Image Processing

Lecture 02 – Image Fundamentals

Yeong Jun Koh

yjkoh@cnu.ac.kr

Image Processing

Lecture 02

- **Digital image**
- Point processing

Image sensor

Image sensor

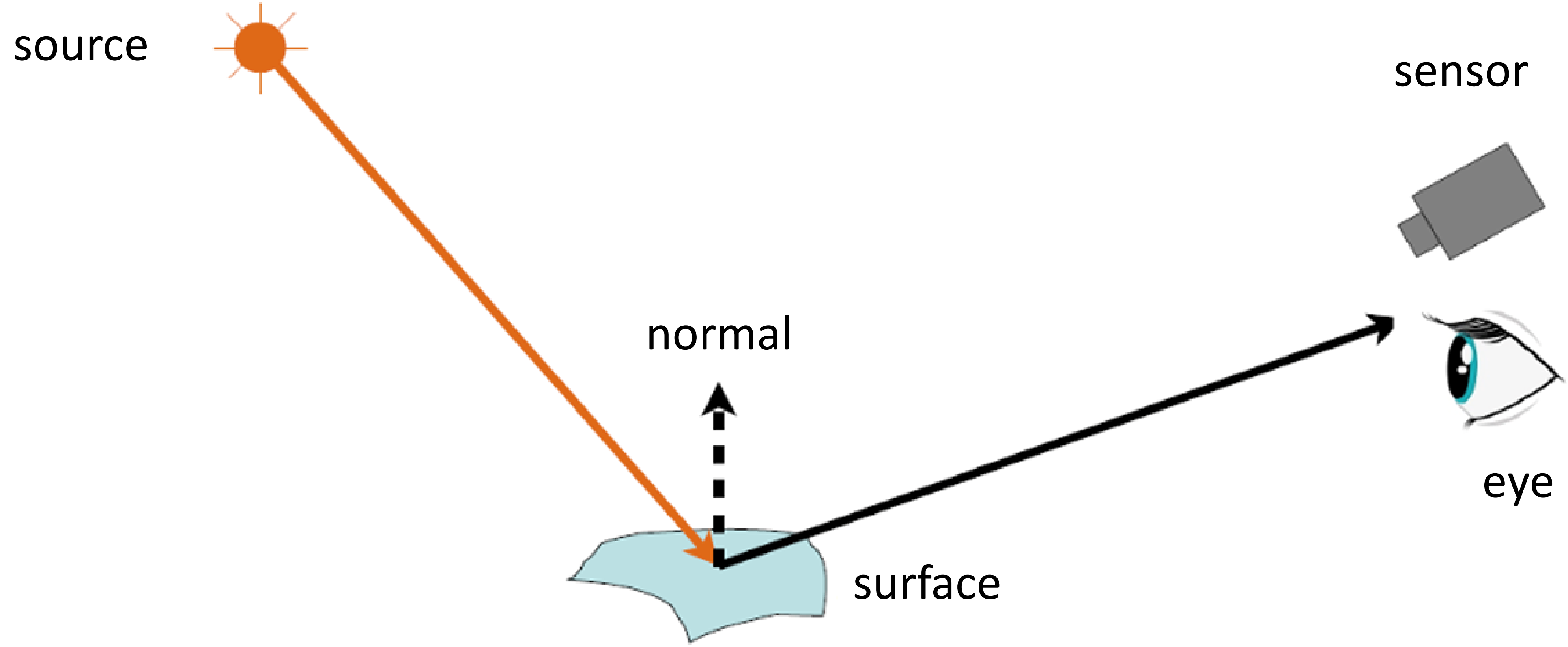
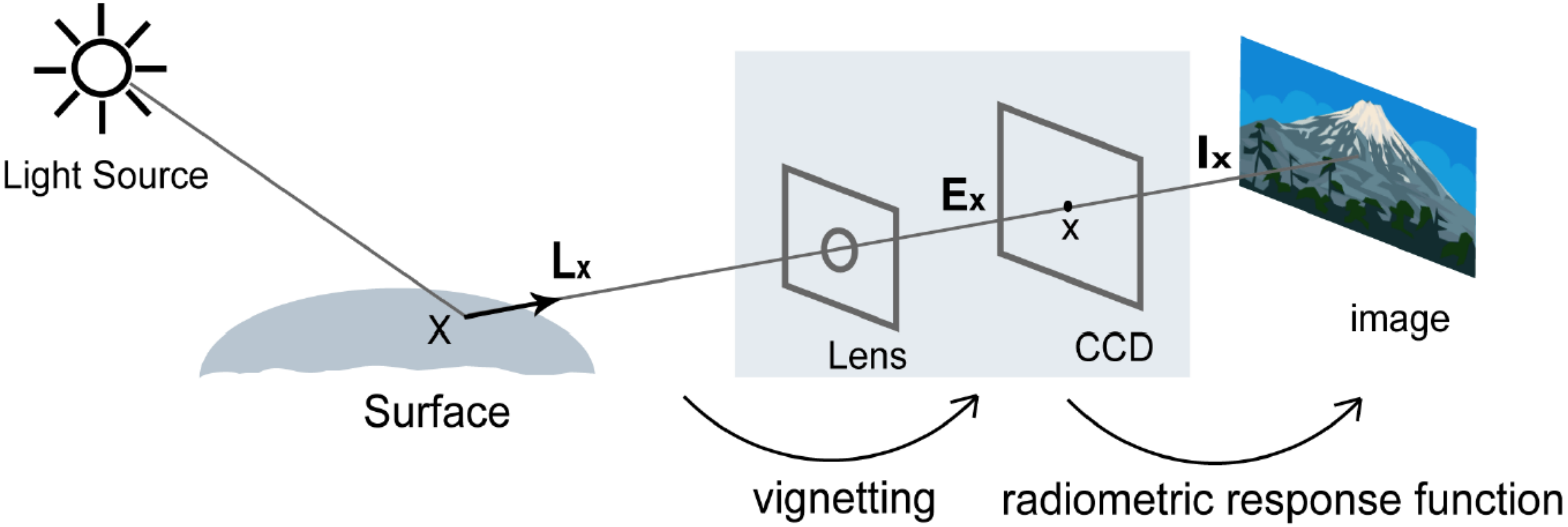


Image sensor **captures amount of light** from the object

Slide Credit: Ioannis Gkioulekas (CMU)

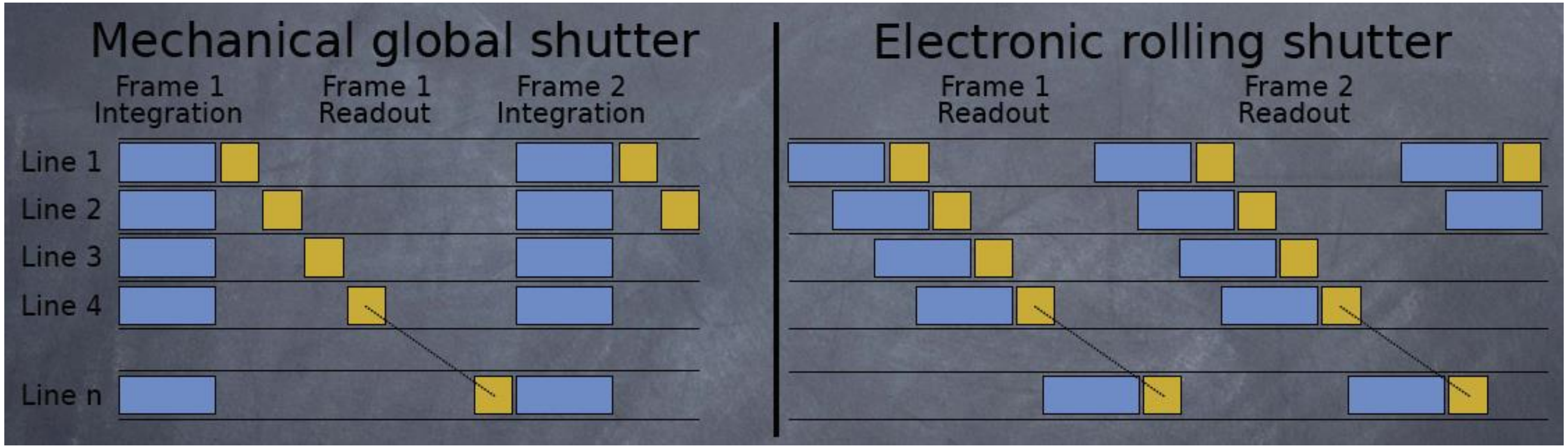
Image sensor



- CCD (Charge coupled device) camera
  - High-end DSLR camera
- CMOS (Complementary metal-oxide semiconductor) camera
  - Smartphone

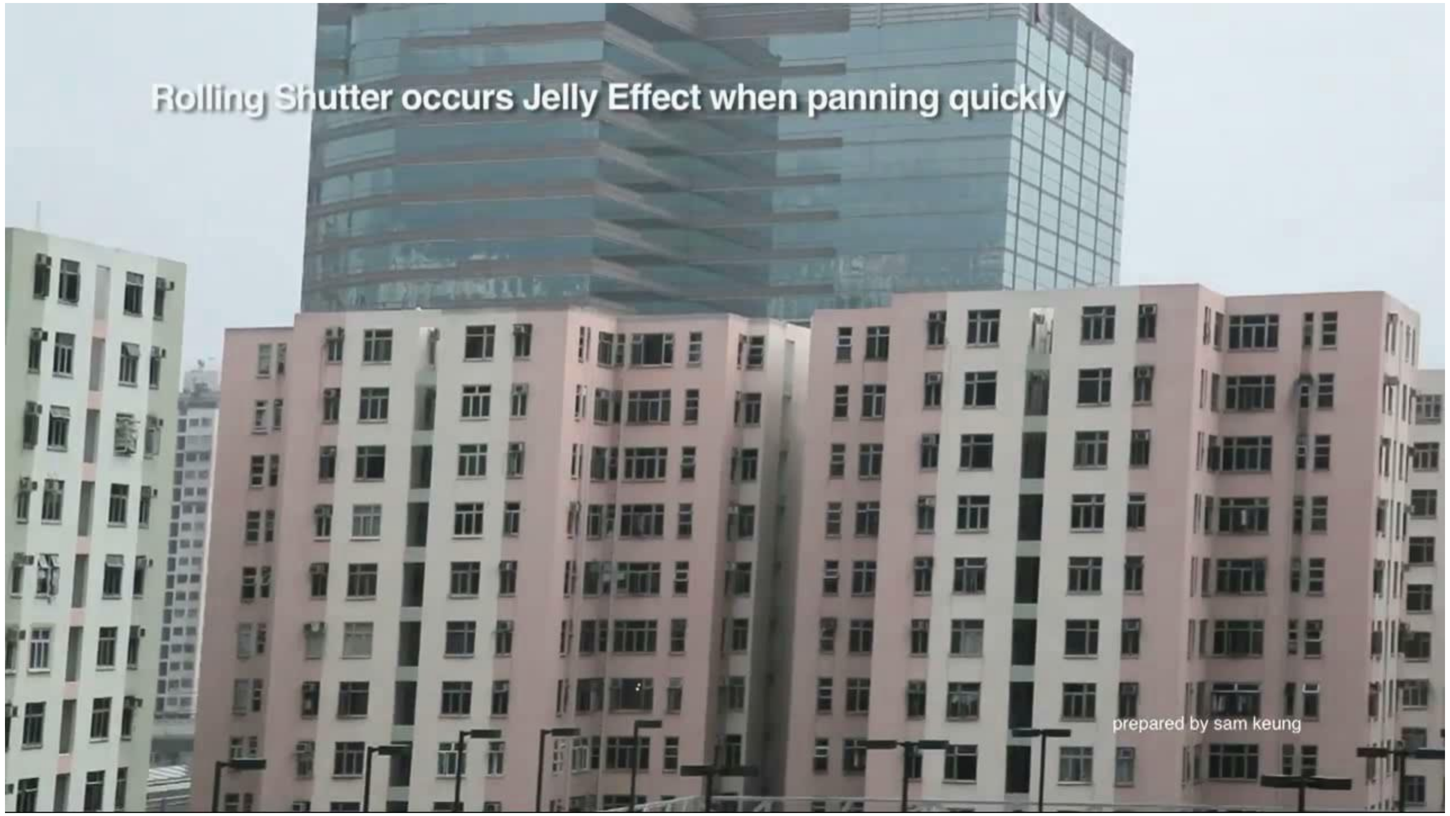
CCD vs. CMOS

- Global(CCD) vs. rolling(CMOS) shutter

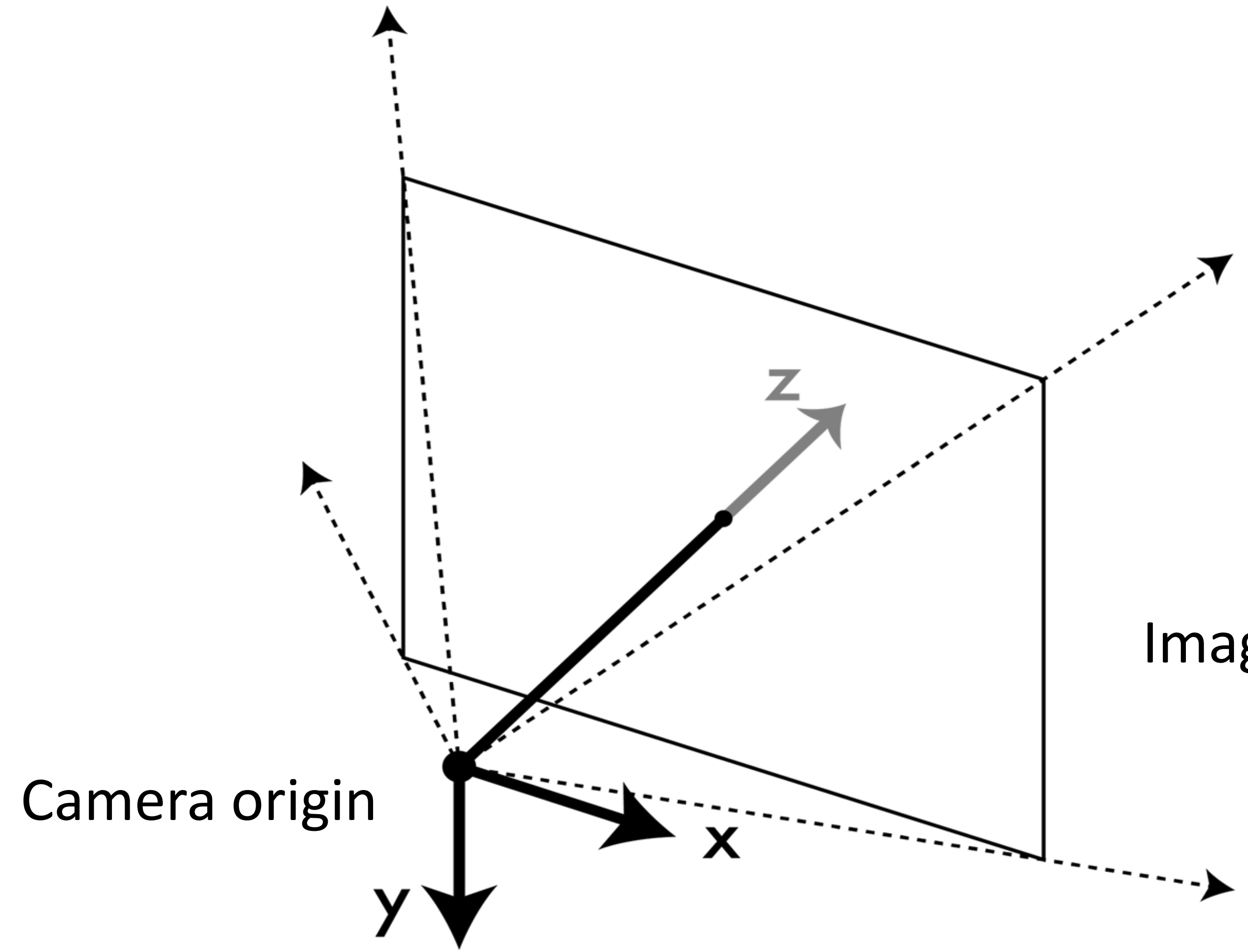


Rolling shutter

- Distortions
  - Skew

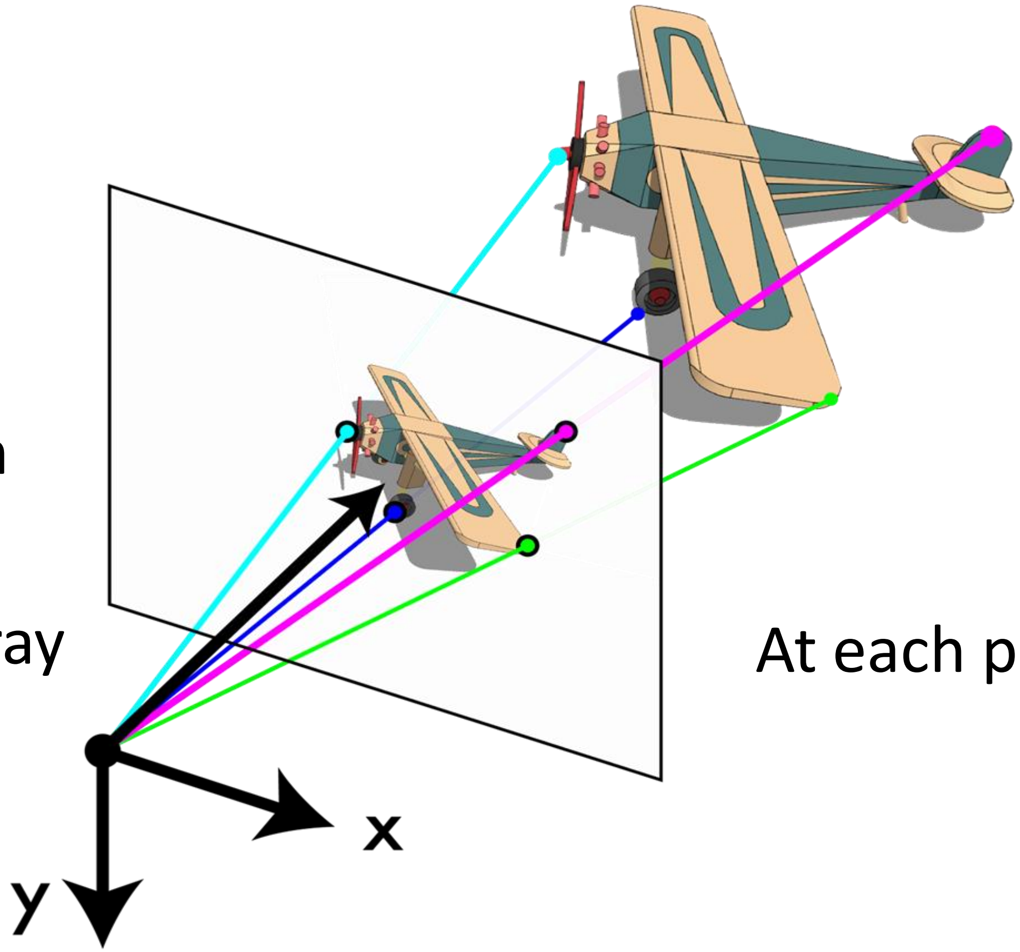


Perspective projection: from 3d world to 2d image



Slide Credit: Joseph Redmon (Washington)

Perspective projection: from 3d world to 2d image



object in 3D world

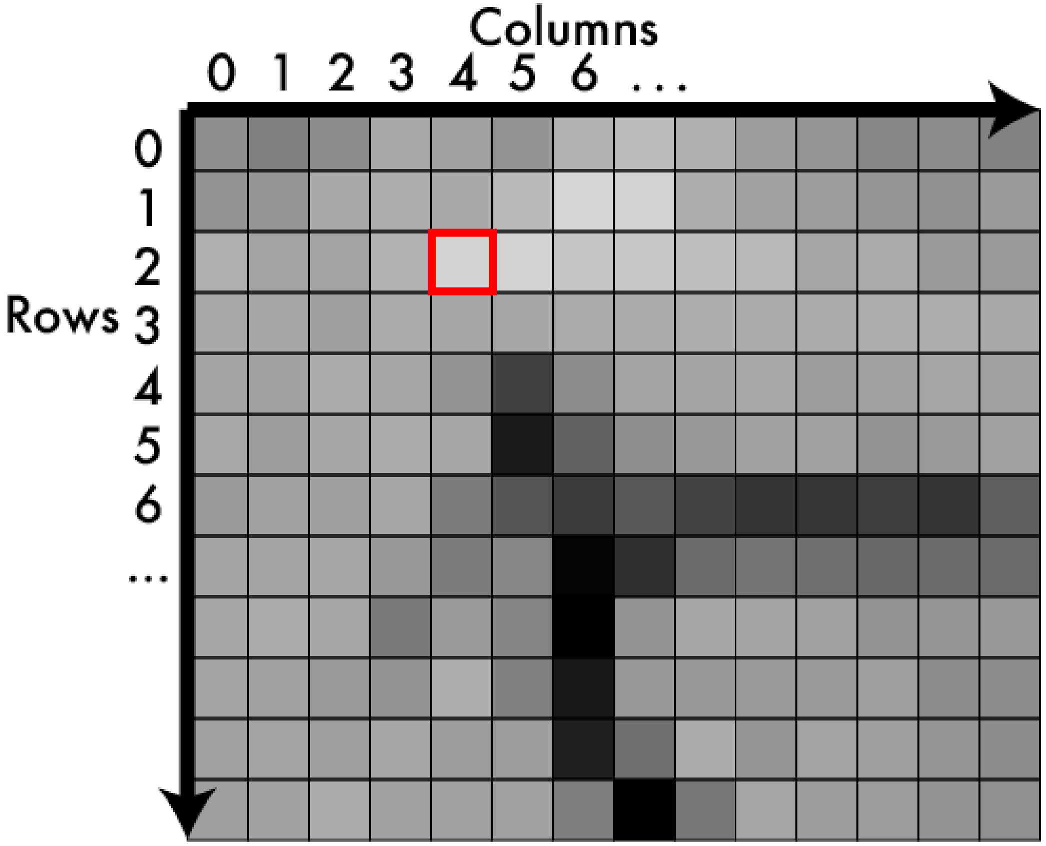
projection
onto 2D
sensor array

At each point sensor records incident light

Slide Credit: Joseph Redmon (Washington)

Image: 2d array of light

- Each point in matrix called a pixel
  - A pixel has a 2D coordinates
  - A pixel has a value (intensity)



Slide Credit: Joseph Redmon (Washington)

Image: 2d array of light

- A pixel value indicates amount of light
  - Higher pixel value = more light
  - Lower pixel value = less light

- A pixel value is bounded:
  - No light (black) = 0
  - Sensor limit (white) = max
  - Typical ranges:
    - [0-255], fit into byte
    - [0-1], floating point

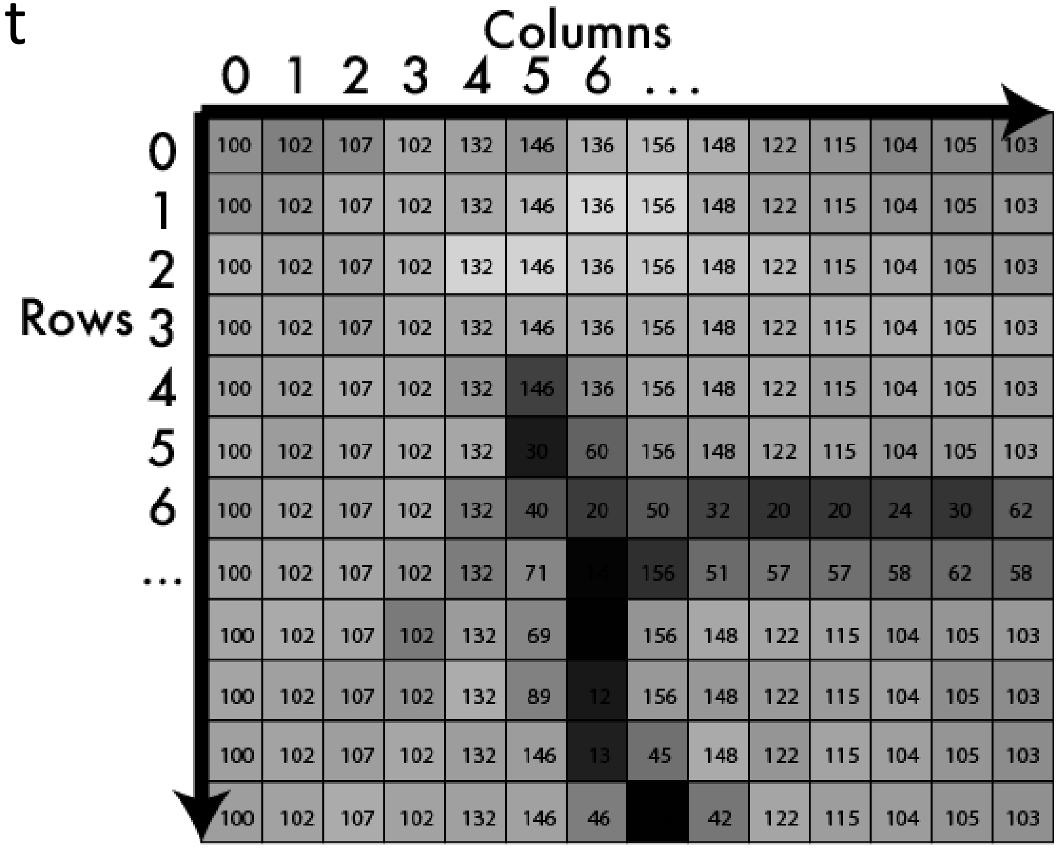


Image Acquisition Process

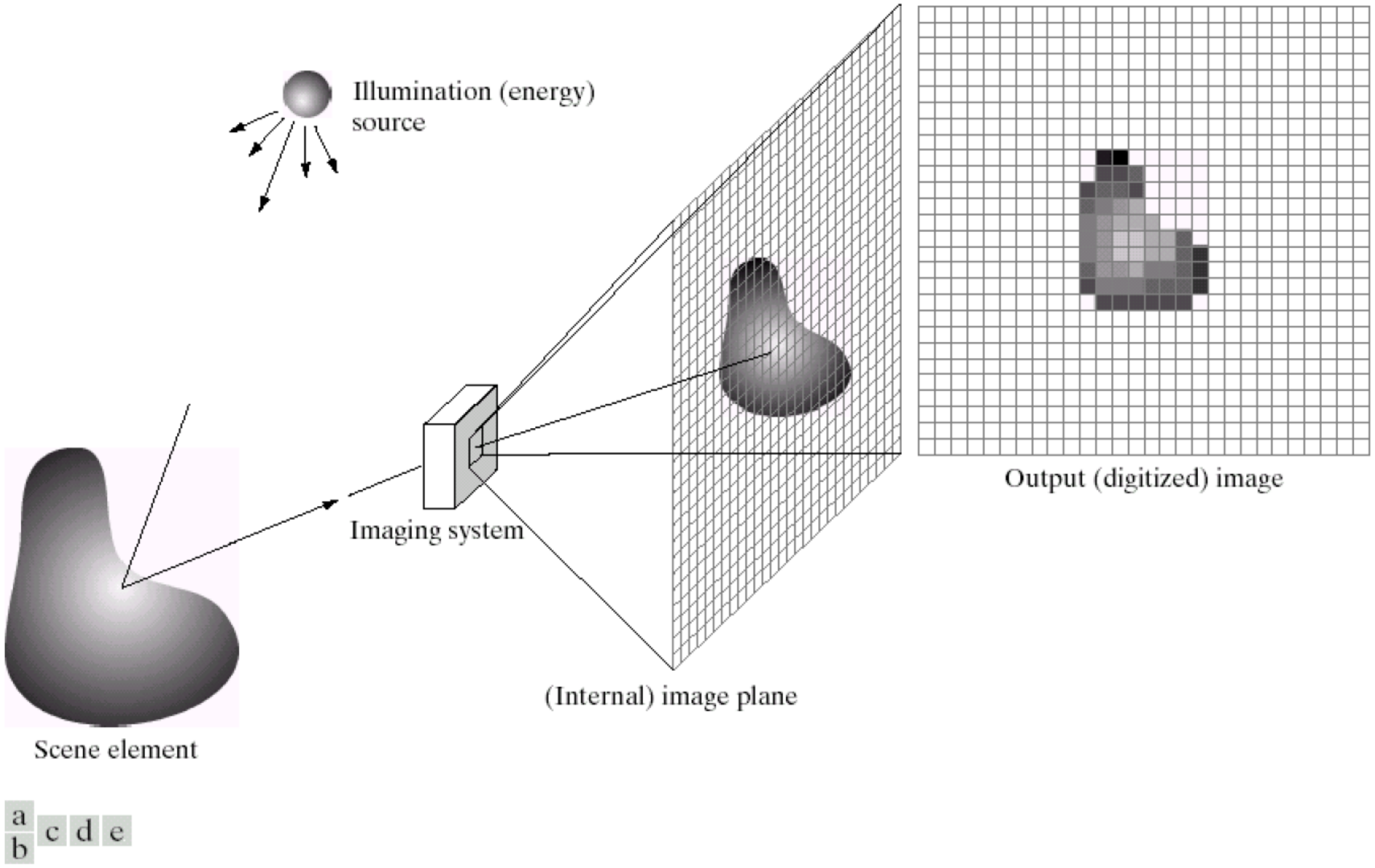


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

From Rafael C. Gonzalez, "Digital image Processing," 2Th Edition

Sensor Array

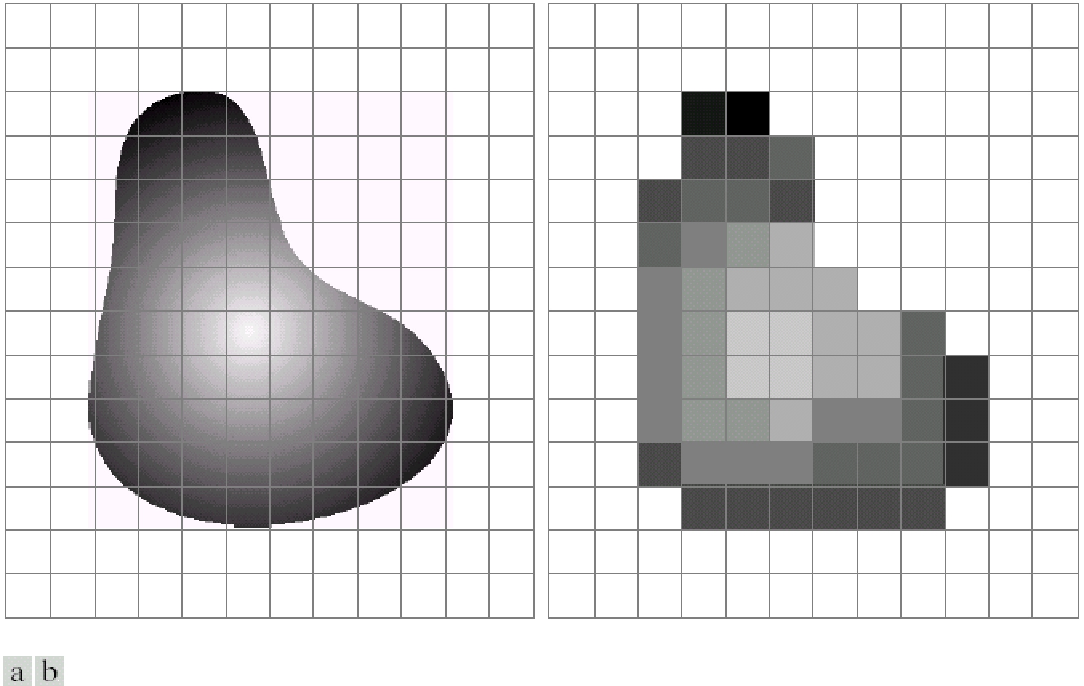
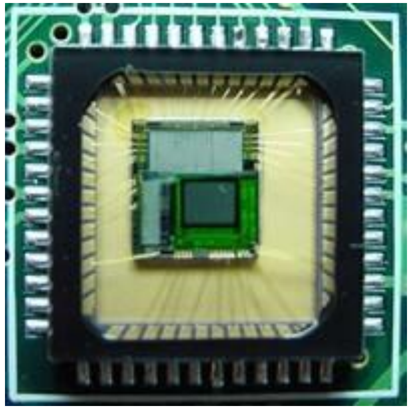


FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.



CMOS sensor

From Rafael C. Gonzalez, “Digital image Processing,” 2Th Edition

Sampling and Quantization

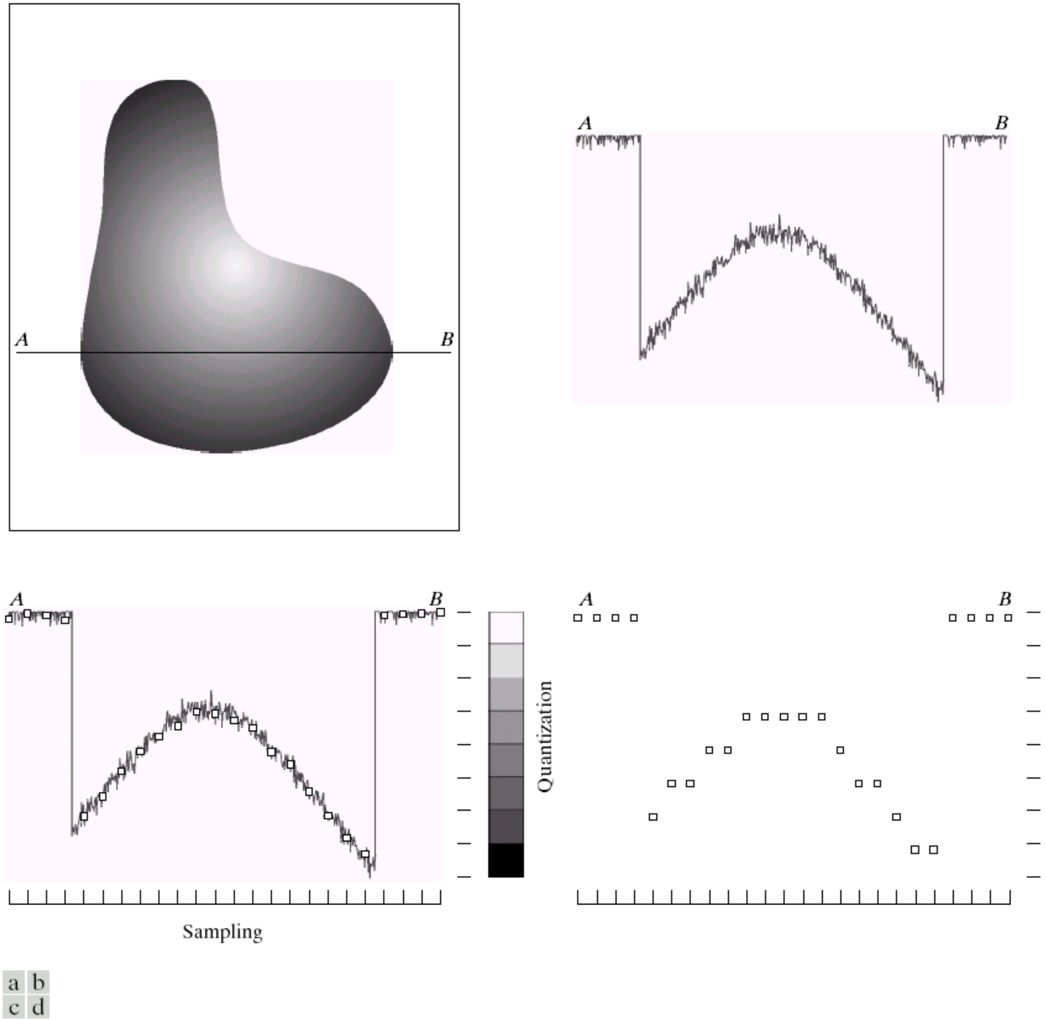


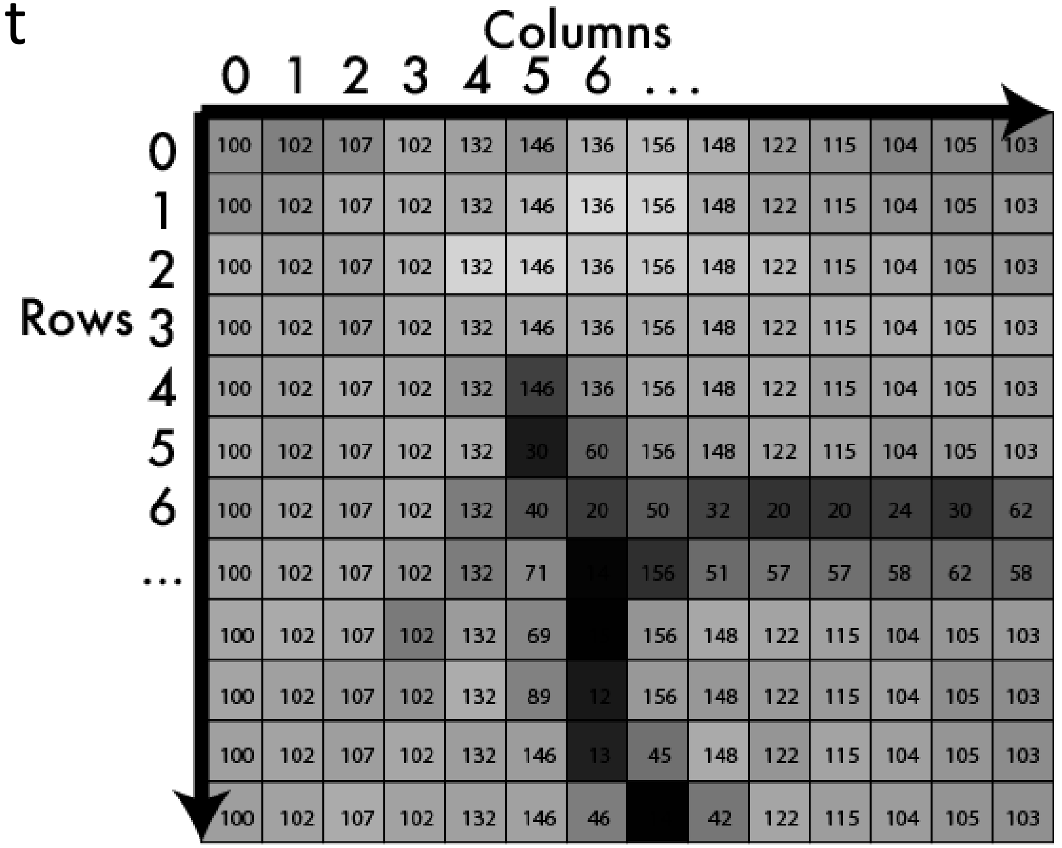
FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

From Rafael C. Gonzalez, “Digital image Processing,” 2Th Edition

Image: 2d array of light

- A pixel value indicates amount of light
  - Higher pixel value = more light
  - Lower pixel value = less light

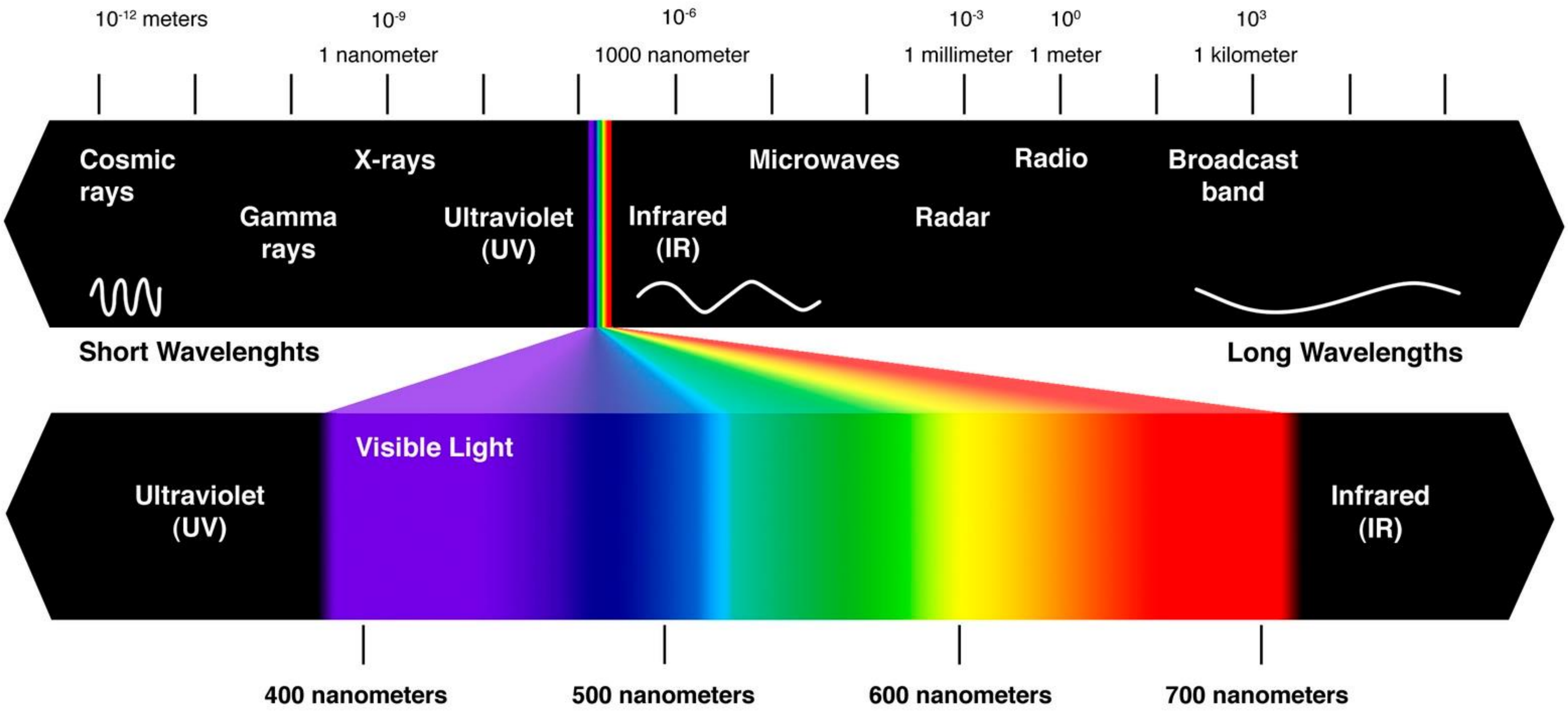
- A pixel value is bounded:
  - No light (black) = 0
  - Sensor limit (white) = 255
  - Typical ranges:
    - [0-255], fit into byte
    - [0-1], floating point



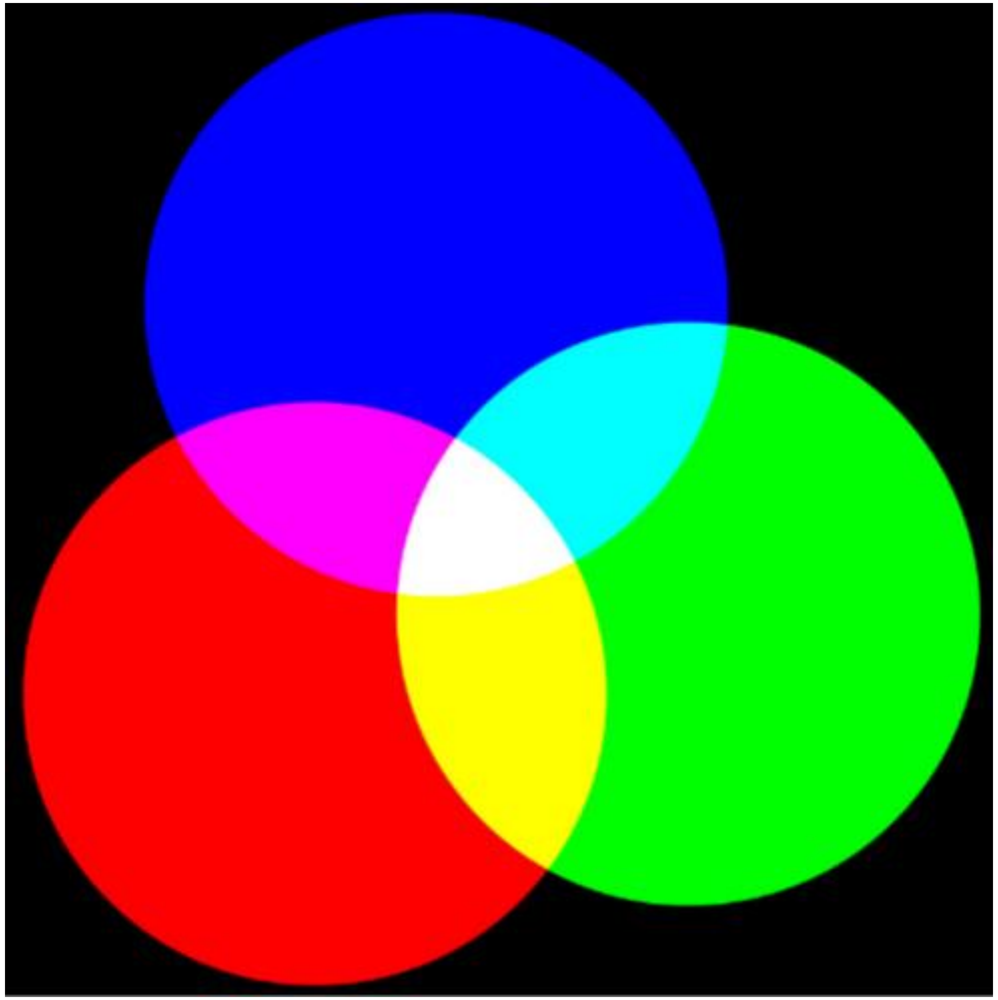
Slide Credit: Joseph Redmon (Washington)

An image is a 2d array of number (matrix)

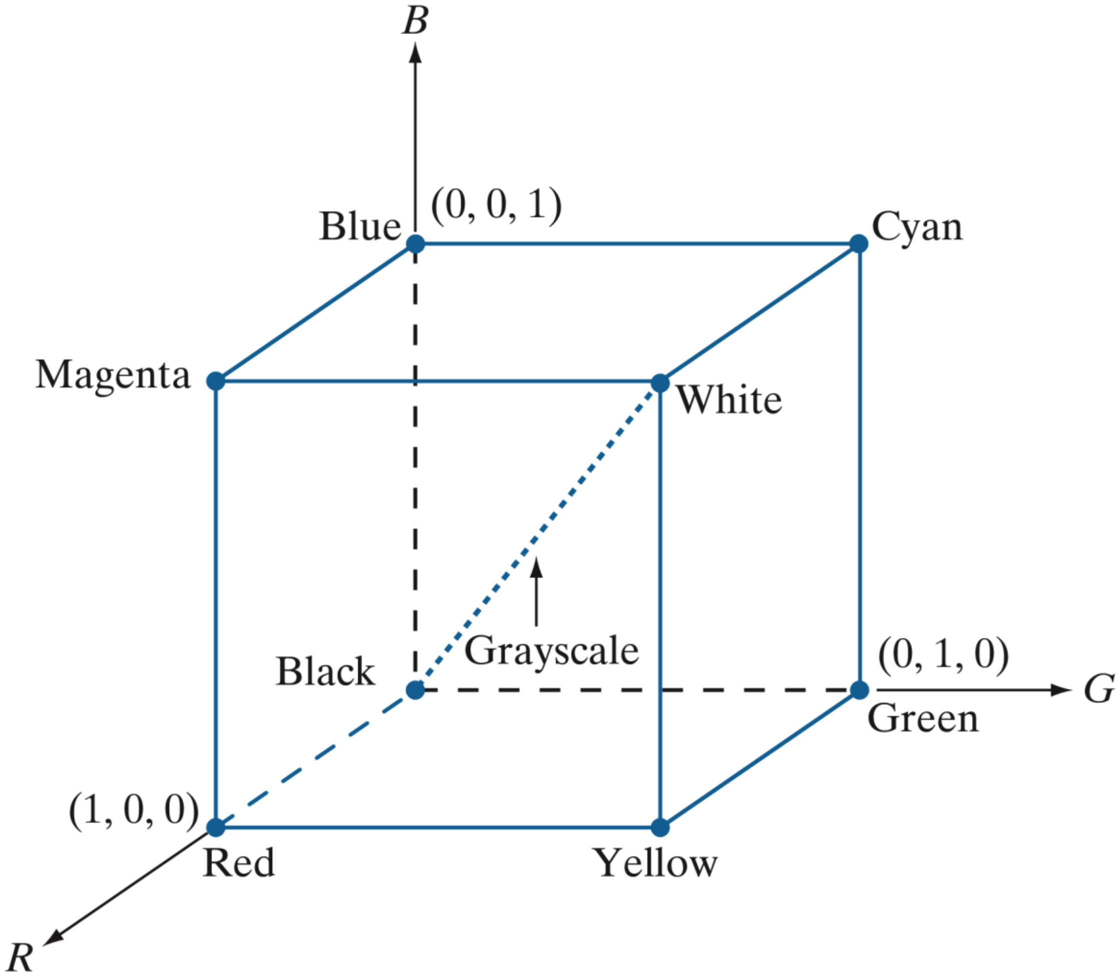
How to record color?



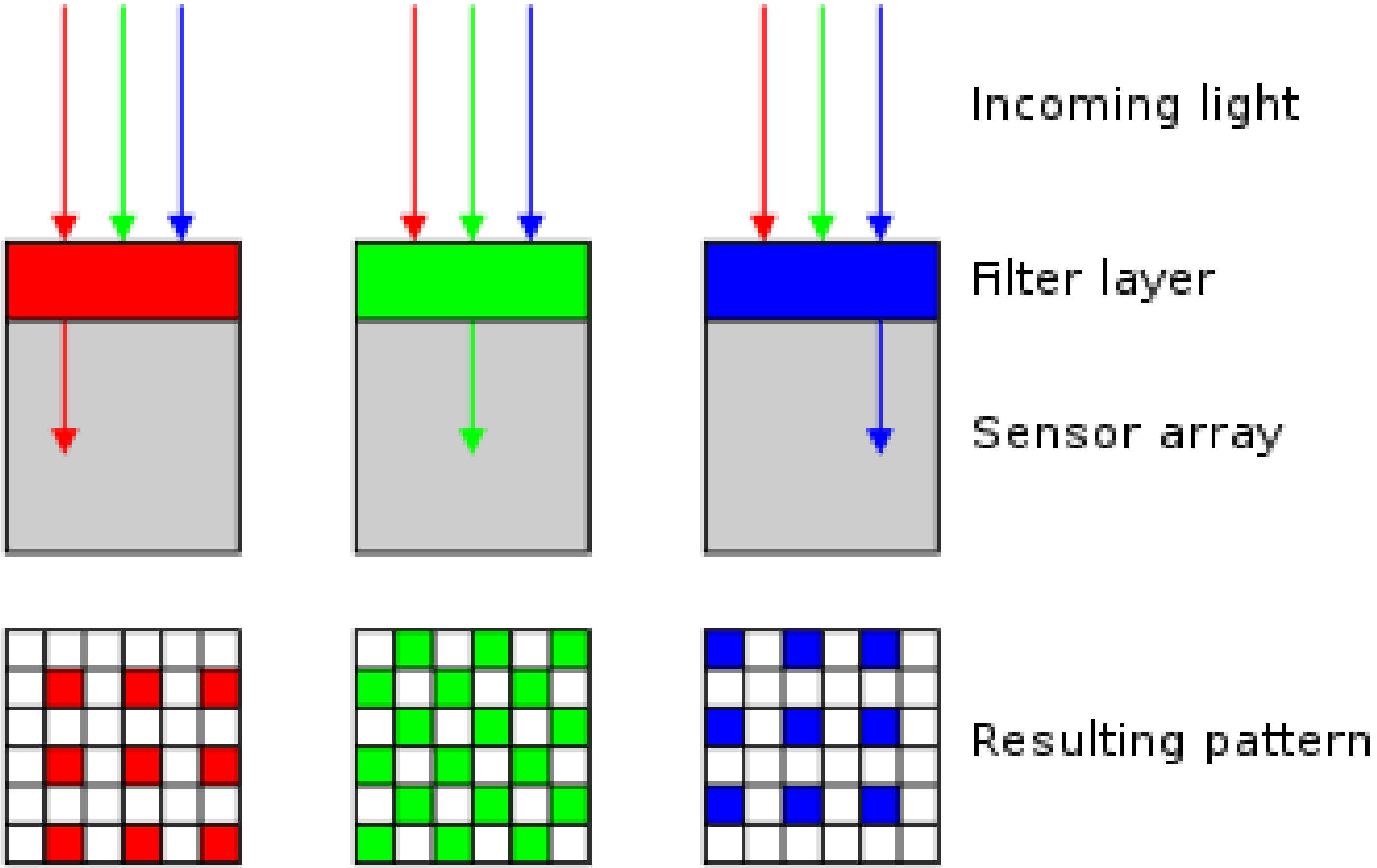
How to record color?

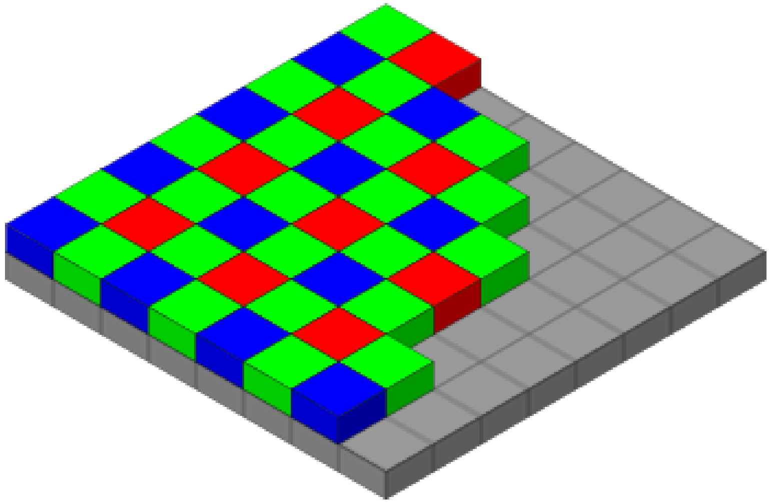


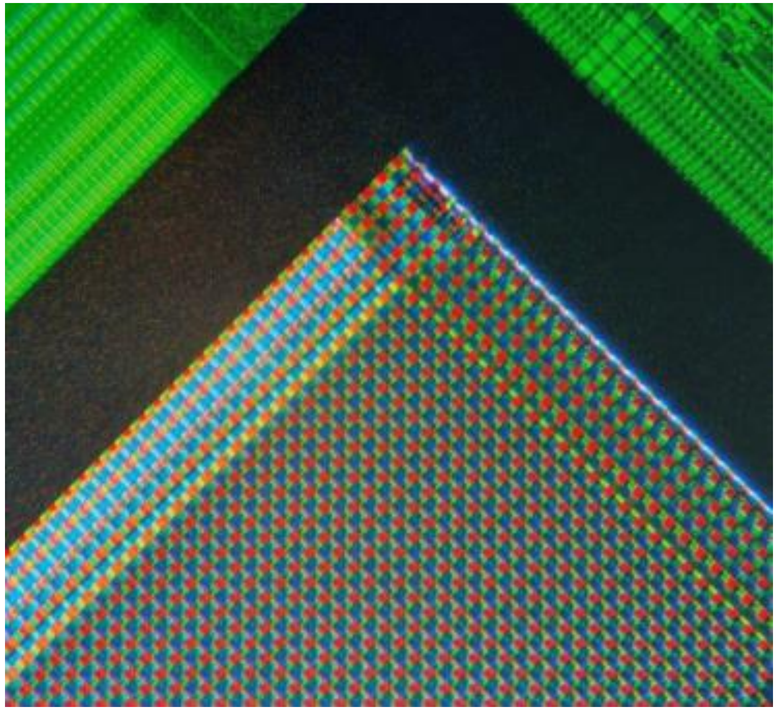
RGB: Primary color of light



Schematic of the RGB color cube

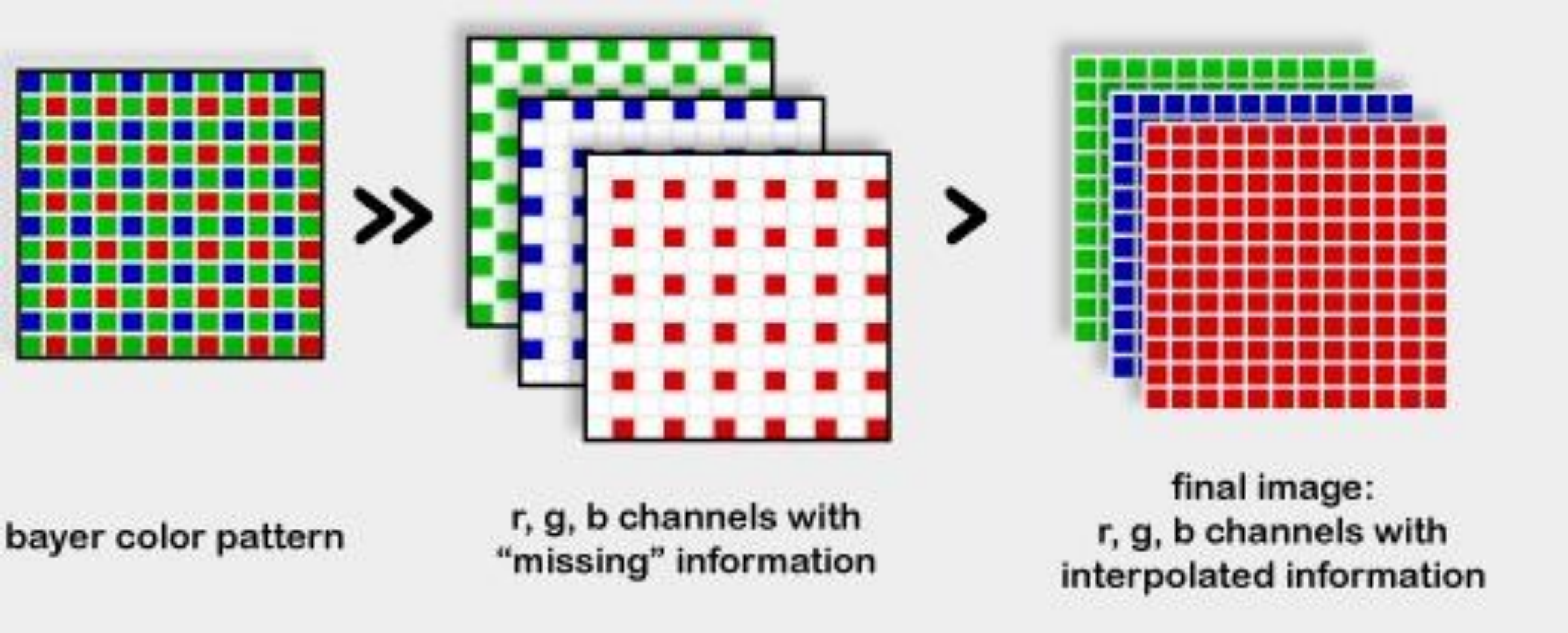






Bayer color pattern for image sensor

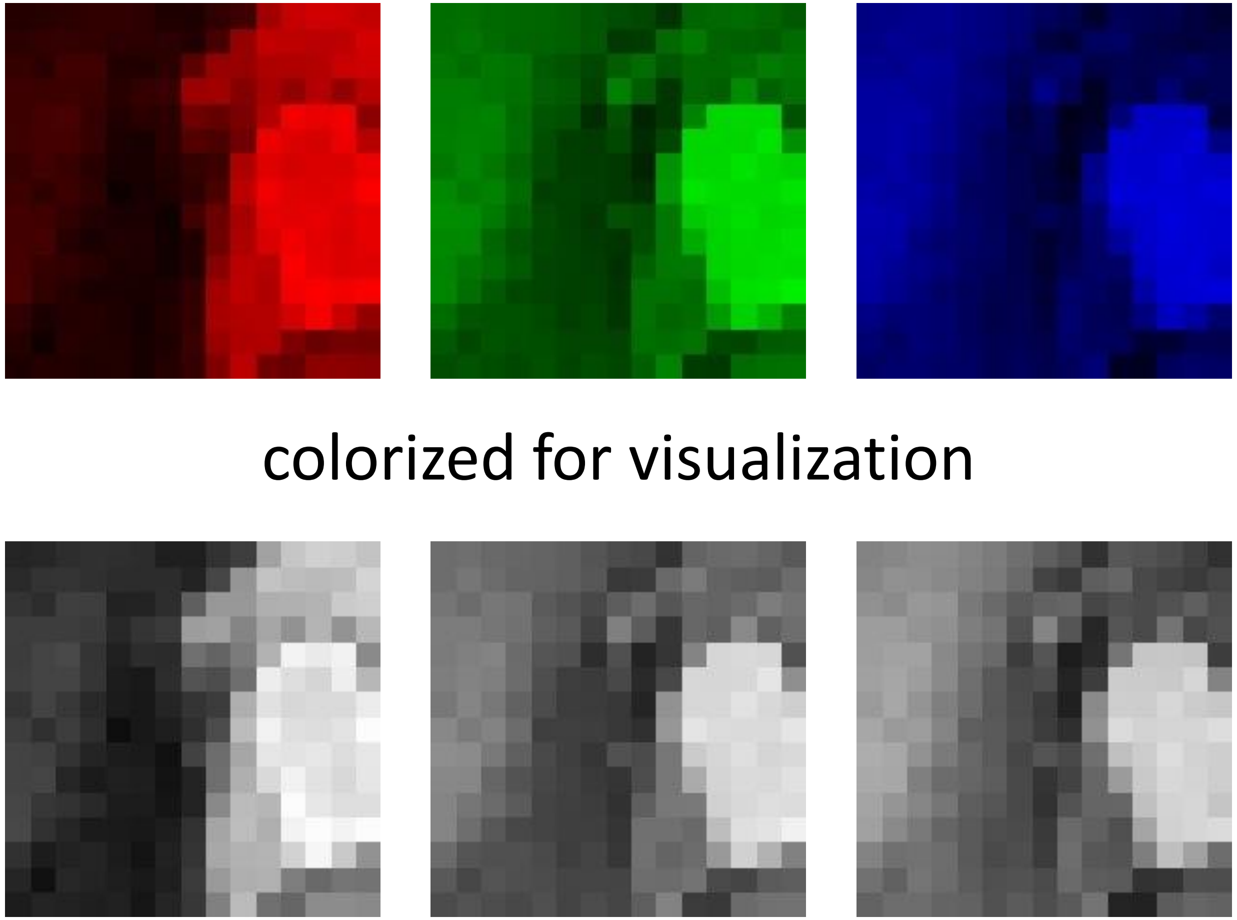
https://en.wikipedia.org/wiki/Bayer_filter


bayer color pattern
r, g, b channels with
“missing” information
final image:
r, g, b channels with
interpolated information

A color image is a 2D array of color (3D tensor)

color image patch

red green blue

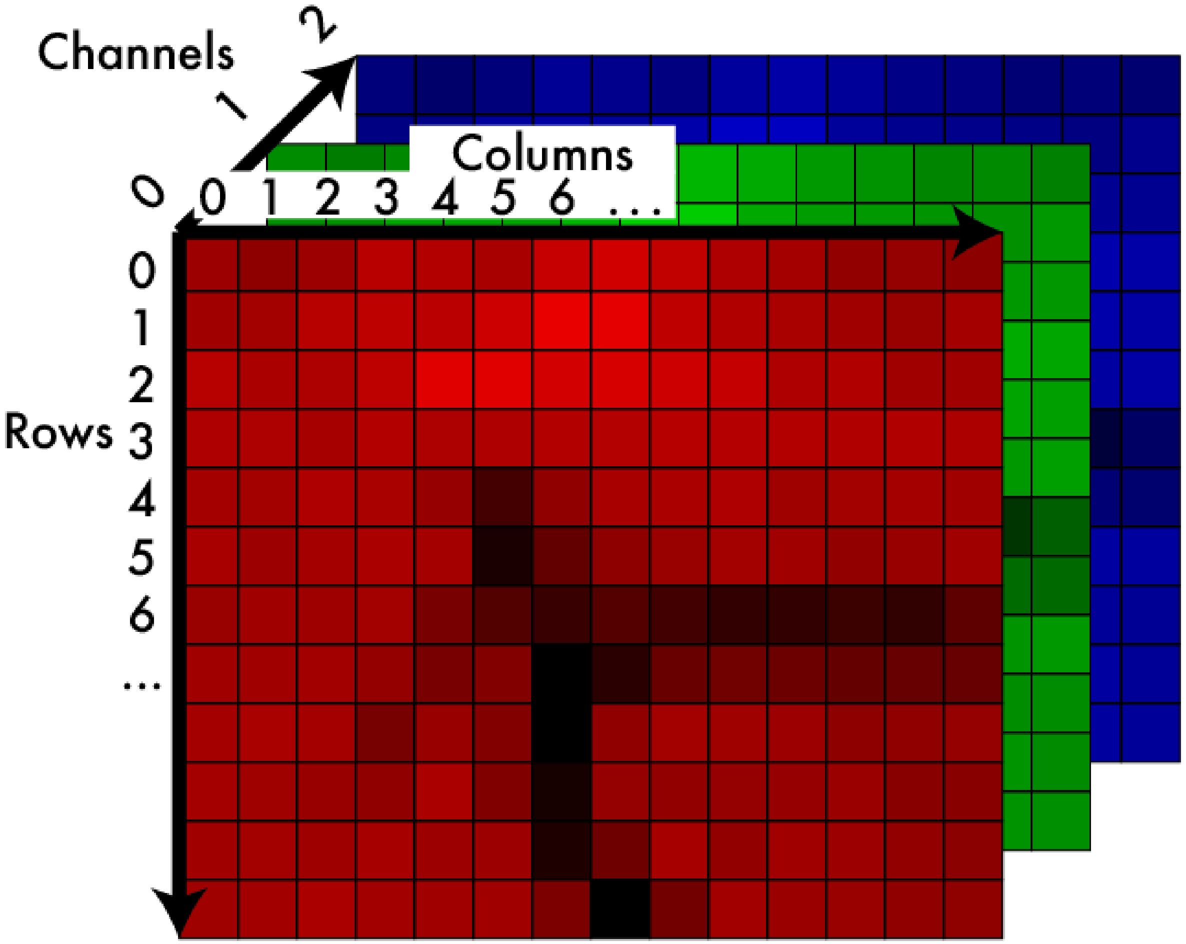


colorized for visualization

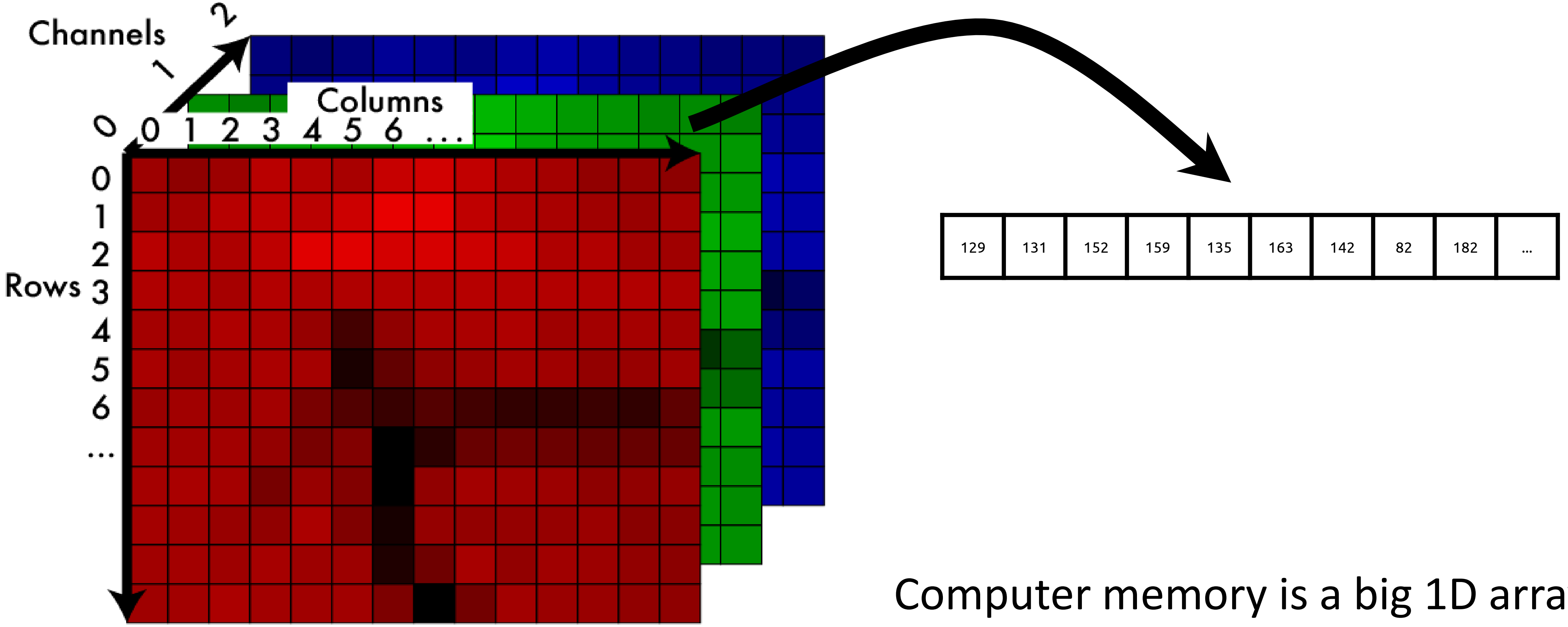
actual intensity values per channel

Each channel is a 2D array of numbers

Color image: 3d tensor in color space



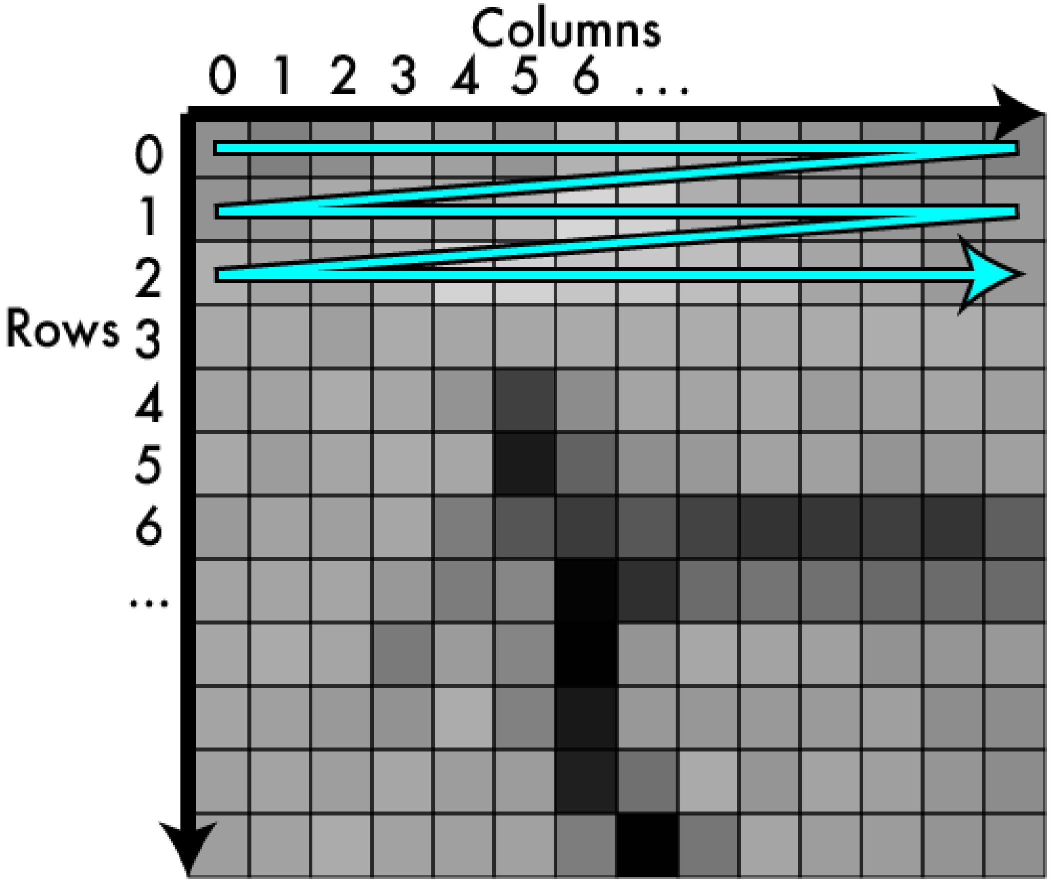
How do computer store them?



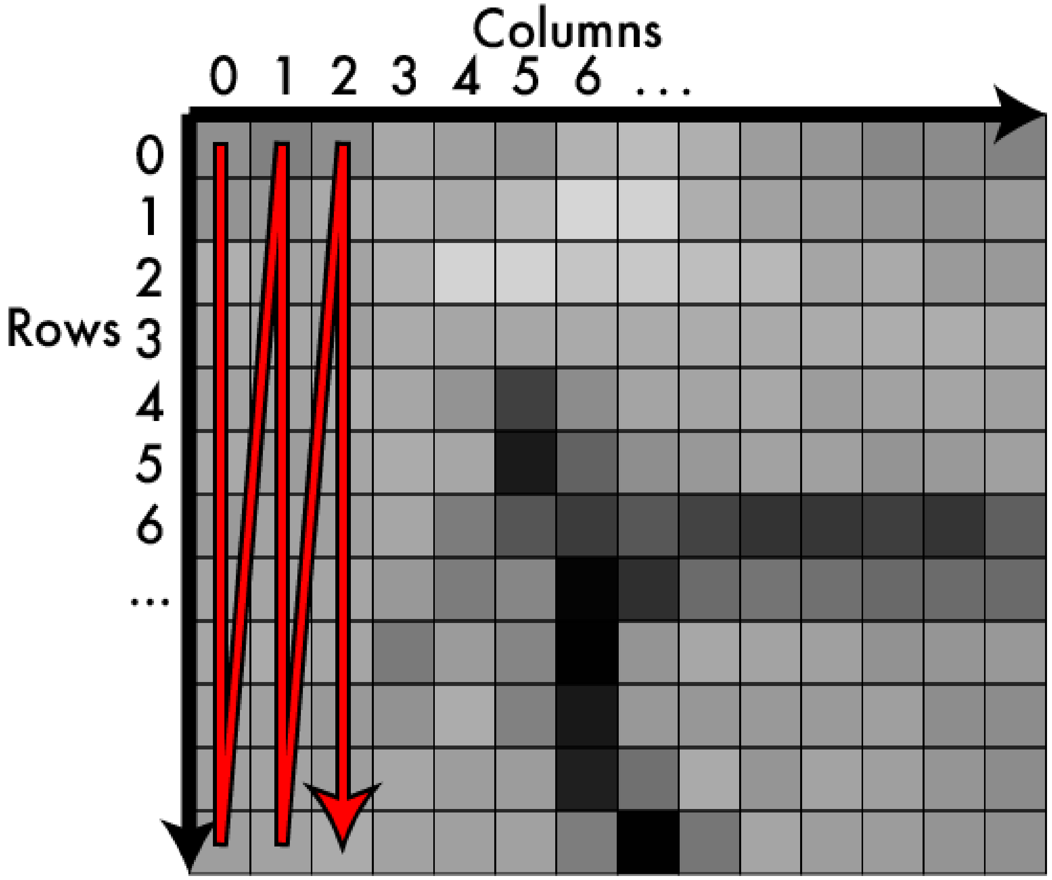
Computer memory is a big 1D array

Storage: row major vs column major

Row major (H, W)

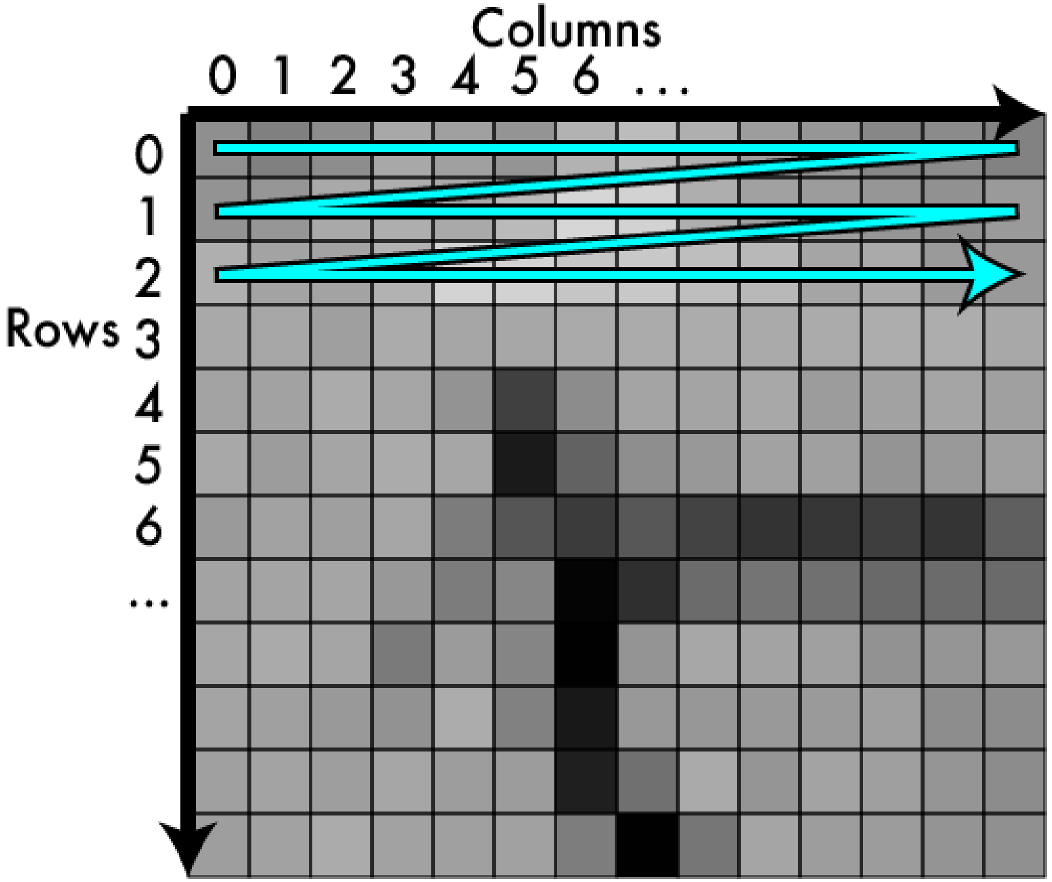


Column major (W, H)



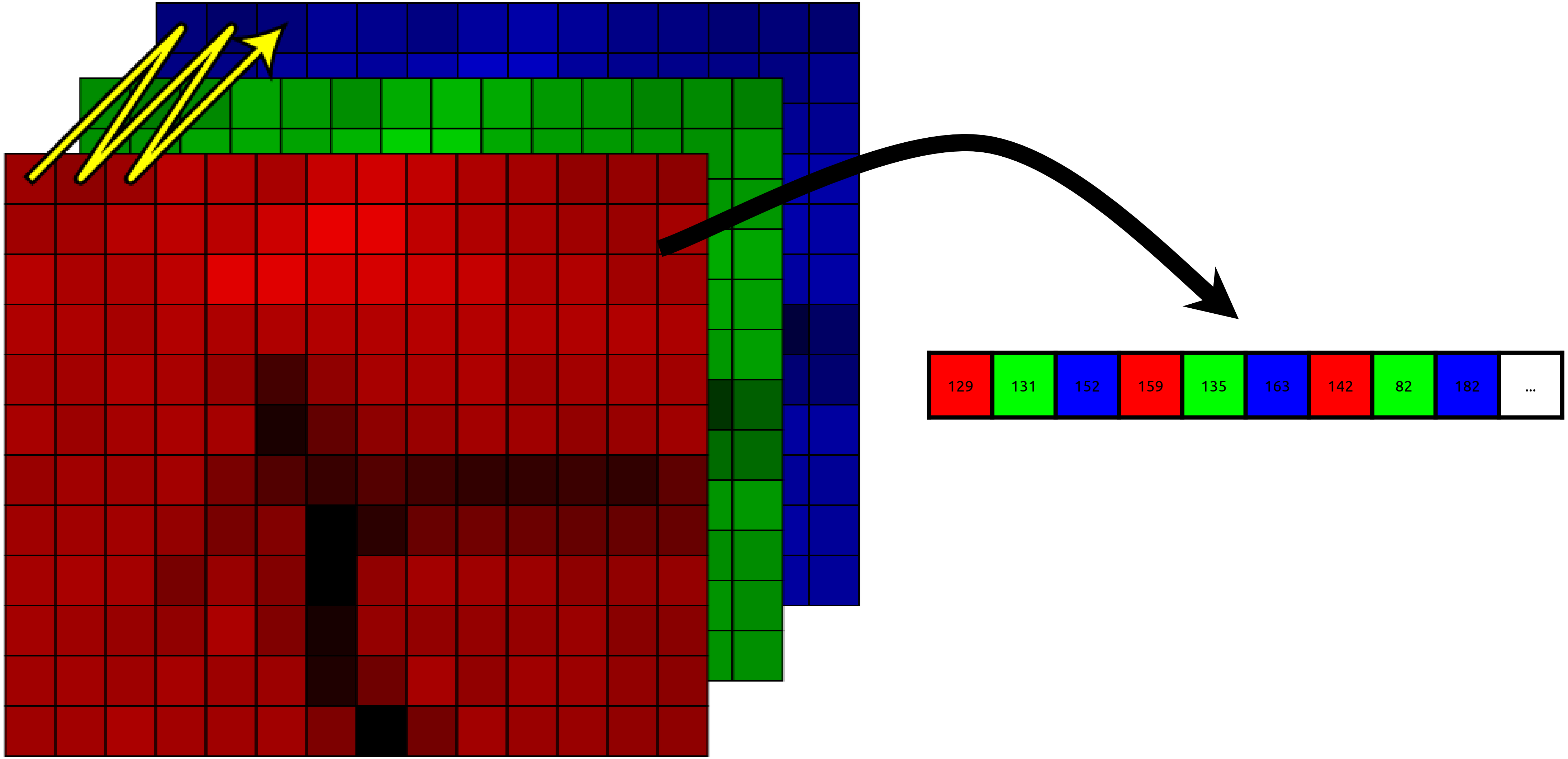
Storage: row major vs column major

Row major (H, W)



Most programming languages assume the row major order system

(H,W,C): channels interleaved



(C,H,W): channels separated

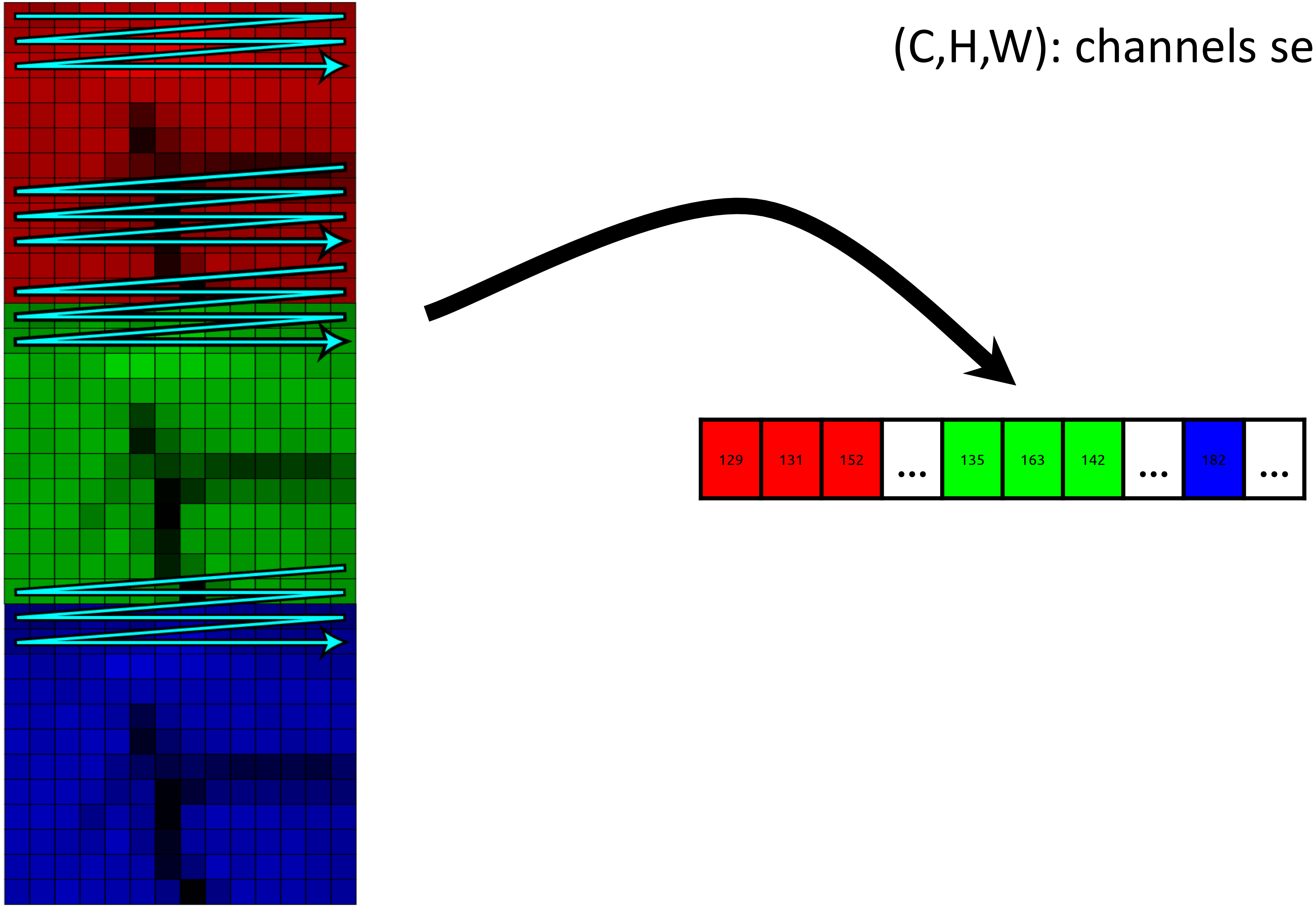
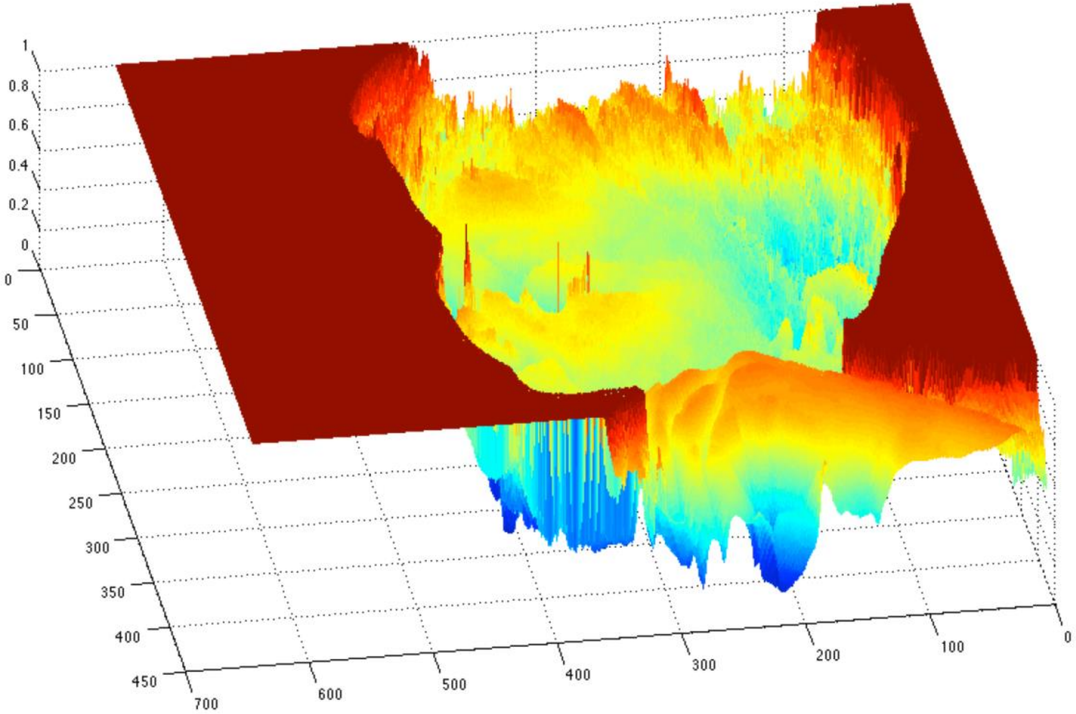


Image as a 2D function

range $f(\mathbf{x})$

grayscale image



An image is a 2D function

domain $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

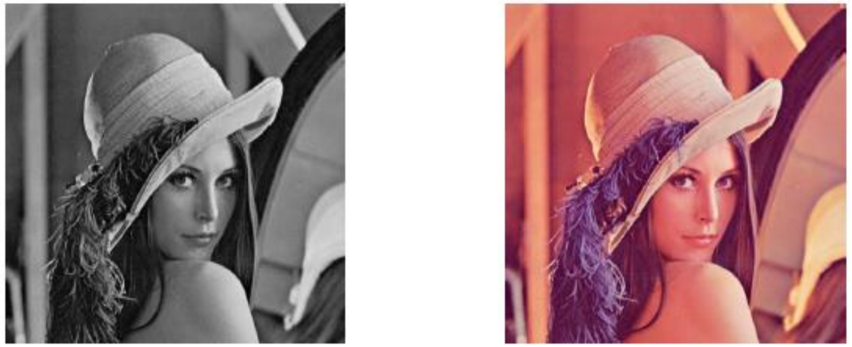
Video?

- Digital Video
  - The set of images
  - Frame rate
    - The number of images (frames) of a video per second
    - Frame per second (FPS)

https://www.mediacollege.com/video/frame-rate/

Image as a 2D function

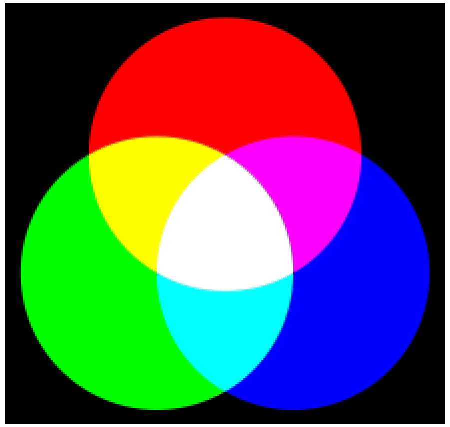
- 2-dimentional function $f(x, y)$
- (x, y): 2-D spatial coordinate
- f: 1-D scalar (gray image), 3-D vector (color image: RGB)
  - Each value is in (0. 255)



- $f(x, y, t)$: video sequence
- $f(x, y, z)$: 3-D object
- $f(x, y, z, t)$: moving 3-D object

Primary Colors of Light

- Primary colors
  - Red, Green, Blue
  - Each color can be represented by a 3D vector, e.g.
    - R = (1, 0, 0)
    - G = (0, 1, 0)
    - B = (0, 0, 1)



- Secondary colors
  - Cyan = G+B = (0, 1, 1)
  - Magenta = B+R = (1, 0, 1)
  - Yellow = R+G = (1,1, 0)

- W = (1,1,1) = R+G+B

From https://ko.wikipedia.org/wiki/RGB

Perceptual Aspects of Color

- **Luminance** is a quantity defining (approximately) the *brightness* by which humans *perceive* different colors
  - e.g. for RGB color base, a common way of computing luminance *Y* is

$$Y = (R + G + B)/3$$

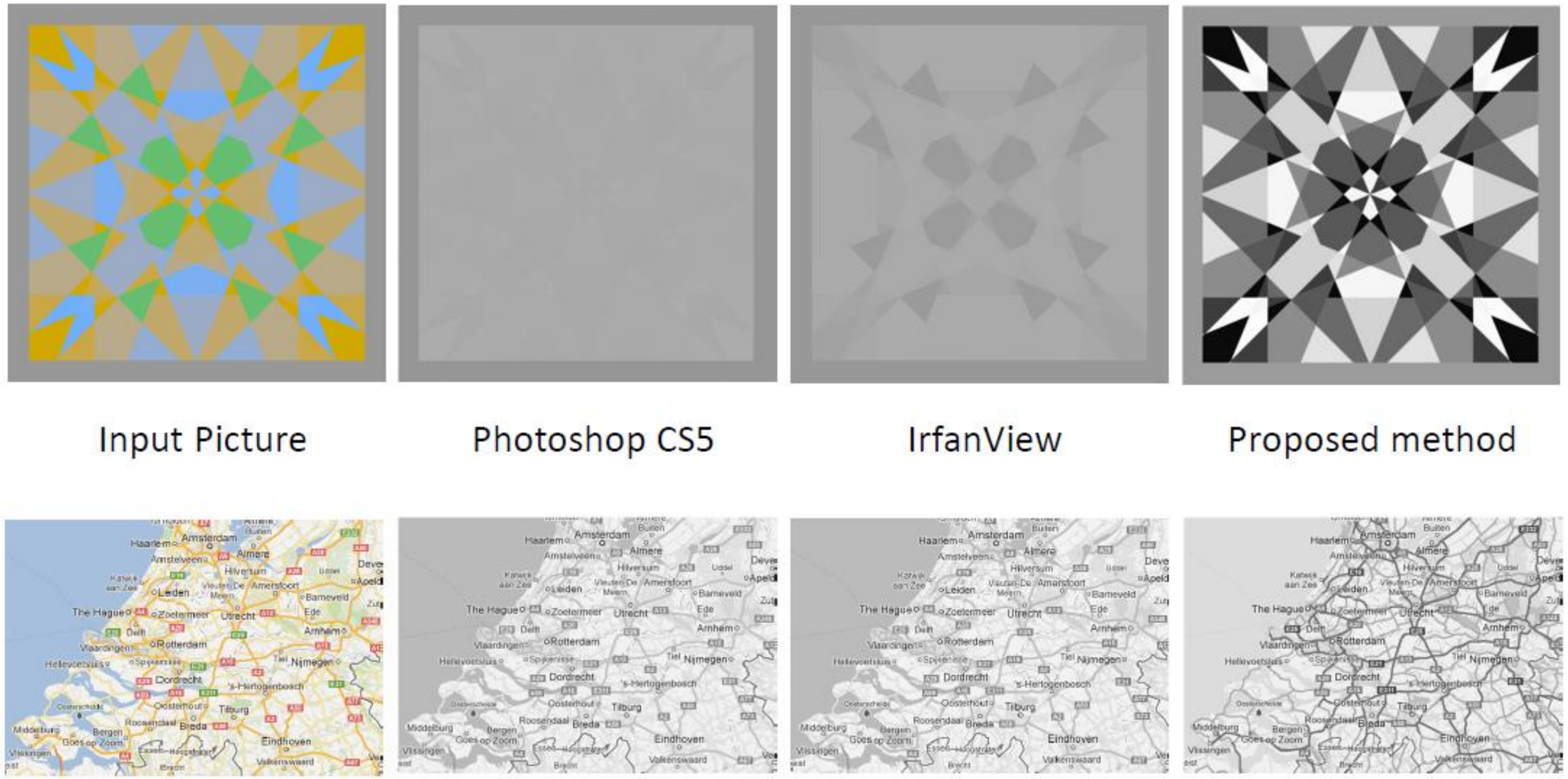
  - However, human visual experiments show that a *blue* light is *perceived* as much more dark than a *red* light, and green light is the brightest (even if they all have the same radiance)

  - Based on experimental data, a more *accurate* computation of luminance ***for phosphor RGB*** is

$$Y = 0.2125\,R + 0.7154\,G + 0.0721\,B$$

Some interesting topics

- Converting RGB to Gray with a minimum perceptual loss



From: "Contrast Preserving Decolorization with Perception-Based Quality Metrics", Int. Journal of Computer Vision, 2014

Some interesting topics

- Image colorization (gray2rgb)
  - Grayscale image + color scribble (given by user)-> color image

From https://sites.google.com/site/globalsmoothing/

Some interesting topics

- Image colorization

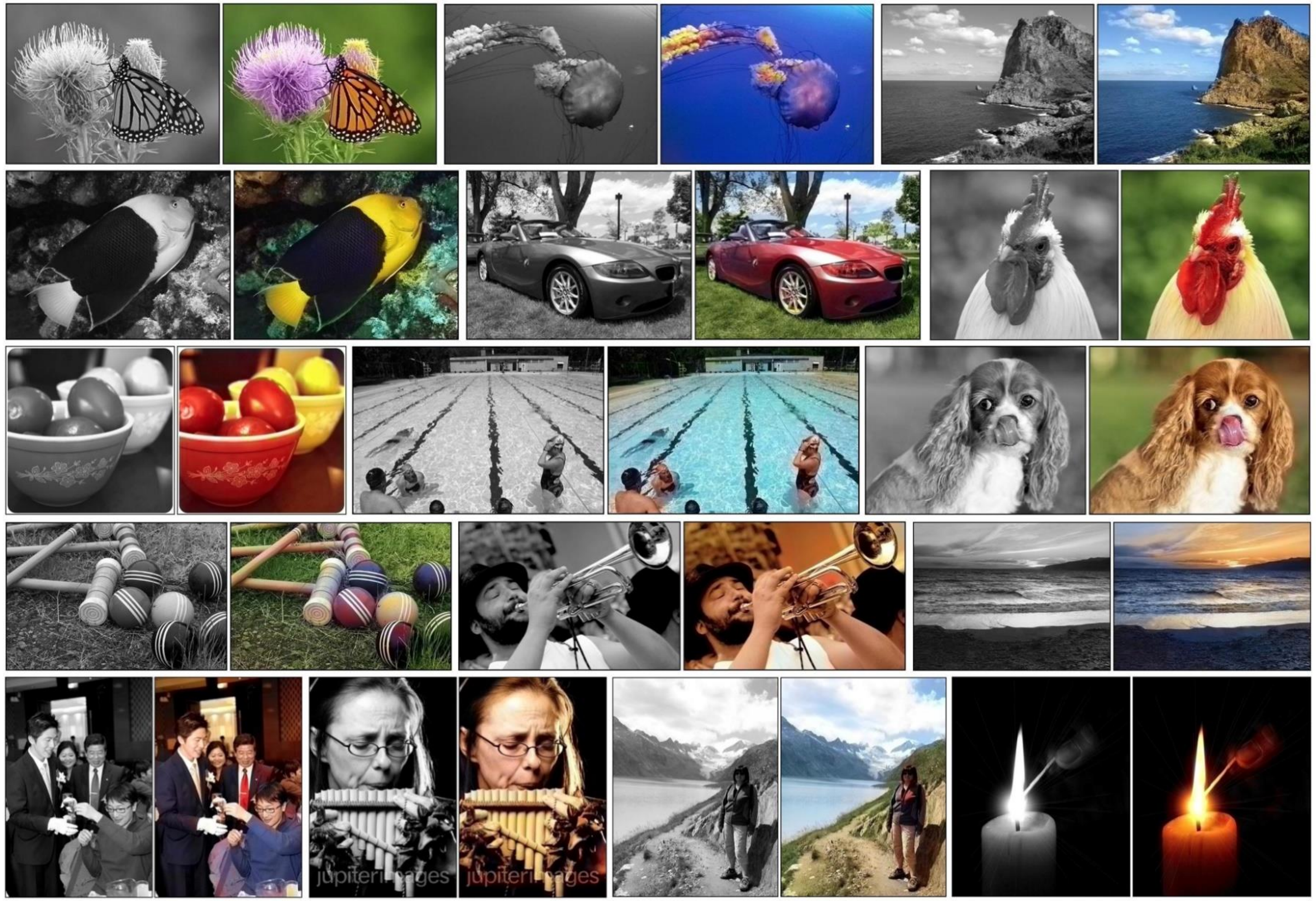


Image Files and Formats

- GIF
  - Colors are stored using a color map
  - Allows multiple images per file: animated GIFs

- PNG
  - Supports true color
  - Lossless compression

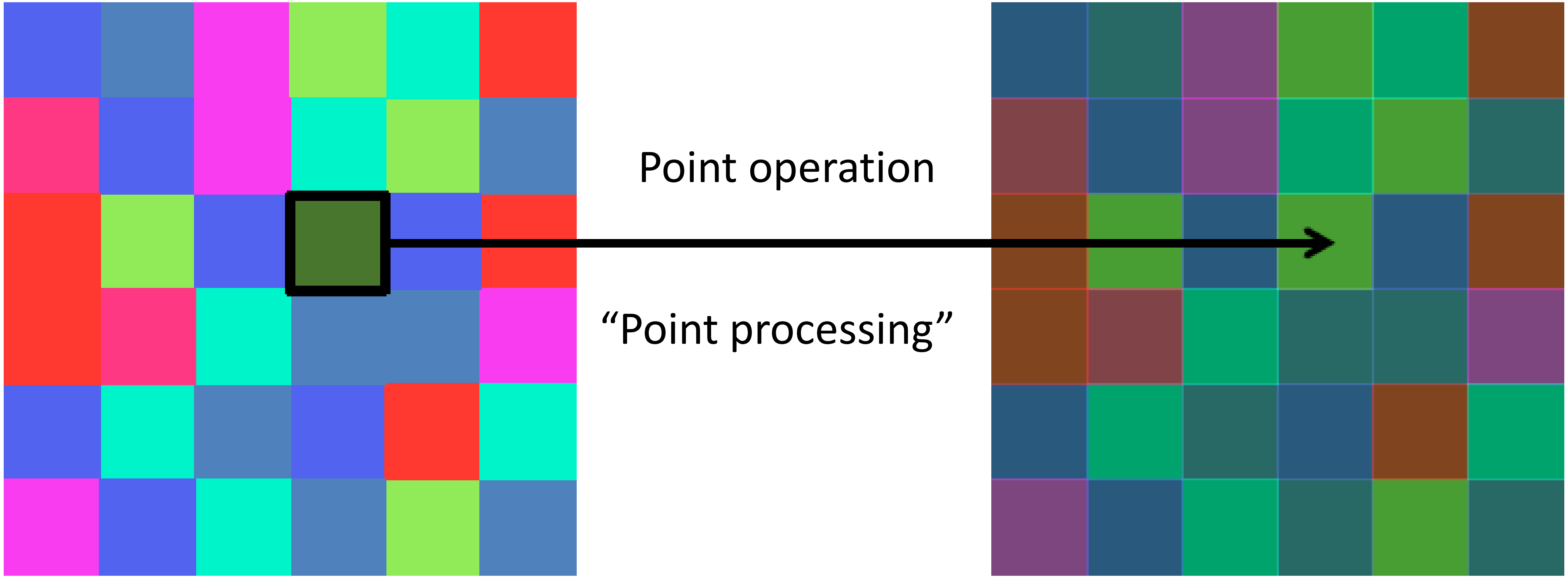
- JPEG
  - Lossy compression

Image Processing

Lecture 02

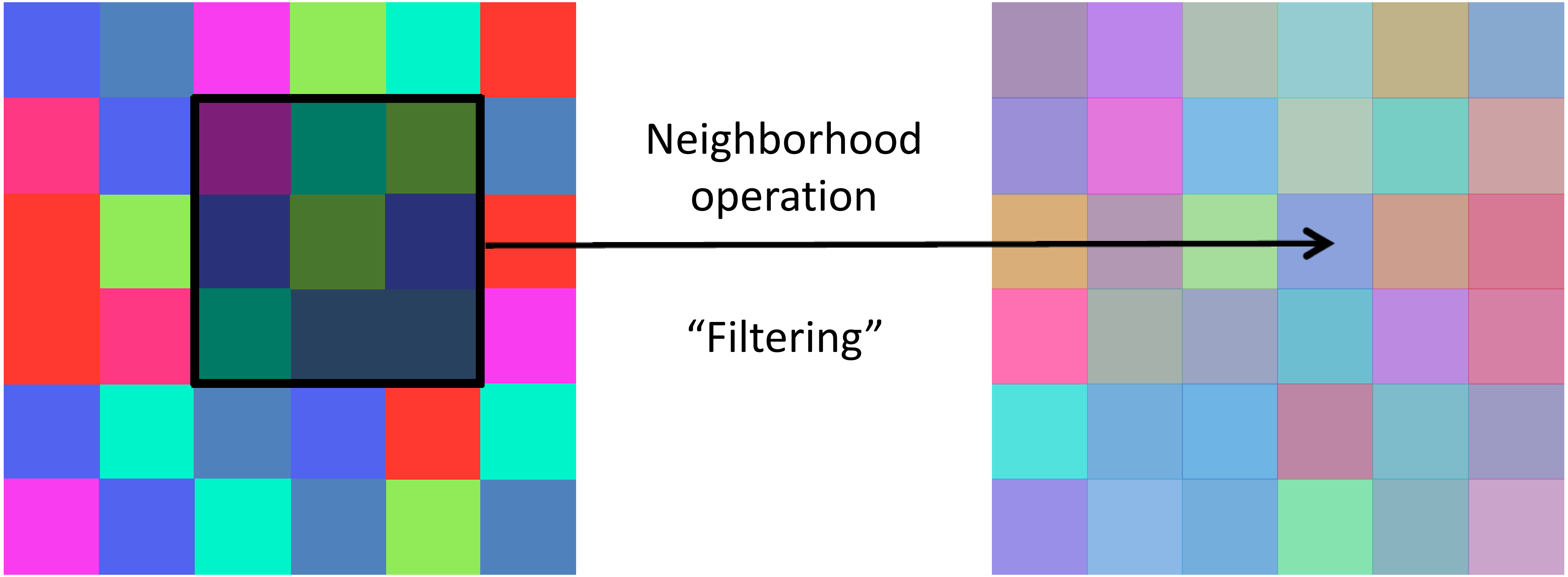
- Digital image
- **Point processing**

What types of image filtering can we do?



Each output pixel's value depends on only the corresponding input pixel value

What types of image filtering can we do?



Each output pixel's value depends on only the corresponding input pixel value and its neighborhood pixel values

Examples of point processing

original

darken

lower contrast

non-linear lower contrast

How would you implement these?

invert

lighten

raise contrast

non-linear raise contrast

Examples of point processing

original

$f(\boldsymbol{x})$

darken

$f(\boldsymbol{x}) - 128$

lower contrast

$f(\boldsymbol{x})/2$

non-linear lower contrast

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

invert

$255 - f(\boldsymbol{x})$

lighten

$f(\boldsymbol{x}) + 128$

raise contrast

$f(\boldsymbol{x}) \times 2$

non-linear raise contrast

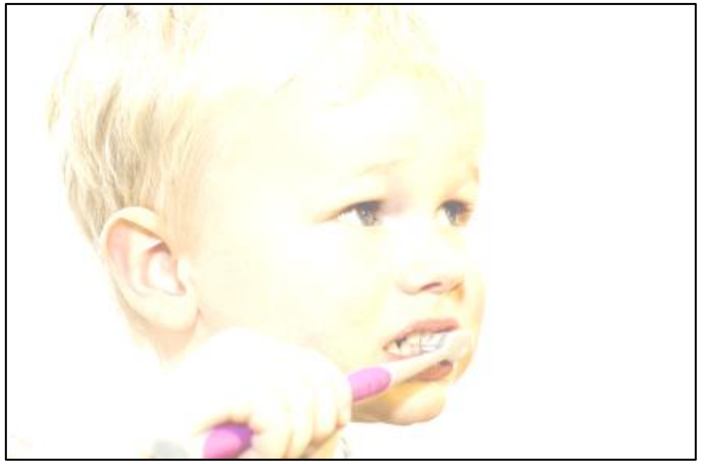
$$\left(\frac{f(\boldsymbol{x})}{255}\right)^{2} \times 255$$

Brightness vs. contrast

Brightness: The mean intensity of image

- Lighten: Increasing the brightness of image
- Darken: Decreasing the brightness of image

$f(x, y)$



$f(x, y) + 128$

$f(x, y) - 128$

Brightness vs. contrast

Contrast: The relative difference between pixel values

$f(x, y)$

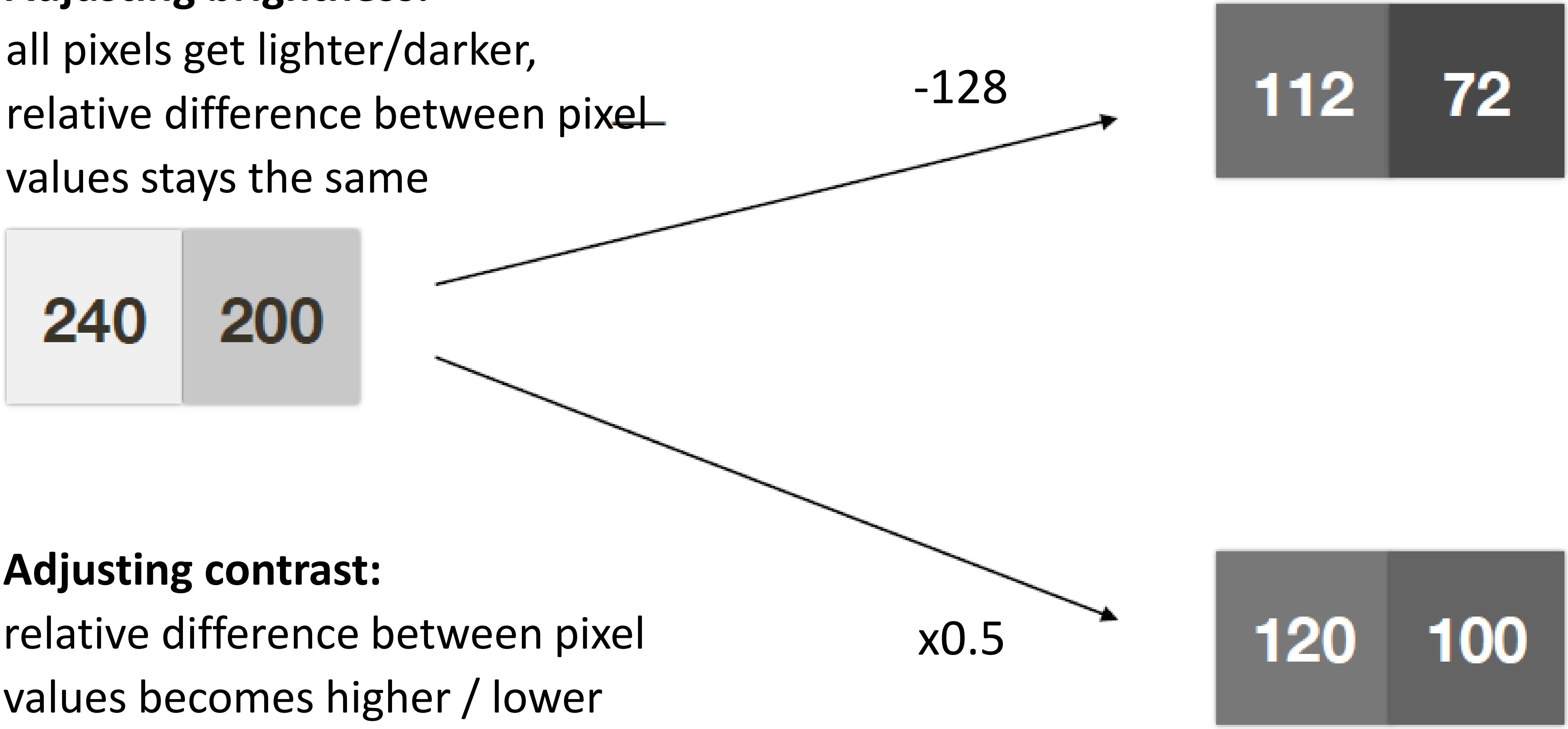
$f(x, y)/2$

$f(x, y) \times 2$

Brightness vs. contrast

Adjusting brightness:
all pixels get lighter/darker,
relative difference between pixel values stays the same

Adjusting contrast:
relative difference between pixel values becomes higher / lower



Intensity transformation

- In the point processing, the operator on spatial domain become

$$g(x,y) = h(f(x,y)) \longrightarrow \quad s = h(r)$$

  where s and r denote the intensity of g and f at any point (x,y).

- We call h an intensity transformation function, because it transform the input intensity r into the output intensity s.

a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast stretching function. (b) Thresholding function.

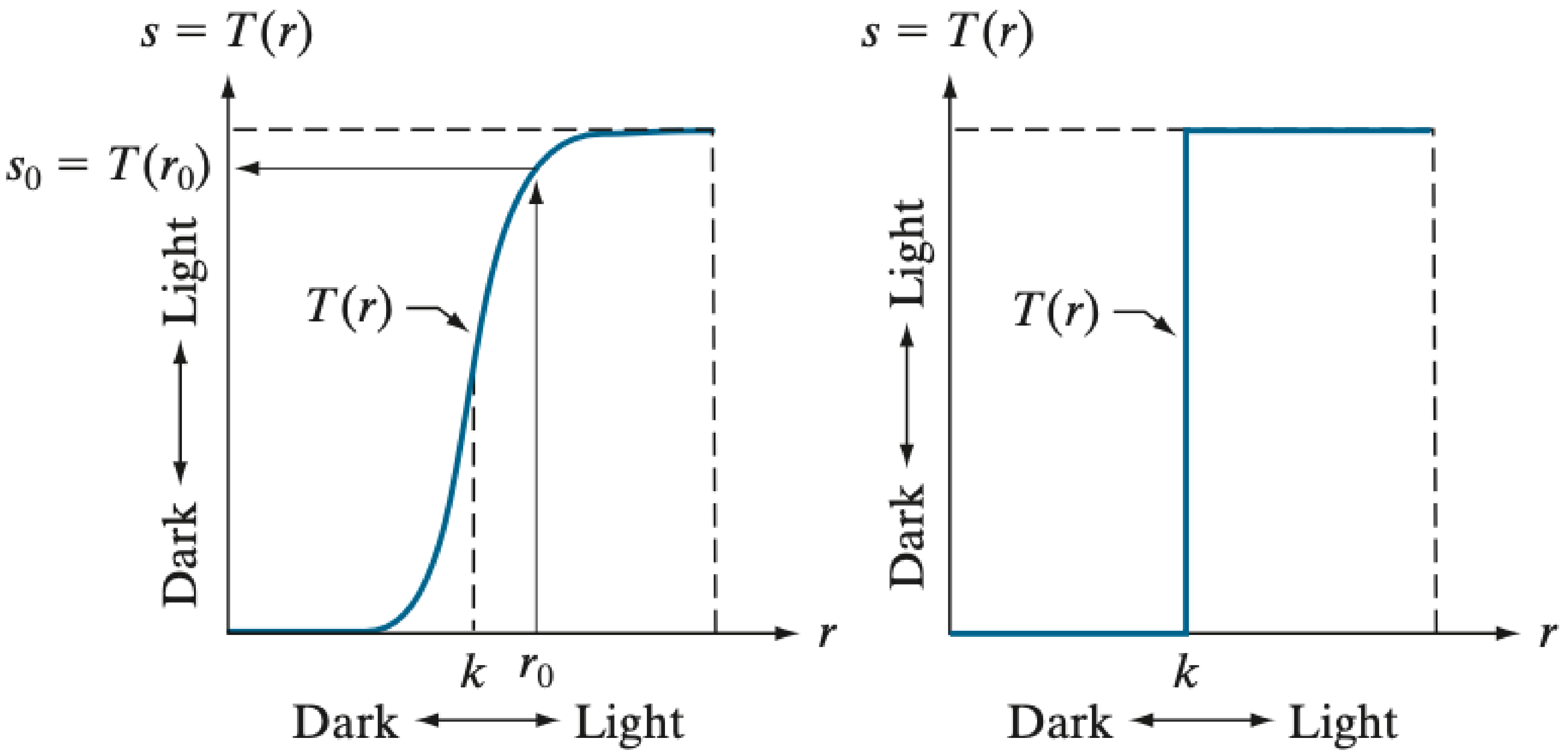
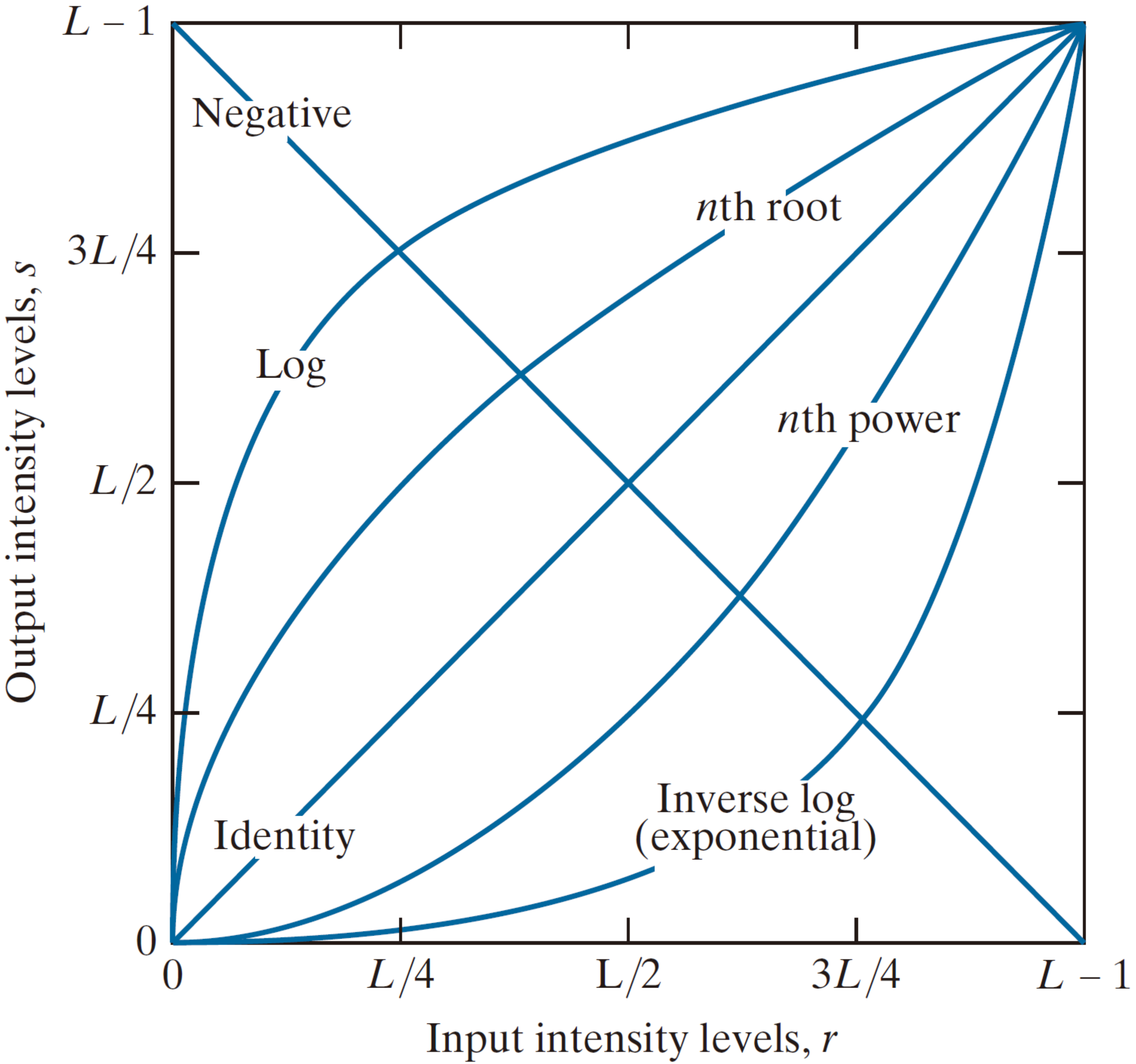


FIGURE 3.3 Some basic intensity transformation functions. Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.



Power-law (gamma) transformation

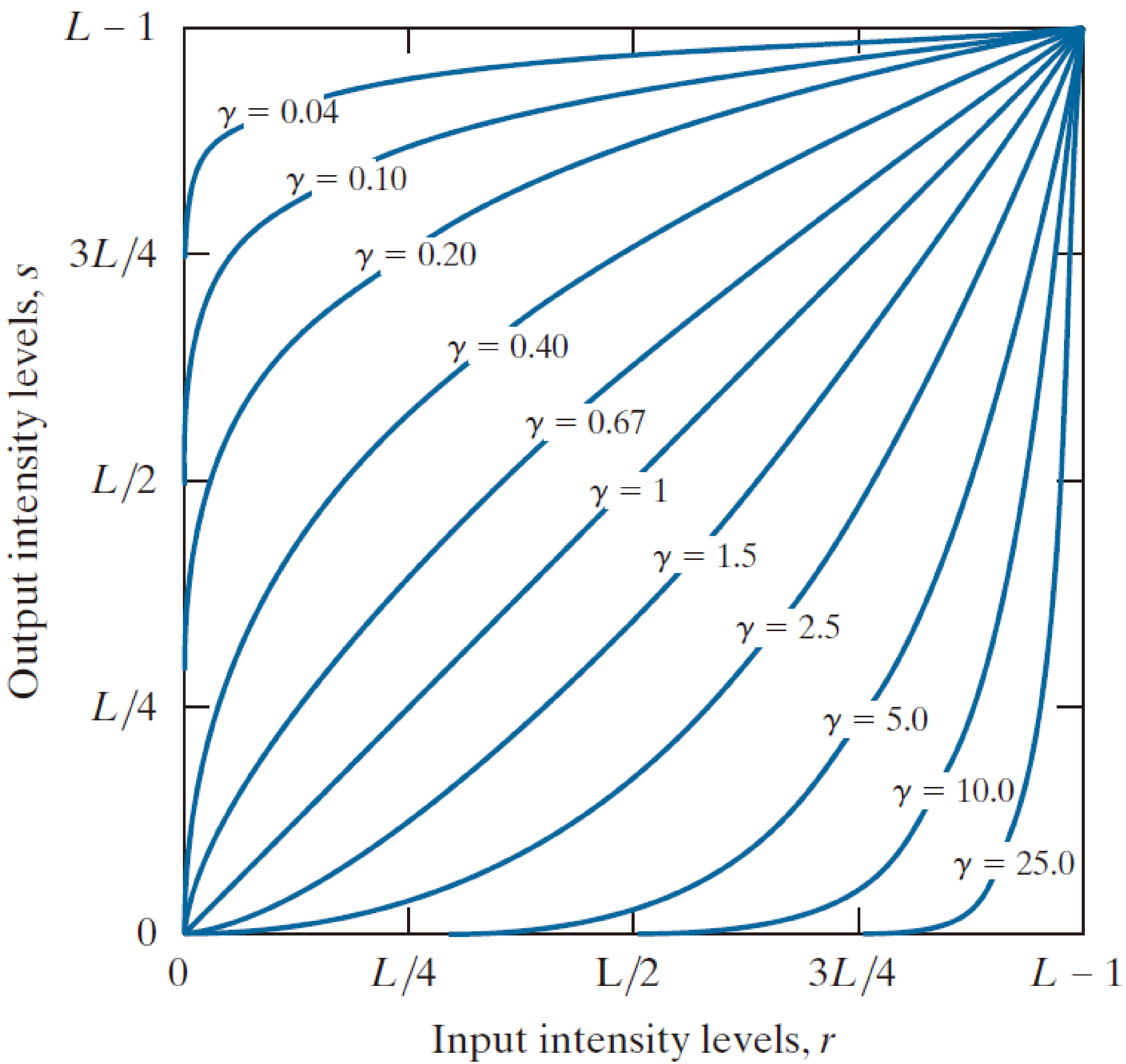
- Power-law (gamma) transformations have the form

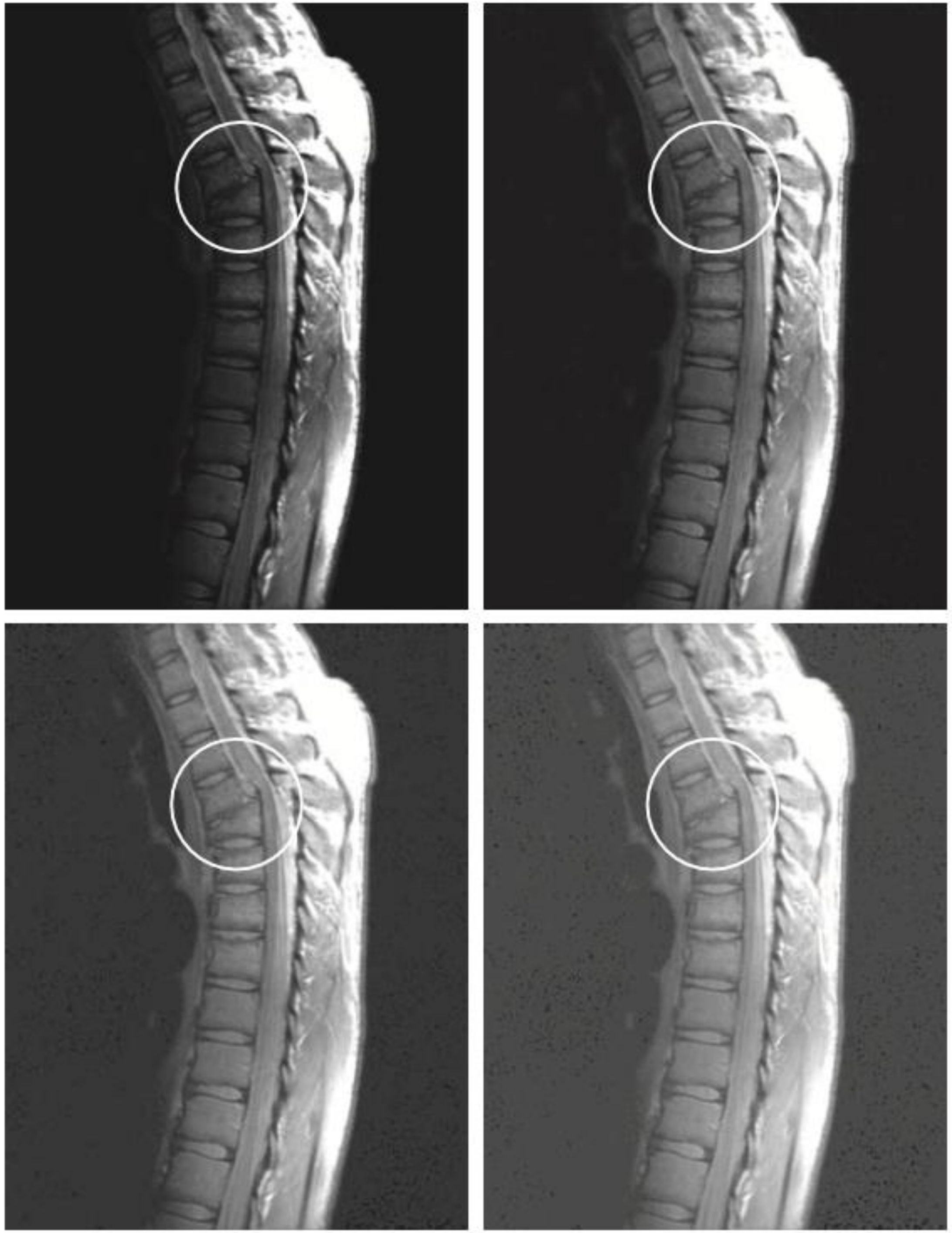
$$s = cr^{\gamma}$$

where c and γ are positive constants

- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels

FIGURE 3.6 Plots of the gamma equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.





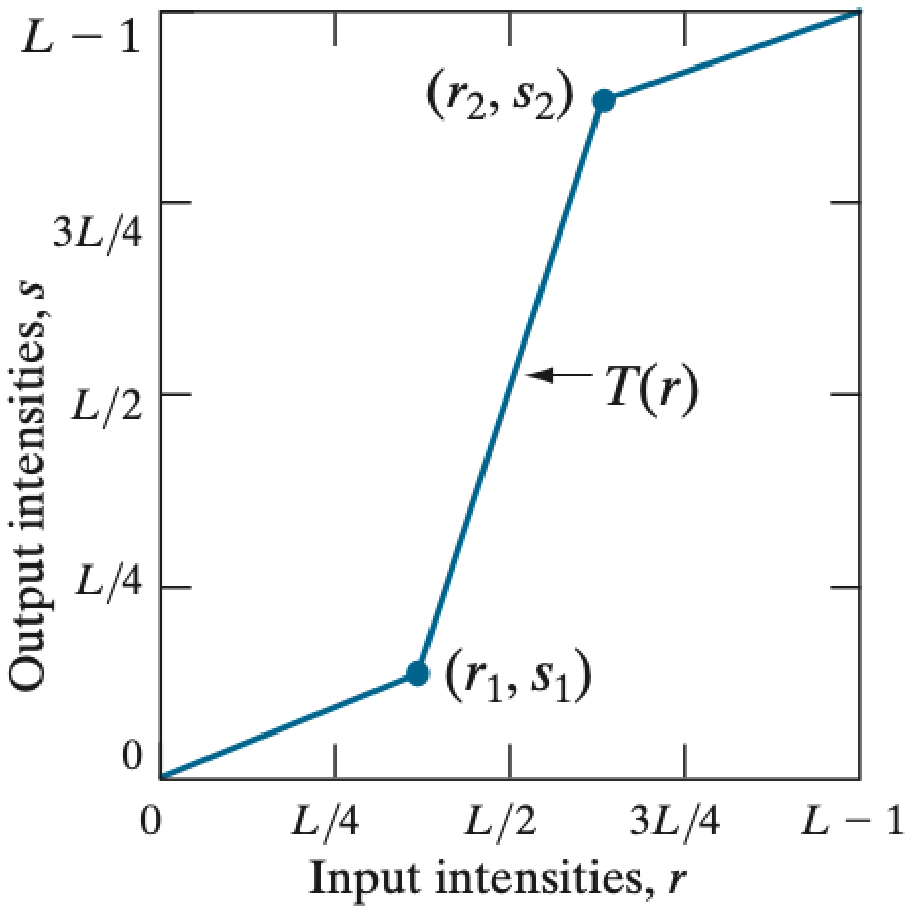
a b
c d

FIGURE 3.8
(a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the transformation in Eq. (3-5) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

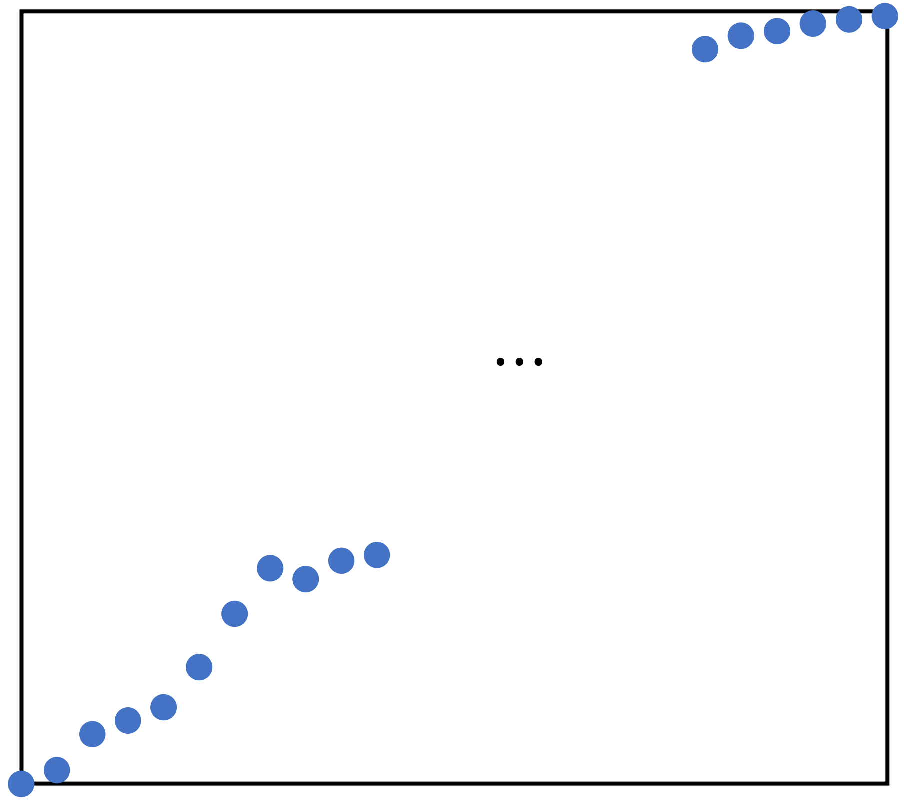
a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3-5) with $\gamma = 3.0, 4.0$, and 5.0, respectively. ($c = 1$ in all cases.) (Original image courtesy of NASA.)

More complex transformation functions



Piecewise linear transformation



More complex piecewise linear transformation

Before

After

Before

After

Thank You