HuStar AI Course: Computer Vision

Digital Image Processing

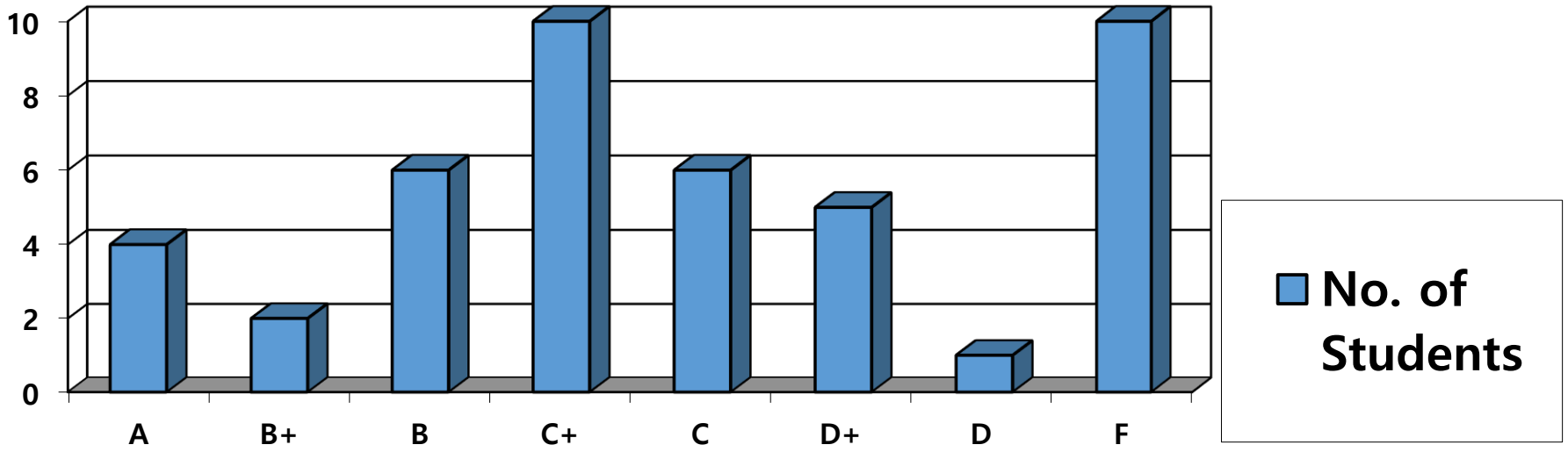
Janghun Jo Geonung Kim

Computer Graphics Lab.

POSTECH

Histogram

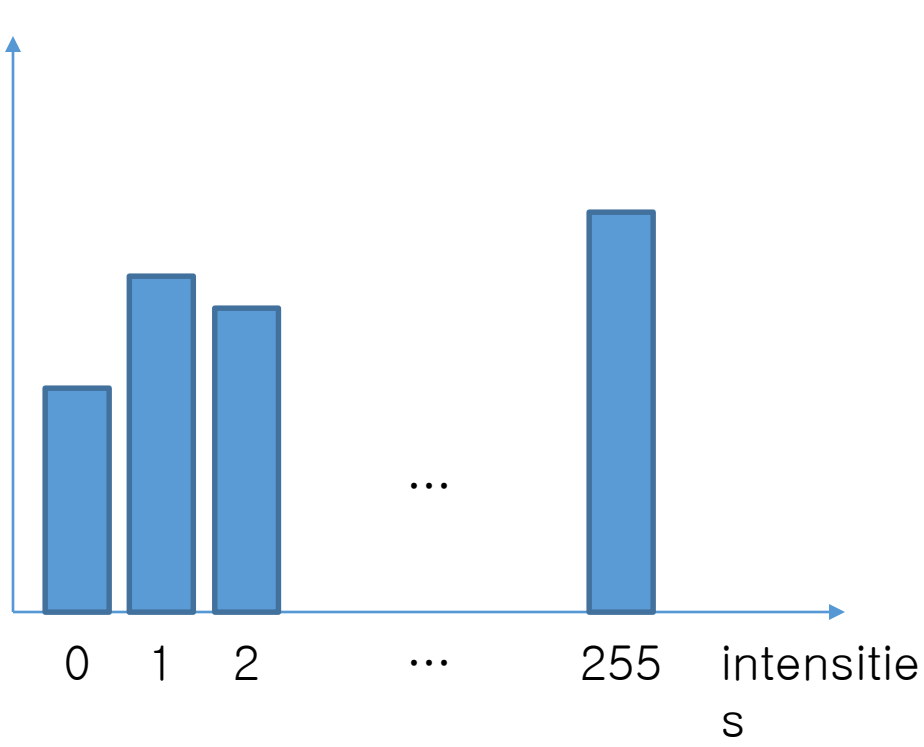
Histogram = Graph of population frequencies



Grades of the course 178 xxx

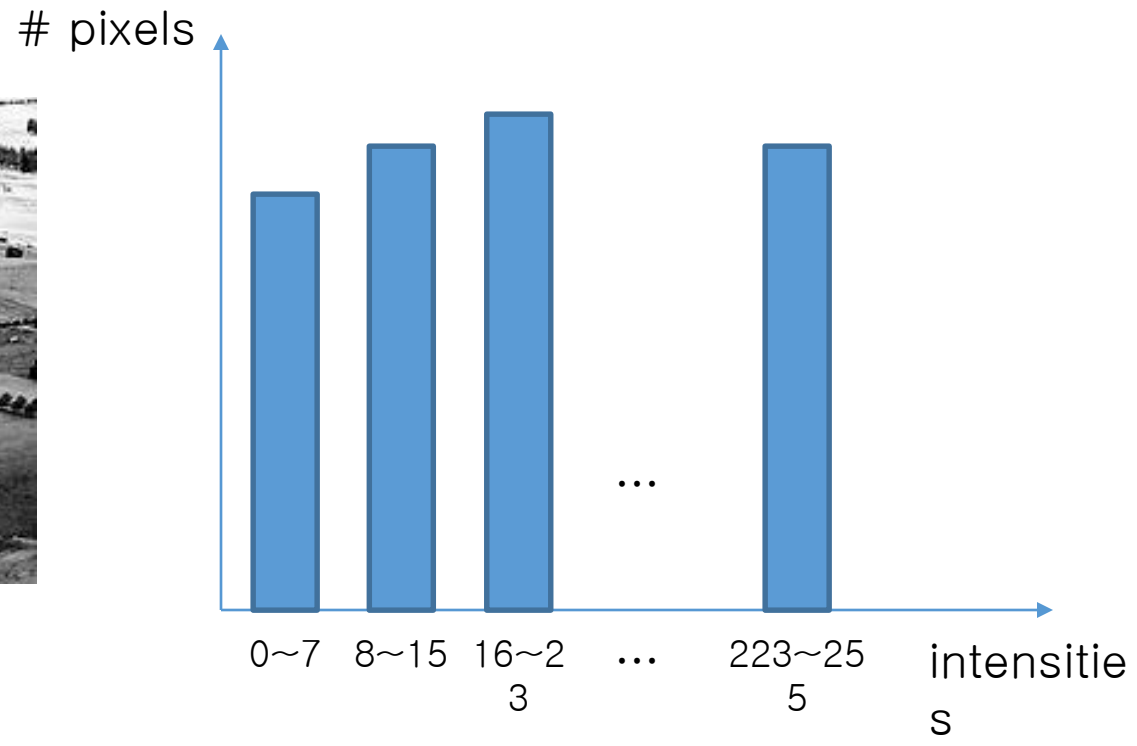
Histogram of an Image

- Assume an image whose pixel values are integers from 0 to 255
- Then, for each $x \in \{0, \ldots, 255\}$, we can count the number of pixels whose values are x.
- We can obtain a histogram of 256 bins



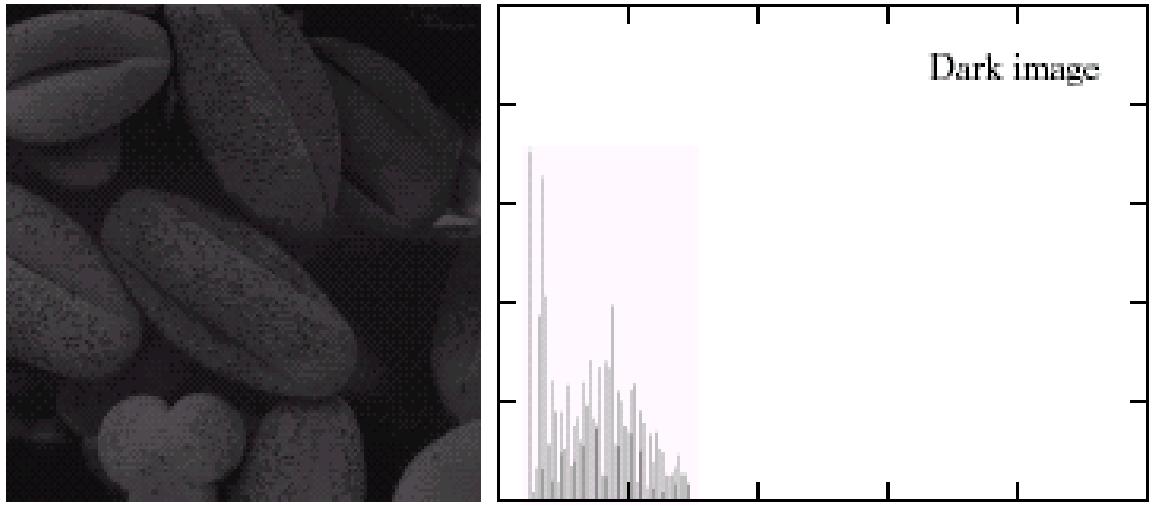
Histogram of an Image

- We can subdivide the intensities into 0~7, 8~15, … 223~255
- Then we have 32 bins

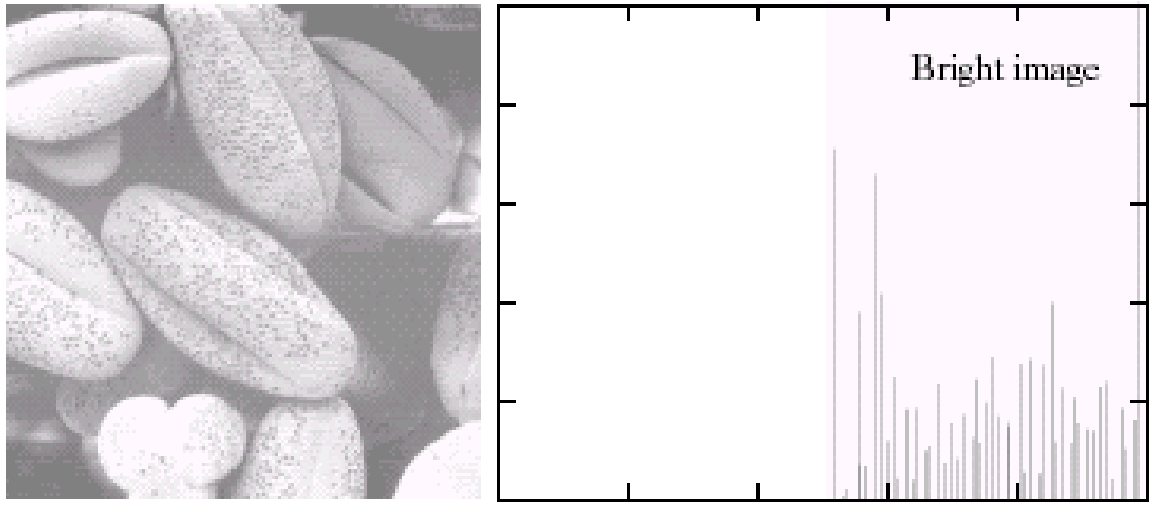


Histogram of an Image

- Useful for analyzing and manipulating brightness and contrast of an image



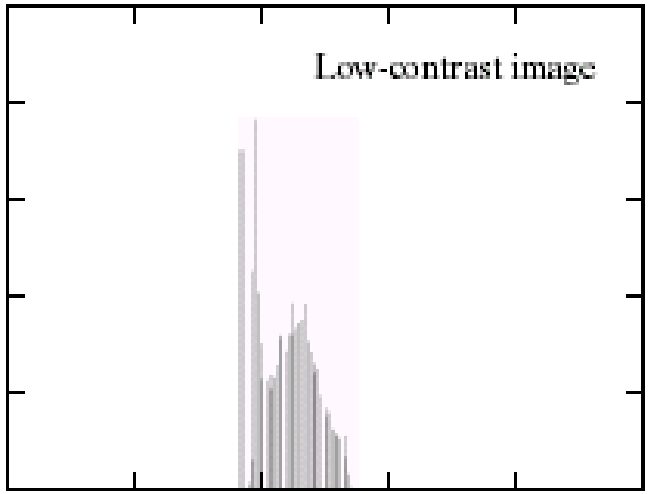
Dark image has histogram on the left



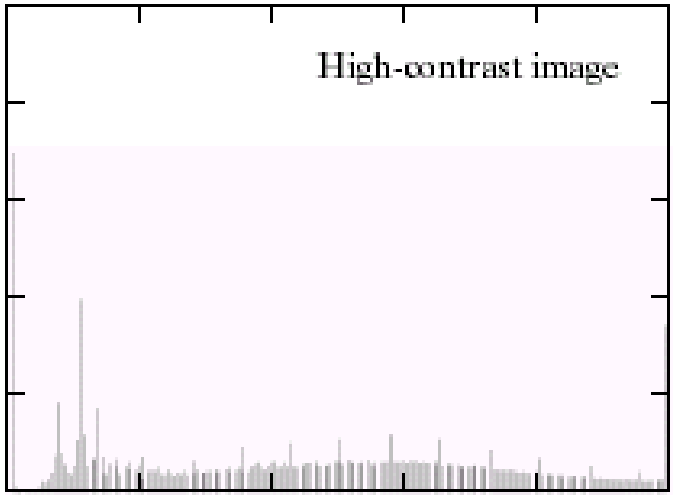
Bright image has histogram on the right

Histogram of an Image

- Useful for analyzing and manipulating brightness and contrast of an image



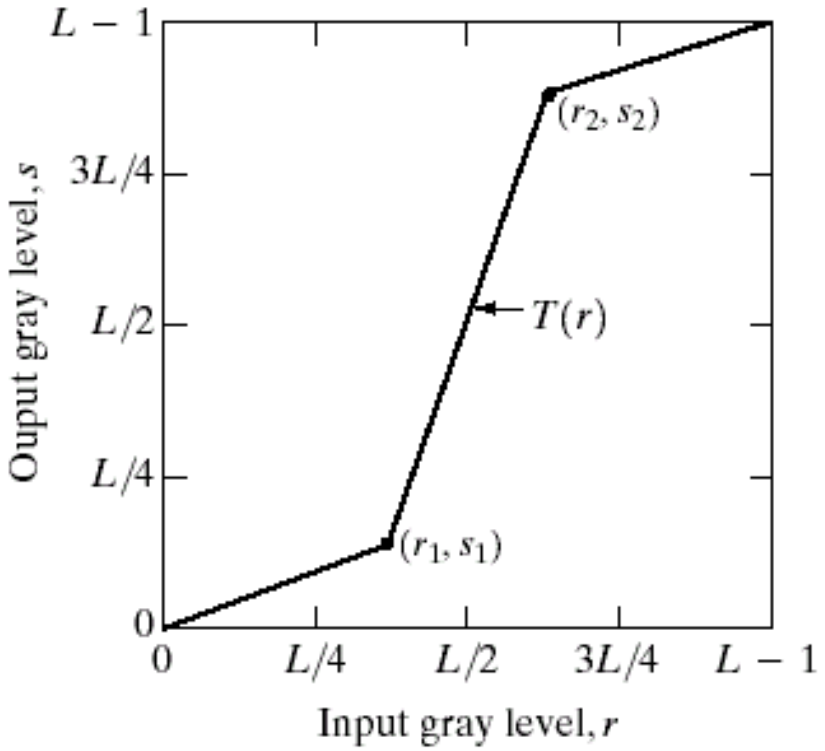
Low contrast image
has narrow histogram



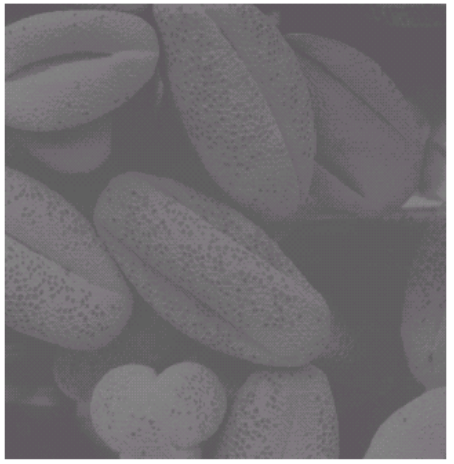
High contrast image
has wide histogram

Contrast stretching

Contrast means the difference between the brightest and darkest intensities

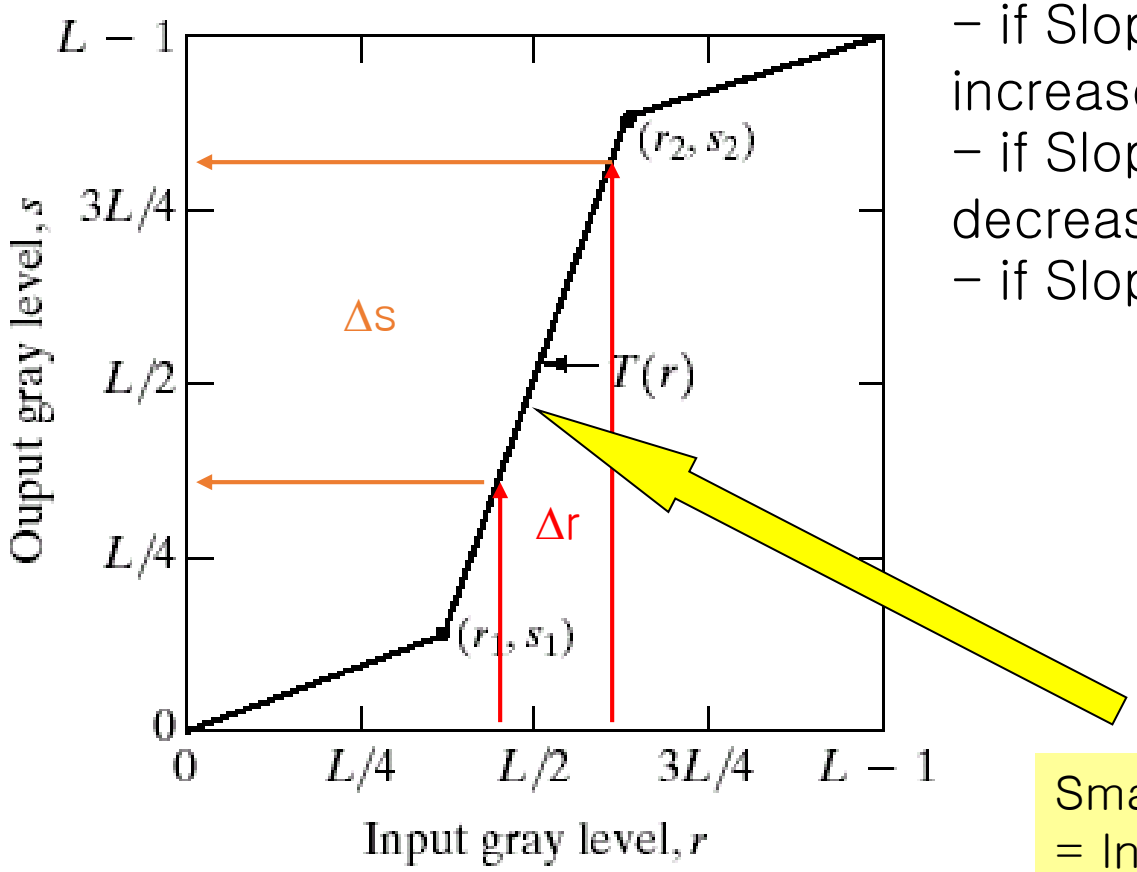


Before contrast enhancement



After

Contrast stretching



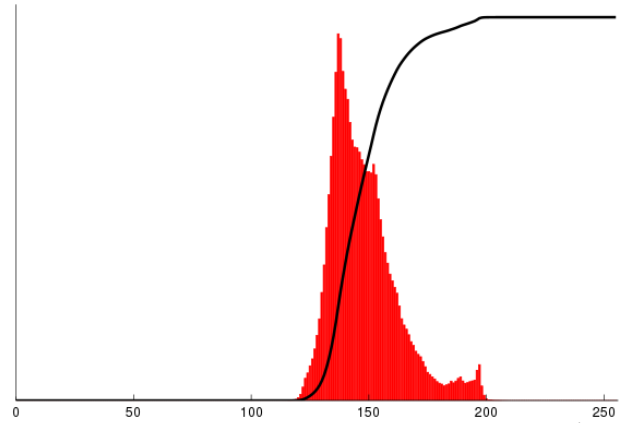
Notice the slope of T(r)
- if Slope > 1 → Contrast increases
- if Slope < 1 → Contrast decrease
- if Slope = 1 → no change

Smaller Δr yields wider Δs = Increasing Contrast

Histogram equalization

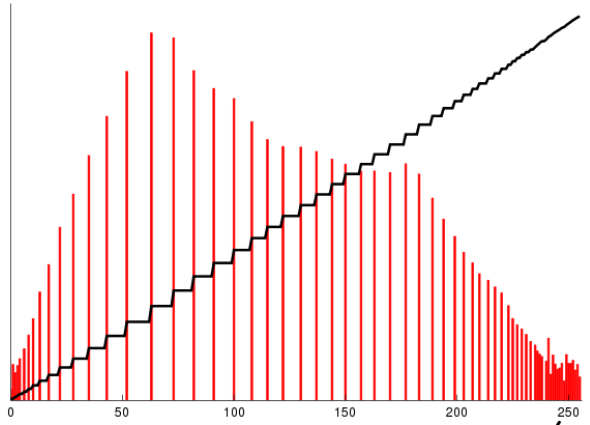
- Adjust image's brightness and contrast

Before histogram equalization



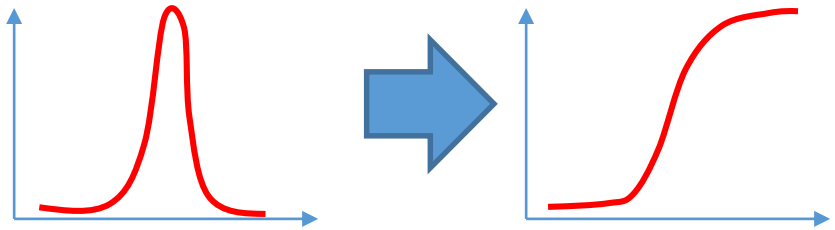
Corresponding histogram (red) and cumulative histogram (black)

After histogram equalization



Corresponding histogram (red) and cumulative histogram (black)

Histogram equalization

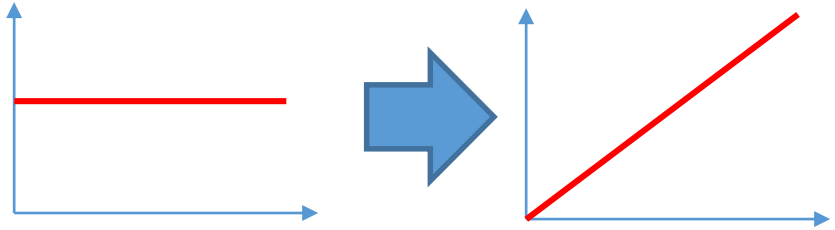


Histogram $y = h_l(x)$ Cumulative histogram

$$y = H_l(x) = \sum_{w=0}^{x} h_l(w)$$

We want $x' = T(x)$ s.t. $H_l\left(T^{-1}(x')\right) = \alpha x'$
i.e., the transformed intensities x' has a cumulative histogram shown below.

$$\Rightarrow T(x) = \frac{1}{\alpha} H_l(x)$$



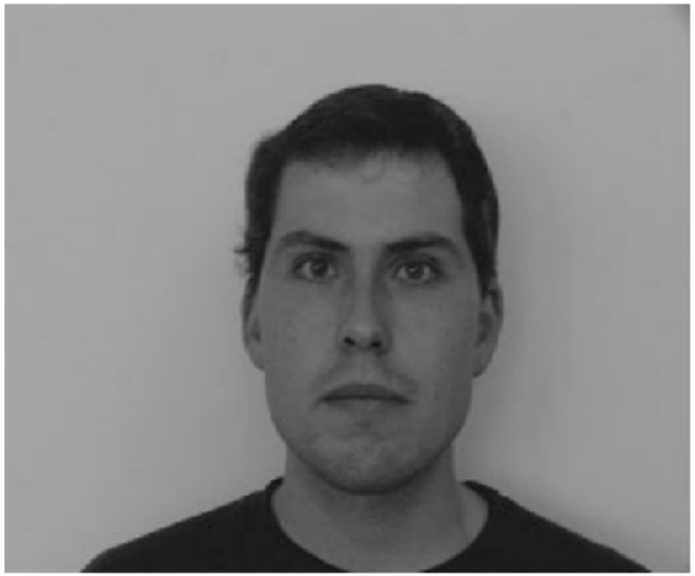
Histogram $y = h_h(x)$ Cumulative histogram

$$y = H_h(x) = \sum_{w=0}^{x} h_h(w) = \alpha x$$

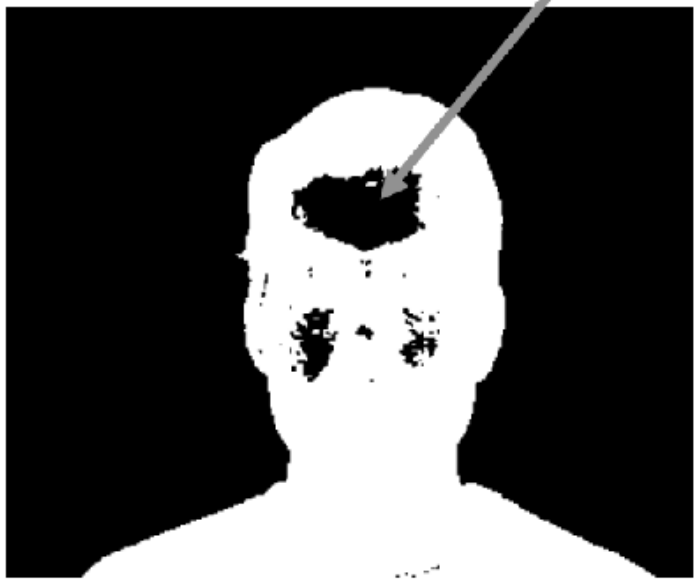
$$\alpha = \frac{\#\ \text{pixels}}{255}$$

Thresholding

- Simplest case of binary segmentation
- Assign white / black to each pixel according to its intensity



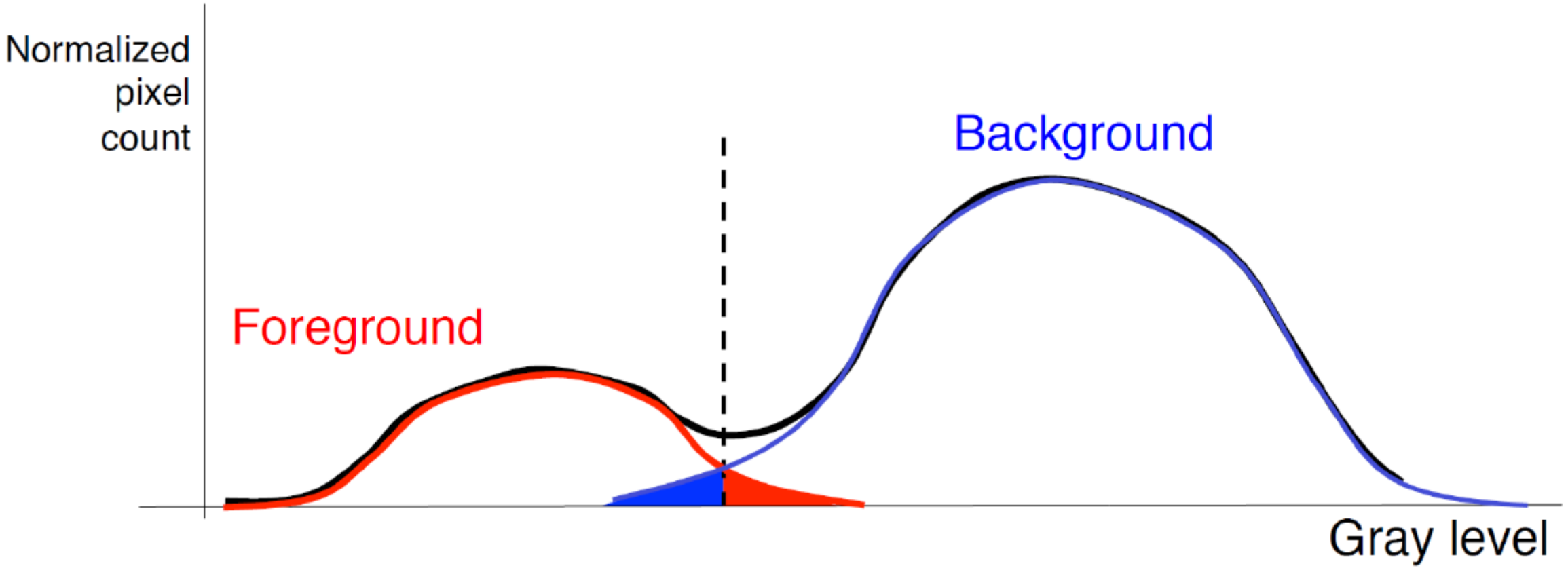
Original image
Peter $f[x,y]$



Thresholded
Peter $m[x,y]$

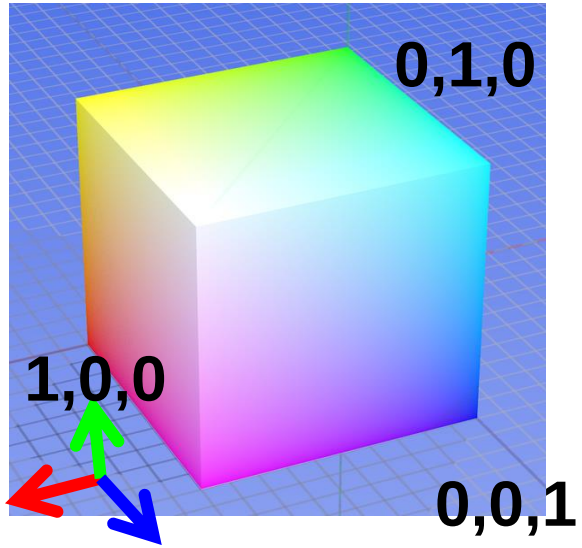
$f[x,y] \cdot m[x,y]$

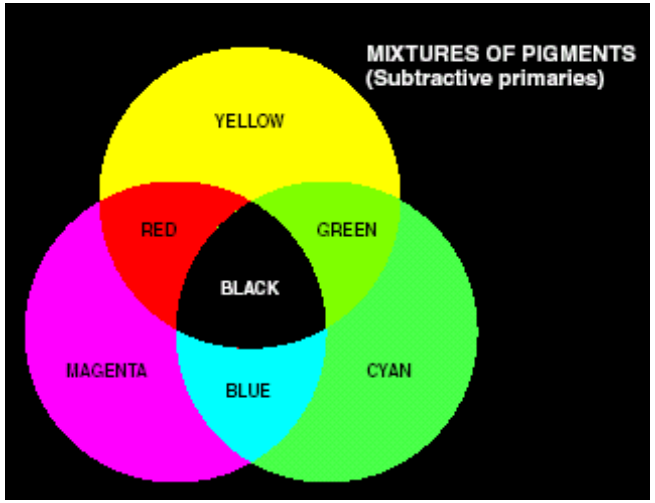
Thresholding

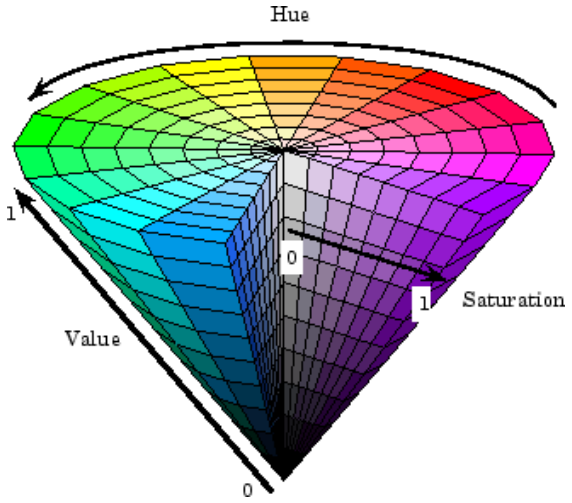


Color model

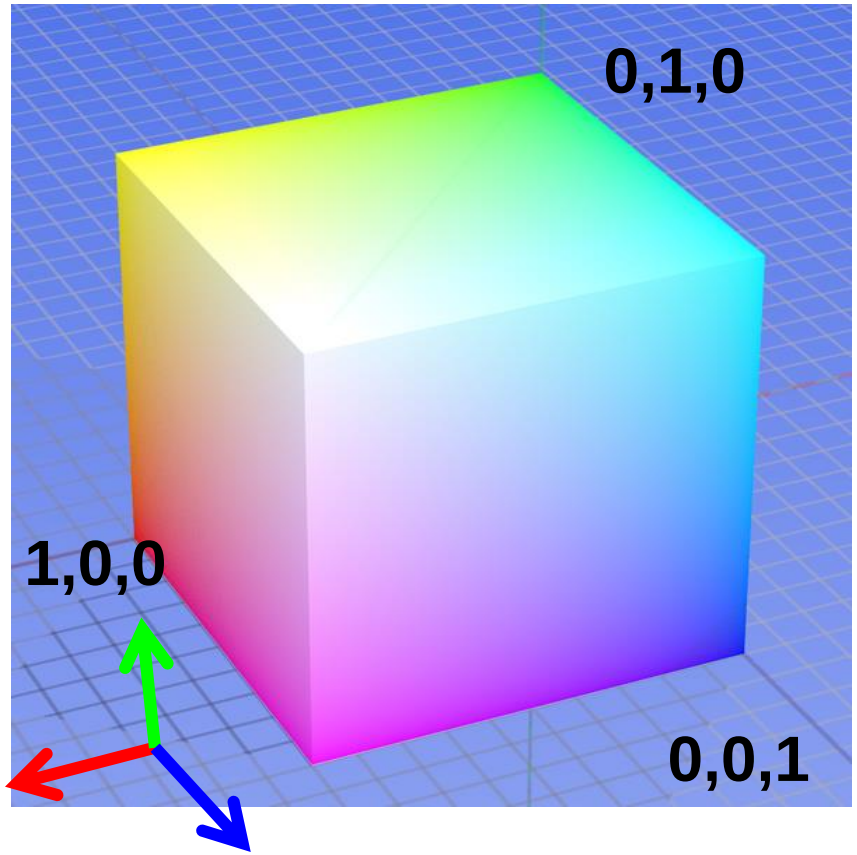
- Widely used color models
  - RGB (Red, Green, Blue) for displays and cameras
  - CMY, CMYK (Cyan, Magenta, Yellow, Black) for printing
  - HSI (Hue, Saturation, Intensity), HSV (Hue, Saturation, Value)







RGB color model



R
(G=0,B=0)

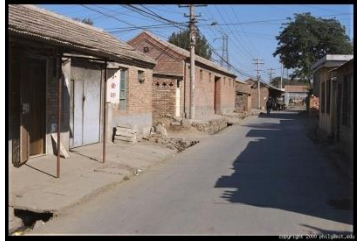
G
(R=0,B=0)

B
(R=0,G=0)

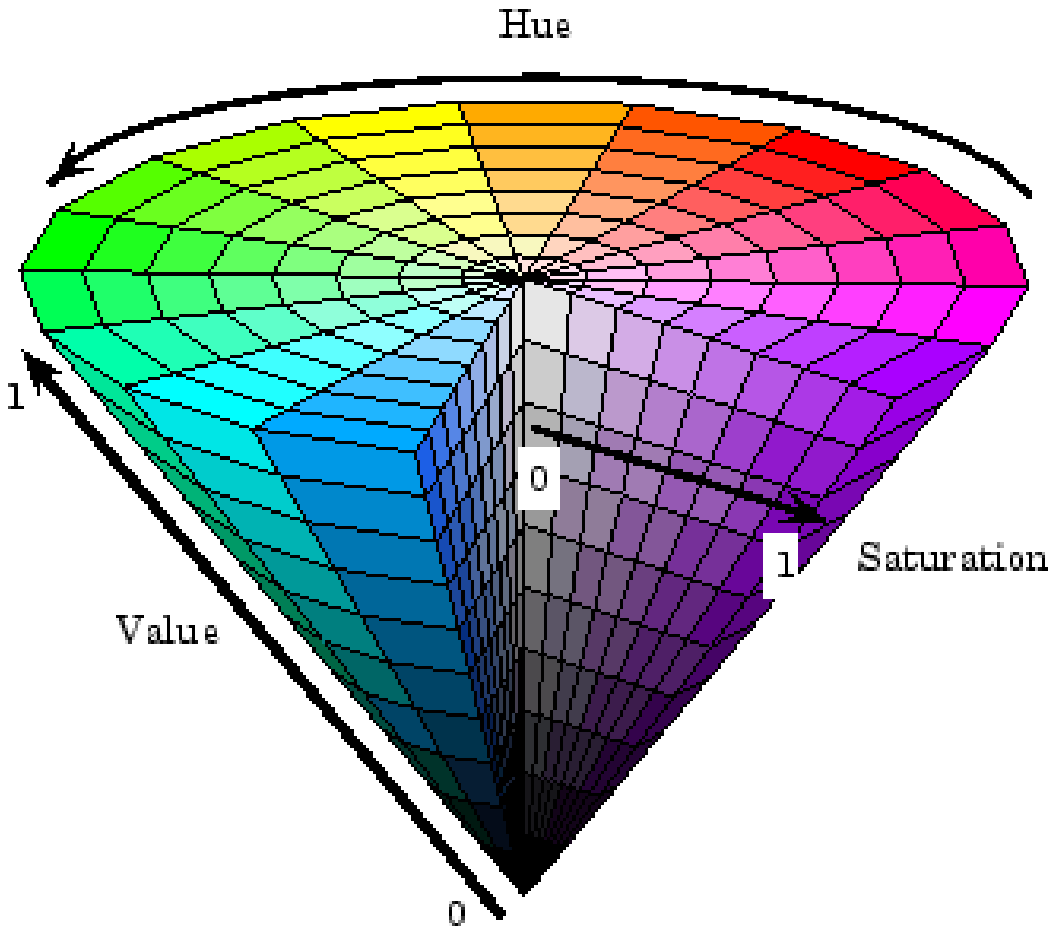
Some drawbacks

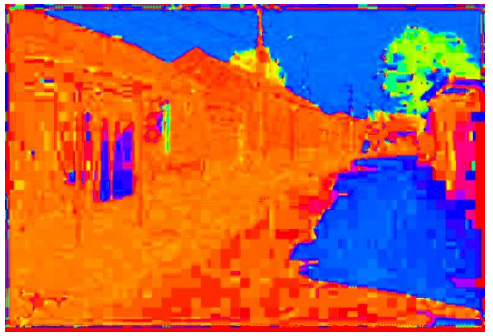
- Strongly correlated channels
- Non-perceptual ➔ Not easy to guess the coordinates of a specific color e.g., purple

HSV color model



Intuitive color space





H
(S=1,V=1)

S
(H=1,V=1)

V
(H=1,S=0)

Splash of color