Image Enhancement

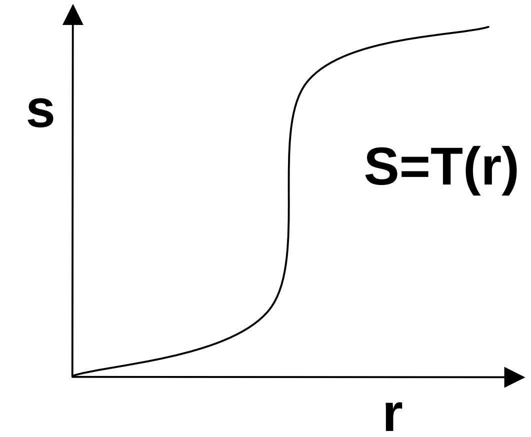
CONTRAST STRETCHING

This is a pixel-based operation, where a given gray level $r \in [0, L]$ mapped into a gray level $s \in [0, L]$ according to a transformation function:

$$s = T(r)$$

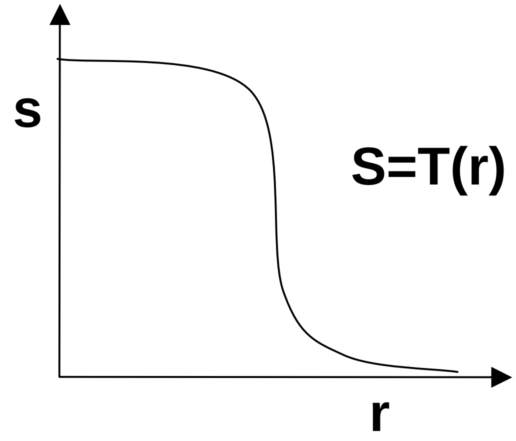
This process is mainly used to enhance done to handle low-contrast images occurring due to poor or non-uniform lighting conditions or due to non-linearity or small dynamic range of the imaging sensor.

Following examples shows some typical contrast stretching transformations.



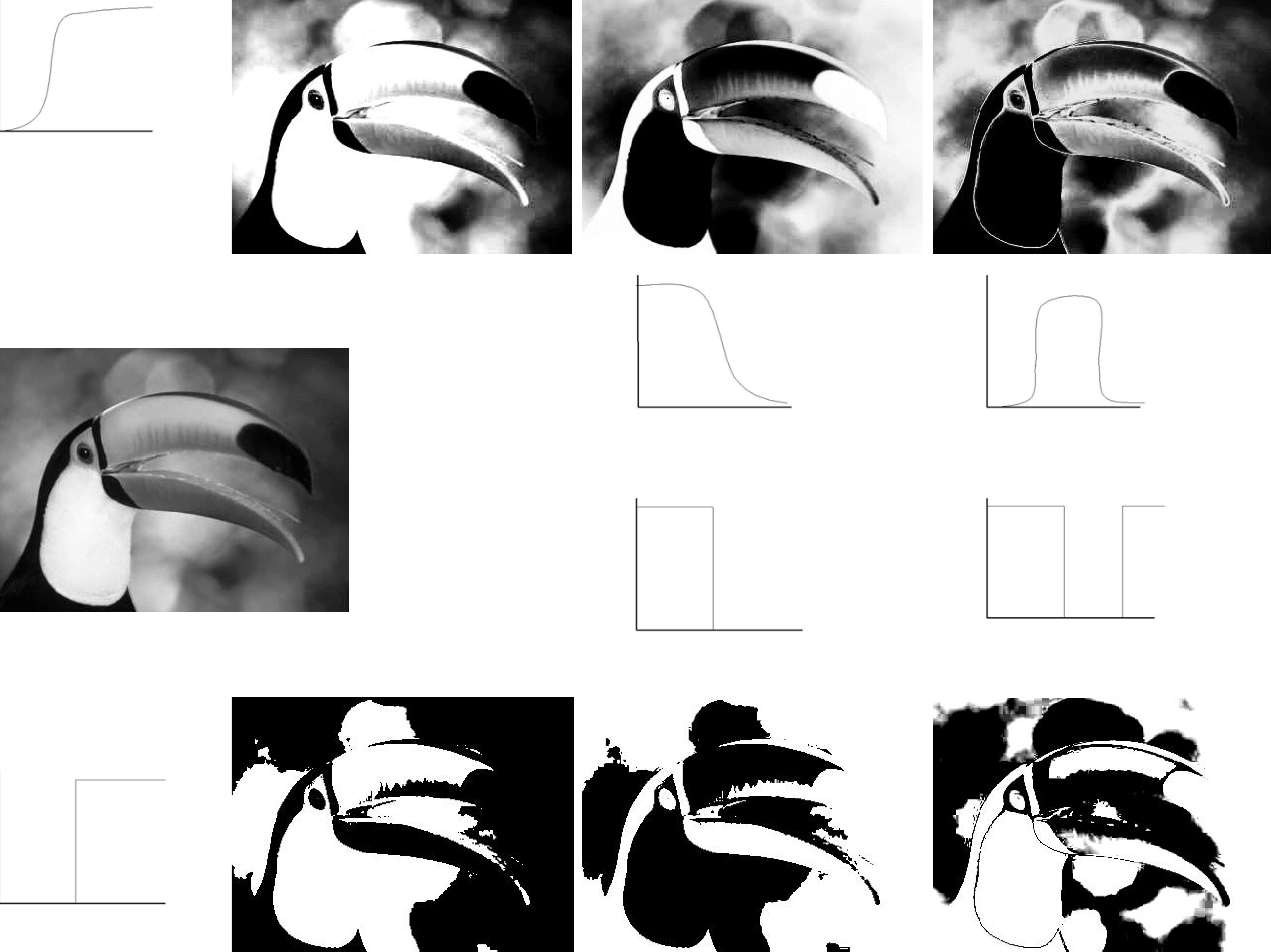
Original image

After contrast stretching



Original image **After contrast stretching**


original image
original image
original image
original image
original image



HISTOGRAM EQUALISATION

Here, the goal is to obtain a uniform histogram for the output image.

Some features of Histogram equalization are as follows:

- **Histogram equalization is a point process**
- **Histogram equalization causes a histogram with closely grouped values to spread out into a flat or equalized histogram.**
- **Spreading or flattening the histogram makes the dark pixels appear darker and the light pixels appear lighter.**
- **Histogram equalization does not operate on the histogram itself but uses the results of one histogram to transform the original image into an image that will have equalized histogram.**
- **Histogram equalization do not introduce new intensities in the image. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities.**

Mathematically, the discrete form of the transformation function for histogram equalization is given by

$$s_k = T(r_k) = \sum_{j=0}^{k} n_j / n = \sum_{j=0}^{k} P_r(r_j)$$

$$0 \le r_k \le 1, k = 0,1,2,...., L-1$$

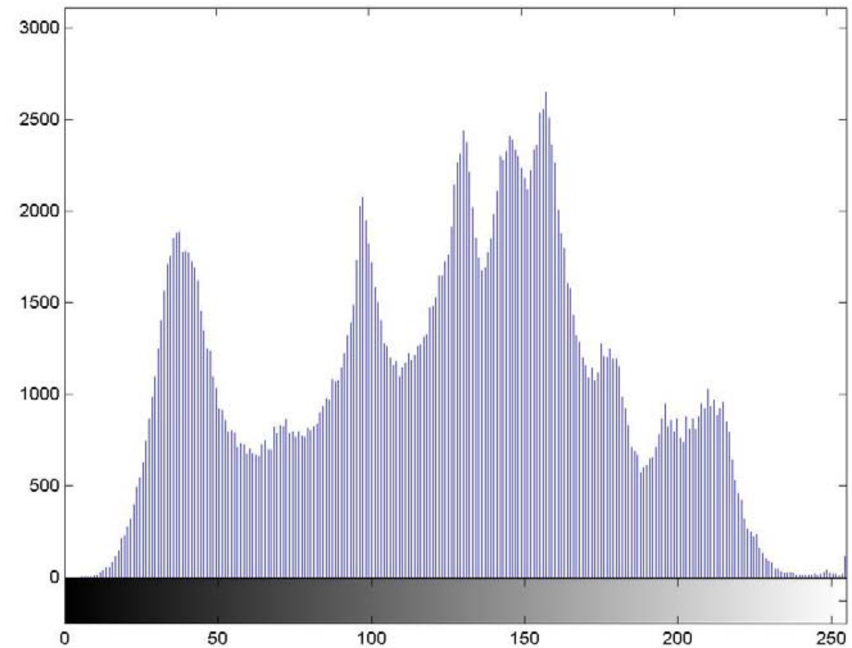
Where nj is the number of times the jth gray level appears in the image, L is the number of gray levels, Pr(rj) is the probability of the jth gray level and n is the total number of pixels in the image. A plot of Pr(rj) vs rj is the histogram of the image.

ALGORITHM

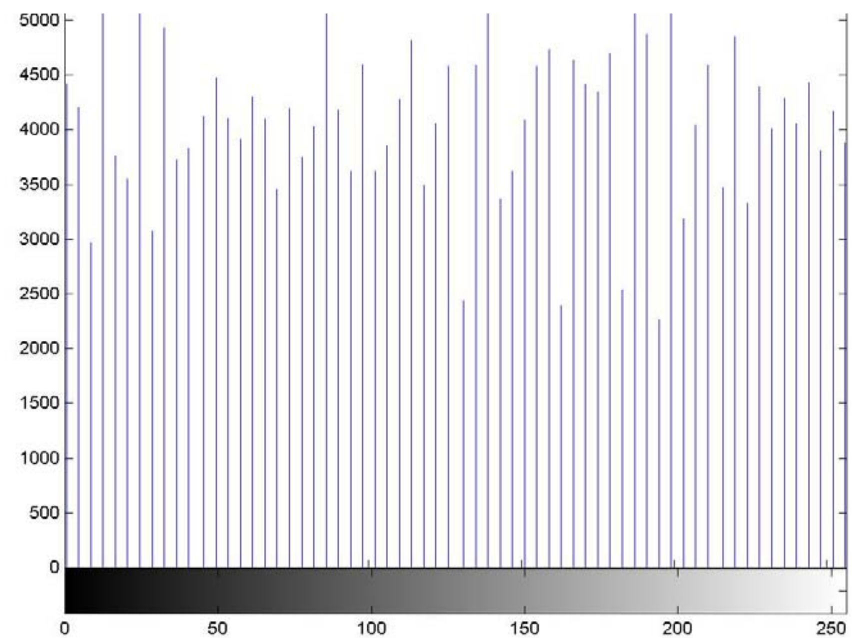
INPUT: Input image
OUTPUT: Output image after equalization

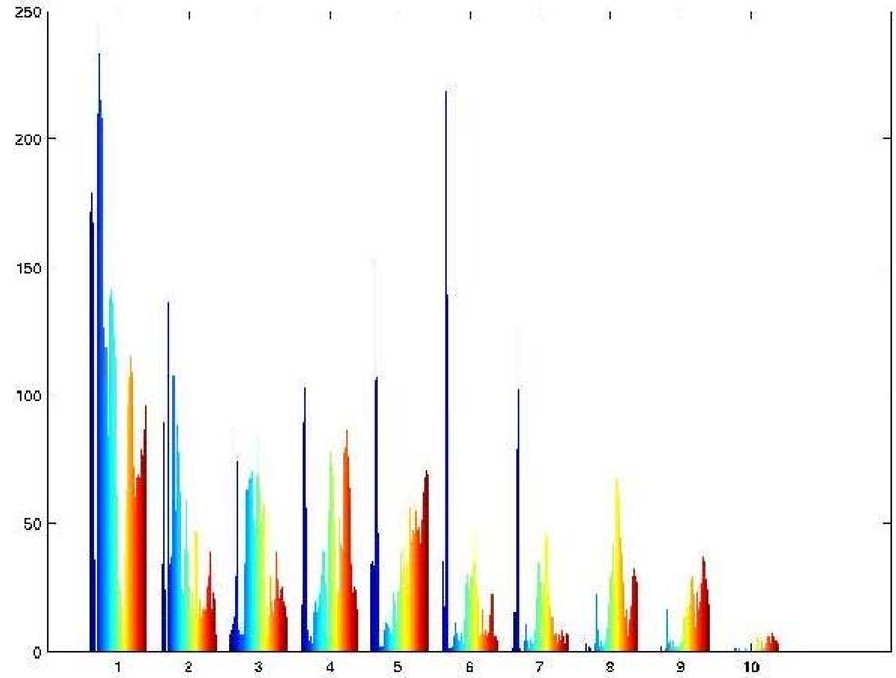
1. Compute histogram h(xi),
2. Calculate normalized sum of histogram
3. Transform input image to output image

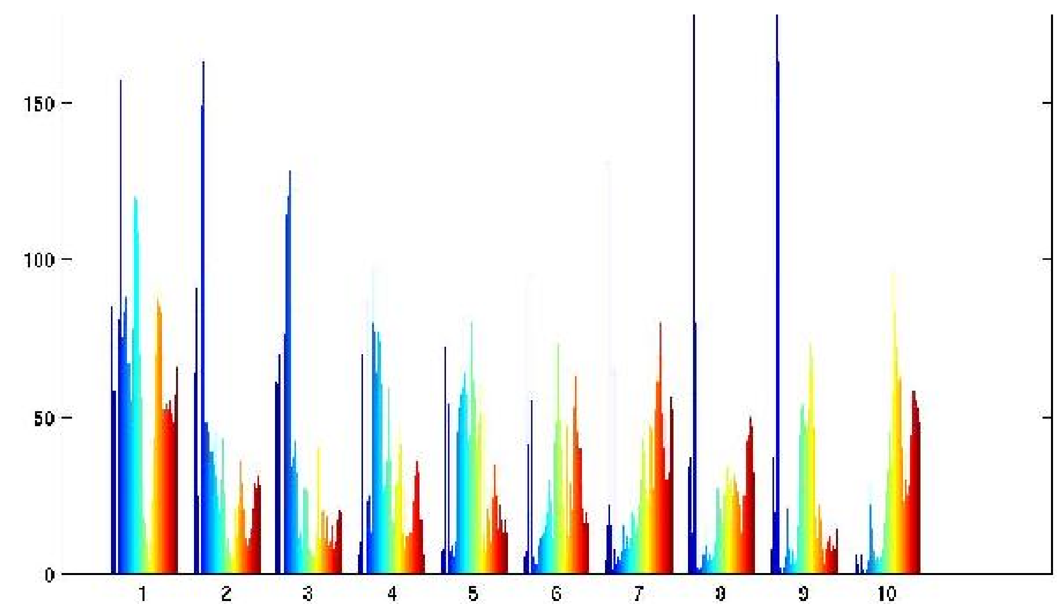

original image

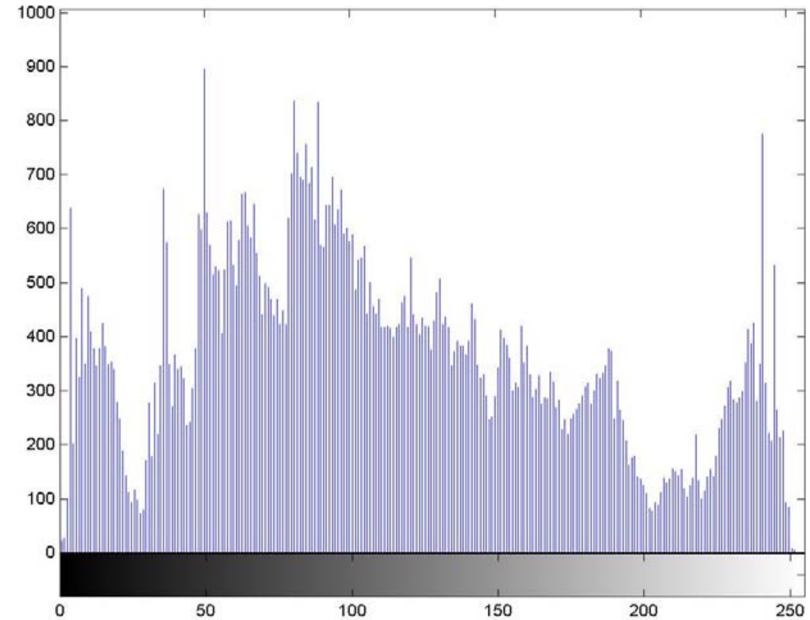

3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

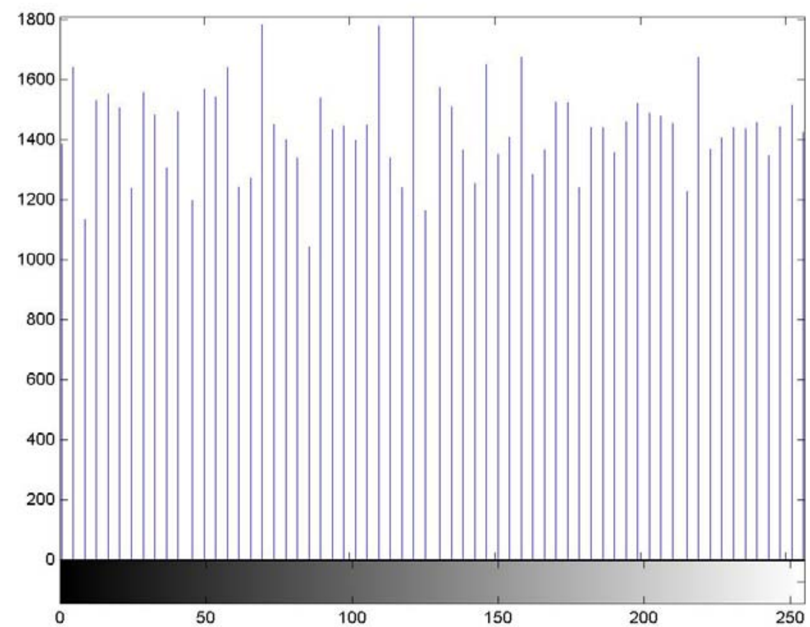

original image

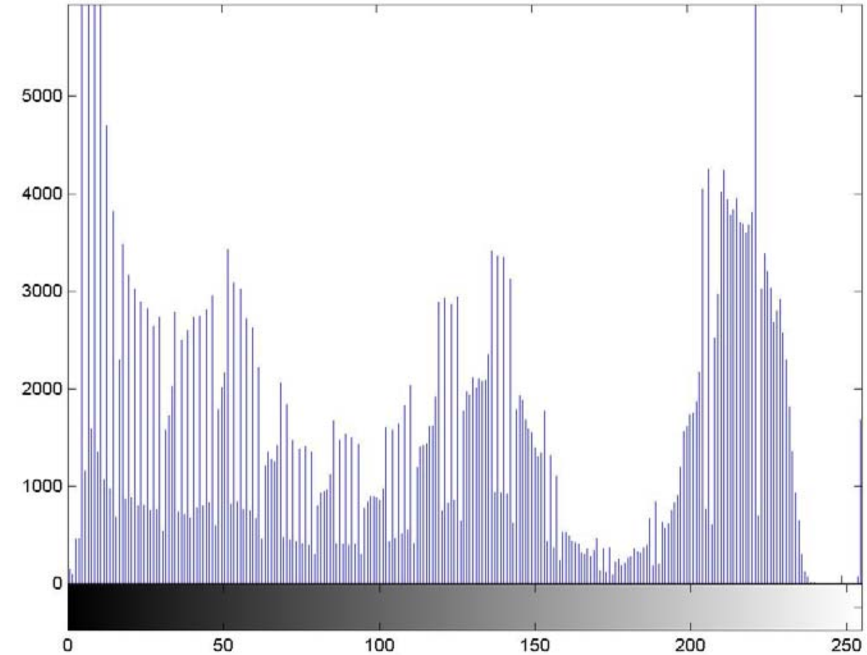

5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

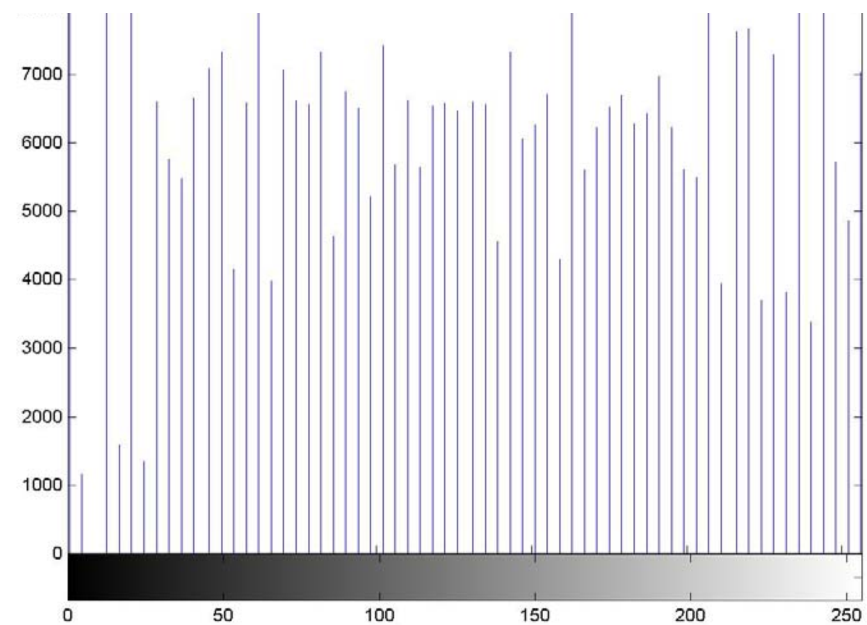

250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10

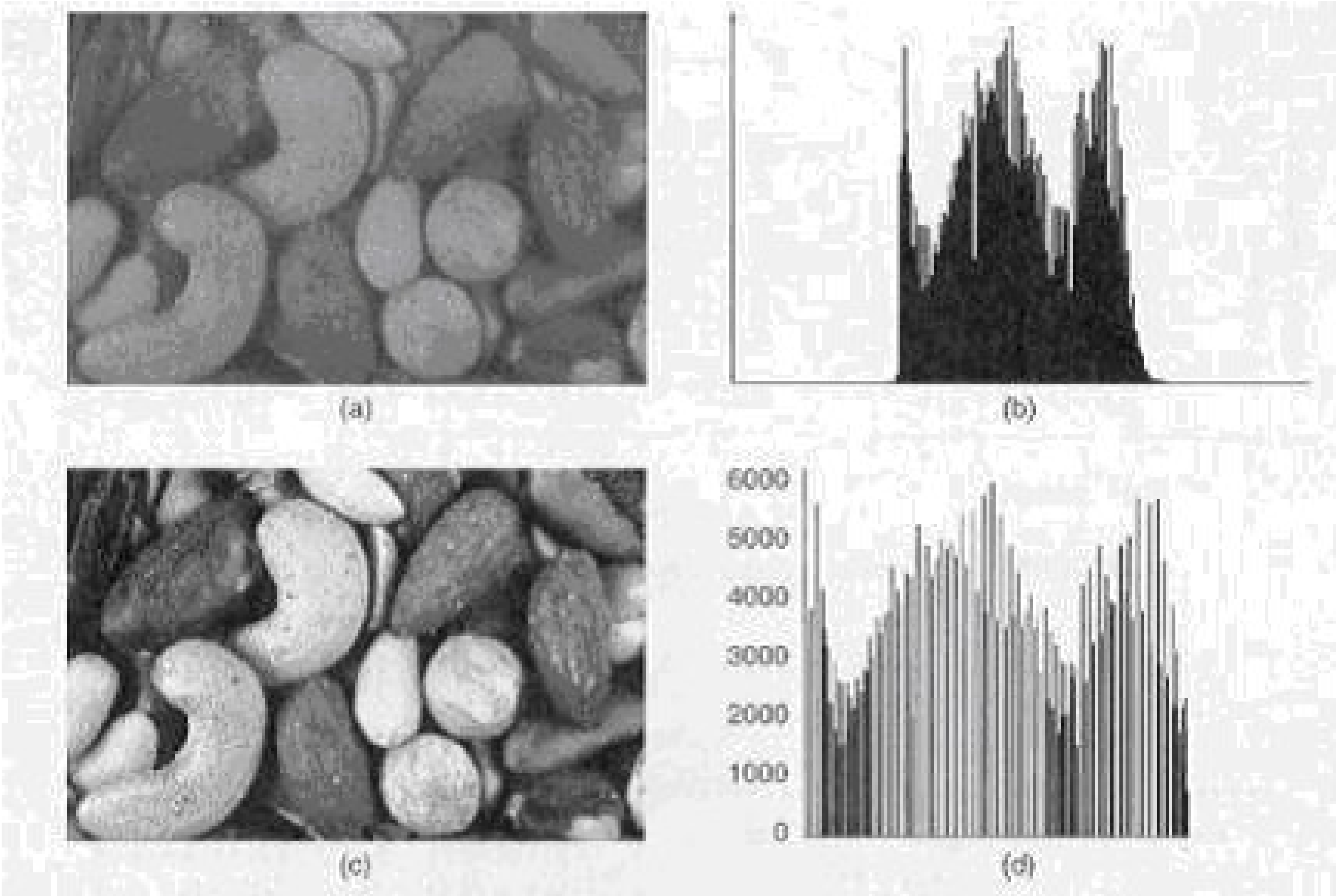

150
100
50
0
1
2
3
4
5
6
7
8
9
10


1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250


1800
1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

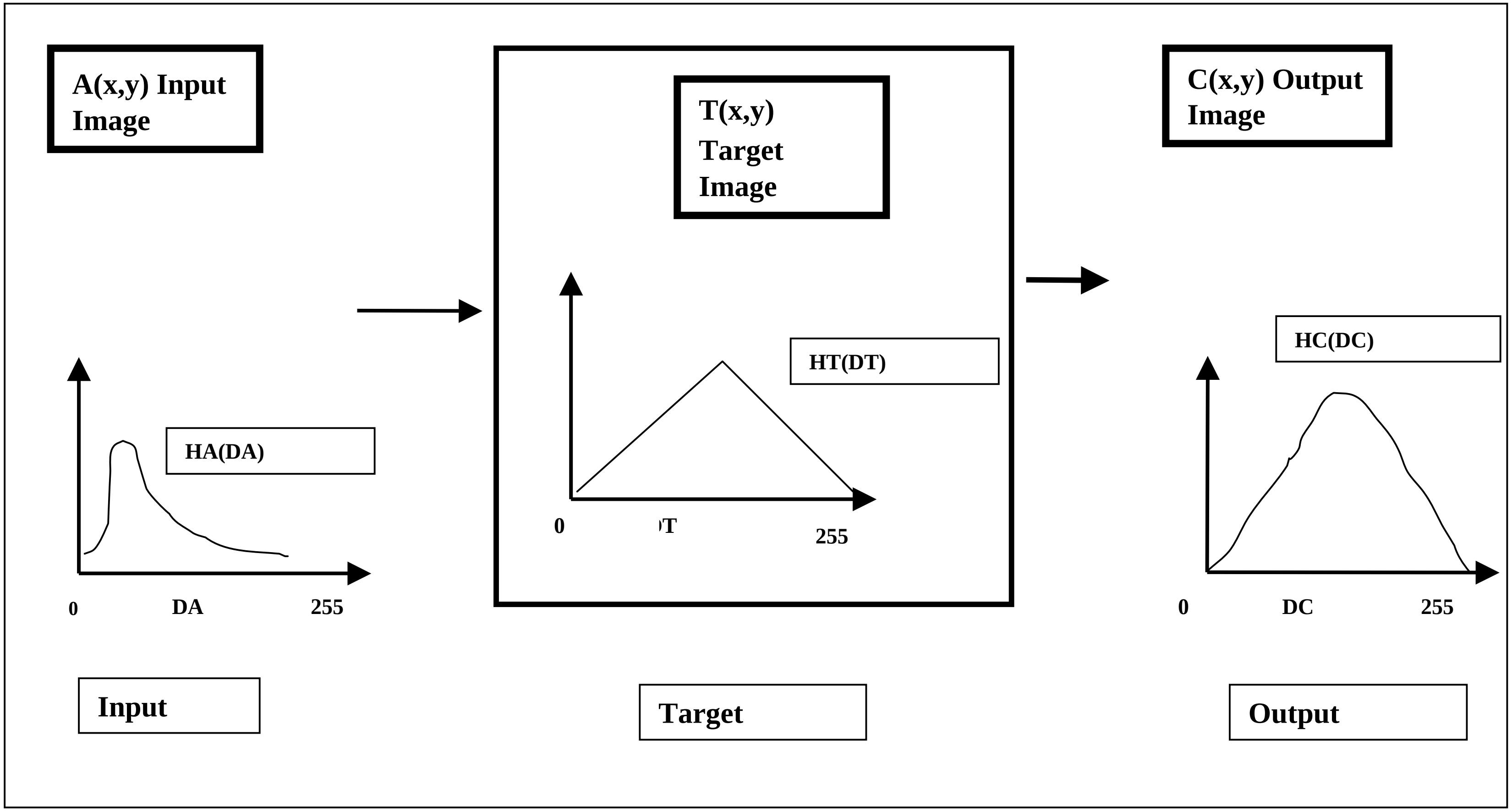

5000
4000
3000
2000
1000
0
0
50
100
150
200
250


7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250


(a)
(b)
6000
5000
4000
3000
2000
1000
0
(c)
(d)

Histogram Specification

Histogram specification method develops a gray level transformation such that the histogram of the output image matches that of the pre-specified histogram of a target image. Figure below shows the flow diagram of histogram specification method.



ALGORITHM

INPUT: Input image, Target image
OUTPUT: Output image that has the same characteristic as the target image

Steps:

- **Read the Input image and the Target image.**
- **Obtain the histogram of the input image and the target image.**
- **Equalize the input and the target images using the equation (1).**
- **Calculate the transformation function G of the target Image.**
- **Map the original image gray level r_k to the final gray level z_k**

In Histogram specification method, histogram is first obtained for both the input and target image. The histogram is then equalized using the formula:

$$s_k = T(r_k) = \sum_{j=0}^{k} n_j / n = \sum_{j=0}^{k} P_r(r_j)$$

$$0 \le r_k \le 1, k = 0,1,2,\ldots,L-1$$

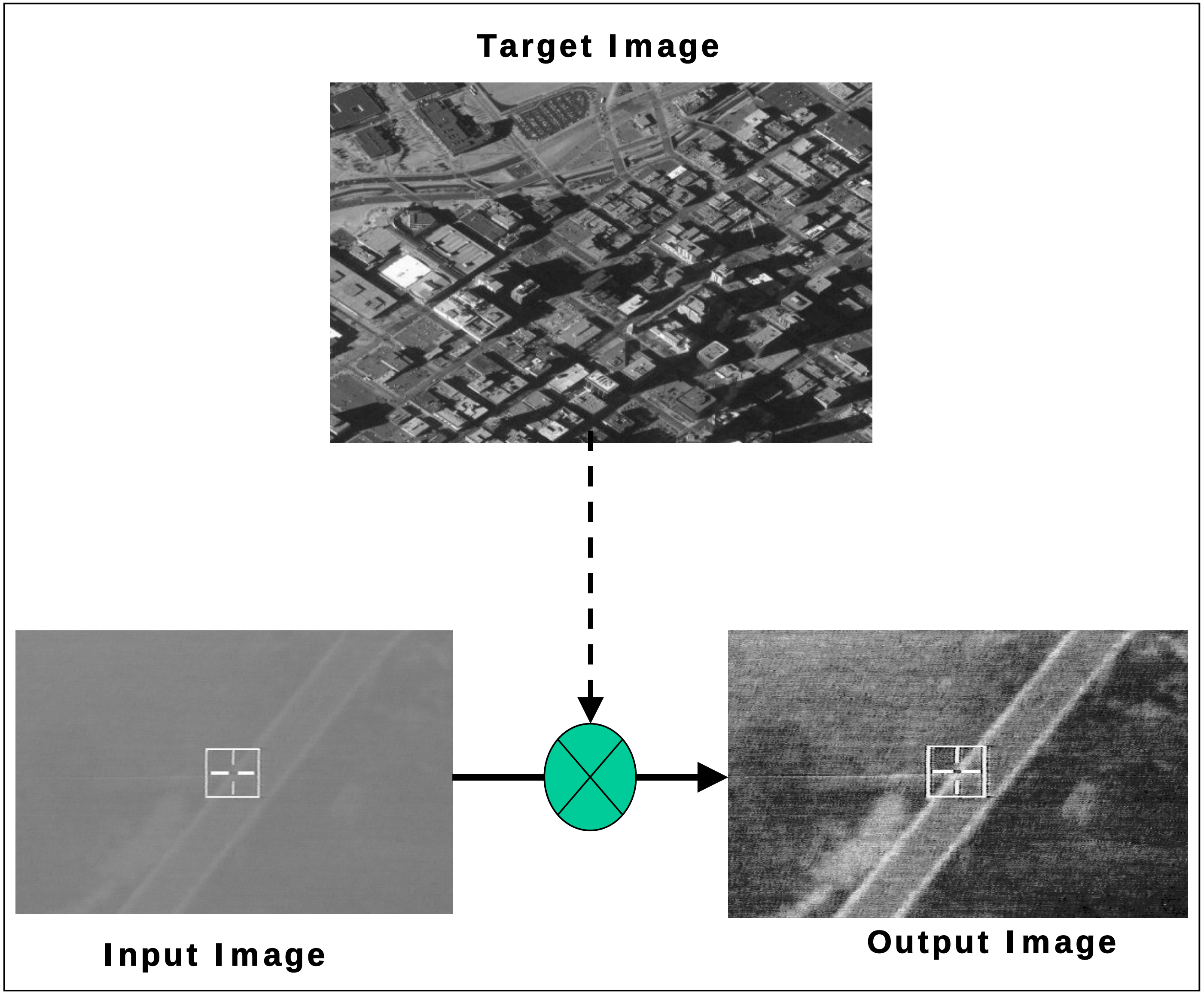
The transformation function should be a monotonically increasing function in the interval 0 and L - 1. The transformation function G is obtained from the given Pz (Z) as:

$$V_k = G(Z_k) = \sum_{i=0}^{k} P_z(Z_i) = S_k; k = 0,1,...., L-1$$

z_k is computed for each value of s_k, such that:
z_k must satisfy the equation **$G(z_k) - s_k = 0$**

Since we are dealing with integers, the closest we can get to satisfy the equation; i.e. $z_k = \varepsilon$ for each value of k, where ε is the smallest integer in the interval k=0,1,2… L-1

For each pixel in the original image if that value of the pixel is r_k, the value is mapped to its corresponding level s_k and then the level s_k is mapped to the final level z_k.


Target Image
Input Image
Output Image