Basic Image Processing on Gray Level Images

1. Thresholding Gray Level Images

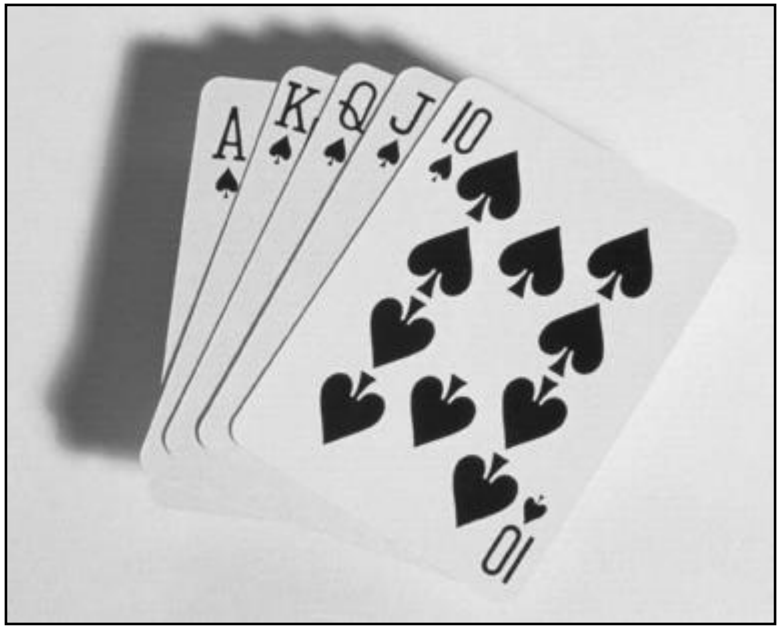
- The simplest approach to segment an image is using thresholding （選取閾值）.

- Single value thresholding can be given mathematically as follows:

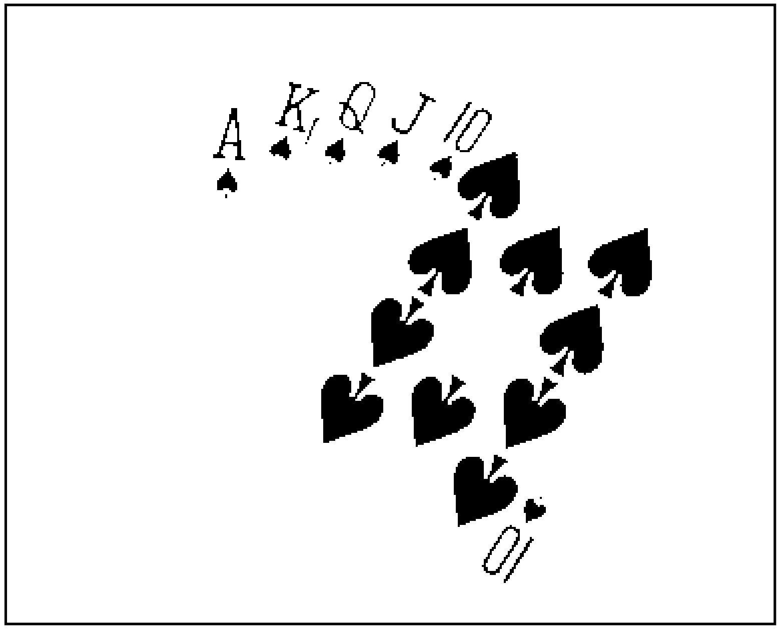
$$g(x,y)=\begin{cases} 1\, if\ \ f(x,y)>T \\ 0\, if\ \ f(x,y)\leq T \end{cases}$$

Thresholding Example

- Imagine a poker playing robot that needs to visually interpret the cards in its hand

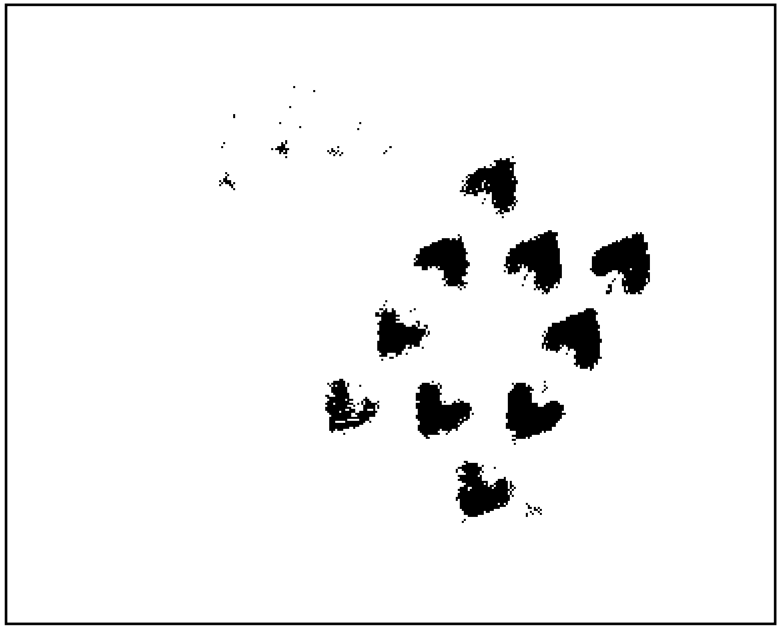


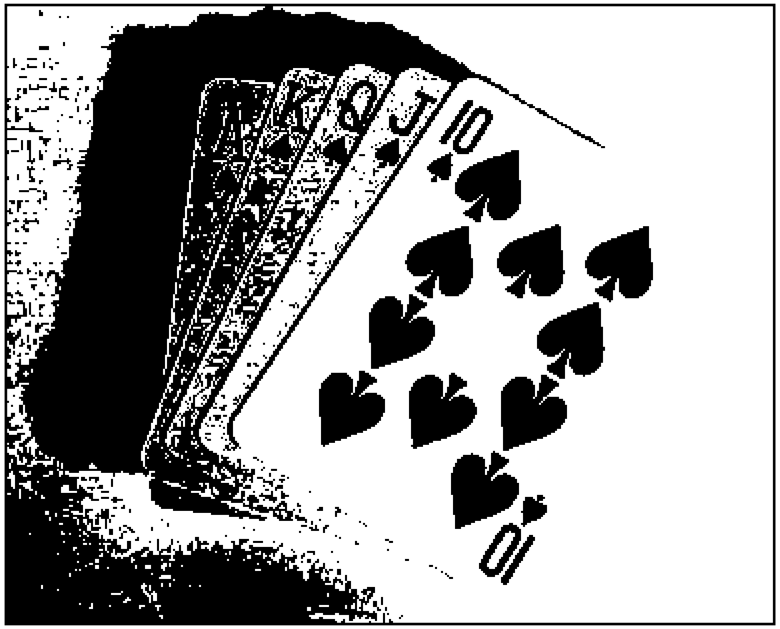
Original Image



Thresholded Image

- But Be Careful , if you get the threshold wrong the results can be disastrous


Threshold Too Low



Threshold Too High

Basic Global Thresholding

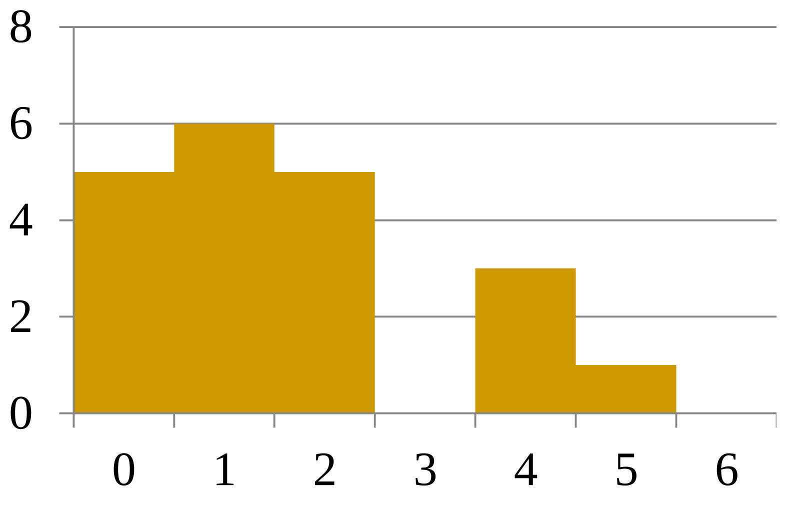
- Based on the histogram of an image
- Partition the image histogram using a single global threshold
- The success of this technique very strongly depends on how well the histogram can be partitioned

Histograms

- The (intensity or brightness) histogram shows how many times a particular grey level (intensity) appears in an image.
- For example, 0 - black, 255 – white

0	1	1	2	4
2	1	0	0	2
5	2	0	0	4
1	1	2	4	1

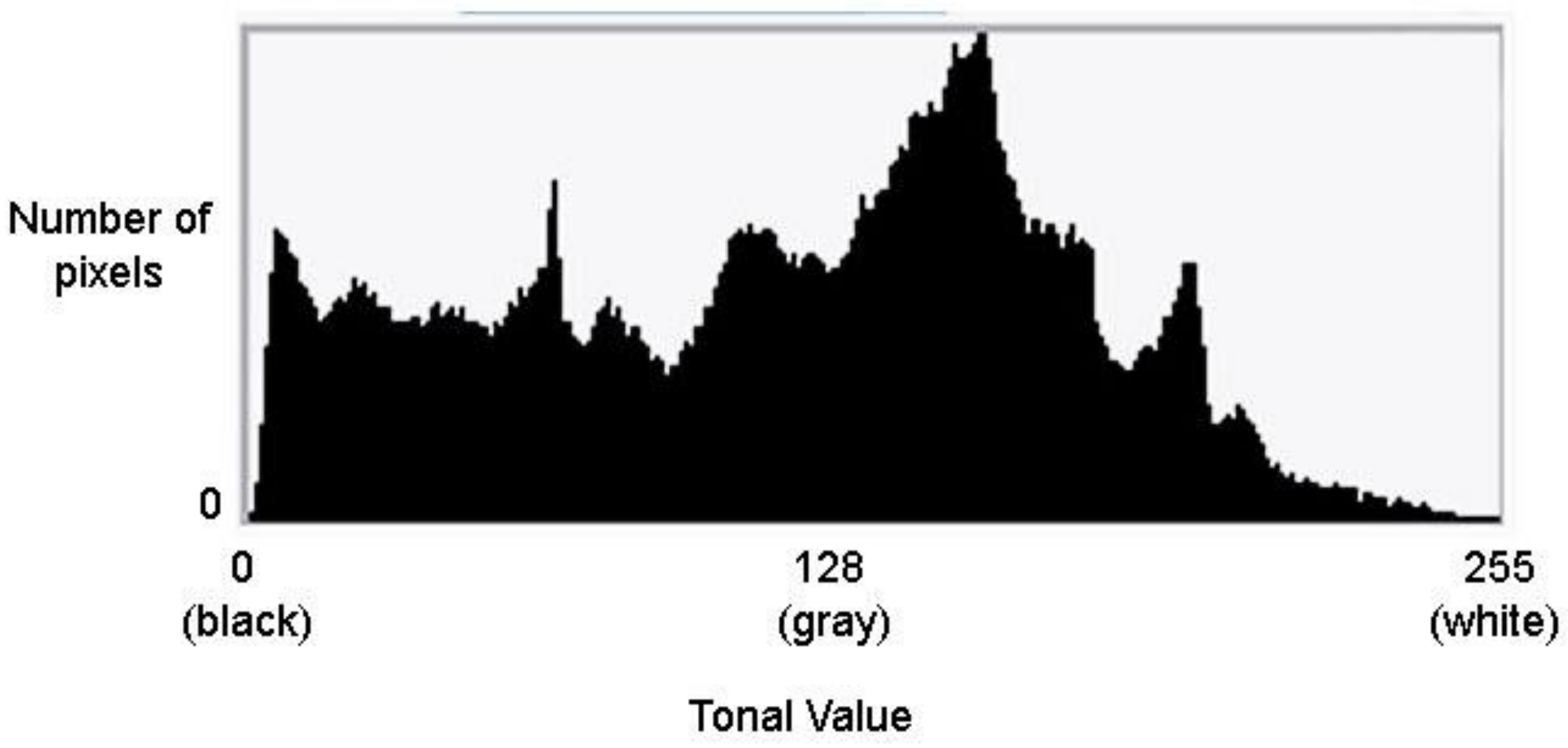
Image



Histogram

What is a histogram?

- A histogram is a “bar chart”.



This bar chart has 256 bars, moved together so there is no space between the bars.

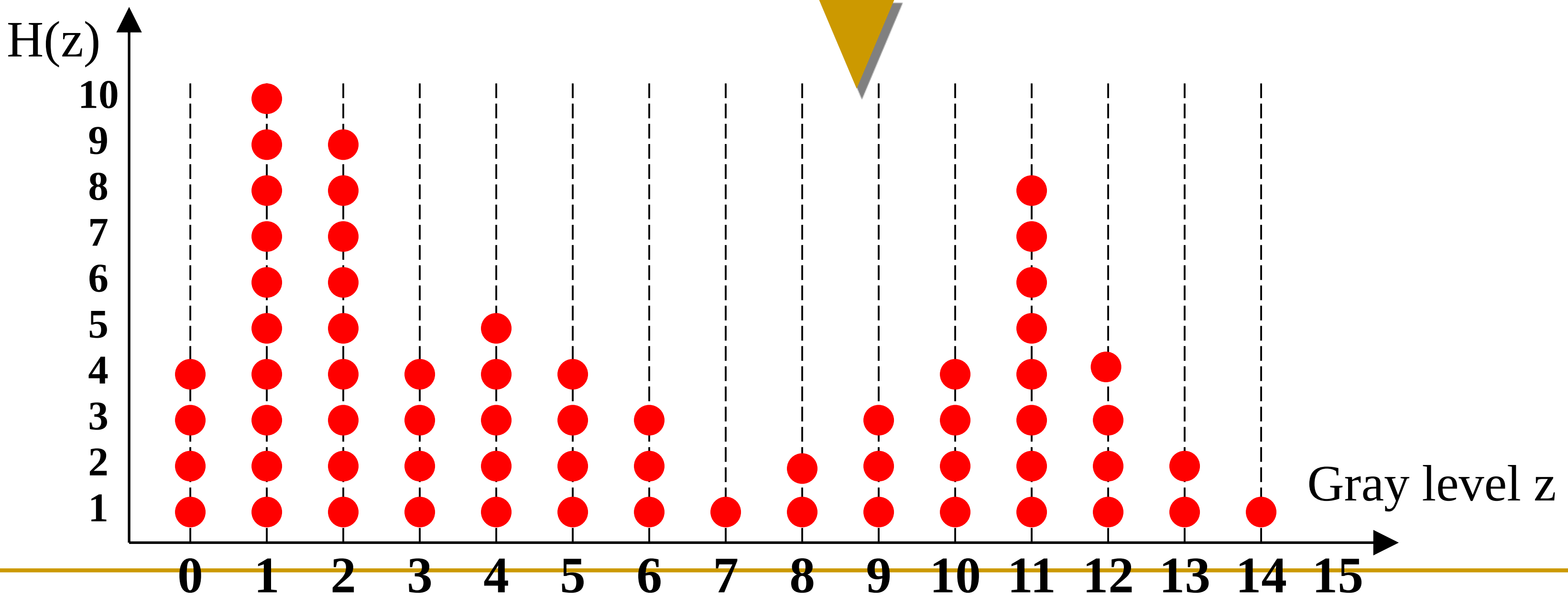
- Histogram are constructed by splitting the range of the data into equal-sized bins (called classes). Then for each bin, the number of points from the data set that fall into each bin are counted.
- Vertical axis: Frequency (i.e., counts for each bin)
- Horizontal axis: Response variable
- In image histograms the pixels form the horizontal axis

- It gives the distribution of the gray-level intensities over the image without reference to their locations, only to their frequencies of occurrence

1	1	1	9	9	0	0	12
1	1	2	2	2	9	1	12
12	6	4	8	0	5	1	1
4	4	2	3	2	6	1	1
4	0	3	7	3	5	5	13
10	10	4	2	3	5	6	13
10	2	10	2	2	8	11	11
11	11	11	11	11	11	12	14

Image B

$$f(x, y) = \begin{pmatrix} 1 & 1 & 1 & 9 & 9 & 0 & 0 & 12 \\ 1 & 1 & 2 & 2 & 2 & 9 & 1 & 12 \\ 12 & 6 & 4 & 8 & 0 & 5 & 1 & 1 \\ 4 & 4 & 2 & 3 & 2 & 6 & 1 & 1 \\ 4 & 0 & 3 & 7 & 3 & 5 & 5 & 13 \\ 10 & 10 & 4 & 2 & 3 & 5 & 6 & 13 \\ 10 & 2 & 10 & 2 & 2 & 8 & 11 & 11 \\ 11 & 11 & 11 & 11 & 11 & 11 & 12 & 14 \end{pmatrix}$$

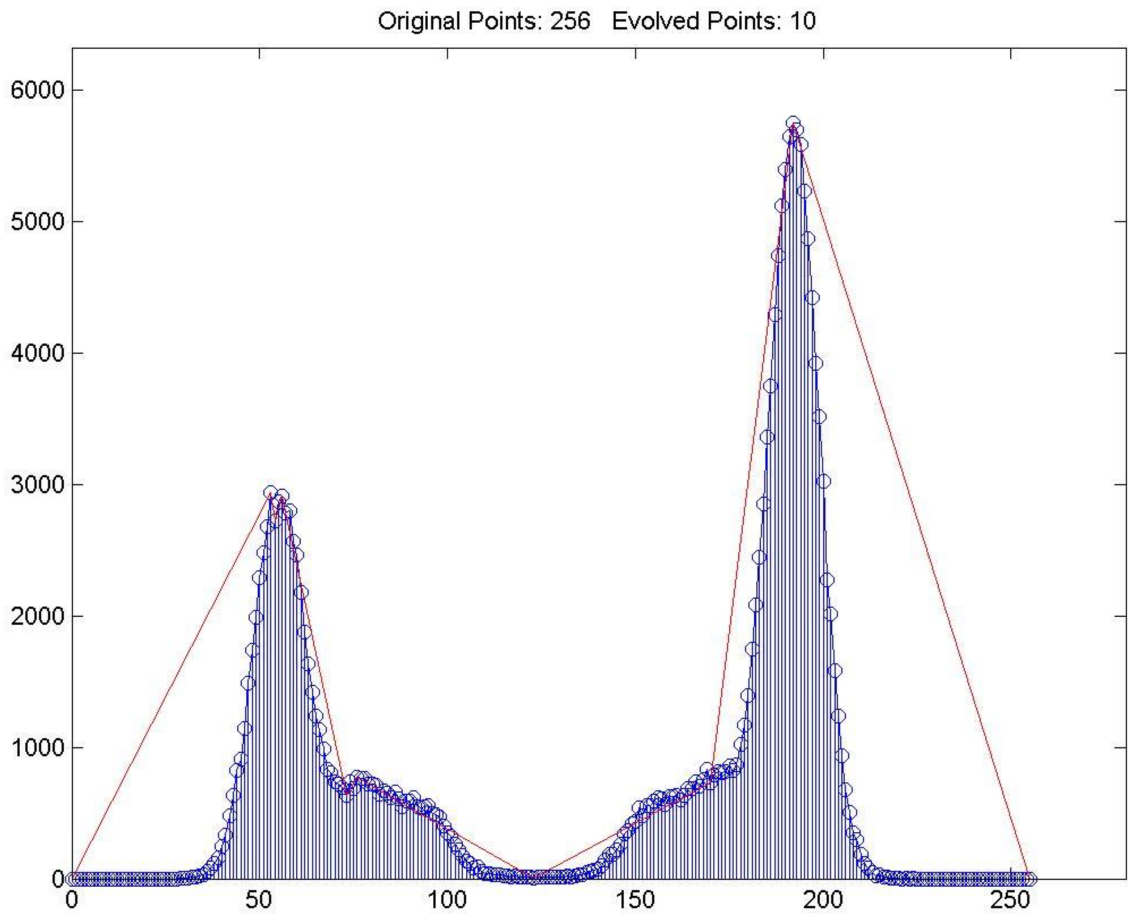


Algorithm – Image histogram

- **Compute the histogram H of gray-level image I.**

```
procedure histogram(I, H);
{
"Initialize the bins of the histogram to zero."
for i : = 0 to MaxVal H[i] := 0;
"Compute values by accumulation."
for L : = 0 to MaxRow
    for P:= 0 to MaxCol
   { grayval :=I[L,P];
       H[grayval]:=H[grayval] + 1;
    }
}
```

- Suppose that the gray-level histogram corresponds to an image, f(x,y), composed of dark objects in a light background, in such a way that object and background pixels have gray levels grouped into two dominant modes.


Original Points: 256 Evolved Points: 10
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

- If two dominant modes characterize the image histogram, it is called a ***bimodal histogram*** **(雙峰)**. Only one threshold is enough for partitioning the image.

Basic Global Thresholding Algorithm

- The basic global threshold, T, is calculated

as follows:

1. Select an initial estimate for T (typically the average grey level in the image)
2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels $>T$ and G_2 consisting pixels with grey levels $\leq T$
3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

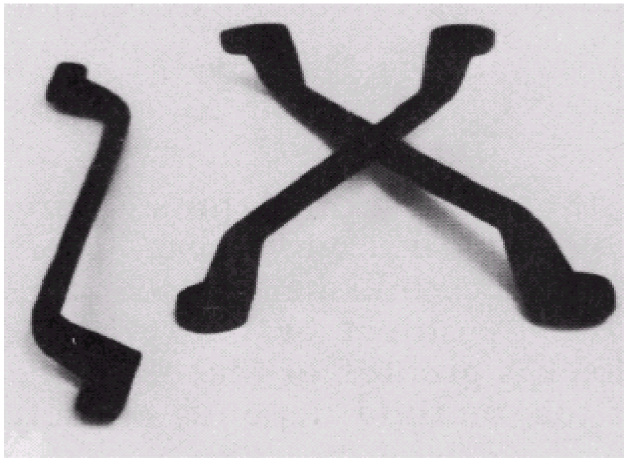
4. Compute a new threshold value:

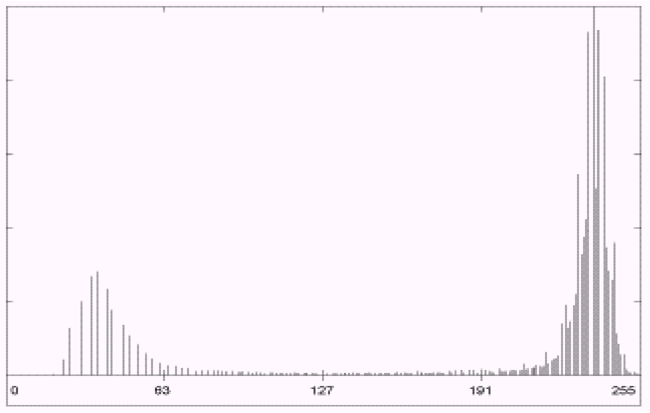
$$T = \frac{\mu_1 + \mu_2}{2}$$

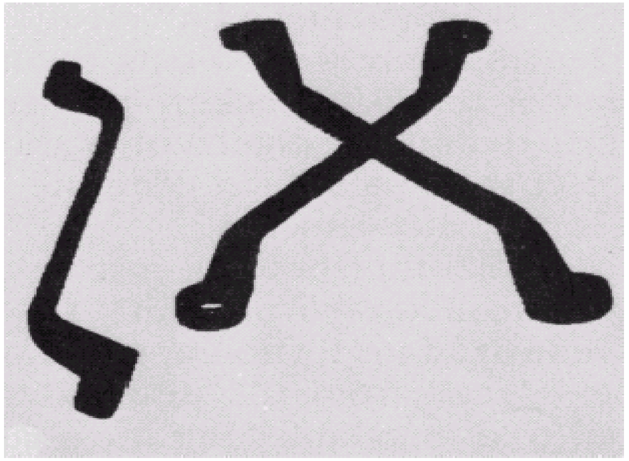
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

- This algorithm works very well for finding thresholds when the histogram is suitable

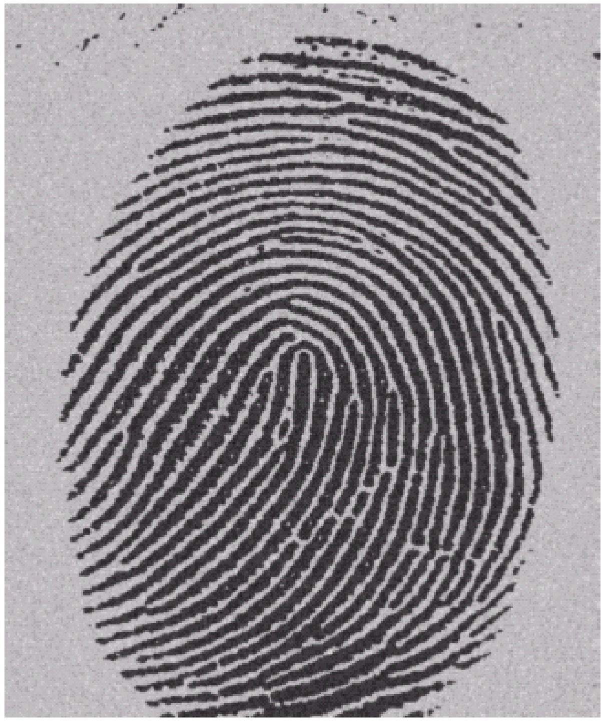
Thresholding Example 1

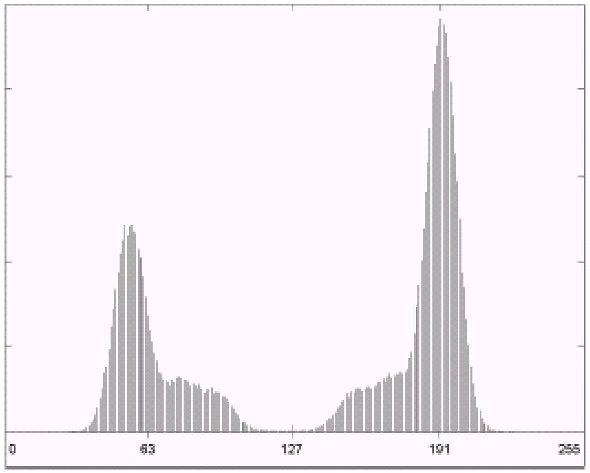


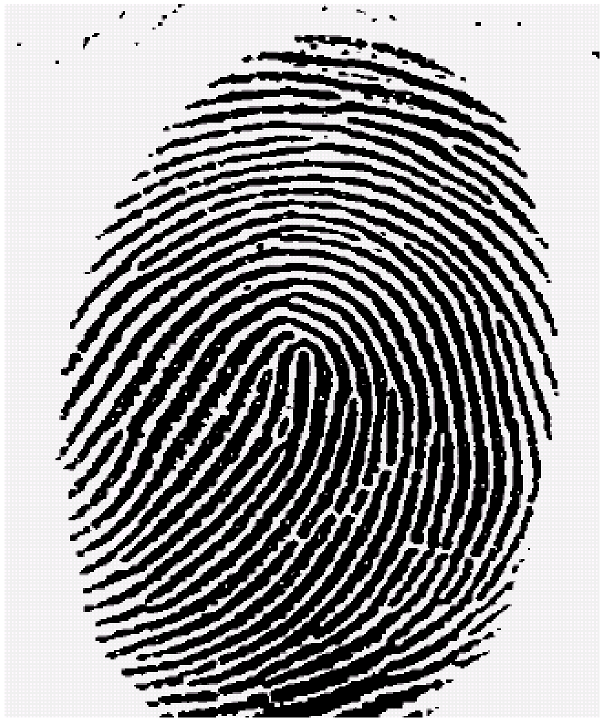




Thresholding Example 2

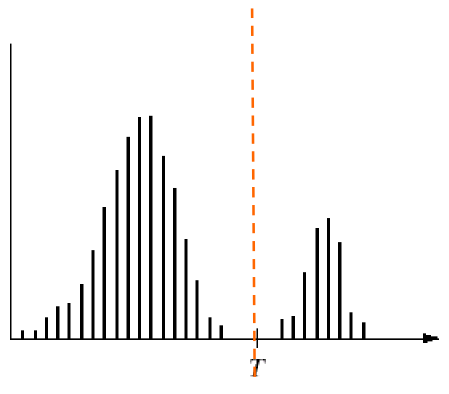


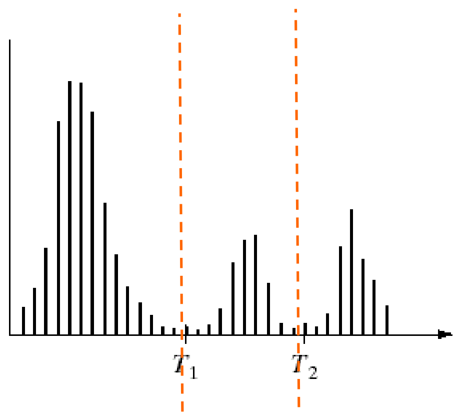




Problems With Single Value Thresholding

- Single value thresholding only works for bimodal histograms
- Images with other kinds of histograms need more than a single threshold





- If for example an image is composed of two types of light objects on a dark background, three or more dominant modes characterize the image histogram

- In such a case the histogram has to be partitioned by multiple thresholds.
- Multilevel thresholding classifies a point (x,y) as
  - belonging to one object class, if $T1 < f(x,y) <= T2$,
  - to the other object class, if $f(x,y) > T2$
  - and to the background, if $f(x,y) <= T1$.

Automatic thresholding

- To make segmentation more robust, the threshold should be automatically selected by the system.

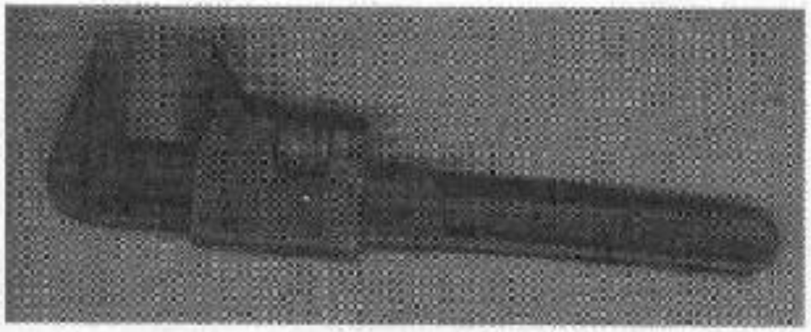
- In general, good threshold can be selected if the histogram peaks are tall, narrow, symmetric, and separated by deep valleys.

- Knowledge about the objects, the application, the environment should be used to choose the threshold automatically:
  - Intensity characteristics of the objects
  - Sizes of the objects
  - Fractions of an image occupied by the objects
  - Number of different types of objects appearing in an image

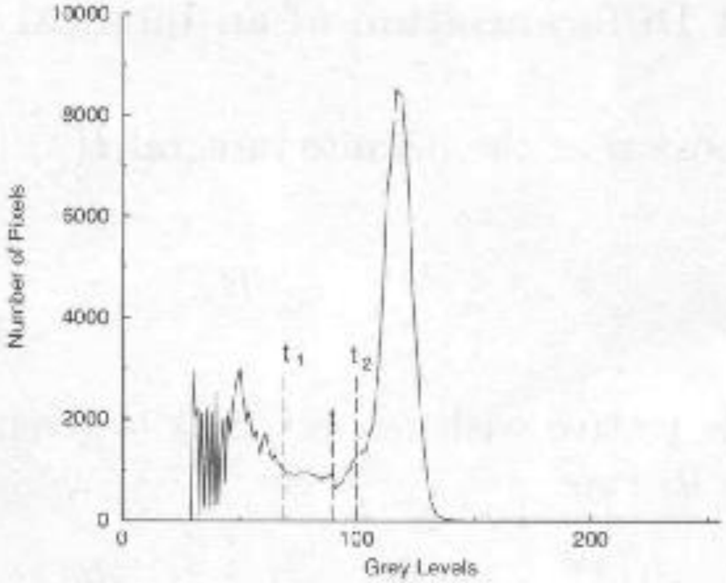
Hysteresis thresholding

- If there is no clear valley in the histogram of an image, it means that there are several background pixels that have similar gray level value with object pixels and vice versa.

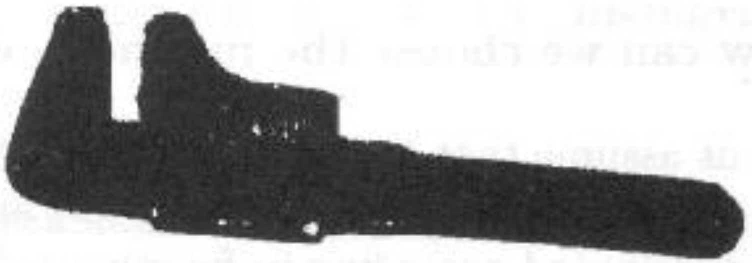
- Hysteresis thresholding (i.e., two thresholds, one at each side of the valley) can be used in this case.



(a) Original image



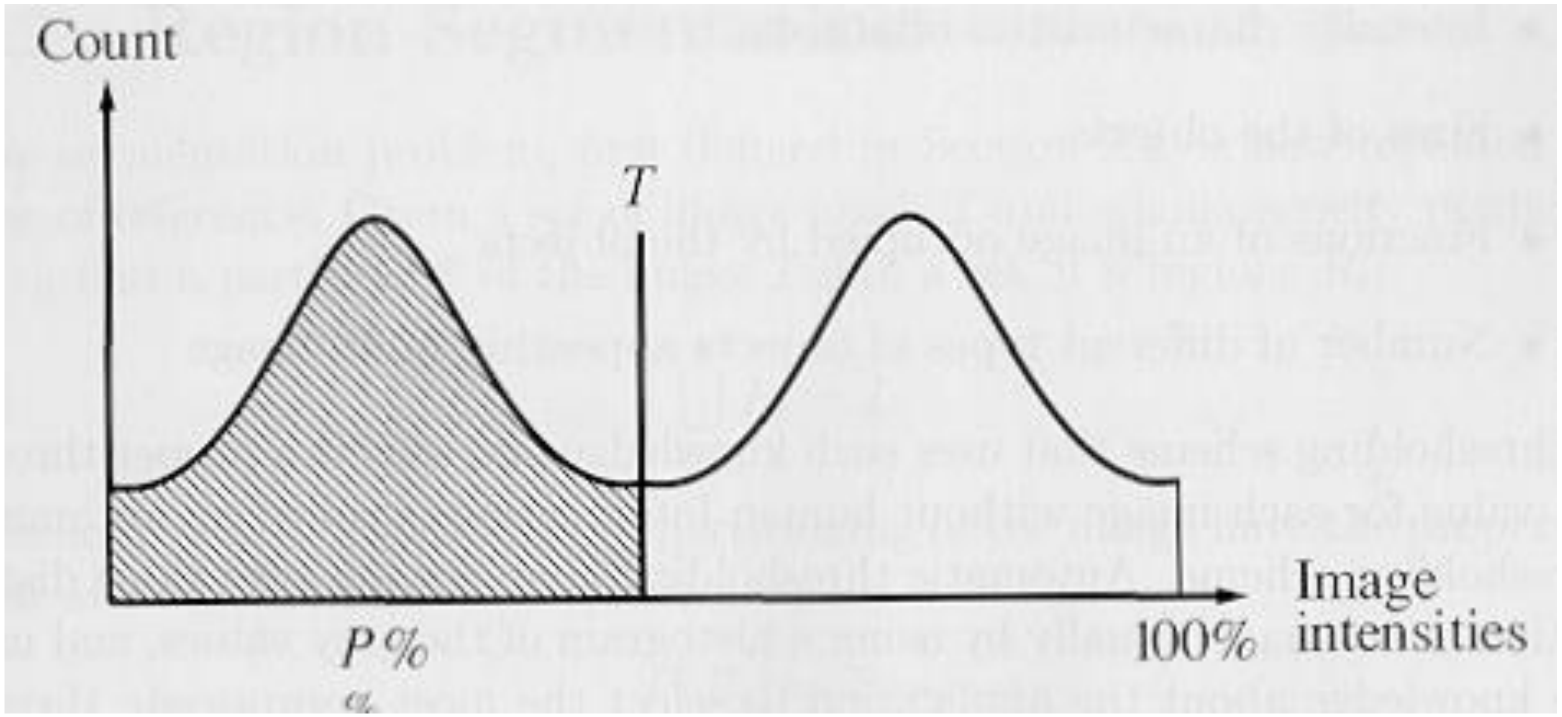
(b) Histogram of (a)



- Pixels above the high threshold are classified as object and below the low threshold as background.
- Pixels between the low and high thresholds are classified as object only if they are adjacent to other object pixels.

Using prior knowledge: the P-Tile method

- This method requires knowledge about the area or size of the objects present in the image
  - Let us assume that we have dark objects against a light background.
  - If, for example, the objects occupy *p*% of the image area, an appropriate threshold can be chosen by partitioning the histogram


Count
T
Image intensities
P %
100%

Examples

- Gray Scale Image - bimodal

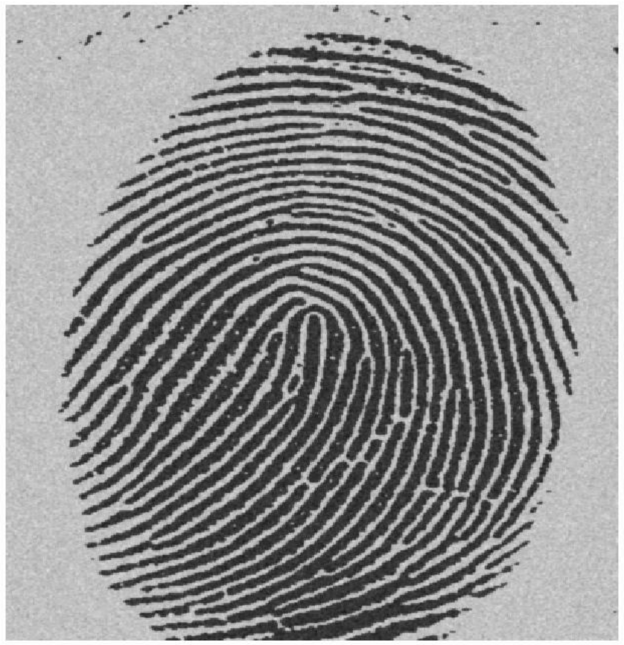


Image of a Finger Print with light background

Bimodal - Histogram

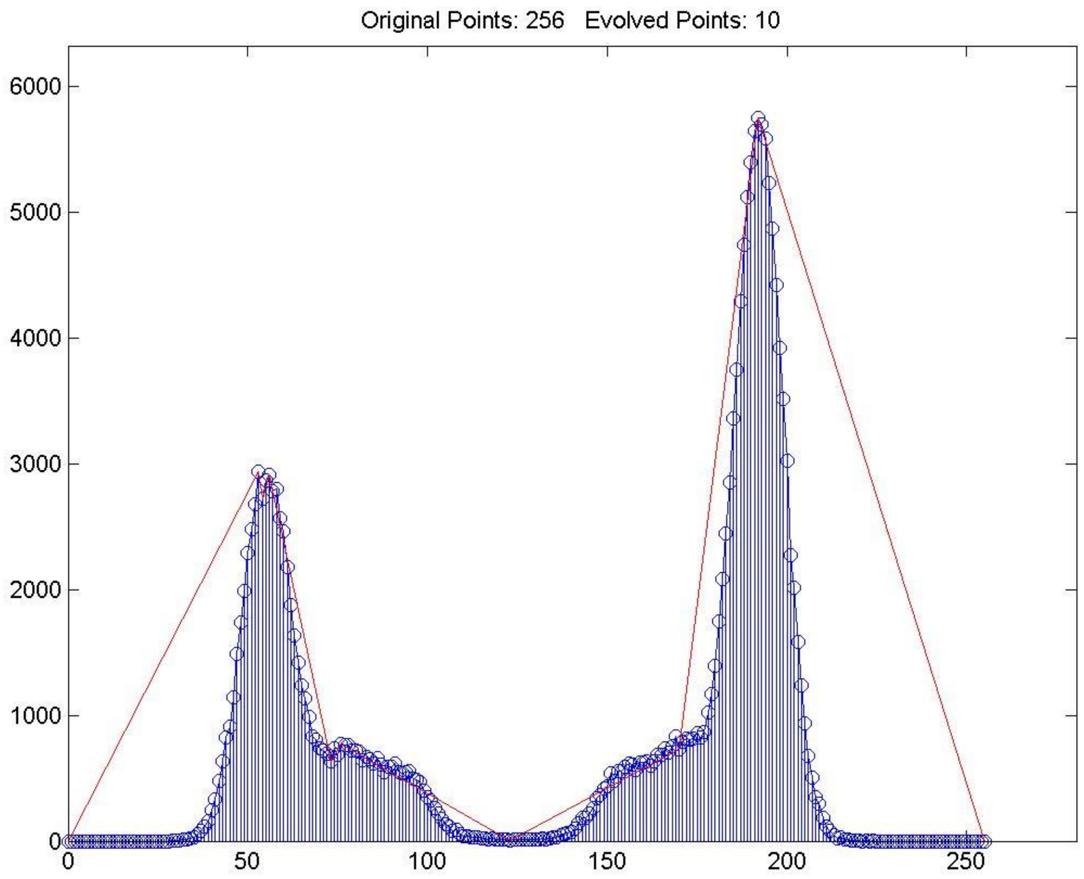
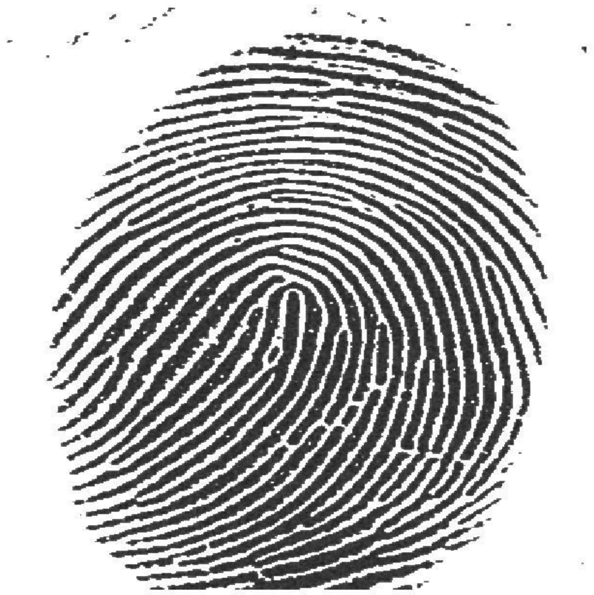


Image Histogram of finger print

Segmented Image



Gray Scale Image (2) - bimodal

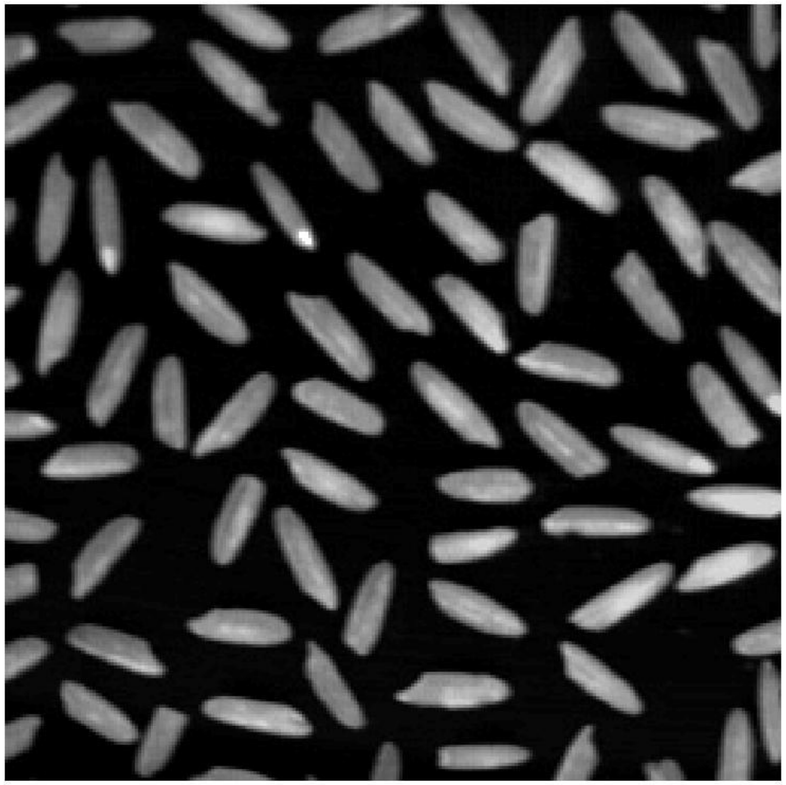


Image of rice with black background

Histogram

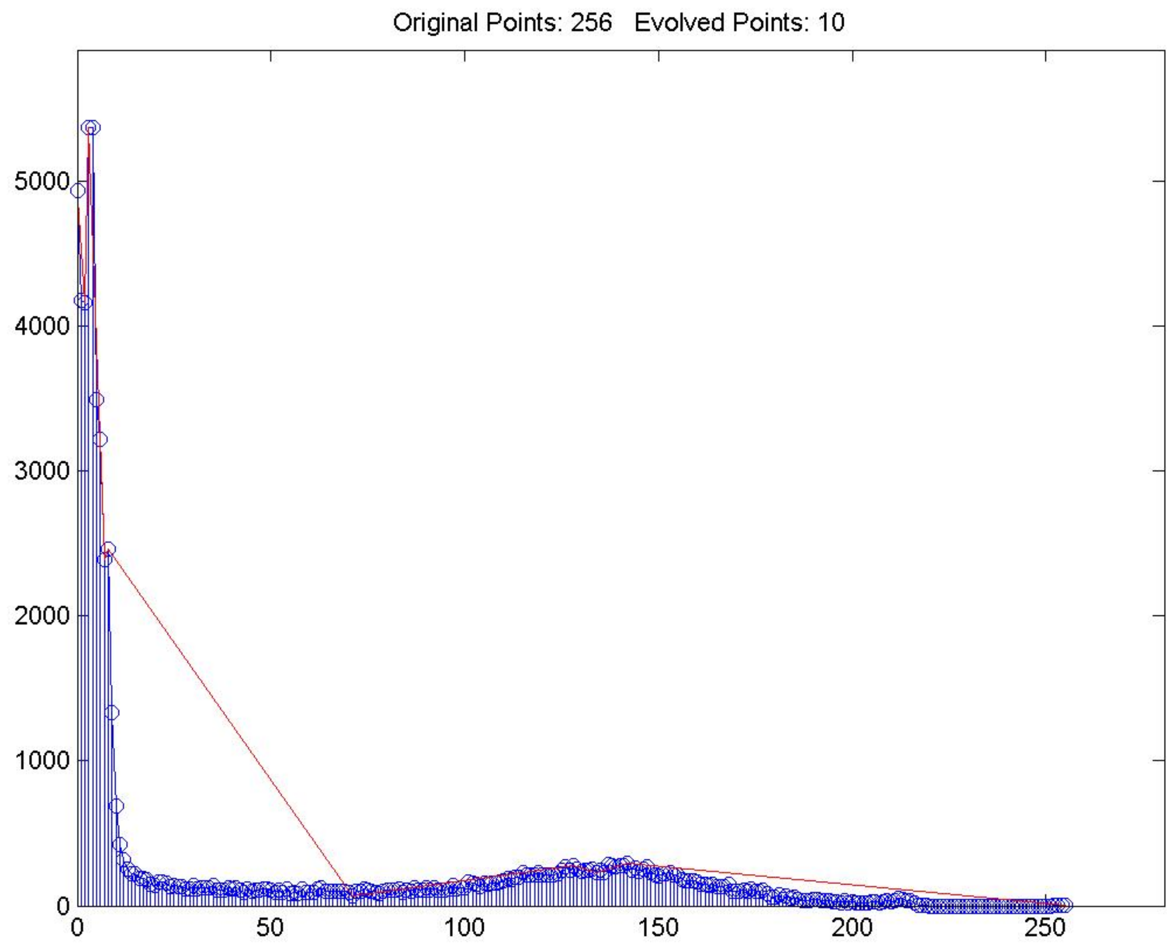
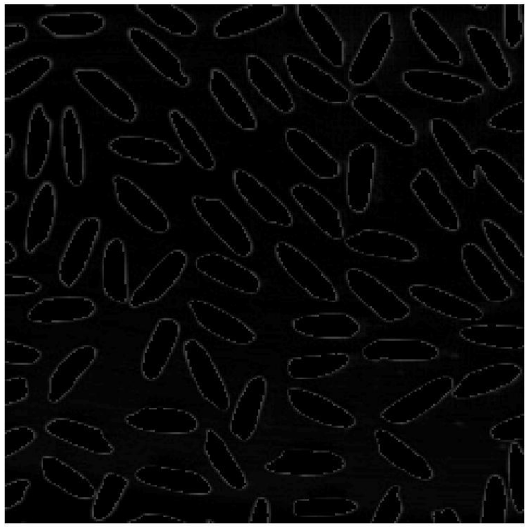
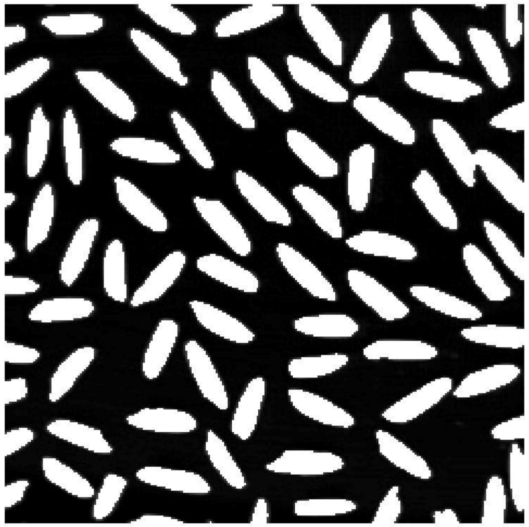


Image histogram of rice

Segmented Image

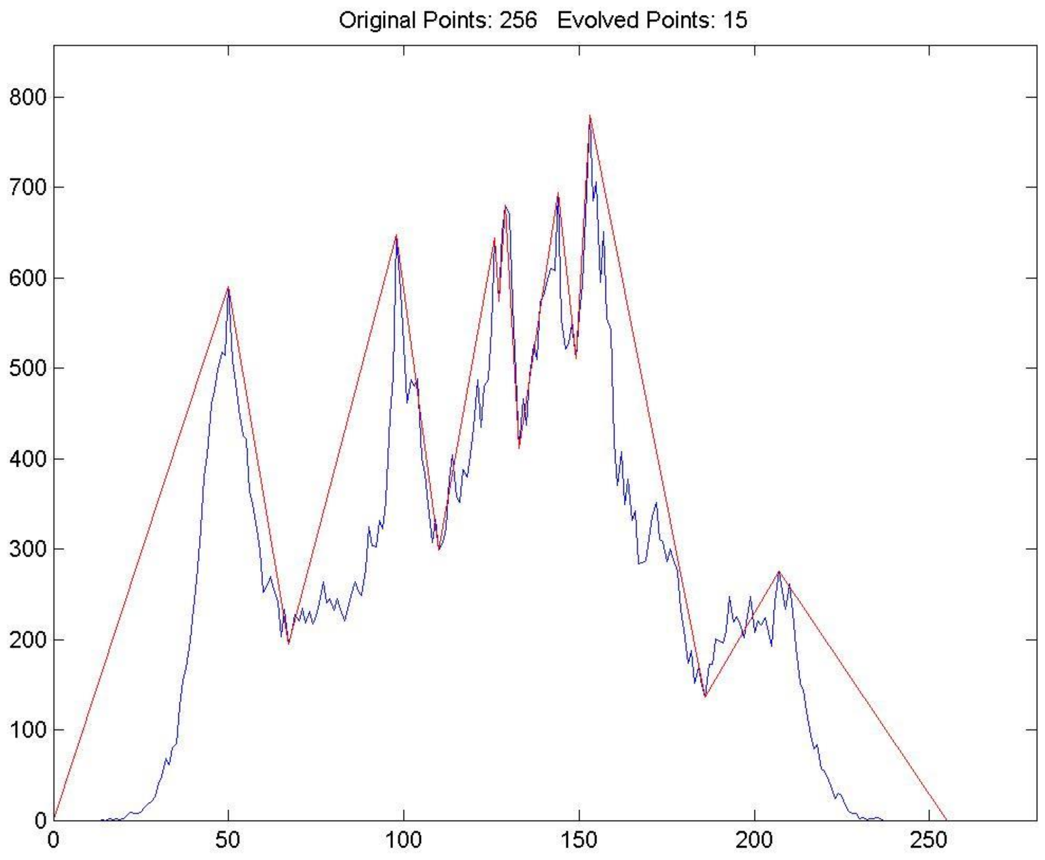




Gray Scale Image - Multimodal

Original Image of lena

Multimodal Histogram



Histogram of lena

Segmented Image

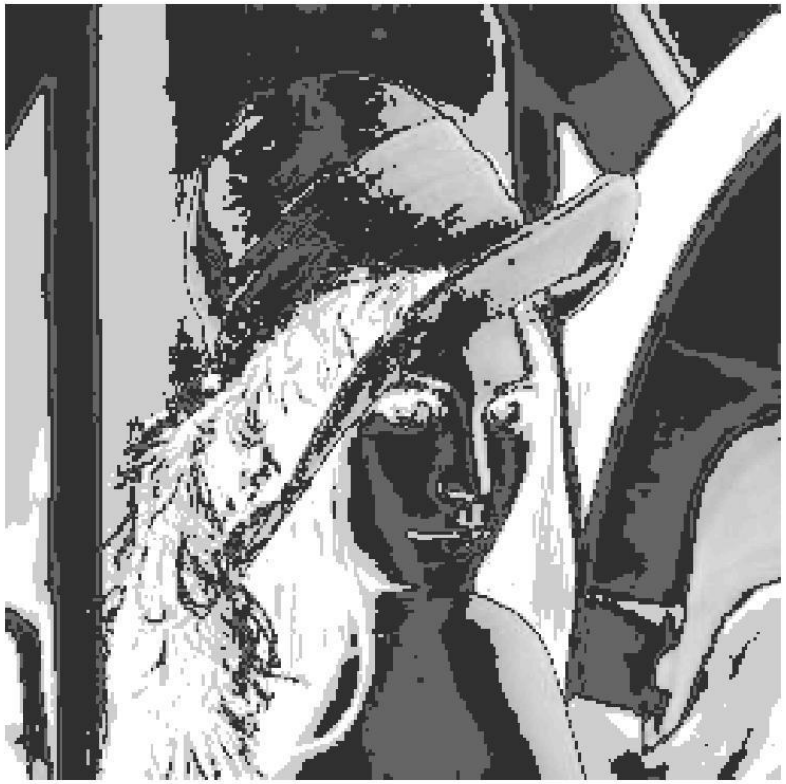
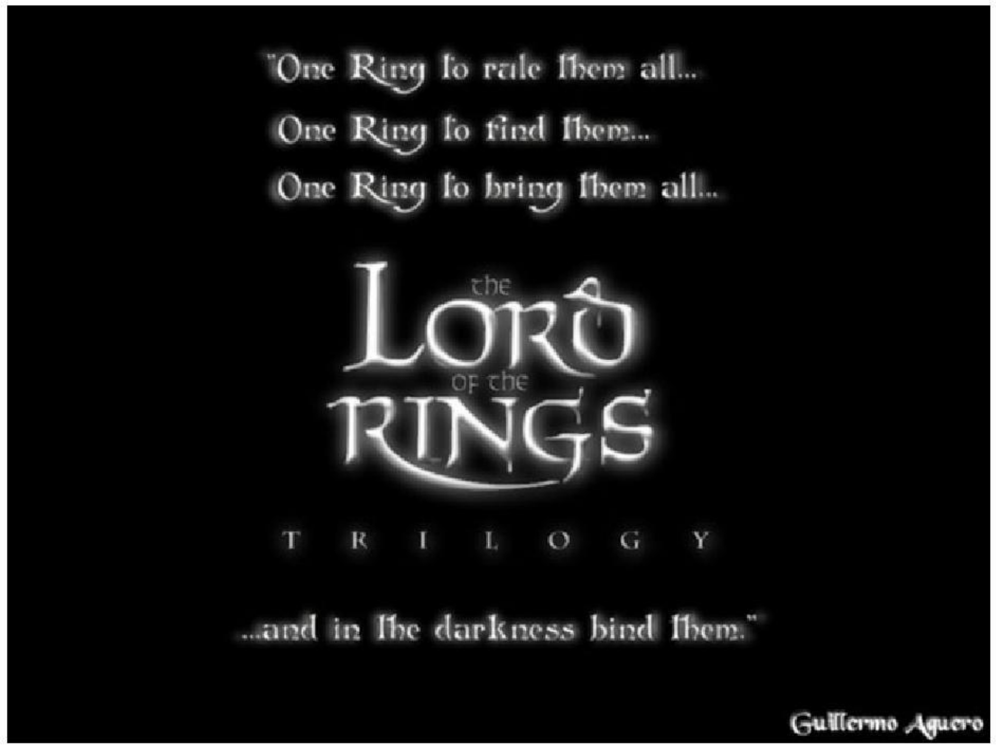


Image after segmentation – we get an outline of her face, hat, shadow etc

Experiments (contd.)


"One Ring to rule them all...

One Ring to find them...

One Ring to bring them all...

the LORD OF the RINGS

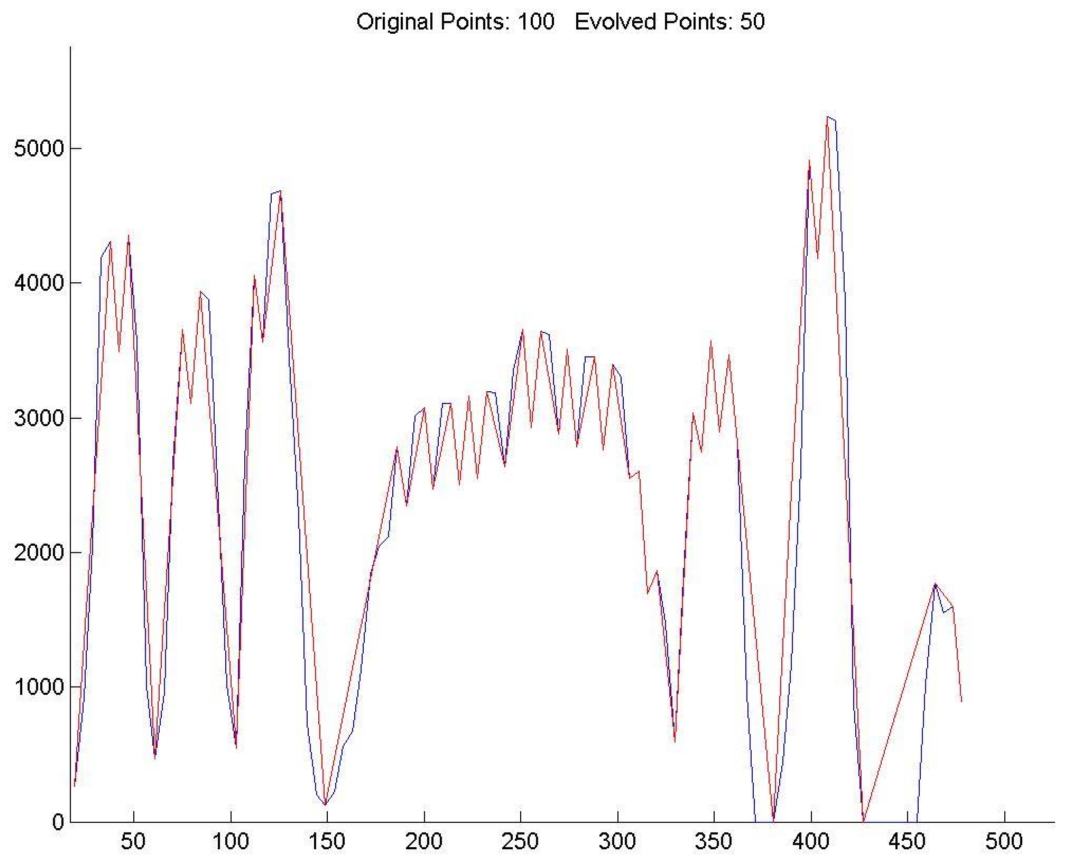
T R I L O G Y

...and in the darkness bind them."

Guillermo Aguero

Original Image, objective is to extract line by line

Histogram



Histogram giving us the six thresholds points, by plotting rows

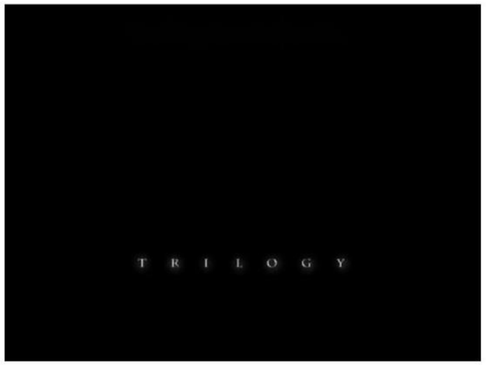
Segmented Image

"One Ring to rule them all...

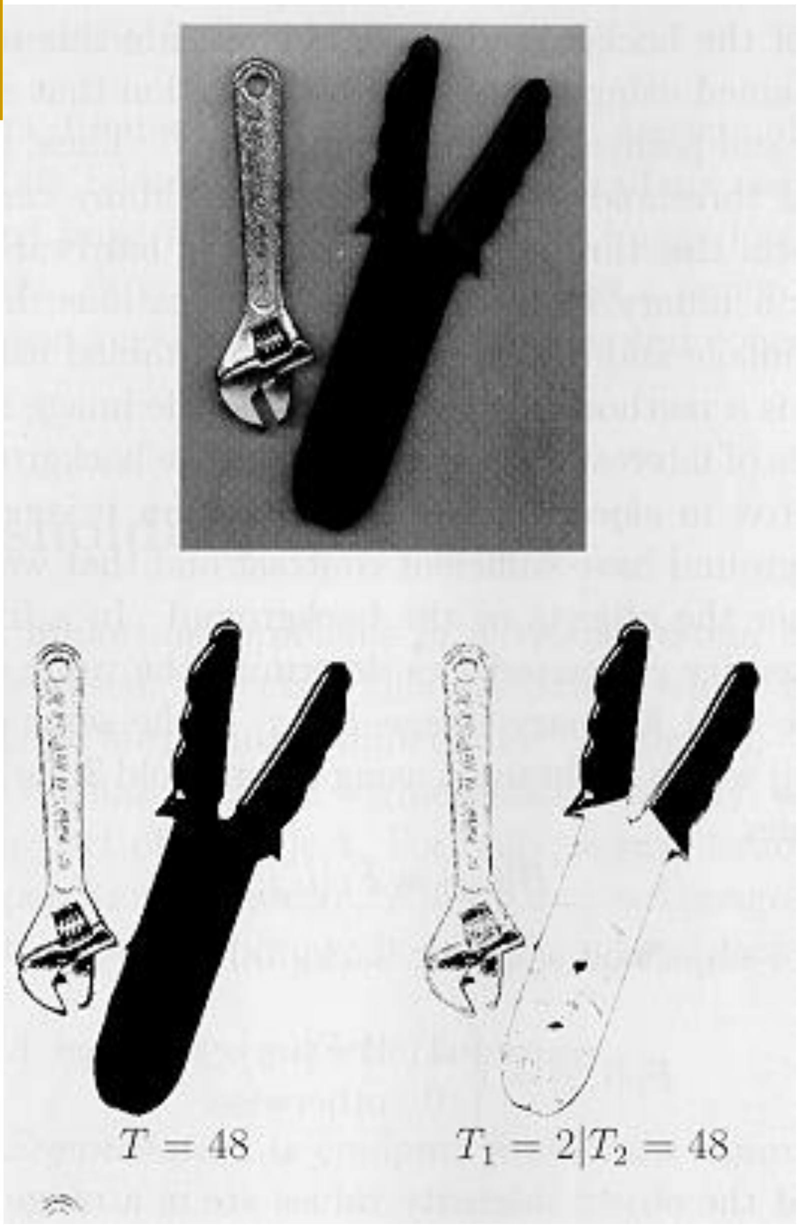
One Ring to find them...

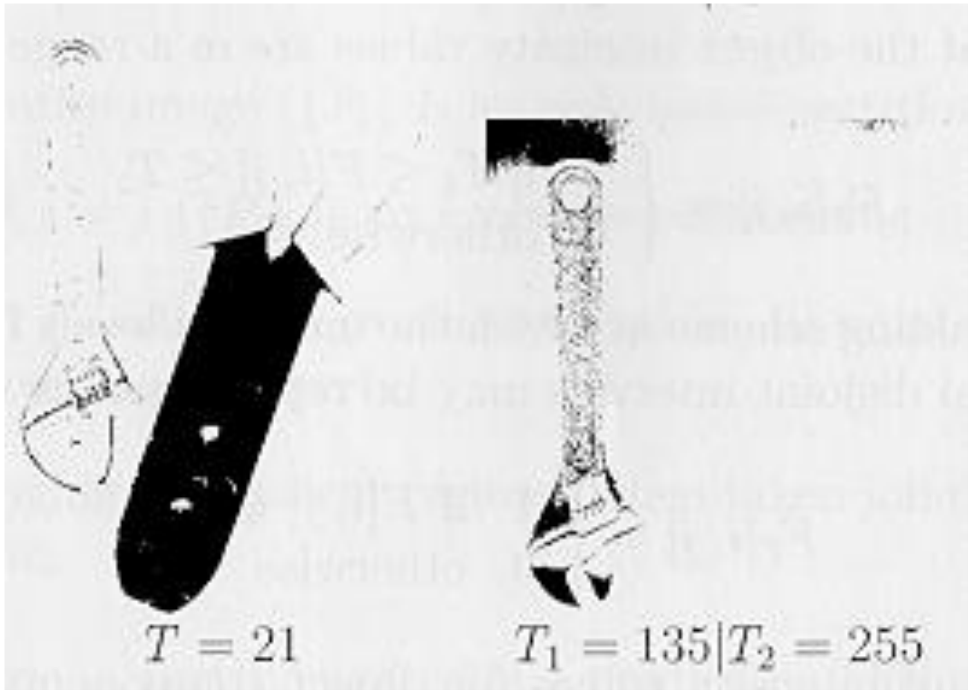
One Ring to bring them all...

the LORD OF the RINGS

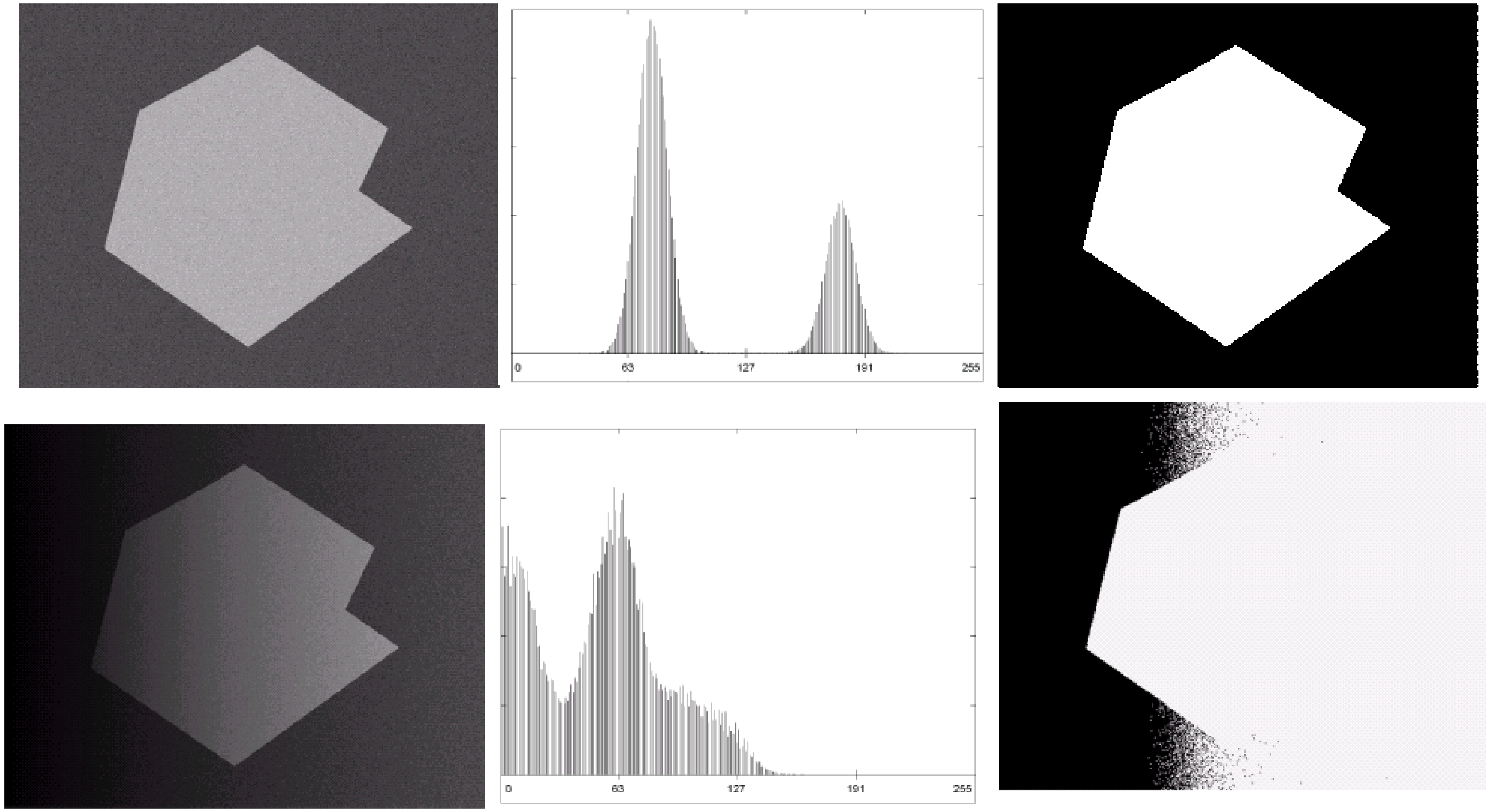

T R I L O G Y

...and in the darkness bind them."


$T = 48$
$T_1 = 2|T_2 = 48$


$T = 21$
$T_1 = 135|T_2 = 255$

Single Value Thresholding and Illumination



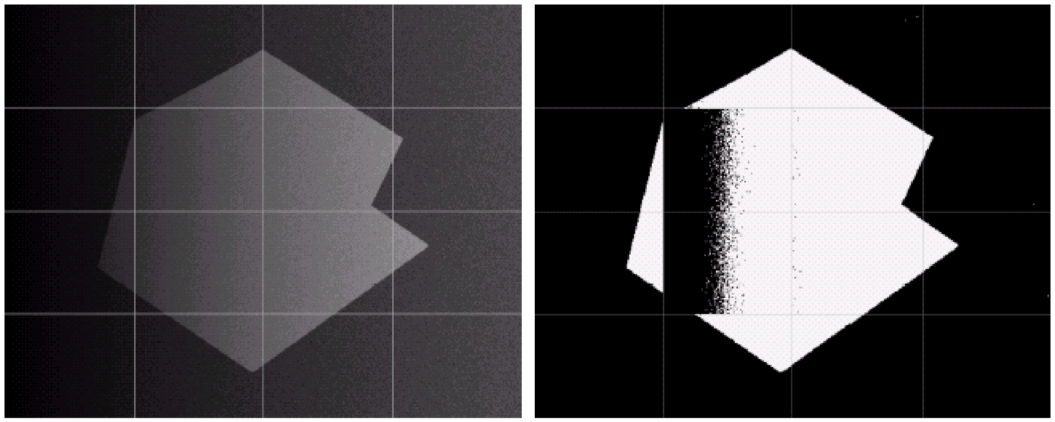
- **Uneven illumination can really upset a single valued thresholding scheme**

Basic Adaptive Thresholding

- An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually
- Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*

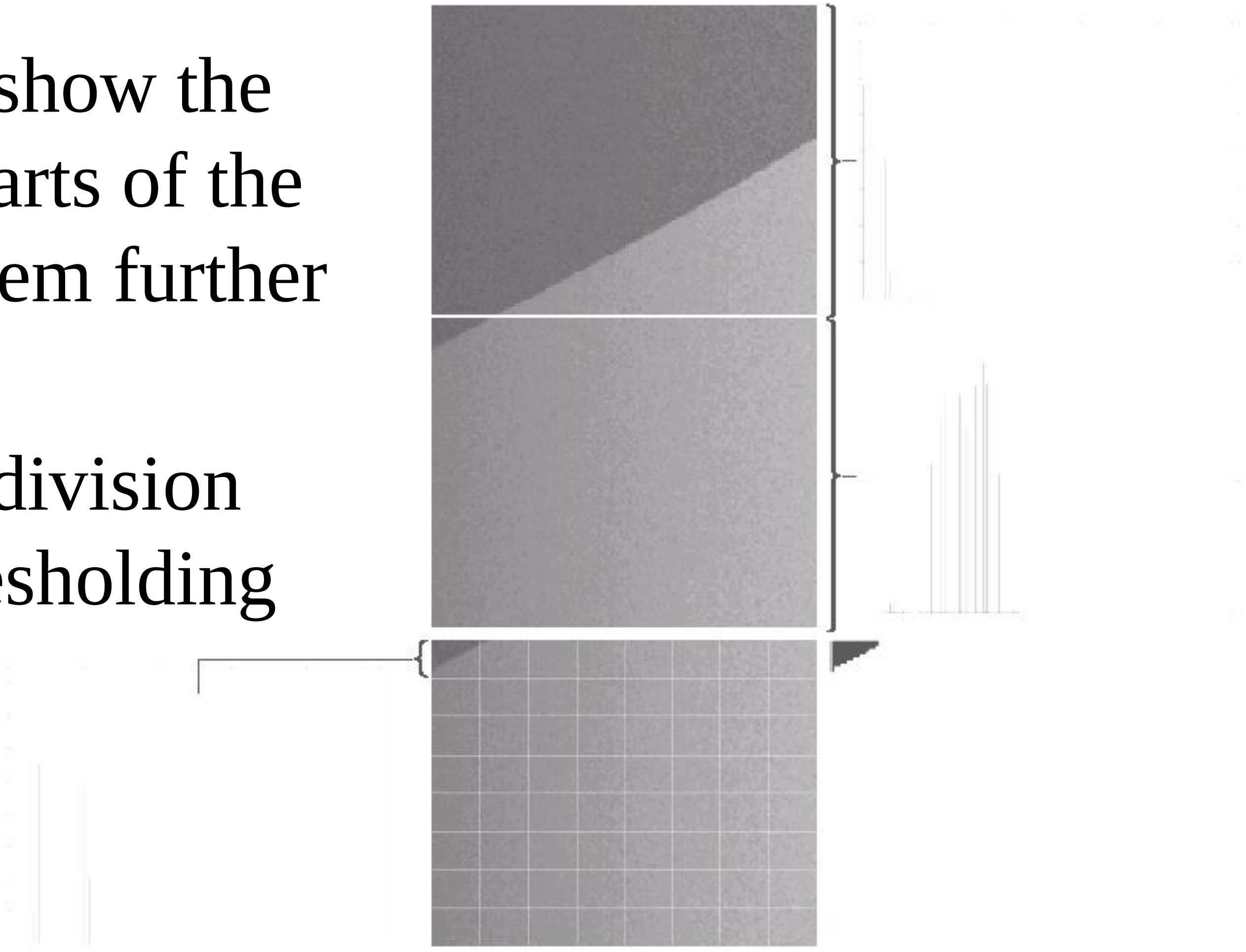
Basic Adaptive Thresholding Example

- The image below shows an example of using adaptive thresholding with the image shown previously



- As can be seen success is mixed
- But, we can further subdivide the troublesome sub images for more success

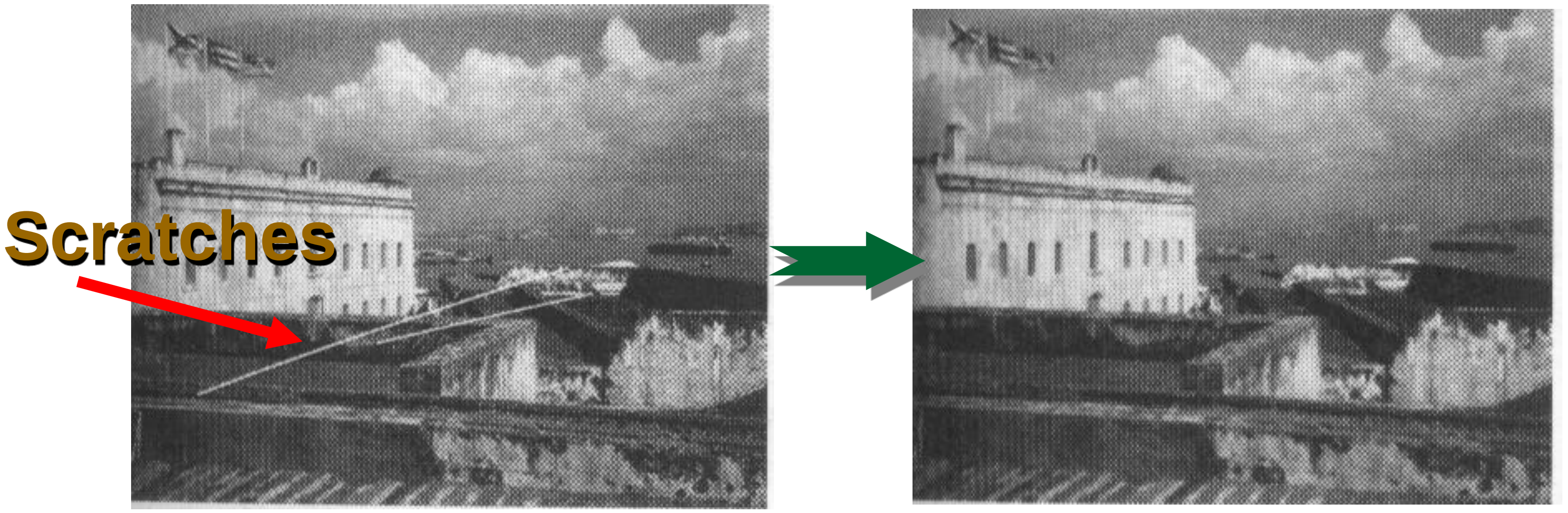
- These images show the troublesome parts of the previous problem further subdivided
- After this sub division successful thresholding can be achieved



2. Image enhancement

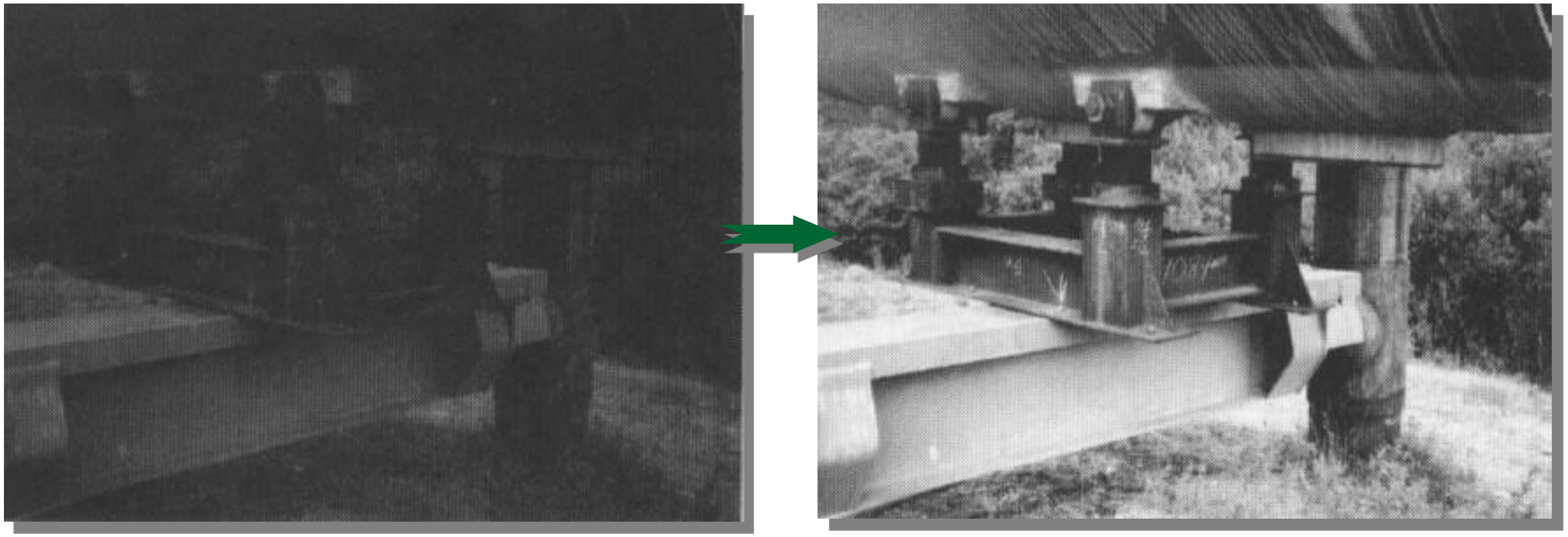
- Image is NOT Perfect in many situations
- Two general categories of problems
  - An image needs improvement
  - Low-level features must be detected

Example 1



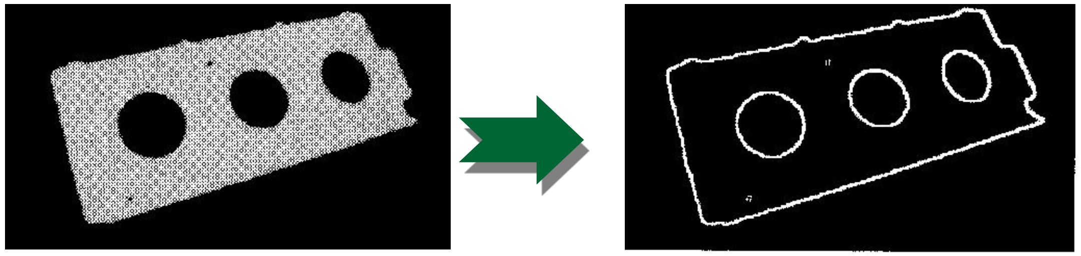
Example 1: Scratches from original photo need to be removed

Example 2



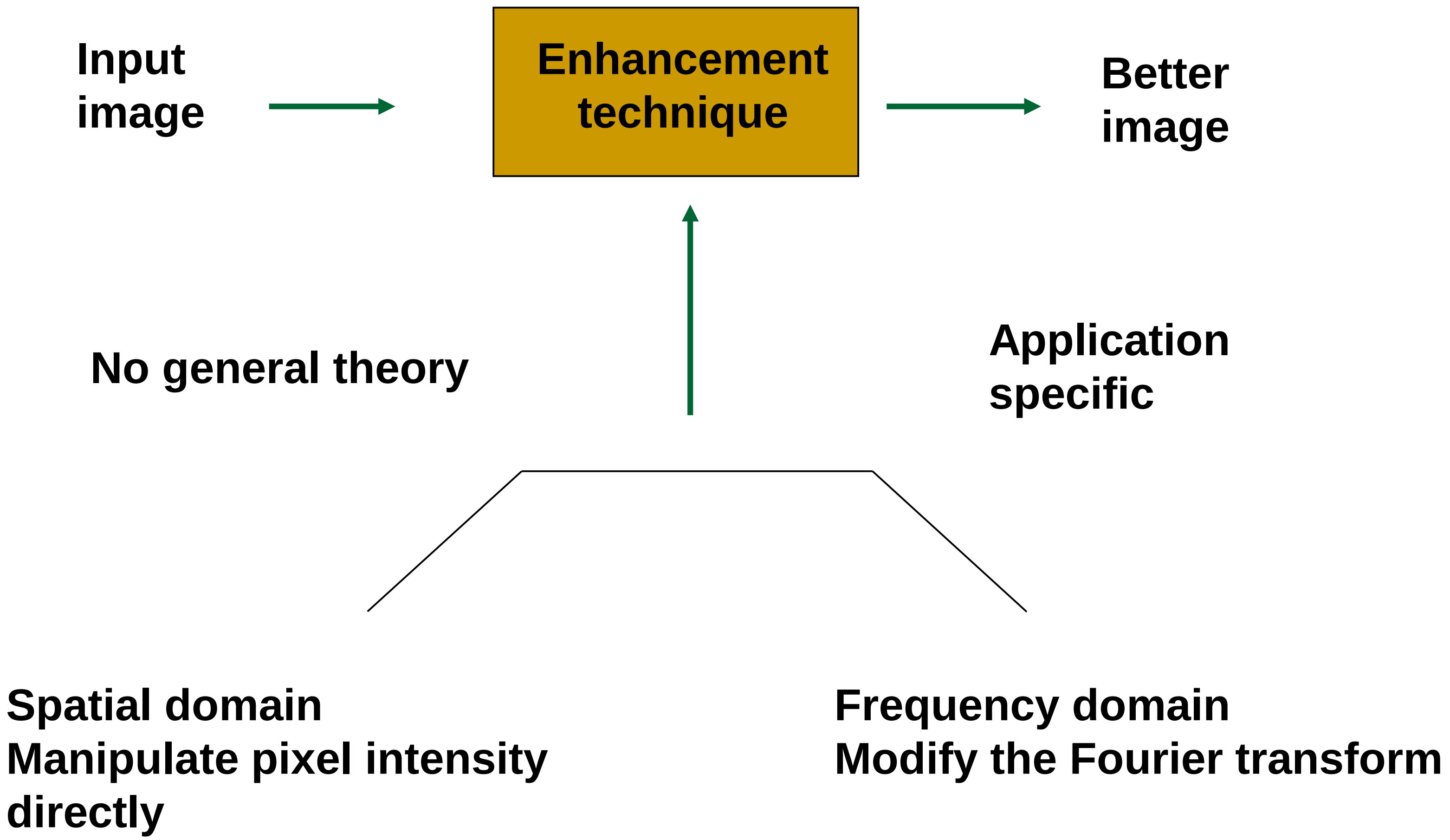
Example 2: Intensity of photo rescaled to show much better detail

Example 3



Example 3: Image of airplane part has edges enhanced to support automatic recognition and measurement

- **Definition 1:** ***Image enhancement*** operators improve the detectability of important image details or objects.
  - Example operations include noise reduction, smoothing, contrast stretching, and edge enhancement.


Input image
Enhancement technique
Better image
No general theory
Application specific
Spatial domain
Manipulate pixel intensity directly
Frequency domain
Modify the Fourier transform

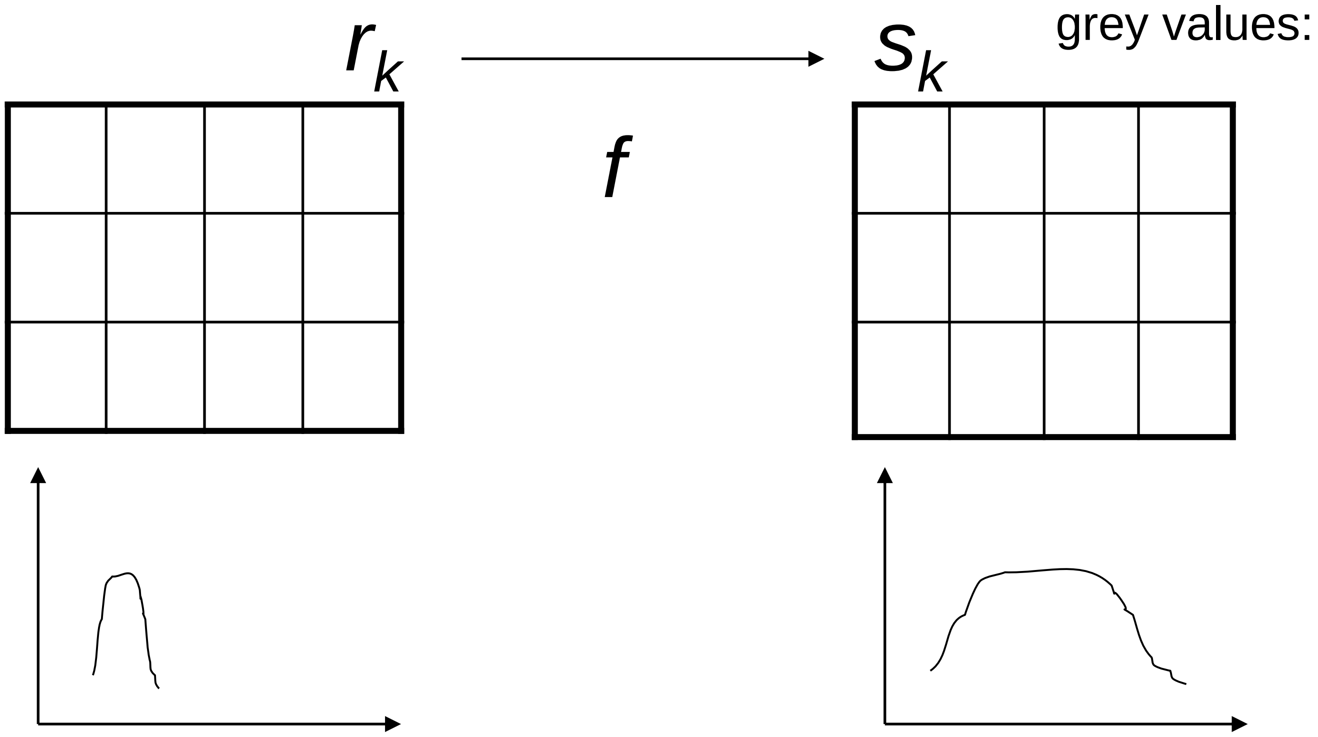
- Spatial domain techniques
  - Point operations
  - Histogram equalization
  - Applications of histogram-based enhancement
- Frequency domain technique
  - Ideal low pass filter
  - Ideal high pass filter
  - Butterworth low pass filter
  - Butterworth high pass filter
  - Gaussian low pass filter
  - Gaussian high pass filter

Point Operations Overview

- Point operations are **zero-memory** operations where a given gray level $x \in [0, L1]$ is mapped to another gray level $y \in [0, L2]$ according to a transformation
- Gray-Level Mapping

- **Two ways to enhance images**:
  - Changing the intensity values of pixels
  - Transforming the pixels via a single function that maps an input gray value into a new output value
- Remapping the gray values is often called *stretching* (伸縮)

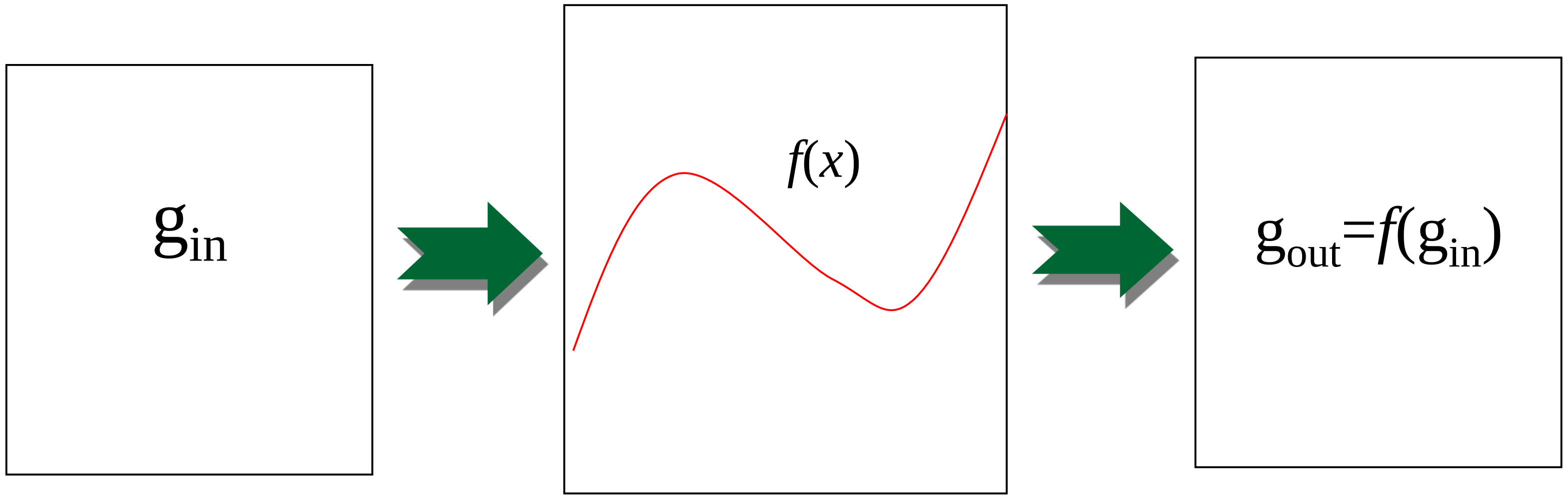
Point operation $f(r_k) = s_k$



Properties of *f*:

keeps the original range of grey values

monotone increasing



Input image **Mapping function** **Output image**

- Definition - A point operator applied to an image is an operator in which the output pixel is determined only by the input pixel,

$$\mathrm{Out}[x, y] = f(\mathrm{In}[x, y]);$$

  possible function f depends upon some global parameters.

- A **contrast stretching** **(對比度伸縮)** operator is a point operator that uses a piecewise smooth function f (In[x, y]) of the input gray level to enhance important details of the image.

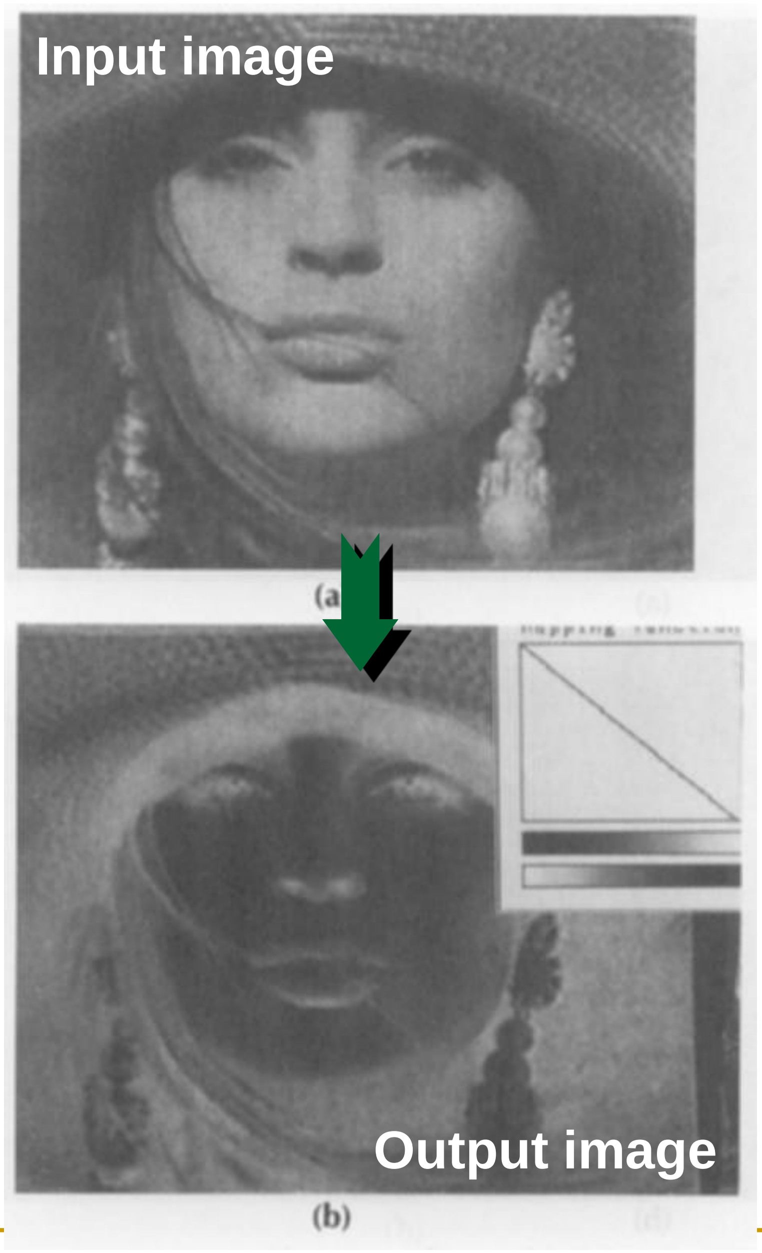
- Using the mapping function $f(x) = x^{1/y}$ is called *Gamma correction* and it might be the proper theoretical model to restore an image to an original form after undergoing known physical distortion.

- When y = 2.0, $f(x) = x^{1/y} = x^{0.5}$

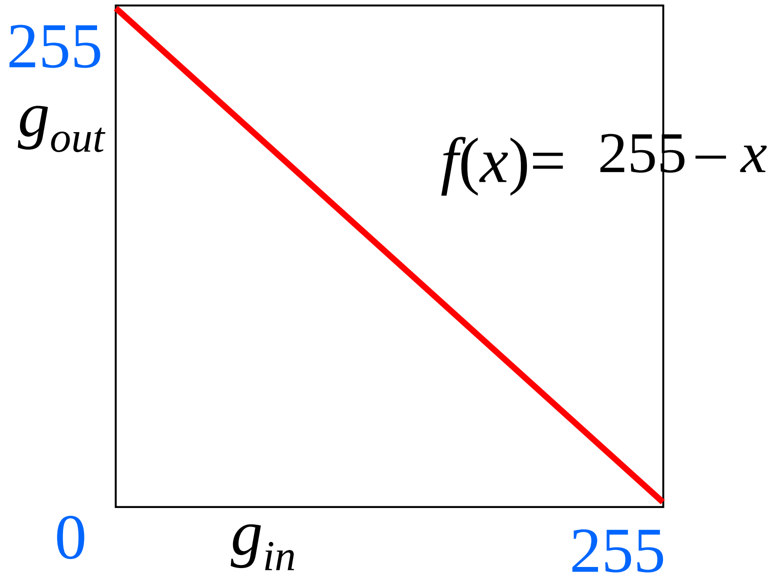
  it is a boosting value. The intensity mapping using the function $f(x) = x^{0.5}$ nonlinearly boosts all intensities, but boosts lower intensities more than the higher ones.

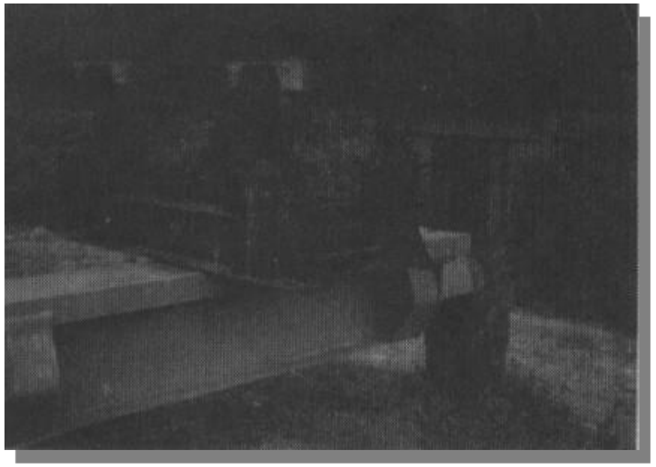
- When y = -1, $f(x) = x^{1/y} = x^{-1}$

  it is a reverse value. The intensity mapping using the function $f(x) = x^{-1}$ reverses all intensities. The dark pixel reverses to bright one, and the bright pixel becomes a dark one.

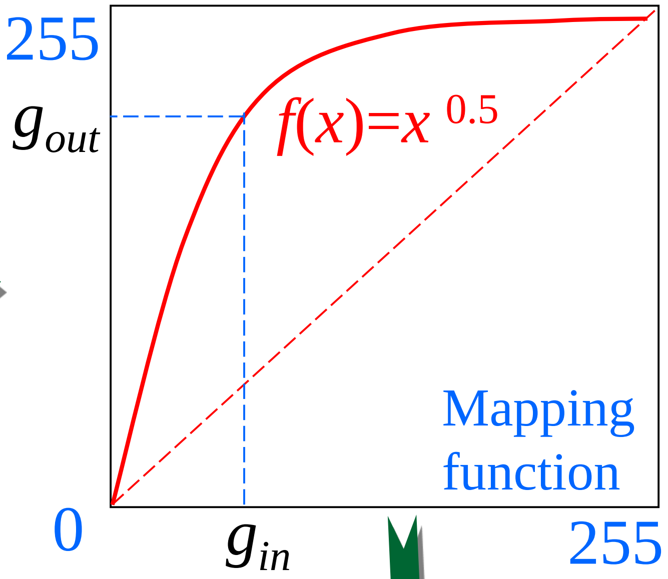


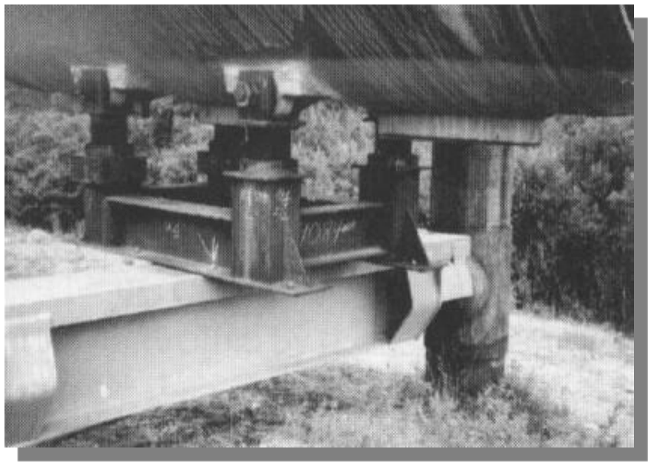
Mapping function (Reverse)





Input image





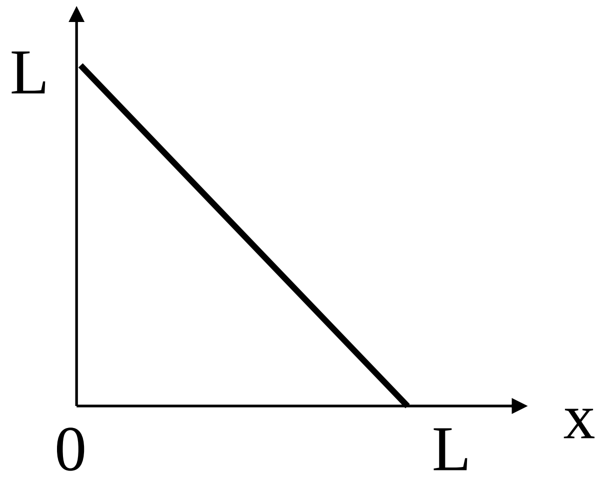
Output image

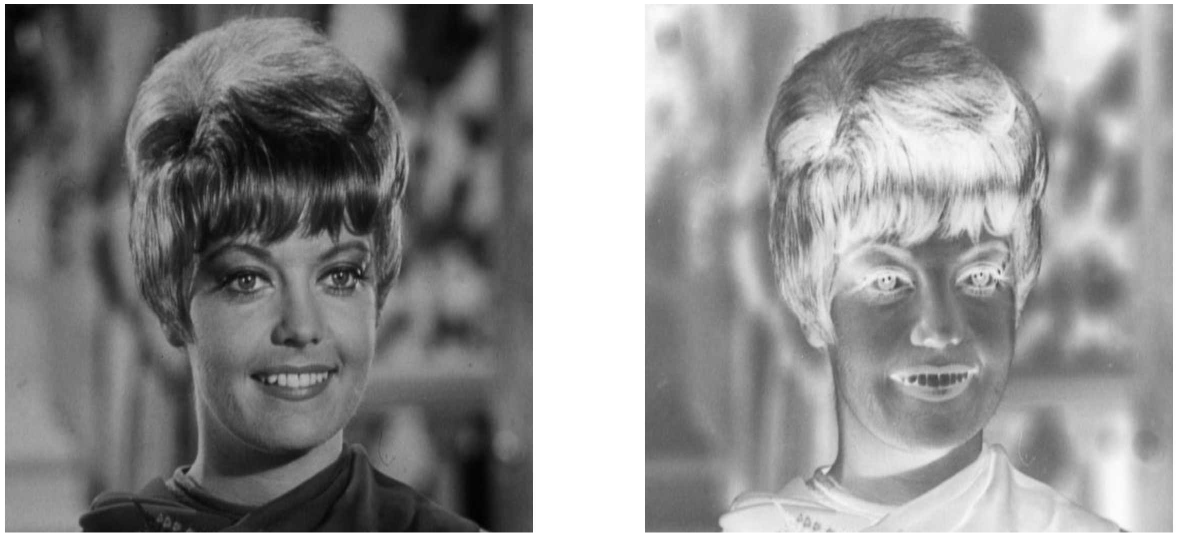
Intensity mapping using the function $f(x) = x^{0.5}$ (Gamma correction) boosts dark pixels much more than bright ones

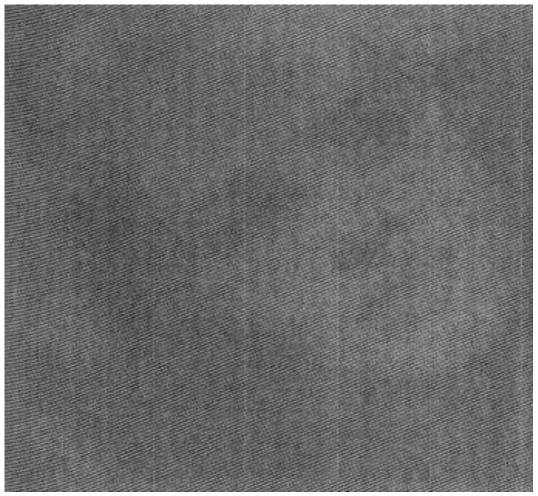
Original, fairly dark, Alaskan Pipeline image

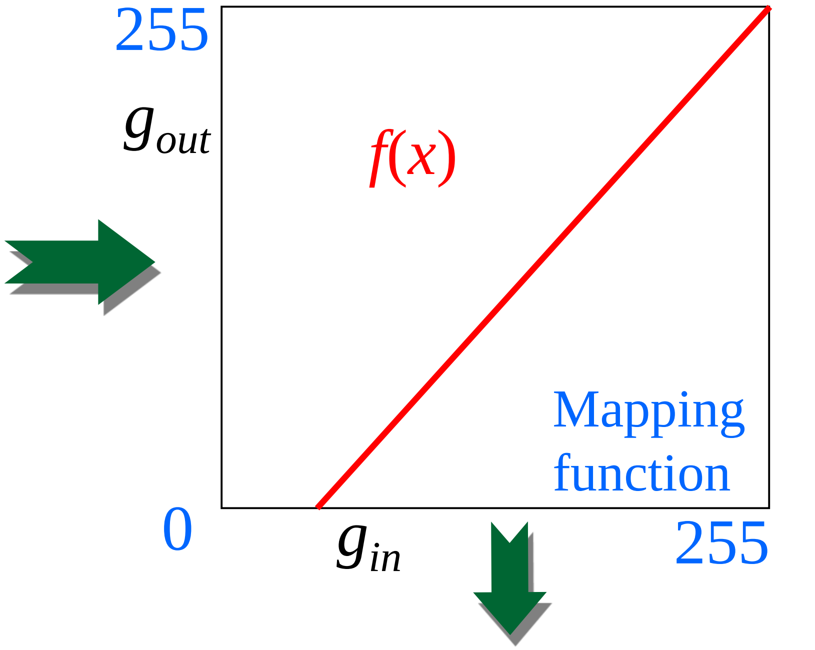
Digital Negative

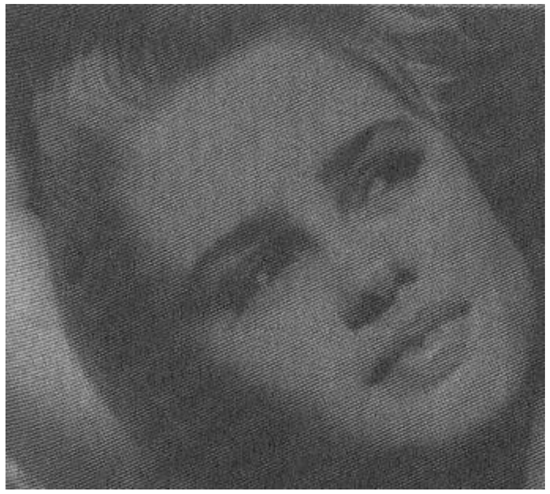
$$y = L - x$$





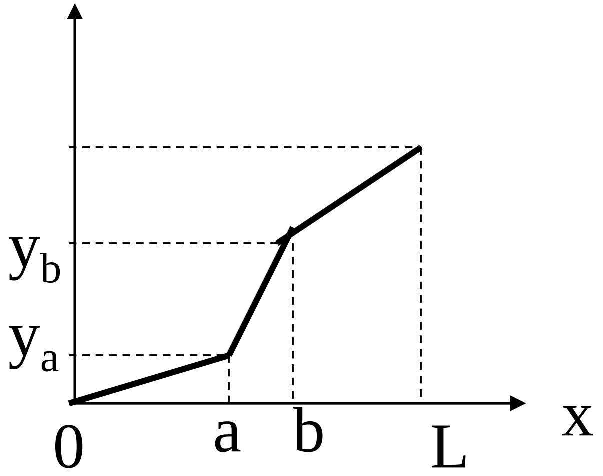

Input image

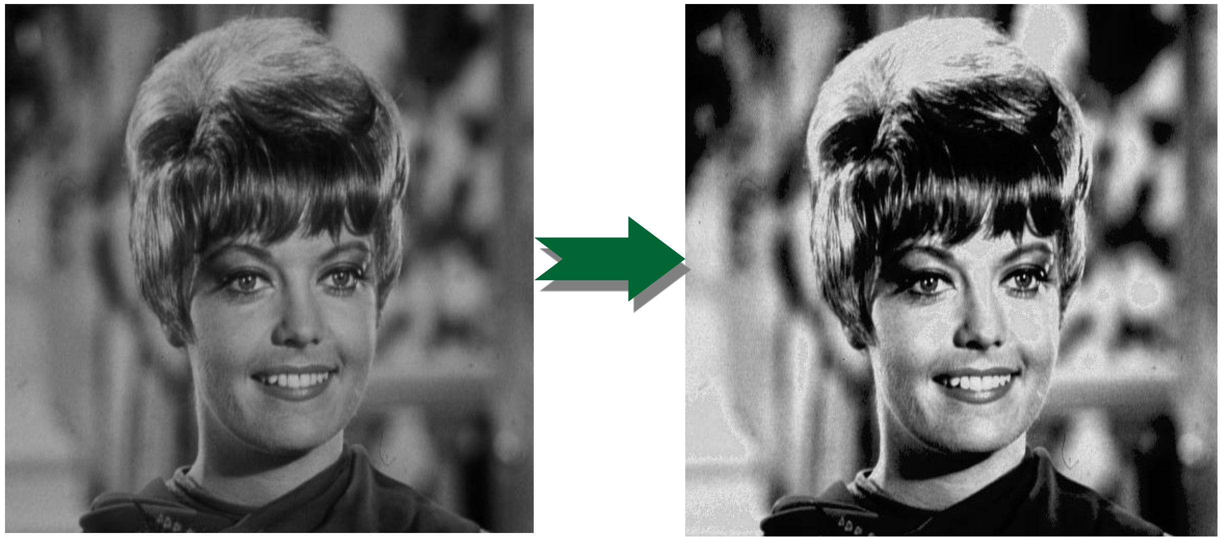



Output image

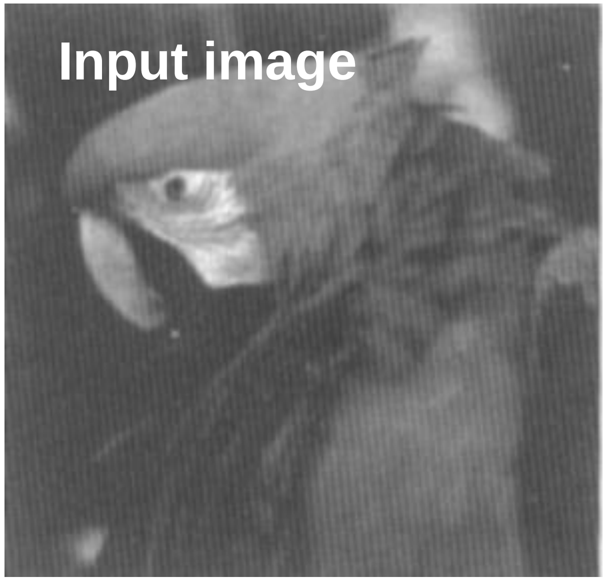
Contrast Stretching

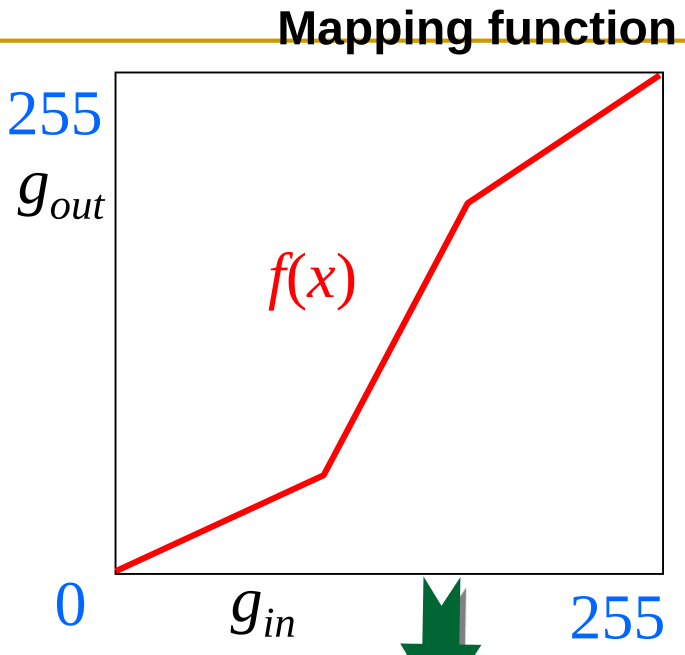
$$y = \begin{cases} \alpha x & 0 \le x < a \\ \beta(x-a)+y_a & a \le x < b \\ \gamma(x-b)+y_b & b \le x < L \end{cases}$$

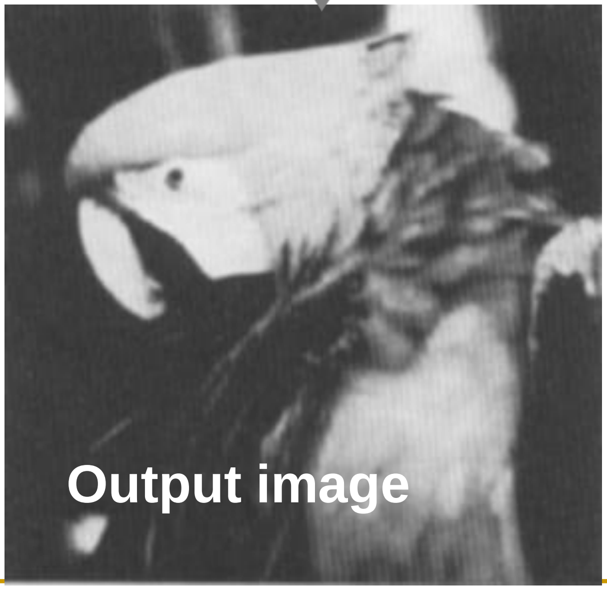




$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200$$

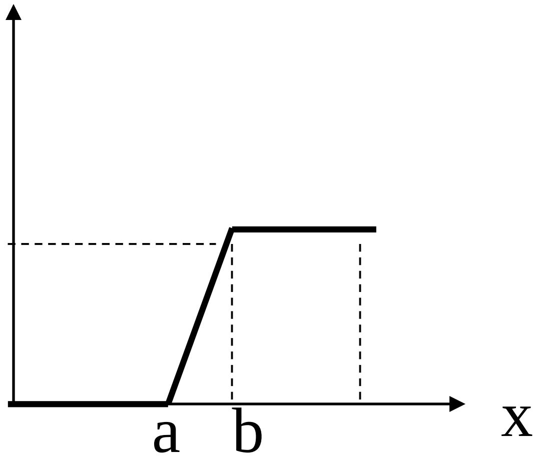

Input image

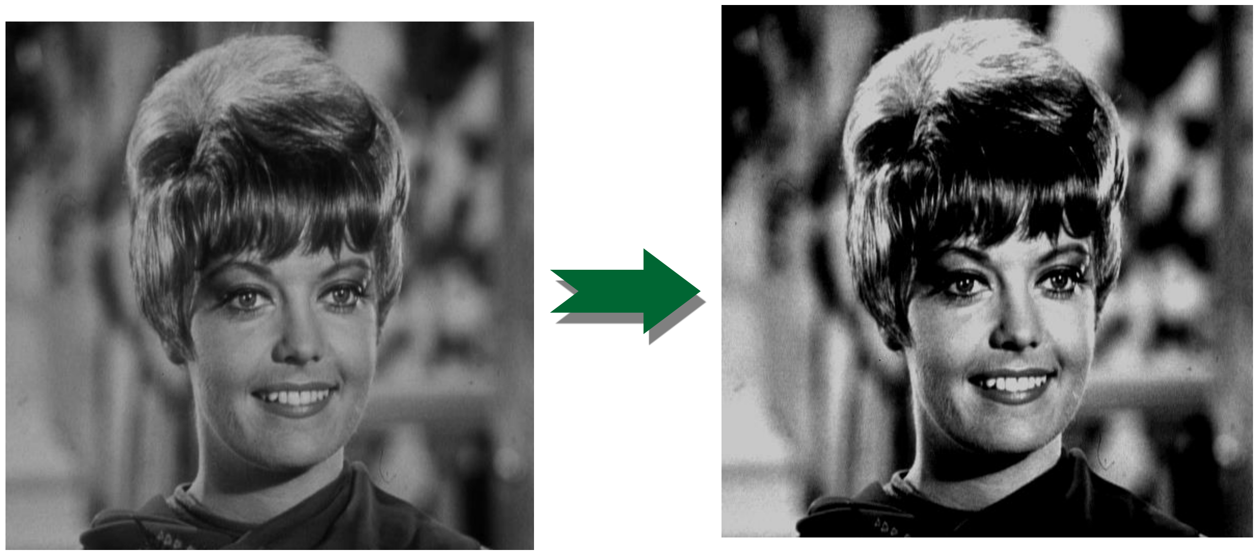

Mapping function
255
g_{out}
$f(x)$
0
g_{in}
255


Output image

Clipping

$$y = \begin{cases} 0 & 0 \le x < a \\ \beta(x-a) & a \le x < b \\ \beta(b-a) & b \le x < L \end{cases}$$

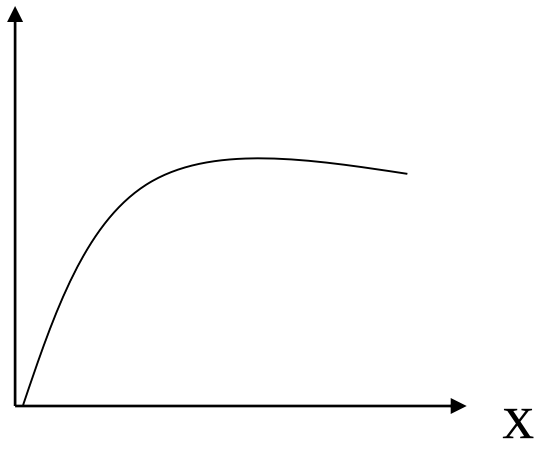


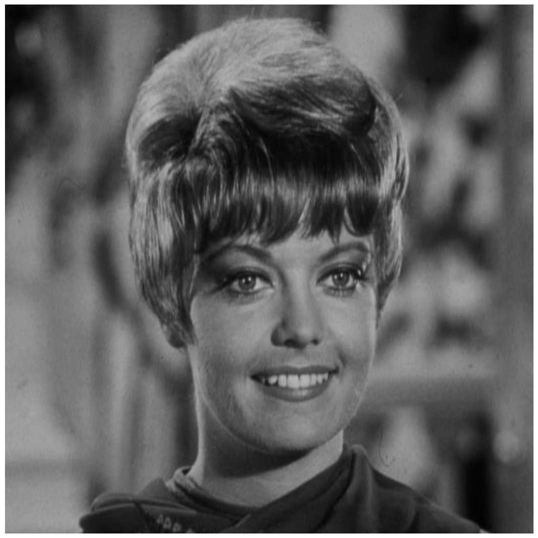


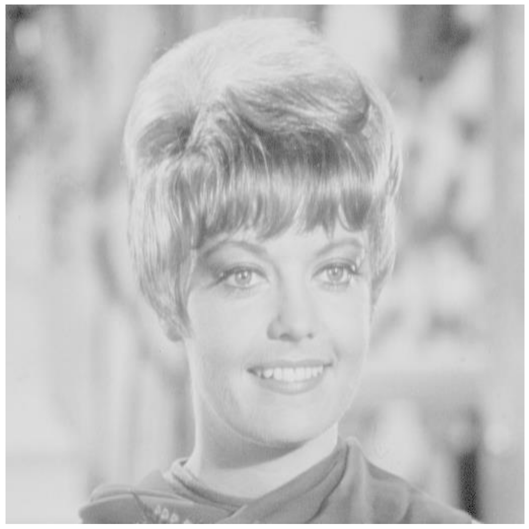
$a = 50, b = 150, \beta = 2$

Range Compression

$$y = c\log_{10}(1+x)$$

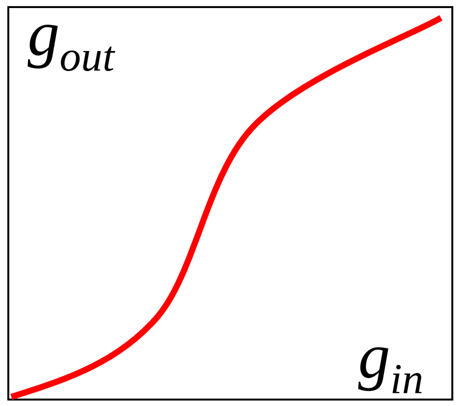


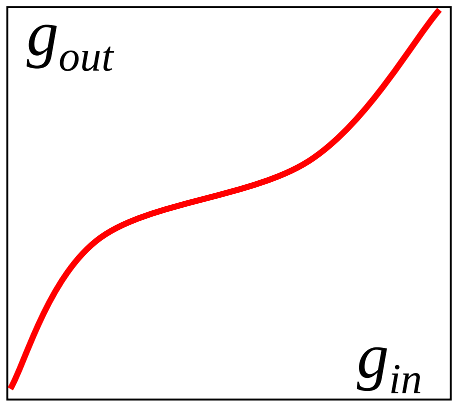


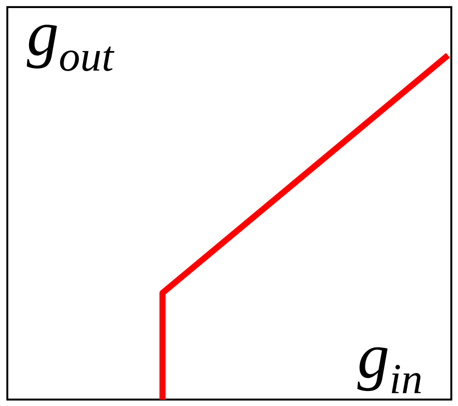


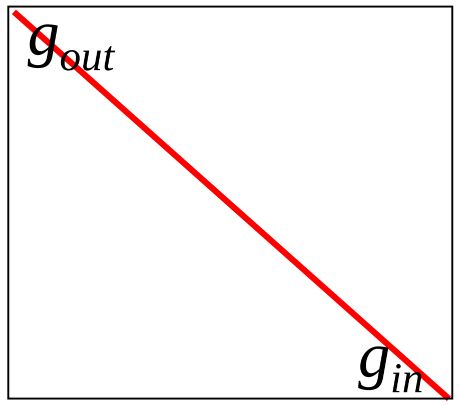
c=100

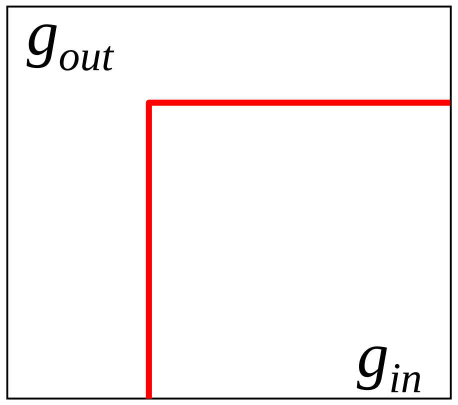
Various Mapping Functions











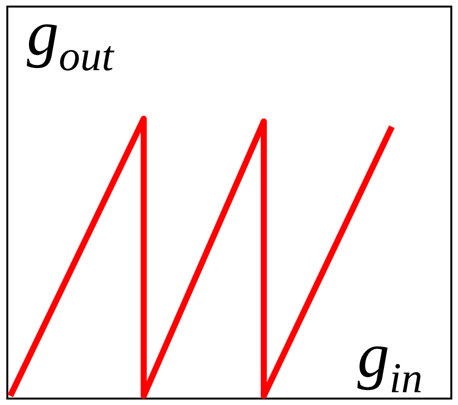
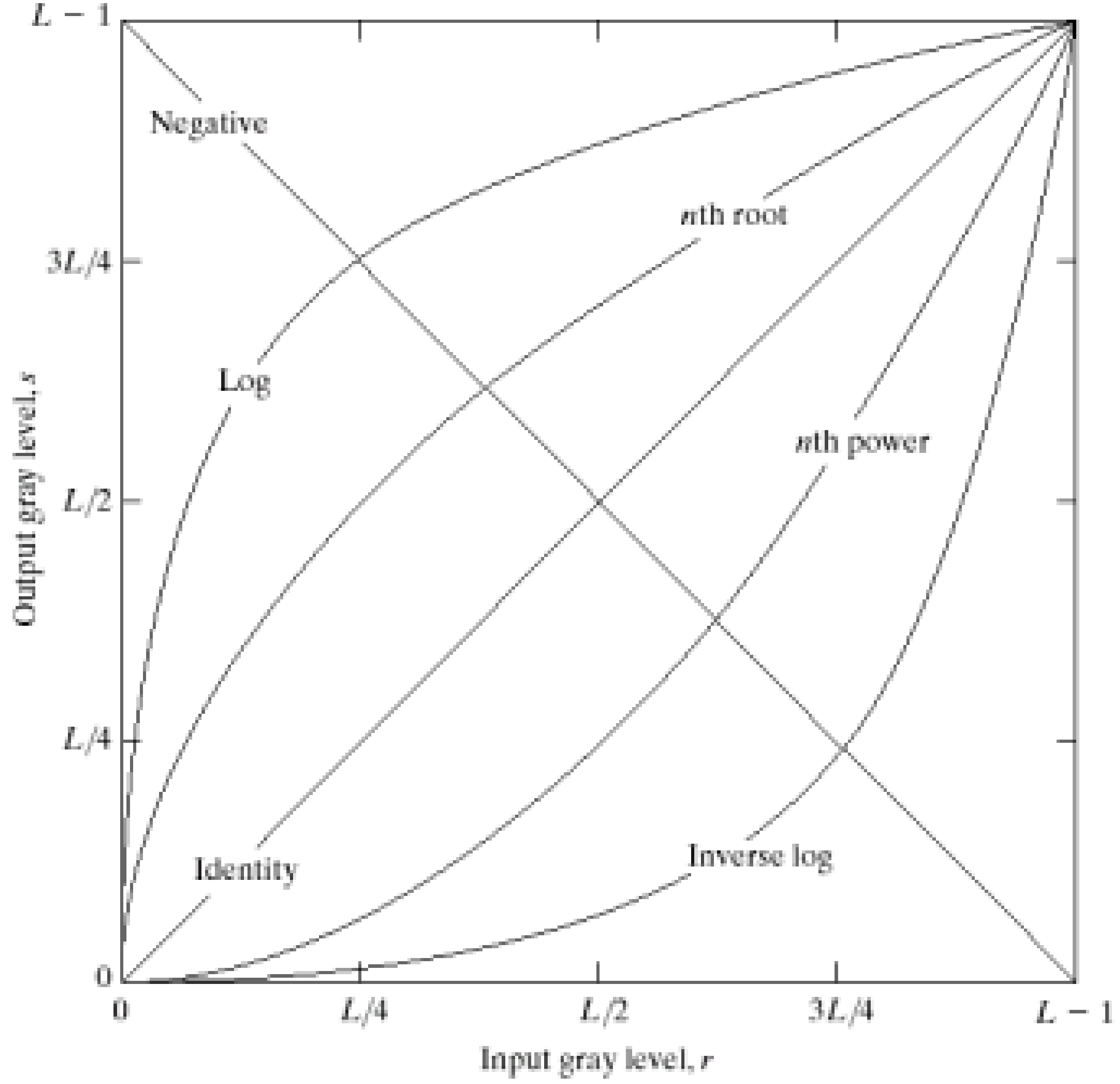


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

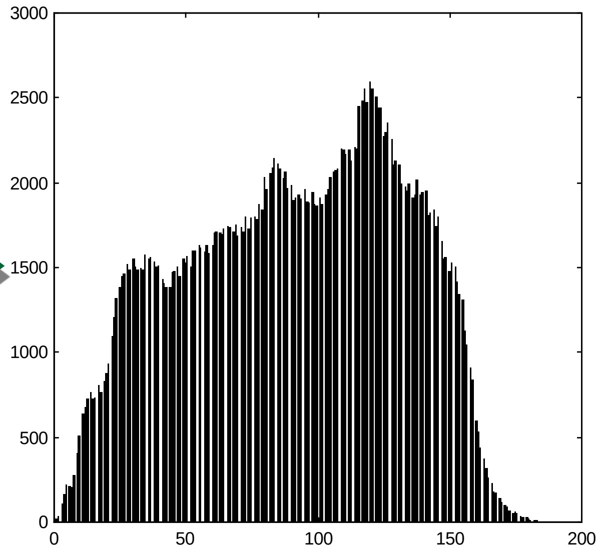


Summary of Point Operation

- So far, we have discussed various forms of mapping function f(x) that leads to different enhancement results
  - MATLAB function > imadjust
- The natural question is: How to select an appropriate f(x) for an arbitrary image?
- One systematic solution is based on the histogram information of an image
  - Histogram equalization and specification

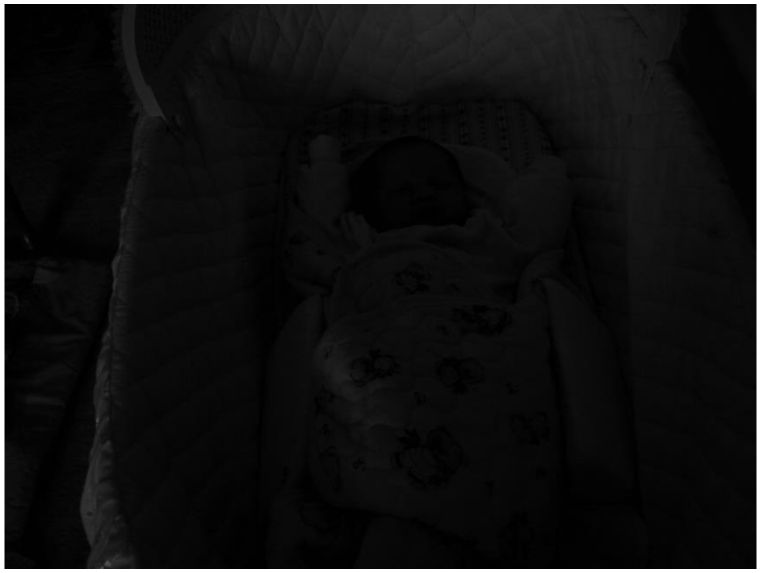
Histogram based Enhancement

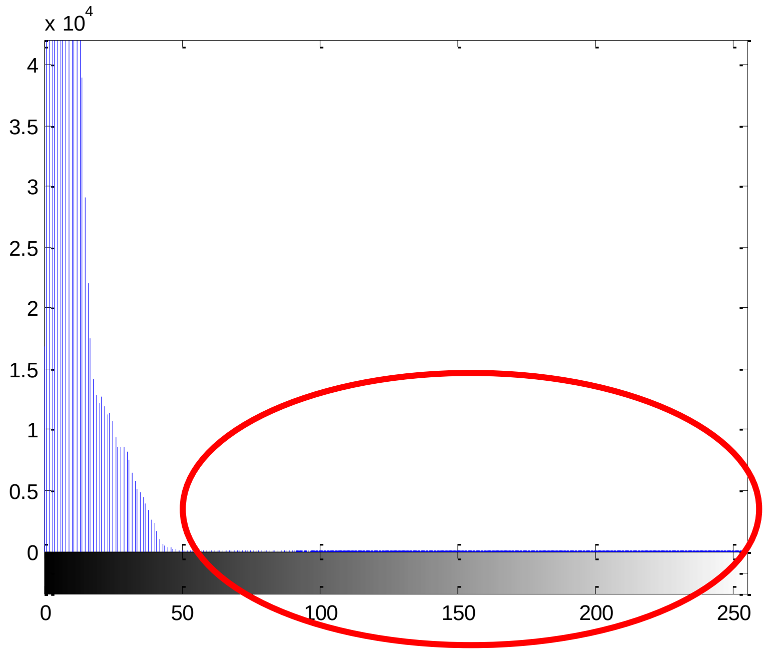
- Histogram of an image represents the relative frequency of occurrence of various gray levels in the image



MATLAB function >imhist(x)

Why Histogram?

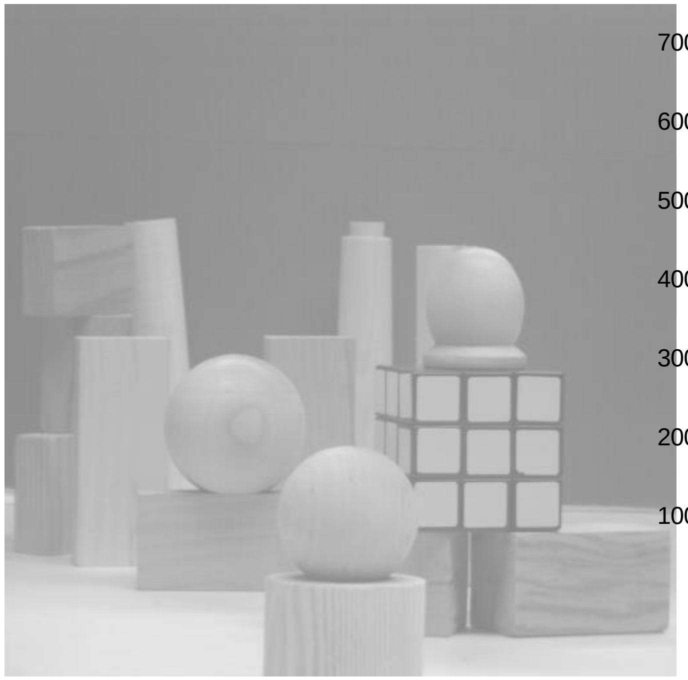


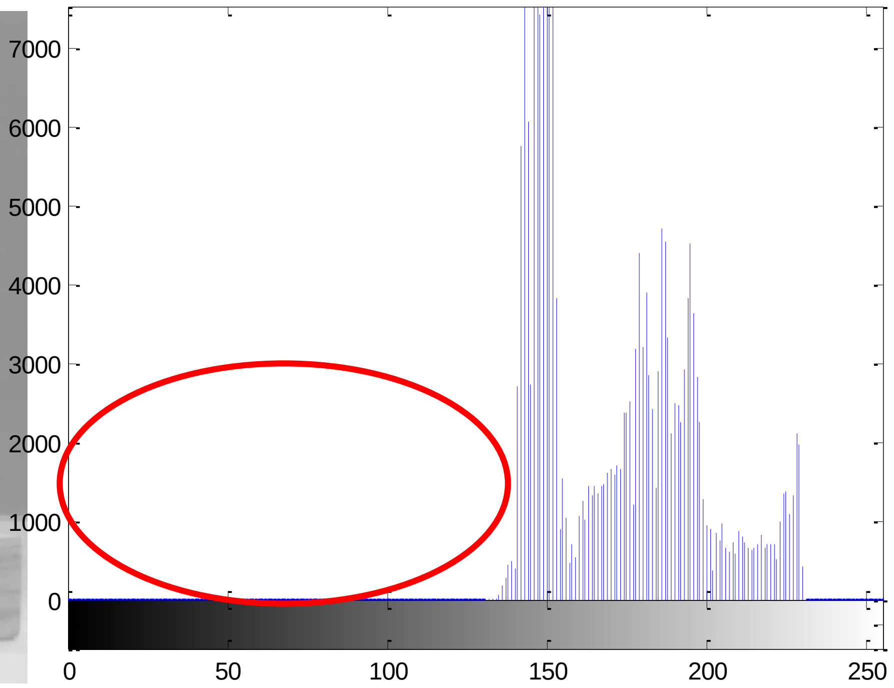


It is a baby in the cradle!

Histogram information reveals that image is under-exposed

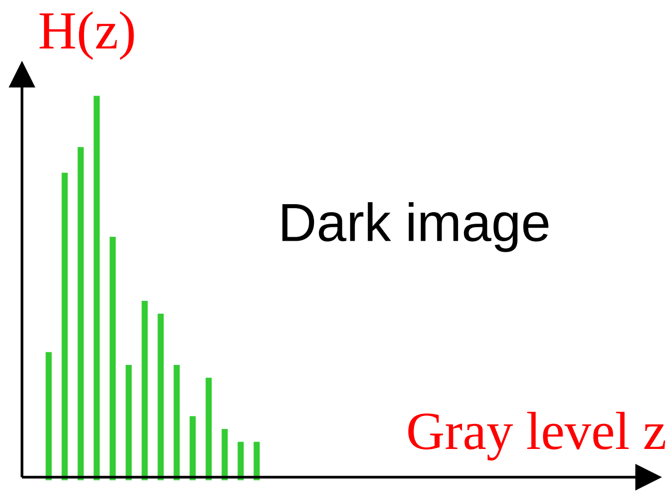
Another Example

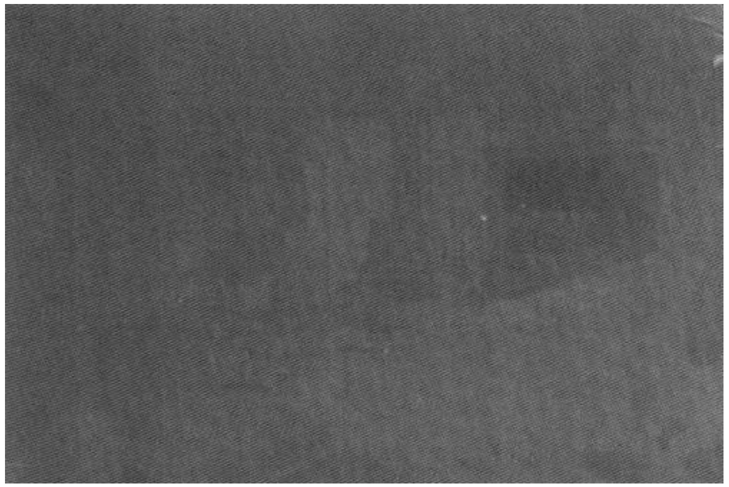

Over-exposed image



- The histograms of **four basic types** of images

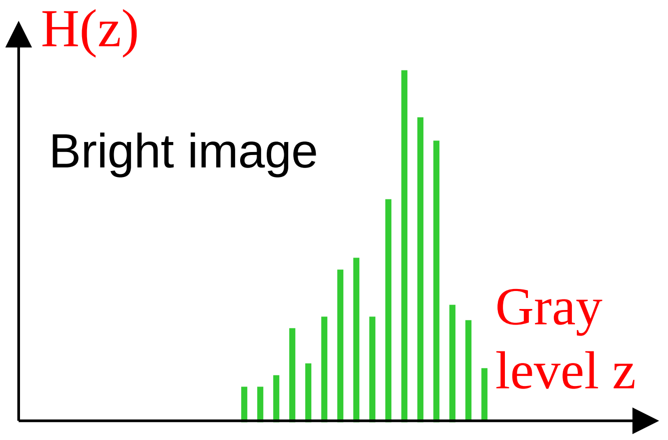
- The gray levels are concentrated toward the dark end of the grayscale range. Thus this histogram corresponds to an image with overall dark characteristics

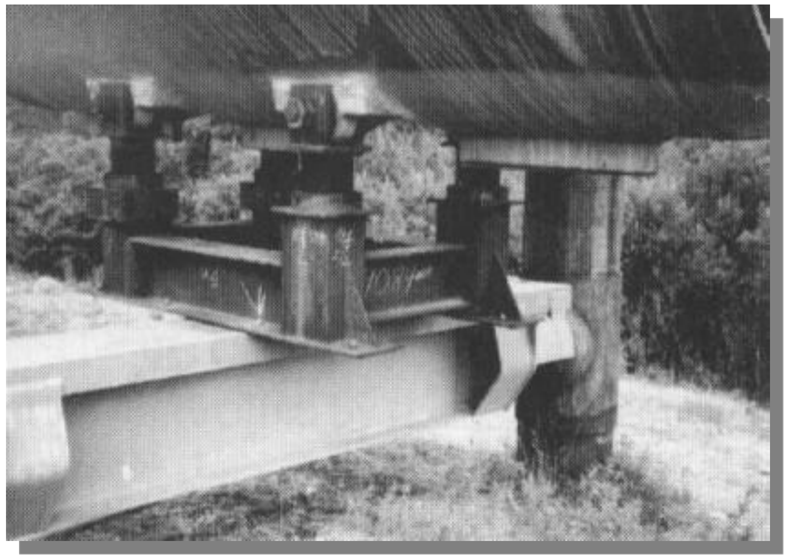




Dark image

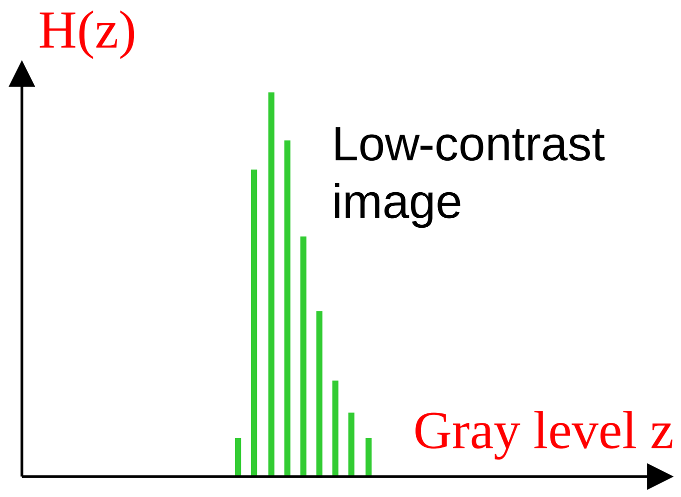
- The gray levels are concentrated toward the bright end of the grayscale range. Thus this histogram corresponds to an image with overall bright characteristics

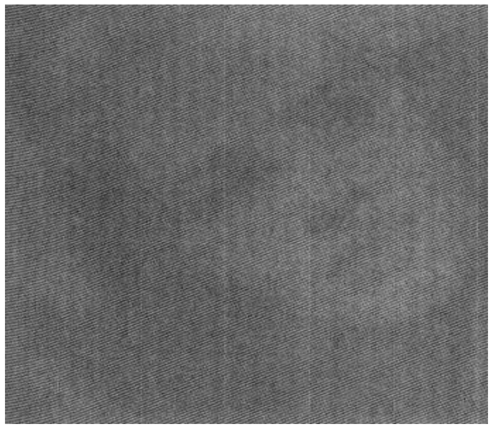




Bright image

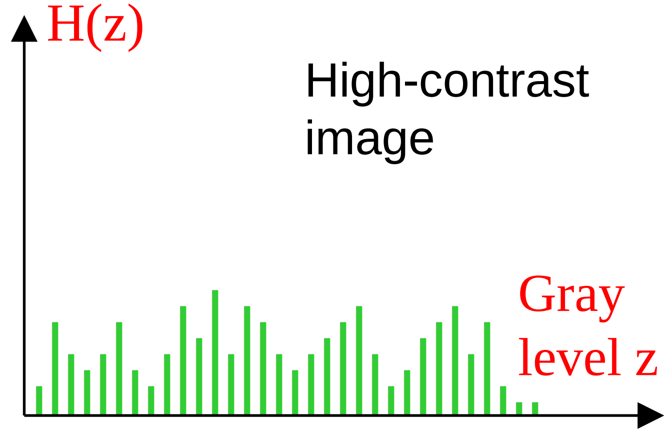
- The histogram has a narrow shape, which indicates little dynamic range and thus corresponds to an image having low contrast

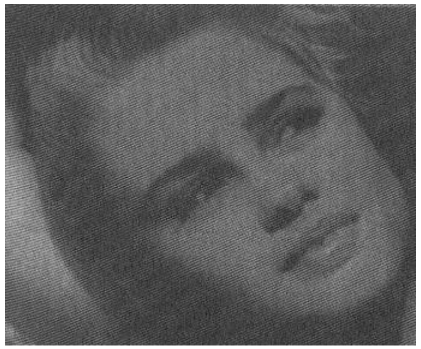




Low-contrast image

- The histogram has significant spread, corresponding to an image with high contrast

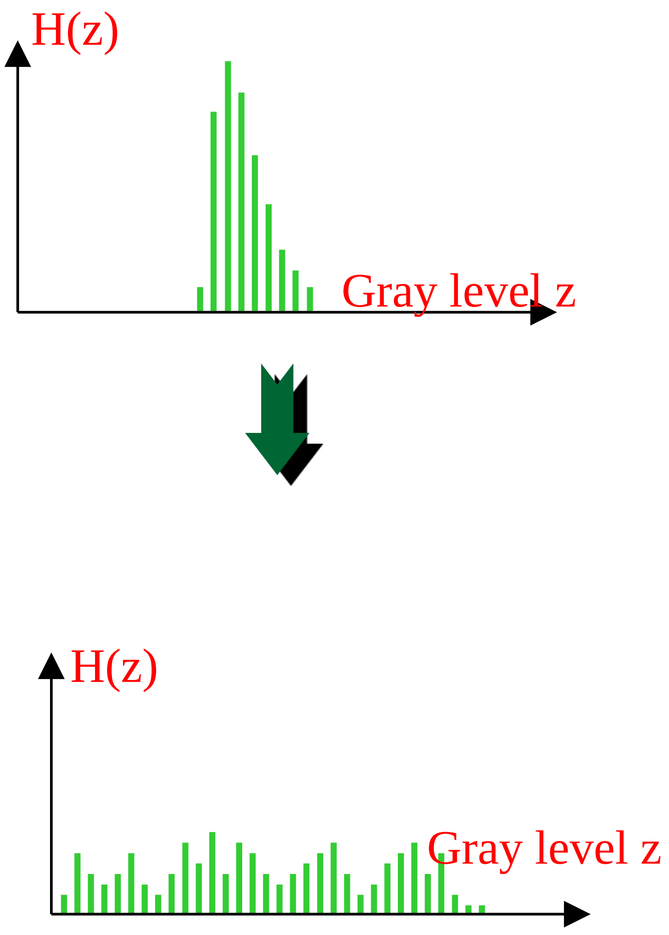




High-contrast image

How to Adjust the Image?

- **Histogram modification:** techniques modify an image so that its histogram has a desired shape. This technique can be used to improve the image contrast.



Histogram Equalization

- ***Histogram equalization*** is an important method in histogram modification.
- Transform the intensity values so that the histogram of the output image approximately matches the flat (uniform) histogram
- Histogram equalization is often tried to enhance an image.

Normalised histogram function

- The normalised histogram function is the histogram function divided by the total number of the pixels of the image:

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

- It gives a measure of how likely is for a pixel to have a certain intensity. That is, it gives the probability of occurrence the intensity.

- The sum of the normalised histogram function over the range of all intensities is 1.

Example

- Consider a 5x5 image with integer intensities in the range between one and eight:

1	8	4	3	4
1	1	1	7	8
8	8	3	3	1
2	2	1	5	2
1	1	8	5	2

Normalised histogram function

$$\begin{array}{lcl}
h(r_1)=8 & & p(r_1)=8/25=0.32 \\
h(r_2)=4 & & p(r_2)=4/25=0.16 \\
h(r_3)=3 & & p(r_3)=3/25=0.12 \\
h(r_4)=2 & \Rightarrow & p(r_4)=2/25=0.08 \\
h(r_5)=2 & & p(r_5)=2/25=0.08 \\
h(r_6)=0 & & p(r_6)=0/25=0.00 \\
h(r_7)=1 & & p(r_7)=1/25=0.04 \\
h(r_8)=5 & & p(r_8)=5/25=0.20
\end{array}$$

- **Histogram equalization**
  - find a map f(x) such that the histogram of the modified (equalized) image is flat (uniform).
  - Key motivation: cumulative probability function of a random variable approximates a uniform distribution

Suppose h(t) is the new histogram

$$T(x) = \sum_{t=0}^{x} p(t)$$

Example

Normalised histogram function	**Intensity transformation function**
$p(r_1) = 0.32$	$T(r_1) = 0.32$
$p(r_2) = 0.16$	$T(r_2) = 0.32 + 0.16 = 0.48$
$p(r_3) = 0.12$	$T(r_3) = 0.32 + 0.16 + 0.12 = 0.60$
$p(r_4) = 0.08$	$T(r_4) = 0.32 + 0.16 + 0.12 + 0.08 = 0.68$
$p(r_5) = 0.08$	$T(r_5) = 0.76$
$p(r_6) = 0.00$	$T(r_6) = 0.76$
$p(r_7) = 0.04$	$T(r_7) = 0.80$
$p(r_8) = 0.20$	$T(r_8) = 1.00$

$T(r_1) = 0.32$	$h(r_1) = 0.32 \times 8 \rightarrow 3$
$T(r_2) = 0.48$	$h(r_2) = 0.48 \times 8 \rightarrow 4$
$T(r_3) = 0.60$	$h(r_3) = 0.60 \times 8 \rightarrow 5$
$T(r_4) = 0.68$	$h(r_4) = 0.68 \times 8 \rightarrow 5$
$T(r_5) = 0.76$	$h(r_5) = 0.76 \times 8 \rightarrow 6$
$T(r_6) = 0.76$	$h(r_6) = 0.76 \times 8 \rightarrow 6$
$T(r_7) = 0.80$	$h(r_7) = 0.80 \times 8 \rightarrow 6$
$T(r_8) = 1$	$h(r_8) = 1.00 \times 8 \rightarrow 8$

The 32% of the pixels have intensity r1. We expect them to cover 32% of the possible intensities.

The 48% of the pixels have intensity r2 or less. We expect them to cover 48% of the possible intensities.

The 60% of the pixels have intensity r3 or less. We expect them to cover 60% of the possible intensities.

…………………………………

3	8	5	5	5
3	3	3	6	8
8	8	5	5	3
4	4	3	6	4
3	3	8	6	4

Histogram equalisation algorithm I:

- Let $r_k, k = 1,2,\ldots,m$ be the intensities of the image
- Let $p(r_k)$ be its normalised histogram function.
- The intensity transformation function for histogram equalisation is

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

- That is, we add the values of the normalised histogram function from 1 to k to find where the intensity will be mapped.

- **Multiply cumulative values by the maximum gray-level value and round the results to obtain** $r_k^{'}$
- **Map the original gay-level value to the resulting value**

Algorithm II

- The two requirements on the operator are that

  (a) the output image should use all available gray levels

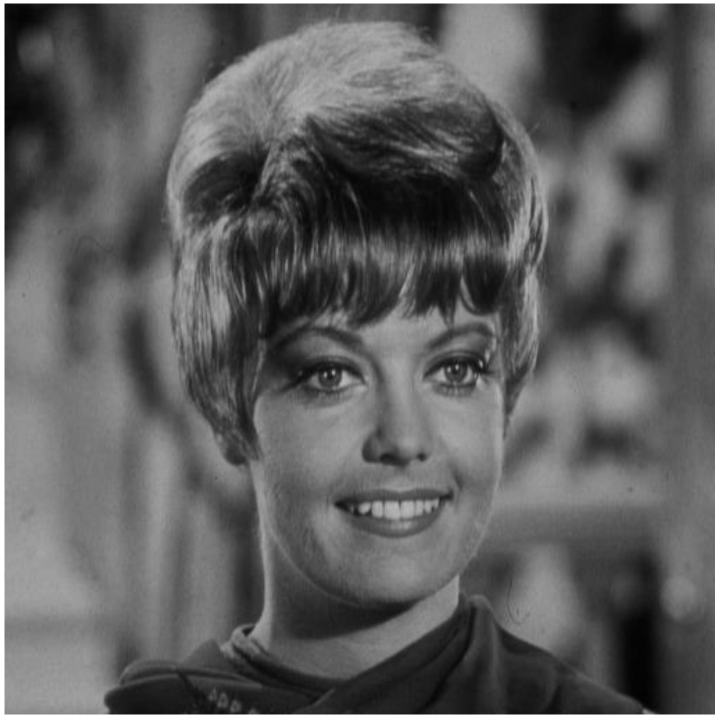
  This requirement means that the target output image uses all gray values $z = z1, z = z2, \ldots z = zn$

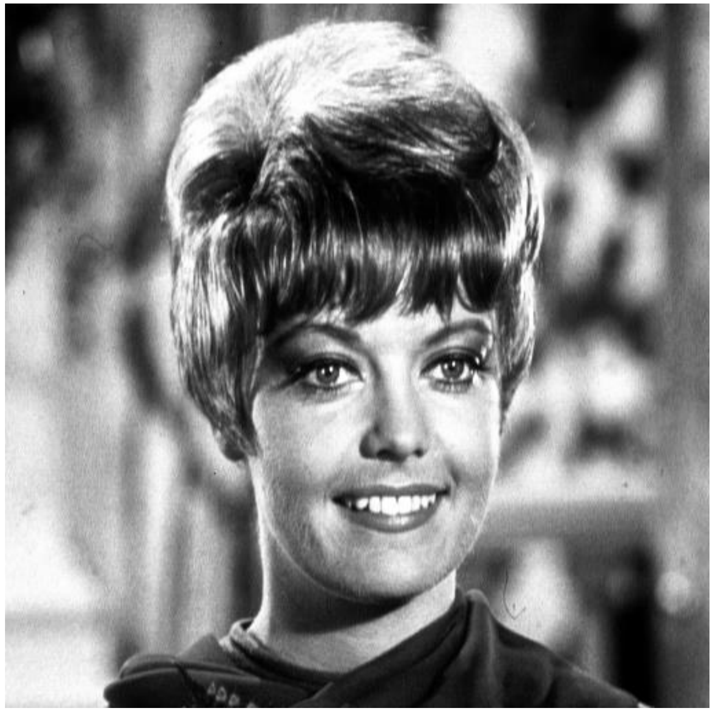
  (b) the output image has approximately the same number of pixels of each gray level.

  This requirement indicates each gray level zk is used approximately $q = (R \times C)/n$ $(n=255)$ times, where R, C are the number of rows and columns of the image.

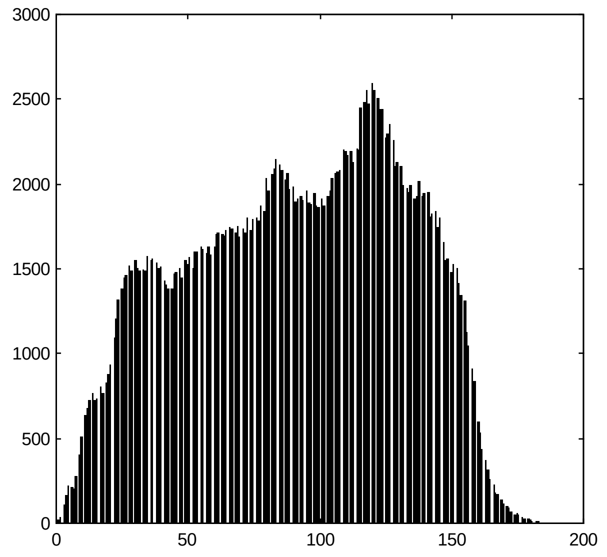
- Compute the average number of pixels per gray level.
- Starting from the lowest gray-level band, accumulate the number of pixels until the sum is closest to the average. All of these pixels are then rescaled to the new reconstruction levels.
- If an old gray-level band is to be divided into several new bands, either do it randomly or adopt a rational strategy----one being to distribute them by region.

Image Example

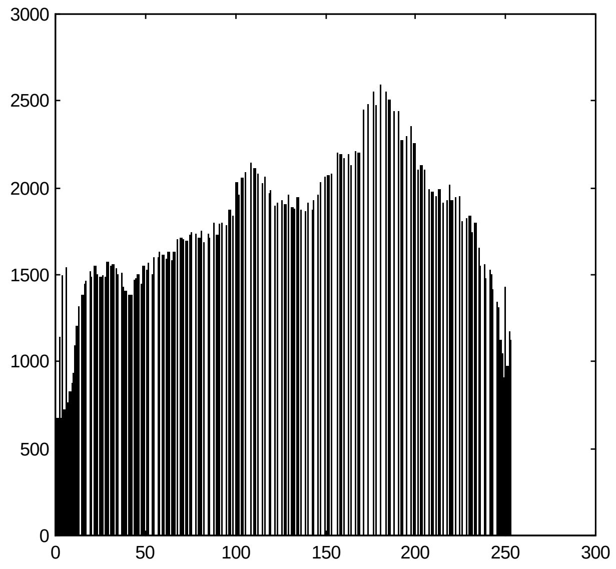

before


after

Histogram Comparison

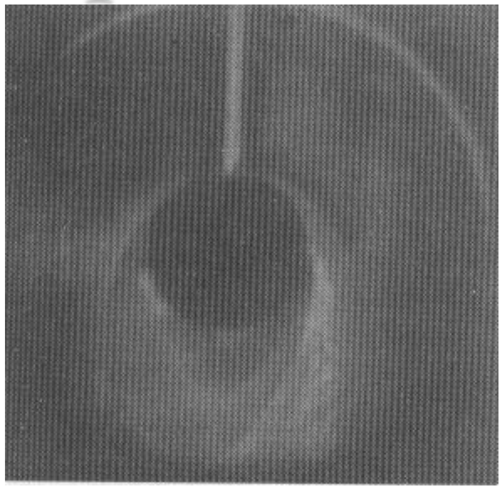


before equalization

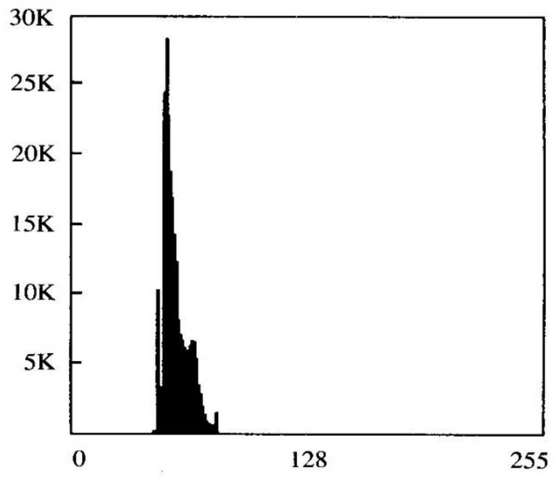


after equalization

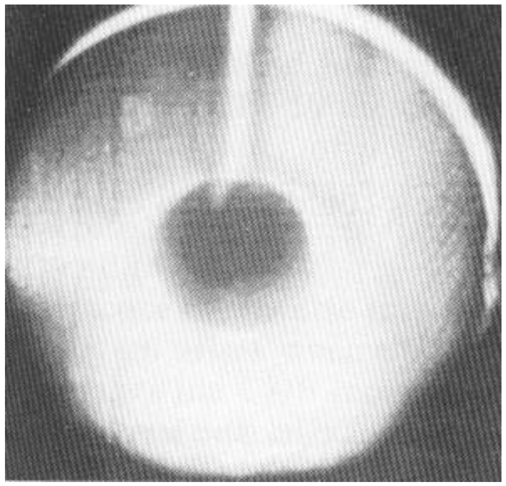
Example of Histogram Equalization



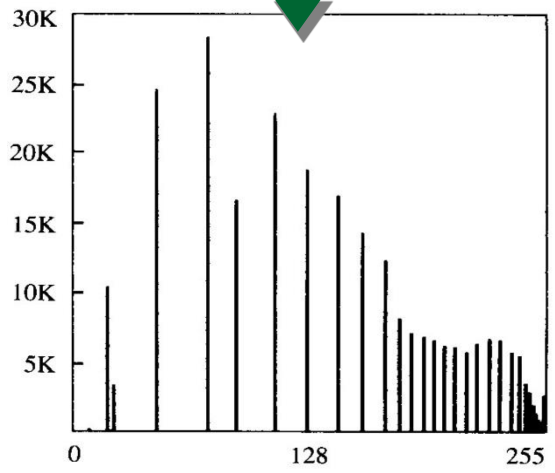
Original image



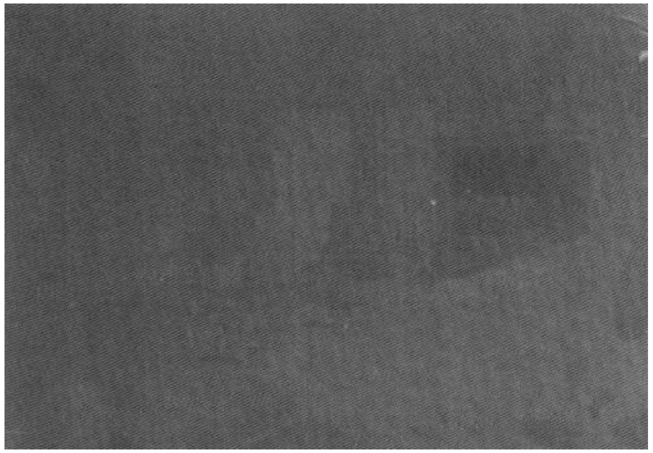
Original histogram

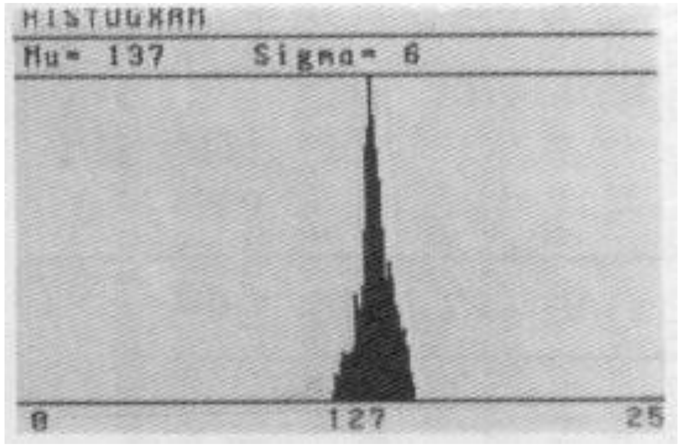


Modified image

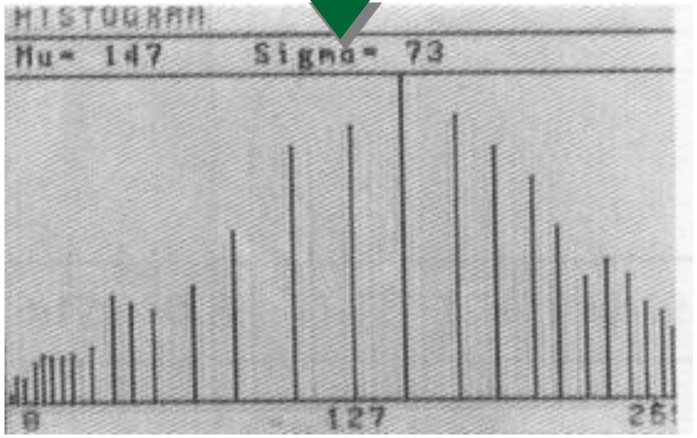


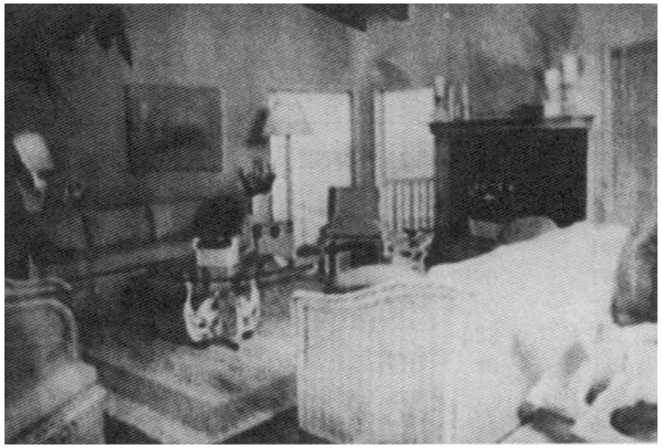
Histogram after equalization


Original image

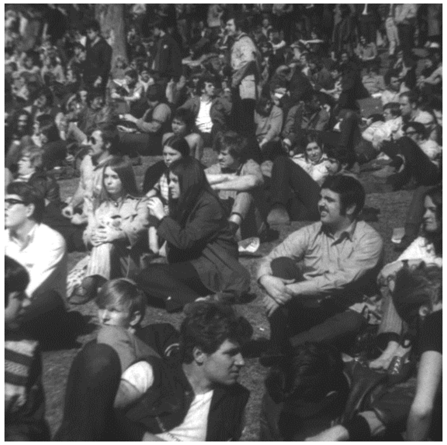


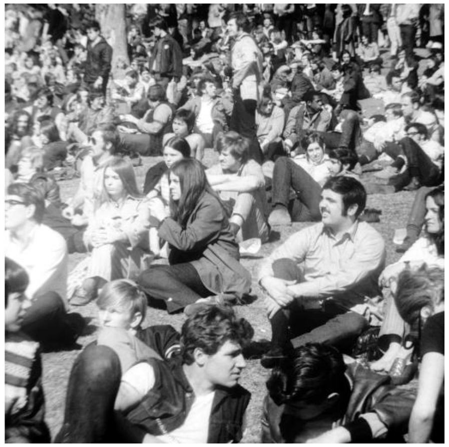
Original histogram

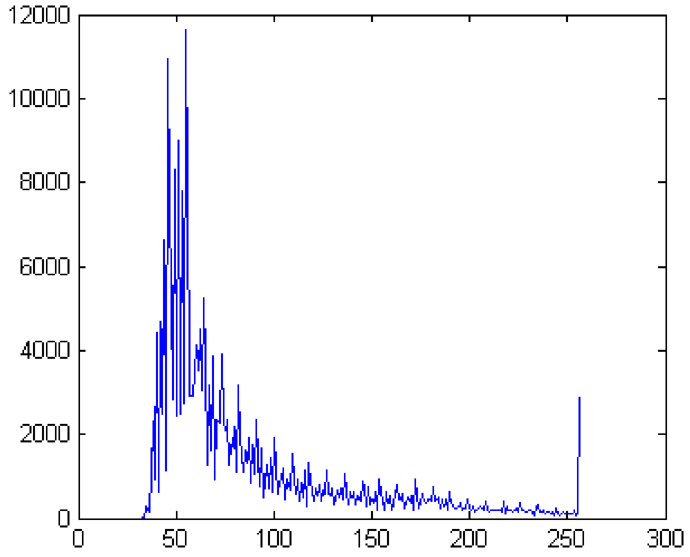


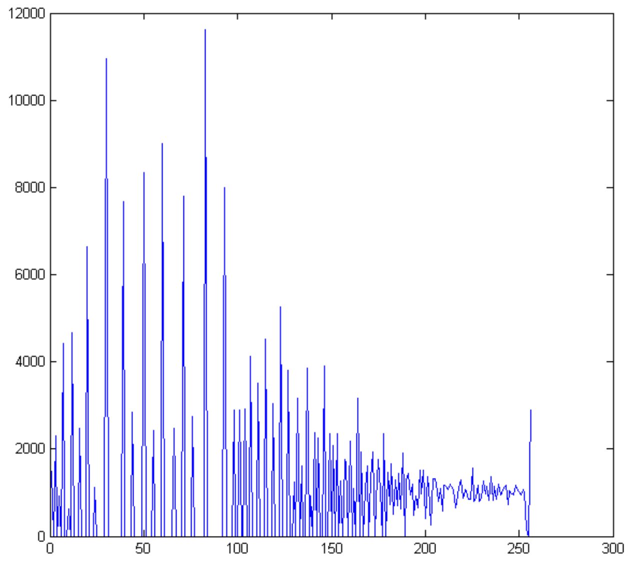

Modified image

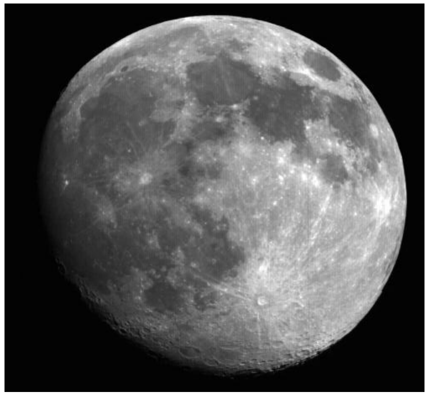
Histogram after equalization

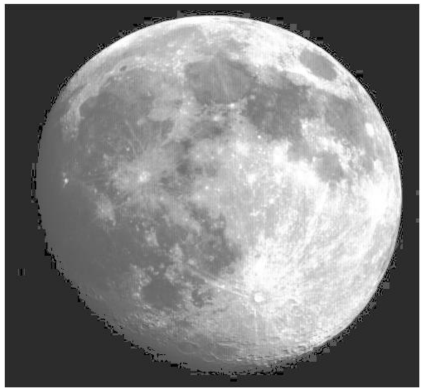


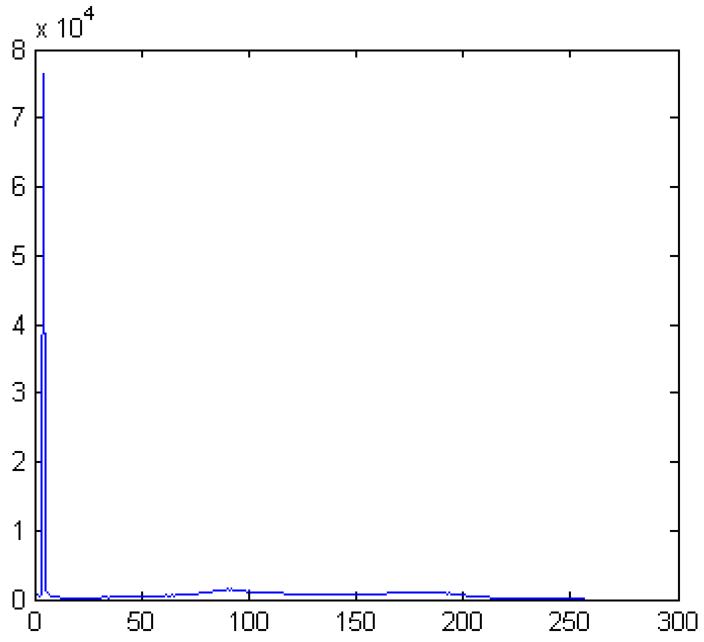


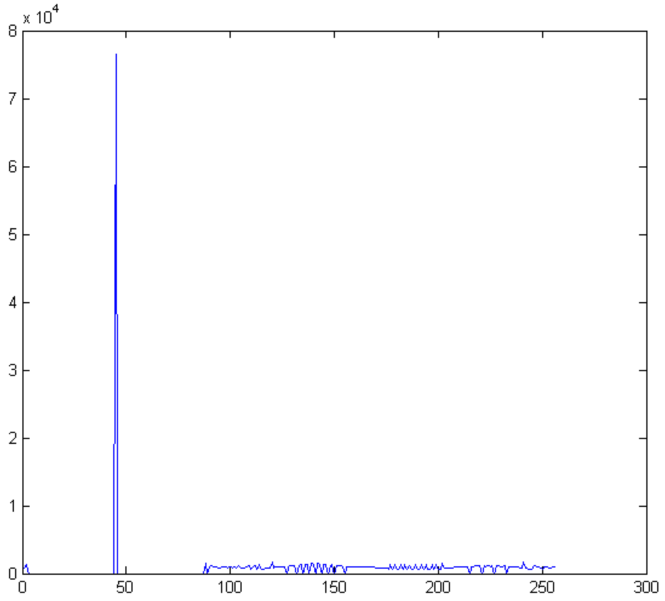

12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300


12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300



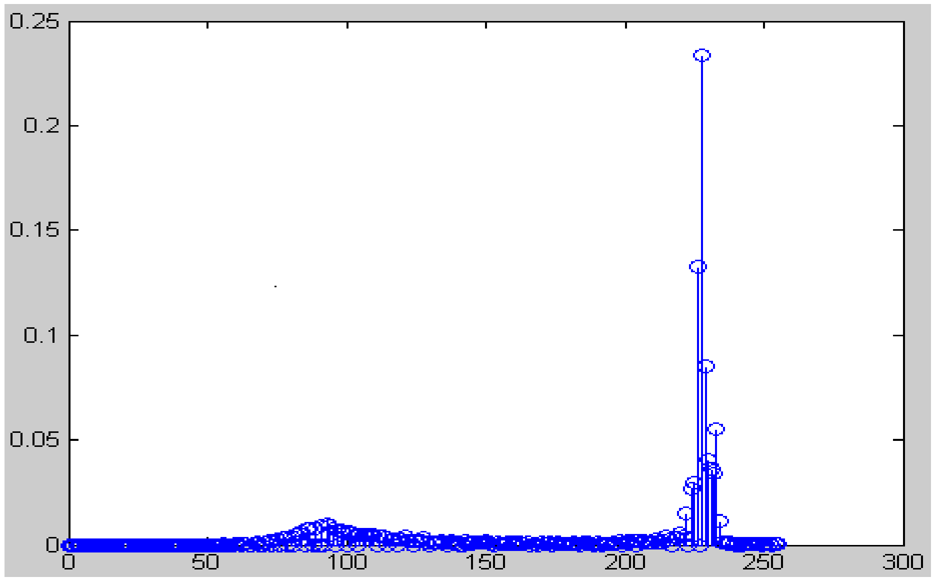


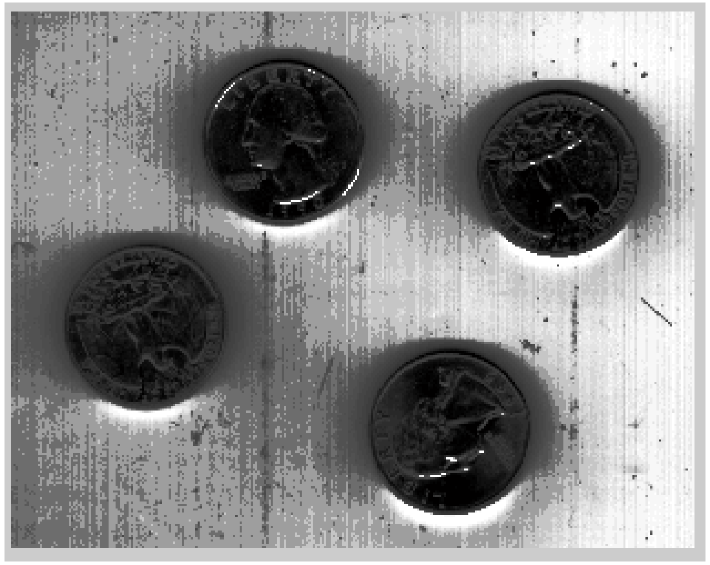

x 10^4
0
1
2
3
4
5
6
7
8
0
50
100
150
200
250
300

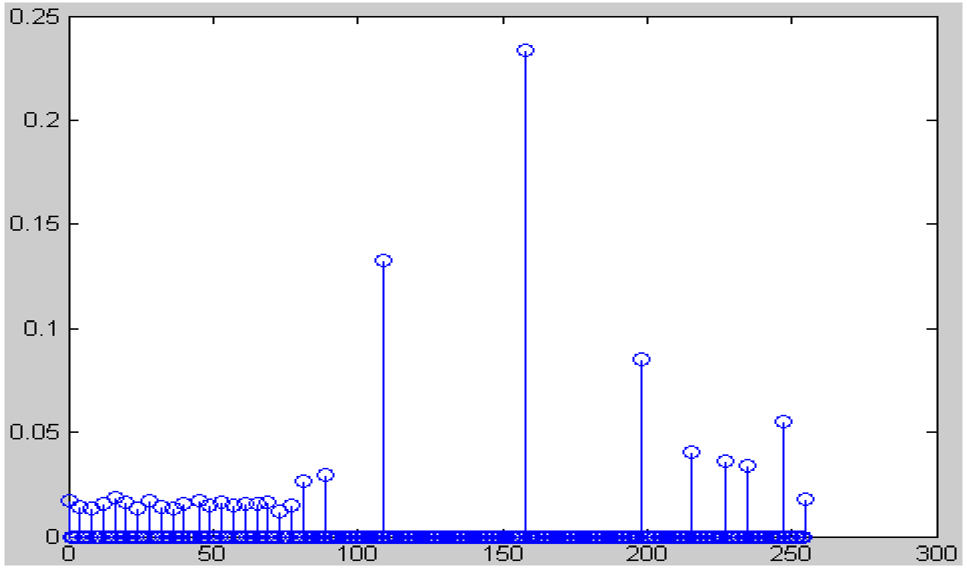

x 10^4
0
1
2
3
4
5
6
7
8
0
50
100
150
200
250
300

- Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

- Histogram equalization may not always produce desirable results, particularly if the given histogram is very narrow. It can produce false edges and regions. It can also increase image “graininess” and “patchiness.”





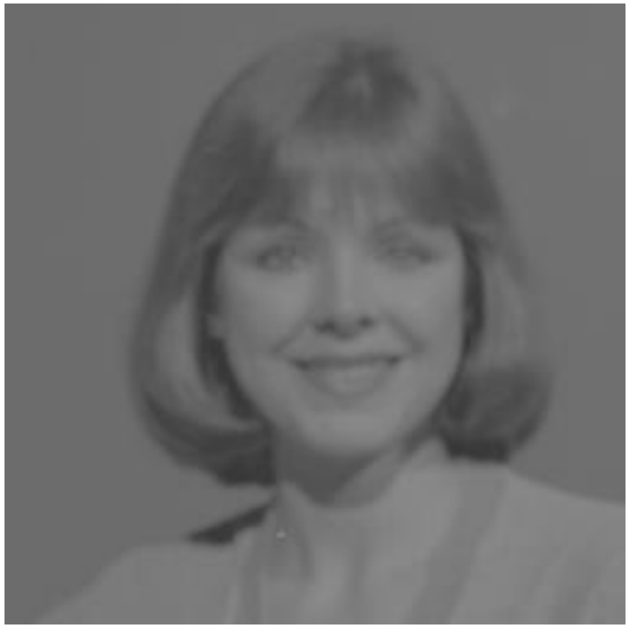


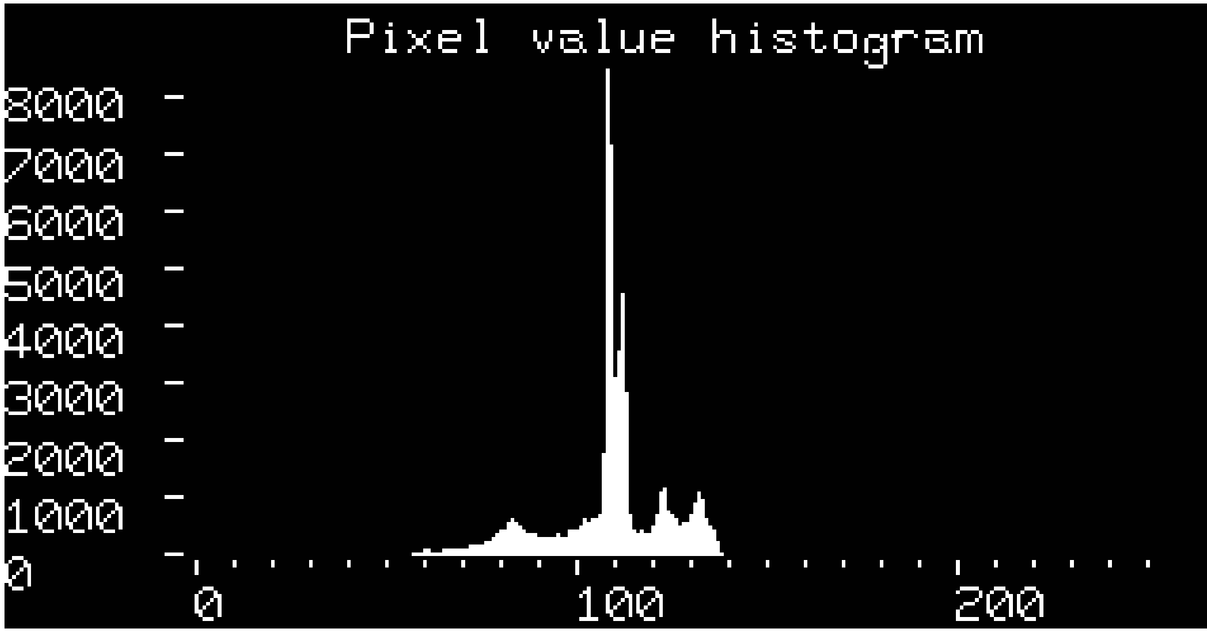
0.25
0.2
0.15
0.1
0.05
0
0
50
100
150
200
250
300

Application of the histogram

- You can use a histogram to determine if you used the correct exposure with your camera or scanner

An image has low contrast when the complete range of possible values is not used. Inspection of the histogram shows this lack of contrast.





- When performing post-processing in your digital darkroom you can use a histogram to :
  - Achieve a desired brightness and contrast range
  - Determine if highlights or shadows have lost detail
  - Identify possible posterization (色調分離)
  - Identify and correct color shifts

Posterize (多色調分色印照片)