

# 图像计算题

## 一、直方图均衡化（P76）

对已知图像进行直方图均衡化修正。

例：表 1 为已知一幅总像素为  $n=64\times 64$  的 8bit 数字图像（即灰度级数为 8），各灰度级（出现的频率）分布列于表中。要求将此幅图像进行均衡化修正（变换），并画出修正（变换）前后的直方图。

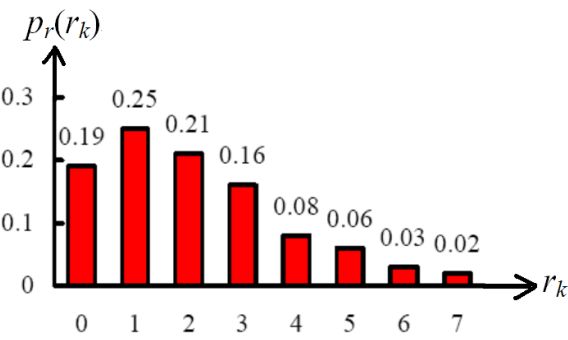
表 1

原图像灰度级 $r_k$	原各灰度级像素个数 $n_k$	原分布概率 $p_r(r_k)$
$r_0=0$	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

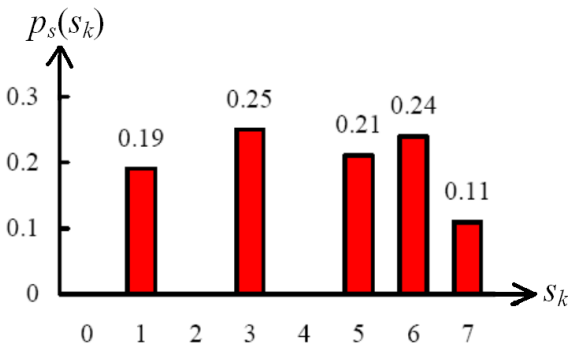
解：对已知图像均衡化过程见下表：

原图像灰度级 $r_k$	原各灰度级像素个数 $n_k$	原分布概率 $p_r(r_k)$	累积分布函数 $S_k$ 计	取整扩展 $S_k$ 并	确定映射对应关系 $r_k \rightarrow s_k$	新图像灰度级 $s_k$	新图像各灰度级像素个数 $n_{sk}$	新图像分布概率 $p_s(s_k)$
$r_0=0$	790	0.19	0.19	1	$0 \rightarrow 1$	1	790	0.19
$r_1=1$	1023	0.25	0.44	3	$1 \rightarrow 3$	3	1023	0.25
$r_2=2$	850	0.21	0.65	5	$2 \rightarrow 5$	5	850	0.21
$r_3=3$	656	0.16	0.81	6	$3 \rightarrow 6$	6	985	0.24
$r_4=4$	329	0.08	0.89	6	$4 \rightarrow 6$			
$r_5=5$	245	0.06	0.95	7	$5 \rightarrow 7$	7	448	0.11
$r_6=6$	122	0.03	0.98	7	$6 \rightarrow 7$			
$r_7=7$	81	0.02	1.00	7	$7 \rightarrow 7$			

画出直方图如下：



(a) 原始图像直方图



(b) 均衡化后直方图

\*\*以下部分不用写在答题中。

其中：

①  $r_k$ 、 $n_k$  中  $k=0, 1, \dots, 7$

②  $p_r(r_k) = n_k / n$ ，即计算各灰度级像素个数占所有像素个数的百分比，其中  $n = \sum_{j=0}^k n_j$ ，在此题中  $n=64 \times 64$ 。

③  $s_{k\text{计}} = \sum_{j=0}^k p_r(r_j)$ ，即计算在本灰度级之前（包含本灰度级）所有百分比之和。

④  $s_{k\text{并}} = \text{int}[(L-1)s_{k\text{计}} + 0.5]$ ，其中  $L$  为图像的灰度级数（本题中  $L=8$ ）， $\text{int}[\ ]$  表示对方括号中的数字取整。

⑤  $s_k = s_{k\text{并}}$

⑥  $n_{sk}$  为映射对应关系  $r_k \rightarrow s_k$  中  $r_k$  所对应的  $n_k$  之和。

⑦  $p_s(s_k) = n_{sk} / n$ ，或为映射对应关系  $r_k \rightarrow s_k$  中  $r_k$  所对应的  $p_r(r_k)$  之和。

## 模板运算 使用空间低通滤波法对图像进行平滑操作（P80）

空间低通滤波法是应用模板卷积方法对图像每一个像素进行局部处理。模板（或称掩模）就是一个滤波器，它的响应为  $H(r,s)$ ，于是滤波输出的数字图像  $g(x,y)$  用离散卷积表示为

$$g(x,y) = \sum_{r=-k}^k \sum_{s=-l}^l f(x-r, y-s) H(r,s) \quad (4.2.6)$$

式中： $x, y=0, 1, 2, \dots, N-1$ ； $k, l$  根据所选邻域大小来决定。

具体过程如下：

- （1）将模板在图像中按从左到右、从上到下的顺序移动，将模板中心与每个像素依次重合（边缘像素除外）；
- （2）将模板中的各个系数与其对应的像素一一相乘，并将所有的结果相加；
- （3）将（2）中的结果赋给图像中对应模板中心位置的像素。

对于空间低通滤波器而言，采用的是低通滤波器。由于模板尺寸小，因此具有计算量小、使用灵活、适于并行计算等优点。常用的  $3 \times 3$  低通滤波器（模板）有：

$$H_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad H_4 = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_5 = \frac{1}{2} \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

模板不同，邻域内各像素重要程度也就不同。但无论怎样的模板，必须保证全部权系数之和为 1，这样可保证输出图像灰度值在许可范围内，不会产生灰度“溢出”现象。

例：以  $H_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  为模板，对下图做低通滤波处理，写出处理结果。

1	7	1	8	1	7	1	1
1	1	1	5	1	1	1	1
1	1	5	5	5	1	1	7
1	1	5	5	5	1	8	1
8	1	1	5	1	1	1	1
8	1	1	5	1	1	8	1
1	1	1	5	1	1	1	1
1	7	1	8	1	7	1	1

解：低通滤波的步骤为：

- （1）将模板在图像中按从左到右、从上到下的顺序移动，将模板中心与每个像素依次重合（边缘像素除外）；
- （2）将模板中的各个系数与其对应的像素一一相乘，并将所有的结果相加；
- （3）将（2）中的结果赋给图像中对应模板中心位置的像素。

如图中第 2 行第 2 列处的值 =  $(1 \times 1 + 1 \times 7 + 1 \times 1 + 1 \times 1 + 2 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 5) / 10 = 2$

（其他位置同样方法计算可得）

由此步骤可得处理结果为（空白处自己计算后填入）

1	7	1	8	1	7	1	1
1	2						1
1							7
1							1
8							1
8							1
1							1

1	7	1	8	1	7	1	1
---	---	---	---	---	---	---	---

## 二、 中值滤波与邻域平均

### 中值滤波（P81）

中值滤波是对一个滑动窗口内的诸像素灰度值排序，用中间值代替窗口中心像素的原来灰度值，它是一种非线性的图像平滑法。

它对脉冲干扰及椒盐噪声的抑制效果好，在抑制随机噪声的同时能有效保护边缘少受模糊。但它对点、线等细节较多的图像却不太合适。

### 局部平滑法（邻域平均法 或 移动平均法）（P76）

局部平滑法是一种直接在空间域上进行平滑处理的技术。用邻域内各像素的灰度平均值代替该像素原来的灰度值，实现图像的平滑。

设有一幅  $N \times N$  的图像  $f(x,y)$ ，若平滑图像为  $g(x,y)$ ，则有

$$g(x,y) = \frac{1}{M} \sum_{i,j \in s} f(i,j) \quad (4.2.1)$$

式中  $x,y = 0,1,\dots,N-1$ ;

$s$  为  $(x,y)$  邻域内像素坐标的集合;

$M$  表示集合  $s$  内像素的总数。

可见邻域平均法就是将当前像素邻域内各像素的灰度平均值作为其输出值的去噪方法。

设图像中的噪声是随机不相关的加性噪声，窗口内各点噪声是独立同分布的，经过（4.2.1）平滑后，信号与噪声的方差比可望提高  $M$  倍。

这种算法简单，但它的主要缺点是在降低噪声的同时使图像产生模糊，特别在边缘和细节处。而且邻域越大，在去噪能力增强的同时模糊程度越严重。

例：对下图做  $3 \times 3$  中值滤波处理和  $3 \times 3$  邻域平均处理，写出处理结果，并比较邻域平均与中值滤波的差异。

1	7	1	8	1	7	1	1
1	1	1	5	1	1	1	1
1	1	5	5	5	1	1	7
1	1	5	5	5	1	8	1
8	1	1	5	1	1	1	1
8	1	1	5	1	1	8	1
1	1	1	5	1	1	1	1
1	7	1	8	1	7	1	1

解：

- (1) 中值滤波是对一个滑动窗口内的诸像素灰度值排序，用中间值代替窗口中心像素的原来灰度值，它是一种非线性的图像平滑法。

题目中的图像经  $3 \times 3$  中值滤波后的结果为（忽略边界）：

1	7	1	8	1	7	1	1
1	1	5	5	5	1	1	1
1	1	5	5	5	1	1	7

1	1	5	5	5	1	1	1
8	1	1	5	1	1	1	1
8	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	7	1	8	1	7	1	1

(2) 局部平滑法（邻域平均法 或 移动平均法）是用邻域内各像素的灰度平均值代替该像素原来的灰度值，实现图像的平滑。

题目中的图像经 3\*3 局部平滑法（邻域平均法 或 移动平均法）后的结果为（忽略边界）：

1	7	1	8	1	7	1	1
1	19/9	38/9	40/9	38/9	23/9	21/9	1
1							7
1							1
8							1
8							1
1							1
1	7	1	8	1	7	1	1

中值滤波法和局部平滑法（邻域平均法 或 移动平均法）均能有效削弱椒盐噪声，但中值滤波法比邻域平均法更有效，且滤波后图像中的轮廓比较清晰。

### 三、霍夫曼编码 (P124)

例：设有一信源  $A=\{a_1, a_2, a_3, a_4, a_5, a_6\}$ ，对应概率  $P=\{0.1, 0.4, 0.06, 0.1, 0.04, 0.3\}$ 。

(1) 进行霍夫曼编码（要求大率的赋码字 0，小率的赋码字 1），给出码字；

(2) 计算平均码长，信源熵和编码效率。

解：

#### (1) 编码步骤

1) 缩减信源符号数量

将信源符号按出现概率从大到小排列，然后结合

初始信源		信源的消减步骤			
符号	概率	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	0.4
$a_1$	0.1	0.1	0.2	0.3	
$a_4$	0.1	0.1	0.1		
$a_3$	0.06	0.1			
$a_5$	0.04				

2) 对每个信源符号赋值

从（消减到）最小的信源开始，逐步回到初始信源

初始信源			对消减信源的赋值			
符号	概率	码字	1	2	3	4
$a_2$	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
$a_6$	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
$a_1$	0.1	011	0.1 011	0.2 010	0.3 01	
$a_4$	0.1	0100	0.1 0100	0.1 011		
$a_3$	0.06	01010	0.1 0101			
$a_5$	0.04	01011				

由此可得哈夫曼编码结果见下表

符号	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
概率	0.1	0.4	0.06	0.1	0.04	0.3
编码结果	011	1	01010	0100	01011	00

$$\text{平均码长 } \bar{B} = \sum_{i=0}^{L-1} \beta_i p_i = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5 = 2.2$$

（其中， $\beta_i$  是灰度值为  $i$  的编码长度， $p_i$  为灰度值为  $a_i$  的概率， $L$  为灰度级数）

$$\text{信源熵 } H = - \sum_{i=0}^{L-1} p_i \log_2 p_i = 2.14$$

$$\text{编码效率 } \eta = \frac{H}{\bar{B}} = \frac{2.14}{2.2} = 0.973$$

#### 四、 费诺—仙农编码 (P126)

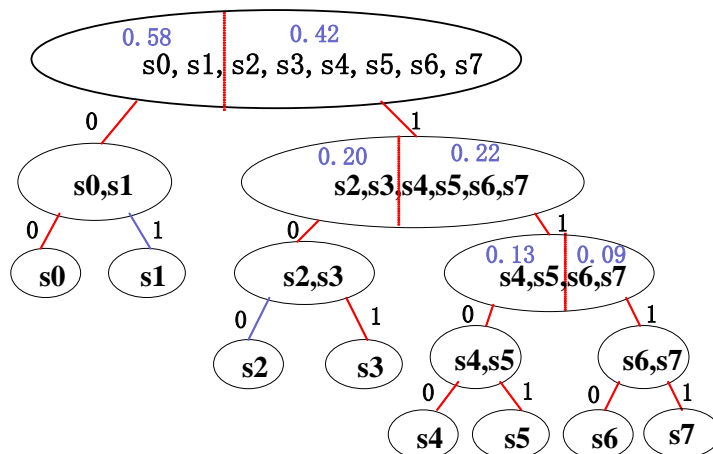
费诺—仙农编码与 Huffman 编码相反，采用从上到下的方法。香农-范诺编码算法步骤：

- (1) 按照符号出现的概率减少的顺序将待编码的符号排成序列。
- (2) 将符号分成两组，使这两组符号概率和相等或几乎相等。
- (3) 将第一组赋值为 0，第二组赋值为 1。
- (4) 对每一组，重复步骤 2 的操作。

例：设一副灰度级为 8 的图象中，各灰度所对应的概率分别为 0.04, 0.05, 0.06, 0.07, 0.10, 0.10, 0.18, 0.40，要求对其进行费诺.仙农编码。

灰度值	S0	S1	S2	S3	S4	S5	S6	S7
出现频率	0.40	0.18	0.10	0.10	0.07	0.06	0.05	0.04

解：根据费诺—仙农编码的方法进行分组和赋值如下图所示



所得编码结果如下表

灰度值	S0	S1	S2	S3	S4	S5	S6	S7
费诺—仙农码	00	01	100	101	1100	1101	1110	1111



## 五、 算术编码（P127）

例：编码来自 1 个 4-符号信源 $\{a_1, a_2, a_3, a_4\}$ 的由 5 个符号组成的符号序列： $b_1b_2b_3b_4b_5 = a_1a_2a_3a_3a_4$

信源符号	概率	初始子区间
$a_1$	0.2	[0, 0.2]
$a_2$	0.2	[0.2, 0.4]
$a_3$	0.4	[0.4, 0.8]
$a_4$	0.2	[0.8, 1.0]

解：由

$N_s = F_s + C_l * L$ （新子区间的起始位置=前子区间的起始位置+当前符号的区间左端\*前子区间长度）

$N_e = F_s + C_r * L$ （新子区间的结束位置=前子区间的起始位置+当前符号的区间右端\*前子区间长度）

可得，对于  $\{a_1, a_2, a_3, a_3, a_4\}$ ，有

$a_1$  [0, 0.2]  
 $a_1a_2$  [0.2\*0.2, 0.2\*0.4]=[0.04, 0.08]  
 $a_1a_2a_3$  [0.04+0.04\*0.4, 0.04+0.04\*0.8]=[0.056, 0.072]  
 $a_1a_2a_3a_3$  [0.056+0.016\*0.4, 0.056+0.016\*0.8]=[0.0624, 0.0688]  
 $a_1a_2a_3a_3a_4$  [0.0624+0.0064\*0.8, 0.056+0.0064\*1]=[0.06752, 0.0688]

解码过程 0.068

(1) 0.068 在区间[0, 0.2]，可知第一个源符号为 $a_1$

(2)  $\frac{0.068-0}{0.2} = 0.34$  在区间[0.2-0.4]中，第二个为 $a_2$

(3)  $\frac{0.34-0.2}{0.2} = 0.7$  在区间[0.4-0.8]中，第三个为 $a_3$

(4)  $\frac{0.7-0.4}{0.4} = 0.75$  在区间[0.4-0.8]中，第四个为 $a_3$

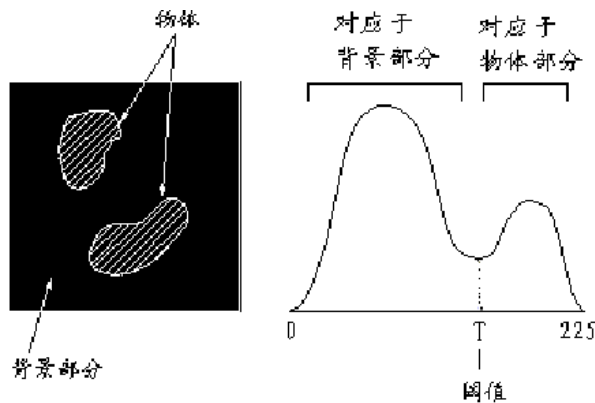
(5)  $\frac{0.75-0.4}{0.4} = 0.875$  在区间[0.8-1]中，第五个为 $a_4$

## 六、 区域分割

状态法（峰谷法、灰度阈值法）（P155）

基本思想是，确定一个合适的阈值  $T$ 。将大于等于阈值的像素作为物体或背景，生成一个二值图像。阈值的选定可以通过如下图中灰度直方图确定。

方法：首先统计最简单图像的灰度直方图，若直方图呈双峰且有明显的谷，则将谷所对应的灰度值  $T$  作为阈值，按图右侧的等式进行二值化，就可将目标从图像中分割出来。这种方法适用于目标和背景的灰度差较大、有明显谷的情况。



$$g(x, y) = \begin{cases} 0 & f(x, y) \leq T \\ 1 & f(x, y) > T \end{cases}$$

在四邻域中有背景的像素，既是边界像素。

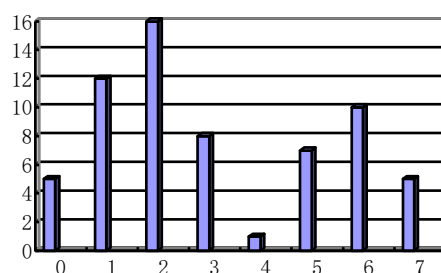
**例：对下面的图像用状态法进行二值化，并计算二值图像的欧拉数。**

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

解：（1）首先根据已知列出灰度级分布表

灰度级	0	1	2	3	4	5	6	7
像素个数	5	12	16	8	1	7	10	5

（2）画出图像的直方图



(3) 由此可确定阈值  $T=4$ ，根据  $g(x,y)=\begin{cases} 0 & f(x,y)\leq T \\ 1 & f(x,y)>T \end{cases}$ ，二值化的图像如下：

0	0	0	0	0	0	0	0
0	1	1	1	0	1	1	1
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	1
0	0	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(4) 在二值图像中，1 像素连接成分数  $C$  减去孔数  $H$  的差值叫做这幅图像的欧拉数。

本题从图中可以看出，其取值为 1 的像素的连接成分数  $C=2$ ，孔数  $H=2$ ，所以这幅图像的欧拉数为

$$E = C - H = 2 - 2 = 0$$

## 七、 区域增长

简单区域扩张法（P159）

步骤：以图像的某个像素为生长点，比较相邻像素的特征，将特征相似的相邻像素合并为同一区域；以合并的像素为生长点，继续重复以上的操作，最终形成具有相似特征的像素是最大连通集合。这种方法称简单（单一型）区域扩张法。

步骤：

- （1）从图像最左上角开始，对图像进行光栅扫描，找到不属于任何的像素。
- （2）把这个像素灰度同其周围（4 邻域或 8 邻域）不属于其他区域的像素的灰度值和已存在区域的像素灰度平均值进行比较，若灰度差值小于阈值，则合并到同一区域，并对合并的像素赋予标记。
- （3）从新合并的像素开始，反复进行（2）的操作。
- （4）反复进行（2）、（3）的操作，直至不能再合并。
- （5）返回（1）操作，寻找新区域出发点的像素。

例：对下面的图像采用简单区域生长法进行区域生长，给出灰度差值 ① $T=1$ ；② $T=3$ ；③ $T=8$  三种情况下的分割图像。

1	0	4	7	5
1	0	4	7	7
0	1	5	5	5
2	0	5	6	5
0	2	5	6	4

解：以 8 邻域为比较范围，① 当阈值  $T=1$  时，图像分割结果如下图所示 6 个区域：

1	0	4	7	5
1	0	4	7	7
0	1	5	5	5
2	0	5	6	5
0	2	5	6	4

② 当阈值  $T=3$  时，图像分割结果如下图所示为 2 个区域

1	0	4	7	5
1	0	4	7	7
0	1	5	5	5
2	0	5	6	5
0	2	5	6	4

③ 当阈值  $T=8$  时，图像分割结果如下图所示为 1 个区域

1	0	4	7	5
1	0	4	7	7
0	1	5	5	5
2	0	5	6	5
0	2	5	6	4

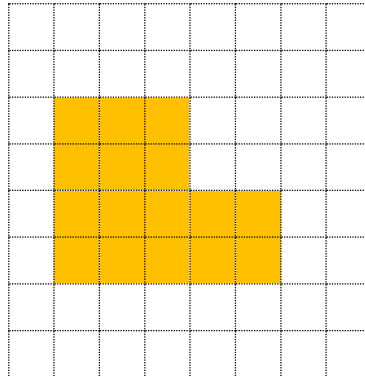
## 八、 分裂合并

分裂合并法（基于四叉树思想的方法）（P161）

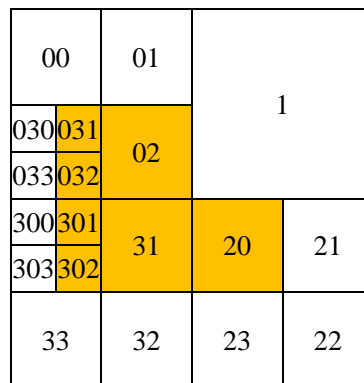
算法实现：

- 1) 对于图像中灰度级不同的区域，均分为四个子区域。
- 2) 如果相邻的子区域所有像素的灰度级相同，则将其合并。
- 3) 反复进行上两步操作，直至不再有新的分裂与合并为止。

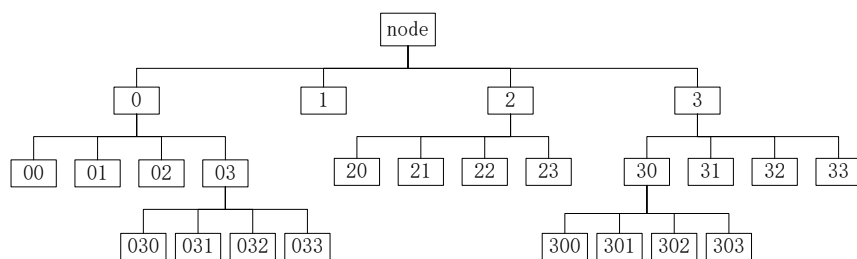
**例：用分裂合并法分割图像，并给出对应分割结果的四叉树。**



解：四叉树法分割结果如下：

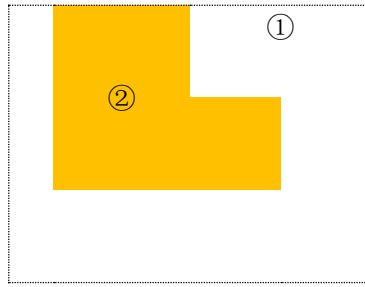


对应四叉树为：



合并后结果为：





其中背景区域①包括四叉树中 00、01、030、033、1、21、22、23、32、33、300、303 子块  
图像区域②包括四叉树中 031、032、02、20、301、302、31 子块

## 九、膨胀和腐蚀、开运算和闭运算 (P172)

### 1) 膨胀

膨胀就是把二值图像各 1 像素连接成分的边界扩大一层的处理。

**膨胀的原理:** 设二值图像为  $F$ , 结构元素为  $B$ ,  $B^s$  代表  $B$  关于原点对称的结构元素。当结构元素  $B^s$  的原点移到  $(x,y)$  处时, 结构元素用  $B_{xy}^s$  表示。则图像  $F$  被结构元素  $B$  膨胀的定义式为:

$$F \oplus B = \{x, y \mid B_{xy}^s \cap F \neq \emptyset\} \quad (\text{即 } B^s \text{ 击中 } F, \text{ 均指各自的非零像素})$$

**其含义是:** 当结构元素  $B^s$  的原点移动到  $(x,y)$  位置时, 如果  $B_{xy}^s$  所覆盖范围内的  $F$  的子图像与结构元素  $B_{xy}^s$

相应位置上至少有一个元素相同且不为 0, 则把该子图像中与  $B_{xy}^s$  的原点位置对应的  $(x,y)$  点的那个像素位置标注为 1, 否则为 0。图像  $F$  上标注出的所有这样的像素组成的集合, 即为膨胀运算的结果。

**膨胀运算的基本过程是:**

- (a) 求结构元素  $B$  关于其原点的反射集合  $B^s$ ;
- (b) 每当结构元素在目标图像  $F$  上平移后, 结构元素  $B^s$  与其覆盖的子图像中至少有一个元素相交时, 就将目标图像中与结构元素  $B^s$  的原点对应的那个位置的像素值置为“1”, 否则置为 0。

**注意:**

- (a) 结构元素中原点位置所对应的目标图像子图像位置处的值是 0 时, 仍可进行膨胀运算, 无需强求是 1。
- (b) 当结构元素在目标图像上平移时, 允许结构元素中的非原点像素超出目标图像范围。

结构元素形状对膨胀运算结果的影响: 当目标图像不变, 但所给的结构元素的形状改变时; 或结构元素的形状不变, 而其原点位置改变时, 膨胀运算的结果会发生改变。

### 2) 腐蚀 (或收缩)

腐蚀是把二值图像各 1 像素连接成分的边界点去掉从而缩小一层的处理。

**腐蚀的原理:** 设  $F$  为目标图像,  $B$  为结构元素, 则目标图像  $F$  被结构元素  $B$  腐蚀可定义为

$$F \odot B = \{x, y \mid B_{xy} \subseteq F\} \quad (\text{即 } B \text{ 包含于 } F, \text{ 均指各自的非零像素})$$

**其含义是:** 当结构元素  $B$  的原点移动到目标图像  $F$  中的  $(x,y)$  位置时, 如果  $(x,y)$  处像素值为 1, 并且  $B_{xy}$  所覆盖范围内的  $F$  的子图像的其他像素能够包含  $B_{xy}$  的其他像素或与  $B_{xy}$  的其他像素完全相同, 则保留该子图像中与  $B_{xy}$  的原点位置对应的  $(x,y)$  点的像素值 1, 否则均为 0。图像  $F$  上保留的所有这样值为 1 的像素组成的集合, 即为腐蚀运算的结果。

这里的“包含”是指结构元素  $B$  和目标图像  $F$  的子图像中值为 1 的像素两两之间的对应关系。

**腐蚀运算的基本过程是:**

把结构元素  $B$  看作为一个卷积模板, 每当结构元素平移到其原点位置与目标图像  $F$  中那些像素值为“1”的位置重合时, 就判断被结构元素覆盖的子图像的其他像素的值是否都与结构元素相应位置的像素值相同; 只有当其都相同时, 就将结果图像中的那个与原点位置对应的像素位置的值置为“1”, 否则置为 0。

**注意:**

- (a) 结构元素中的原点位置处的像素值可以不为 1, 但要求目标图像中的子图像与结构元素  $B$  的原点对应的那个位置的像素值是 1。
- (b) 当结构元素在目标图像上平移时, 结构元素中的任何元素不能超出目标图像的范围。

腐蚀运算的结果不仅与结构元素的形状(矩形、圆形、菱形等)选取有关, 而且还与原点位置的选取有关。

### 3) 膨胀和腐蚀的作用:

膨胀的作用是使孔洞收缩, 目标扩大。对消除图像目标中的小颗粒噪声和填补凹陷非常有效。

腐蚀的作用是使目标收缩, 孔洞扩大。对去除图像小颗粒噪声和目标之间的粘连非常有效。

### 4) 开运算 V.S. 闭运算

**A、开运算 (先腐蚀再膨胀):**  $F \circ B = (F \odot B) \oplus B$

作用: 光滑目标轮廓、消除小目标 (如去掉毛刺和孤立点等), 在纤细点处分离物体, 同时并不明显改变目

标面积；

**B、闭运算（先膨胀再腐蚀）：**  $F \bullet B = (F \oplus B) \odot B$

作用：在保持原目标的大小与形态的同时，填充凹陷、弥合孔洞和裂缝。

例 1：用结构元素  $B$  对目标图像  $F$  进行腐蚀运算。结构元素  $B$  中红色为原点。

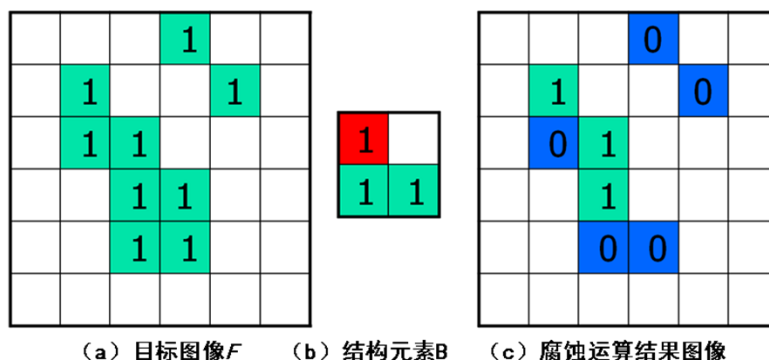


图8.4 腐蚀运算实例

例 2：结构元素不同时的腐蚀运算

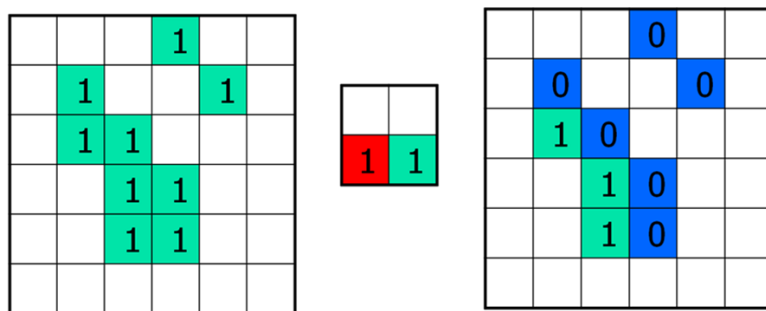


图8.5 与图8.4结构元素不同时的腐蚀运算实例

例 3：结构元素原点不同时的腐蚀运算

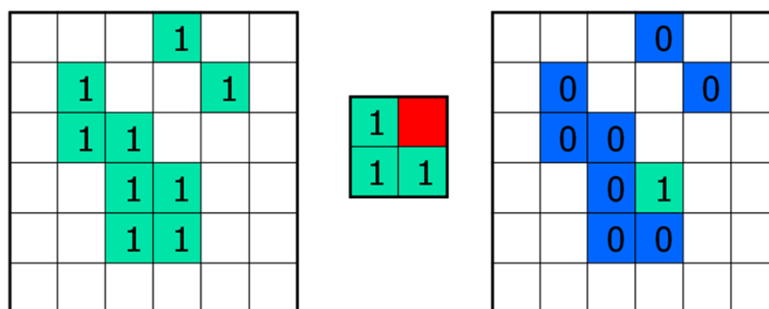


图8.6 与图8.4的结构元素的原点不同时的腐蚀运算实例

例 4：用结构元素  $B$  对目标图像  $F$  进行膨胀运算。结构元素  $B$  中红色为原点。



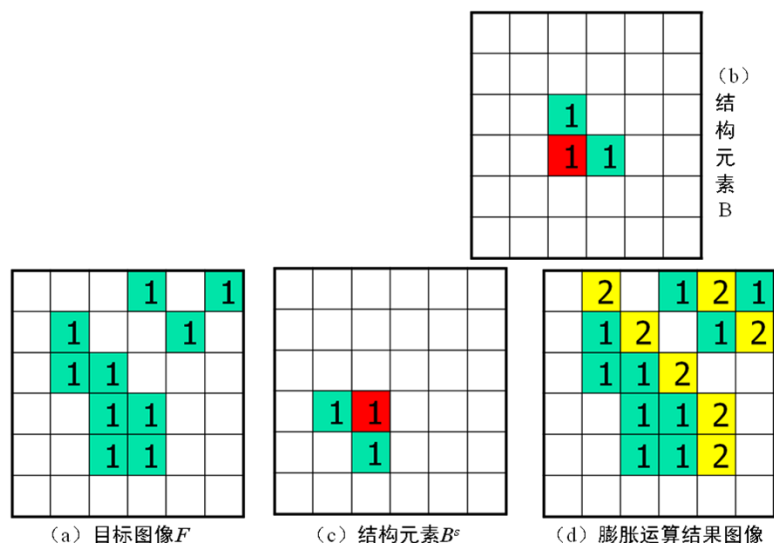
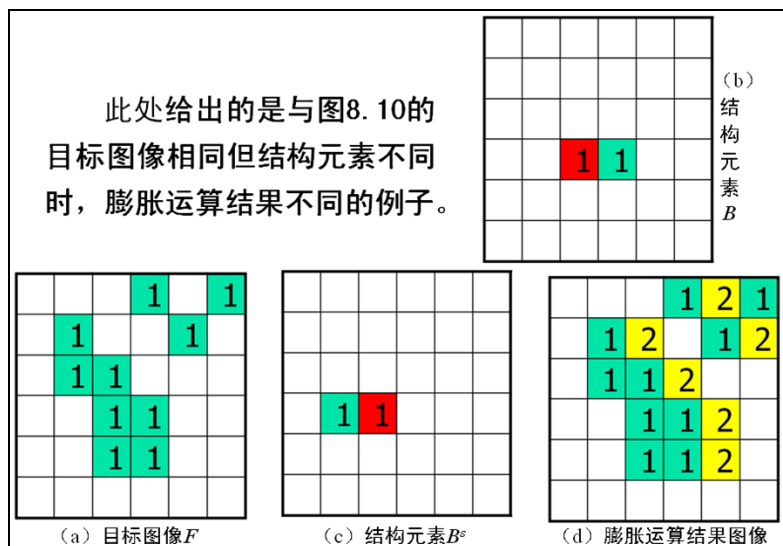


图 8.10 膨胀运算实例

例 5



例 6

