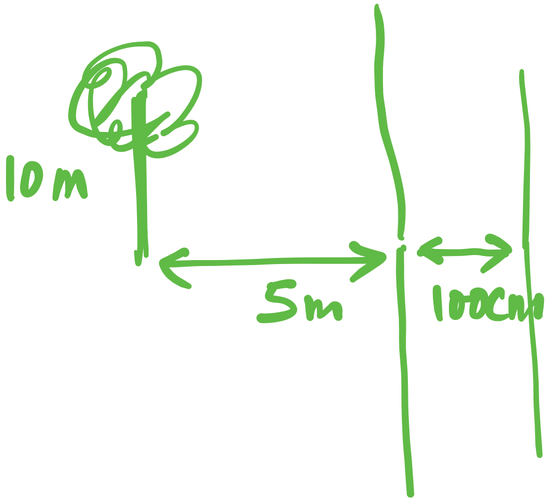


$$y' = f\frac{y}{5} = \frac{(1)(10)}{5} = 2$$

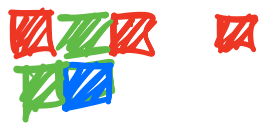
Point Processes

Computational Photography (CSCI 3240U)

Faisal Z. Qureshi
http://vclab.science.ontariotechu.ca
Ontario Tech University

Imaging pipeline
Image formation

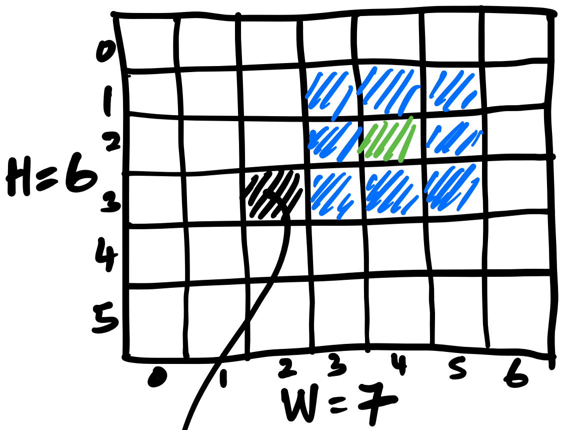
Raw image



↓

Color or grayscale image

ANATOMY OF AN IMAGE



$= I \in \mathbb{R}^{6\times 7}$

"uint8"

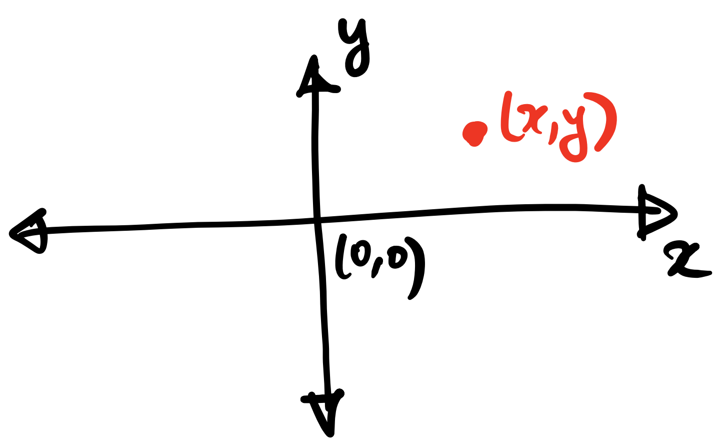
$I[3,2]$ ⇝ # of allowed values 8-bits, total $2^8 = 256$

$I[3,2] \in [0, 255]$

CONVERT TO FLOATING PTS.

$I[i,j] / 255.0$] $I[3,2] \in \mathbb{R}$

$\in [0,1]$



$I/255.0 \rightarrow [0,1]$

$\uparrow$ $\downarrow$

$I \leftarrow *255 \leftarrow$

Acknowledgments

- These slides borrow and adapt materials developed by others, including
  - Michael Brown
  - Kyros Kutulakos
  - David Lindell
  - Gordon Wetzstein
  - Marc Levoy
  - Fredo Durand
  - Paul Debevec
  - Ramesh Raskar

Slide credits

- A lot of inspiration and quite a few examples for these slides were taken directly from:
  - Kayvon Fatahalian (15-769, Fall 2016).
  - Michael Brown (CVPR 2016 Tutorial on understanding the image processing pipeline).
  - Marc Levoy (Stanford CS 178, Spring 2014).

Special thanks to Ioannis Gkioulekas

- Many of the slides are taken with his permission from the computational photography course that he has developed at CMU

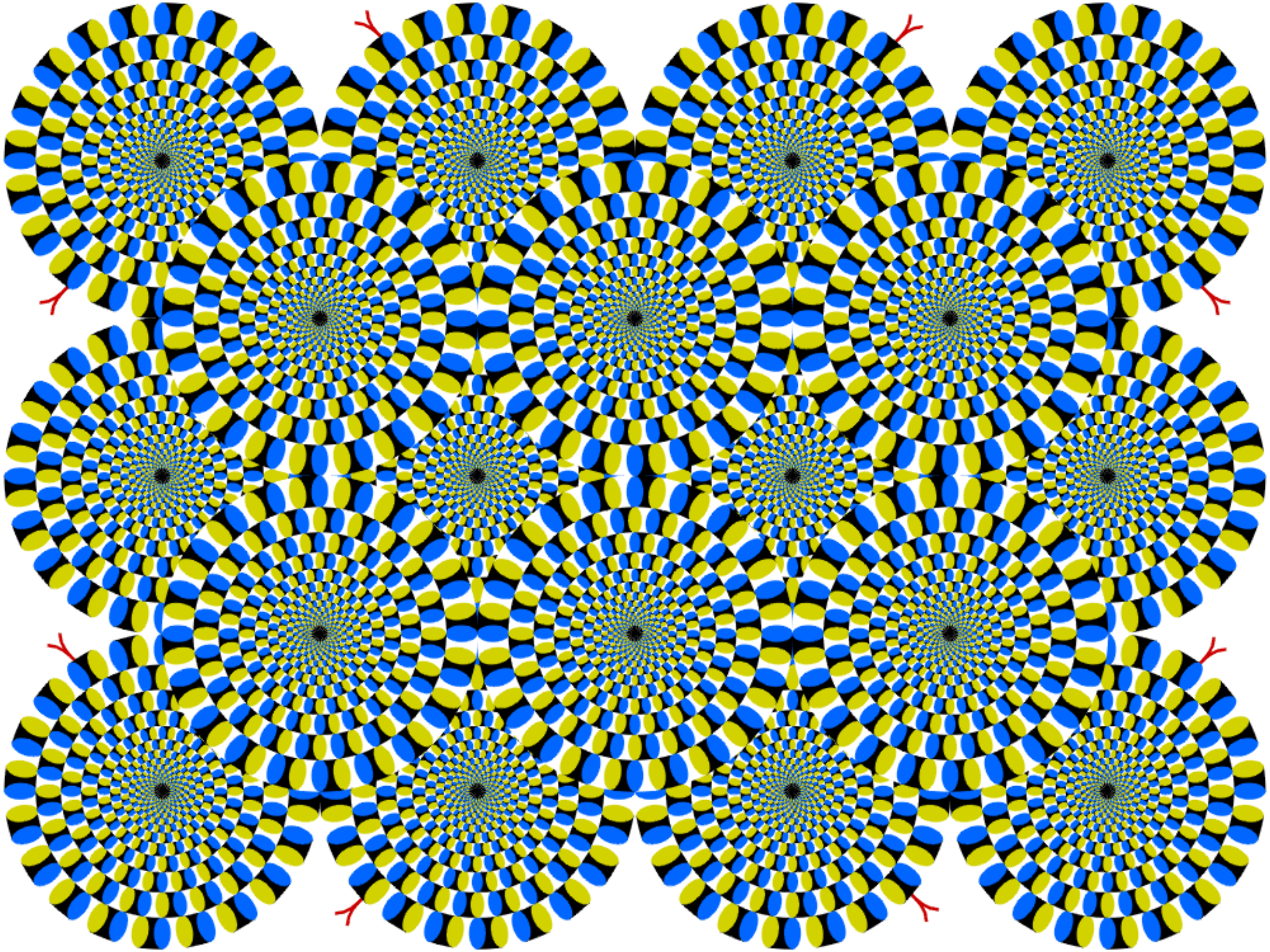


Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing
  - Spatial processing
  - Frequency domain processing

↑

E.g., Human perception

Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing
  - Spatial processing (pixel neighbourhoods)
  - Frequency domain processing

E.g., Human perception

Image Enhancement

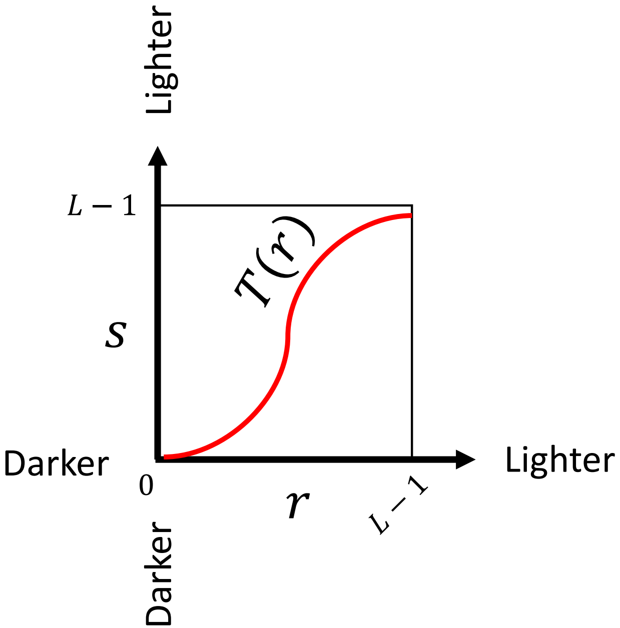
- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing ← Today's Focus
  - Spatial processing (pixel neighbourhoods)
  - Frequency domain processing

Image Enhancement

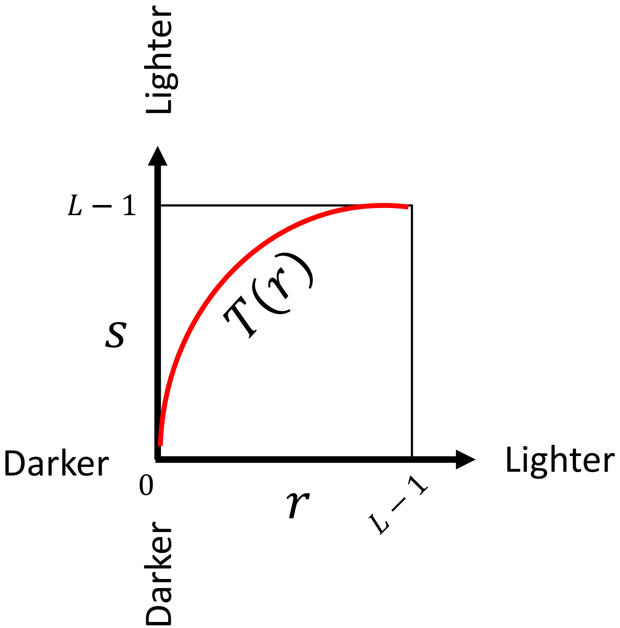
$$g(x,y) = T\left(f(x,y)\right)$$

- Input image: $f(x, y)$
- Output image: $g(x, y)$
- T is an operator on f or a set of f
  - T is defined over some neighbourhood of (x, y)
  - T can operate over a set of images
  - For point processing the neighbourhood of (x, y) is just (x, y) itself

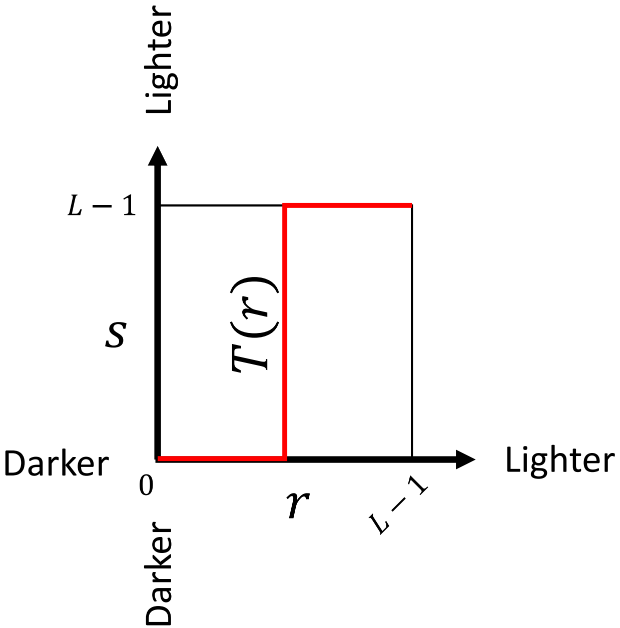
Point Processing Example



Point Processing Example



Point Processing Example



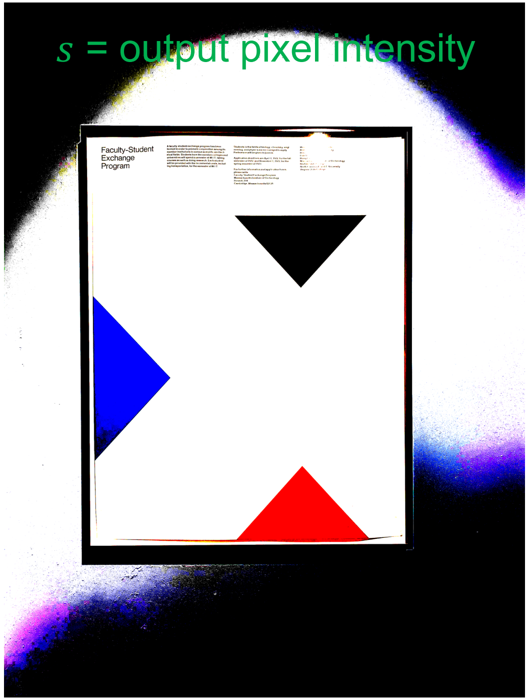
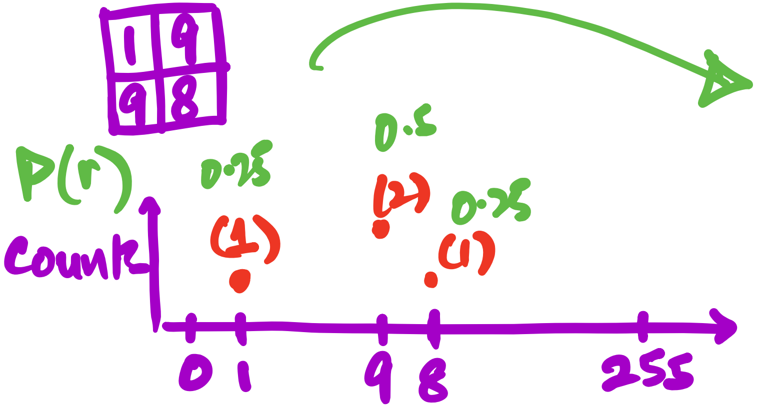


Image Histogram

- Consider an $H \times W$ image with L gray levels.
- Record the count n_k of pixels at each gray level r_k where $k \in [0, L-1]$
- The probability of a pixel at gray level r_k is



$$p(r_k) = \frac{n_k}{H \cdot W}$$

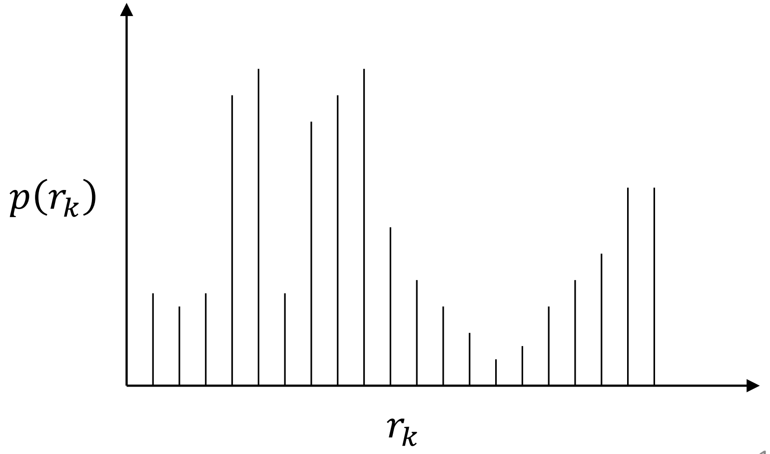
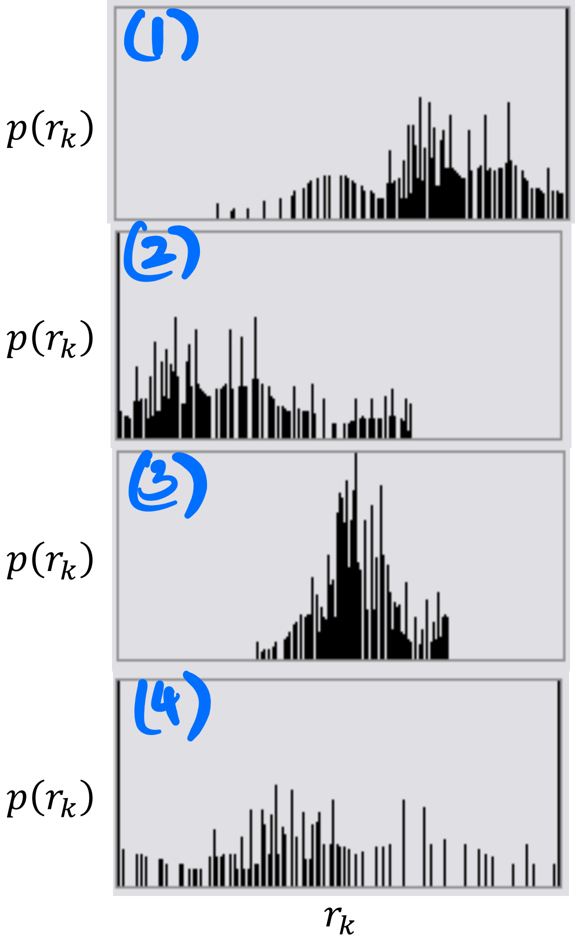


Image Histogram



A

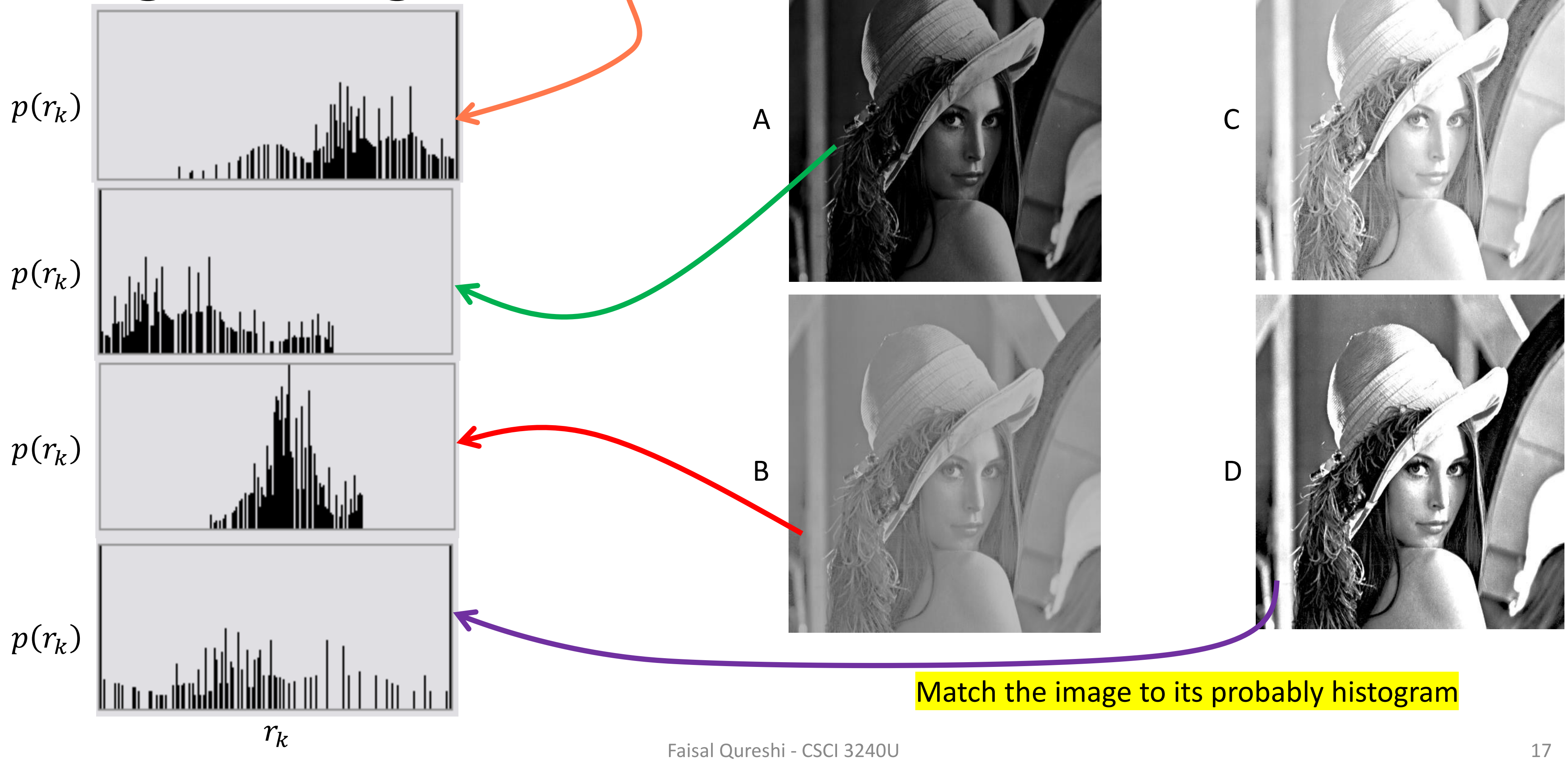
C

B

D

Match the image to its probably histogram

Image Histogram



Match the image to its probably histogram

Image Histogram

- Contrast is defined as the ratio of the maximum intensity to minimum intensity

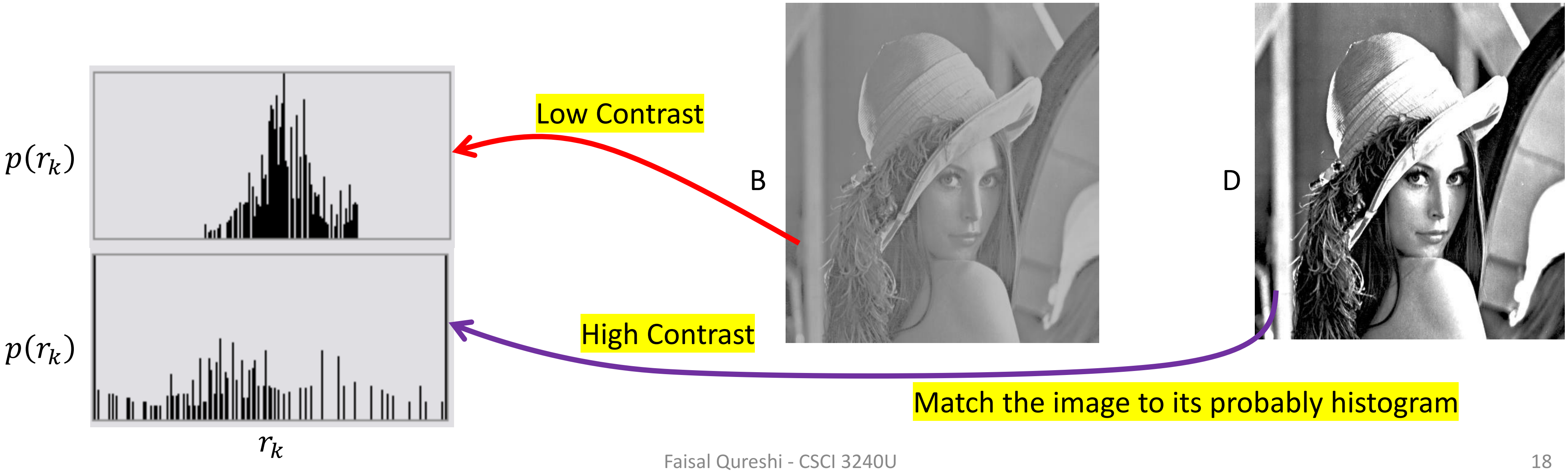


Image Histogram

- Do the following two images have the same or different histograms?

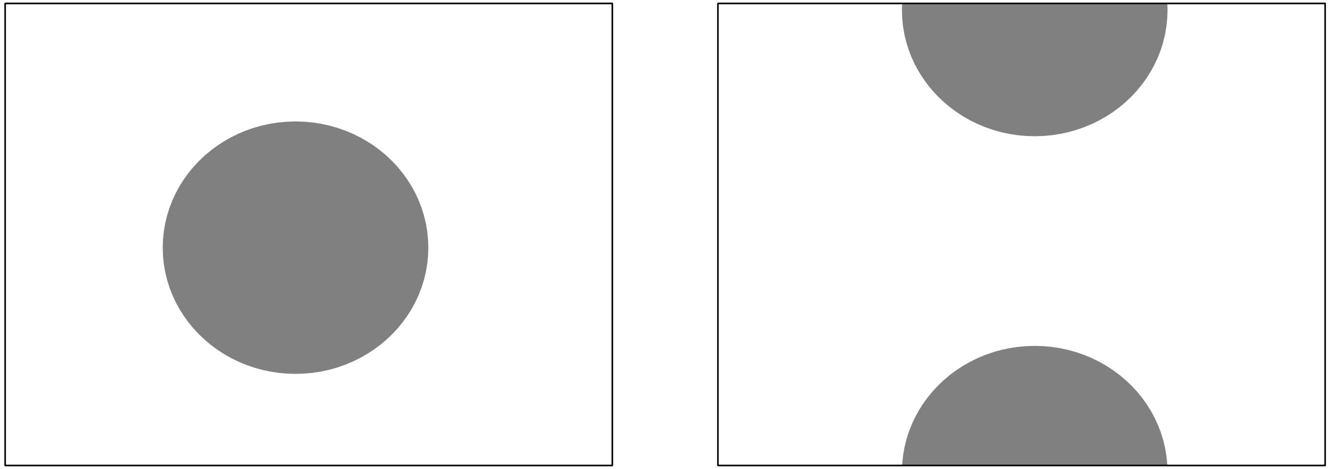
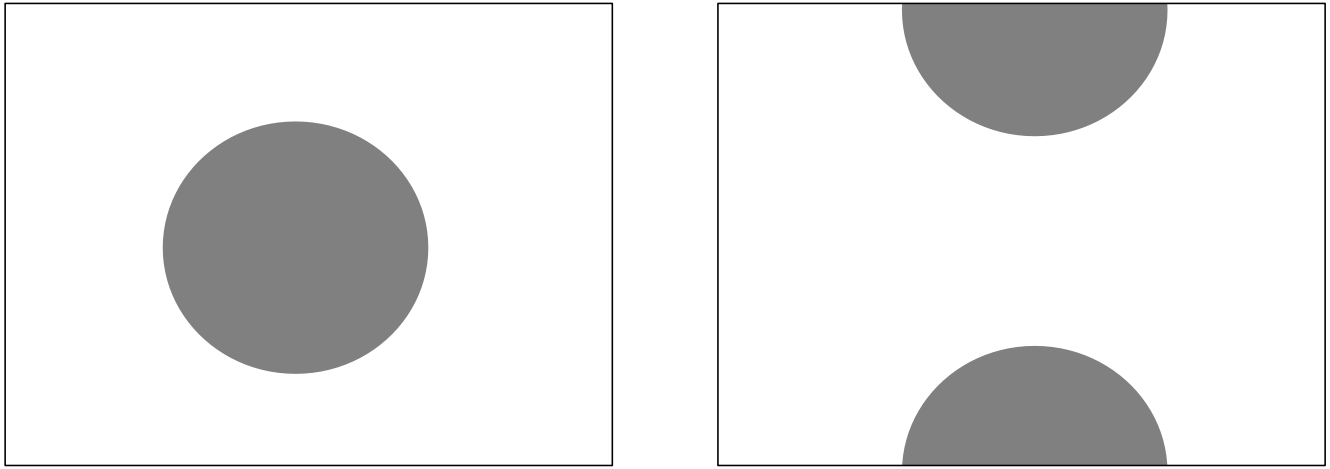


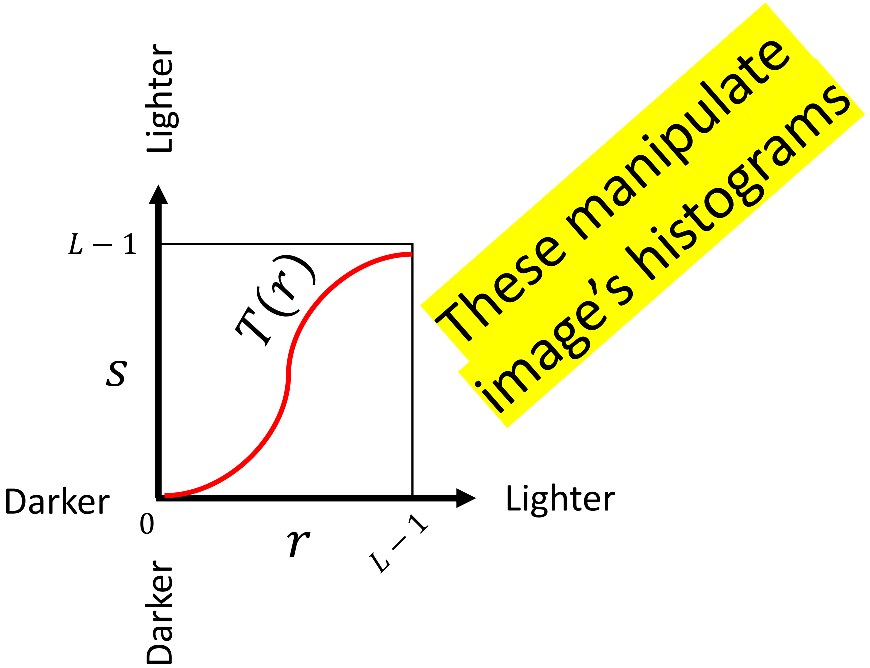
Image Histogram

- Do the following two images have the same or different histograms?



Same. Histograms are just counts.
These are not spatially aware.

Adjustment Curves

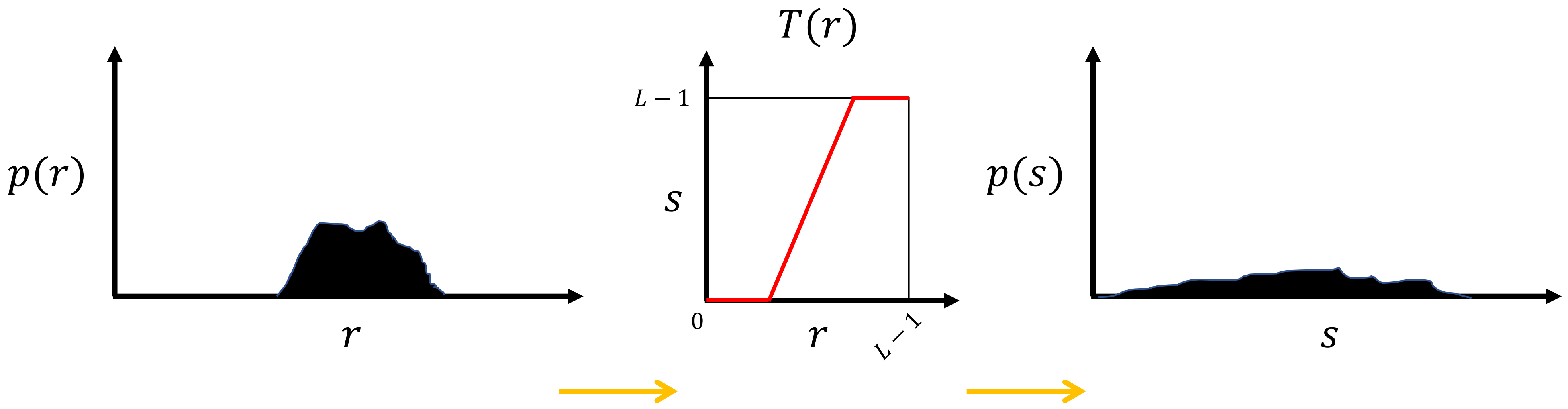


r = input pixel intensity
s = output pixel intensity

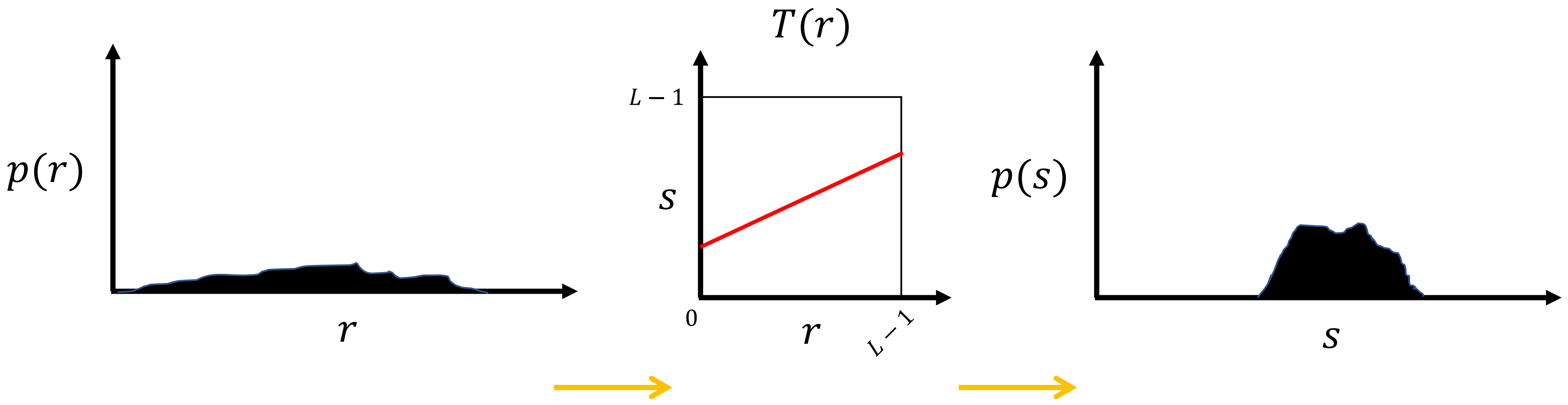
Image using pixels r

Image using pixels s

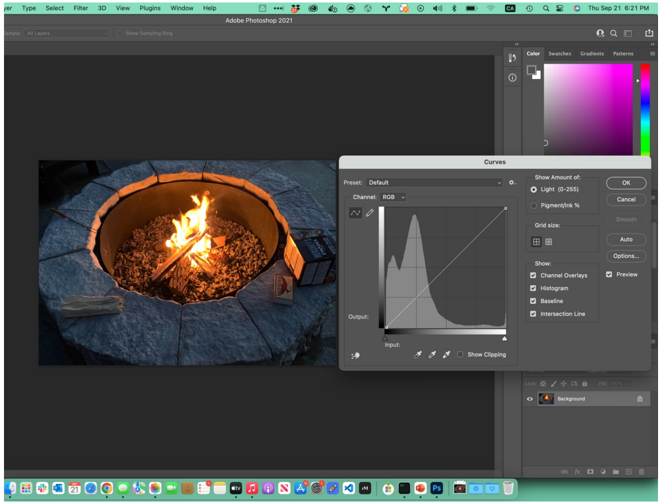
Increase Contrast

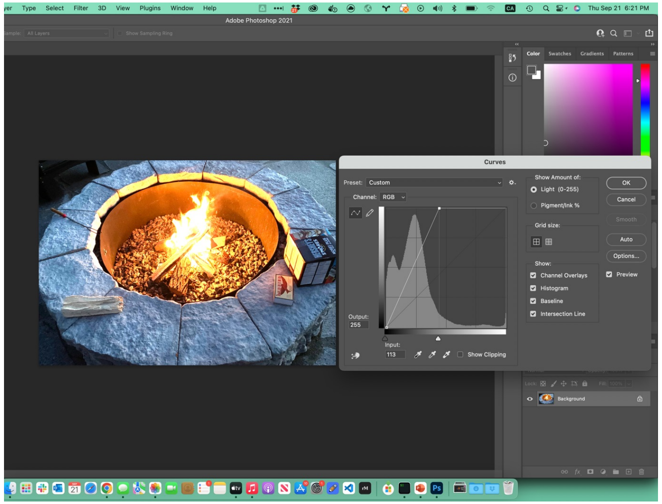


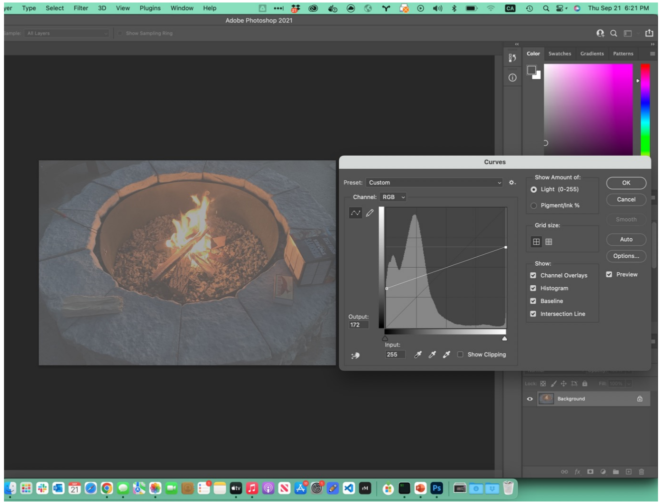
Decrease Contrast



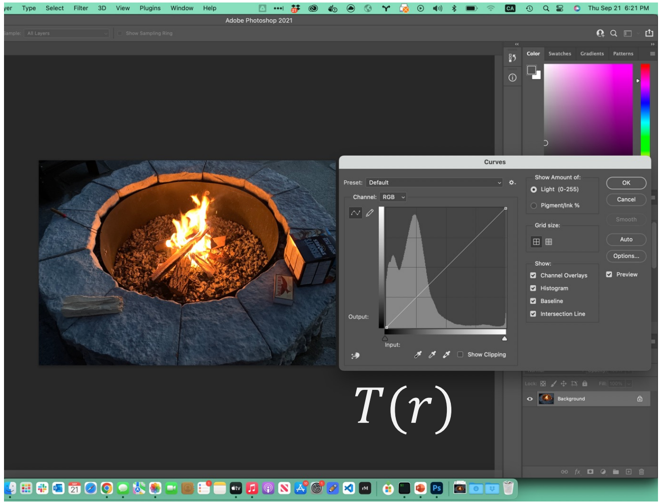
Photoshop: Image > Adjustment > Curves

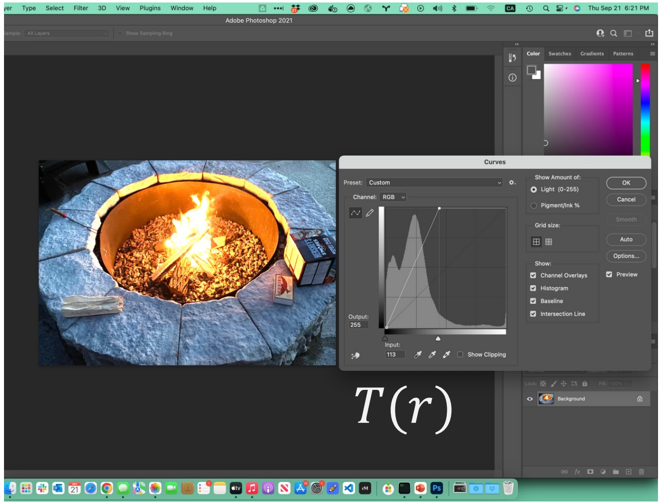


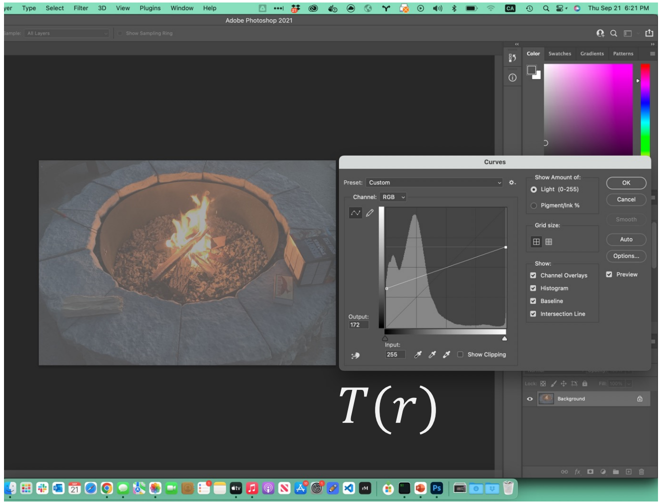




Photoshop: Image > Adjustment > Curves







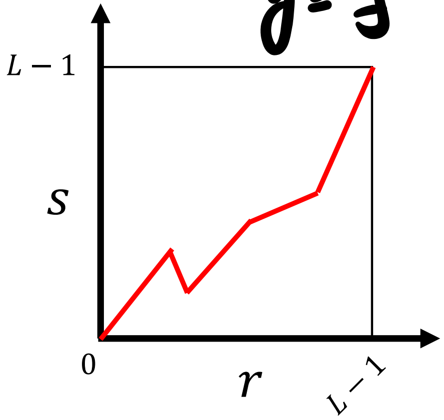
Properties of $T(r)$

- Non-monotonic
  - Does not preserve gray level ordering
  - Looks unnatural
  - Does not have an inverse

$y = f(x)$

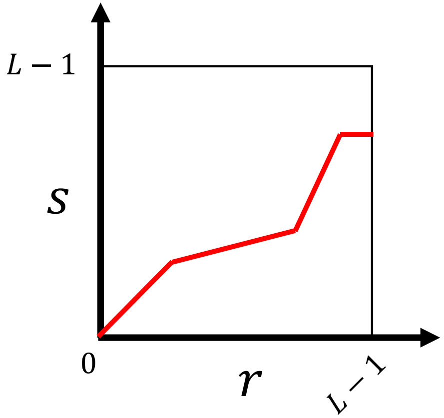
$g = f^{-1}$

$x = g(y)$



Non-monotonic

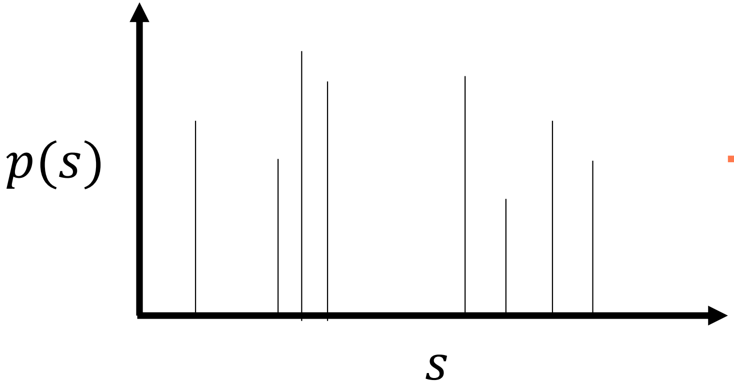
- Monotonic
  - Preserves gray level ordering
  - Looks natural
  - Inverse exists

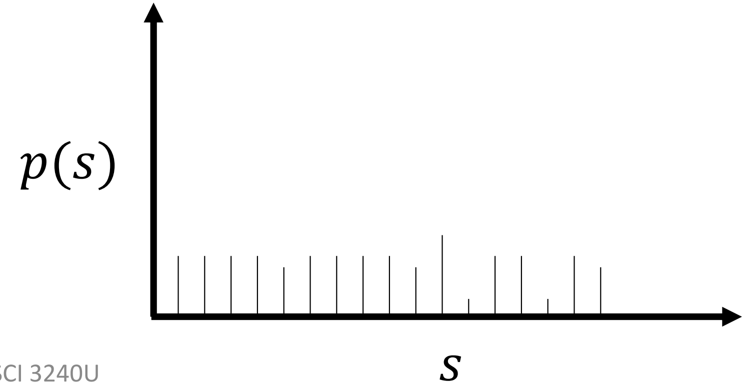


Monotonically increasing

Histogram Equalization

- Construct an image with equally many pixels at each gray level
  - Image is perceptually pleasant (nice to look at)
  - Pixel resources are maximized, so to speak
- Such an image will have an equal histogram
- Counts of pixels at each level r_k will be the "same"
  - The counts will be *roughly* equal to $(H \cdot W)/L$

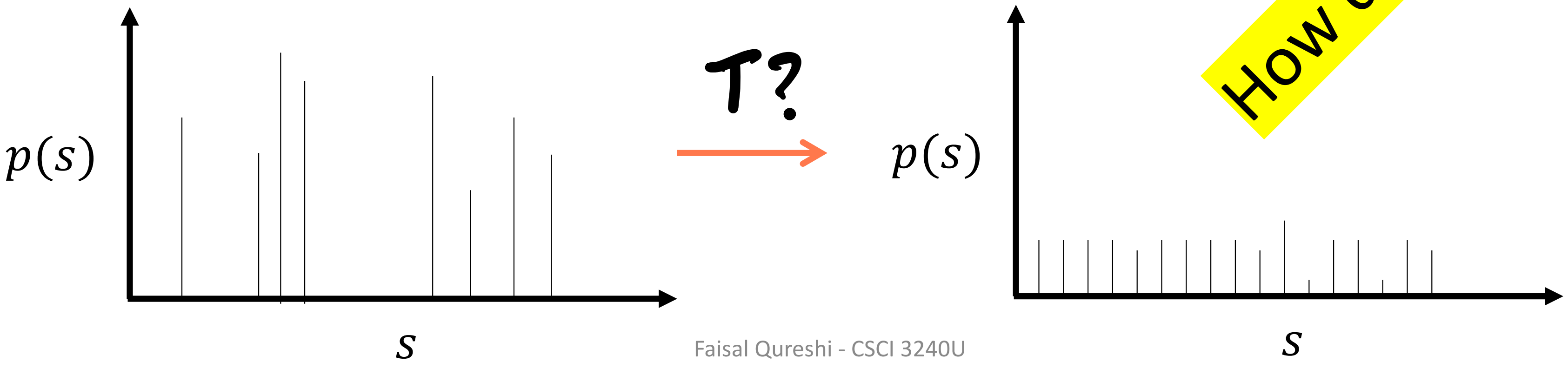




Histogram Equalization

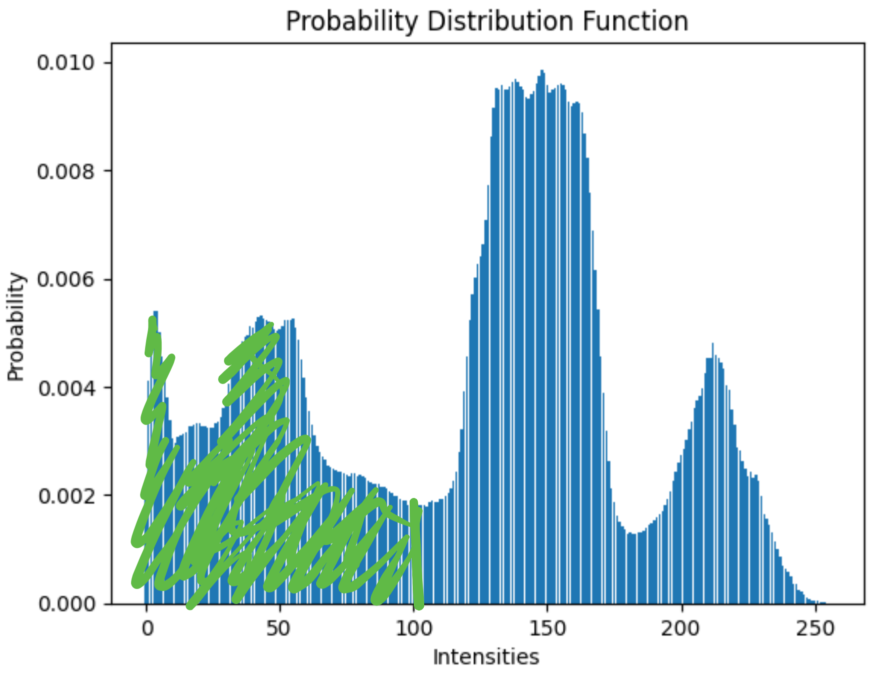
- Construct an image with equally many pixels at each gray level
  - Image is perceptually pleasant (nice to look at)
  - Pixel resources are maximized, so to speak
- Such an image will have an equal histogram
- Counts of pixels at each level r_k will be the "same"
  - The counts will be *roughly* equal to $(H \cdot W)/L$

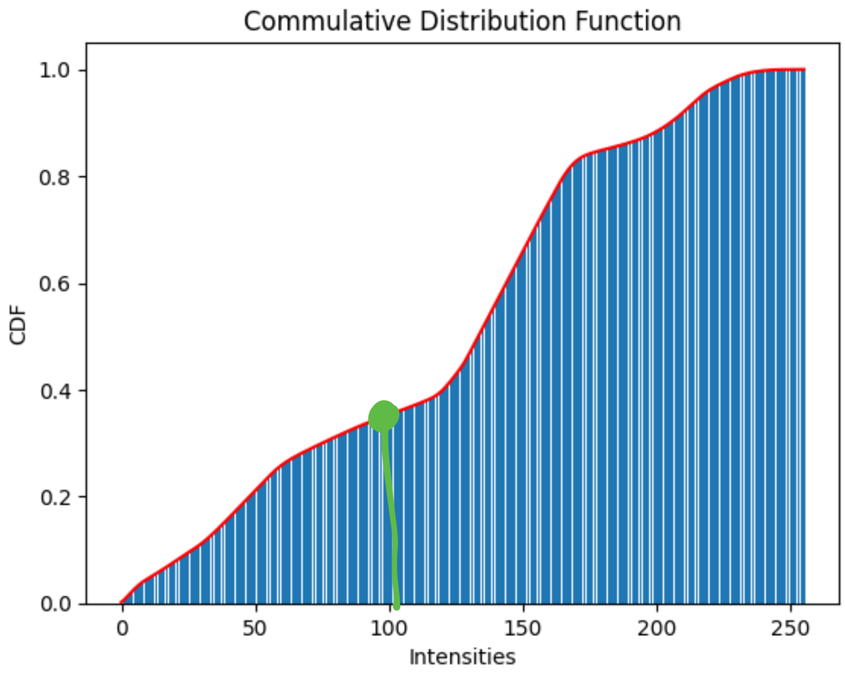
How do achieve it?



Histogram Equalization

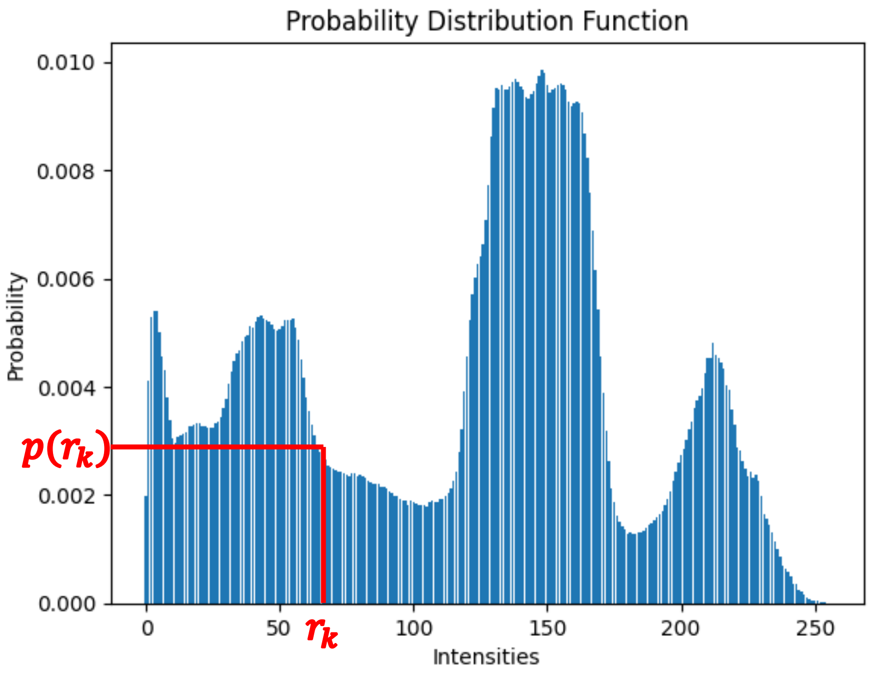
- Use cumulative distribution function to construct the necessary $T(r)$

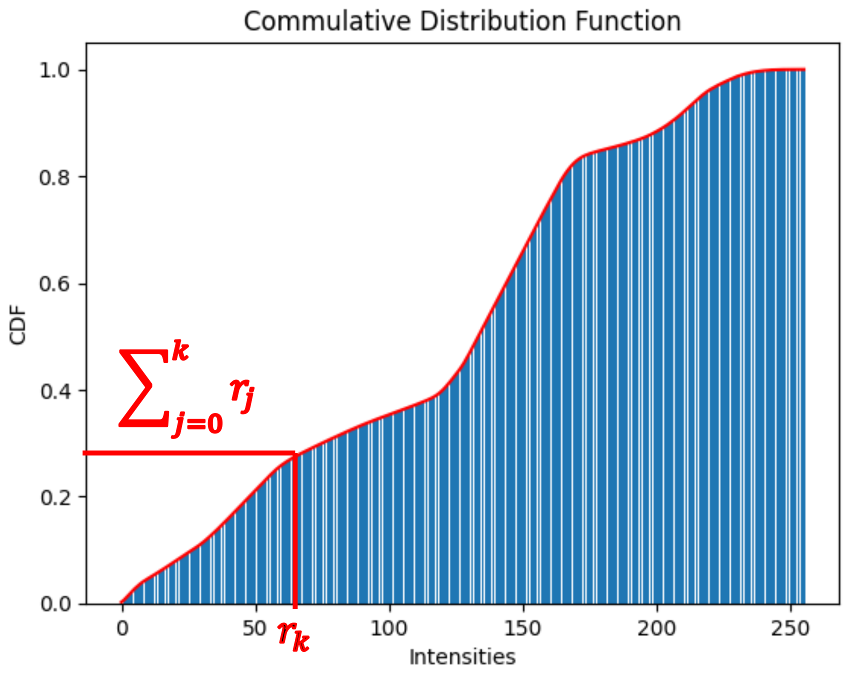




Histogram Equalization

- Use cumulative distribution function to construct the necessary $T(r)$





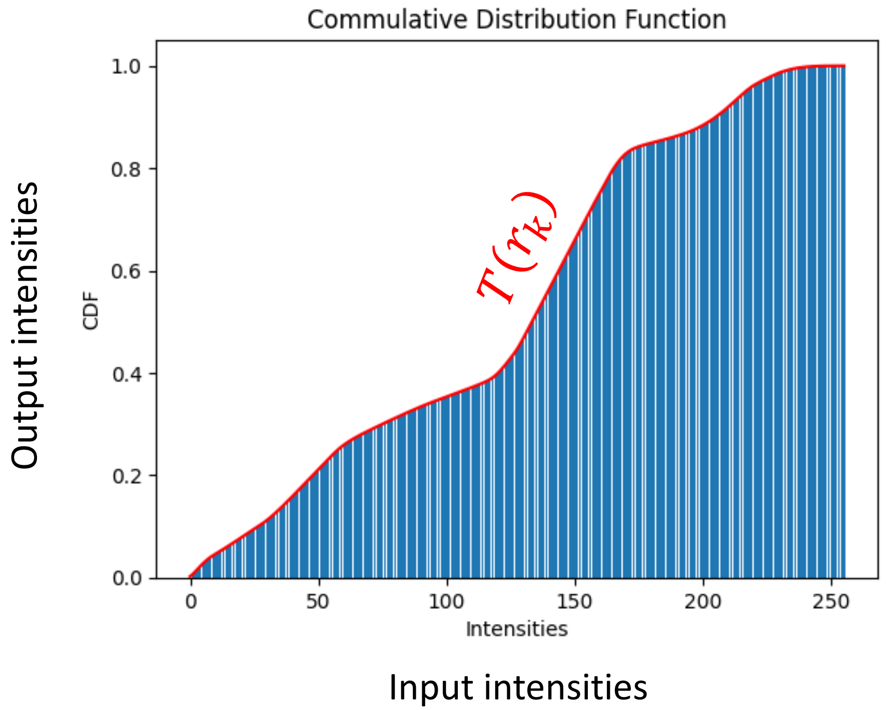
Histogram Equalization

- Use cumulative distribution function to construct the necessary $T(r)$

$$s_k = T(r_k)$$

$$= \frac{1}{H \cdot W} \sum_{j=1}^{k} n_j$$

$$= \sum_{j=1}^{k} p(r_j)$$

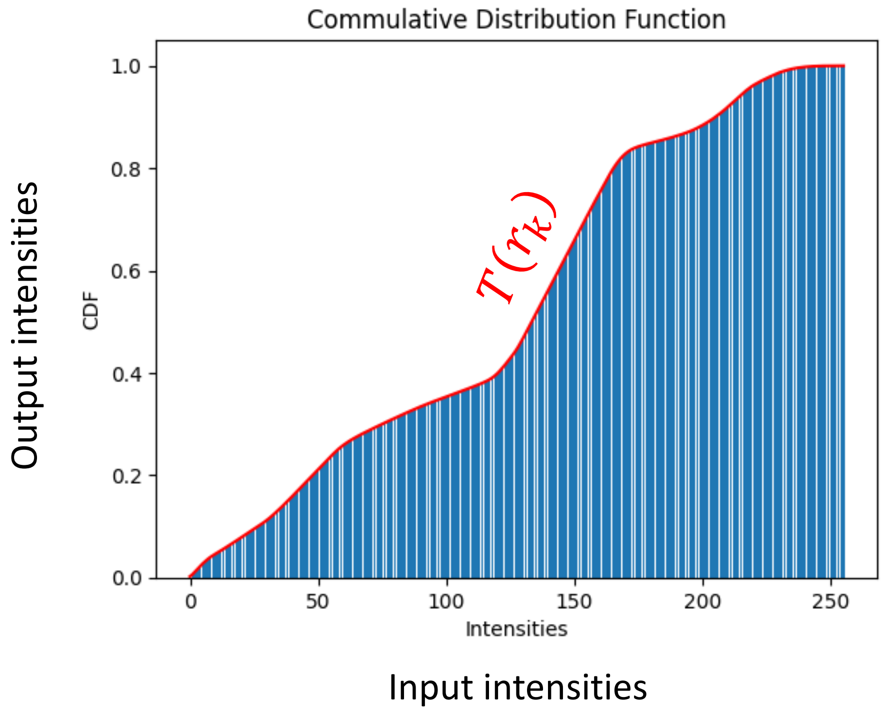


Histogram Equalization

- Use cumulative distribution function to construct the necessary $T(r)$

$$s_k = T(r_k)$$
$$= \frac{1}{H \cdot W} \sum_{j=1}^{k} n_j$$
$$= \sum_{j=1}^{k} p(r_j)$$

Input intensities fall between 0 and L-1, whereas output intensities are between 0 and 1?

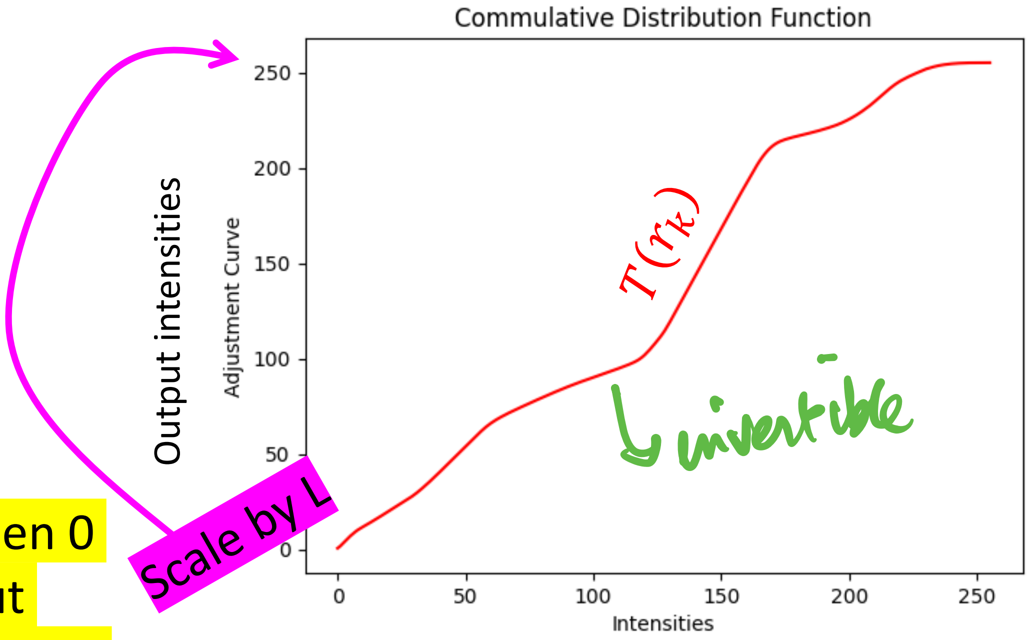


Histogram Equalization

- Use cumulative distribution function to construct the necessary $T(r)$

$$s_k = T(r_k)$$
$$= \frac{1}{H \cdot W} \sum_{j=1}^{k} n_j$$
$$= \sum_{j=1}^{k} p(r_j)$$

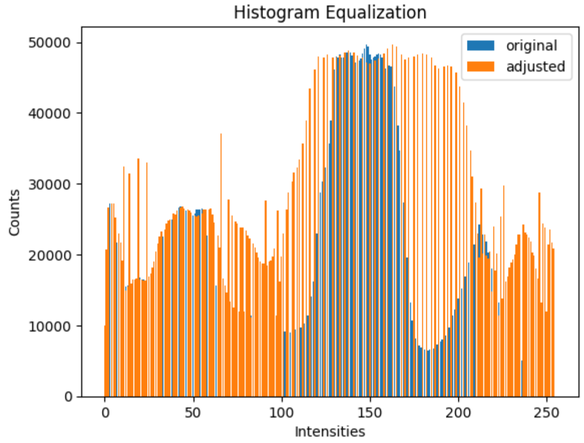
Input intensities fall between 0 and L-1, whereas output intensities are between 0 and 1?

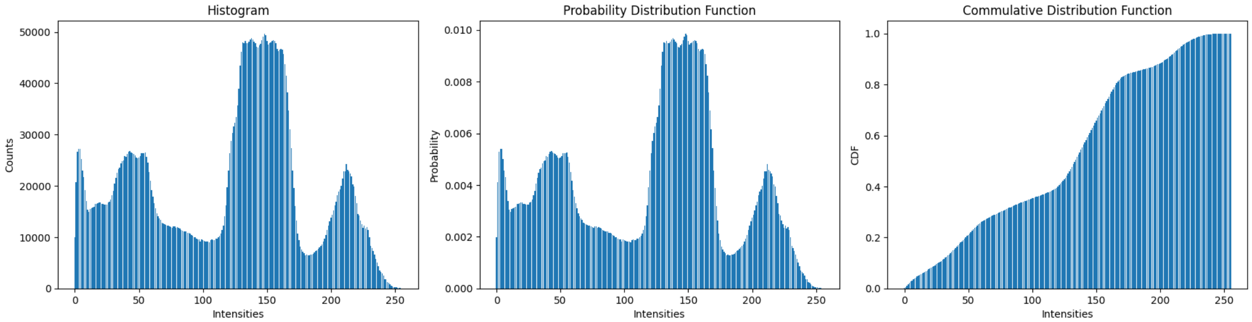


Input intensities

Histogram Equalization

Histogram before and after adjustment





Histogram Equalization

- Can significantly improve image appearance
- Automatic
  - Derived fully from the input image
- Often used as a pre-processing step
  - Accounts for lighting variations (somewhat)
  - Accounts for camera/device characteristics (somewhat)
  - Helps with image comparison
- It is possible to "recover" the original since $r = T^{-1}(s)$ exists (at least in theory)
  - Assuming a reasonable distribution of gray scales in the original image
  - This won't work if the original image was black-and-white

Summary

- Point processes for image enhancement
- Adjustment curves
- Histogram equalization

Something to Think About

- How would you use what you have learned in this lecture to develop tools to enhance the appearance of color images?