Image Processing

Lecture 03 – Point Processing

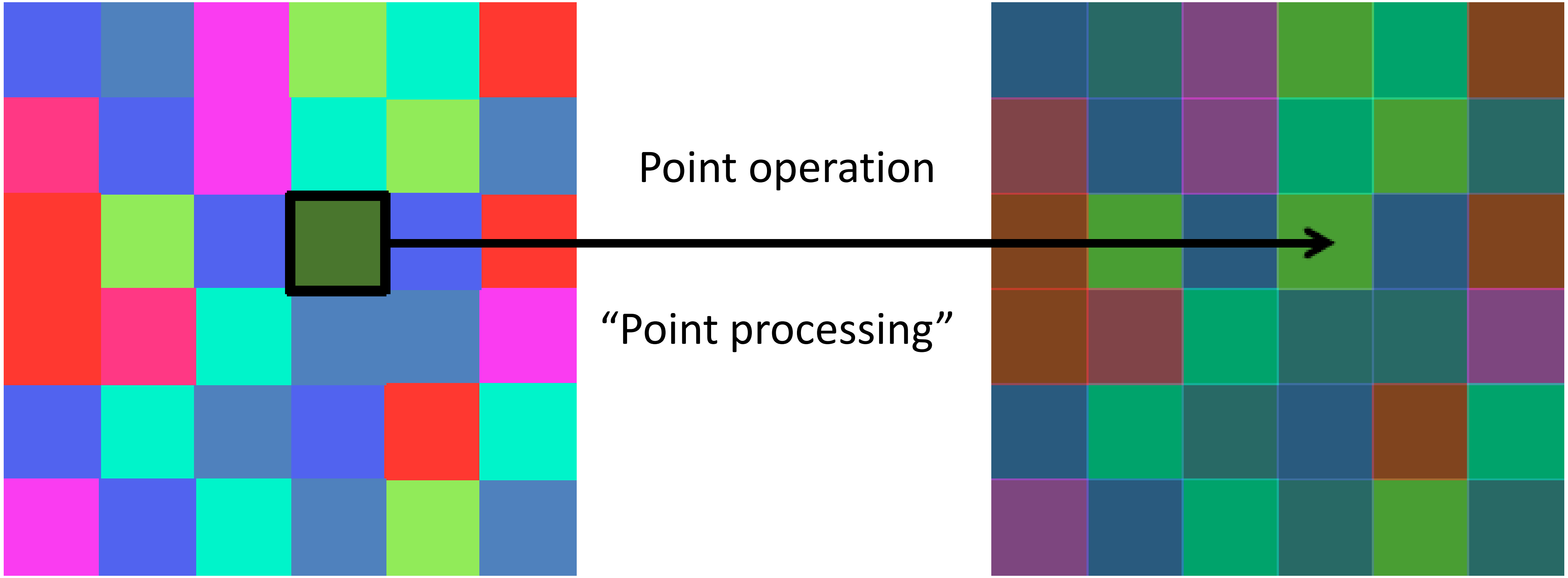
Yeong Jun Koh

yjkoh@cnu.ac.kr

Image Processing
Lecture 02

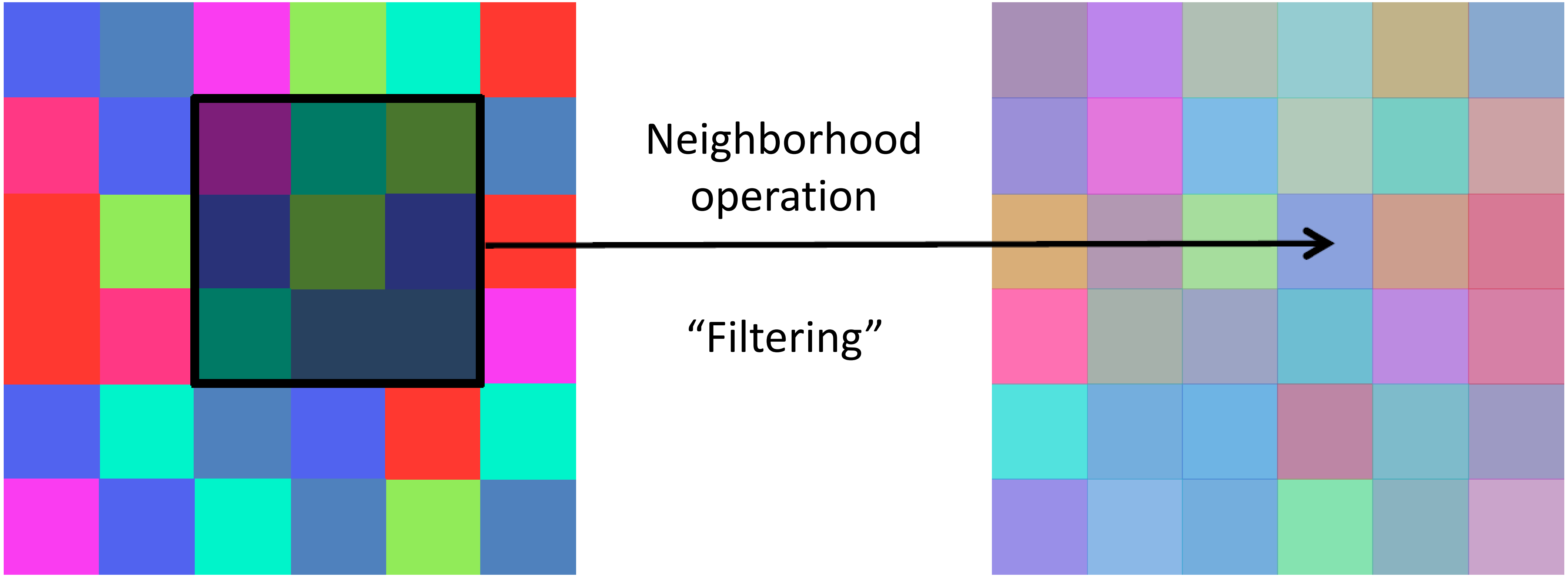
- Point processing
- Histogram equalization

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

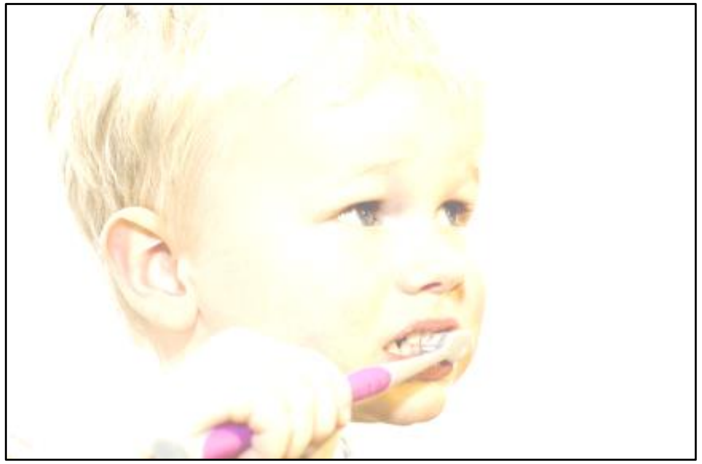
$$\left(\frac{f(\boldsymbol{x})}{255}\right)^{2} \times 255$$

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

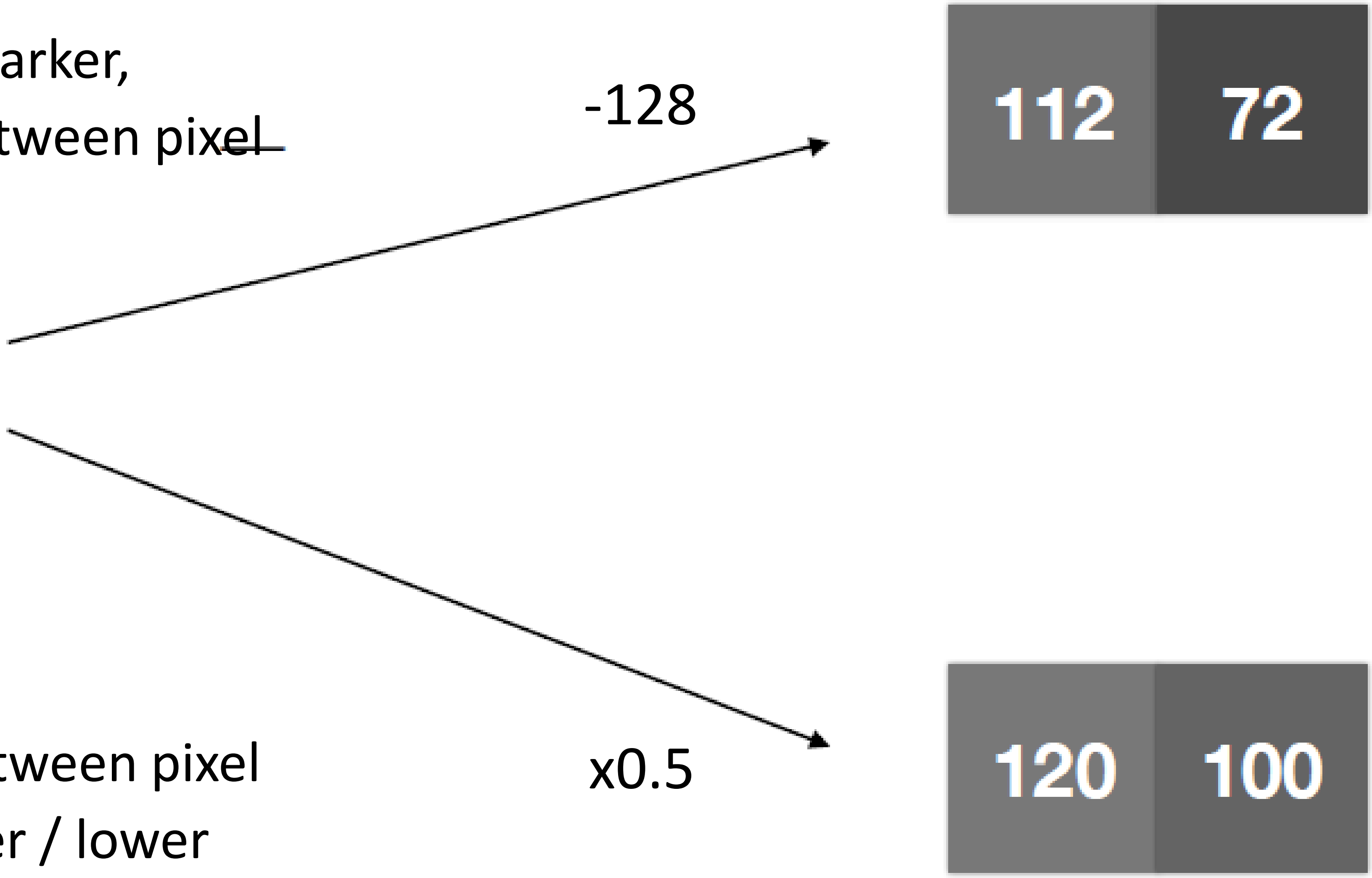
$f(x, y)$

$f(x, y)/2$

$f(x, y) \times 2$

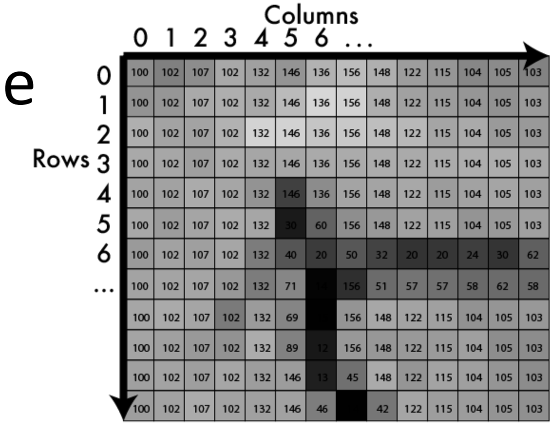
Brightness vs. contrast

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

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

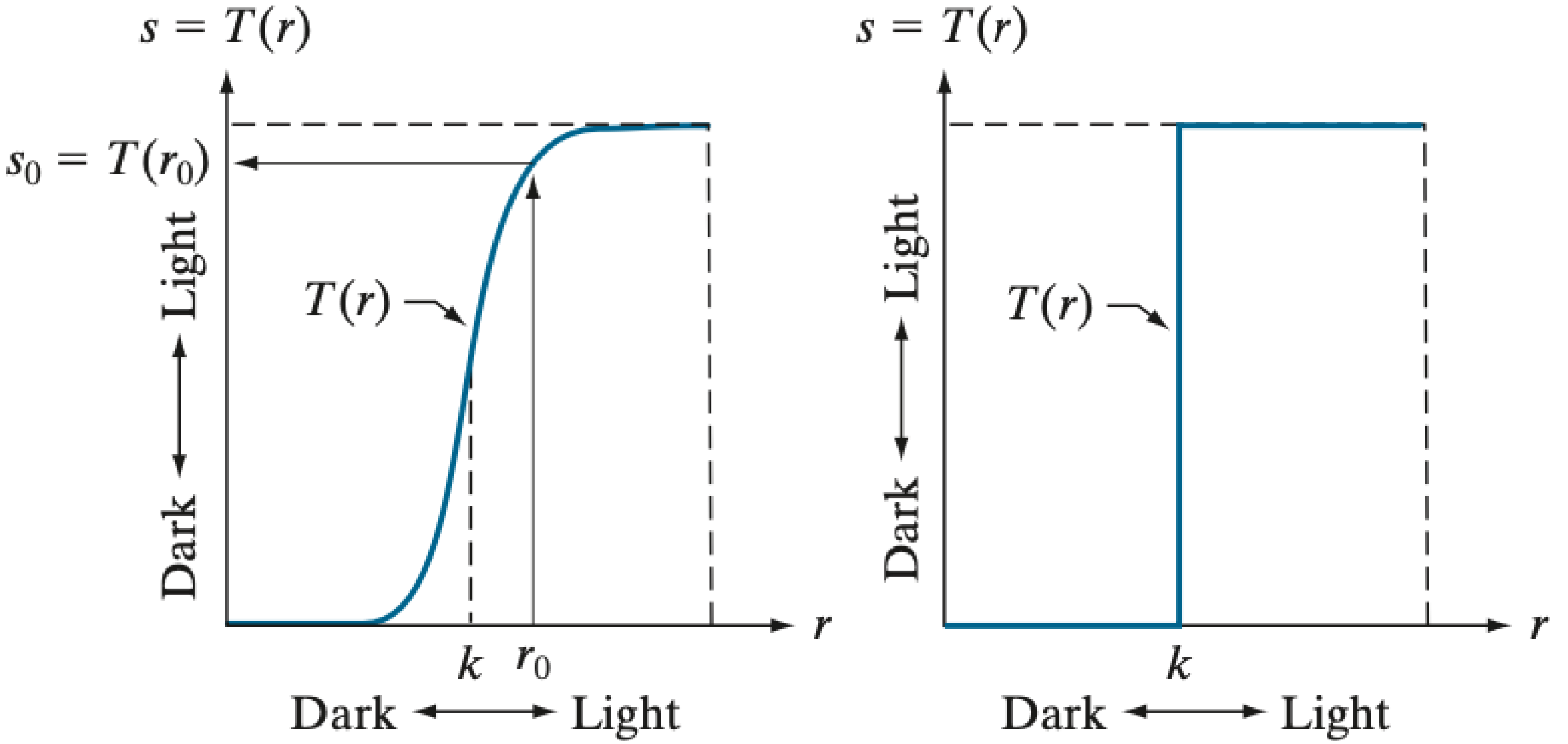
where s and r denote the intensity of g and f at any point (x, y).

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

$$\left.\begin{array}{l} e.g)\ s = r + 1 \\ e.g)\ s = 3r \\ e.g)\ s = r^2 \end{array}\right\} \text{intensity transformation function}$$

a b

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



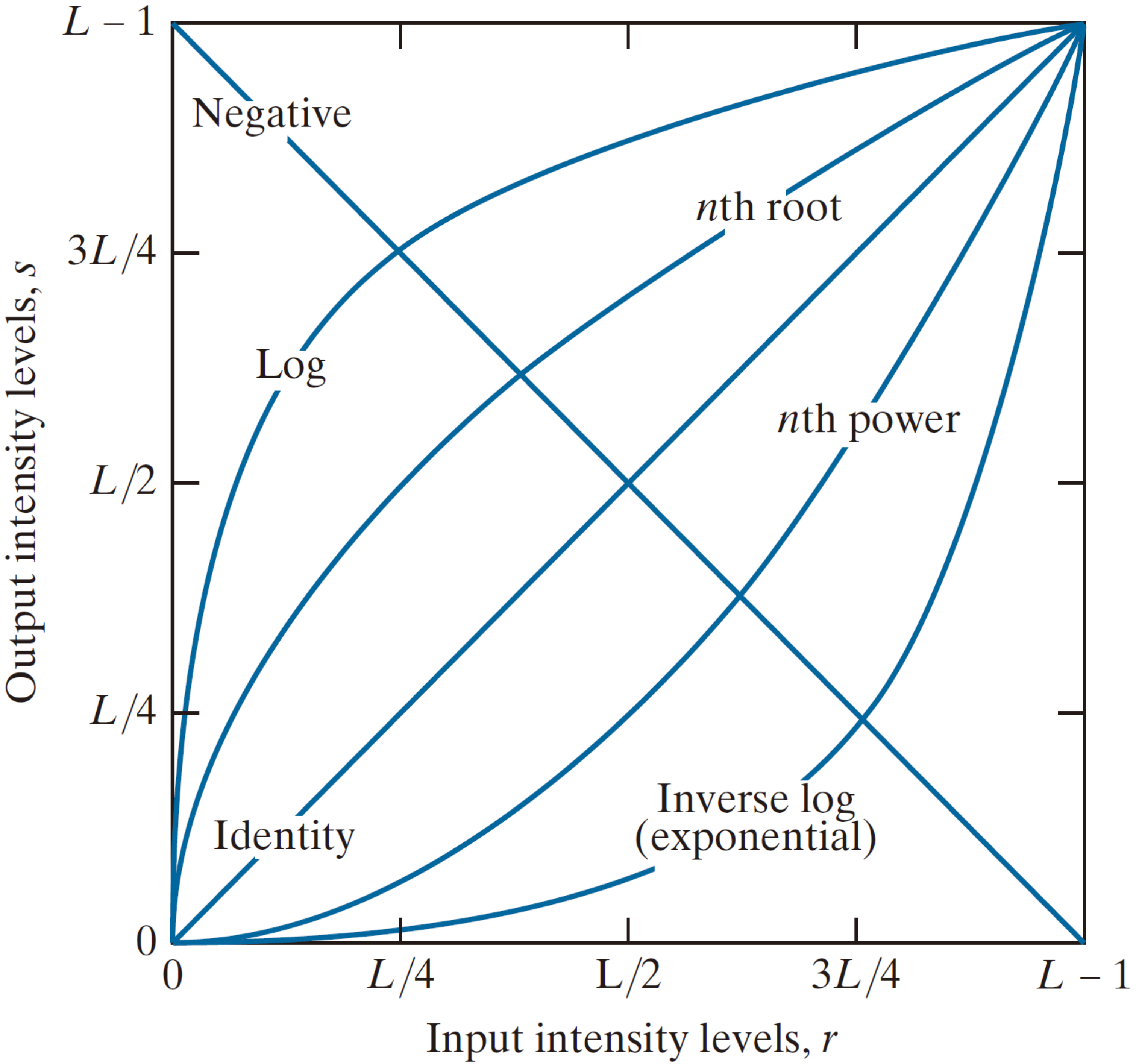


FIGURE 3.3 Some basic intensity transformation functions. Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.

Power-law (gamma) transformation

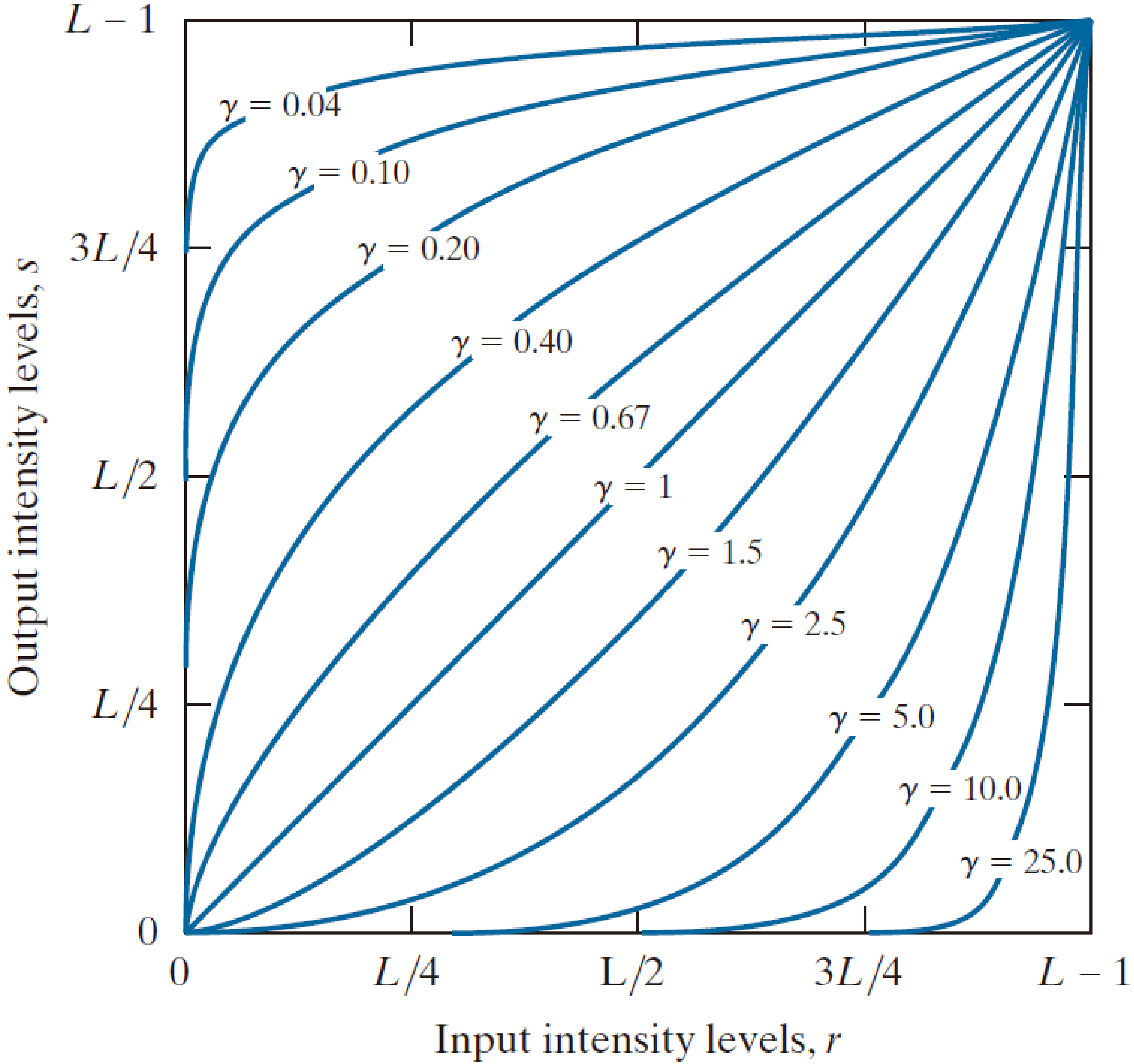
- Power-law (gamma) transformations have the form

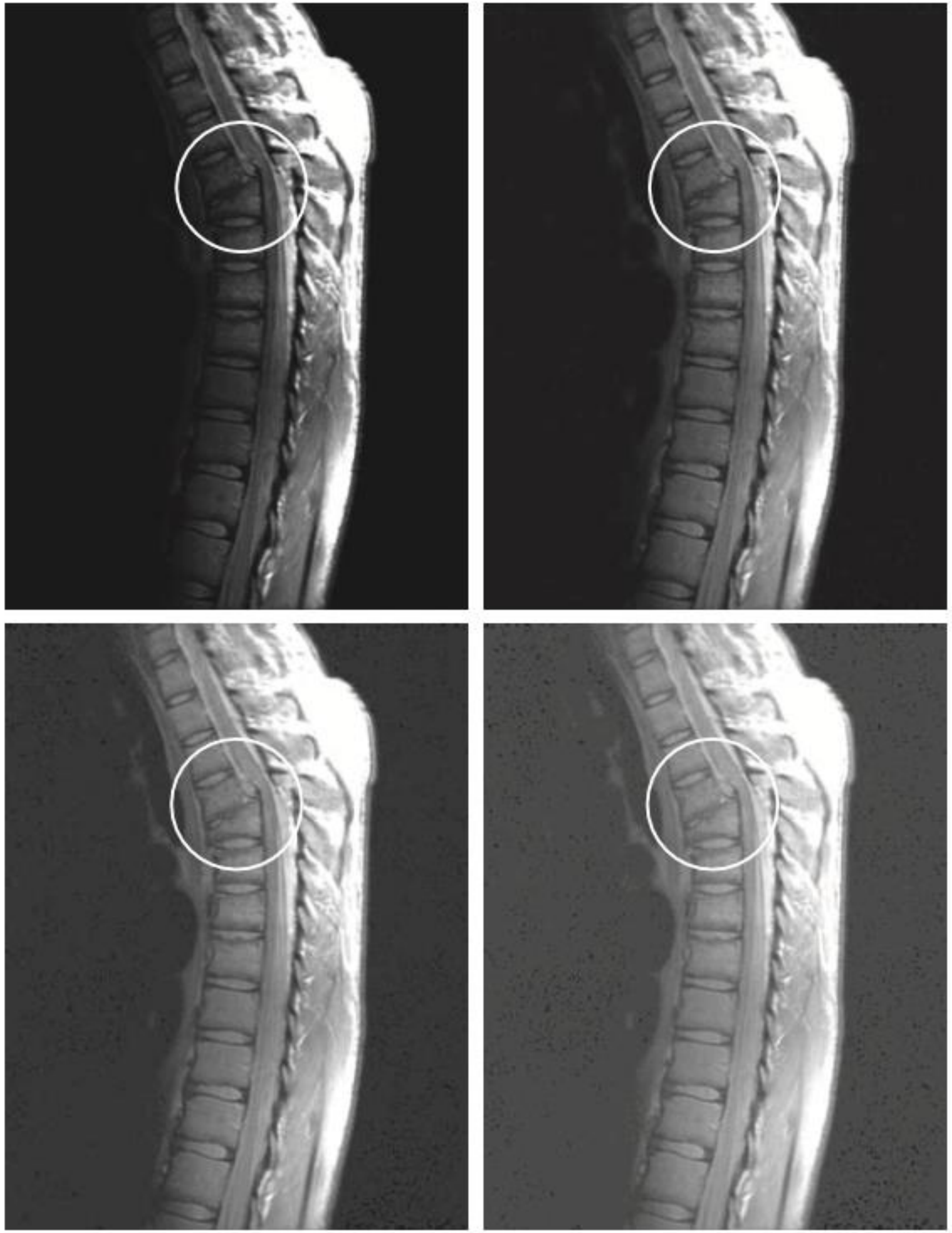
$$s = cr^{\gamma}$$

  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.





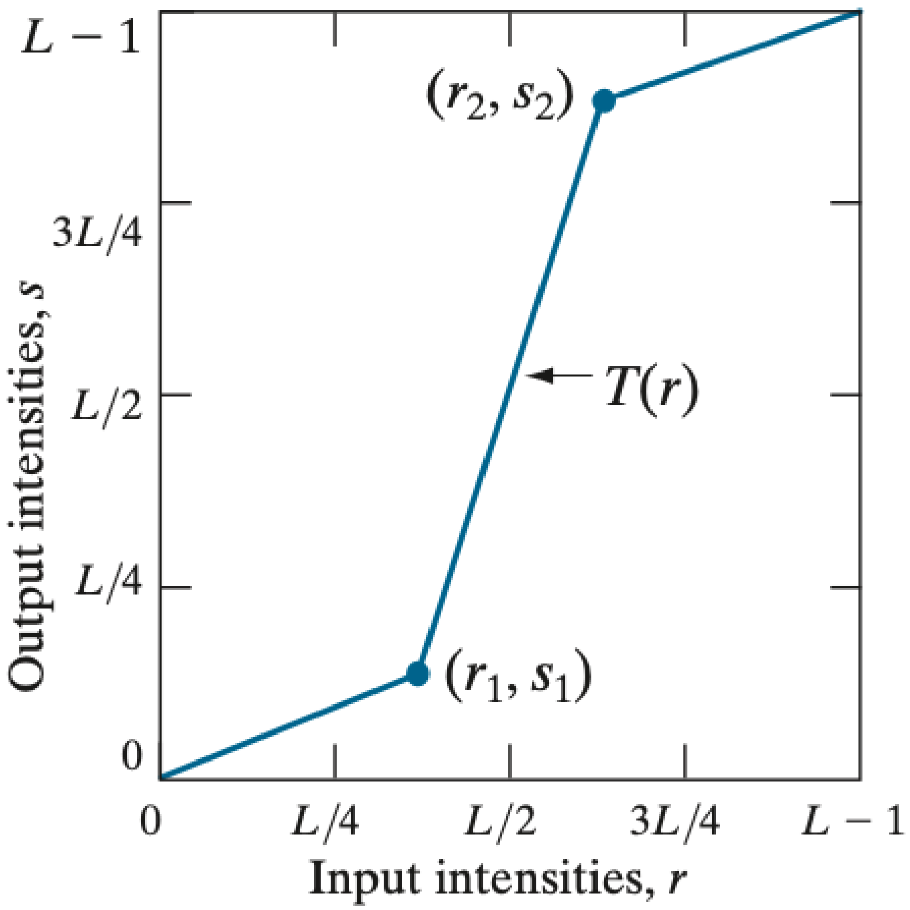
a b
c d

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.)

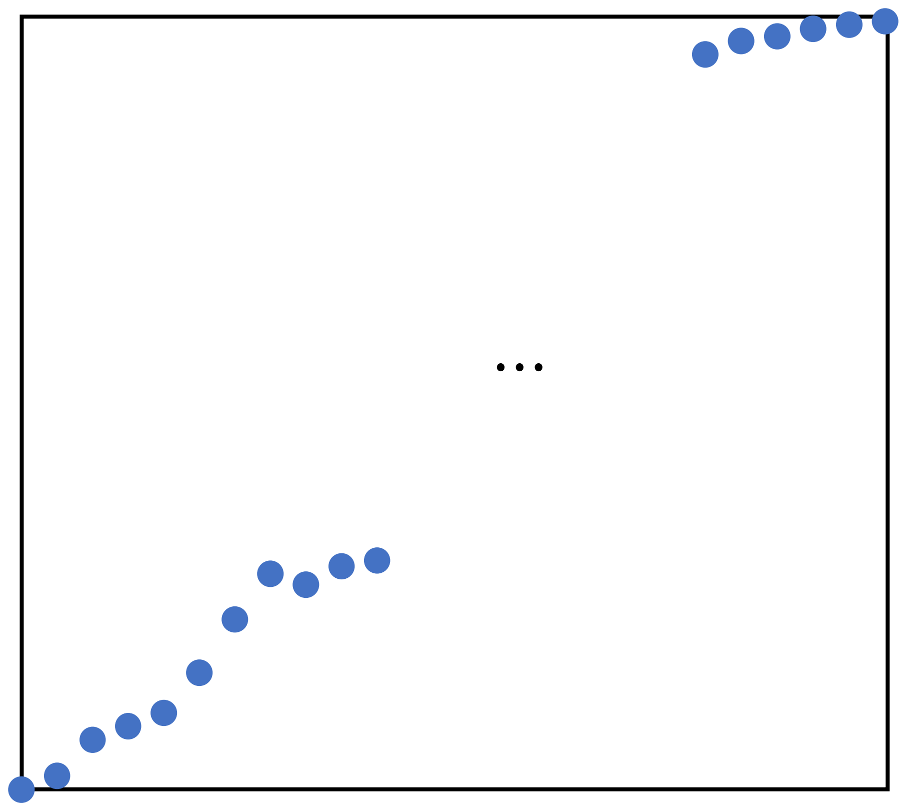
a b
c d

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

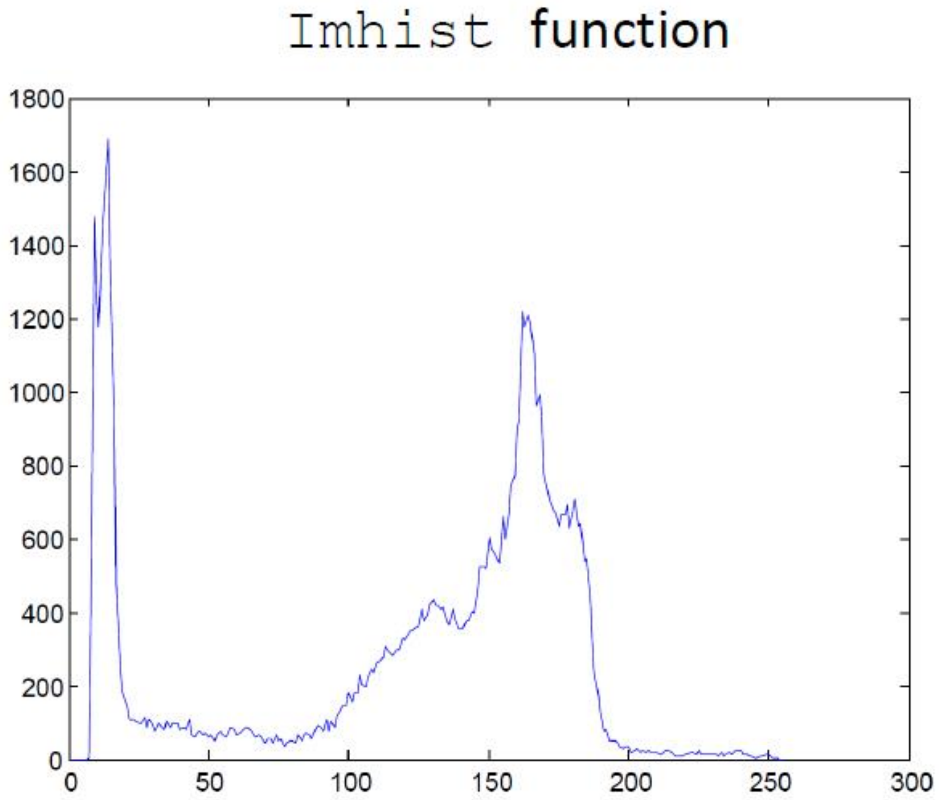
After

Before

After

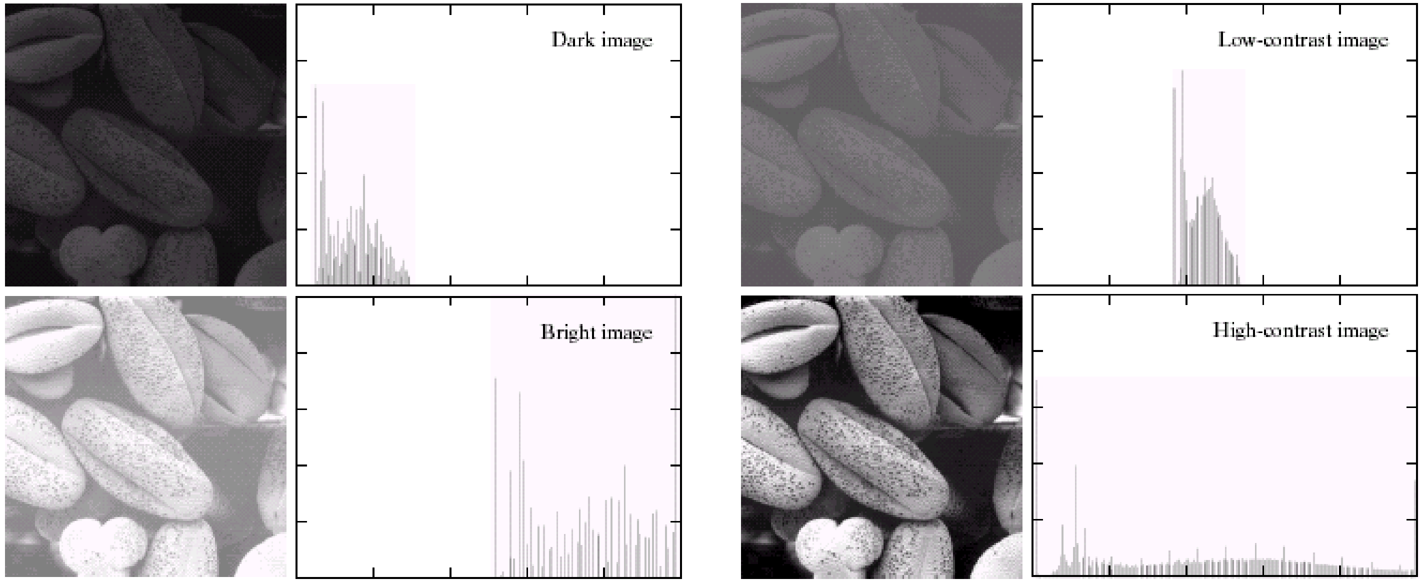
Histogram

- Counts how many times each intensity value (pixel value) occurs in an image



Histogram Example

- The uniform distribution of gray levels is desirable
  - High contrast
  - A great deal of details



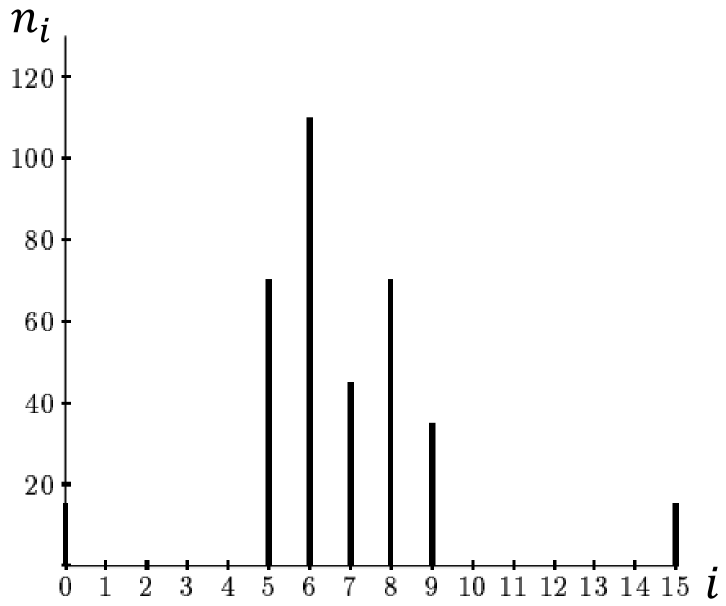
Histogram Stretching

- A table of the numbers of gray values

Grey level i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n_i	15	0	0	0	0	70	110	45	70	35	0	0	0	0	0	15

(Sum $n = 360$)

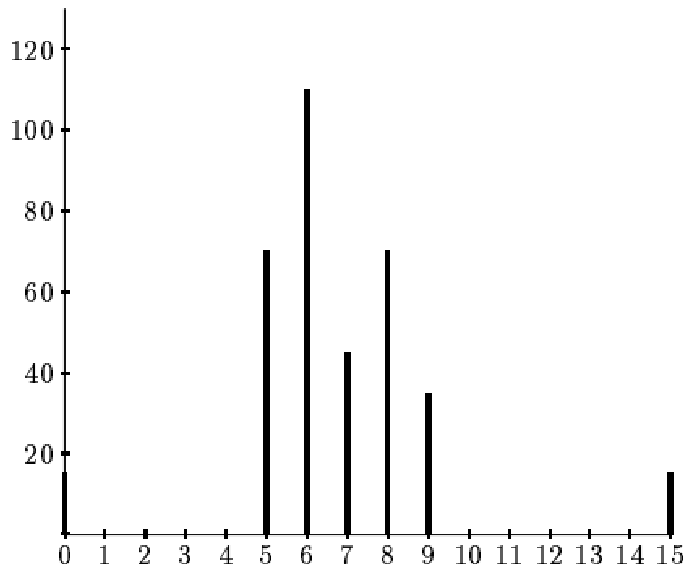
We can stretch out the gray levels
in the center of the range by applying
the piecewise linear function



Histogram Stretching

Histogram

Stretch gray levels 5-9 to gray levels 2-14



Stretching function $j = f(i)$

$$j = \frac{14 - 2}{9 - 5}(i - 5) + 2$$

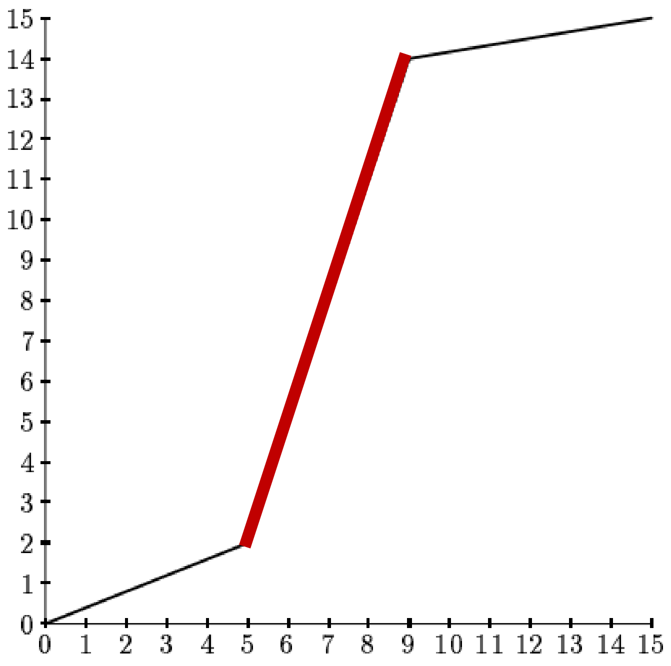


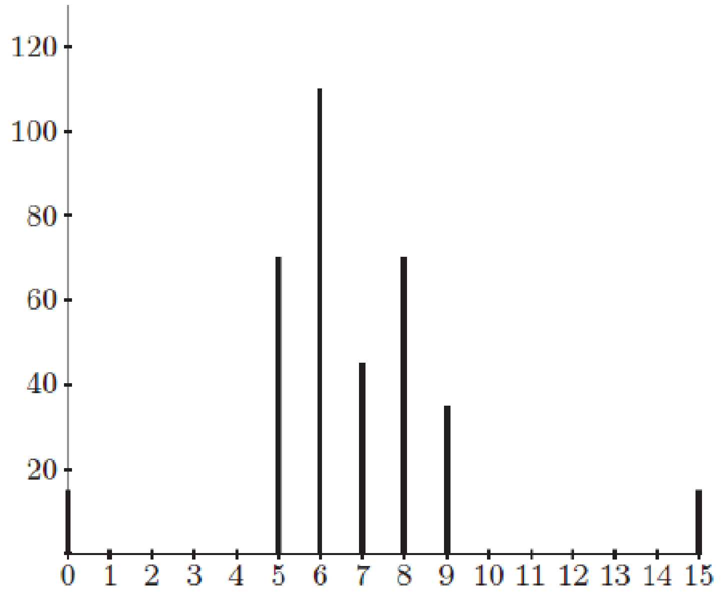
Figure 2.9: A histogram of a poorly contrasted image, and a stretching function

Histogram Stretching

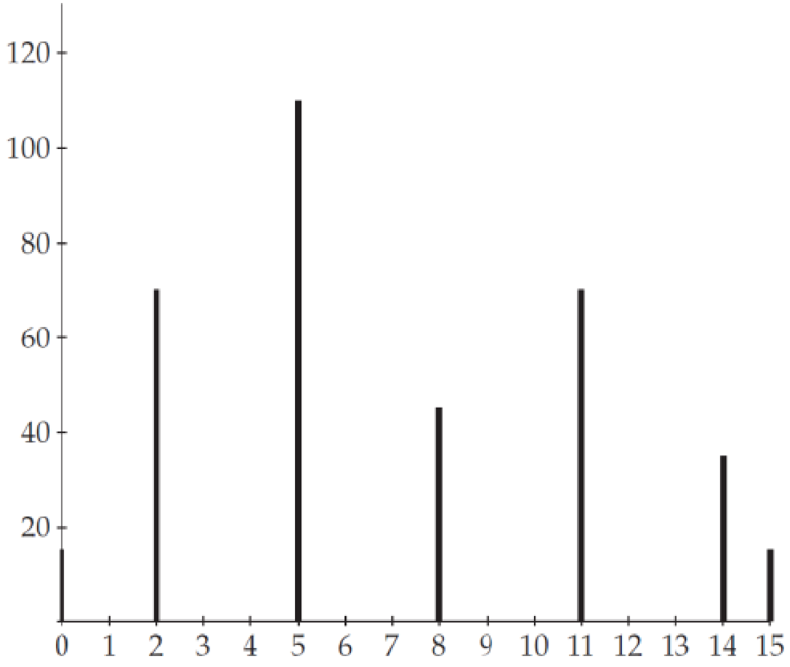
Histogram

Stretching function $j = f(i)$

$$j = \frac{14-2}{9-5}(i-5)+2$$



Before



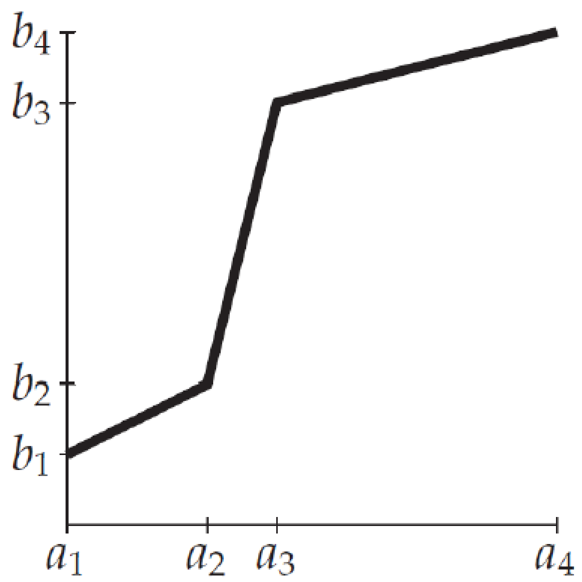
After

Histogram Stretching

- A piecewise linear stretching function

Stretch gray levels a_1- a_{i+1} to gray levels b_1- b_{i+1}

$$y = \frac{b_{i+1} - b_i}{a_{i+1} - a_i}(x - a_i) + b_i.$$



In MATLAB?

- `im`: input, `out`: output

```
pix=find(im >= a(i) & im < a(i+1));
out(pix)=(im(pix)-a(i))*(b(i+1)-b(i))/(a(i+1)-a(i))+b(i)
```

Histogram Stretching

Histogram

Stretch gray levels 5-9 to gray levels 2-14

Stretching function $j = f(i)$

$$j = \frac{14-2}{9-5}(i-5)+2$$

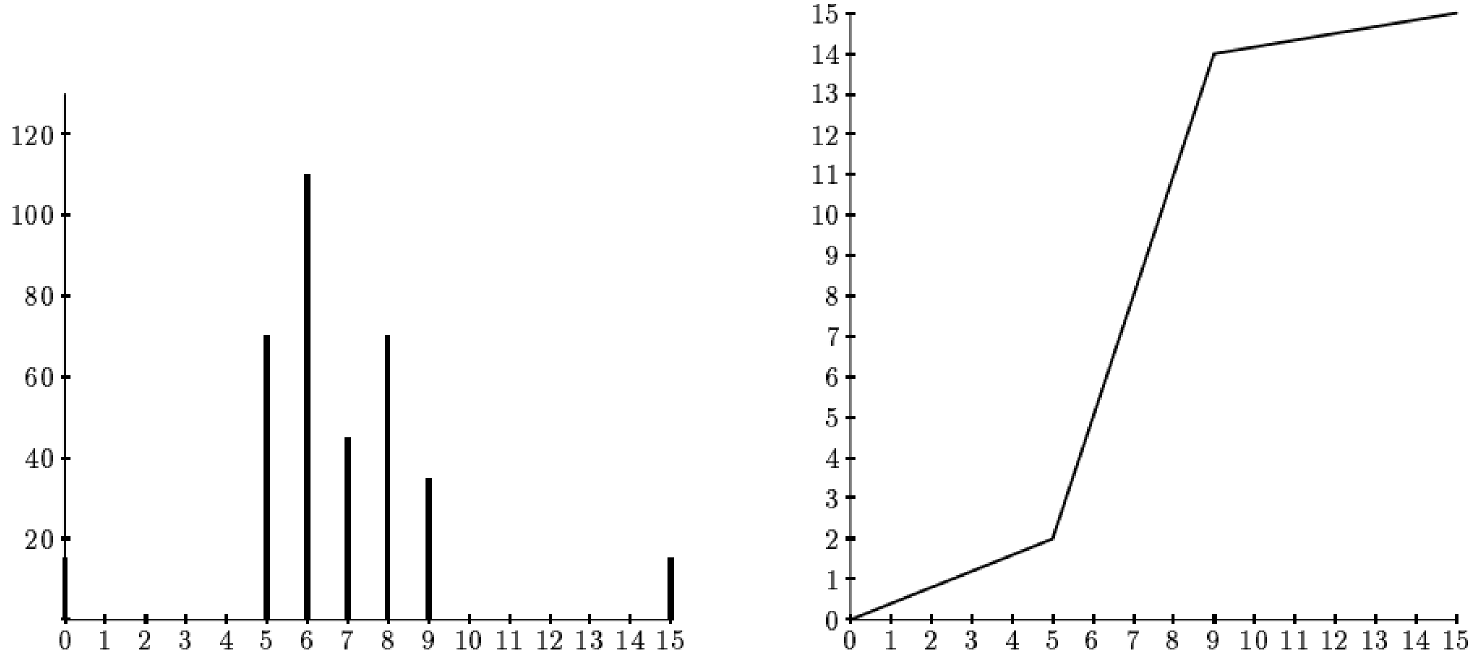
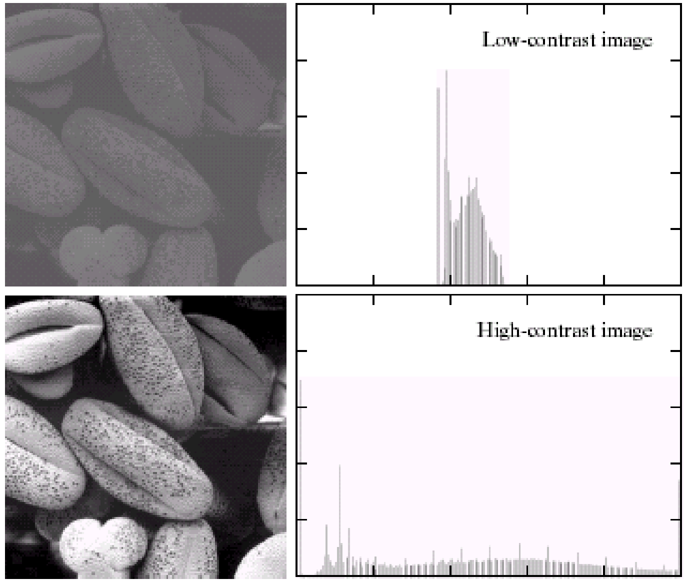


Figure 2.9: A histogram of a poorly contrasted image, and a stretching function

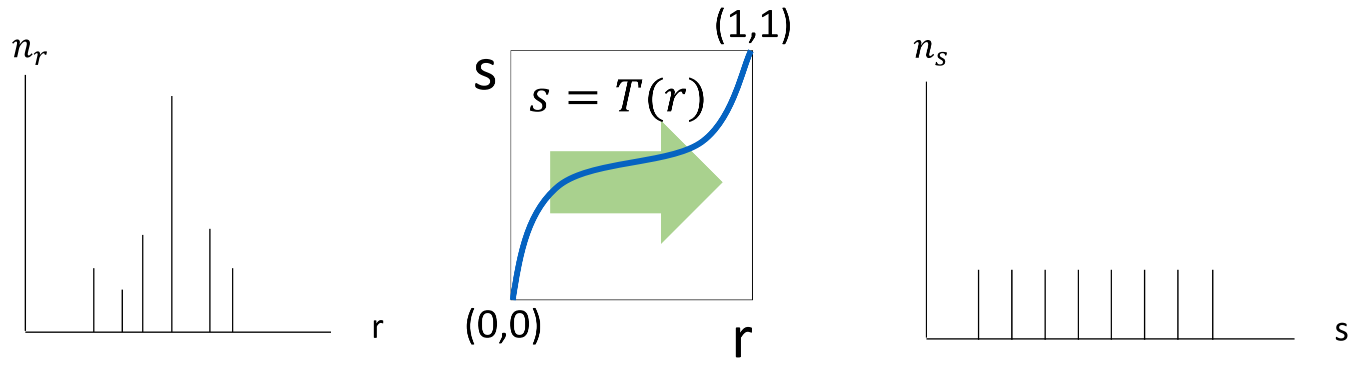
Histogram Equalization

- Goal
  - Enhance an input image to have the gray level distribution, which is as uniform as possible



Histogram Equalization

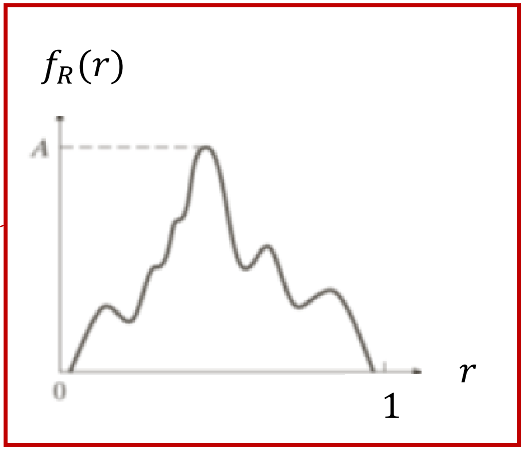
- Ideal case
  - Enhance an input image to have the gray level distribution, which is as uniform as possible

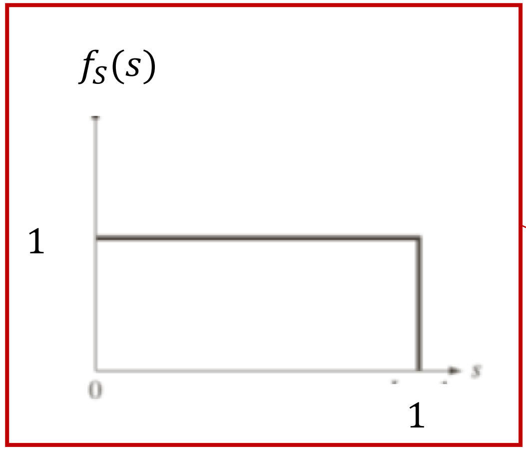


Histogram Equalization

- Continuous case
  - Regard histograms as probability density function (PDF)
  - Normalized r to [0, 1]

which we have

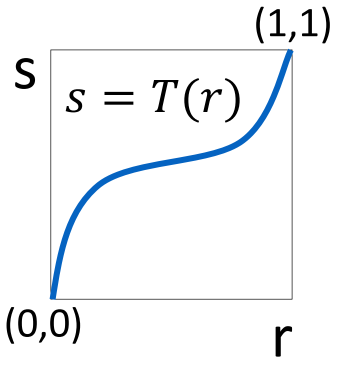




which we want

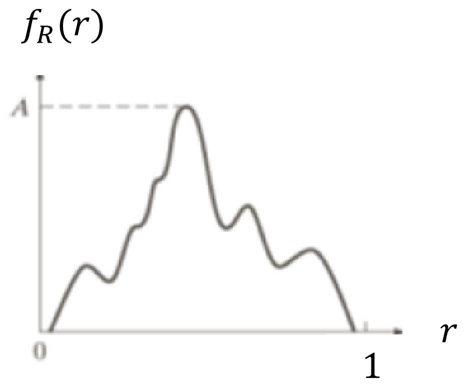
Histogram Equalization

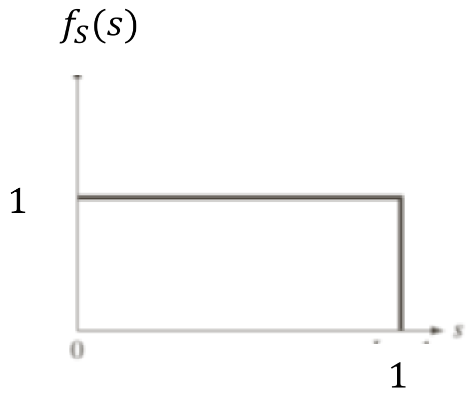
- Continuous case
  - $f_S(s) = 1$
  - $f_S(s)ds = f_R(r)dr$
    - Monotonic increasing function: $s = T(r)$



$$f_R(r) = f_S(s)\frac{ds}{dr} = f_S(s)\frac{dT(r)}{dr} = T'(r)$$

$$T(r) = \int_{-\infty}^{r} f_R(t)dt = \int_{0}^{r} f_R(t)dt$$





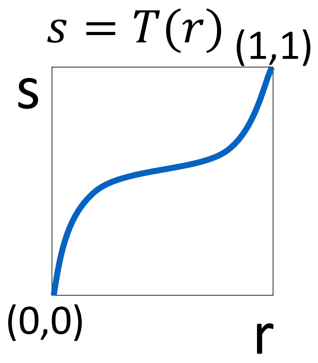
Histogram Equalization

- Continuous case

$$s = T(r) = CDF(r) = \int_0^r f_R(t)dt$$

- Discrete approximation

$$s_k = T(r_k) = \sum_{j=0}^{k} f_R(r_j) = \sum_{j=0}^{k} \frac{n_j}{N}$$



input gray level k	0	1	2	3	4	5	6	7
normalized input r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
histogram n_k	1	3	2	7	8	3	0	1
normalized histogram n_k/N	1/25	3/25	2/25	7/25	8/25	3/25	0	1/25
normalized output s_k	1/25	4/25	6/25	13/25	21/25	24/25	24/25	1
denormalized output $o_k = s_k \times 7$	7/25	28/25	42/25	91/25	147/25	168/25	168/25	7
output gray level floor(o_k)	0	1	1	3	5	6	6	7
m	0	1	2	3	4	5	6	7
output histogram n_m	1	5	0	7	0	8	3	1

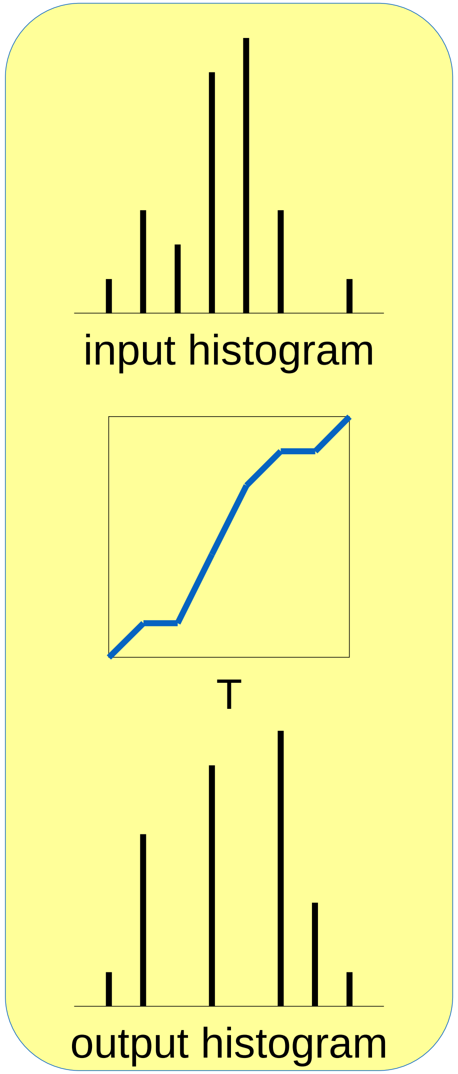
Histogram Equalization

- Discrete approximation

input gray level k	0	1	2	3	4	5	6	7
normalized input r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
histogram n_k	1	3	2	7	8	3	0	1
normalized histogram n_k/N	1/25	3/25	2/25	7/25	8/25	3/25	0	1/25
normalized output s_k	1/25	4/25	6/25	13/25	21/25	24/25	24/25	1
denormalized output $o_k = s_k \times 7$	7/25	28/25	42/25	91/25	147/25	168/25	168/25	7
output gray level floor(o_k)	0	1	1	3	5	6	6	7
m	0	1	2	3	4	5	6	7
output histogram n_m	1	5	0	7	0	8	3	1

Histogram Equalization

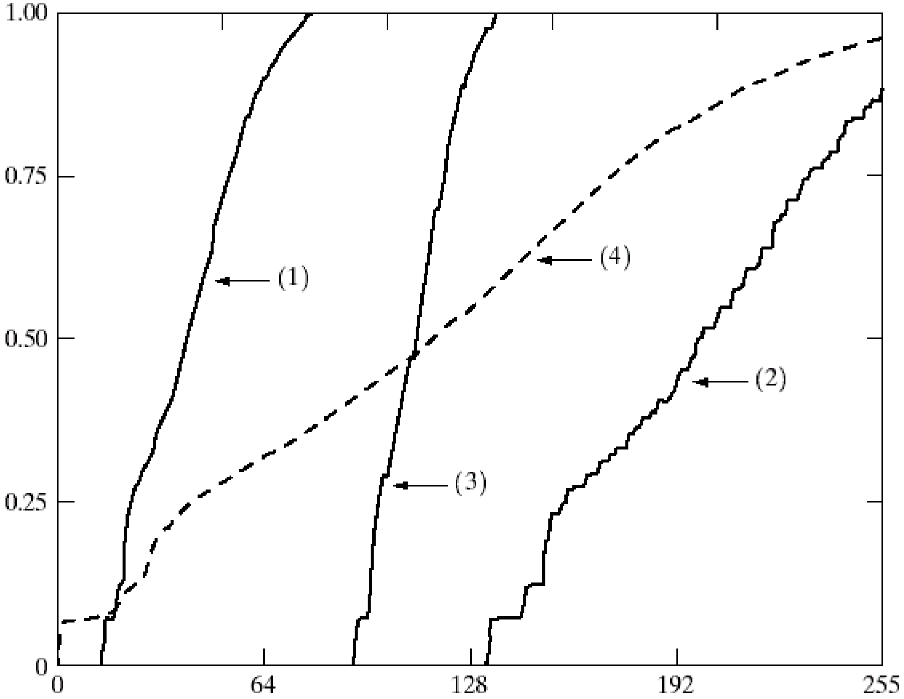
input gray level	0	1	2	3	4	5	6	7
output gray level	0	1	1	3	5	6	6	7
input histogram	1	3	2	7	8	3	0	1
output histogram	1	5	0	7	0	8	3	1



- Does not provide the exactly uniform output
  - Discrete approximation

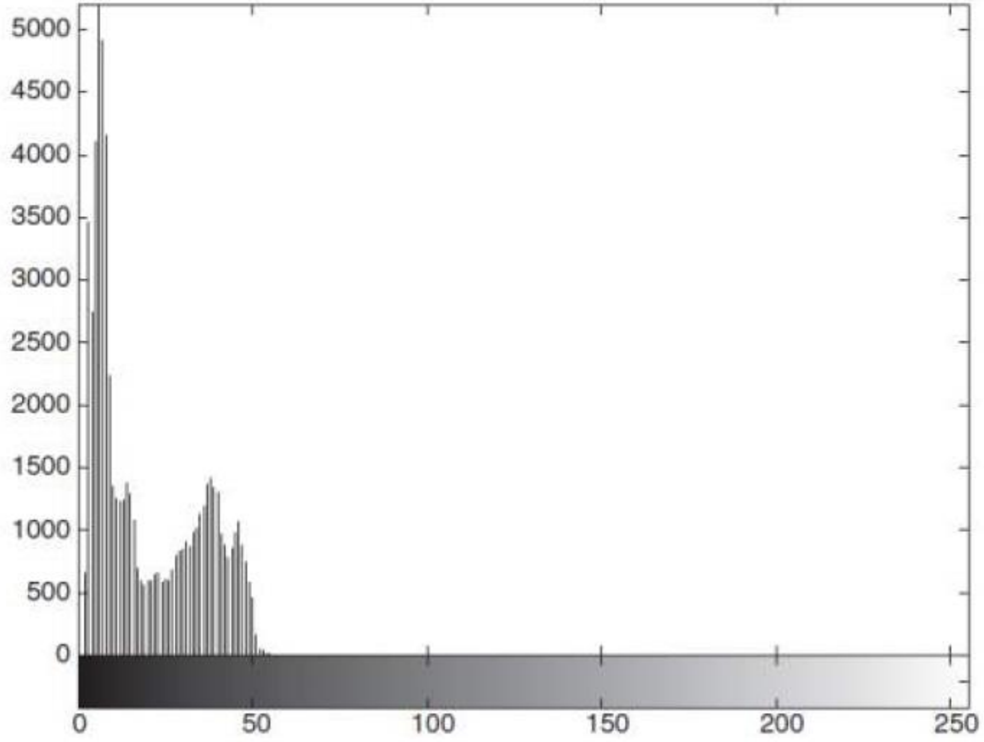
- But, spread the histogram automatically

Histogram Equalization



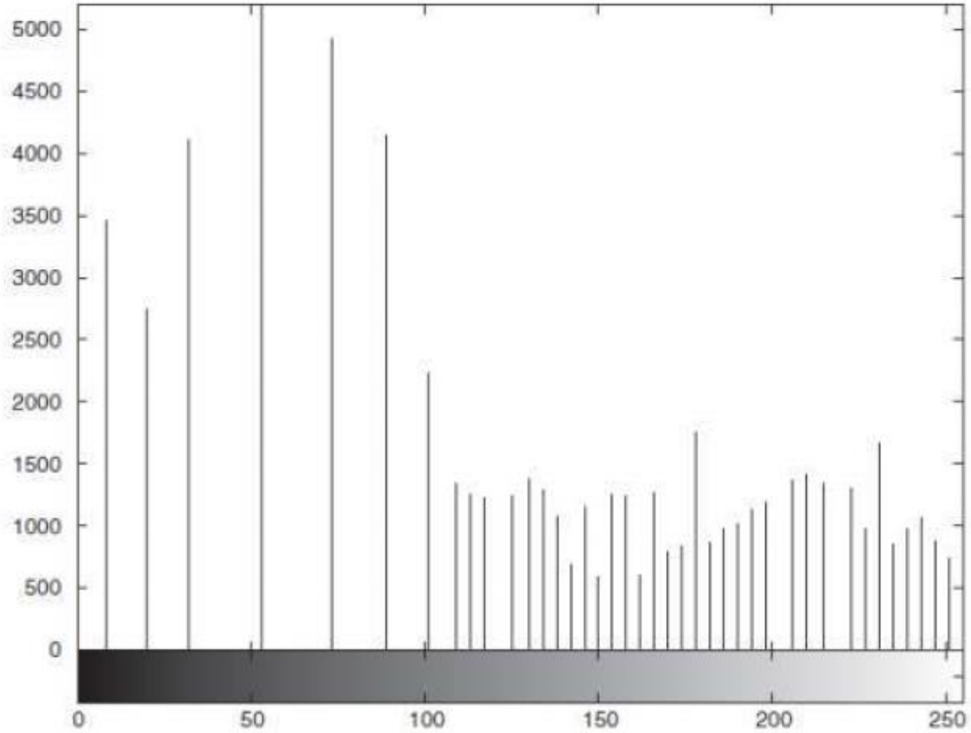
Histogram Equalization

```
>> en=imread('engineer.tif');
>> e=imdivide(en,4);
>> imshow(e),figure,imhist(e),axis tight
```



Histogram Equalization

```
>> eh=histeq(e);
>> imshow(eh),figure,imhist(eh),axis tight
```



Thank You