第5章 二值图像分析

计算机科学与技术学院

本次课程内容

1. 什么是二值图像
2. 二值图像的创建
3. 二值图像的表达方法
4. 利用二值图像特征计算
5. 二值图像的处理
6. 二值图像的应用实例

1. 什么是二值图像

- 什么是二值图像?

  只有黑白两个灰度等级的图像。

二值图像实例

使用二值图像处理的优点:

去掉无关信息的干扰

几何与拓扑特性的表示与分析

节省资源

算法非常简单，容易理解和实现，并且计算速度很快

二值视觉所需的内存小，对计算设备要求低

使用二值图像的优点

- 去掉无关信息的干扰，不考虑颜色等干扰信息
- 几何与拓扑特性的表示与分析

  例如，适用于形状分析
- 二值视觉所需的内存小，对计算设备要求低，节省资源

  如果256个灰度等级，每像素占8位

  二值图像，每像素占一个位bit
- 算法非常简单，容易理解和实现，并且计算速度很快

2. 二值图像的创建

- 如果彩色图像：先灰度化，再二值化
- 如果灰度化：直接二值化

彩色图像灰度化方法

- 在NTSC美制电视制亮度规范中，灰度强度计算：

$$Y = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}$$

此方法常用

- 在PAL电视制亮度规范中，灰度强度计算：

$$Y = 0.222\mathrm{R} + 0.707\mathrm{G} + 0.071\mathrm{B}$$

每个像素的RGB都需要转换

灰度图像二值化方法

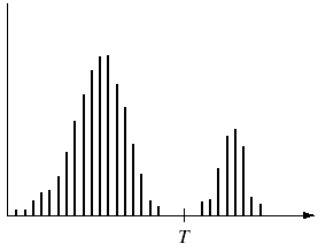
- 二值化：经常利用阈值分割，从图像分出中前景和背景，实现二值化

- 阈值确定的常用方法：
  - 简单阈值：直方图
  - 全局阈值迭代方法
  - OTSU方法（最大类间方差法）

利用阈值二值化—— 直方图确定阈值方法

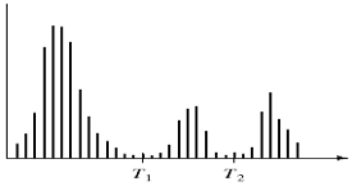
- 单阈二值化，所有像素都使用相同的阈值，适用于直方图双峰值情况

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

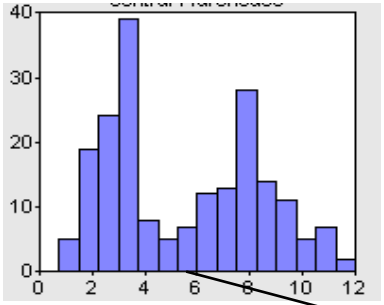


- 多阈值二值化，所有像素都使用相同的阈值，适用于直方图多峰值情况

$$g(x,y)=\begin{cases}1 & T_{k-1}<f(x,y)\le T_k\\ 0 & f(x,y)\le T\end{cases}$$



单阈值二值化实例



直方图

阈值

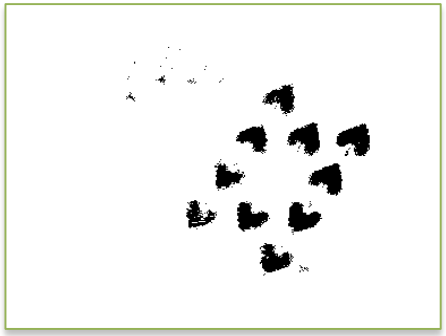
图像二值化结果

二值化理想结果

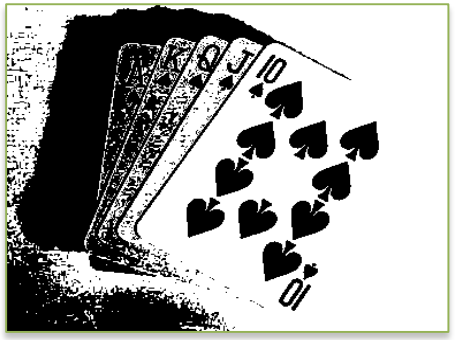
二值化不理想结果

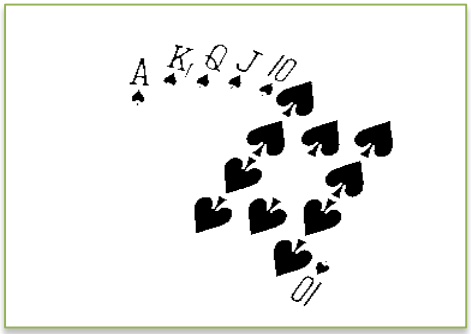
图像二值化结果

灰度图像



二值化不理想结果





二值化理想结果

确定阈值方法——全局阈值迭代方法

1. 选择一个初始化的阈值T (通常取灰度值的平均值)
2. 使用阈值T 将图像的像素分为两部分: G1包含灰度满足大于T， G2包含灰度满足小于T。
3. 计算G1中所有像素的均值μ1，以及G2中所有像素的均值μ2
4. 计算新的阈值：

$$T = \frac{\mu_1 + \mu_2}{2}$$

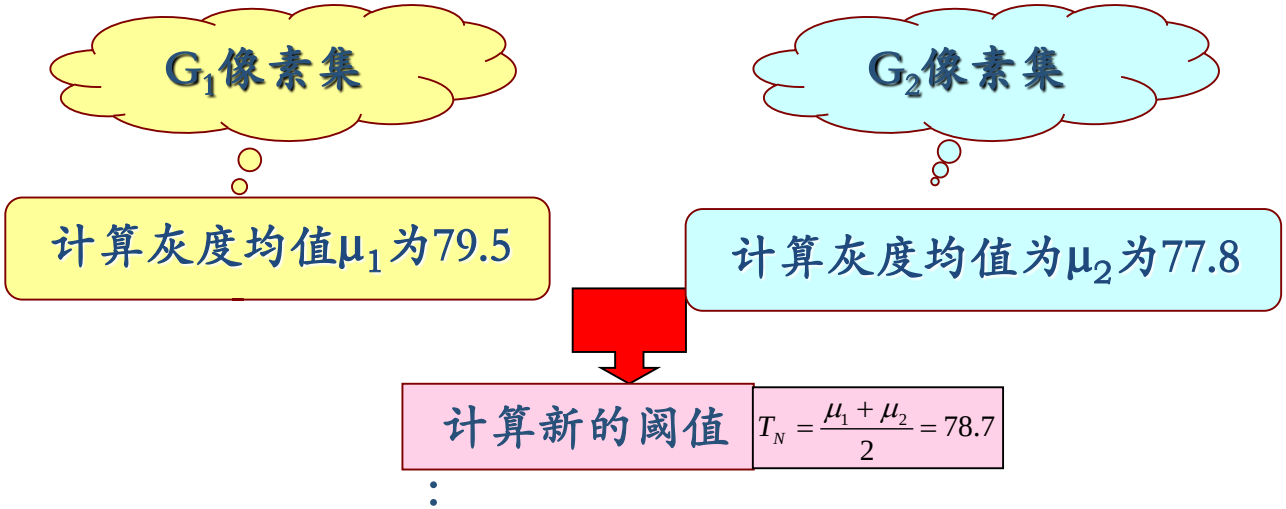
5. 重新步骤2-4，直到在前后计算的结果小于一个预先确定的阈值为止

全局阈值迭代计算的实例

例如：假设阈值误差E_T（表示相继两次计算阈值间的误差）为0.5，初始化的阈值T_P为78。

那么，根据上述步骤：

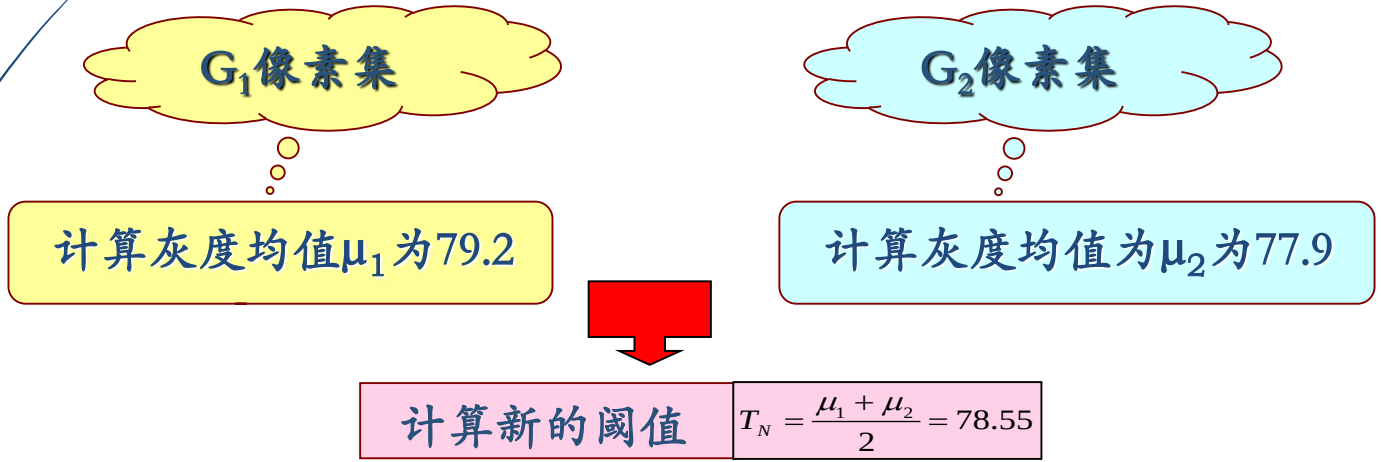
（1）利用78将全部像素分为两个集合G_1和G_2



全局阈值迭代计算的实例（续）

（2）计算$|T_P-T_N|=0.7$ 因为$0.7>0.5$（当前阈值还不够好），因此，继续进行下一次迭代过程，并且把78.7作为当前阈值，即T_P为78.7

（3）利用78.7将全部像素分为两个集合G_1和G_2



全局阈值迭代计算的实例

（4）计算$|T_P-T_N|=0.15$ 因为$0.15<0.5$, 因此，即相邻两次计算的阈值相差很小，小于预先指定的阈值误差0.5, 终止迭代过程。

（5）当前阈值$T_N=$ 78.55就是迭代两次计算得到的结果。

利用78.55进行分割，任一像素：

- 如果灰度大于78.55，则为前景像素
- 如果灰度小于78.55，则为背景像素

3. 二值图像的表达方法

- 游程长度编码(run-length encoding)
  - 用图像像素值连续为1的个数（像素1的长度）来描述图像.
  - 已被用于图像传输. 另外，图像的某些性质，如物体区域面积，也可以从游程长度编码直接计算出来
- 在游程长度编码中经常运用两种方法:
  - 使用1的起始位置和1的游程长度
  - 使用0和1的游程长度描述

实例回顾

1的游程（2, 2） （6, 3） （13, 6） （20, 1）
（4, 6） （11, 10）
（1, 5 ） （11， 1） （17， 4）

1和0的游程长度：0，2，2，3，4，6，1，1
0，3，6，1，10
5，5，1，5，4

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

4. 利用二值图像特征计算

- 利用二值图像可以计算以下图像特征：
  - 尺寸（面积）
  - 位置（质心）
  - 目标方向及伸长率
  - 密集度（A/p^2）：散布性或密集性度量方法
  - 体态比：最小外接矩形长宽比

（1）面积及位置的计算

- 二值图像前景部分的面积（零阶矩）计算：

$$A=\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}B[i,j]$$

如果B[i, j]是二值图像

- 质心位置计算为：

$$\bar{x}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}B[i,j]=\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}jB[i,j]$$

$$\bar{y}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}B[i,j]=-\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}iB[i,j]$$

$\bar{x}$ 和 $\bar{y}$ 是物体的中心位置的坐标.

$$\bar{x}=\frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}jB[i,j]}{A}$$

$$\bar{y}=-\frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}iB[i,j]}{A}$$

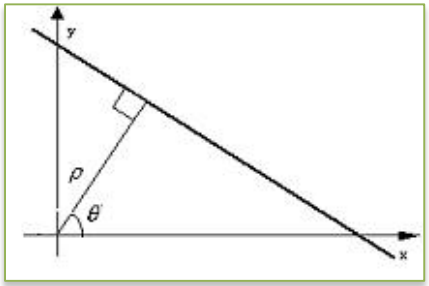
（2）二值图像目标的方向及伸长率

- 假设二值图像前景是一个长形状的目标，长轴方向被定义为物体的方向
- 计算物体方向的原理：
  - 利用物体目标全部前景点到直线的距离之和最小。
  - 距离之和计算为

$$\chi^2 = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} r_{ij}^2 B[i,j]$$

r_{ij} 是物体点 $[i,j]$ 到直线的距离.

二值图像方向计算



一般把直线表示成极坐标形式：

$$\rho = x\cos\theta + y\sin\theta$$

θ是直线的法线与x轴的夹角

ρ是直线到原点的距离.

直线的极坐标方程得出距离 r

$$r^2 = (x\cos\theta + y\sin\theta - \rho)^2$$

求极小化问题 ：

$$\chi^2 = a\cos^2\theta + b\sin\theta\cos\theta + c\sin^2\theta$$

$$a = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(x_{ij} - \bar{x})^2 B[i,j]$$

$$b = 2\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(x_{ij} - \bar{x})(y_{ij} - \bar{y})B[i,j]$$

$$c = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(y_{ij} - \bar{y})^2 B[i,j]$$

二值图像方向计算

$$\chi^2 = \frac{1}{2}(a+c) + \frac{1}{2}(a-c)\cos 2\theta + \frac{1}{2}b\sin 2\theta$$

- 对χ^2微分，并置微分结果为零，求解θ 值：

$$\tan 2\theta = \frac{b}{a-c}$$

- 惯性轴的方向由下式给出：

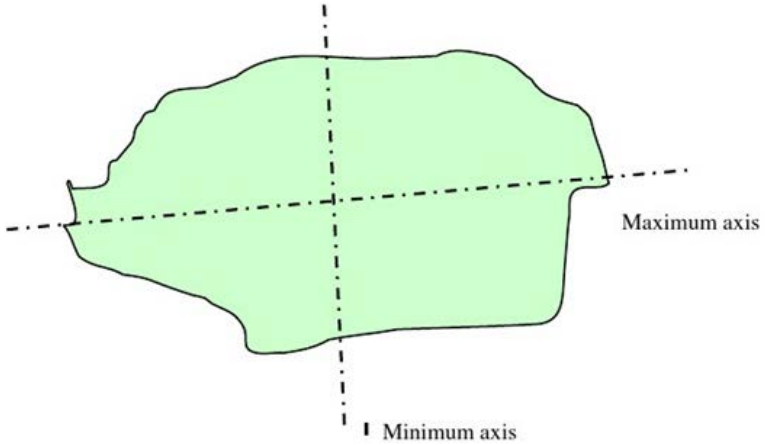
$$\sin 2\theta = \pm\frac{b}{\sqrt{b^2 + (a-c)^2}}$$

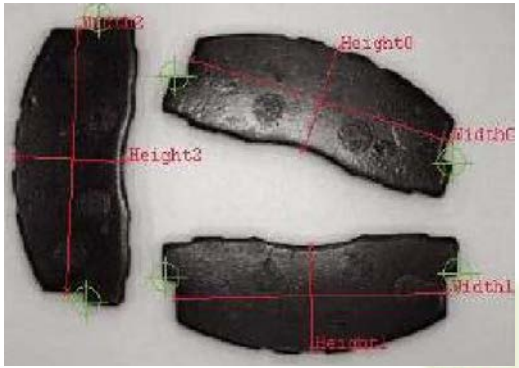
$$\cos 2\theta = \pm\frac{a-c}{\sqrt{b^2 + (a-c)^2}}$$

伸长率

物体的伸长率E是$\chi^{'}$的最大值与最小值之比 $E=\frac{\chi_{\max}}{\chi_{\min}}$

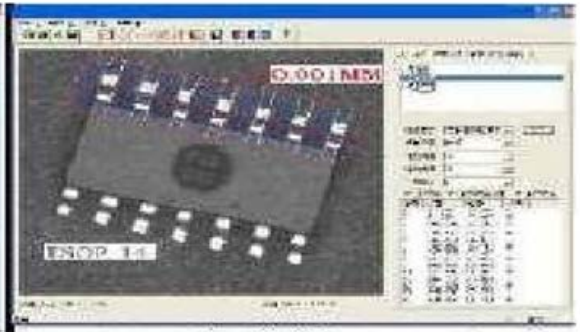
取最大与最小时，对应的方向称为最大最小惯性轴。





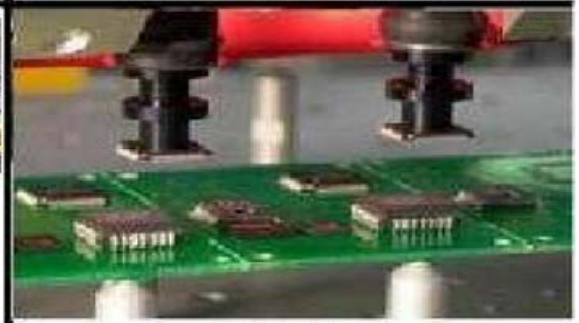
应用

视觉定位



间距测量

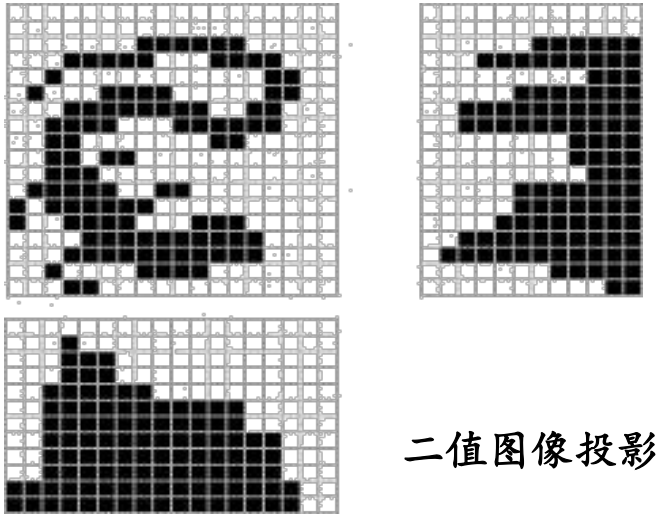
视觉检测



方向辨别

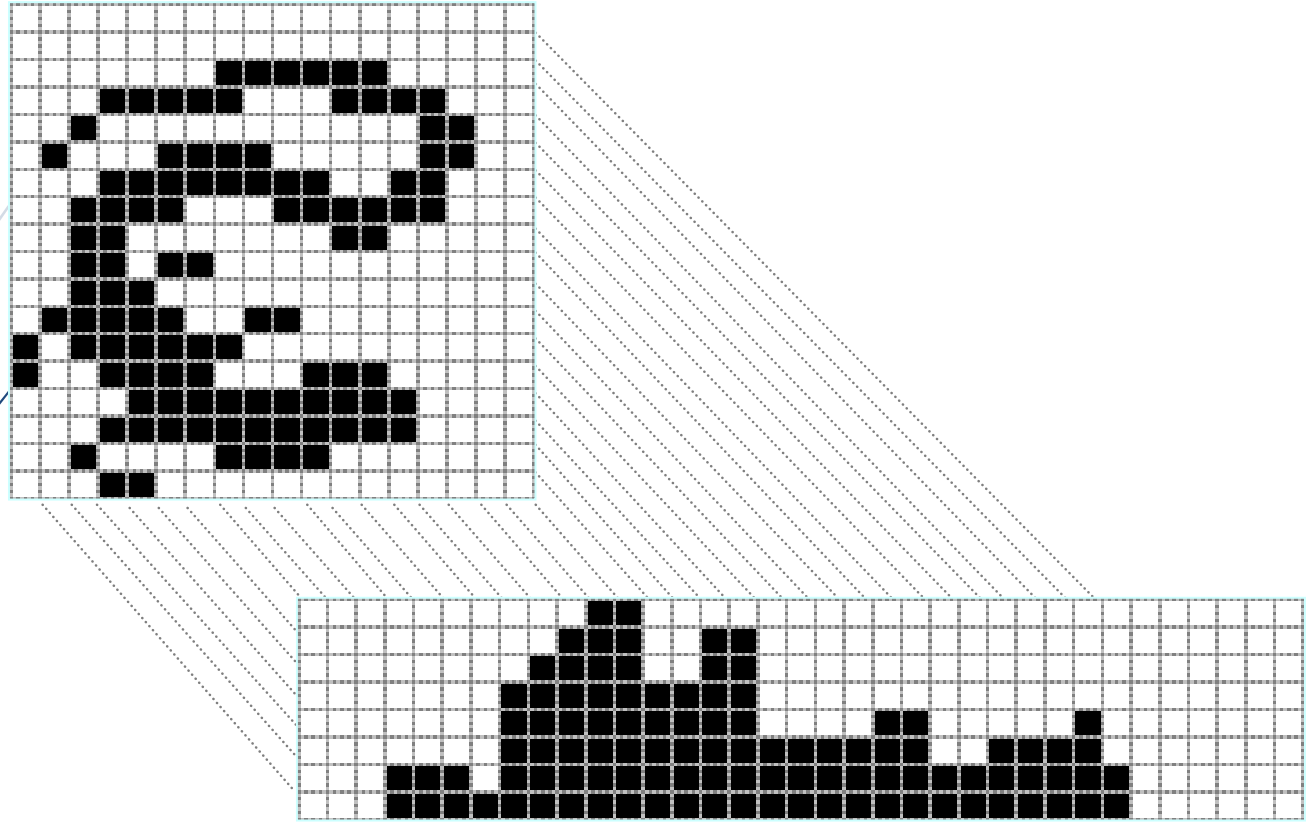
根据投影计算特征

- 投影（projection）:指定方向上前景像素的个数
- 方法：在给定直线上的投影projection.
- 计算二值图像每一列或每一行上像素值为1的像素数量，就得到了二值图像的水平和垂直投影。



二值图像投影图

对角线投影



5. 二值图像的处理

- 连通性及连通成分标记
- 区域的边界
- 图像骨架化
- 数学形态学操作

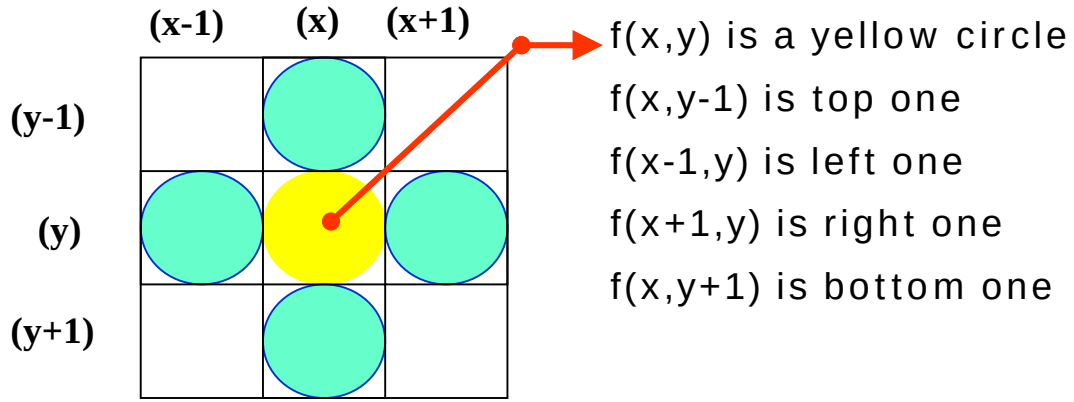
（1）连通性及连通成分标记

- 如何将一幅图像中所有被标记的点组合成物体图像
- 假设物体点在空间上是非常接近
- 把空间上非常接近的点聚合在一起，构成图像的一个成分（component）

近邻及连通

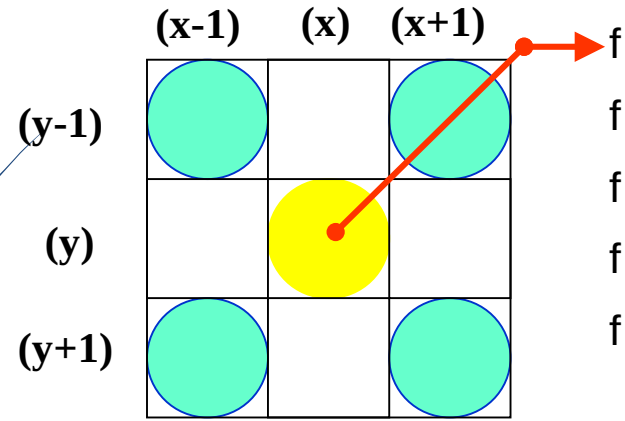
a）像素的4-邻像素

- 像素的4-邻像素表示为N4(p)
- 它是水平及垂直的临像素



近邻及连通

b）对角-邻像素表示为$N_D(p)$



f(x,y) is a yellow circle

f(x-1,y-1) is top-left one

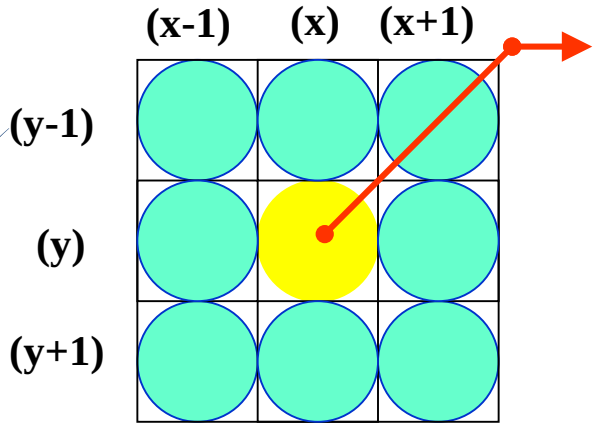
f(x+1,y-1) is top-right one

f(x-1,y+1) is bottom-left one

f(x+1,y+1) is bottom-right one

近邻及连通

c）8-邻像素表示为$N_8(p)$



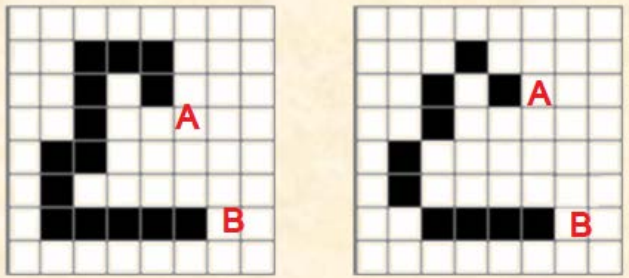
f(x,y) is a yellow circle

(x-1,y-1)	(x,y-1)	(x+1,y-1)
(x-1,y)	(x,y)	(x+1,y)
(x-1,y+1)	(x,y+1)	(x+1,y+1)

Path路径

从像素$[i_0, j_0]$到像素$[i_n, j_n]$的路径(path)是指一个像素序列$[i_0, j_0]$，$[i_1, j_1]$，...，$[i_n, j_n]$，其中像素$[i_k, j_k]$是像素$[i_{k+1}, j_{k+1}]$的近邻像素，$0 \le k \le n-1$.

如果近邻关系是4—连通的，则路径是4—路径；如果是8—连通的，则称为8—路径.



4—路径和8—路径示意图

连通性

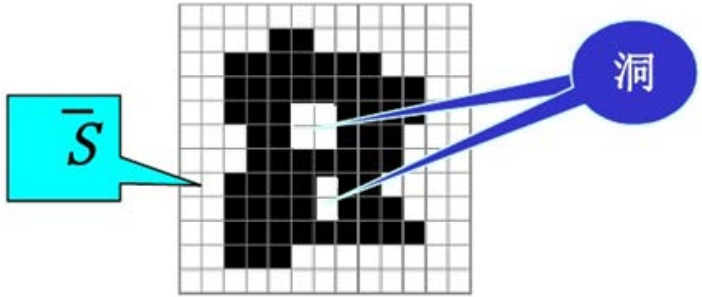
- 前景：图像中值为1的全部像素的集合称为前景(foreground)，用S表示
- 连通性：已知像素，如果存在一条从p到q的路径，且路径上的全部像素都包含在S中，则称p与q是连通的.
- 等价关系：对属于S的任意三个像素p、q和r，有下列性质：
  - 自反性-- 像素p与p本身连
  - 互换性-- 如果p与q连通，则q与p连通
  - 传递性-- 如果p与q连通且q与r连通，则p与r连通

连通成份概念

- **连通成份：**一个像素集合，如果集合内的每一个像素与集合内其它像素连通，则称该集合为一个连通成份

- **背景：** $\bar{S}$（S的补集）中包含图像边界点的所有连通成份的集合称为背景。

  $\bar{S}$中所有其它元称为洞.

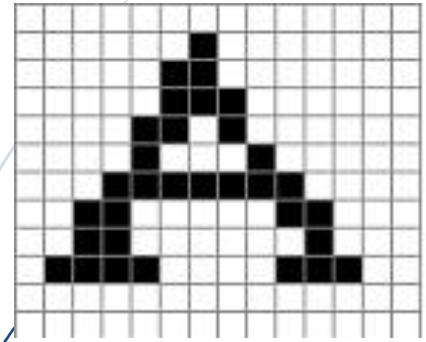


欧拉数

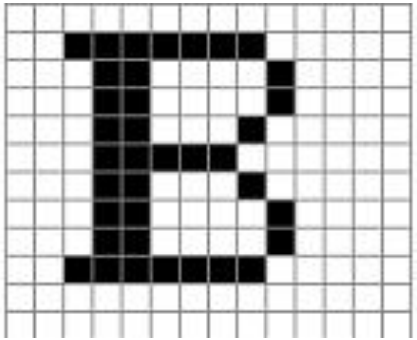
- 欧拉数：在许多应用中，亏格数(genus)（或欧拉数）可作为识别物体的特征．亏格数定义为连通成份数减去空洞数

$$E = C - H$$

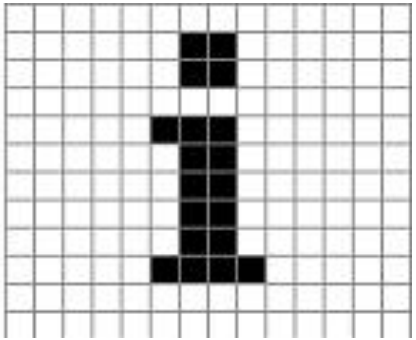
其中E、 C和H分别是欧拉数、连通成份数与空洞数．此式给出了一个简单的拓朴特征。



$E = 0$



$E = -1$

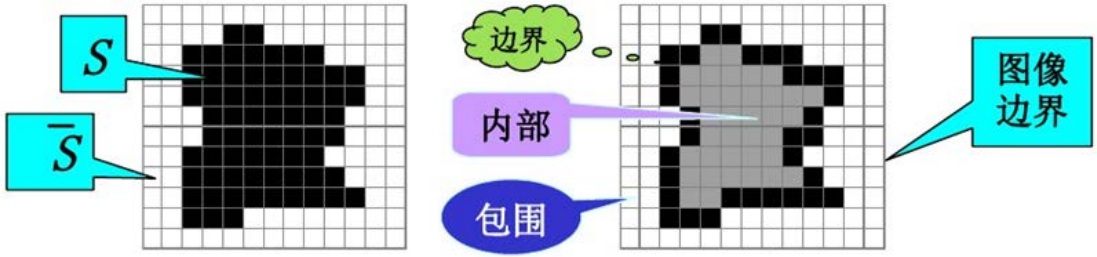


$E = 2$

包围

- 如果从S中任意一点到图像边界的4－路径必须与区域T相交，则区域T包围区域S（或S在T内）.

- 下图即为一幅简单二值图像和它的边界、内部、包围示意图.



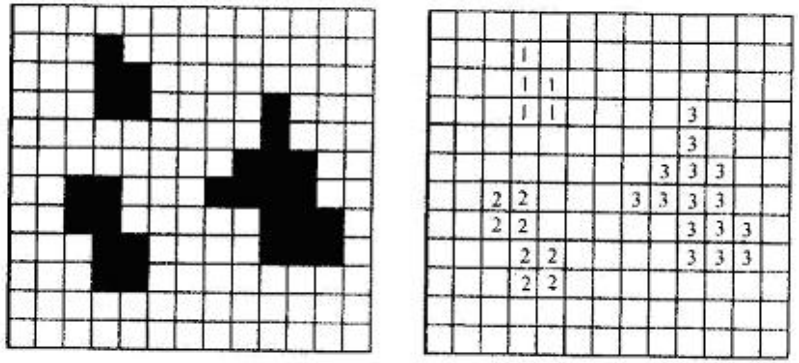
一幅二值图像

连通标记算法

- 在一幅图像中找出连通成份是视觉问题中最常见的运算之一．连通区域内的点构成表示物体的候选区域
- 连通成份算法常常会在二值视觉系统中形成瓶颈效应
- 如果图像中仅有一个物体，那么找连通成份就没有必要；
- 如果图像中有许多物体，且需要求出物体的特性与位置，则必须确定连通成份．
- 连通标记算法可以找到图像中的所有连通成份，并对同一连通成份中的所有点分配同一标记．

连通成份标记

- 连通成份标记常用算法
  - 递归算法（我们这里以这种为例）
  - 序贯算法



图像及其连通成分

递归算法连通成份标记

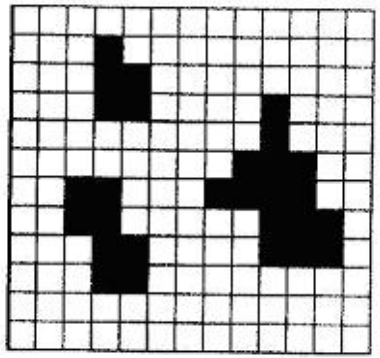
连通成份递归算法（假设L初值为0）

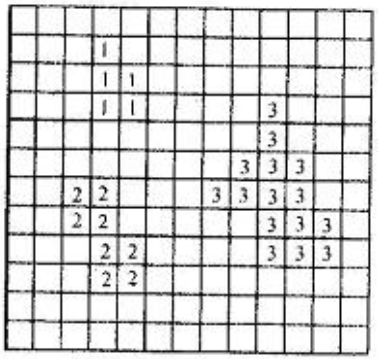
1．扫描图像，找到没有标记的1点，给它分配一个新的标记L.

3．递归分配标记L给1点的邻点.

3．如果不存在没标记的点，则停止.

4．L=L+1,返回第一步.



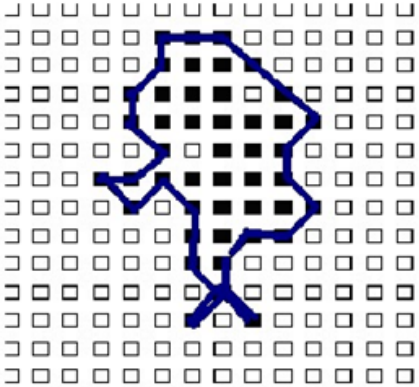


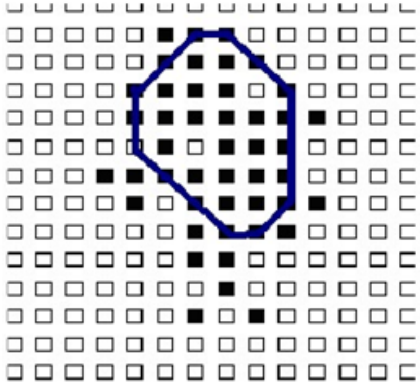
（2）区域的边界

- S的边界(boundary)是S中与S补集中有4－连通关系的像素集合．边界通常记为S'．
- 在大多数应用中，用一特定的顺序跟踪边界点，一般的算法是按顺时针方向跟踪区域的所有点，采用边界跟踪算法
- 假定物体边界不在图像的边界上（即物体完全在图像内部），边界跟踪算法先选择一起始点，然后跟踪边界直到回到起始点．

区域的边界

- 为了得到平滑的图像边界，可以在检测和跟踪图像边界后，利用边界点的方向信息来平滑边界。
- 设置一个边界点方向变化数阈值，将方向变化值大于该阈值的图像边界点滤除，可得到平滑的图像边界





（a）图像边界连接结果；　　（b）边界平滑结果

边界跟踪算法

① 从左到右、从上到下扫描图像，求区域S的起始点，

$$s(k) = (x(k), y(k)), k = 0$$

② 用c表示当前边界上被跟踪的像素点．置 $c = s(k)$，记c左4邻点为b，$b \in \overline{S}$ ；

③ 按逆时针方向从b开始将c的8个8邻点分别记为：

$n_1, n_2, \cdots, n_8$　　$k = k + 1$

④ 从b开始，沿逆时针方向找到第一个 $n_i \in S$ ；

⑤ 置 $c = s(k) = n_i$ ，$b = n_{i-1}$ ；

⑥ 重复步骤③、④、⑤，直到 $s(k) = s(0)$ 。

（3）图像骨架化

- 骨架特点：
  - 单像素厚度
  - 等距性

图像骨架化方法

- 中轴方法
- 细化方法
- 基于主曲线的方法
- 数学形态学方法

（a）中轴方法

距离概念：

- 欧几里得距离：

$$d_{\text{Euclidean}}([i_1, j_1],[i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$$

- 街区距离：

$$d_{\text{Block}} = | i_1 - i_2 | + | j_1 - j_2 |$$

- 棋盘距离：

$$d_{\text{Chess}} = \max(| i_1 - i_2 |, | j_1 - j_2 |)$$

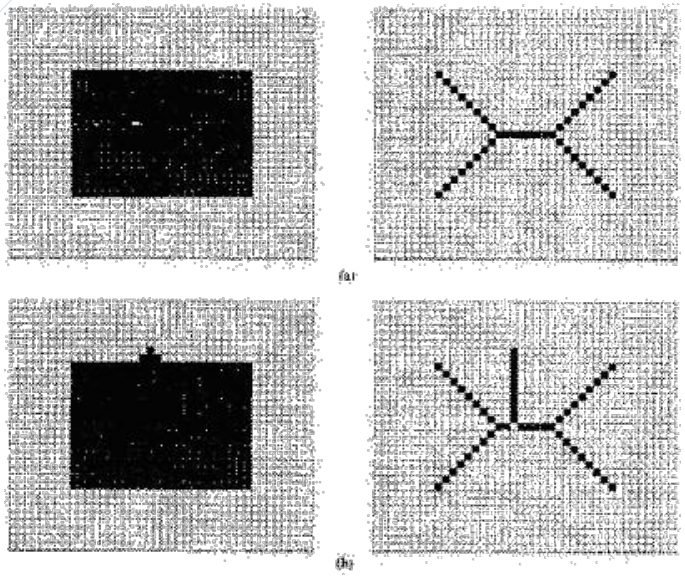
中轴概念

如果对中像素 $[i,j]$ 的所有邻点 $[u,v]$ 有下式成立：

$$d([i,j],\overline{S}) \geq d([u,v],\overline{S})$$

则S中像素 $[i,j]$ 到 $\overline{S}$ 的距离 $d([i,j],\overline{S})$ 是局部最大值．s中所有到 $\overline{S}$ 的距离是局部最大值的像素点集合称为对称轴或中轴 。 通常记为 S^*

中轴举例



（b）细化方法

- **细化(thinning)：**是一种图像处理运算，可以把二值图像区域缩成线条，以逼近区域的中心线，也称之为骨架或核线.
- **细化的目的：**减少图像成份，直到只留下区域的最基本信息，以便进一步分析和识别.
- 虽然细化可以用在包含任何区域形状的二值图像，但它主要对细长形(而不是凸圆形或水滴状)区域有效.
- 细化一般用于文本分析预处理阶段，以便将文本图像中线条图画或字符笔画表示成单像素线条.

细化要求

（1）连通图像区域必须细化成连通线结构

（2）细化结果最少应该是8－连通

（3）保留近似终止线的位置

（4）细化结果应该近似于中轴线

（5）由细化引起的附加突刺(短分支)应该是最小的

细化是把区域缩成线条、逼近中心线（骨架或核线）的一种图像处理。细化的目的是减少图像成份，直到只留下区域的最基本信息，以便进一步分析和识别．虽然细化可以用在包含任何区域形状的二值图像，但它主要对细长形(而不是凸圆形或水滴状)区域有效．细化一般用于文本分析预处理阶段，以便将文本图像中线条图画或字符笔画表示成单像素线条．

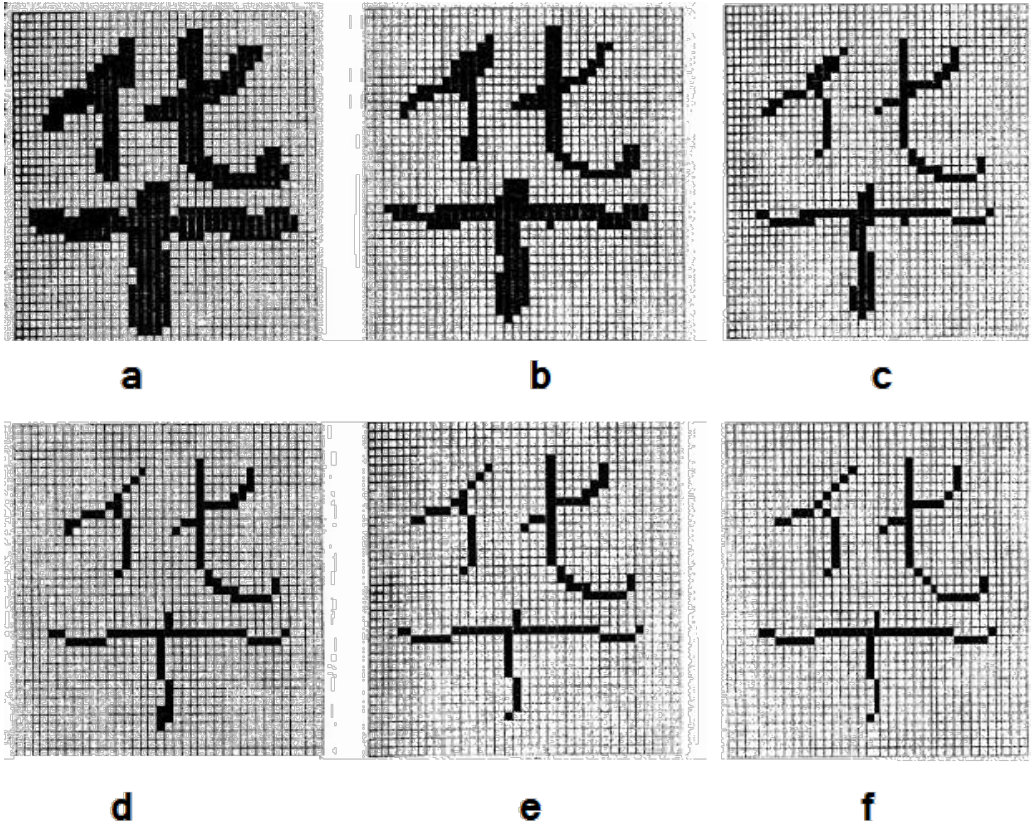
细化措施

- 一种常用的细化手段是在至少 3×3 邻域内检查图像的每一点，剥去区域边界．一次剥去一层图像，直至区域被细化成一条线．这一过程是用迭代法实现的。

- 在每次迭代时，每一个像素点用 $n\times n$ 窗函数检查，为了保持连通性或线末端位置，将单像素厚的边界擦除
- 在每次迭代中，值为1外层区域就是用这种方式削掉．当迭代结果没有变化时，迭代过程结束，图像得到细化

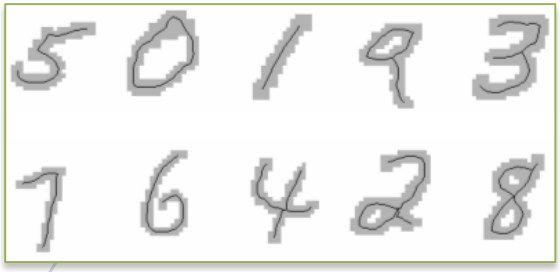
在满足一定条件下迭代

删除外围前景像素



(a) 原图像，(b)—(f)为

五次迭代过程，每次迭代削去一层边界



细化实例

（4）利用数学形态学方法处理二值图像

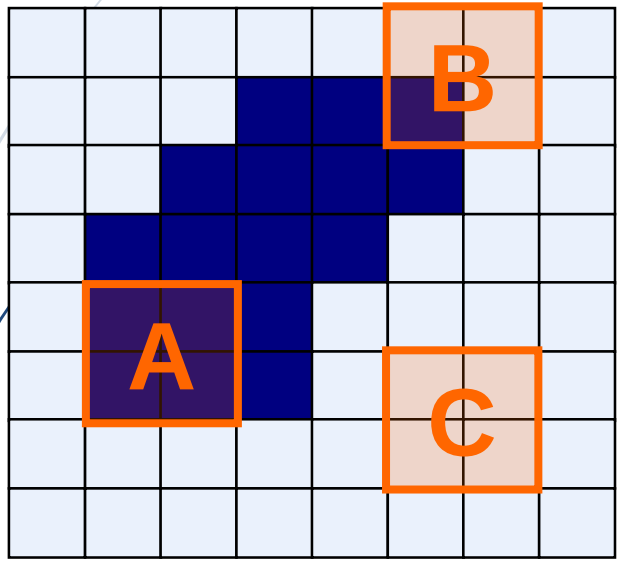
- 图像分析与识别思想：基于形状
- 理论基础：集合论
- 作用：保持形状特征，同时简化图像
- 工具：结构元

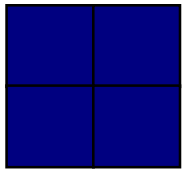
形态学算子

- 膨胀（dilation）
- 腐蚀（erosion）
- 开运算
- 闭运算

结构元

- 数学形态学中利用结构元对图像进行二值化处理





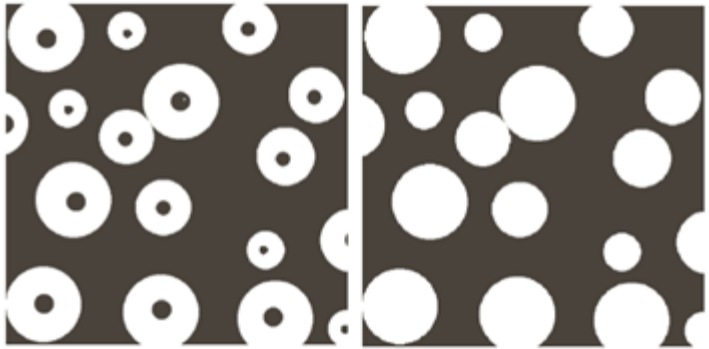
Structuring Element
结构元

应用形态学算子处理二值图像

1. 边缘提取
2. 区域（孔洞）填充
3. 连通成分标记
4. 图像骨架化

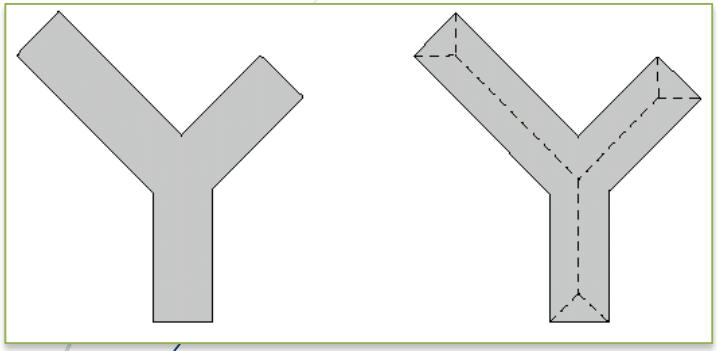
形态学处理二值图像实例

边缘提取的实例

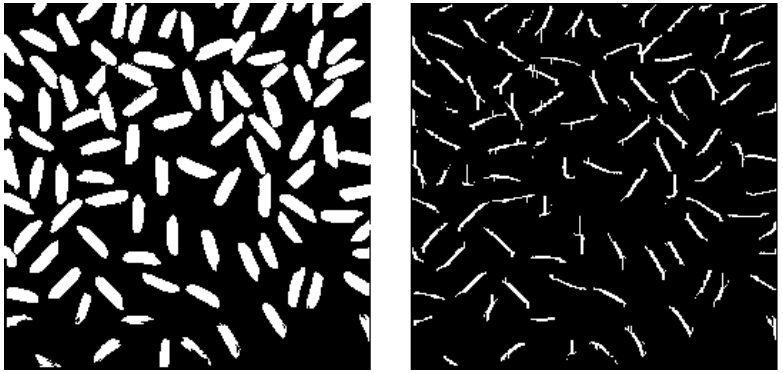


填充孔的实例

形态学处理二值图像实例



图像骨架实例



图像骨架实例

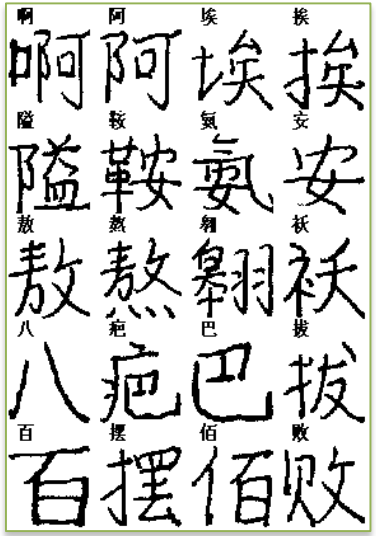
6. 二值图像的应用实例

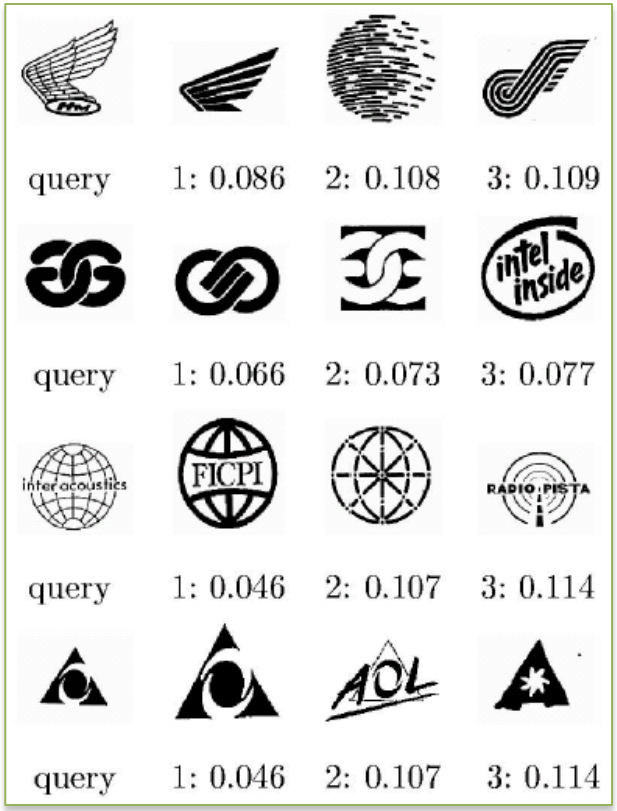
- 工业应用控制中的应用
- 数字水印
- 扫码程序
- 二值化图像特征:边缘、区域
- 图像检索
- 笔画识别
- 草图建模技术

工业应用

- 工业应用中，物体通常出现在已知表面（如工作台面）上，而且摄像机相对台面的位置也是已知的．在这种情况下，图像中的物体位置决定了它的空间位置．
- 物体位置计算方法：用物体的外接矩形、物体矩心（区域中心）等来表示物体的位置．区域中心是通过对图像进行“全局”运算得到的一个点，因此它对图像中的噪声相对来说是不敏感的
- 在大多数应用中，物体的数量不是很多，如果物体的尺寸和形状完全不同，则可以利用尺度和形状特征来识别这些物体
- 实际上在许多工业应用中，经常使用区域的一些简单特征，如大小、位置和方向，来确定物体的位置并识别它们

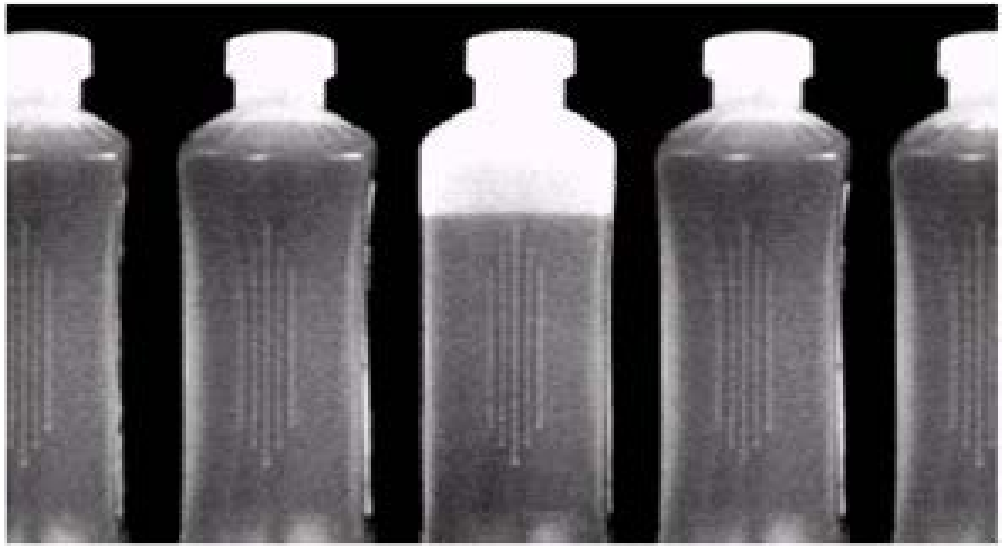
应用实例





文字识别及图像检索中的应用实例

产品检测中应用二值图像



先二值化分割再检测