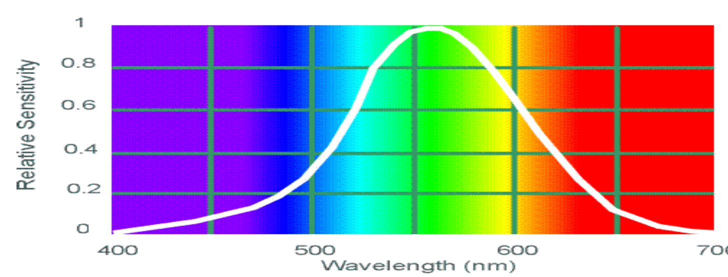
- 3-chip vs. 1-chip: quality vs. cost



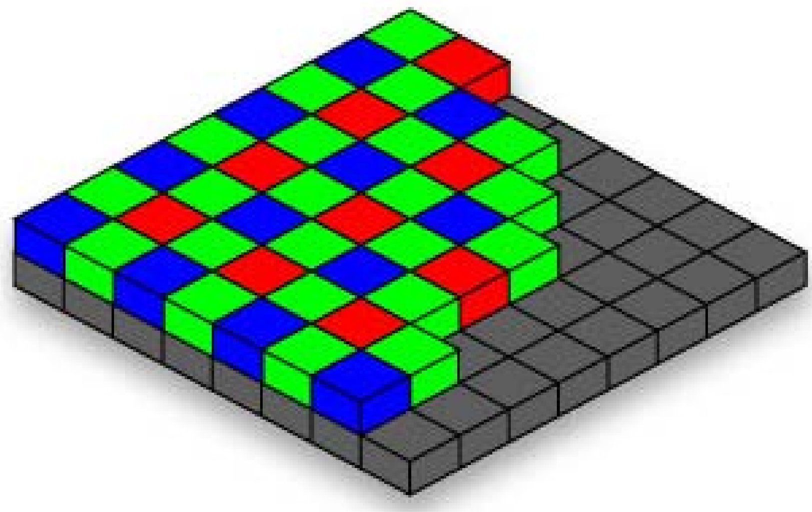
Spatial Multiplexing: Bayer Grid

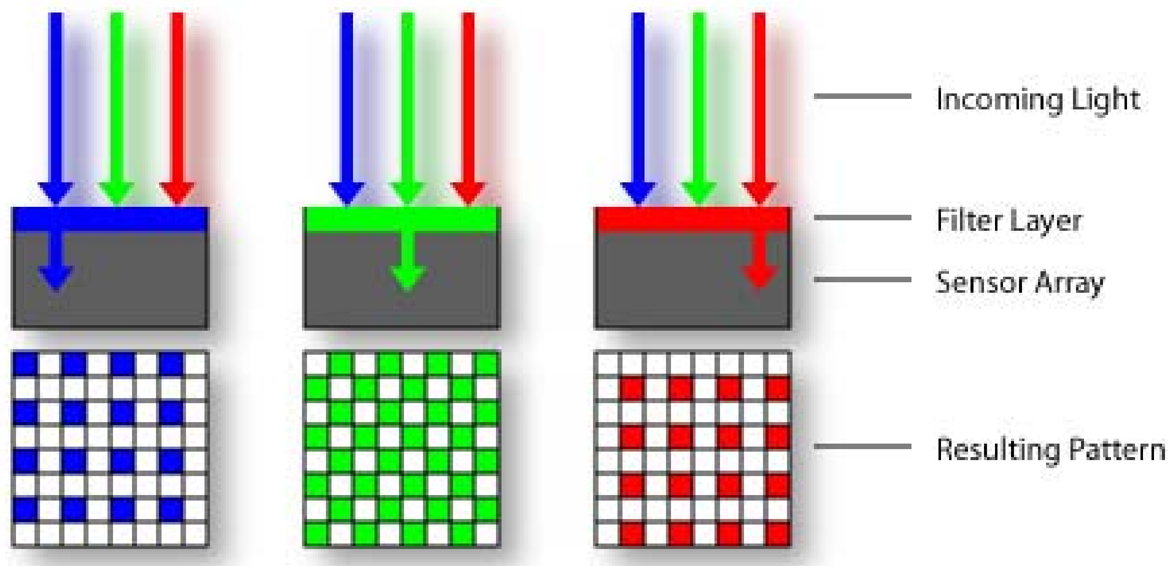
- Why more green?
  - We have 3 channels and square lattice doesn't like odd numbers
  - It's the spectrum "in the middle"
  - More important to human perception of brightness





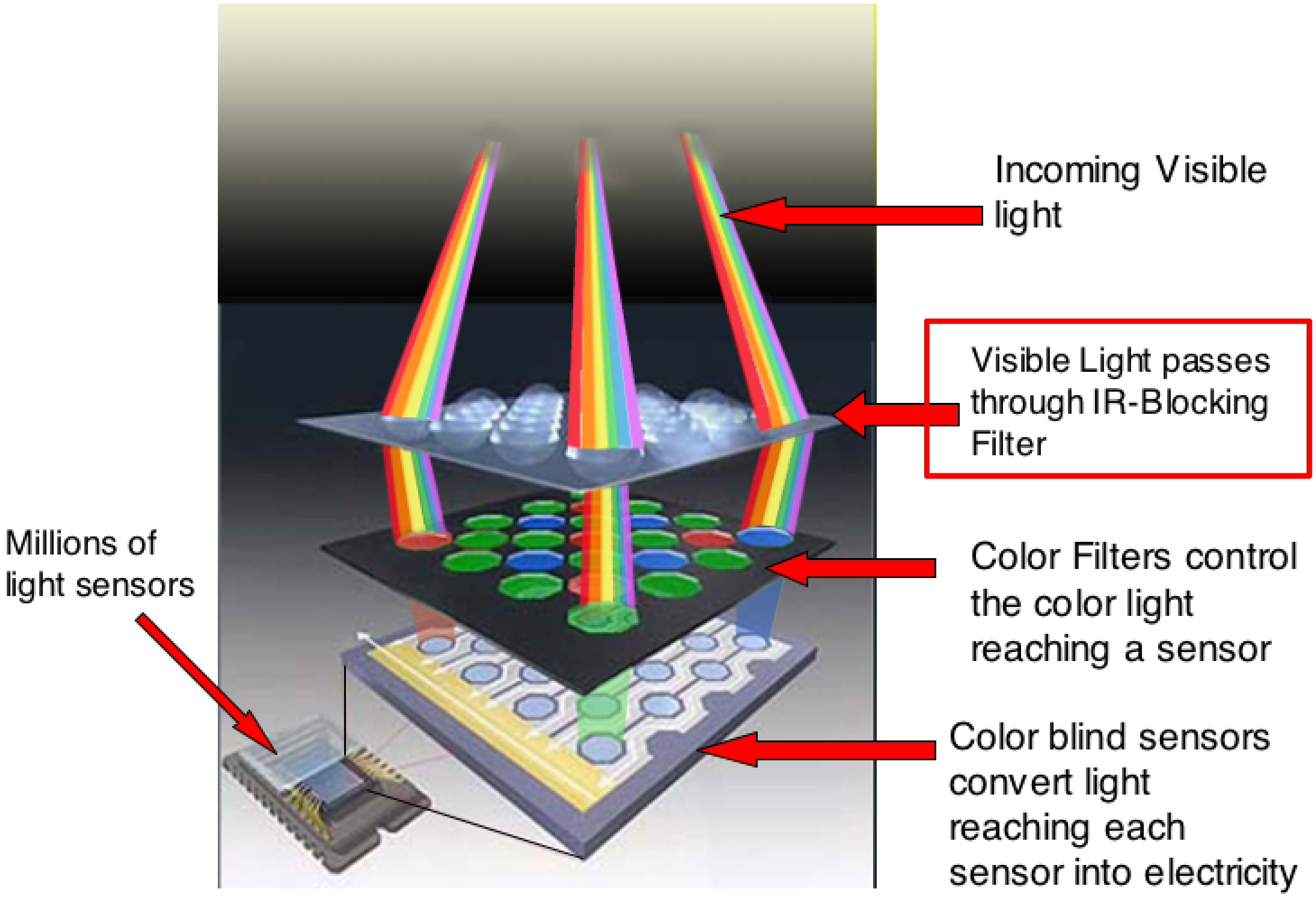
Practical Color Sensing: Bayer Grid





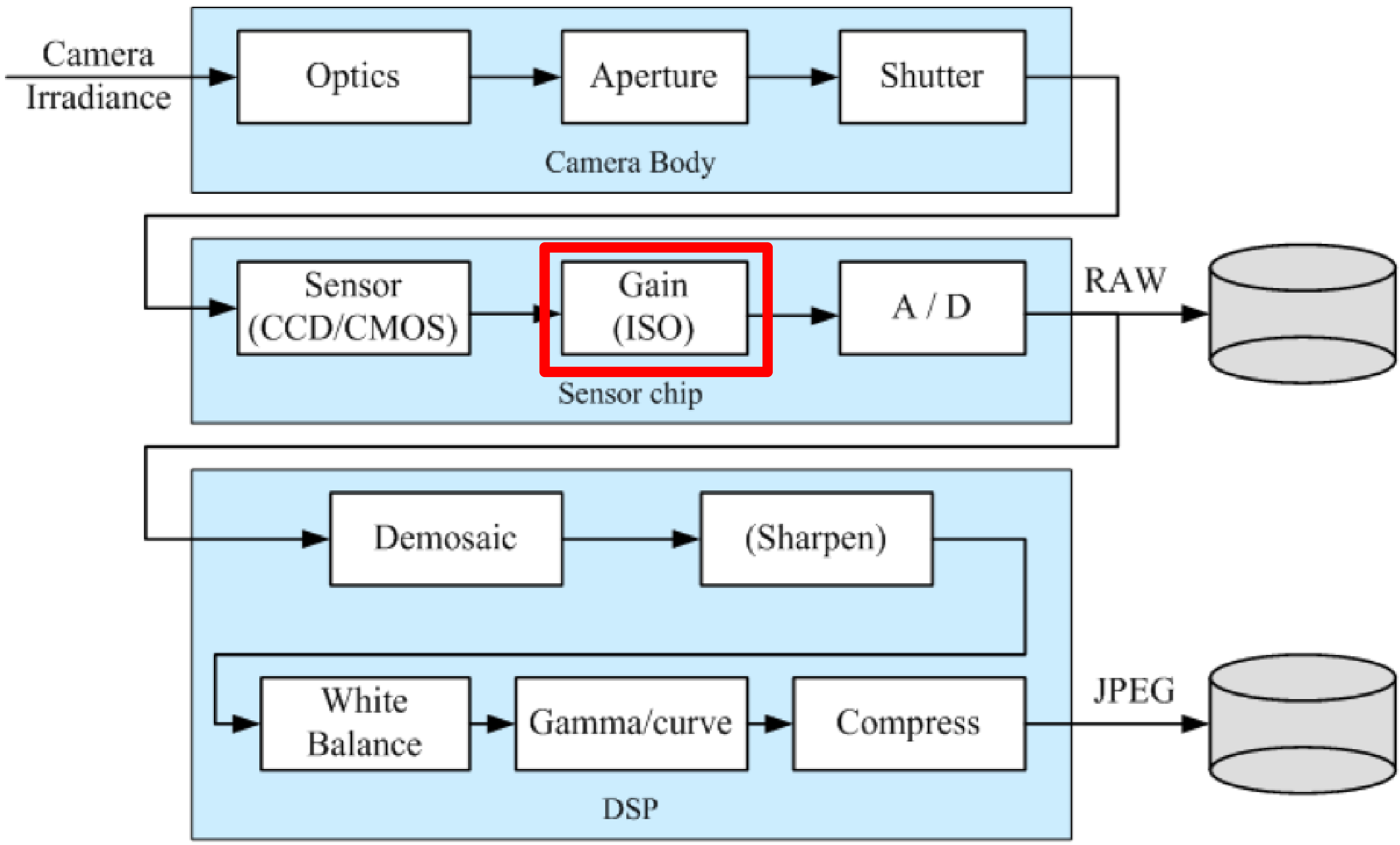
Recap: Camera sensor

RGB Inside the Camera



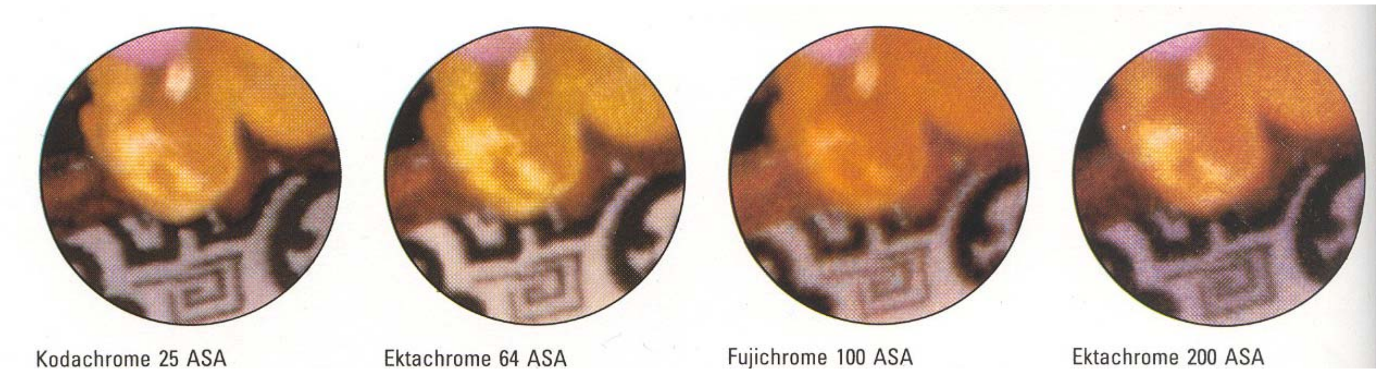
http://www.photoaxe.com/wp-content/uploads/2007/04/camera-sensor.jpg

Sensor Chip: Gain

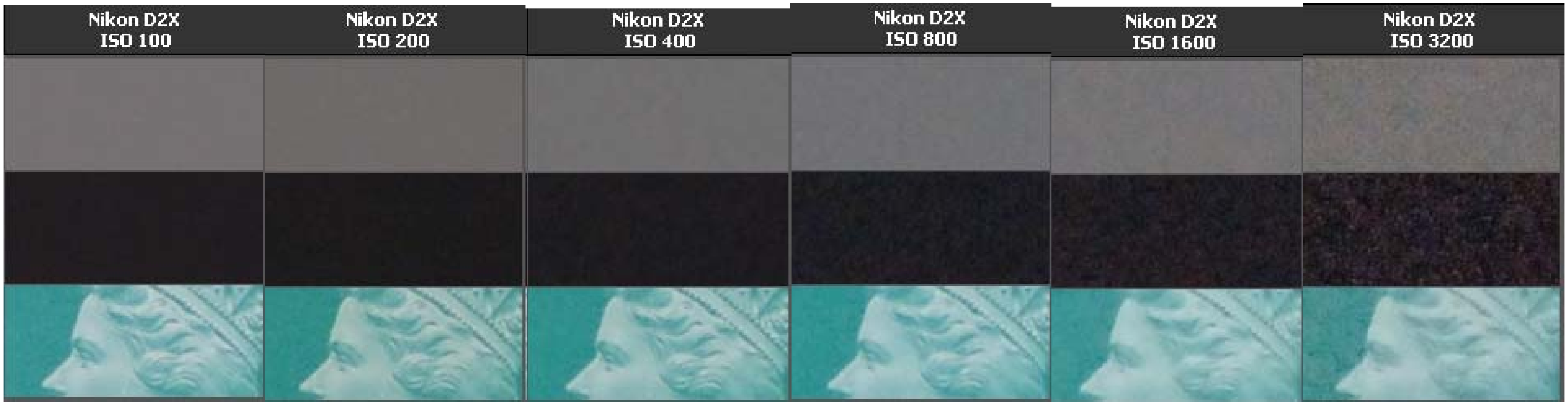


Sensitivity (ISO)

- Third variable for exposure: gain applied to sensor
- Linear effect (200 ISO needs half the light as 100 ISO)
- Film photography: trade sensitivity for grain



- Digital photography: trade sensitivity for noise



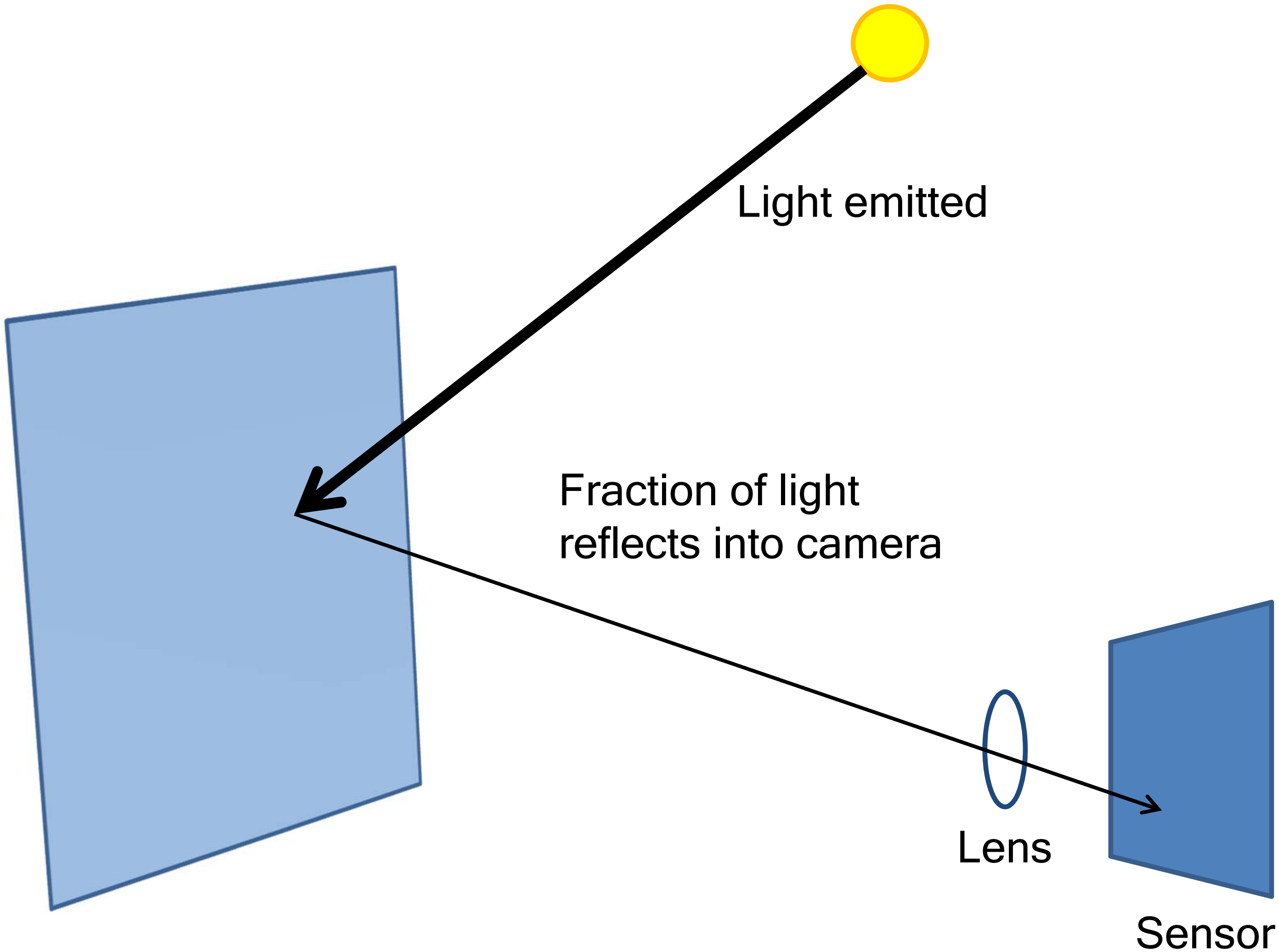
From dpreview.com

Light and Shading

Slides by D. Hoiem

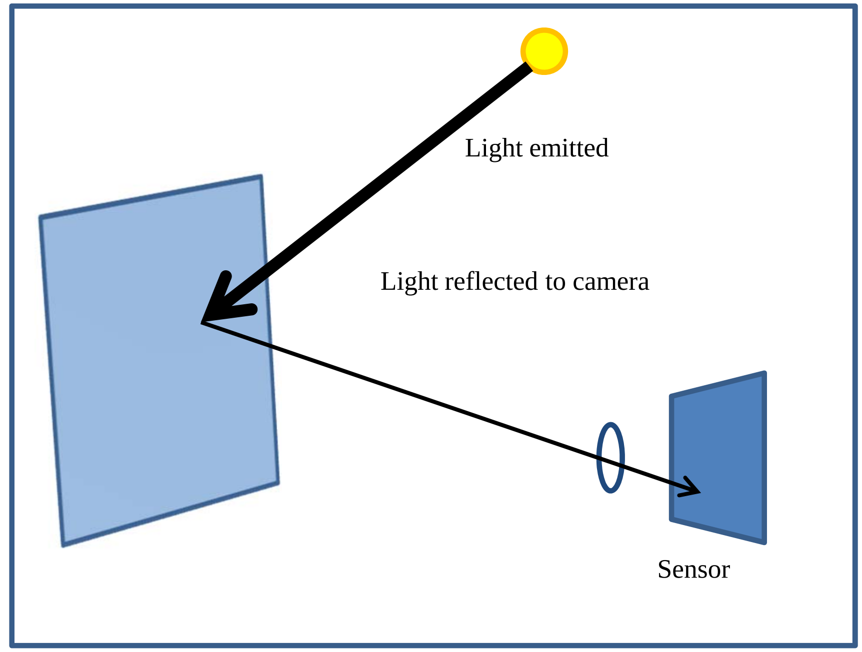
- What determines a pixel’s intensity?
- What can we infer about the scene from pixel intensities?

How does a pixel get its value?



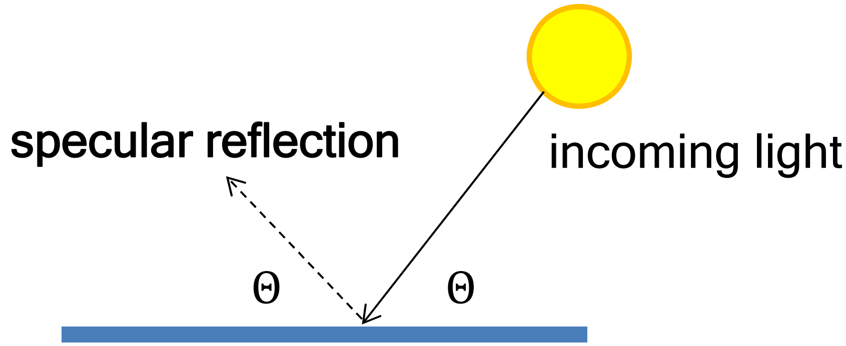
How does a pixel get its value?

- Major factors
  - Illumination strength and direction
  - Surface geometry
  - Surface material
  - Nearby surfaces
  - Camera gain/exposure

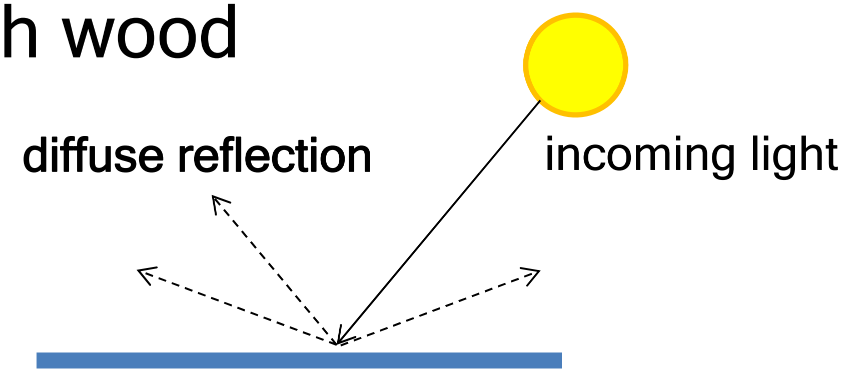


Basic models of reflection

- Specular: light bounces off at the incident angle
  - E.g., mirror

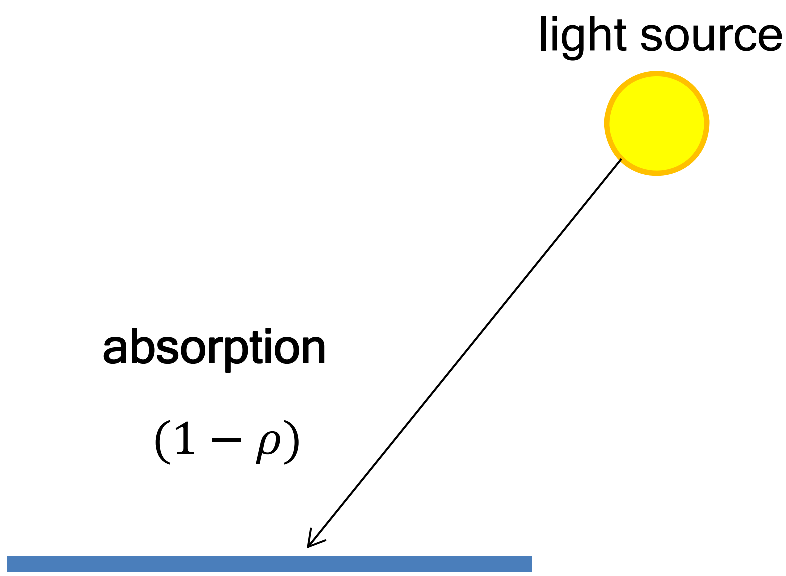


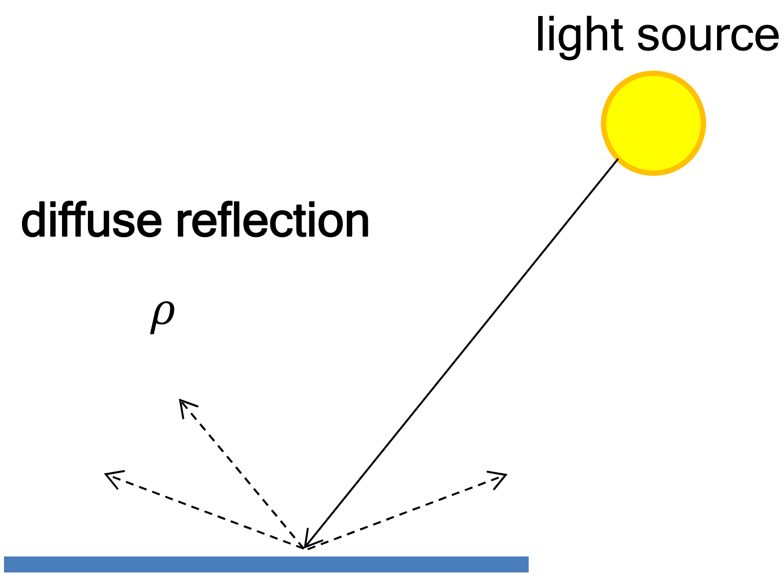
- Diffuse: light scatters in all directions
  - E.g., brick, cloth, rough wood



Lambertian reflectance model

- Some light is absorbed (function of albedo ρ)
- Remaining light is scattered (diffuse reflection)
- Examples: soft cloth, concrete, matte paints

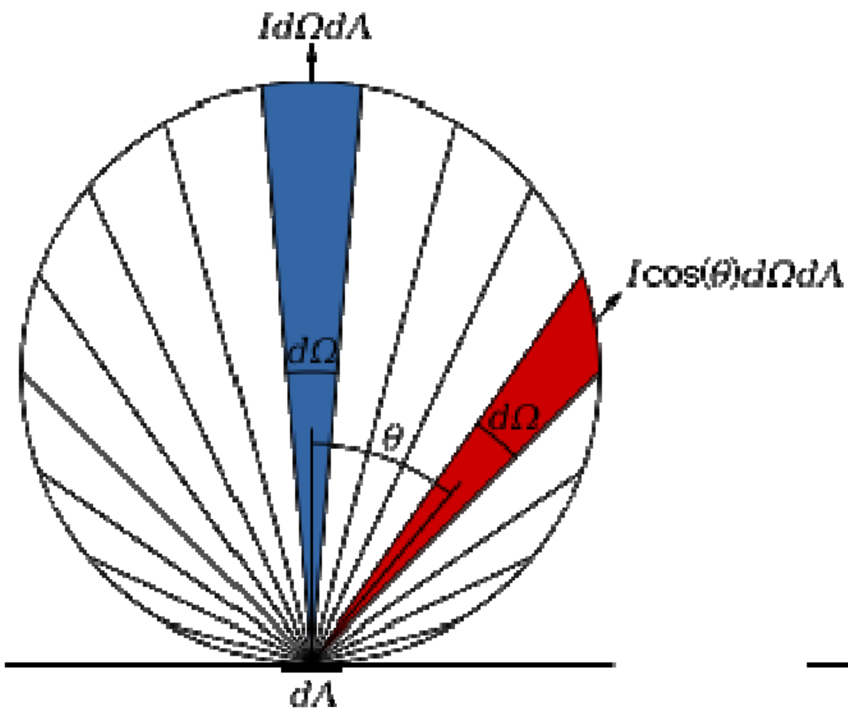


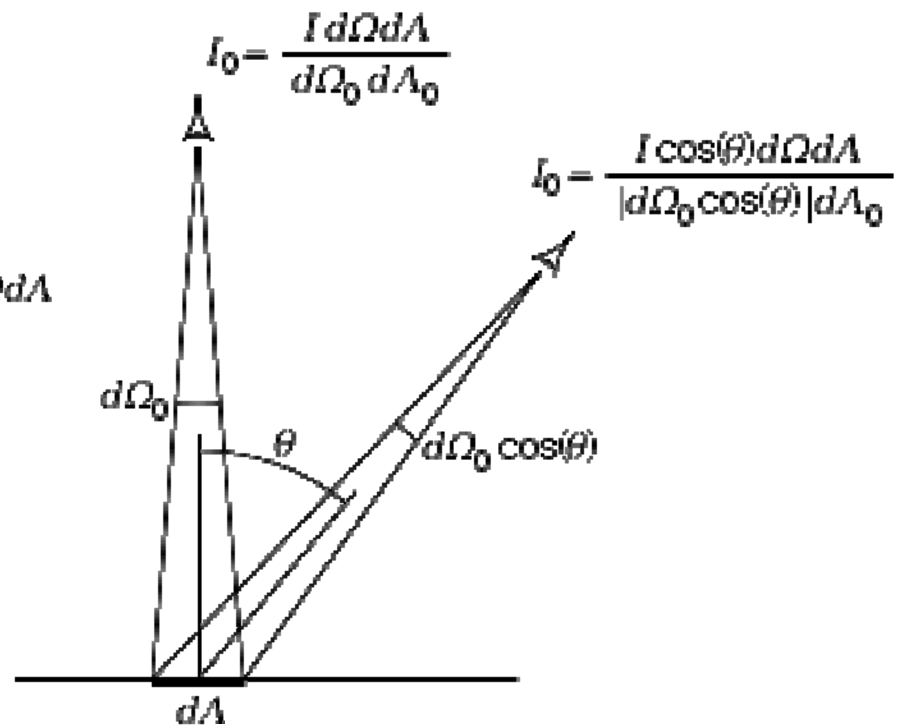


Diffuse reflection: Lambert's cosine law

Intensity does *not* depend on viewer angle.

- Amount of reflected light proportional to $\cos(\theta)$
- Visible solid angle also proportional to $\cos(\theta)$





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

Intensity and Surface Orientation

Intensity depends on illumination angle because less light comes in at oblique angles.

ρ = albedo

$\boldsymbol{S}$ = directional source

$\boldsymbol{N}$ = surface normal

I = reflected intensity

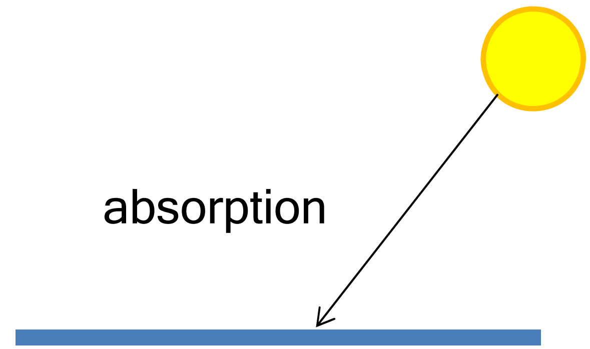
$$I(x) = \rho(x)(\boldsymbol{S} \cdot \boldsymbol{N}(x))$$

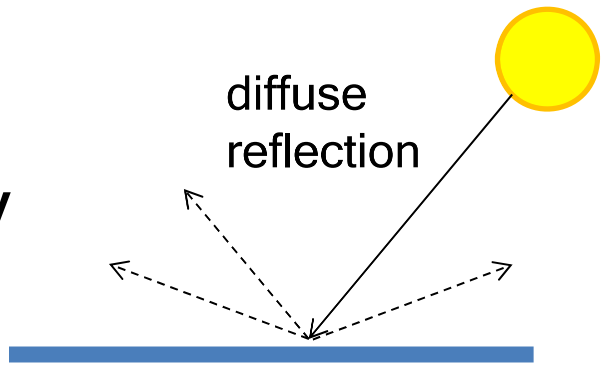
Slide: Forsyth

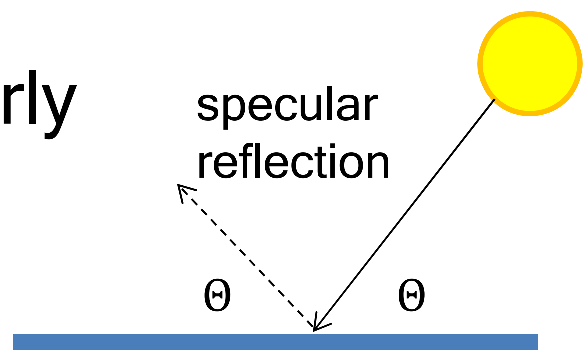

1
2

Recap

- When light hits a typical surface
  - Some light is absorbed (1-ρ)
    - More absorbed for low albedos
  - Some light is reflected diffusely
    - Independent of viewing direction
  - Some light is reflected specularly
    - Light bounces off (like a mirror), depends on viewing direction

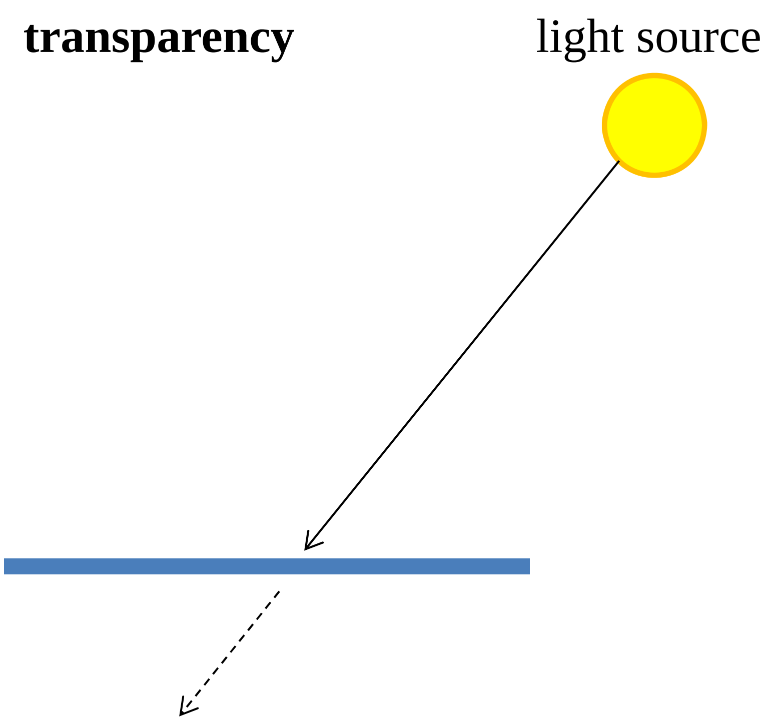




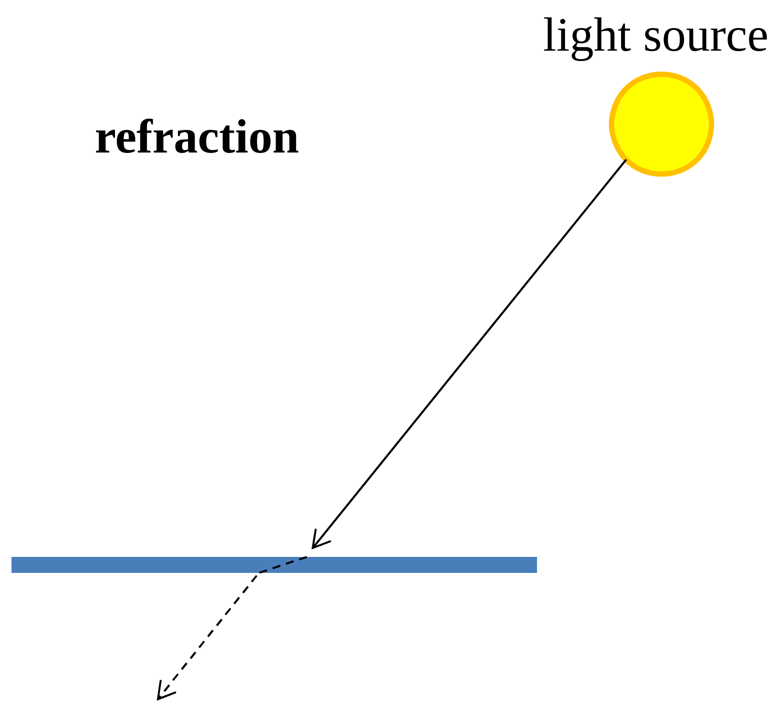


Other possible effects

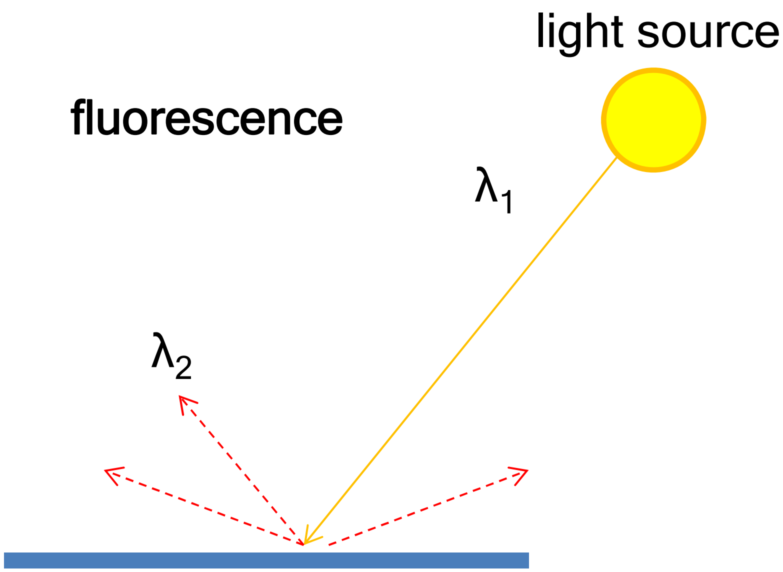
transparency



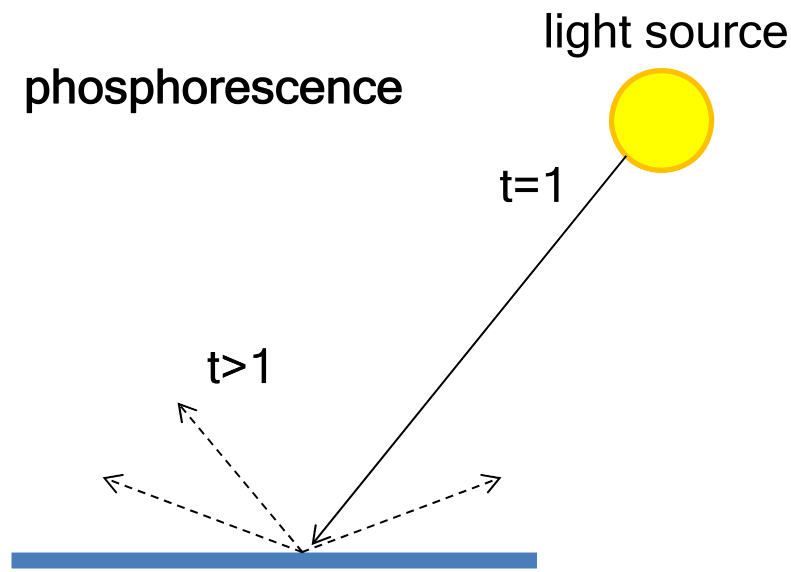
refraction



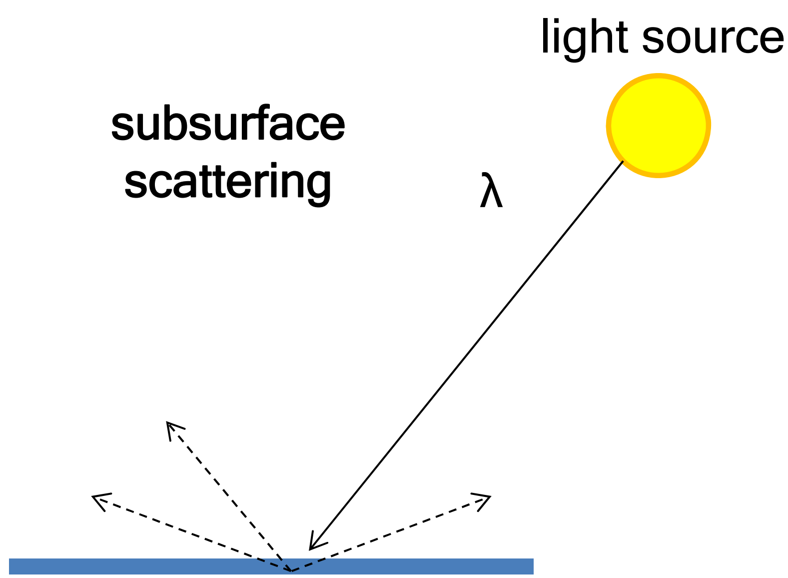
fluorescence



phosphorescence

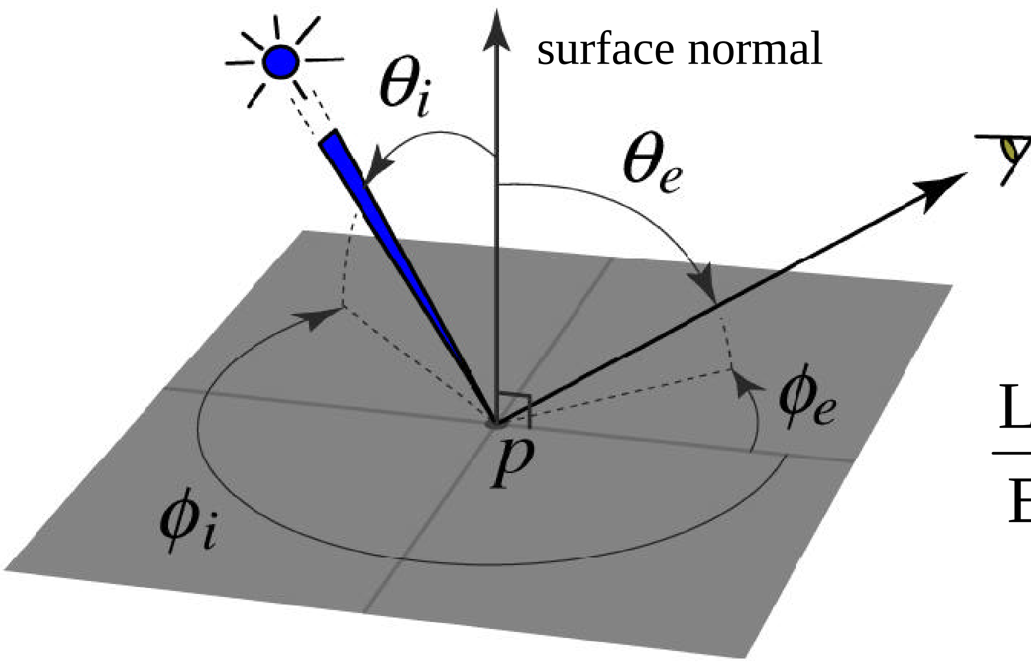


subsurface scattering



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
  - Ratio of measured outgoing radiance in direction (θ_e,ϕ_e) to irradiance from direction (θ_i,ϕ_i)
  - Reciprocal



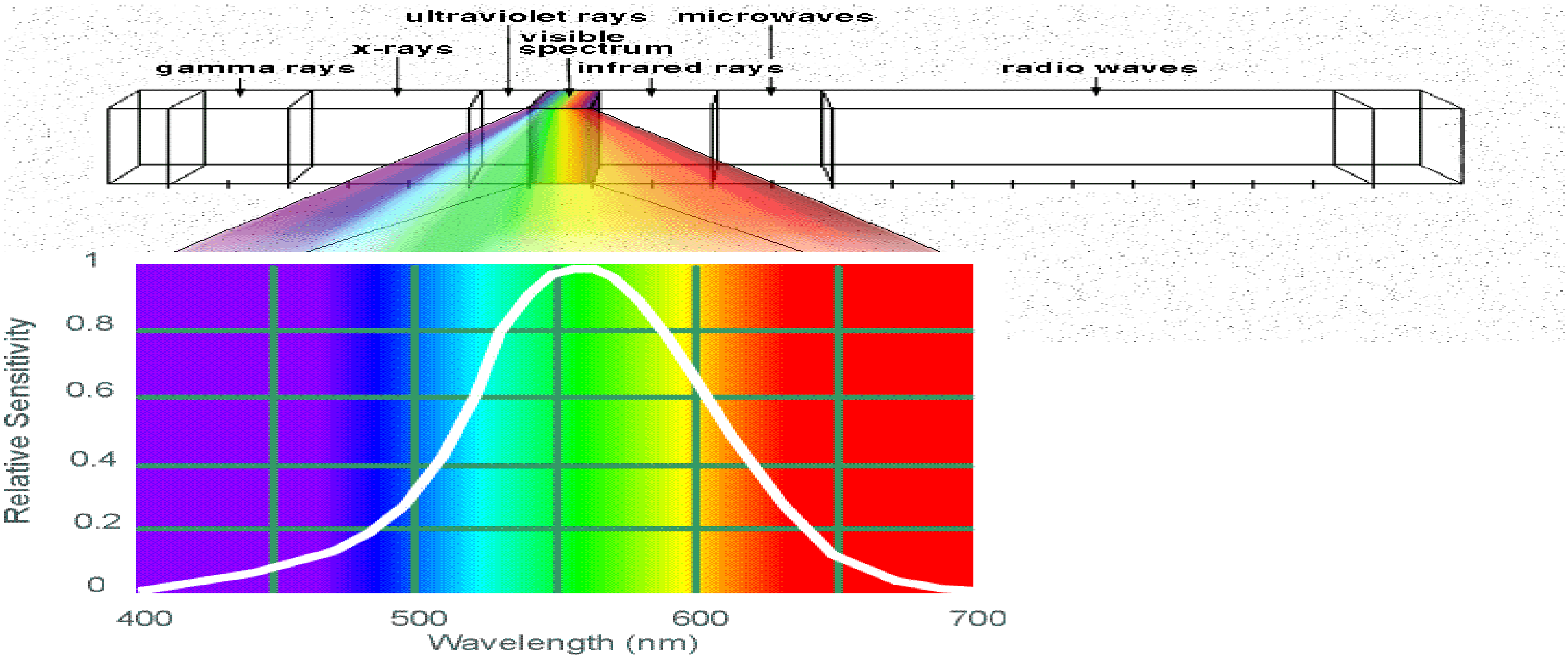
$$\rho(\theta_i,\phi_i,\theta_e,\phi_e;\lambda)=$$

$$\frac{\mathrm{L_e}(\boldsymbol{\theta}_\mathrm{e},\boldsymbol{\phi}_\mathrm{e})}{\mathrm{E_i}(\boldsymbol{\theta}_\mathrm{i},\boldsymbol{\phi}_\mathrm{i})}=\frac{\mathrm{L_e}(\boldsymbol{\theta}_\mathrm{e},\boldsymbol{\phi}_\mathrm{e})}{\mathrm{L_i}(\boldsymbol{\theta}_\mathrm{i},\boldsymbol{\phi}_\mathrm{i})\cos\boldsymbol{\theta}_\mathrm{i}\,\mathrm{d}\boldsymbol{\omega}}$$

Slide credit: S. Savarese

Color

Light is composed of a spectrum of wavelengths

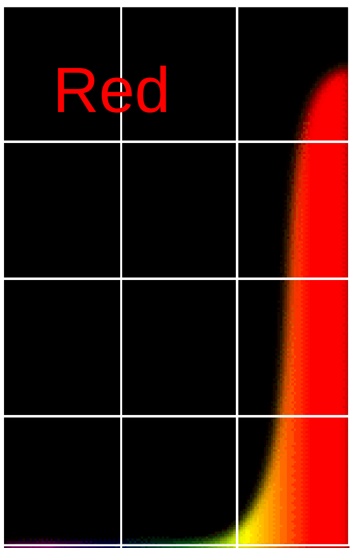


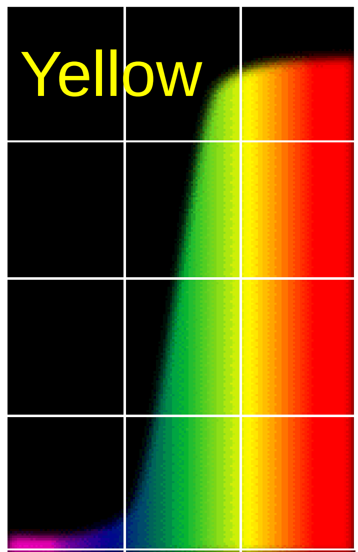
Human Luminance Sensitivity Function

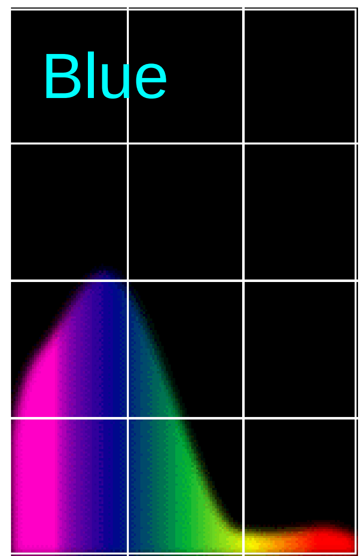
Slide Credit: Efros

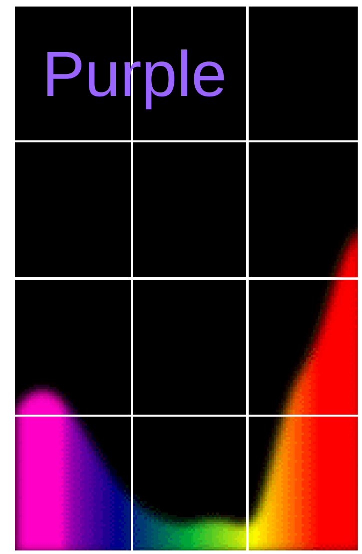
http://www.yorku.ca/eye/photopik.htm

Some examples of the reflectance spectra of surfaces



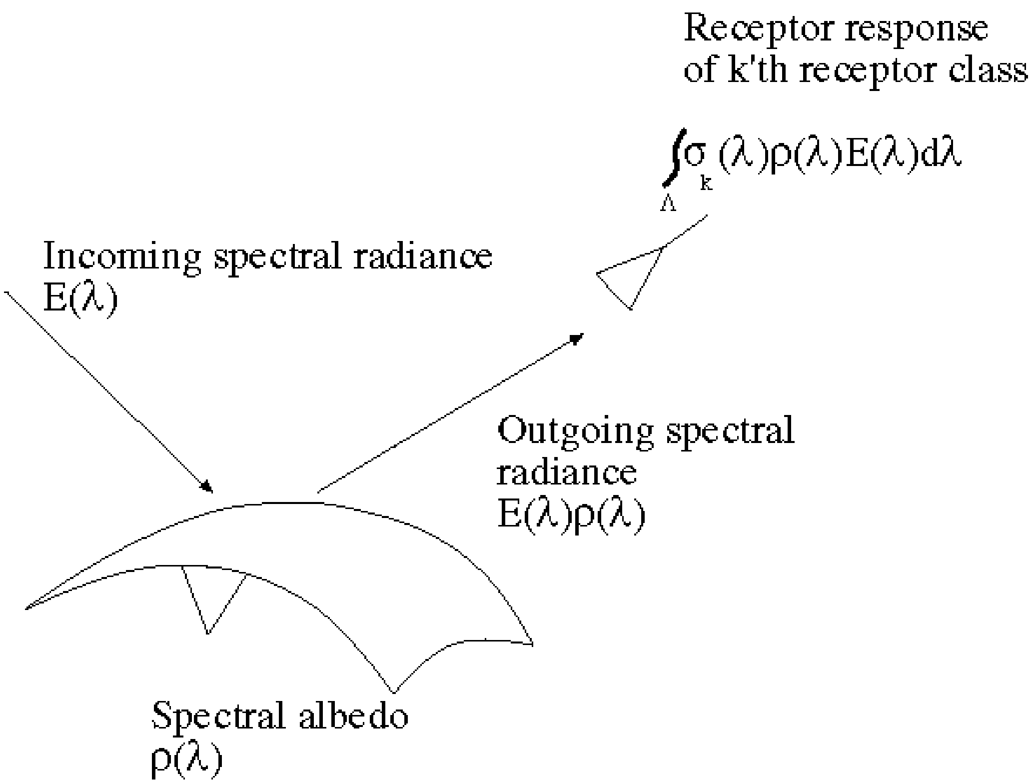






The color of objects

- Colored light arriving at the camera involves two effects
  - The color of the light source (illumination + inter-reflections)
  - The color of the surface

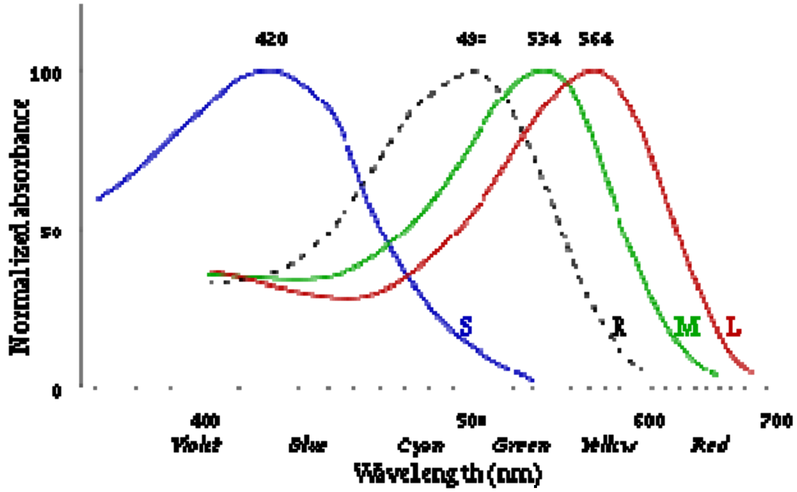


Slide: Forsyth

Why RGB?

If light is a spectrum, why are images RGB?

Human color receptors

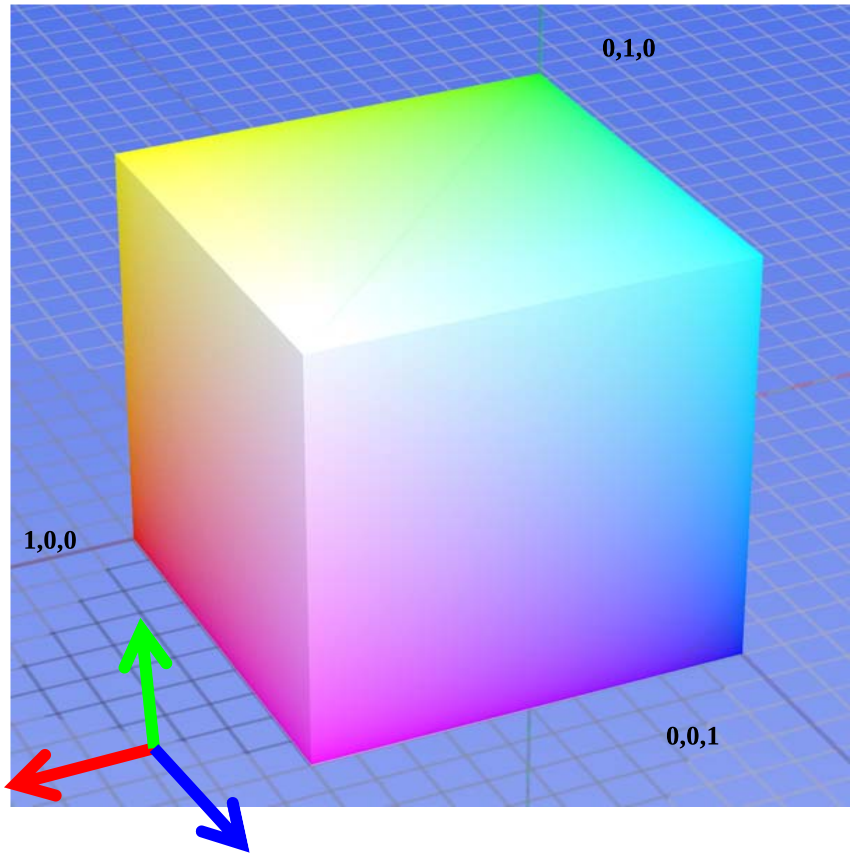


- Long (red), Medium (green), and Short (blue) cones, plus intensity rods
- Fun facts
  - "M" and "L" on the X-chromosome
    - That's why men are more likely to be color blind
  - "L" has high variation, so some women are tetrachromatic
  - Some animals have 1 (night animals), 2 (e.g., dogs), 4 (fish, birds), 5 (pigeons, some reptiles/amphibians), or even 12 (mantis shrimp) types of cones

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

Color spaces: RGB

Default color space



R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)

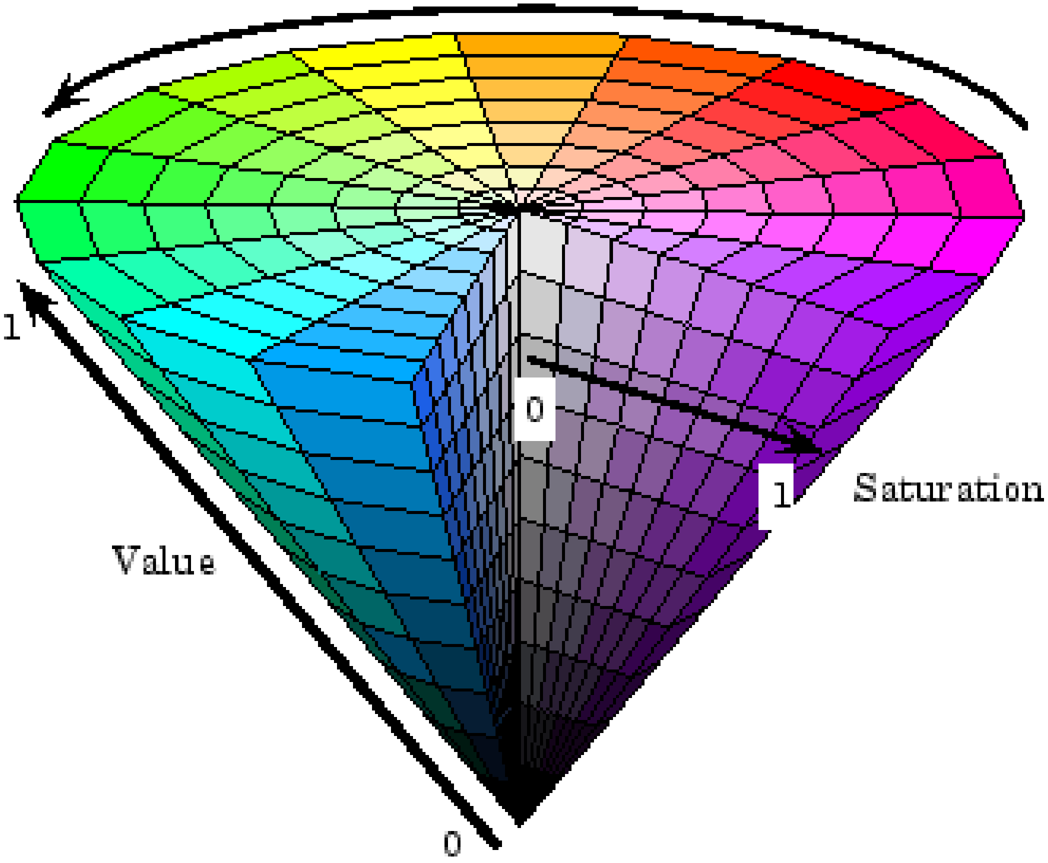
Some drawbacks

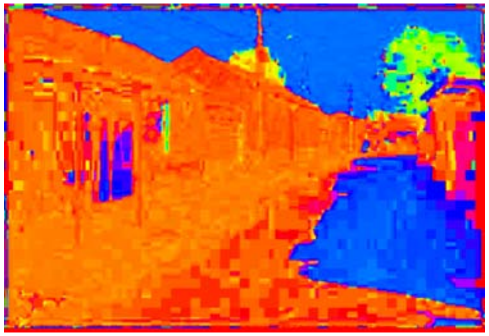
- Strongly correlated channels
- Non-perceptual

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Color spaces: HSV

Intuitive color space





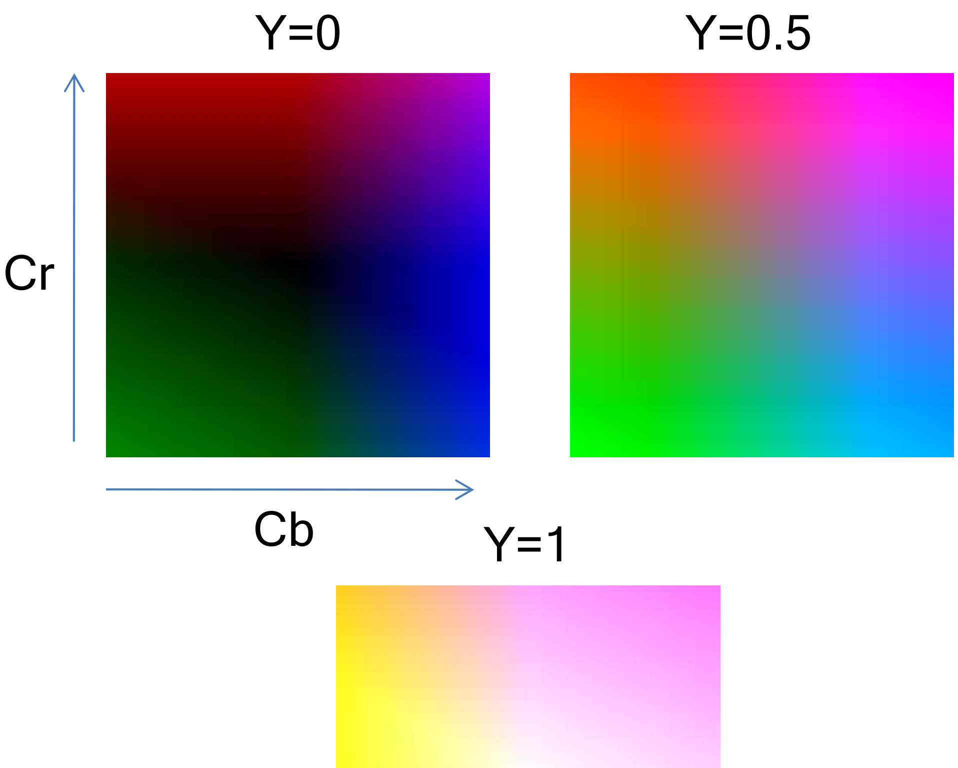
H
(S=1,V=1)

S
(H=1,V=1)

V
(H=1,S=0)

Color spaces: YCbCr

Fast to compute, good for compression, used by TV



$$Y' = 16 + \frac{65.738 \cdot R'_D}{256} + \frac{129.057 \cdot G'_D}{256} + \frac{25.064 \cdot B'_D}{256}$$

$$C_B = 128 + \frac{-37.945 \cdot R'_D}{256} - \frac{74.494 \cdot G'_D}{256} + \frac{112.439 \cdot B'_D}{256}$$

$$C_R = 128 + \frac{112.439 \cdot R'_D}{256} - \frac{94.154 \cdot G'_D}{256} - \frac{18.285 \cdot B'_D}{256}$$

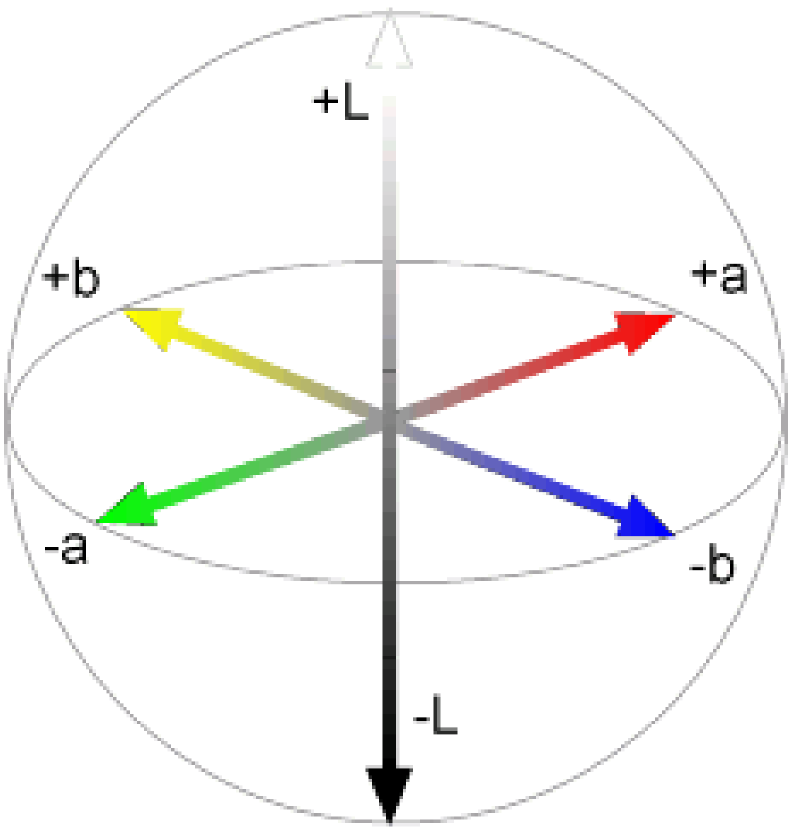
Y
(Cb=0.5,Cr=0.5)

Cb
(Y=0.5,Cr=0.5)

Cr
(Y=0.5,Cb=05)

Color spaces: CIE L*a*b*

"Perceptually uniform" color space



Luminance = brightness
Chrominance = color

L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Which contains more information?

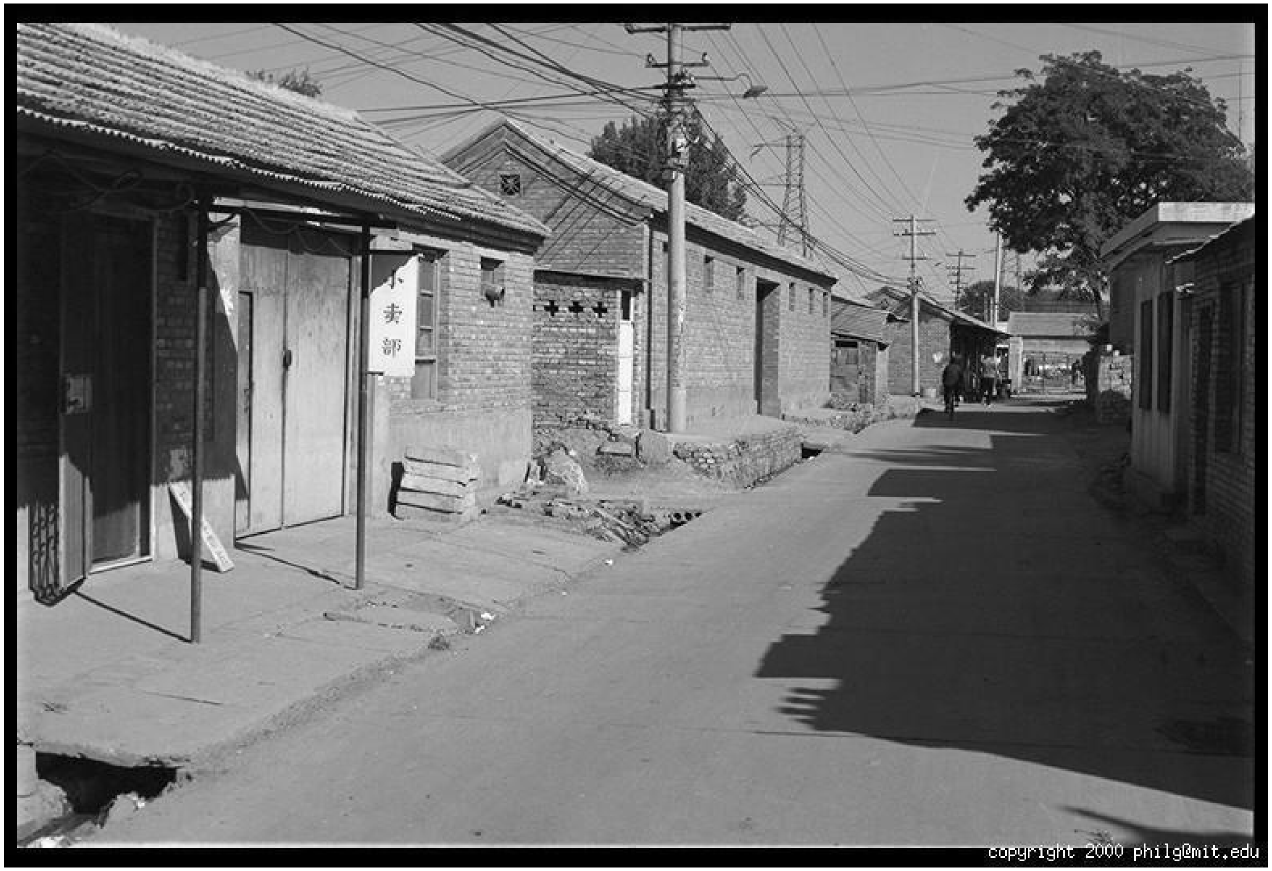
(a) **intensity** (1 channel)

(b) **chrominance** (2 channels)

Most information in intensity

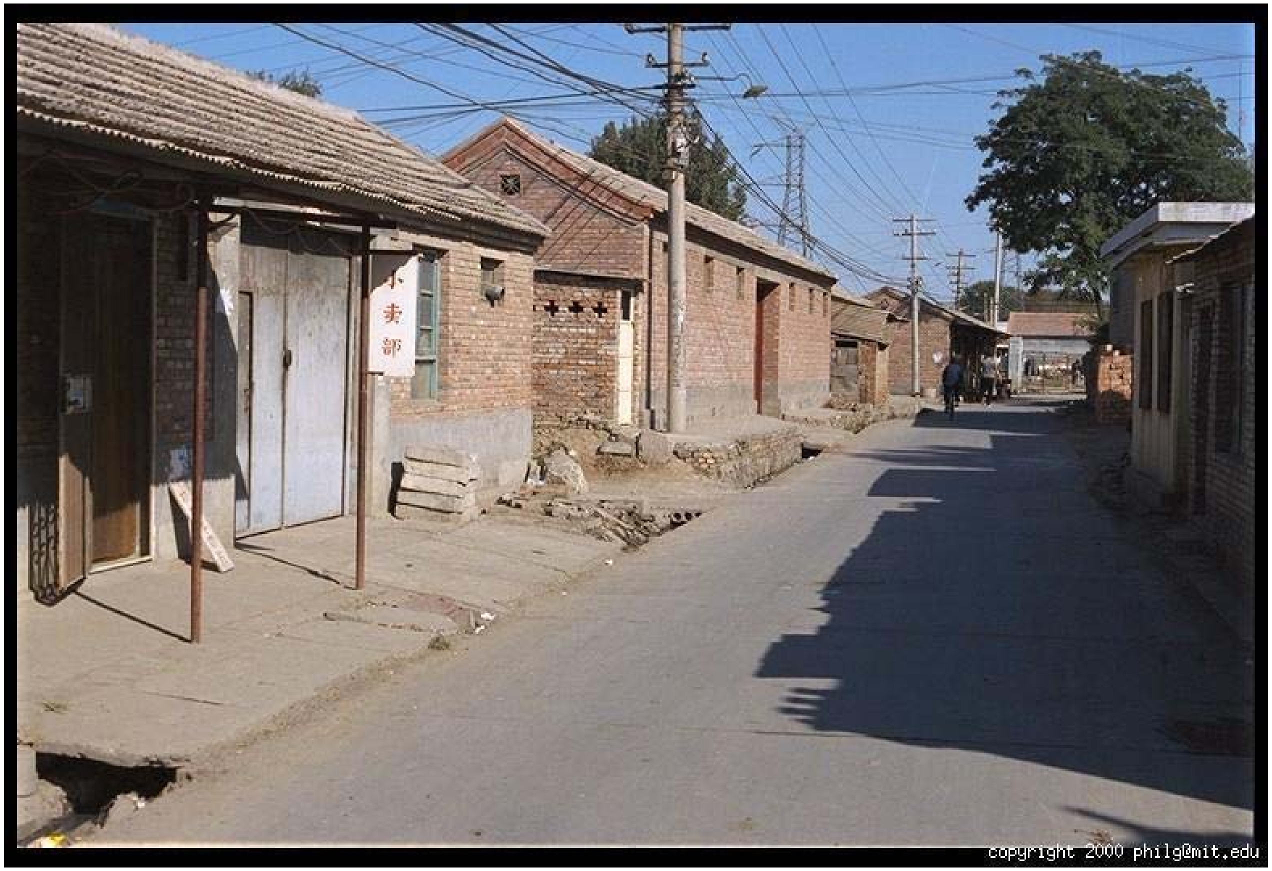
Only color shown – constant intensity

Most information in intensity



Only intensity shown – constant color

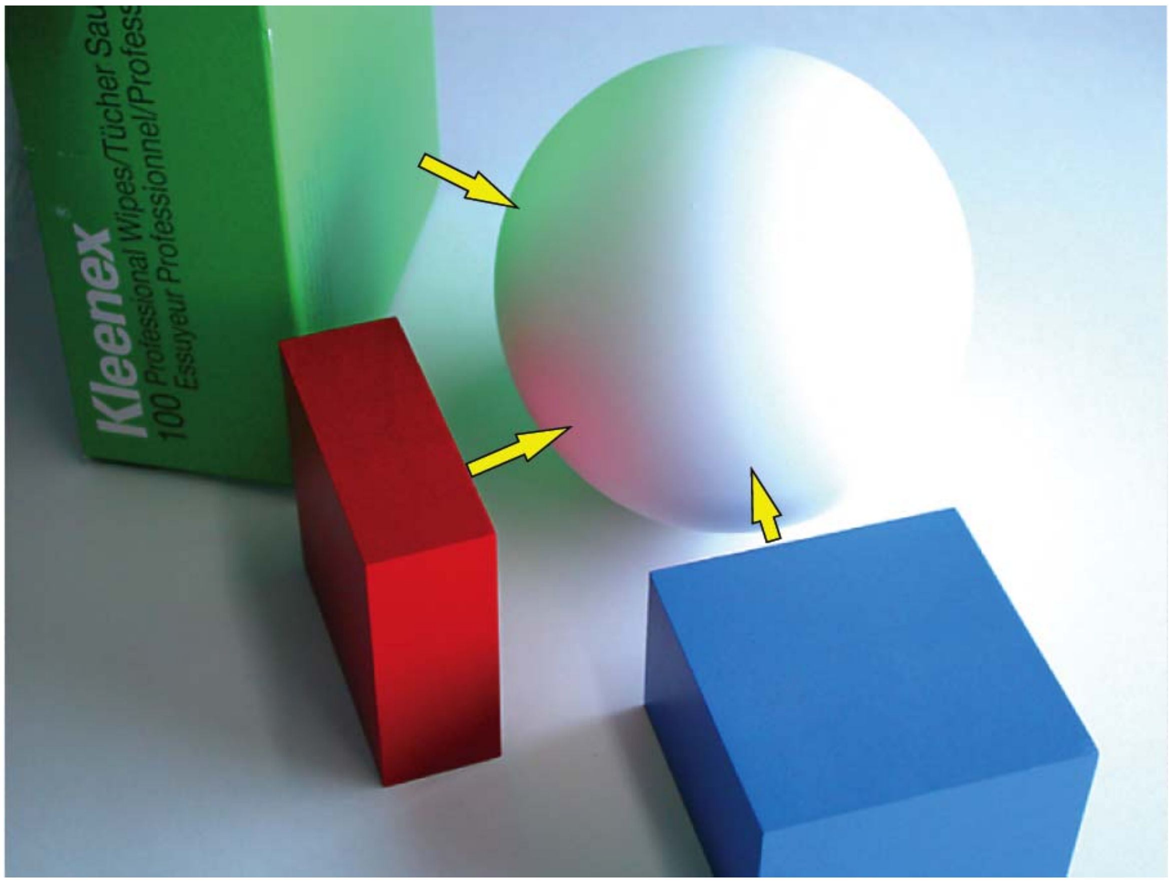
Most information in intensity



Original image

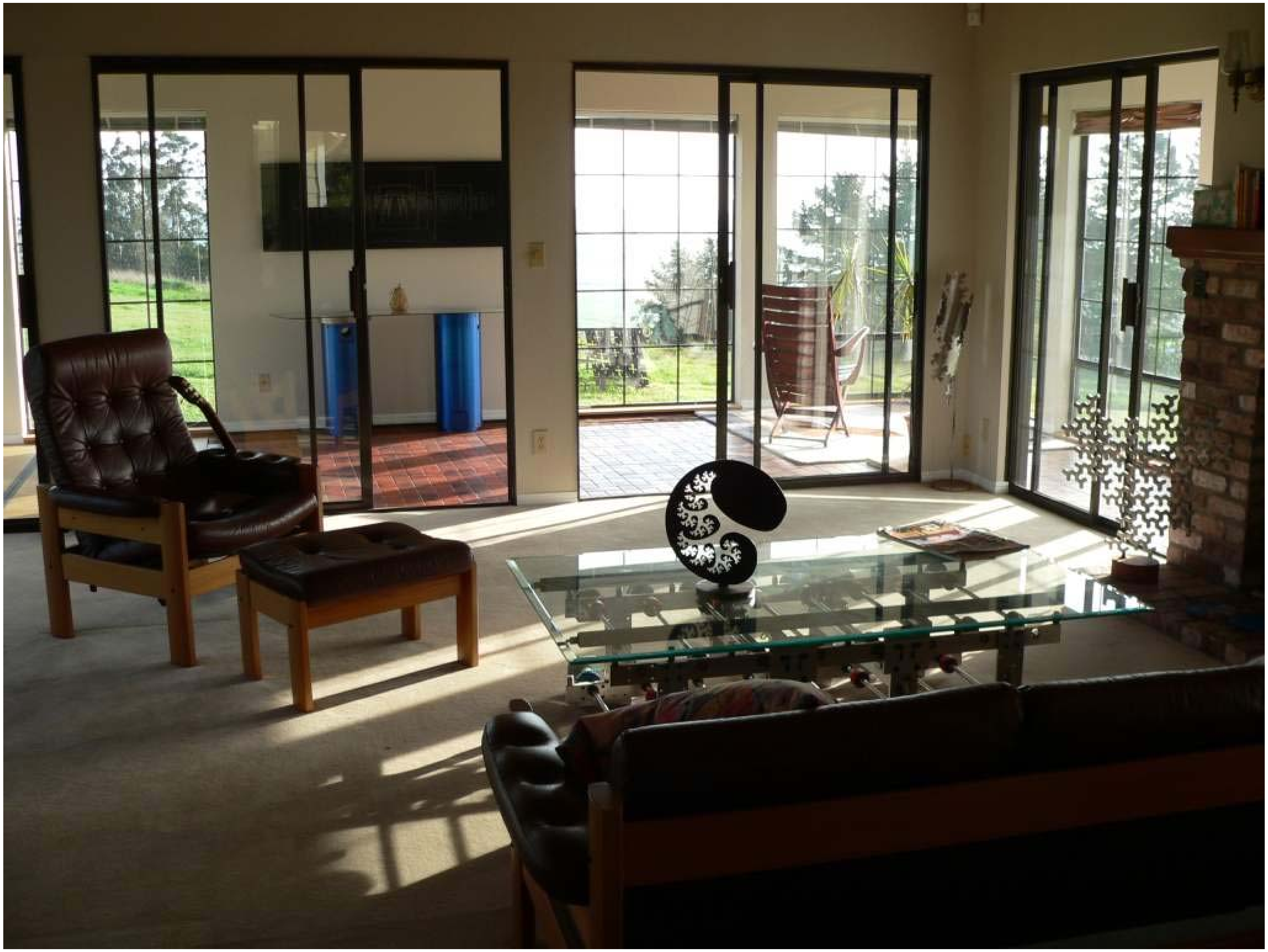
So far: light→surface→camera

- Called a local illumination model
- But much light comes from surrounding surfaces

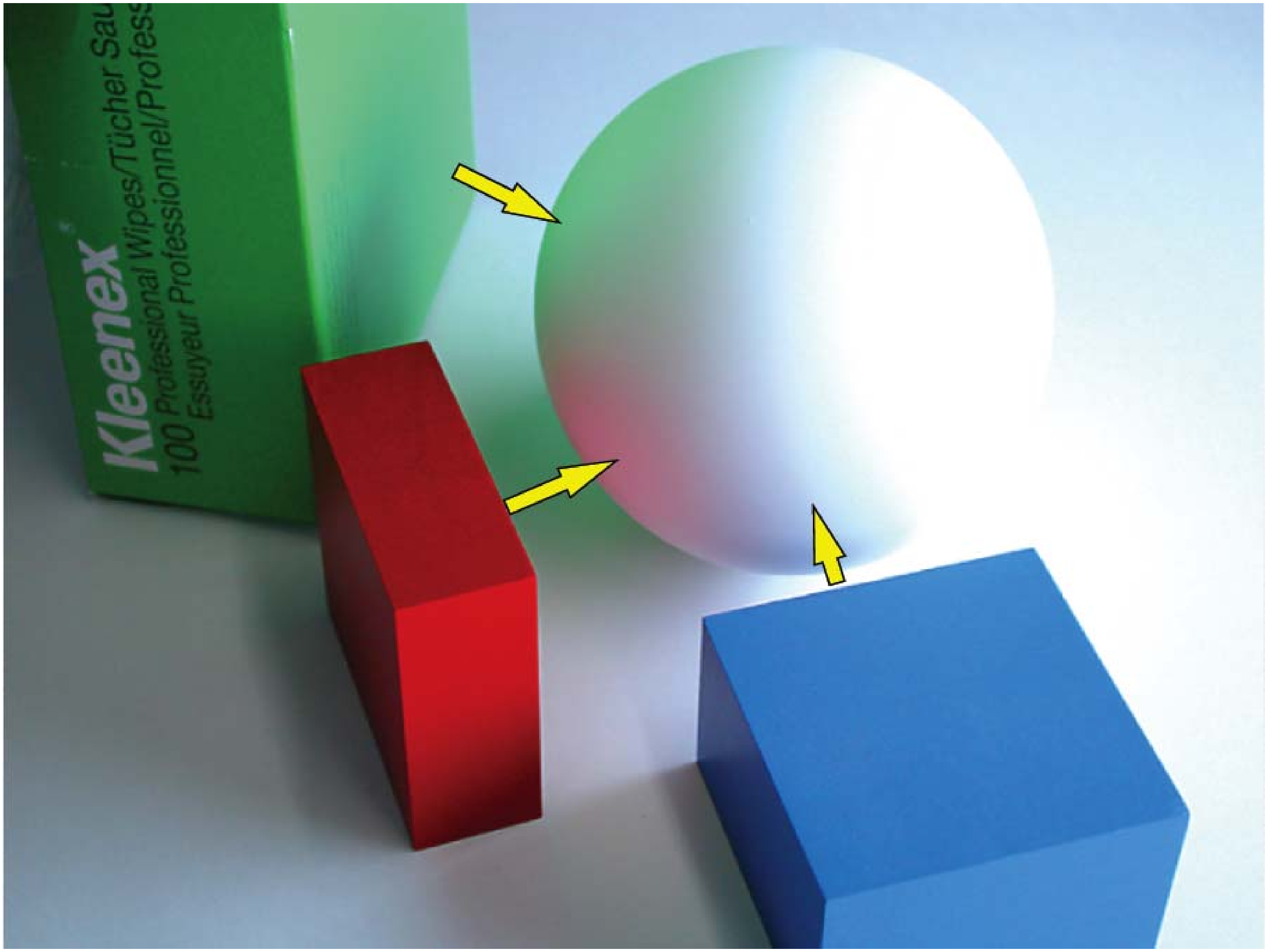


From Koenderink slides on image texture and the flow of light

Inter-reflection is a major source of light

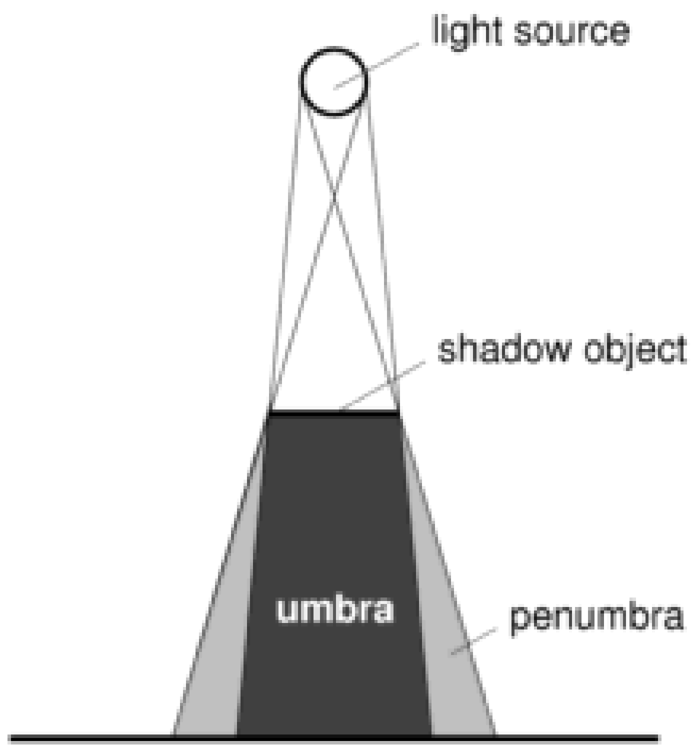


Inter-reflection affects the apparent color of objects

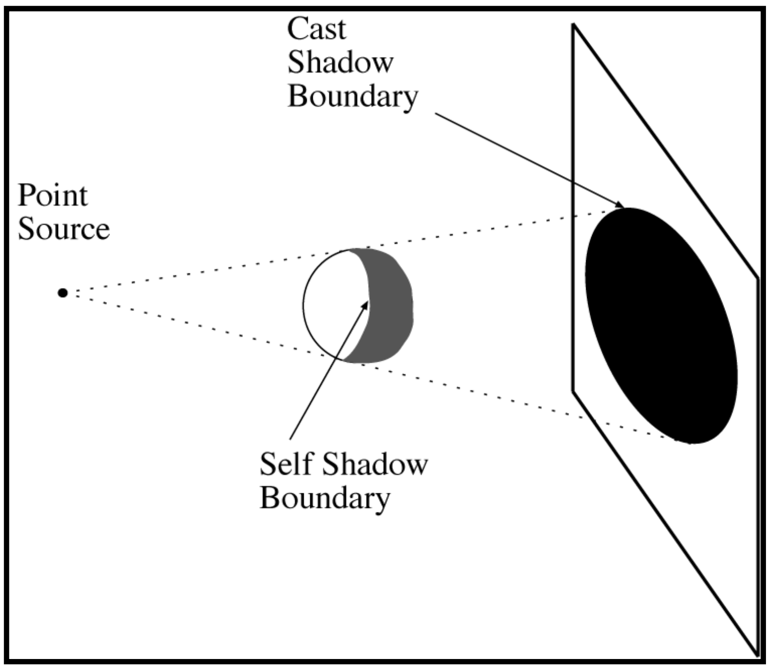


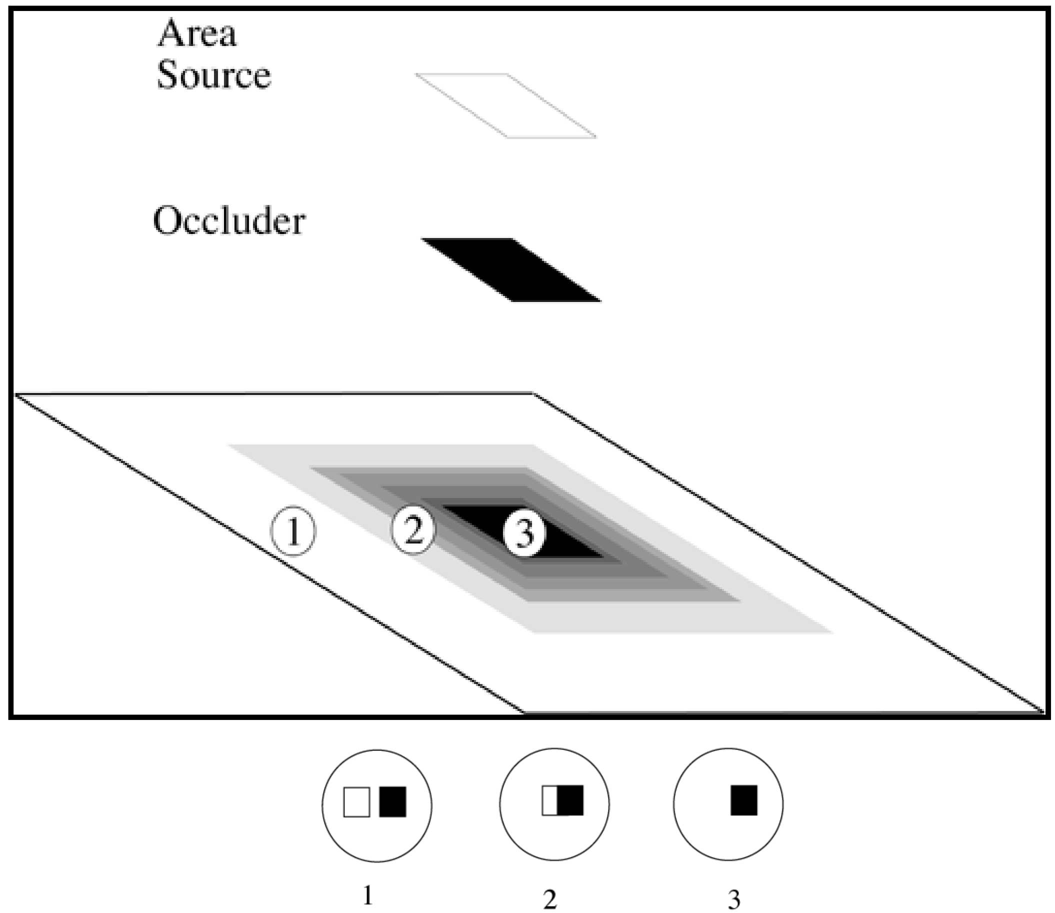
Scene surfaces also cause shadows

- Shadow: reduction in intensity due to a blocked source



Shadows


Cast
Shadow
Boundary
Point
Source
Self Shadow
Boundary

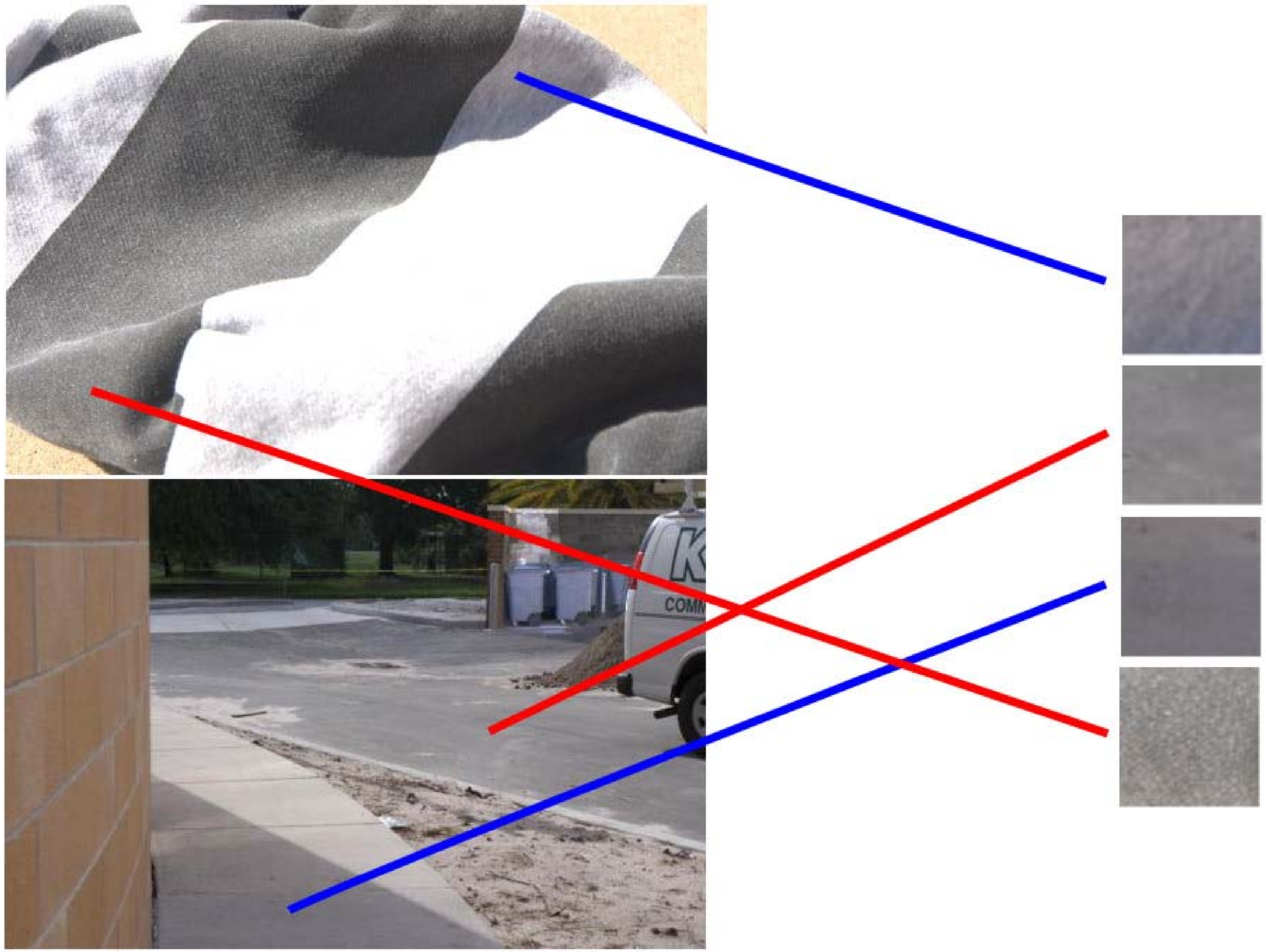

Area
Source
Occluder
1
2
3
1
2
3

Models of light sources

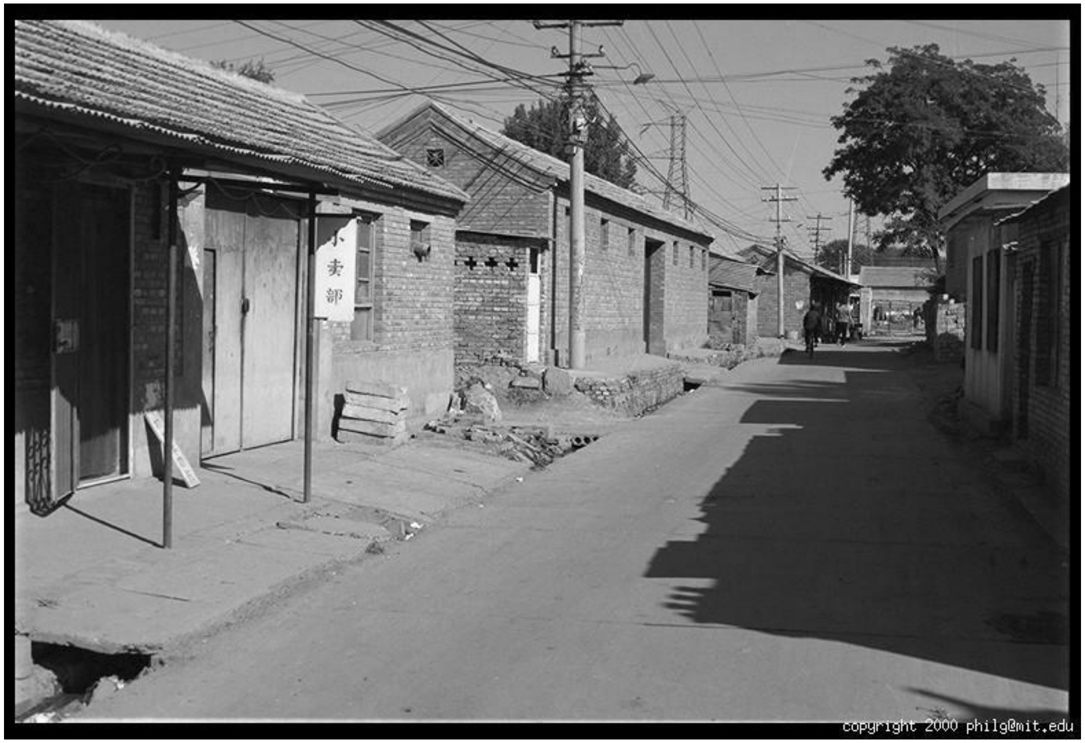
- Distant point source
  - One illumination direction
  - E.g., sun
- Area source
  - E.g., white walls, diffuser lamps, sky
- Ambient light
  - Substitute for dealing with interreflections
- Global illumination model
  - Account for interreflections in modeled scene

The plight of the poor pixel

- A pixel's brightness is determined by
  - Light source (strength, direction, color)
  - Surface orientation
  - Surface material and albedo
  - Reflected light and shadows from surrounding surfaces
  - Gain on the sensor
- A pixel's brightness tells us nothing by itself

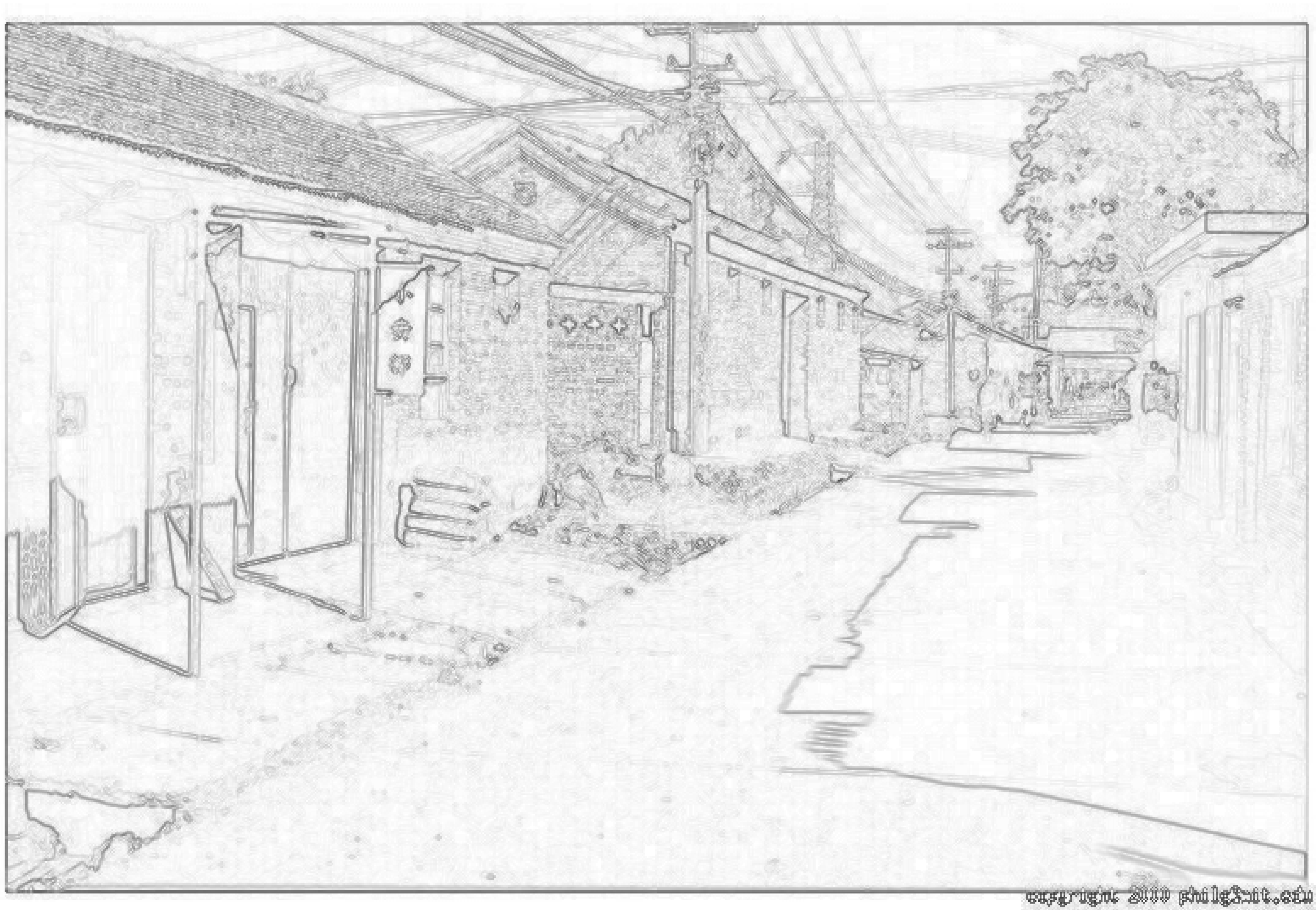


And yet we can interpret images…



- Key idea: for nearby scene points, most factors do not change much
- The information is mainly contained in *local differences* of brightness

Darkness = Large Difference in Neighboring Pixels



What is this?


HONDA

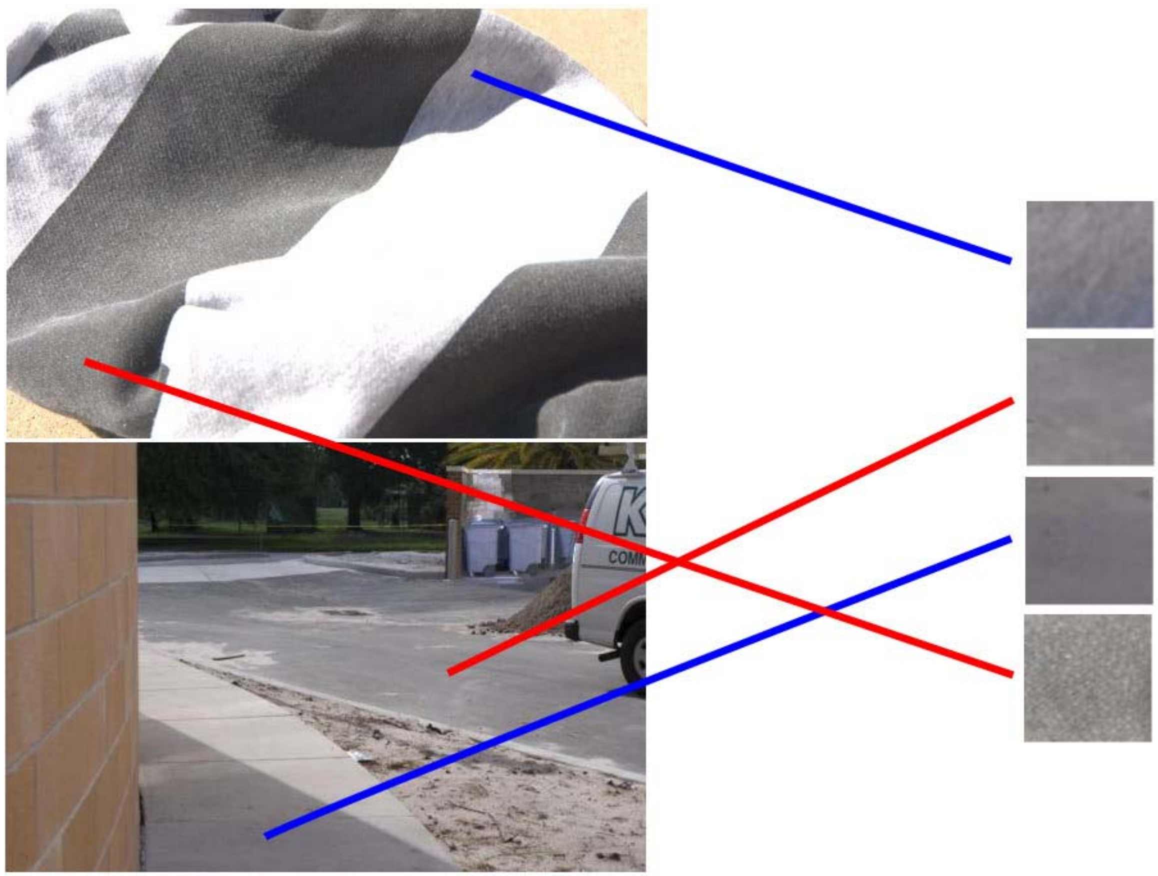
What differences in intensity tell us about shape

- Changes in surface normal
- Texture
- Proximity
- Indents and bumps
- Grooves and creases

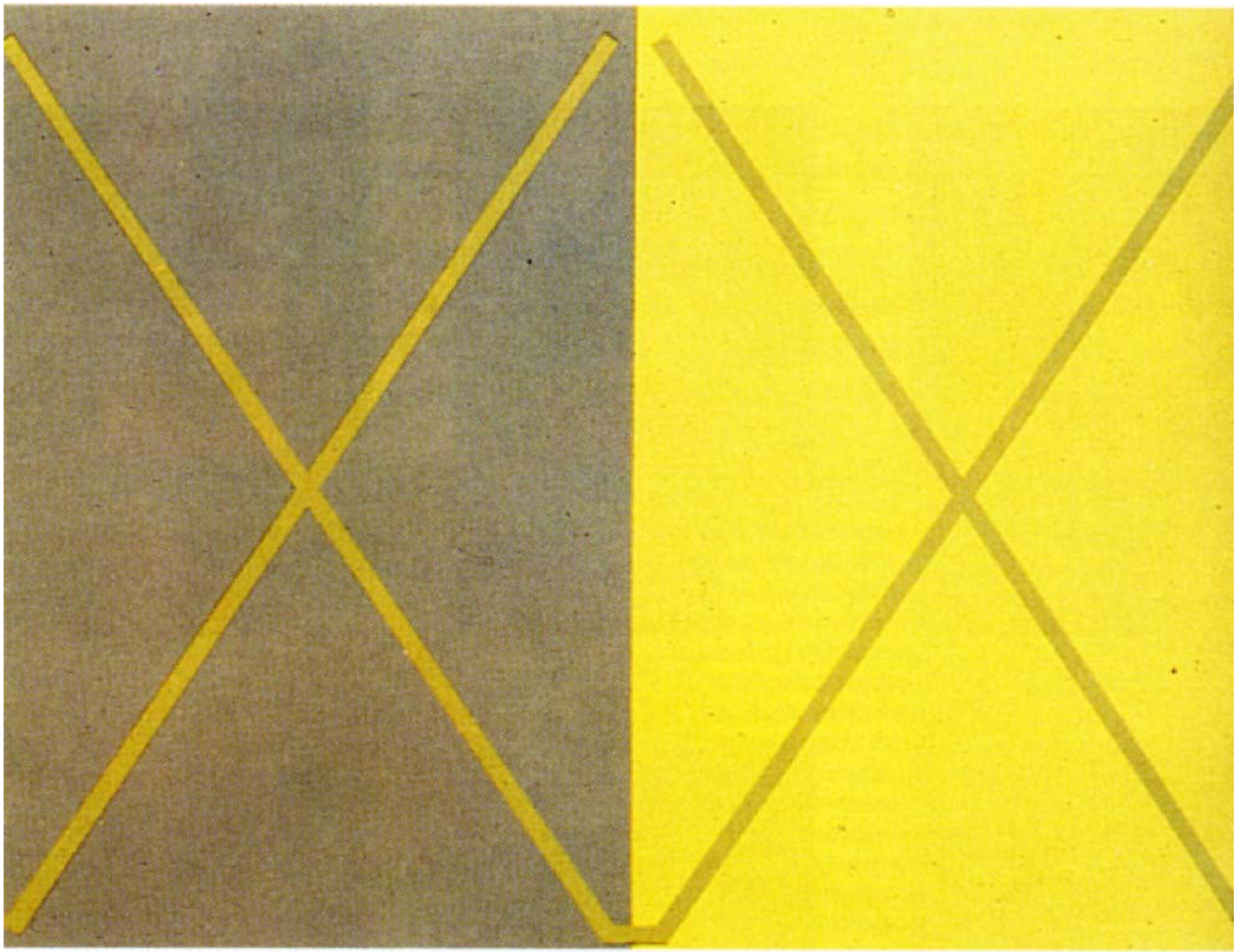
Photos Koenderink slides on image texture and the flow of light

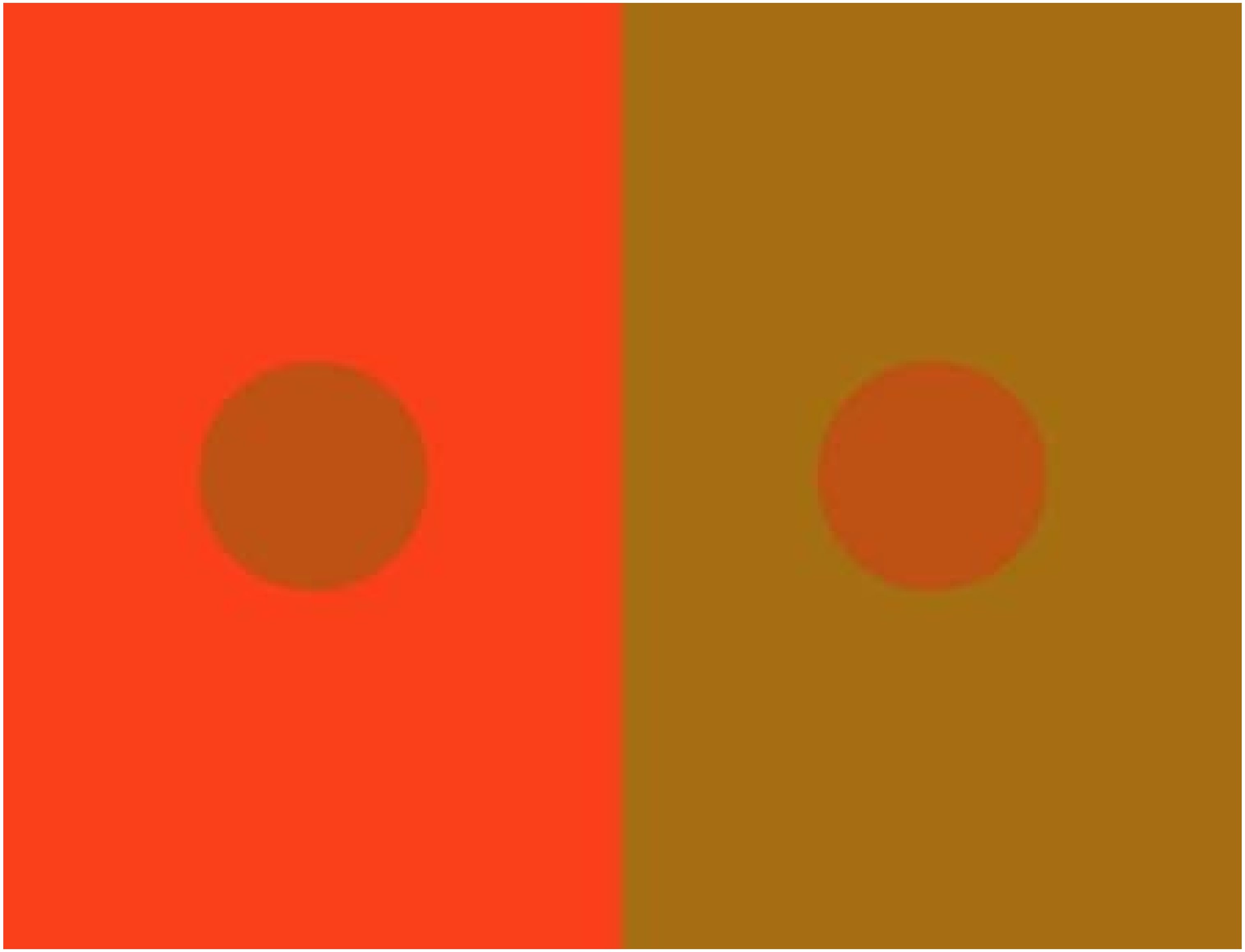
Color constancy

- Interpret surface in terms of albedo or “true color”, rather than observed intensity
  - Humans are good at it
  - Computers are not nearly as good



One source of constancy: local comparisons

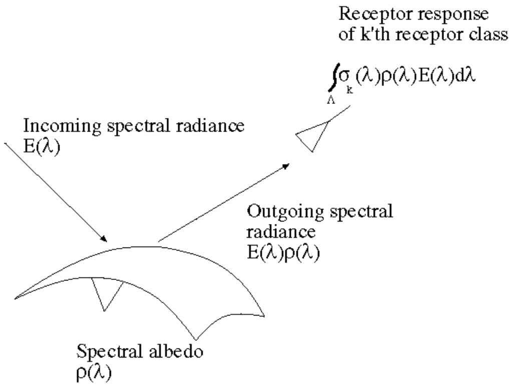


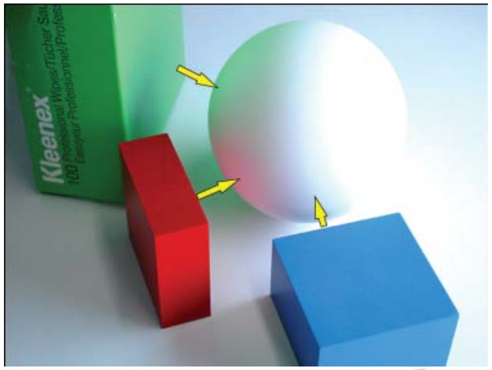


http://www.echalk.co.uk/amusements/OpticalIllusions/colourPerception/colourPerception.html

Things to remember

- Important terms: diffuse/specular reflectance, albedo, umbra/penumbra
- Observed intensity depends on light sources, geometry/material of reflecting surface, surrounding objects, camera settings
- Objects cast light and shadows on each other
- Differences in intensity are primary cues for shape

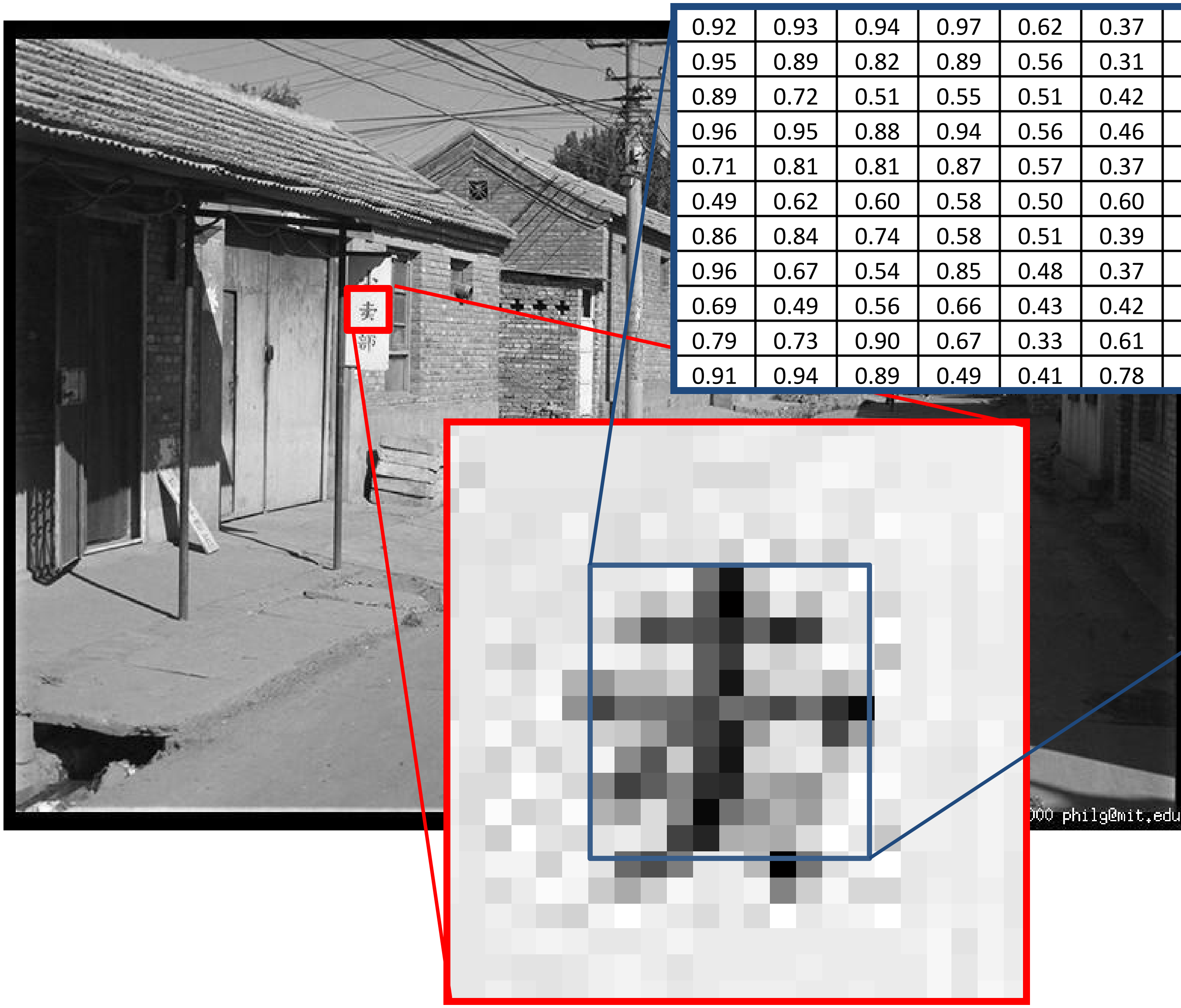




Pixels and Linear Filters

Slides by D. Hoiem

The raster image (pixel matrix)



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

Image filtering

- Image filtering: for each pixel, compute function of local neighborhood and output a new value
  - Same function applied at each position
  - Output and input image are typically the same size

Image filtering

- Linear filtering: function is a weighted sum/difference of pixel values

- Really important
  - Enhance images
    - Denoise, smooth, increase contrast, etc.
  - Extract information from images
    - Texture, edges, distinctive points, etc.
  - Detect patterns
    - Template matching

Example: box filter

$$g[\cdot\,,\cdot\,]$$

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

Slide credit: David Lowe (UBC)

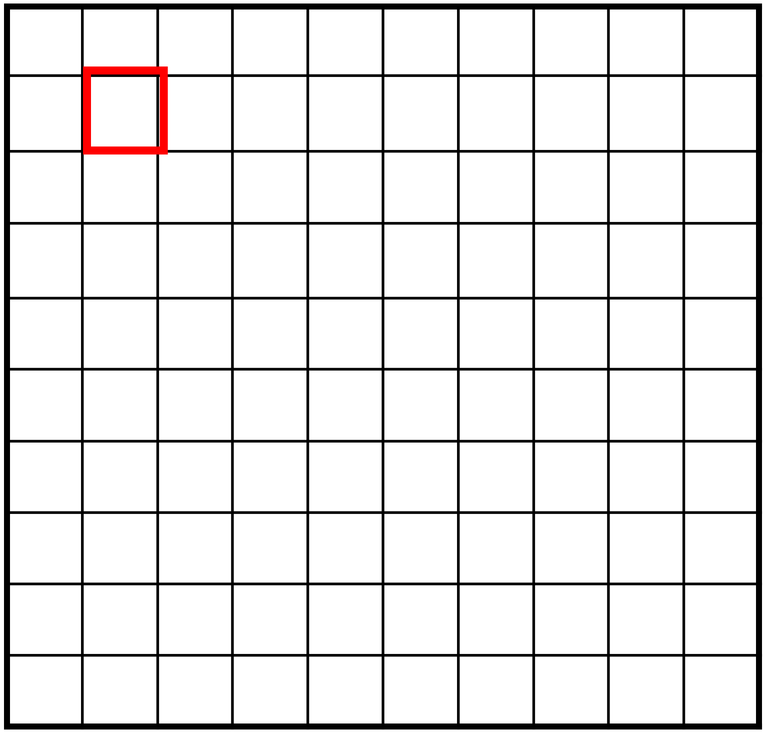
Image filtering

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$



$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot\,,\cdot\,]$ $\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\; f[m+k, n+l]$$

Credit: S. Seitz

Box Filter

$$g[\cdot\,,\cdot\,]$$

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

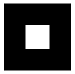
What does it do?

- Replaces each pixel with an average of its neighborhood

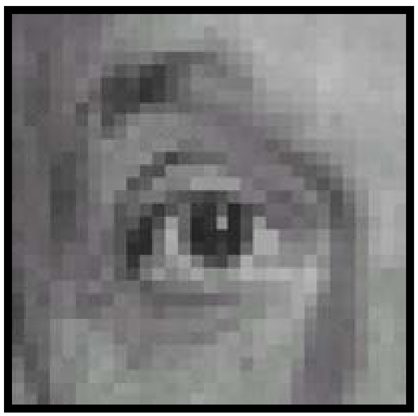
- Achieve smoothing effect (remove sharp features)

Slide credit: David Lowe (UBC)

Smoothing with box filter



Practice with linear filters



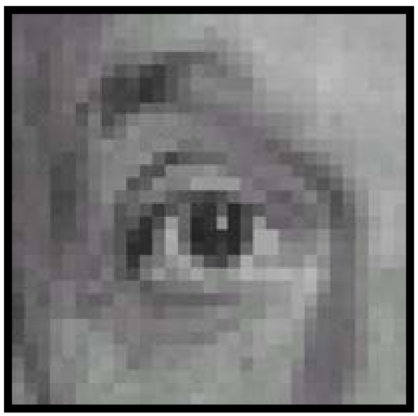
Original

0	0	0
0	1	0
0	0	0

?

Source: D. Lowe

Practice with linear filters



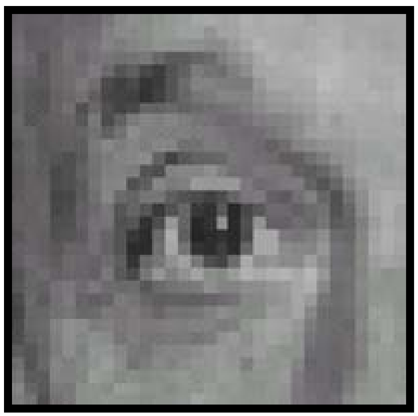
Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Source: D. Lowe

Practice with linear filters



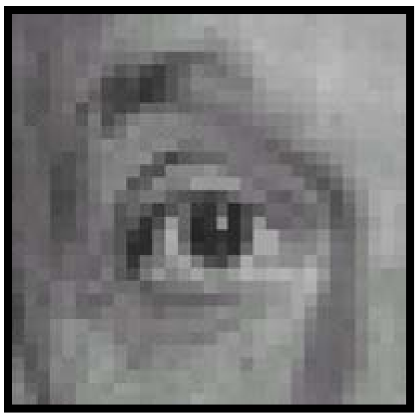
Original

0	0	0
0	0	1
0	0	0

?

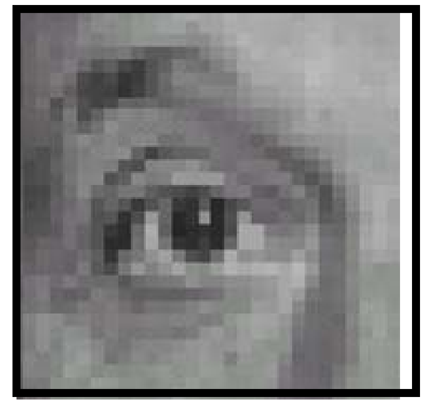
Source: D. Lowe

Practice with linear filters



Original

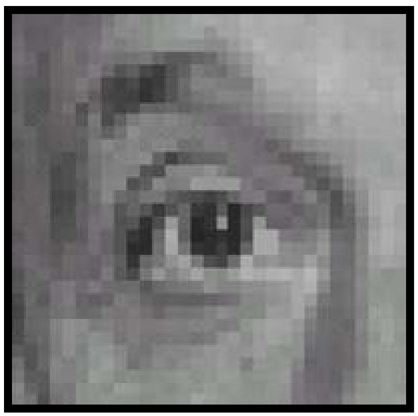
0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Practice with linear filters



Original

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

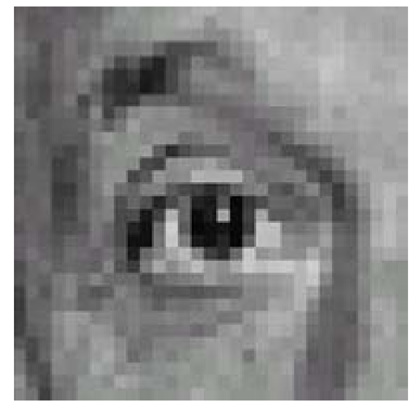
(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters

Original

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

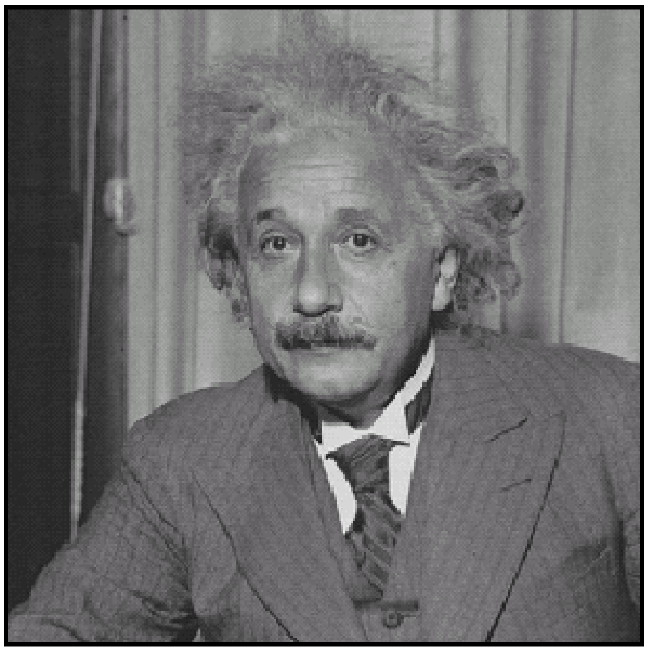


Sharpening filter

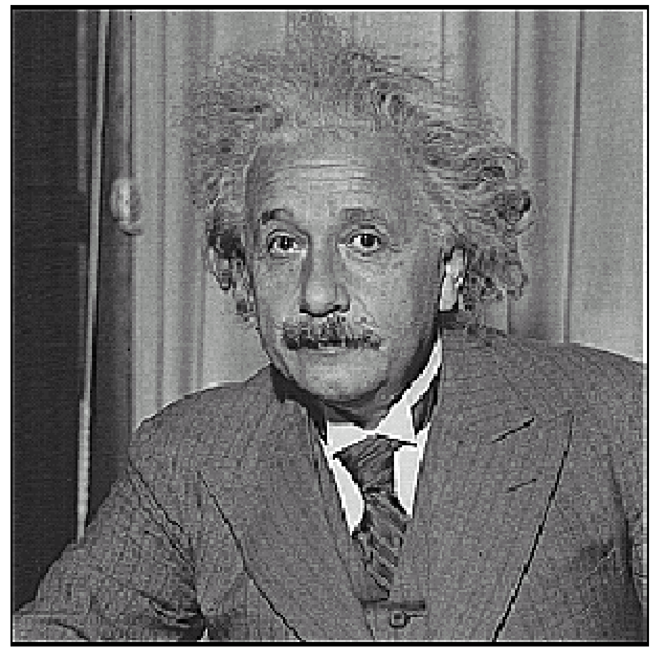
- Accentuates differences with local average

Source: D. Lowe

Sharpening



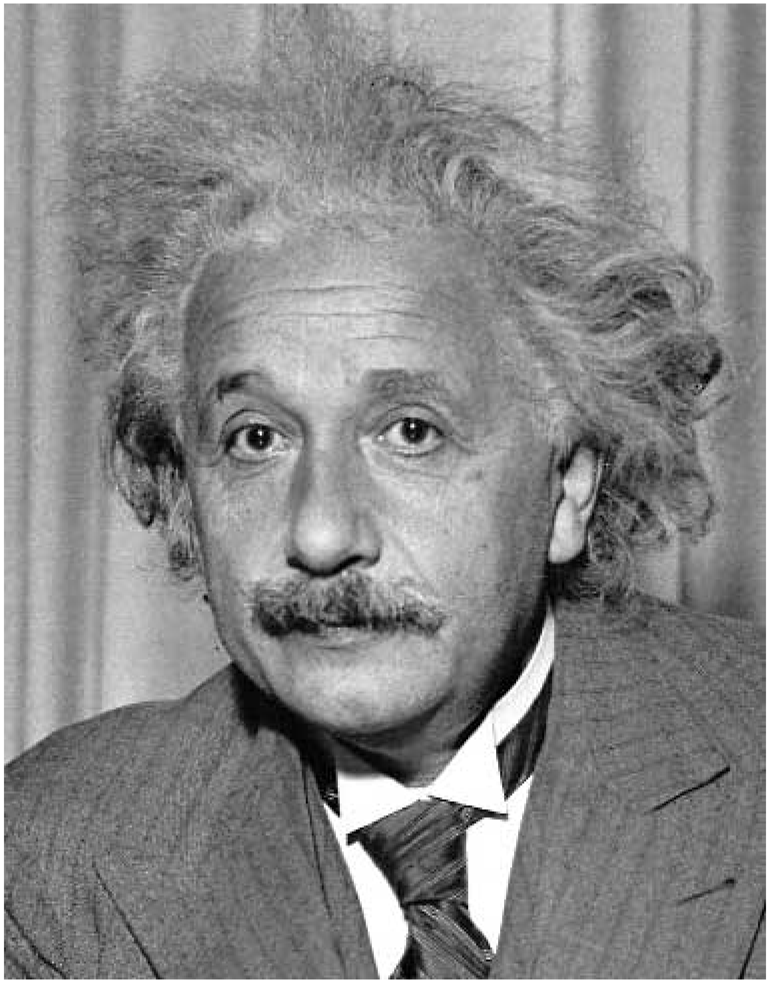
before



after

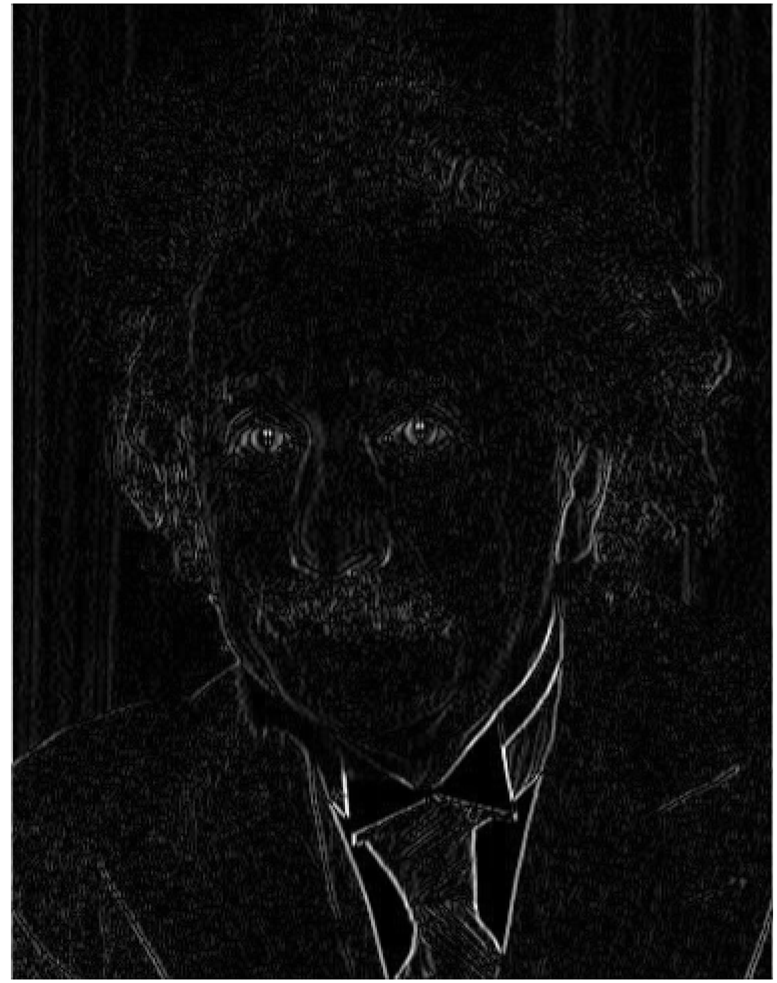
Source: D. Lowe

Other filters



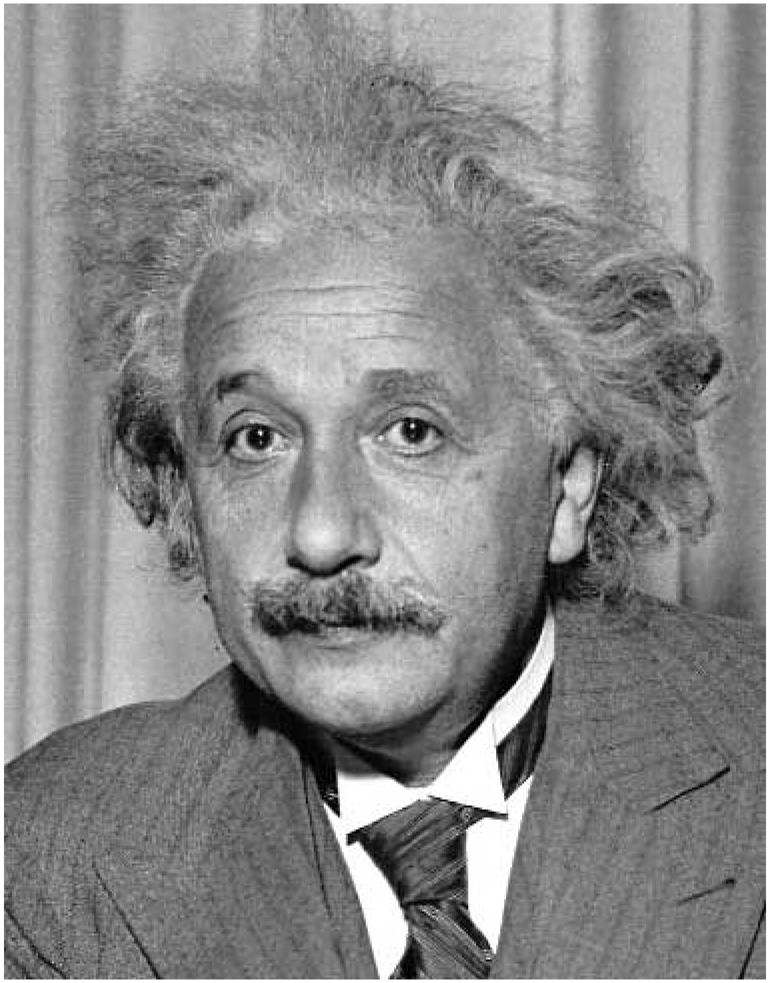
1	0	-1
2	0	-2
1	0	-1

Sobel



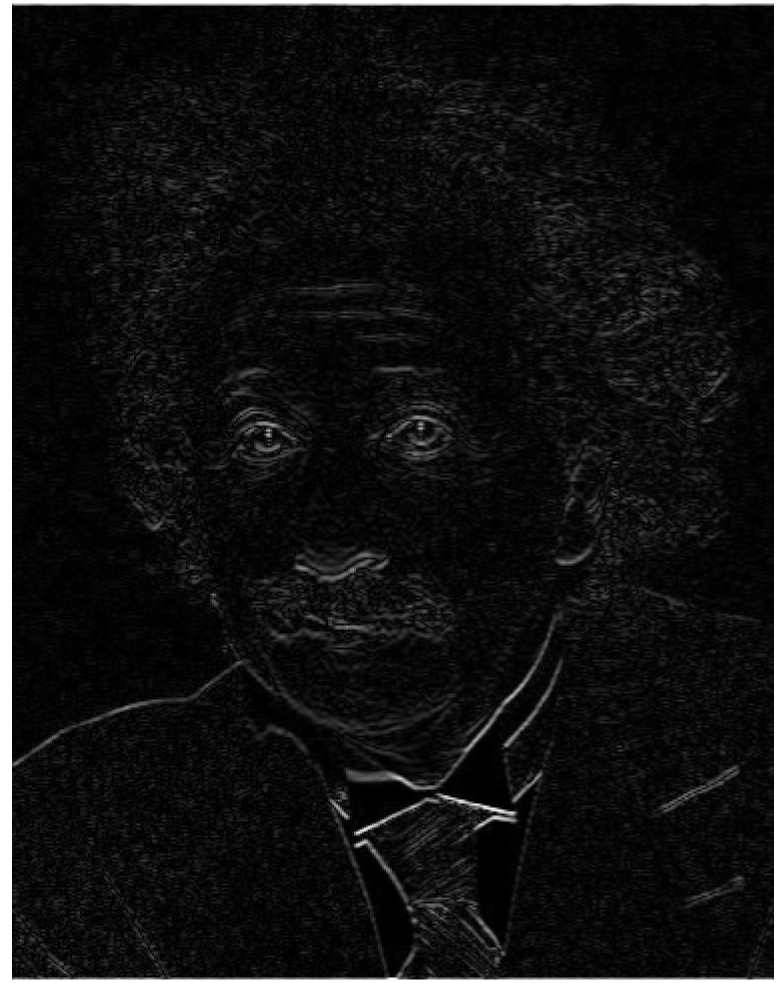
Vertical Edge
(absolute value)

Other filters



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

Sobel



Horizontal Edge
(absolute value)

Basic gradient filters

Horizontal Gradient

0	0	0
-1	0	1
0	0	0

or

-1	0	1

Vertical Gradient

0	-1	0
0	0	0
0	1	0

or

-1
0
1

Examples

Write as filtering operations, plus some pointwise operations: +, -, .*,>

1. Sum of four adjacent neighbors plus 1

$$out(m,n) = 1 + \sum_{k,l\in\{-1,1\}} in(m+k, n+l)$$

2. Sum of squared values of 3x3 windows around each pixel:

$$out(m,n) = \sum_{k,l\in\{-1,0,1\}} in(m+k, n+l)^2$$

3. Center pixel value is larger than the average of the pixel values to the left and right:

$$out(m,n) = 1 \quad \text{if} \quad in(m,n) > (in(m,n-1) + in(m,n+1))/2$$

$$out(m,n) = 0 \quad \text{if} \quad in(m,n) \le (in(m,n-1) + in(m,n+1))/2$$

Filtering vs. Convolution

- 2d filtering

  g=filter f=image

  - `h=filter2(g,f);` or
    `h=imfilter(f,g);`

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

- 2d convolution
  - `h=conv2(g,f);`

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

Key properties of linear filters

Linearity:

$\mathtt{filter}(f_1 + f_2) = \mathtt{filter}(f_1) + \mathtt{filter}(f_2)$

Shift invariance: same behavior regardless of pixel location

```
filter(shift(f)) = shift(filter(f))
```

Any linear, shift-invariant operator can be represented as a convolution

Source: S. Lazebnik

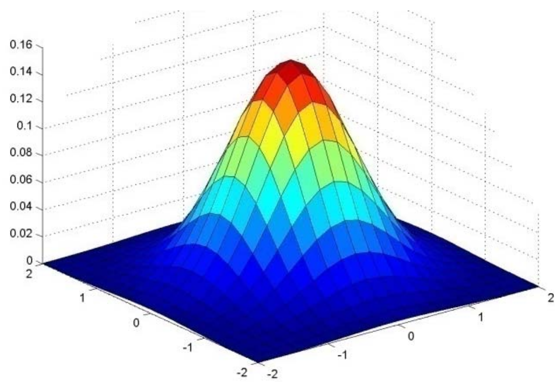
More properties

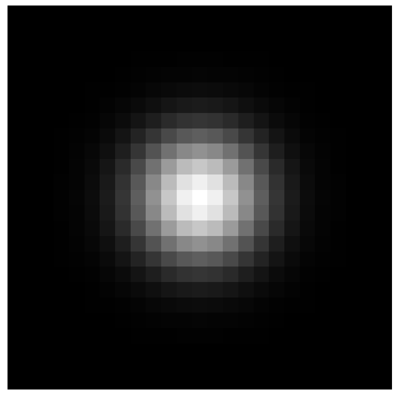
- Commutative: $a * b = b * a$
  - Conceptually no difference between filter and signal
- Associative: $a * (b * c) = (a * b) * c$
  - Often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - This is equivalent to applying one filter: a * $(b_1 * b_2 * b_3)$
- Distributes over addition: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

Source: S. Lazebnik

Important filter: Gaussian

- Spatially-weighted average





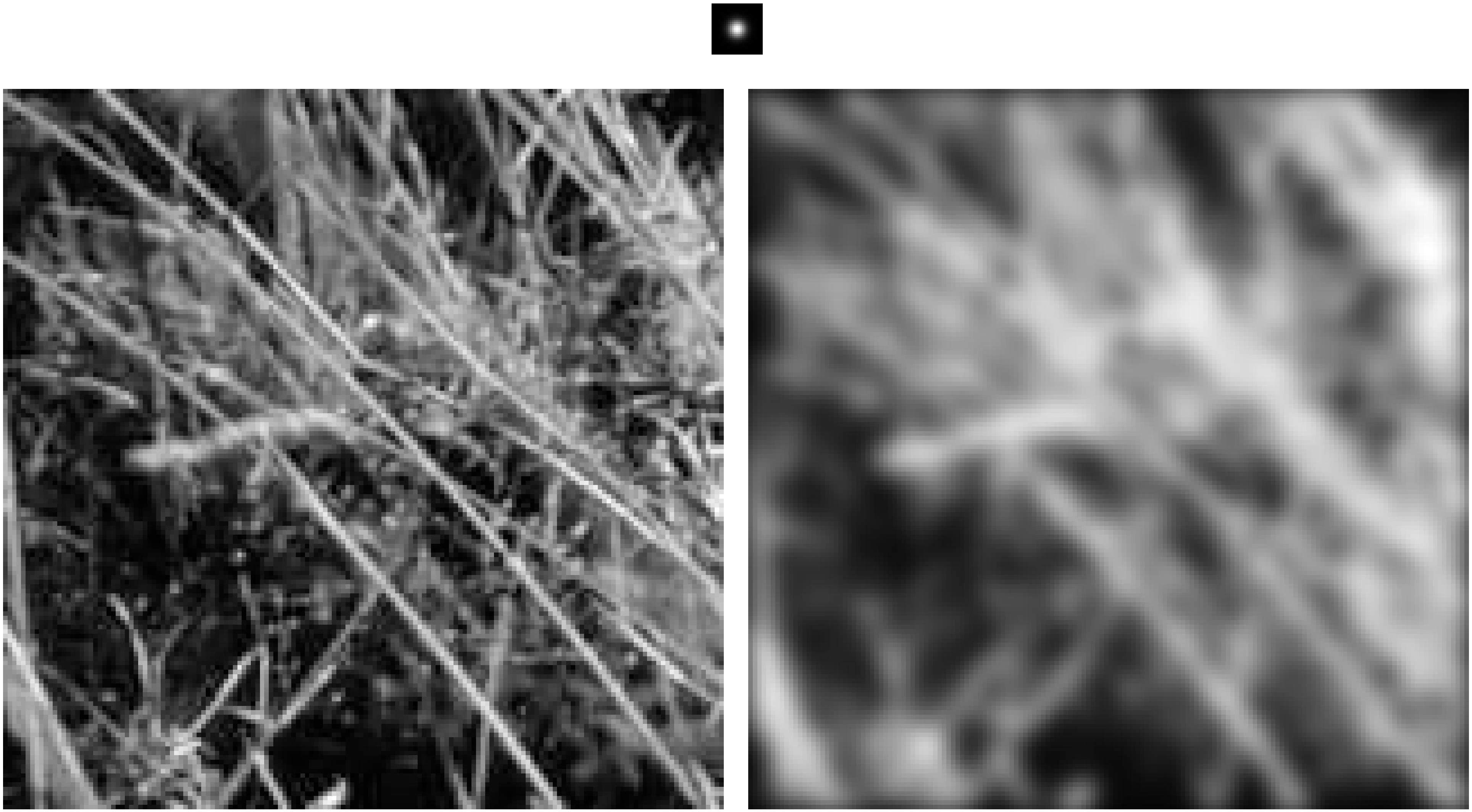
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, σ = 1

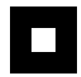
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

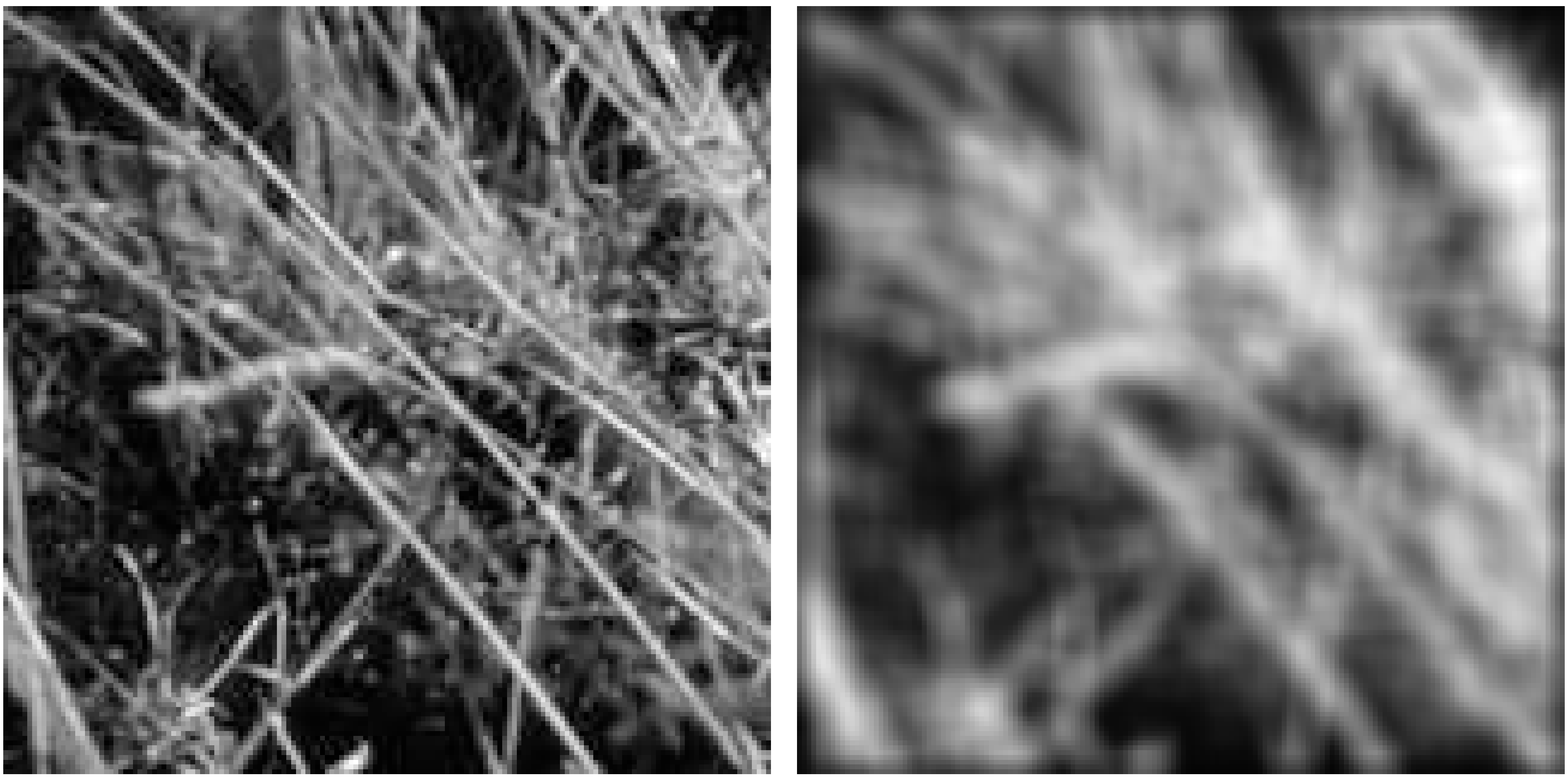
Slide credit: Christopher Rasmussen

Smoothing with Gaussian filter



Smoothing with box filter





Gaussian filters

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

Source: K. Grauman

Separability of the Gaussian filter

$$
\begin{aligned}
G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2}\exp^{-\frac{x^2+y^2}{2\sigma^2}} \\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{y^2}{2\sigma^2}}\right)
\end{aligned}
$$

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 example

2D filtering
(center location only)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix}$$

The filter factors
into a product of 1D
filters:

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

Perform filtering
along rows:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Followed by filtering
along the remaining column:

Source: K. Grauman

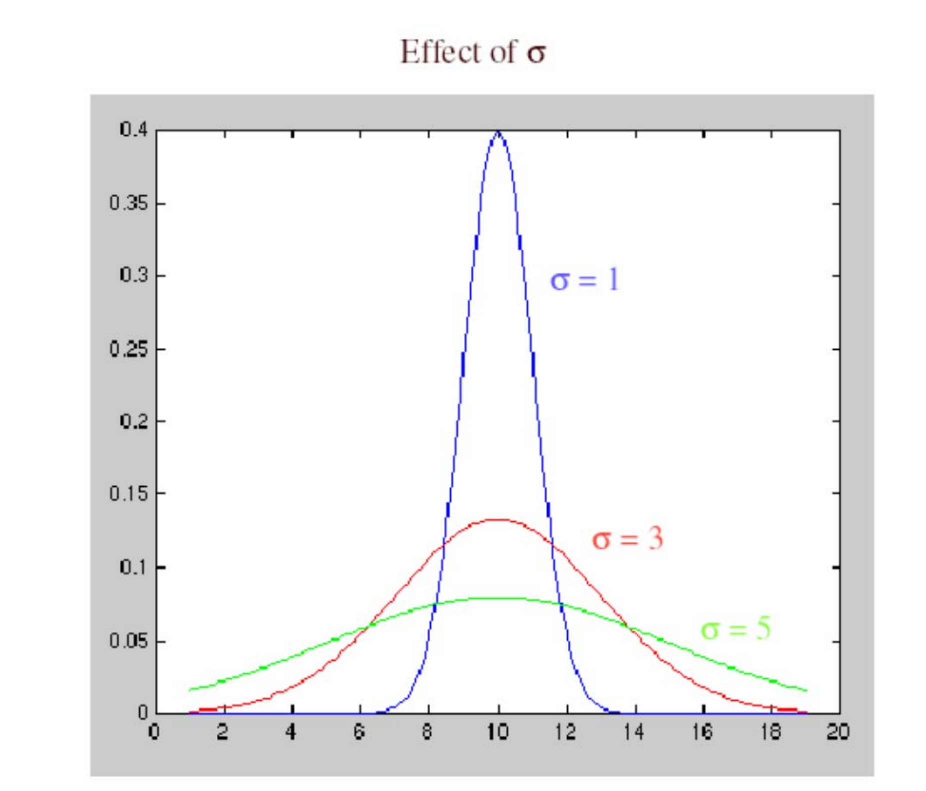
Separability

- Why is separability useful in practice?

Practical matters

How big should the filter be?

- Values at edges should be near zero ← important!
- Rule of thumb for Gaussian: set filter half-width to about 3 σ



Practical matters

- What about near the edge?
  - the filter window falls off the edge of the image
  - need to extrapolate
  - methods:
    - clip filter (black)
    - wrap around
    - copy edge
    - reflect across edge

Source: S. Marschner

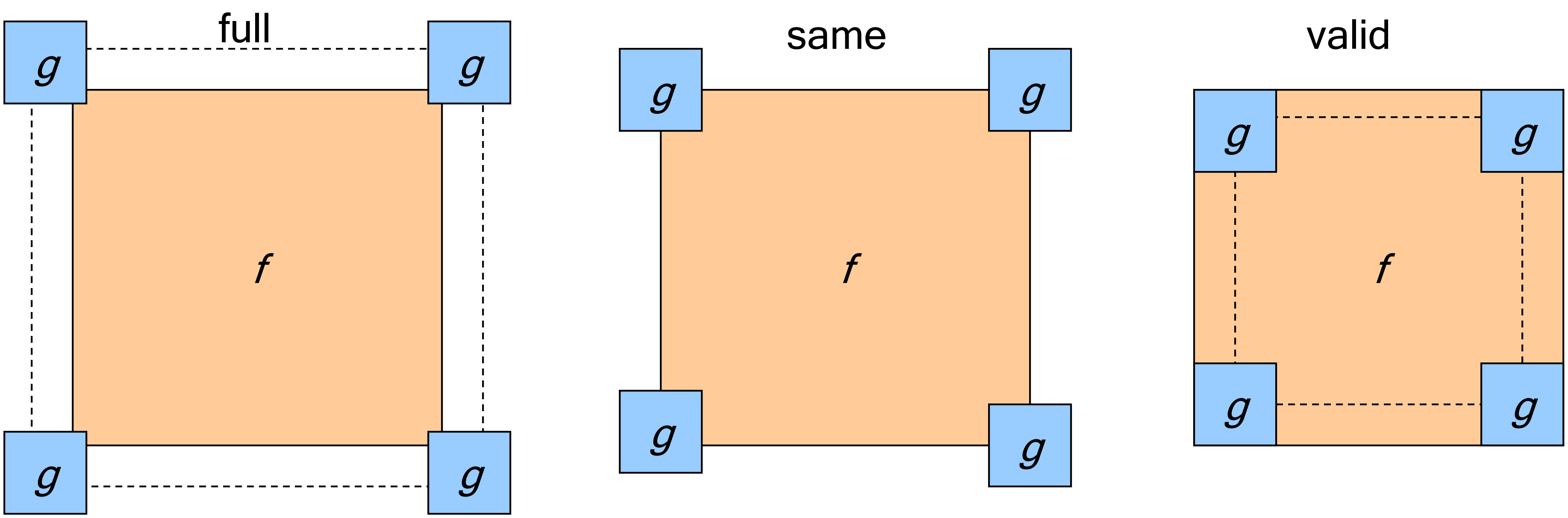
Practical matters

– methods (MATLAB):
  - clip filter (black): imfilter(f, g, 0)
  - wrap around: imfilter(f, g, ‘circular’)
  - copy edge: imfilter(f, g, ‘replicate’)
  - reflect across edge: imfilter(f, g, ‘symmetric’)

Source: S. Marschner

Practical matters

- What is the size of the output?
- MATLAB: filter2(g, f, *shape*)
  - *shape* = ‘full’: output size is sum of sizes of f and g
  - *shape* = ‘same’: output size is same as f
  - *shape* = ‘valid’: output size is difference of sizes of f and g



Source: S. Lazebnik

Slide Credits

- This set of sides contains contributions kindly made available by the following authors
  - Gianfranco Doretto
  - Derek Hoiem
  - Alexei Efros
  - Svetlana Lazebnik
  - Kristen Grauman
  - Frédo Durand
  - David Lowe
  - Steve Marschner
  - Steve Seitz
  - Richard Szeliski