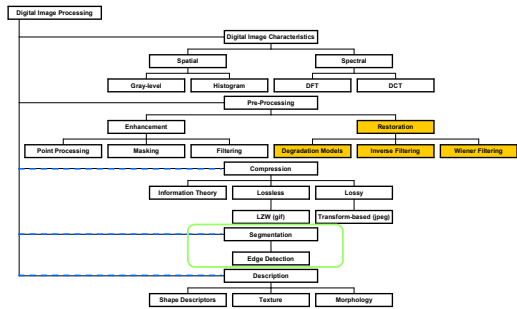


Digital Image Processing: Details



- 1 图像分割定义与策略
- 2 点检测
- 3 线检测
- 4 边缘检测
- 5 基于跟踪的图像分割
- 6 Hough变换
- 7 基于灰度阈值的分割方法
- 8 空间聚类
- 9 区域生长
- 10 分裂合并法

1、图像分割定义

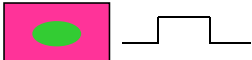
- 图像分割就是把图像划分为不同的区域



——感兴趣部分称为目标或前景（其余称为背景）

图像分割策略

- 基于亮度值的两个基本特性：
- (1) 区域之间的不连续性
- 边界方法，找图像中有区域间突变的边缘。
- 找突变的点、线、边

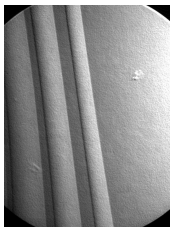


- (2) 区域内部的相似性
- 区域方法：子区域同质性，不同区域异质。
- 通过阈值，区分出不同亮度值子区域。
- 区域外轮廓即是分割目标的边界



2、点检测

- 计算与周边邻域像素加权差值



图像

-1	-1	-1
-1	8	-1
-1	-1	-1

模板



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

442

3、线检测



水平模板

45度模板

垂直模板

135度模板



HIT-Visual Intelligence Lab



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

443

4、边缘检测



截面图



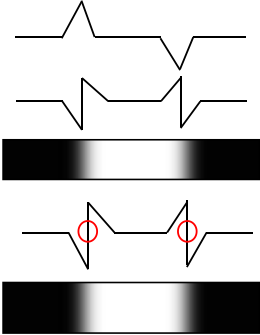
HIT-Visual Intelligence Lab



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

444


- 一阶导数:
- 二阶导数:



——一阶导数的极大值，二阶导数过零点



HIT-Visual Intelligence Lab



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

445

-1

1

-1

1

-1

0

0

-1

0

1

1

0

Roberts

-1

-1

-1

-1

0

0

0

0

1

1

1

1

-1

0

1

-1

0

1

-1

0

1

-1

0

1

Prewitt

0

-1

0

-1

-1

-1

-1

4

-1

-1

8

-1

0

-1

0

-1

-1

-1

拉普拉斯算子

-1

-2

-1

-1

0

0

0

2

0

2

-1

0

1

1

2

1

-1


0

1

Sobel



HIT-Visual Intelligence Lab



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

446



Laplacian二阶边缘检测算子的边缘检测示例



HIT-Visual Intelligence Lab



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

447

高斯拉普拉斯（LOG）







0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



HIT-Visual Intelligence Lab

2

哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

448

平滑后的边缘检测

原始图像

水平梯度部分

原始图像

水平梯度部分



垂直梯度部分

组合边缘图像

垂直梯度部分

组合边缘图像

HIT-Visual Intelligence Lab

哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

449

5、基于跟踪的图像分割

• 轮廓跟踪：检测的边缘点连接成线

• (1) 轮廓跟踪法

边界跟踪函数：bwboundaries

——一种适用于二值图像的图像分割方法

HIT-Visual Intelligence Lab

哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

450

轮廓跟踪算法

• 步骤1：扫描图像，寻找 A_0 。初始化扫描方向变量 dir （记录上一个边界点到当前边界点的移动方向）。

• 步骤2：按逆时针方向搜索当前像素的 3×3 邻域，搜索到的第一个与当前像素值相同的像素便为新的边界点 A_n ，同时更新变量 dir 为新的方向值。

• (i) 对4连通区域取 $dir = 3$ ，取 $(dir + 3) \bmod 4$ 。

$dir=0 \quad (0+3) \bmod 4 = 3$

$dir=1 \quad (1+3) \bmod 4 = 0$

$dir=2 \quad (2+3) \bmod 4 = 1$

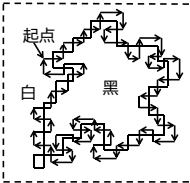
$dir=3 \quad (3+3) \bmod 4 = 2$

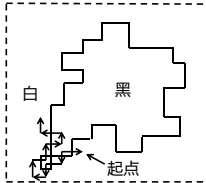
HIT-Visual Intelligence Lab

哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

451





(a)某些小凸部分可能被漏掉

(b)利用不同起点跟踪小凸部分

轮廓跟踪法示例

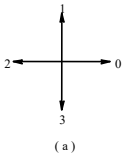
HIT-Visual Intelligence Lab

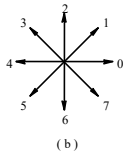
哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

452

• (ii) 对8连通区域取 $dir = 7$ 。若 dir 为奇数取 $(dir + 7) \bmod 8$ ，若 dir 为偶数取 $(dir + 6) \bmod 8$ 。





(a)

(b)

4方向与8方向

HIT-Visual Intelligence Lab

哈尔滨工业大学

HARBIN INSTITUTE OF TECHNOLOGY

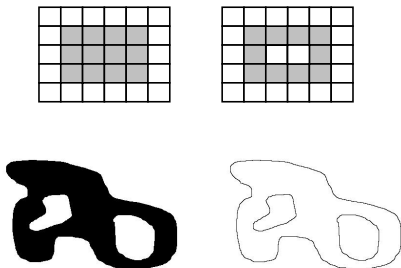
453

• 步骤3：如果 A_n 等于第二个边界点 A_1 且边界点 A_{n-1} 等于第一个边界点 A_0 ，则停止搜索，结束跟踪，否则重复步骤2继续搜索。

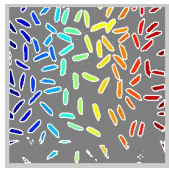
• 步骤4：由边界点 A_0 、 A_1 、 A_2 、...、 A_{n-2} 构成的边界便为要跟踪的边界。

• 轮廓提取：如果原图像中有一点为黑，且它的8个邻点都是黑色时，说明该点是内部点，将该点删除。

HIT-Visual Intelligence Lab



```
• I = imread('rice.png');
• BW = im2bw(I, graythresh(I));
• [B,L] = bwboundaries(BW,'noholes');
• imshow(label2rgb(L, @jet, [.5 .5 .5]))
• hold on
• for k = 1:length(B)
•     boundary = B{k};
•     plot(boundary(:,2), boundary(:,1), 'w',
'LineWidth', 2)
• end
```



(2) 光栅跟踪法

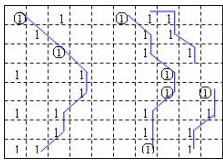
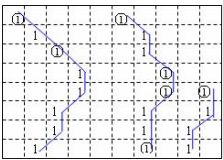
- 光栅跟踪方法的基本思想是：
- 先利用检测准则确定接受对象点
- 然后根据已有的接受对象点和跟踪准则确定新的接受对象点
- 最后将所有标记为1且相邻的对象点联接起来就得到了检测到的细曲线。

光栅跟踪算法

- 先确定检测阈值 d 、跟踪阈值 t ，且要求 $d > t$ 。
- 检测准则：对图像逐行扫描，将每一行中灰度值大于或等于检测阈值 d 的所有点（称为接受对象点）记为1。
- 跟踪准则：设位于第 i 行的点 (i, j) 为接受对象点，如果位于第 $i+1$ 行上的相邻点 $(i+1, j-1)$ 、 $(i+1, j)$ 和 $(i+1, j+1)$ 的灰度值大于或等于跟踪阈值 t ，就将其确定为新的接受对象点，并记为1。
- 重复以上过程，直至图像中除最末一行以外的所有接受点扫描完为止。

$d=7, t=4$

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



(a) 跟踪过程和检测结果 (b) 直接取阈值为4时的检测结果

6、Hough变换

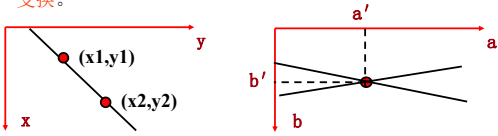
- 在找出边界点集之后，需要连接，形成完整的边界图形描述。
- 它的主要优点在于受噪声和曲线间断的影响较小



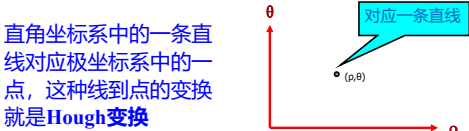
- Hough**：对图像进行Hough变换
- Houghpeaks**：用来提取Hough变换后参数平面上的峰值点。
- Houghlines**：用于在图像中提取参数平面上的峰值点对应的直线

- 点线对偶性**：
 - XY空间的一条直线，对应于参数空间PQ的一个点；同理参数空间PQ的一条直线，对应于XY空间的一个点。
 - XY空间的一条蓝色直线 $y = px + q$ 对应于参数空间一点 (p, q)
 - 参数空间的一条直线 $q = -xp + y$ 对应于XY空间一点 (x, y)

- Hough变换的基本思想**：
 - 在 xy 平面内的一条直线可以表示为： $y = ax + b$
 - 将 a 、 b 作为变量， ab 平面内直线可以表示为： $b = -xa + y$
 - 如果点 $(x1,y1)$ 与点 $(x2,y2)$ 共线，那么这两点在参数 ab 平面上的直线将有一个交点
 - 在参数 ab 平面上相交直线最多的点，对应的 xy 平面上的直线就是我们的解。这种从线到点的变换就是 **Hough变换**。

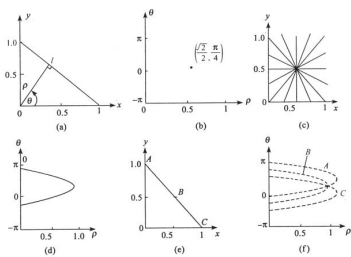


- 极坐标系Hough变换的基本思想**：
 - 对于边界上的 n 个点的点集，找出共线的点集和直线方程。
 - 对于直角坐标系中的一条直线 l ，可用 ρ 、 θ 来表示该直线，且直线方程为： $\rho = x \cos \theta + y \sin \theta$
 - 其中， ρ 为原点到该直线的垂直距离， θ 为垂线与 x 轴的夹角，这条直线是唯一的。
 - 构造一个参数 ρ - θ 的平面，从而有如下结论：



直角坐标系中的一条直线对应极坐标系中的一点，这种线到点的变换就是Hough变换

- 参数平面为 ρ, θ 对应不是直线而是正弦曲线
- 找出相交线段最多的参数空间的点，再根据该点求出对应的 xy 平面的直线段

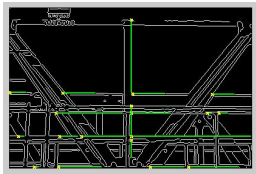
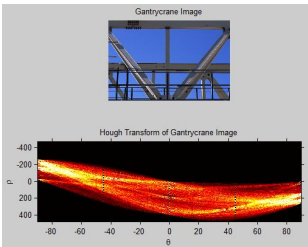


- 算法步骤**：
 - 在 ρ 、 θ 的取值范围内对其分别进行 m 、 n 等分，设一个二维数组的下标与 ρ_i 、 θ_j 的取值对应；
 - 对图像上的所有边缘点作 Hough 变换，求每个点在 θ_j ($j=0, 1, \dots, n$) Hough 变换后的 ρ_i ，判断 (ρ_i, θ_j) 与哪个数组元素对应，则让该数组元素值加1；
 - 比较数组元素值的大小，**最大值** 所对应的 (ρ_i, θ_j) 就是这些共线点对应的直线方程的参数。

Hough函数检测图像中的直线

- RGB = imread('gantrycrane.png');
- I = rgb2gray(RGB); % Convert to intensity.
- BW = edge(I,'canny'); % Extract edges.
- [H,T,R] = hough(BW,'RhoResolution',0.5,'ThetaResolution',0.5);
- % Display the original image.
- subplot(2,1,1);imshow(RGB);title('Gantrycrane Image');
- % Display the Hough matrix.
- subplot(2,1,2);
- imshow(imadjust(mat2gray(H)),'XData',T,'YData',R,...
- 'InitialMagnification','fit');
- title('Hough Transform of Gantrycrane Image');
- xlabel('\theta'), ylabel('\rho');
- axis on, axis normal, hold on;
- colormap(hot);

- P = houghpeaks(H, 5, 'threshold', ceil(0.3 * max(H(:))));
- lines = houghlines(BW, T, R, P, 'FillGap', 5, 'MinLength', 7);
- figure,imshow(BW),hold on;
- max_len=0;
- for k = 1 : length(lines)
- xy = [lines(k).point1; lines(k).point2];
- plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
- plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
- plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
- len = norm([lines(k).point1-lines(k).point2]);
- if (len > max_len)
- max_len = len;
- xy_long = xy;
- end
- end



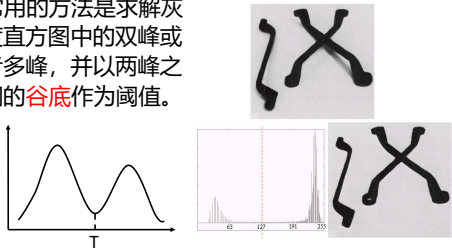
应用Hough变换对倾斜表格图像纠偏

7、基于灰度阈值的分割方法

- 基于阈值的图像分割适用于那些物体（前景）与背景在灰度上有较大差异的图像分割问题。
- 基于阈值的图像分割方法是提取物体与背景在灰度上的差异，把图像分为具有不同灰度级的目标区域和背景区域的一种图像分割技术。
- graythresh: 基于Otsu法求取灰度阈值
- watershed: 实现基于分水岭算法的图像分割

7、基于灰度阈值的分割方法

常用的方法是求解灰度直方图中的双峰或者多峰，并以两峰之间的谷底作为阈值。



◆ 阈值分割

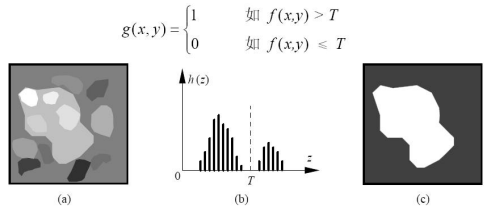
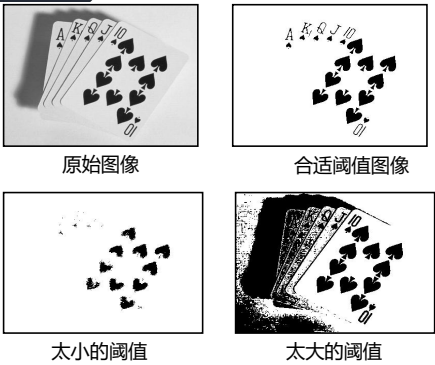


图 7.5.1 单阈值分割图示

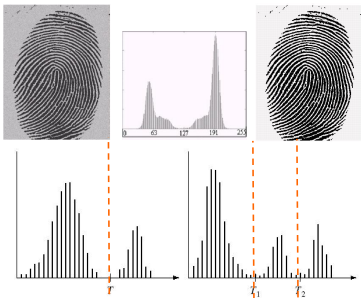


7.1 基本的全局阈值算法

- 全局阈值是指整幅图像使用同一个阈值做分割处理，并产生一个二值图，区分出前景对象和背景。适用于背景和前景对比度大的图像。
- 算法实现：
 - 选取一个合适的阈值T，逐行扫描图像：灰度值大于T，置为255；灰度值小于T，置为0
 - 基本的迭代式阈值 T 可以按如下计算：
 - 1、选择一个初始估计值T（一般为图像的平均灰度值）

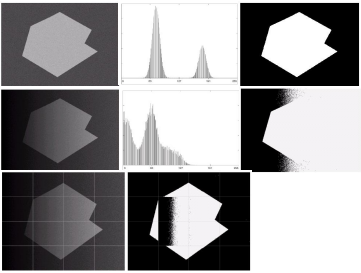
- 2、使用T分割图像，产生两组像素：G1包括灰度值大于 T 的像素，G2包括灰度值小于等于 T 的像素
- 3、计算G1 中像素的平均值并赋值给μ₁，计算G2 中像素的平均值并赋值给μ₂
- 4、计算一个新的阈值：
$$T = \frac{\mu_1 + \mu_2}{2}$$
- 5、重复步骤 2 - 4，一直到两次连续的T之间的差小于预先给定的上界M。

单阈值只能对双峰直方图效果较好。对于其它类型的直方图，需要更多的阈值



7.2 单值阈值和光照

- 不均匀的光照会使单值阈值方案失效。
- 解决单值阈值无法工作的一个方法是 将图像分割为子图像，并分别进行阈值化处理
- 由于每个像素的阈值依赖于其在图像中的位置，因此称为自适应 (adaptive) 阈值



我们看到图像得到了改善，但是需要对出错的图像进行进一步的细分，从而得到更好的效果。

7.3 大津阈值分割

大津阈值法可以自动寻找阈值，对图像进行划分，将目标物和背景区分开来。

把直方图在某一阈值处分割成两组，当被分成的两组间方差为最大时，确定阈值。例如，设一幅图像的灰度值为 $1 \sim m$ 级，灰度值 i 的像素数为 n_i ，此时我们得到：

总像素数
$$N = \sum_{i=1}^m n_i$$

各灰度值的归一化值
$$p_i = \frac{n_i}{N}$$

然后用 k 将其分成两组 $C_0 = \{1 \sim k\}$ 和 $C_1 = \{k+1 \sim m\}$

各组产生的概率如下：

C_0 产生的概率
$$\omega_0 = \sum_{i=1}^k p_i = \omega(k)$$

C_1 产生的概率
$$\omega_1 = \sum_{i=k+1}^m p_i = 1 - \omega(k)$$

C_0 的平均值
$$\mu_0 = \frac{\sum_{i=1}^k ip_i}{\omega_0} = \frac{\mu(k)}{\omega(k)}$$

C_1 的平均值
$$\mu_1 = \frac{\sum_{i=k+1}^m ip_i}{\omega_1} = \frac{\mu - \mu(k)}{1 - \omega(k)}$$

其中 $\mu = \sum_{i=1}^m ip_i$ 是整体图像的灰度平均值；

$\mu(k) = \sum_{i=1}^k ip_i$ 是阈值为 k 时的灰度平均值

两组直方图的灰度平均值为

$$\mu = \omega_0 \mu_0 + \omega_1 \mu_1$$

两组间的方差用下式求出

$$\begin{aligned} \sigma^2(k) &= \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu)^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \\ &= \frac{[\mu \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \end{aligned}$$

证明此式存在

$$k^* = \arg \max_k \sigma^2(k)$$

k^* 值便是阈值，不管图像的直方图有无明显的双峰，都能得到较满意的结果。

8、空间聚类

- 特征空间聚类 (feature space clustering) 的方法是将图像空间的元素按照特征值用对应的特征空间点表示，通过将特征空间点聚集成不同的类团，然后再将它们划分开。
- 聚类方法也是一种全局的方法，比仅基于边缘检测的方法更抗噪声。

8.1 K-均值聚类

- 将一个特征空间分成 K 个聚类的常用方法是 **K-均值法**(K-means)。令 $x = (x_1, x_2)$ 代表特征空间的坐标， $g(x)$ 代表在这个位置的特征值，K-均值法是要**最小化**如下指标

- $$E = \sum_{i=1}^K \sum_{x \in Q_j^{(i)}} \|g(x) - \mu_j^{(i+1)}\|^2 \quad (1)$$

- 其中 $Q_j^{(i)}$ 代表在第 i 次迭代后赋给类 j 的特征点集合， μ_j 表示第 j 类的均值。式 (1)给出**每个特征点与其对应类均值的距离和**。

• K-均值法计算步骤如下:

- (1) 任意选取 K 个初始类均值， $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)}$;
- (2) 在第 i 次迭代时，根据下述准则将每个特征点都赋给 K 类之一 ($j=1, 2, \dots, K, i=1, 2, \dots, K, j \neq i$)，即

$$x \in Q_j^{(i)} \text{ 如果 } \|g(x) - \mu_j^{(i)}\| < \|g(x) - \mu_{j'}^{(i)}\|$$

- 即将每个特征点赋给均值离它最近的类。

- (3) 对 $j=1, 2, \dots, K$, 更新类均值 $\mu_j^{(i+1)}$

$$\mu_j^{(i+1)} = \frac{1}{N_j} \sum_{x \in Q_j^{(i)}} g(x)$$

- 其中 N_j 是 $Q_j^{(i)}$ 中的特征点个数。
- (4) 如果对所有的 $j=1, 2, 3, \dots, K$, 有 $\mu_j^{(i+1)} = \mu_j^{(i)}$, 则算法收敛，结束；否则退回到步骤(2)继续下一次迭代。

8.2 ISODATA 聚类

- 主要步骤如下:
- (1) 设定 N 个聚类中心位置的 **初始值**;
- (2) 对每个特征点求取离其 **最近的聚类中心** 位置，通过赋值把特征空间分成 N 个区域;
- (3) 分别计算属于各聚类模式的 **平均值**;
- (4) 将最初的聚类中心位置与步骤(3)得到的新平均值进行比较，如果相同则停止，如果不同则返回步骤(2)继续进行。

9、区域生长

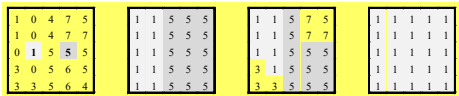
- 从单个像素出发，逐渐合并以形成所需的分割区域，称为**区域生长**。
- 区域生长的基本思想是**将具有相似性质的像素结合起来构成区域**。
- 具体是先对每个需要分割的区域找一个**种子像素**作为生长的起点，然后将种子像素周围邻域中与种子像素相同或相似性质的像素（根据某种事先确定的生长或相似准则来判定）合并到种子像素所在的区域中。

- 将这些像素作为新的种子像素继续进行上面的过程，直到再没有满足条件的像素可被包括进来。这样一个区域就长成了。

• 基于区域灰度差准则的主要步骤:

- (1) 对图像进行逐行扫描，找出尚没有归属的像素;
- (2) 以该像素为中心检查它的邻域像素，即将邻域中的像素逐个与它比较，如果灰度差小于预先确定的阈值，则将它们合并;

- (3) 以新合并的像素为中心，返回步骤(2)，检查新像素的邻域，直到区域不能进一步扩张;
- (4) 返回到步骤(1)，继续扫描直到不能发现没有归属的像素，则结束整个生长过程。



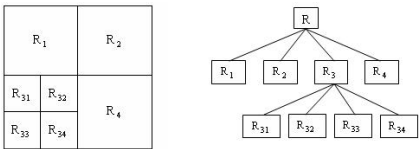
原始图 T=3 T=2 T=7

10. 分裂合并法

- 分裂-合并分割法是从整个图像出发，根据图像和各区域的不均匀性，把图像或区域分裂成新的子区域；根据毗邻区域的均匀性，把毗邻的子区域合并成新的较大区域。

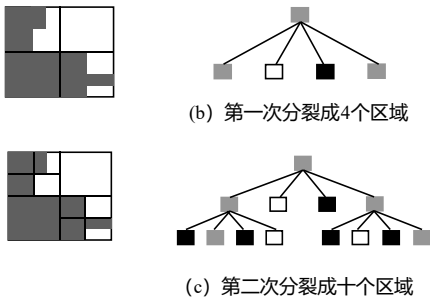
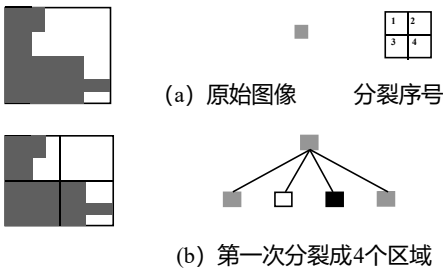
- 1. 图像四叉树
- 如果把整幅图像分成大小相同的4个方形象限区域，并接着把得到的新区域进一步分成大小相同的4个更小的象限区域，如此不断继续分割下去，就会得到一个以该图像为树根，以分成的新区域或更小区域为中间结点或树叶结点的四叉树。

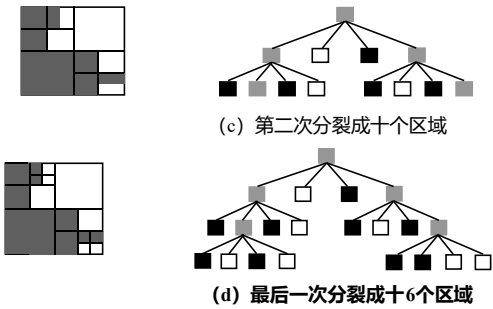
分裂合并法



- (a) 图像R (b) 图像R的四叉树示例
图像的四叉树表示

- 2. 分裂-合并分割法
- 设同一区域 R_i 中的所有像素满足某一相似性准则时， $P(R_i) = \text{TRUE}$ ，否则 $P(R_i) = \text{FALSE}$ 。
- (1) 将图像 R 分成4个大小相同的象限区域 R_i ， $i=1, 2, 3, 4$ ；
- (2) 对于任何的 R_i ，如果 $P(R_i) = \text{FALSE}$ ，则将该 R_i 再进一步拆分成4个更小的象限区域；
- (3) 如果此时存在任意相邻的两个区域 R_j 和 R_k 使 $P(R_j \cup R_k) = \text{TRUE}$ 成立，就将 R_j 和 R_k 进行合并；
- (4) 重复 (2) 和 (3)，直到无法进行拆分和合并为止。





- Matlab提供了实现四叉树分解算法功能的函数 `qtdecomp`。
- 该函数调用语法格式为：
`S=qtdecomp (F)`
- 其功能是对灰度图像F执行四叉树分解，返回四叉树结构存于稀疏矩阵

分裂合并法

- 对于灰度图像的一些可以选择的分裂-合并准则：
 - (1) 同一区域中最大灰度值与最小灰度值之差或方差小于某选定的阈值；
 - (2) 两个区域的平均灰度值之差及方差小于某个选定的阈值；
 - (3) 两个区域的灰度分布函数之差小于某个选定的阈值；
 - (4) 两个区域的某种图像统计特征值的差小于等于某个阈值。