Ch2
灰度变换与空间滤波

荆丽雯 助理教授
大数据与互联网学院

Part 1：空间域和空间域处理

- **空间域**：图像平面本身

- **空间域处理:**

$$\boldsymbol{g(x,y)=T[f(x,y)]}$$

其中$f(x,y)$为输入图像，$g(x,y)$为输出（处理后的）图像，T是在点(x,y)的一个指定邻域上定义的对图像f进行处理的算子。

此外，T还可以对一组图像进行处理，如为了降低噪声而叠加K幅图像。

Part 2：灰度变换

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

- 变换T最简单的形式是邻域大小为1×1（一个单独的像素）的情形。
- 对单色图像来说，
  - 输入：图像f在(x,y)处的灰度
  - 输出：经过T变换的灰度
- 以上过程成为**灰度变换**，可简写成:

$$s=T(r)$$

式中$\mathrm{r}=f(x,y)$，是图像f在(x,y)处的灰度。

Part 2：
灰度变换

$$s = T(r)$$

- 相关工具箱函数：

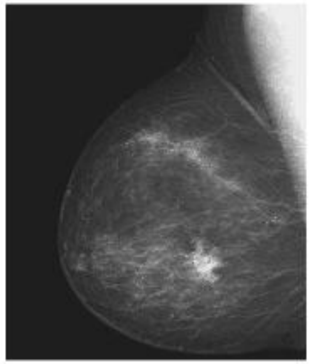
  Imadjust和stretchlim

Part 2：灰度变换

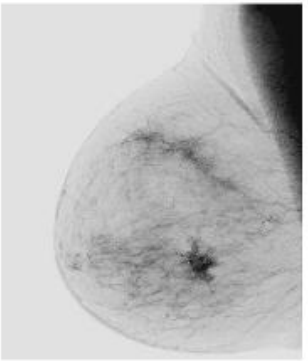
Imadjust

DIP2_2_1.m

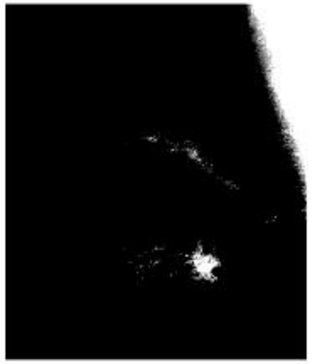
- J = imadjust(I,[low_in high_in],[low_out high_out],gamma) maps intensity values in I to new values in J, where gamma specifies the shape of the curve describing the relationship between the values in I and J.



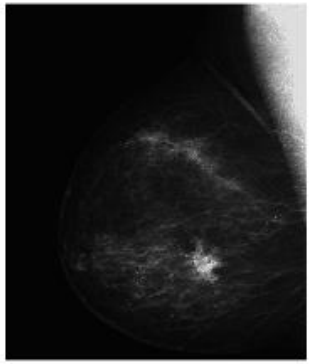
原片



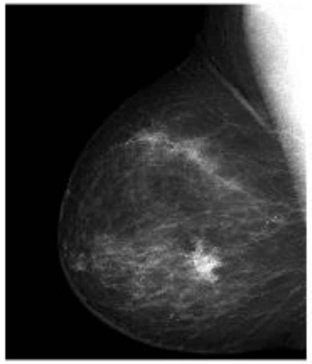
反片



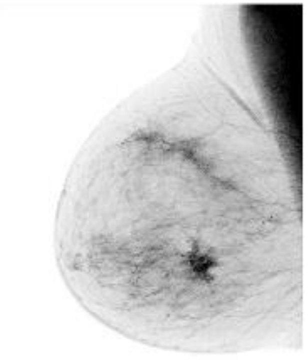
[0.5-0.75]扩展到[0 1]



gamma = 2

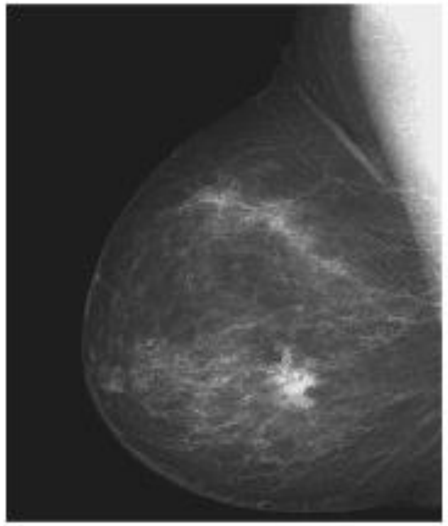
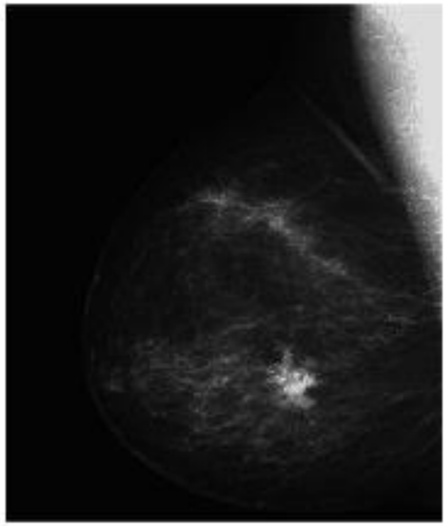


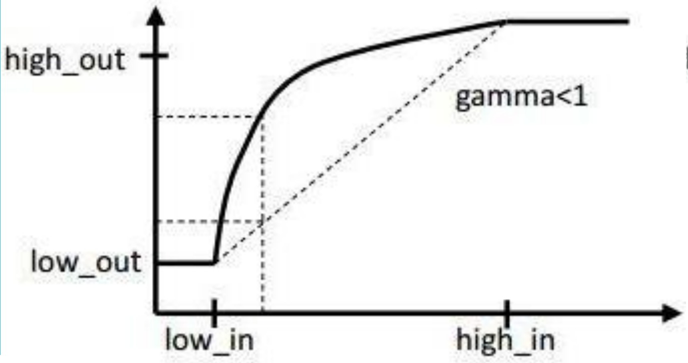
stretchlim

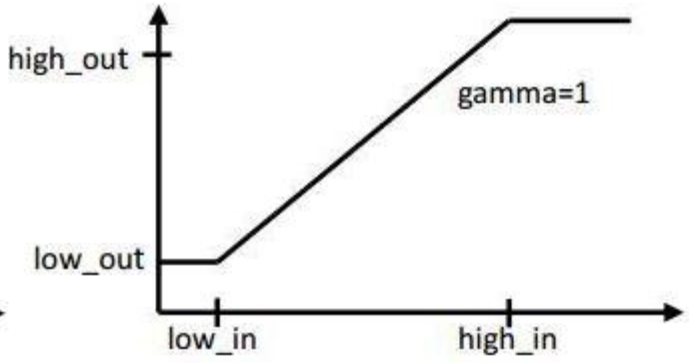


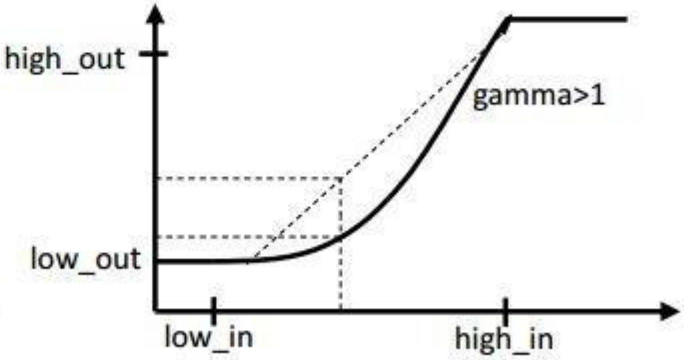
Stretchlim反片

Part 2：
灰度变换
Imadjust



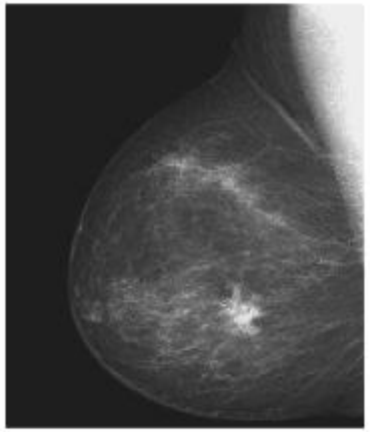


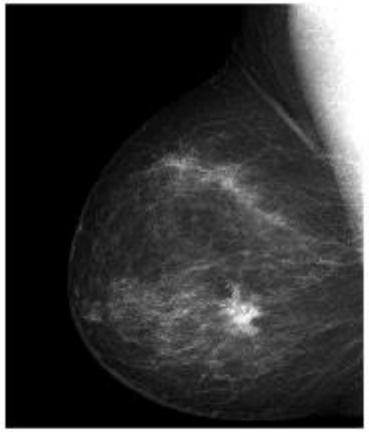


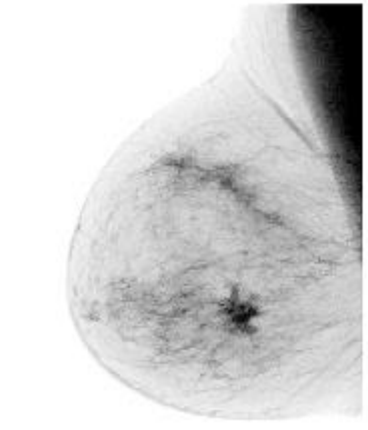
DIP2_2_1.m

Part 2：灰度变换
Imadjust & stretchlim

- lowhigh = stretchlim(I) computes the lower and upper limits that can be used for contrast stretching grayscale or RGB image I. The limits are returned in lowhigh. By default, the limits specify the bottom 1% and the top 1% of all pixel values.


原片


stretchlim


Stretchlim反片

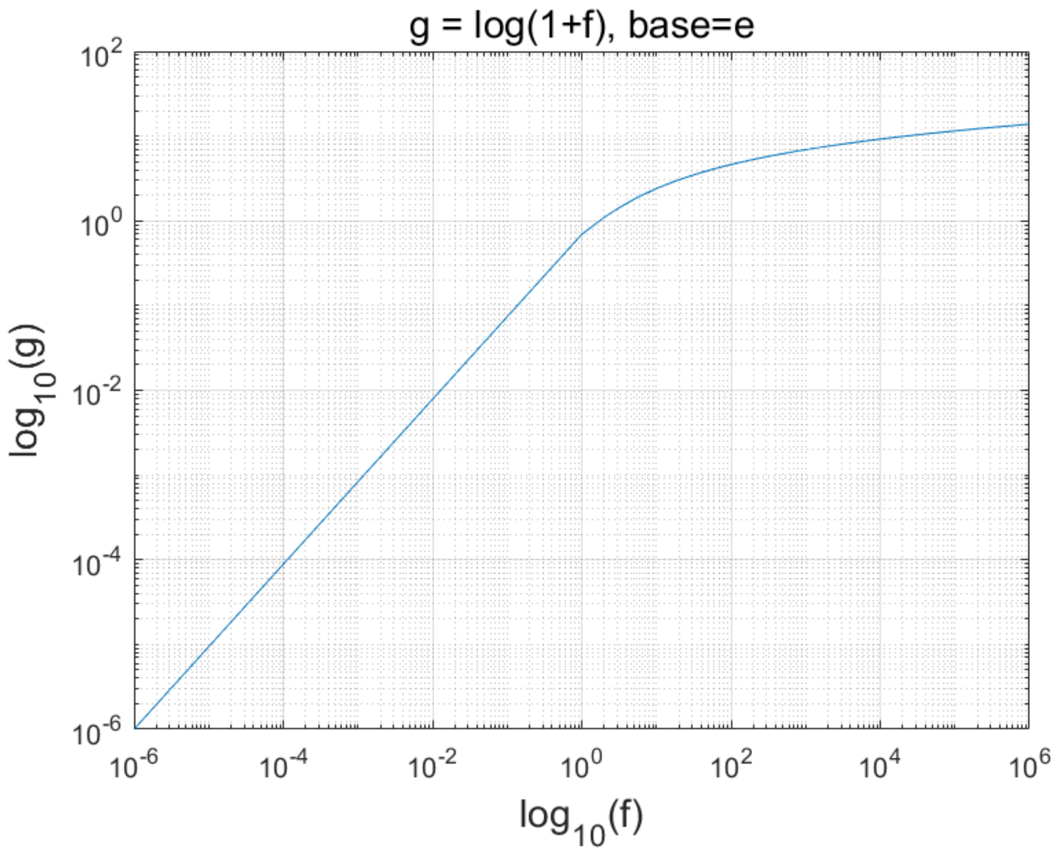
DIP2_2_1.m

Part 2：
灰度变换
对数变换

- 对数变换公式

  $g = c*\log(1+f)$

- 压缩动态范围：
  - From无穷小~10^6
  - To无穷小~13.8155

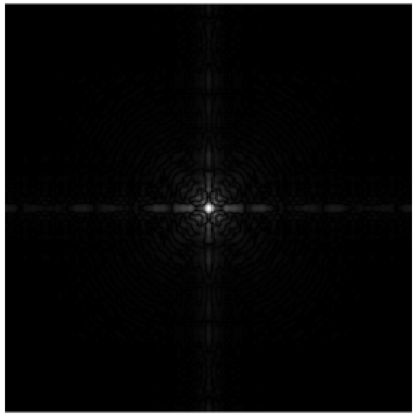


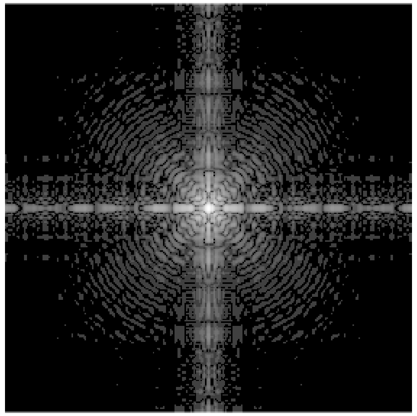
DIP2_2_2.m

Part 2：
灰度变换
对数变换

DIP2_2_2.m

- Matlab函数：
- im2unit8:
  - J = im2uint8(I) converts the grayscale, RGB, or binary image I to uint8, rescaling or offsetting the data as necessary.
  - If I is of class single or double with values outside the range [0, 1] then you can use rescale function to rescale values to the expected range.
- mat2gray:
  - I = mat2gray(A,[amin amax]) converts the matrix A to an intensity image I that contains values in the range 0 (black) to 1 (white).amin and amax are the values in A that correspond to 0 and 1 in I. Values less than amin become 0, and values greater than amaxbecome 1.
  - I = mat2gray(A) sets the values of amin and amax to the minimum and maximum values in A.



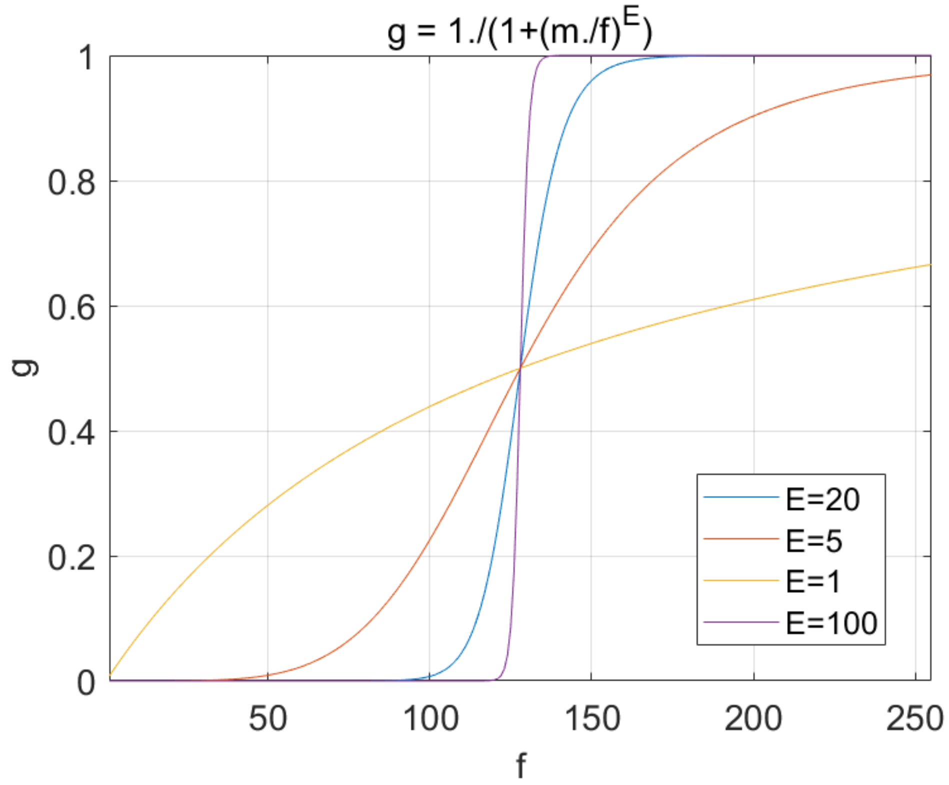


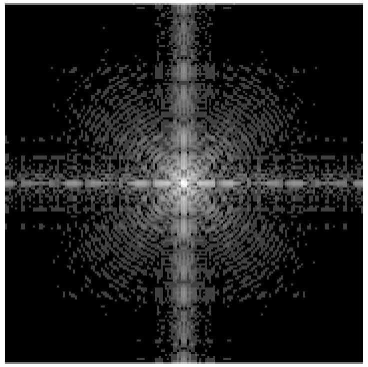
Part 2：
灰度变换
对比度拉伸变换

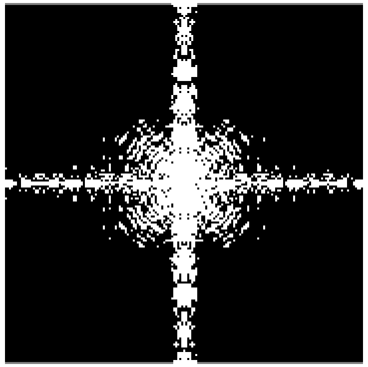
- 公式

  g = 1./(1+(m./f).^E)

- **扩展动态范围：**把窄范围的输入灰度级扩展为宽范围的输出灰度级







m=128, E=10

DIP2_2_2.m

Part 2：灰度变换
指定任意灰度变换

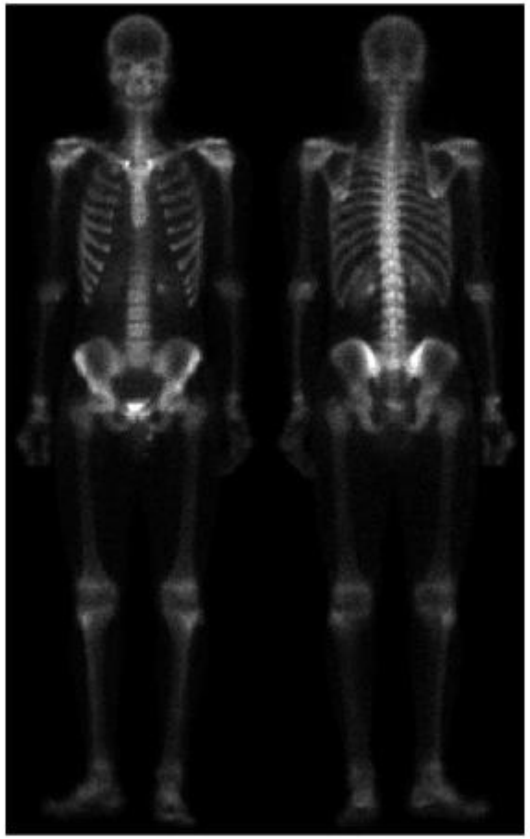
- 可以用任何指定的变换函数来变换一副图像的灰度。
- 步骤：
  - （1）构建列向量T(1:1:256),对应0-255所有灰度值的映射结果
  - （2）将输入图像f转换为映射后的输出图像g
- 函数：

g = interp1(z, T, f) % 内插

Z = linspace(0, 1, numel(T)) %生成n个[0,1]间等间隔的点

Part 2：
灰度变换
intrans函数
（其中用到tofloat函数）

- Intrans能执行如下变换功能：负片变换、对数变换、伽马变换、对比度拉伸变换和自定义函数变换



DIP2_2_4.m

Part 2：
灰度变换

课堂TASK:

小组学习intrans和□tofloat函数（10min）

Page 29 & Page 369

Part 3: 直方图处理与函数绘图

- 定义：一幅数字图像在区间[o,G]内共有L个灰度级，其直方图和归一化直方图分别为：

$$h(r_k) = n_k$$

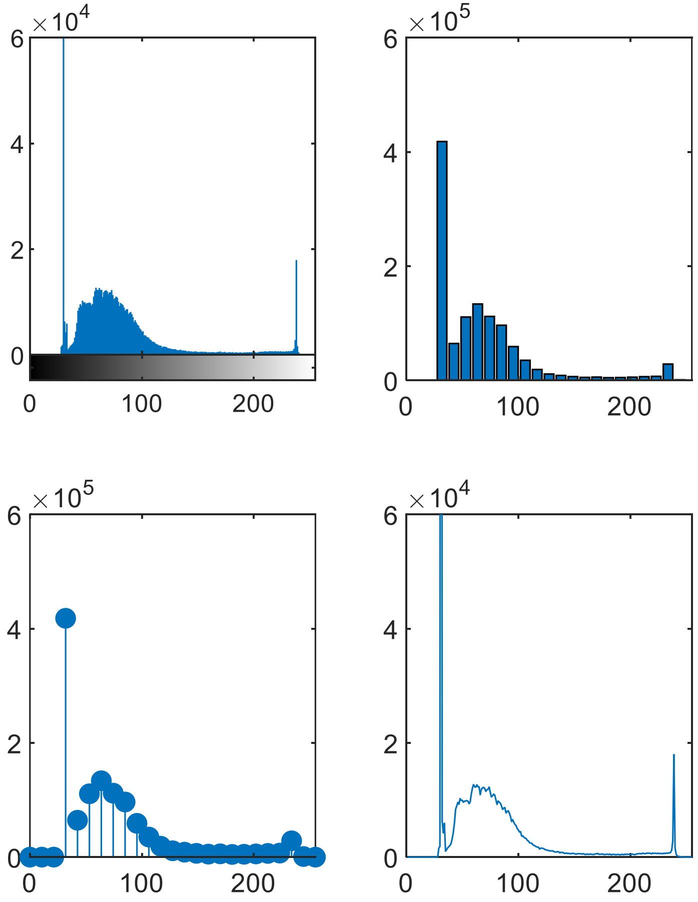
$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

其中r_k是区间[o,G]内的第k级灰度，n_k为图像中出现r_k这种灰度级的像素数，n为总像素数。

Matlab 函数：imhist以及numel

Part 3: 直方图处理与函数绘图

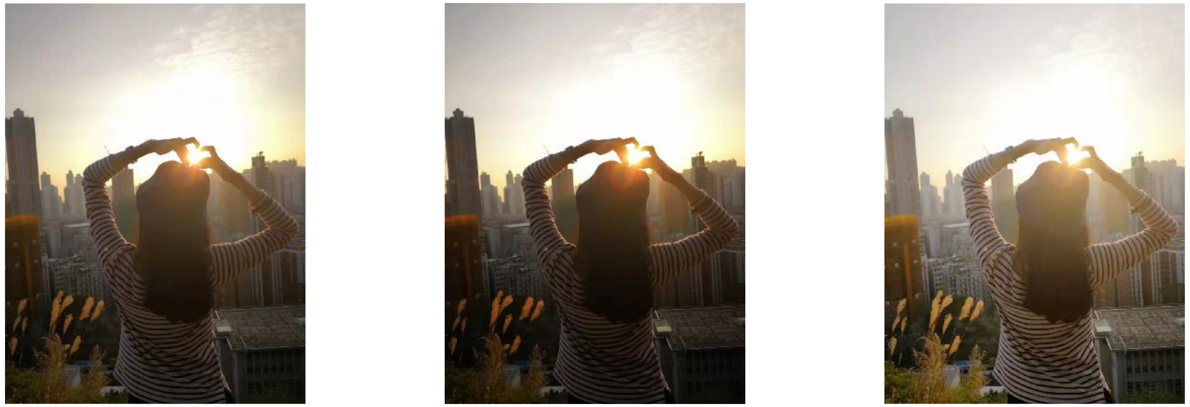
$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

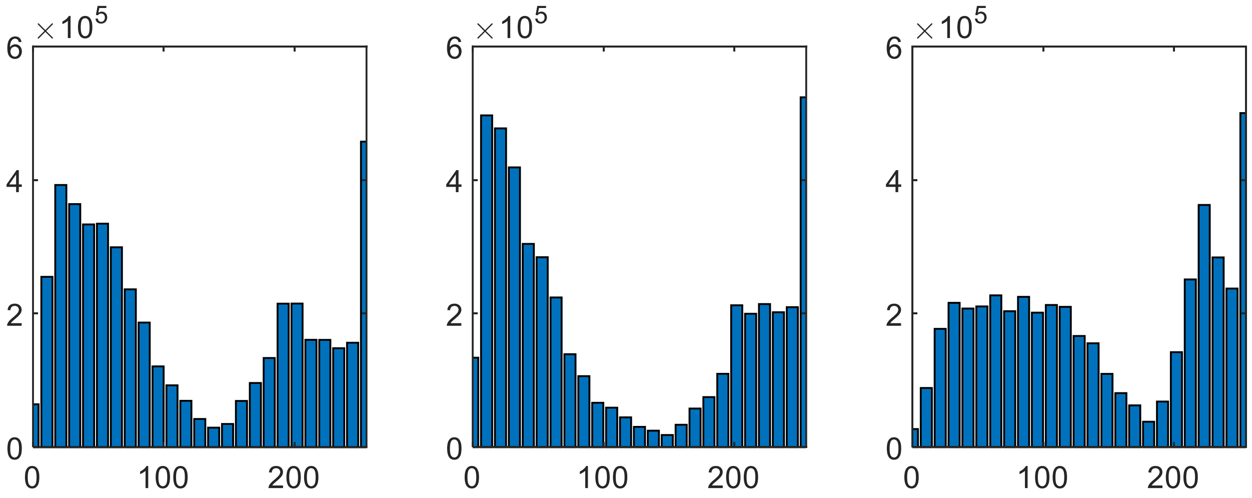


Matlab 函数：imhist，bar，stem，plot

DIP2_3_1.m

Part 3: 直方图处理与函数绘图





Matlab 函数：imhist，bar，stem，plot

DIP2_3_2.m

Part 3: 直方图处理与函数绘图
（直方图均衡化）

- 直方图均衡化(Histogram Equalization)是一种增强图像对比度(Image Contrast)的方法，其主要思想是将一副图像的直方图分布通过累积分布函数变成近似均匀分布，从而增强图像的对比度。变换函数如下：

（1）连续：

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

（2）离散：

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n}$$

Part 3:
直方图处理与函数绘图
（直方图均衡化-连续）

（1）连续：

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

其中， r表示待处理图像的灰度，值域[0,L-1]，常见unit8中[0,255]；L表示离散灰度级数，通常取2的整数次幂；$p_r(r)$表示待处理图像中灰度值r的PDF(概率密度函数)。

特性：（a）$T(r)$在区间$0 \le r \le L-1$上是一个单调递增函数

（b）对于$0 \le r \le L-1$，有$0 \le T(r) \le L-1$

（c）$p_s(s)$始终是均匀的，与$p_r(r)$形式无关

Part 3: 直方图处理与函数绘图

（直方图均衡化-连续）

$$s = T(r) \tag{1}$$

$$p_s(s) = p_r(r)\left|\frac{\mathrm{d}r}{\mathrm{d}s}\right| \tag{2}$$

$$s = T(r) = (L-1)\int_0^r p_r(w)dw \tag{3}$$

$$\frac{\mathrm{d}s}{\mathrm{d}r} = \frac{\mathrm{d}T(r)}{\mathrm{d}r} = (L-1)\frac{\mathrm{d}}{\mathrm{d}r}\left[\int_0^r p_r(w)dw\right] = (L-1)p_r(r) \tag{4}$$

$$p_s(s) = p_r(r)\left|\frac{\mathrm{d}r}{\mathrm{d}s}\right| = p_r(r)\frac{1}{(L-1)p_r(r)} = \frac{1}{(L-1)} \tag{5}$$

Part 3:
直方图处理与函数绘图
（直方图均衡化-离散）

（2）离散：IMPORTANT数字图像都是离散的

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n},$$
$$k = 0,1,2,\ldots,L-1$$

由于直方图是PDF的近似，不连续，用此方法很少出现完全平坦的直方图，但可以使灰度级覆盖更宽的灰度范围，进而增强图像的对比度。

Matlab 函数：histeqd

Part 3:
直方图处理与函数绘图
（直方图均衡化-离散）

（2）离散：EXAMPLE (Q)

$$s_k = T(r_k)$$
$$= (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n},$$
$$k = 0,1,2,\ldots,L-1$$

假设一副大小为64×64像素的3比特图像（L=8）具有表中的灰度分布，求s_k及$p_s(s_k)$。

r_k	n_k	$p_r(r_k)$
$r_0=0$	790	0.19
$r_1=1$	1023	0.25
$r_2=2$	850	0.21
$r_3=3$	656	0.16
$r_4=4$	329	0.08
$r_5=5$	245	0.06
$r_6=6$	122	0.03
$r_7=7$	81	0.02

Part 3: 直方图处理与函数绘图

（直方图均衡化-离散）

（2）离散：EXAMPLE (A)

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k}\frac{n_j}{n},$$
$$k = 0,1,2,\dots,L-1$$

假设一副大小为64×64像素的3比特图像（L=8）具有表中的灰度分布，求s_k及$p_s(s_k)$。

r_k	$p_r(r_k)$	s_k
$r_0 = 0$	0.19	$s_0 = 1.33 \rightarrow 1$
$r_1 = 1$	0.25	$s_1 = 3.08 \rightarrow 3$
$r_2 = 2$	0.21	$s_2 = 4.55 \rightarrow 5$
$r_3 = 3$	0.16	$s_3 = 5.67 \rightarrow 6$
$r_4 = 4$	0.08	$s_4 = 6.23 \rightarrow 6$
$r_5 = 5$	0.06	$s_5 = 6.65 \rightarrow 7$
$r_6 = 6$	0.03	$s_6 = 6.86 \rightarrow 7$
$r_7 = 7$	0.02	$s_7 = 7 \rightarrow 7$

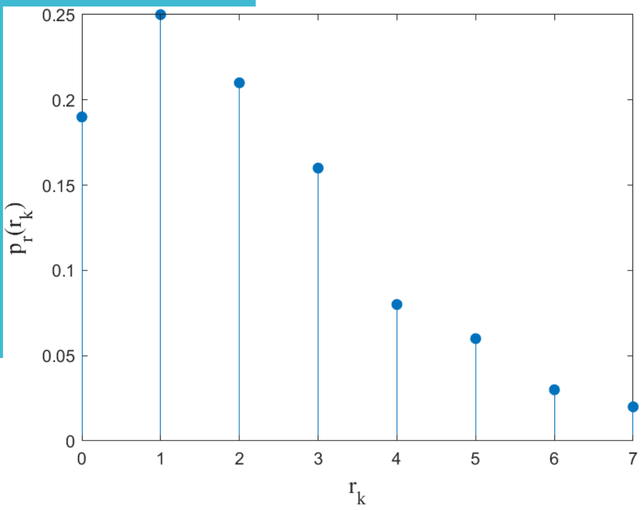
Part 3: 直方图处理与函数绘图

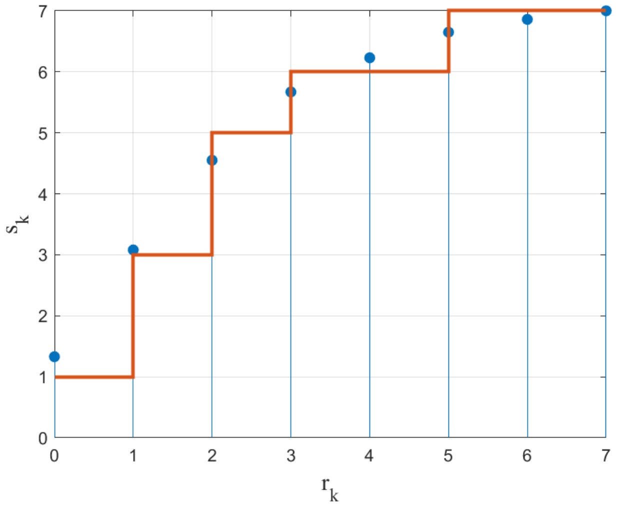
（直方图均衡化-离散）

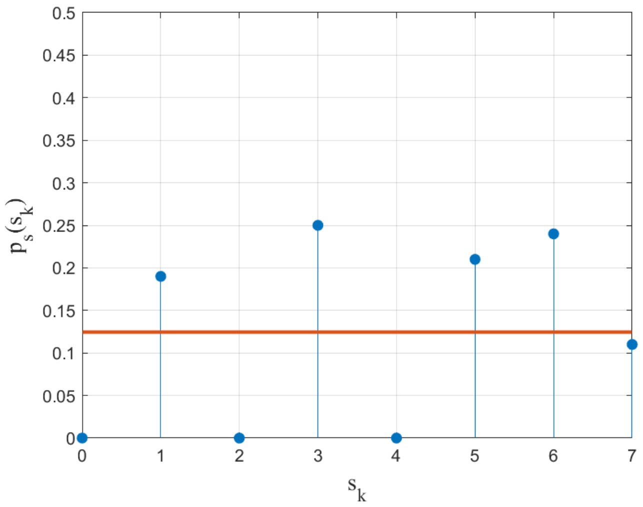
（2）离散：EXAMPLE (A)

假设一副大小为64×64像素的3比特图像（L=8）具有表中的灰度分布，求s_k及$p_s(s_k)$。

r_k	$p_r(r_k)$	s_k
$r_0=0$	0.19	$s_0=1.33\rightarrow 1$
$r_1=1$	0.25	$s_1=3.08\rightarrow 3$
$r_2=2$	0.21	$s_2=4.55\rightarrow 5$
$r_3=3$	0.16	$s_3=5.67\rightarrow 6$
$r_4=4$	0.08	$s_4=6.23\rightarrow 6$
$r_5=5$	0.06	$s_5=6.65\rightarrow 7$
$r_6=6$	0.03	$s_6=6.86\rightarrow 7$
$r_7=7$	0.02	$s_7=7\rightarrow 7$





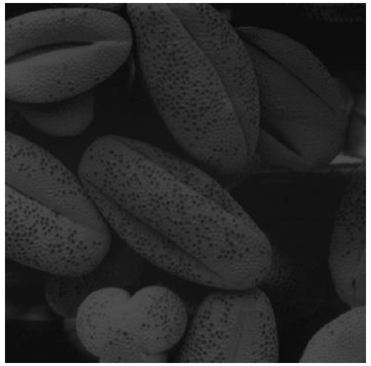


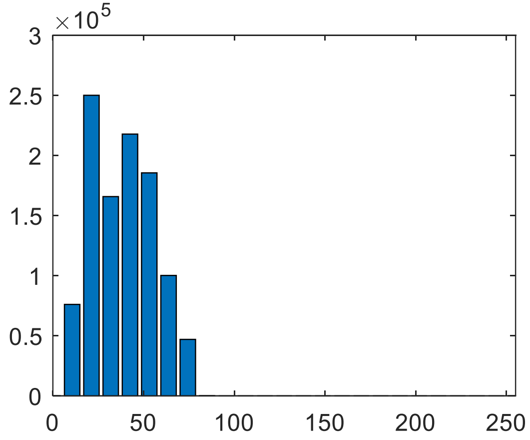
DIP2_3_4.m

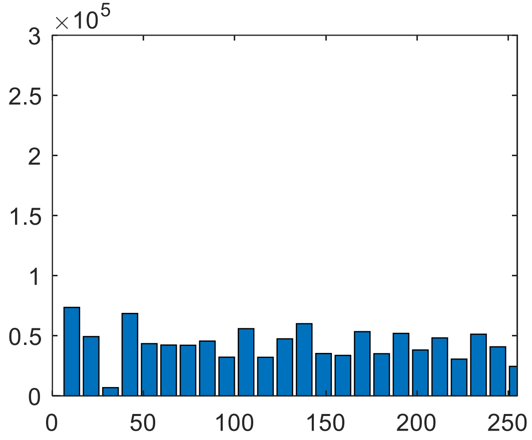
Part 3:
直方图处理与函数绘图
（直方图均衡化-离散）

Matlab 函数： histeq

通过把输入图像的灰度级扩展到较宽灰度范围来实现图像增强。







DIP2_3_3.m

Part 3:
直方图处理与函数绘图
（直方图规定化）

用于生成有规定直方图的图像的方法，称为直方图匹配或直方图规定化。

连续：输入$p_r(r)$，输出$p_z(z)$

离散：输入$p_r(r_k)$，输出$p_z(z_k)$

如何实现？

Part 3: 直方图处理与函数绘图（直方图规定化）

连续：输入$p_r(r)$，输出$p_z(z)$

Step 1: 进行均衡化变换，找到一个灰度均匀分布的中间变量s

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

$$p_s(s) = \frac{1}{(L-1)}$$

Step 2: z具有如下性质

$$s = H(z) = (L-1)\int_0^z p_z(v)dv$$

$$\frac{\mathrm{ds}}{\mathrm{dz}} = (L-1)p_z(z)$$

Step 3: 采用如下映射关系得到输出图像

$$z = H^{-1}(s) = H^{-1}[T(r)]$$

$p_z(z) = p_s(s)\left|\frac{\mathrm{ds}}{\mathrm{dr}}\right|$ 一定是你要的分布，but解析解不好算

Part 3:
直方图处理与函数绘图
（直方图规定化）

离散：输入$p_r(r_k)$，输出$p_z(z_k)$

Step 1: 进行均衡化变换，找到一个灰度接近均匀分布的中间变量s_k，$k = 0,1,2, \dots L-1$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

Step 2: z具有如下性质

$$s_k = H(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

Step 3: 采用如下映射关系得到输出图像

$$z_q = H^{-1}(s_k)$$

此处的数值解可以用穷举法试出来。

r_k	$p_r(r_k)$	z_q	$p_z(z_q)$
$r_0 = 0$	0.19	$z_0 = 0$	0
$r_1 = 1$	0.25	$z_1 = 1$	0
$r_2 = 2$	0.21	$z_2 = 2$	0
$r_3 = 3$	0.16	$z_3 = 3$	0.15
$r_4 = 4$	0.08	$z_4 = 4$	0.2
$r_5 = 5$	0.06	$z_5 = 5$	0.3
$r_6 = 6$	0.03	$z_6 = 6$	0.2
$r_7 = 7$	0.02	$z_7 = 7$	0.15

离散：输入$p_r(r_k)$，输出$p_z(z_k)$ EXAMPLE (Q)

Step 1: 进行均衡化变换，找到一个灰度接近均匀分布的中间变量s_k，$k = 0,1,2,\dots L-1$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

Step 2: z具有如下性质

$$s_k = H(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

Step 3: 采用如下映射关系得到输出图像

$$z_q = H^{-1}(s_k)$$

此处的数值解可以用穷举法试出来。

考虑与均衡化例子相同的输入图像，一副大小为64×64像素的3比特图像（L=8）具有左表中的灰度分布，我们希望变换该直方图，使它具有右表中的规定值。求s_k、$H(z_q)$和s_k到z_q的映射关系。

离散：输入$p_r(r_k)$，输出$p_z(z_k)$ EXAMPLE (A)

Step 1: 进行均衡化变换，找到一个灰度接近均匀分布的中间变量s_k，$k=0,1,2,\ldots L-1$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

r_k	$p_r(r_k)$
$r_0=0$	0.19
$r_1=1$	0.25
$r_2=2$	0.21
$r_3=3$	0.16
$r_4=4$	0.08
$r_5=5$	0.06
$r_6=6$	0.03
$r_7=7$	0.02

z_q	$p_z(z_q)$
$z_0=0$	0
$z_1=1$	0
$z_2=2$	0
$z_3=3$	0.15
$z_4=4$	0.2
$z_5=5$	0.3
$z_6=6$	0.2
$z_7=7$	0.15

r_k	$p_r(r_k)$	s_k
$r_0=0$	0.19	$s_0=1.33\rightarrow 1$
$r_1=1$	0.25	$s_1=3.08\rightarrow 3$
$r_2=2$	0.21	$s_2=4.55\rightarrow 5$
$r_3=3$	0.16	$s_3=5.67\rightarrow 6$
$r_4=4$	0.08	$s_4=6.23\rightarrow 6$
$r_5=5$	0.06	$s_5=6.65\rightarrow 7$
$r_6=6$	0.03	$s_6=6.86\rightarrow 7$
$r_7=7$	0.02	$s_7=7\rightarrow 7$

离散：输入$p_r(r_k)$，输出$p_z(z_k)$ EXAMPLE (A)

Step 2: z具有如下性质

r_k	$p_r(r_k)$
$r_0 = 0$	0.19
$r_1 = 1$	0.25
$r_2 = 2$	0.21
$r_3 = 3$	0.16
$r_4 = 4$	0.08
$r_5 = 5$	0.06
$r_6 = 6$	0.03
$r_7 = 7$	0.02

z_q	$p_z(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	0.15
$z_4 = 4$	0.2
$z_5 = 5$	0.3
$z_6 = 6$	0.2
$z_7 = 7$	0.15

$$s_k = H(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

z_q	$p_z(z_q)$	$H(z_q)$
$z_0 = 0$	0	$H(z_0) = 0 \to 0$
$z_1 = 1$	0	$H(z_1) = 0 \to 0$
$z_2 = 2$	0	$H(z_2) = 0 \to 0$
$z_3 = 3$	0.15	$H(z_3) = 1.05 \to 1$
$z_4 = 4$	0.2	$H(z_4) = 2.45 \to 2$
$z_5 = 5$	0.3	$H(z_5) = 4.55 \to 5$
$z_6 = 6$	0.2	$H(z_6) = 5.95 \to 6$
$z_7 = 7$	0.15	$H(z_7) = 7.00 \to 7$

Part 3:
直方图处理与函数绘图
（直方图规定化）

离散：输入$p_r(r_k)$，输出$p_z(z_k)$ EXAMPLE (A)

Step 3: 采用如下映射关系得到输出图像

$$z_q = H^{-1}(s_k)$$

此处的数值解可以用穷举法试出来。

对每个s_k，找到z_q的对应值，使得$H(z_q)$最接近s_k。当映射不唯一，算则最小的z_q值。

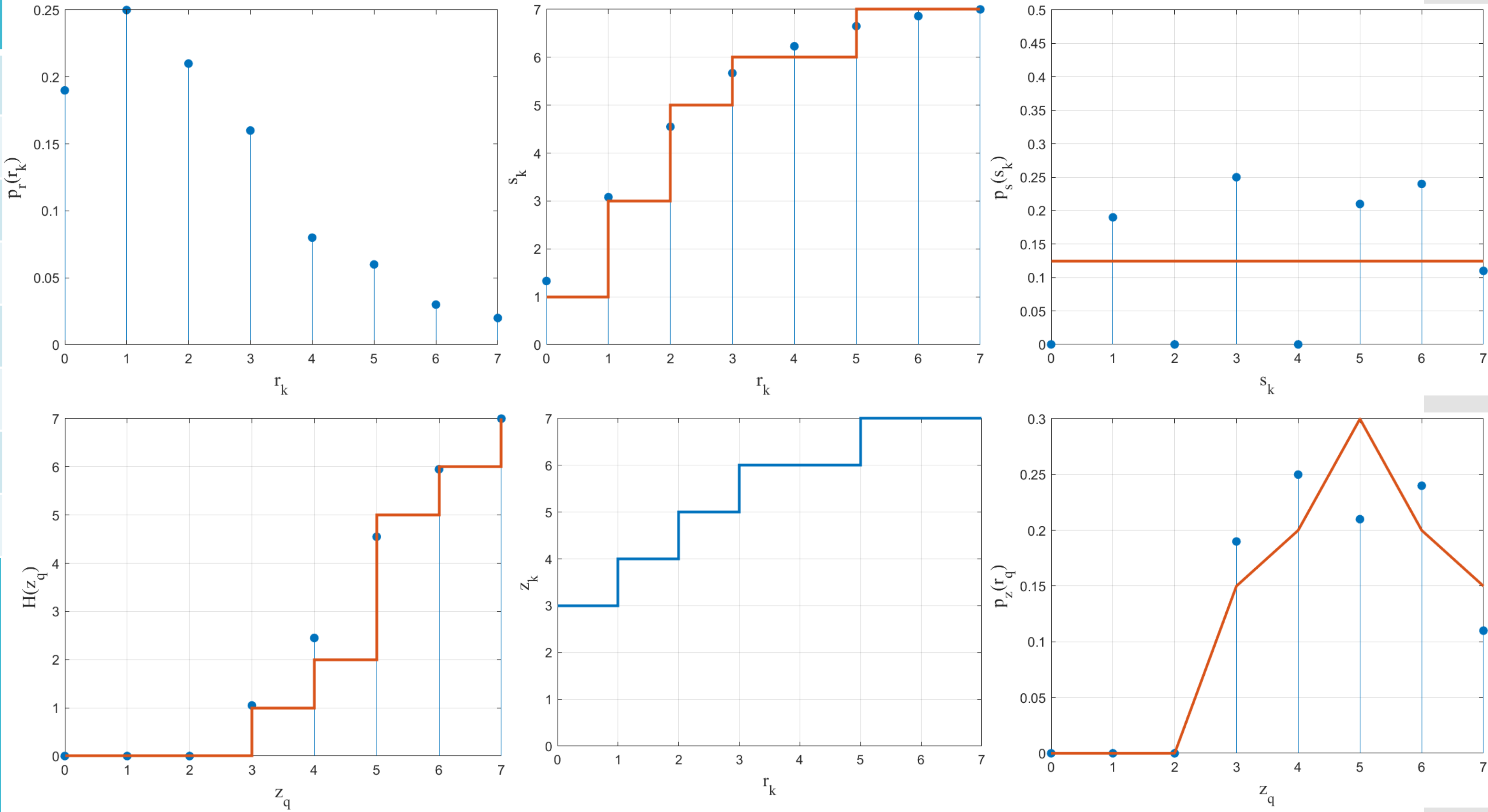
s_k	$H(z_q)$
$s_0 = 1$	$H(z_0) = 0$
$s_1 = 3$	$H(z_1) = 0$
$s_2 = 5$	$H(z_2) = 0$
$s_3 = 6$	$H(z_3) = 1$
$s_4 = 6$	$H(z_4) = 2$
$s_5 = 7$	$H(z_5) = 5$
$s_6 = 7$	$H(z_6) = 6$
$s_7 = 7$	$H(z_7) = 7$

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

离散：输入$p_r(r_k)$，输出$p_z(z_k)$ EXAMPLE (A)

r_k	$p_r(r_k)$	z_q	$p_z(z_q)$
$r_0 = 0$	0.19	$z_0 = 0$	0
$r_1 = 1$	0.25	$z_1 = 1$	0
$r_2 = 2$	0.21	$z_2 = 2$	0
$r_3 = 3$	0.16	$z_3 = 3$	0.15
$r_4 = 4$	0.08	$z_4 = 4$	0.2
$r_5 = 5$	0.06	$z_5 = 5$	0.3
$r_6 = 6$	0.03	$z_6 = 6$	0.2
$r_7 = 7$	0.02	$z_7 = 7$	0.15

s_k	→	z_q
1	→	3
3	→	4
5	→	5
6	→	6
7	→	7



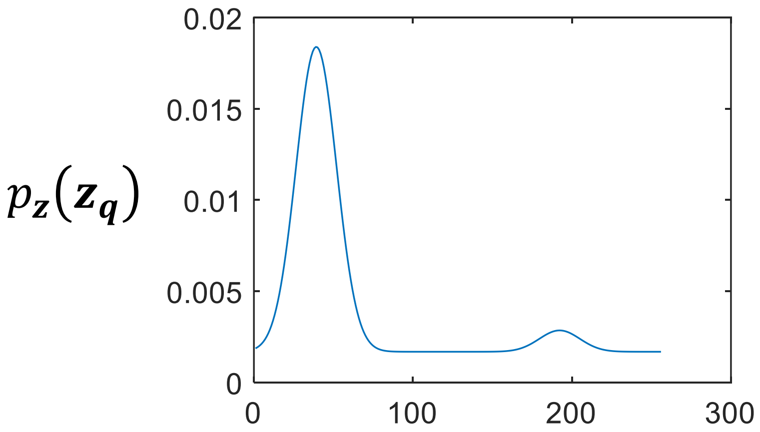
考虑与均衡化例子相同的输入图像，一副大小为64×64像素的3比特图像（L=8）具有左表中的灰度分布，我们希望变换该直方图，使它具有右表中的规定值。求s_k、$H(z_q)$和s_k到z_q的映射关系。

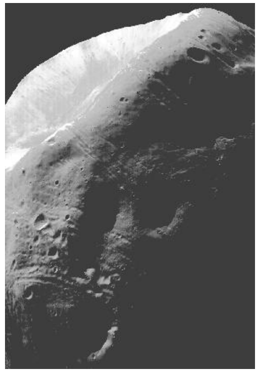
Part 3:
直方图处理与函数绘图
（直方图规定化）

Matlab 函数： histeq 、 twomodegauss （自己写的）

原图

均衡后





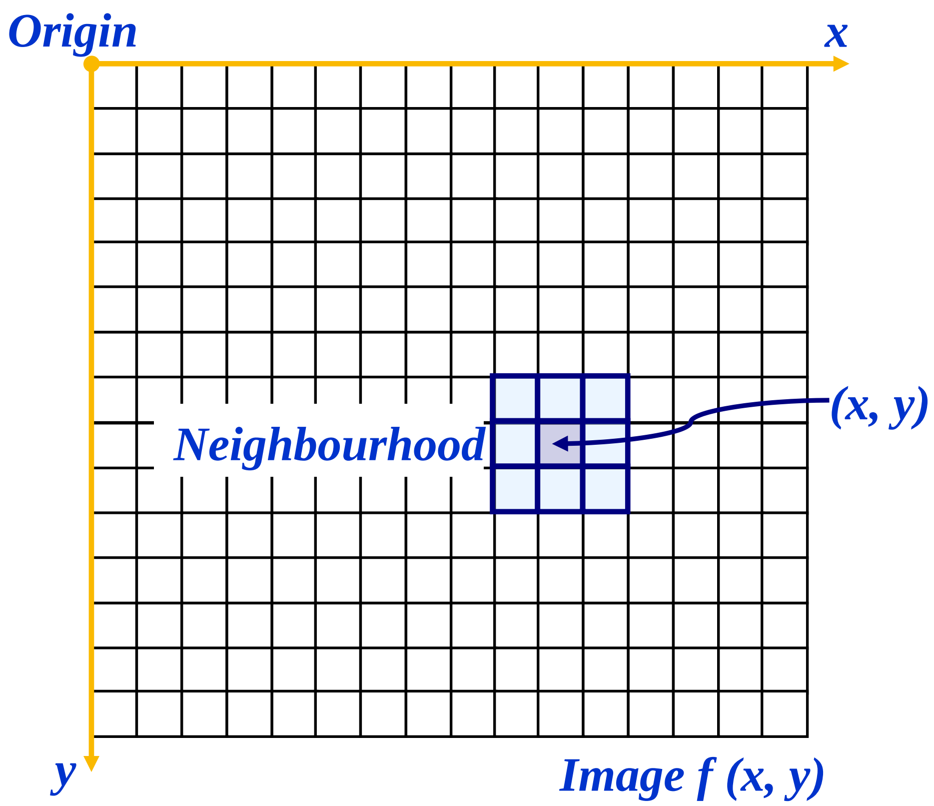
规定化后

Part 4:
空间滤波

- 空域滤波是一种邻域处理方法，通过把每个像素的值替换为该像素及其邻域的函数值来修改图像。
- 作用域：像素及其邻域
- 目的： 达到平滑或锐化图像的作用。

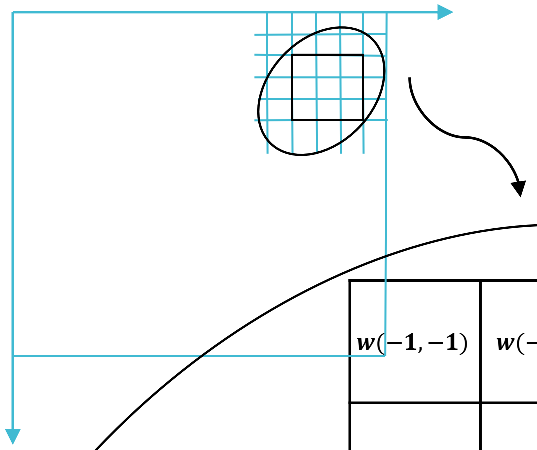
Part 4:
空间滤波

- 空域滤波通常使用空域模板进行的图像处理，模板本身被称为空域滤波器（$m \times n$矩阵，也称为滤波模板、核、掩模或窗口）。



Part 4:
空间滤波
（线性空间滤波）

- 线性空间滤波（空间卷积）：对邻域中像素执行线性运算。
- 像素的输出值是计算该像素邻域内像素值的线性组合，系数矩阵称为**模板**。
- 通常使用滤波模板与图像的空域卷积来实现的，因此滤波模板也称为**卷积模板**。



滤波器核

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

$f(x-1,y-1)$	$w(x-1,y)$	$w(x-1,x+1)$
$f(x,y-1)$	$w(x,y)$	$w(x,x+1)$
$f(x+1,y-1)$	$w(x+1,y)$	$w(x+1,x+1)$

以（x,y）为中心时核下面的像素值

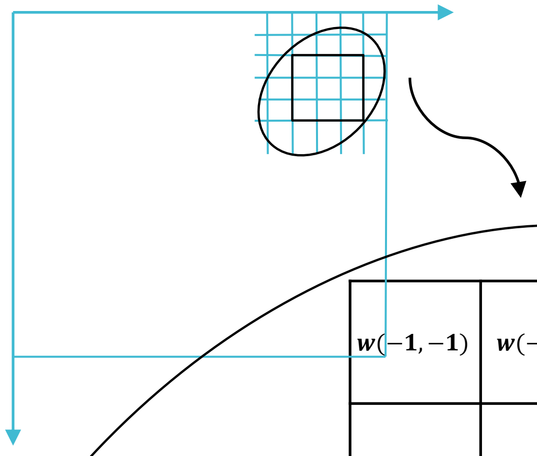
Part 4:
空间滤波
（线性空间滤波）

- 相关和卷积
- 相关：

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

- 卷积：

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x-s,y-t)$$



滤波器核

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

$f(x-1,y-1)$	$w(x-1,y)$	$w(x-1,x+1)$
$f(x,y-1)$	$w(x,y)$	$w(x,x+1)$
$f(x+1,y-1)$	$w(x+1,y)$	$w(x+1,x+1)$

以（x,y）为中心时核下面的像素值

Part 4:

空间滤波

（线性空间滤波）

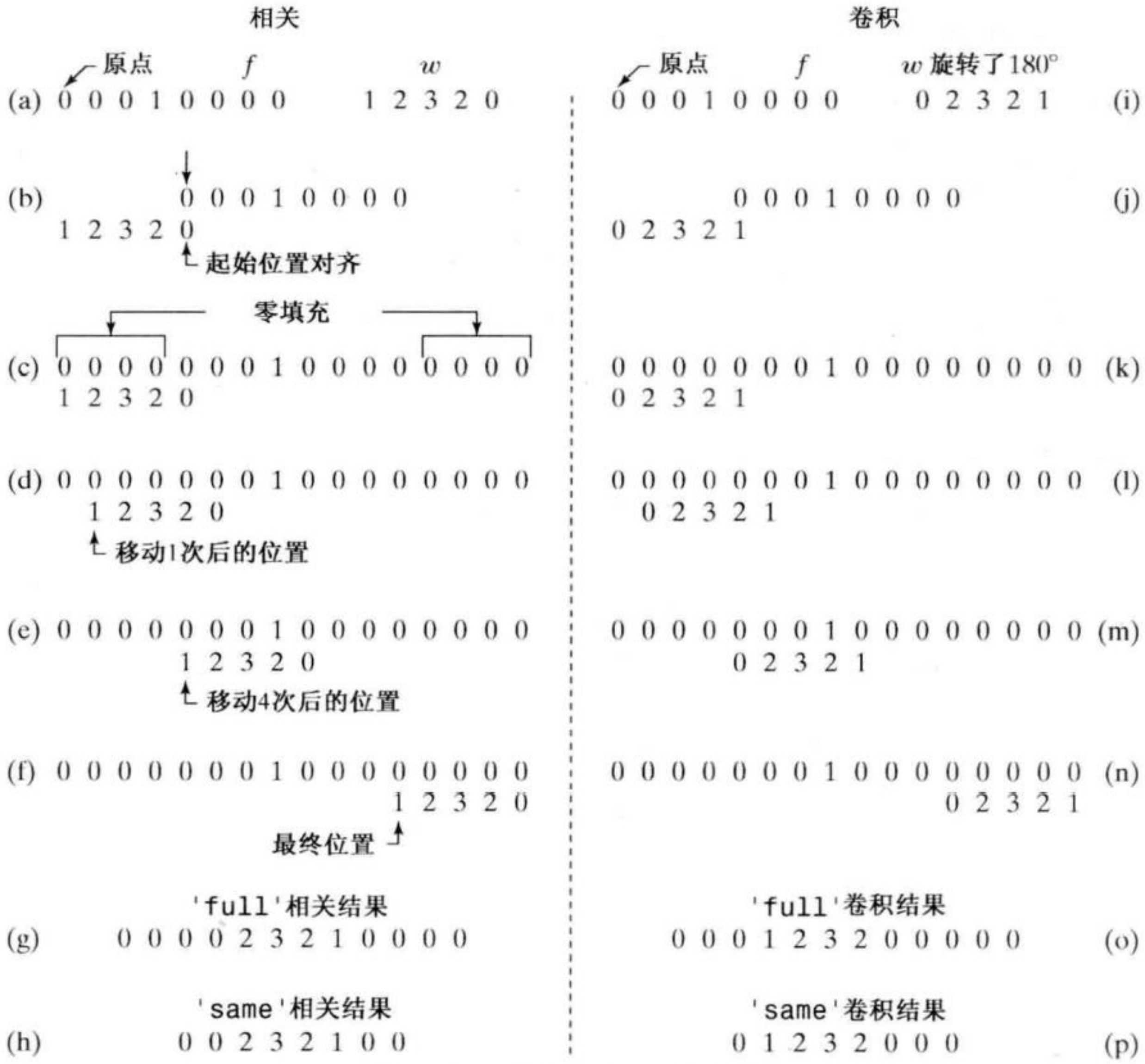


图 2.14 一维相关和卷积的说明

Part 4:
空间滤波
（线性空间滤波）

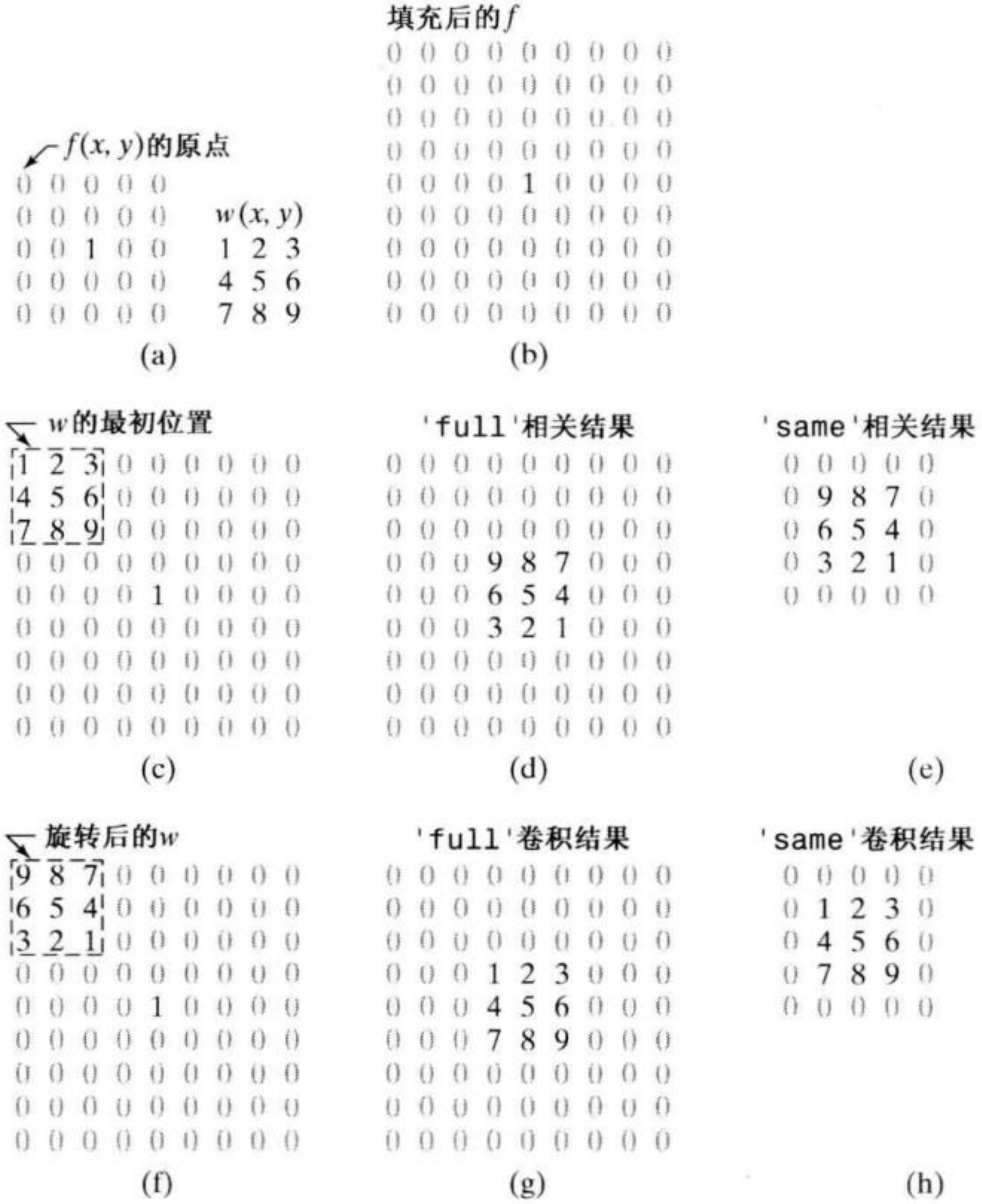
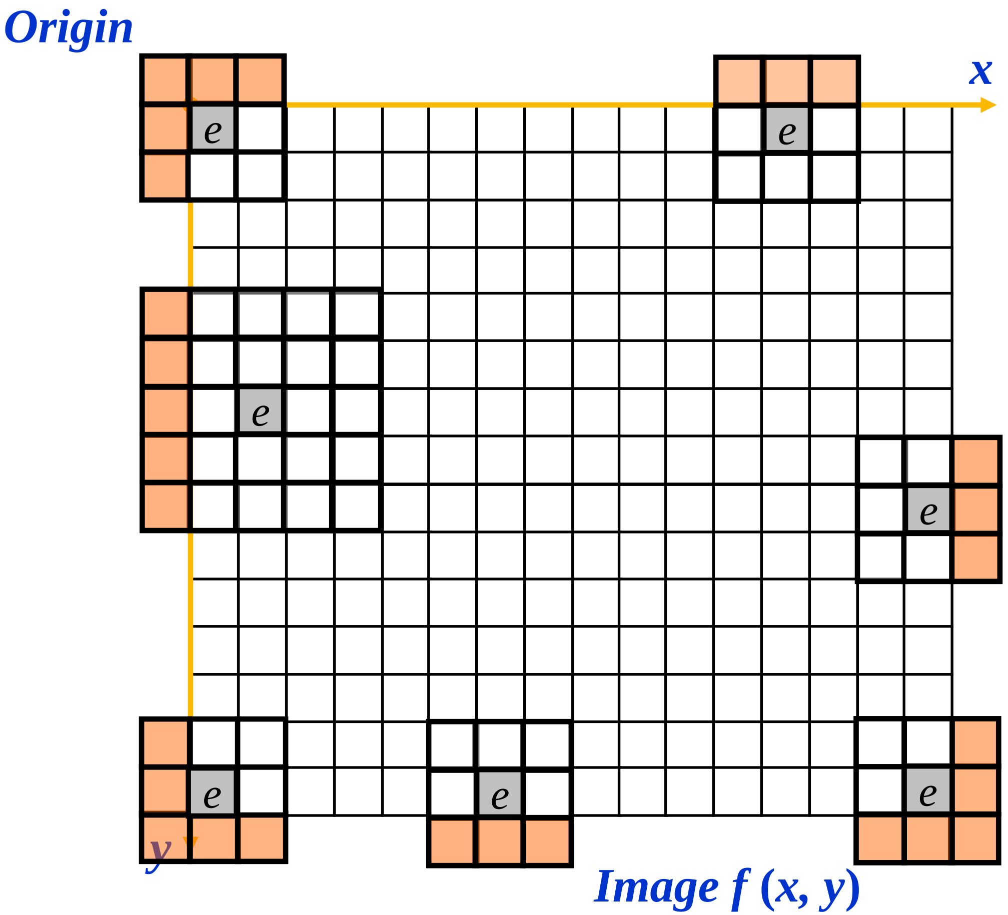


图 2.15　二维相关和卷积的说明。为便于观察，0 用灰色显示

Part 4:
空间滤波
（线性空间滤波）

- 图像边缘扩展（零填充、镜像填充、复制填充等）

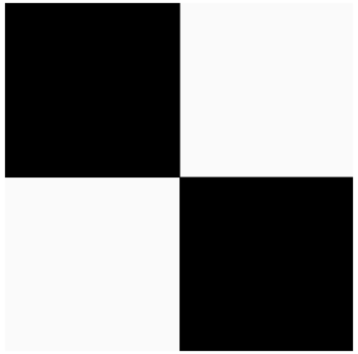


Part 4:
空间滤波
（线性空间滤波）

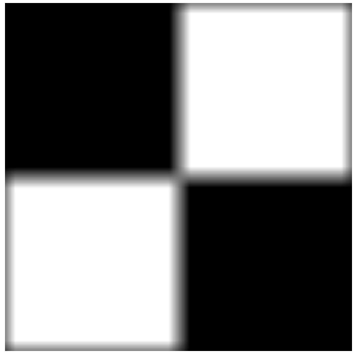
Matlab 函数： imfilter

例1： 平滑低通 （盒式滤波器）

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\qquad \frac{1}{4.8976}\times\begin{bmatrix}0.3679&0.6065&0.3679\\0.6065&1&0.6065\\0.3679&0.6065&0.3679\end{bmatrix}$$



原图

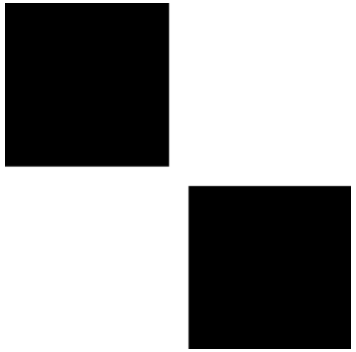


31*31盒式核滤波+零填充

31*31盒式核滤波+replicate

31*31盒式核滤波+symmetric

31*31盒式核滤波+circular

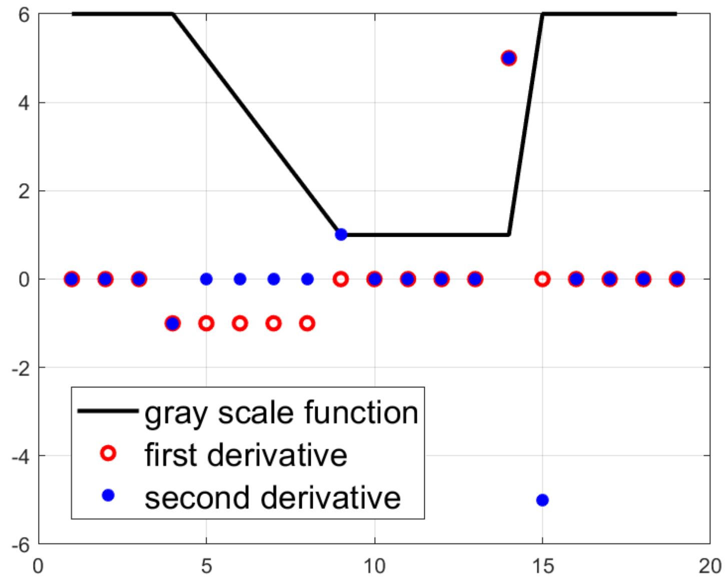


unit8 31*31盒式核滤波

Part 4:
空间滤波
（线性空间滤波）

Matlab 函数： imfilter

例2： 锐化高通（拉普拉斯滤波器）



原理：原图像、一阶导数在斜坡处不为零，二阶导数在边缘处不为零，因此二阶导数用于锐化。

DIP2_4_2.m

Part 4:
空间滤波
（线性空间滤波）

Matlab 函数： imfilter

例2： 锐化高通（拉普拉斯滤波器）

拉普拉斯算子：

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

矩阵： $\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \quad \begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \quad \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \quad \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$

Part 4:
空间滤波
（线性空间滤波）

Matlab 函数： imfilter

例2：锐化高通（拉普拉斯滤波器）

原图

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

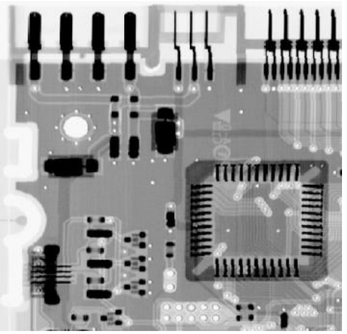
$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

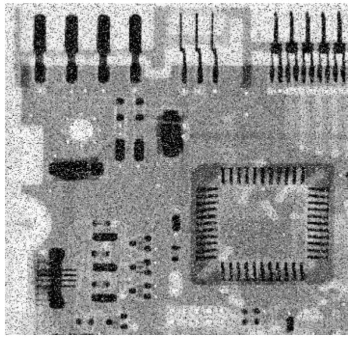
DIP2_4_3.m

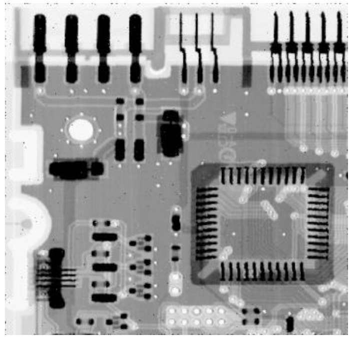
Part 4:
空间滤波
（非线性空间滤波）

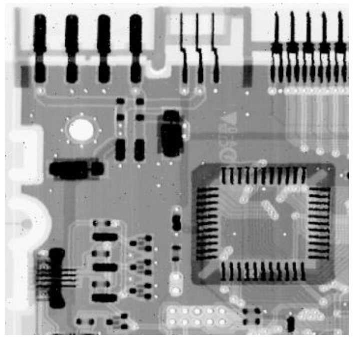
Matlab 函数： ordfilt2

例：统计排序滤波器（中值滤波）


原图


加椒盐噪声后


中值滤波后


中值滤波后
（symmetric）选项

DIP2_4_4.m