Image Processing

Pixel-wise Operation

Yeong Jun Koh
Department of Computer Science & Engineering
Chungnam National University

Objectives

- Pixel-wise operation
- Image histogram
- Histogram stretch
- Histogram equalization

Arithmetic Operation

- Pixel intensity ← Pixel intensity
- $O = f(I)$
  - I: input intensity
  - O: output intensity
  - $f(I) = I + 128$ or $f(I) = I - 128$

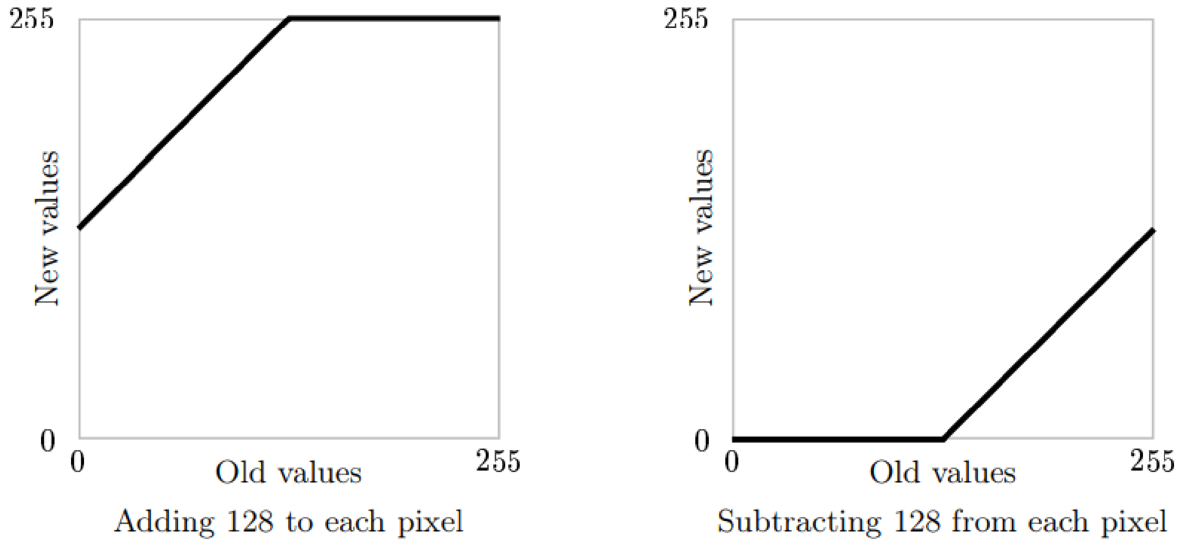
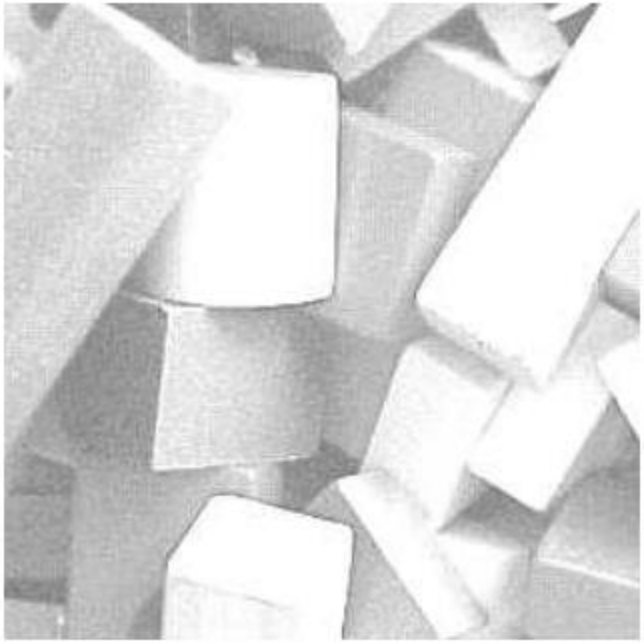
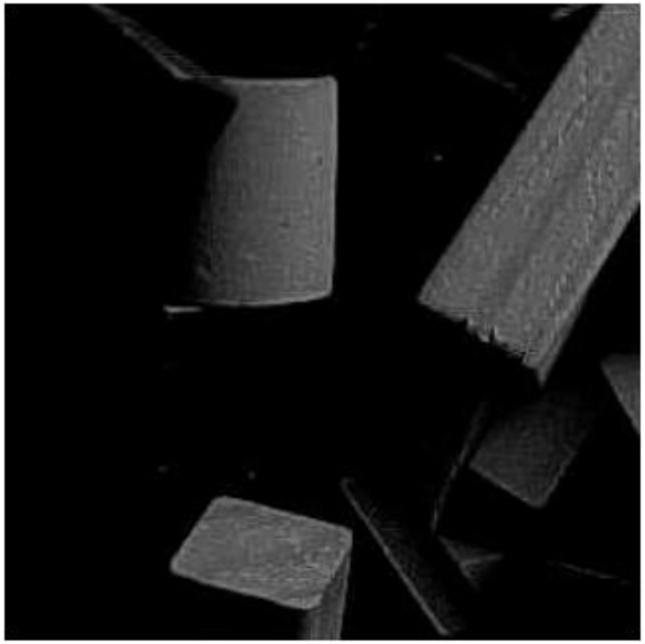


Figure 2.2: Adding and subtracting a constant

Arithmetic Operation



b1: Adding 128



b2: Subtracting 128

Arithmetic Operation

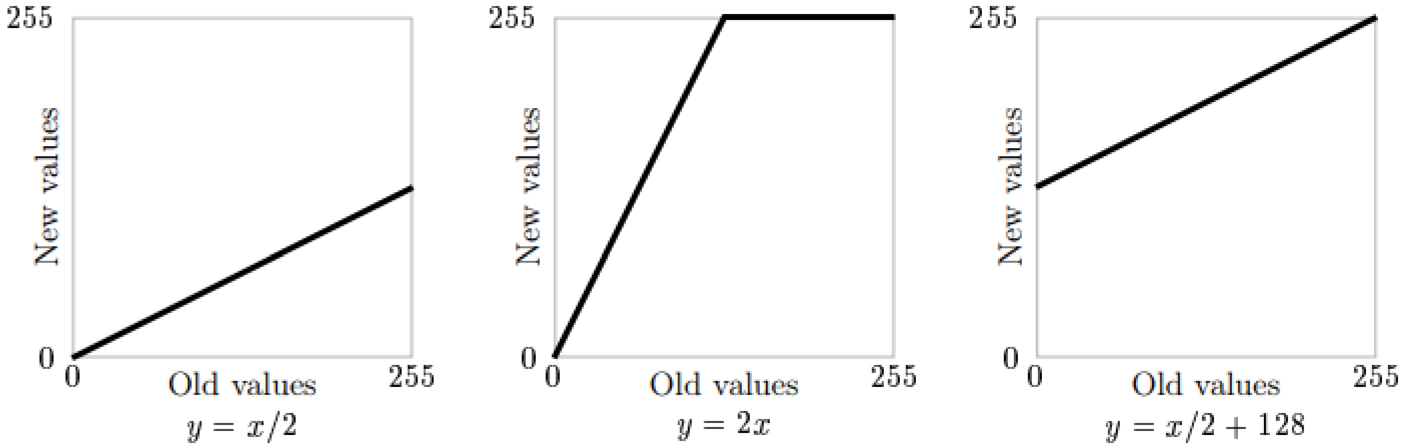


Figure 2.4: Using multiplication and division

b3: $y = x/2$ b4: $y = 2x$ b5: $y = x/2 + 128$

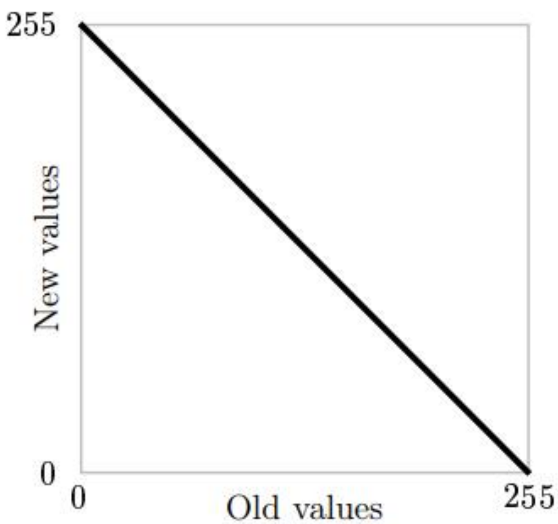
Figure 2.5: Arithmetic operations on an image: multiplication and division

Arithmetic Operation

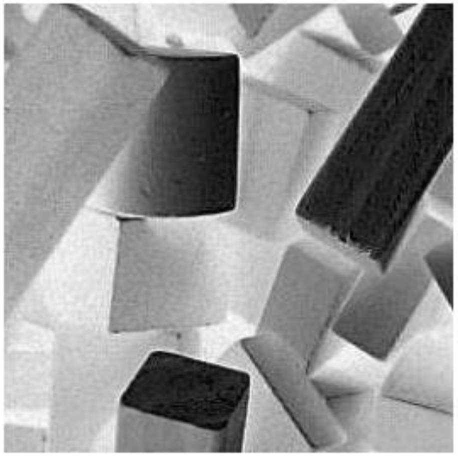
- Complements
  - type double (0.0~1.0)

    $1 - x$
  - type uint8 (0~255)

    $255 - x$



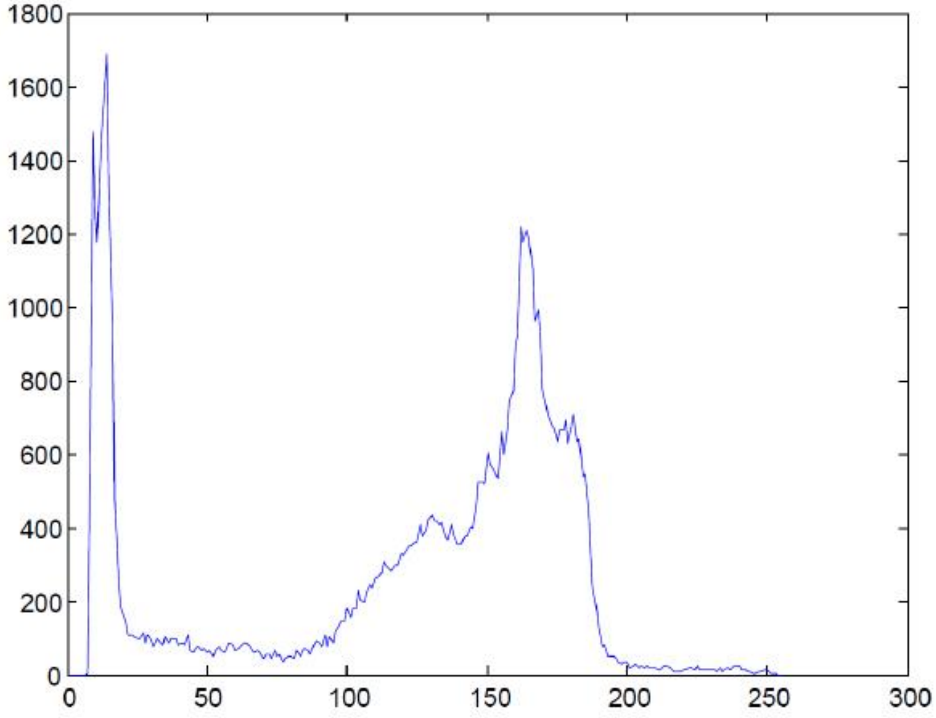
$y = 255 - x$



Histogram

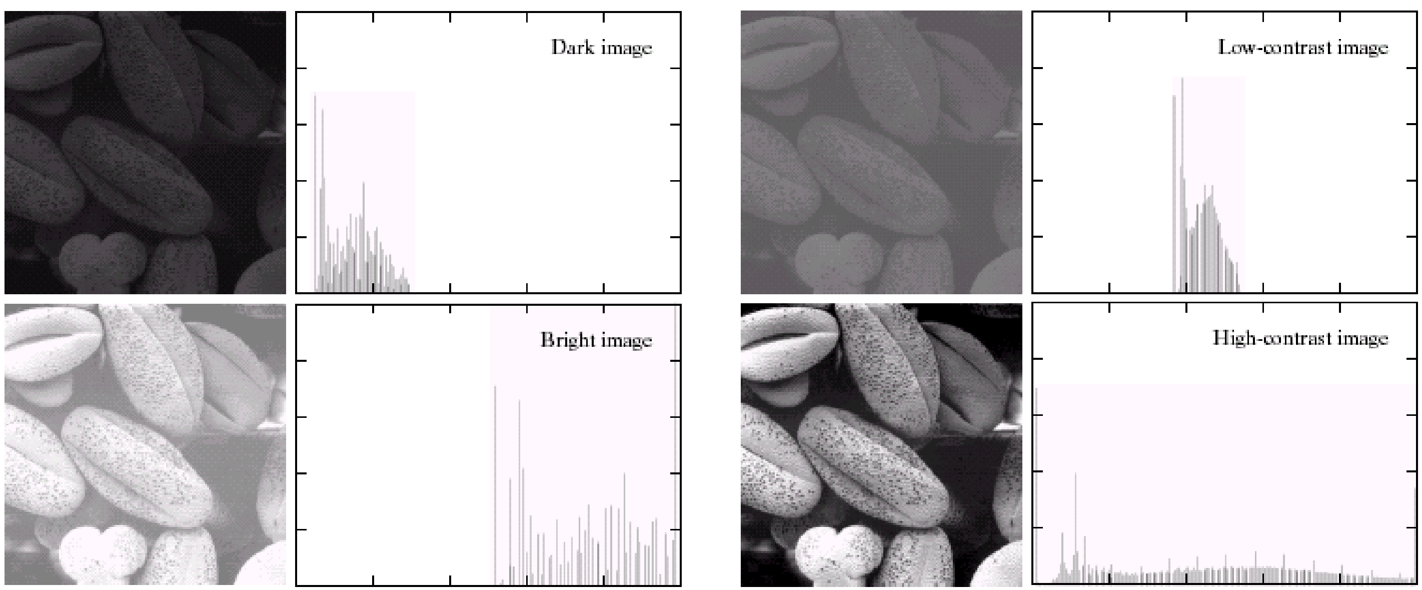
- Counts how many times each intensity value (pixel value) occurs in an image

Imhist function



Histogram Example

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



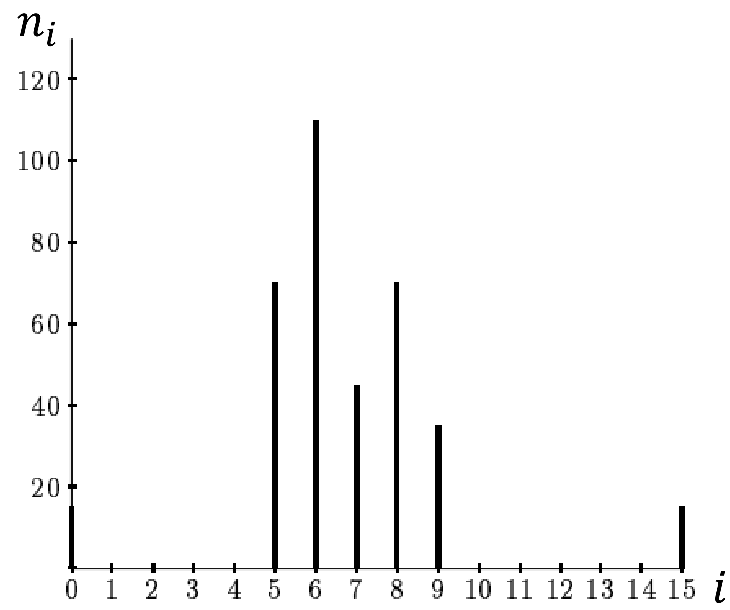
Histogram Stretching

- 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 $j = f(i)$

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

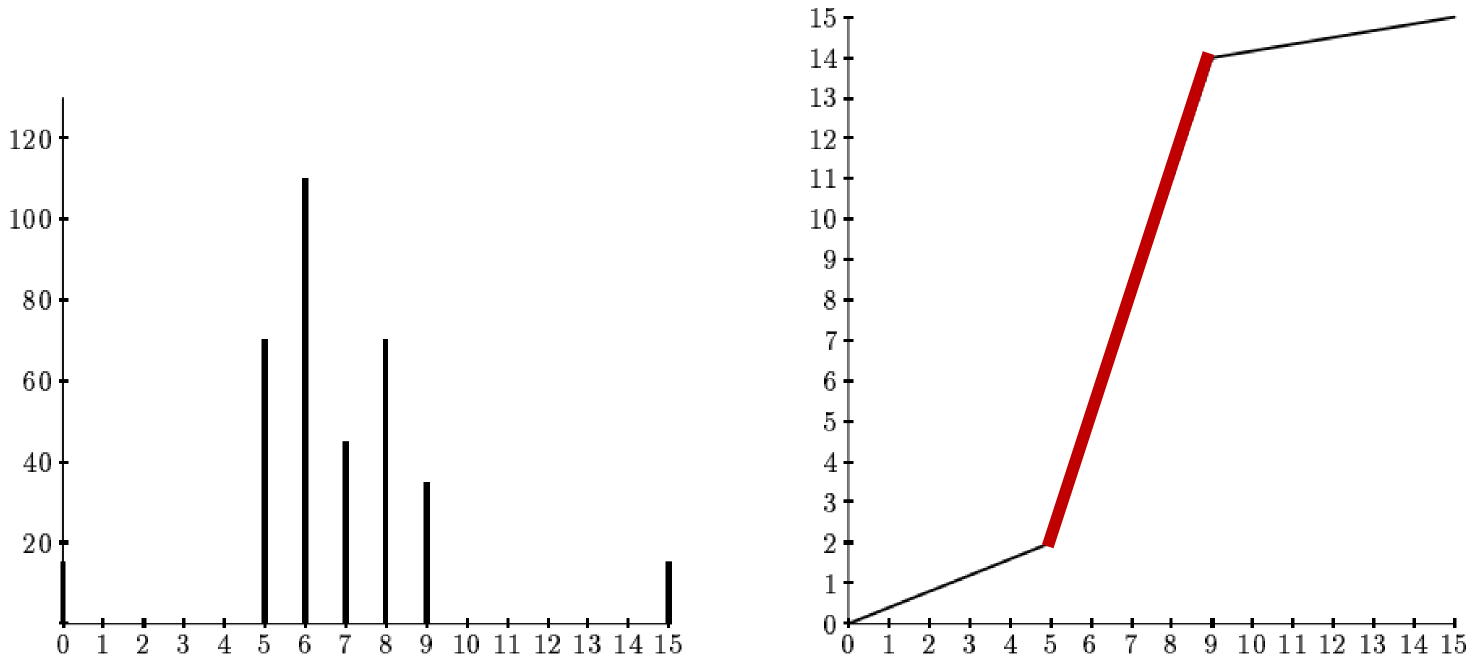


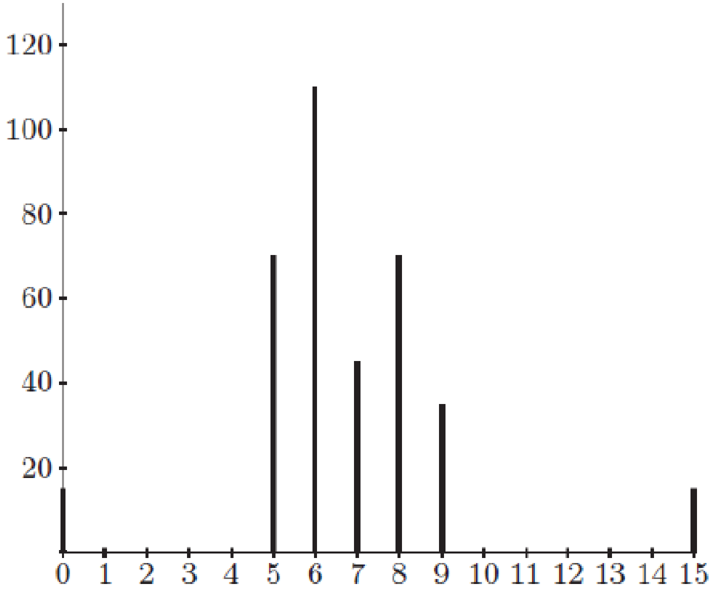
Figure 2.9: A histogram of a poorly contrasted image, and a stretching function

Histogram Stretching

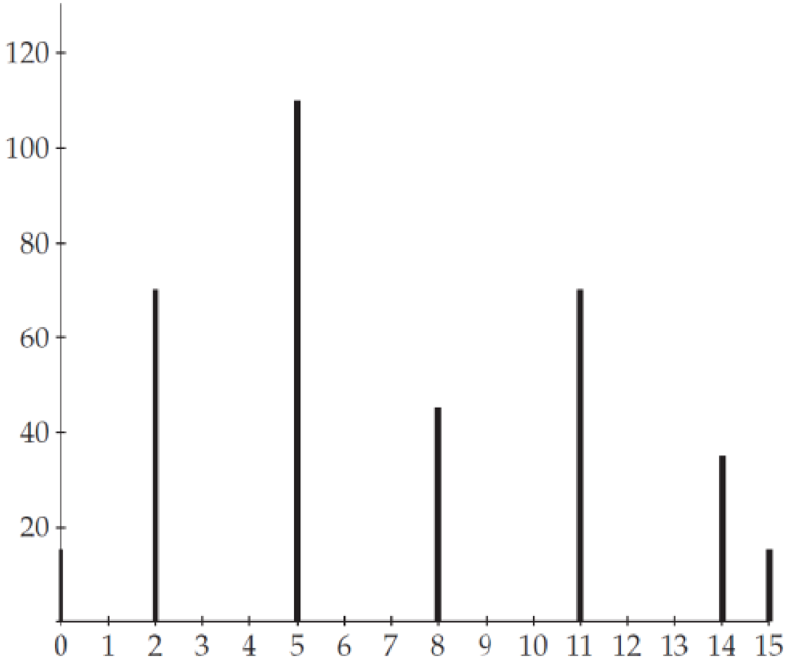
Histogram

Stretching function $j = f(i)$

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



Before



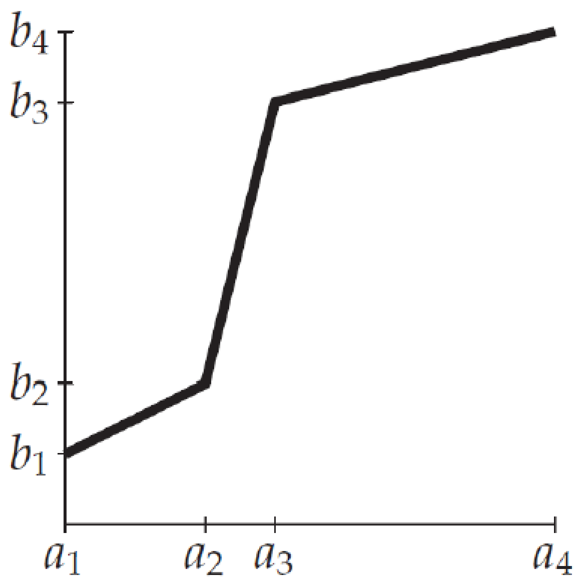
After

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)
```

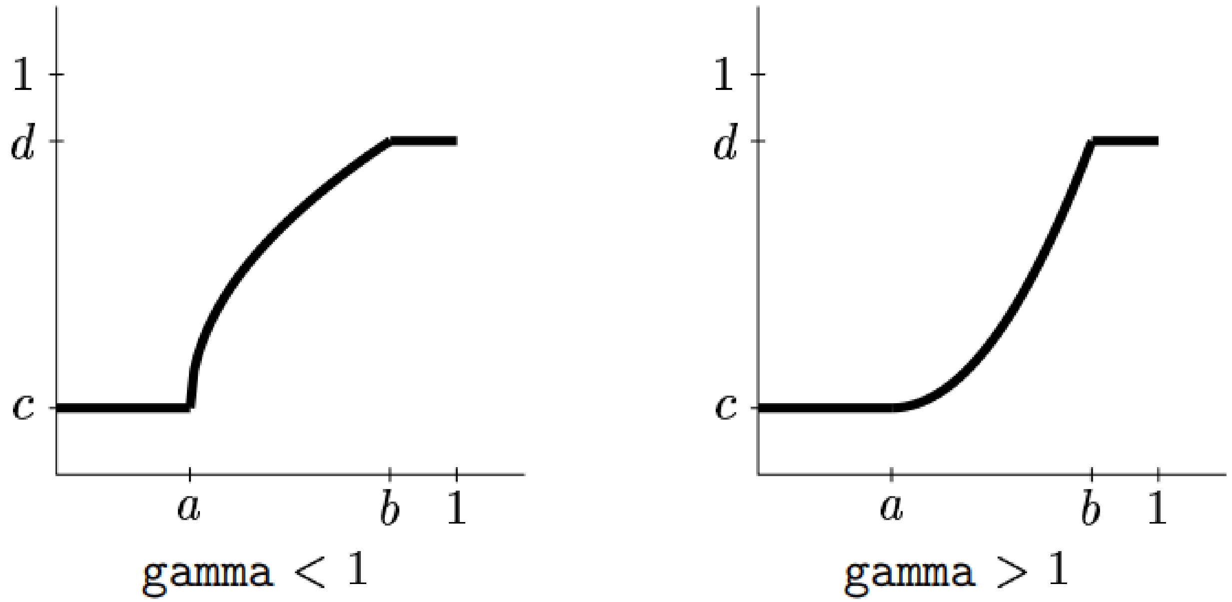
Histogram Stretching

- Gamma correction

$$y = \left(\frac{x-a}{b-a}\right)^{\gamma}(d-c)+c$$

$\gamma < 1$: expand bright levels and compress dark levels

$\gamma > 1$: expand dark levels and compress bright levels



Histogram Stretching

- $\gamma = 0.5$

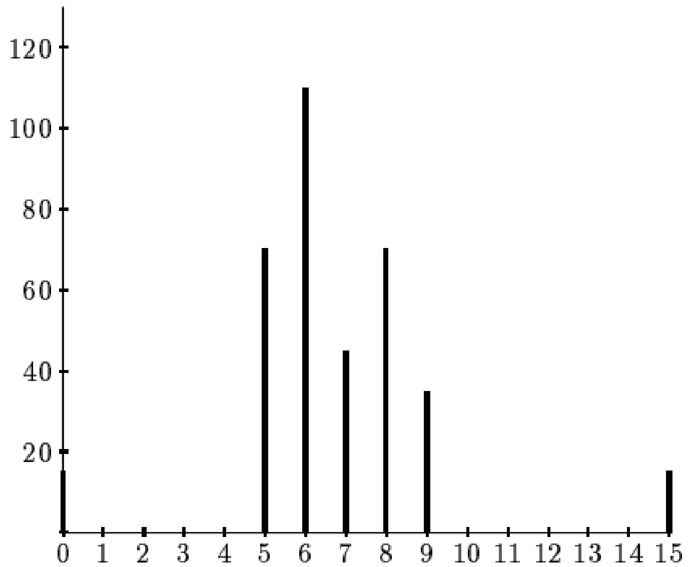
Histogram Stretching

Histogram

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

Stretching function $j = f(i)$

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



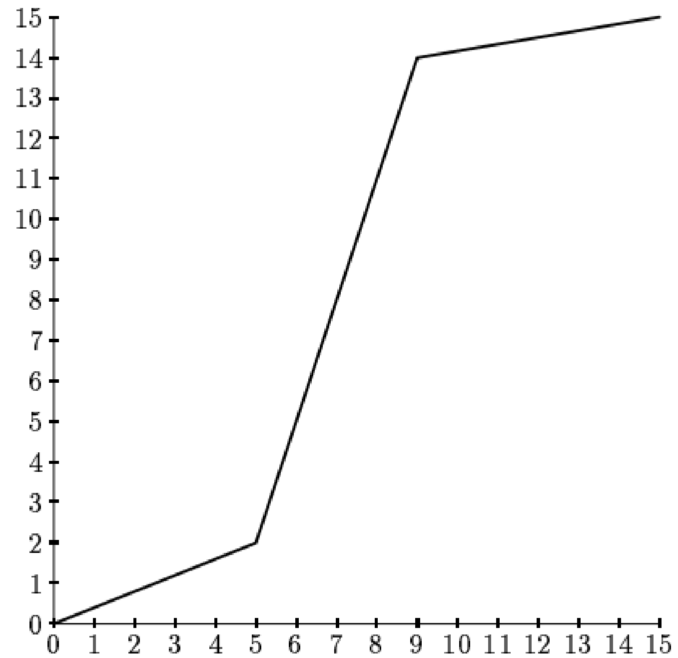
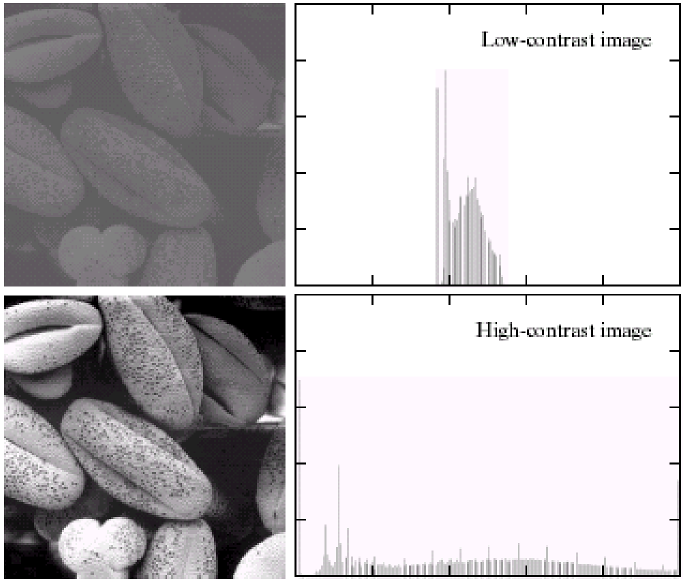


Figure 2.9: A histogram of a poorly contrasted image, and a stretching function

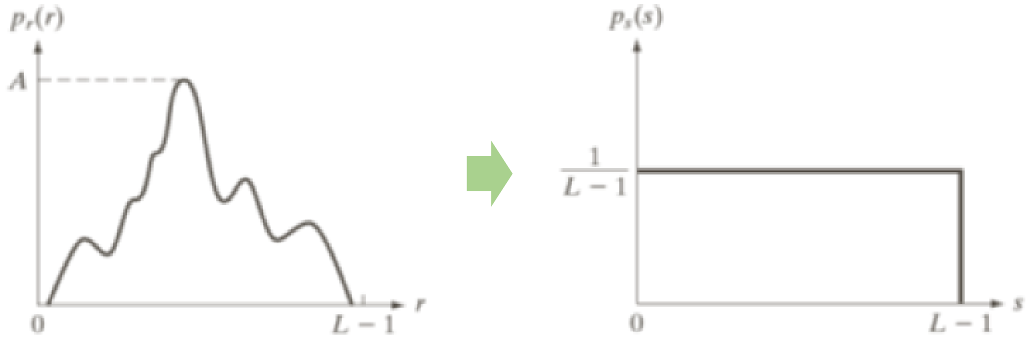
Histogram Equalization

- A fully automatic procedure
- Suppose our image has L different gray levels, $0,1,2,\ldots,L-1$, and gray level k occurs n_k times in image
- Ideal case
  - All intensity values has an uniform probability distribution function (PDF)



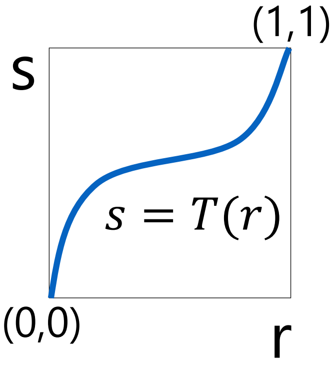
Histogram Equalization

- A fully automatic procedure
- Suppose our image has L different gray levels, $0,1,2,\ldots,L-1$, and gray level k occurs n_k times in image
- Ideal case
  - All intensity values has an uniform probability distribution function (PDF)
  - Enhance an input image to have the gray level distribution, which is as uniform as possible



Histogram Equalization

- A fully automatic procedure
- Suppose our image has L different gray levels, $0,1,2,\ldots,L-1$, and gray level k occurs n_k times in image
- Ideal case
  - All intensity values has an uniform probability distribution function (PDF)
  - Enhance an input image to have the gray level distribution, which is as uniform as possible
- Practical solution
  - $s = T(r) = CDF(r)$
  - CDF: Cumulative Distribution Function

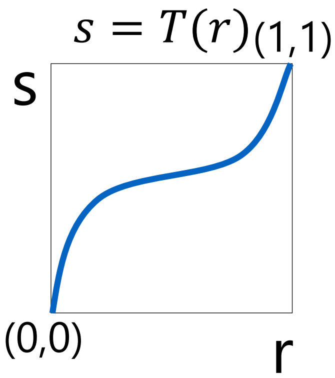


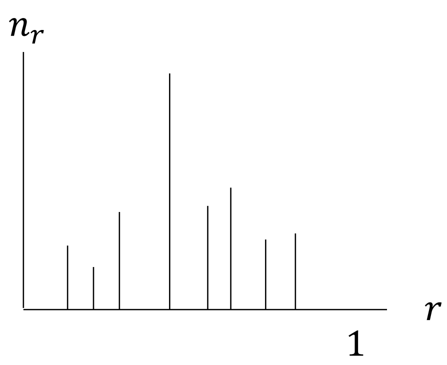
CDF and PDF

- For a given random variable X, CDF $F_X(x)$ is defined by probability density function (PDF) $f_X(x)$
  - $F_X(x) = \int_{-\infty}^{x} f_X(t)dt \ = P(X \leq x)$
    - $\frac{d}{dx} F_X(x) = f_X(x)$
  - $P(a \leq X \leq b) = F(b) - F(a) = \int_{a}^{b} f_X(t)dt$

Histogram Equalization

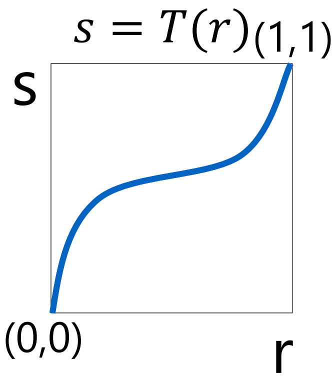
- Problem definition
  - r: gray level of input image
    - Normalized to [0, 1]
    - Probability density function: $f_R(r)$
  - Monotonic increasing function: $s = T(r)$
  - Goal is to find the function T, such that
    - $f_S(s) = 1$ for all $0 \le s \le 1$
    - i.e s is a uniform random variable

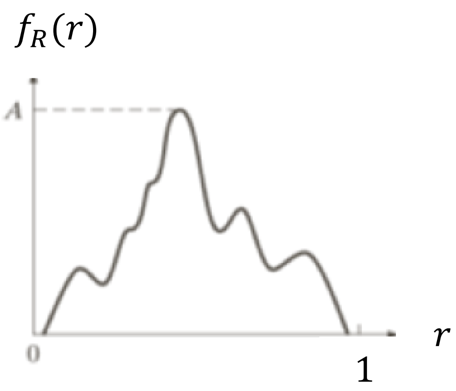


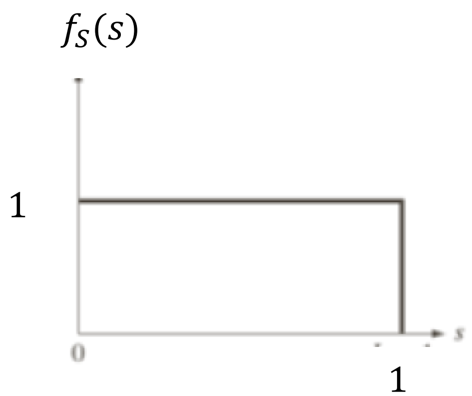


Histogram Equalization

- Problem definition
  - r: gray level of input image
    - Normalized to [0, 1]
    - Probability density function: $f_R(r)$
  - Monotonic increasing function: $s = T(r)$
  - Goal is to find the function T, such that
    - $f_S(s) = 1$ for all $0 \leq s \leq 1$
    - i.e s is a uniform random variable

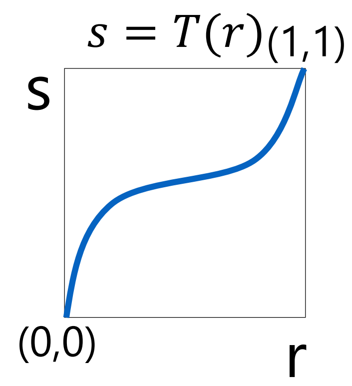






Histogram Equalization

- Problem constraints
  - $f_S(s) = 1$
  - $f_S(s)ds = f_R(r)dr$
    - $s = T(r)$ is monotonic increasing function
    - $r = T^{-1}(s)$



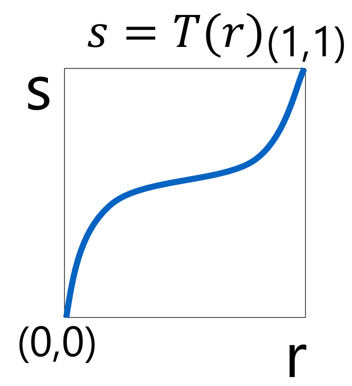
$$F_S(s) = P(S \le s) = P(T(R) \le s) = P(R \le T^{-1}(s)) = P(R \le r) = F_R(r)$$

$$\frac{d}{ds}F_S(s) = \frac{d}{ds}F_R(r)$$

$$f_S(s) = f_R(r) \cdot \frac{dr}{ds}$$

Histogram Equalization

- Problem constraints
  - $f_S(s) = 1$
  - $f_S(s)ds = f_R(r)dr$
    - $s = T(r)$ is monotonic increasing function
    - $r = T^{-1}(s)$



- $f_R(r) = T'(r)$

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

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

Histogram Equalization

- Continuous case $$s = T(r) = CDF(r) = \int_0^r f_R(t)dt$$
- Discrete approximation $$s_k = T(r_k) = \sum_{j=0}^{k} f_R(r_j) = \sum_{j=0}^{k} \frac{n_j}{N}$$

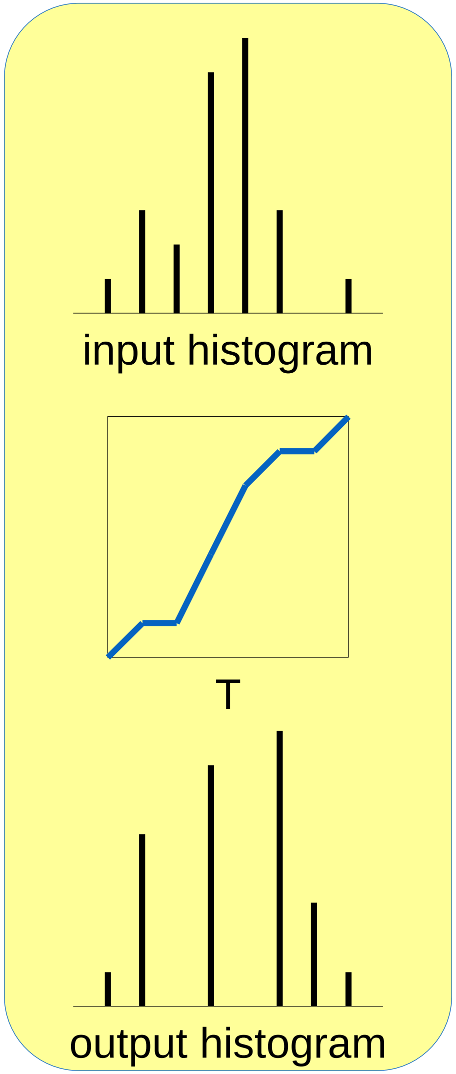
input gray level k	0	1	2	3	4	5	6	7
normalized input r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
histogram n_k	1	3	2	7	8	3	0	1
normalized histogram n_k/N	1/25	3/25	2/25	7/25	8/25	3/25	0	1/25
normalized output s_k	1/25	4/25	6/25	13/25	21/25	24/25	24/25	1
denormalized output $o_k = s_k \times 7$	7/25	28/25	42/25	91/25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

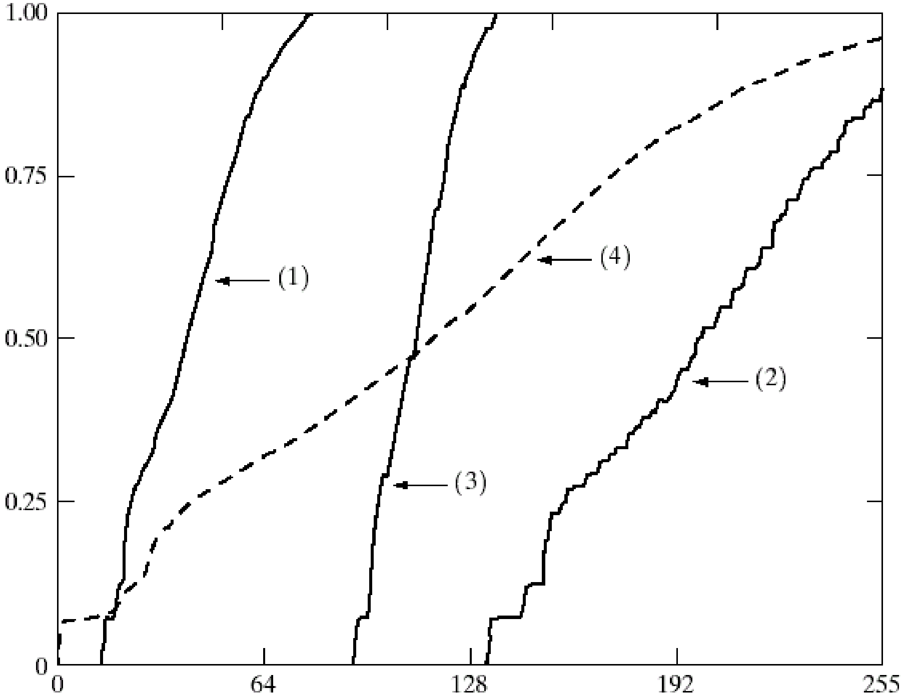
input gray level	0	1	2	3	4	5	6	7
output gray level	0	1	1	3	5	6	6	7
input histogram	1	3	2	7	8	3	0	1
output histogram	1	5	0	7	0	8	3	1

- Does not provide the exactly uniform output
  - Discrete approximation

- But, spread the histogram automatically

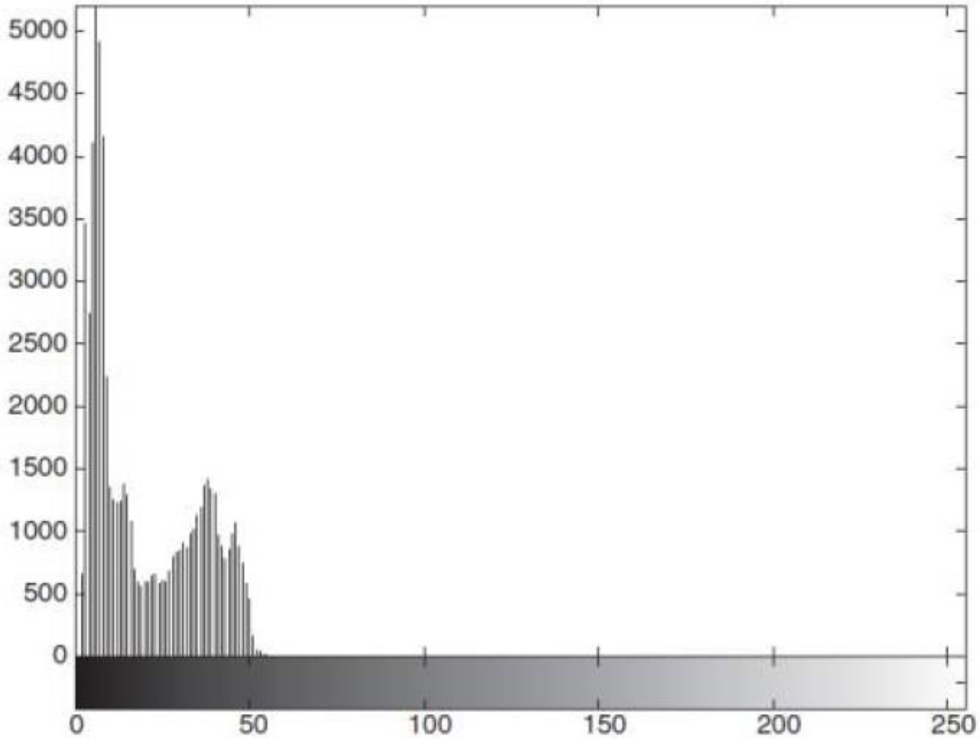


Histogram Equalization



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
```