

# 一、简答

## 1.颜色模型及适用领域 p254

### 1) RGB

CCD 技术直接感知 R,G,B 三个分量

是图像成像、显示、打印等设备的基础

### 2) CMY(青、深红、黄)、CMYK (青、深红、黄、黑)

运用在大多数在纸上沉积彩色颜料的设备，如彩色打印机和复印机

CMYK

打印中的主要颜色是黑色

等量的 CMY 原色产生黑色，但不纯

在 CMY 基础上，加入黑色，形成 CMYK 彩色空间

### 3) HSI(色调、饱和度、亮度)

两个特点：

I 分量与图像的彩色信息无关

H 和 S 分量与人感受颜色的方式是紧密相连的

将亮度(I)与色调(H)和饱和度(S)分开

避免颜色受到光照明暗(I)等条件的干扰

仅仅分析反映色彩本质的色调和饱和度

广泛用于计算机视觉、图像检索和视频检索

#### 4) YIQ

Y 指亮度(Brightness),即灰度值

I 和 Q 指色调, 描述色彩及饱和度

用于彩色电视广播, 被北美的电视系统所采用 (属于 NTSC 系统)

Y 分量可提供黑白电视机的所有影像信息

#### 5) YUV

Y 指亮度,与 YIQ 的 Y 相同

U 和 V 也指色调, 不同于 YIQ 的 I 和 Q

用于彩色电视广播, 被欧洲的电视系统所采用 (属于 PAL 系统)

Y 分量也可提供黑白电视机的所有影像 信息

#### 6) YCbCr

Y 指亮度,与 YIQ 和 YUV 的 Y 相同

Cb 和 Cr 由 U 和 V 调整得到

**JPEG 采用的彩色空间**

## 2.图像量化级数 影响

函数值的数据化

量化级数越高, 图像视觉效果越好

量化等级越多, 所得图像层次越丰富, 灰度分辨率高, 图像质量好, 但数据量大; 量化等级越少, 图像层次欠丰富, 灰度分辨率低,

会出现假轮廓现象，图像质量变差，但数据量小.

### 3.傅里叶变换 低通、高通、滤波应用原理

低通滤波：截断傅里叶变换中的所有**高频**成分，这些高频成分位于指定距离  $D_0$  之外

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

频率矩形的中心在  $(u, v) = (\frac{M}{2}, \frac{N}{2})$ ，从点  $(u, v)$  到中心（原点）的距离如

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

高通滤波：截断傅里叶变换中的所有**低频**成分，这些低频成分位于指定距离  $D_0$  之内

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

频率矩形的中心在  $(u, v) = (\frac{M}{2}, \frac{N}{2})$ ，从点  $(u, v)$  到中心（原点）的距离如

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

### 4.存储空间计算方法 灰度级→大小

图片所占内存大小 = 图片长度（像素） \* 图片宽度（像素） \* 一个像素所占内存空间（单位：字节）

如：一幅 256\*256 的图像，若灰度级数为 16，则该图像的大小？

256\*256 表示像素个数，16 级灰度用二进制表示需要 4 位，故存储图

像所需的二进制位数为：256\*256\*4。即 256Kbit，所需字节数除以 8，为 32KB

## 5.图像基本特征

灰度、直方图、边缘、

颜色特征、纹理特征、形状特征和空间关系特征

## 6.快速傅里叶变换基本原理

FFT 算法基于一个叫做逐次加倍的方法。通过推导将原始傅里叶转换成两个递推公式

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad u = 0, 1, 2, \dots, M-1$$

$$F(u) = \frac{1}{2} [F_{\text{even}}(u) + F_{\text{odd}}(u) W_{2K}^u]$$

$$F(u+K) = \frac{1}{2} [F_{\text{even}}(u) - F_{\text{odd}}(u) W_{2K}^u]$$

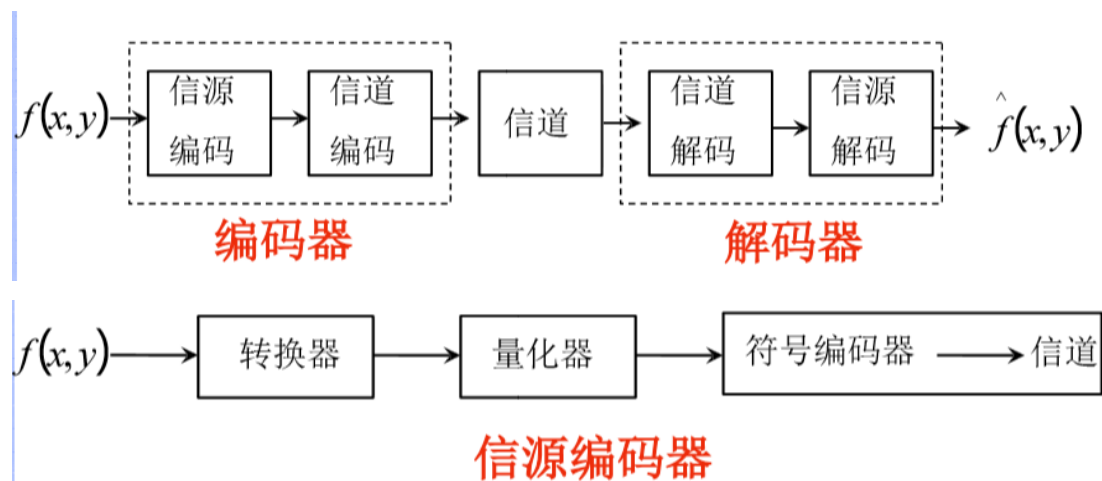
归纳快速傅立叶变换的思想：

- (1) 通过计算两个单点的 DFT，来计算两个点的 DFT，
- (2) 通过计算两个双点的 DFT，来计算四个点的 DFT，…，以此类推
- (3) 对于任何  $N=2^m$  的 DFT 的计算，通过计算两个  $N/2$  点的 DFT，

来计算  $N$  个点的 DFT

## 7. 图像编码基本原理

由于图像存在：编码冗余、像素间冗余、心理视觉冗余三种冗余，  
当去一个或多个冗余被减少或消除时，压缩就实现了。



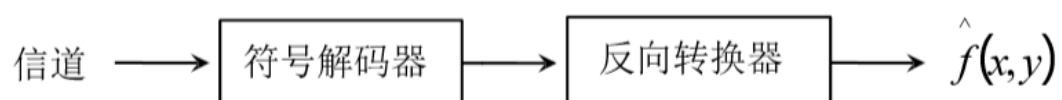
信源编码器：减少或消除输入图像中的编码冗余 间冗余及心理视觉冗余

转换器：减少像素间冗余

量化器：减少心理视觉冗余，该步操作是不可逆的

符号编码器：减少编码冗余

并不是每个图像压缩系统都必须包含这 行无误差压缩时，必须去掉量化器



符号解码器：进行符号编码的逆操作

反向转换器：进行转换器的逆操作

## 8.视频压缩过程

### MPEG 帧的分类

- ✧ I 帧（Intra-picture） 不需要参考其它画面而**独立**进行压缩**编码**的画面
- ✧ P 帧（Predicted-picture） 参考**前面**已编码的 I 或 P 画面进行预测**编码**的画面
- ✧ B 帧（Bidirectional-picture） 既参考前 面的 I 或 P 画面、又参考后面的 I 或 P 画面 进行双 向预测编码的画面

### 连续帧图像压缩的基本思想

可以根据同帧附近像素来加以预测，被 称为：帧内编码技术

可以根据附近帧中的像素来加以预测， 被称为：帧间编码技术

对于视频帧，通常使用变换编码来消去除图像的帧内冗余，用运动估计和运动补偿来去除图像的帧间冗余，用熵编码来进一步提高压缩的效率

## 9.图像增强及效果

### 空间域

**平滑空间滤波器**：对图像进行模糊处理，去除不重要的细节，减少噪声，包括线性滤波器和非线性滤波器

**锐化滤波器**：突出图像中的细节

### 频率域

**陷波滤波器：**由于图像平均值为 0，而产生整体平均灰度级的降低，  
用于识别由**特定的、局部化频域**成分引起的空间图像效果

**低通滤波器** 进行平滑处理，模糊效果

**高通滤波器** 突出边缘等细节部分，锐化效果

## 二、大题

### 1.应用哈夫曼编码,对图像进行编码 步骤、结果、编码长度、存储空间

P354

步骤:

- 1 将需要考虑的符号概率排序,并将两个最小概率的符号合并为一个符号,从而化简信源
2. 对每个化简后的信源进行编码,从最小的信源开始,一直编码到原始的信源

#### 步骤1

Original source		Source reduction			
Symbol	Probability	1	2	3	4
$a_2$	0.4	0.4	0.4	0.4	0.6
$a_6$	0.3	0.3	0.3	0.3	
$a_1$	0.1	0.1	0.2	0.3	0.4
$a_4$	0.1	0.1			
$a_3$	0.06	0.1	0.1	0.1	0.1
$a_5$	0.04				

FIGURE 8.11  
Huffman source  
reductions.

#### 步骤2

FIGURE 8.12  
Huffman code  
assignment  
procedure.

Original source			Source reduction			
Sym.	Prob.	Code	1	2	3	4
$a_2$	0.4	1	0.4	1	0.4	1
$a_6$	0.3	00	0.3	00	0.3	00
$a_1$	0.1	011	0.1	011	0.2	010
$a_4$	0.1	0100	0.1	0100	0.1	011
$a_3$	0.06	01010	0.1	0101		
$a_5$	0.04	01011				

编码平均长度:

$$L_{avg}=0.4*1+0.3*2+0.1*3+0.1*4+0.06*5+0.04*5=2.2 \text{ 比特/像素}$$

哈夫曼编码是块编码,唯一可解的



空间大小= $MNL_{avg}$

## 2.JPEG 压缩过程

- ✧ 1.先把整个图像**分解**成多个  $8 \times 8$  的图像块；
- ✧ 2. $8 \times 8$  的图像块经过 **DCT 变换**后，**低频分量**都集中在左上角，**高频分量**则分布在右下角（DCT 变换类似于低通滤波器），因为低频分量包含了图像的主要信息，所以可以忽略高频分量，达到压缩的目的；
- ✧ 3.使用**量化操作**去掉高频分量，量化操作就是将某一个值除以量化表中的对应值。由于量化表中左上角的值较小，而右下角的值较大，这样达到**保持低频分量**，抑制高频分量的目的；
- ✧ 4.在左上角低频分量中， $F(0,0)$ 代表了直流(DC)系数，即  $8 \times 8$  子块的平均值。由于两个相邻图像块的 DC 系数相差很小，所以采用**差分编码 DPCM**,其它 63 个元素是交流（AC）系数，采用之字型（zig-zag）顺序进行**行程编码**，使系数为 0 的值更集中；
- ✧ 5.在得到 DC 码字和 AC 行程码字后，为了进一步提高压缩比，再进行**熵编码**，采用了 Huffman 编码

## 3.直方图均衡化具体应用 累积和→映射→四舍五入

使用的方法是灰度级变换： $s = T(r)$

- ✓ 基本思想是把原始图的直方图变换为**均匀分布**的形式，这样就增加了像素灰度值的**动态范围**，从而达到增强图像整体**对比度**的效果

$$s=T(r)$$

$T(r)$ 满足下列两个条件：

- (1)  $T(r)$ 在区间  $0 \leq r \leq 1$  中为单值且单调递增
- (2) 当  $0 \leq r \leq 1$  时,  $0 \leq T(r) \leq 1$

条件(1)保证原图各灰度级在变换后仍保持从黑 到白(或从白到黑)的排列次序

条件 (2) 保证变换前后灰度值动态范围的一致性

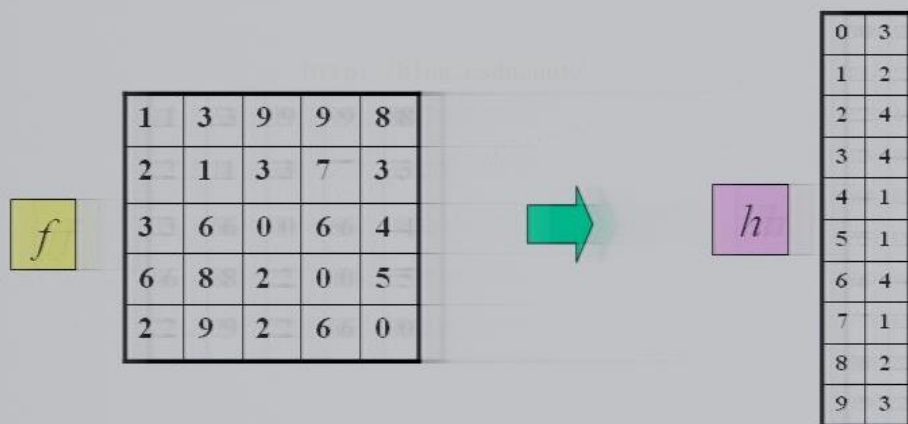
<https://blog.csdn.net/superjunenaruto/article/details/52431941>

## ■ 直方图均衡

➤ 算法：设  $f$ 、 $g$  分别为原图像和处理后的图像。

1) 求出原图  $f$  的灰度直方图，设为  $h$ 。

$h$  为一个 256 维的向量。



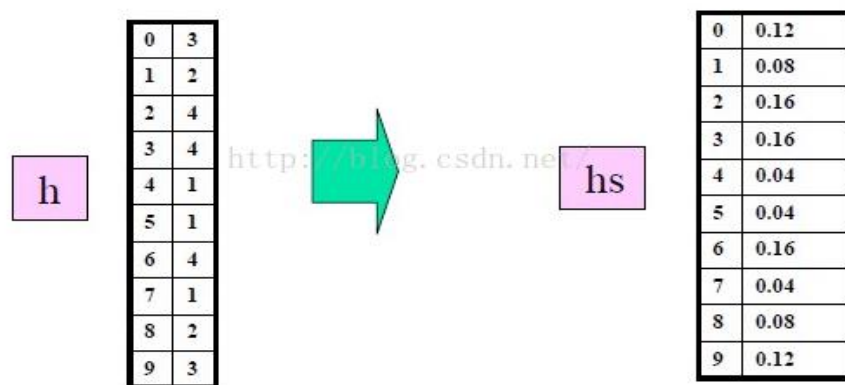
2. 求出图像  $f$  的总体像素个数

$N_f = m * n$  ( $m, n$  分别为图像的长和宽)

求得  $N_f = 25$

计算每个灰度的像素个数在整个图像中所占的百分比

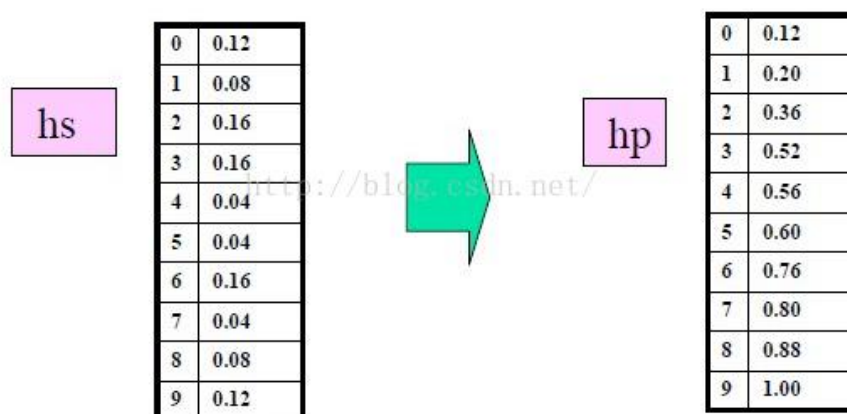
$hs(i) = h(i) / N_f$  ( $i = 0, 1, \dots, 255$ )



3) 计算图像各灰度级的累计分布  $hp$ 。

$$hp(i) = \sum_{k=0}^i h(k)$$

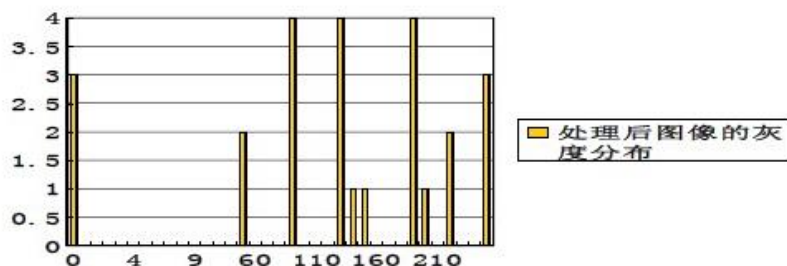
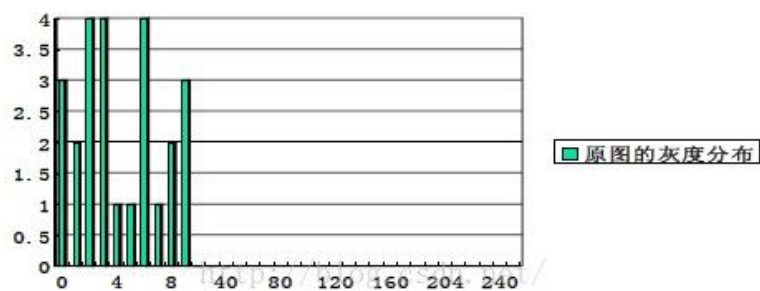
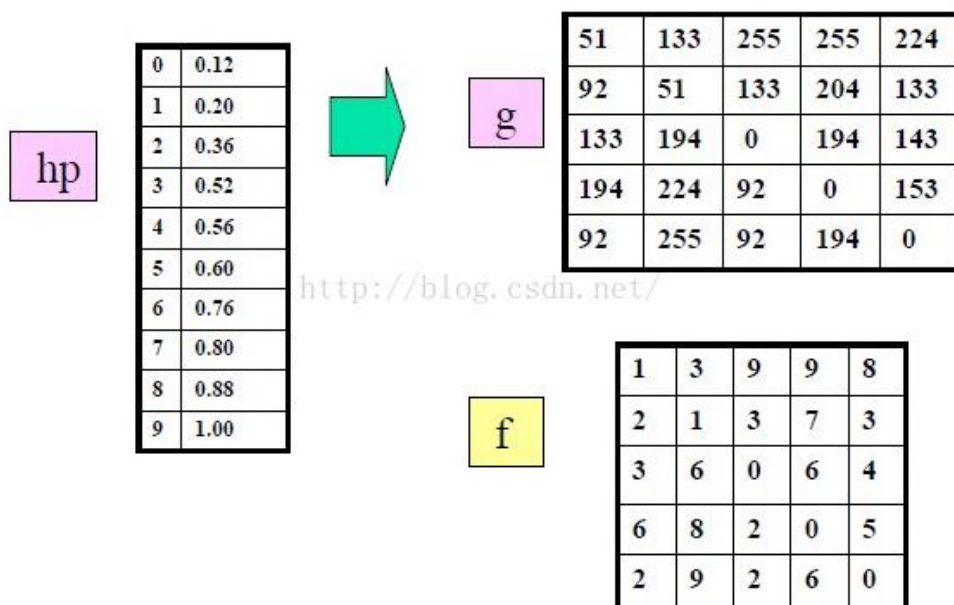
$$i = 1, 2, \dots, 255$$



求出新图像g的灰度值。

$$g = 255 \cdot hp(i) \quad i = 1, 2, \dots, 255$$

$$g = 0 \quad i = 0$$



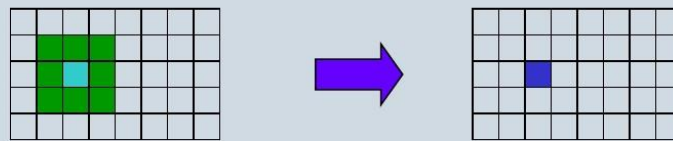
#### 4.计算空间滤波 模板+图像

##### 均值滤波

### 均值滤波器

#### —— 原理

- 在图像上，对**待处理的像素**给定一个模板，该模板包括了其周围的邻近像素。将模板中的全体像素的**均值**来替代原来的像素值的方法。



#### —— 处理方法

以模块运算系数表示即： $H_0 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

待处理像素

1	2	1	4	3
1	2	2	3	4
5	7	6	8	9
5	7	6	8	8
5	6	7	8	9

C=6.6316



1	2	1	4	3
1	3	4	4	4
5	4	5	6	9
5	6	7	8	8
5	6	7	8	9

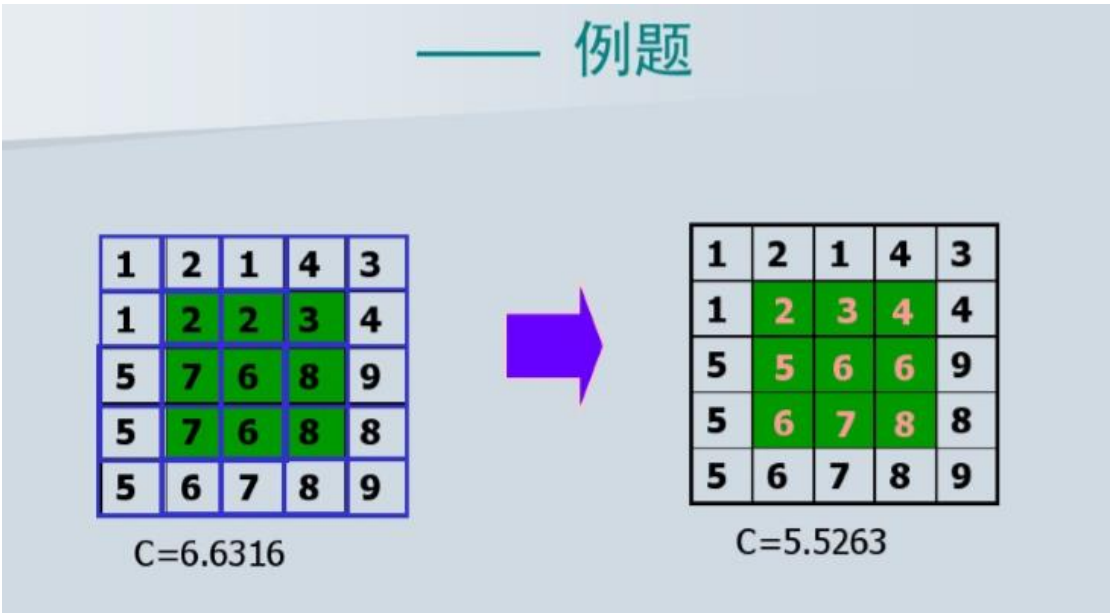
C=5.5263

边框保留不变的效果示例

示例

中值滤波

将模板区域内的像素排序，求出中间值，赋值给模板中间位置



例如：

3x3 的模板，第 5 大的是中值，

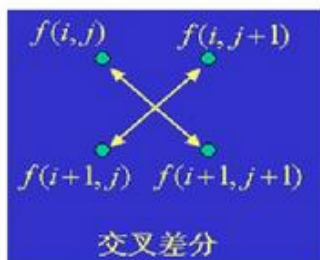
5x5 的模板，第 13 大的是中值，

7x7 的模板，第 25 大的是中值，

9x9 的模板，第 41 大的是中值

交叉微分算法 ([Roberts算法](#)) 计算公式如下：

$$g(i, j) = |f(i+1, j+1) - f(i, j)| + |f(i+1, j) - f(i, j+1)|$$



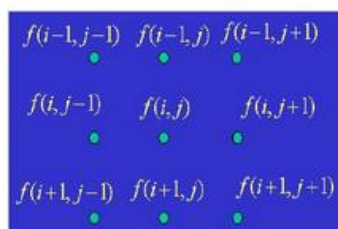
$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

**Roberts梯度算子**

特点：算法简单

[Sobel锐化](#)的计算公式如下：

$$g(i, j) = \{d_x^2(i, j) + d_y^2(i, j)\}^{\frac{1}{2}}$$



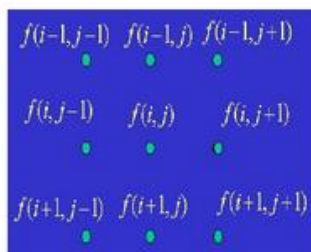
$$d_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad d_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

特点：锐化的边缘信息较强



Priwitt锐化算法 的计算公式如下:

$$g(i, j) = \{d_x^2(i, j) + d_y^2(i, j)\}^{\frac{1}{2}}$$

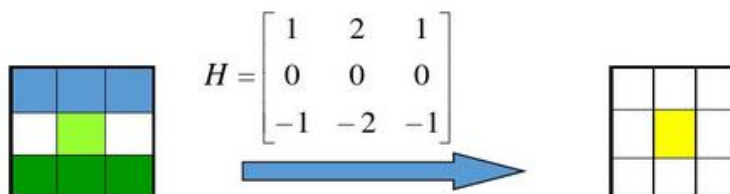


$$d_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

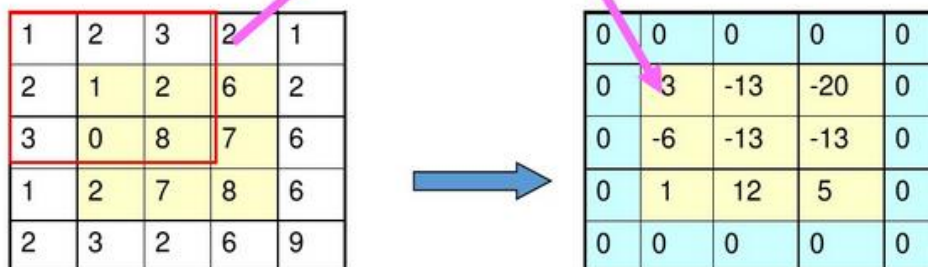
$$d_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

特点: 与Sobel相比, 有一定的抗干扰性。图像效果比较干净。

例1



$$1 + 2 \times 2 + 3 - 3 - 2 \times 0 - 8 = -3$$





## Vertical edge detection

$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6x6

"convolution"

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

3x3  
filter

=

-5			

4x4

## Vertical edge detection

$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6x6

"convolution"

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

3x3  
filter

=

-5	-4		

4x4