



# 第七章：边缘检测

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 ([lihq@ustc.edu.cn](mailto:lihq@ustc.edu.cn))  
周文罡 ([zhwg@ustc.edu.cn](mailto:zhwg@ustc.edu.cn))  
李 礼 ([li11@ustc.edu.cn](mailto:li11@ustc.edu.cn))  
胡 洋 ([eeyhu@ustc.edu.cn](mailto:eeyhu@ustc.edu.cn))



# 边缘检测

---

- ☐ 边缘模型
- ☐ 边缘检测算子
- ☐ 边缘拟合
- ☐ 边缘搜索



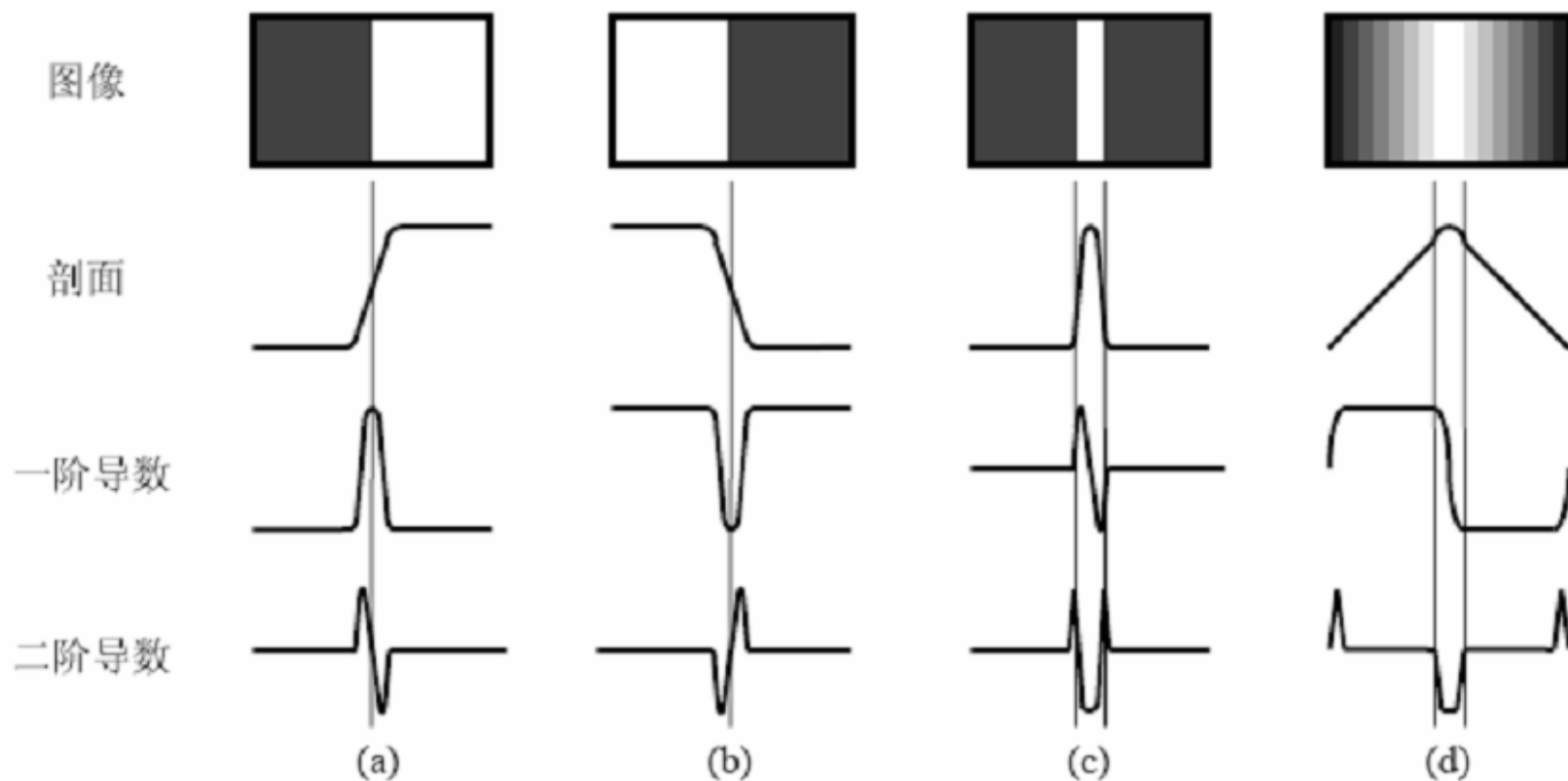
# 边缘检测

---

- 边缘模型
- 边缘检测算子
- 边缘拟合
- 边缘搜索

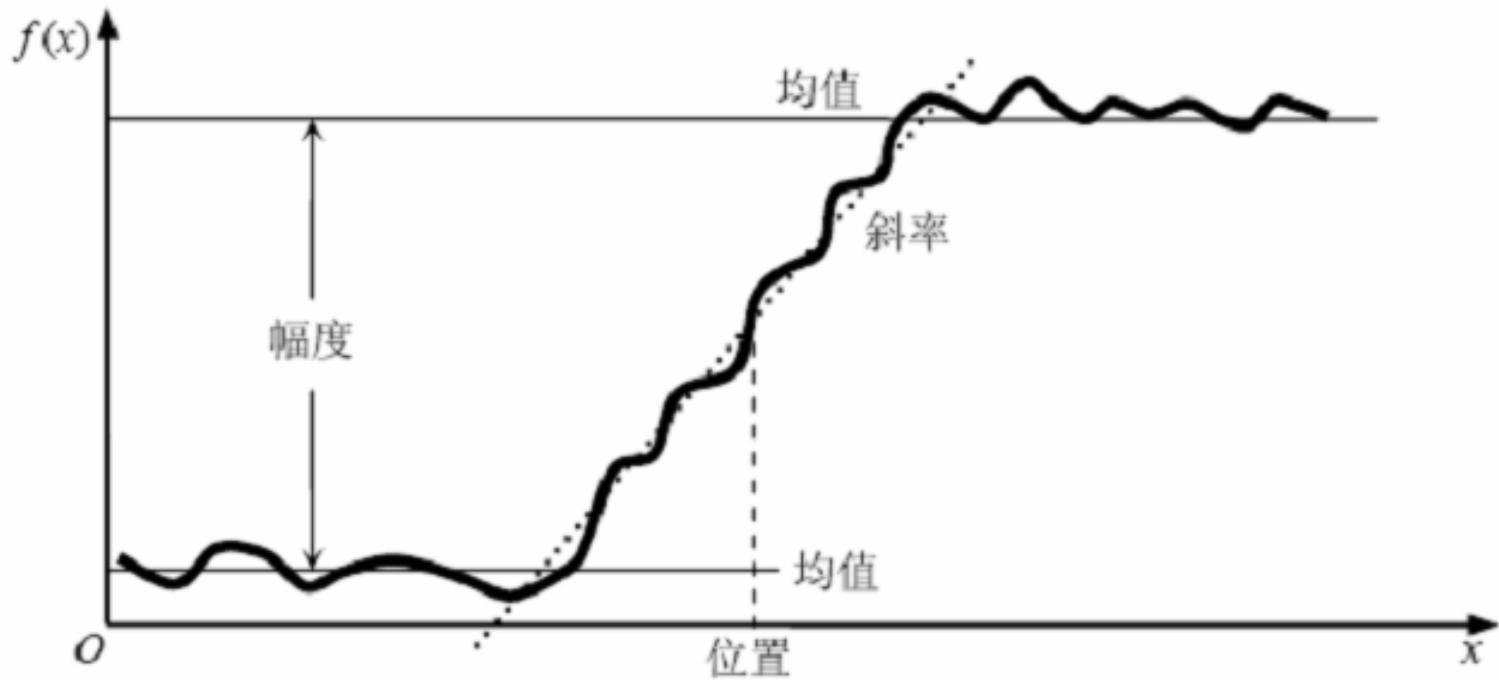
# 边缘模型

常见的边缘剖面：阶梯状、脉冲状、屋顶状



图象边缘模型及其一阶、二阶导数

# 描述边缘的参数



图象边缘模型与描述的参数



# 描述边缘的参数

## □ 位置

- 边缘（等效的）最大灰度不连续处（最重要的参数）

## □ 朝向

- 跨越灰度最大不连续的方向（XY平面）

## □ 幅度

- 灰度不连续方向上的灰度差

## □ 均值

- 分属边缘两边（近邻）像素的灰度均值

## □ 斜率

- 边缘在其朝向上的倾斜程度（幅值）



# 边缘检测

---

- ☐ 边缘模型
- ☐ 边缘检测算子
- ☐ 边缘拟合
- ☐ 边缘搜索



# 边缘检测算子

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# 正交梯度算子

## □ 梯度算子

### ■ 一阶差分算子

矢量  $\nabla f(x, y) = \begin{bmatrix} G_x & G_y \end{bmatrix}^T = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$

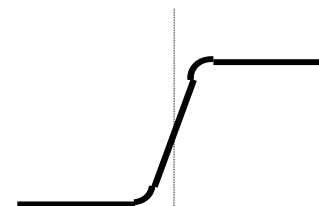
幅度  $\text{mag}(\nabla f) = \left[ G_x^2 + G_y^2 \right]^{1/2}$

方向角  $\phi(x, y) = \arctan(G_y / G_x)$

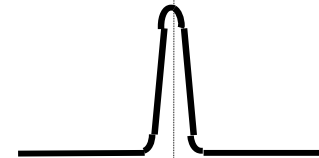
图象



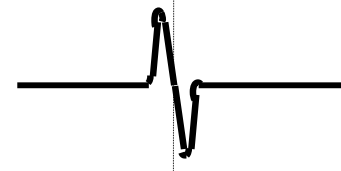
剖面



一阶导数



二阶导数



# 正交梯度算子

## □ 梯度算子

- 利用模板（与图象进行）卷积
- 模板中系数之和为零，使得在恒定灰度区域的相应为0

1	
	-1

	1
-1	

(a) Roberts

-1		1
-1		1
-1		1

1	1	1
-1	-1	-1

(b) Prewitt

-1		1
-2		2
-1		1

1	2	1
-1	-2	-1

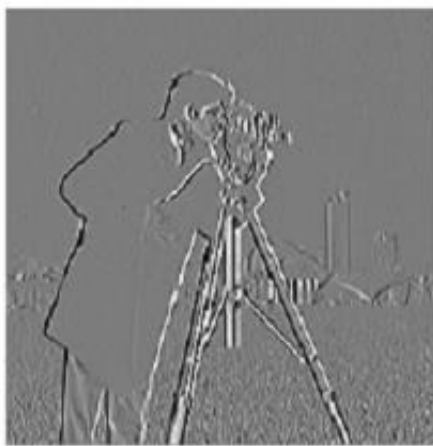
(c) Sobel

- Sobel算子应用最为广泛

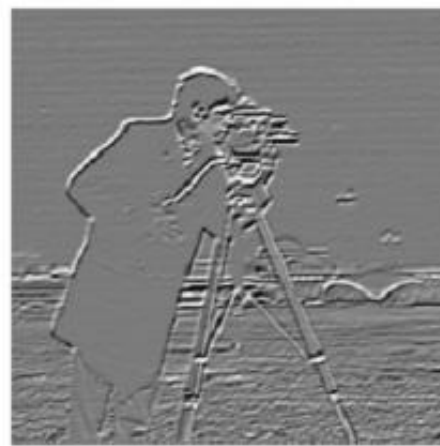
# 梯度图示例



(a)



(b)



(c)



(d)



(e)



(f)

梯度图示例：a)原图；b) Sobel 水平模板；c) Sobel 垂直模板；d) Sobel梯度图（范数2） e) Sobel梯度图（范数1） f) Sobel梯度图（范数 $\infty$ ）



# 边缘检测算子

---

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子

# 方向微分算子

## □ 基于特定方向上的微分来检测边缘

八方向Kirsch ( $7 \times 7$ ) 模板

<table><tr><td>-5</td><td>3</td><td>3</td></tr><tr><td>-5</td><td></td><td>3</td></tr><tr><td>-5</td><td>3</td><td>3</td></tr></table>	-5	3	3	-5		3	-5	3	3	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>-5</td><td>0</td><td>3</td></tr><tr><td>-5</td><td>-5</td><td>3</td></tr></table>	3	3	3	-5	0	3	-5	-5	3	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>3</td><td>0</td><td>3</td></tr><tr><td>-5</td><td>-5</td><td>-5</td></tr></table>	3	3	3	3	0	3	-5	-5	-5	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>3</td><td>0</td><td>-5</td></tr><tr><td>3</td><td>-5</td><td>-5</td></tr></table>	3	3	3	3	0	-5	3	-5	-5
-5	3	3																																					
-5		3																																					
-5	3	3																																					
3	3	3																																					
-5	0	3																																					
-5	-5	3																																					
3	3	3																																					
3	0	3																																					
-5	-5	-5																																					
3	3	3																																					
3	0	-5																																					
3	-5	-5																																					
<table><tr><td>3</td><td>3</td><td>-5</td></tr><tr><td>3</td><td></td><td>-5</td></tr><tr><td>3</td><td>3</td><td>-5</td></tr></table>	3	3	-5	3		-5	3	3	-5	<table><tr><td>3</td><td>-5</td><td>-5</td></tr><tr><td>3</td><td>0</td><td>-5</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	3	-5	-5	3	0	-5	3	3	3	<table><tr><td>-5</td><td>-5</td><td>-5</td></tr><tr><td>3</td><td>0</td><td>3</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	-5	-5	-5	3	0	3	3	3	3	<table><tr><td>-5</td><td>-5</td><td>3</td></tr><tr><td>-5</td><td>0</td><td>3</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	-5	-5	3	-5	0	3	3	3	3
3	3	-5																																					
3		-5																																					
3	3	-5																																					
3	-5	-5																																					
3	0	-5																																					
3	3	3																																					
-5	-5	-5																																					
3	0	3																																					
3	3	3																																					
-5	-5	3																																					
-5	0	3																																					
3	3	3																																					

## □ 模板的对称性 → 模板数减半

- 边缘强度：卷积值的极大值的绝对值
- 边缘方向：卷积值的极大值的符号

# 方向微分算子

## □ 方向微分算子不局限于4个

■ 下面是每隔30度的模板

■ 0度，30度，60度，90度，120度，150度

1.0	1.0	1.0
-1.0	-1.0	-1.0

(a)

1.0	1.0	0.7
0.8		-0.8
-0.7	-1.0	-1.0

(b)

1.0	0.8	-0.7
1.0		-1.0
0.7	-0.8	-1.0

(c)

-1.0		1.0
-1.0		1.0
-1.0		1.0

(d)

-0.7	0.8	1.0
-1.0		1.0
-1.0	-0.8	0.7

(e)

0.7	1.0	1.0
-0.8		0.8
-1.0	-1.0	-0.7

(f)

## □ 基于梯度最大的两个方向可以进一步精细化幅值和方向



# 边缘检测算子

---

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# 边缘检测算子

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# 二阶导数算子

## □ 拉普拉斯算子

### ■ 二阶差分算子（过零点性质）

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

(a)

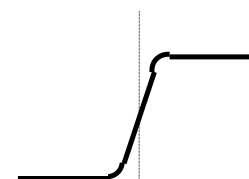
-1	-1	-1
-1	8	-1
-1	-1	-1

(b)

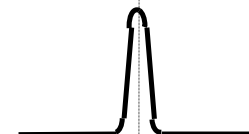
图象



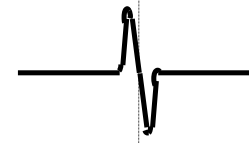
剖面



一阶导数



二阶导数



# 二阶导数算子

## □ 拉普拉斯算子

- 对图象中的噪声相当敏感
- 产生双像素宽的边缘
- 不能提供边缘方向的信息



- 拉普拉斯算子很少直接用于检测边缘，主要用于已经边缘像素后确定该像素是在图像的暗区或明区一边



# 边缘检测算子

---

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# 二阶导数算子

## □ 马尔算子

1. 用一个2-D的高斯平滑模板与源图象卷积  
✓ 等价于低通滤波
2. 计算卷积后图象的拉普拉斯值
3. 检测拉普拉斯图象中的过零点作为边缘点

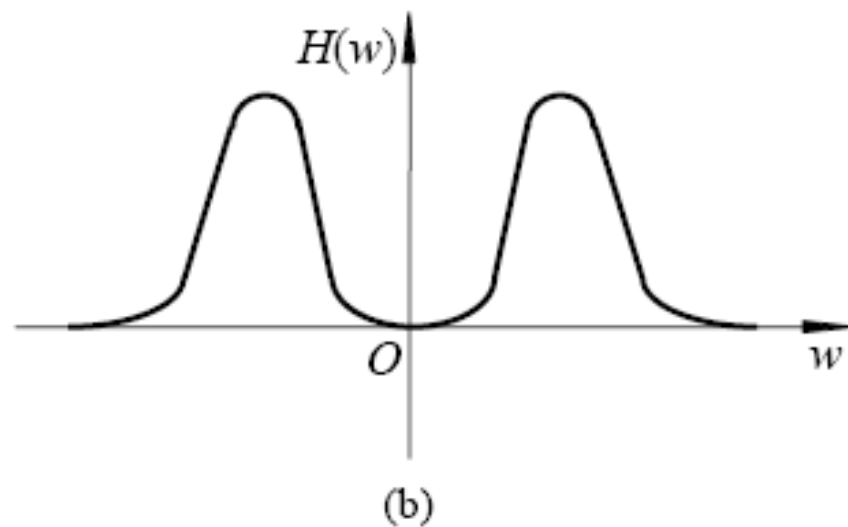
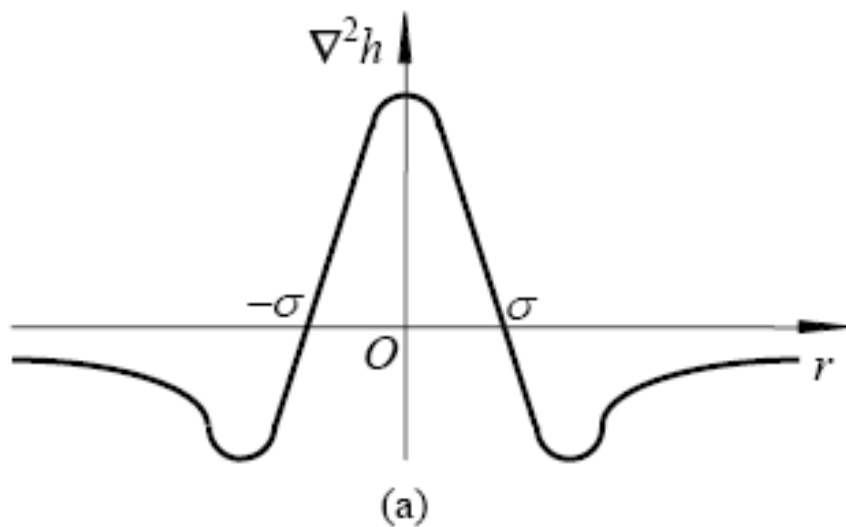
$$h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad g(x, y) = h(x, y) \otimes f(x, y)$$

$$\begin{aligned} \nabla_g^2 &= \nabla^2[h(x, y) \otimes f(x, y)] = \nabla^2 h(x, y) \otimes f(x, y) \\ &= \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \otimes f(x, y) \end{aligned}$$

$$\nabla^2 h = h''(r) = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (LoG)$$

# 二阶导数算子

## □ 马尔算子（墨西哥草帽）



$\nabla^2 h$ 的剖面和对应的转移函数



# 边缘检测算子

- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# 最优边缘检测算子

## □ 坎尼算子

好的边缘检测算子应具有的三个指标

- 低失误概率
  - ✓ 既要少将真正的边缘丢失也要少将非边缘判为边缘
- 高位置精度
  - ✓ 检测出的边缘应在真正的边界上
- 对每个边缘有唯一的响应
  - ✓ 得到的边界为单像素宽

- **J. Canny.** *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986



# 坎尼边缘检测方法

## □ Canny算子近似

- Canny 算子可以用高斯函数的一阶微分算子来近似

## □ Canny边缘检测流程：

- 高斯滤波平滑
- 计算梯度大小与方向
- 非极大值抑制
  - ✓ 实现单像素宽的边缘
- 双阈值检测和连接
  - ✓ 保证低失误差率

- **J. Canny.** *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986





# 高斯平滑及差分

- 先对图像进行高斯卷积平滑滤波，然后计算梯度
  - 卷积算子和差分算子均为线性算子，可以交换运算顺序

$$S = \nabla(I * h) = \nabla(h * I) = (\nabla h) * I = \begin{bmatrix} h_x \\ h_y \end{bmatrix} * I = \begin{bmatrix} h_x * I \\ h_y * I \end{bmatrix}$$

$$\text{高斯滤波器: } h = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$\text{高斯差分滤波器: } h_x = \frac{\partial h}{\partial x}, \quad h_y = \frac{\partial h}{\partial y}$$

$$\text{图像梯度幅值: } G = \sqrt{(h_x * I)^2 + (h_y * I)^2}$$

# 实例结果

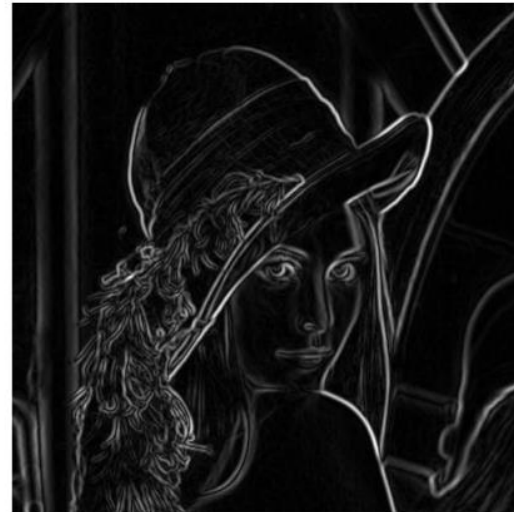
$I$



$h_x * I$



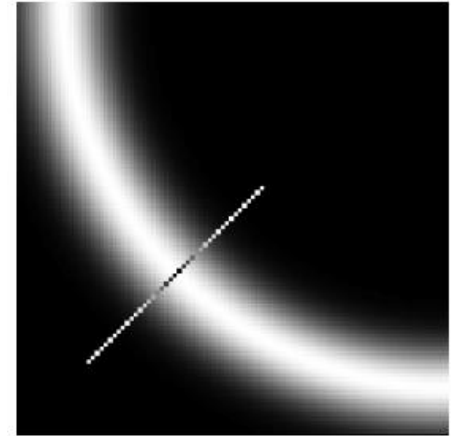
$h_y * I$



# 非极大值抑制(non-maxima suppression, NMS)

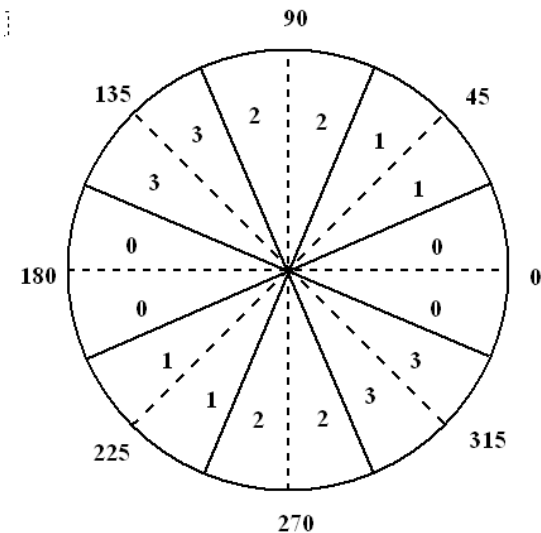
## □ 沿着梯度方向，抑制梯度值非最大的点

- 细化幅值图像 $M[i, j]$ 中的屋脊带(ridge)，只保留幅值局部变化最大的点。
- NMS通过抑制梯度线上所有非屋脊峰值的幅值来细化边缘。



## □ 基于最近邻进行最大值消除

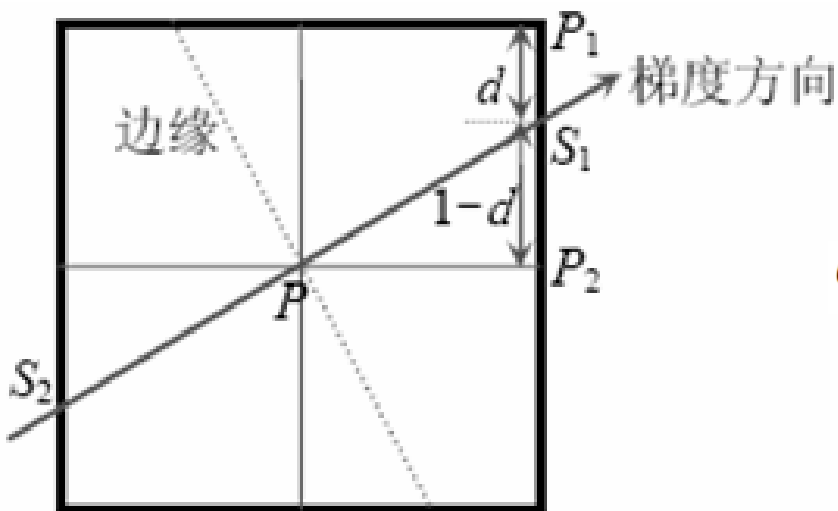
- 将梯度角 $\theta[i, j]$ 的变化范围分为四个扇区  
$$\zeta[i, j] = \text{Sector}(\theta[i, j]);$$
- 用3x3邻域作用于幅值图像 $M[i, j]$ ，邻域中心像素 $M[i, j]$ 与沿着梯度线方向的两个像素进行比较
- 若 $M[i, j]$ 不比沿梯度线方向的两个相邻点幅值大，则像素 $(i, j)$ 被抑制， $M[i, j]$  被置为0。



# 用插值进行非最大消除

## □ 用插值进行最大值消除：

- 通过对相邻单元的梯度幅值的插值估计梯度线上的相邻幅值
- 如果P点的梯度值小于 $S_1$ 或 $S_2$ 的梯度值，则P点梯度值被置为0
- 精确但计算量大



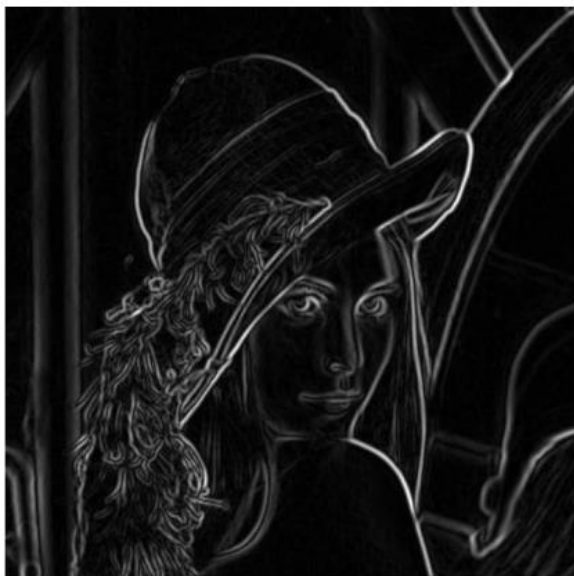
$$G_{S_1} = (1 - d)G_{P_1} + dG_{P_2}$$

2-D 梯度插值

# 实例结果



原图



梯度图



非极大值抑制结果图

# 双阈值算法

- 双阈值算法采用两个阈值 $\tau_1$ 和 $\tau_2$ ，且 $\tau_2 \approx 2\tau_1$
- 得到两个阈值边缘图像 $T_1[i,j]$ 和 $T_2[i,j]$
- $T_2[i,j]$ 含有的假边缘少，但有断点
- 以 $T_2[i,j]$ 为指导，在 $T_1[i,j]$ 中相应8邻域点寻找可以连接到轮廓上的点
- 不断在 $T_1[i,j]$ 收集边缘，直到将 $T_2[i,j]$ 中所有的间隙连接起来为止



原图



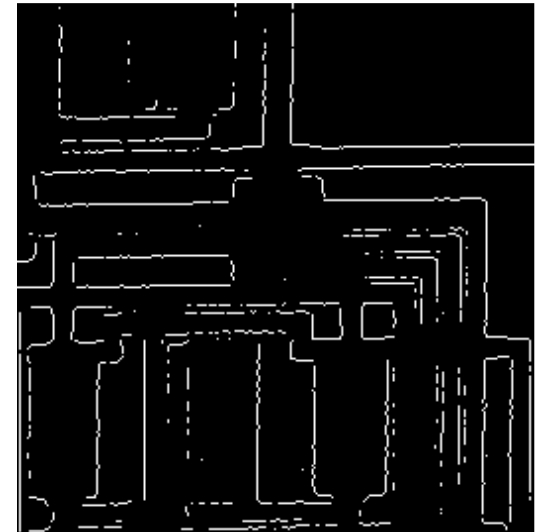
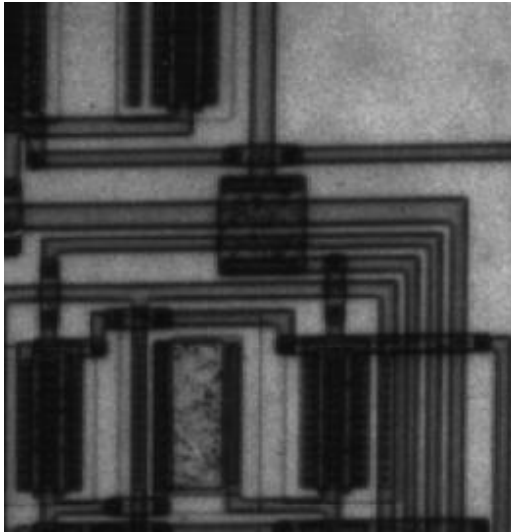
强/弱边缘检测结果



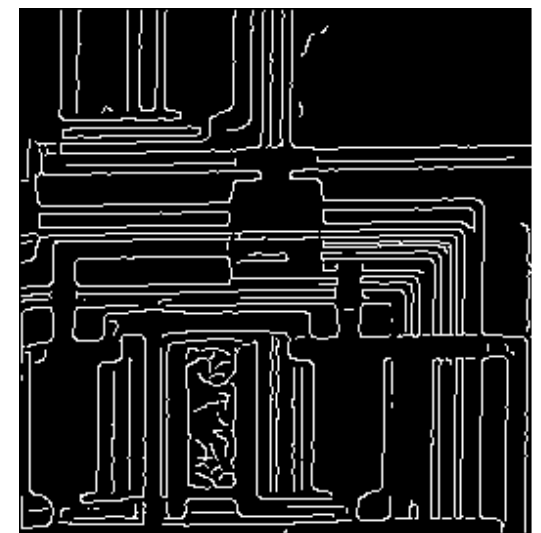
最后边缘检测结果



# 边缘检测对比



Roberts	Sobel
Log	Canny





# 边缘检测算子

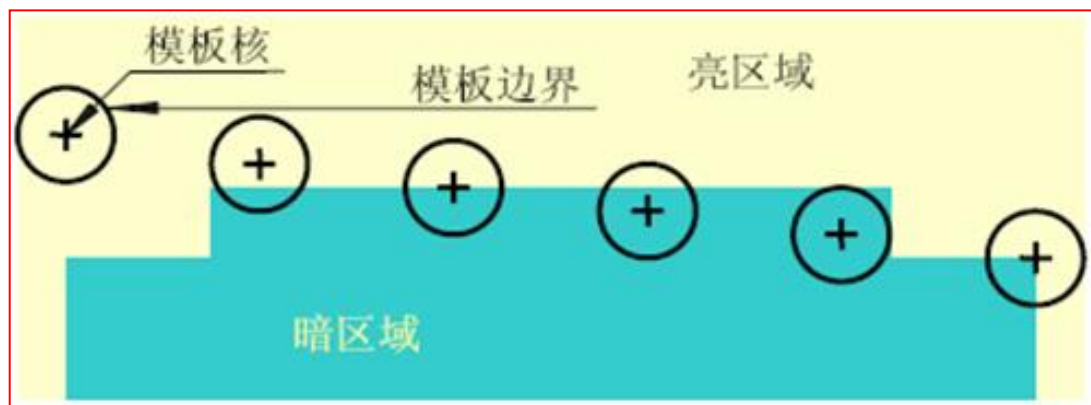
- 正交梯度算子
  - 梯度算子
- 方向微分算子
  - Kirsch算子
- 二阶导数算子
  - 拉普拉斯(Laplacian)算子
  - 马尔(Marr)算子
- 最优边缘检测算子
  - 坎尼(Canny)算子
- SUSAN 算子



# SUSAN算子

## □ USAN (Univalue Segment Assimilating Nucleus)

- 核同值区：相对于模板的核，模板中有一定的区域与它有相同的灰度



## □ USAN面积携带了关于图象中核像素处结构的主要信息

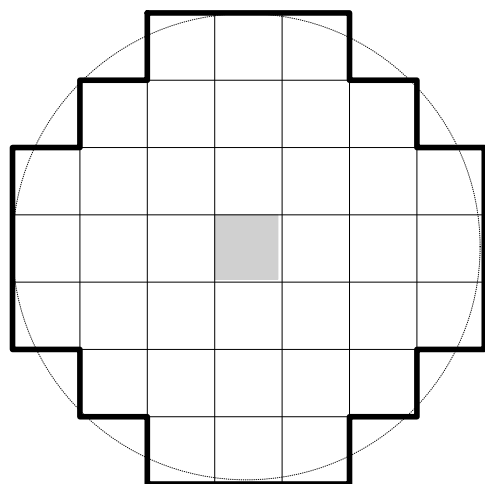
- 当核像素处在图象中的灰度一致区域，USAN的面积会达到最大
- 当核处在直边缘处该面积约为最大值的一半，而当核处在角点处则为最大值的1/4

## □ 使用USAN面积作为特征起到了增强边缘和角点的效果

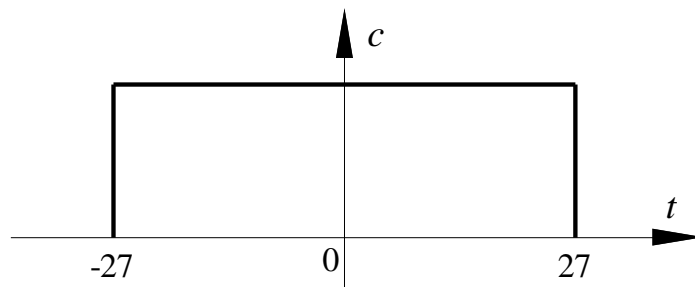
# SUSAN算子

□ SUSAN: 最小 (Smallest) 核同值区(USAN)

检测模板: 37个像素, 半径为3.4像素



$$C(x_0, y_0; x, y) = \begin{cases} 1 & \text{如果 } |f(x_0, y_0) - f(x, y)| \leq T \\ 0 & \text{如果 } |f(x_0, y_0) - f(x, y)| > T \end{cases}$$





# SUSAN算子

- 检测对模板中的每个像素进行
- 得到输出的游程和（running total）

$$S(x_0, y_0) = \sum_{(x,y) \in N(x,y)} C(x_0, y_0; x, y)$$

- 边缘响应

$$R(x_0, y_0) = \begin{cases} G - S(x_0, y_0) & \text{如果 } S(x_0, y_0) < G \\ 0 & \text{否则} \end{cases}$$

几何阈值  $G = 3S_{\max}/4$  （为了达到最佳信噪比），其中  $S_{\max}$  是  $S$  所能取的最大值，即模板面积。

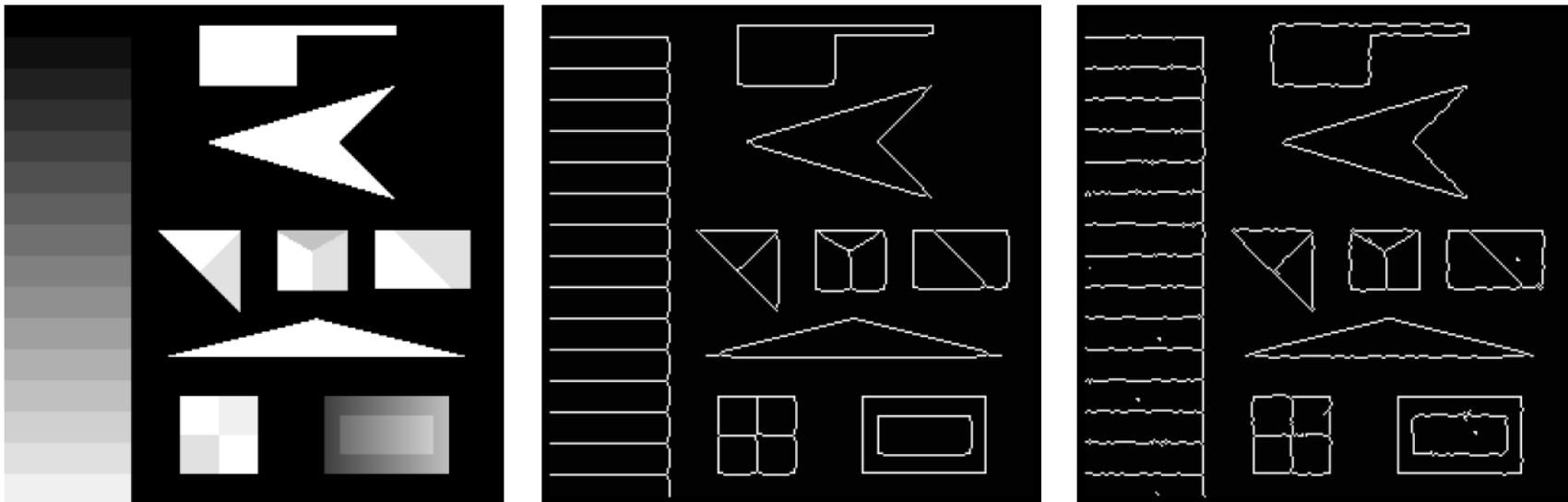


# SUSAN边缘检测

## □ 特点

- 有噪声时的性能较好
  - ✓ 不需要计算微分
  - ✓ 对面积计算中的各个值求和（积分）
  - ✓ 非线性响应特点
  
- 易自动化实现
  - ✓ 控制参数的选择简单
  - ✓ 参数的任意性较小

# SUSAN算子检测实例



左：原图。

中：SUSAN检测结果。

右：含高斯白噪声的结果。(SNR=0.5)



# 边界闭合

- 有噪声时：边缘像素常孤立/分小段连续
- 封闭边界（轮廓）：连接边缘像素
- 一种具体方法
  - 利用像素梯度的幅度和方向：

$$|\nabla f(x, y) - \nabla f(s, t)| \leq T$$

$$|\varphi(x, y) - \varphi(s, t)| \leq A$$

- 像素(s, t)在像素(x, y)的邻域



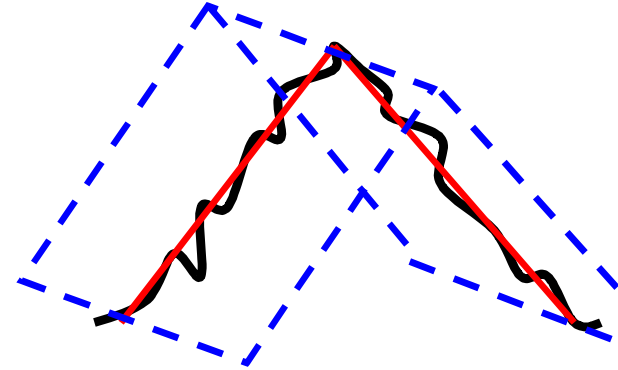
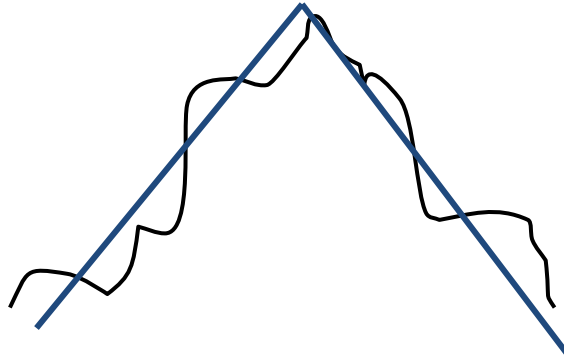
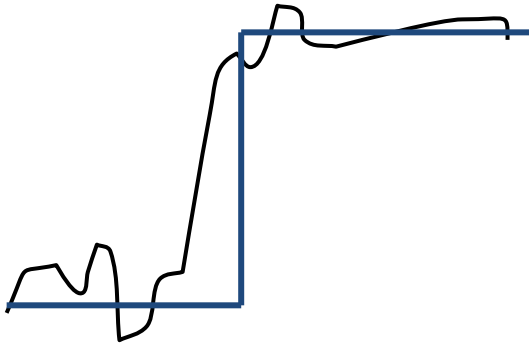
# 边缘检测

---

- ☐ 边缘模型
- ☐ 边缘检测算子
- ☐ 边缘拟合
- ☐ 边缘搜索

# 边缘拟合(Edge Fitting)

- 对图象中一个子区域，用理想灰度**阶跃**或**斜变**去拟合实际图象数据，从而求出拟合的理想模型参数，如阶跃幅度、斜变倾角等，并以此为这个子区域的边缘强度和方向度量
- 从某种意义上说，拟合是匹配滤波，旨在从失真和噪声中检测出理想边缘来。因此，有较强的**抗噪声**能力







# 基于斜面模型的边缘检测

## □ 拟合模型

- 将 $M \times N$ 的数字图像划分为相连接的区域集合 $P$  ,  $P = (P_1, P_2, \dots, P_i, \dots)$
- $P_i$ 的大小设定 $R \times C$  (通常为 $3 \times 3$ )

$$\hat{f}(x, y) = \alpha x + \beta y + \gamma$$

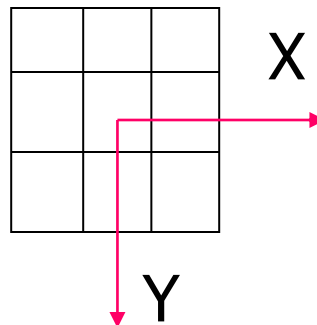
## □ 对每一个小区域，用一个斜平面来近似

## □ 拟合误差 $$e^2 = \sum_R \sum_{\times C} [\alpha x + \beta y + \gamma - f(x, y)]^2$$

# 基于斜面模型的边缘检测

## □ 斜面拟合

- 由最小化拟合均方误差来求  $\alpha$ 、 $\beta$ 、 $\gamma$
- 区域  $3 \times 3$ ，原点取在中心点时，有



$$\alpha = \frac{\sum \sum_{R \times C} x f(x, y)}{\sum \sum x^2}$$

$$\beta = \frac{\sum \sum_{R \times C} y f(x, y)}{\sum \sum y^2}$$

$$\gamma = \frac{\sum \sum_{R \times C} f(x, y)}{\sum \sum 1}$$

$$\rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

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

$\Sigma \Sigma$

代入误差算式，得出误差值。

误差较小时，认为拟合可靠。

# 基于斜面模型的边缘检测

## □ 斜面交界的判断

- 对于“可靠”的斜面，把该拟合斜平面的参数作为小区域的参数（不重叠划分）或小区域中心点的参数（重叠划分）。

- 可得：梯度值  $\sqrt{\alpha^2 + \beta^2}$

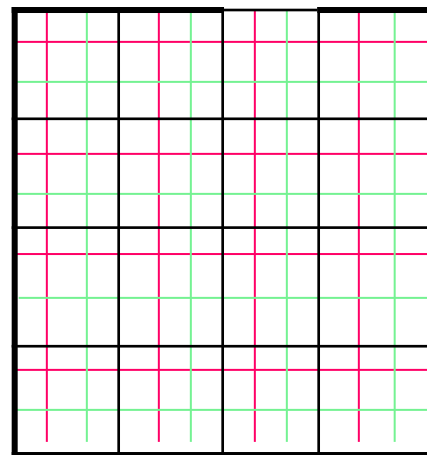
$$\text{方向为 } \theta = \tan^{-1} \frac{\alpha}{\beta}$$

- 然后用斜面参数判断各个斜面间是否有边缘存在。

即考察  $\alpha_1 = \alpha_2?$

$$\beta_1 = \beta_2?$$

$$\gamma_1 = \gamma_2?$$





# 一般步骤:

---

1. 选取适合的拟合区域
2. 根据模型求解拟合系数
3. 求拟合斜面各点灰度值
4. 计算误差，判断斜面的可靠性
5. 对于可靠的斜面，计算相邻点或区域不在同一斜面上的度量
6. 选出边缘度量值局部最大的点，定为边缘点



# 边缘检测

---

- ☐ 边缘模型
- ☐ 边缘检测算子
- ☐ 边缘拟合
- ☐ 边缘搜索



# 图搜索

## □ 图搜索的一般步骤：

- 确定搜索的起点，对于边缘搜索则起点是某一边缘点
- 采取一种合适的**数据结构**和**搜索机理**，在已有边缘点的基础上进行搜索，不断确定新的边缘点
- 规定搜索终止的条件，在满足条件时停止搜索

## □ 基本概念

- **边界点**和**边界段**可以用图结构表示，通过在图中搜索达到某一目标的最佳路径（最短路径，最小消耗路径）寻找边缘。

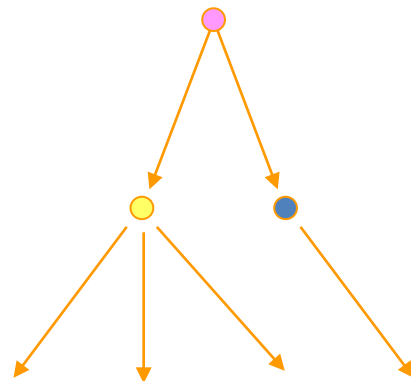
# 图搜索

- 将边缘像素和边界段用图表示

图:  $G = \{N, A\}$

结点集  $\{n_1, \dots\}$

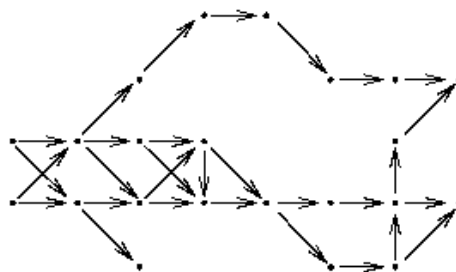
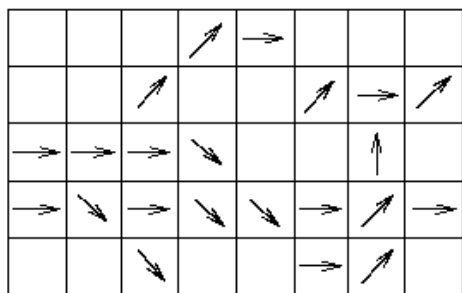
结点对集  $\{(n_i, n_j)\}$



- 通路代价: 通常跟灰度值的变化是相关的

$$C = \sum_{i=2}^K c(n_{i-1}, n_i)$$

- 图的建立



(左) 显著边缘点的方向 (右) 相应的图



# 启发式图搜索

---

- 路径评价函数可以定义为： $f(n)=g(n)+h(n)$ 
  - $n$ ：搜索过程进行到的当前节点。
  - $g(n)$ ：为从起始节点到当前节点所有路径代价。
  - $h(n)$ ：是当前节点到目标节点将要经过的所有路径的代价。
    - ✓ 一般为对真实代价 $h(n)$ 的估计值，从而为启发项



# 启发式图搜索

## □ 利用问题拥有的启发信息来引导搜索

- 减少搜索范围
- 降低问题复杂度

## □ 代价函数 $f(n)=g(n)+h(n)$

- **启发项**  $h(n)$  能加快搜索速度，但不一定能找到全局最优。
- 当无启发项（ $h(n)=0$ ）时，一定能找到全局最优，但搜索范围加大。



普通搜索过程演示



启发式搜索过程演示



# 代价函数的设计

- 使用“启发”信息，构造评价函数，计算路径的耗费，是启发式搜索的关键。
- 具体如何构造，则的分析该问题的具体情况，将多种因素合理的分离开来，把实际情况中的约束转化为计算机可操作的表达式。
  - 灰度梯度的幅度
  - 灰度梯度的方向
  - 路径的曲率
  - 路径与某一函数的近似程度
  - 到目标点的距离等