CHAPTER 2

POINT PROCESSES

What is an image?

[Albrecht Dürer, 1525]

- **Image**: a visual representation in form of a function $f(x,y)$ where f is related to the brightness (or color) at point (x,y)
- Most images are defined over a rectangle
- Continuous in amplitude and space

Digital Images and Pixels

- **Digital image**: discrete samples $f[x,y]$ representing continuous image $f(x,y)$
- Each element of the 2-d array $f[x,y]$ is called a **pixel** or **pel** (from "picture element")

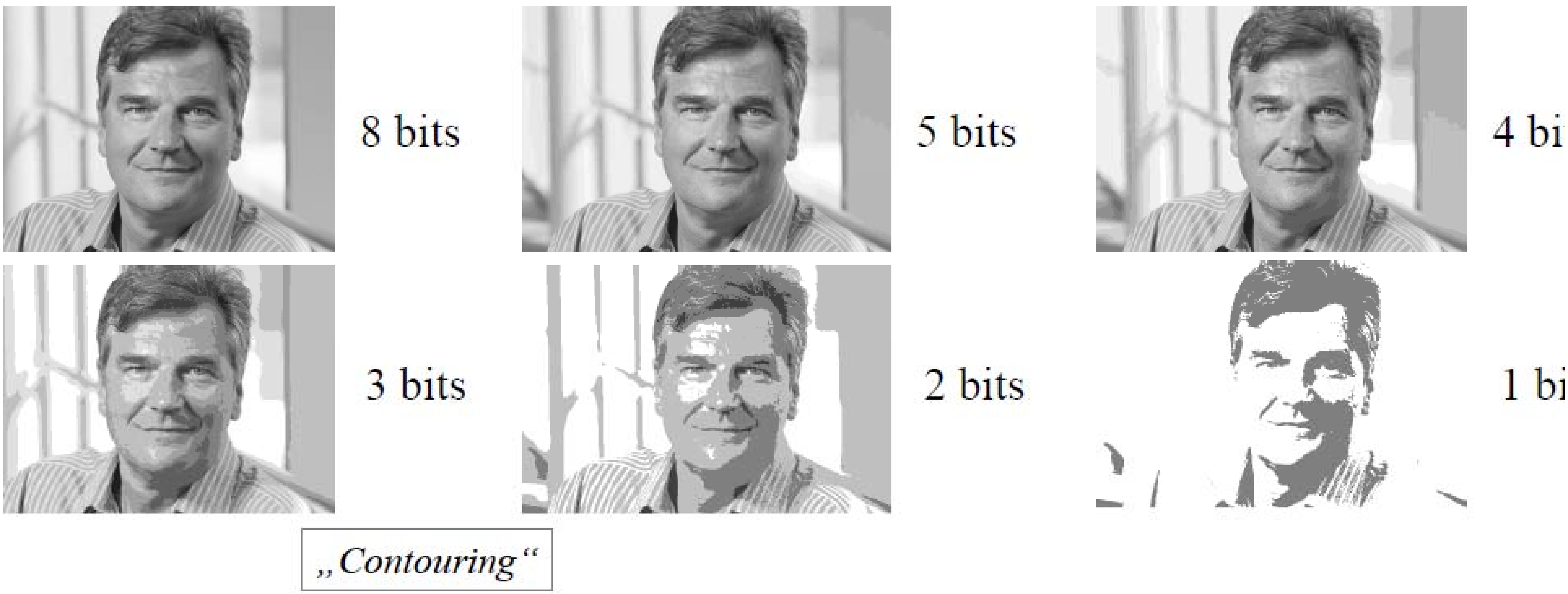

200x200


100x100


50x50

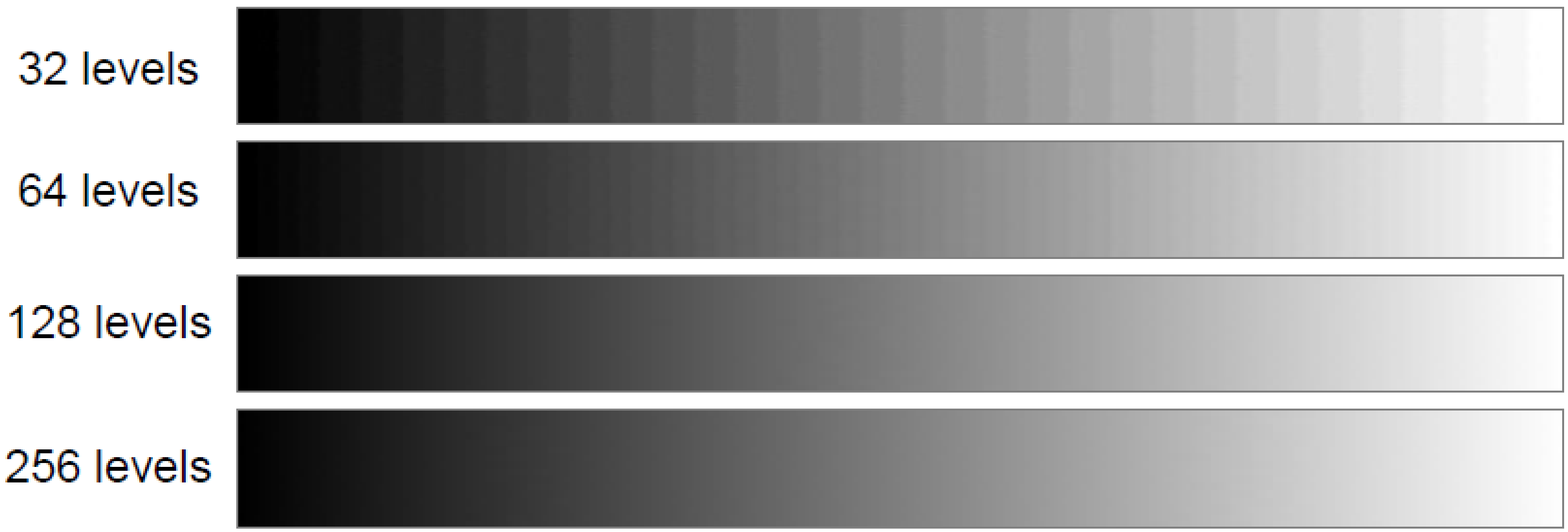

25x25

Quantization: how many bits per pixel?



How many gray levels are required?

- Contouring is most visible for a ramp



- Digital images typically are quantized to 256 gray levels.

Introduction

- Point Processes vs. Area Processes
    - Point processes: operate on a pixel based solely on that pixel's value.
    - Area processes: use the input pixel as well as the pixels around it to generate a new pixel.
    - Point processes are easily implemented as "look-up tables".

Arithmetic Operations

- Adding, subtracting, dividing, and multiplying pixels by a constant value.

addition (+40)

subtraction (-40)

original image

multiplication (x1.2)

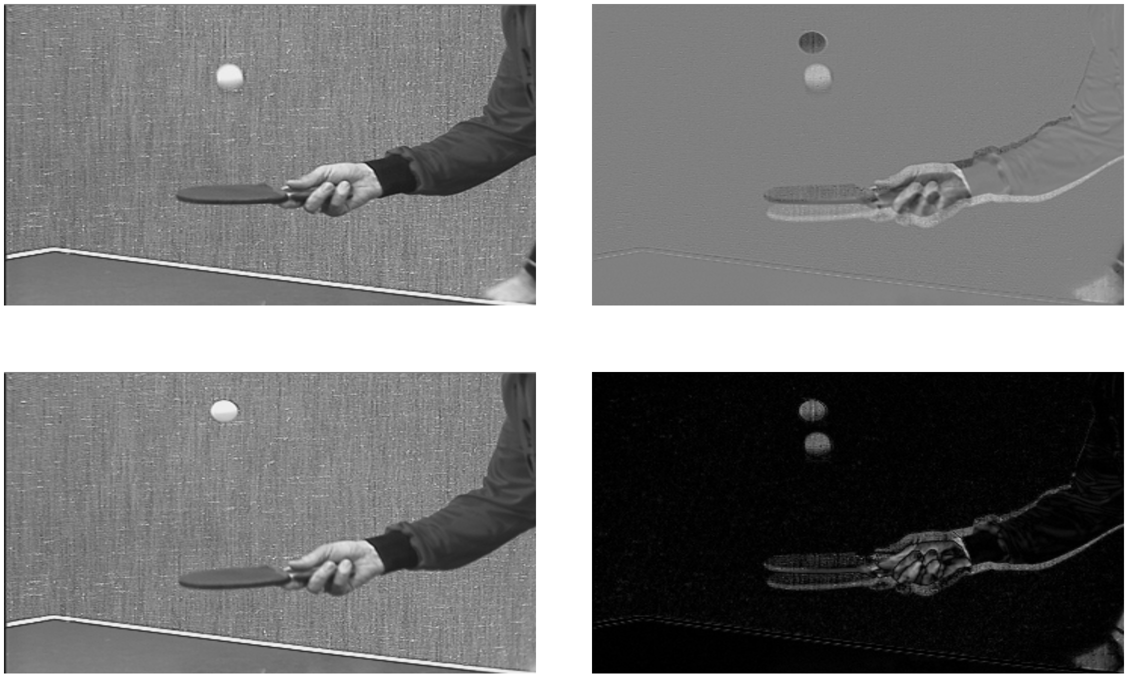
division (/1.2)

Arithmetic Operations

- Problems
  - Can create negative values and values greater than the maximum possible values.
- Clamping
  - Set negative values to 0.
  - Set values greater than 255 to 255.

Arithmetic Operations

- Image Subtraction and Change Detection
  - medical imaging application : display blood-flow paths
  - automated inspection of printed circuits
  - security monitoring



Arithmetic Operations

add operation

Original image

subtract operation

XOR Operations

- XOR (Exclusive OR): X'Y+XY'

X	Y	X'Y+XY'
0	0	0
0	1	1
1	0	1
1	1	0

The XOR function can be used to find all pixels of a certain value. Every pixel that is the specified value will be set to black. All other pixels will be non-black.


original image


XOR 48


XOR 128


XOR 255

XOR Operations

- The XOR function is frequently used on graphics system to generate a cursor for the mouse.
- By XORing the cursor mask with the existing pixels, the colors are changed but you can still see the shape of the image below the cursor.
- The beauty of the XOR function when superimposing is that you need not know the value of the background to create a high contrast.

Look-Up Tables

- Use the current pixel value as the array index.
- New value is the array element pointed by this index.

7	7	5	4	2
7	6	4	3	0
7	6	4	3	1
6	6	4	2	0
5	5	3	1	0

0	0
1	0
2	1
3	1
4	2
5	3
6	4
7	5

5	5	3	2	1

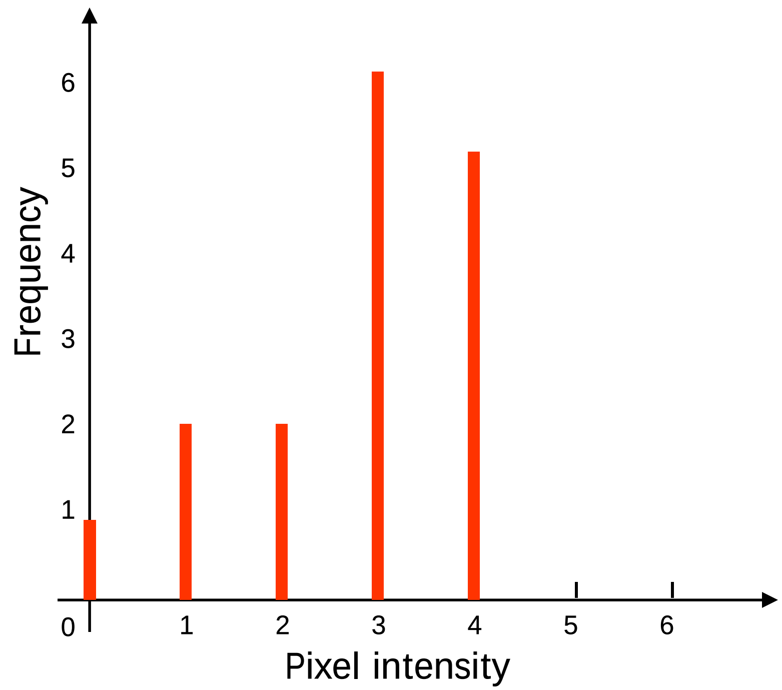
Operation of a 3-bit look-up table

Histograms

- A bar graph of pixel intensities.
- The pixel intensities are plotted along the x-axis and the number of occurrences (frequency) for each intensity represents the y-axis.

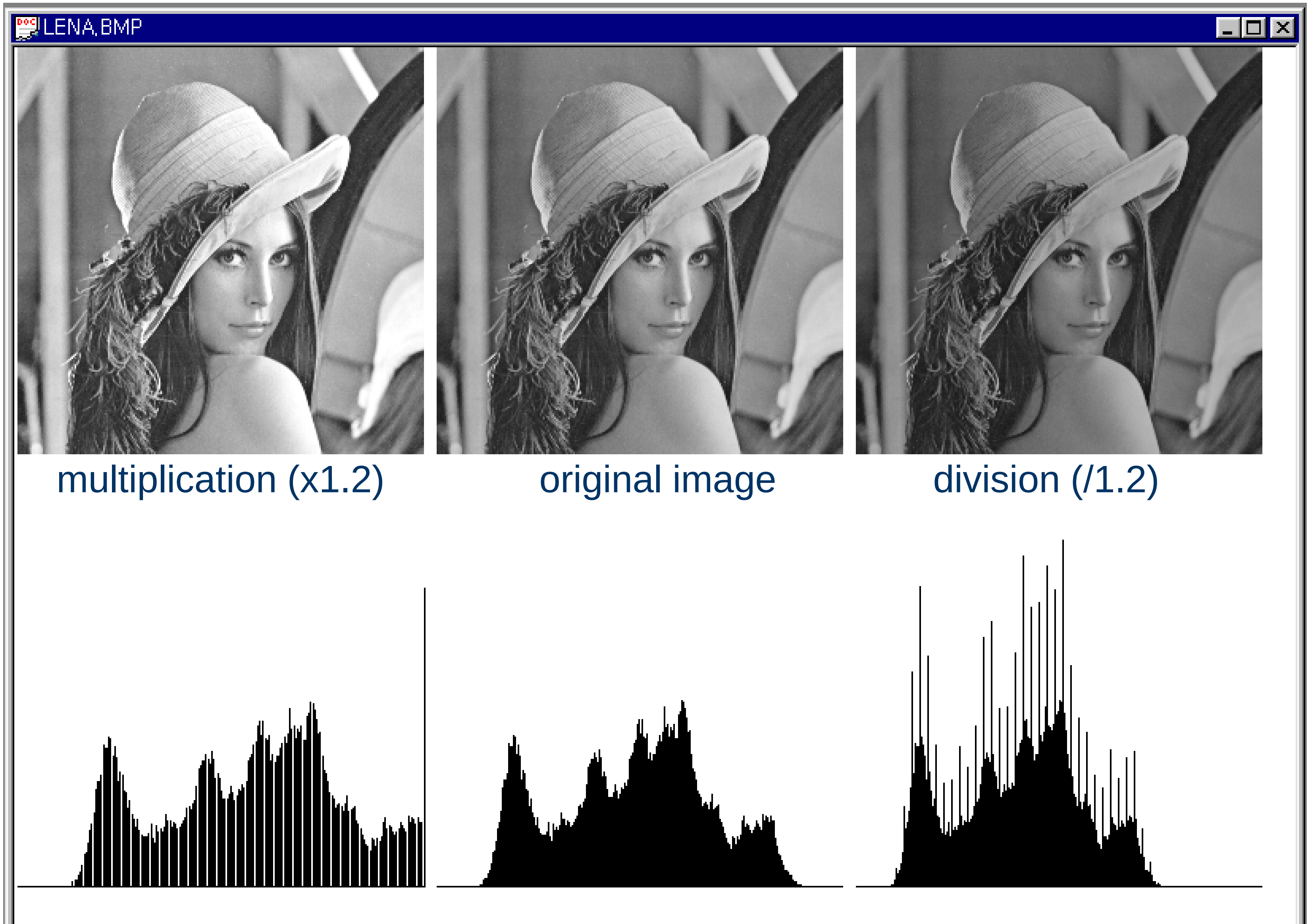
4	4	3	3
4	4	3	3
4	1	2	3
0	1	2	3

image



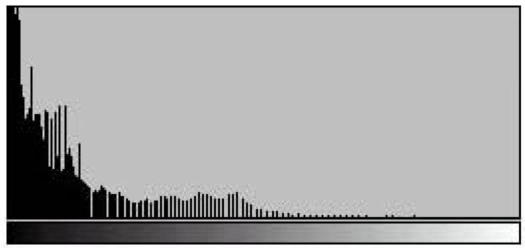
Histograms

Histograms



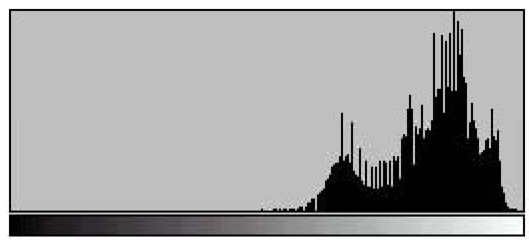
Histograms

image



Histogram

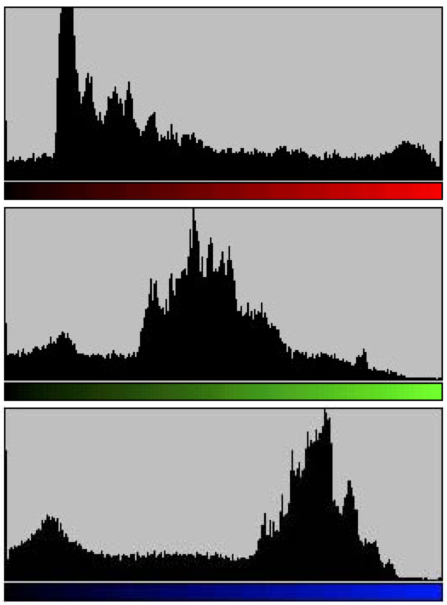
image



Histogram

Histograms

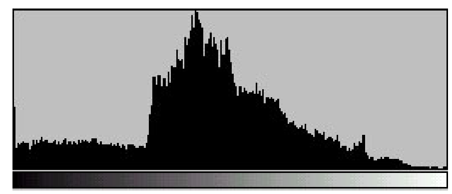
- RGB Color vs. Gray Images



Histogram

Original image

Original image



Histogram

Histograms

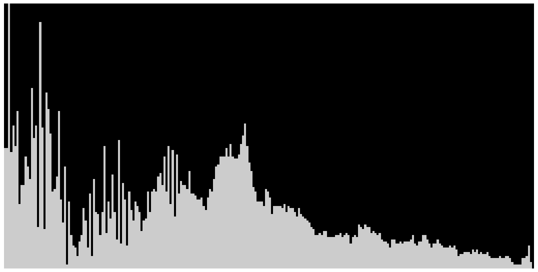
- Compression, Expansion, and Shift

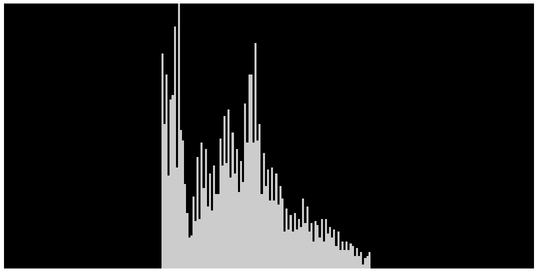

(a)

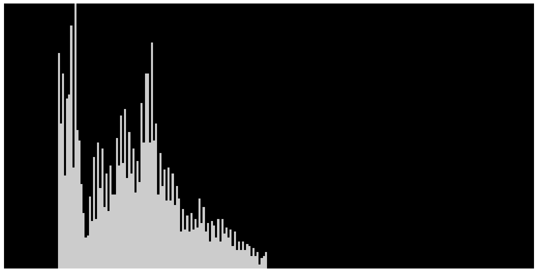

(b)

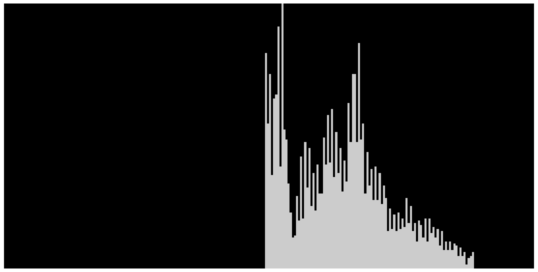

(c)


(d)


(1)


(2)


(3)


(4)

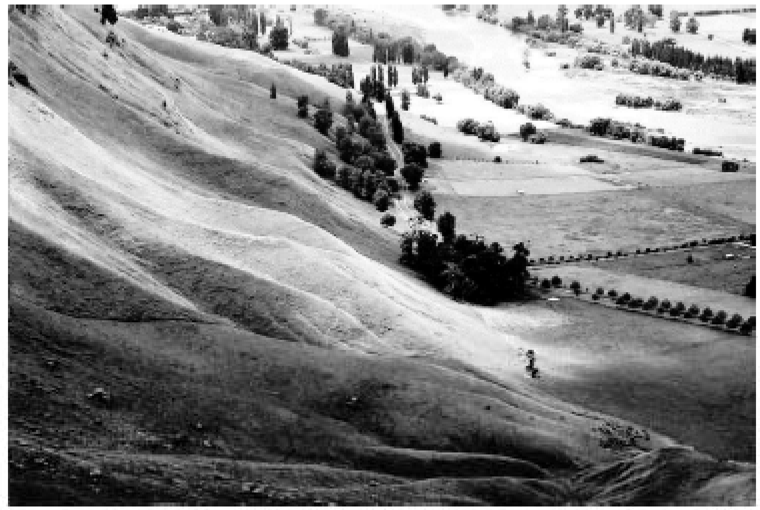
Histogram Equalization

- Goal: To obtain a uniform histogram.
- Histogram equalization will NOT “flatten” a histogram.
- It redistributes intensity distributions.
- “Spreading” is a better term than “flattening” to describe histogram equalization.

- Three Steps for Histogram Equalization
  - Compute histogram
  - Calculate normalized sum of histogram
  - Transform input to output image

Histogram equalization example

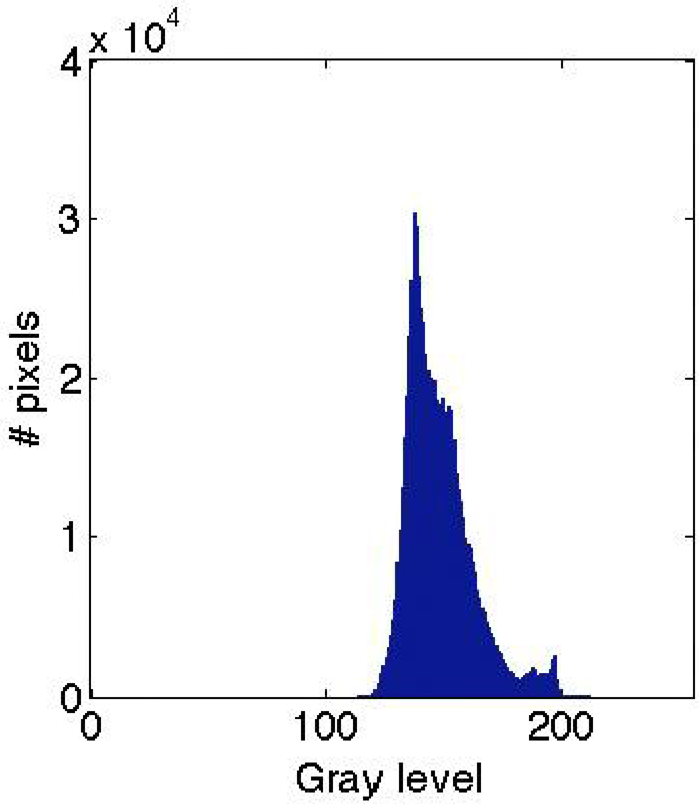
Original image *Bay*



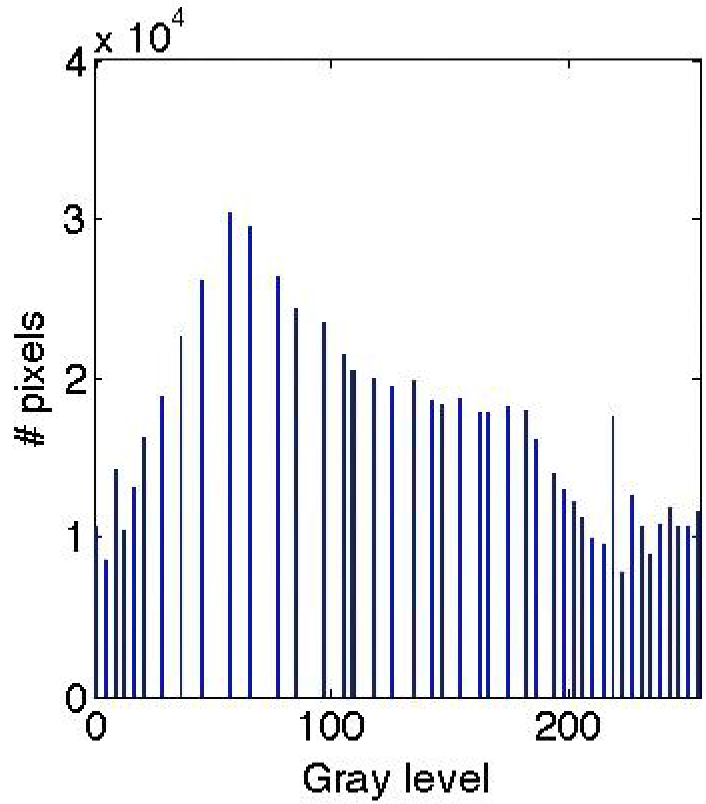
... after histogram equalization

Histogram equalization example

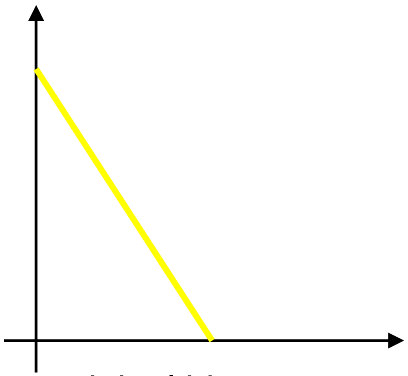
Original image *Bay*



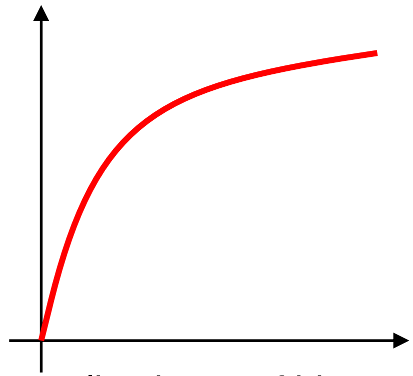
. . . after histogram equalization



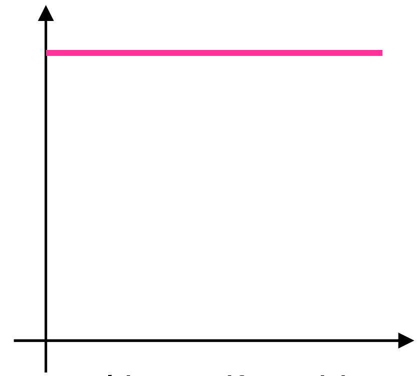
Histogram Equalization



original histogram

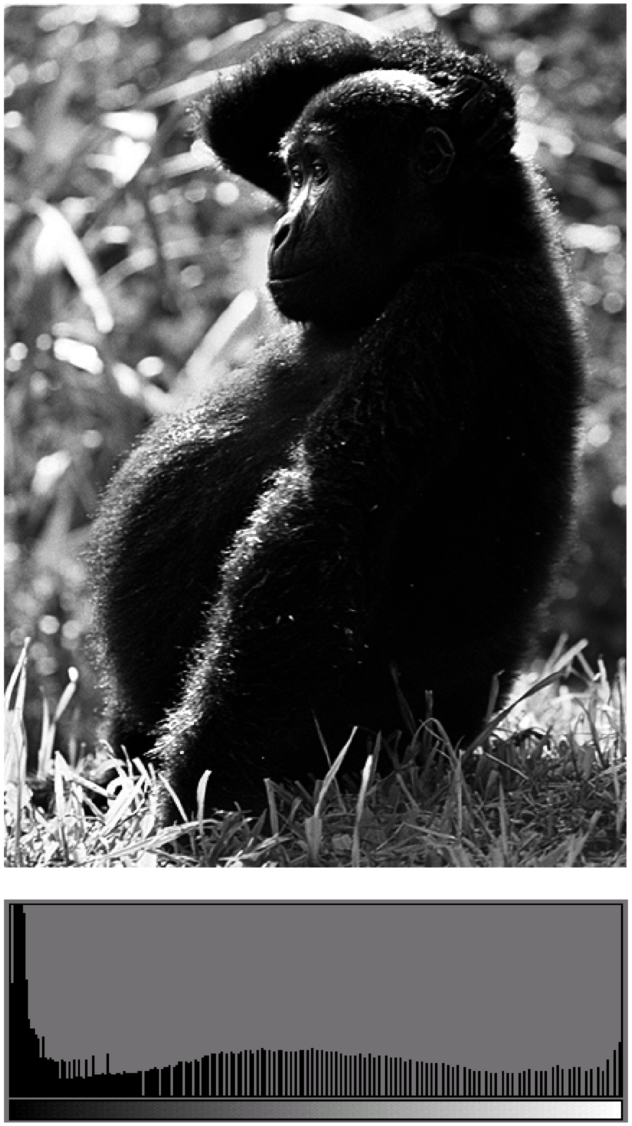


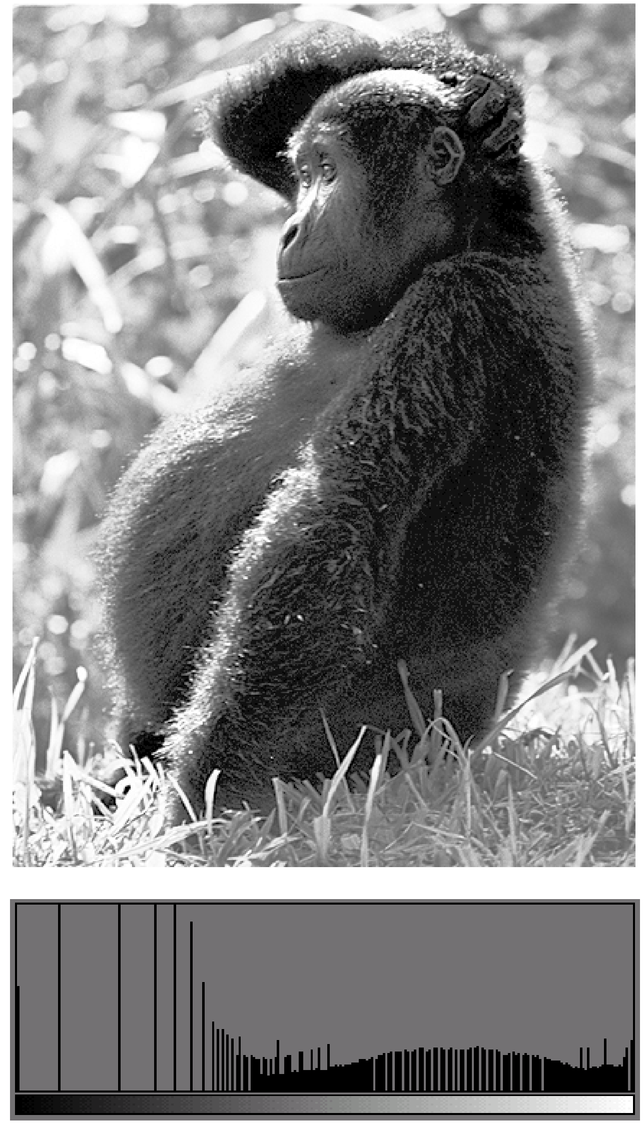
normalized sum of histogram



resulting uniform histogram

Histogram Equalization (Spreading)



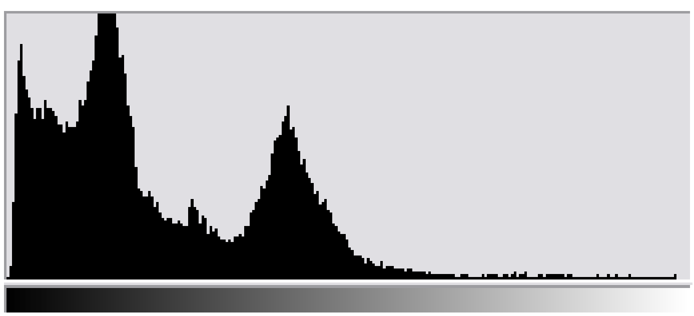


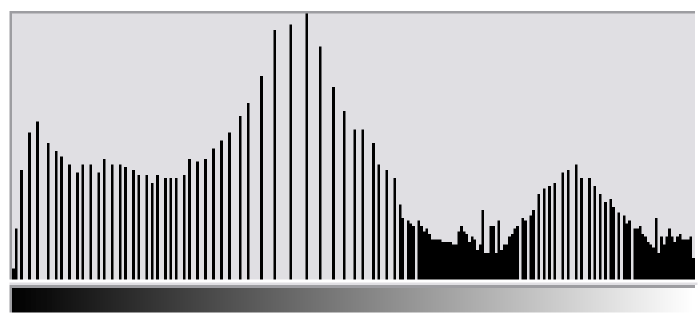
Histogram Equalization (HE)

- Effects of HE

  - HE stretches contrast (expand the range of gray levels) for gray levels near histogram maxima.
  - Compresses contrast in areas with gray levels near histogram minima.
  - Contrast is expanded for the most of the image pixels.
  - HE usually improves detectability of many image features.
  - Similar effect of enhancement could be achieved by manual contrast stretching approach, but the advantage of HE is fully automatic.

Histogram Equalization



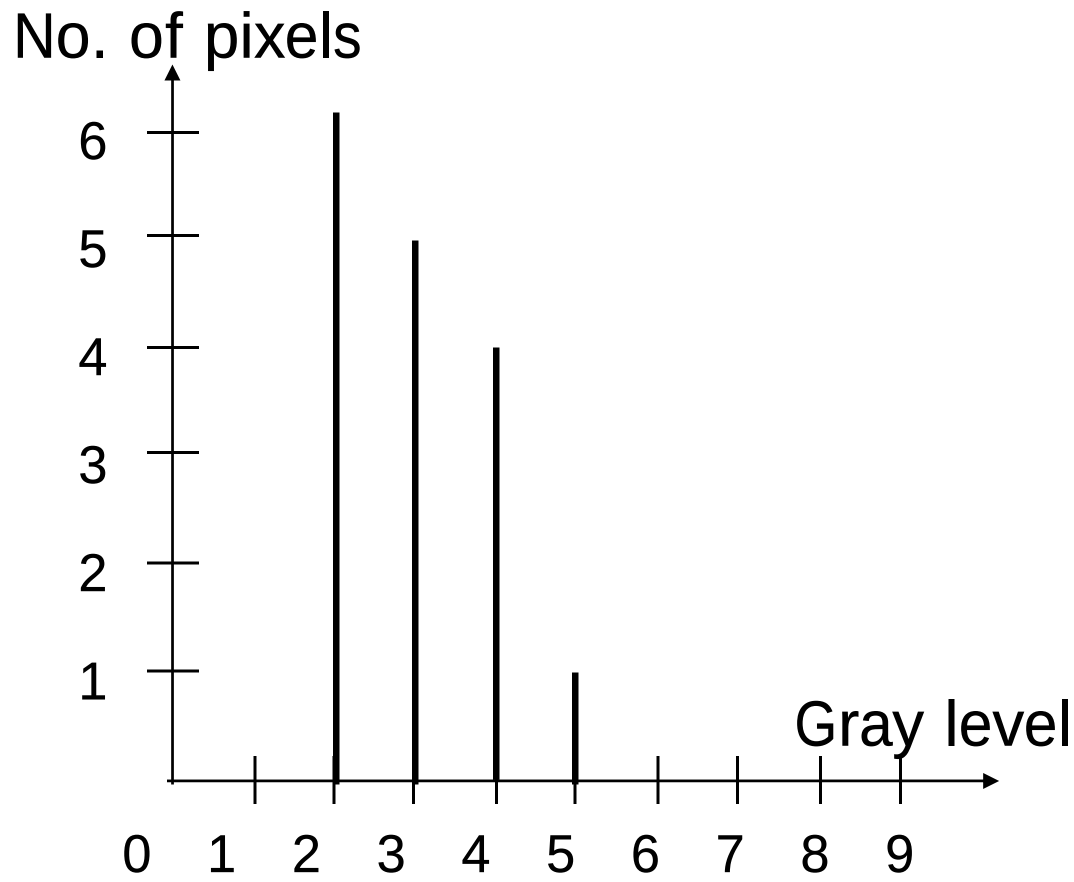


Histogram Equalization - Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Histogram Equalization - Example

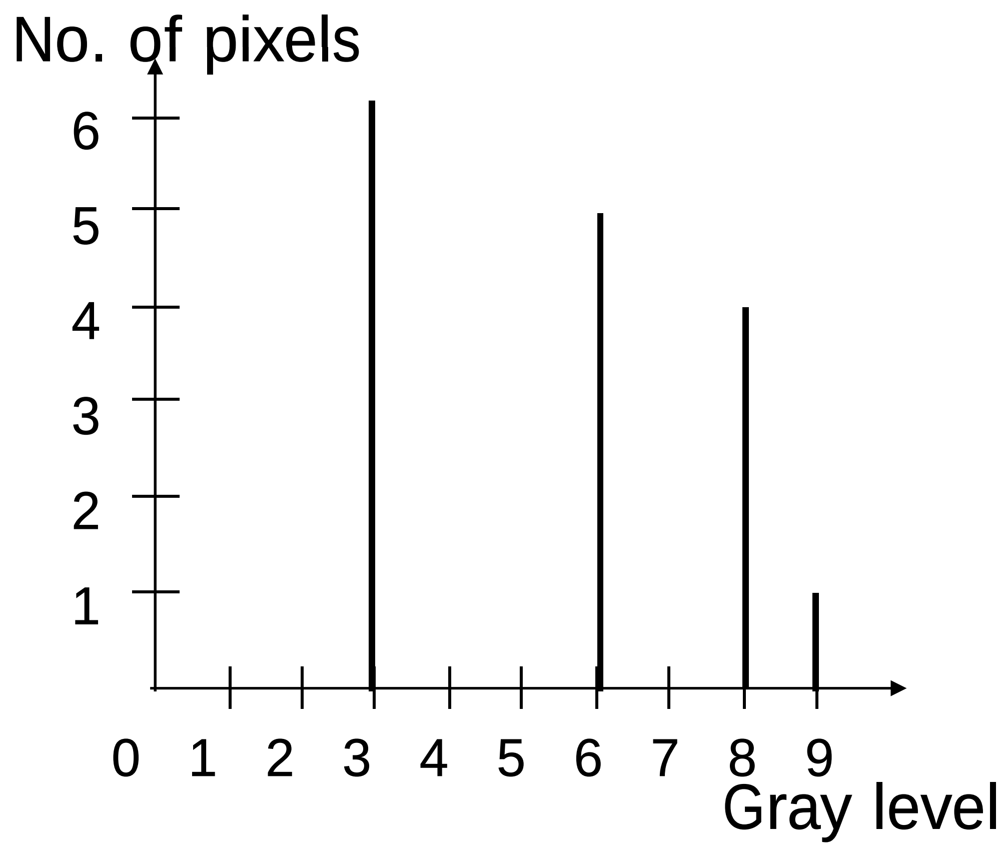
Gray Level	0	1	2	3	4	5	6	7	8	9
No.of pixels	0	0	6	5	4	1	0	0	0	0
	0	0	6	11	15	16	16	16	16	16
	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16/ 16	16/ 16	16/ 16	16/ 16
s* x *9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Histogram Equalization - Example

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]

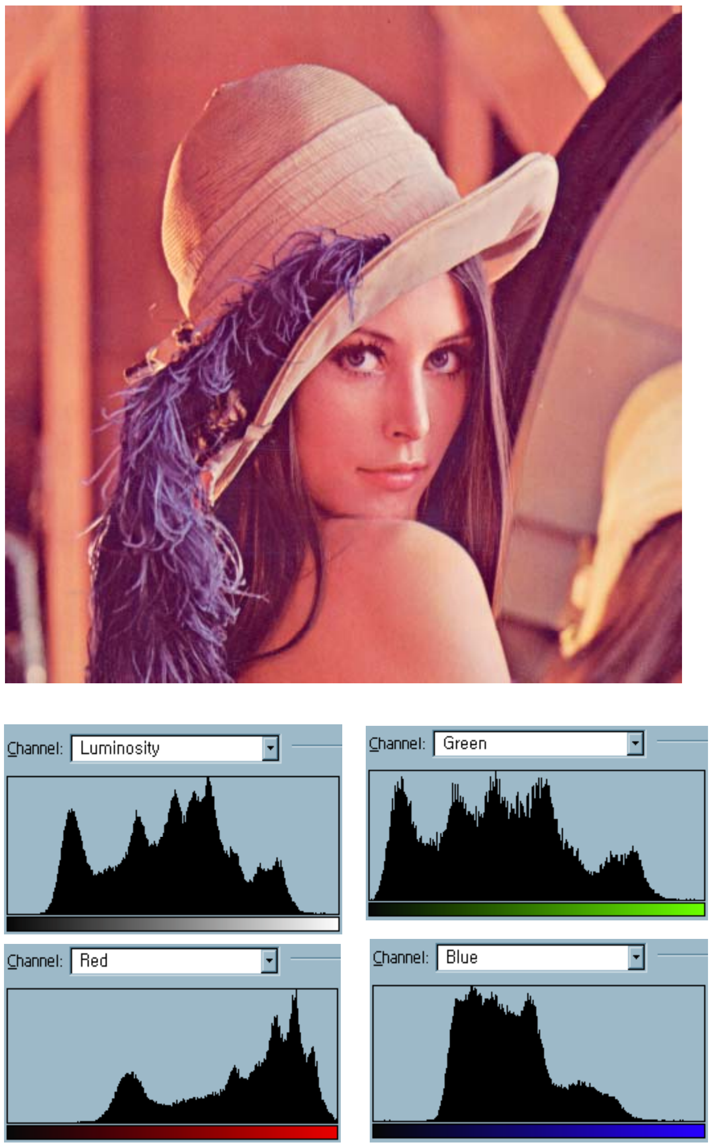


Histogram equalization

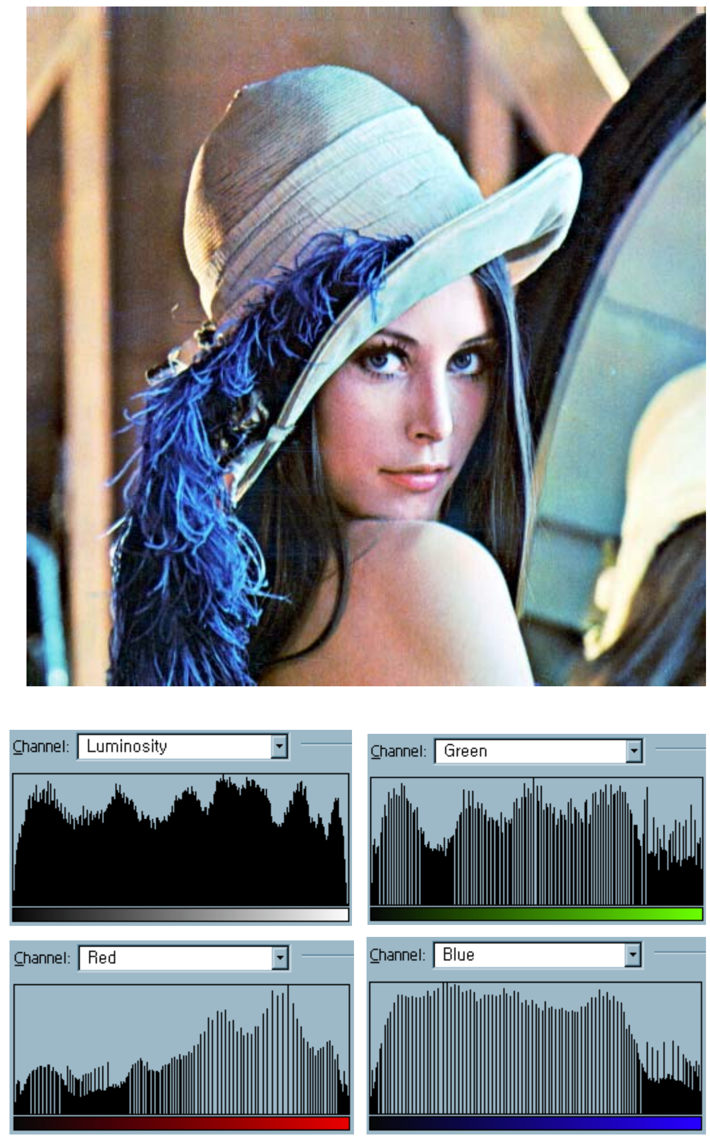
Histogram Equalization

```
nrow=512; ncol=512; npixels = nrow*ncol;
maxlevel = 255;
void hist_equalize(int im[][], nrow, ncol) {
   int hist[maxlevel+1], sum[maxlevel+1];
// initialize : hist[]=0;
   for (int k=0; k<=maxlevel; k++)   hist[k]=0;
// calc histogram
  for (int i =0; i<nrow;i++)
    for (int j=0; j<ncol; j++)
         hist[ im[i][j] ]++;
// calc sum
   sum[0] = hist[0];
   for (int k=1; k<=maxlevel; k++) sum[k]=sum[k-1]+hist[k];
// calc normalized sum :
   for (k=0; k<=maxlevel; k++)
     sum[k] = (int) ((float) (sum[k]*maxlevel) / (float)npixels + 0.5);
// transform : sum[] acts as LUT
  for (int i =0; i<nrow;i++)
     for (int j=0; j<ncol; j++)
          im[i][j] = sum[ im[i][j] ];
}
```

Color Histogram Equalization



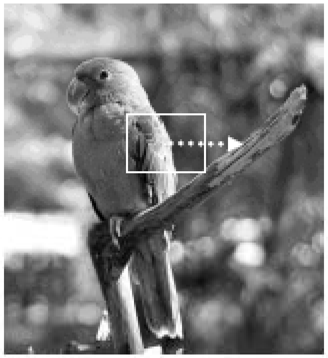
Original Image



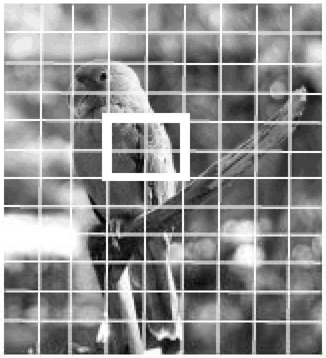
Equalized Image

Adaptive histogram equalization

Histogram equalization based on a histogram obtained from a portion of the image



Sliding window approach:
different histogram (and mapping) for every pixel



Tiling approach:
subdivide into overlapping regions, mitigate blocking effect by smooth blending between neighboring tiles

Limit contrast expansion in flat regions of the image, e.g., by clipping histogram values.
("Contrast-limited adaptive histogram equalization")

[Pizer, Amburn et al. 1987]

Adaptive histogram equalization

Original image
Parrot

Global histogram
equalization

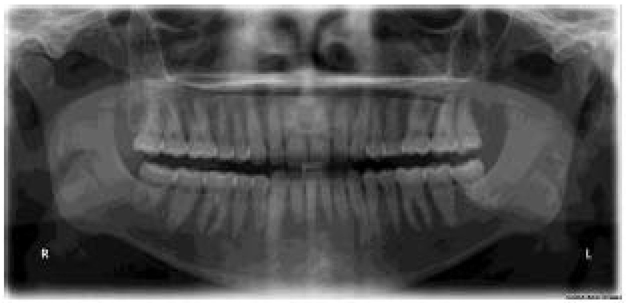
Adaptive histogram
equalization, 8x8 tiles

Adaptive histogram
equalization, 16x16 tiles

STOP HERE 2014/01/21

Adaptive histogram equalization

Original image
Dental Xray



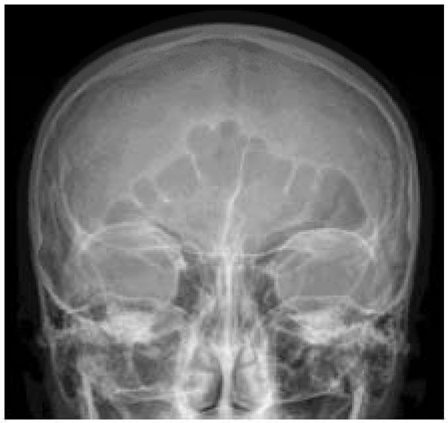
Global histogram
equalization

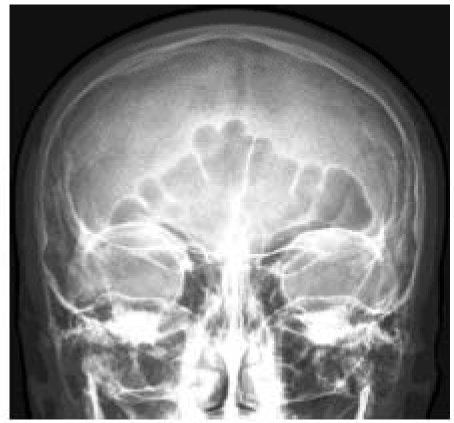
daptive histogram
ualization, 8x8 tiles

Adaptive histogram
equalization, 16x16 t

Adaptive histogram equalization

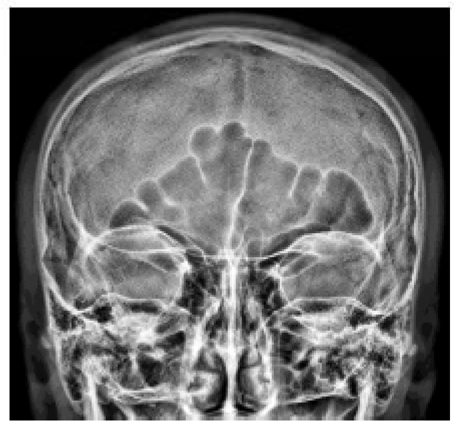
Original image
Skull Xray





Global histogram
equalization

Adaptive histogram
equalization, 8x8 tiles

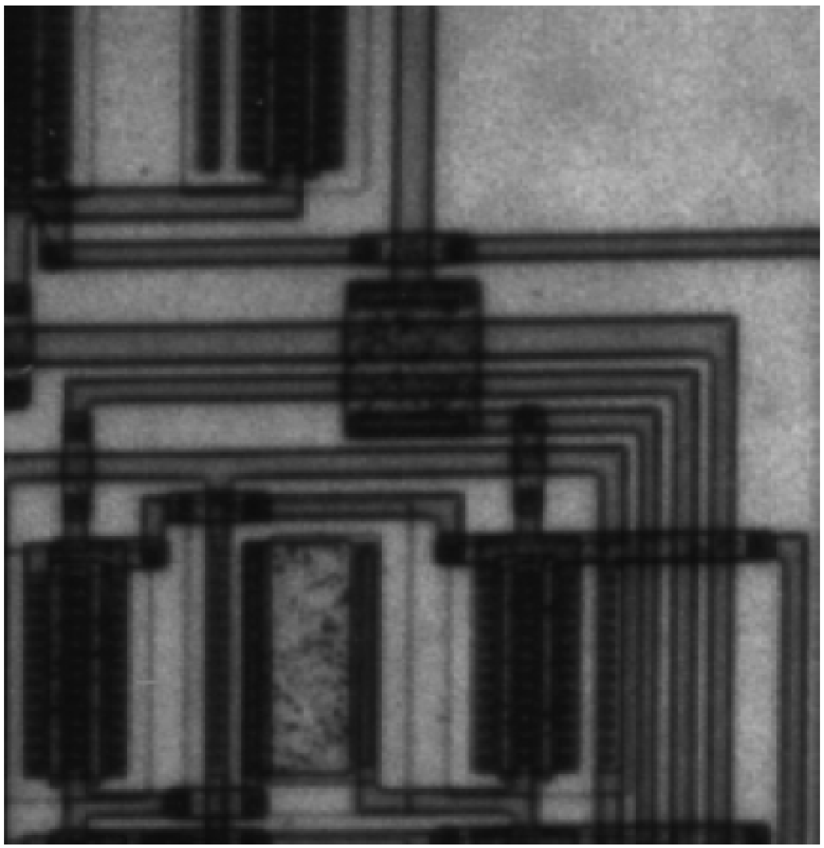


Adaptive histogram
equalization, 16x16 tiles

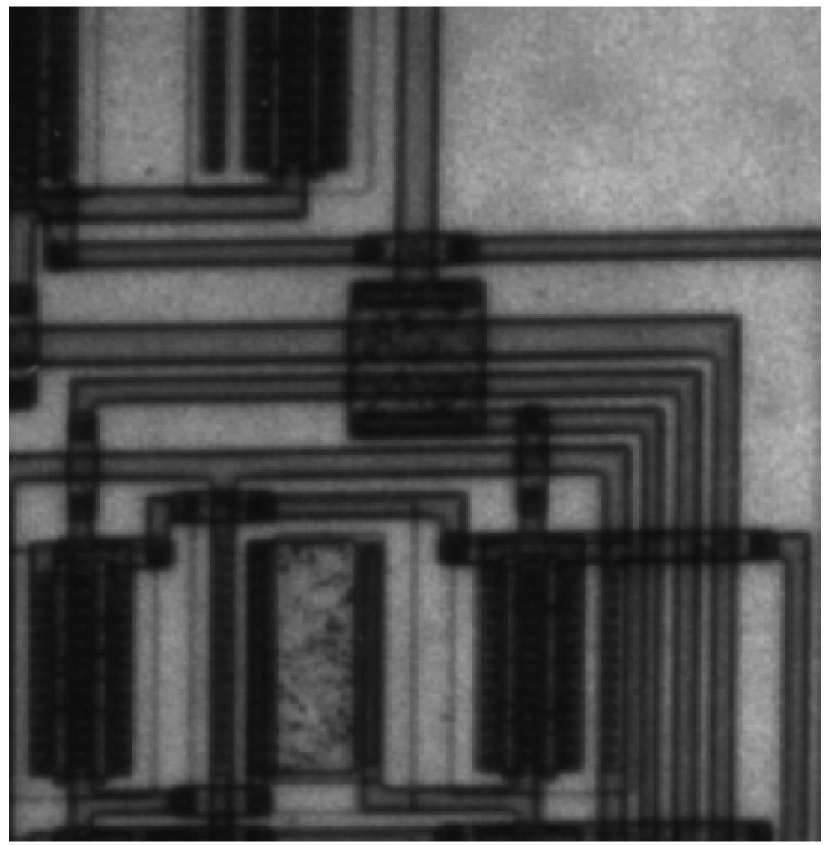
Histogram Specification - Application

- We wish to check if a circuit board (image 1) matches the template (image 2) from which it was manufactured. Any defects?

1 - Manufactured

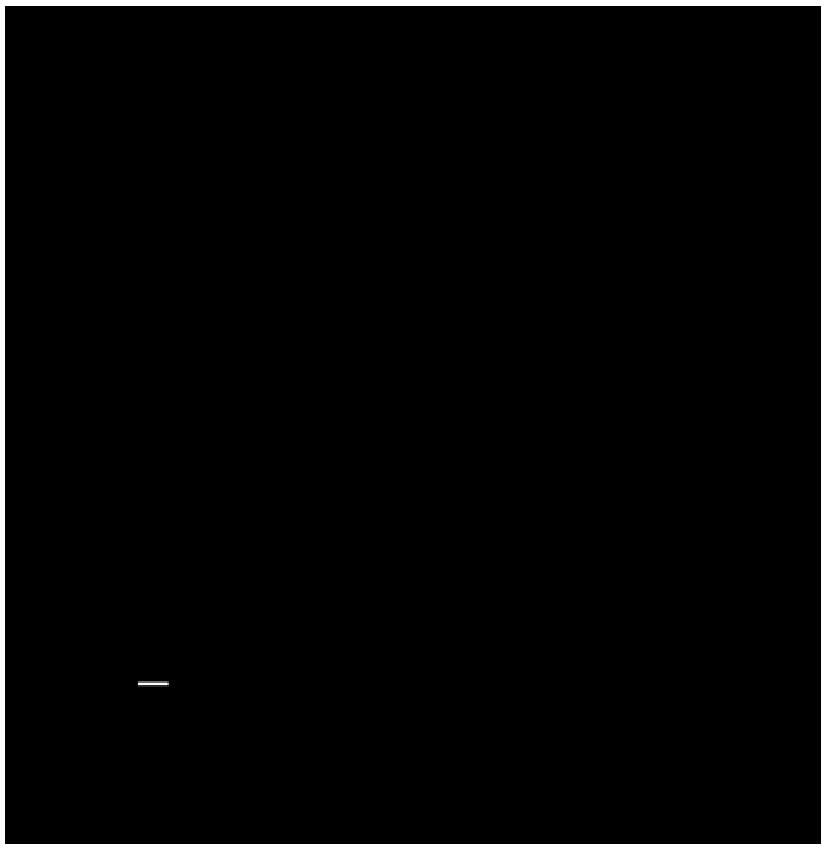


2 - Template



Histogram Specification - Application

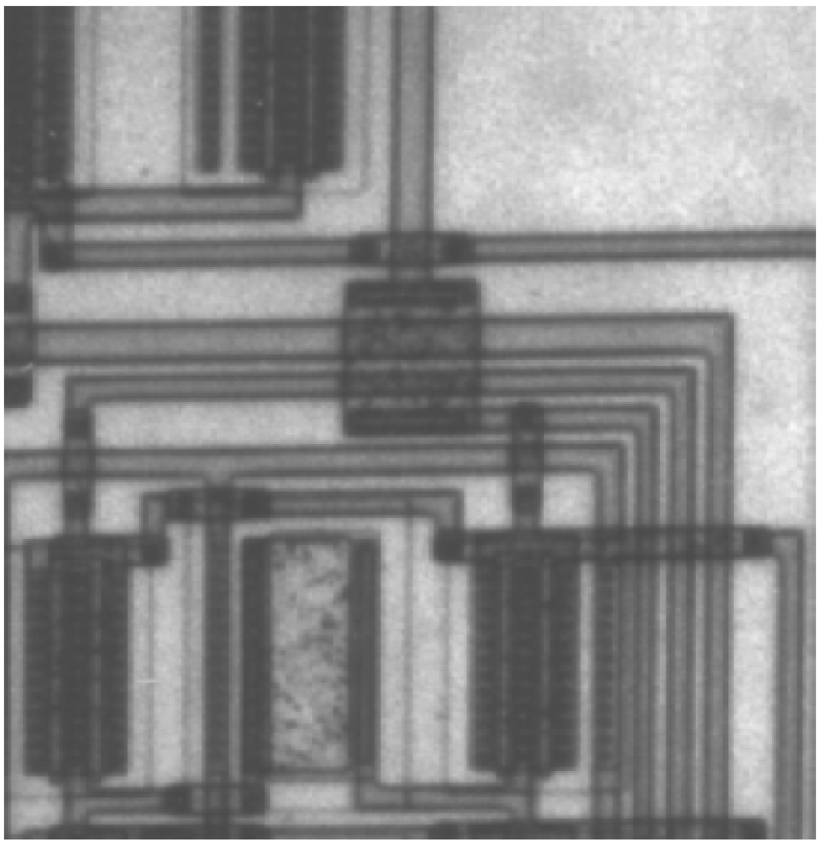
- Compute difference image (defined later), threshold by setting pixels with non-zero absolute difference to 1 and all other pixels to 0:



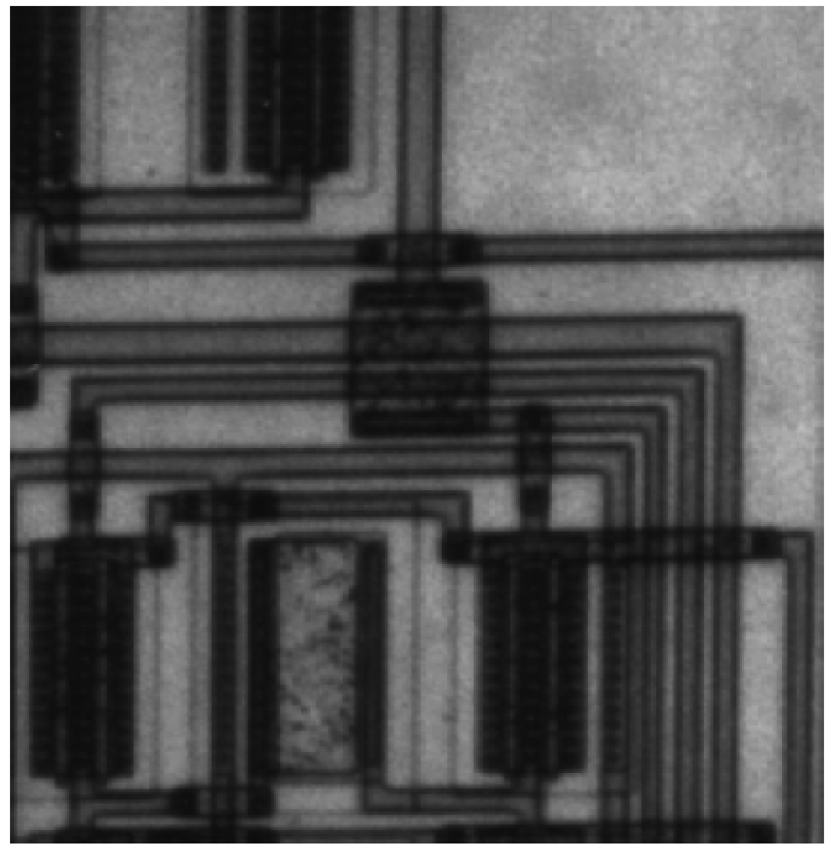
Histogram Specification - Application

- What if the overall brightness of image 1 is different from that of image 2?

1 - Manufactured

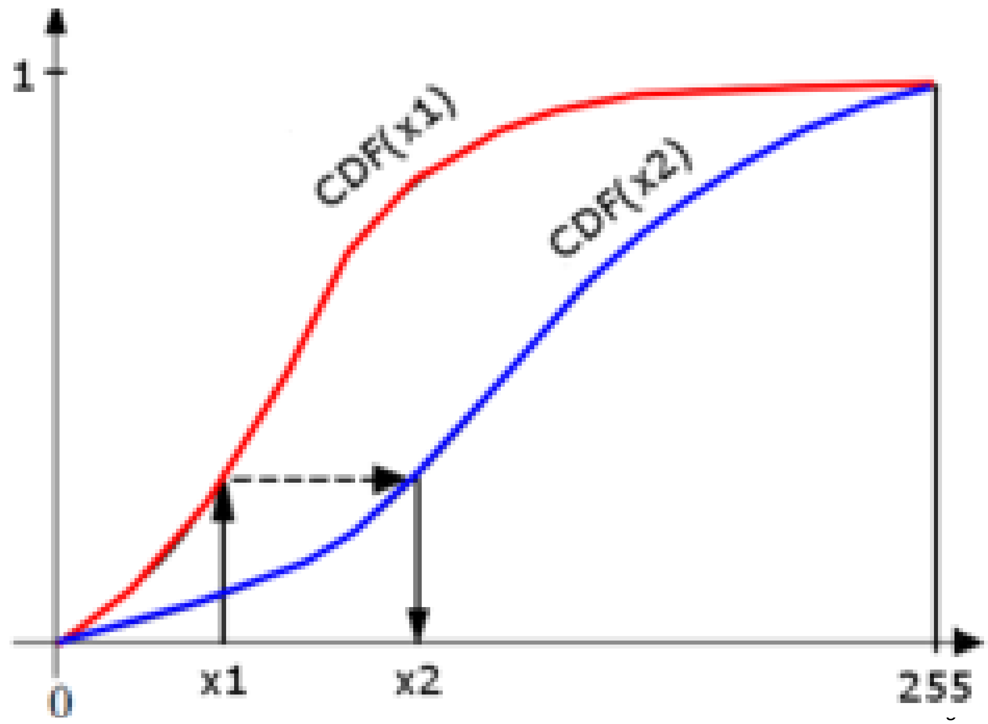


2 - Template



Histogram Specification

Given two images, the reference and the adjusted images, we compute their histograms. Following, we calculate the cumulative distribution functions of the two images' histograms - $F_1()$ for the reference image and $F_2()$ for the target image. Then for each gray level $G_1 \in [0, 255]$, we find the gray level G_2 for which $F_1(G_1) = F_2(G_2)$, and this is the result of histogram matching function: $M(G_1) = G_2$. Finally, we apply the function $M()$ on each pixel of the reference image.



Steps of Algorithm

- Step 1:

  Find histogram of input image h_x, and find its cumulative H_x, the histogram equalization mapping function: $H_x[j] = \sum_{i=0}^{j} h_x[i]$

- Step 2:

  Specify the desired histogram h_z, and find its cumulative H_z, the histogram equalization mapping function: $H_z[j] = \sum_{i=0}^{j} h_z[i]$

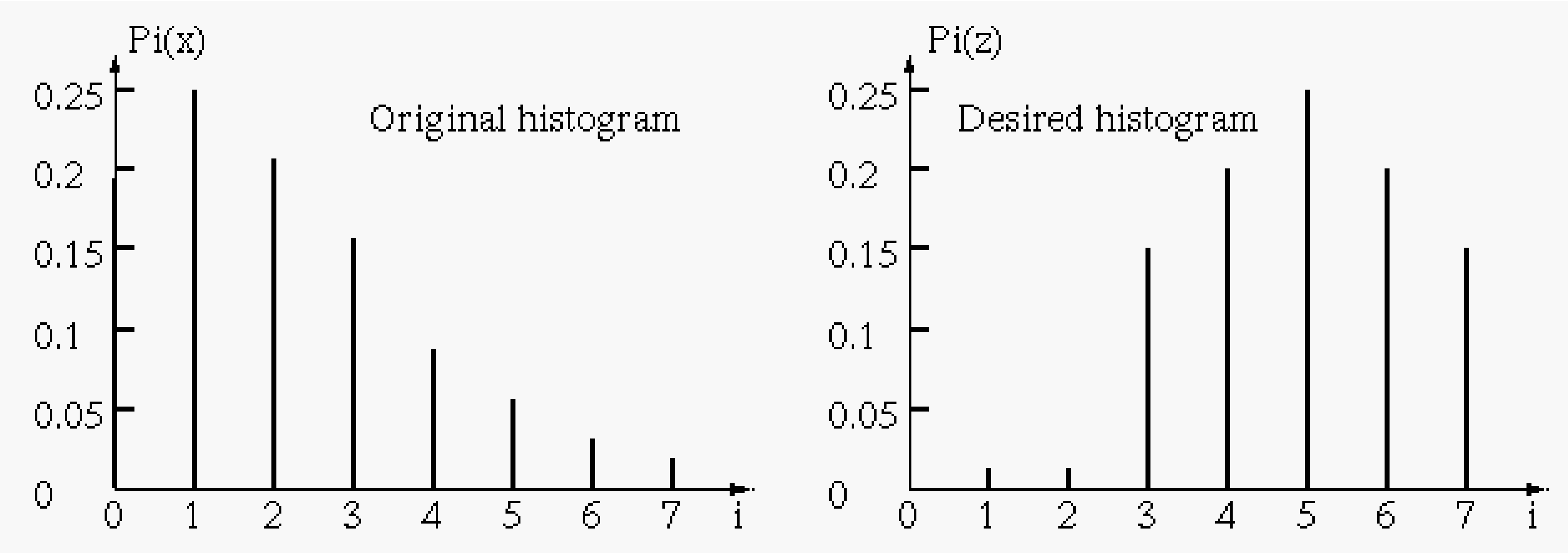
- Step 3: Relate the two mapping above to build a **lookup table** for the overall all mapping.

  Specifically, for each input level *i*, find and output level *j* so that $H_z[j]$ best matches $H_x[i]$

$$H_x[i] - H_z[j] = \min_k |H_x[i] - H_z[k]|$$

And then, $lookup[i] = j$

Histogram Specification - Example



Histogram Specification - Example

- Step 1: Equalize p_x

x_i	n_j	h_x	$y = H_x$
0	790	0.19	0.19
1	1023	0.25	0.44
2	850	0.21	0.65
3	656	0.16	0.81
4	329	0.08	0.89
5	245	0.06	0.95
6	122	0.03	0.98
7	81	0.02	1.00

Histogram Specification - Example

- Step 2: Equalize p_z

z_i	p_z	$\underline{y' = H_z}$
0	0.0	0.0
1	0.0	0.0
2	0.0	0.0
3	0.15	0.15
4	0.20	0.35
5	0.30	0.65
6	0.20	0.85
7	0.15	1.0

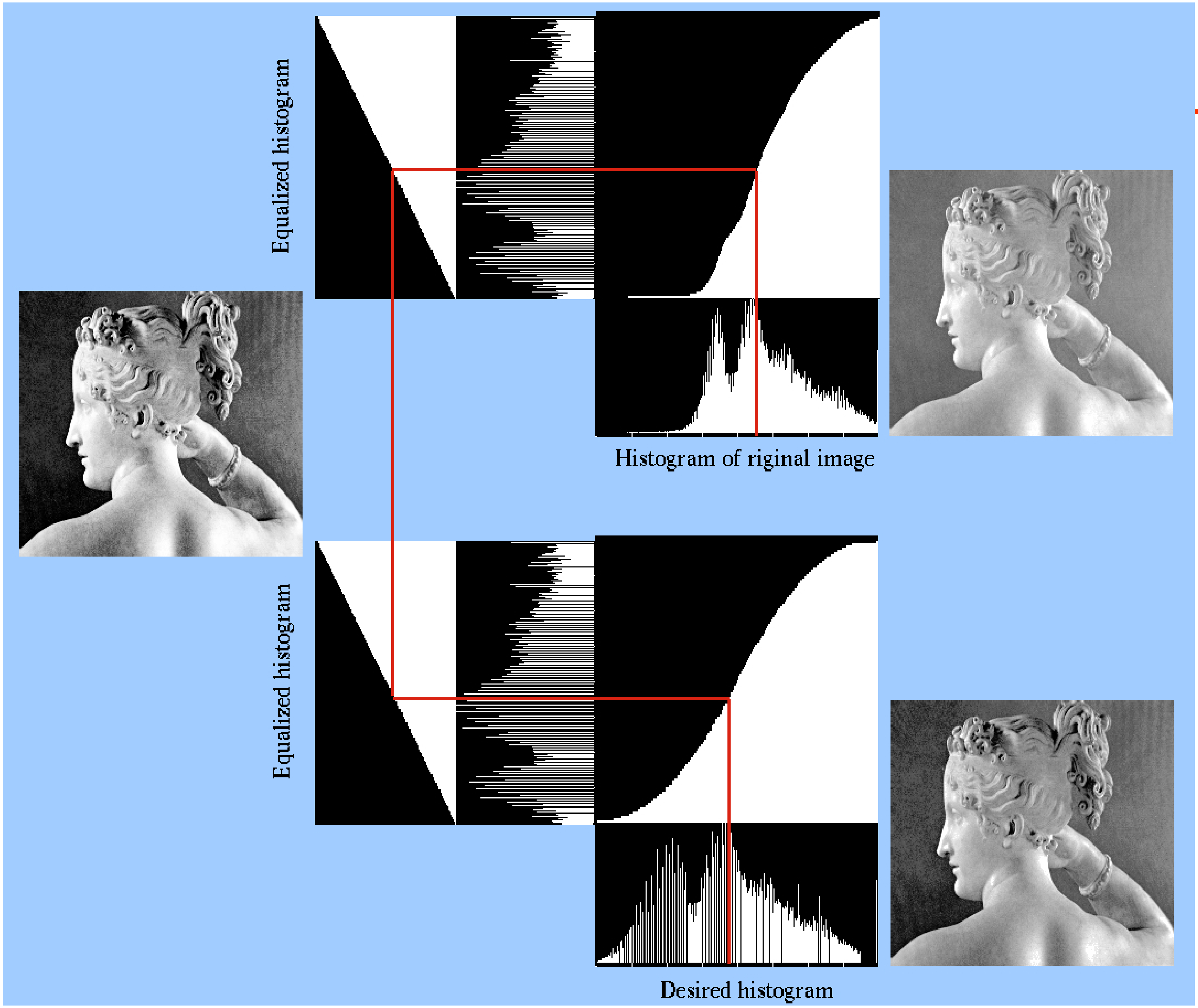
Histogram Specification - Example

- Step 3: Obtain overall mapping $x \rightarrow y \rightarrow y' \rightarrow z$

$x_i = i$	$y_j = H_x$	$\underline{y'_j = H_z}$	$z_j = j$
0	0.19	0.0	3
1	0.44	0.0	4
2	0.65	0.0	5
3	0.81	0.15	6
4	0.89	0.35	6
5	0.95	0.65	7
6	0.98	0.85	7
7	1.0	1.0	7

Here is the look-up table:

i	0	1	2	3	4	5	6	7
j	3	4	5	6	6	7	7	7


Equalized histogram
Histogram of riginal image
Equalized histogram
Desired histogram

Histogram Specification - Programming

j=0;
for $(i = 0;\ \ i < 256;\ \ i++)$ {
    $if\ \ (Hx[i] \leq Hz[j])\ \ lookup[i] = j;$
    $else$ {
        $while(Hx[i] > Hz[j])\ \ j++;$
        $if\ \ (Hz[j] - Hx[i] > Hx[i] - Hz[j-1])\ \ \ lookup[i] = j - 1;$
        $else\ \ \ lookup[i] = j;$
    }
}

Contrast

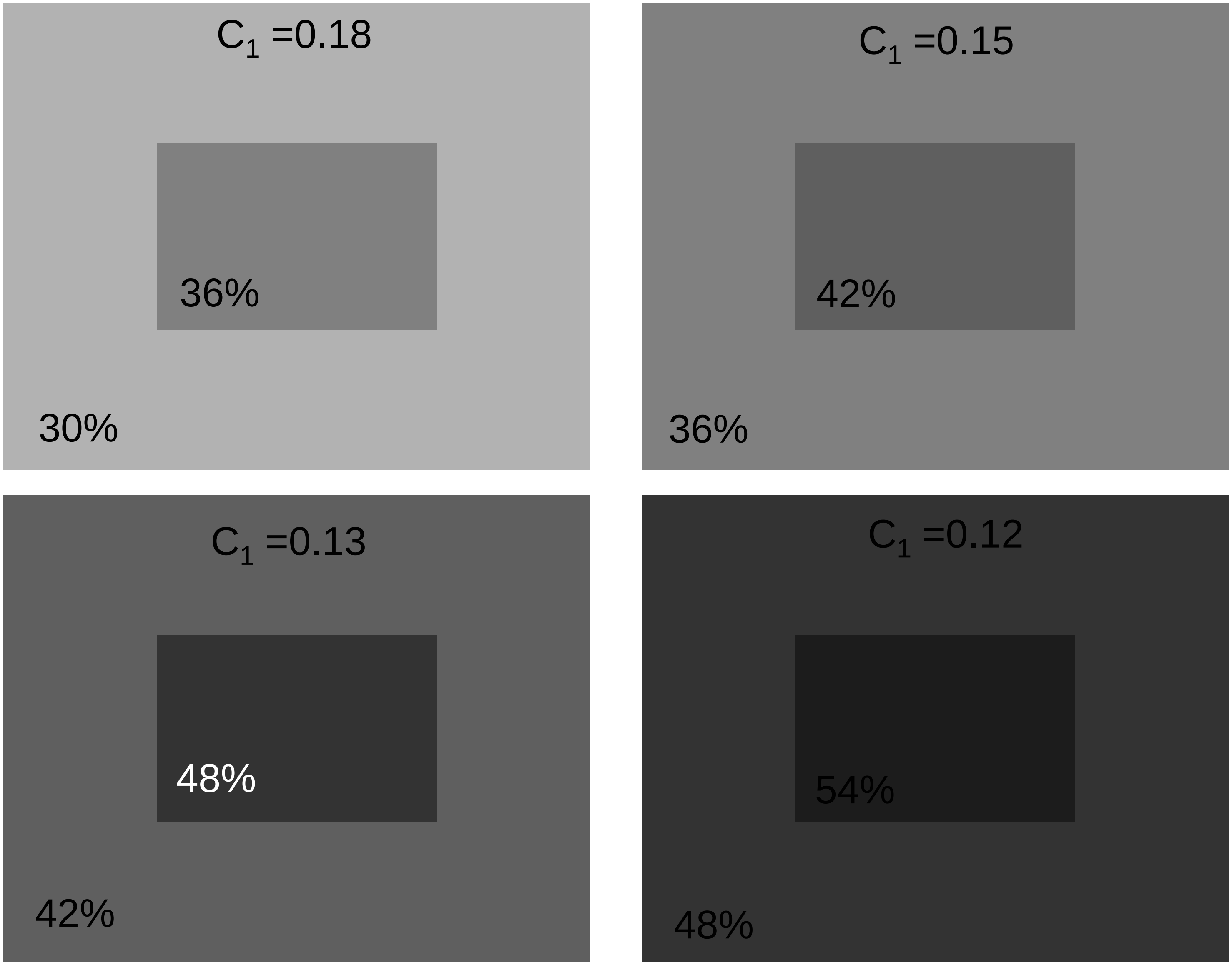
- Separation of signal (image) features from background
- Contrast describes relative brightness of a feature

$$\boxed{C_1 = \frac{S-b}{\frac{1}{2}(S+b)} \quad \Rightarrow \quad C_1 : -2 \ldots 2}$$

$$C_2 = \frac{S-b}{b} \quad \Rightarrow \quad C_2 : -1 \ldots \infty$$

$$C_3 = \frac{|S-b|}{S} \quad \Rightarrow \quad C_3 : 0 \ldots \infty$$

Contrast Values



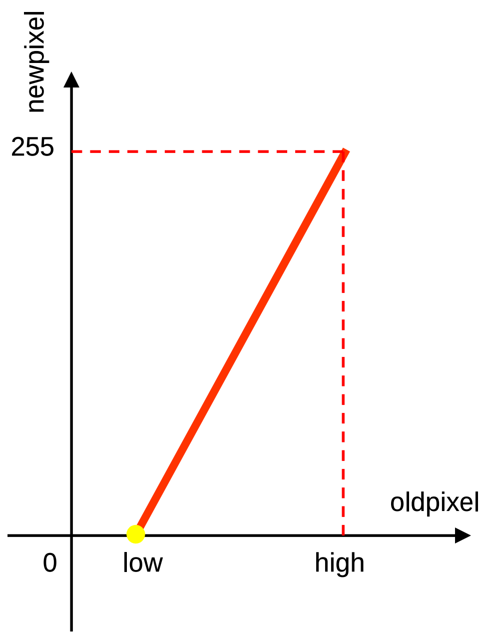
Contrast Stretching

- Contrast stretching is applied to an image to stretch a histogram to fill the full dynamic range of the image.
- It works best with images that have a Gaussian or near-Gaussian distribution.

Low and high contrast histograms

Contrast Stretching

- Basic Contrast Stretch
  - Expands the image histogram to cover all ranges (0~255) of pixels.



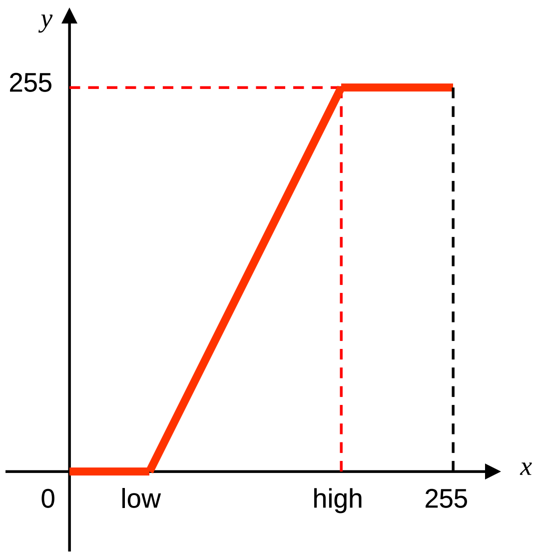
$$y - y_1 = a(x - x_1)$$

$$newpixel = \frac{255}{high - low}(oldpixel - low)$$

  - Works best on images that have all pixels concentrated in one part of the histogram, the middle.

Contrast Stretching

- End-In-Search
  - A certain percentage of the pixels must be saturated to full white or full black.

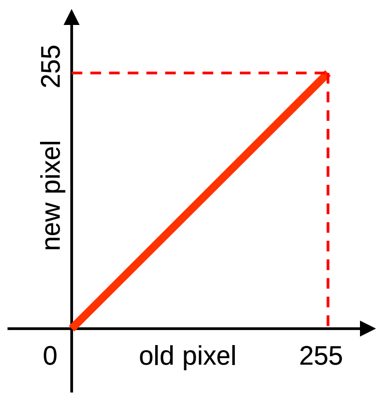


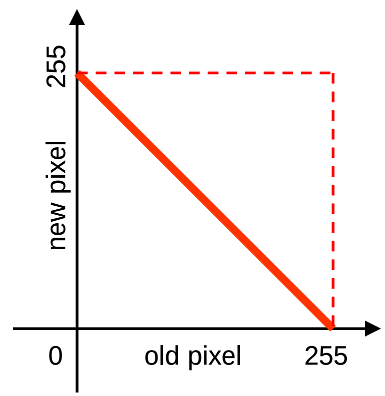
$$y = \begin{cases} 0 & \text{for} \quad x \le low \\ 255 \times \dfrac{(x - low)}{high - low} & \text{for} \quad \mathrm{low} \le \mathrm{x} \le \mathrm{high} \\ 255 & \text{for} \quad \mathrm{high} \le \mathrm{x} \end{cases}$$

  - Works well for images that have pixels of all possible intensities but have a pixel concentration in one part of the histogram.

Intensity Transformations

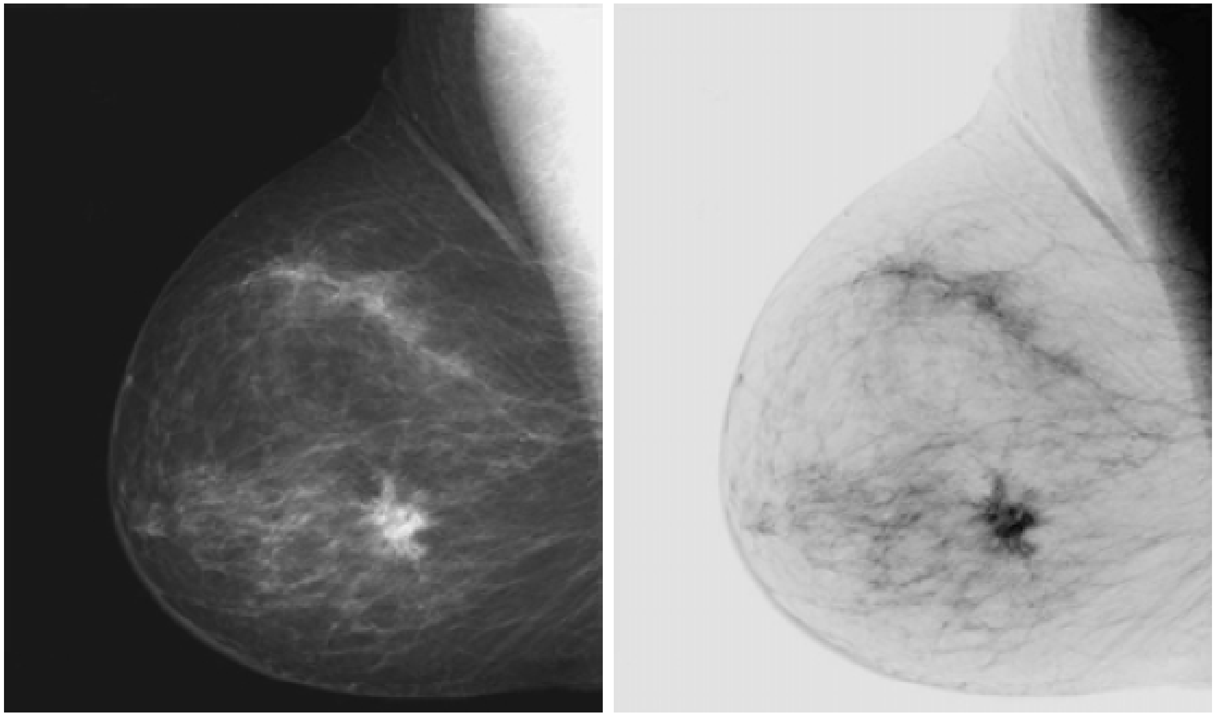
- Convert an old pixel into a new pixel based on some predefined function.
- Easily implemented with simple look-up tables.
  (ex: image negative)





Intensity Transformations

- Image Negative



a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Intensity Transformations

- Power-law:

Power-law transformations have the basic form

$$s = cr^{\gamma} \tag{3.2-3}$$

where c and γ are positive constants.

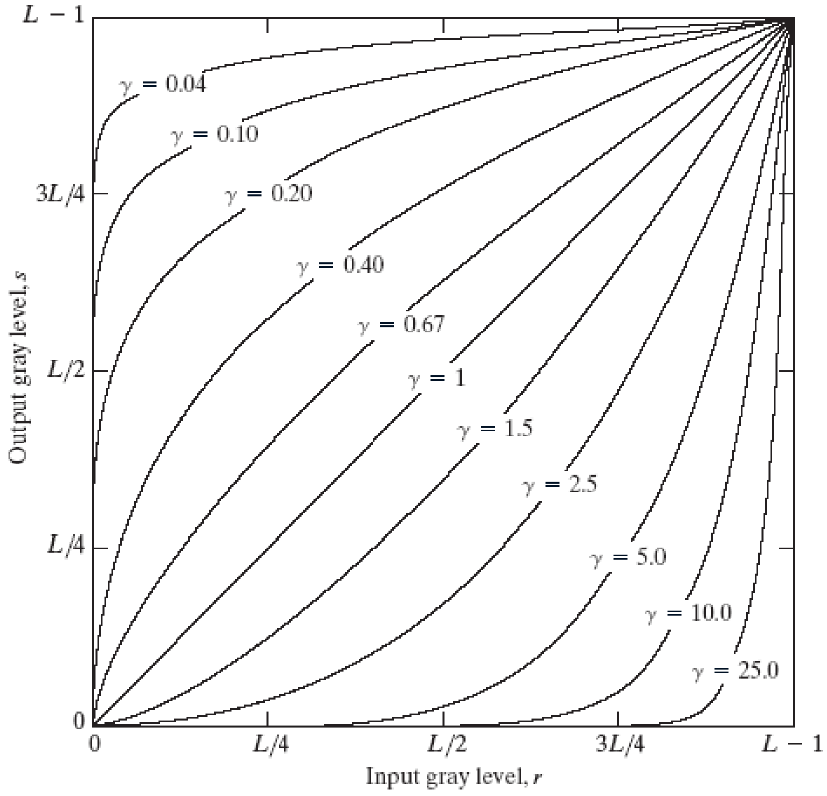


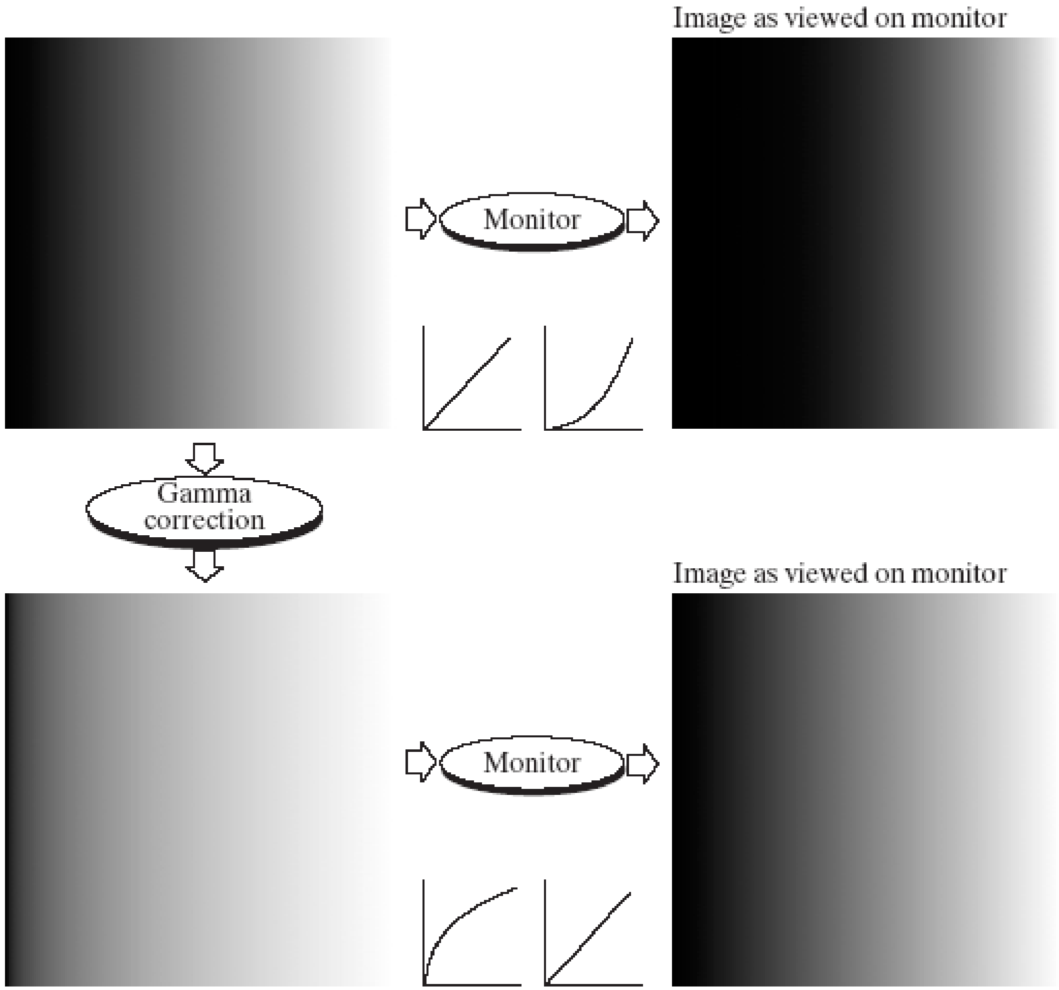
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Intensity Transformations

- Power-law:

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



Intensity Transformations

- Power-law:

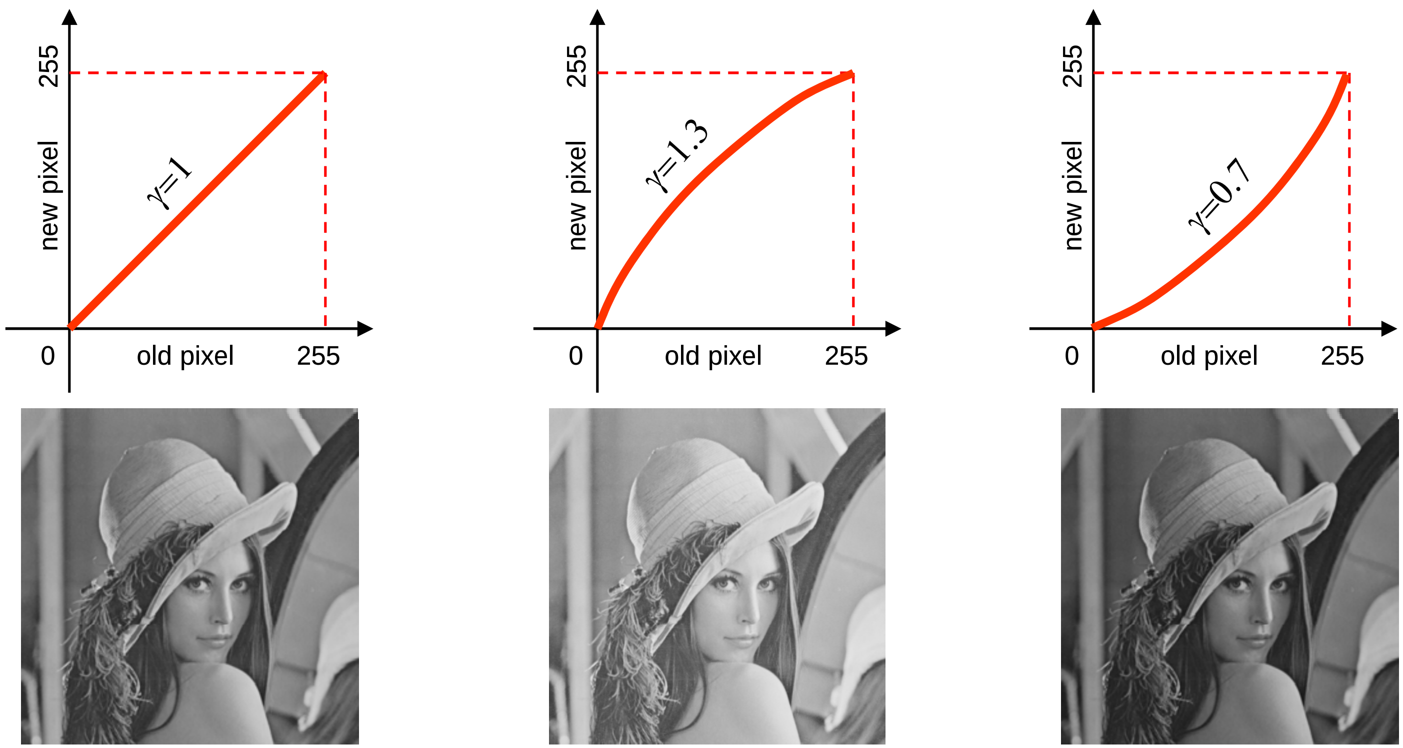
a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Intensity Transformations

- Gamma Correction
  Often used in image processing to compensate for nonlinear responses in imaging sensors, displays and film.

$$X \longrightarrow \boxed{\text{Gamma Correction}} \longrightarrow Y = 255 \times \left(\frac{X}{255} \right)^{\gamma}$$



Intensity Transformations

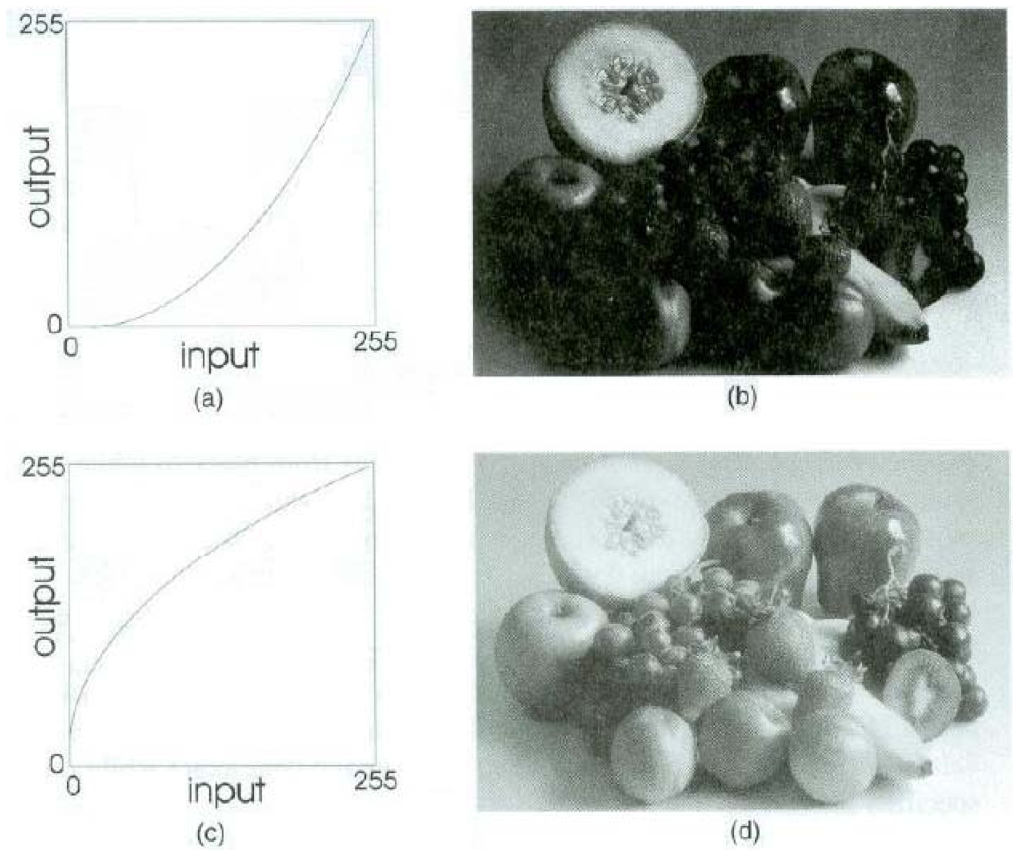
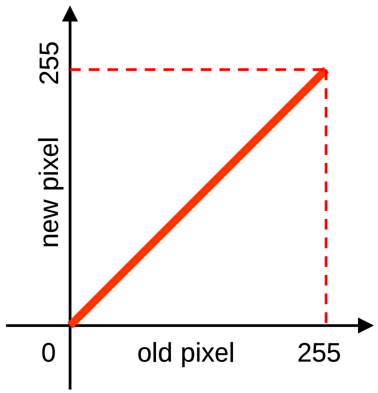


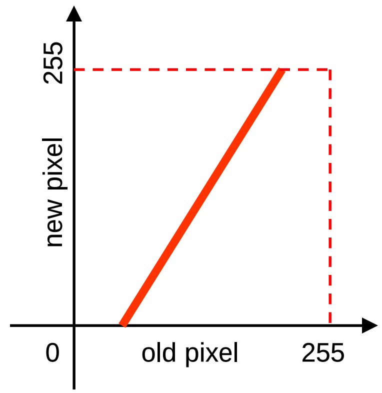
Fig. 2.16 (a) Gamma correction transformation with γ=0.45; (b) gamma corrected image; (c) gamma correction transformation with γ=2.2; (d) gamma corrected image.

Intensity Transformations

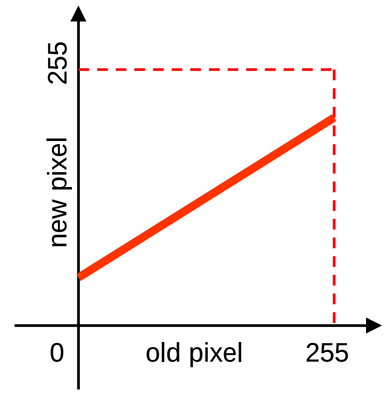
- Contrast Stretch/Compression Transformations

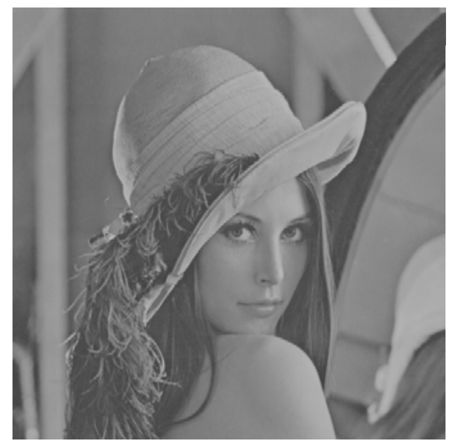


original



contrast stretched





contrast compressed

Intensity Transformations

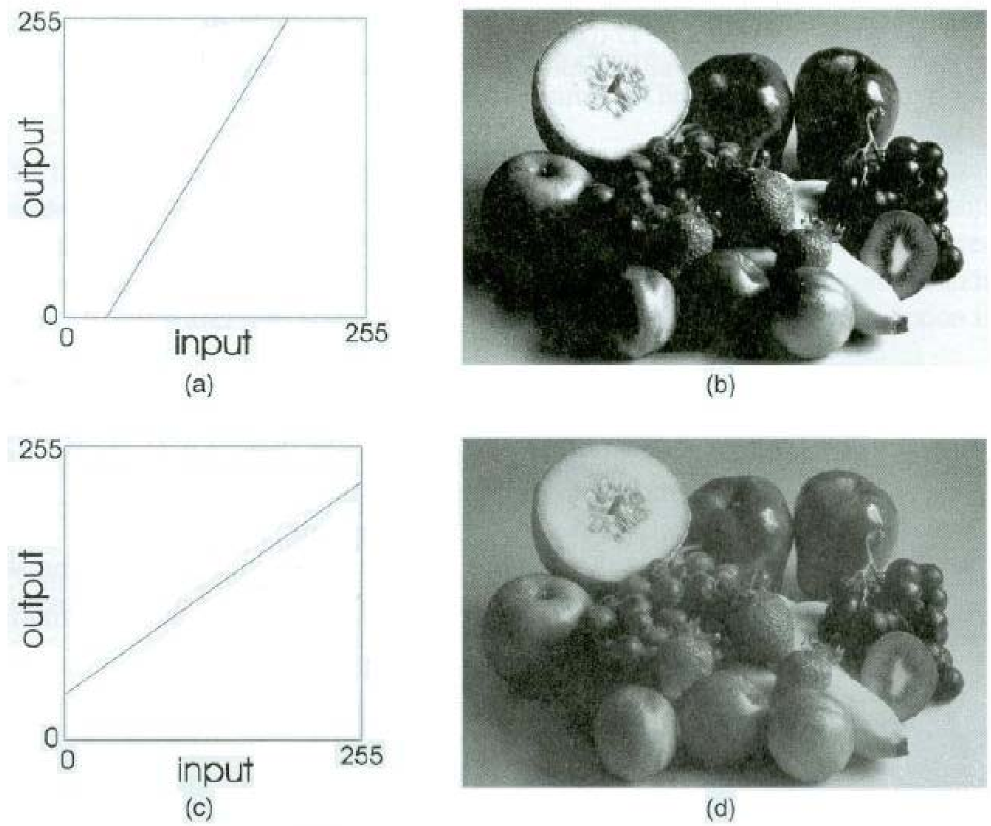
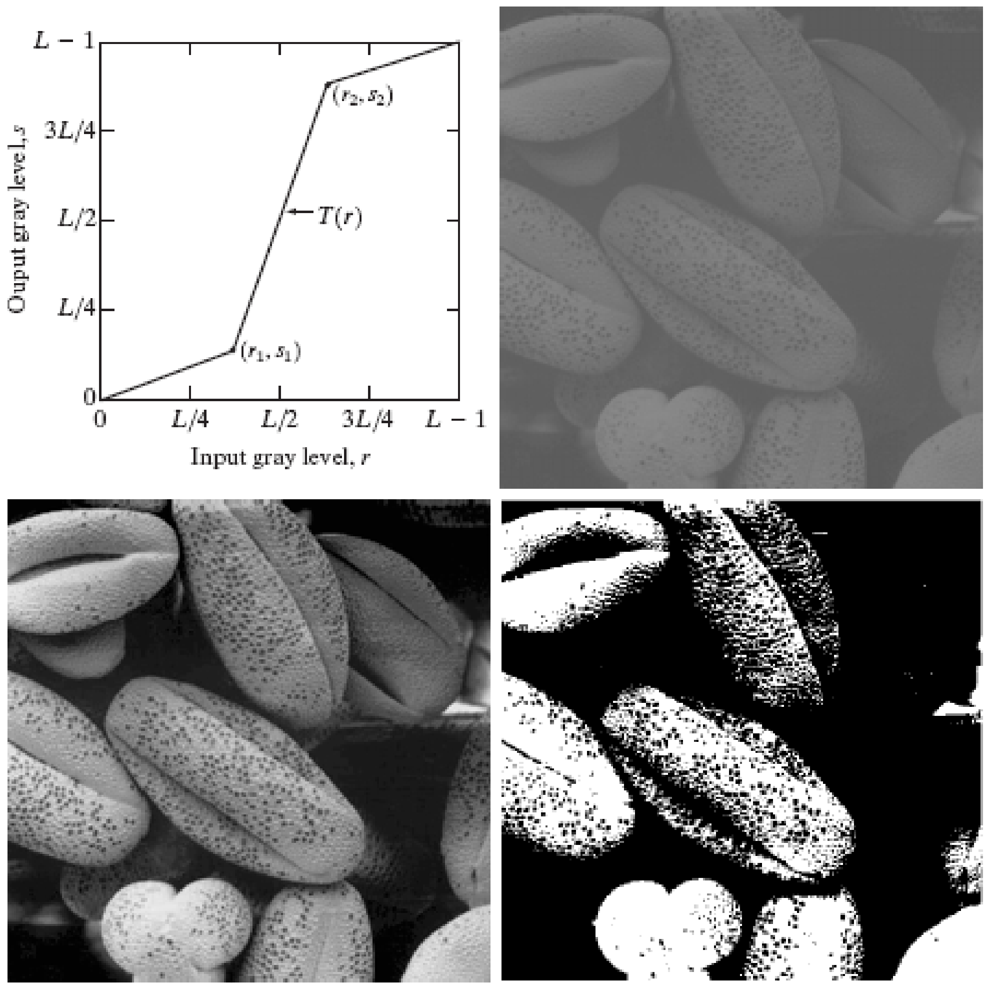


Fig. 2.17 (a) Contrast stretch transformation; (b) contrast stretched image; (c) Contrast compression transformation; (d) contrast compressed image.

Intensity Transformations



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Intensity Transformations

- Posterize Transformation
  Reduces the number of gray levels in an image

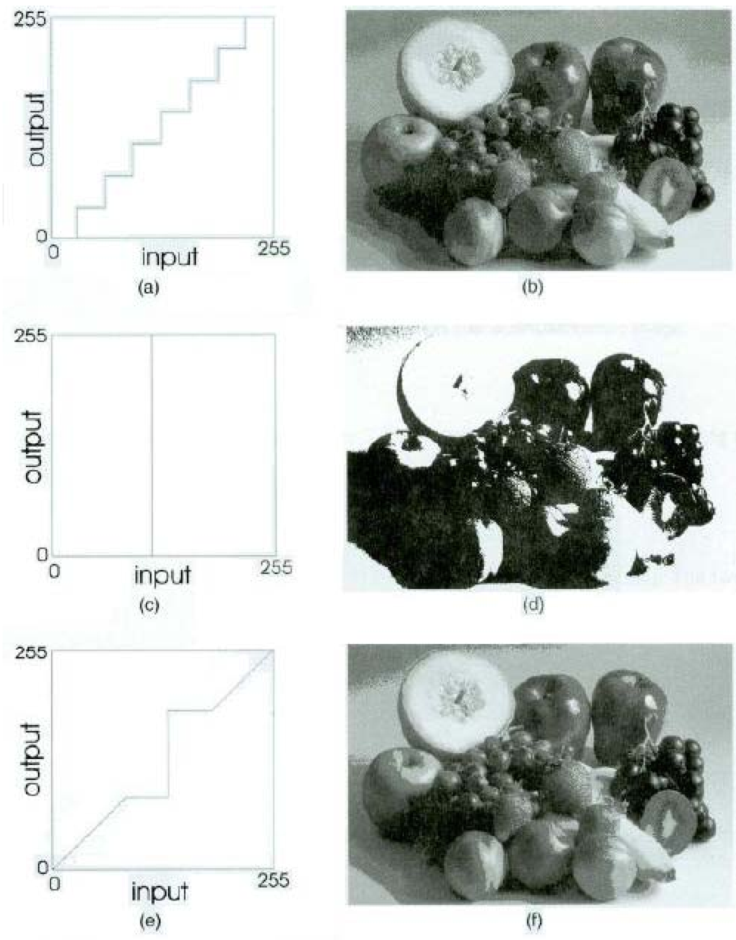


Fig. 2.18 (a) 8-Level posterize transformation;
(b) posterized image; (c) threshold transformation;
(d) threshold image; (e) bounded threshold;
(f) bounded threshold image.

Intensity Transformations

- Bit-Clipping
  - Sets a certain number of the most significant bits of a pixel to 0.
  - Effect of breaking up an image into several subregions with the same intensity cycles.

Setting 1 MSB to zero

000	000	0
001	001	1
010	010	2
011	011	3
100	000	0
101	001	1
110	010	2
111	011	3

Setting 2 MSB's to zero

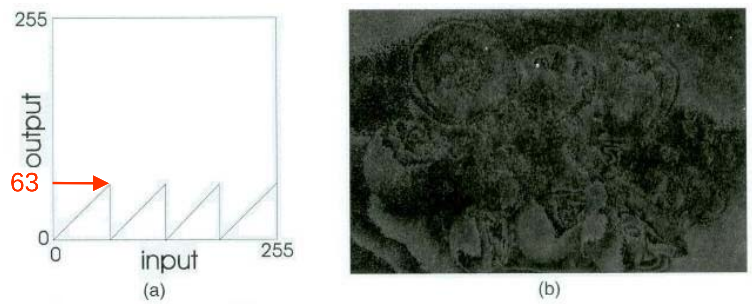


Fig. 2.19 (a) 2-bit bit-clipping transformation; (b) resulting image.

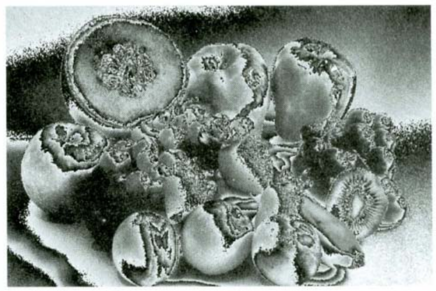


Fig. 2.20 Bit clipped image contrast stretched.

Intensity Transformations

- Iso-Intensity Transformation
  - Sets particular input intensity values to black or white.
  - Can be used to create contours on an image at specific intervals.

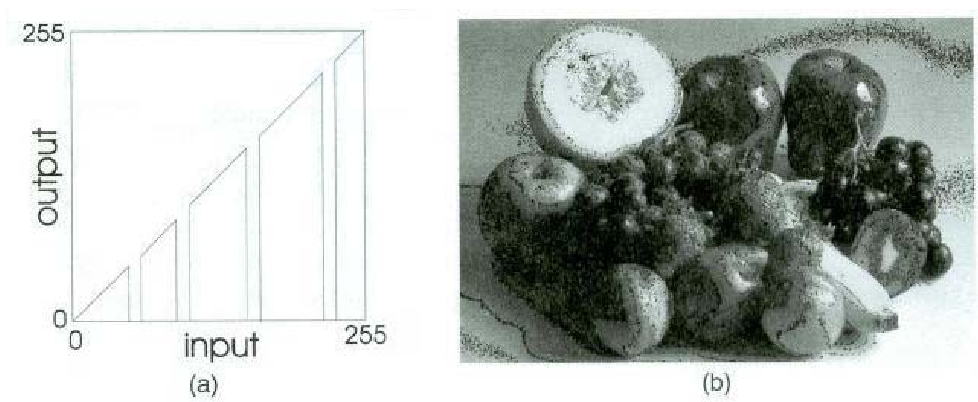


Fig. 2.21 (a) Iso-intensity contouring transformation; (b) contoured image.

Intensity Transformations

- Range-Highlighting Transformation

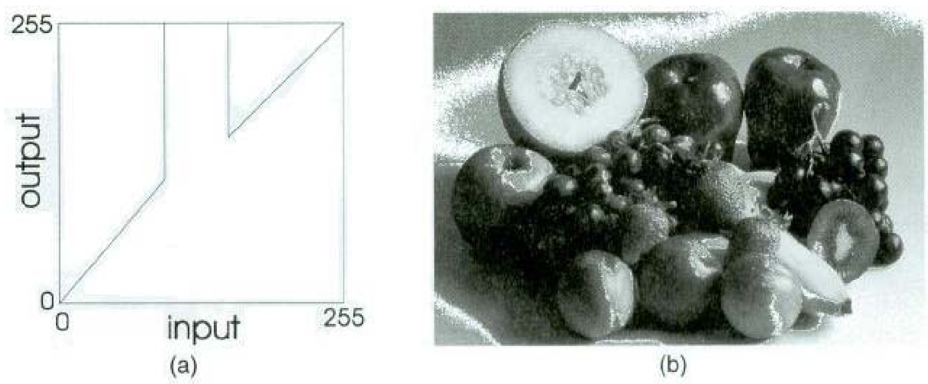


Fig. 2.22 (a) Range-highlighting transformation; (b) resulting image.

- Solarize Transformation

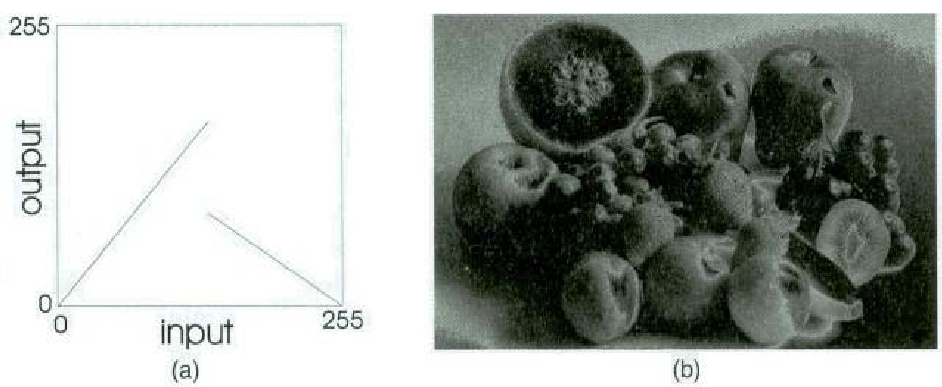


Fig. 2.23 (a) Solarize transformation using a threshold of 150; (b) solarized image.

Intensity Transformations

- Parabola Transformations

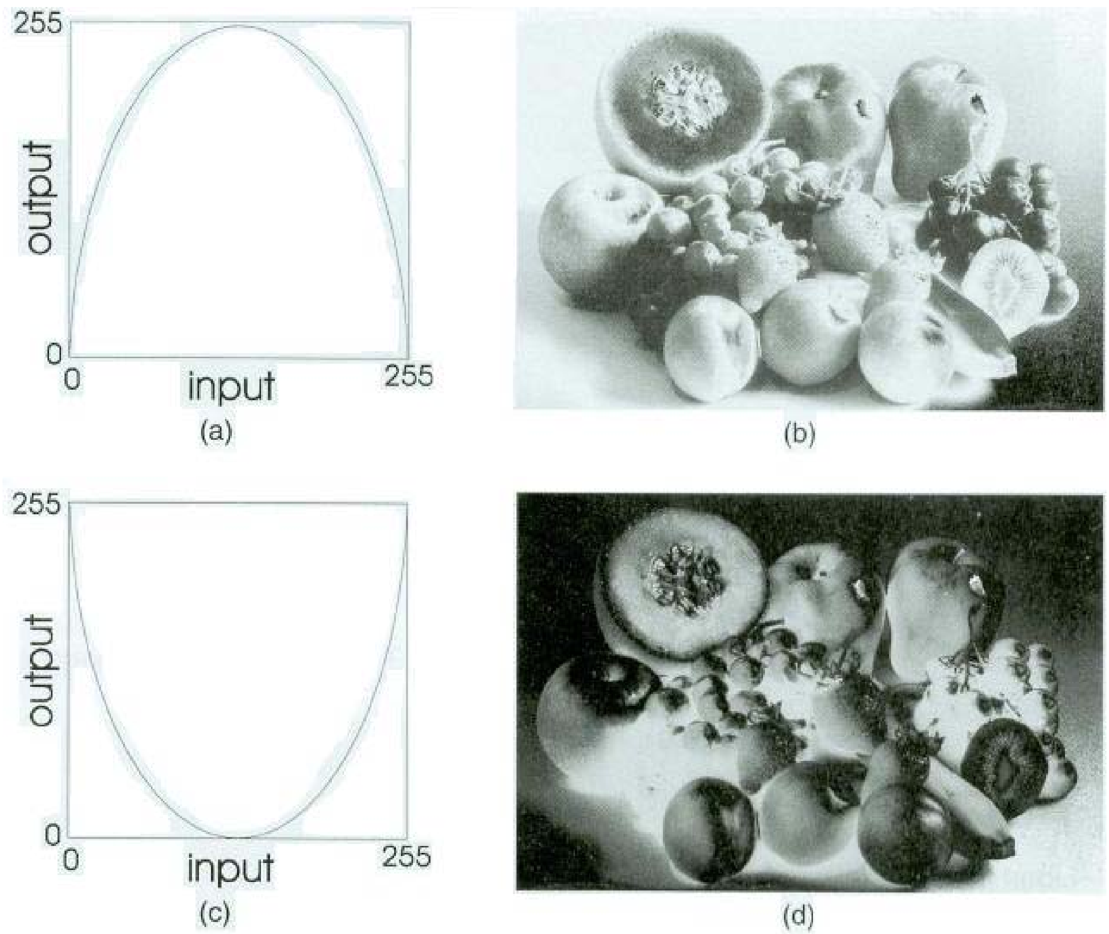


Fig. 2.24 (a) First parabola transformation; (b) transformed image; (c) second parabola transformation; (d) second transformed image.