Digital Image Processing Using MATLAB

SP_Tutorial2

- TA

崔子昂 cuiza@shanghaitech.edu.cn

李超凡 lichf1@shanghaitech.edu.cn

Outline

- **Color image processing**

  -Digital image and color image

  -Color balance

  -HSI color space

  -Demosaicing
- **Spatial filtering**

  -Spatial noise

  -Linear and nonlinear spatial filter
- **Intensity transformation**

  -Histogram

  -Log transformation and Gamma transformation

  -Histogram processing

Digital image and color image

- What is Digital Image?
- Digital image compose of a finite number of elements – Pixel.
- A visual representation in form of a function $f(x, y)$,

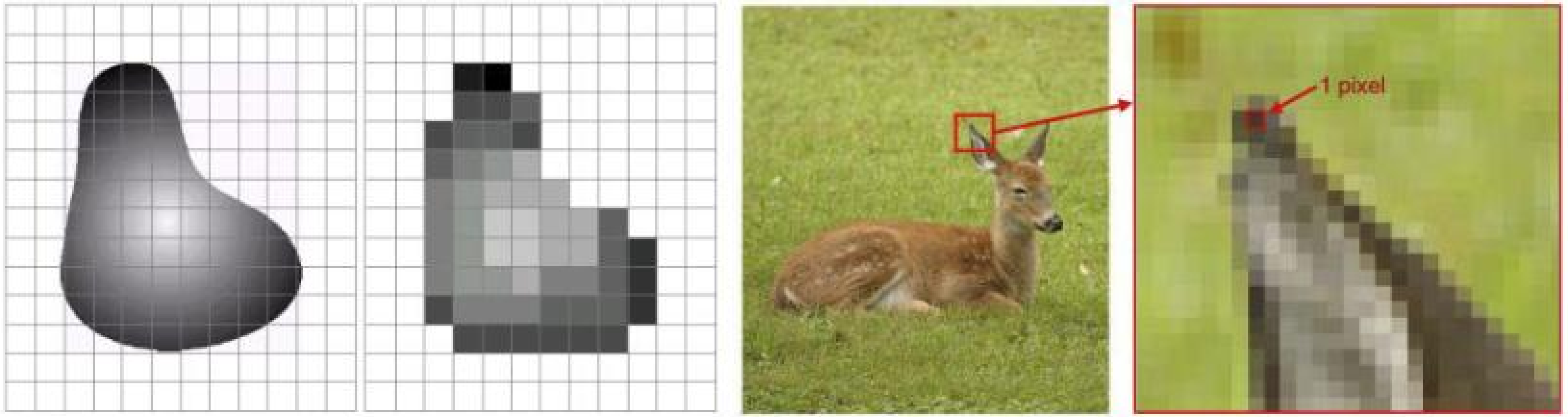
  -f is related to the intensity or brightness at point

  -(x, y) are spatial coordinates

  -x, y, and the amplitude of f are finite and discrete quantities

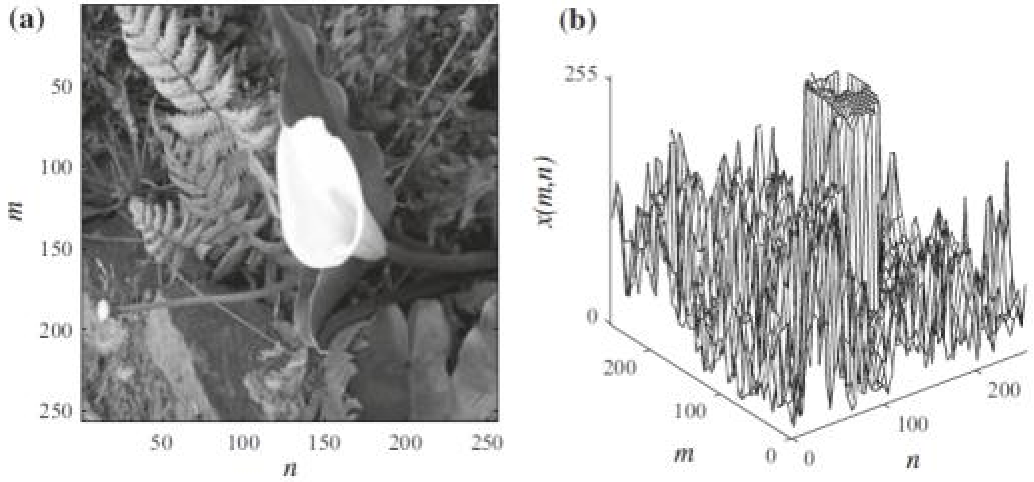
Digital image and color image

- What is Digital Image?



Digital image and color image

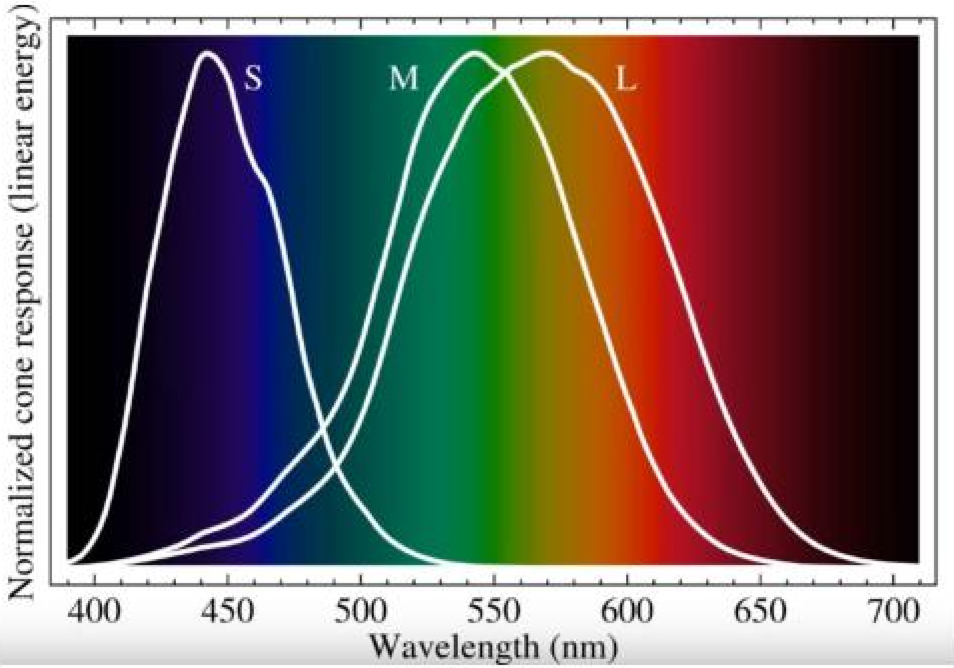
- What is Digital Image?



(a) A 256X256 image with 256 gray levels; (b) its amplitude profile

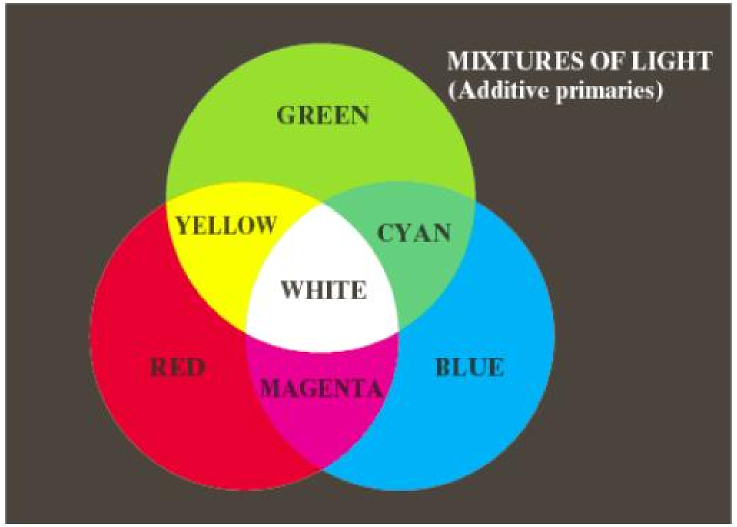
Digital image and color image

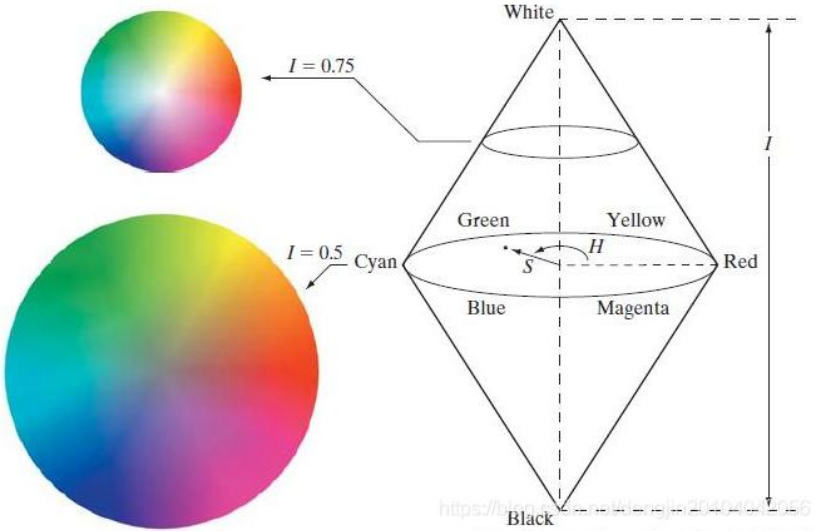
- Color image
- Human visual system color space – the LMS color space
- 3 types of cones sensitive to red, green and blue respectively



Digital image and color image

- Color image





$$\cancel{f(x, y)} \longrightarrow f(x, y, c)$$

Outline

- **Color image processing**

  -Digital image and color image

  -Color balance

  -HSI color space

  -Demosaicing

- **Spatial filtering**

  -Spatial noise

  -Linear and nonlinear spatial filter

- **Intensity transformation**

  -Histogram

  -Log transformation and Gamma transformation

  -Histogram processing

Color balance

- White balance

$$I(x,y) = 0.299 f_R(x,y) + 0.587 f_G(x,y) + 0.114 f_B(x,y)$$

$$k_R = \frac{\bar{I}}{\bar{f_R}} \quad k_G = \frac{\bar{I}}{\bar{f_G}} \quad k_B = \frac{\bar{I}}{\bar{f_B}}$$

$$\begin{bmatrix} g_R(x,y) \\ g_G(x,y) \\ g_B(x,y) \end{bmatrix} = \begin{bmatrix} k_R & & \\ & k_G & \\ & & k_B \end{bmatrix} \begin{bmatrix} f_R(x,y) \\ f_G(x,y) \\ f_B(x,y) \end{bmatrix}$$

- Calculate $I(x,y)$;
- Find means of I, f_R, f_G & f_B;
- Calculate coefficient k_R, k_G & k_B;
- $g(x,y) = k * f(x,y)$

Outline

- **Color image processing**

  -Digital image and color image

  -Color balance

  -HSI color space

  -Demosaicing

- **Spatial filtering**

  -Spatial noise

  -Linear and nonlinear spatial filter

- **Intensity transformation**

  -Histogram

  -Log transformation and Gamma transformation

  -Histogram processing

HSI color space

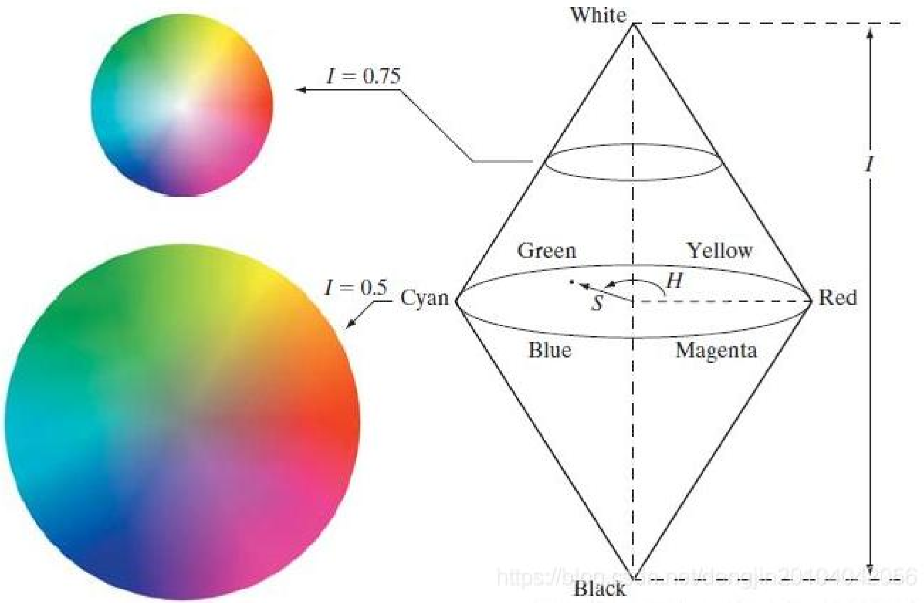
- RGB color space to HSI color space

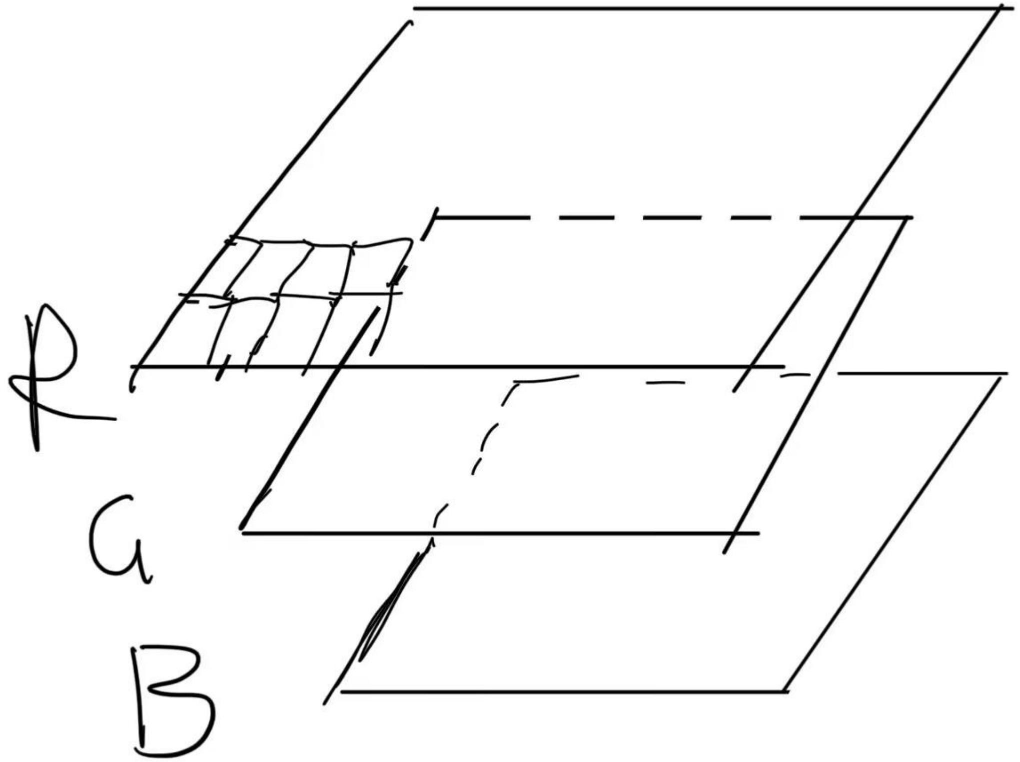
$$I(x,y) = \frac{1}{3}(R(x,y) + G(x,y) + B(x,y))$$

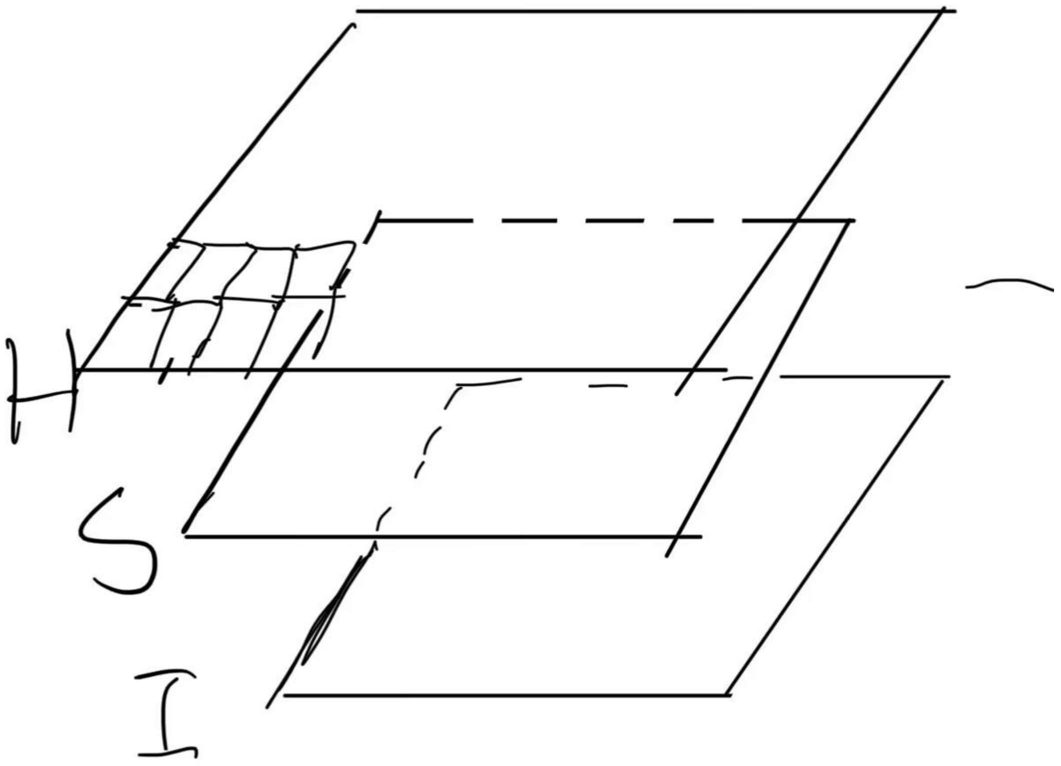
$$S(x,y) = 1 - \frac{3min(R(x,y), G(x,y), B(x,y))}{R(x,y) + G(x,y) + B(x,y)}$$

$$H(x,y) = \begin{cases} \theta. G(x,y) \geq B(x,y) \\ 2\pi - \theta, G(x,y) < B(x,y) \end{cases}$$

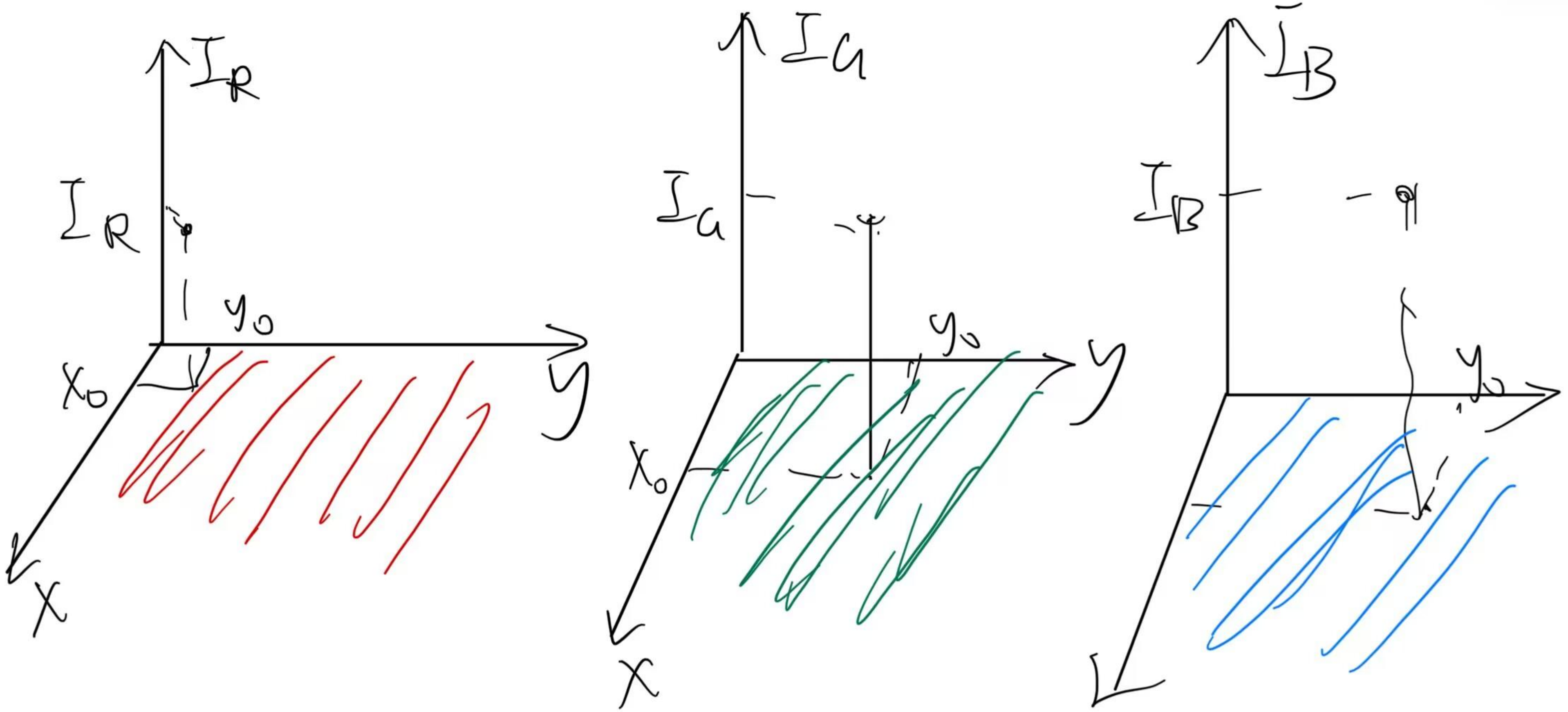
$$\theta = cos^{-1}[\frac{\frac{1}{2}[(R(x,y) - G(x,y)) + (R(x,y) - B(x,y))]}{\sqrt{(R(x,y) - G(x,y))^2 + (R(x,y) - B(x,y))(G(x,y) - B(x,y))}}]$$







Saturation


I_R
I_R
y_0
X_0
y
X
I_G
I_G
y_0
y
X_0
X
I_B
I_B
y_0

HSI color space

- HSI color space to RGB color space

$$
\begin{aligned}
&1^\circ if\ H(x,y) \in [0^\circ, 120^\circ) : \\
&H(x,y) = H(x,y) \\
&\begin{cases}
B(x,y) = I(x,y)(1 - S(x,y)) \\
R(x,y) = I(x,y)[1 + \frac{S(x,y)\cos H(x,y)}{\cos(60^\circ - H(x,y))}] \\
G(x,y) = 3I - (R(x,y) + B(x,y))
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&2^\circ if\ H(x,y) \in [120^\circ, 240^\circ) : \\
&H(x,y) = H(x,y) - 120^\circ \\
&\begin{cases}
R(x,y) = I(x,y)(1 - S(x,y)) \\
G(x,y) = I(x,y)[1 + \frac{S(x,y)\cos H(x,y)}{\cos(60^\circ - H(x,y))}] \\
B(x,y) = 3I - (R(x,y) + G(x,y))
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&3^\circ if\ H(x,y) \in [240^\circ, 360^\circ) : \\
&H(x,y) = H(x,y) - 240^\circ \\
&\begin{cases}
G(x,y) = I(x,y)(1 - S(x,y)) \\
B(x,y) = I(x,y)[1 + \frac{S(x,y)\cos H(x,y)}{\cos(60^\circ - H(x,y))} \\
R(x,y) = 3I - (G(x,y) + B(x,y))
\end{cases}
\end{aligned}
$$

Outline

- **Color image processing**

  -Digital image and color image

  -Color balance

  -HSI color space

  -Demosaicing
- **Spatial filtering**

  -Spatial noise

  -Linear and nonlinear spatial filter
- **Intensity transformation**

  -Histogram

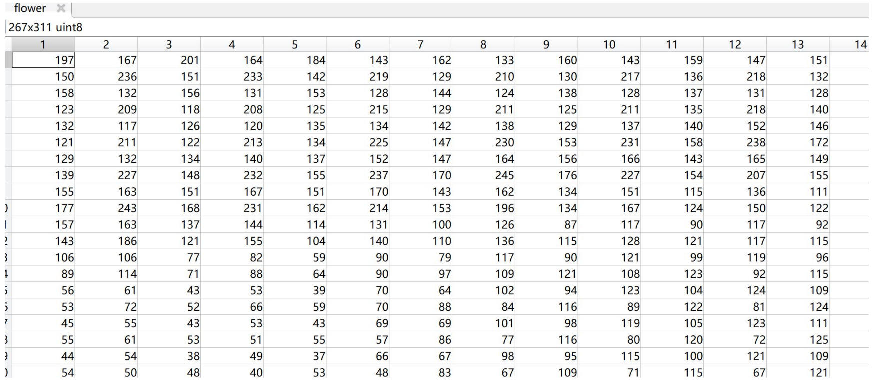
  -Log transformation and Gamma transformation

  -Histogram processing

Demosaicing

- Bayer filter

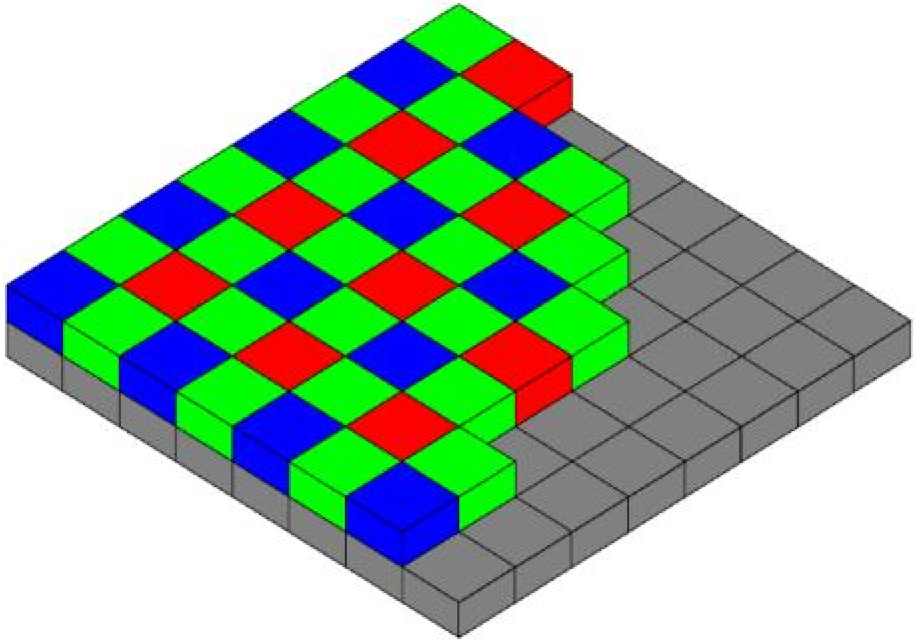
flower.mat



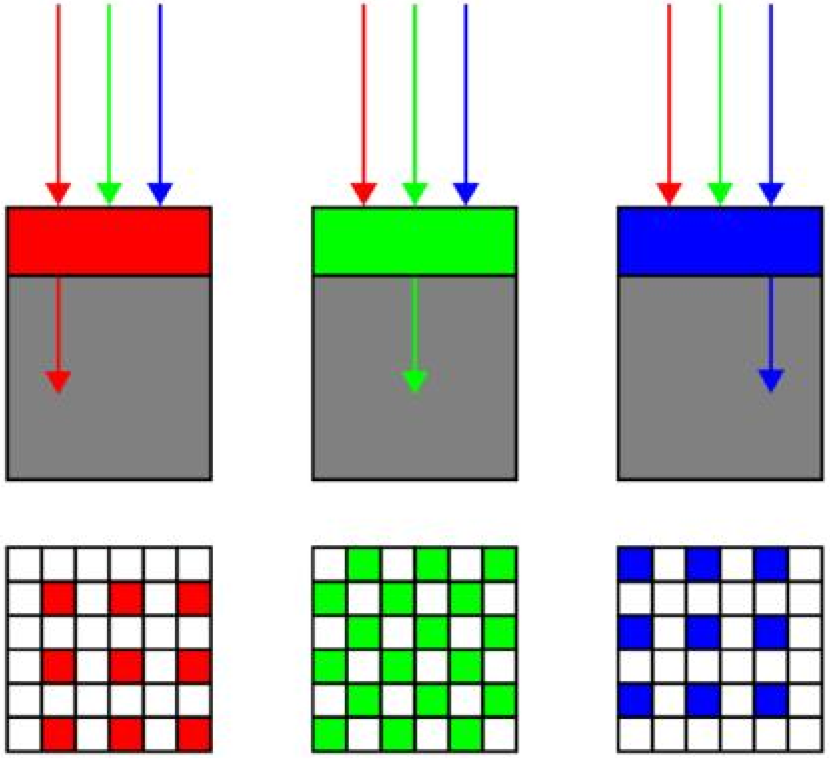
flower

267x311 uint8

1	2	3	4	5	6	7	8	9	10	11	12	13	14
197	167	201	164	184	143	162	133	160	143	159	147	151	
150	236	151	233	142	219	129	210	130	217	136	218	132	
158	132	156	131	153	128	144	124	138	128	137	131	128	
123	209	118	208	125	215	129	211	125	211	135	218	140	
132	117	126	120	135	134	142	138	129	137	140	152	146	
121	211	122	213	134	225	147	230	153	231	158	238	172	
129	132	134	140	137	152	147	164	156	166	143	165	149	
139	227	148	232	155	237	170	245	176	227	154	207	155	
155	163	151	167	151	170	143	162	134	151	115	136	111	
177	243	168	231	162	214	153	196	134	167	124	150	122	
157	163	137	144	114	131	100	126	87	117	90	117	92	
143	186	121	155	104	140	110	136	115	128	121	117	115	
106	106	77	82	59	90	79	117	90	121	99	119	96	
89	114	71	88	64	90	97	109	121	108	123	92	115	
56	61	43	53	39	70	64	102	94	123	104	124	109	
53	72	52	66	59	70	88	84	116	89	122	81	124	
45	55	43	53	43	69	69	101	98	119	105	123	111	
55	61	53	51	55	57	86	77	116	80	120	72	125	
44	54	38	49	37	66	67	98	95	115	100	121	109	
54	50	48	40	53	48	83	67	109	71	115	67	121	

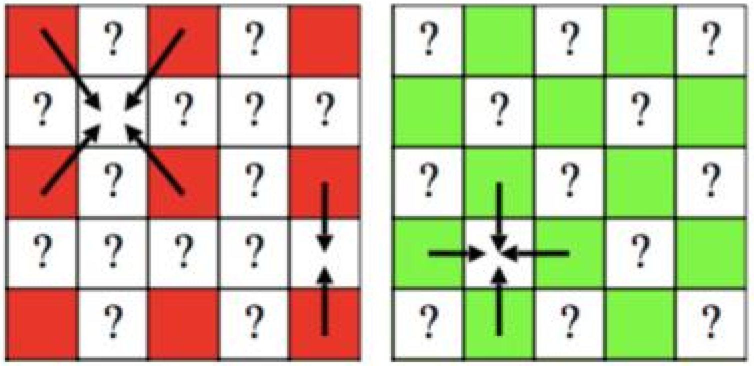


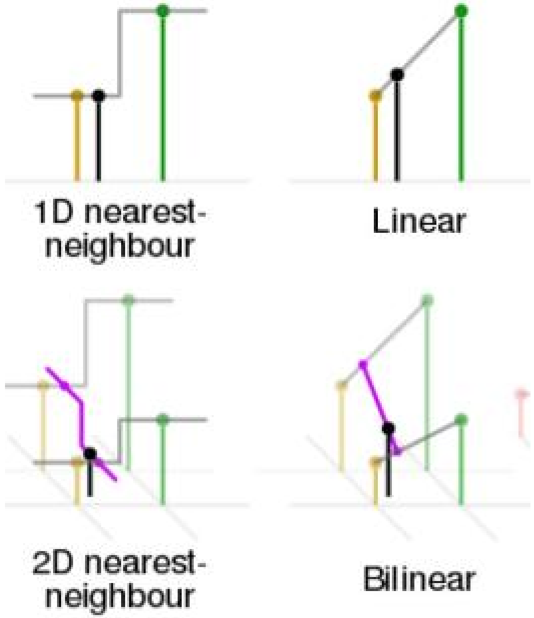
$$f(x,y) \rightarrow f(x,y,c)$$



Demosaicing

- Color image reconstruction





- Nearest-neighbour interpolation (最近相邻插值)
- Bilinear interpolation (双线性插值)

Demosaicing

- Some tips and functions may used in your homework:
- img_R = img(:, :, 1);
- RGB_img = cat(3, img_R, img_G, img_B);
- uint8 & double:

  -uint8 range from 0 to 255;

  -double range from 0 to 1;

  -If you use double(img), the intensity range of the image will not change. It can only change data type.

  -You should use im2double(img) to change the intensity range to [0,1]

Outline

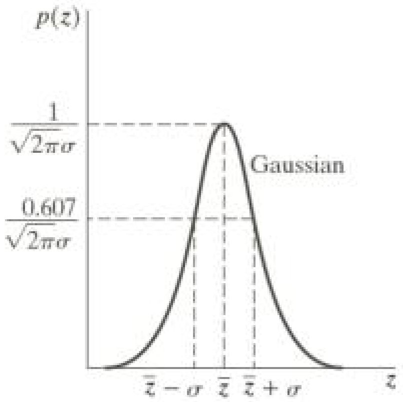
- **Color image processing**
  - -Digital image and color image
  - -Color balance
  - -HSI color space
  - -Demosaicing
- **Spatial filtering**
  - -Spatial noise
  - -Linear and nonlinear spatial filter
- **Intensity transformation**
  - -Histogram
  - -Log transformation and Gamma transformation
  - -Histogram processing

Spatial noise

- Gaussian Noise

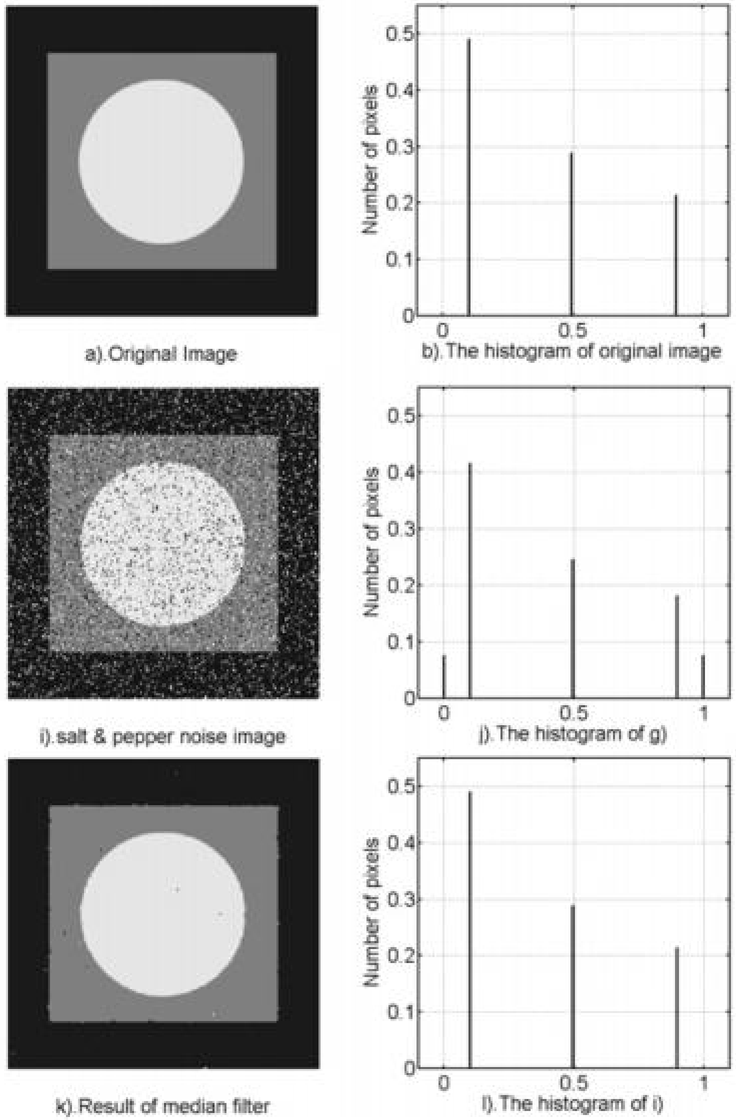
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$mean = \mu\,; variance = \sigma^2$$



- Salt-and-pepper Noise(椒盐噪声)

  Pulse Noise(冲激噪声)



Outline

- **Color image processing**

  -Digital image and color image

  -Color balance

  -HSI color space

  -Demosaicing

- **Spatial filtering**

  -Spatial noise

  -Linear and nonlinear spatial filter

- **Intensity transformation**

  -Histogram

  -Log transformation and Gamma transformation

  -Histogram processing

Spatial filter

- A spatial filter is directly applied on the image.
- A spatial filter is also called spatial masks(掩模), kernels(核), templates(模板) or windows(窗口).
- A spatial filter consists of

a) neighborhood

b) a predefined operation

- A spatial filter can be linear and nonlinear

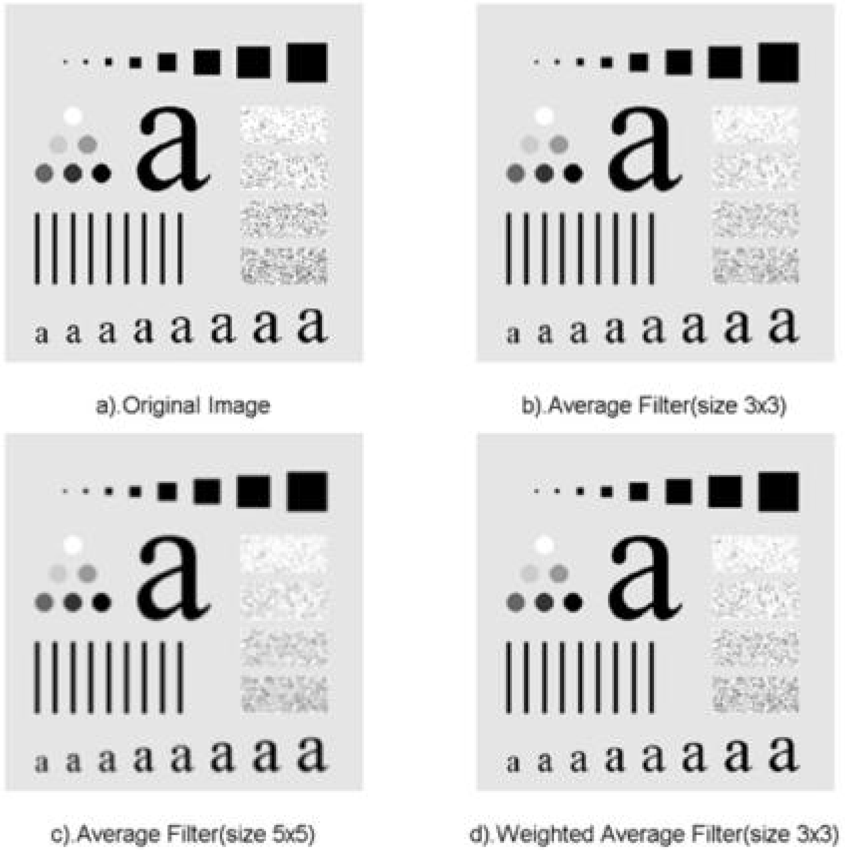
Spatial filter

- Linear Spatial Filter: Average Filter

$$(1/9)*\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix} \qquad (1/16)*\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$$

- Nonlinear Spatial Filter: Median Filter

Order-statistic Filter(统计排序滤波器)



$$g(x,y) = median\{m \times n \; pixel \; neighbouring \; around \; I(x,y)\}$$

Spatial filter

- Try these built-in functions:
- imnoise(input_image, type, para);
- fspecial(type, para);
- imfilter(input_image, spatial_mask, …);
- medfilt2(input_image, [m n]);
- Other functions...

Outline

- **Color image processing**
  - -Digital image and color image
  - -Color balance
  - -HSI color space
  - -Demosaicing
- **Spatial filtering**
  - -Spatial noise
  - -Linear and nonlinear spatial filter
- **Intensity transformation**
  - -Histogram
  - -Log transformation and Gamma transformation
  - -Histogram processing

Histogram

$$h(r_k) = n_k$$

Where r_k : the kth intensity value in the level range of [0, L-1]

n_k : the number of pixels in the image with intensity r_k

Normalized Histogram（归一化直方图）

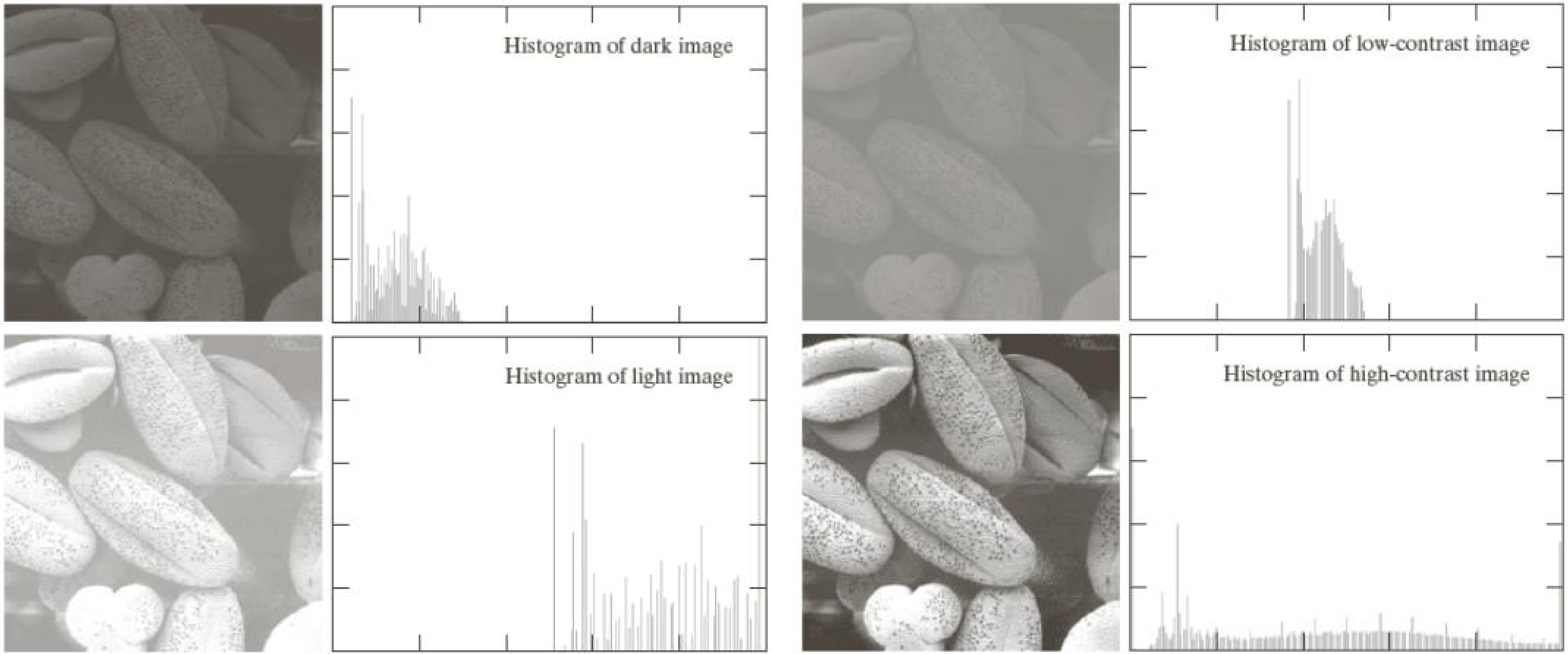
$$p(r_k) = \frac{n_k}{MN}$$

Where $p(r_k)$: the probability of occurrence of intensity r_k in an image

M, N: the row and column dimensions of the image

Histogram

- Histogram examples



Histogram

Histogram properties:

- Describe the number or probability of intensity, no location(spatial) information;
- Can be same as other images;
- $\sum_0^{L-1} n_k = M \cdot N \;\; or \;\; \sum_0^1 p(r_k) = 1$
- If region $C = A \cup B$,A and B are disjoint, $H_c = H_A + H_B$

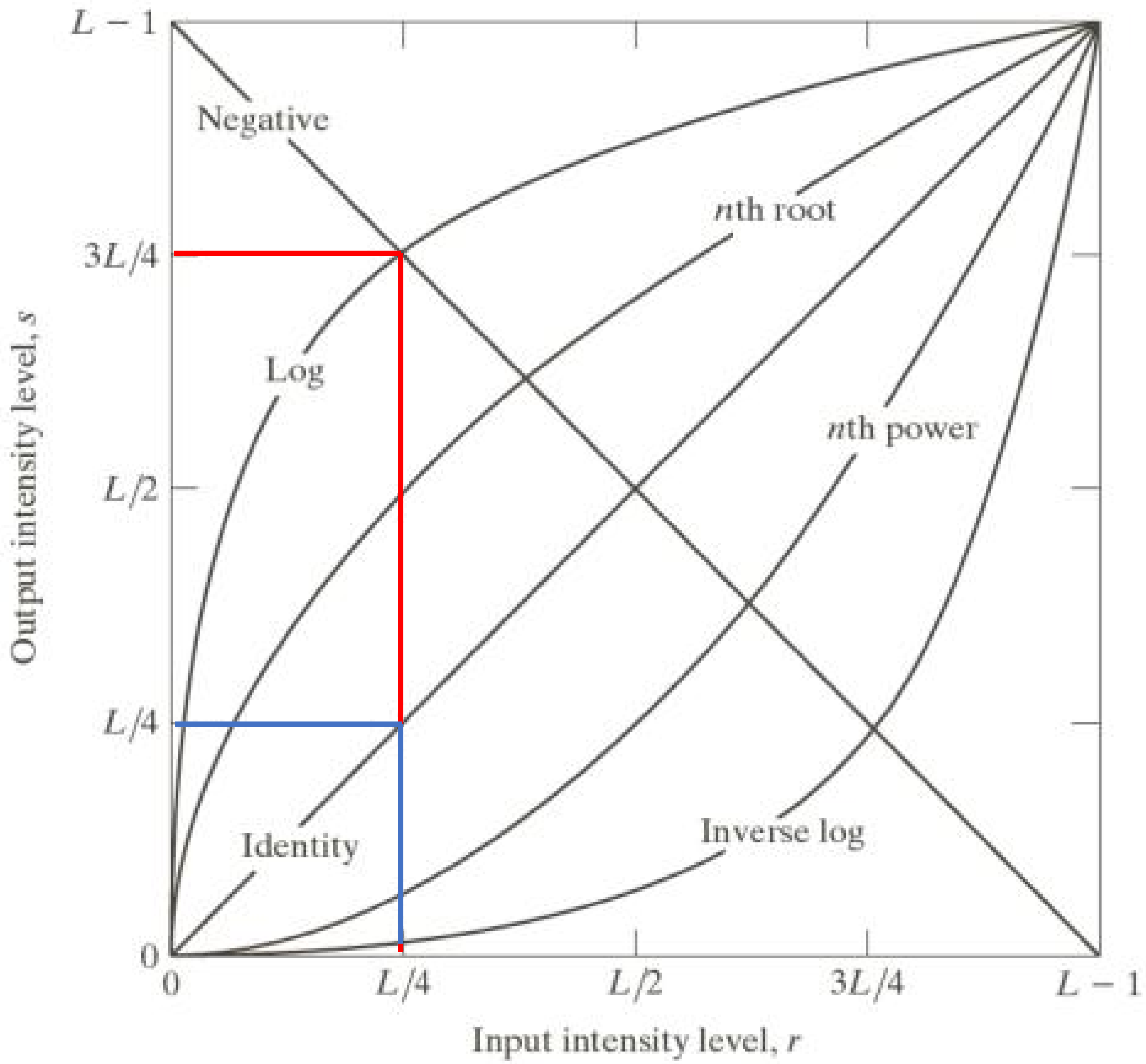
Outline

- **Color image processing**
  - -Digital image and color image
  - -Color balance
  - -HSI color space
  - -Demosaicing
- **Spatial filtering**
  - -Spatial noise
  - -Linear and nonlinear spatial filter
- **Intensity transformation**
  - -Histogram
  - -Log transformation and Gamma transformation
  - -Histogram processing

Log transformation

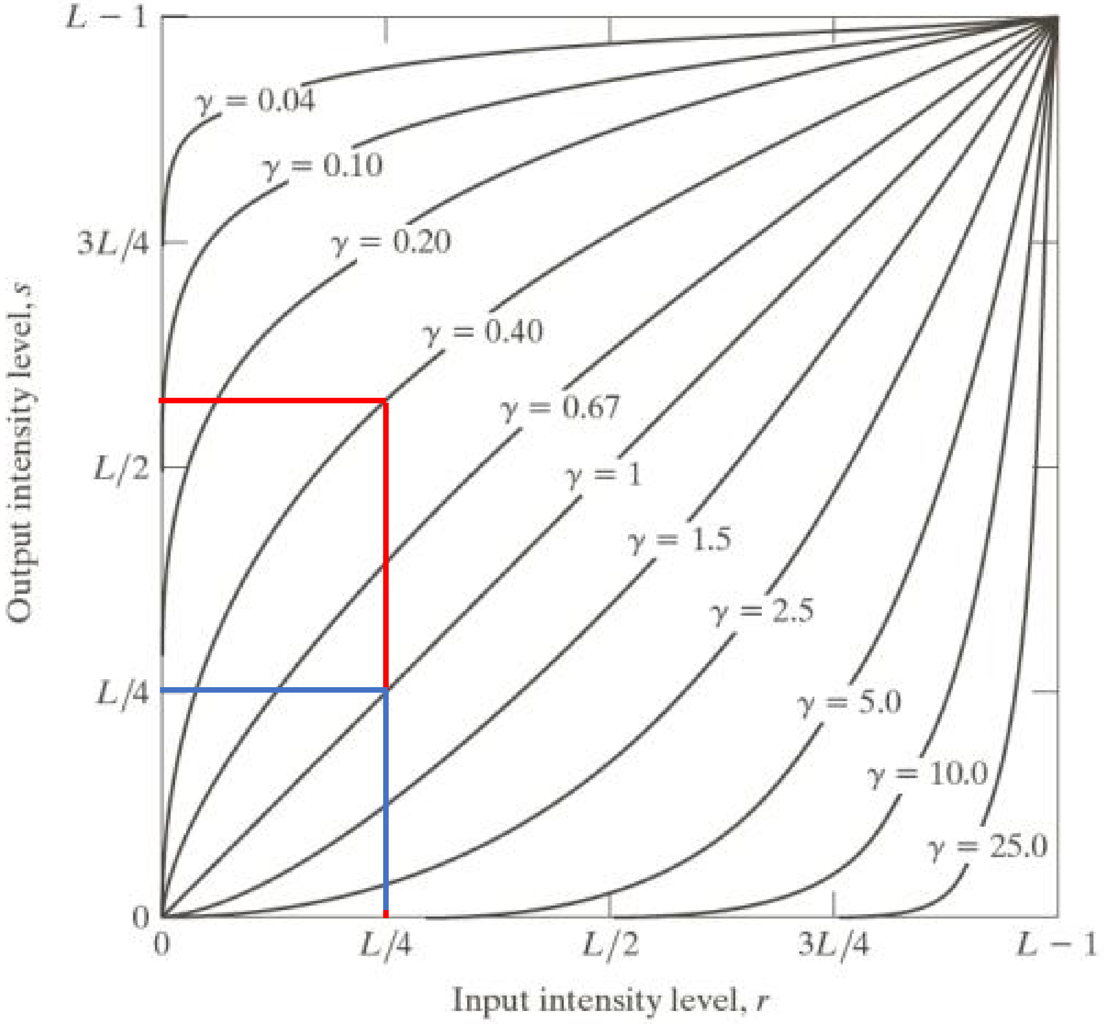
- $s = c\log(1 + r)$
- $s, r \in [0, L-1]$

$$S = (L-1)log_L(r+1)$$



Gamma transformation

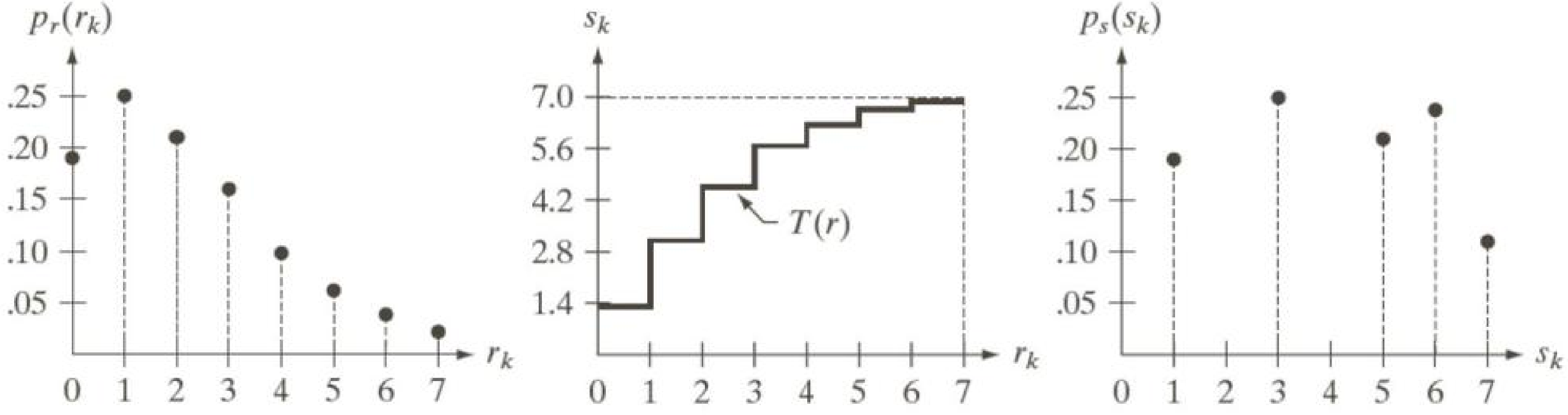
- $s = c \cdot r^{\gamma}$
- $s, r \in [0, L-1]$



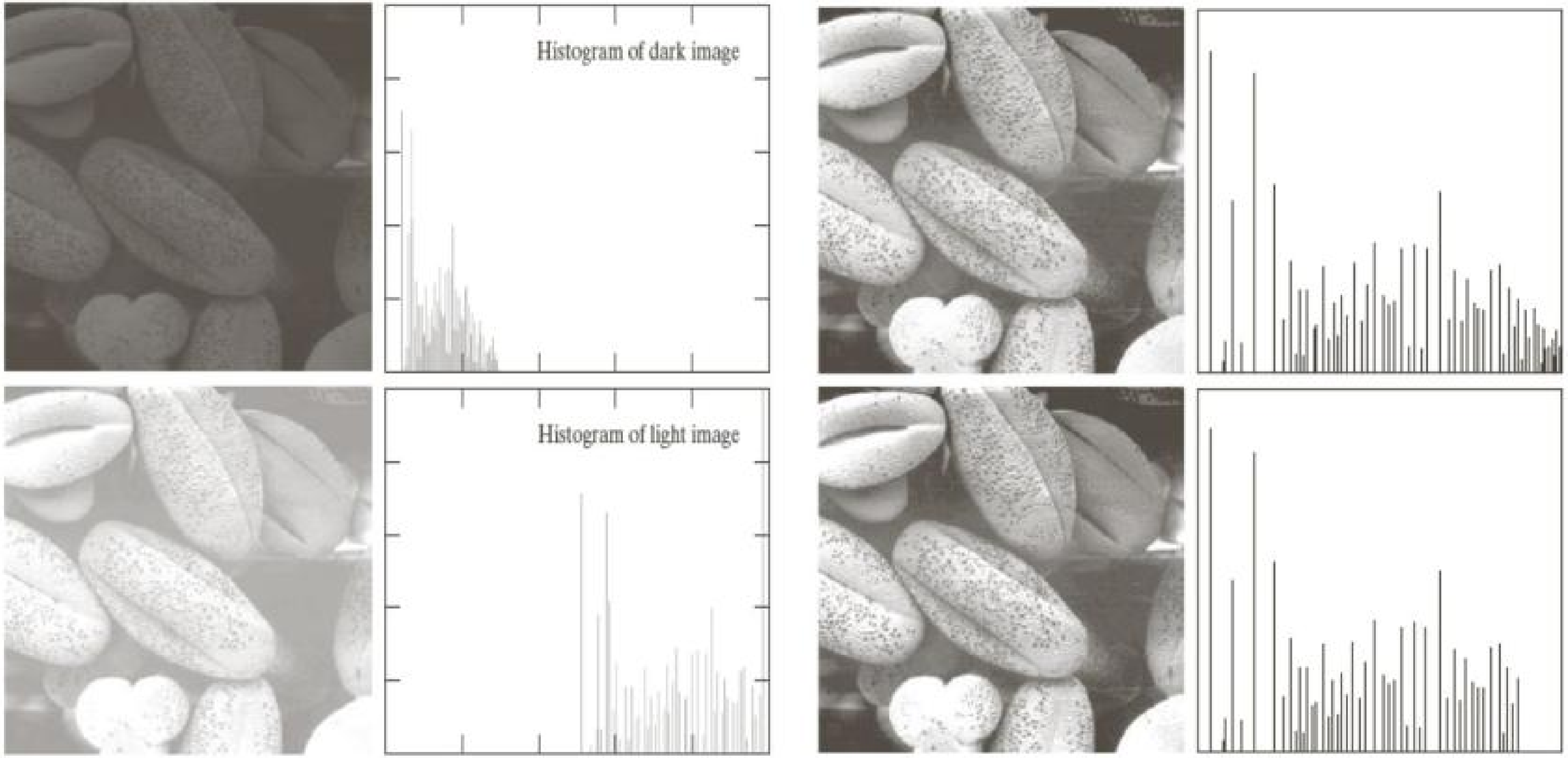
Outline

- **Color image processing**
  - -Digital image and color image
  - -Color balance
  - -HSI color space
  - -Demosaicing
- **Spatial filtering**
  - -Spatial noise
  - -Linear and nonlinear spatial filter
- **Intensity transformation**
  - -Histogram
  - -Log transformation and Gamma transformation
  - -Histogram processing

Histogram equalization



Histogram equalization

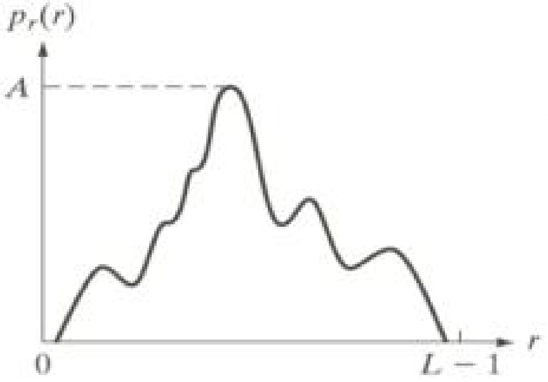


Histogram equalization

- Uniform Probability density function : $p_s(s) = \frac{1}{L-1}$
- The probability density function (PDF) of s is

$$p_s(s) = p_r(r) \cdot \frac{dr}{ds} \Rightarrow \; p_r(r) \cdot \frac{dr}{ds} = \frac{1}{L-1} \; \Rightarrow (L-1)p_r(r) \cdot dr \; = \; ds$$

- Transformation function : $s = T(r) = (L-1)\int_0^r p_r(w)dw$



Histogram equalization

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \ \mathrm{k}$$
$$= 1,2,\ldots,\mathrm{L}-1$$

r_k	n_k	$p_r(r_k)$	s_k		s_k	$p_s(s_k)$
0	790	0.19	1.33	1	0	0
1	1023	0.25	3.08	3	1	0.19
2	850	0.21	4.55	5	2	0
3	656	0.16	5.67	6	3	0.25
4	329	0.08	6.23	6	4	0
5	245	0.06	6.65	7	5	0.21
6	122	0.03	6.86	7	6	0.24
7	81	0.02	7.00	7	7	0.11

Histogram matching

- Generate a processed image with a specified histogram:

  For input: $s = T(r) = (L-1)\int_0^r p_r(w)dw$

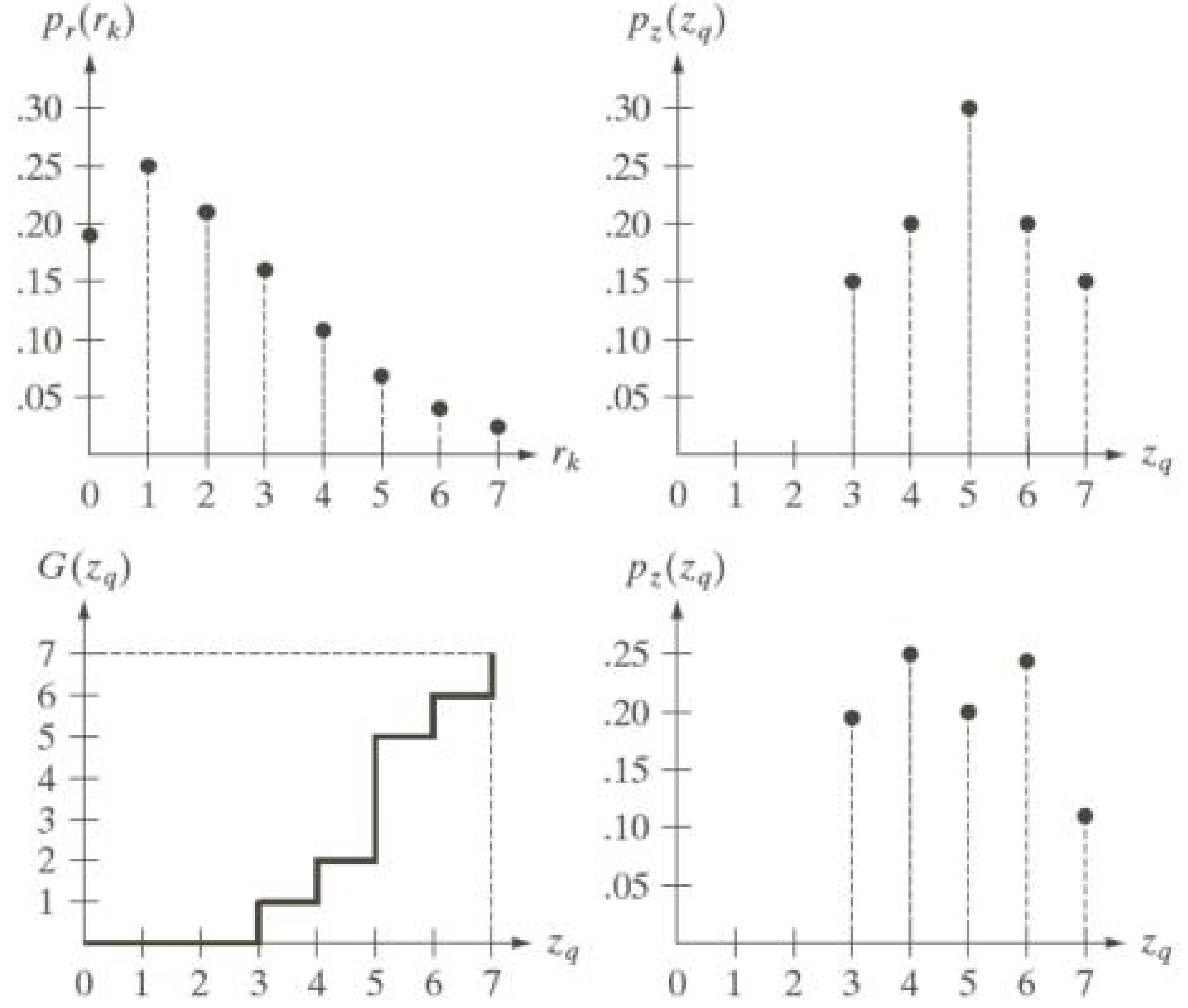
  For output: $G(z) = (L-1)\int_0^z p_z(t)dt$

  Therefore, $z = G^{-1}(s) = G^{-1}[T(r)]$

Histogram matching

r_k	$p(r_k)$	$s_k=T(r_k)$	z_q	$p(z_q)$	$s_k = G(z_q)$	$s_k \to z_k$	$r_k \to z_k$	z_k	$p(z_k)$
0	0.19	1	0	0	0	$0 \to 0, 1, 2$	$0 \to 3$	0	0
1	0.25	3	1	0	0	$1 \to 3$	$1 \to 4$	1	0
2	0.21	5	2	0	0	$2 \to 4$	$2 \to 5$	2	0
3	0.16	6	3	0.15	1		$3 \to 6$	3	0.19
4	0.08	6	4	0.20	2		$4 \to 6$	4	0.25
5	0.06	7	5	0.30	5	$5 \to 5$	$5 \to 7$	5	0.21
6	0.03	7	6	0.20	6	$6 \to 6$	$6 \to 7$	6	0.24
7	0.02	7	7	0.15	7	$7 \to 7$	$7 \to 7$	7	0.11

Histogram matching



Histogram matching