CS 89.15/189.5, Fall 2015

COMPUTATIONAL ASPECTS OF DIGITAL PHOTOGRAPHY

Image Processing Basics

Wojciech Jarosz
wojciech.k.jarosz@dartmouth.edu

Dartmouth

Domain, range

Domain vs. range

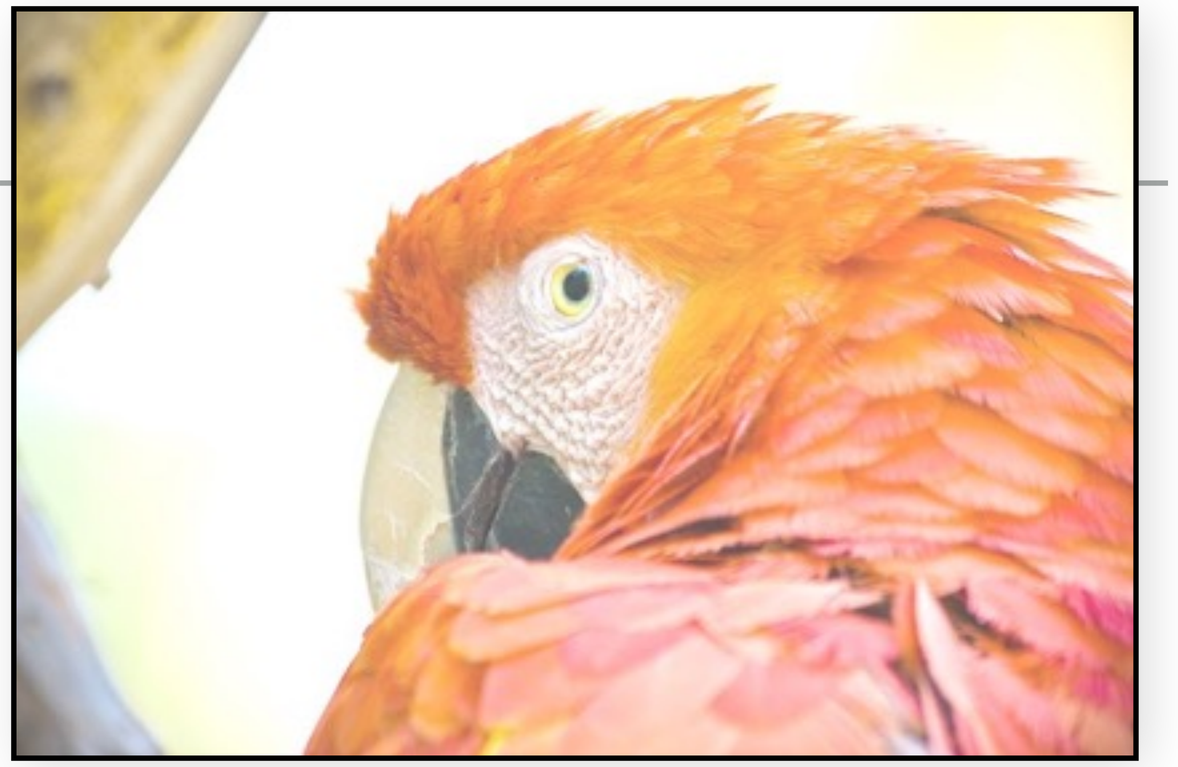
2D plane: domain of images

color value: range (R^3 for us)

- red, green and blue components stored in im(x, y, 0), im(x, y, 1), im(x, y, 2), respectively

Basic types of operations

output(x,y)



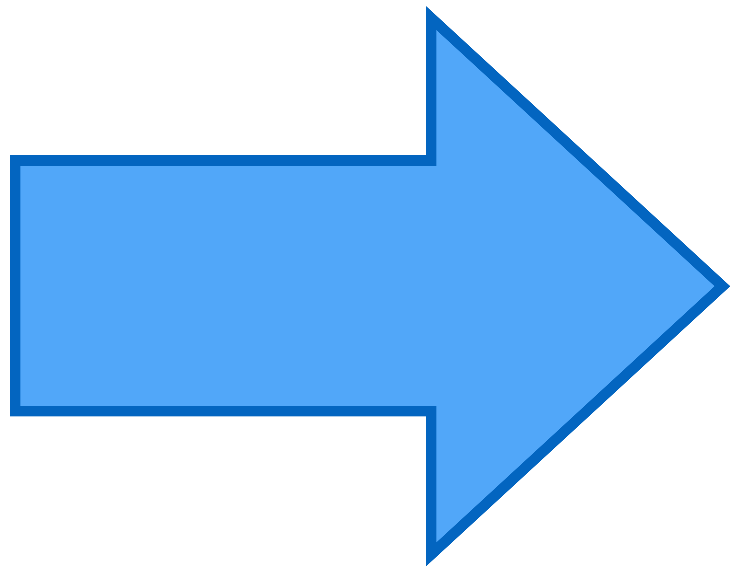
output(x,y) = f(image(x,y))

Point operations:

range only

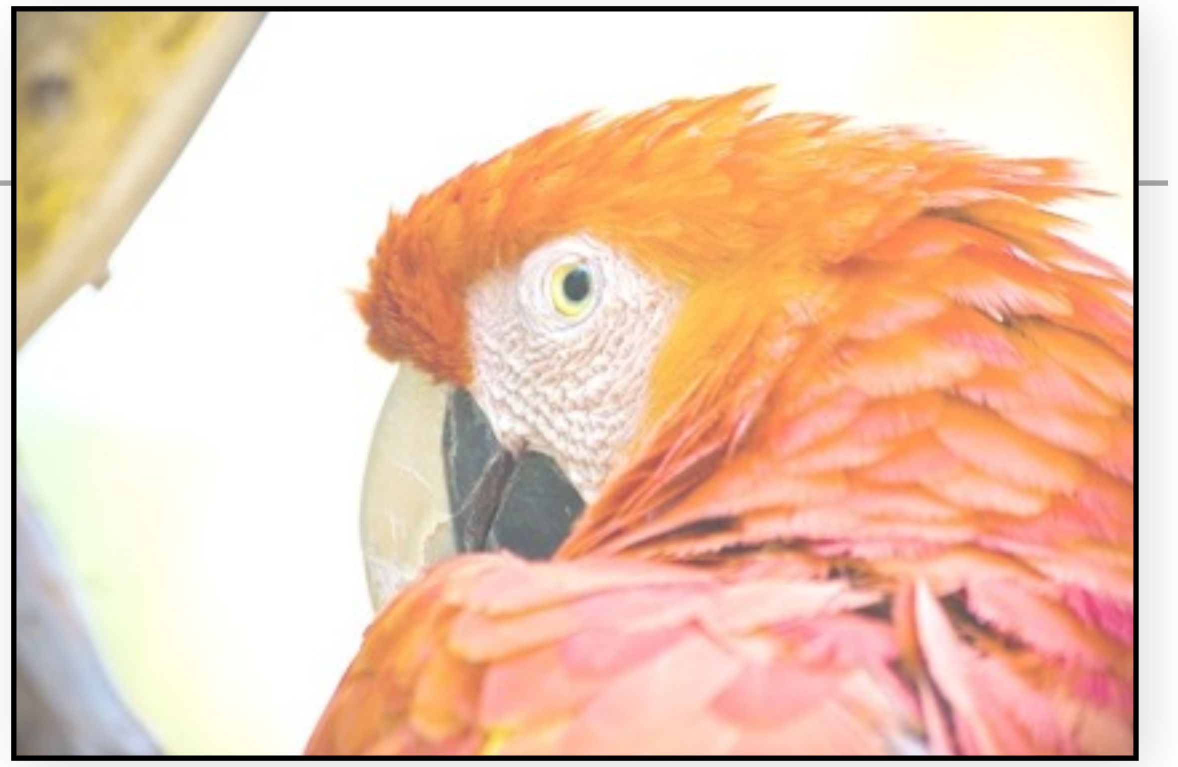
Assignment 2

image(x,y)



Basic types of operations

output(x,y)



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

Point operations:
range only
Assignment 2

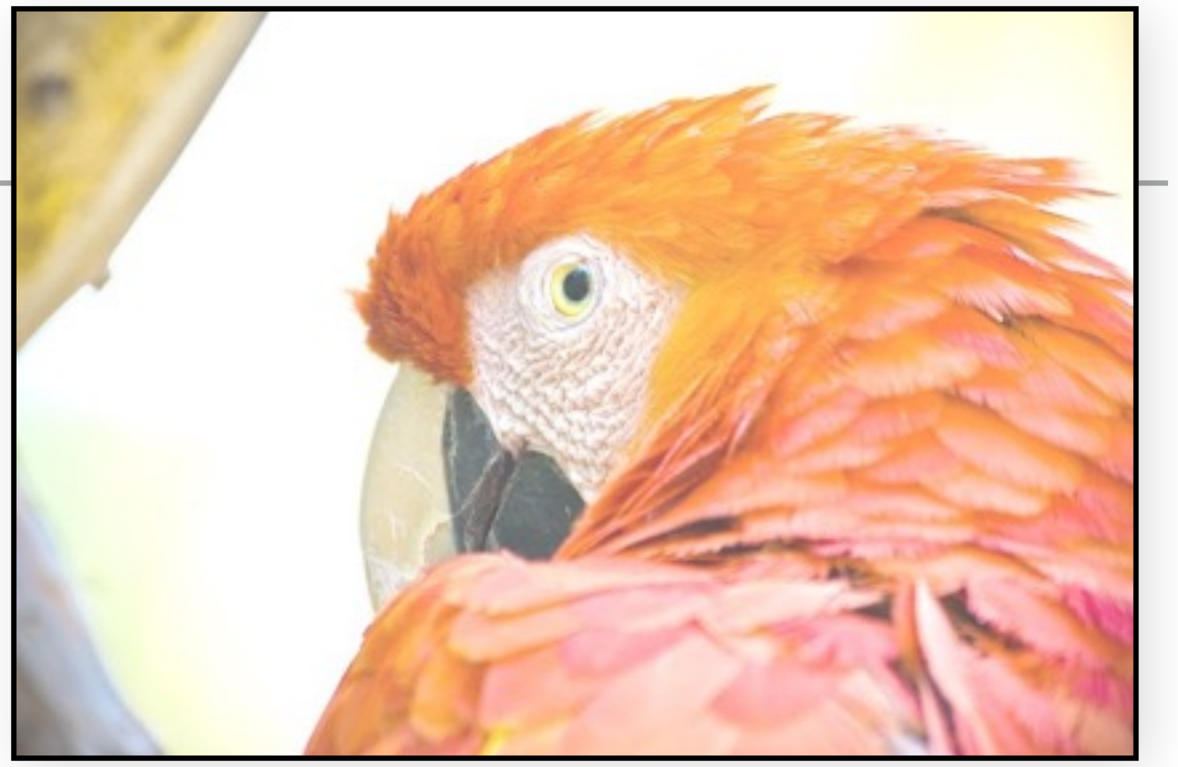
image(x,y)

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

Domain
operations
Assignment 6

Basic types of operations

output(x,y)



image(x,y)

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

Point operations:
range only
Assignment 2

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

Domain
operations
Assignment 6

Neighborhood operations:
domain and range
Assignments 3, 4, 5

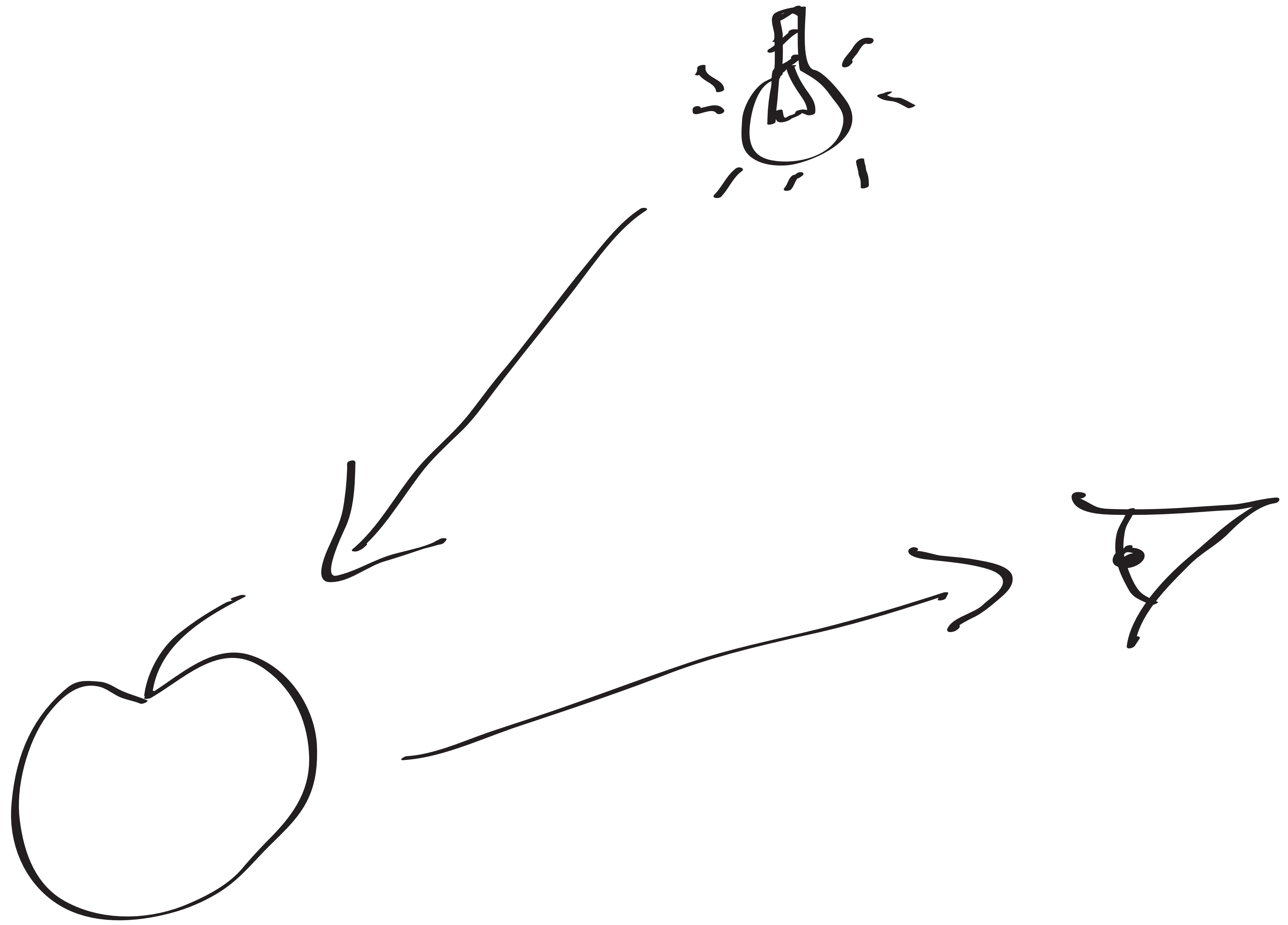
Light & perception

Light matter, eyes

Light from sources is reflected by objects and reaches the eye

The amount of light from the source gets multiplied by the object reflectance

- on a per-wavelength basis



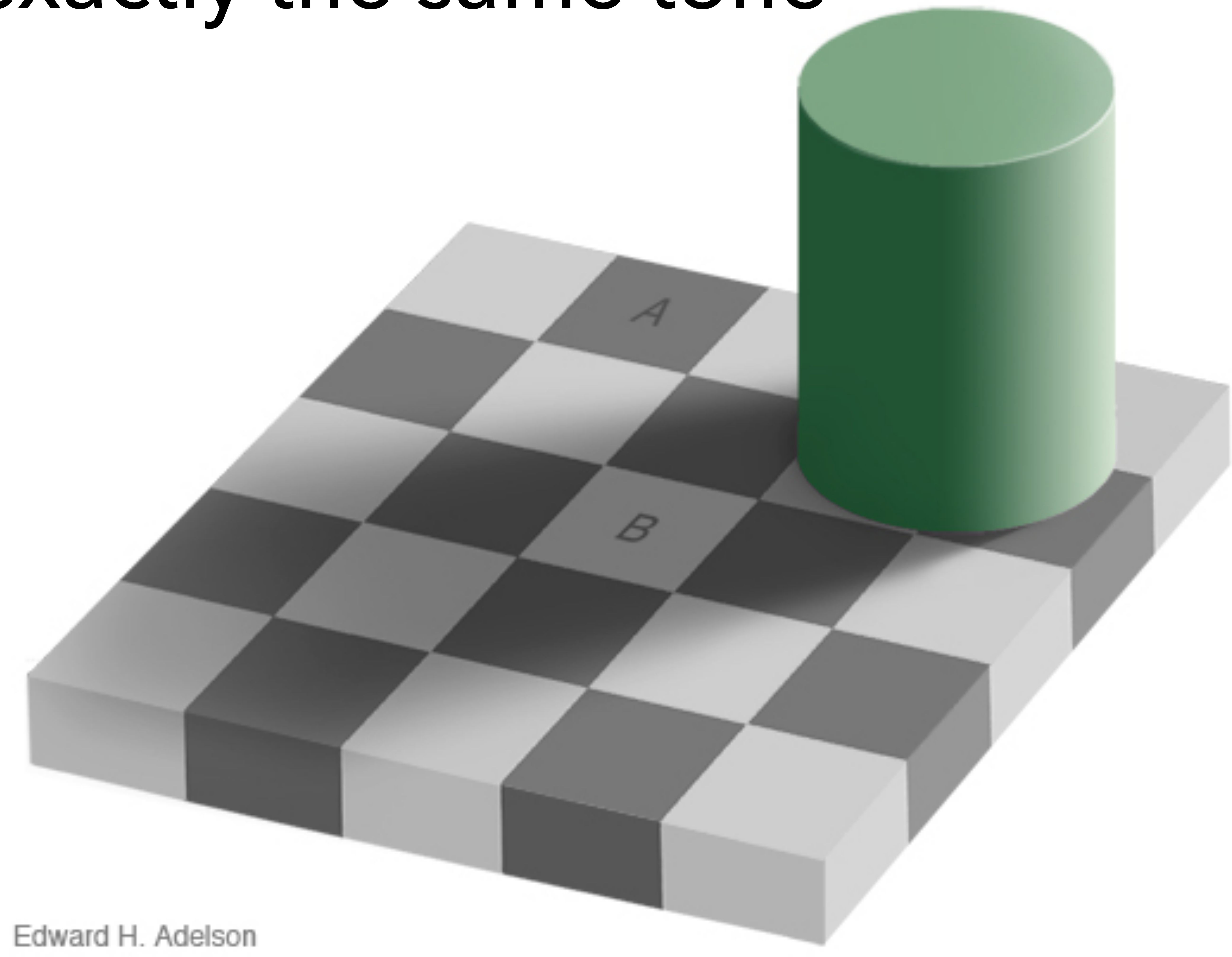
Human perception

Our eyes have an uncanny ability to discount the illumination

- Only objects really matter for survival
- Light is only useful to understand if you're a photographer or to choose your sun lotion

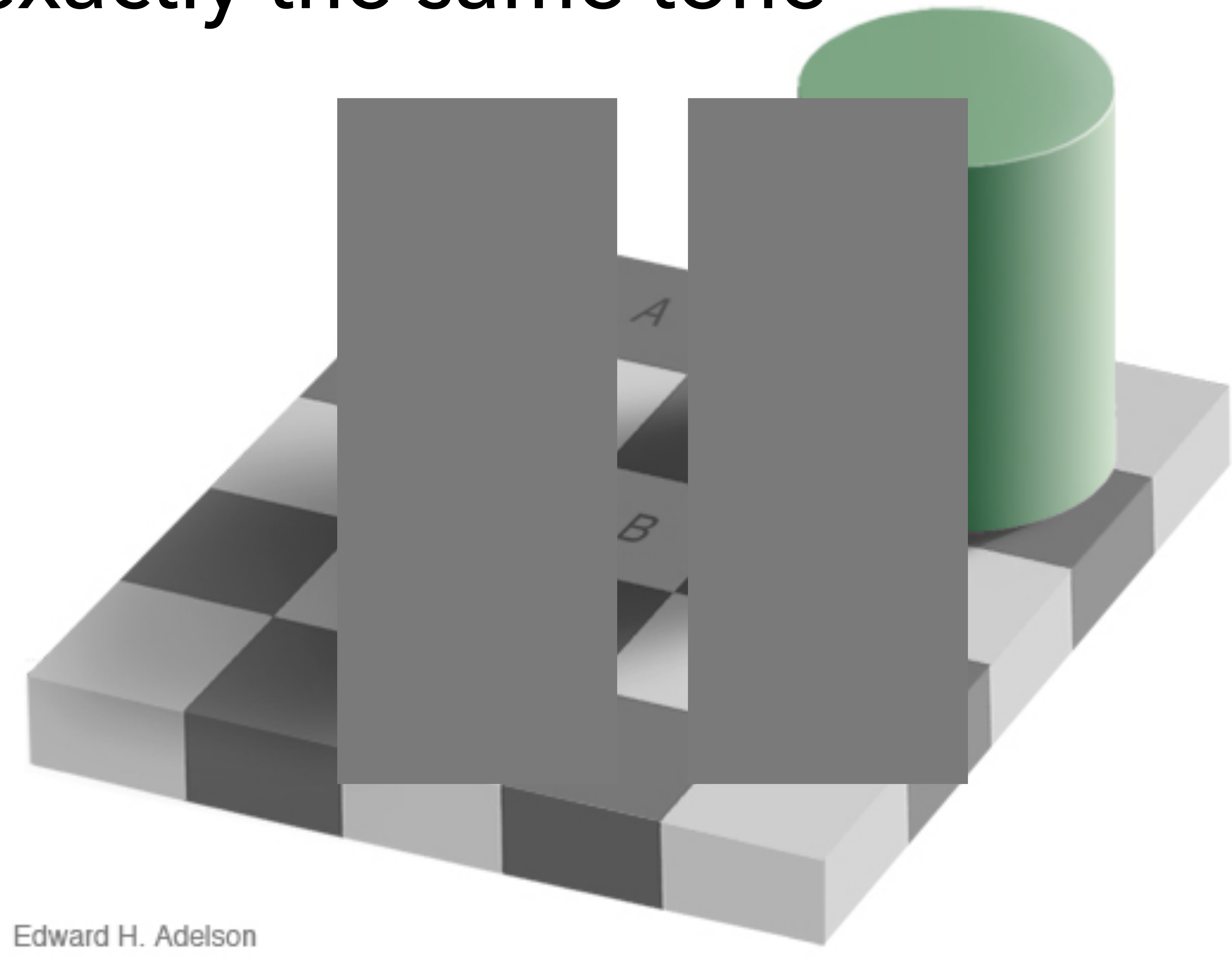
Illusion by Adelson

A & B have exactly the same tone



Illusion by Adelson

A & B have exactly the same tone



Mechanism to discount light

Light adaptation

- We re-center our neural response around the current average brightness
- neural + chemical + pupil

Chromatic adaptation

- eliminate color cast due to light sources
  e.g. Daylight is white but tungsten is yellowish
- Related to white balance - more soon
- and Spanish Castle illusion

Contrast is about ratios

Contrast between 1 & 2 is the same as between 100 & 200

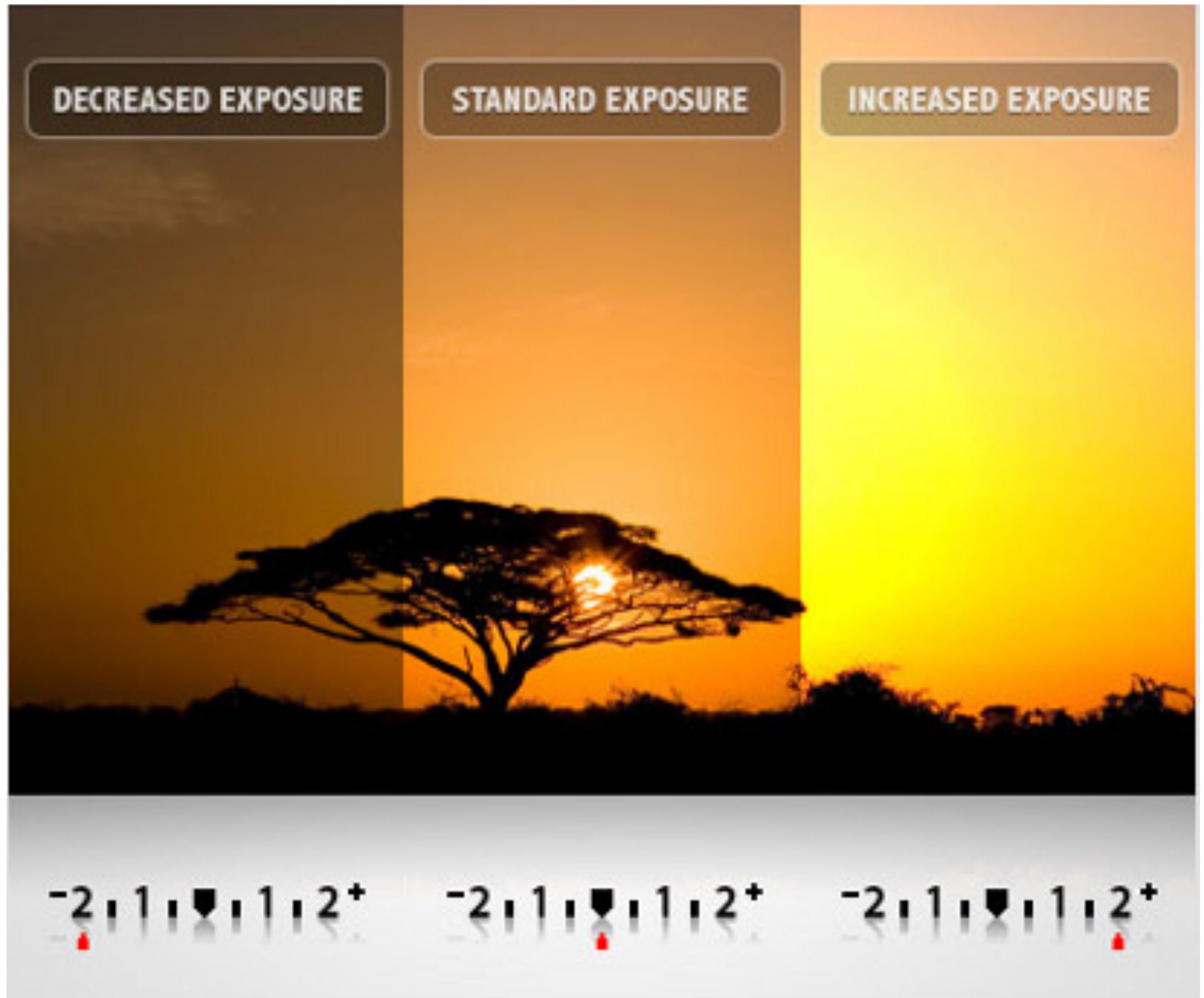
Useful to discount the multiplicative effect of light

0.1 to 0.2   0.4 to 0.8

Exposure

On cameras, exposure (shutter speed, aperture, ISO) has a multiplicative effect on the values recorded by the sensor.

Changes the "brightness", not contrast



http://photographystepbystep.com/exposure-2/auto-bracketing/

White balance

White balance & Chromatic adaptation

Different illuminants have different color temperature

Our eyes adapt: chromatic adaptation

- We actually adapt better in brighter scenes
- This is why candlelit scenes still look yellow

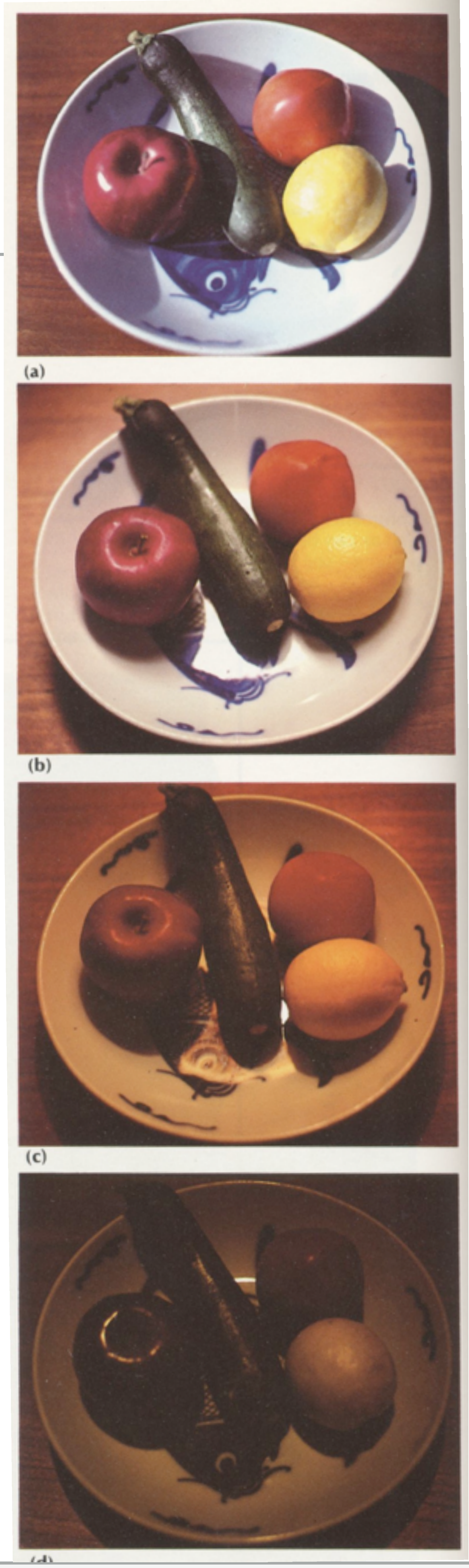
www.shortcourses.com/guide/guide2-27.html

White balance problem

When watching a picture on screen or print, we adapt to the illuminant of the room, not that of the scene in the picture

The eye cares more about objects' intrinsic color, not the color of the light leaving the objects

We need to discount the color of the light source



Same object,
different illuminants

White balance & Film

Different types of film for fluorescent, tungsten, daylight

Need to change film!

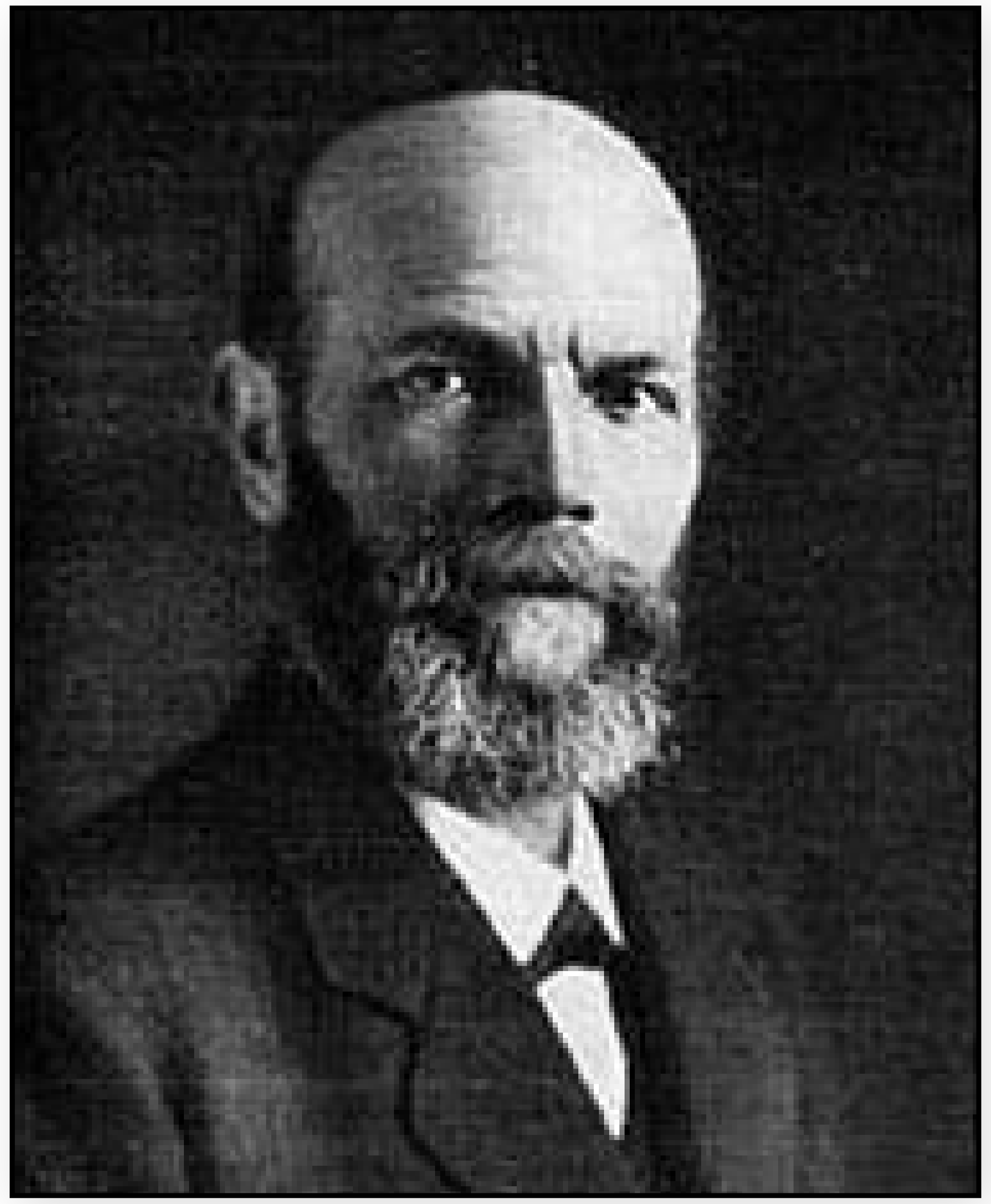
Electronic & Digital imaging are more flexible

Von Kries adaptation

Multiply each channel by a gain factor

- $R' = R * k_r$
- $G' = G * k_g$
- $B' = B * k_b$

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



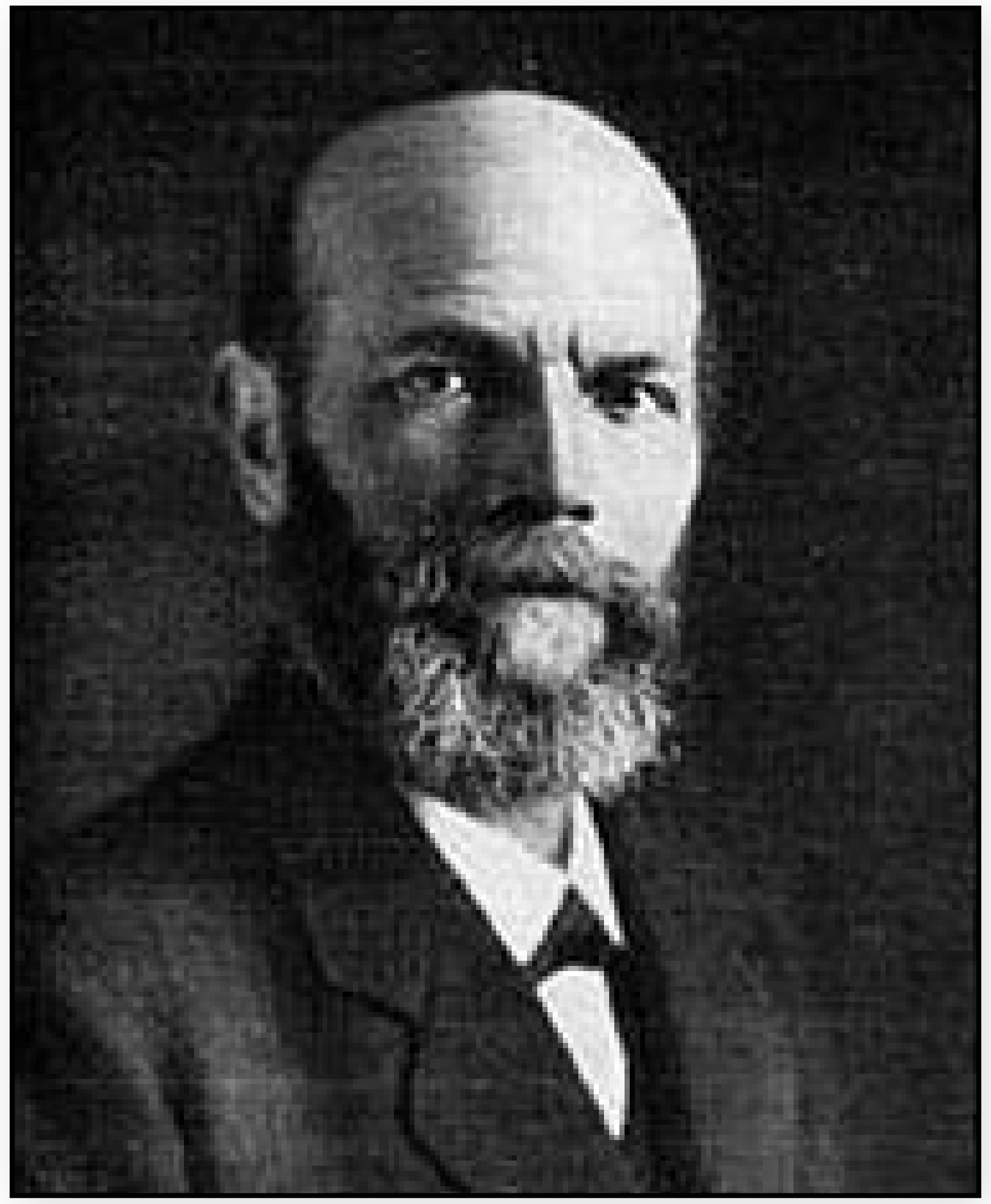
Von Kries adaptation

Multiply each channel by a gain factor

Note that the light source could have a more complex effect

- Arbitrary 3×3 matrix
- More complex spectrum transformation

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



White balance challenge

How do we find the scaling factors for r, g, and b?

Best way to do white balance

Grey card:

Take a picture of a neutral object (white or gray)

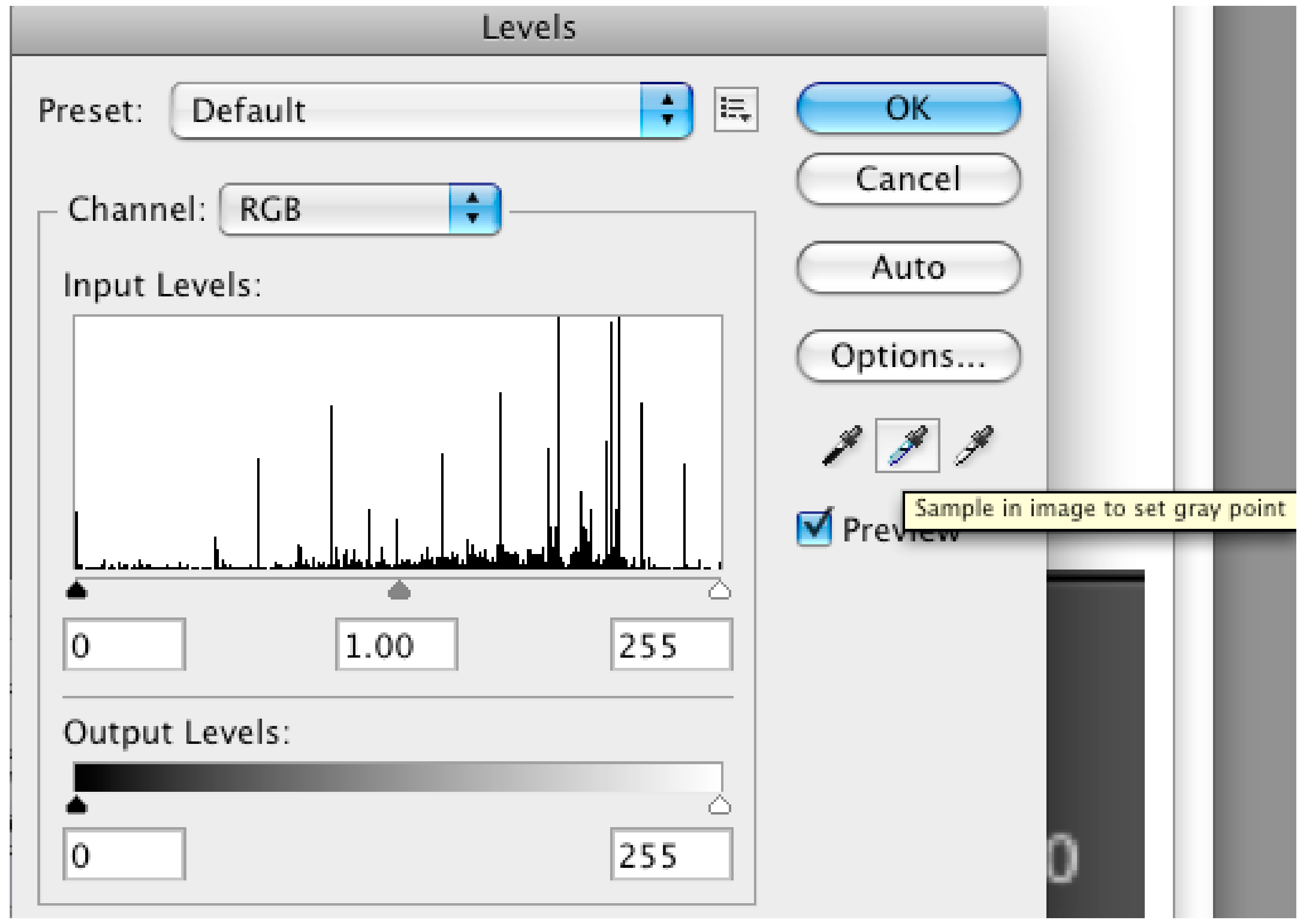
Deduce the weight of each channel

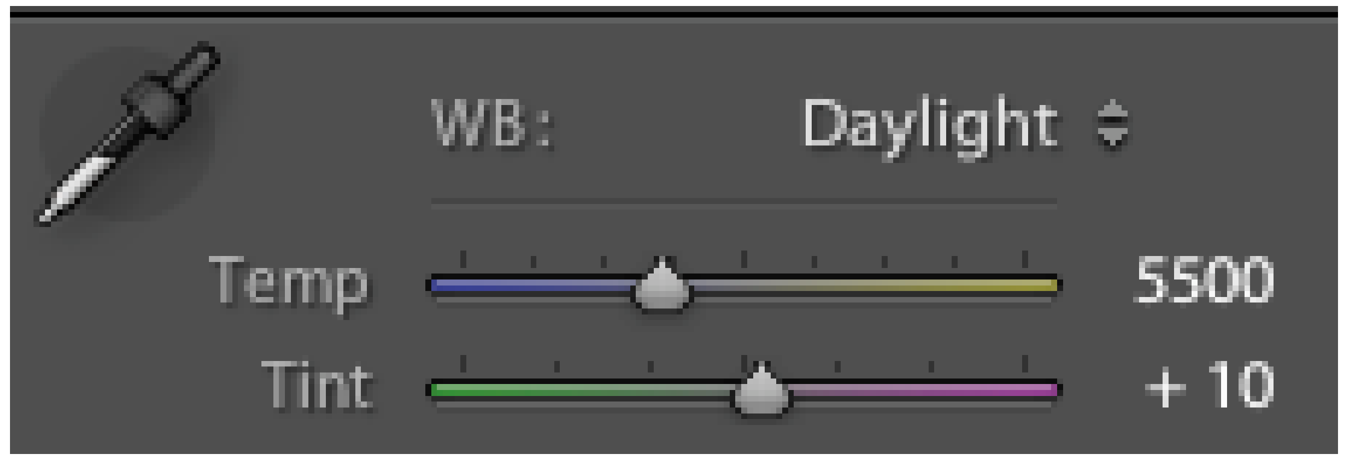
If the object is recoded as r_w, g_w, b_w
use weights k/r_w, k/g_w, k/b_w
where k controls the exposure

Lightroom demo

Most photo editing software lets you click on a neutral object to achieve white balance

- In "Levels" in Photoshop
- In "Basic" in Lightroom
- You also often have presets such as daylight, tungsten





Party name tags

Provide excellent white references!


```
write(im/im(300, 214))
```

Without grey cards

We need to “guess” which pixels correspond to white objects

Grey world assumption

Assume average color in the image is grey

Use weights proportional to

$$\frac{1}{\int_{image} r}, \frac{1}{\int_{image} g}, \frac{1}{\int_{image} b}$$

Usually assumes 18% grey to set exposure

Brightest pixel assumption

Highlights usually have the color of the light source

- At least for dielectric materials

White balance by using the brightest pixels

- Plus potentially a bunch of heuristics
- In particular use a pixel that is not saturated/clipped

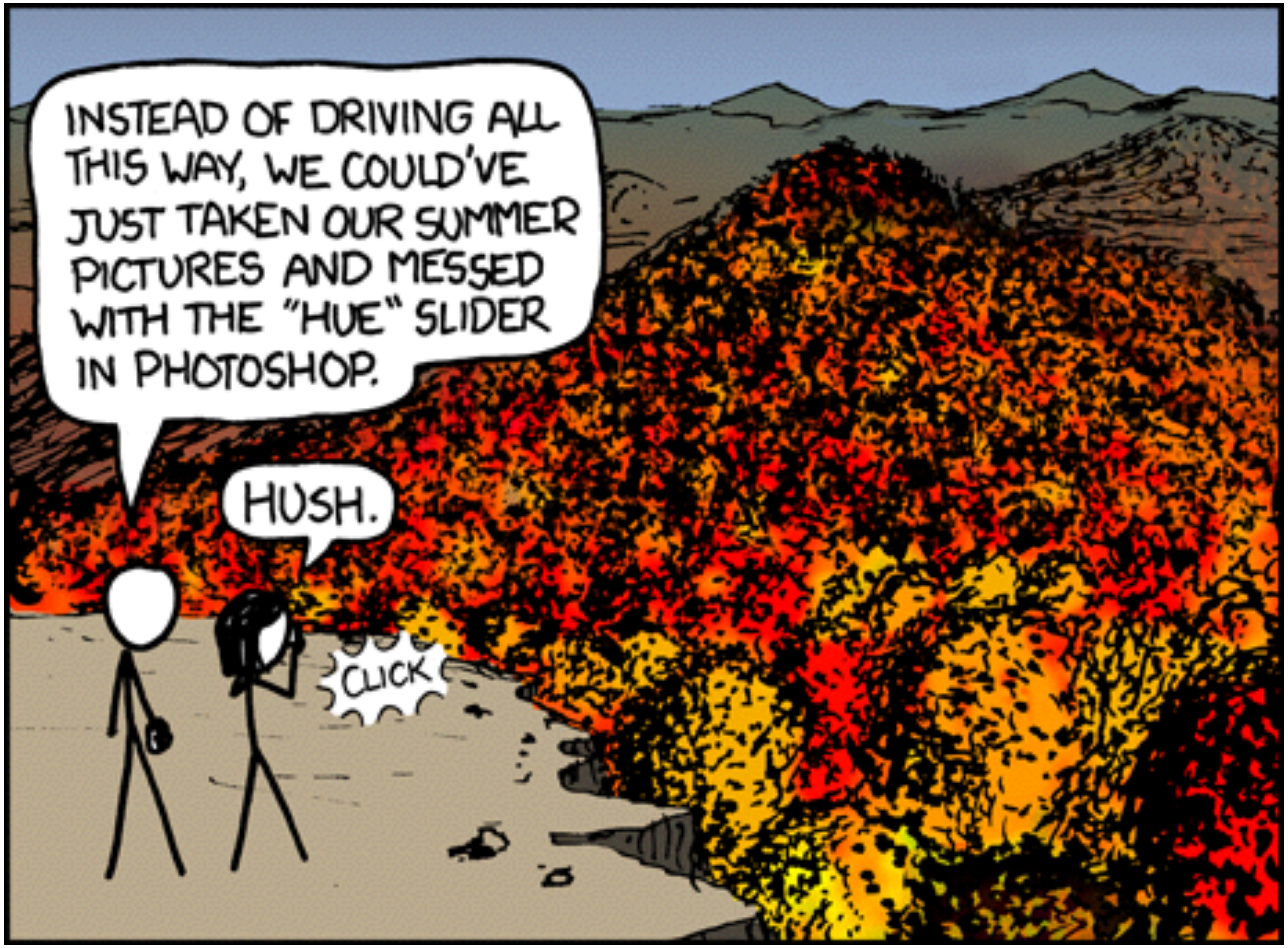
Refs

Recent work on color constancy

- http://gvi.seas.harvard.edu/paper/perceptionbased-color-space-illuminationinvariant-image-processing
- http://gvi.seas.harvard.edu/paper/color-subspaces-photometric-invariants
- http://people.csail.mit.edu/billf/papers/BayesJOSA.pdf

Still an open problem!

Questions?



from xkcd

Take home messages

Discounting the illumination is useful

Ratios matter

Optical illusions are not optical but fun

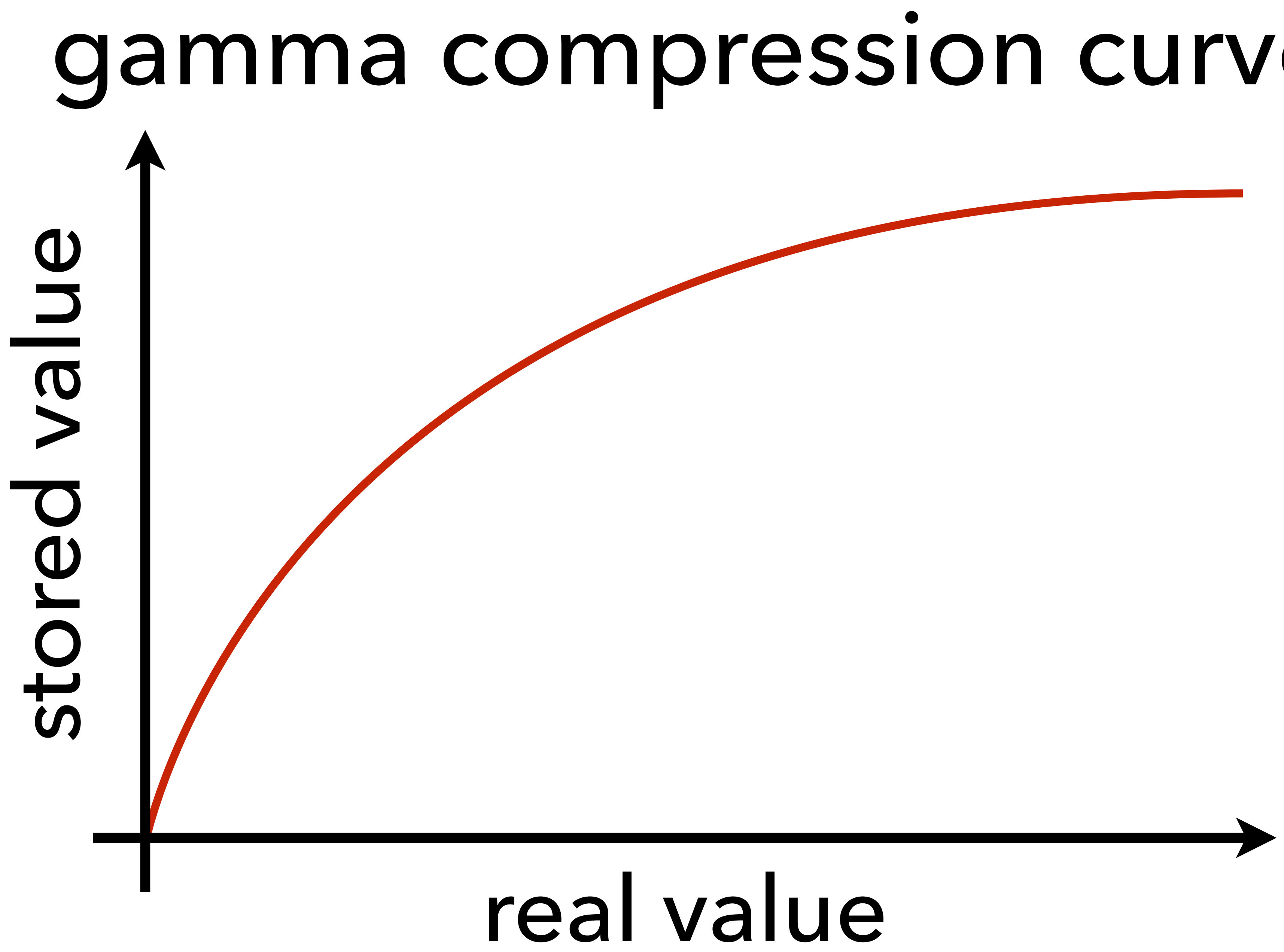
Gamma

Linearity and gamma

Images are usually gamma encoded

Instead of storing the light intensity x, they store x^γ

to get more precision in dark areas for 8-bit encoding



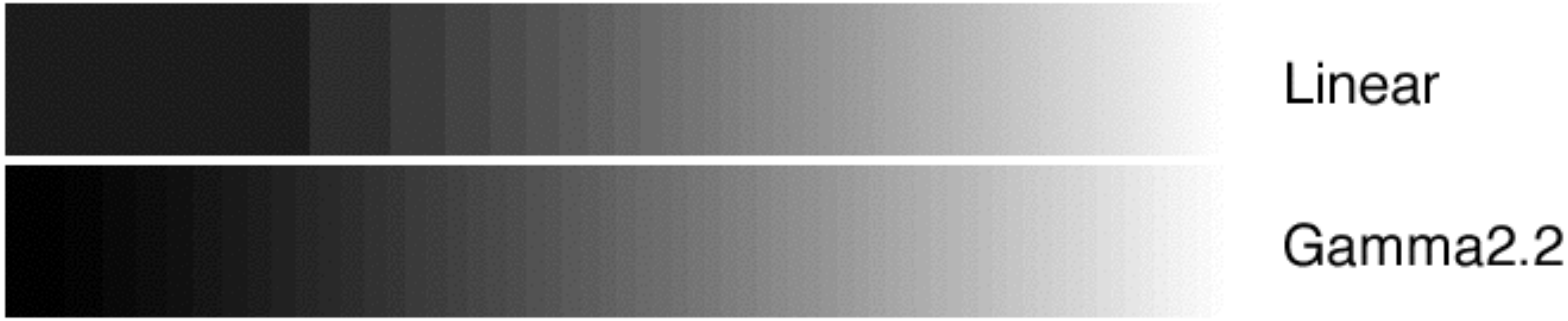
Linearity and gamma

Images are usually gamma encoded

Instead of storing the light intensity x, they store x^γ

to get more precision in dark areas for 8-bit encoding

6 bit encoding for emphasis:



Gamma demo

http://web.mit.edu/lilis/www/gammavis.html

Linearity and gamma

Images are usually gamma encoded

Instead of storing the light intensity x, they store x^γ

Half of image processing algorithms work better in linear space

- If linearity is important
- To deal with ratios and multiplicative factors better

Half work better in gamma space

- closer to logarithmic scale

How to capture linear images

http://www.mit.edu/~kimo/blog/linear.html

Take home message

Images are usually gamma-encoded

gamma ~2.2

provides better quantization

sometimes good for algorithms

sometimes bad

- convert to linear values!

Histograms

Histogram

Histogram:

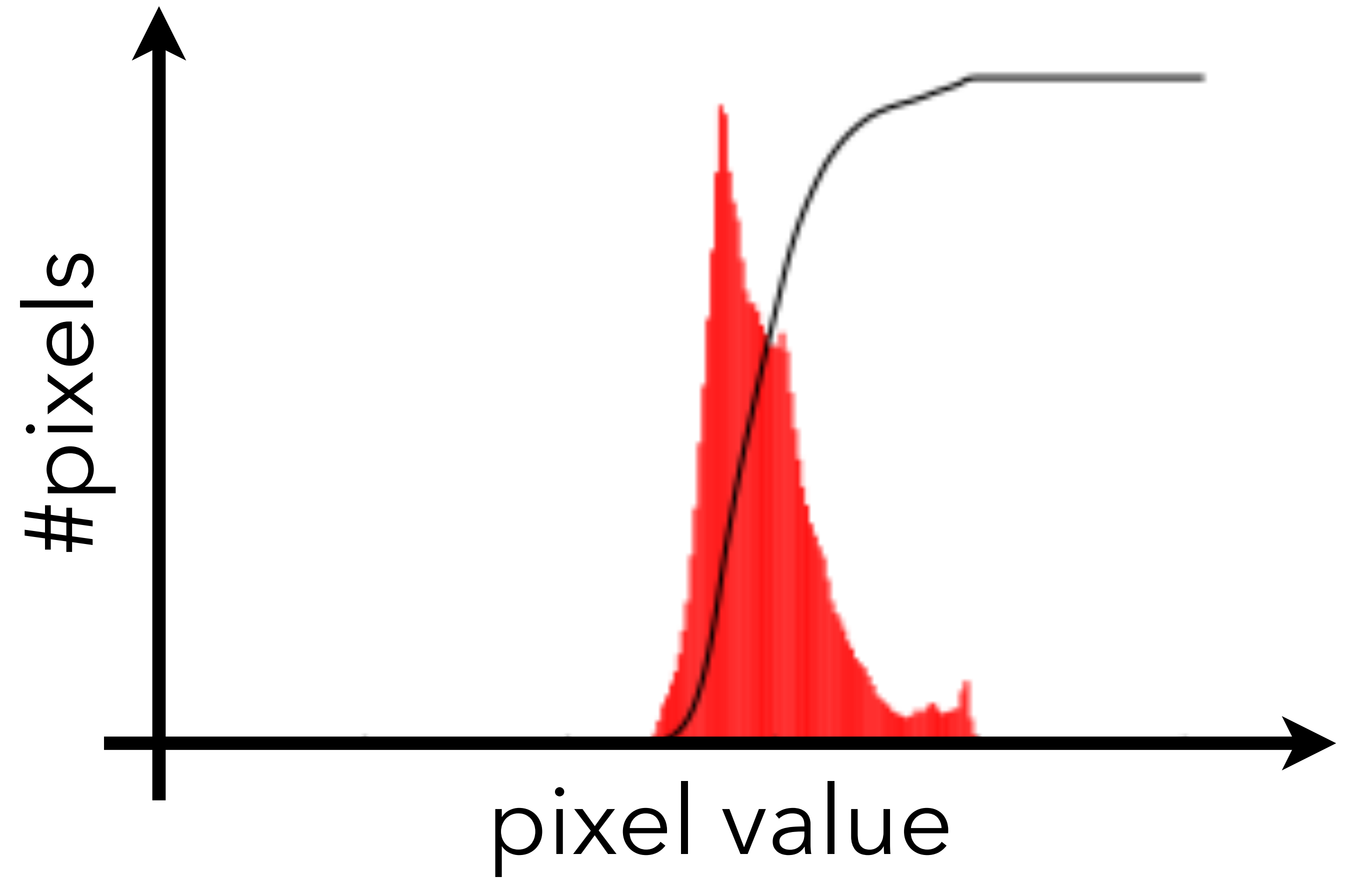
- For each value (e.g. 0-255), how many pixels have this value?

Cumulative histogram:

- for each value x, how many pixels have a value smaller than x?

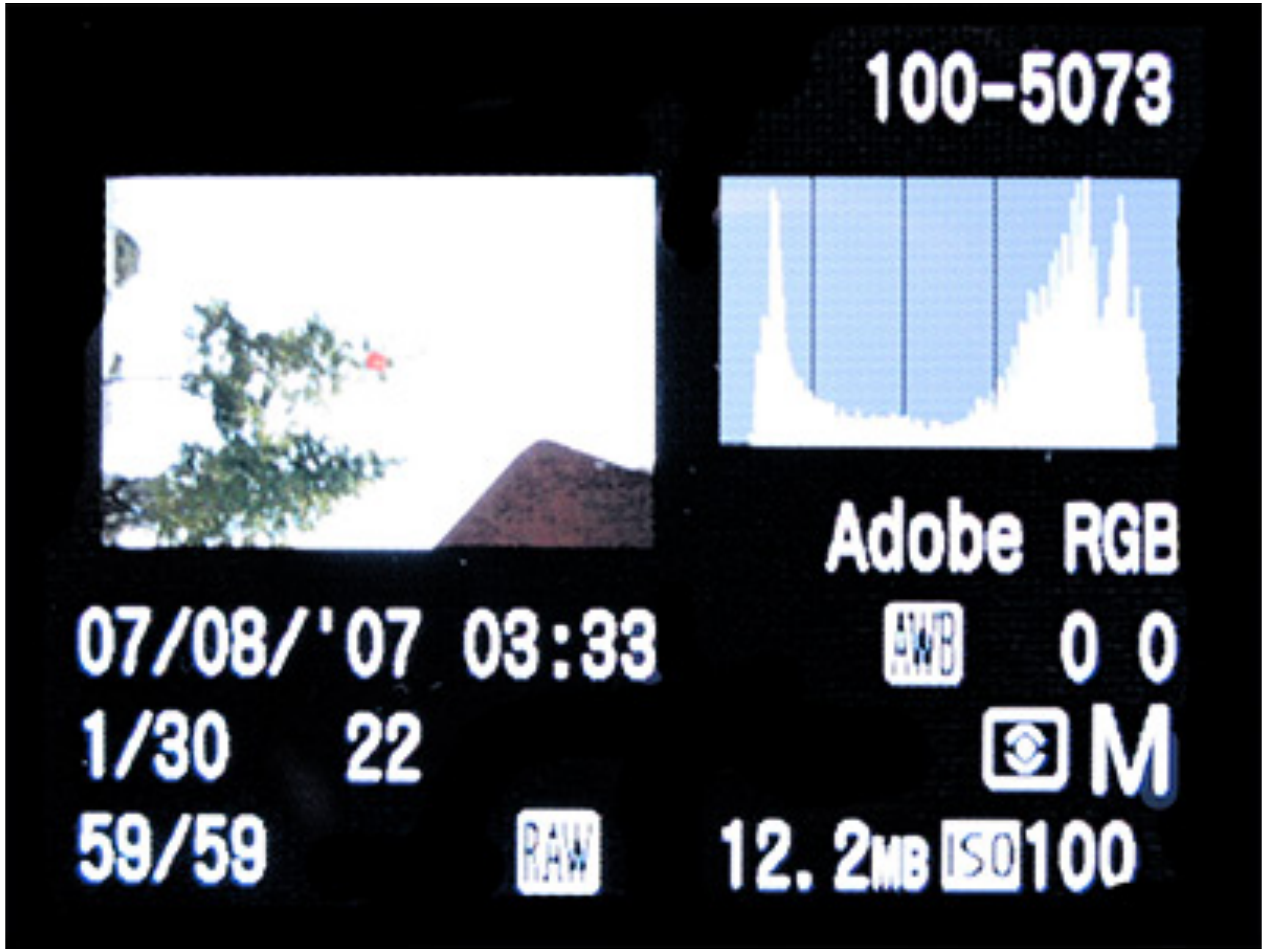
Normalized: divide value of each bin by total number of pixels

- histogram = discrete PDF
- cumulative histogram = discrete CDF



Very useful on camera

Allows you to tell if you use the dynamic range entirely



Bad: bright values under-used (underexposure)

Bad: bright values saturate (overexposure)

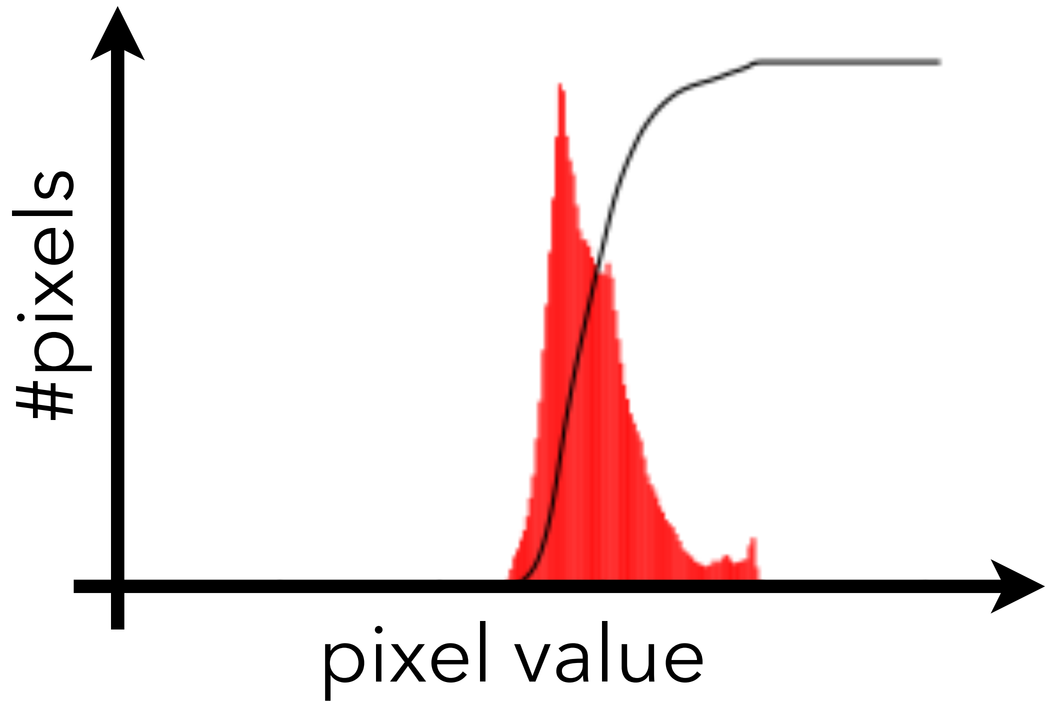
http://www.luminous-landscape.com/tutorials/understanding-series/understanding-histograms.shtml

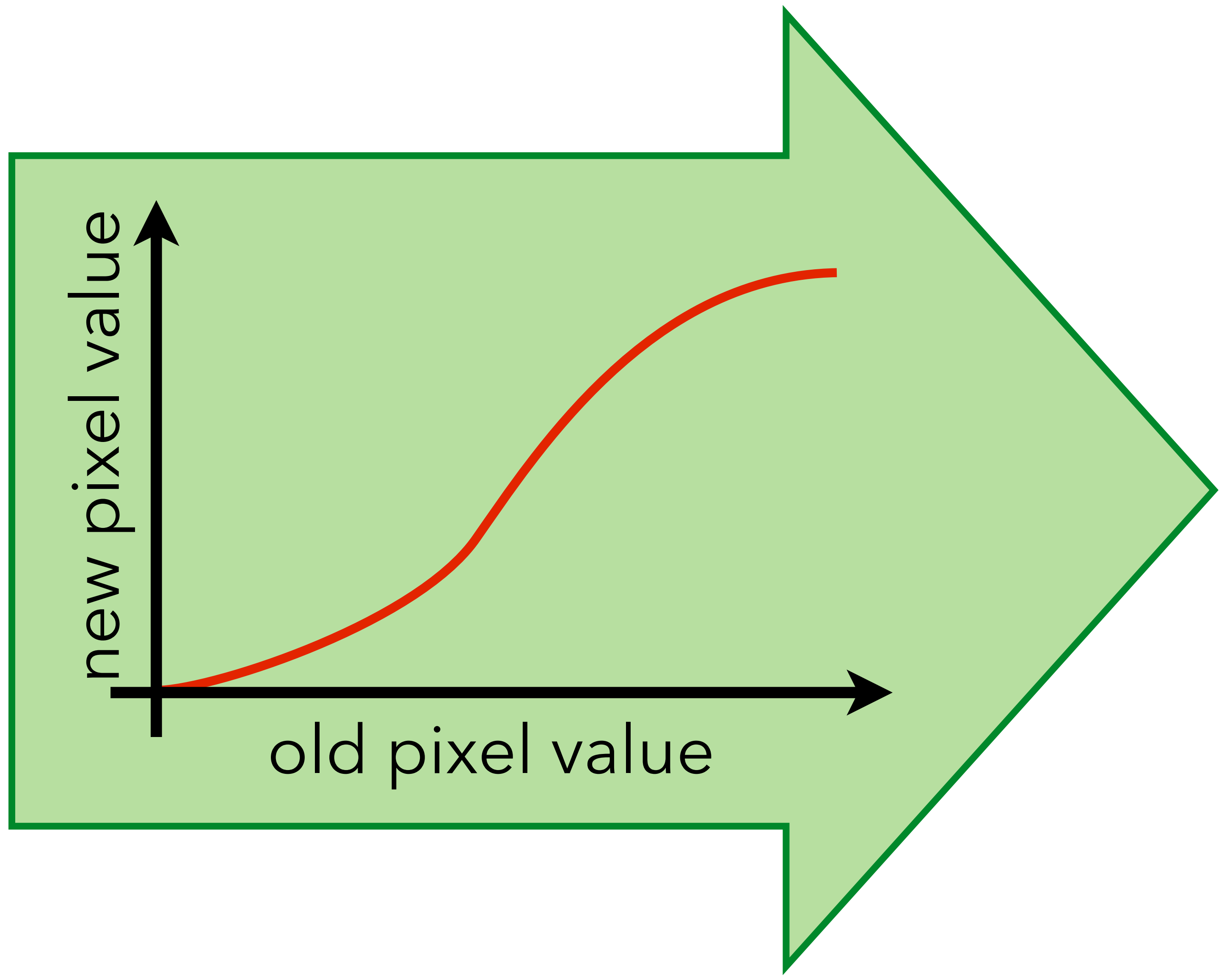
Histogram equalization

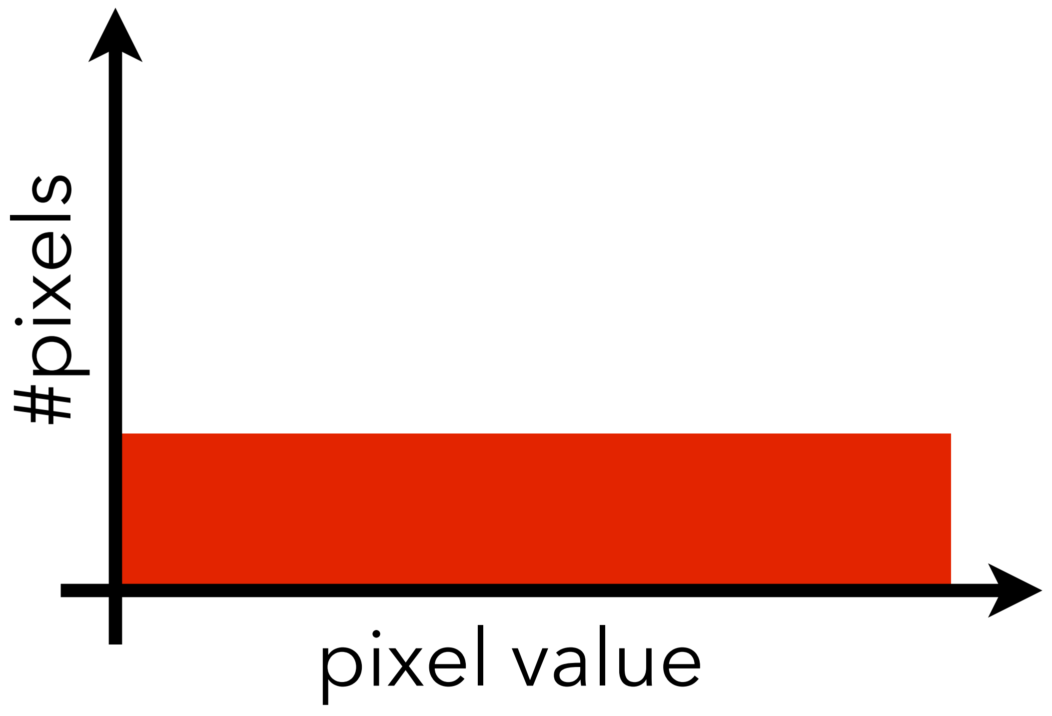
Histogram equalization

Given an image with a certain histogram,

monotonic remapping to get a flat histogram







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

Histogram equalization

Ideal flat histogram: y% of pixels have a value less than y%

- assuming everything is normalized to [0,1]

Flip it: a pixel with value larger than y% of all pixels should have value y%

For an old value x%, we know the number of pixels that have value < x%: cumulative histogram (also called CDF)

Therefore, we want x to be mapped to its cumulative histogram value.

Histogram matching

Histogram matching

- Given a desired histogram
- Map each value of an image channel to a new value, such that the new histogram matches the desired histogram

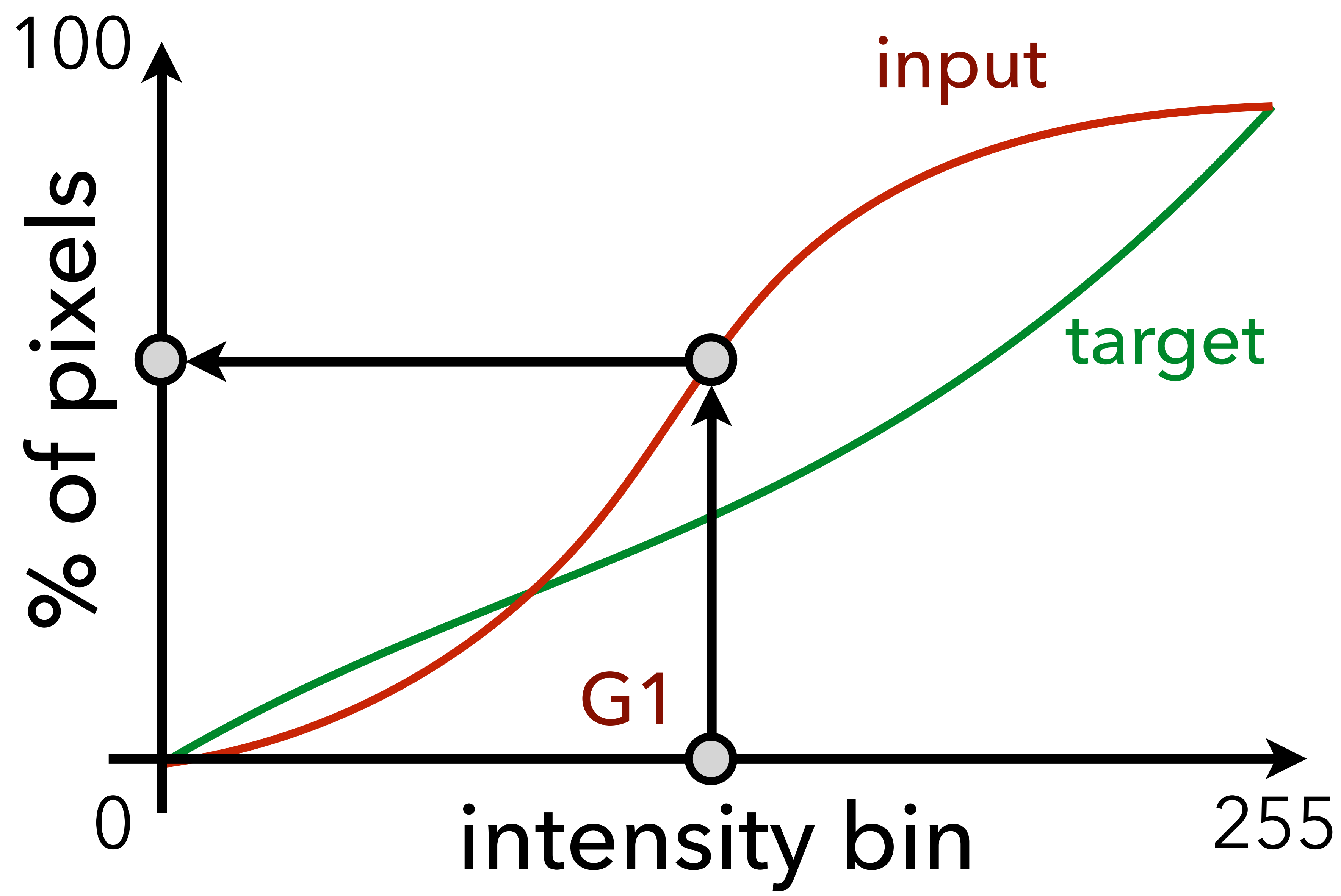
Histogram equalization

- The desired histogram is simply constant
- What shape is the cumulative histogram?

Histogram matching

Histogram matching is done by adjusting the cumulative distribution function (cdf)

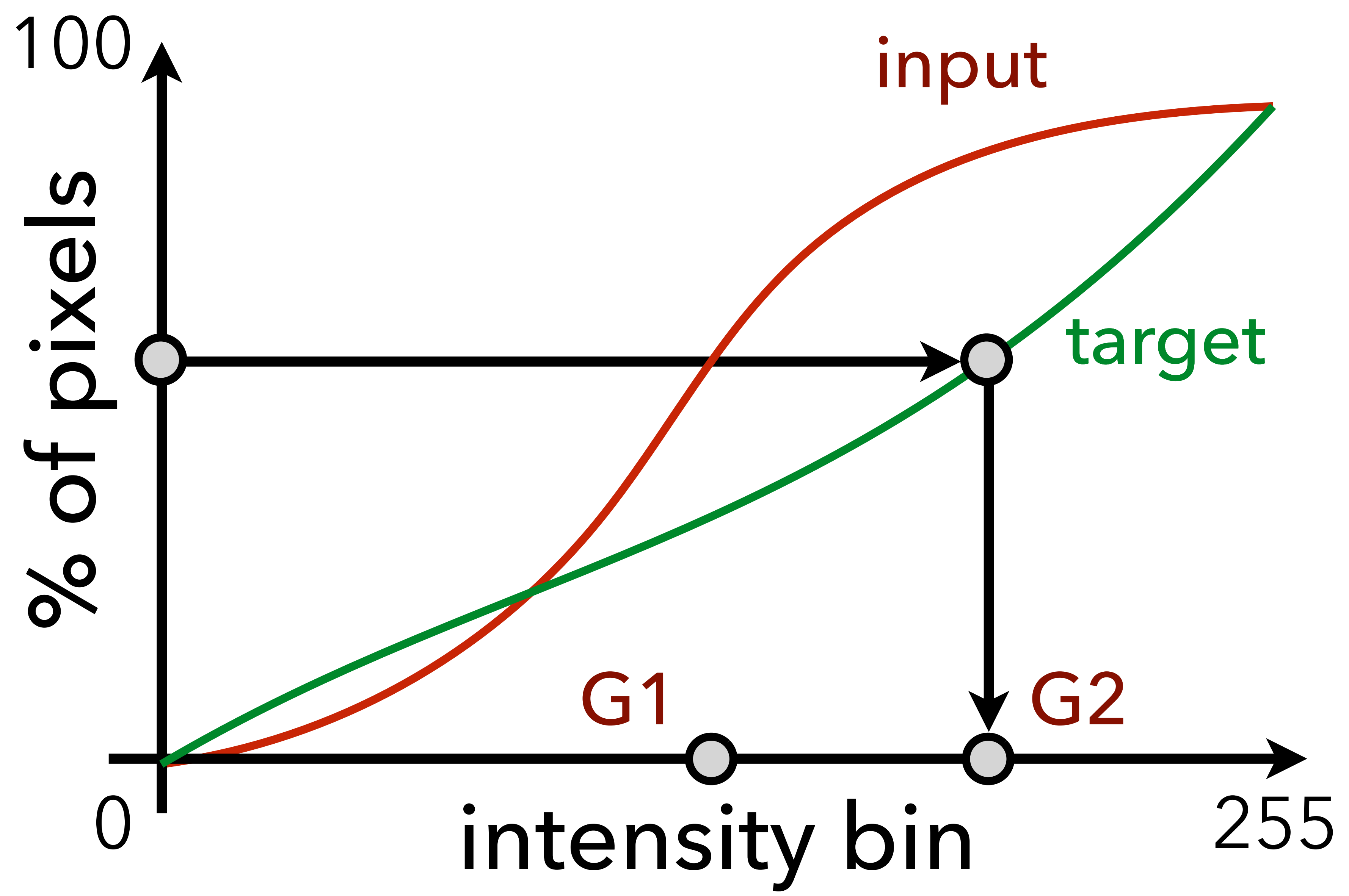
- Cumulative histogram of input



Histogram matching

Histogram matching is done by adjusting the cumulative distribution function (cdf)

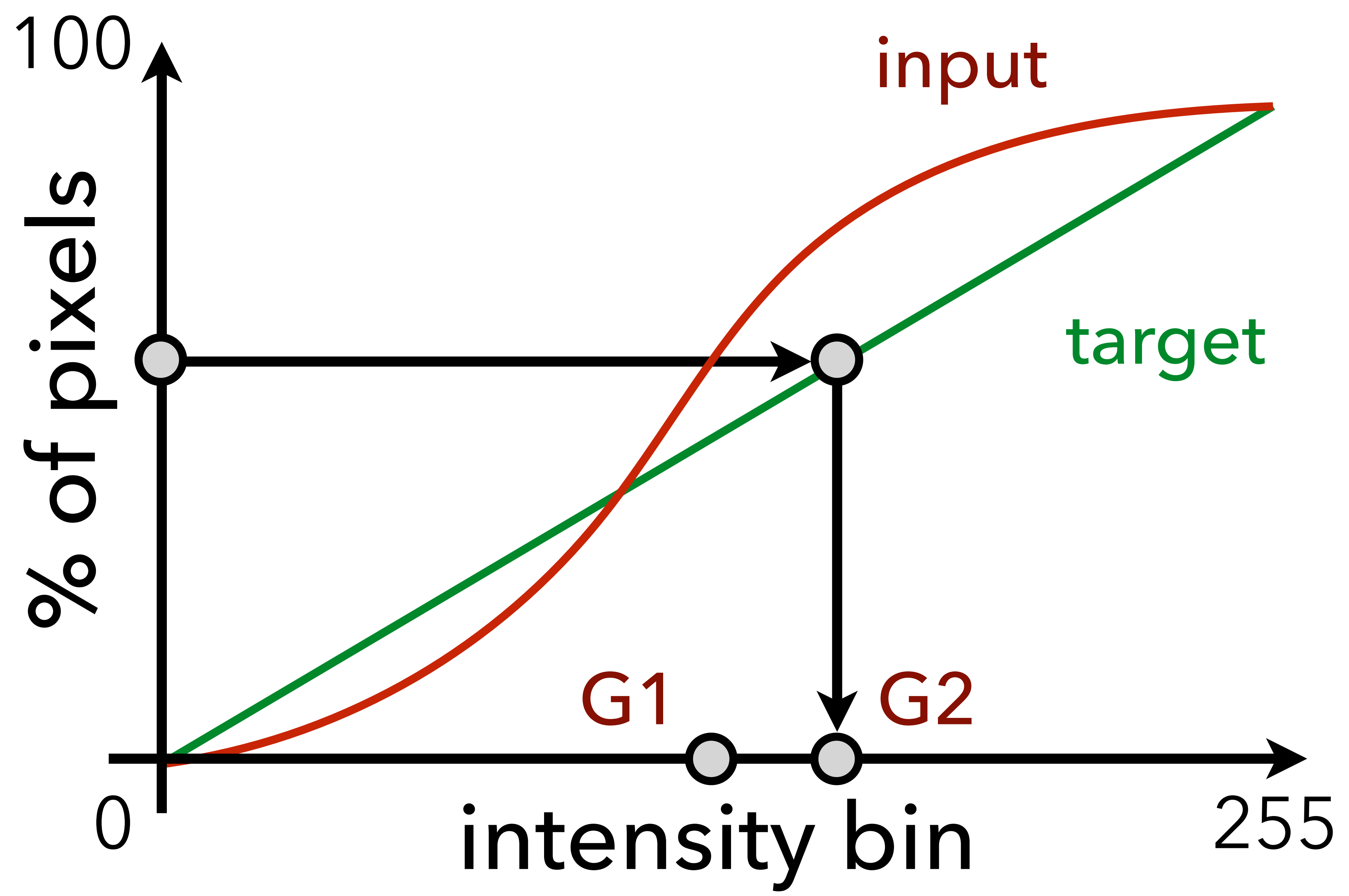
- Cumulative histogram of input
- Followed by inverse cumulative histogram of target



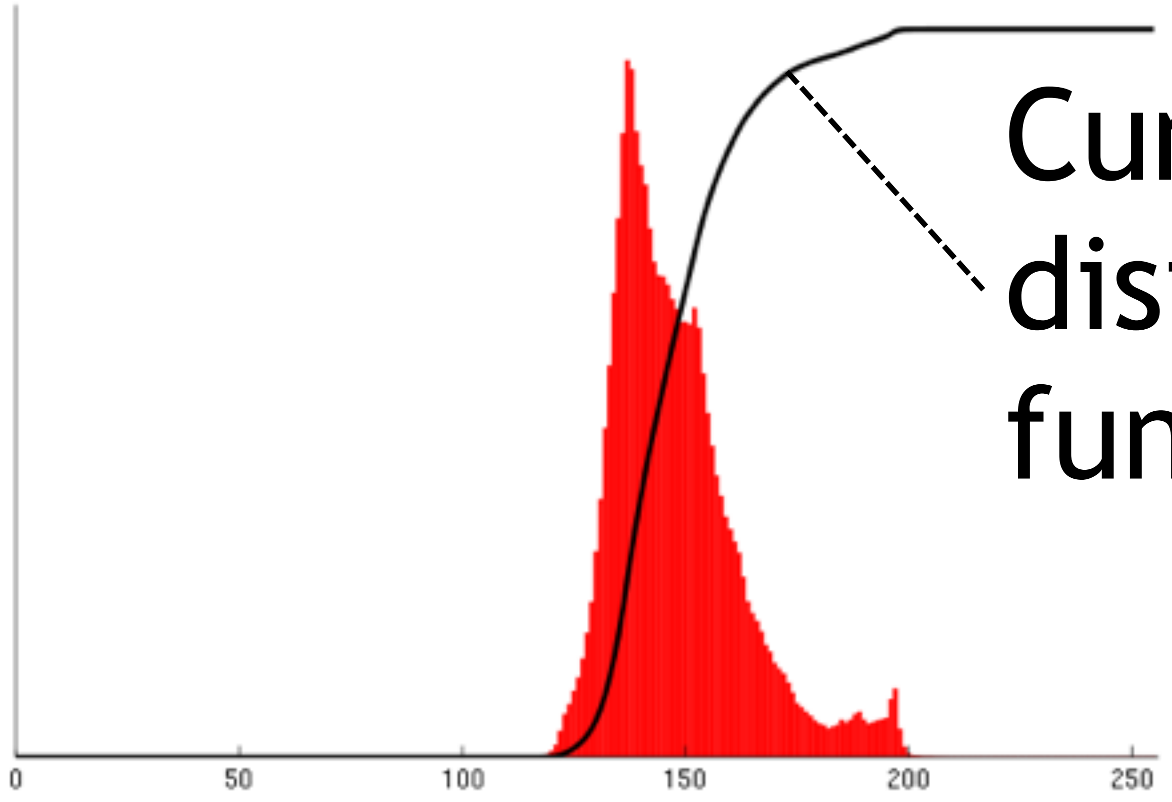
Histogram equalization

Histogram matching is done by adjusting the cumulative distribution function (cdf)

- Cumulative histogram of input
- Followed by inverse cumulative histogram of target (linear)



Histogram equalization



Cumulative distribution function (CDF)

Input

Equalized

Debugging

Debugging

Doubt everything

Debug pieces in isolation

- Binary search/divide and conquer

Display/print everything

- In particular intermediate results

Create simple inputs

- where you can easily manually compute the result
- e.g. constant image, edge image, etc.
- use small images (e.g. 3x3)
- including (especially) inputs to intermediate stages
- use input thats isolate different failure modes

Change one thing at a time

- e.g. to verify that a given command has the effect you want, modify it to break it

Slide credits

Frédo Durand