Digital Image Processing

Lecture 3

Image Pre-processing

(Image Color Transformations)

Lecturer: Associate Prof. Lý Quốc Ngọc

3. **Image Pre-processing**

3.1. Color Transformations

3.2. Geometric Transformations

3.3. Local Pre-processing

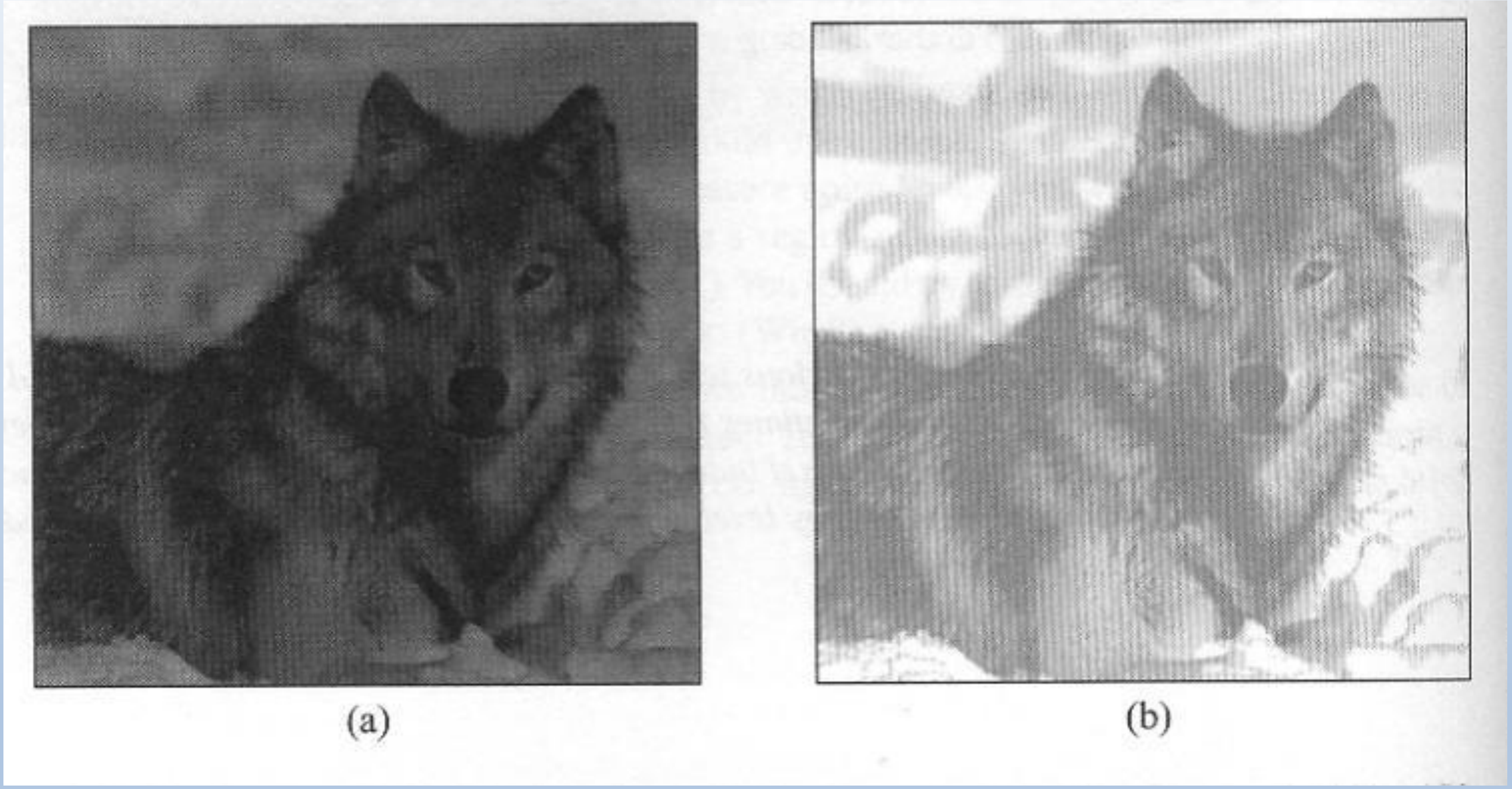
3.1. Color Transformations

3.1.1. Linear mapping

3.1.2. Non-linear mapping

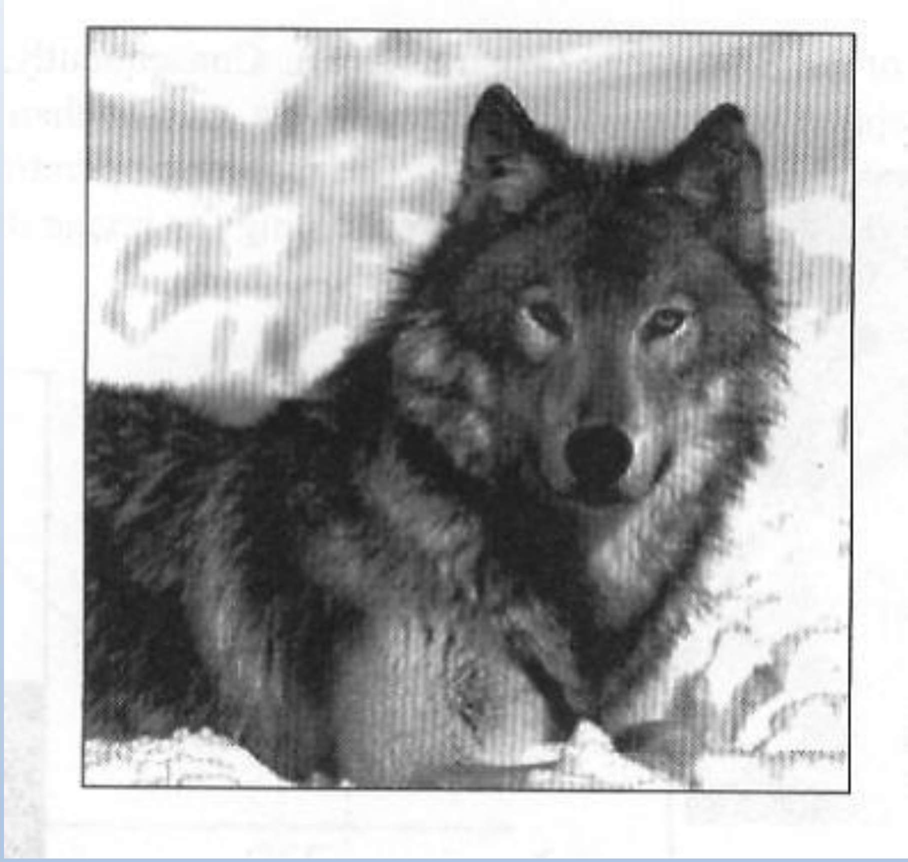
3.1.3. Probability Density Function-based mapping

3.1.1. Linear mapping



3.1.1. Linear mapping

(a)



3.1.1. Linear mapping

- **Brightness modification**

$$g(x,y) = f(x,y) + b$$

- **Contrast modification**

$$g(x,y) = a.f(x,y)$$

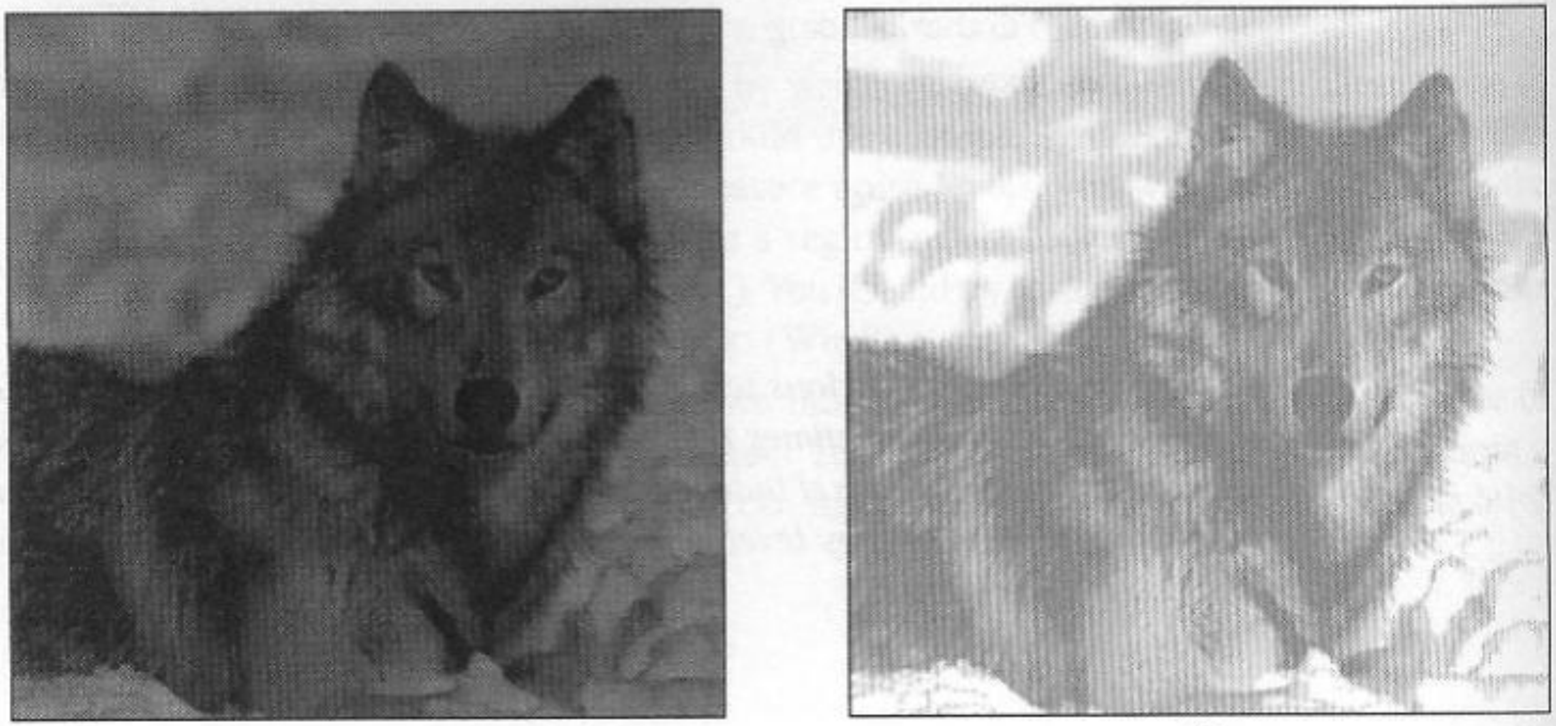
- **Brightness**+ Contrast **modification**

$$g(x,y) = a.f(x,y) + b$$

3.1.1. Linear mapping

- **Brightness modification**

$$g(x,y) = f(x,y) + b$$



(a) (b)

3.1.1. Linear mapping

- **Contrast modification**

$$g(x,y)=a.f(x,y)$$

3.1.1. Linear mapping

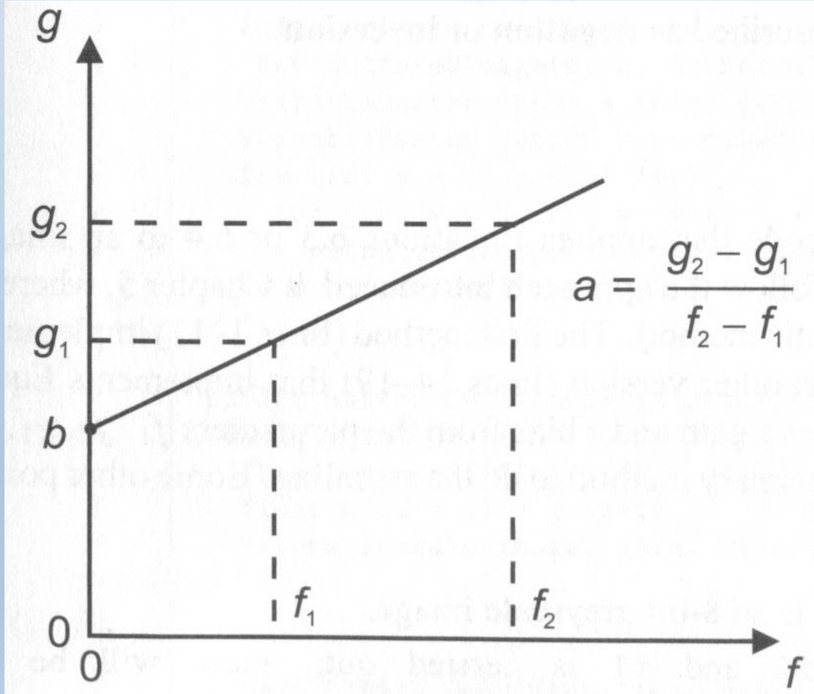
- **Brightness+ Contrast modification**

$$g(x, y) = a.f(x, y) + b$$

3.1.1. Linear mapping

- **Map a particular range of grey levels onto a new range**

Ex: map image from range of grey levels **[f_1 f_2]** onto range **[g_1 g_2]**.



3.1.2. Non-linear mapping

?

3.1.2. Non-linear mapping

?

3.1.2. Non-linear mapping

- **Logarithmic mapping function**

$$g(x, y) = c \log f(x, y)$$

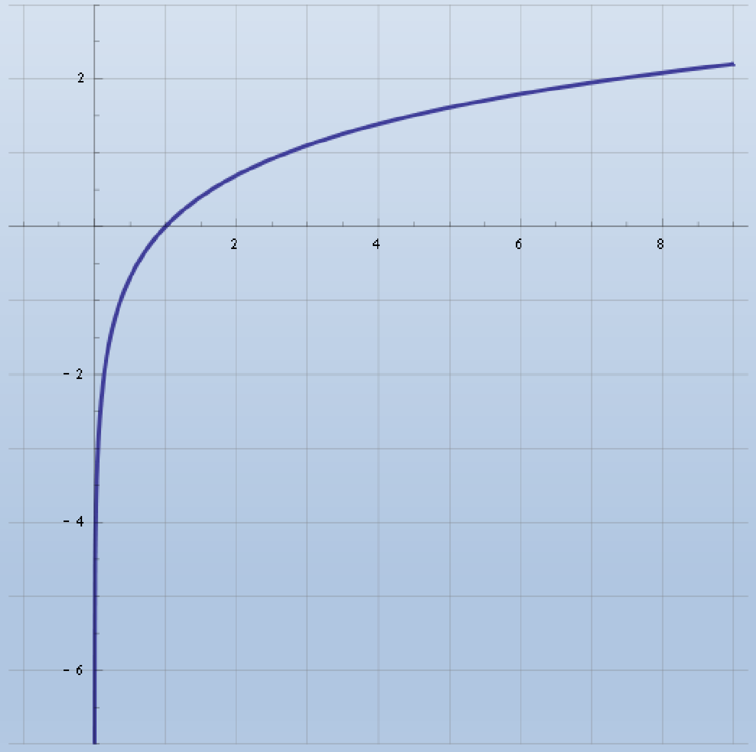
- **Exponential mapping function**

$$g(x, y) = e^{f(x,y)}$$

3.1.2. Non-linear mapping

- **Logarithmic mapping function**

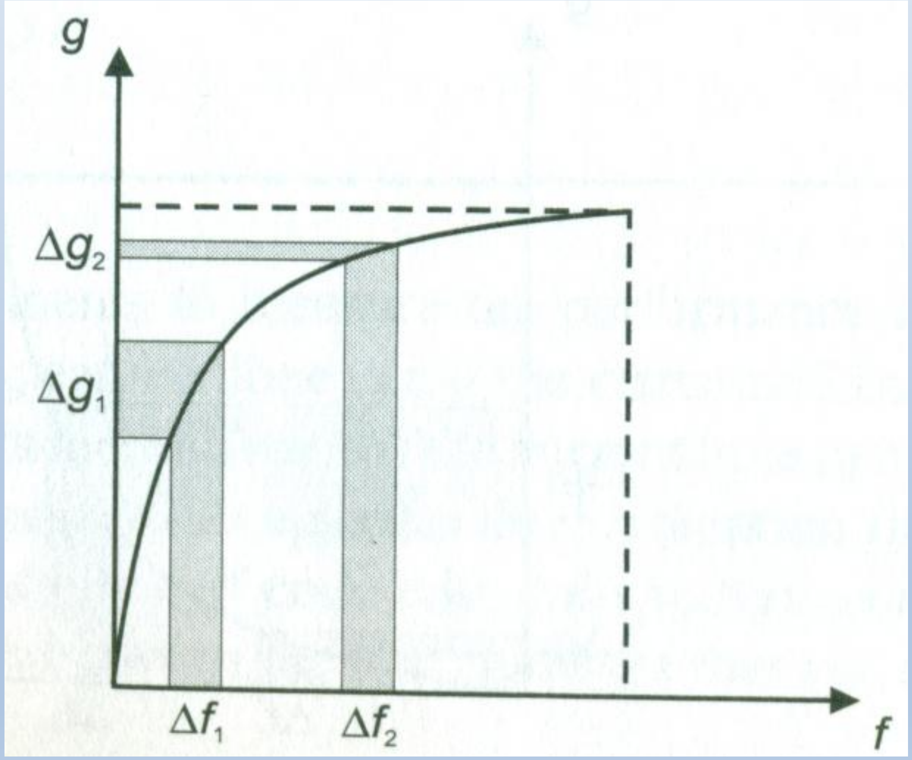
$$g(x, y) = c \log f(x, y)$$



3.1.2. Non-linear mapping

- **Logarithmic mapping function**

$$g(x, y) = c \log f(x, y)$$



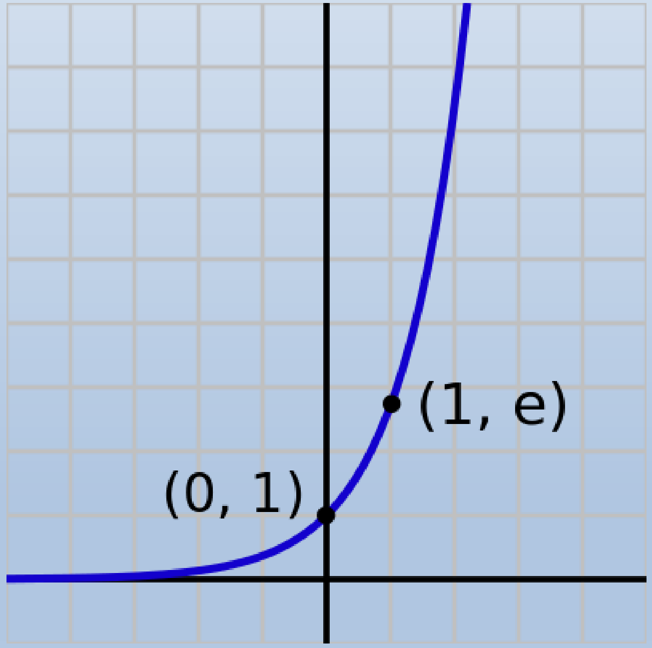
3.1.2. Non-linear mapping

- **Logarithmic mapping function**

3.1.2. Non-linear mapping

- **Exponential mapping function**

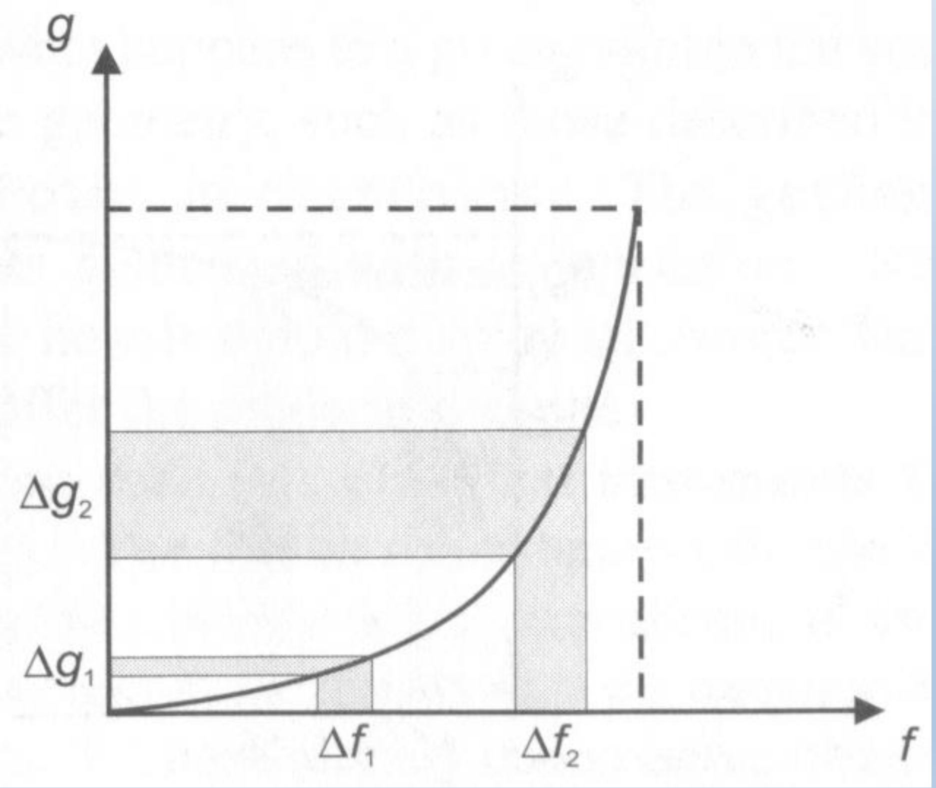
$$g(x,y) = e^{f(x,y)}$$



3.1.2. Non-linear mapping

- **Exponential mapping function**

$$g(x,y) = e^{f(x,y)}$$



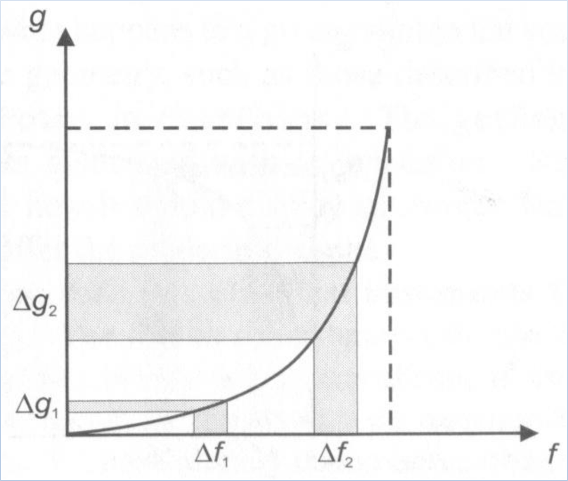
3.1.2. Non-linear mapping

- **Ex: Exponential mapping function**

?

3.1.2. Non-linear mapping

- **Ex: Exponential mapping function**



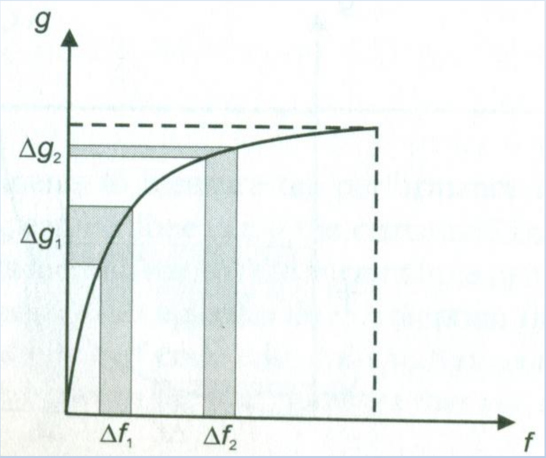
3.1.2. Non-linear mapping

- **Ex: Logarithmic mapping function**

?

3.1.2. Non-linear mapping

- **Ex: Logarithmic mapping function**



3.1.3. Probability Density Function-based mapping

3.1.3. Probability Density Function - based mapping

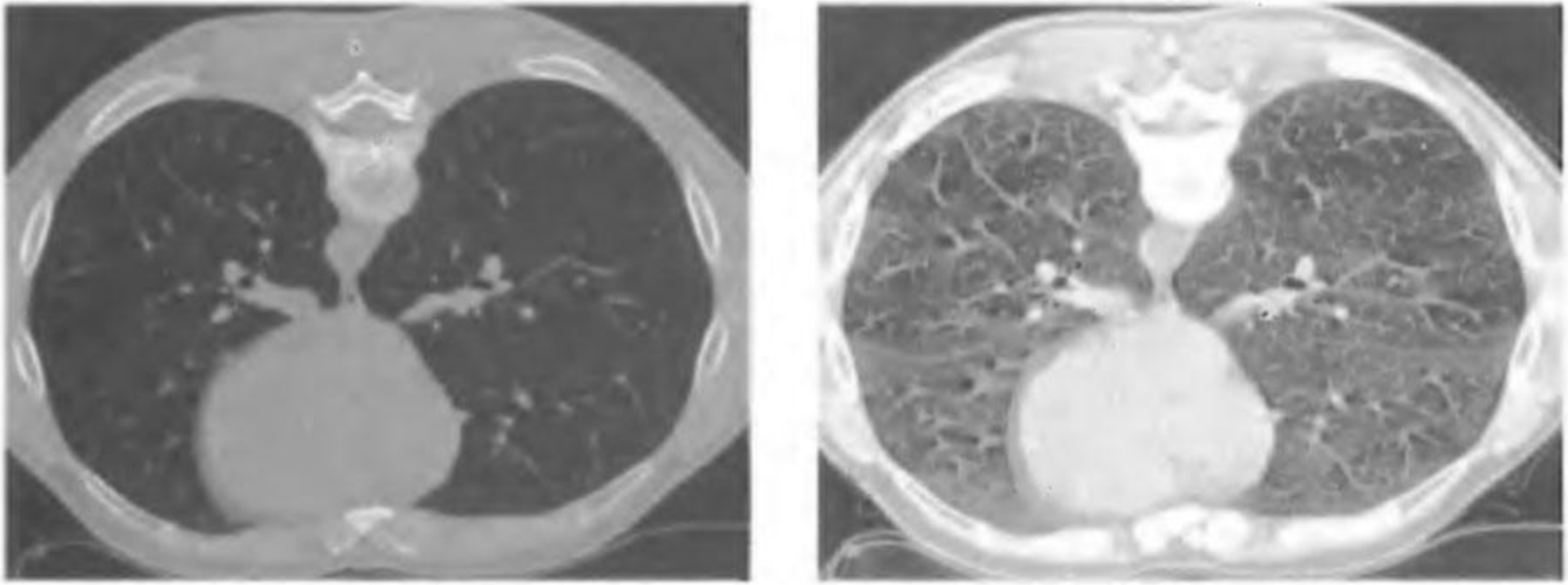
3.1.3.1. Histogram Equalization

Problem statement

Given image **f(x,y)**, it is necessary to define the nonlinear transformation **T** such that the output image **g=T(f)** whose **pdf is uniform**.

3.1.3. Probability Density Function - based mapping

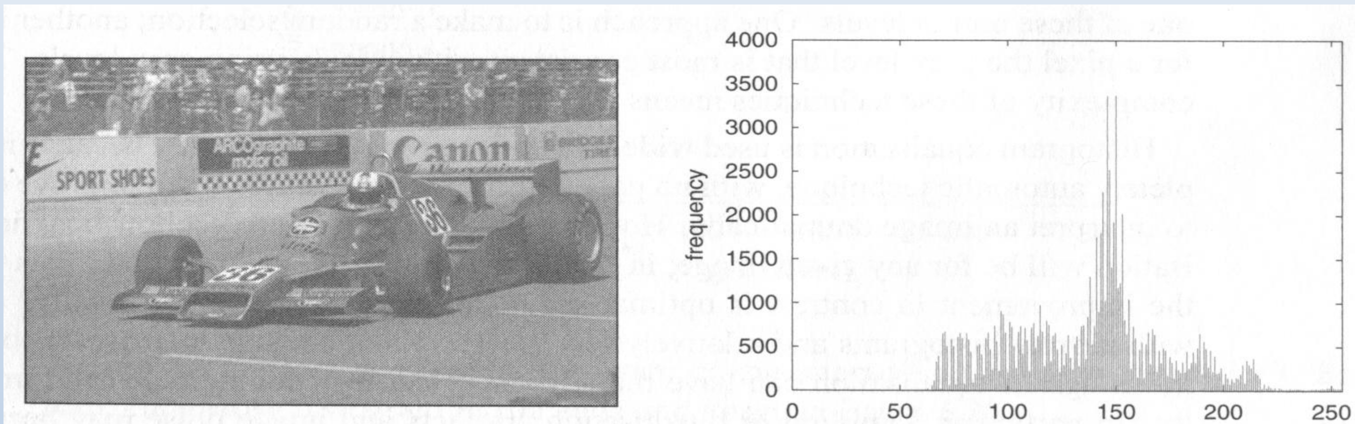
3.1.3.1. Histogram Equalization

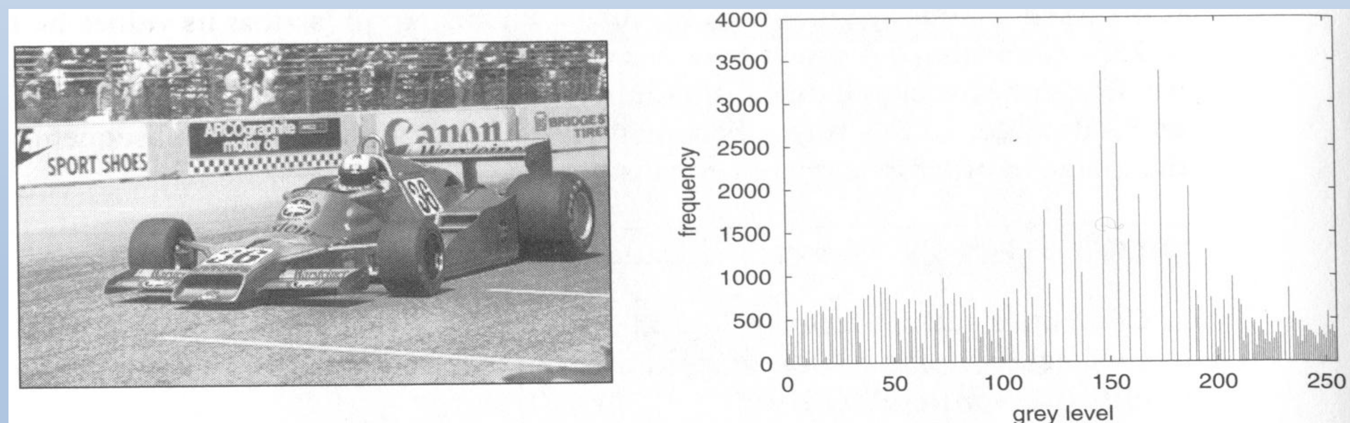


(a) (b)

3.1.3. Probability Density Function - based mapping

3.1.3.1. Histogram Equalization





3.1.3. Probability Density Function - based mapping

3.1.3.1. Histogram Equalization

Method

$$g = T(f) \Rightarrow p_g(g) = \frac{p_f(f)}{|dT(f)/df|}$$

$$p_g(g) = 1 \Rightarrow T(f) = \int_0^f p_f(w)dw$$

$$g_k = T(f_k) = \sum_{j=0}^{k} p_f(f_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization (**Algorithm)**

Step1. Create an array H of length nG initialized with 0 values (for an NxM image f of nG grey-levels).

Step2. Form **the image histogram** of f, save to H

$$H[f(x,y)] += 1$$

Step3. Form **the cumulative image histogram** of f, save to T

$$T[0] = H[0]; \quad T[p] = T[p-1] + H[p], \quad p = 1,2,...,nG-1$$

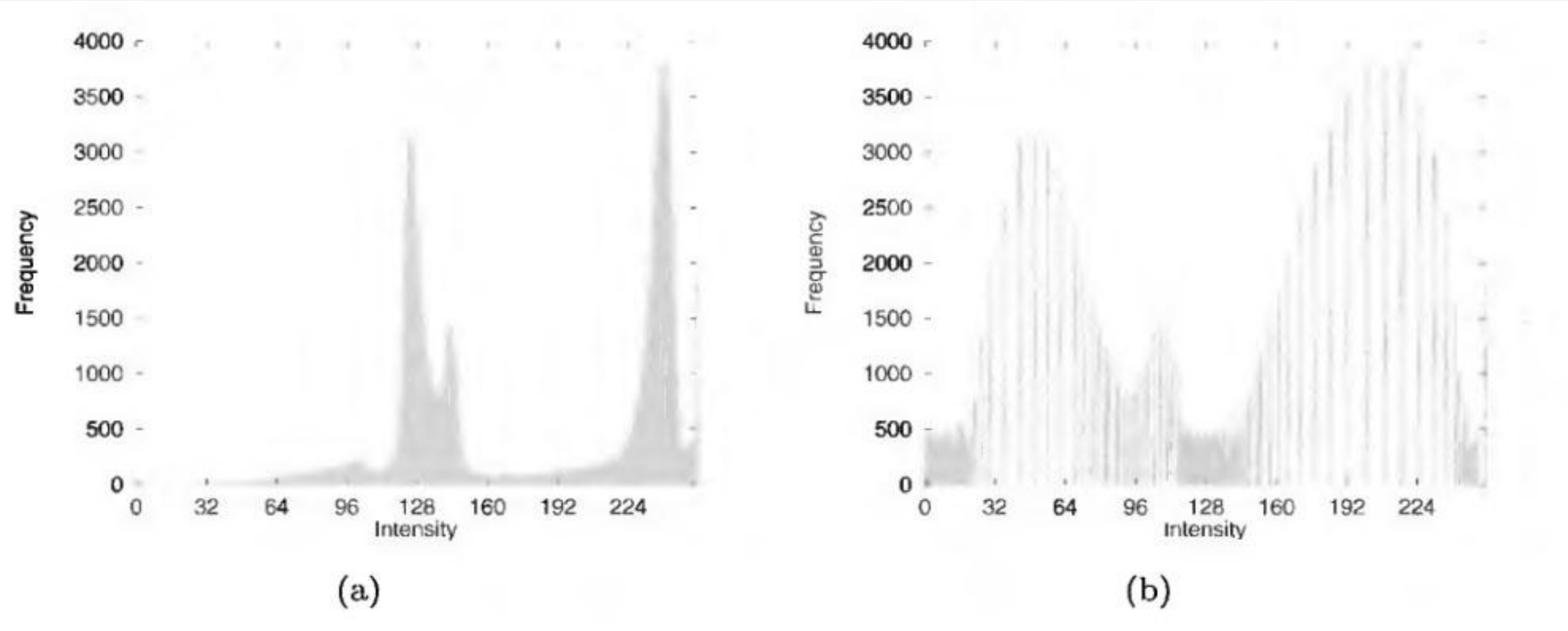
Step4. Constructing **a lookup table T** in range [0;nG-1]

$$T[p] = round((nG-1)/NM)T[p])$$

Step5. Form the **output image g**: $g(x,y) = T[f(x,y)]$

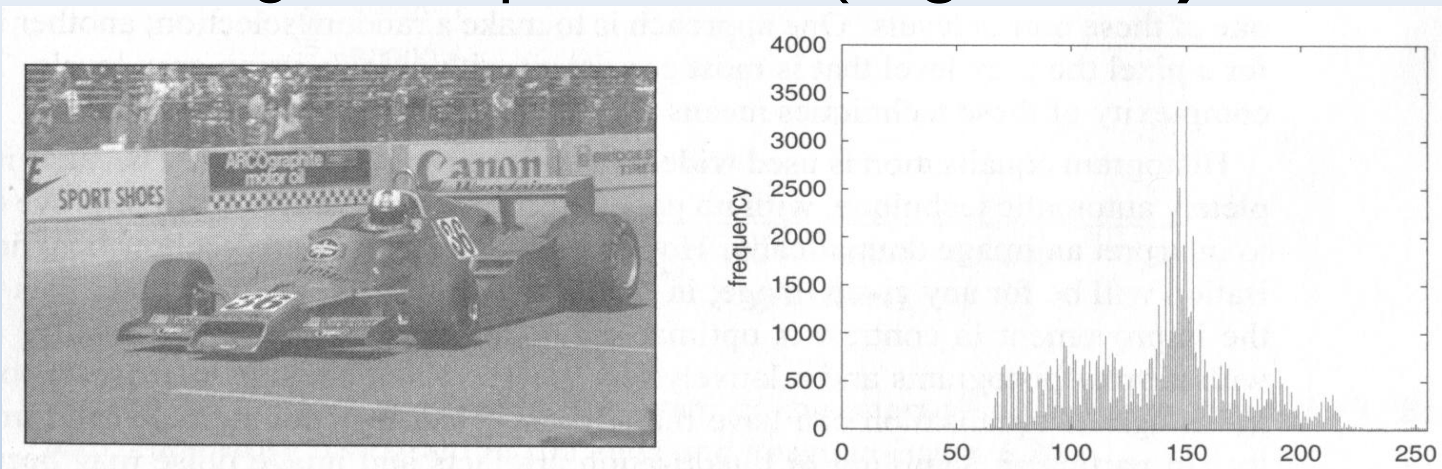
3.1.3. Probability Density Function-based mapping

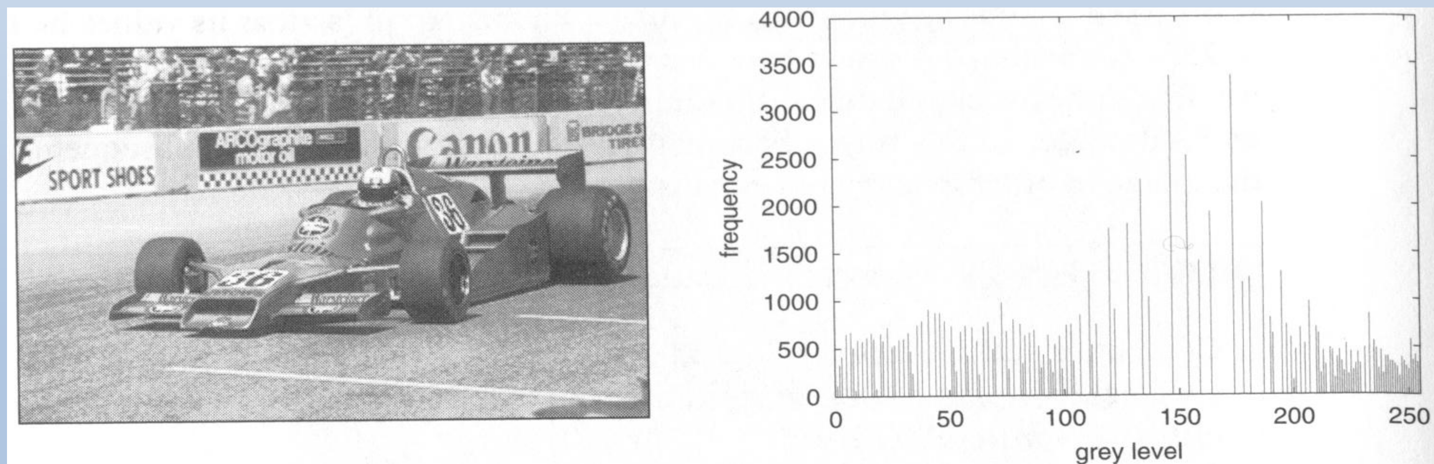
3.1.3.1. Histogram Equalization (**Algorithm)**



3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization (Algorithm)





3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization

Ex: Histogram equalization of f

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

3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization

nG=15, N.M=64

f	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
H	0	4	16	5	2	7	8	4	5	4	4	1	2	1	1
T	0	4	20	25	27	34	42	46	51	55	59	60	62	63	64
TR	0	1	4	5	6	7	9	10	11	12	13	13	14	14	14

$$T[0] = H[0]$$

$$T[p] = T[p-1] + H[p],\ \ p = 1,2,\ldots,nG-1$$

$$T[p] = round((nG-1/NM)T[p])$$

3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization

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

4	4	4	4	4	4	4	7
4	9	9	9	9	7	7	7
1	10	10	9	9	7	7	7
1	10	10	1	1	11	9	6
4	12	11	11	11	11	9	6
4	12	13	13	13	14	14	5
4	12	12	13	13	14	14	5
4	4	4	4	4	5	5	5

$$g(x,y)] = T[f(x,y)]$$

3.1.3. Probability Density Function-based mapping

3.1.3.1. Histogram Equalization

g	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Hg	0	4	0	0	16	5	0	7	0	8	4	4	4	5	4

4	4	4	4	4	4	4	7
4	9	9	9	9	7	7	7
1	10	10	9	9	7	7	7
1	10	10	1	1	11	9	6
4	12	11	11	11	11	9	6
4	12	13	13	13	14	14	5
4	12	12	13	13	14	14	5
4	4	4	4	4	5	5	5

3.1.3. Probability Density Function - based mapping

3.1.3.2. Histogram Specification

Problem statement

Given an image **f(x,y)**, a predefined histogram $\mathbf{p_g(g)}$, it is necessary to define the non-linear transformation **F** such that the output image **g=F(f).**

3.1.3. Probability Density Function - based mapping

3.1.3.2. Histogram Specification

Method

$$s = T(f) = \int_0^f p_f(w)\,dw$$

$$z = G(g) = \int_0^g p_g(w)\,dw$$

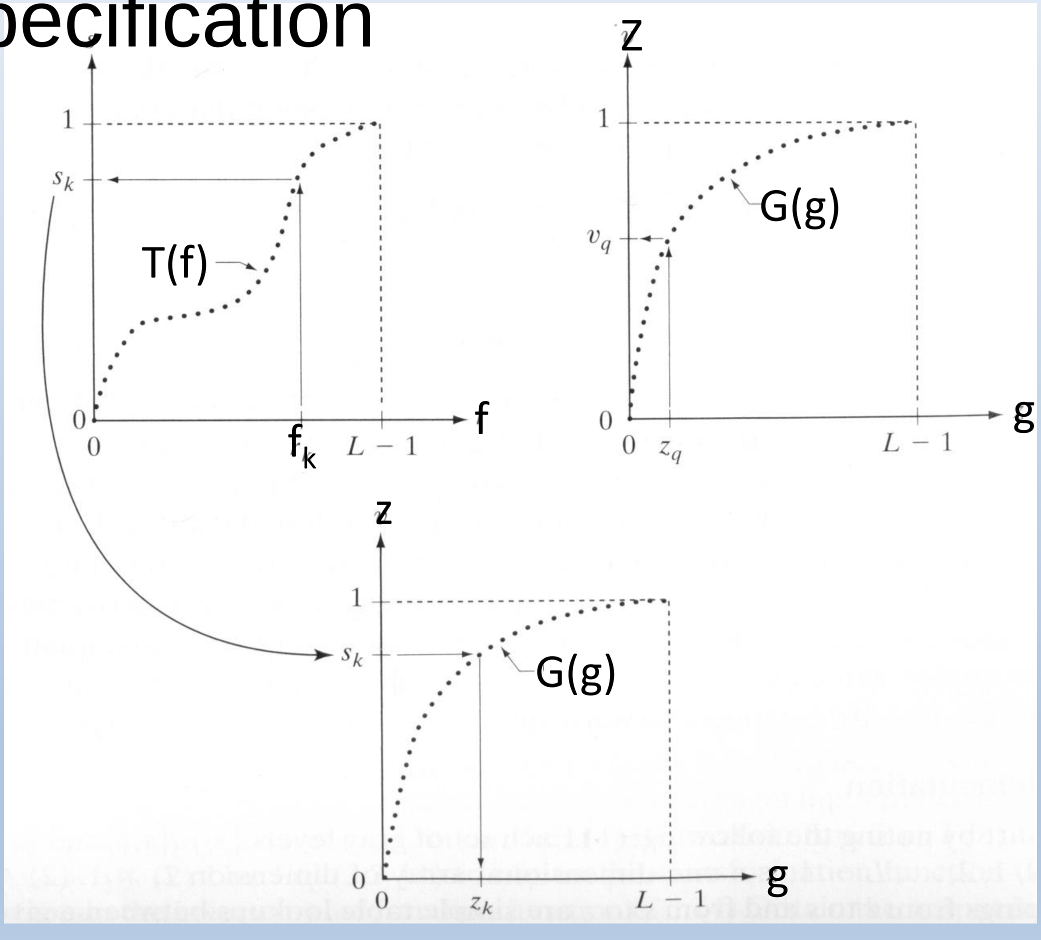
$$g = G^{-1}(z) = G^{-1}(s) = G^{-1}[T(f)]$$

$$g = F(f), F = G^{-1} \circ T$$

3.1.3. Probability Density Function - based mapping

3.1.3.2. Histogram Specification

Method



3.1.3. Probability Density Function-based mapping

3.1.3.2. Histogram Specification (**Algorithm)**

Step1. Create an array H of length nG initialized with 0 values (for an NxM image f of nG grey-levels).

Step2. Form **the image histogram** of f, save to Hf

$$Hf[f(x,y)]+=1$$

Step3. Form **the cumulative image histogram** of f, save to T

$$T[0]=H[0];\ \ T[p]=T[p-1]+Hf[p],\ \ p=1,2,...,nG-1$$

Step4. Constructing **a lookup table T** in range [0;nG-1]

$$T[p]=round((nG-1)/NM)T[p])$$

3.1.3. Probability Density Function-based mapping

3.1.3.2. Histogram Specification (**Algorithm)**

Step5. Form **the cumulative image histogram of g, save to G**

$$G[0] = Hg[0]$$

$$G[p] = G[p-1] + Hg[p], \quad p = 1,2,\ldots,nG-1$$

Step 6. Constructing **a lookup table G** in range [0;nG-1]

$$G[p] = round((nG-1/NM)G[p])$$

Step 7. Form the output image **g**:

$$g(x,y)] = G^{-1}[T[f(x,y)]]$$