第四章　图像增强

4.1 概述与分类

定义：

图像增强是指按特定的需要突出一幅图像中的某些信息，同时，削弱或去除某些不需要的信息的处理方法。

原始图像

增强后的图像



A. 原始图像



B. 增强后的图像

- **主要目的**：

  是使处理后的图像对某种特定的应用来说，比原始图像更适用。

- **前提**：不考虑图像降质的原因。

- **结果**：改善后的图像不一定要逼近原图像。

注意：

1、图像增强处理**并不能增加原始图像的信息**，其结果只能增强对某种信息的辨别能力，而这种处理肯定会损失一些其它信息.

2. 图像增强处理最大的困难：　**增强后图像质量的好坏**主要依靠人的主观视觉来评定，也就是说，**难以定量描述**。

图像增强技术基本上可分成两大类：

1. 空间域法

主要是在空间域对图像像素灰度值直接运算处理。

空间域法的图像增强技术可以用下式和图来描述。

$$g(x,y)=f(x,y)\ h(x,y)$$

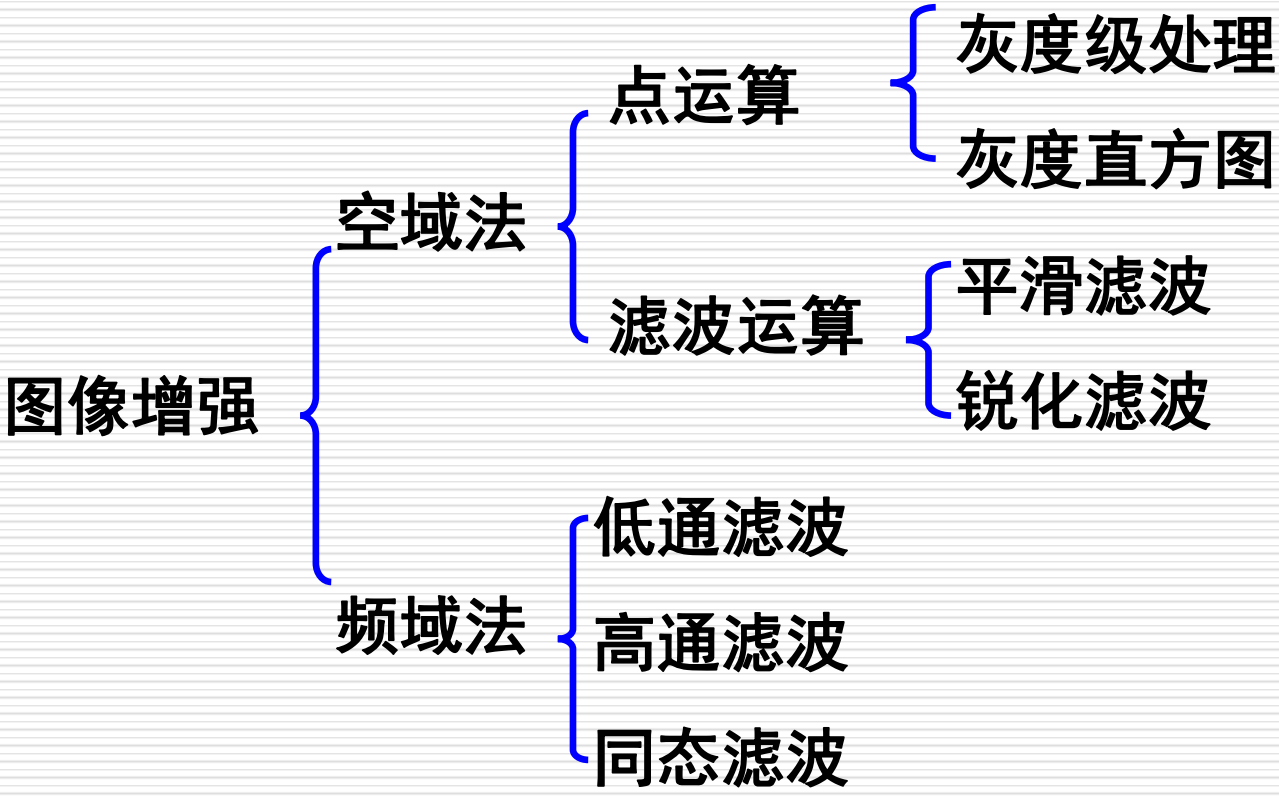
其中，f(x, y)、g(x, y)为增强处理前后的函数，h(x, y)为空间运算函数。

2. 频率域法

在图像的某种变换域内，对图像的变化值进行运算（增强感兴趣的频率分量），然后通过逆变换获得图像增强效果。

如先对图像进行傅里叶变换，再对图像的频谱进行某种修正（如滤波等），最后将修正后的变化值逆变换到空间域，从而获得增强后的图像。

其中，F(u,v)、G(u,v)分别为增强处理前后图像f(x, y)、g(x, y)的频率正变换，H(u,v)为滤波函数。


图像增强
空域法
点运算
灰度级处理
灰度直方图
滤波运算
平滑滤波
锐化滤波
频域法
低通滤波
高通滤波
同态滤波

4.2 基于点运算的增强

当算子T的作用域是以每一单个像素为单位，图像的输出g(x, y)只与位置(x, y)处的输入f(x, y)有关，实现的是像素点到点的处理时，称这种运算为“点运算”。

点运算可以表示为

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

其中T(.)称为灰度变换函数，它描述了输入灰度值和输出灰度值之间的转换关系。

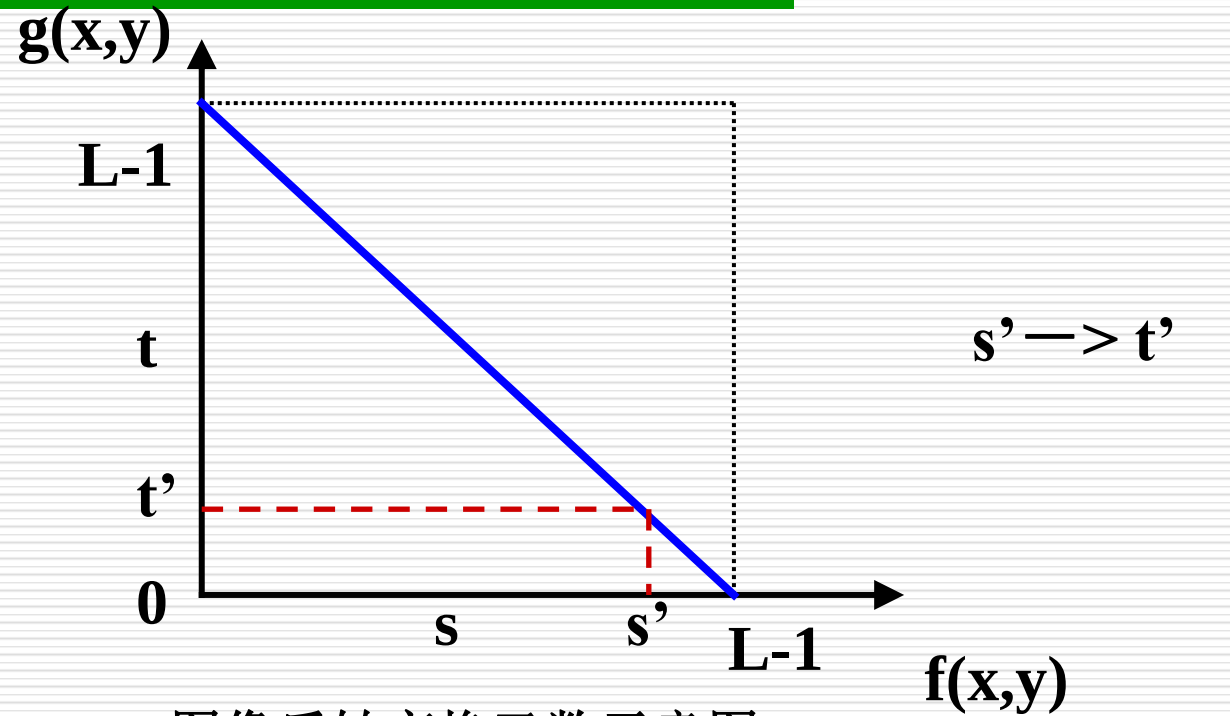
4.2.1 直接的灰度变换

1. 图像反转

对图像求反是**将原始图像的灰度值进行反转**，使输出图像的强度随输入图像的强度增加而减少。

普通的黑白底片和照片的关系就是如此。

灰度反转处理效果



图像反转变换函数示意图

其数学表达式为：

$$g(x, y)=-f(x, y)+(L-1)$$

其中，[0, L-1]是图像灰度级范围。

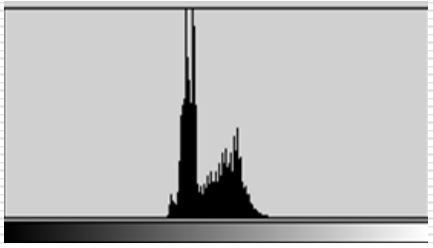
%图像反转的程序实现

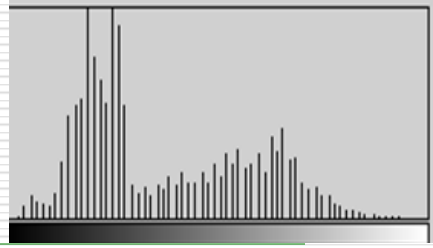
```
I=imread('a.bmp');
J=double(I);        %将图像矩阵转化为double类型
J=-J+(256-1);       %图像反转线性变换
H=uint8(J);         %double数据类型转化为unit8数据类型
imshow(H);
```

2. 灰度线性变换

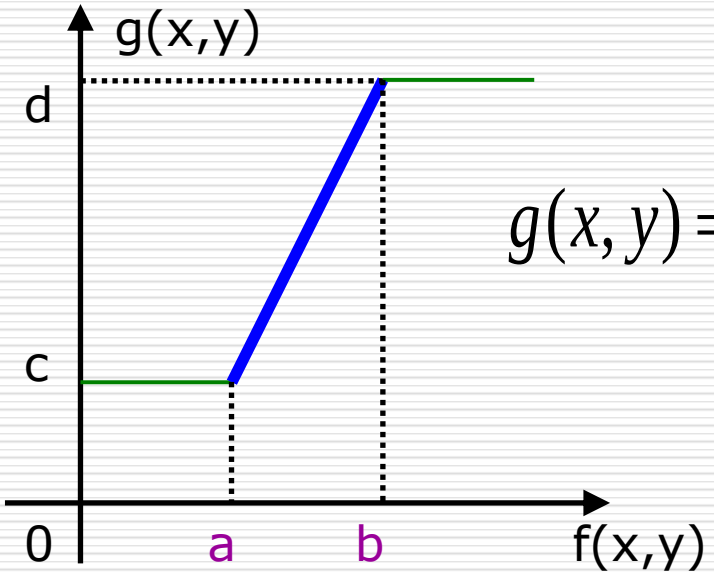
在曝光不足或过度的情况下，图像灰度可能会局限在一个很小的范围内，使图像的细节分辨不清，显示出来的图像模糊不清，看上去没有灰度层次感。

下图是对曝光不足的图像采用线性变换对图像每一个像素灰度作线性拉伸，可有效地改善图像视觉效果。





设图像f(x,y)的灰度范围为0~M，其中大部分像素的灰度级分布在区间[a,b]内，根据图像处理的需要，将其灰度范围变换到[c,d]，如图所示，变换后的图像为g(x,y)，则：



$$g(x,y)=\begin{cases} c & 0\le f(x,y)<a \\ \dfrac{d-c}{b-a}[f(x,y)-a]+c & a\le f(x,y)\le b \\ d & b<f(x,y)\le M \end{cases}$$

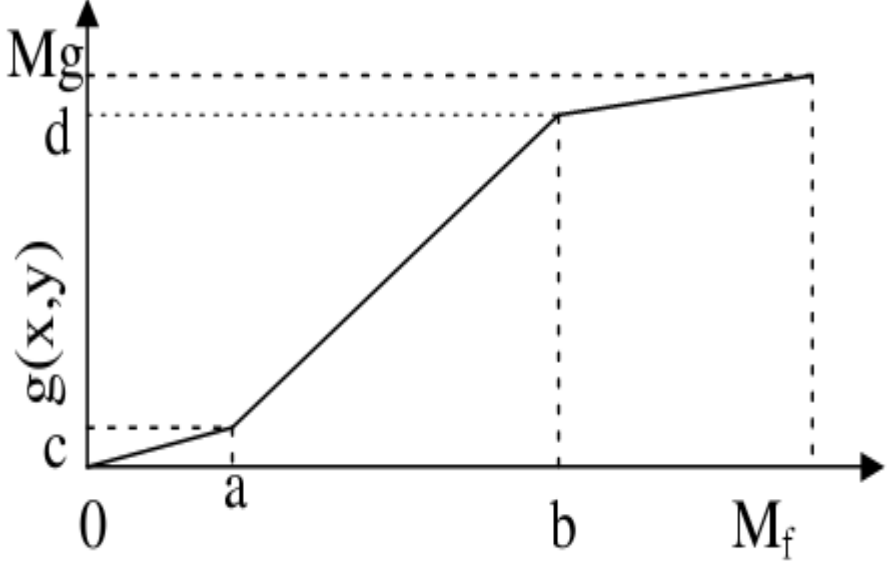
图像灰度变换前后效果对比图：

变换前　　变换后

3. 分段线性变换

突出感兴趣的灰度区间内的图像细节，相对抑制不感兴趣的灰度级，将图像灰度区间分成两段乃至多段分别作线性变换。

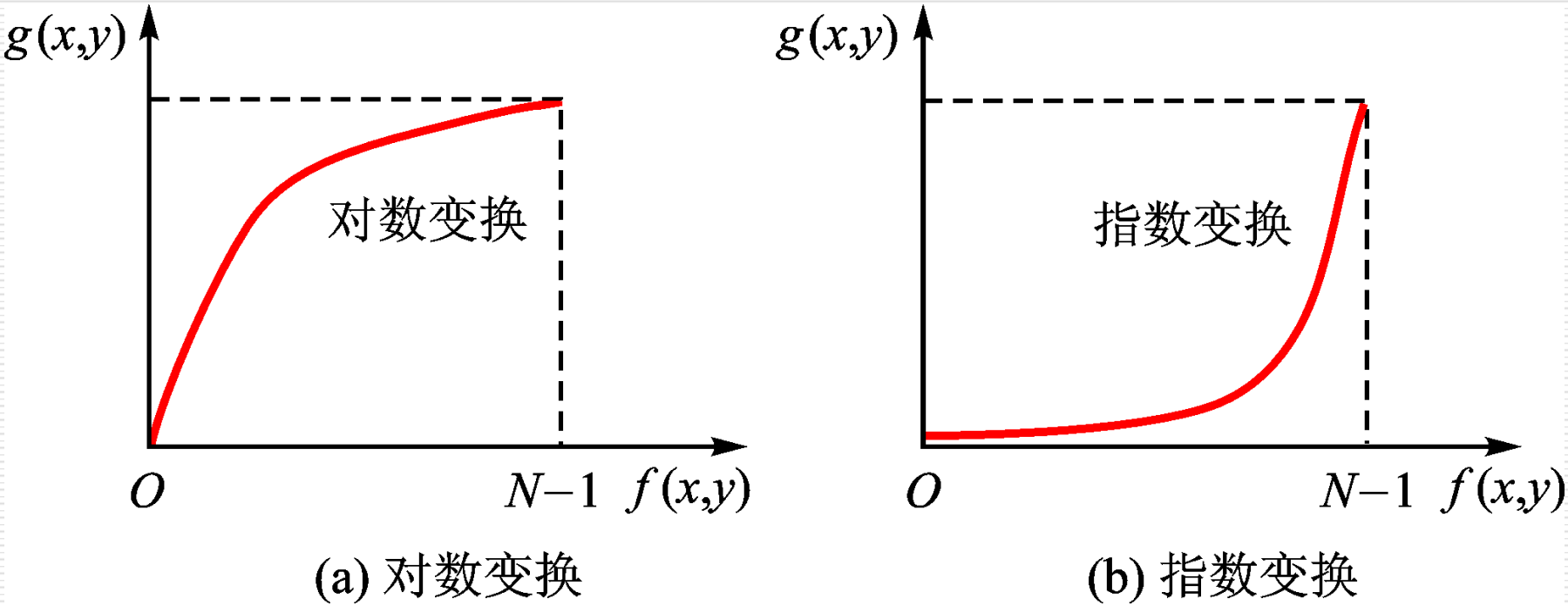
在黑色或白色附近有噪声干扰时用：如照片中的划痕。



$$g(x,y)=\begin{cases}\dfrac{c}{a}f(x,y) & 0\le f(x,y)\le a\\ \dfrac{d-c}{b-a}[f(x,y)-a]+c & a\le f(x,y)\le b\\ \dfrac{M_g-d}{M_f-b}[f(x,y)-b]+d & b\le f(x,y)\le M_f\end{cases}$$

使用分段线性变换，通过调节折线拐点的位置及控制分段直线的斜率，可对不同区间进行压缩或扩展。

4. 灰度非线性变换



对数变换特点：

扩展低值灰度区，压缩高值灰度区。

指数变换特点：

压缩低值灰度区，扩展高值灰度区。

① 对数变换 (Log Transformations)

$$g(x, y) = a + \frac{\ln[f(x, y) + 1]}{b \cdot \ln c}$$ a, b, c 为参数，

② 指数变换(Power-Law Transformations)

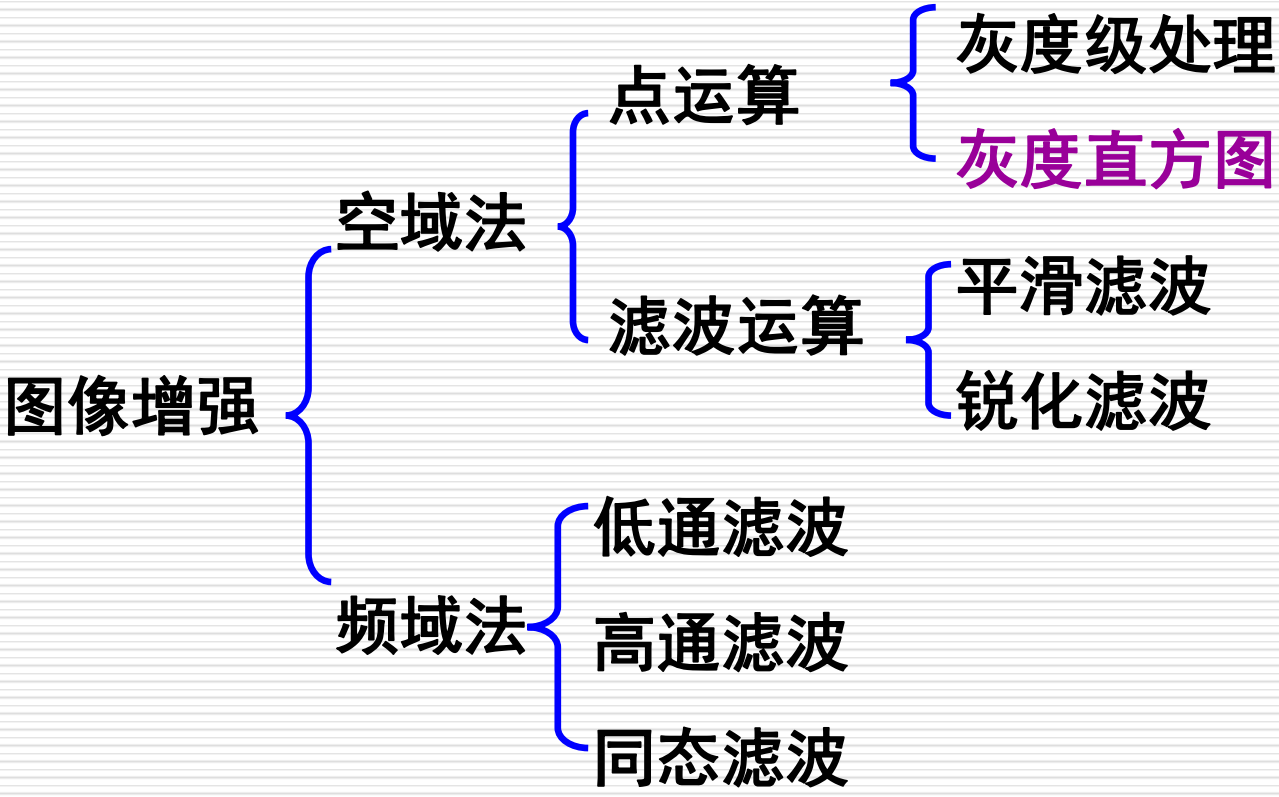
$$g(x, y) = b^{c[f(x, y) - a]} - 1$$ 其中 a, b, c 为调节参数。

采用对数变换处理的图像

(a) 原始图像

(b) 对数变换后的图像

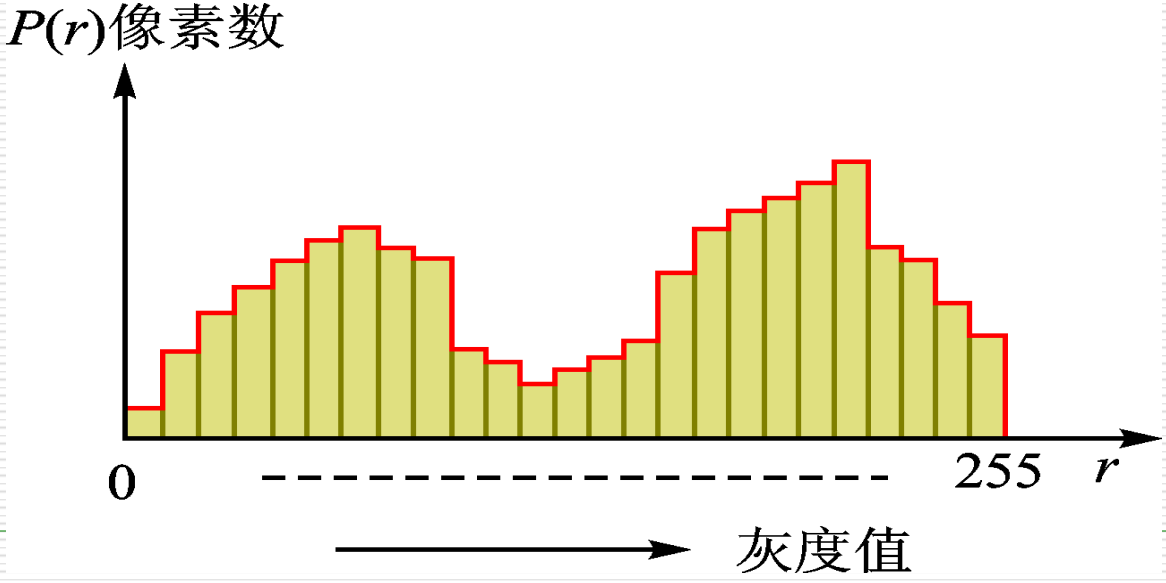
对数变换前、后图像效果图


图像增强
空域法
点运算
灰度级处理
灰度直方图
滤波运算
平滑滤波
锐化滤波
频域法
低通滤波
高通滤波
同态滤波

4.2.2 灰度直方图的处理

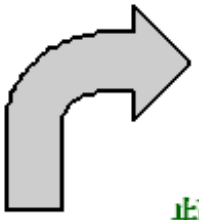
1. 灰度直方图

图像的灰度直方图是图像的重要统计特性，是反映一幅图像中的灰度级与出现这种灰度的频率之间的关系的图形。

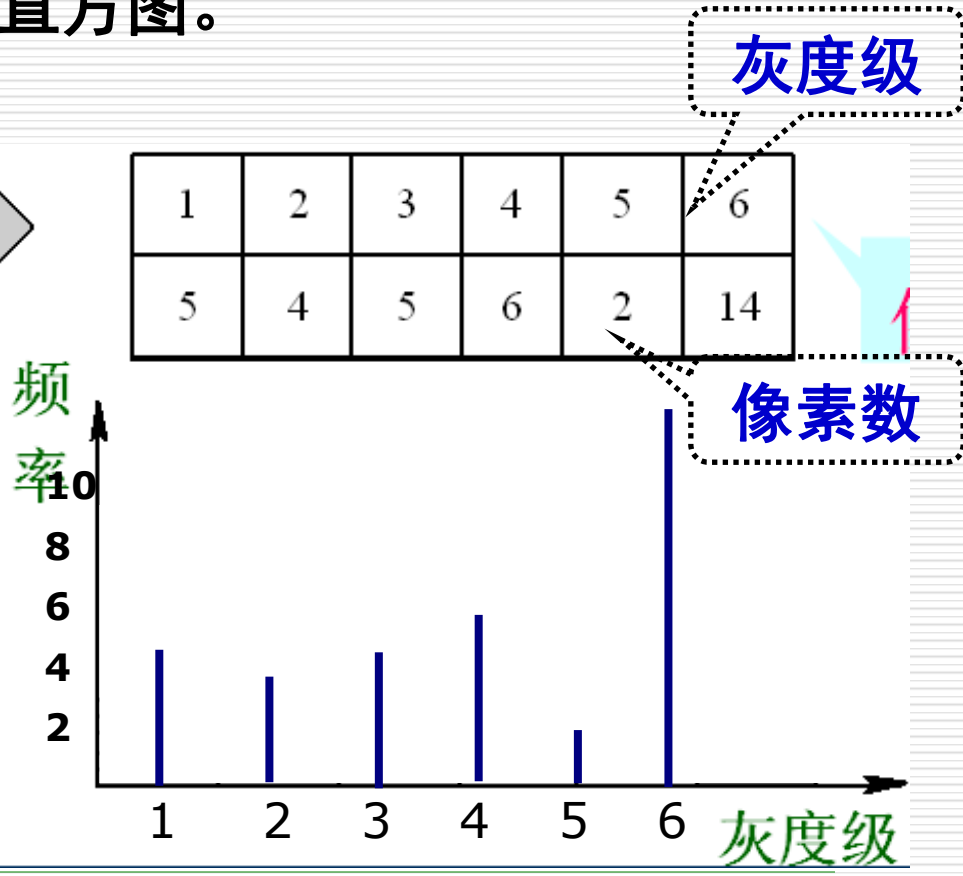


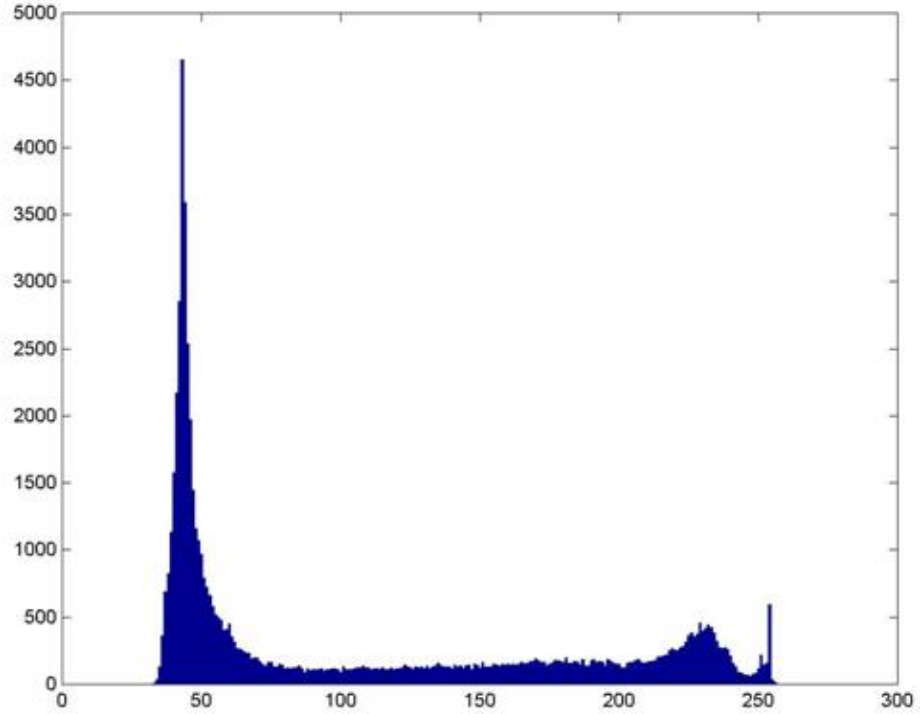
以灰度级为横坐标，灰度级的频率为纵坐标，绘制频率与灰度级的关系图就是灰度直方图。

1	2	3	4	5	6
6	4	3	2	2	1
1	6	6	4	6	6
3	4	5	6	6	6
1	4	6	6	2	3
1	3	6	4	6	6



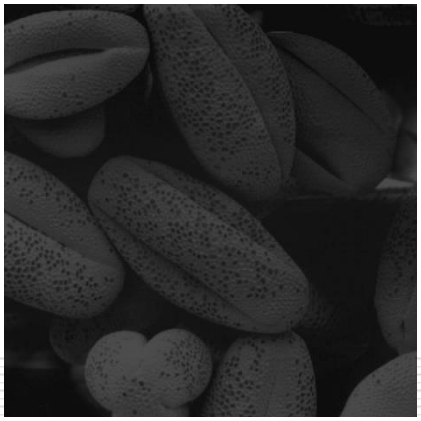
1	2	3	4	5	6
5	4	5	6	2	14

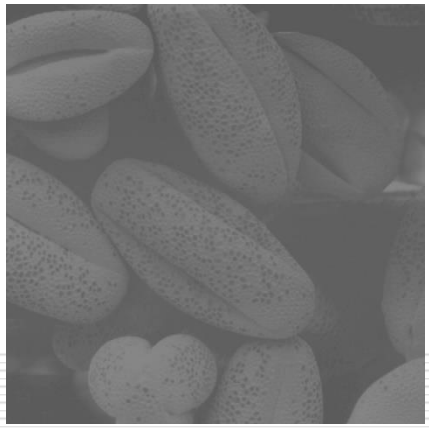
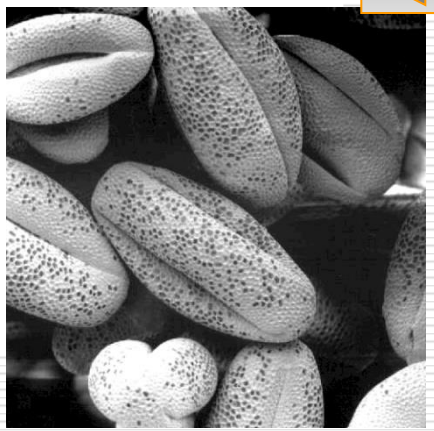


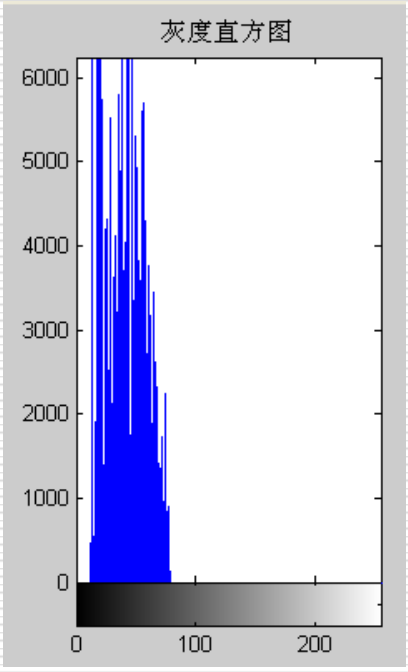

5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

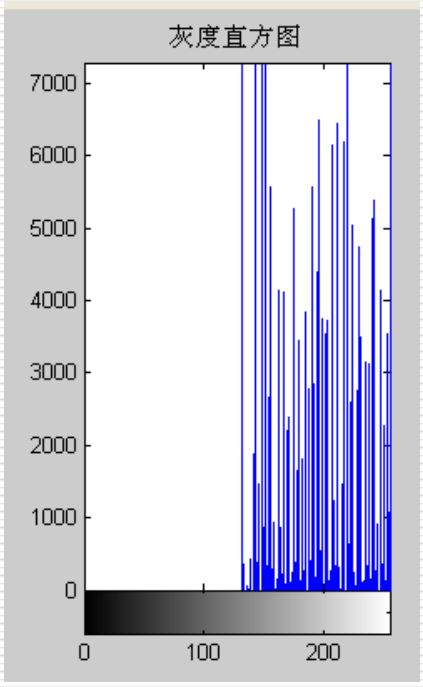
图像灰度直方图

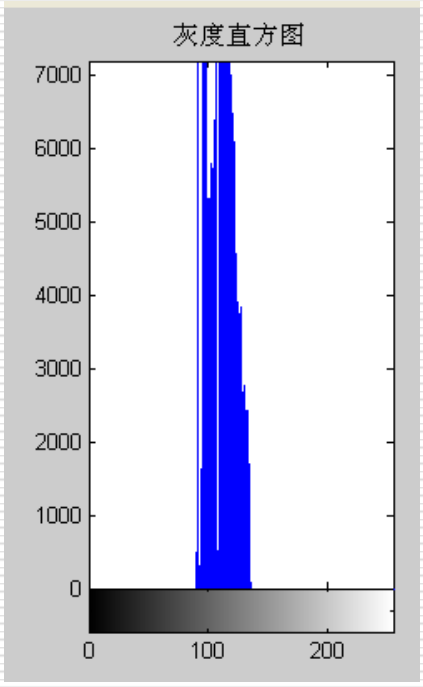
图像的灰度直方图是反映一幅数字图像概貌性特征的重要手段，如它可以给出一幅数字图像的灰度范围、灰度分布、整幅图像的平均亮度和明暗对比度等，并可以由此得出进一步处理的重要依据。

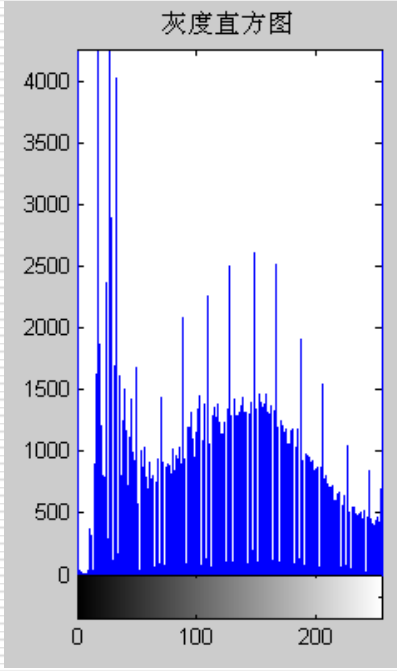


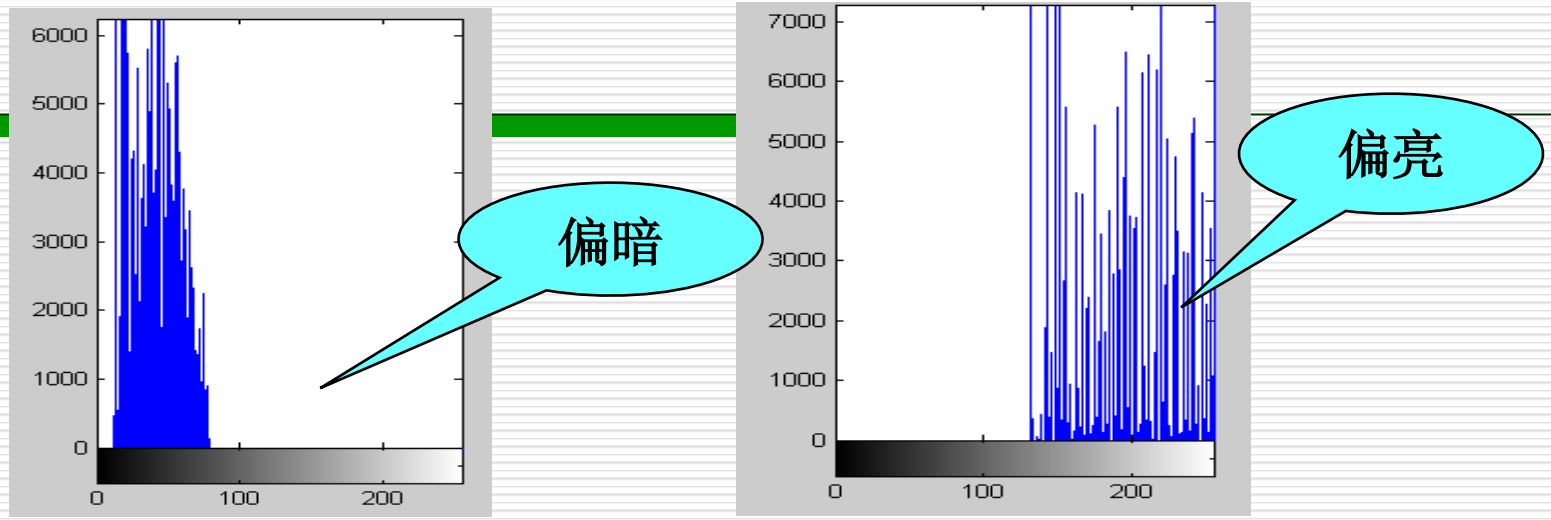








花粉图像的暗、亮、低对比度和高对比度的四种基本图像类型及其相应的直方图

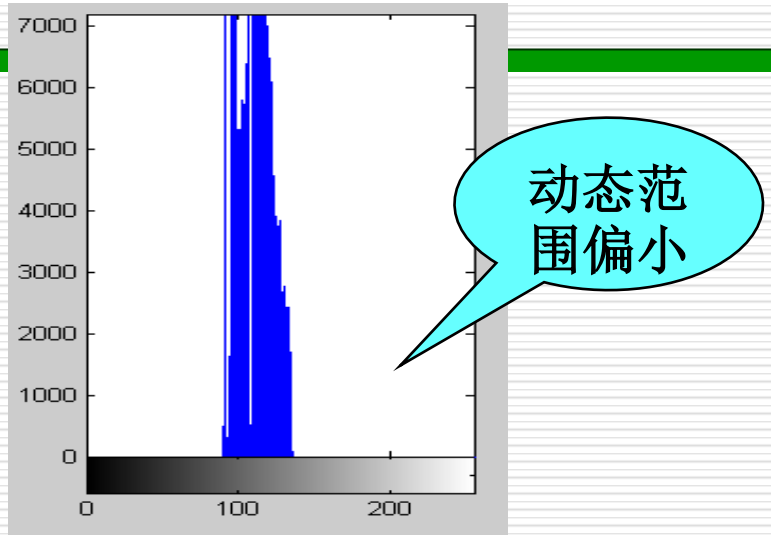


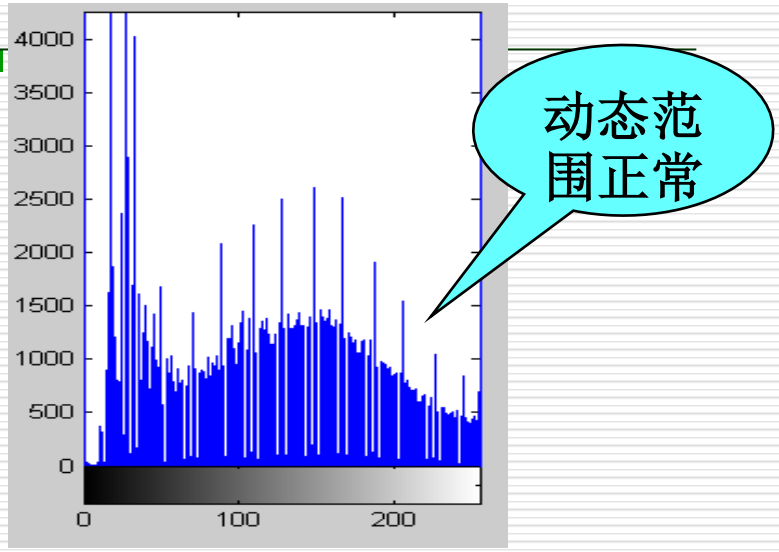
a **b**

不同类型的直方图(1)

- 图a的大多数像素灰度值取在较暗的区域。所以这幅图像肯定**较暗**。

- 图b图像的像素灰度值集中在亮区，因此图像的特性**偏亮**，。





c **d**

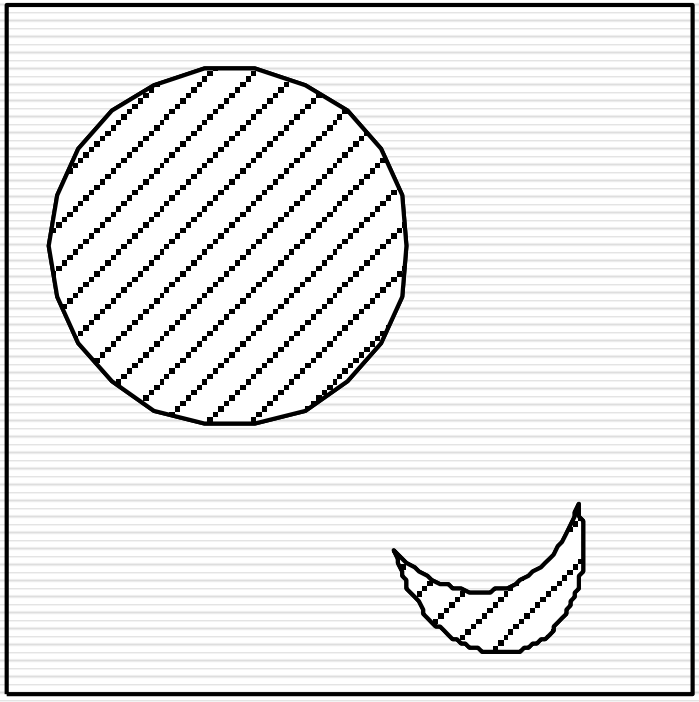
不同类型的直方图(2)

- **图c图像的像素灰度值集中在某个较小的范围内。**

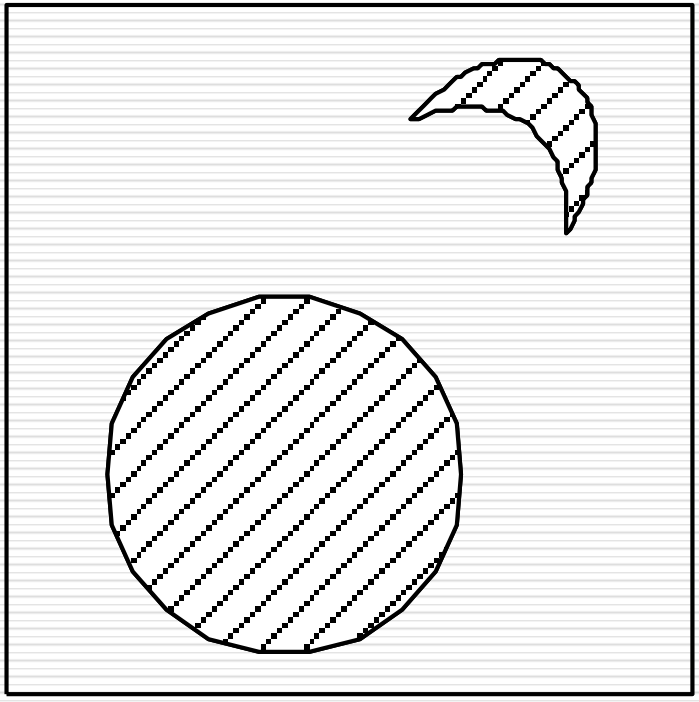
直方图的性质

（1） 直方图是一幅图像中各像素灰度值出现次数（或频数）的统计结果，它只反映该图像中不同灰度值出现的次数（或频数），而未反映某一灰度值像素所在位置。也就是说，它只包含了该图像中某一灰度值的像素出现的概率，而丢失了其所在位置的信息。

（2）任一幅图像，都能唯一地确定出一幅与它对应的直方图，但不同的图像，可能有相同的直方图。也就是说，图像与直方图之间是多对一的映射关系。



(a)



(b)

图像与直方图间的多对一关系

灰度直方图的编程实现

MATLAB图像处理工具箱提供了imhist函数来计算和显示图像的直方图，imhist函数的语法格式为：

imhist(I, n)

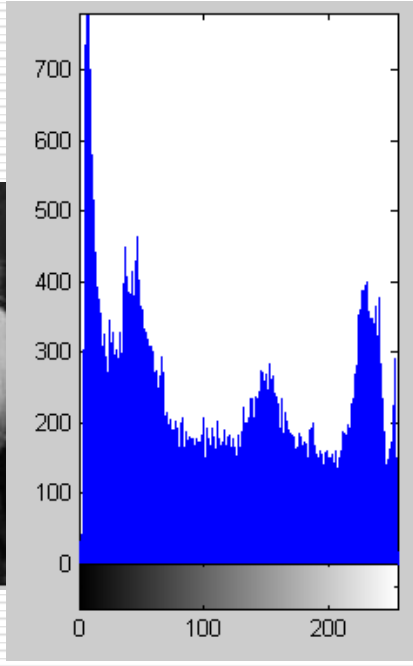
I为灰度图像，n为指定的灰度级数目，默认值为256。

如要计算和显示灰度图像dog.bmp的灰度直方图：

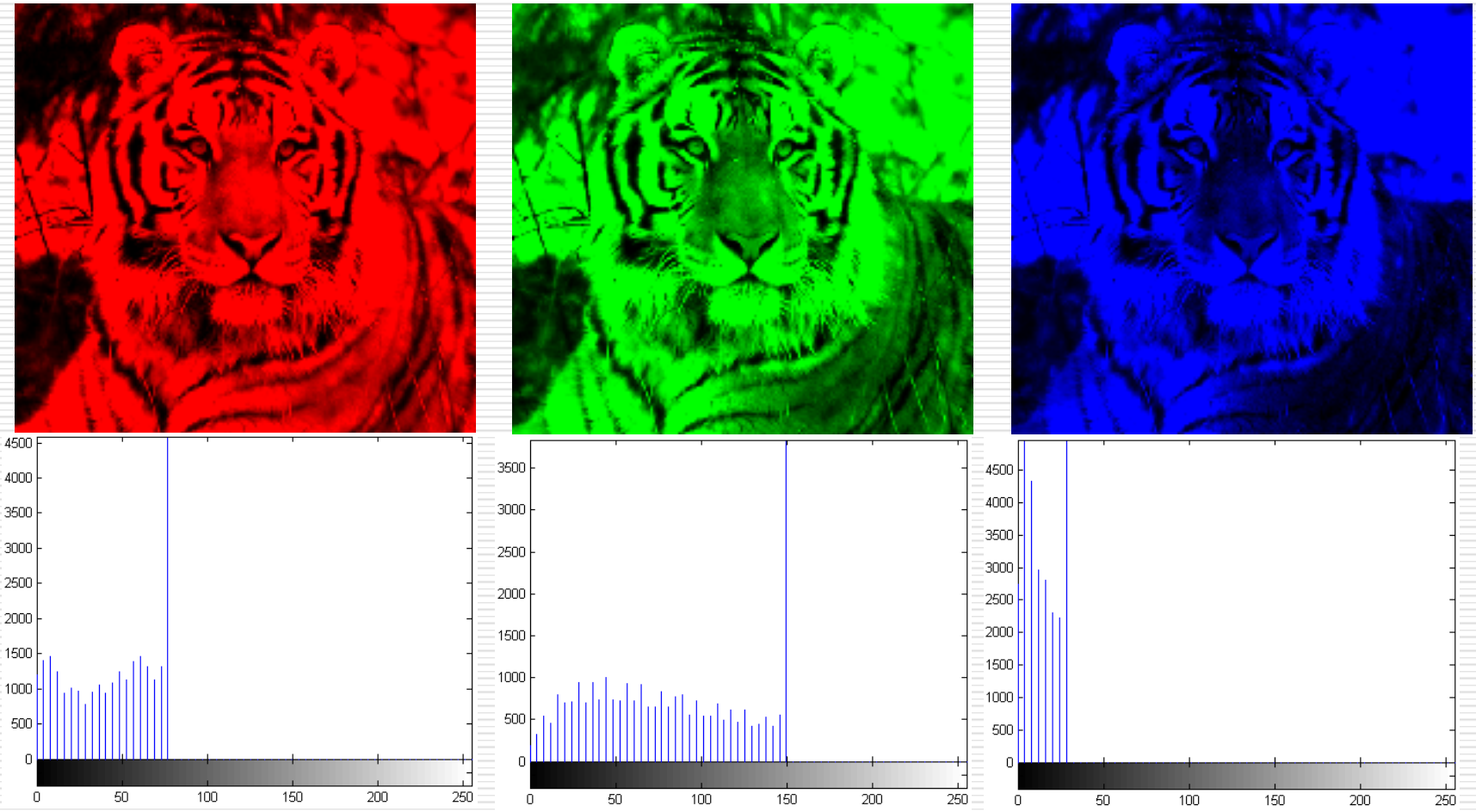
I=imread('dog.bmp')

subplot(1,2,1); imshow(I)

subplot(1,2,2); imhist(I)



彩色图像直方图



三个颜色分量的灰度直方图

图像直方图一般能从整体上描述一幅图像，如原图像曝光不足或曝光过度，也可以看出动态范围选择的合适与否。我们可以对图像的直方图作一些修改，可以达到改善图像的目的。

例：Matlab中直方图灰度变换

使用MATLAB图像处理工具箱中提供的imadjust灰度变换函数，可以实现图像的直方图调节。

该函数的一般调用格式：

J=imadjust(I,[low_in, high_in], [low_out, high_out])

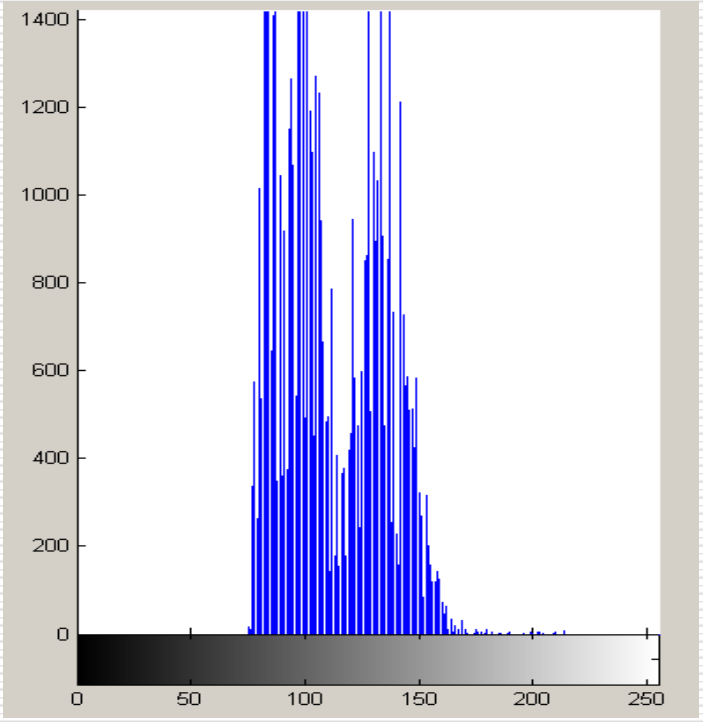
low_in和high_in参数

分别指定输入图像 I 需要映射的灰度范围；

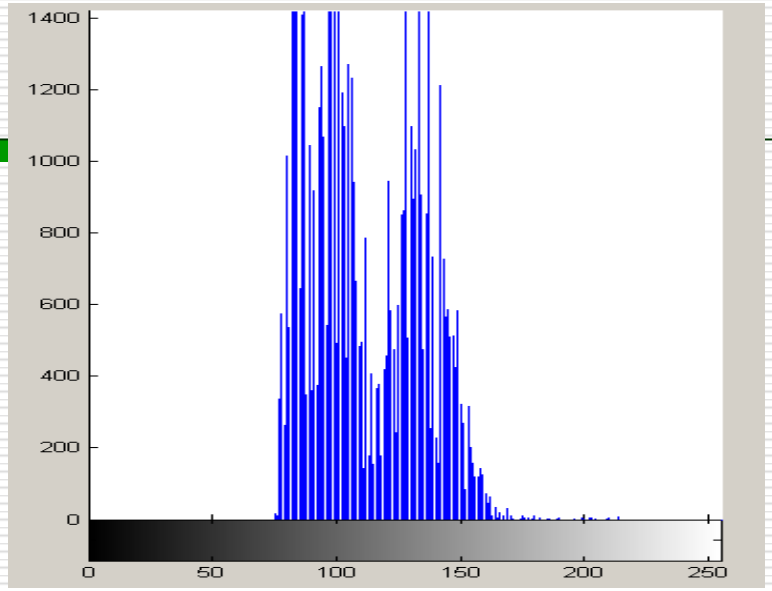
low_out和high_out参数

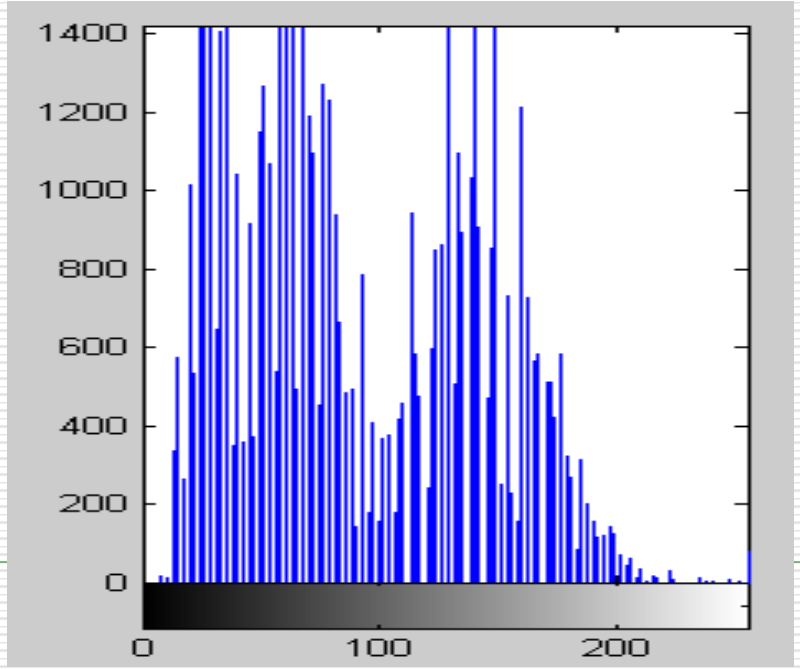
分别指定输出图像 J 所在的灰度范围。

下图图像的灰度值主要分布在80~220之间，现利用imadjust将它的灰度值重新进行映射，使之填满整个灰度值允许范围[0，255]。



```
I=imread('boy.bmp')
subplot(2,2,1), imshow(I)
subplot(2,2,2), imhist(I)          %显示原图直方图
J = imadjust(I,[0.28,0.82],[0 1])
% [0.28,0.82]这里指定需要映射的灰度值范围，[0 1]指定希望映射到的灰度值范围，因此，
% 这里输入图像中的灰度值 0.28 将被映射为输出图像中的 0，0.82 将被映射为输出图像中
% 的 1，这时直方图将填满整个灰度范围。
subplot(2,2,3), imshow(J)
subplot(2,2,4), imhist(J)
```

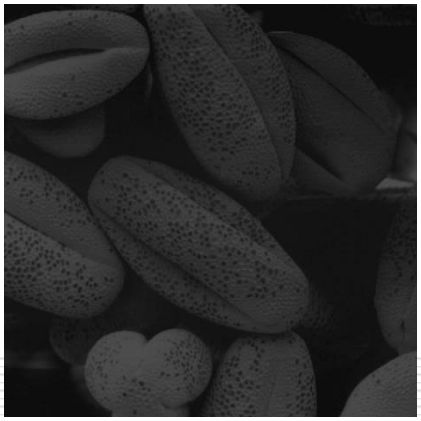

1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

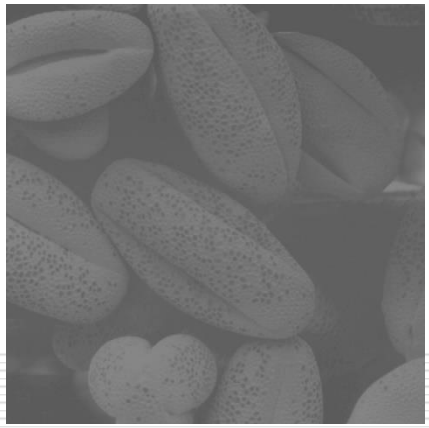
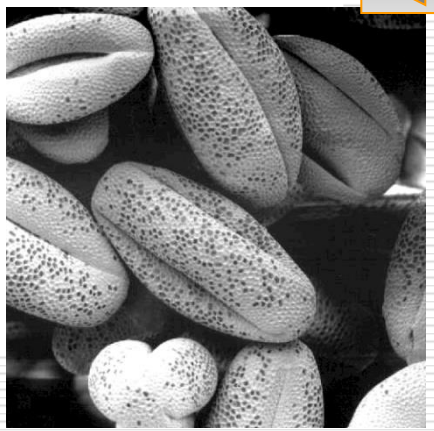

1400
1200
1000
800
600
400
200
0
0
100
200

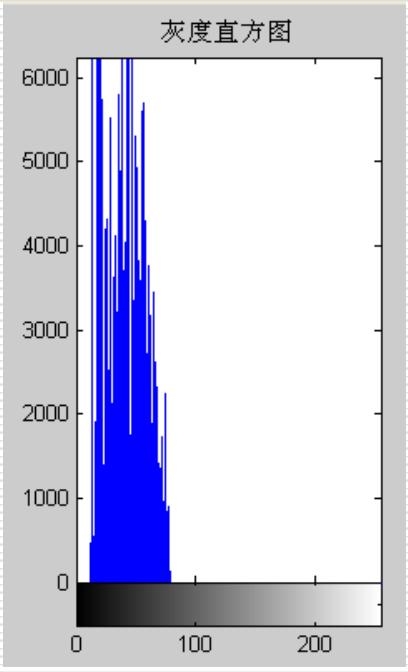
2. 直方图修正法

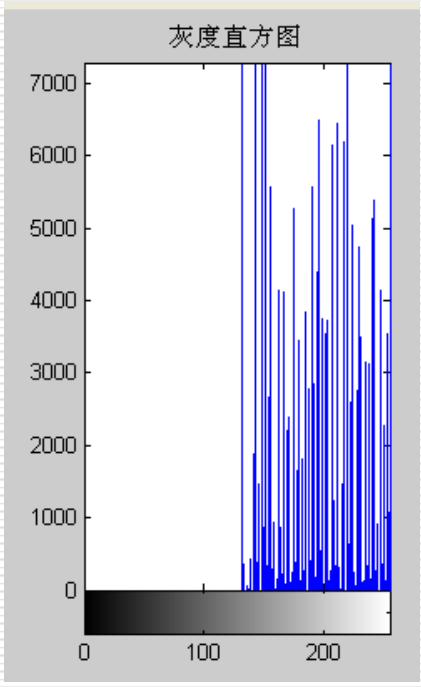
灰度直方图反映了数字图像中每一灰度级与其出现像素频率间的统计特性。它能描述图像的概貌，如图像的灰度范围、每个灰度级出现的频率、灰度级的分布等，为对图像进一步处理提供了重要依据。

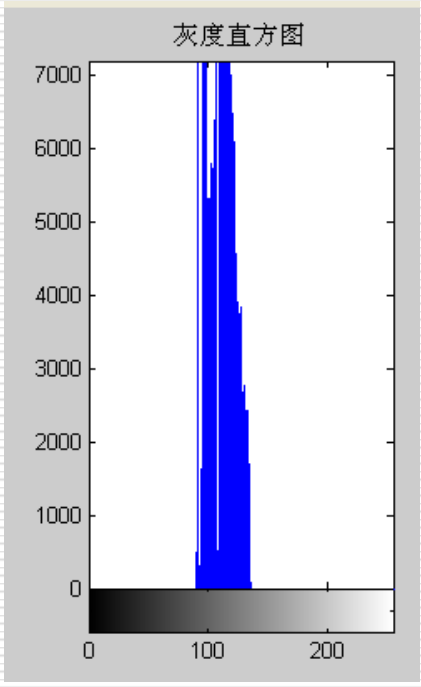
大多数自然图像由于其灰度分布集中在较窄的区间，引起图像细节不够清晰。

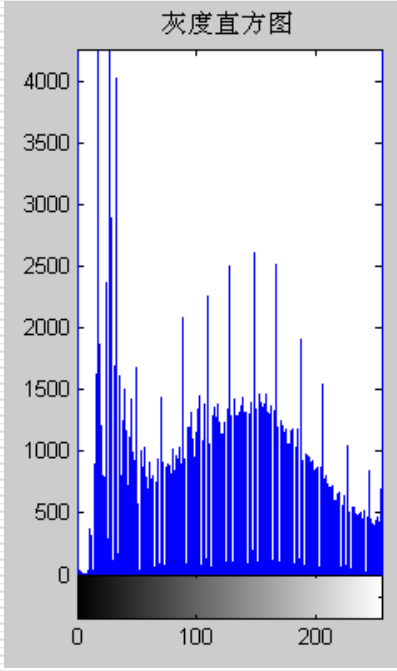










花粉图像的暗、亮、低对比度和高对比度的四种基本图像类型及其相应的直方图

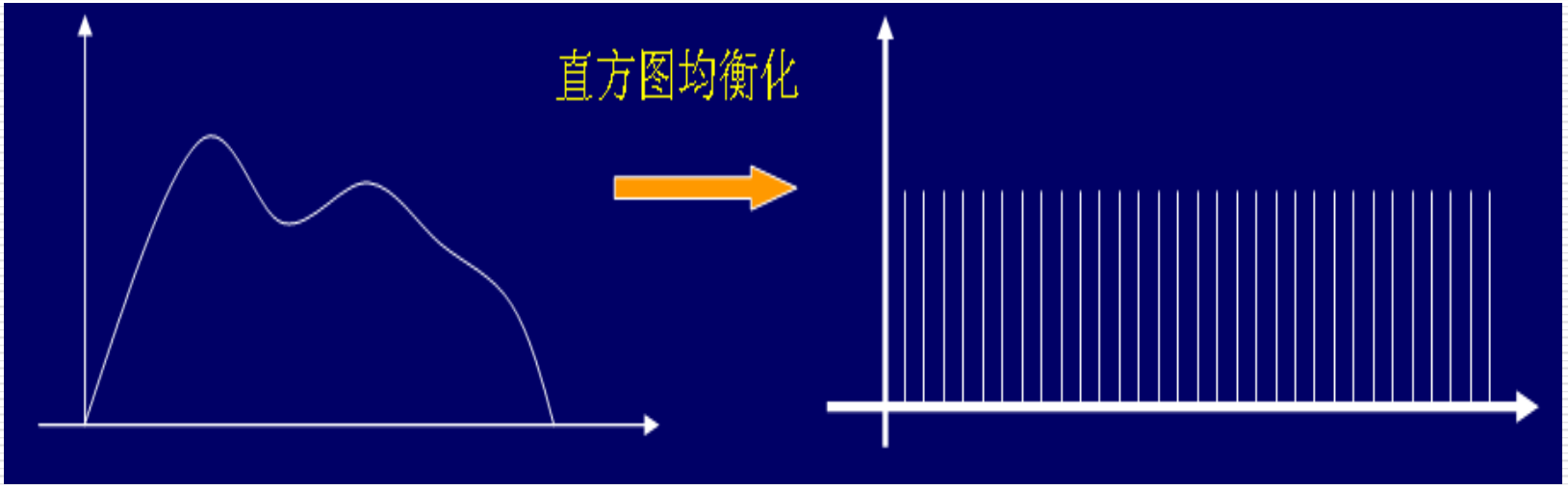
可以直观地看到，若一幅图像其像素占有全部可能的灰度级并且分布均匀，则这样的图像具有高对比度和多变的灰度色调。

直方图修正技术可使图像的灰度间距拉开或使灰度分布均匀，从而增大反差，使图像细节清晰，达到增强的目的。

直方图修正法包括直方图均衡化及直方图规定化两类。

（1） 直方图均衡化

直方图均衡化是将原图像通过某种变换，使原图像的灰度直方图修正为均匀分布的直方图的一种方法。



直方图均衡化原理：

一幅给定图像f(x, y)的灰度级分布在0≤r≤1范围内，可以对[0, 1]区间内的任一个r值进行如下变换：

$$s=T(r)$$

通过上述变换，每个原始图像的像素灰度值r都对应产生一个s值。变换函数T(r)应满足下列条件：

（1） 在0≤r≤1区间内，T(r)值单调增加；

（2） 对于0≤r≤1，有0≤T(r)≤1

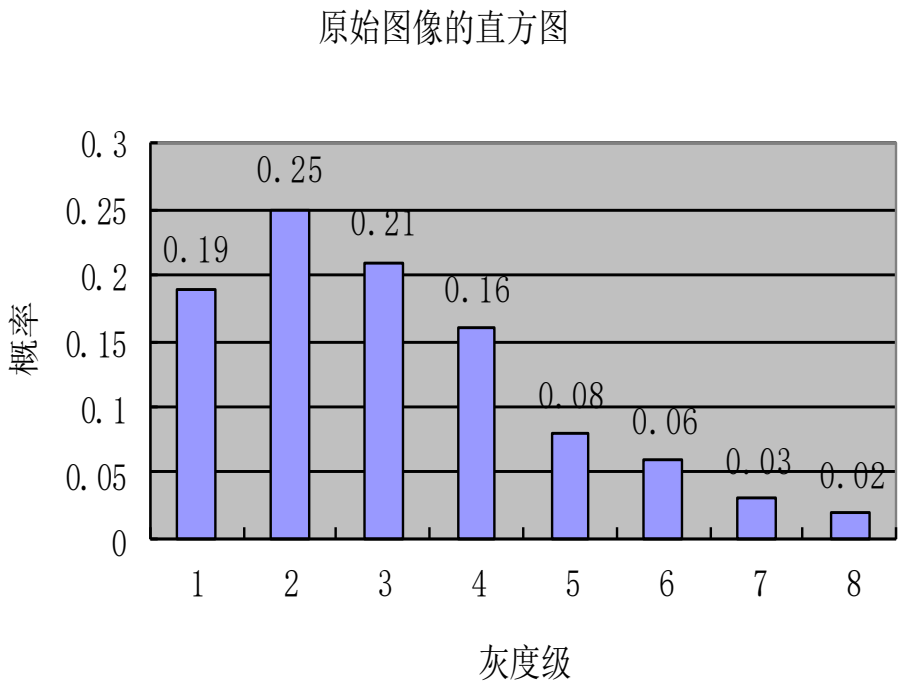
可以证明满足上述两个条件并能将r的原始分布转换为T(r)中的均匀分布的函数关系可由图像f(x,y)的累积直方图得到。

■直方图均衡化处理技术是用累积分布函数作变换函数的直方图修正法。

实例

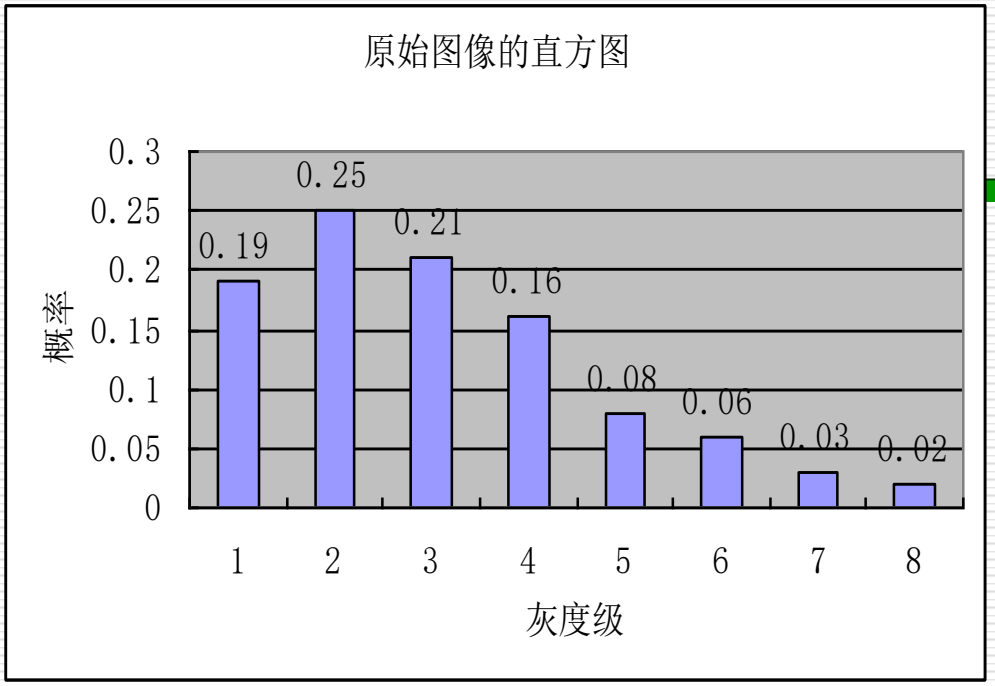
假设一幅64×64，有8个灰度级的图像，其概率分布见表，试进行直方图均衡化处理。

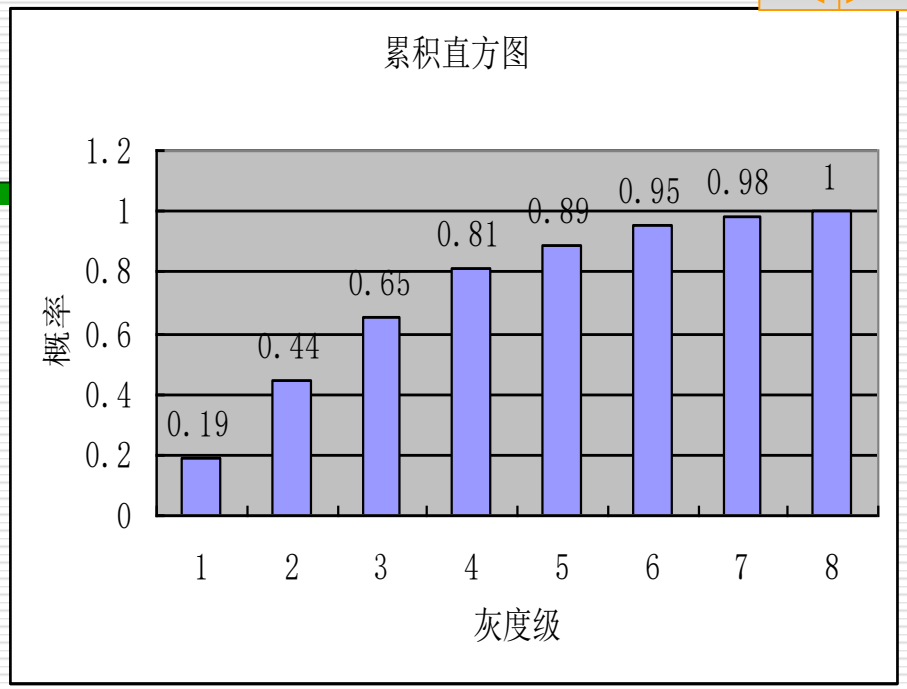
r_k	n_k	$P_r(r_k)=n_k/n$
$r_0=0$	790	0.19
$r_1=1/7$	1023	0.25
$r_2=2/7$	850	0.21
$r_3=3/7$	656	0.16
$r_4=4/7$	329	0.08
$r_5=5/7$	245	0.06
$r_6=6/7$	122	0.03
$r_7=1$	81	0.02

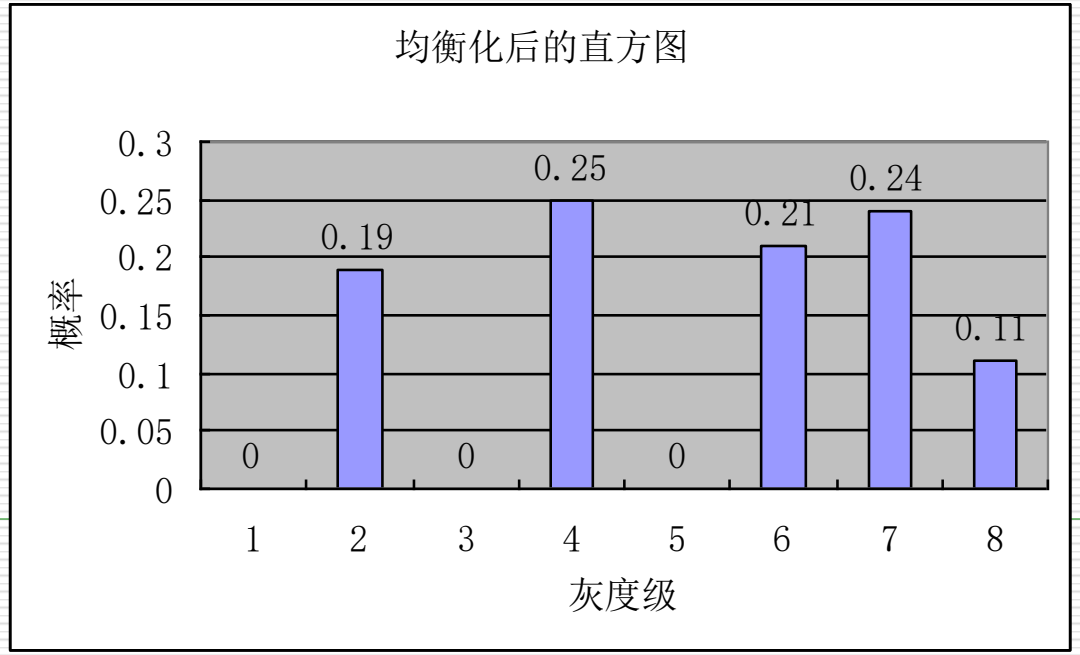


直方图均衡化处理过程

序号	运算	步骤和结果							
1	列出原始图像灰度级 r_k $k=0, 1..7$	0/7	1/7	2/7	3/7	4/7	5/7	6/7	7/7
2	统计原始直方图各灰度级象素 n_k	790	1023	850	656	329	245	122	81
3	计算原始直方图 p_i	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
4	计算累积直方图 $S_k=\sum_{i=0}^{k} Pi$	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
5	量化级	0/7= 0.00	1/7= 0.14	2/7= 0.29	3/7= 0.43	4/7= 0.57	5/7= 0.71	6/7= 0.86	7/7= 1.00
6	确定映射对应关系 r_k->S_k	0→1	1→3	2→5	3、4→6		5、6、7→7		
7	统计新直方图各灰度级象素 n_k		790		1023		850	985	448
8	计算均衡化直方图		0.19		0.25		0.21	0.24	0.11


原始图像的直方图
概率
0.3
0.25
0.2
0.15
0.1
0.05
0
0.19
0.25
0.21
0.16
0.08
0.06
0.03
0.02
1
2
3
4
5
6
7
8
灰度级


累积直方图
概率
1.2
1
0.8
0.6
0.4
0.2
0
0.19
0.44
0.65
0.81
0.89
0.95
0.98
1
1
2
3
4
5
6
7
8
灰度级


均衡化后的直方图
概率
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.19
0
0.25
0
0.21
0.24
0.11
1
2
3
4
5
6
7
8
灰度级

由上面的例子可见，利用累积分布函数作为灰度变换函数，经变换后得到的新灰度的直方图虽然不很平坦，但毕竟比原始图像的直方图平坦得多，而且其动态范围也大大地扩展了。因此这种方法对于对比度较弱的图像进行处理是很有效的。

因为直方图是近似的概率密度函数，所以用离散灰度级作变换时很少能得到完全平坦的结果。另外，从上例中可以看出变换后的灰度级减少了，这种现象叫做“简并”现象，由于简并现象的存在，处理后的灰度级总是要减少的，这是像素灰度有限的必然结果。由于上述原因，数字图像的直方图均衡只是近似的。

作业:

1. 一幅64×64大小的图像，灰度级有8个，各灰度级出现的频数如下表所示，用直方图均衡化方法对该图像进行增强处理。

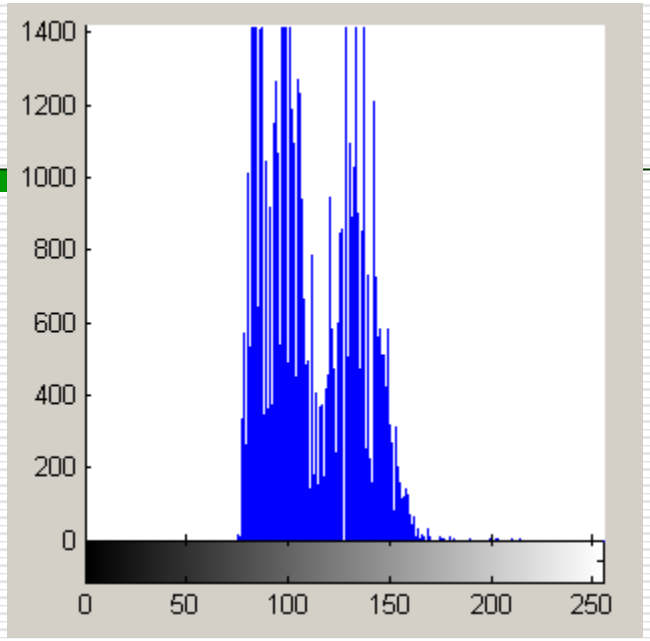
r_k	n_k	n_k/n
0	750	0.18
1	982	0.24
2	568	0.14
3	515	0.13
4	215	0.05
5	647	0.16
6	273	0.07
7	136	0.03

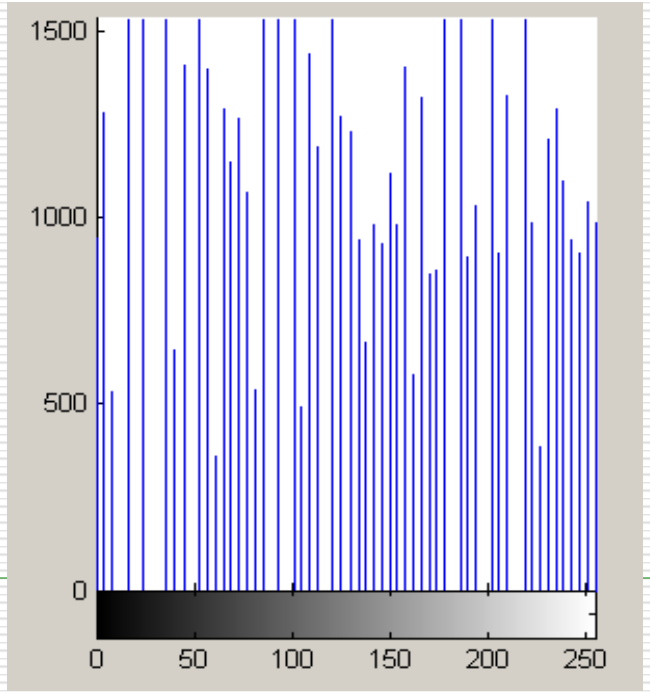
在MATLAB中，可以调用函数histeq自动完成图像的直方图均衡化。对于灰度图像，histeq函数的调用格式如下：

J=histeq(I,n)

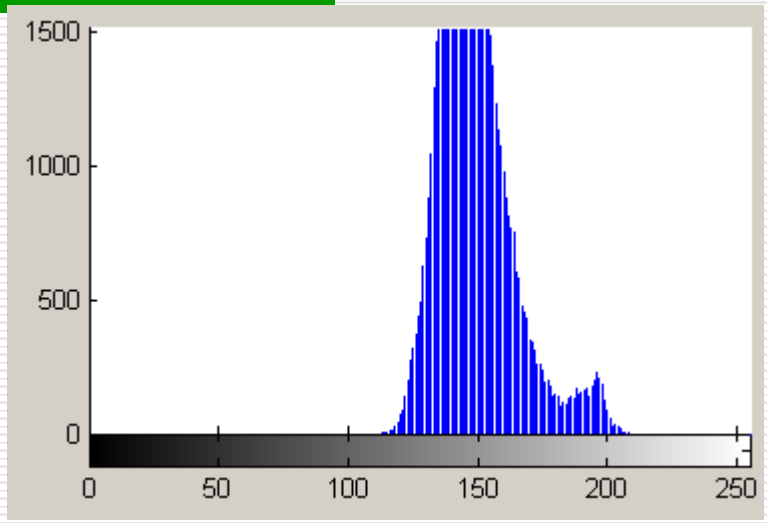
其中，n表示输出图像的灰度级数目，是一个可选参数，缺省值为64。

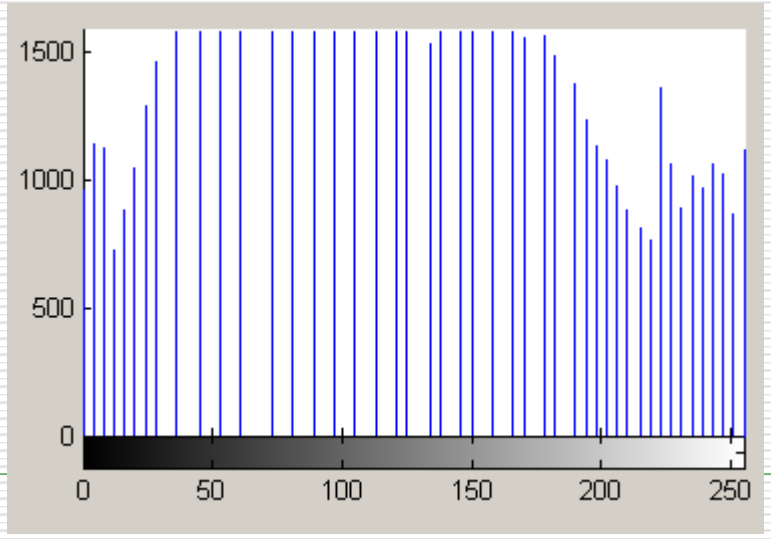
```
%例　直方图均衡化前后的图像灰度分布
I=imread('boy.bmp')
J=histeq(I)      %对图像I进行直方图均衡化
figure, imshow(I)
figure, imshow(J)
figure, imhist(I, 64)   %显示原图像的灰度直方图
figure, imhist(J, 64)   %显示均衡化后的图像的灰度直方图
```


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


1500
1000
500
0
0
50
100
150
200
250

直方图均衡化示例


1500
1000
500
0
0
50
100
150
200
250


1500
1000
500
0
0
50
100
150
200
250

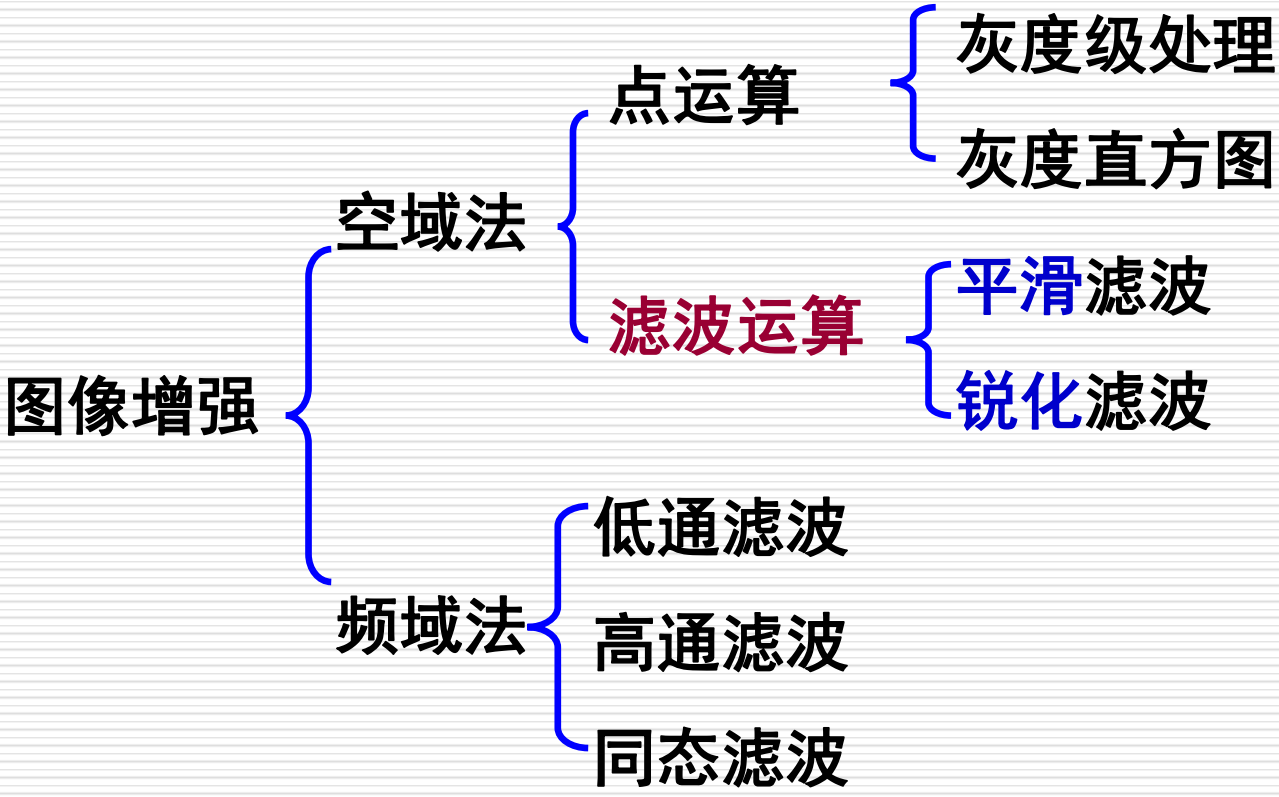
■ 经过直方图均衡化，可以看出，图像的细节更加清楚了，而从图像的直方图可以看出，在直方图调节之前，图像的灰度值范围比较窄，经过直方图调整，各灰度等级的比例更加平衡。

■ 但是由于直方图均衡没有考虑图像的内容，简单地将图像进行直方图均衡，使得图像看起来亮度过高。也就是说直方图均衡的方法不够灵活，因此人们又提出了别的增强方法。

(2) 直方图规定化

在某些情况下，并不一定需要具有均匀直方图的图像，有时需要具有特定形状的直方图的图像，以便能够增强图像中某些灰度级。直方图规定化就是针对上述思想提出来的。

直方图规定化是使原图像灰度直方图变成规定形状的直方图而对图像作修正的增强方法。


图像增强
空域法
点运算
灰度级处理
灰度直方图
滤波运算
平滑滤波
锐化滤波
频域法
低通滤波
高通滤波
同态滤波

4.3 基于空域滤波的增强

4.3.1 背景和原理

空域滤波是在图像空间借助模板进行邻域操作完成的。

空域滤波器根据功能主要分成平滑滤波和锐化滤波。

平滑滤波可用低通滤波实现，其目的：

a. 消除噪声

b. 实现模糊　　去除太小的细节或将目

锐化滤波可用高通滤波实现。

其目的是为了增强被模糊的细节。

在增强中用到的空间滤波器主要有低通(平滑)滤波器、高通(锐化)滤波器和带通滤波器：

(1) 低通(平滑)滤波器

当信号通过该滤波器时，频域中信号的高频部分被衰减或去掉，信号的低频部分则可以无衰减的通过。

因为图像的高频部分刻画了图像的边缘或其它尖锐细节，滤波器将这些部分滤去可使图像平滑。

(2) 高通(锐化)滤波器　衰减或去除信号的低频部分。

因为低频分量对应图像中灰度值缓慢变化的区域，因而与图像的整体特征，如整体对比度和平均灰度值有关。图像通过高通滤波器，抑制了图像的这些缓慢变化特征而突出了图像的边缘及其它细节信息，使图像锐化。

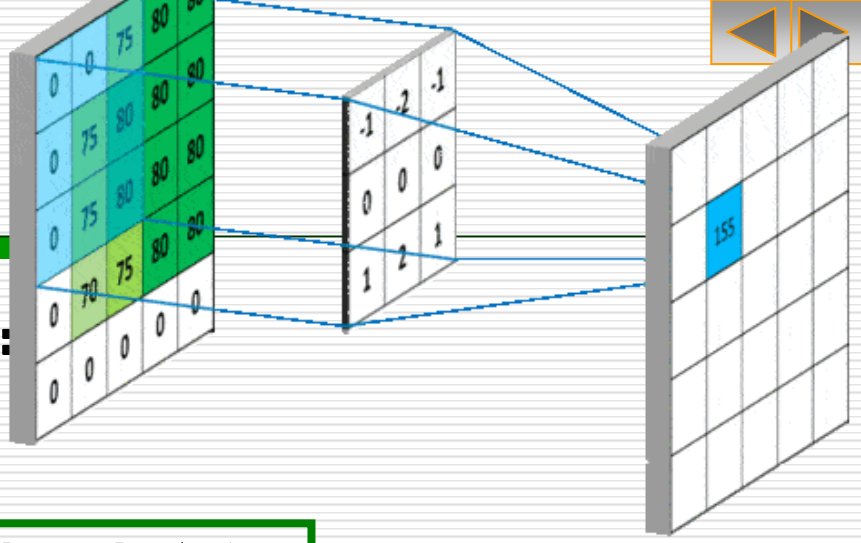
(3) 带通滤波器

能够减弱或消除某一特定频率区域内的信号分量。

不管使用何种滤波器，但在空间域实现图像滤波都是利用模板卷积运算。

模板：如下图有点类似于矩阵的式子，大小有3*3、5*5、9*9等。带星号的数据表示该元素为中心元素，即这个元素是将要处理的元素。

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1^{*} & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



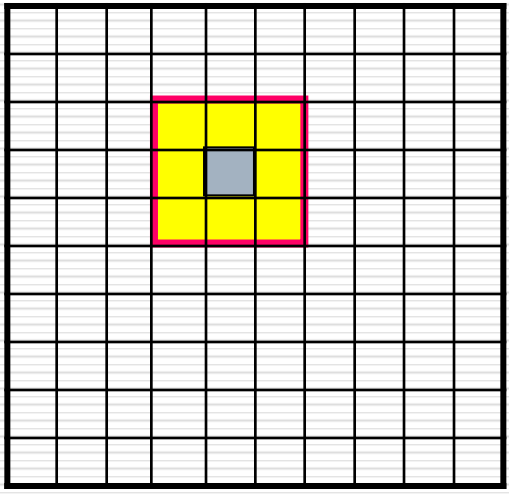
利用模板卷积运算的主要步骤为：

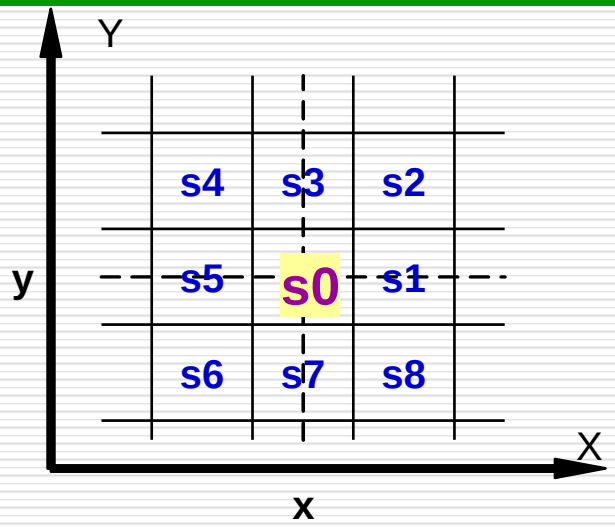
1、将模板在图像中漫游，并将模板中心与图中某个像素位置重合；

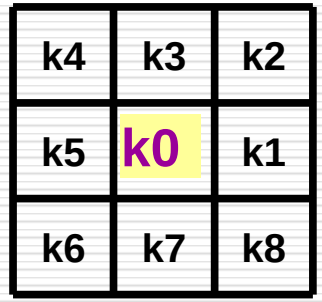
2、将模板上系数与模板下对应像素相乘；

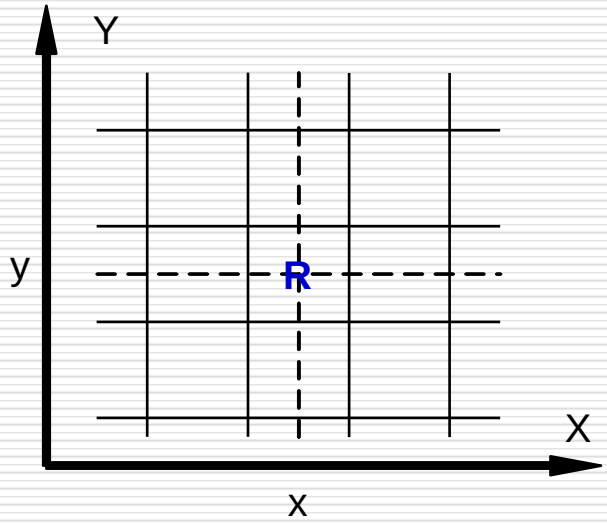
3、将所有乘积相加；

4、将和（模板的输出响应）赋给图中对应模板中心位置的像素。









a. 图像一部分　　b. 3×3模板　　c.

将k0的位置于图中灰度值为s0的象素重合（即将模板中心放在图中(x, y)位置），模板的输出响应R为：

$$R = k_0 s_0 + k_1 s_1 + \cdots + k_8 s_8$$

将R赋给增强图，作为在(x, y)位置的灰度值(图c)

习题：

已知一幅图像f(x, y)= $\begin{bmatrix} 2 & 7 & 3 \\ 5 & 8 & 1 \\ 9 & 2 & 8 \end{bmatrix}$，现用模板h(x, y)= $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ 对其进行卷积操作，求输出图像（输出图像与原图像尺寸一样，当模板超出图像边界时，利用常数0来填充图像边界）。

习题：

已知一幅图像f(x, y)= $\begin{bmatrix} 2 & 7 & 3 \\ 5 & 8 & 1 \\ 9 & 2 & 8 \end{bmatrix}$，现用模板h(x, y)= $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ 对其进行卷积操作，求输出图像（输出图像与原图像尺寸一样，当模板超出图像边界时，利用常数0来填充图像边界）。

	2	7	3	
	5	8	1	
	9	2	8	

0	-1	0
-1	4	-1
0	-1	0

在模板卷积运算中，需注意：

1.图像边界问题

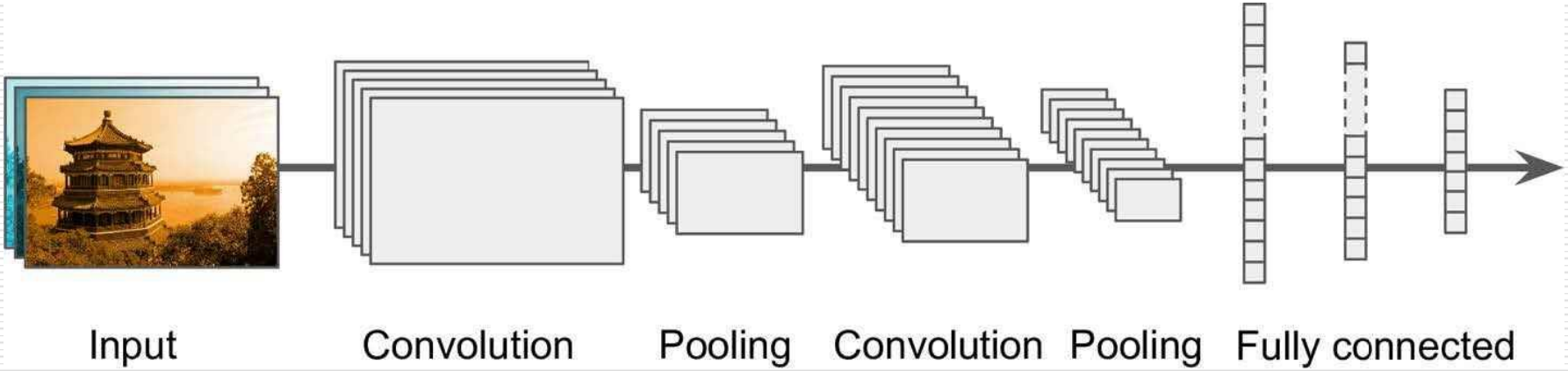
当模板原点移至图像边界时，部分模板系数可能在原图像中找不到相对应的像素。处理方法：

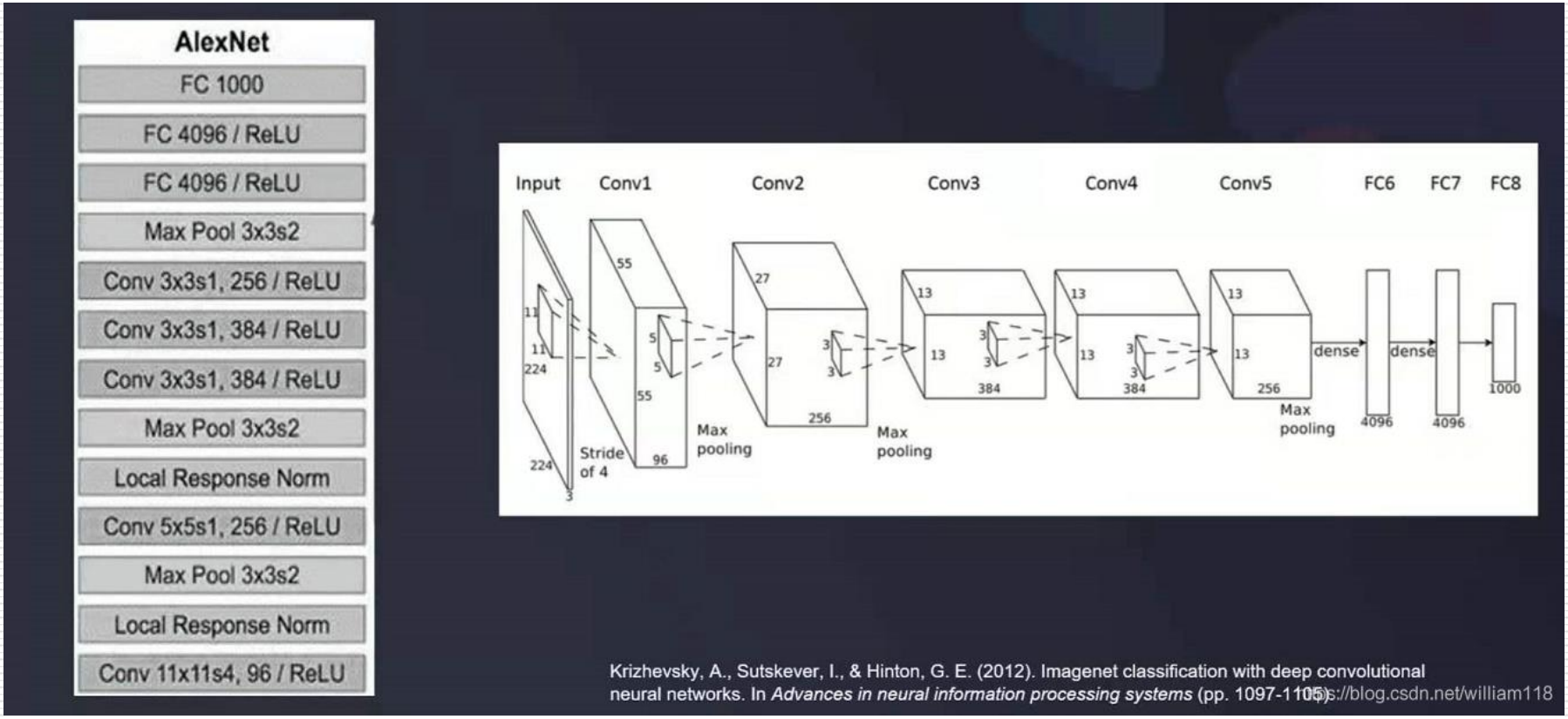
(1).当模板超出图像边界时不作处理。

(2).扩充图像，可以复制原图像边界像素或利用常数来扩充图像边界，使得卷积在图像边界也可以计算。

2.计算结果可能超出灰度范围

例如，对于8位灰度图像，当计算结果超出[0,255]时，可以将其置为0或255。


Input
Convolution
Pooling
Convolution
Pooling
Fully connected


AlexNet
FC 1000
FC 4096 / ReLU
FC 4096 / ReLU
Max Pool 3x3s2
Conv 3x3s1, 256 / ReLU
Conv 3x3s1, 384 / ReLU
Conv 3x3s1, 384 / ReLU
Max Pool 3x3s2
Local Response Norm
Conv 5x5s1, 256 / ReLU
Max Pool 3x3s2
Local Response Norm
Conv 11x11s4, 96 / ReLU
Input
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
Stride of 4
Max pooling
Max pooling
Max pooling
dense
dense
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
https://blog.csdn.net/william118

4.3.2 图像平滑滤波

1. 邻域平均法

邻域平均法是简单的空域处理方法。这种方法的基本思想是用几个像素灰度的平均值来代替每个像素的灰度。

假定有一幅N*N个像素的图像f(x, y)，平滑处理后得到一幅图像g(x, y)。g(x, y)由下式决定

$$g(x,y)=\frac{1}{M}\sum_{(m,n)\in s} f(m,n)$$

$$x,\ y=0,\ 1,\ 2,\cdots\cdots,N-1$$

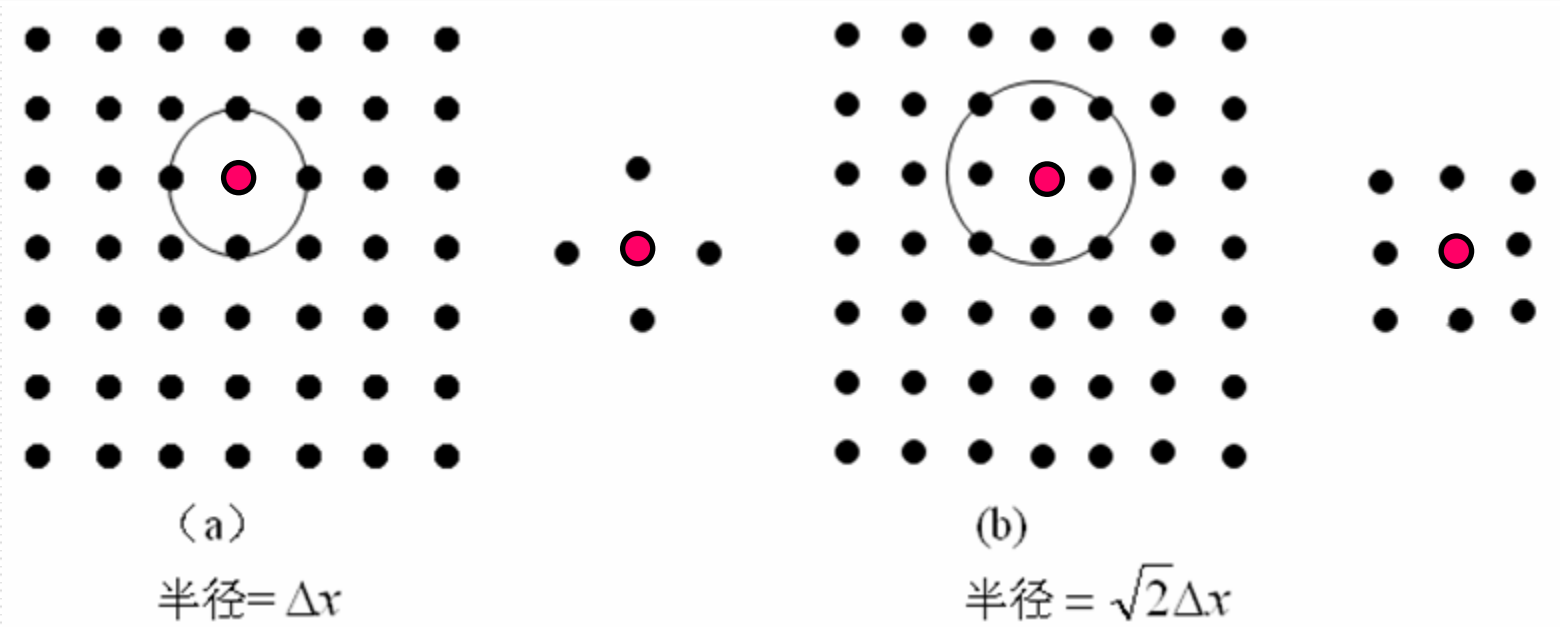
M为邻域内所包含的像素总数，
S为事先确定的邻域
（如该邻域不包括（x, y)点，则为去心邻域)。

上式说明，平滑后的图像g(x,y)中的每个像素的灰度值均由包含在(x,y)点的预定邻域中的f(x,y)的几个像素的灰度值的平均值来决定。

在邻域平均法中，如何选取邻域是一个相对重要的问题，下图给出了两种从图像阵列中选取邻域的方法。

4-邻域

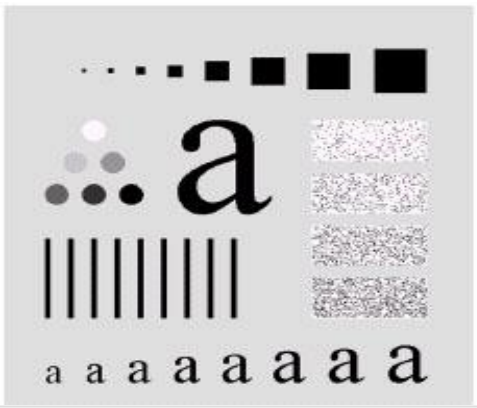
8-邻域



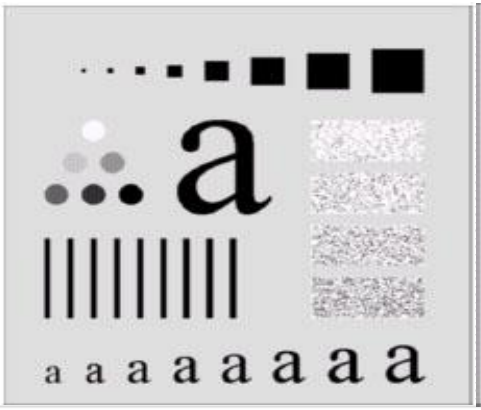
图(a)的方法是一个点的邻域定义为以该点为中心的一个圆的内部或边界上的点的集合。图中像素间的距离为△x，选取△x为半径作圆，那么，点R 的灰度值就是圆周上4个像素灰度值的平均值。

图(b)是选 √2 △x为半径的情况下构成的点R的邻域，选择在圆的边界上的点和在圆内的点为S的集合。

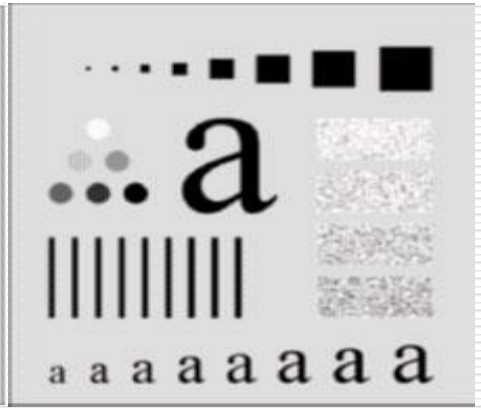
邻域平均法实验结果



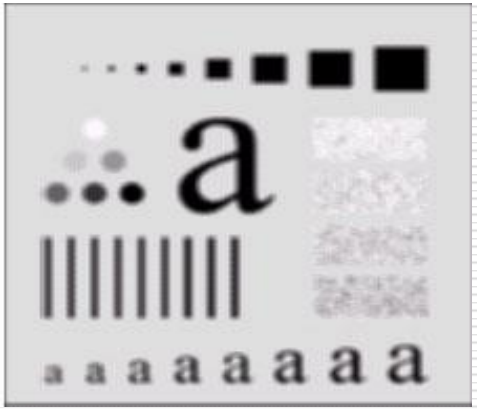
(a) 原始图像



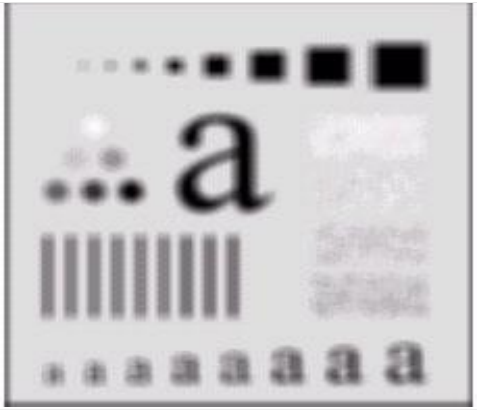
(b)M=3



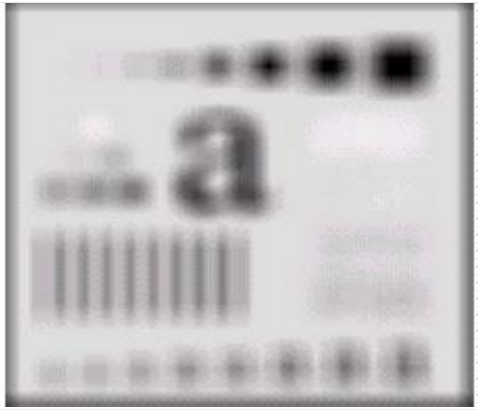
(c)M=5



(d) M=9



(e)M=15



(f)M=35

不同尺度下的邻域平均法

优点：算法简单，计算速度快

缺点：

图像易模糊。对抑制噪声有效，但随着邻域的加大，图像的模糊程度也愈加严重。

可能的解决办法：

采用**阈值法**减少由于邻域平均所产生的模糊效应。

$$g(x,y)=\begin{cases}\dfrac{1}{M}\displaystyle\sum_{(m,n)\in s} f(m,n) & \text{若}\left|f(x,y)-\dfrac{1}{M}\displaystyle\sum_{(m,n)\in s} f(m,n)\right|>T \\ f(x,y) & \text{其他}\end{cases}$$

式中T就是规定的非负**阈值**。当一些点和它的邻域内的点的灰度的平均值的差不超过规定的阈值T时，就仍然保留其原灰度值不变，如果大于阈值T时就用它们的平均值来代替该点的灰度值。这样就可以大大减少模糊的程度。

□ 2　中值滤波

邻域平滑滤波在消除噪声的同时，会将图像中的一些细节模糊，如果既要消除噪声，又要保持图像的细节可以使用中值滤波器。

中值滤波是对一个滑动窗口内的诸像素灰度值排序，用中值代替窗口中心像素的原来灰度值。

例：　采用1×3窗口进行中值滤波

原图像为：　2 2 6 2 1 2 4 4 4 2 4

处理后为：　2 2 2 2 2 2 4 4 4 4 4

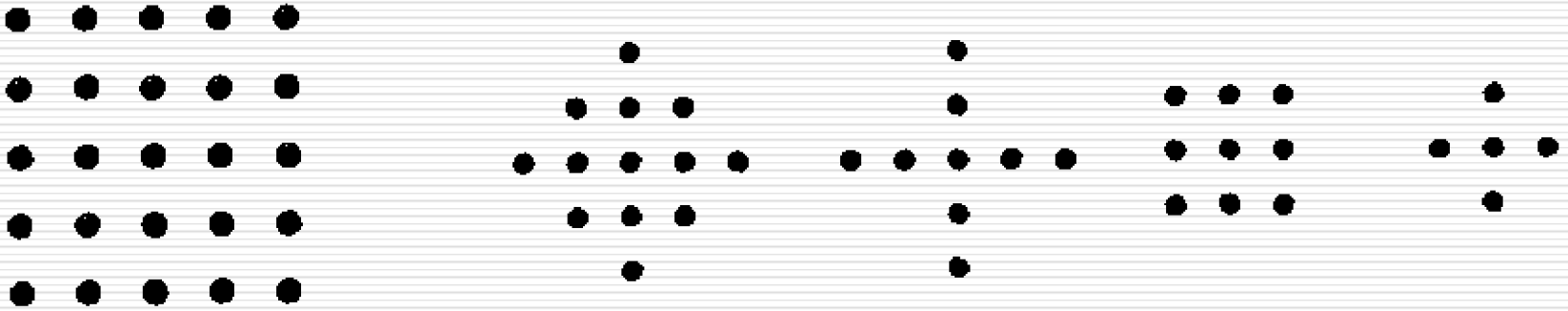
由此可以看出，中值滤波器的主要功能就是让与周围像素灰度值的差比较大的像素改取与周围的像素值接近的值，从而可以消除孤立的噪声点。

中值滤波滤波原理与均值滤波方法类似，二者不同之处在于：

中值滤波器的输出像素是由邻域像素的中间值而不是平均值决定。由于它不是简单的取均值，所以产生的模糊比较少。

对于二维的数字图像，中值滤波就是用一个活动窗口沿图像移动，窗口中心位置的像素灰度用窗口内所有像素灰度的中值来代替。

考虑到一般图像在二维方向上均具有相关性，因此，活动窗口一般选为二维窗口(如3×3、5×5或7×7等)。窗口的形状常用的有方形、十字形、圆形或X字形等。

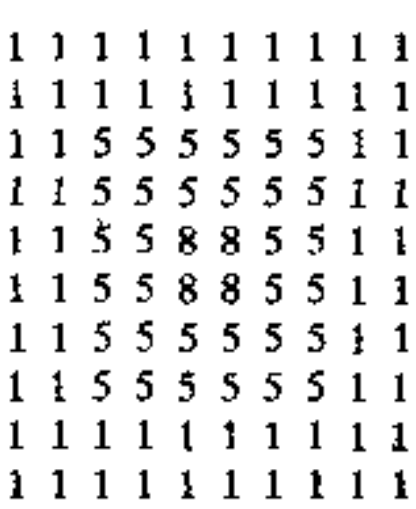


5×5方形窗口　5×5圆形窗口　5×5十字形窗口　3×3方形窗口　3×3十字形窗口

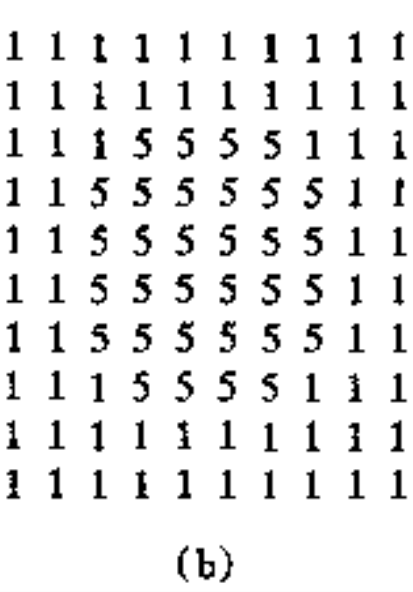
二维中值滤波常用窗口

使用二维中值滤波时需要注意以下问题：

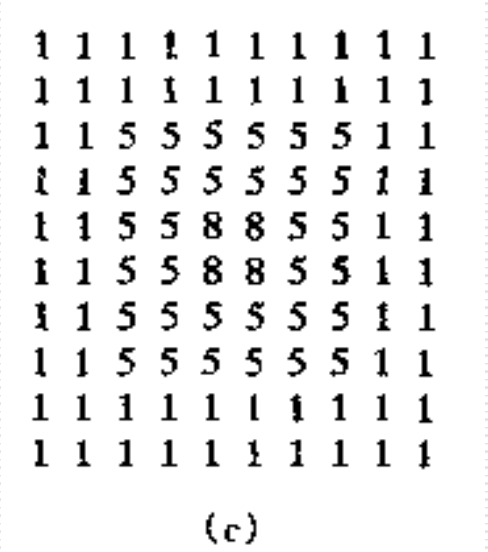
(1) 具有丰富的尖顶角几何结构的图像，一般采用十字形窗口，窗口大小最好不要超过图像中最小有效物体的尺寸。否则，中值滤波后会丢失细小的几何特征。



原图像



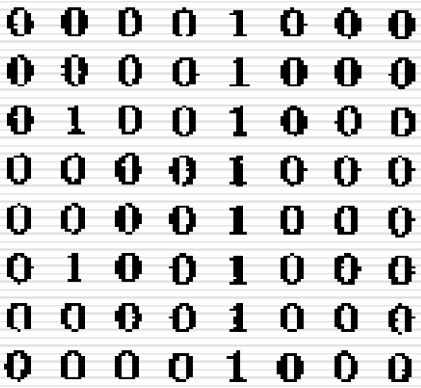
用3×3方形窗口作中值滤波后



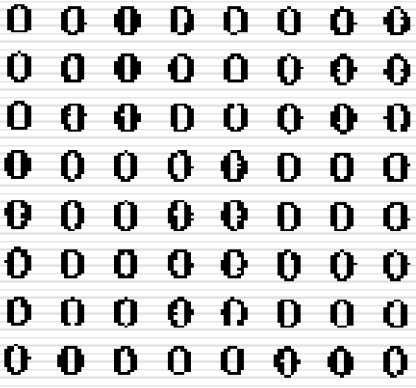
用3×3十字形窗口作中值滤波后

b图除尖角几何结构损失以外，中心部分的几何结构信息也丢失。

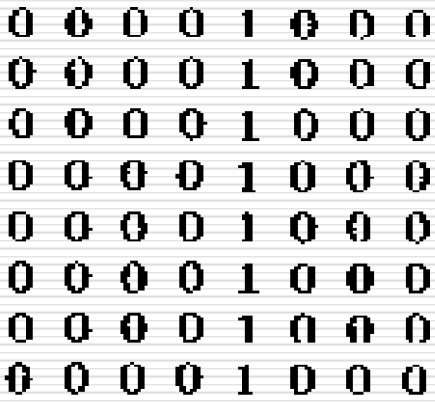
（2）在进行二维中值滤波时，需特别注意保持图像中有效的细线状及尖顶角物体，如果图像中有较多的点、线和尖顶角细节结构，最好不要使用中值滤波法做图像平滑处理。



A.原图像



B. 用3×3方形窗口作中值滤波后



C.用3×3十字形窗口作中值滤波后

B图点和线的细节均丢失了；C图虽保留了线状细节，但丢失了点状细节。

设图像如下：

分别使用3×3的模板对其进行均值滤波处理与中值滤波处理，写出处理结果（四舍五入取整数），当模板超出图像边界时扩充图像，利用常数0来填充扩充的图像边界，输出图像与原图像大小一样。

1	2	2	2	3
1	15	1	2	2
2	1	2	0	3
0	2	2	3	1
3	2	0	2	2

Matlab图像平滑滤波函数及应用

图像增强操作主要是针对图像的各种噪声而言的，为了说明图像处理中的滤波方法和用途，需要模拟数字图像的各种噪声来分析滤波效果。

实际获得的图像一般都因受到某种干扰而含有噪声。引起噪声的原因有敏感元器件的内部噪声、照相底片上感光材料的颗粒、传输通道的干扰及量化噪声等。噪声产生的原因决定了噪声的分布特性及它和图像信号的关系。

imnoise函数

Matlab图像处理工具箱提供了模拟噪声生成的函数imnoise，它可以对图像添加一些典型的噪声。

imnoise的语法格式为：

J=imnoise(I, type)

J=imnoise(I, type, parameters)

对原图像 I 添加典型噪声得到含噪声图像 J，

参数type和parameters用于确定噪声的类型和相应的参数。

参数Type对应的噪声类型如下：

‘gaussian’	高斯噪声
‘localvar’	0均值高斯白噪声
‘poisson’	泊松噪声
‘salt & pepper’	椒盐噪声
'speckle'	乘性噪声

类型	参数	说明
gaussian	m,v	均值为m,方差为v的高斯噪声
localvar	v	均值为0,方差为v的高斯噪声
possion	无	泊松噪声
Salt & pepper	d	密度为d的椒盐噪声
speckle	v	均值为0,方差为v的均匀分布的随机噪声

下面的命令是对图像cat.bmp分别加入高斯噪声、椒盐噪声和乘性噪声，其结果如图所示：

```
I=imread('cat.bmp');
J1=imnoise(I, 'gaussian', 0, 0.02);
        %添加均值为0，方差为 0.02的高斯噪声
J2=imnoise(I, 'salt & pepper', 0.02); %添加椒盐噪声
J3=imnoise(I, 'speckle', 0.02);
subplot(2,2,1), imshow(I)
subplot(2,2,2), imshow(J1)
subplot(2,2,3), imshow(J2)
subplot(2,2,4), imshow(J3)
```

原图像

添加高斯噪声

添加椒盐噪声

添加乘性噪声

由于**噪声的随机性**，它们对某一像点的影响将使其灰度和邻点的灰度显著不同，因此可以利用这种不同来消除噪声。

fspecial函数

fspecial函数用于创建预定义的滤波算子，其语法格式为：

```
h=fspecial(type)
h=fspecial(type,para)
```

参数type指定算子的类型，para指定相应的参数，具体意义如下：

- type='average'，为均值滤波器，参数为n，代表模板尺寸，默认值为[3 3].
- type='laplacian'，为拉普拉斯算子，参数为alpha,用于控制拉普拉斯算子的形状，取值范围为[0,1]，默认值为0.2。
- type='prewit'，为Prewitt算子，用于边缘增强，无参数。
- type='sobel'，为著名的Sobel算子，用于边缘提取，无参数。

```
>>  h=fspecial('average',3)
```

```
h =

    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
```

```
>> h=fspecial('average',5)
```

```
h =

    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
```

filter2函数

MATLAB图像处理工具箱提供了基于卷积的图像滤波函数filter2，filter2的语法格式为：

B=filter2(h, A)

B=filter2(h, A, shape)

其中B=filter2(h, A)返回图像A经算子h滤波后的结果，参数shape指定滤波的计算范围，即：

shape='full'，作边界补零。(返回图像比原图像大)

shape='same'，返回图像B与输入图像A大小相同。

shape='valid'，不考虑边界补零，只计算有效输出部分。

```
%
img=[2 7 3;5 8 1;9 2 8];
h=[0 -1 0;-1 4 -1;0 -1 0];
```

```
imgnew: 5x5 double =

     0    -2    -7    -3     0
    -2    -4    15     4    -3
    -5     1    17   -15    -1
    -9    29   -17    29    -8
     0    -9    -2    -8     0
```

```
imgnew=filter2(h,img,'full');
imshow(imgnew);
```

```
img=[2 7 3;5 8 1;9 2 8];
h=[0 -1 0;-1 4 -1;0 -1 0];
```

```
imgnew: 3x3 double =

    -4    15     4
     1    17   -15
    29   -17    29
```

```
imgnew=filter2(h,img,'same');
imshow(imgnew);
```

```
img=[2 7 3;5 8 1;9 2 8];
h=[0 -1 0;-1 4 -1;0 -1 0];
```

```
imgnew: 1x1 double =

    17
```

```
imgnew=filter2(h,img,'valid');
imshow(imgnew);
```

例1　均值滤波

对图像加入椒盐噪声，并实现均值滤波

```
img=imread('dog.bmp');
imgnew=imnoise(img,'salt & pepper',0.02); %添加椒盐噪声
H1=fspecial('average',3);     %H1=ones(3,3)/9;
M1=filter2(H1,imgnew);
H2=fspecial('average',5);
M2=filter2(H2,imgnew);
H3=fspecial('average',7);
M3=filter2(H3,imgnew);
figure,
subplot(2,3,1),imshow(img);
subplot(2,3,2),imshow(imgnew);
subplot(2,3,4),imshow(uint8(M1));
subplot(2,3,5),imshow(uint8(M2));
subplot(2,3,6),imshow(uint8(M3));
% h=fspecial('average', n)  创建n*n的均值滤波器
% B=filter2(h,A)    返回图像A经算子h滤波的结果。
```

加入椒盐噪声的图像和均值滤波的图像分别如下图所示：

(1). 原图像

(2). 有噪声的图像

(3). 3*3均值滤波后的结果

(4). 5*5均值滤波后的结果

(5). 7*7均值滤波后的结果

◆经过邻域平均处理后，图像的噪声得到了抑制，但图像变得相对模糊了。

◆邻域平均法的平滑效果与所采用的邻域的半径（模板大小）有关。模板尺寸（半径）越大，则图像的模糊程度越大。此时，消除噪声的效果将增强，但同时所得到的图像将变得更模糊，图像细节的锐化程度逐步减弱。

例2　中值滤波：

B=medfilt2(A)

用3*3的滤波窗口对图像A进行中值滤波

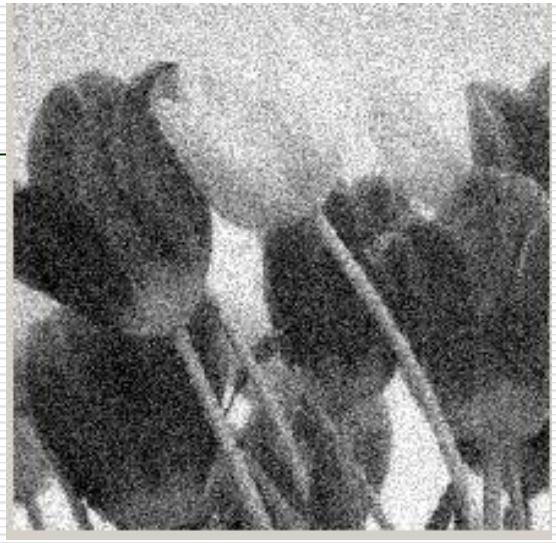
B=medfilt2(A,[m,n]);

用指定大小为m*n的窗口对图像A进行中值滤波

```
%中值滤波处理程序
I=imread('flower1.bmp');
J1=imnoise(I,'salt & pepper',0.02);   %加均值为0,方差为0.02的椒盐噪声
J2=imnoise(I,'gaussian',0.02);        %加均值为0,方差为0.02的高斯噪声
subplot(2,2,1), imshow(J1);
subplot(2,2,2), imshow(J2);
I1=medfilt2(J1,[5,5]);                %对椒盐噪声图像进行5*5方形窗口中值滤波
I2=medfilt2(J2,[5,5]);
subplot(2,2,3), imshow(I1);
subplot(2,2,4), imshow(I2);
```

(1)原图像

(2)有椒盐噪声图像



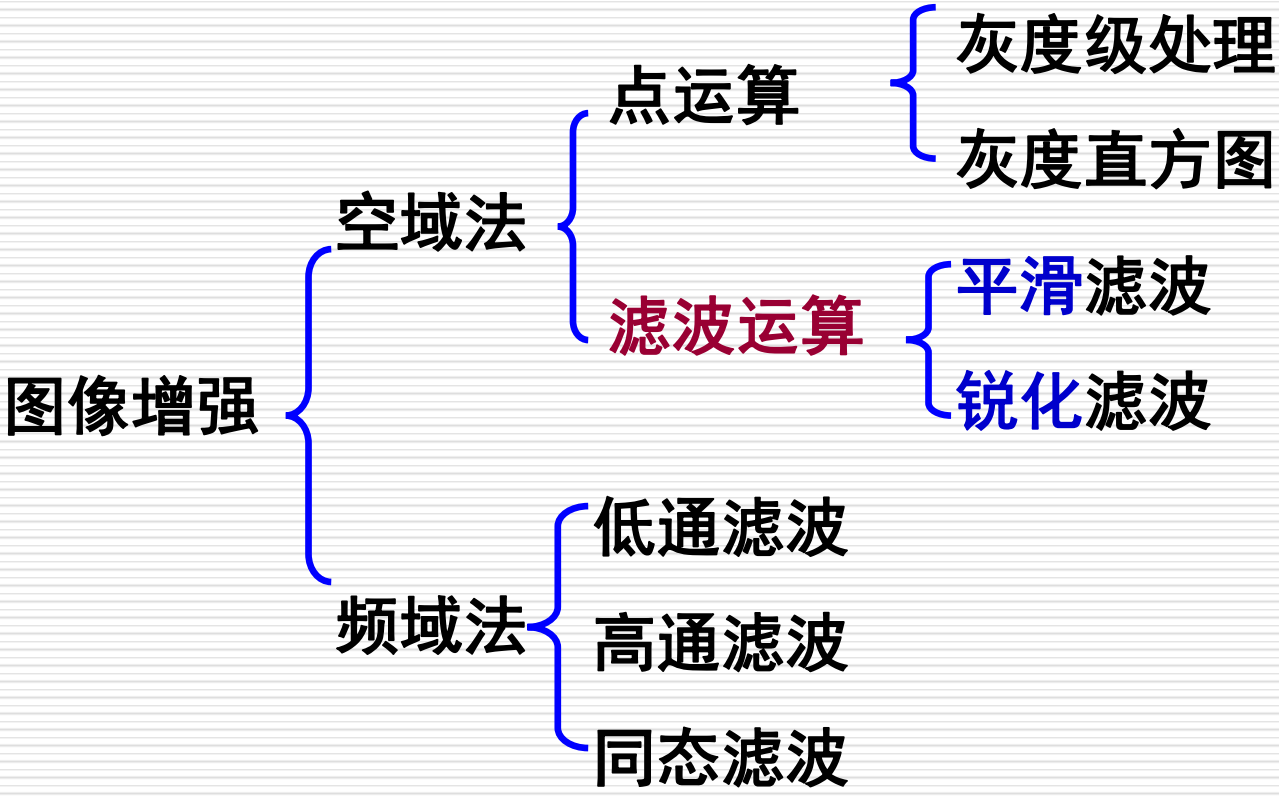
(3)有高斯噪声图像

(4)对椒盐噪声图像中值滤波

(5) 对高斯噪声图像中值滤波

◆ 对有椒盐、高斯噪声的图像，进行中值滤波，对于消除孤立点和线段的干扰中值滤波十分有效，对于高斯噪声则效果不佳。

◆ 中值滤波的优点在于去除图像噪声的同时，还能够保护图像的边缘信息。


图像增强
空域法
点运算
灰度级处理
灰度直方图
滤波运算
平滑滤波
锐化滤波
频域法
低通滤波
高通滤波
同态滤波

4.3.3 图像锐化滤波

在图像的判读或识别中常需要突出边缘和轮廓信息。图像锐化的目的就是增强图像的边缘或轮廓。

图像平滑(邻域平均法)—通过积分过程—使图像的边缘模糊

图像锐化—通过微分过程—使图像的边缘突出、清晰

□ 1梯度运算

图像锐化中最常用的方法是梯度法。对图像f(x,y)，在点(x,y)的梯度是一个二维列矢量，可定义为：

$$grad[f(x,y)]=\begin{bmatrix}\dfrac{\partial f}{\partial x}\\ \dfrac{\partial f}{\partial y}\end{bmatrix}$$

如果用G[f(x,y)]来表示梯度的幅度(大小)，那么

$$G[f(x,y)]=\sqrt{(\frac{\partial f}{\partial x})^2+(\frac{\partial f}{\partial y})^2}$$

梯度的方向为：

$$\theta=\arctan\left(\frac{\partial f/\partial y}{\partial f/\partial x}\right)$$

对于数字图像处理，微分可近似用差分代替

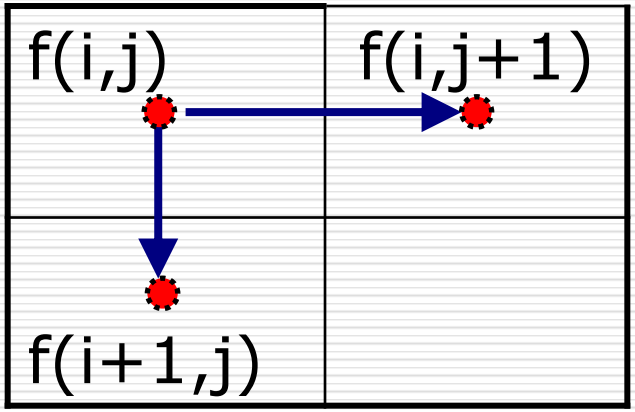
$$G[f(i,j)] \approx \sqrt{[f(i,j)-f(i+1,j)]^2+[f(i,j)-f(i,j+1)]^2}$$

在实际计算中，为了降低图像的计算量，常用绝对值来代替平方和平方根运算，因此可将上式简化为：

$$G[f(i,j) \approx |f(i,j)-f(i+1,j)|+|f(i,j)-f(i,j+1)|$$

$$G[f(i,j)] \approx |f(i,j) - f(i+1,j)| + |f(i,j) - f(i,j+1)|$$

下图a示出了上式中像素间的关系。

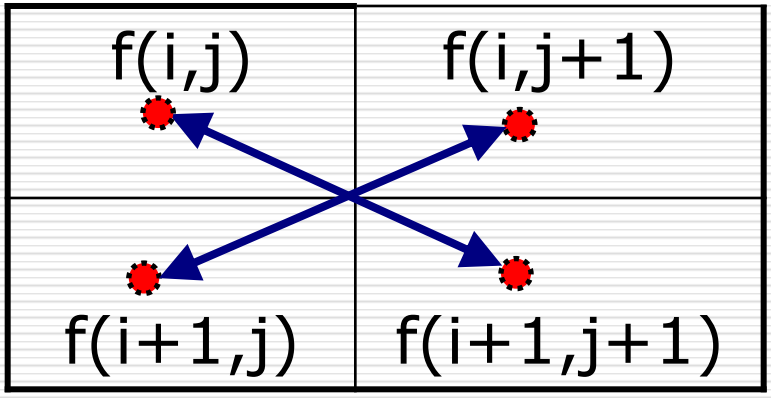


a. 典型梯度算法（直接沿x和y方向的一阶差分方法）

计算二维离散梯度的一种方法

关于梯度处理的另一种方法是罗伯茨（Roberts ）梯度法。这是一种交叉差分法。其近似计算值如下式

$$G[f(i,j)] \approx \max(|f(i+1,j+1)-f(i,j)|, |f(i+1,j)-f(i,j+1)|)$$



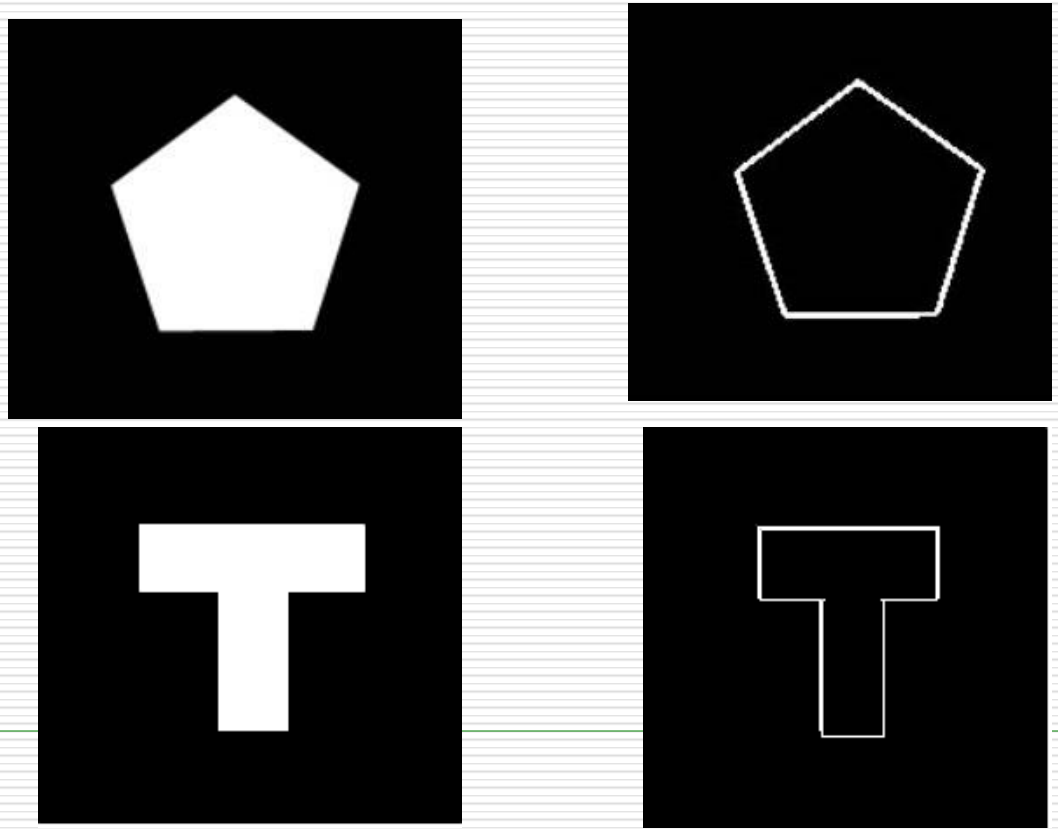
b. Roberts梯度算法(交叉差分方法)

计算二维离散梯度的另一种方法

由上面的公式可见，梯度的近似值都和相邻像素的灰度差成正比。这正象所希望的那样，在一幅图像中，边界区灰度变化较大，因而梯度值也较大，而那些灰度变化缓慢的区域，梯度值也相应较小，对于灰度级为常数的区域梯度值为零。

因此图像经过梯度运算后可使其细节清晰从而达到锐化的目的。

这种性质正如下图所示。由于梯度运算的结果，使得图像中不变的白区变为零灰度值，黑区仍为零灰度值，只留下了灰度值急剧变化的边源处的点。



利用梯度进行图像增强方法有很多。最简单的方法是使其输出图像 g(x,y) 的各点等于该点的梯度幅度，即

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

这个简单方法的缺点是使 f(x, y) 中所有平滑区域在 g(x,y) 中变成暗区，因为平滑区内各点梯度很小。

为克服这一缺点可采用阈值法（或叫门限法）。其方法如下式表示

$$g(x,y)=\begin{cases} G[f(x,y)] & \text{若 } G[f(x,y)]\geq T \\ f(x,y) & \text{其他} \end{cases}$$

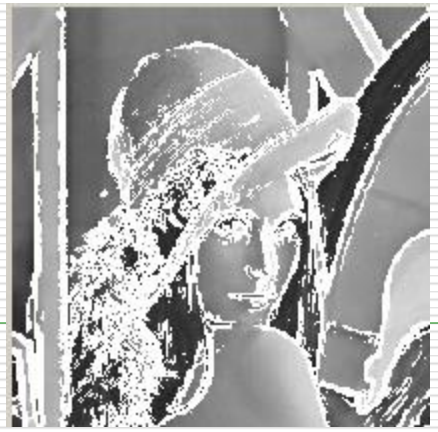
非负[illegible]当梯度值大于或等于T[illegible]就取[illegible]度值，如果梯度值小于[illegible]原来[illegible]

这样，通过合理地选择阈值，就有可能既不破坏平滑区域的灰度值又能有效地强调了图像的边缘。

基于上述思路的另一种作法是给边缘处的像素值规定一个特定的灰度级L_G，即

$$g(x,y)=\begin{cases} L_G & \text{若 } G[f(x,y)]\geq T \\ f(x,y) & \text{其他} \end{cases}$$

这种处理会使图像边缘的增强效果更加明显。



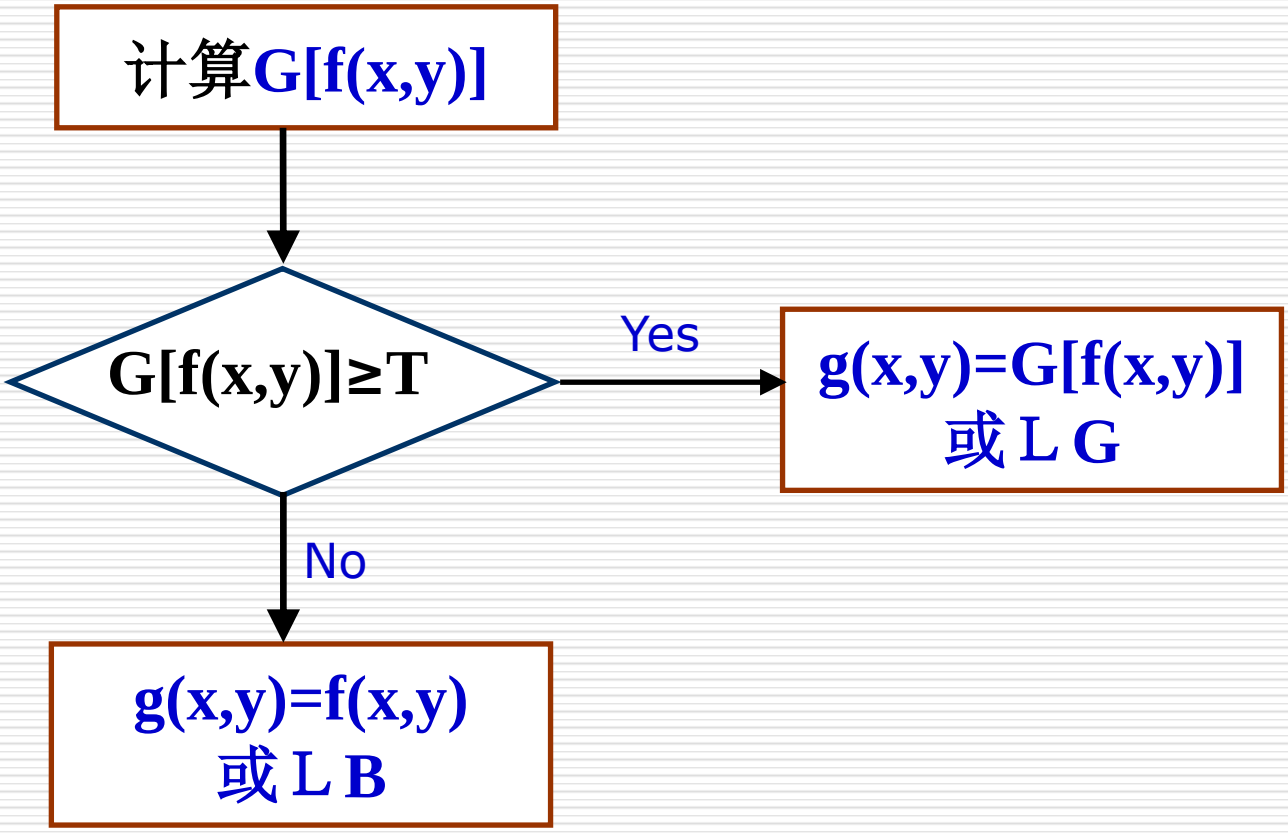
当只研究图像边缘灰度级变化时，要求不受背景的影响，则用下式来构成梯度图像

$$g(x,y)=\begin{cases} G[f(x,y)] & \text{若 } G[f(x,y)]\geq T \\ L_B & \text{其它} \end{cases}$$

式中 L_B 是规定的背景灰度值。

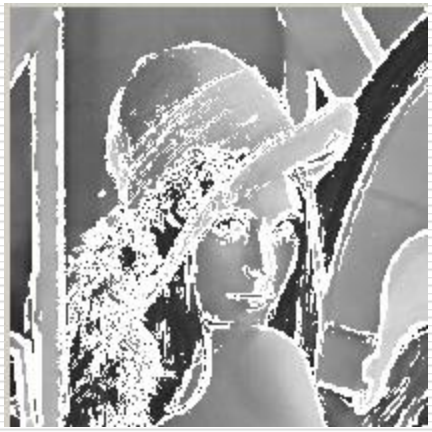
另外，如果只对边缘的位置感兴趣，则可采用下式的规定产生图像。

$$g(x,y)\begin{cases} L_G & \text{若 } G[f(x,y)] \geq T \\ L_B & \text{其它} \end{cases}$$



梯度法图像锐化处理计算框图



梯度法图像锐化的5种方法比较

□ 2 Laplacian算子

一个连续的二元函数 f(x,y)， 其Laplacian运算定义为

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

其中∇^2为Laplacian算子。

对数字图像而言，二阶偏导数可用二阶差分近似，可推导出Laplacian算子表达式为

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

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

上式也可由拉普拉斯算子模板来表示：

拉普拉斯变换对图像增强的基本方法：

H1=

0	1	0
1	-4	1
0	1	0

$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

$$= 5f(x,y) - [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]$$

0	-1	0
-1	5	-1
0	-1	0

用拉普拉斯算子对图像进行增强

```
I=imread('lenna.bmp');
subplot(1,3,1), imshow(I);
h=[0 1 0,1 -4 1,0 1 0];          % 拉普拉斯算子
%J=conv2(I, h,' same');
J=filter2(h,I);   %用拉普拉斯算子对图像滤波运算
J=uint8(J);
subplot(1,3,2), imshow(J);
K=I-J;            %增强的图像为原始图像减去拉普拉斯算子滤波的图像
subplot(1,3,3), imshow(K);
```

a. 原始图像

b. 拉普拉斯图像

C. 原图像-拉普拉斯图像

原始图像及经过增强的图像

1. 试用拉普拉斯算子对下图进行增强运算，并将增强后的图像画出来(当模板超出图像边界时，扩充图像，利用常数0来填充扩充的图像边界，使得增强后的图像与原图像大小一样)。

0	0	0	0	0
2	2	2	2	2
3	4	5	4	3
2	3	4	3	2
0	0	0	0	0

```
389  function laplacian_Callback(hObject, eventdata, handles)
390  % 拉普拉斯算子
391  img=[0 0 0 0 0;2 2 2 2 2;3 4 5 4 3;2 3 4 3 2;0 0 0 0 0];
392  H=[0 -1 0;-1 5 -1;0 -1 0];
393  imgnew=filter2(H,img);
394  img
395  ims
396  tit
397
398  %H1
399  %M1
400
401  %figure,
```

imgnew: 5x5 double =

-2	-2	-2	-2	-2
5	2	1	2	5
7	7	11	7	7
4	5	9	5	4
-2	-3	-4	-3	-2

```
389  function laplacian_Callback(hObject, eventdata, handles)
390  % 拉普拉斯算子
391  img=[0 0 0 0 0;2 2 2 2 2;3 4 5 4 3;2 3 4 3 2;0 0 0 0 0];
392  H=[0 -1 0;-1 5 -1;0 -1 0];
393  imgnew=filter2(H,img);
394  imgnew=uint8(imgnew);
395  ims
396  tit                                                    ;
397
398  %H1                                                  .5);%
399  %M1
400
401  %fi
402  %subplot(1,3,1),imshow(img);
```

imgnew: 5x5 uint8 =

0	0	0	0	0
5	2	1	2	5
7	7	11	7	7
4	5	9	5	4
0	0	0	0	0

□ 3 其它锐化算子

利用梯度与差分原理组成的锐化算子还有以下几种：

Sobel算子：$S=(d_x^2+d_y^2)^{1/2}$

$$dx=[f_{i-1,j-1}+2f_{i,j-1}+f_{i+1,j-1}]-[f_{i-1,j+1}+2f_{i,j+1}+f_{i+1,j+1}]$$

$$dy=[f_{i+1,j-1}+2f_{i+1,j}+f_{i+1,j+1}]-[f_{i-1,j-1}+2f_{i-1,j}+f_{i-1,j+1}]$$

用模板表示：

$dx=$

1	0	-1
2	0	-2
1	0	-1

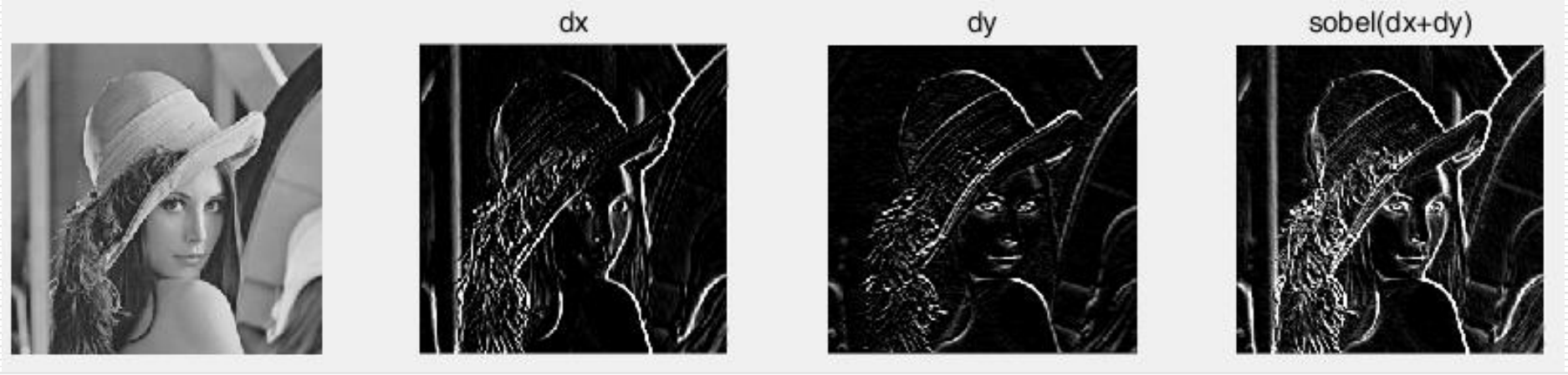
$dy=$

-1	-2	-1
0	0	0
1	2	1

用Sobel算子对图像进行增强

```
dx=[1  0 -1; 2 0 -2;1 0 -1];
dy=[-1 -2 -1;0 0 0;1 2 1];
gradx=filter2(dx,img);
grady=filter2(dy,img);

subplot(1,4,1);imshow(img);
subplot(1,4,2);imshow(uint8(gradx));title('dx');
subplot(1,4,3);
imshow(uint8(grady));title('dy');
subplot(1,4,4);
imshow(uint8(gradx)+uint8(grady));title('sobel(dx+dy)');
```



原图像　　x方向梯度　　y方向梯度　　sobel算子

Prewitt算子：$S=(dx^2+dy^2)^{1/2}$

用模板表示：

$dx=$

1	0	-1
1	0	-1
1	0	-1

$dy=$

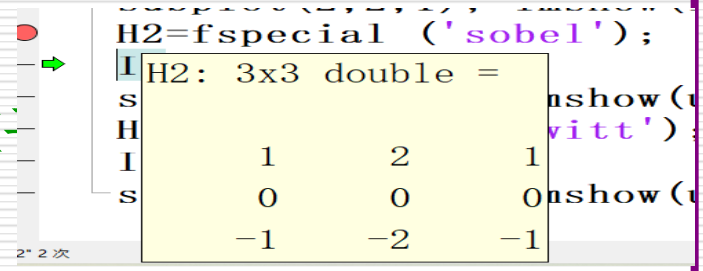
-1	-1	-1
0	0	0
1	1	1

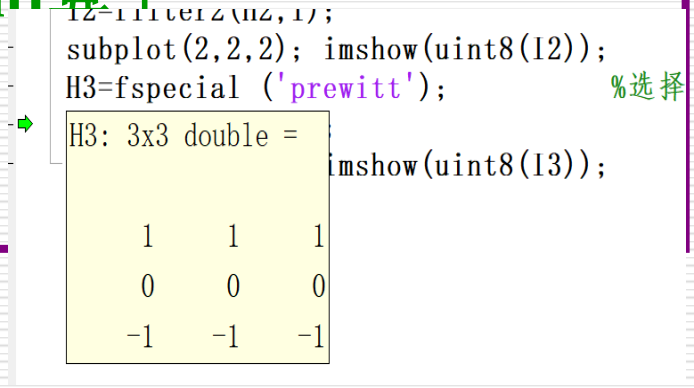
用prewitt算子对图像进行增强

原图像　　x方向梯度　　y方向梯度　　prewitt算子

利用sobel算子、prewitt算子(matlab函数)对模糊图像进行增强

```
I=imread('lenna.bmp');
subplot(1,3,1); imshow(I);
H2=fspecial ('sobel');   %选择sobel算子
I2=filter2(H2,I);
subplot(1,3,2); imshow(uint8(I2));
H3=fspecial ('prewitt');   %选择prewitt算子
I3=filter2(H3,I);
subplot(1,3,3); imshow(uint8(I3));
```



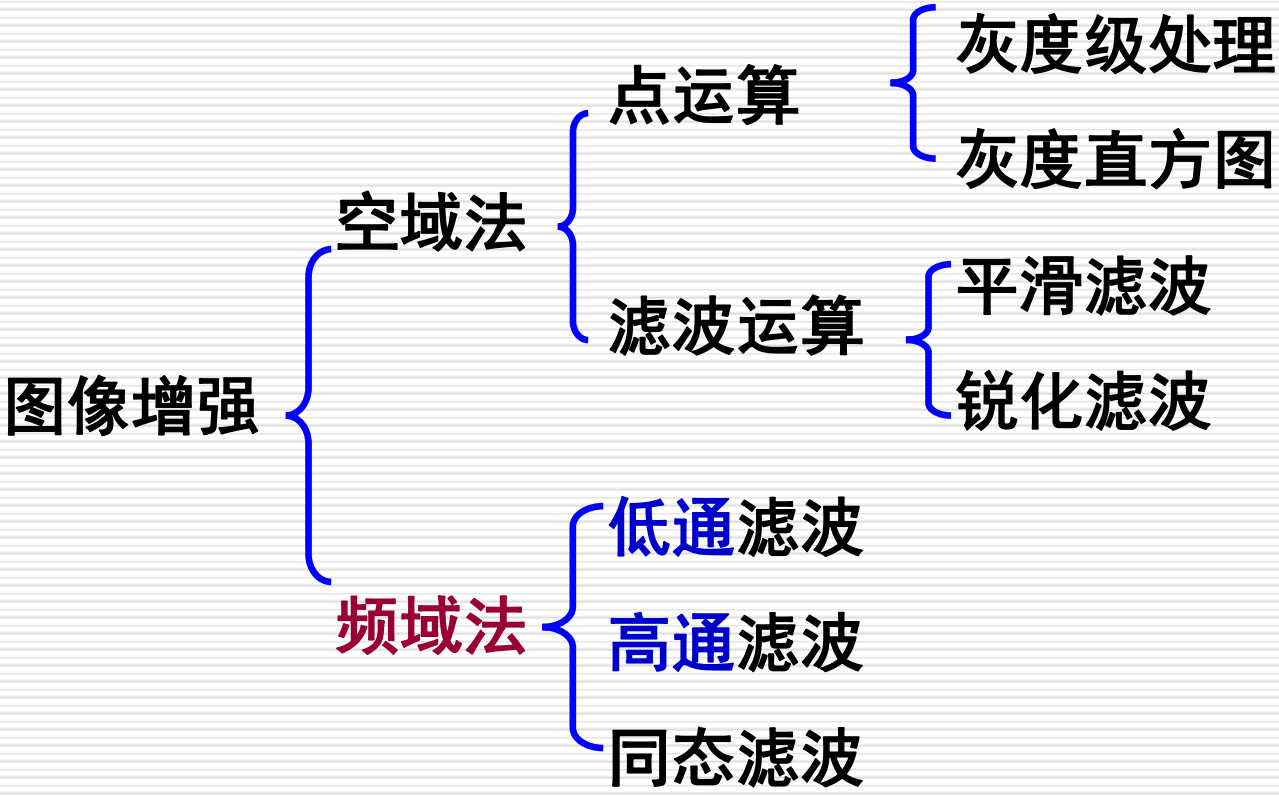


a. 原图像

b. sobel算子

c. prewitt算子

各个算子对图像锐化结果


图像增强
空域法
点运算
灰度级处理
灰度直方图
滤波运算
平滑滤波
锐化滤波
频域法
低通滤波
高通滤波
同态滤波

4.4 基于频率变换的图像增强

图像增强的目的主要包括：

①消除噪声，改善图像的视觉效果；

②突出边缘，有利于识别和处理。

前面是关于图像空间域增强的知识，下面介绍频率域增强的方法。

假定原图像为f(x，y)，经傅立叶变换为F(u，v)。频率域增强就是选择合适的滤波器H(u, v)对F(u, v)的频谱成分进行处理，然后经傅立叶逆变换得到增强的图像g(x, y)。

频率域增强的一般过程如下：

计算步骤

（1）计算所需增强图像的傅立叶变换

（2）将其与1个(根据需要设计)转移函数相乘

（3）再将结果傅立叶反变换，以得到增强的图像

4.4.2 频域图像平滑

图像的平滑除了在空间域中进行外，也可以在频率域中进行。由于图像的边缘和噪声主要集中在高频部分，为去除噪声改善图像质量，滤波器需采用低通滤波器H(u,v)来抑制高频成分，然后再进行逆傅立叶变换获得滤波图像，就可达到平滑图像的目的。

常用的频率域低通滤波器H(u,v)有：

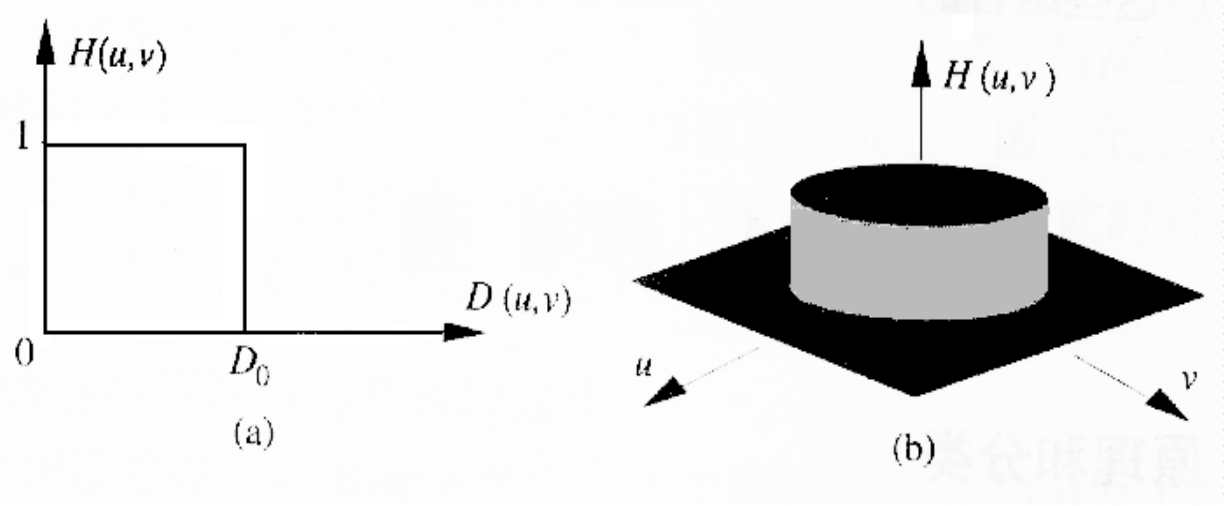
- **理想低通滤波器**

一个理想的二维低通滤波器的传递函数H(u,v)表示为

$$H(u,v)=\begin{cases}1 & D(u,v)\le D_0\\ 0 & D(u,v)>D_0\end{cases}$$

式中D_0是理想低通滤波器的截止频率, D(u,v)是(u,v)点到频率域原点的距离

$$\mathrm{D}(u,v)=\left[u^2+v^2\right]^{\frac{1}{2}}$$



这里的**理想**是指以D_0为半径的圆内的所有频率可以完全不受影响地通过滤波器，而在D_0外的频率则完全通不过。

由于高频成分包含有大量的边缘信息，因此采用该滤波器在去噪声的同时将会导致边缘信息损失而使图像变模糊。

```
% 例    理想低通滤波器
F=fft2(noisy);        %傅里叶变换,   noisy是添加噪声的图像
F=fftshift(F);        %将图像频谱中心从矩阵原点移到矩阵中心
[M N]=size(F);        %求矩阵F的大小
u1=fix(M/2);    u2=fix(N/2);
D0=40;                %截止频率
for u=1:M
   for v=1:N
      D=sqrt((u-u1)^2+(v-u2)^2);
      if   (D<=D0)   H=1;
      else  H=0;
      end             %理想低通滤波器
      result(u,v)=H*F(u,v);
   end
end
result=ifftshift(result);
J1=ifft2(result);     %傅里叶逆变换
J2=uint8(real(J1));
imshow(J2);
```

理想低通滤波器例-----具有振铃和模糊现象

原图(添加噪声)

D0=10

D0=30

D0=50

D0=80

D0=200

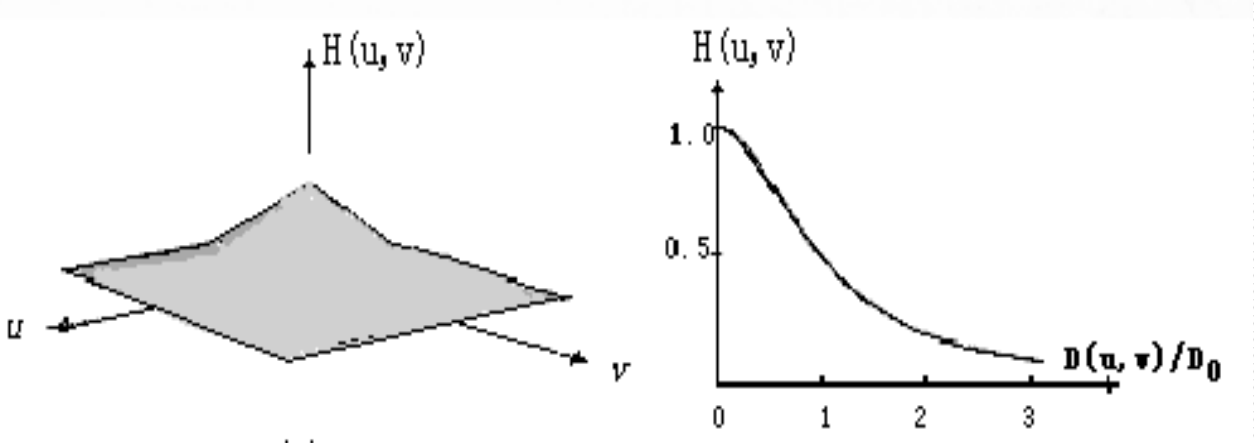
结论：半径D0越小，模糊现象越严重。

➢ 2. Butterworth低通滤波器

n阶Butterworth滤波器的传递函数为：

$$H(u,v)=\frac{1}{1+[\frac{D(u,v)}{D_0}]^{2n}}$$

它的特性是连续性衰减，而不象理想滤波器那样陡峭变化，即明显的不连续性。因此采用该滤波器滤波在抑制噪声的同时，图像边缘的模糊程度大大减小，没有振铃效应产生。



巴特沃斯低通滤波器　n =2

原图(添加噪声)

D0=10

D0=30

D0=50

D0=80

D0=200

结论：与理想低通滤波器相比，经巴特沃斯滤波器处理的图像的模糊程度会大大减少，并且没有振铃现象 。

4.4.2 频率域锐化

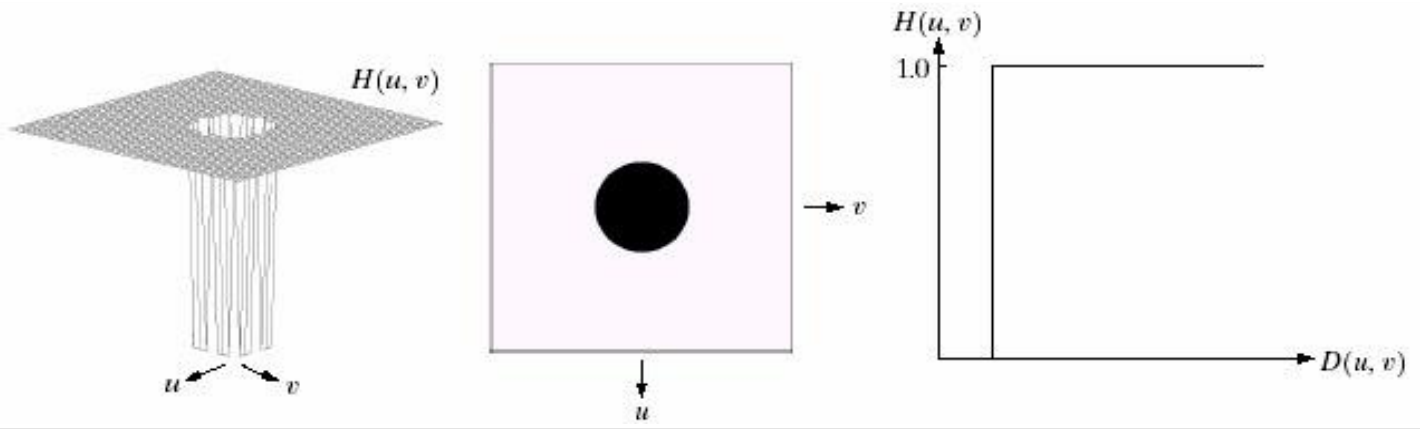
图像的边缘、细节主要位于高频部分。频率域锐化就是为了消除模糊，突出边缘。因此采用高通滤波器让高频成分通过，使低频成分削弱，再经逆傅立叶变换得到边缘锐化的图像。

常用的高通滤波器有：

1）理想高通滤波器

二维理想高通滤波器的传递函数为

$$H(u,v)=\begin{cases}0 & D(u,v)\le D_0\\ 1 & D(u,v)>D_0\end{cases}$$

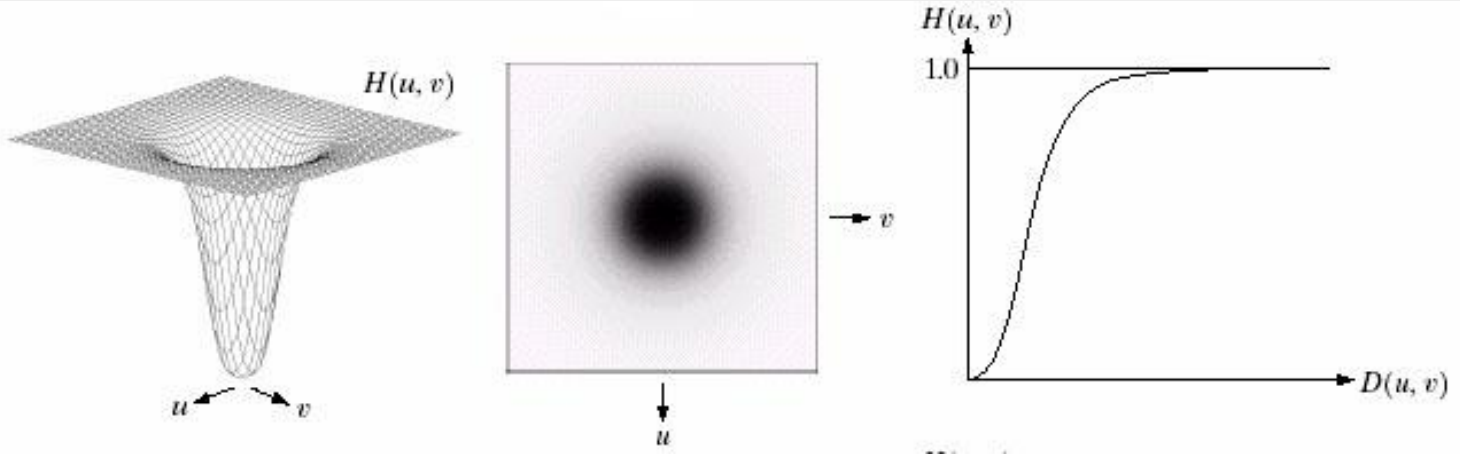


通过高通滤波正好把以D0为半径的圆内的频率成分衰减掉，对圆外的频率成分则无损通过。

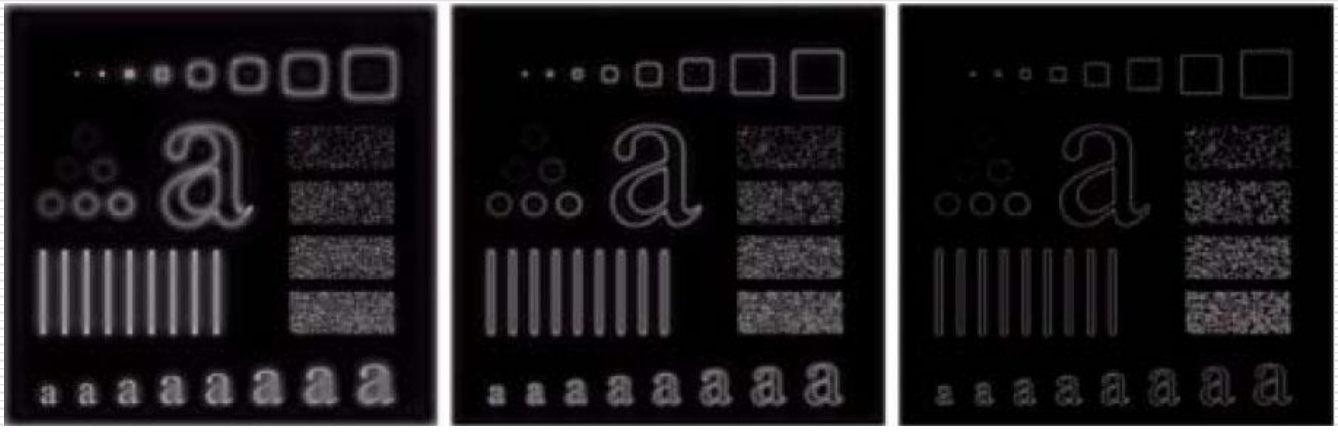
2）巴特沃斯高通滤波器

n阶巴特沃斯高通滤波器的传递函数定义如下

$$H(u,v)=\frac{1}{1+[\frac{D_0}{D(u,v)}]^{2n}}$$



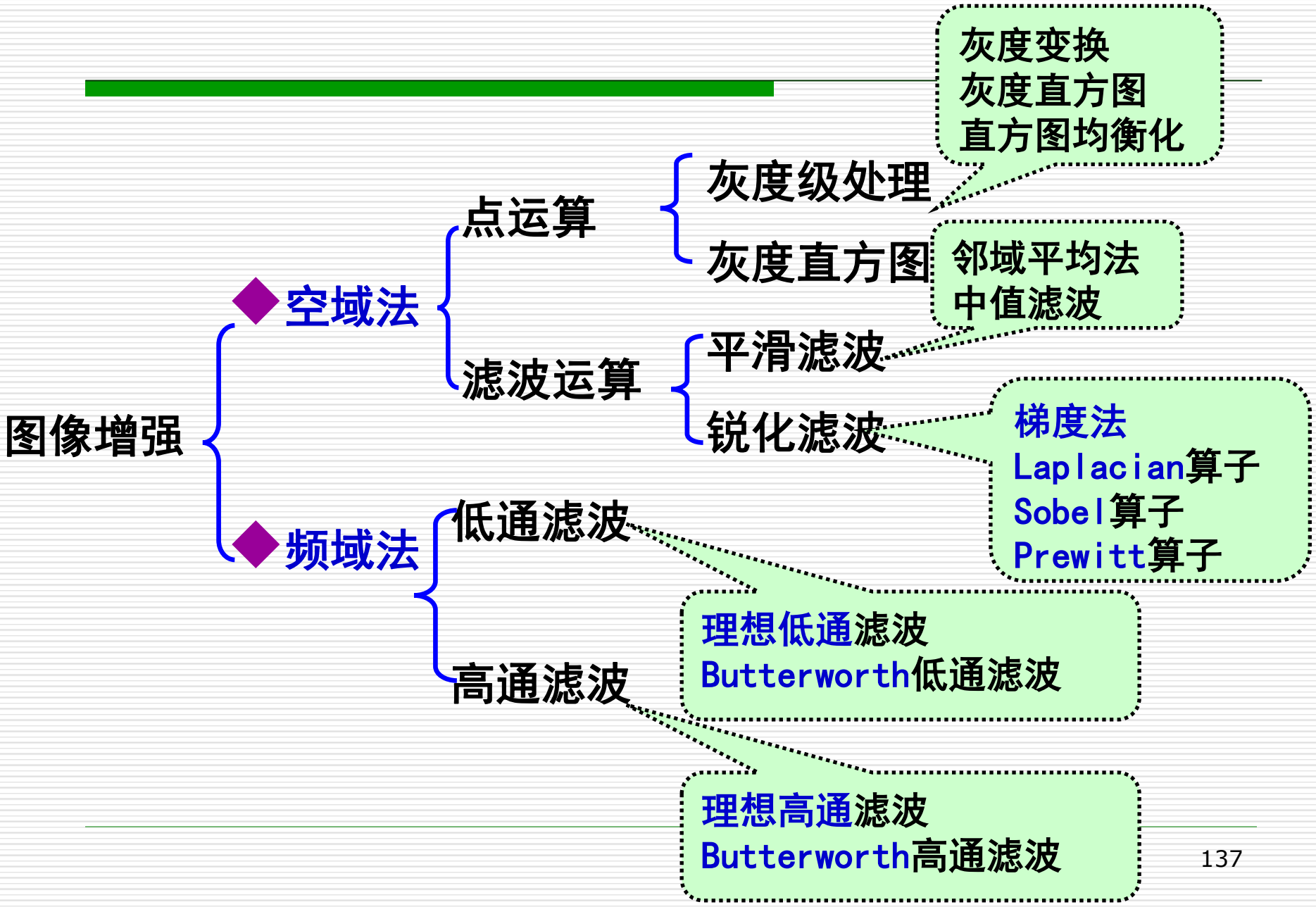
理想高通滤波器



巴特沃斯高通滤波器

D0=15　　D0=30　　D0=80

高通滤波处理结果


灰度变换
灰度直方图
直方图均衡化
灰度级处理
点运算
灰度直方图
邻域平均法
中值滤波
空域法
平滑滤波
滤波运算
锐化滤波
梯度法
Laplacian算子
Sobel算子
Prewitt算子
图像增强
低通滤波
频域法
理想低通滤波
Butterworth低通滤波
高通滤波
理想高通滤波
Butterworth高通滤波