Imperial College London

4 – Intensity Transformation

Prof Peter YK Cheung

Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE4_DVS/
E-mail: p.cheung@imperial.ac.uk

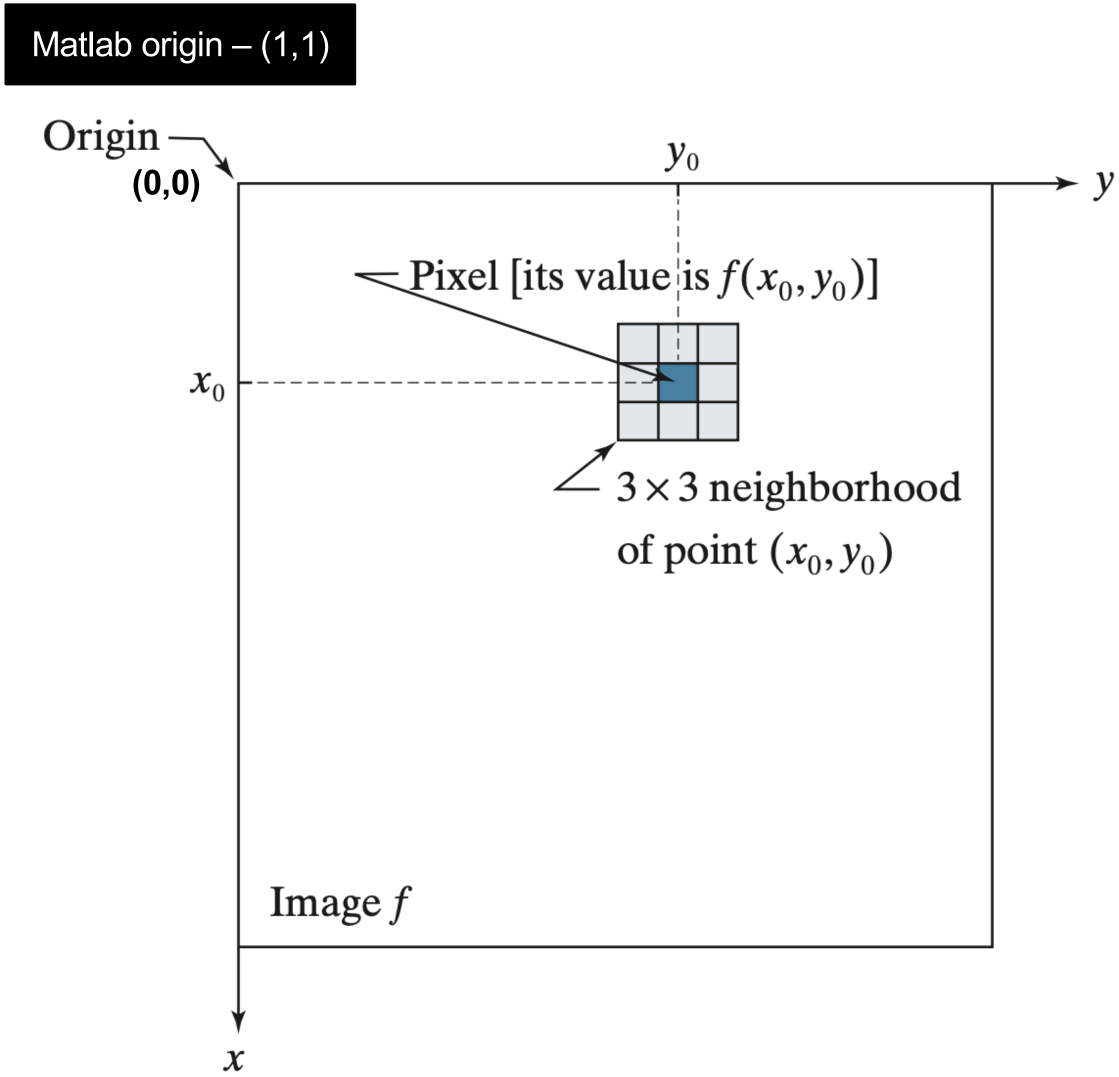
Spatial Domain Processing

- Visual data is captured and stored in computer
- Only using intensity (or luminous) information
- Scene is sampled at regular spatial grid in X and Y direction
- General equation of processing is:

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

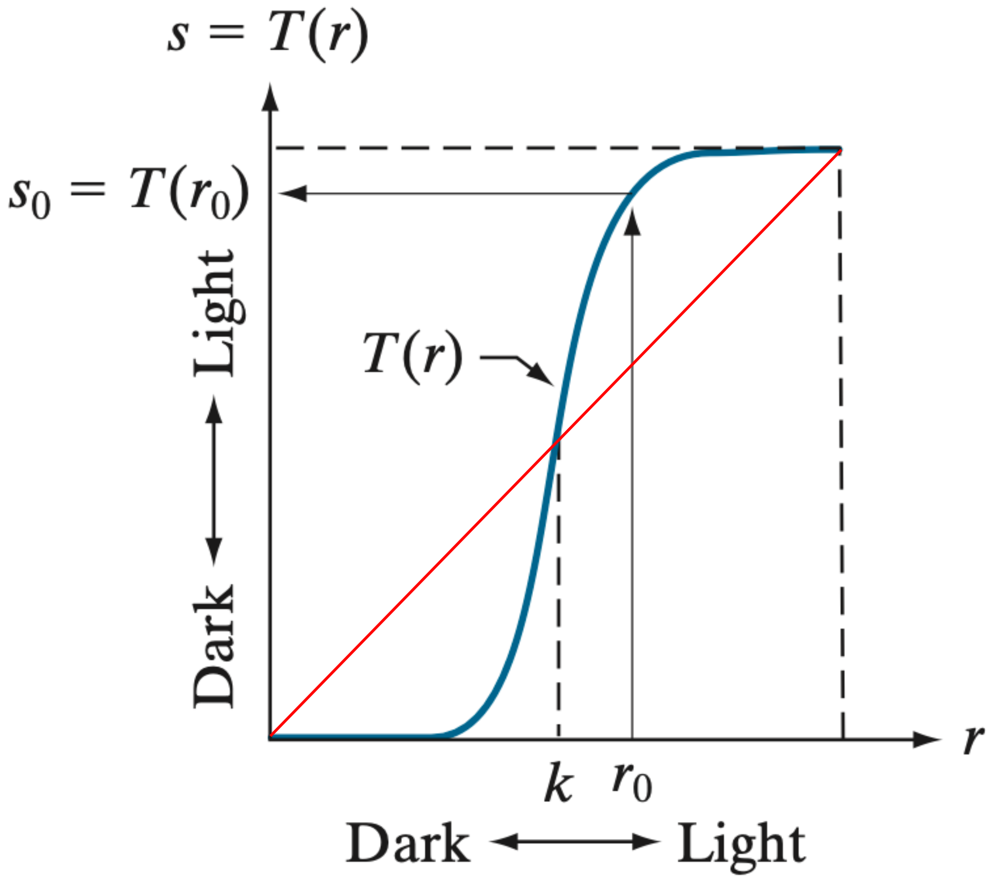
  - $f(x,y)$ is the captured visual information at (x,y)
  - $T[.]$ is the operator defined over a neighborhood of point (x, y)
  - $g(x,y)$ is the output image at (x,y)

Neighbourhood operation

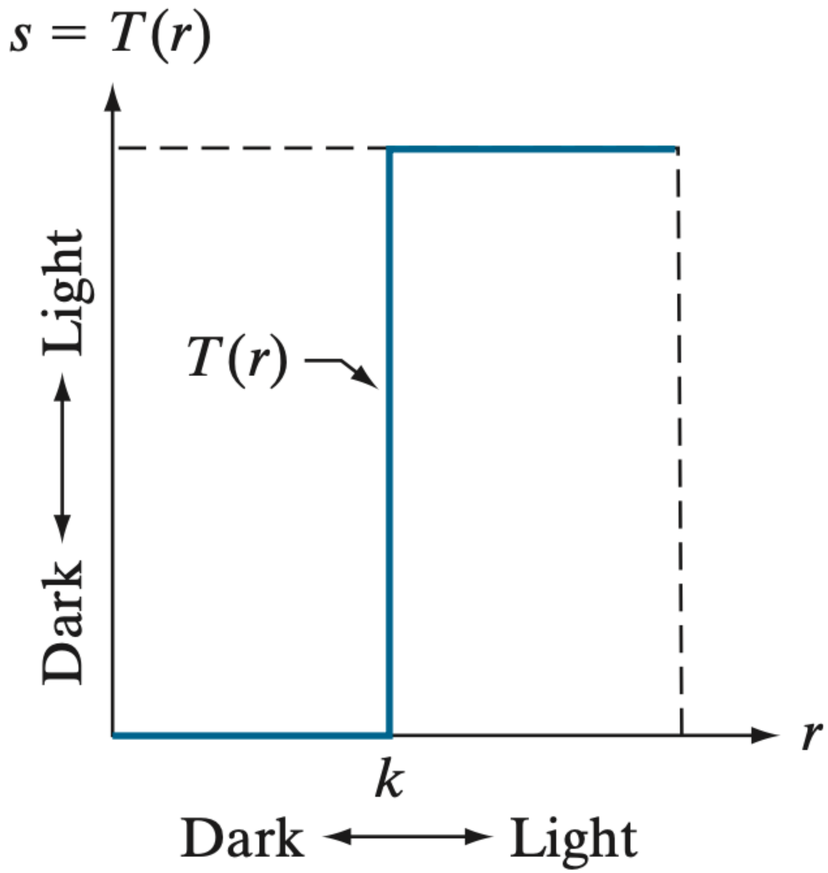


- (x_0, y_0) – arbitrary location in image
- Neighbourhood is a rectangle centred on (x_0, y_0)
- This shows a 3x3 neighbourhood
- Operator T is applied to the pixel values of neighbouring pixel surrounding $f(x_0, y_0)$
- The result of T is output image value $g(x_0, y_0)$
- Processing the entire image requires such an operation performed over the entire image, pixel-by-pixel, starting from the origin

Examples of Operator

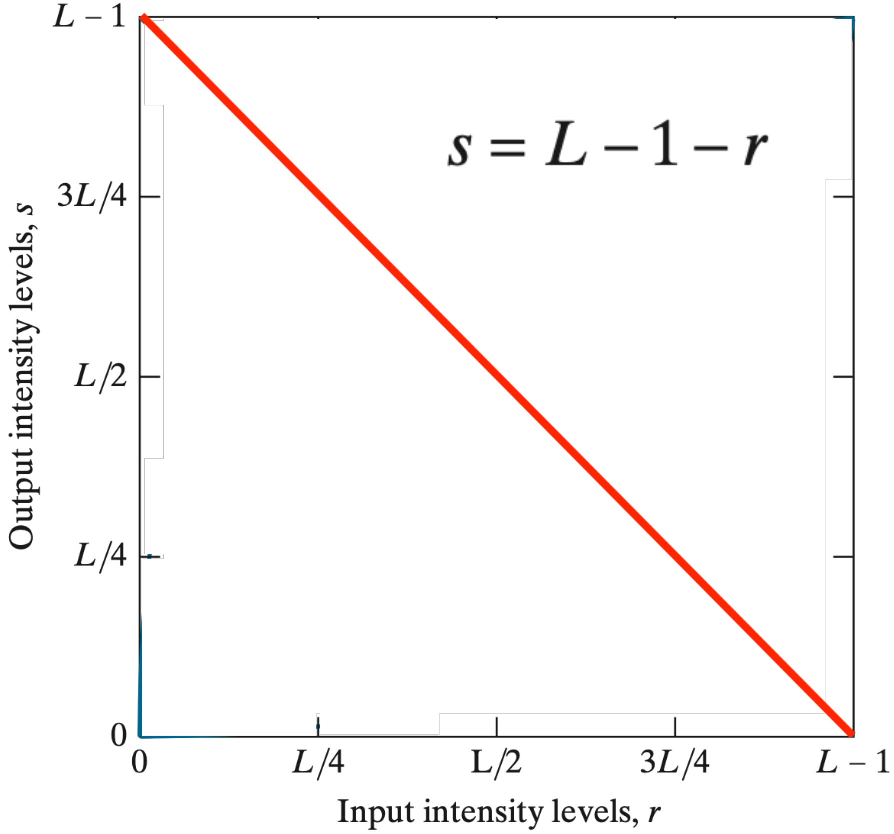


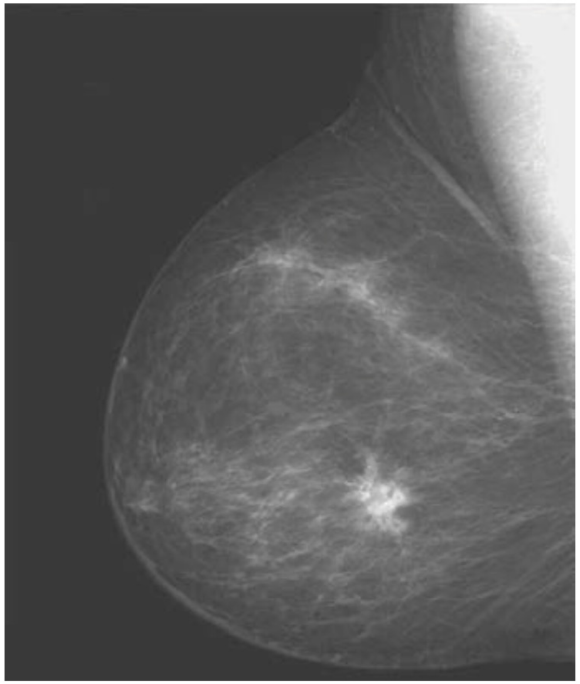
Contrast Stretching function



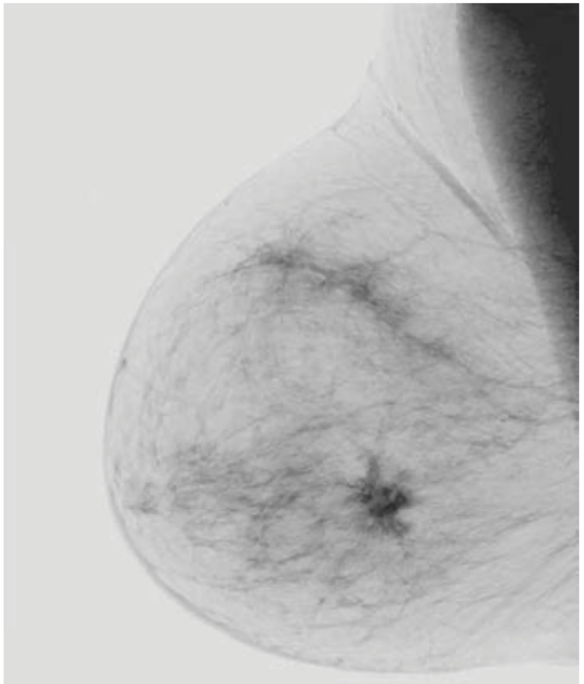
Thresholding function

Negative Transformation





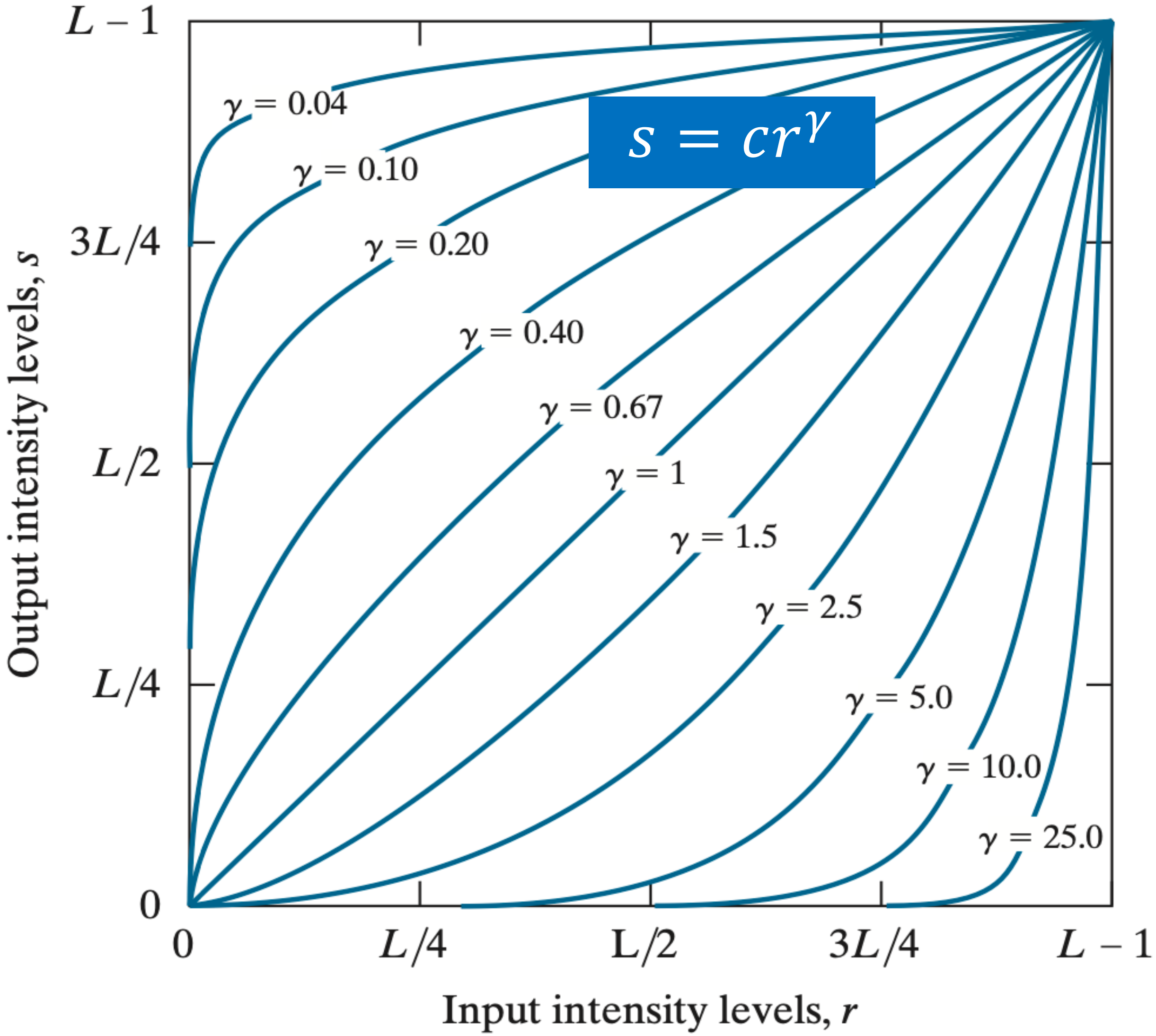
Original Image



Negative Image

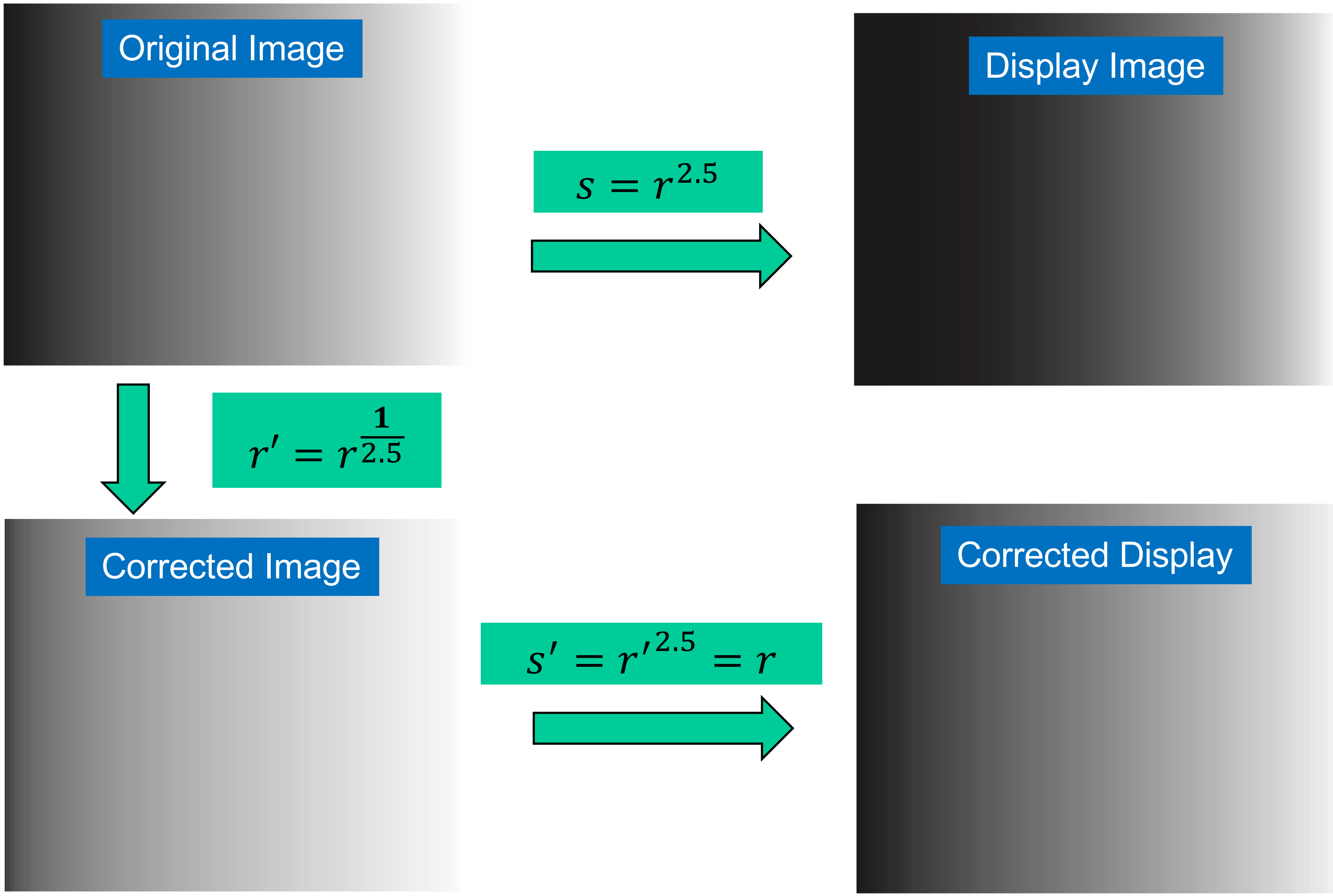
- r is the original pixel value in range [0, L-1]
- s is the resulting pixel value after negative operator

Power-law Transformation (Gamma Correction)



- c and γ are positive constants
- Better known as Gamma Correction
- Assume c = 1
- Used for processing an image for display on monitors
- Different values of γ produce a family of functions
- $\gamma = 1$ means image is not changed
- $\gamma > 1$ produces in darker image
- $\gamma < 1$ produces a ligher image
- $s = r^{\gamma}$ and $s = r^{\frac{1}{\gamma}}$ are inverse of each other

Gamma Correction for Monitor Display



Power-law Contrast Enhancement (1)

$s = r^{\gamma}$

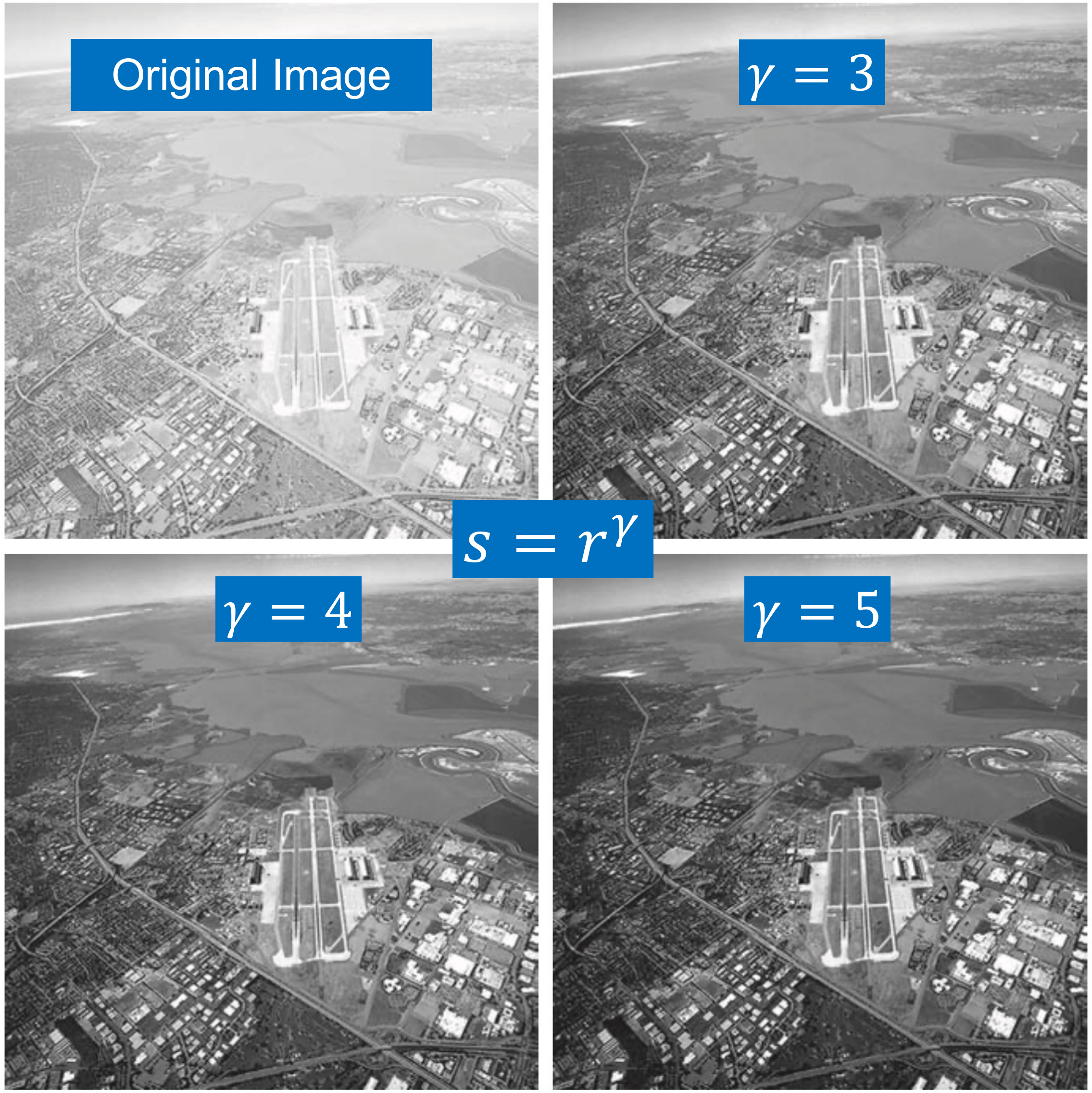
- $\gamma < 1$ lightens the image and shows break in spine

Original Image

$\gamma = 0.6$

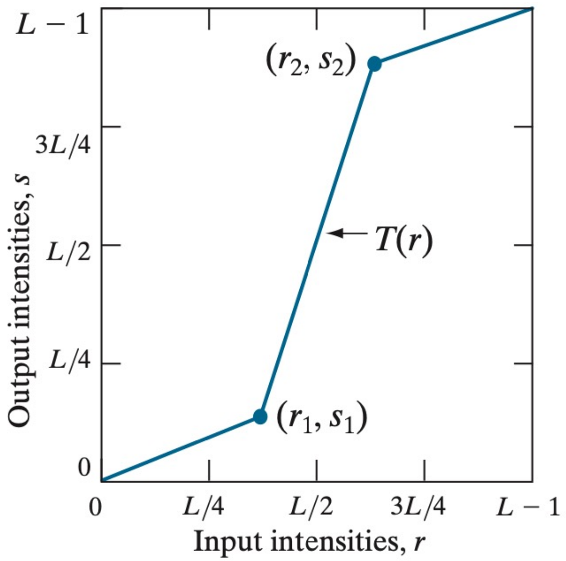
$\gamma = 0.3$

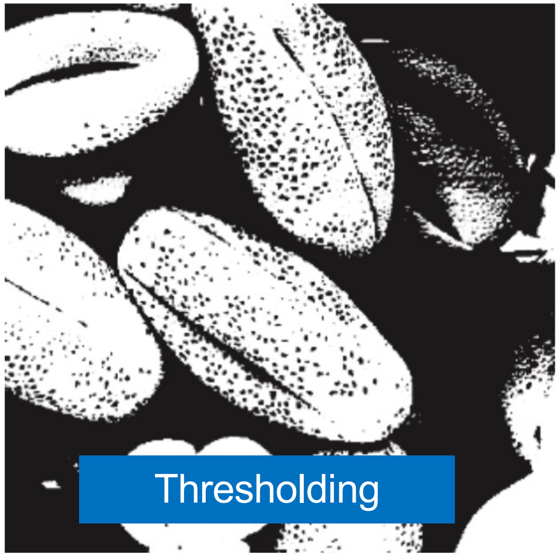
Power-law Contrast Enhancement (2)



- Original image over exposed
- $\gamma > 1$ darkens the image

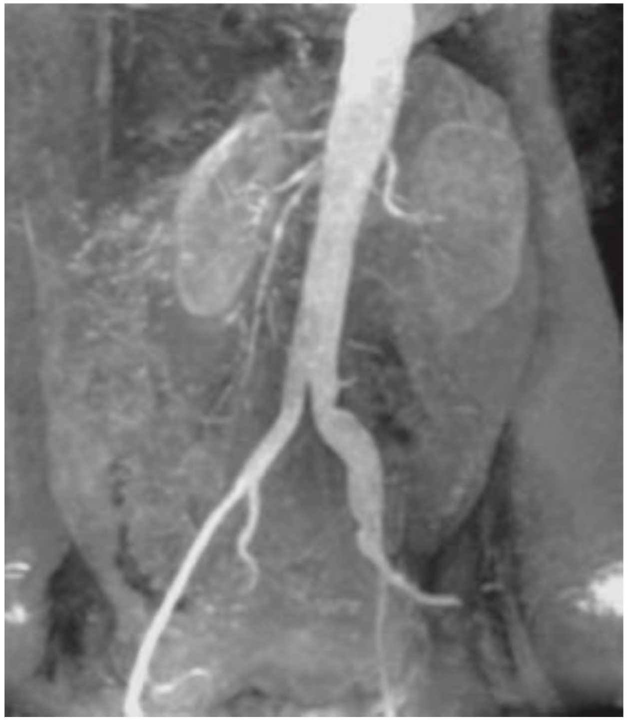
Piecewise Linear Transformation (Contrast Stretching)

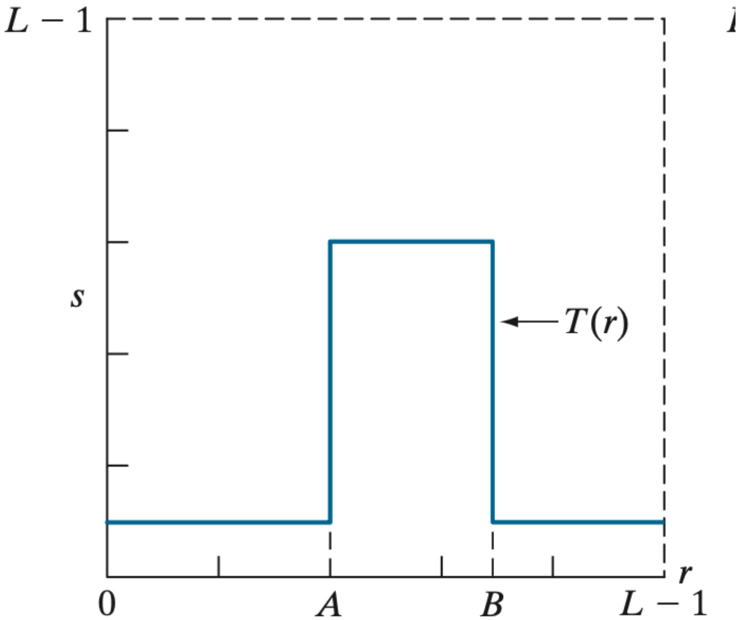


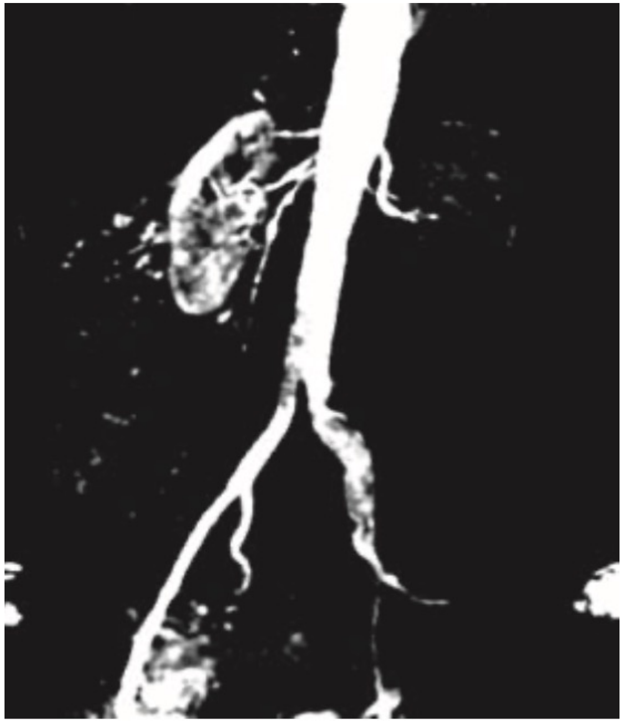


- Transformation function made up of segments of lines
- Gradient of line segments always positive
- Not a mathematical function
- Lighten and darken different grayscale regions
- E.g. low contrast electron microscope image of pollens enlarged by x700
- After contrast stretching using the piecewise linear transformation
- After thresholding only

Piecewise Linear Transformation (Intensity-level slicing) (1)

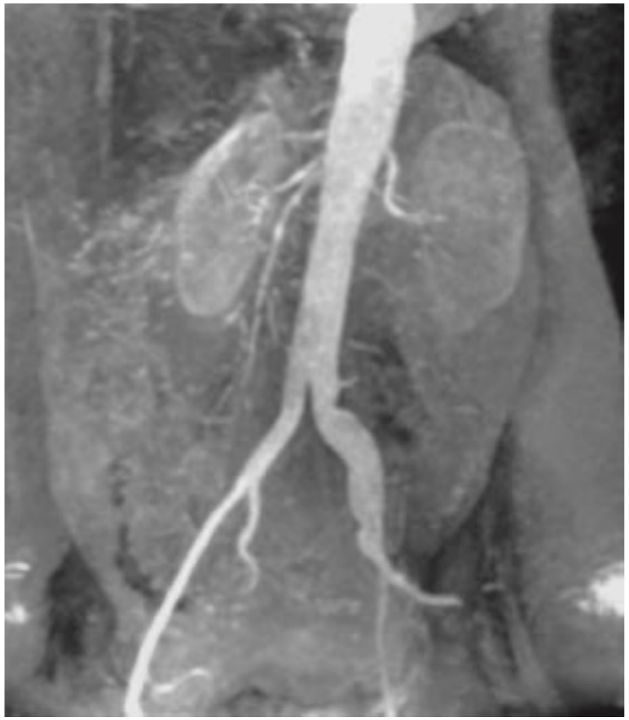


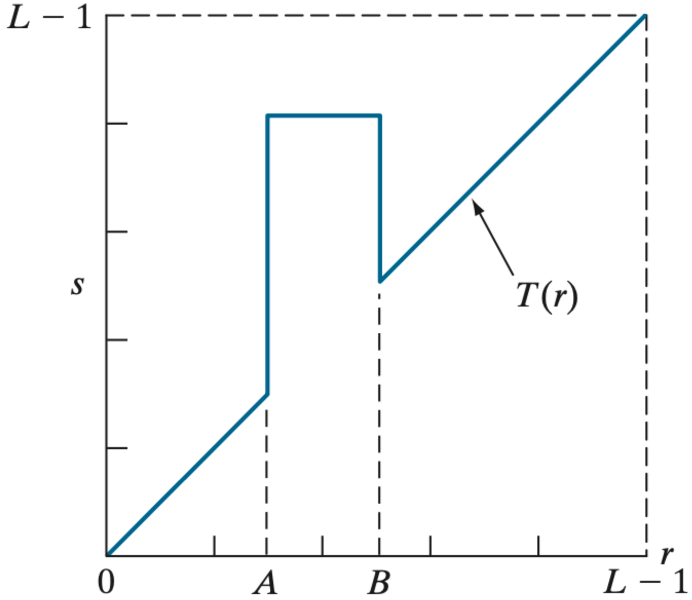


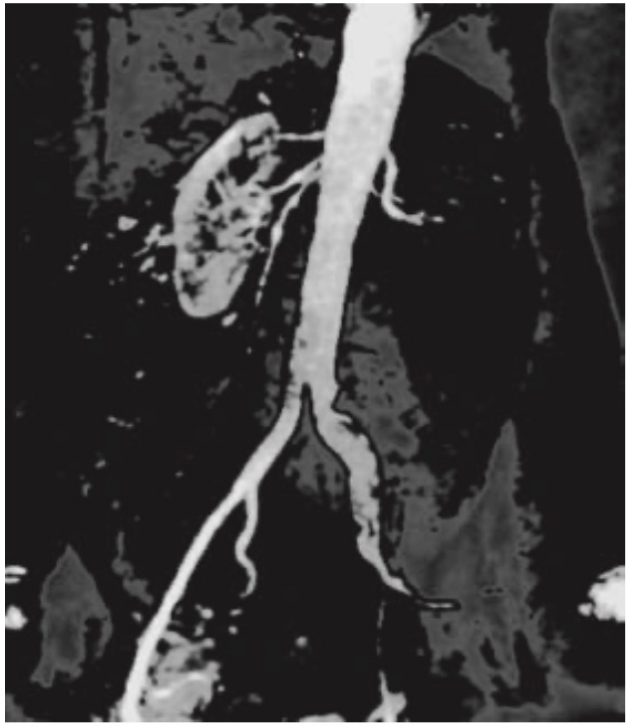


- Similar to thresholding, but use two thresholds instead of one
- Result: highlight region of image with intensity between A and B
- E.g. Aortic angiogram image with interesting feature having intensity with defined region
- Only the arteries and part of heart highlighted

Piecewise Linear Transformation (Intensity-level slicing) (2)

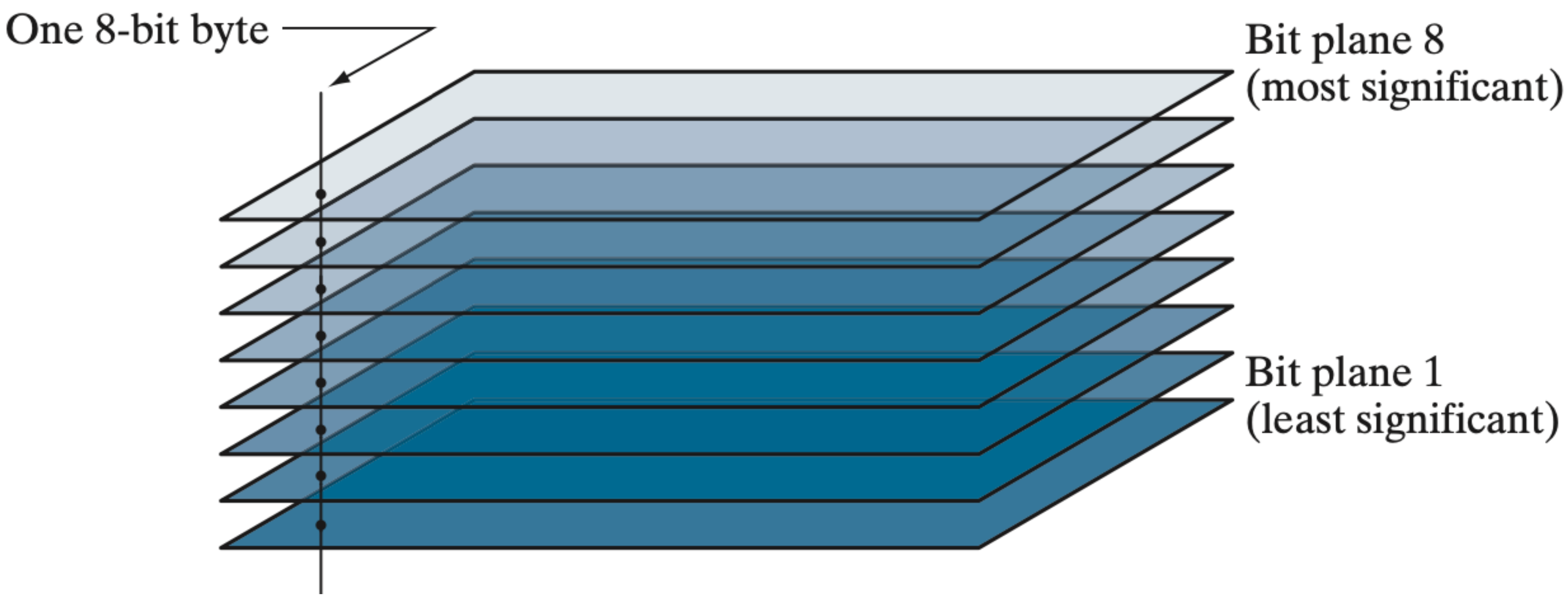






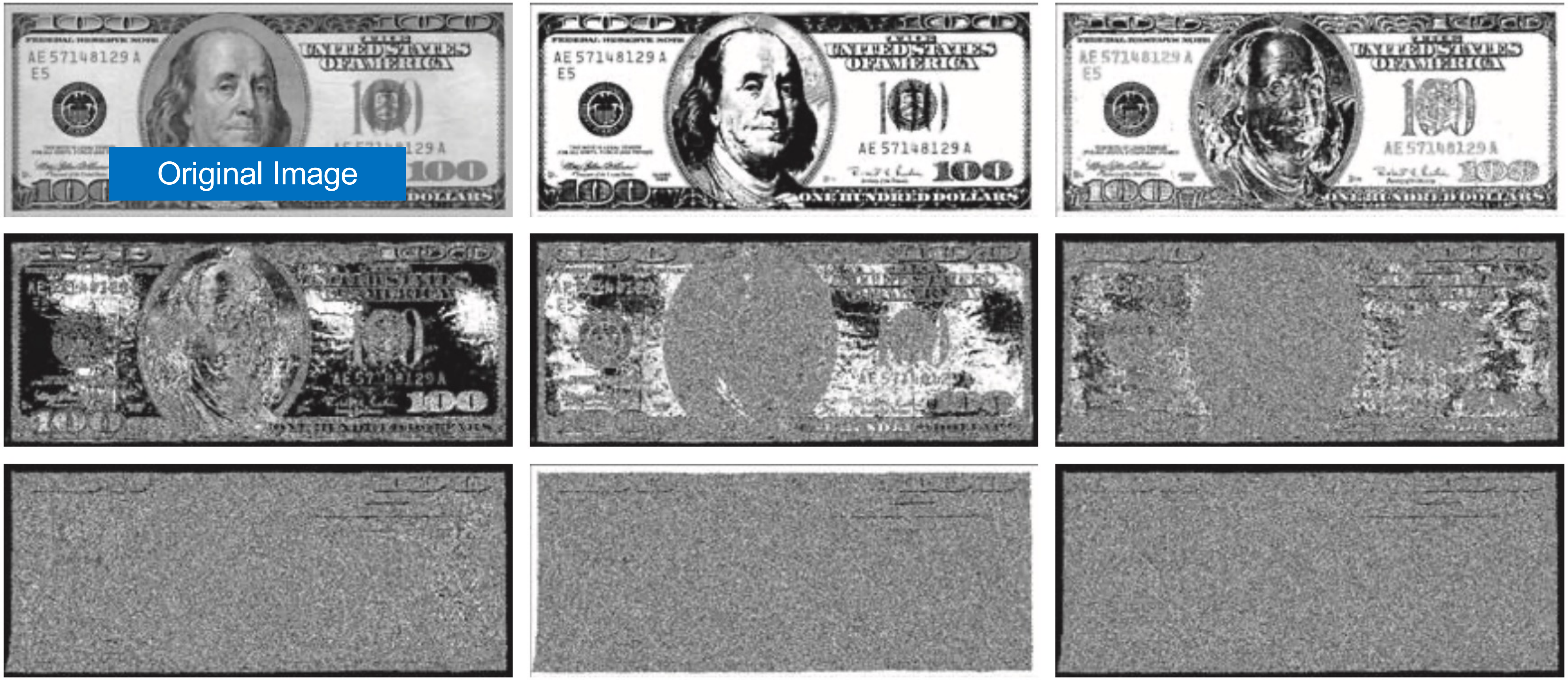
- Similar to the previous, but preserve original image intensity except for the region with intensity between A and B
- Resulting image shows features of other part of body, but still highlighting the region of interest.
- Effectively superimposing the highlighted part onto the original image

Bit-plane slicing



- Intensity is 8-bit number
- Slice each pixel intensity values into images of EACH BIT
- Produces 8 separate images – what for?

Bit-plane slicing of a US bank note (1)



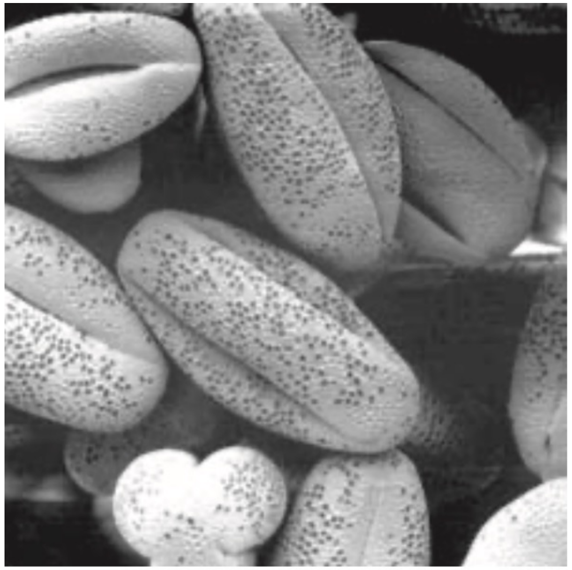
- Slicing original image of bank note shown in top left
- Clearly the most-significant bit image is most important (not surprising)
- Also shows that image of the bottom 4-bits contain little information

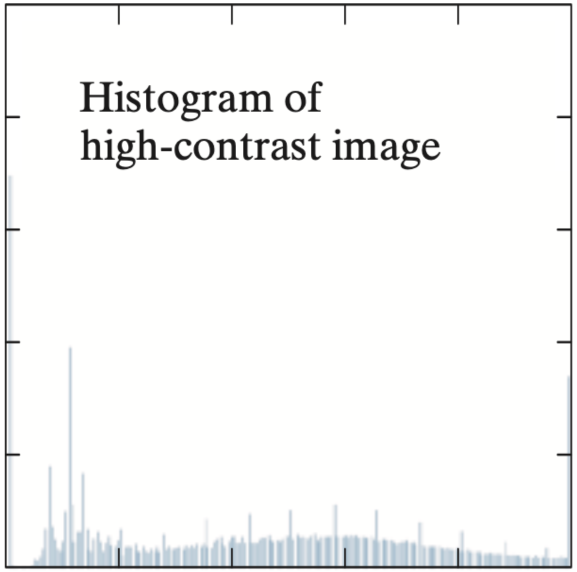
Bit-plane slicing of a US bank note (2)

Original with 8-bit intensity

With only most-significant 4 bits

Formal Definition of Histogram





Let r_k, for $k = 0,1,2,..., L-1$, denote the intensities of an *L*-level digital image, $f(x, y)$.

The **unnormalized histogram** of f is defined as:

$$h(r_k) = n_k \quad \text{for } k = 0,1,2,\ldots,L-1$$

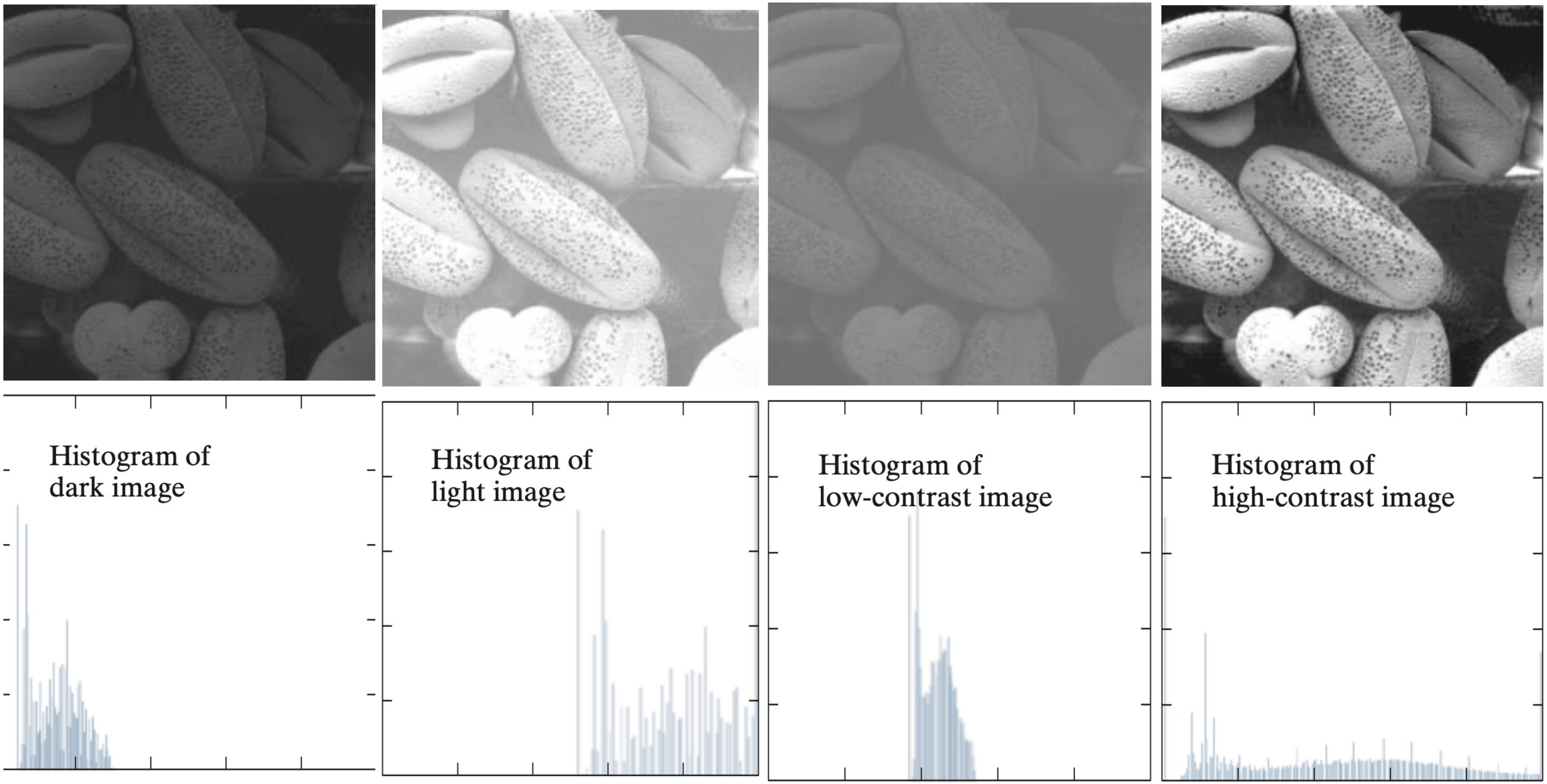
n_k is the number of pixels in f with intensity r_k.

The subdivisions of the intensity scale are called **histogram bins**.

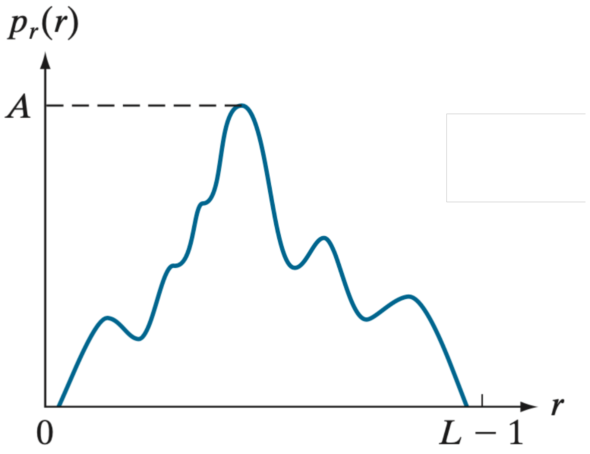
The **normalized histogram** of image f of dimension (MxN)is defined as:

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

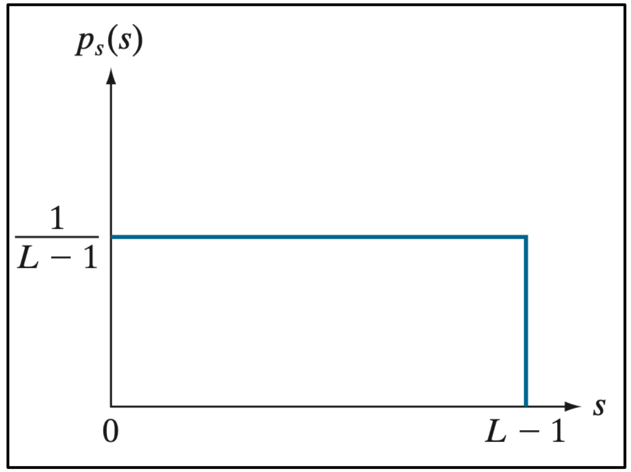
Histogram affects contrast

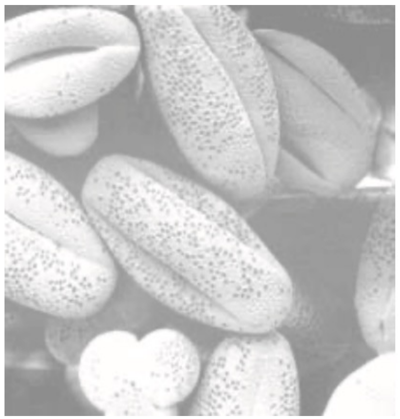


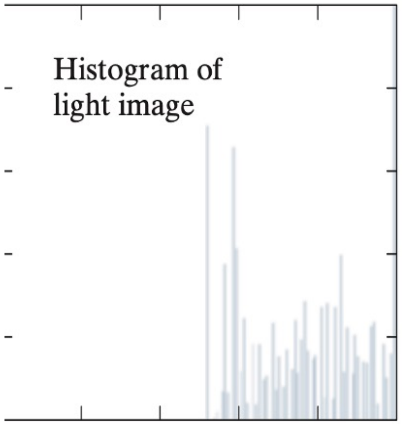
Histogram Equalization



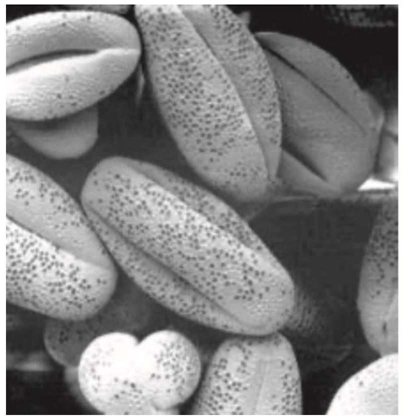
Ideal equalization

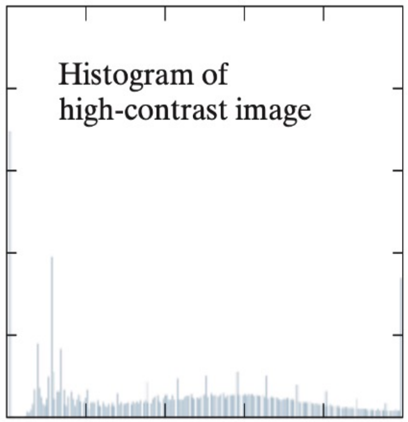




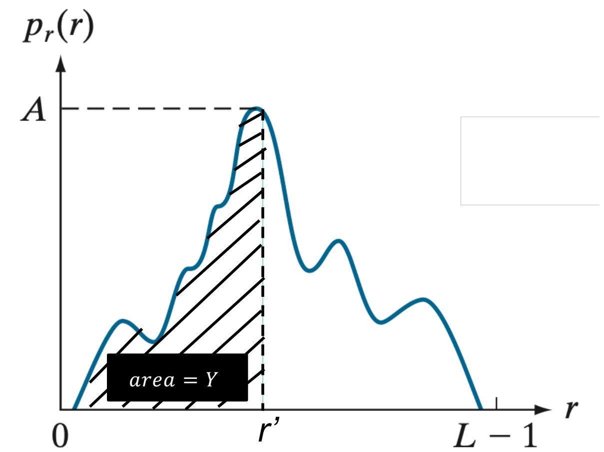


Practical equalization



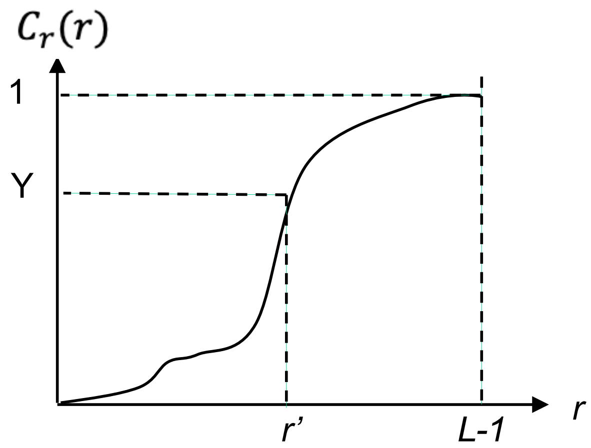


PDF and CDF



Probability Distribution Function
PDF

Integrate = area under curve



Type equation here.Cumulative Distribution Function

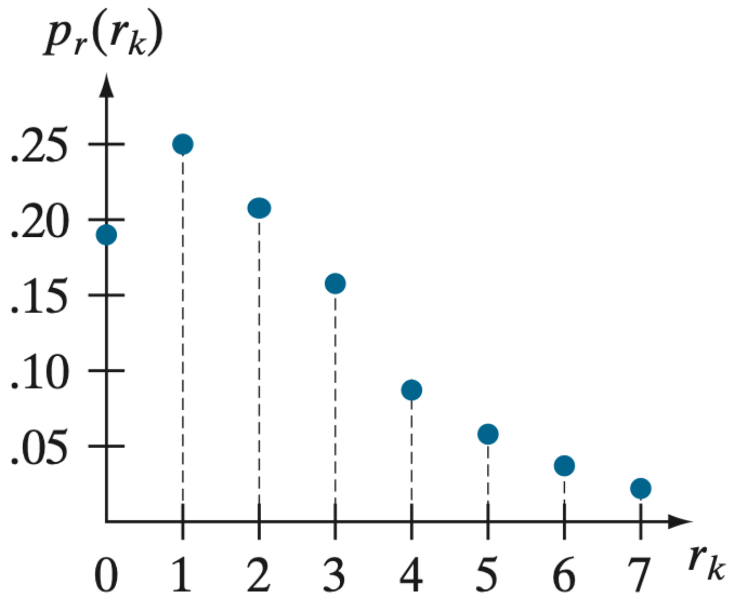
$$C_r(r) = (L-1)\int_0^r p_r(w)dw$$

- Histogram equalization (i.e. flattening of the PDF) can be achieved by using the CDF as the intensity transformation function

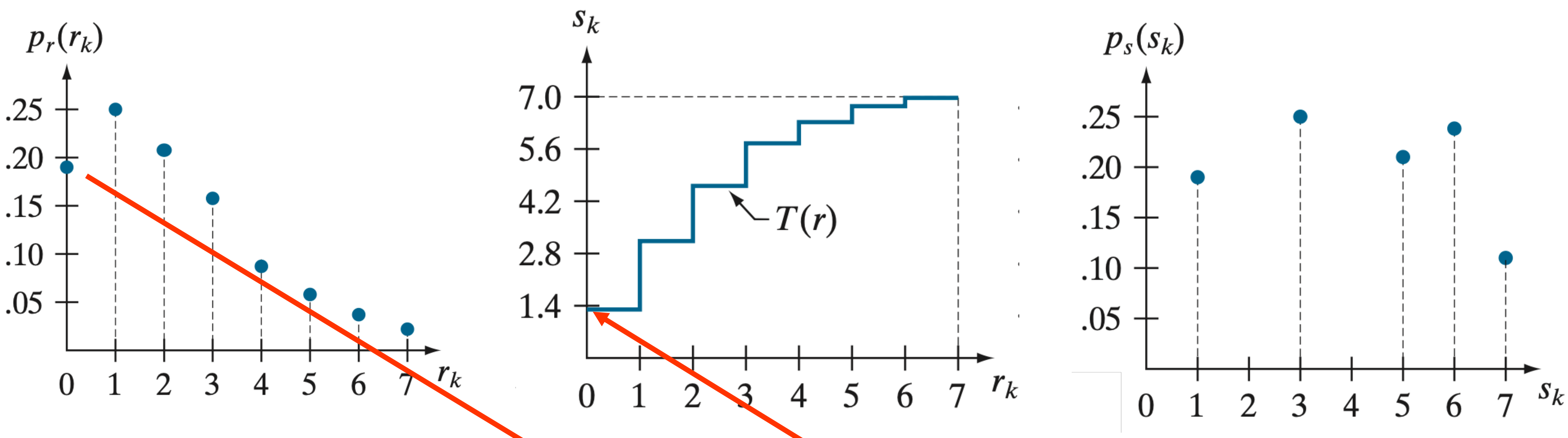
$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

An example – 3-bit intensity distribution of a 64 x 64 image

r_k	n_k	$p_r(r_k) = n_k / MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



Compute the CDF and use as intensity transform function

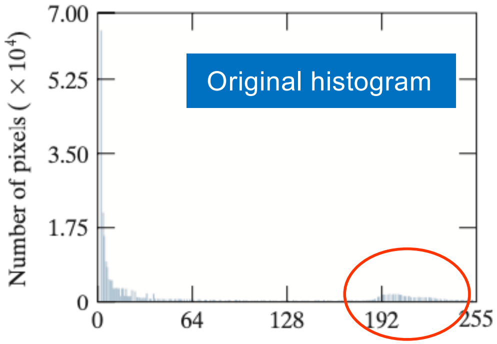


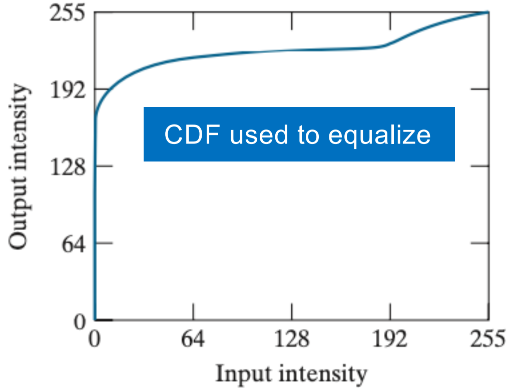
$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

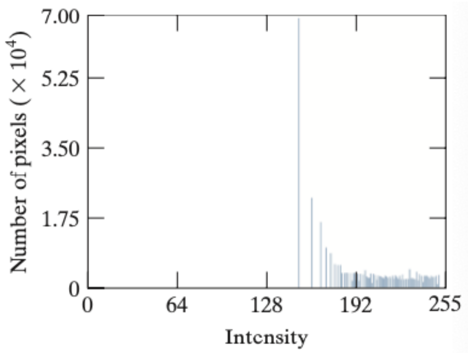
$s_0 = 1.33 \rightarrow 1$	$s_2 = 4.55 \rightarrow 5$	$s_4 = 6.23 \rightarrow 6$	$s_6 = 6.86 \rightarrow 7$
$s_1 = 3.08 \rightarrow 3$	$s_3 = 5.67 \rightarrow 6$	$s_5 = 6.65 \rightarrow 7$	$s_7 = 7.00 \rightarrow 7$

- Note that after equalization, the final histogram have no values at 0 and 2!
- Equalized histogram is approximately flat.

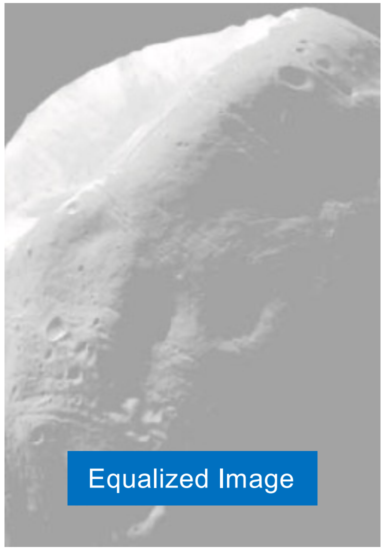
Histogram equalization may not work



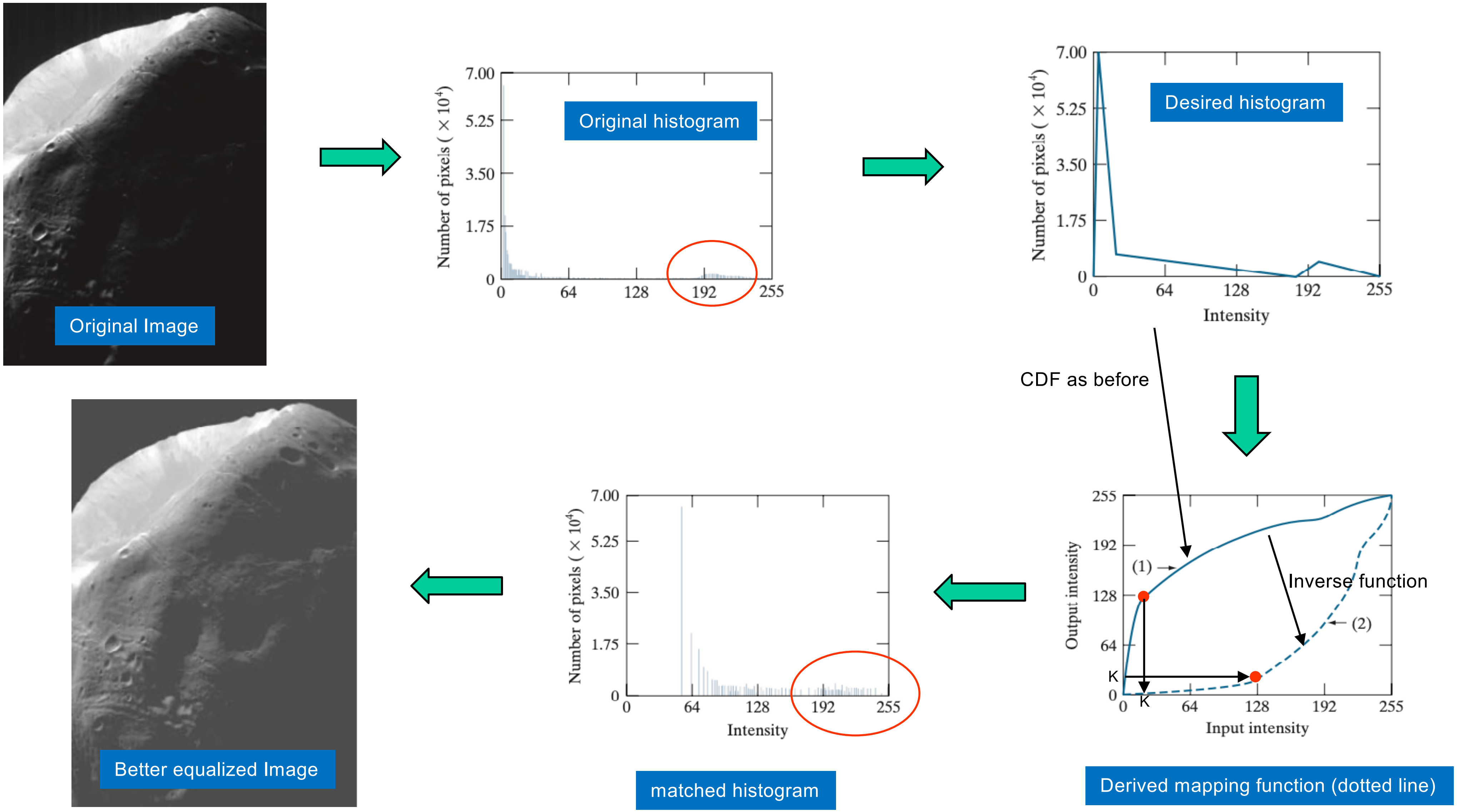




Equalized histogram



Histogram Matching



Matlab Functions related to this Lecture

imadjust

Adjust image intensity values or colormap

Syntax

```
J = imadjust(I)
J = imadjust(I,[low_in high_in])
J = imadjust(I,[low_in high_in],[low_out high_out])
J = imadjust(I,[low_in high_in],[low_out high_out],gamma)
```

imhist

Histogram of image data

Syntax

```
[counts,binLocations] = imhist(I)
[counts,binLocations] = imhist(I,n)
```

histeq

Enhance contrast using histogram equalization

Syntax

```
J = histeq(I)
J = histeq(I,n)
J = histeq(I,hgram)
```

imhistmatch

Adjust histogram of 2-D image to match histogram of reference image

Syntax

```
J = imhistmatch(I,ref)
J = imhistmatch(I,ref,nbins)
```

stretchlim

Find limits to contrast stretch image

Syntax

```
lowhigh = stretchlim(I)
lowhigh = stretchlim(I,Tol)
```