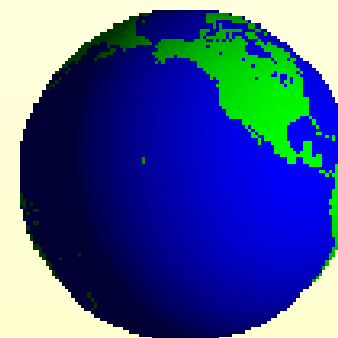




北京航空航天大学

数字图像处理 图像分割



计算机学院

图像处理的三个层次

图像处理

低层

- 降噪、增强、锐化
- 输入输出都是图像

中层

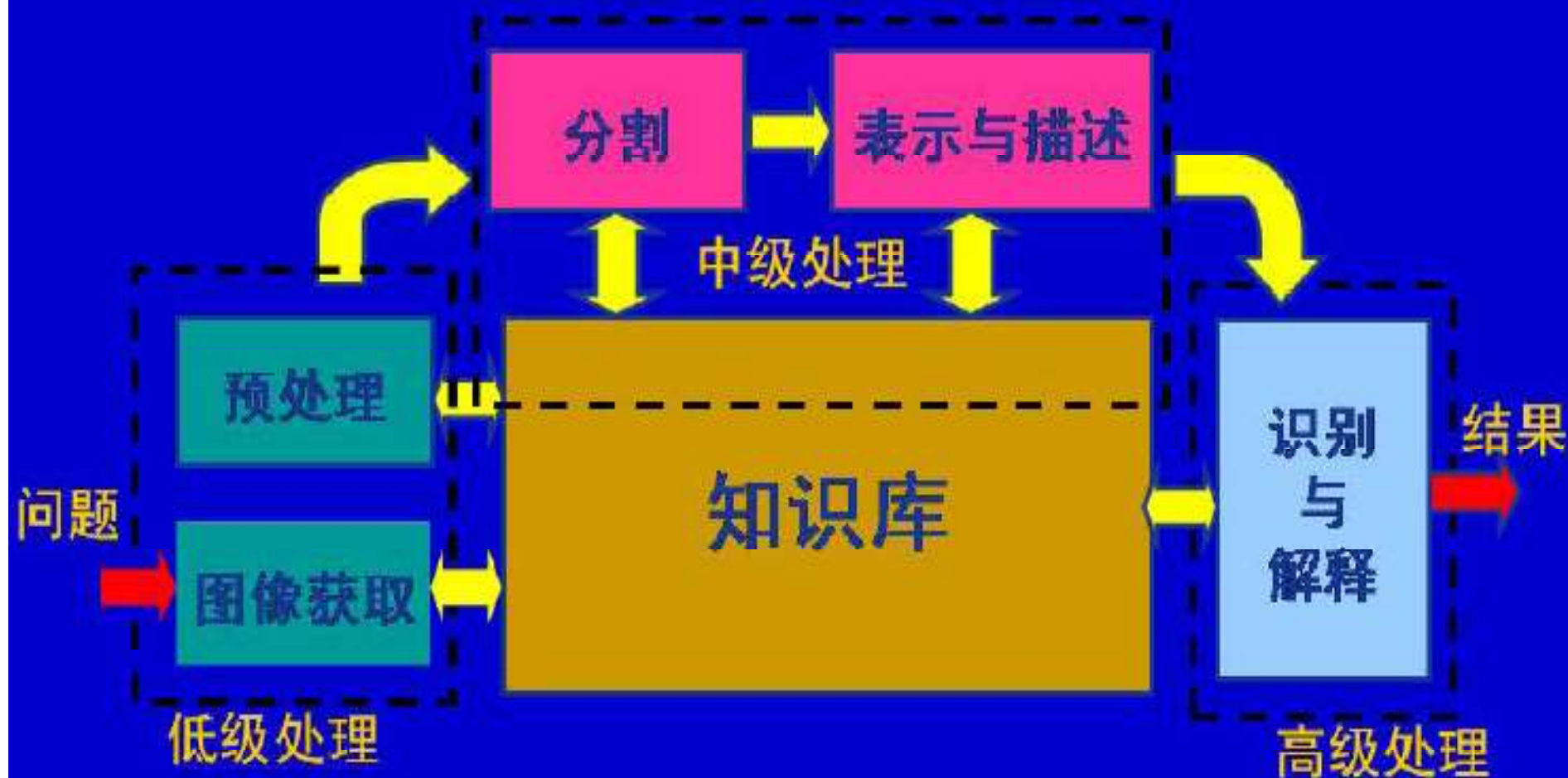
- 分割、特征提取
- 输入为图像，输出为图像中提取的特征

高层

- 分类、识别
- 输入为图像，输出为对图像的语义理解

计算机视觉

图像理解的基本构成





- 引言

- 图像分析的概念
- 图像分析系统的基本构成
- 图像分割的概念
- 图像分割的基本思路
- 图像分割的基本策略



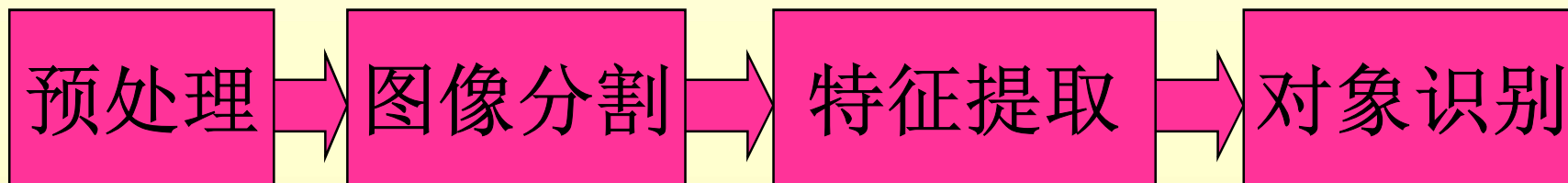
- 图像处理

- 输入输出均为图像

- 图像分析的概念

- 从图像中提取信息的技术。输入是图像、输出是从图像中提取出来的属性。对图像中感兴趣的目标进行检测和测量，以获得它们的客观信息，从而建立对图像的描述。

- 图像分析系统的基本构成





- 图像分割的概念
 - 把图像分解成构成它的部件(子区域)和对象的过程
 - 有选择性地定位感兴趣对象在图像中的位置和范围





- 分割将图像细分为互不重叠的区域并提取感兴趣目标的技术

令集合 R 代表整个图像区域，对 R 的分割可看作将 R 分成 N 个满足以下五个条件的非空子集（子区域） R_1, R_2, \dots, R_N ：

对所有的 i 和 j ， $i \neq j$ ，有 $R_i \cap R_j = \phi$ ；

对 $i = 1, 2, \dots, N$ ，有 $P(R_i) = \text{TRUE}$ ；

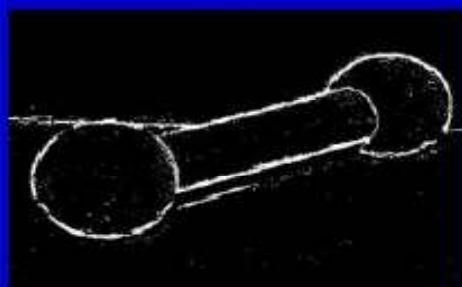
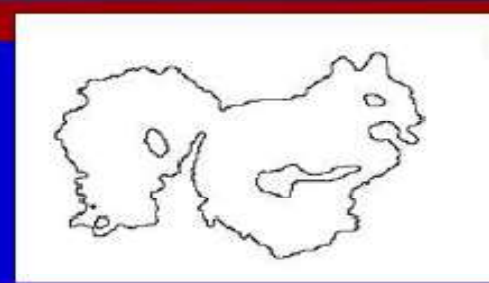
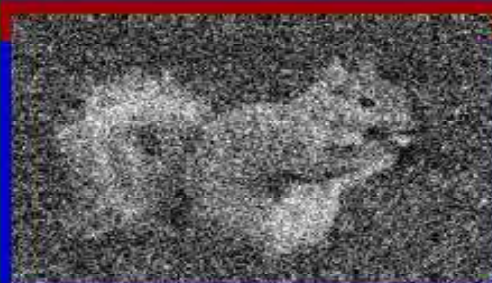
对 $i \neq j$ ，有 $P(R_i \cup R_j) = \text{FALSE}$ ；

对 $i = 1, 2, \dots, N$ ， R_i 是连通的区域。

其中 $P(R_i)$ 是对所有在集合 R_i 中元素的逻辑谓词， ϕ 代表空集



实例





- 分割依赖于低层视觉：不连续性、相似性
- 分割依赖于高层视觉：不完全属于图像特征提取问题，还涉及到各种图像特征的知觉组织。



从一般意义上来说，只有对图像内容的彻底理解，才能产生完美的分割。



图像分割的基本思路

- 从简到难，逐级分割
 - ✓ 控制矩形区域、定位拍照、定位文字
- 控制背景环境，降低分割难度
 - ✓ 背景环境、路面、天空
- 把焦点放在增强感兴趣对象，缩小不相干图像成分的干扰
 - ✓ 感兴趣的对象：汽车牌照
 - ✓ 不相干图像成分：非矩形区域





图像分割的基本策略

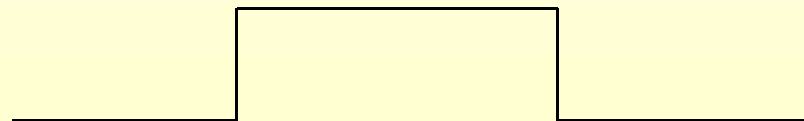
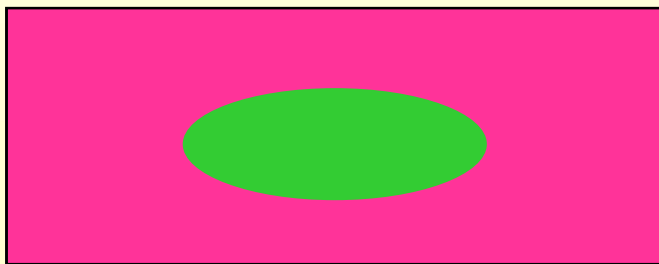
基于图像灰度值的两个基本特性

- 不连续性——区域之间
- 相似性——区域内部



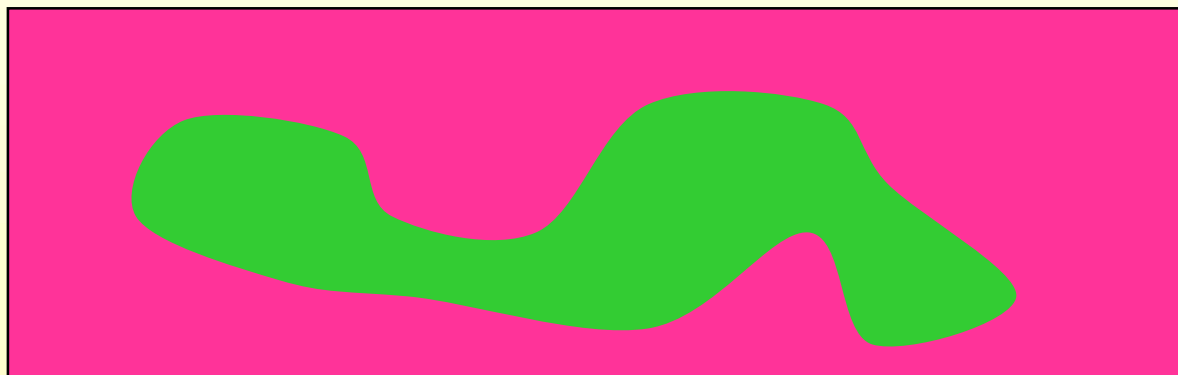


- **不连续性**——基于亮度的突变来分割一幅图像，如图像中的边缘
 - 检测图像像素灰度级的**不连续性**，找到点、线（宽度为1）、边（不定宽度）
 - 先找到边，再确定区域





- **相似性**——根据事先定义的准则把图像分割成相似的区域
 - 通过选择阈值，找到灰度值相似的区域，区域的外轮廓就是对象的边。
 - 门限处理、区域生长、拆分与合并等。





基于边缘的方法：边缘检测

基于区域的方法：阈值法、区域生长法

对图像特征空间做分类的方法

基于函数优化的方法：Bayesian等

综合考虑边缘的区域信息的混合分割方法



- 1、边界分割法：点、线、边的检测
- 2、边缘连接法
- 3、阈值分割法
- 4、面向区域的分割



- 边界分割法：基于亮度的不连续变化分割图像
 - 点的检测
 - 线的检测
 - 边的检测





• 点的检测

—用空域的高通滤波器来检测孤立点

例：3×3的掩模，掩模在该图像中任何一点处的响应R
由下式给出

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

z_i 是与掩模系数 w_i 相关的像素的亮度

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

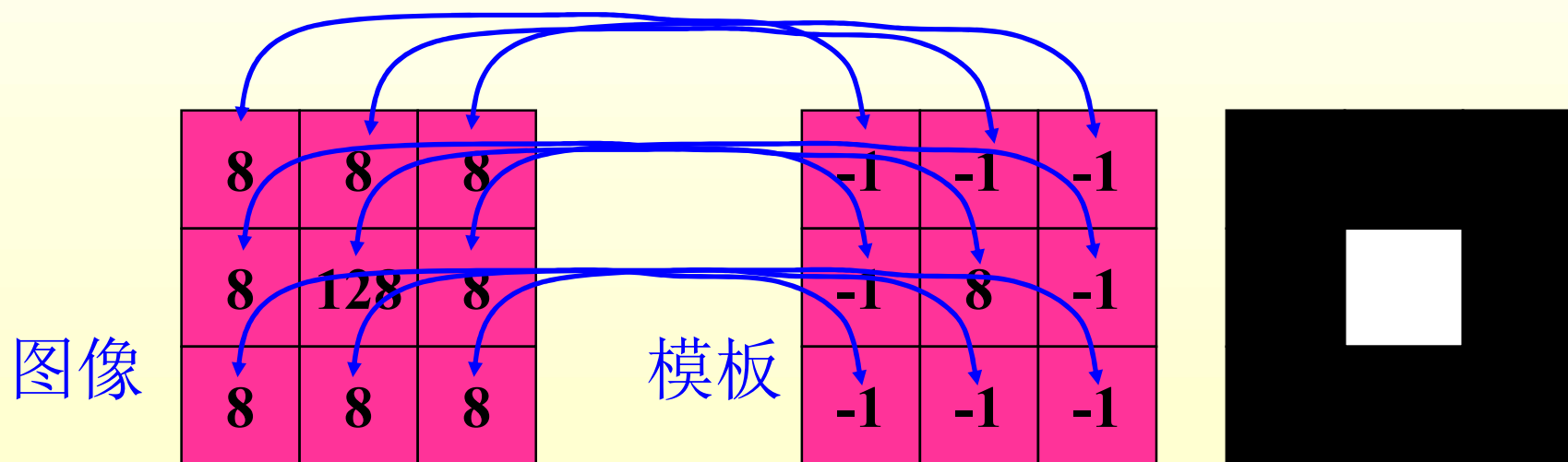
点检测： $|R| \geq T$

T是非负阈值



$$R = |-1 * 8 * 8 + 128 * 8) / 9|$$
$$= |(120 * 8) / 9| = |960 / 9| = 106$$

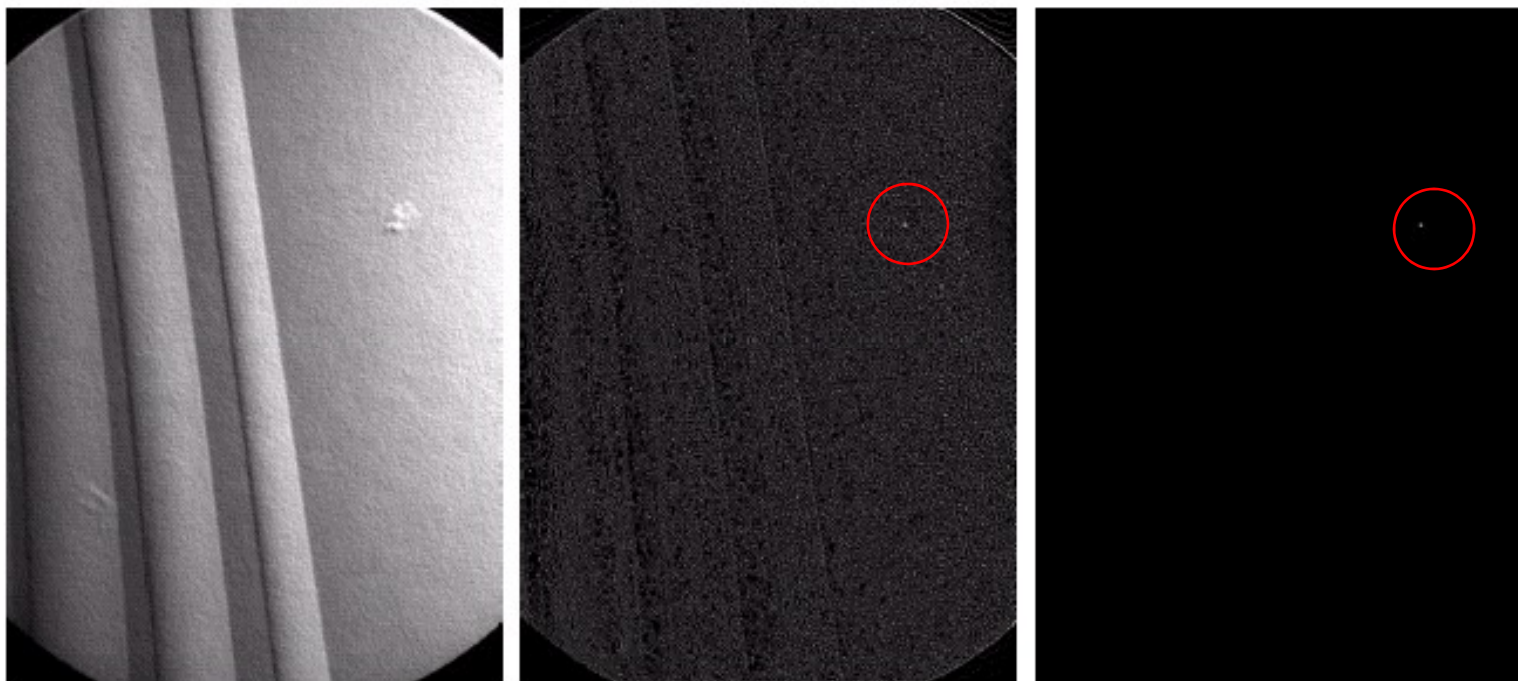
设：阈值： $T = 64$





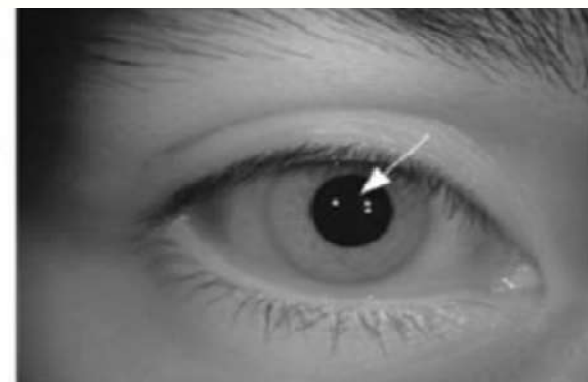
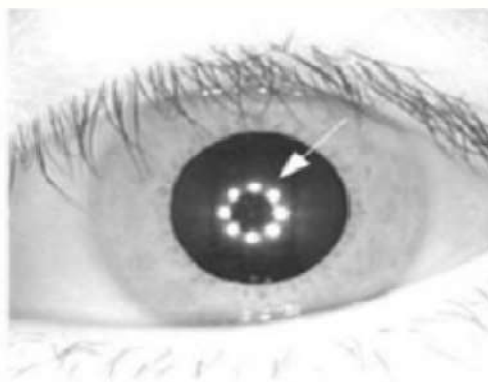
• 点的检测

- 当掩模的中心位于一个孤立点时，掩模的响应必须最强。即当R的值足够大时，说明该点的值与周围的点非常不同，是孤立点。通过阈值T来控制如 $T = 32、64、128$ 等，如 $|R| > T$ ，则便检测到一个孤立点
- 在亮度不变的区域中响应为零

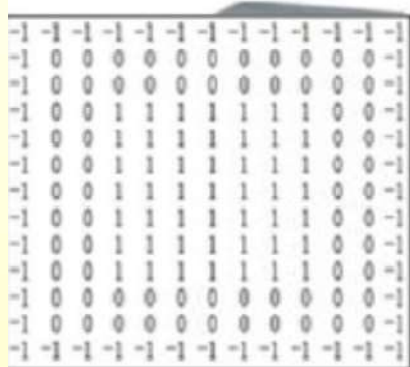


一个黑色像素，表示涡轮叶片上的孔洞





(a) 虹膜图像中的光斑



(a) 光斑检测算子



(b) 原始图像



(c) 光斑检测和验证结果



• 线的检测

- 通过比较典型模板的计算值，确定一个点是否在某个方向的线上

-1	-1	-1
2	2	2
-1	-1	-1

最佳响应水平线

-1	-1	2
-1	2	-1
2	-1	-1

最佳响应+45度线

-1	2	-1
-1	2	-1
-1	2	-1

最佳响应垂直线

2	-1	-1
-1	2	-1
-1	-1	2

最佳响应-45度线

每个掩模的系数之和为零，表明亮度不变区域中来自掩模的响应为零。





- 线的检测

1	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5
1	1	1	1	1	1	1	1	1

-1	-1	-1
2	2	2
-1	-1	-1

$$R1 = -6 + 30 = 24$$

-1	-1	2
-1	2	-1
2	-1	-1

$$R2 = -14 + 14 = 0$$

-1	2	-1
-1	2	-1
-1	2	-1

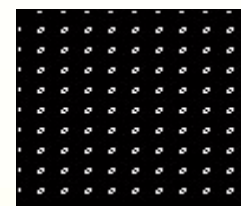
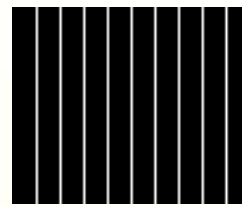
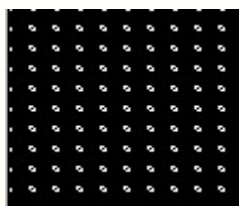
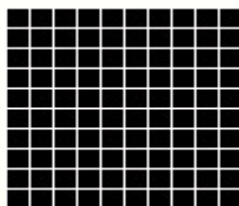
$$R3 = -14 + 14 = 0$$

2	-1	-1
-1	2	-1
-1	-1	2

$$R4 = -14 + 14 = 0$$



1—边界分割法



➤ 二值电路接线模板

➤ 问题：？

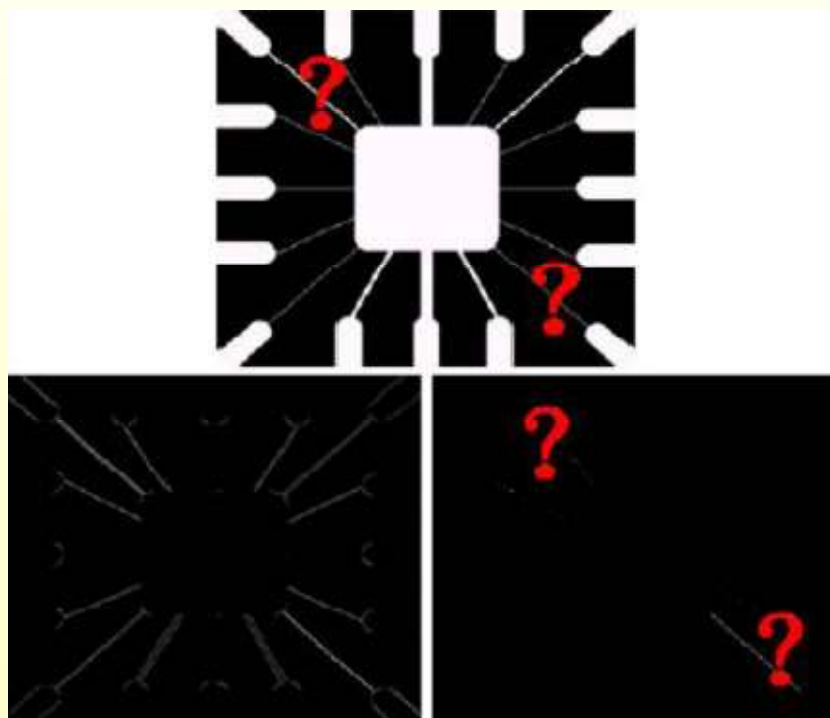
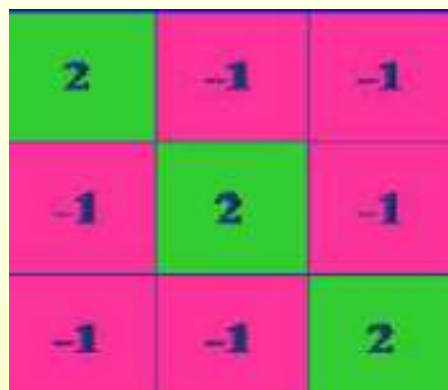


Illustration of line detection.
(a) Binary wire-bond mask.
(b) Absolute value of result after processing with -45° line detector.
(c) Result of thresholding image (b).

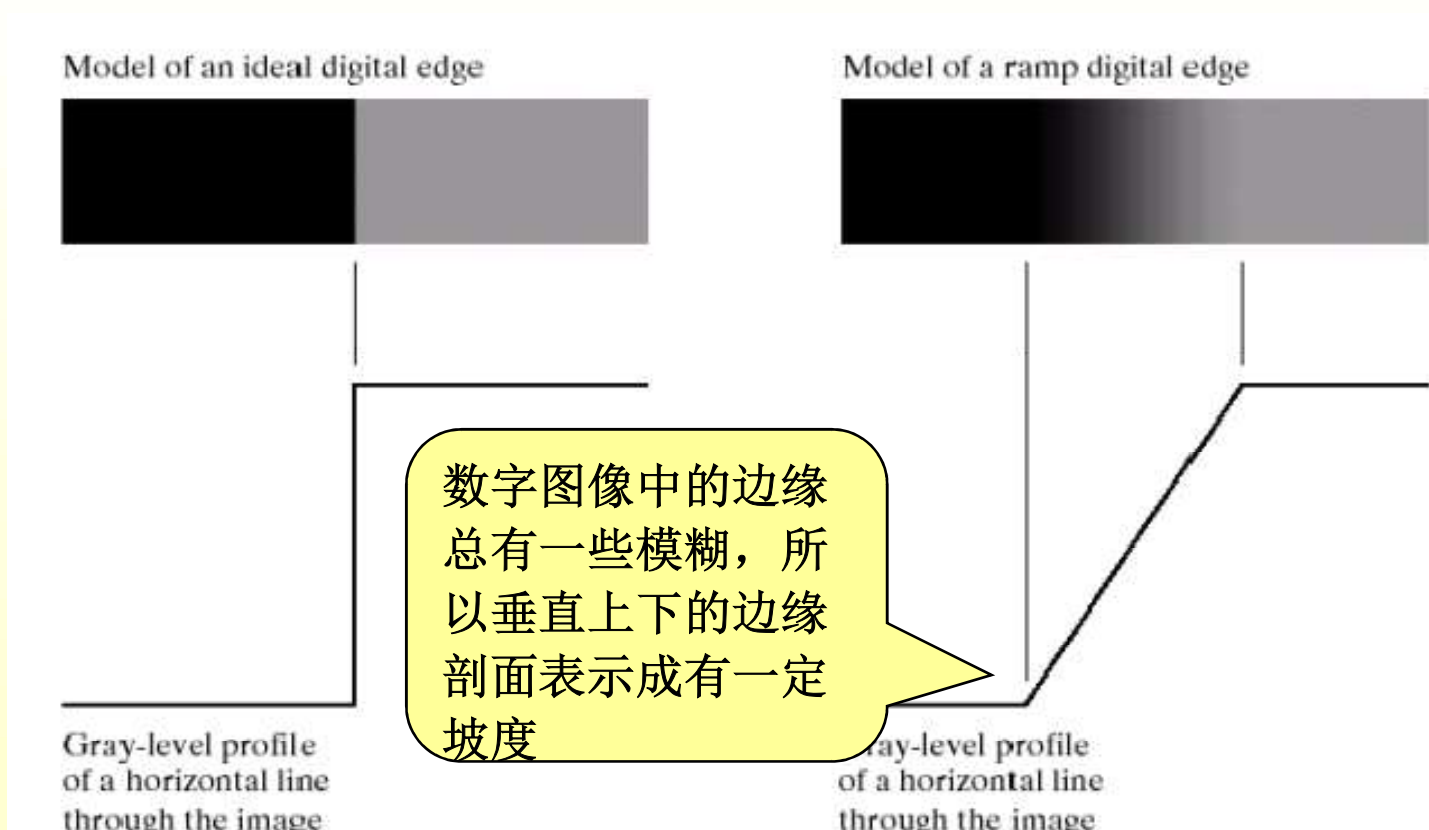


- 边的检测

- 边界的定义：一段边是两个具有相对不同灰度值特性的区域的边界线
- 适用于：假定问题中的区域是非常类似的，两个区域之间的过渡，仅仅根据灰度的不连续性便可确定
- 不适用于：当假定不成立时，阈值分割技术一般来说比边缘检测更加实用。
 - 阈值分割技术包括单阈值分割<图像是由具有单峰灰度分布的目标和背景组成>和多阈值分割<如果图像中有灰度值不同的区域>
 - 边缘检测是通过边缘检测算子找到物体的边缘（边缘检测算子是一组用在图像强度函数中定位变化重要的局部图像预处理方法）



— 边缘的宽度取决于图像中边缘的模糊程度

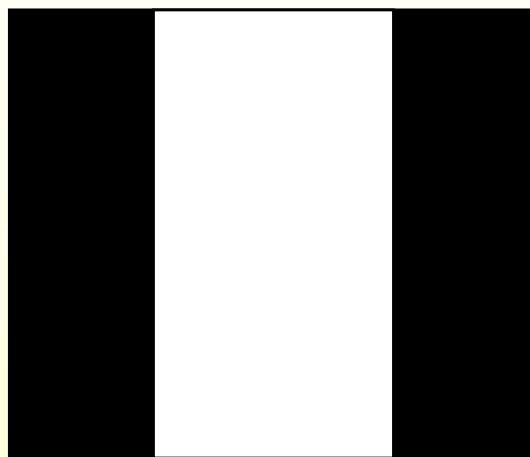




- 边的检测

- 基本思想：
计算局部微分算子

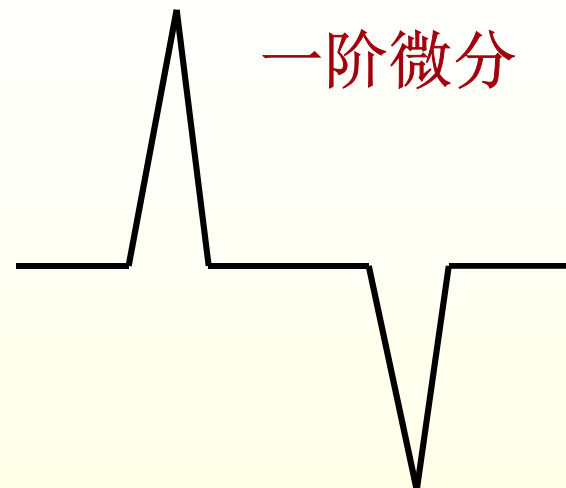
边界图像



剖面图



一阶微分



二阶微分



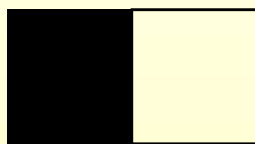


• 边的检测

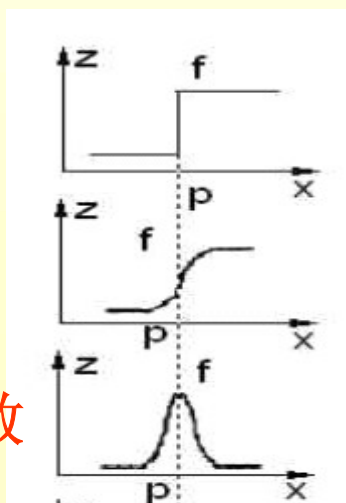
— 一阶微分：用梯度算子来计算

- ✓ 特点：对于由暗到亮的边缘，边的一阶微分为正值；对于由亮到暗的边缘，边的一阶微分为负值；常数部分为零；
- ✓ 用途：用于检测图像中边的存在。

图像

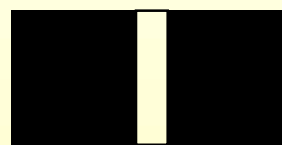


剖面

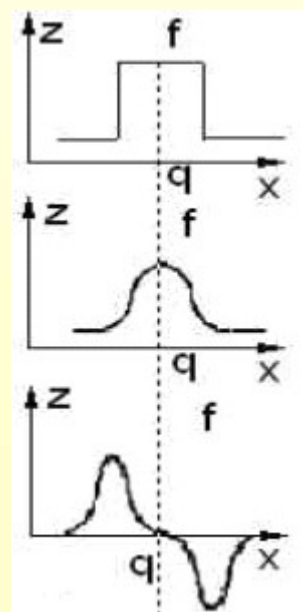


一阶导数

图像



剖面



一阶导数



• 边的检测

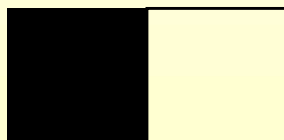
— 二阶微分：通过二阶微分算子来计算

- 特点：二阶微分边缘与黑色一边相关的跃变点二阶微分为正，与亮色一边相关的跃变点二阶微分为负；过零点位置正对应原图像中边缘位置。

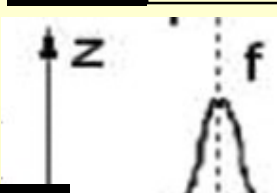
- 用途：

- 1) 二次导数的符号，用于确定边上的像素是在亮的一边，还是暗的一边。
- 2) 零跨越，确定边的准确位置

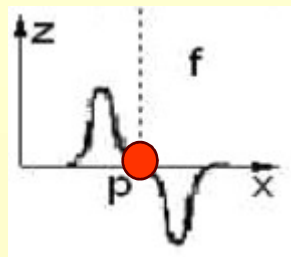
图像



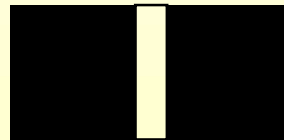
一阶导数



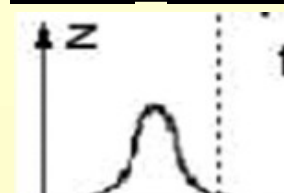
二阶导数



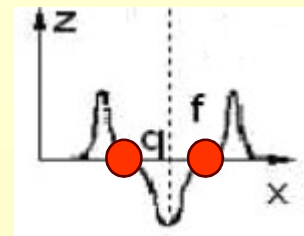
图像



一阶导数



二阶导数



计算机学院



实际上数字图像中求导数是利用差分近似微分来进行的。

- 边的检测——梯度算子

函数 $f(x,y)$ 在 (x,y) 处的梯度为一个向量：

$$\nabla f = [\partial f / \partial x, \partial f / \partial y]$$

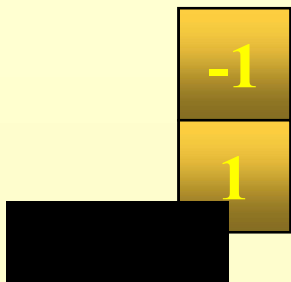
计算这个向量的大小为：

$$|\nabla f| = \text{mag}(\nabla f) = [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2}$$

近似为： $\nabla f \approx |\partial x| + |\partial y|$

梯度的方向角为：

$$\alpha(x,y) = \tan^{-1}(\partial y / \partial x)$$



$$g(x,y) = \begin{cases} 1 & \text{Grad}(x,y) \geq T \\ 0 & \text{其它} \end{cases}$$



- 为了检测边缘点，选取适当的阈值 T ，对梯度图像进行二值化
- 这样形成了一幅边缘二值图像 $g(x,y)$
- 特点：仅计算相邻像素的灰度差，对噪声比较敏感，无法抑止噪声的影响



- 边的检测

- 梯度算子

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

 ∂x

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

 ∂y

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

- 偏导数需要对每个像素位置计算，在实际中常用小区域模板卷积来近似计算。

Sobel算子为：

$$\partial x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$\partial y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$



- 边的检测

- Sobel梯度算子的使用与分析

1. 直接计算 ∂y 、 ∂x 可以检测到边的存在，以及从暗到亮，从亮到暗的变化
2. 仅计算 $|\partial x|$ ，产生最强的x方向的响应； $|\partial y|$ 则是产生最强的y方向的响应。
3. 由于微分增强了噪音，平滑效果是Sobel算子特别引人注意的特性



北京航空航天大学

a b
c d

FIGURE 10.10

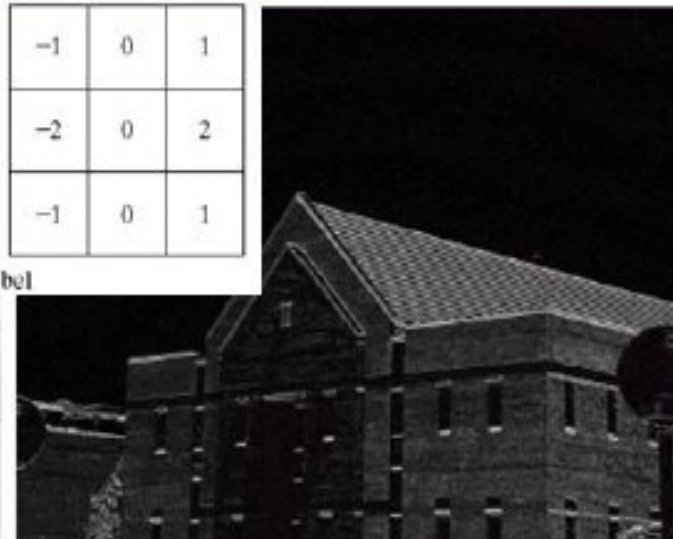
(a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

y
方
向
的
响
应



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

Sobel



x
方
向
的
响
应



根据梯度计算方法不同

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0	0	-1
0	1	1	0

Roberts

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

Prewitt

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

Sobel

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

Prewitt

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

Sobel



- 边的检测

- 拉普拉斯算子

二维函数 $f(x,y)$ 的拉普拉斯是一个二阶的微分定义为： $\nabla^2 f = [\partial^2 f / \partial x^2, \partial^2 f / \partial y^2]$

可以用多种表示方式。对于一个 3×3 的区域，经验上被推荐最多的形式是：

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

或者 $\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1



- 边的检测

- 拉普拉斯算子

- 定义数字形式的拉普拉斯的基本要求是，作用于中心像素的系数是一个正数，而且其周围像素的系数为负数，系数之和必为0。

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



- 边的检测

- 拉普拉斯算子的分析：

- 应用：

- 检测一个像素是在边的亮的一边还是暗的一边
 - 利用零跨越，确定边的位置
 - 对细线和孤立点检测效果较好

- 缺点：

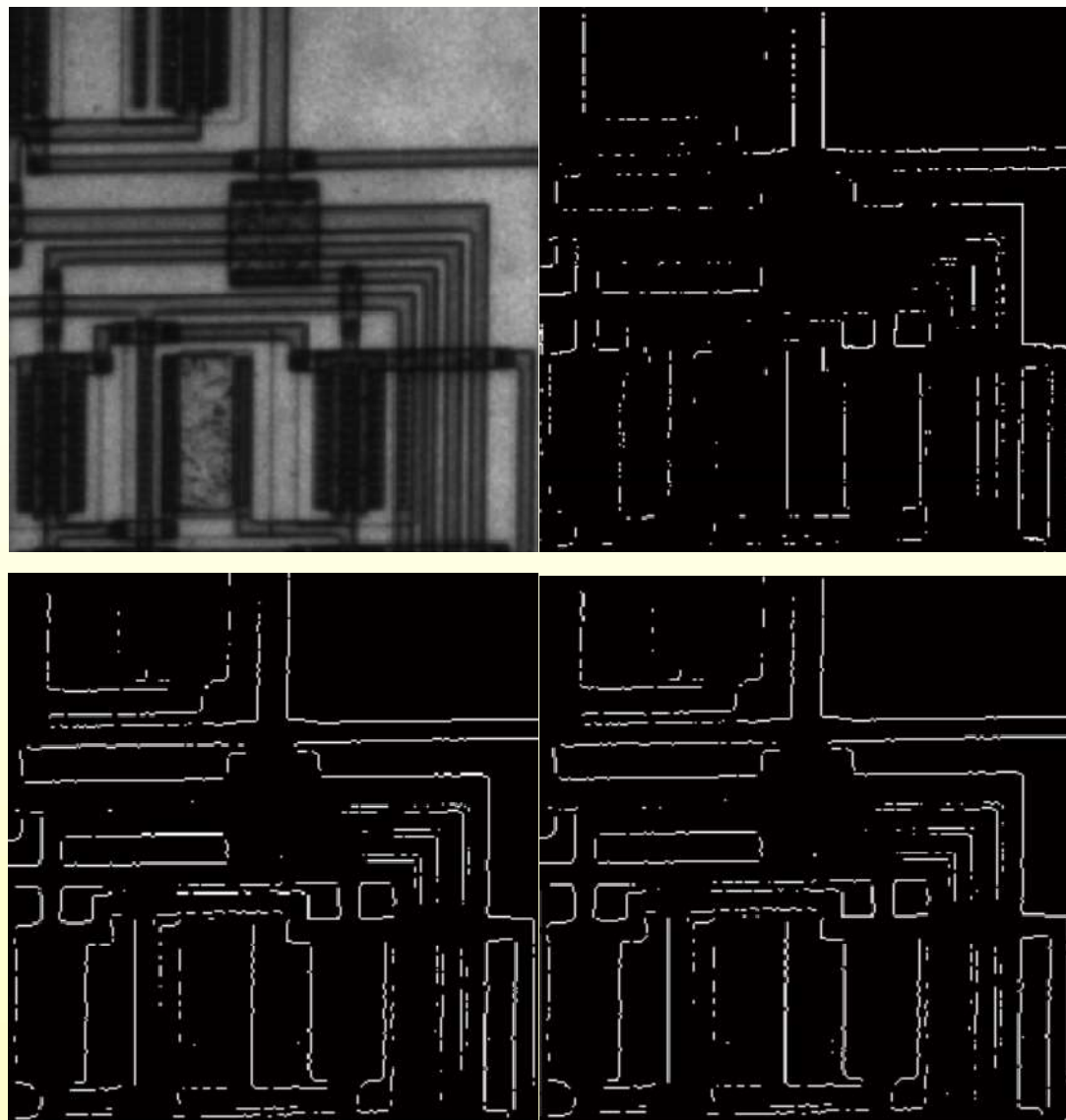
- 对噪音敏感
 - 会产生双边效果
 - 不能检测出边的方向



北京航空航天大学

边的检测——示例

(a)原图 (b)roberts
(c)prewitt (d)sobel



计算机学院



(a) 原始图像



(b) 梯度算子



(c) Roberts算子



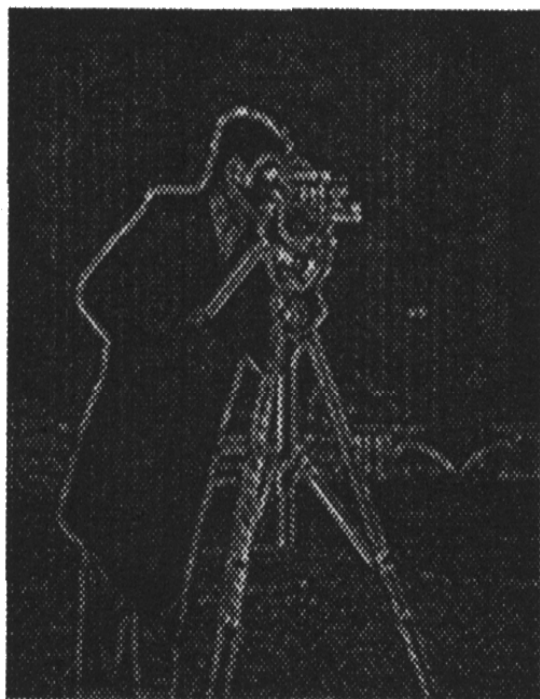
(d) Prewitt算子



(e) Sobel算子



(f) Kirsch算子



(a) Laplacian算子



(b) Marr算子



(c) 曲面拟合法



在边缘检测方法后，通过边缘连接将边缘像素组合成有意义的边缘





- 1、边界分割法：点、线、边的检测
- 2、边缘连接法
- 3、阈值分割法
- 4、面向区域的分割



- 边缘连接法
 - 局部处理法
 - Hough变换



- 边缘连接法

- 由于噪音的原因，边界的特征很少能够被完整地描述，在亮度不一致的地方会中断。
- 因此典型的边检测算法后面总要跟随着连接过程和其它边界检测过程，用来归整边像素，成为有意义的边。



- 局部连接处理
 - 连接处理的时机和目的
 - 连接处理的原理
 - 局部连接算法描述



- 局部连接处理

- 连接处理的时机和目的：

- 时机：对做过边界检测的图像进行

- 目的：连接间断的边

- 连接处理的原理：

- 用比较梯度算子的响应强度（梯度幅值）和梯度方向确定两个点是否同属一条边



- 局部连接处理

- 连接处理的原理：

- **梯度幅值**—通过比较梯度，确定两个点的连接性：

对于点 (x',y') ，判断其是否与邻域内的边界点 (x,y) 相似，当：

$$|\nabla f(x,y) - \nabla f(x',y')| \leq T$$

其中 T 是一个非负的阈值



- 局部连接处理

- 连接处理的原理：

- 梯度方向**——比较梯度向量的方向角

对于点 (x',y') ，判断其是否与邻域内的边界点 (x,y) 的方向角相似，当：

$$|\alpha(x,y) - \alpha(x',y')| < A$$

其中A是一个角度阈值



- 局部连接处理

- 连接处理的原理：

当梯度值和方向角都是相似的，则点 (x', y') ，与边点界 (x, y) 是连接的

$$|\nabla f(x, y) - \nabla f(x', y')| \leq T$$

$$|\alpha(x, y) - \alpha(x', y')| < A$$



- 局部连接处理

- 局部连接算法描述：

- 1) 设定A、T的阈值大小，确定邻域的大小
- 2) 对图像上每一个像素的邻域点进行分析，判断是否需要连接。
- 3) 记录像素连接的情况，另开一个空间，给不同的边以不同的标记。
- 4) 最后，删除孤立线段，连接断开的线段。

1个像素是否与其邻域中的另1个像素连通并不需要在其他判断后作出，在这个意义上，边界连接可并行完成。

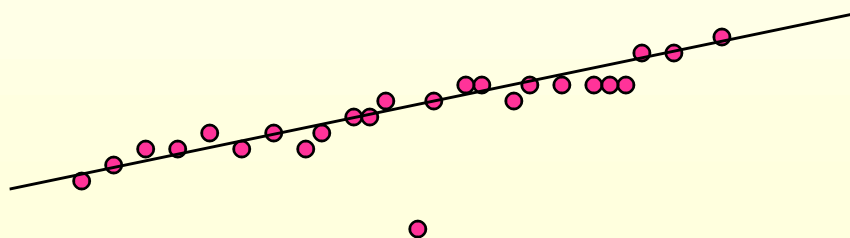


- Hough变换
 - 问题的提出
 - Hough变换的基本思想
 - 算法实现
 - Hough变换的扩展



- Hough变换问题的提出

- 在找出边界点集之后，需要连接，形成完整的边界图形描述

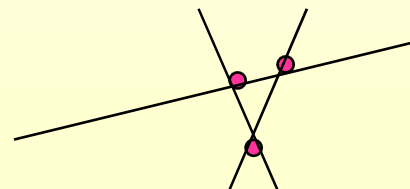


设给定图像中的 n 个点，要从中确定连在同1条直线上的点的子集。即已检测出1条直线上的若干个点，需要求它们所在的直线。



- 直接办法

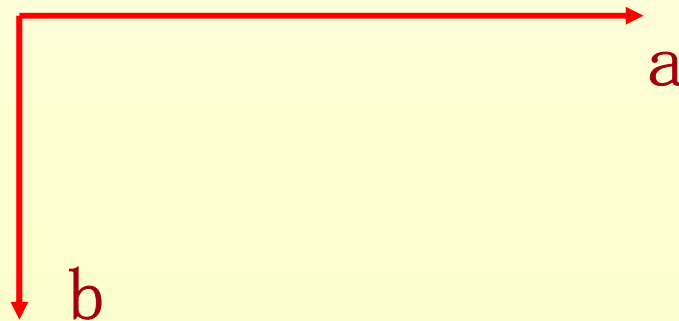
- 先确定所有由任意2点决定的直线（需约 n^2 次运算以确定 $n(n-1)/2$ 条线），再找出接近具体直线的点的集合（需约 n^3 次运算以比较 n 个点中的每一个与 $n(n-1)/2$ 条直线中的每一条）
- 计算量大！





- Hough变换的基本思想：点 - 线对偶性

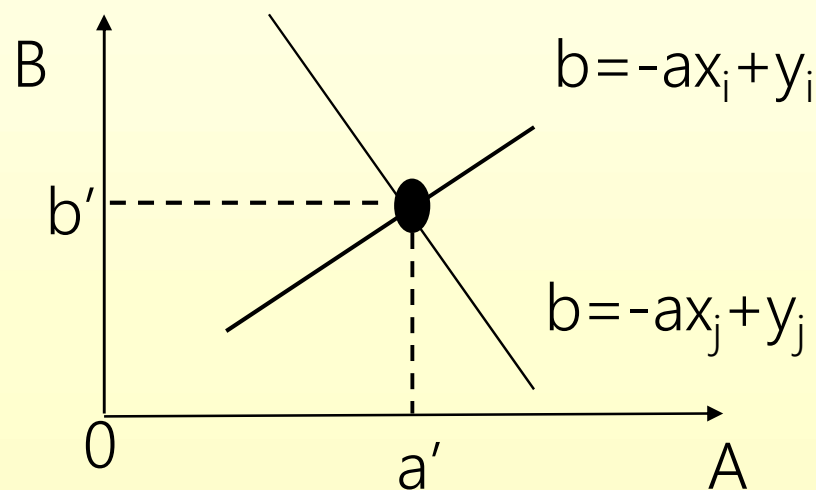
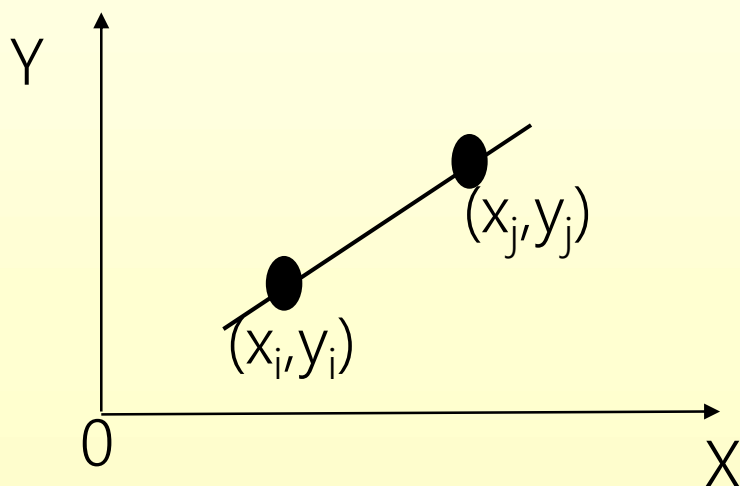
- 对于边界上的n个点的点集，找出共线的点集和直线方程。
- 对于任意两点的直线方程： $y = ax + b$ ，构造一个参数ab的平面：
- $b = -ax + y$





• 点 - 线对偶性

- 图像空间中共线的点对应在参数空间里相交的线
- 在参数空间中相交于同1点的所有直线在图像空间里都有共线的点与之对应。



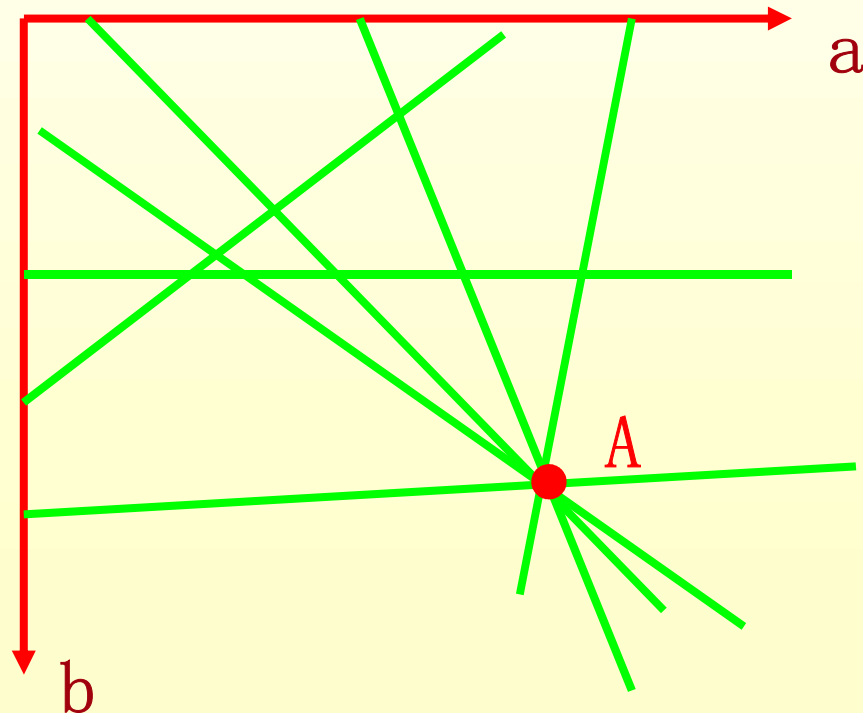
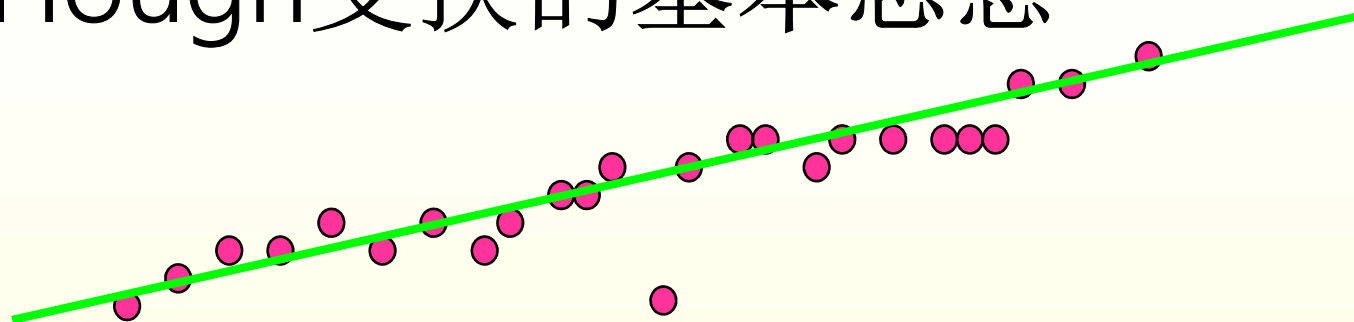


- Hough变换的基本思想

- xy 平面上的任意一条直线，对应参数 ab 平面上都有一个点。
- 过 xy 平面一个点 (x,y) 的所有直线，构成参数 ab 平面上的一条直线。
- 如果点 (x_1,y_1) 与点 (x_2,y_2) 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。



- Hough变换的基本思想





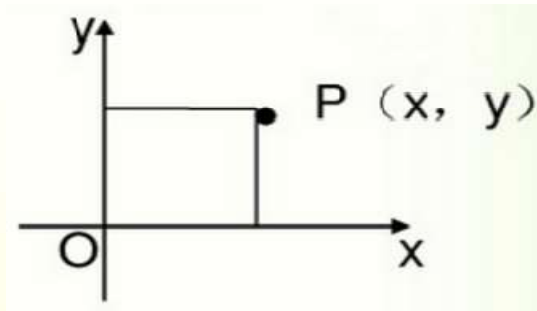
- Hough变换算法实现

- $y = ax + b$. 如果直线接近竖直方向，则会由于 a, b 的值接近无穷大，使计算量大增，我们改用极坐标形式：
$$x \cos \theta + y \sin \theta = \rho$$
- 参数平面为 θ, ρ ，对应不是直线而是正弦曲线。原图像空间中的点对应新参数空间中的一条正弦曲线。
- 使用交点累加器，或交点统计直方图，找出相交线段最多的参数空间的点，然后找出该点对应的 xy 平面的直线线段。

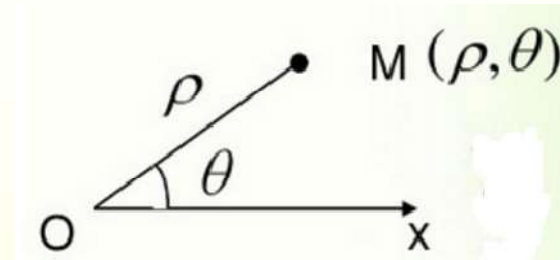


北京航空航天大学

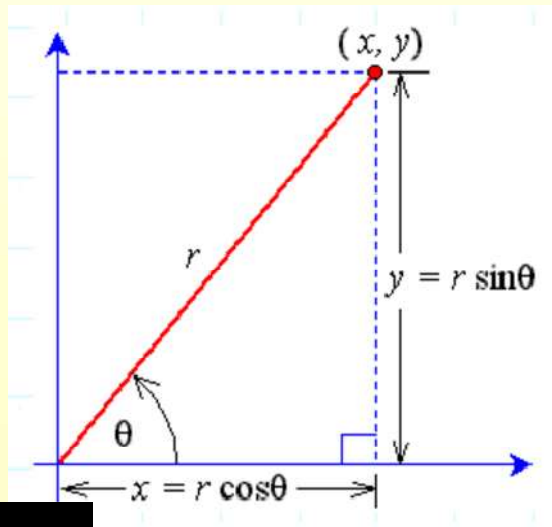
在直角坐标系下，点P的坐标表示



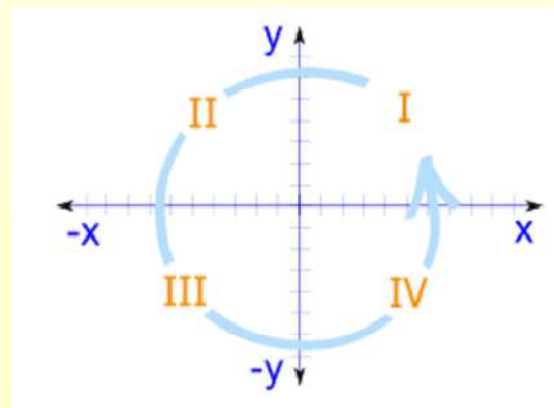
在极坐标下，点P的坐标表示



某一点由极坐标转换为直角坐标

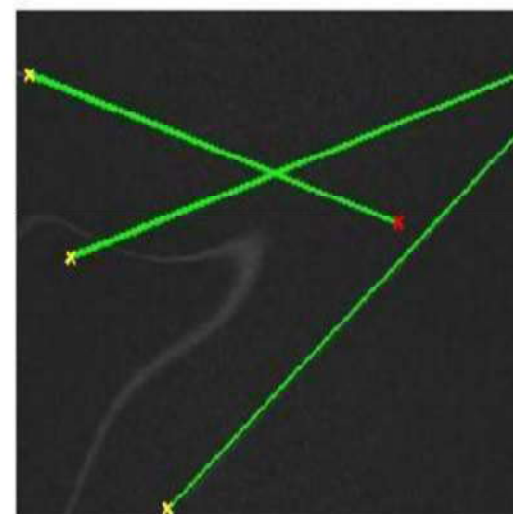
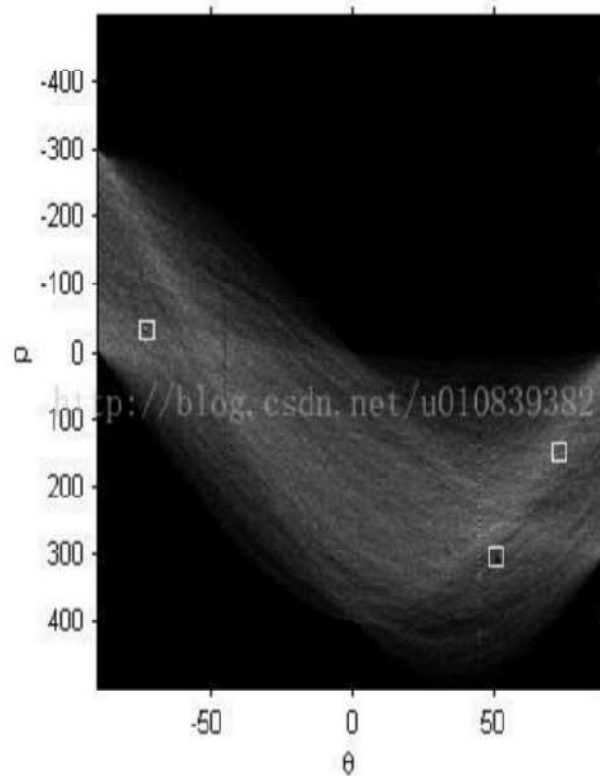
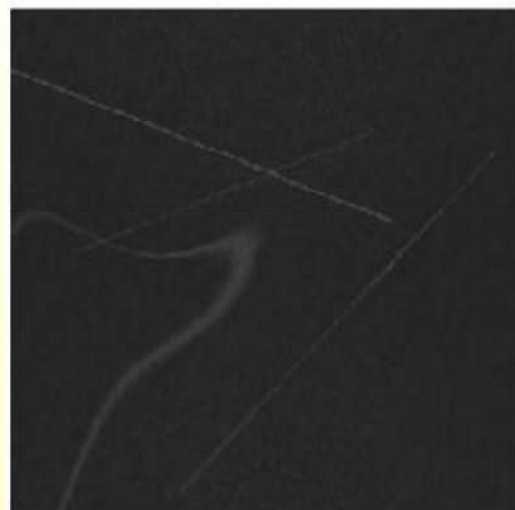


某一点直角坐标转换为极坐标



根据点具体落到四个象限中的哪一个，在arctan结果的基础上进行修正

$$(\sqrt{x^2 + y^2}, \tan^{-1}(y/x)) \Rightarrow (r, \theta)$$





北京航空航天大学



(a).原图像



(a).Canny 算子进行边缘检测结果



(b).Hough 变换检测直线



- 1、边界分割法：点、线、边的检测
- 2、边缘连接法
- 3、阈值分割法
- 4、面向区域的分割



- 阈值分割法
 - 通过交互方式得到阈值
 - 简单全局阈值分割
 - 通过直方图得到阈值
 - 通过边界特性选择阈值
 - 分割连通区域
 - 基于多个变量的阈值



• 阈值分割法

— 阈值分割法的基本思想：

- 确定一个合适的阈值T（阈值选定的好坏是此方法成败的关键）。
- 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。

If $f(x,y) \geq T$ set 255

Else set 0

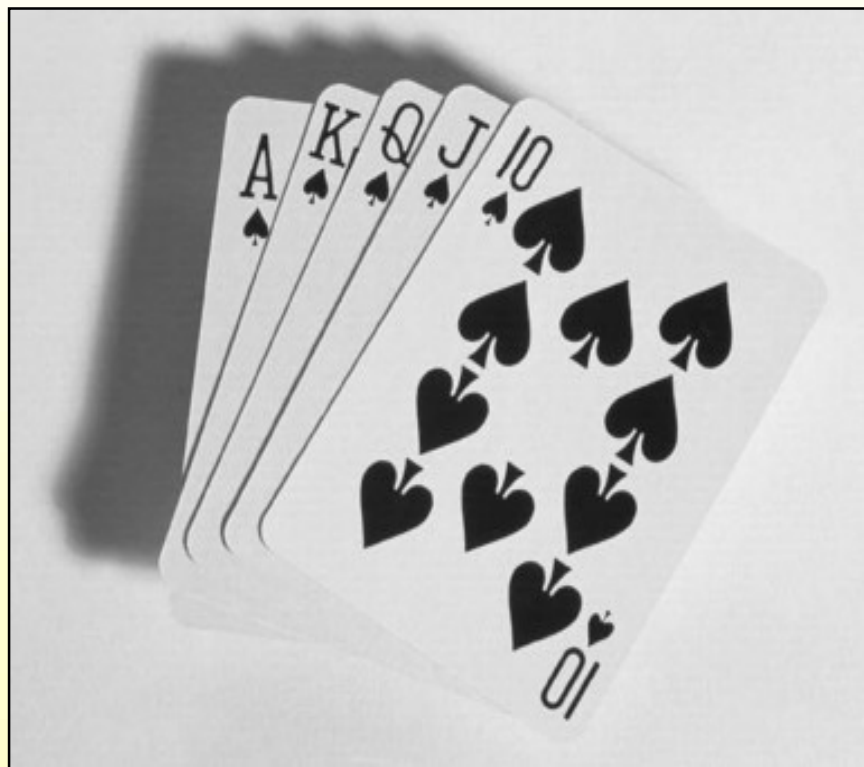
- 在四邻域中有背景的像素，即是边界像素。
- 全局阈值

0	0	255
0	255	255
255	255	255

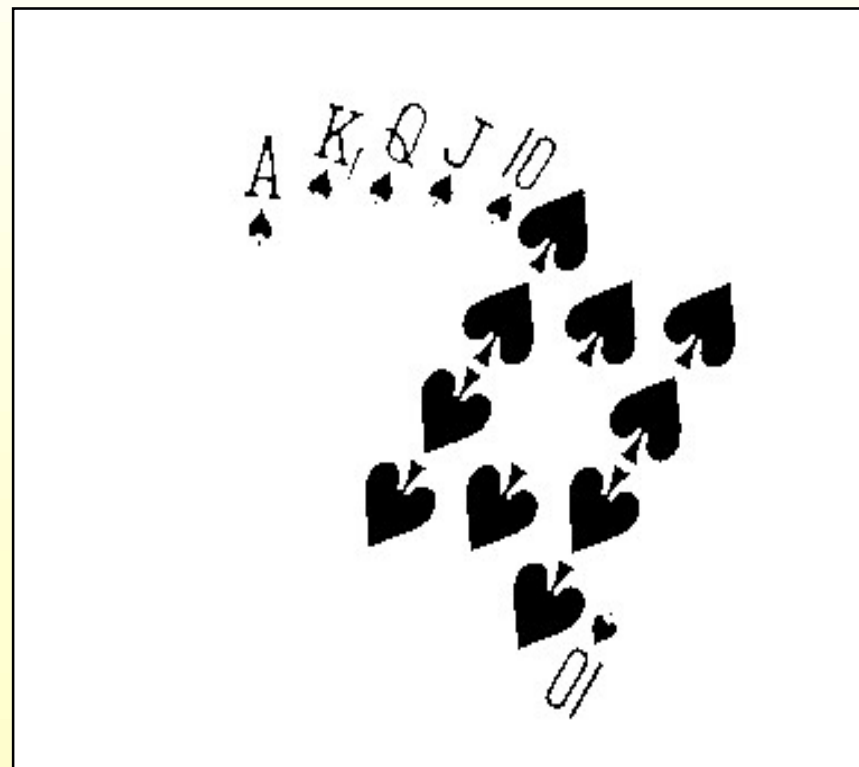




- 例子：设想电脑玩家手中的扑克牌，我们需要对其进行视觉上的分析



原始图像

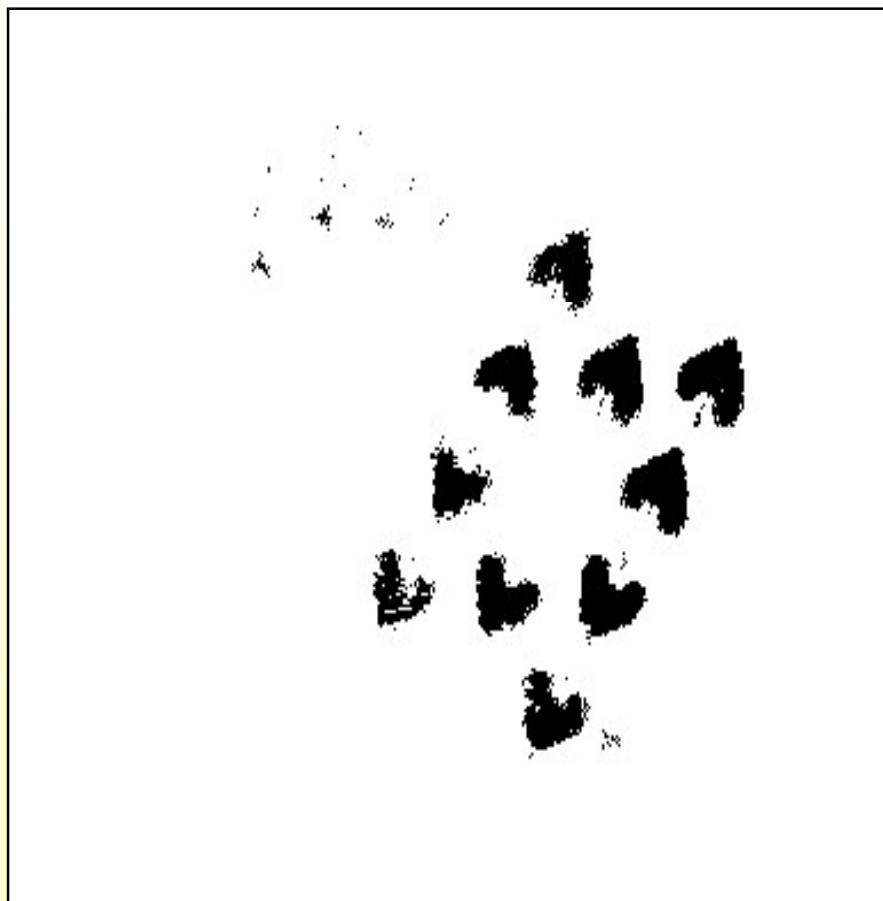


阈值图像

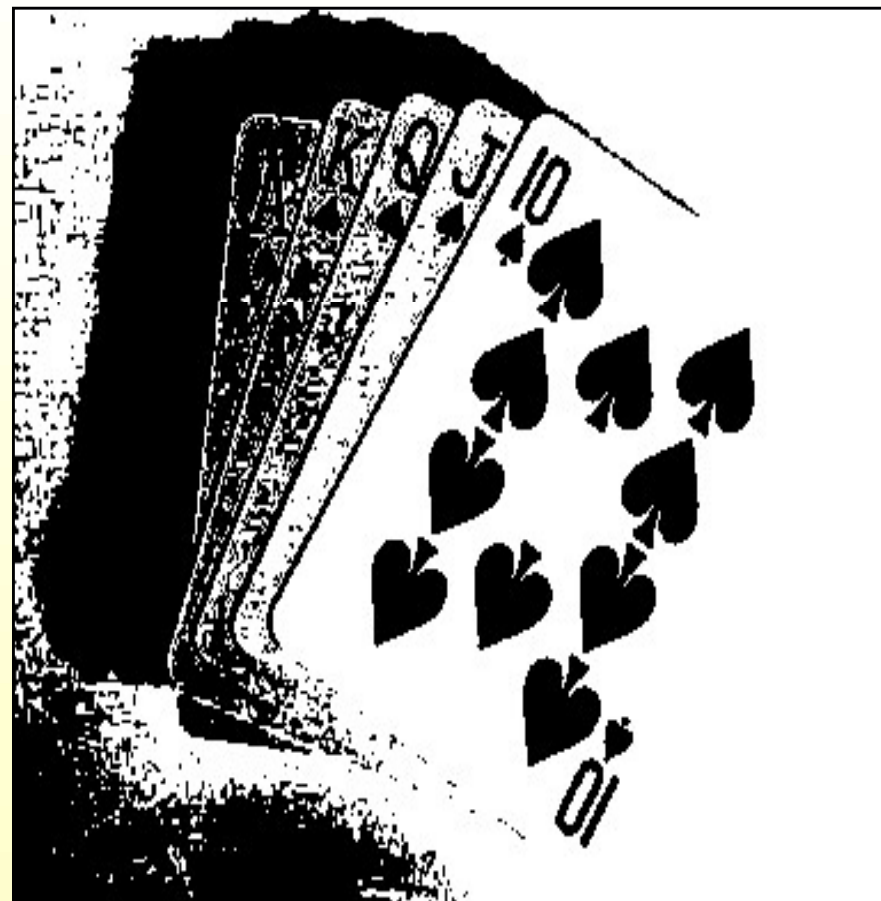


3—阈值分割法

- 如果设置了错误的阈值，结果会很糟糕



小的阈值



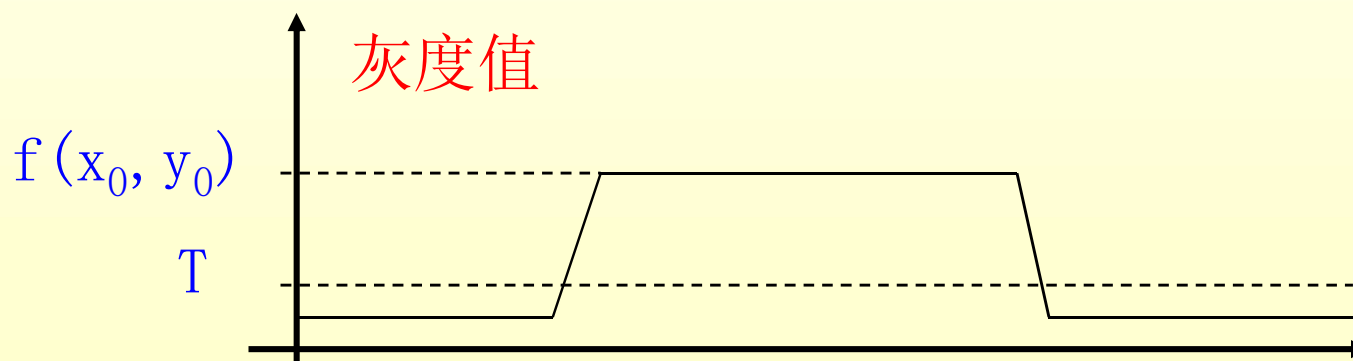
太大的阈值



• 阈值分割法

— 阈值分割法的特点：

- 适用于物体与背景有较强对比的情况，重要的是背景或物体的灰度比较单一。（可通过先求背景，然后求反得到物体）
- 这种方法总可以得到封闭且连通区域的边界。





- 通过交互方式得到阈值

- 基本思想：

- 通过交互方式，得到对象（或背景）的灰度值，比得到阈值 T 容易得多。

假设：对象的灰度值(也称样点值)为 $f(x_0, y_0)$, 且： $T = f(x_0, y_0) - R$ 有：

$$f(x, y) \geq T$$

$$f(x, y) \geq f(x_0, y_0) - R$$

$$|f(x, y) - f(x_0, y_0)| \leq R$$

其中 R 是容忍度，可通过试探获得。



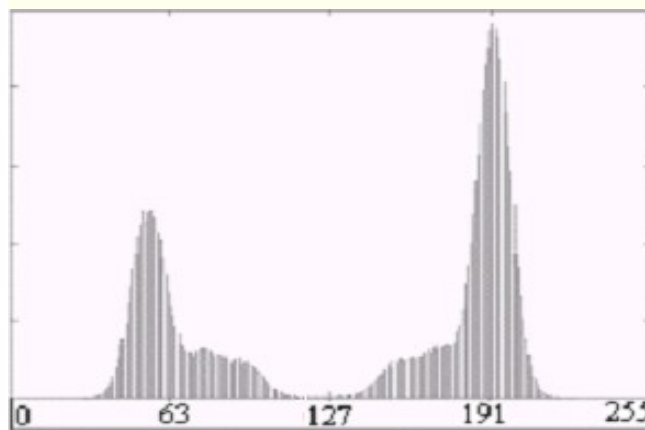
- 通过交互方式得到阈值
 - 实施方法：
 - (1) 通过光标获得样点值 $f(x_0, y_0)$
 - (2) 选取容忍度 R
 - (3) if $|f(x, y) - f(x_0, y_0)| \leq R$ set 255
else set 0



- **全局阈值分割**，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： $G1$ 包括灰度级大于 T 的像素， $G2$ 包括灰度级小于等于 T 的像素
 - 3、计算 $G1$ 中像素的平均值并赋值给 μ_1 ，计算 $G2$ 中像素的平均值并赋值给 μ_2
 - 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

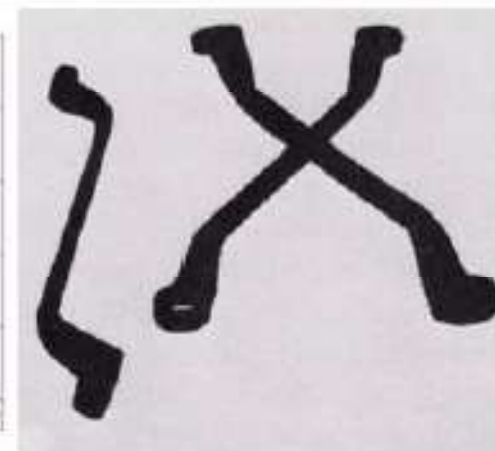
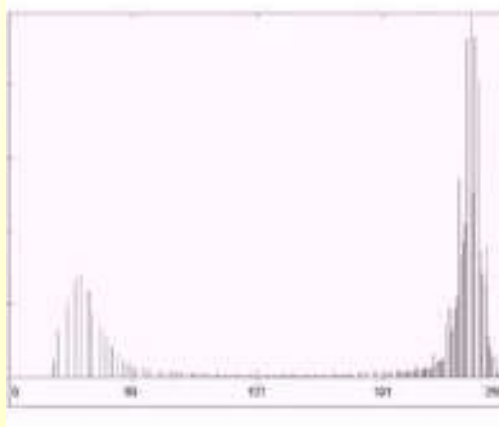
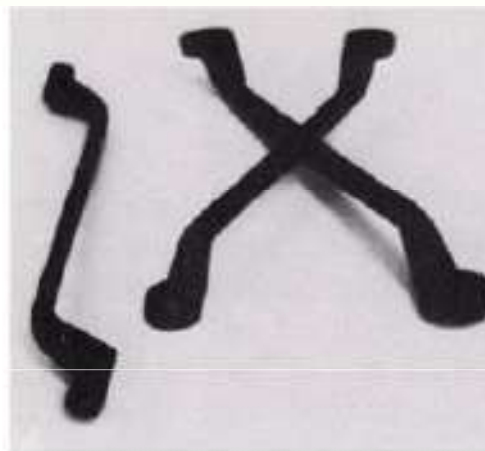


- 通过算法迭代产生全局阈值





- 简单阈值法
 - 经验知识
 - $0.5(\min + \max)$





任意选取一个灰度值 t ，则可以将图像分成前后两部分。我们称这两部分分别为 A 和 B 。对应的就是前景色和背景色。这两部分各自的平均值成为 MA 和 MB 。

A 部分里的像素数占总像素数的比例记作 PA ， B 部分里的像素数占总像素数的比例记作 PB 。

则整个图像的均值为 $M = PA * MA + PB * MB$

Nobuyuki Otsu 给出的类间方差定义为：

$$ICV = PA * (MA - M)^2 + PB * (MB - M)^2$$

这个最佳的阈值 t 就是使得 ICV 最大的那个值。

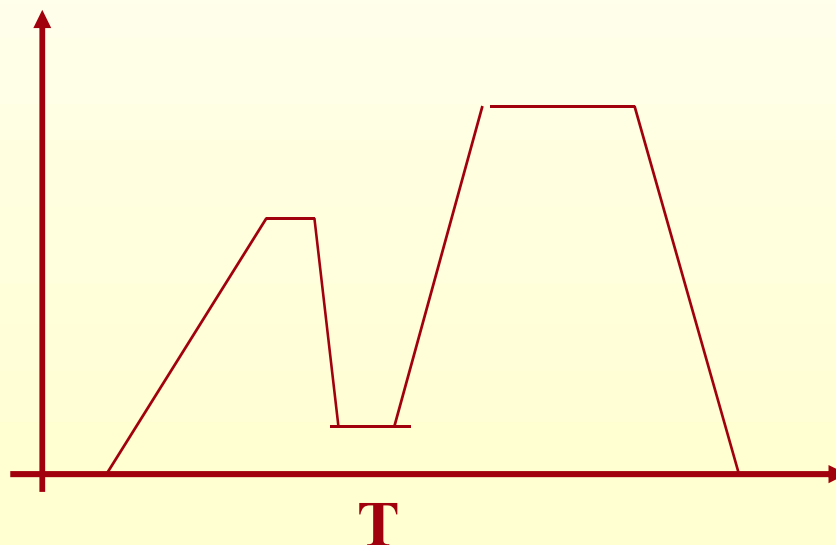
遍历灰度值 t 得到最佳阈值。



- 通过直方图得到阈值

- 基本思想

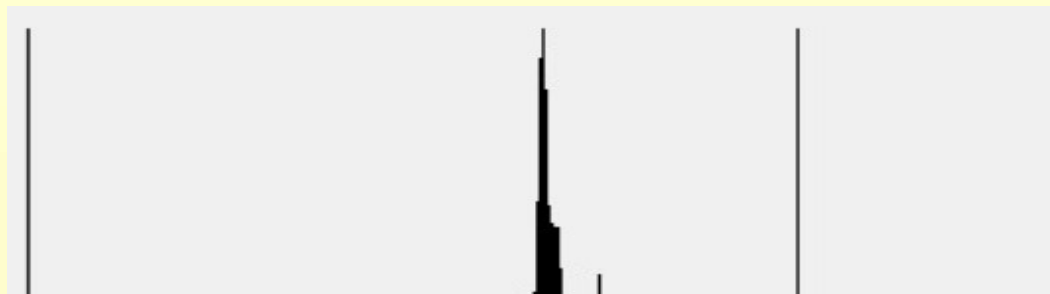
边界上的点的灰度值出现次数较少





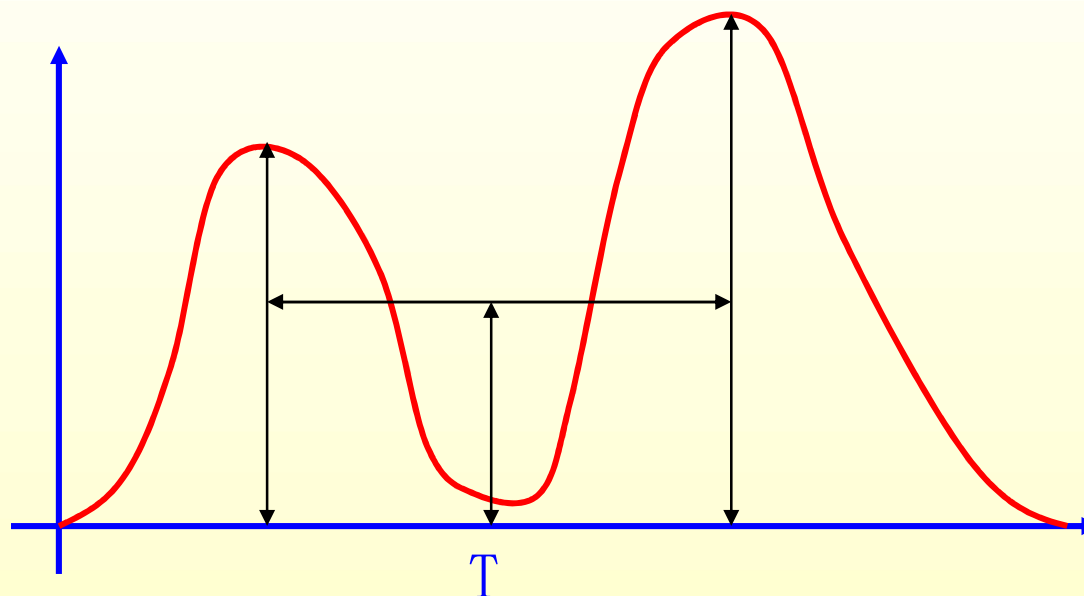
- 通过直方图得到阈值

- 取值方法：取直方图谷底，为最小值的灰度值作为阈值 T
- 缺点：会受到噪音的干扰，最小值不是预期的阈值，而偏离期望的值；
- 改进1：由于峰值代表的是区域内的典型值，一般情况下比选谷底更可靠，可排除噪音干扰。
- 改进2：对直方图进行平滑处理，如最小二乘法



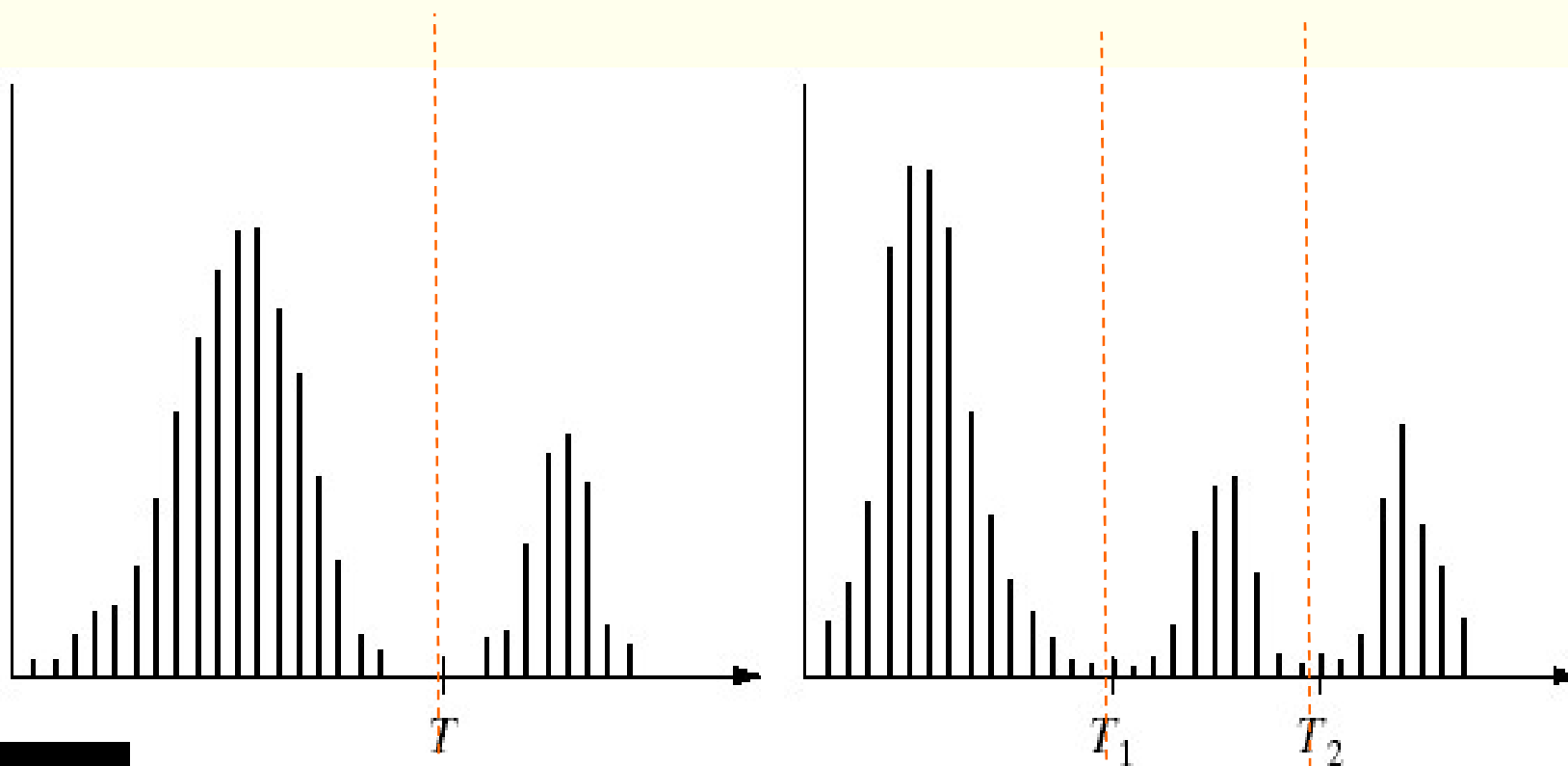


- 通过直方图得到阈值





- 单值阈值只能对双峰直方图工作得较好
- 对于其它类型的直方图，需要更多的阈值





• 通过边界特性选择阈值

— 基本思想：

- 如果直方图的各个波峰很高、很窄、对称，且被很深的波谷分开时，有利于选择阈值。
- 为了改善直方图的波峰形状，我们只把区域边缘的像素绘入直方图，而不考虑区域中间的像素。
- 用微分算子，处理图像，使图像只剩下边界中心两边的值。



- 通过边界特性选择阈值

- 基本思想：

- 这种方法有以下优点：

- 1)在前景和背景所占区域面积差别很大时，不会造一个灰度级的波峰过高，而另一个过低
 - 2)边缘上的点在区域内还是区域外的概率是相等的，因此可以增加波峰的对称性
 - 3)基于梯度和拉普拉斯算子选择的像素，可以增加波峰的高度



- 通过边界特性选择阈值
 - 算法的实现：
 - 1) 对图像进行梯度计算，得到梯度图像。
 - 2) 得到梯度值最大的那一部分（比如10%）的像素直方图
 - 3) 通过直方图的谷底，得到阈值T。
 - 如果用拉普拉斯算子，不通过直方图，直接得到阈值，方法是使用拉普拉斯算子过滤图像，将0跨越点对应的灰度值为阈值T。
- 本质：通过局部区域特性优化检测阈值



- 分割连通区域

- 基本思想：用前述方法获得阈值 T ，并产生一个二值图，区分出单独的连通前景对象和背景区域

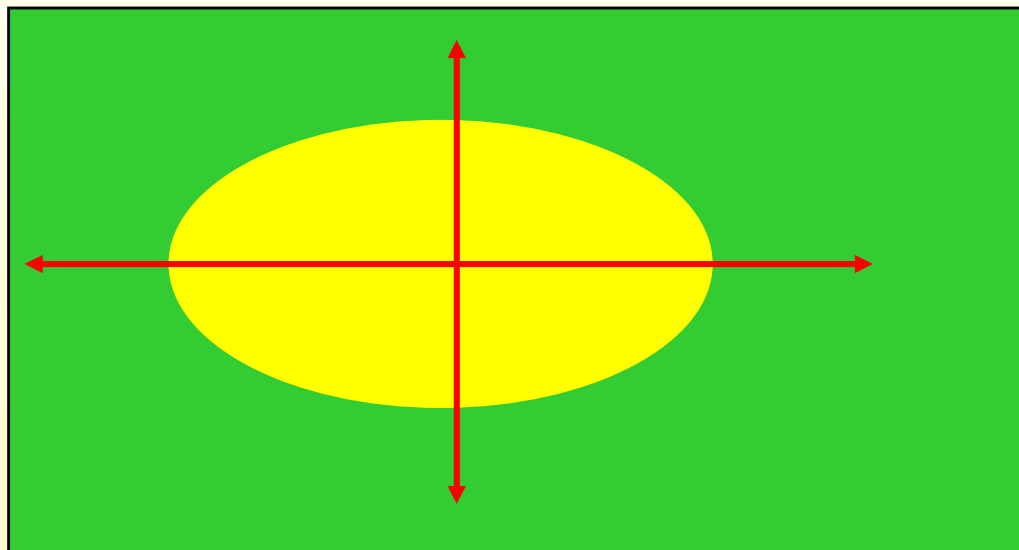
- 算法实现：

- 规定一个阈值 T ，上下左右4个方向进行逐行扫描图像
 - 凡灰度级大于 T 的，颜色置为255；凡灰度级小于 T 的，颜色置为0。



- 分割连通区域

先左后右，先上半部分、后下半部分





- 基于多个变量的阈值

- 基本思想：把前面的方法扩展到多维空间，则寻找波谷的过程，变为寻找点簇的过程。
- 算法实现：
 - 各维分量波谷之间进行逻辑与运算，从波谷重合的点，得到实际的阈值 T 。
- 应用场合：有多个分量的颜色模型，如RGB模型、CMYK模型、HSI模型



- 1、边界分割法：点、线、边的检测
- 2、边缘连接法
- 3、阈值分割法
- 4、面向区域的分割



- 面向区域的分割
 - 基本概念
 - 通过像素集合的区域增长
 - 区域分裂与合并



- 基本概念

- 目标：将区域R划分为若干个子区域

- R_1, R_2, \dots, R_n ，这些子区域满足5个条件：

- 1)完备性：
$$\bigcup_{i=1}^n R_i = R$$
 分割应将图像中的每个像素都分进某1个子区域中

- 2)连通性：每个 R_i 都是一个连通区域

- 3)独立性：对于任意 $i \neq j$ ， $R_i \cap R_j = \emptyset$

各个子区域互不重叠



- 基本概念

4)单一性：每个区域内的灰度级相等，

$$P(R_i) = \text{TRUE}, i = 1, 2, \dots, n$$

属于同一区域的像素应该具有某些相同特性

5)互斥性：任两个区域的灰度级不等，

$$P(R_i \cup R_j) = \text{FALSE}, i \neq j$$

属于不同区域的像素应该具有某些不同特性



- 通过像素集合的区域增长

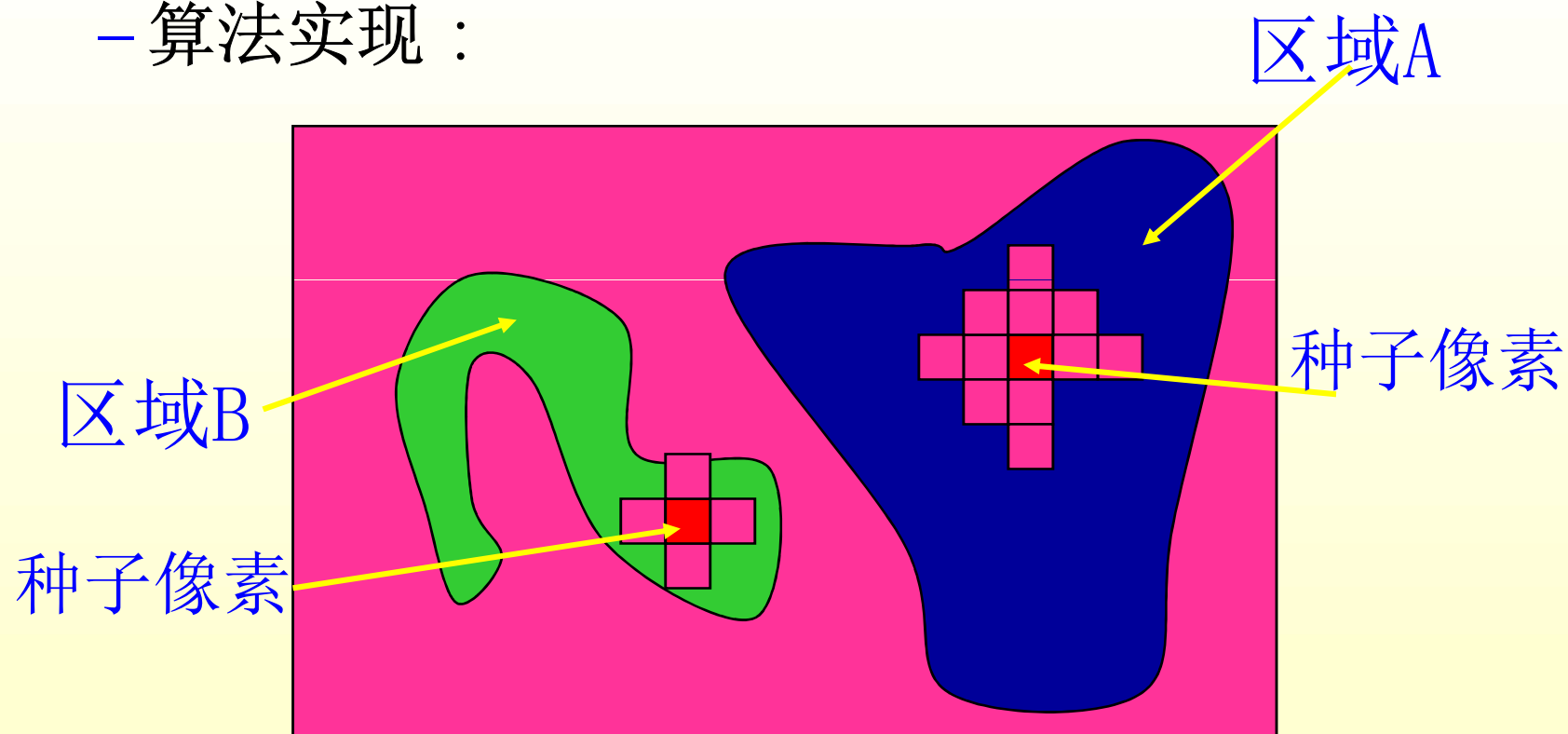
- 算法实现：

- 1) 根据图像的不同应用选择一个或一组种子，它或者是最亮或最暗的点，或者是位于点簇中心的点。
- 2) 选择一个描述符（条件）
- 3) 从该种子开始向外扩张，首先把种子像素加入集合，然后不断将与集合中各个像素连通、且满足描述符的像素加入集合
- 4) 上一过程进行到不再有满足条件的新结点加入集合为止。



- 通过像素集合的区域增长

— 算法实现：





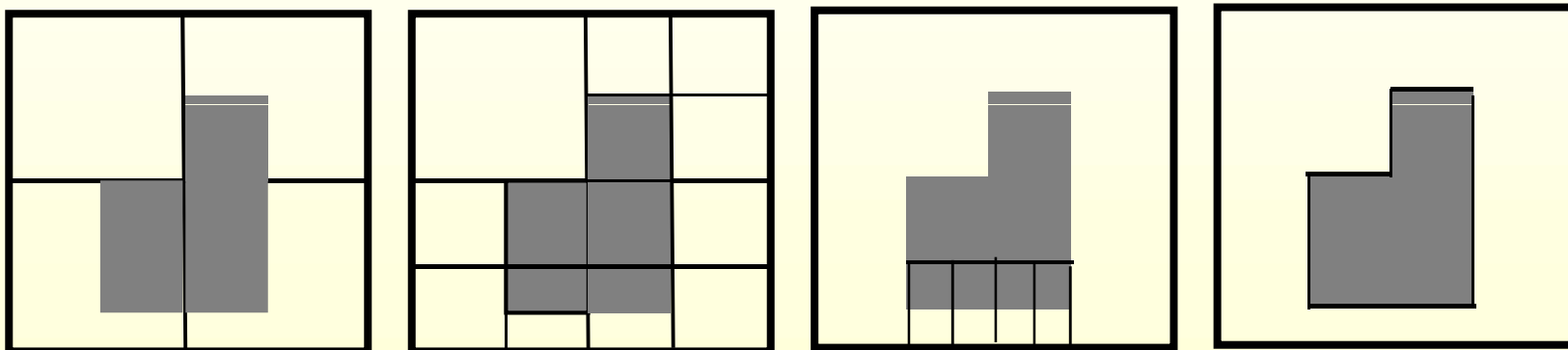
- 区域分裂与合并

— 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（分裂）。
- 2) 如果相邻的子区域所有像素的灰度级相同，则将其合并（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。



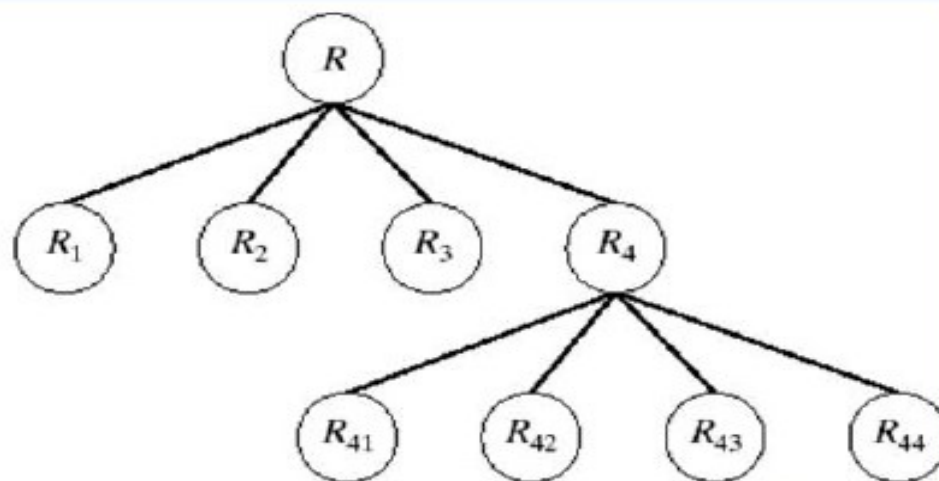
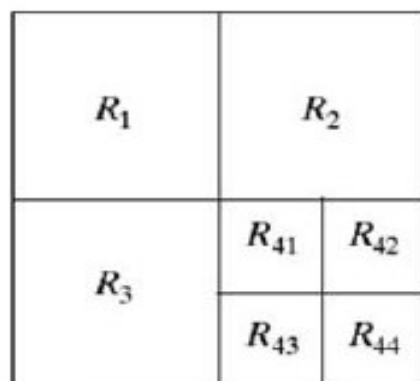
- 区域分裂与合并
 - 算法实现：





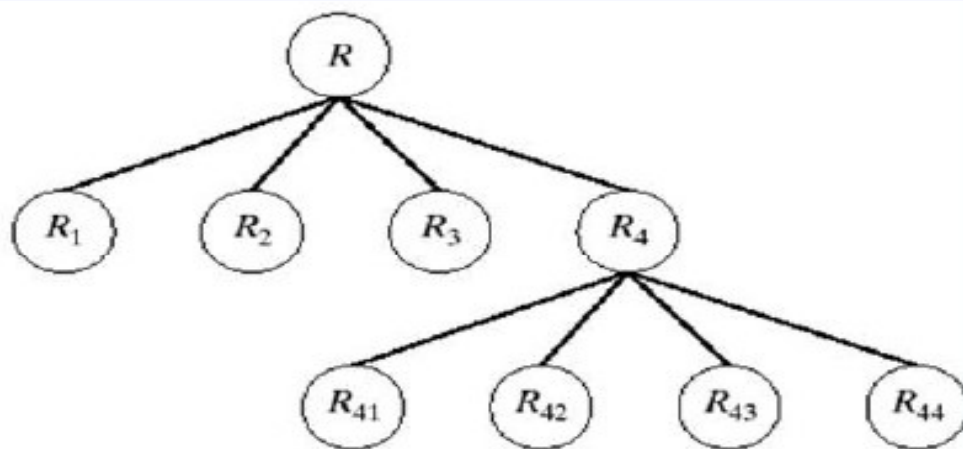
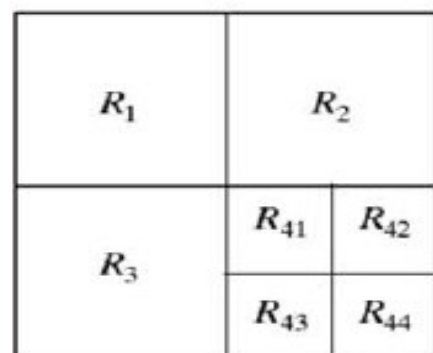
区域拆分与合并

- R 表示整幅图像区域， P 表示一个分割谓词
- 区域拆分的过程就是反复将分割的到的子区域在此分为四个区域，直到对任意区域 R_i ，都有 $P(R_i) = \text{TRUE}$





区域拆分与合并



➤ 如果仅拆分，最后分区可能会包含具有相同性质的相邻区域

□ 通过合并来矫正

□ $P(R_i \cup R_j) = \text{TRUE}$ 时合并

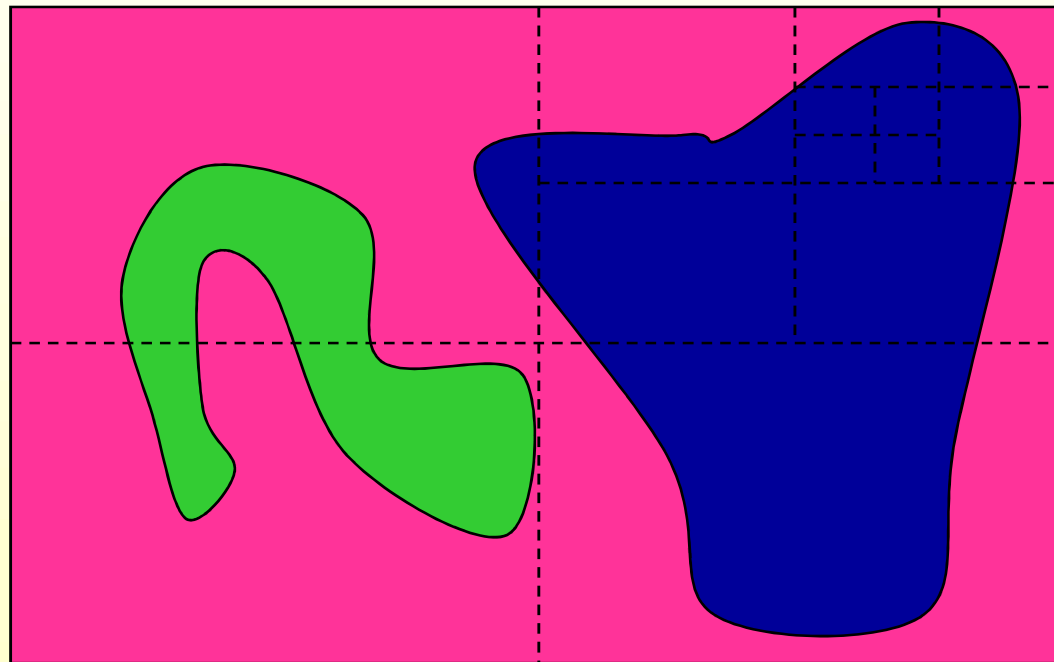


区域拆分与合并

- 对于图像中任何区域 R_i , 如果 $P(R_i) = \text{FALSE}$, 则将 R_i 拆分为4个相连的象限区域。
- $P(R_i \cup R_j) = \text{TRUE}$ 的任意两个相邻区域进行合并。
- 反复进行上两步操作, 直至不再有新的拆分与合并为止。



- 区域分裂与合并
 - 算法实现：





- 区域分裂与合并

- 算法实现：实际应用中还可作以下修改：

- $P(R_i)$ 的定义为：

- 1) 区域内多于80%的像素满足不等式

- $|z_j - m_i| \leq 2\sigma_i,$

- 其中： z_j 是区域 R_i 中第 j 个点的灰度级，

- m_i 是该区域的平均灰度级，

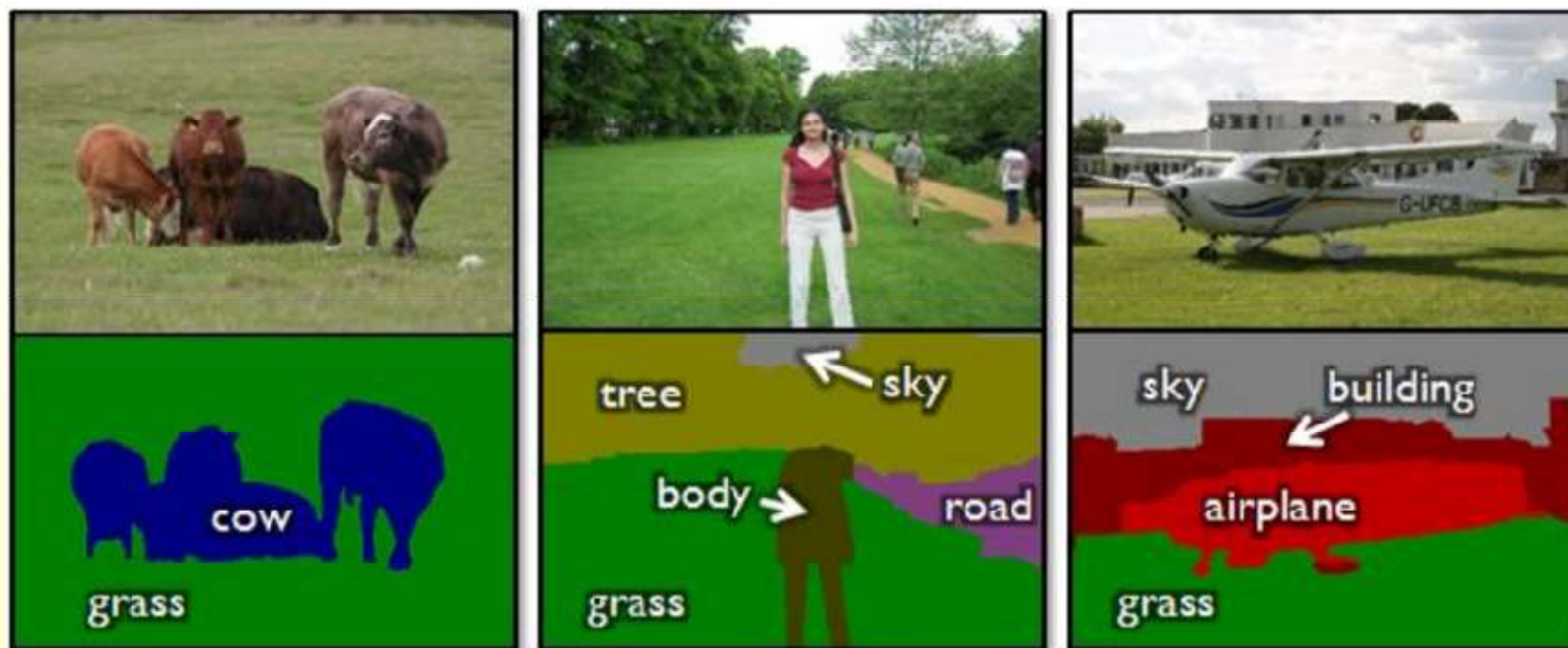
- σ_i 是区域的灰度级的标准方差。

- 2) 当 $P(R_i) = \text{TRUE}$ 时，将区域内所有像素的灰度级置为 m_i 。



北京航空航天大学

目标分割、目标识别、目标检测和目标跟踪



object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat

计算机学院



北京航空航天大学

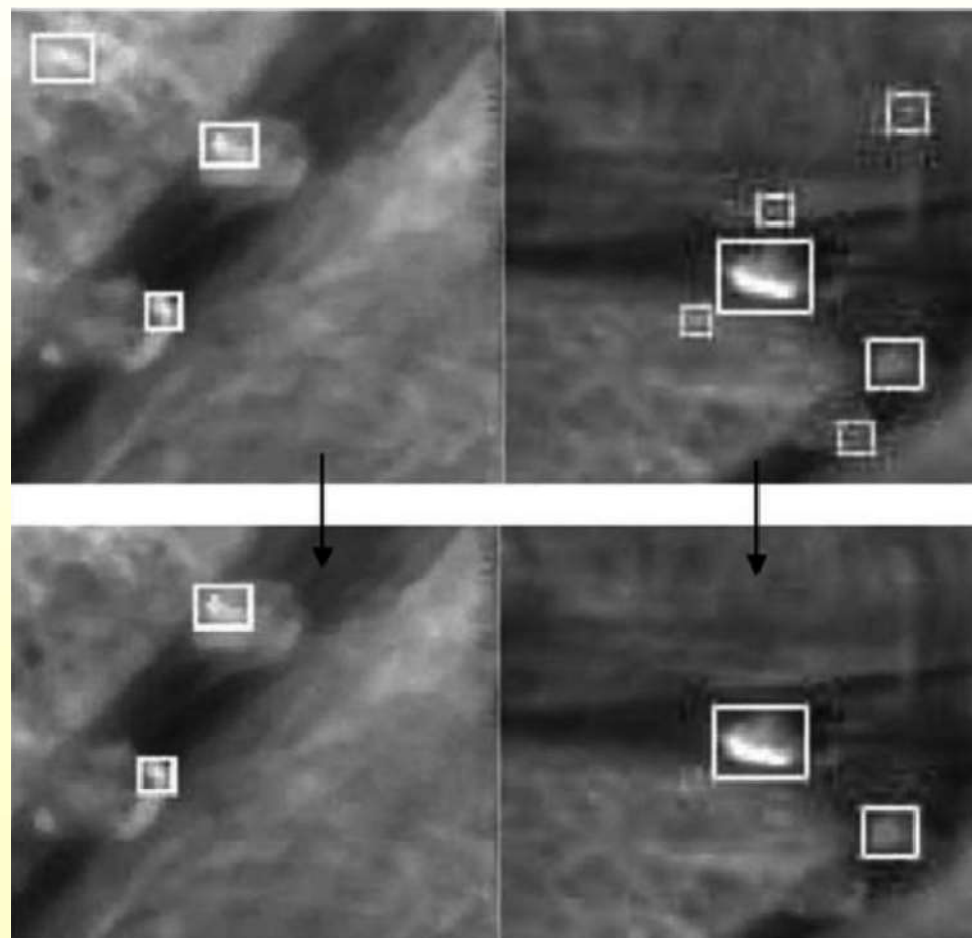
目标分割、目标识别、目标检测和目标跟踪





北京航空航天大学

目标分割、目标识别、**目标检测**和目标跟踪



计算机学院



北京航空航天大学

目标分割、目标识别、目标检测和**目标跟踪**

