

# Canny边缘检测器



J.Canny, “A Computational Approach to Edge Detection”,  
*IEEE Trans. on PAMI*, 8(6),1986.

John Canny

CS Division, University of California, Berkeley

BID: The Berkeley Institute of Design

*National Laboratory of Pattern Recognition*

Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所

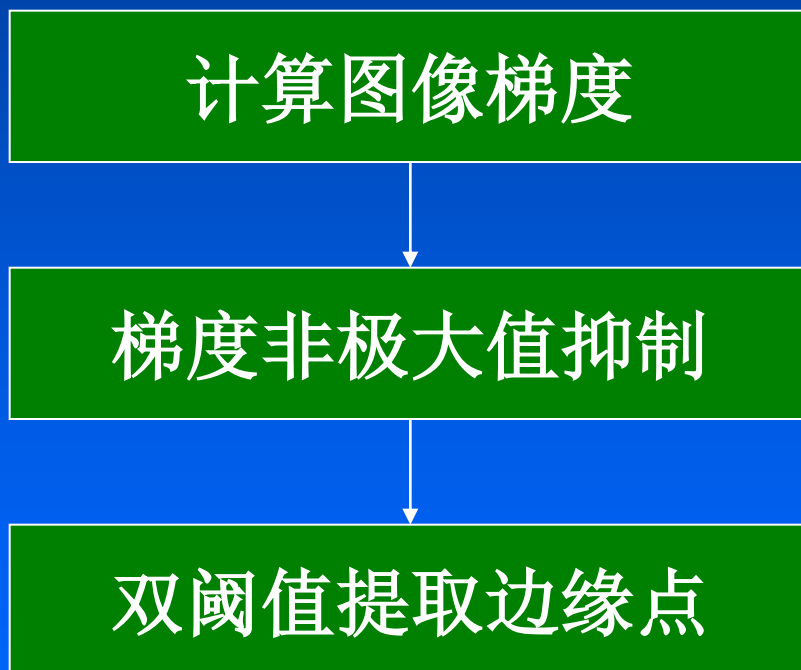
# Canny边缘检测器

- J. Canny, “A Computational Approach to Edge Detection”, *IEEE Trans. on PAMI*, 8(6), 1986.
- 也许是最常用的边缘检测方法
- 一个优化的方案
  - 噪声抑制
  - 边缘增强
  - 边缘定位



# Canny边缘检测算法

- 算法基本过程:



幅值大小 $M(x,y)$   
方向 $\Theta(x,y)$

NMS:  
Non-Maxima Suppression



# 计算图像梯度：高斯函数的一阶导数

- 高斯函数的一阶导数 (Derivative of Gaussian)
- 可以很近似地满足以下三条边缘检测最优准则：
  - 好的边缘检测性能：Good detection  
对边缘的响应大于对噪声的响应
  - 好的定位性能：Good localization  
其最大值应接近边缘的实际位置
  - 低的错误检测率：Low false positives  
在边缘附近只有一个极大值点



# 计算图像梯度：高斯函数的一阶导数

(1) 求图像与高斯平滑滤波器卷积：

$$S(x, y) = G(x, y; \sigma) * I(x, y)$$

$\sigma$  代表对图像的平滑程度

(2) 使用一阶有限差分计算偏导数的两个阵列：

$$D_x(x, y) \approx (S(x, y+1) - S(x, y) + S(x+1, y+1) - S(x+1, y)) / 2$$

$$D_y(x, y) \approx (S(x, y) - S(x+1, y) + S(x, y+1) - S(x+1, y+1)) / 2$$

相当于与模版进行卷积运算：

-1	1
-1	1

1	1
-1	-1



# 计算图像梯度：高斯函数的一阶导数

(3) 幅值和方位角:

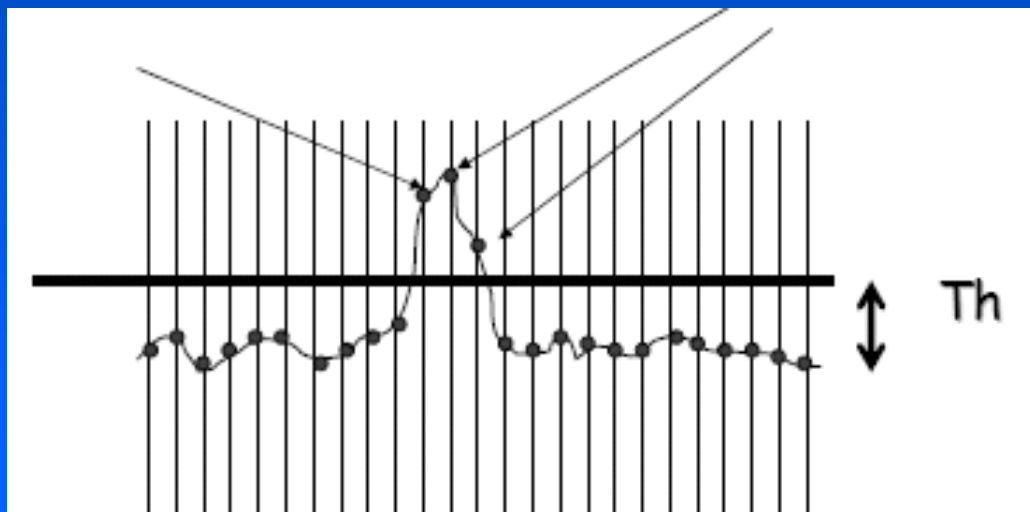
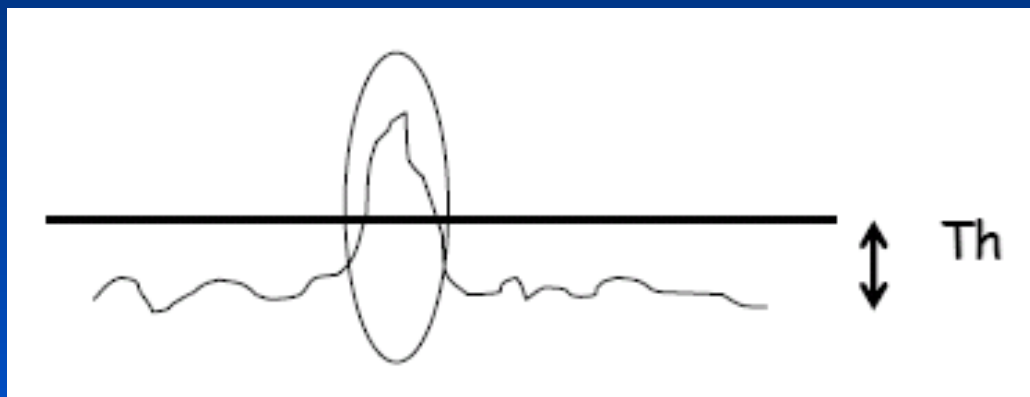
$$M(x, y) = \sqrt{D_x(x, y)^2 + D_y(x, y)^2}$$

$$\theta(x, y) = \arctan(D_y(x, y) / D_x(x, y))$$

$M$  代表梯度幅值的大小，在存在边缘的图像位置处， $M$  的值变大，图像的边缘特征被“增强”



# 如何检测边缘？



局部极值周围存在相近数值的点

# 非极大值抑制 NMS

- 非极大值抑制（ NMS: Non-Maxima Suppression ）
- 主要思想：由梯度幅值图像 $M(x,y)$ ，仅保留极大值。  
（严格地说，保留梯度方向上的极大值点。）

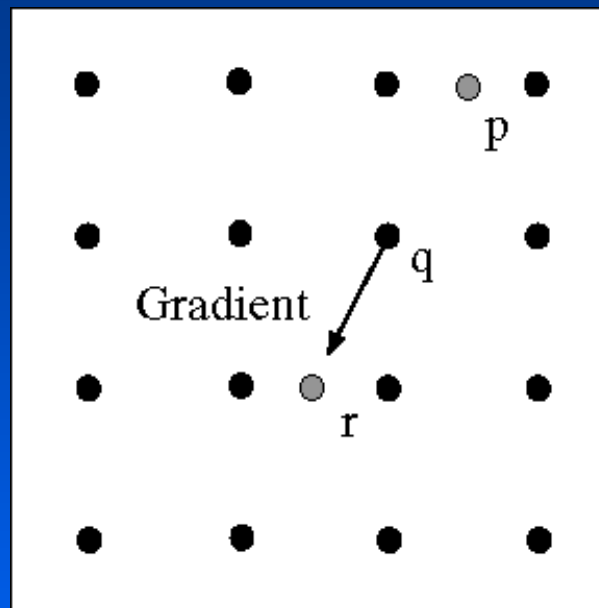
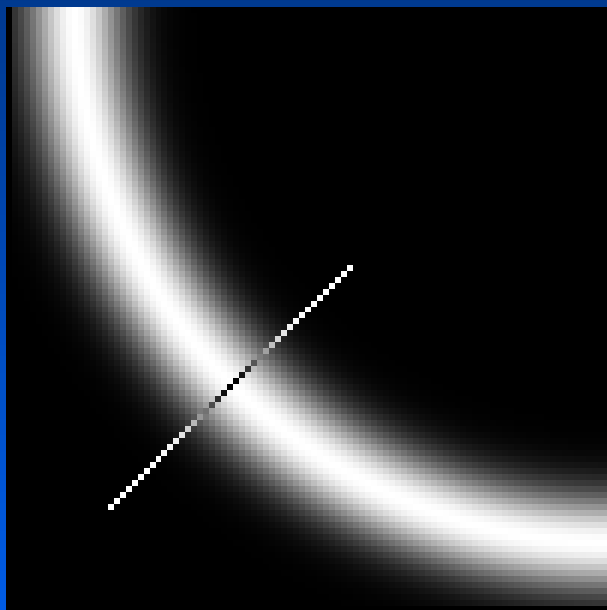
得到的结果为 $N(x,y)$ ，具体过程：

- 初始化 $N(x,y) = M(x,y)$
- 对于每个点，在梯度方向和反梯度方向各找 $n$ 个像素点。  
若 $M(x,y)$ 不是这些点中的最大点，则将 $N(x,y)$ 置零，否则保持 $N(x,y)$ 不变。
- $N(x,y)$  单像素宽度：
  - 问题：额外的边缘点，丢失的边缘点





# 非极大值抑制 NMS



- 在梯度方向的沿线上检测该点是否为局部极大值
- 简化的情形，只使用4个方向：{0, 45, 90, 135}
- 得到的结果 $N(x, y)$  包含边缘的宽度为1个像素

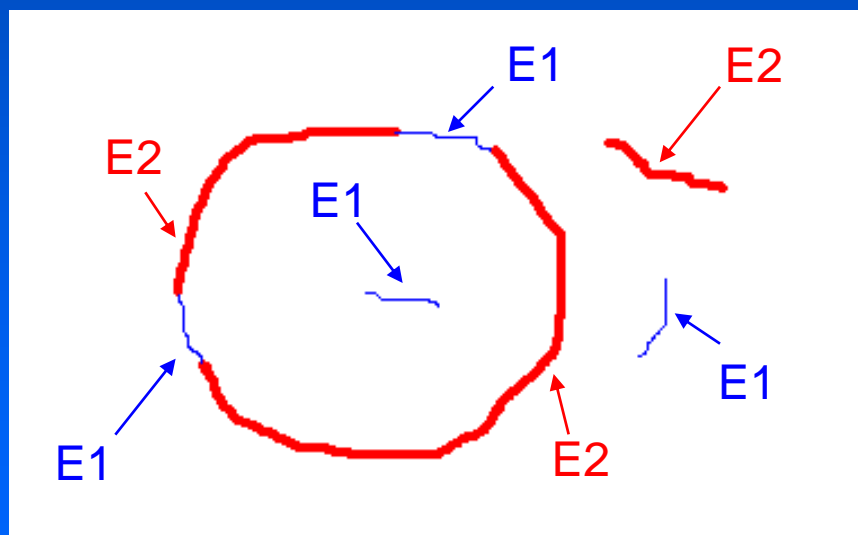
# 对NMS结果进行二值化

- 对上述得到的 $N(x,y)$ 使用阈值进行二值化
- 使用大的阈值，得到：
  - 少量的边缘点
  - 许多空隙
- 使用小的阈值，得到：
  - 大量的边缘点
  - 大量的错误检测



# 使用双阈值检测边缘

- 两个阈值 $T_1$ ,  $T_2$ :  $T_2 \gg T_1$ 
  - 由 $T_1$ 得到  $E_1(x, y)$ , 低阈值边缘图: 更大的误检测率
  - 由 $T_2$ 得到  $E_2(x, y)$ , 高阈值边缘图: 更加可靠
- 边缘连接:



# 边缘连接

- 将 $E_2(x, y)$ 中相连的边缘点输出为一幅边缘图像 $E(x, y)$
- 对于 $E(x, y)$ 中每条边，从端点出发在 $E_1(x, y)$ 中寻找其延长的部分，直至与 $E(x, y)$ 中另外一条边的端点相连，否则认为 $E_1(x, y)$ 中没有它延长的部分
- 将 $E(x, y)$ 作为结果输出



# Canny算子：流程



原始图像



原始图像经过Gauss平滑

*National Laboratory of Pattern Recognition*

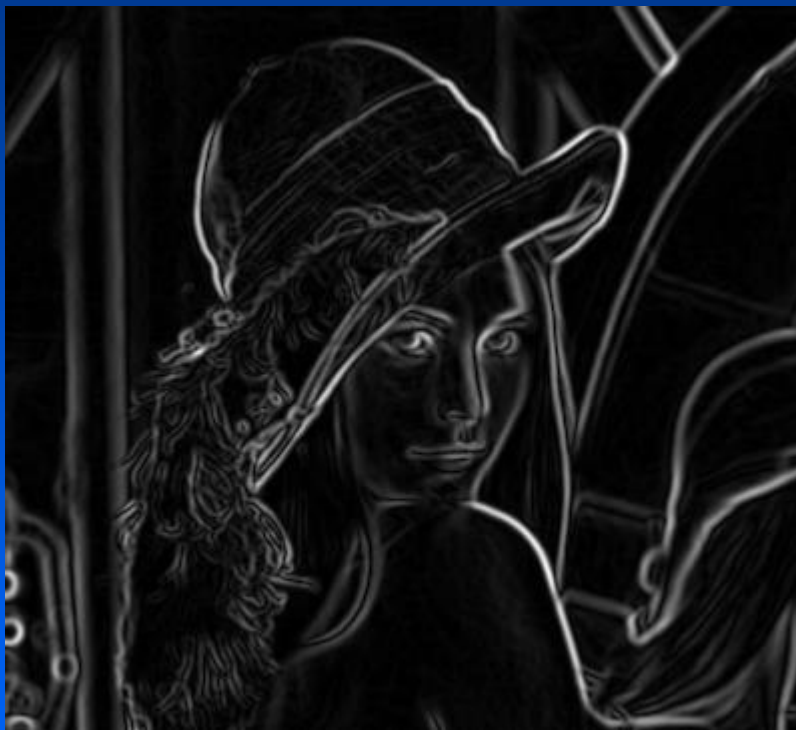
Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所

# Canny算子：流程

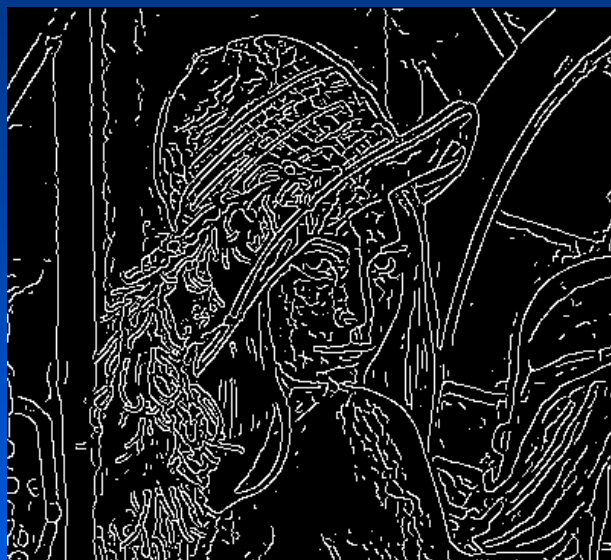


梯度幅值图像



梯度幅值经过非极大值抑制

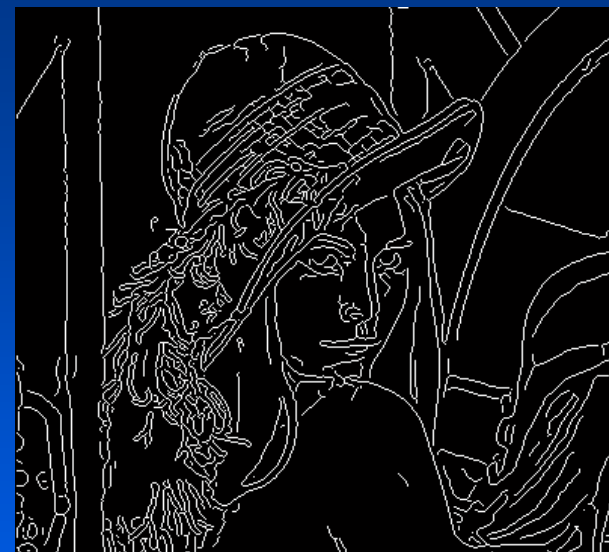
# Canny算子：流程



低阈值边缘图像



高阈值边缘图像



Canny输出边缘图像

# 使用Canny算子需要注意的问题

- Canny算子的优点：
  - 参数较少
  - 计算效率
  - 得到的边缘连续完整
- 参数的选择：
  - Gauss滤波的尺度
  - 双阈值的选择( $LOW=HIGH*0.4$ )





canny: 0.04 0.1 Gaussian: 1

canny: 0.04 0.1 Gaussian: 3

canny: 0.04 0.1 Gaussian: 5

canny: 0.04 0.1 Gaussian: 7

canny: 0.04 0.1 Gaussian: 9

canny: 0.04 0.1 Gaussian: 11

渐增高斯滤波模版的尺寸

*National Laboratory of Pattern Recognition*

Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所

canny: 0.04 0.1



canny: 0.08 0.2



canny: 0.12 0.3



canny: 0.16 0.4



canny: 0.2 0.5



canny: 0.24 0.6



渐增双阈值的大小，保持 $\text{low} = \text{high} \times 0.4$

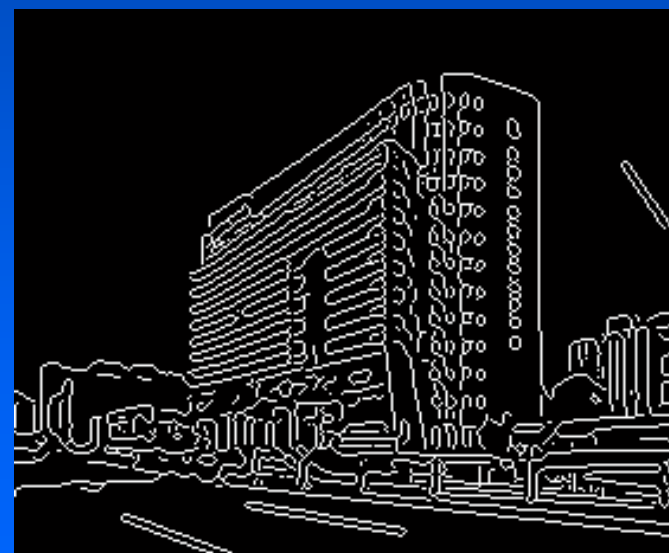
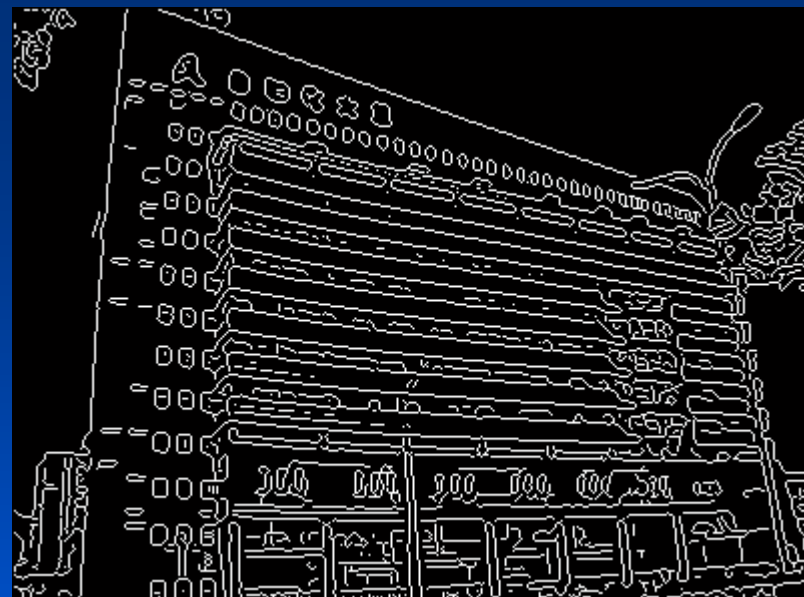
*National Laboratory of Pattern Recognition*

Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所



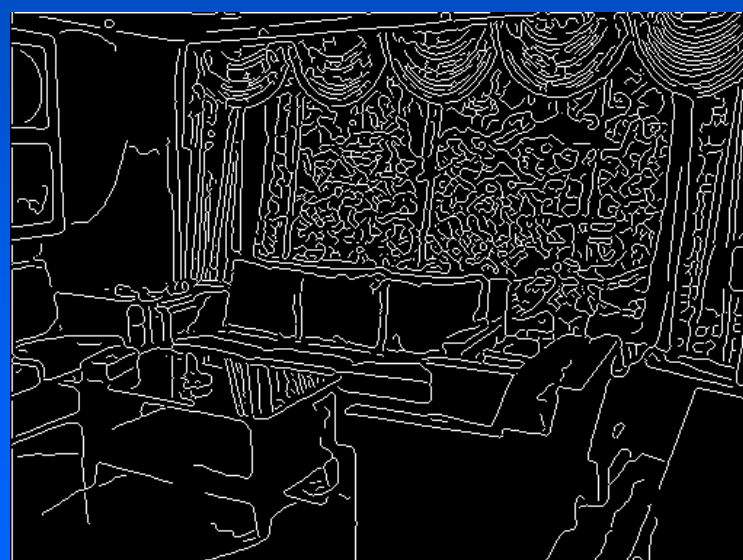
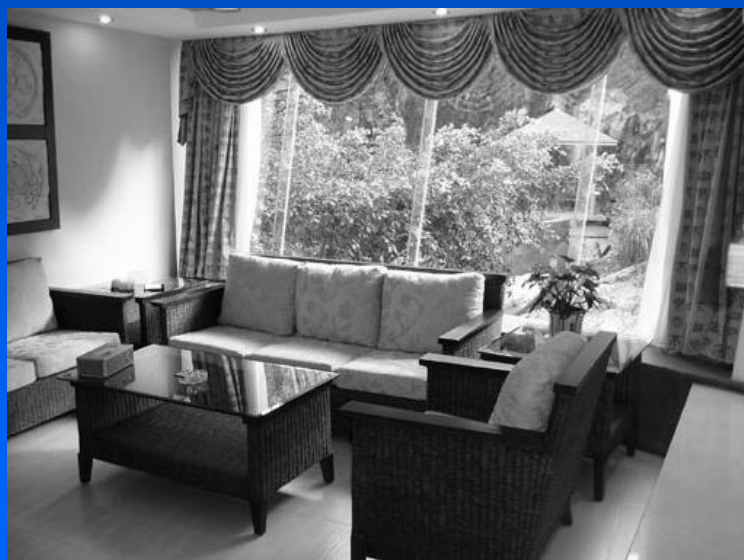
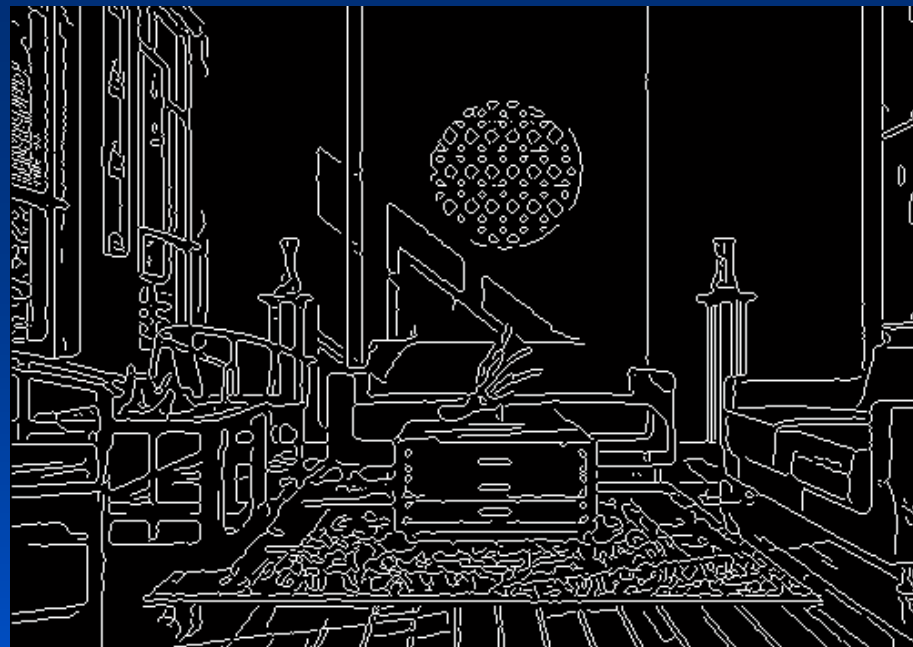
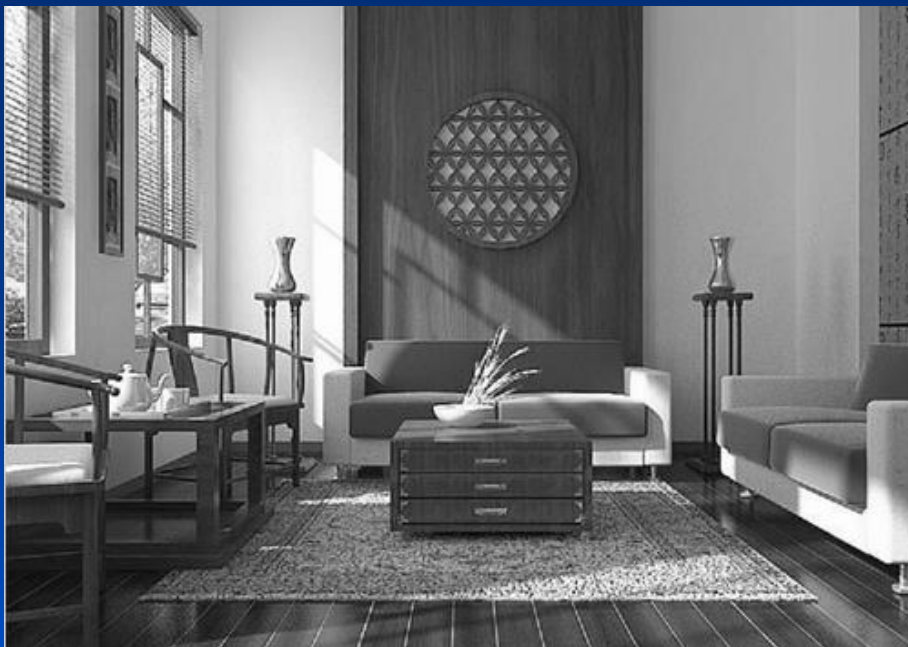
*National Laboratory of Pattern Recognition*

Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所



*National Laboratory of Pattern Recognition*

Institute of Automation, Chinese Academy of Sciences



**模式识别国家重点实验室**

中国科学院自动化研究所

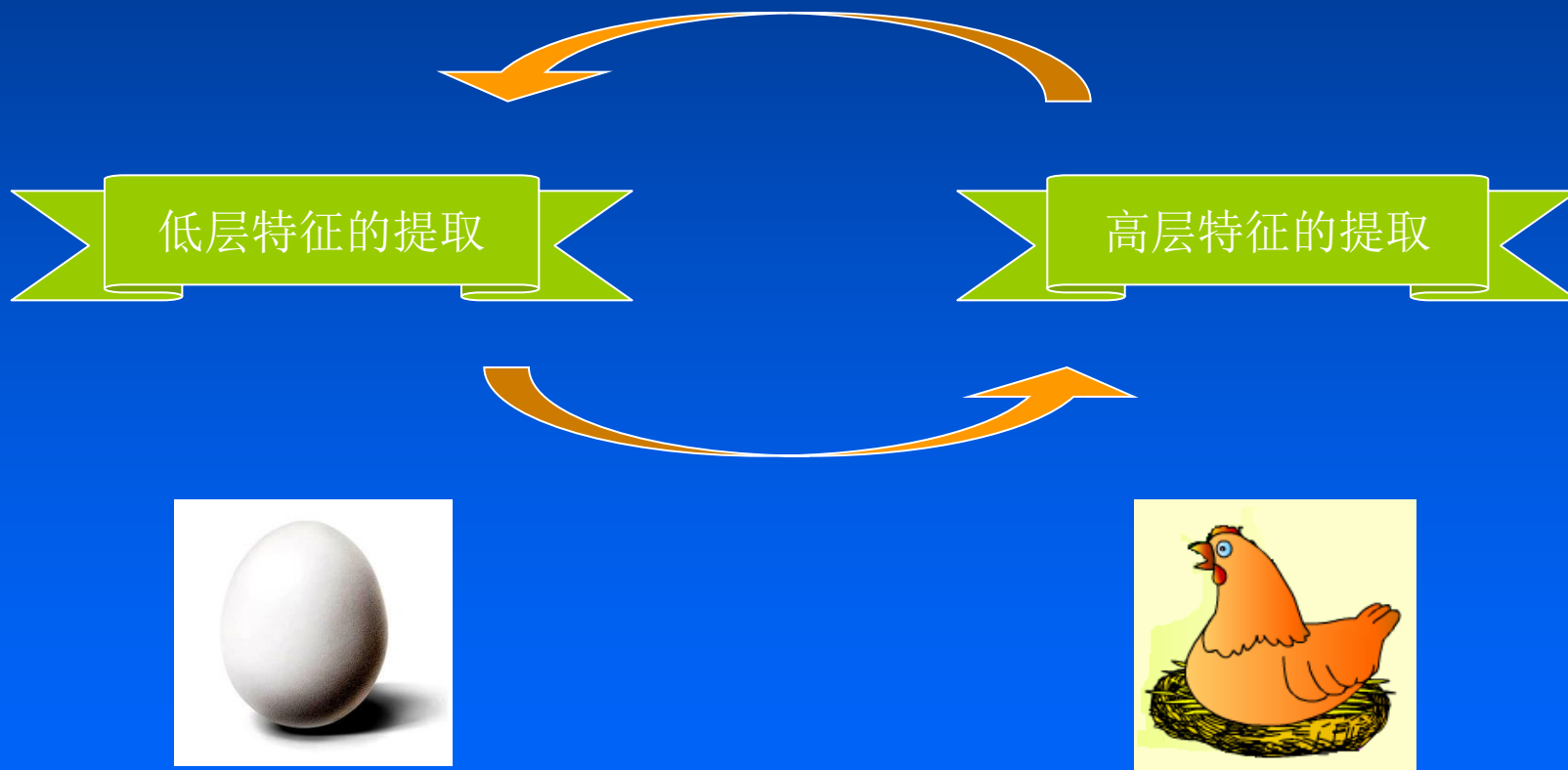


# 边缘检测小结

- 边缘检测是计算机视觉中最基本的问题之一
- 没有一种统一的方法可以解决所有的边缘分割问题：
  - 抑制噪声的能力
  - 定位精度
  - 计算的复杂程度
- 困难的原因
  - 让计算机理解图像：从数值矩阵到语义概念
  - 实际问题的复杂性：噪声，光照，阴影...



# 图像特征分割的困难？



# 推荐文献阅读

- J.Canny, “A Computational Approach to Edge Detection”, *IEEE Trans. on PAMI*, 8(6),1986.
- F. A. Pellegrino, W. Vanzella, and V. Torre, Edge Detection Revisited, *IEEE TRANS. on SYSTEMS, MAN, AND CYBERNETICS*, VOL. 34, NO. 3, JUNE 2004

