

计算机视觉—特征检测

申抒含

中国科学院自动化研究所
模式识别国家重点实验室



Robot Vision Group

National Laboratory of Pattern Recognition

Institute of Automation, Chinese Academy of Sciences

有关计算机视觉——整体框架



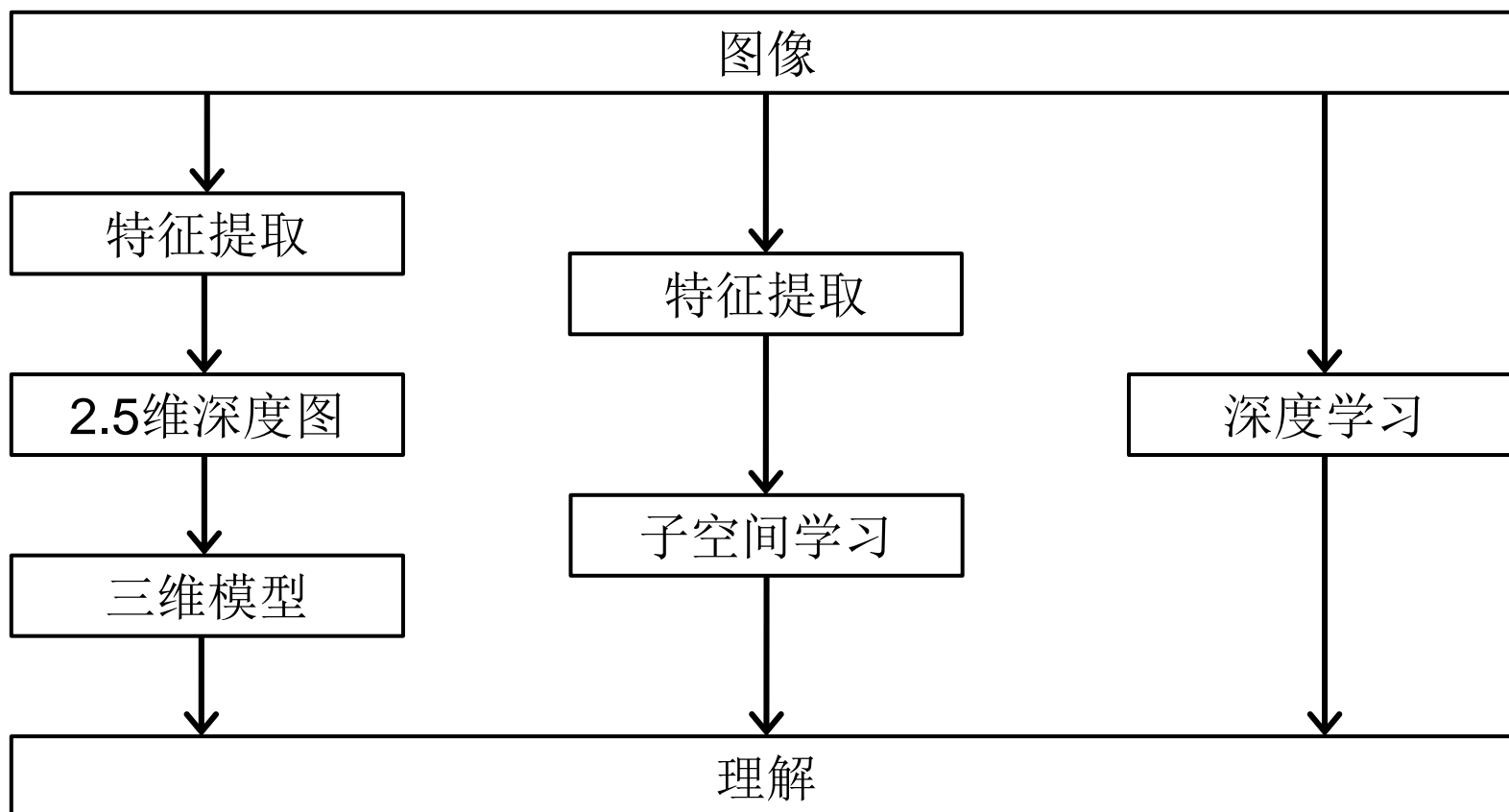
Marr, 1981



Poggio, 2000



Hinton, 2006



有关计算机视觉——模型比较

这三种视觉计算模型哪个最好？

Marr模型：

- 人类视觉的主要功能是复原三维场景的可见几何表面；
- 开创了计算机视觉学科；
- 在此模型上形成了以摄影几何为基础的完整的几何视觉理论；
- 这一模型目前看来是不符合人类视觉的；
- 但Marr提出的层次化三维重建框架，至今是计算机视觉中的主流方法。

有关计算机视觉——模型比较

这三种视觉计算模型哪个最好？

Poggio模型：

- 人类对物体的表达，不是物体的三维几何形状，而是该物体在不同视点下的一组图像；
- 有心理学和生理学基础；
- 在此基础上发展了大量基于机器学习的方法（图像—特征—学习—识别）；
- 这一模型仍不完善，基于子空间学习的方法遭遇瓶颈。

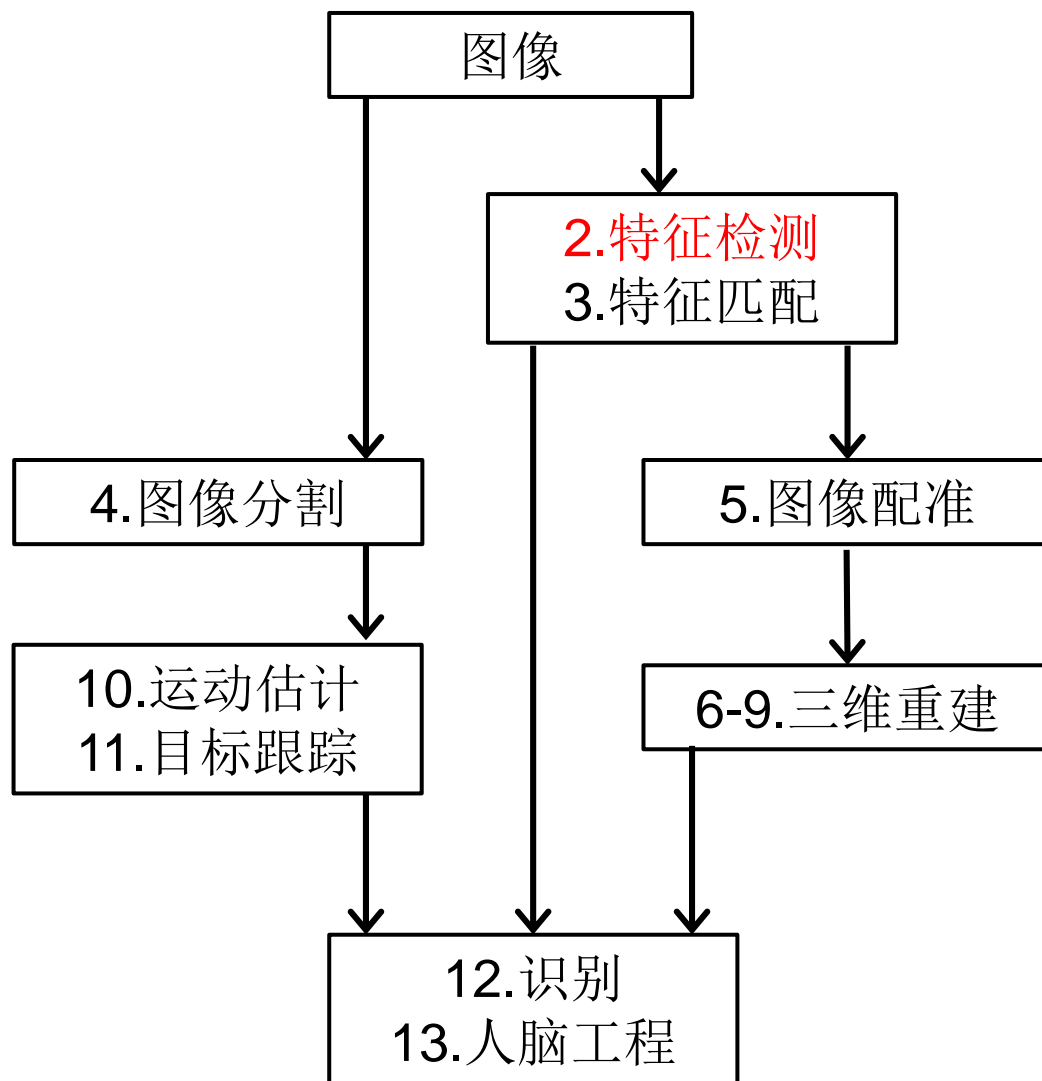
有关计算机视觉——模型比较

这三种视觉计算模型哪个最好？

Hinton模型：

- 人类感知（包括视觉）的处理过程，是一个深层网络加工过程；
- 结构决定功能。人脑中一个神经元功能有限，但分层+联结的脑网络则具有高级智慧；
- 多层结构+分层学习的DNN在有效提升了图像、语音、文字的识别水平；
- DNN并非万能，也不是尽善尽美；DNN只是对大脑非常粗略的近似（况且我们对人脑认知机理了解还很肤浅）；DNN网络内部的本质机理还不明确。

计算机视觉课程结构图



为什么要检测特征

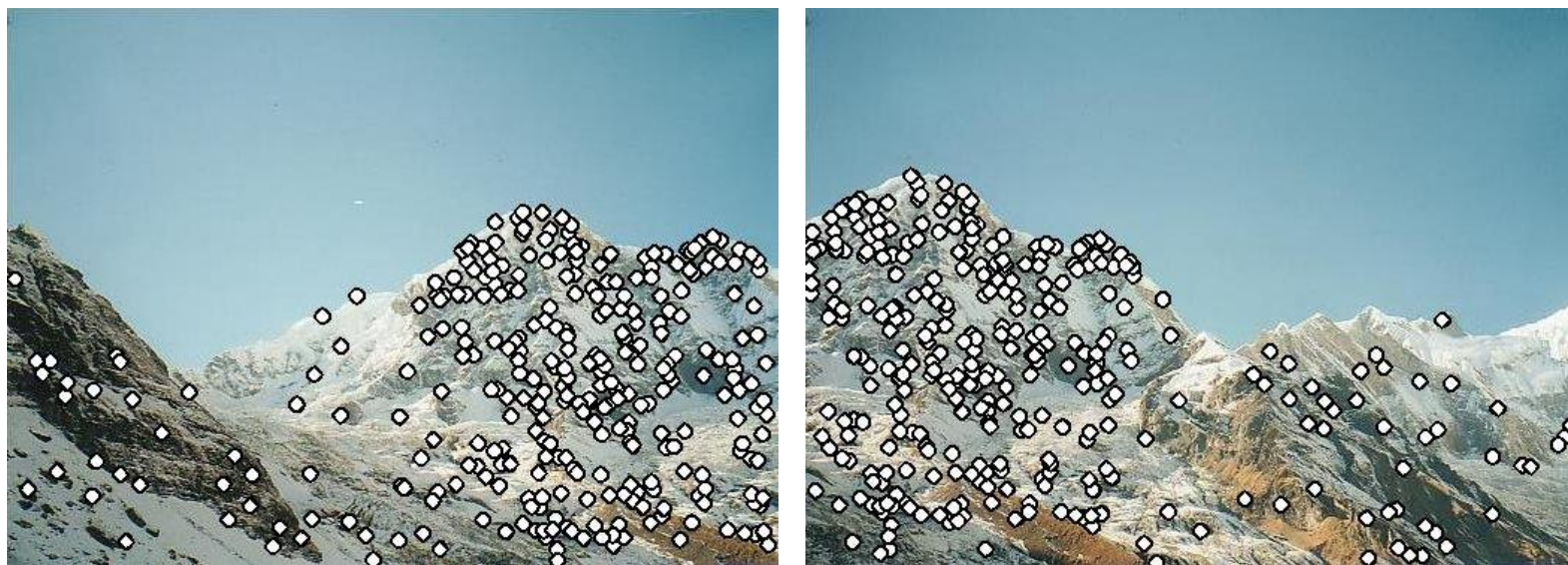
一个例子：全景图像拼接



给定两张图像，如何拼成大图？

为什么要检测特征

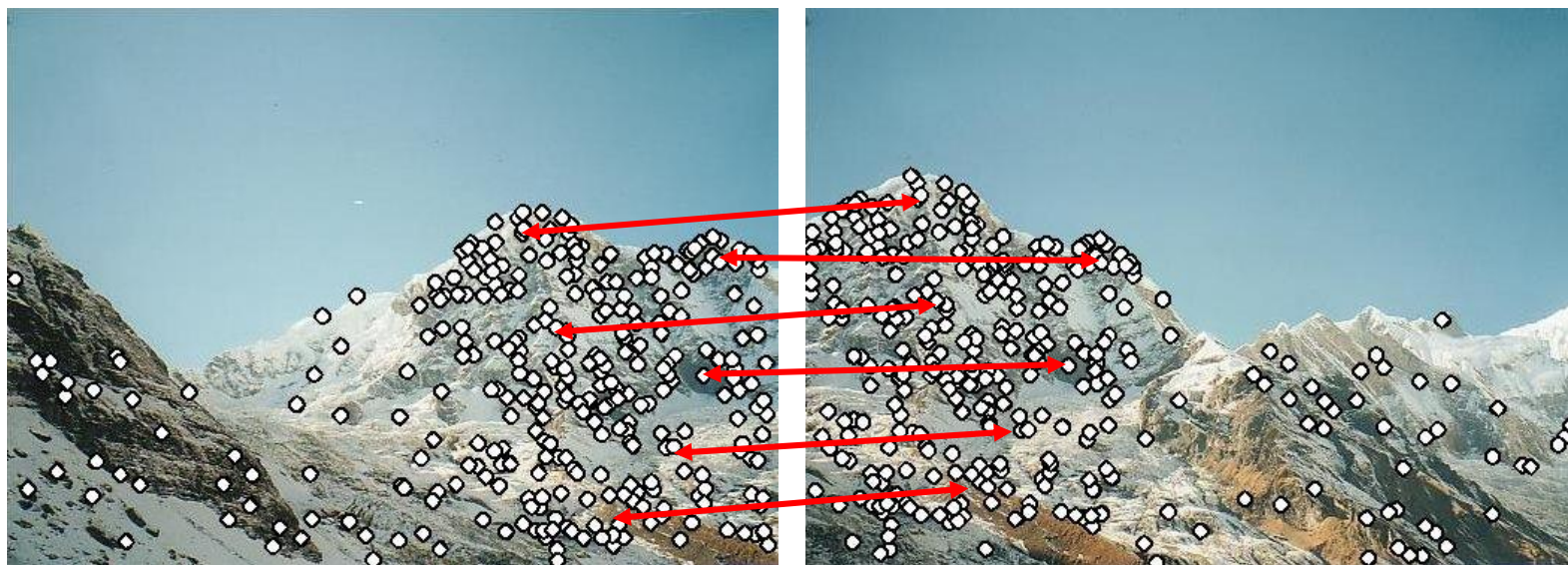
一个例子：全景图像拼接



步骤1：检测特征点

为什么要检测特征

一个例子：全景图像拼接



步骤1：检测特征点

步骤2：匹配特征点

为什么要检测特征

一个例子：全景图像拼接

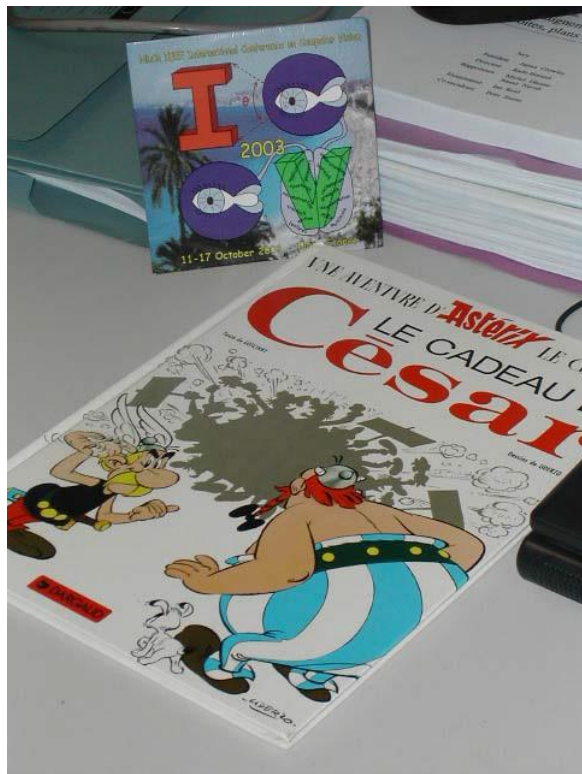


步骤1：检测特征点

步骤2：匹配特征点

步骤3：将图像配准

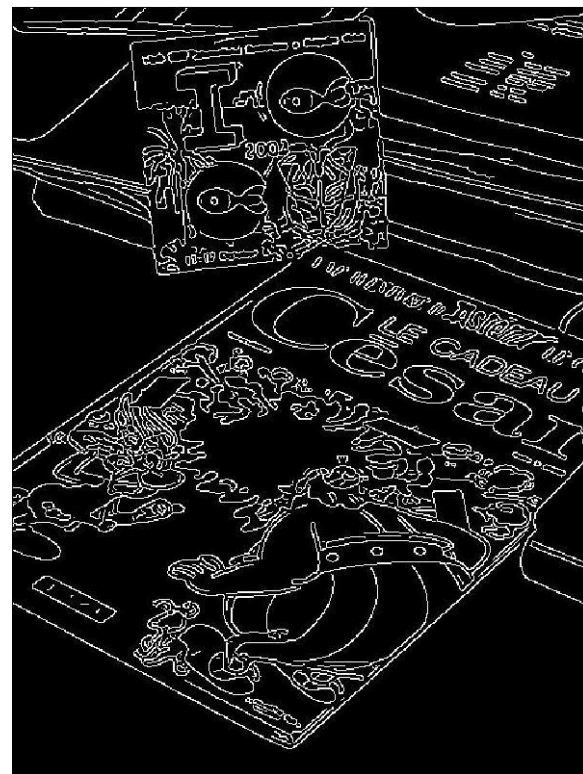
图像的特征有哪些



图像



Harris角点



Canny边缘

计算机视觉中常用的图像特征包括：点、边缘、直线、曲线等

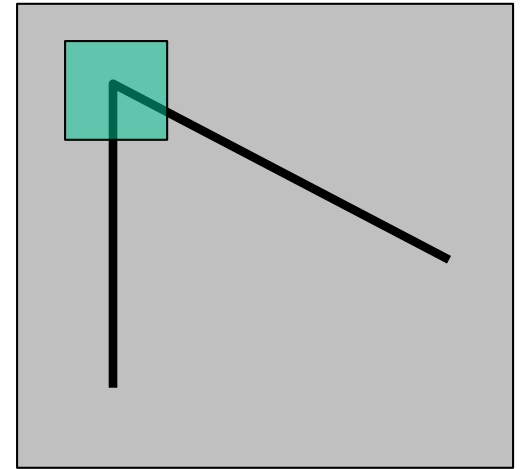
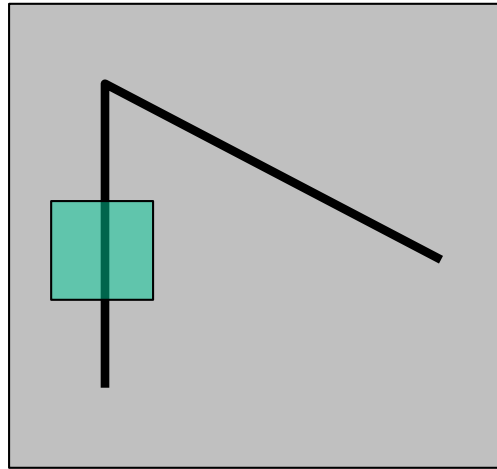
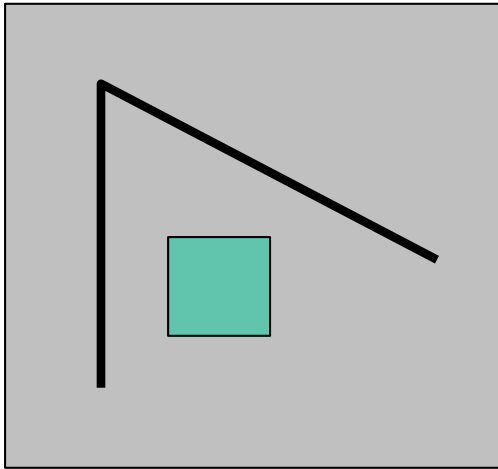
点特征检测

点特征的优势：

- 点特征属于局部特征，对遮挡有一定鲁棒性；
- 通常图像中可以检测到成百上千的点特征，以量取胜；
- 点特征有较好的辨识度，不同物体上的点容易区分；
- 点特征提取通常速度很快。

点特征检测

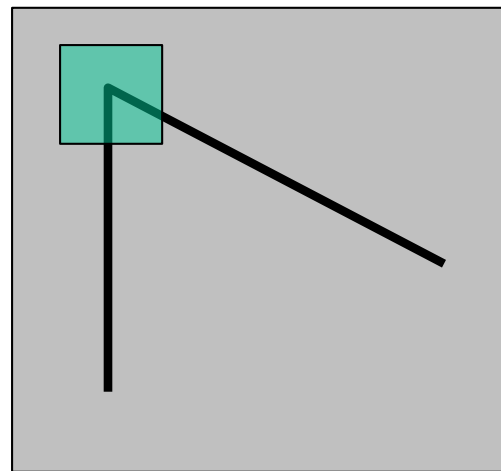
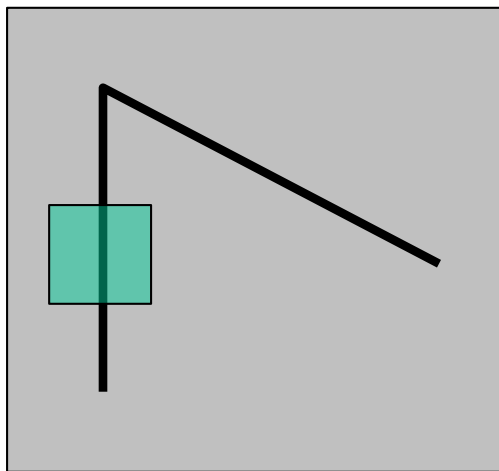
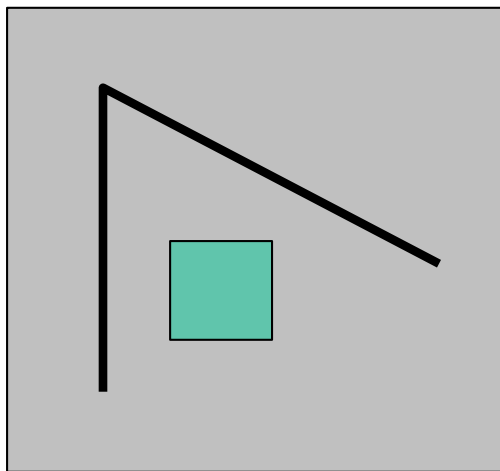
什么是好的点特征？



考虑图像上一个小窗口

点特征检测

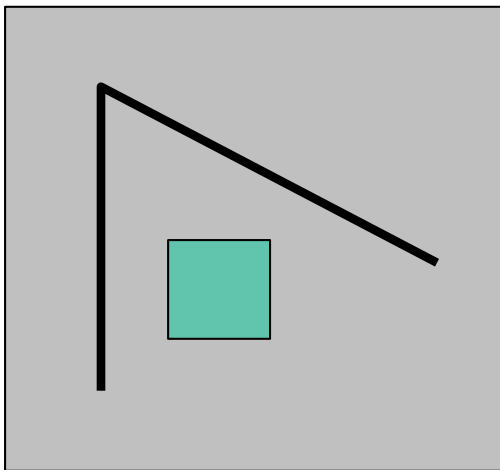
什么是好的点特征？



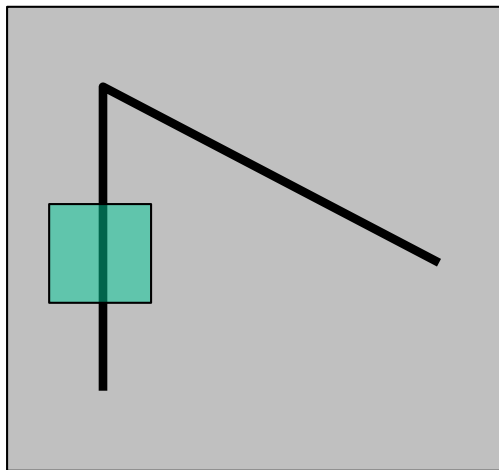
当窗口位置发生微小变化时，窗口图像如何变化？

点特征检测

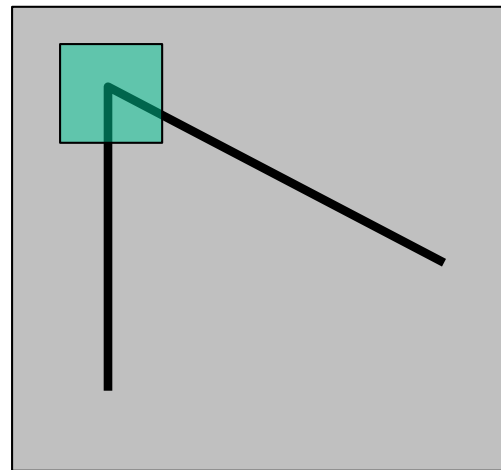
什么是好的点特征？



平坦区域：
任意方向移动，无
灰度变化



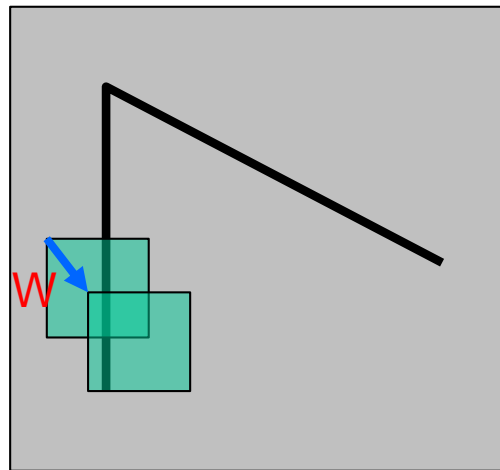
边缘：
沿着边缘方向移动
，无灰度变化



角点：
沿任意方向移动，
明显灰度变化

点特征检测—数学表达

- 假设窗口 W 发生位置偏移 (u,v) ;
- 比较偏移前后窗口中每一个像素点的灰度变化值;
- 使用误差平方和定义误差函数 $E(u,v)$

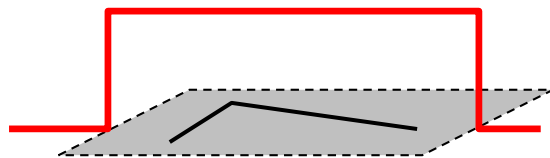


$$E(u, v) = \sum_{(x,y) \in W} \omega(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Diagram illustrating the components of the error function $E(u, v)$:

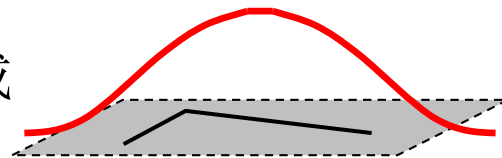
- $\omega(x, y)$: 窗口函数 (Window Function)
- $I(x + u, y + v)$: 平移后的图像灰度 (Translated Image Intensity)
- $I(x, y)$: 图像灰度 (Image Intensity)

窗口函数 $\omega(x, y)$:



1 in window, 0 outside

或

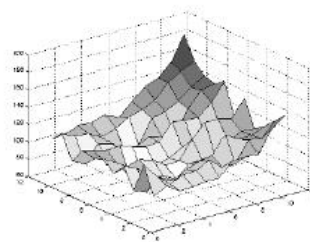
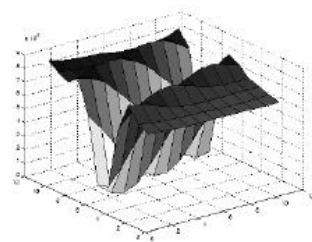
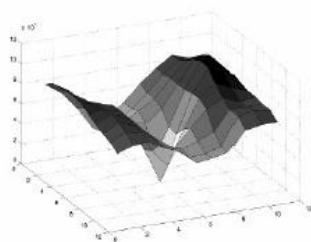
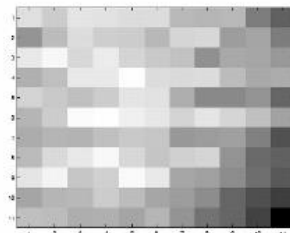
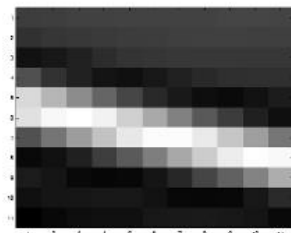
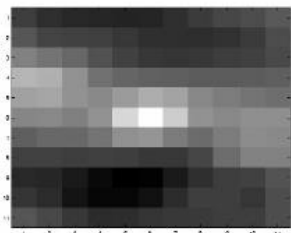


Gaussian

点特征检测—数学表达



不同位置点计算得到的 $E(u,v)$



点特征检测—数学表达

将 $I(u, v)$ 进行Taylor展开:

$$\begin{aligned} I(x + u, y + v) &= I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + O(u^2, v^2) \\ &\approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \quad \text{舍去高阶小量} \\ &\approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

$$\begin{aligned} E(u, v) &= \sum_{(x,y) \in W} \omega(x, y) [I(x + u, y + v) - I(x, y)]^2 \\ &= \sum_{(x,y) \in W} \omega(x, y) \left[[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2 \end{aligned}$$

点特征检测—数学表达

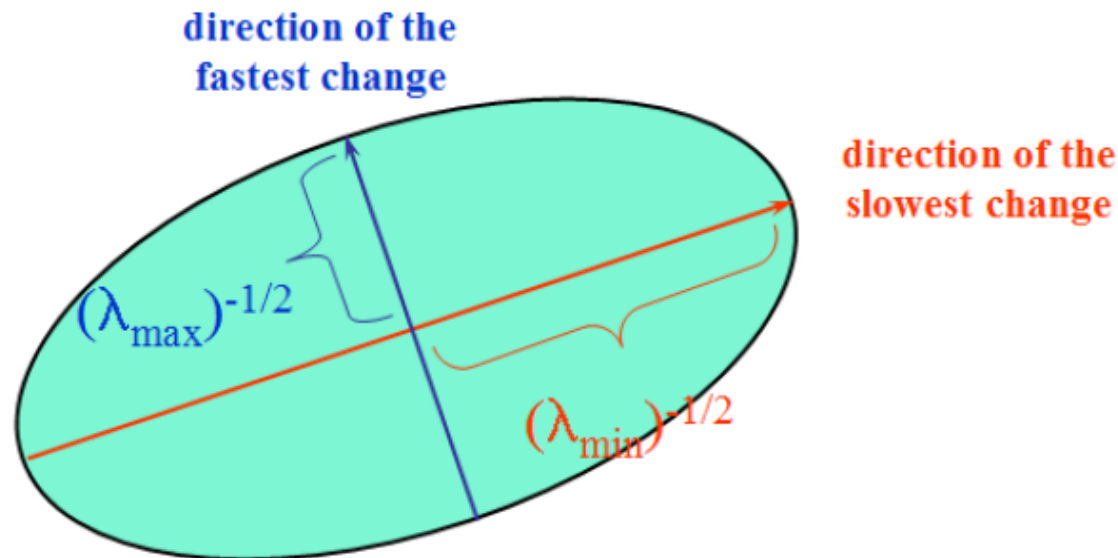
进一步展开，可以写成：

$$\begin{aligned} E(u, v) &= \sum_{(x,y) \in W} \omega(x, y) \left[[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2 \\ &= \sum_{(x,y) \in W} [u, v] \omega(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ &= [u, v] \underbrace{\left(\sum_{(x,y) \in W} \omega(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_H \end{aligned}$$

点特征检测—数学表达

$$H = \sum_{(x,y) \in W} \omega(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$
$$= \omega(x,y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

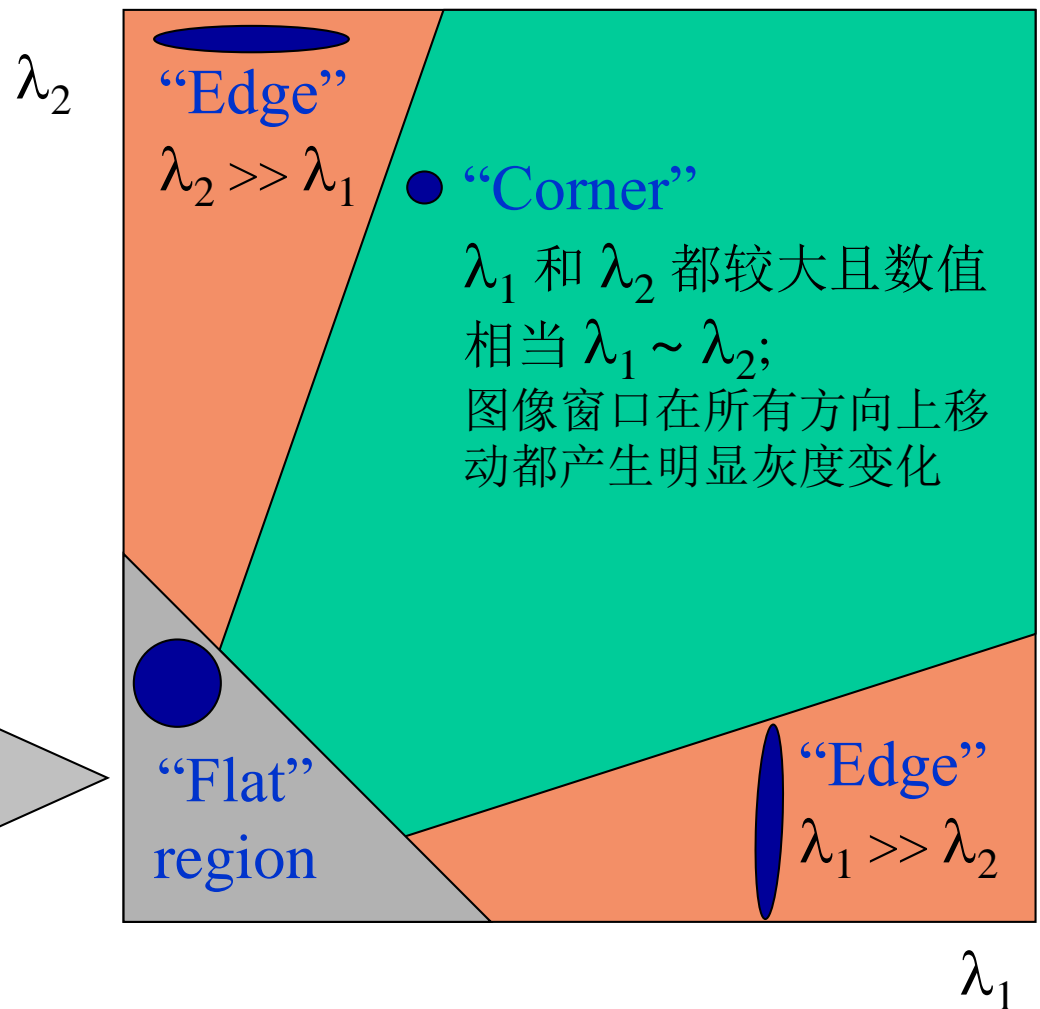
H 称为自相关矩阵, λ_{\max} 和 λ_{\min} 是 H 的2个特征值, $E(u,v)$ 的变化如下图所示



点特征检测—数学表达

根据 H 的2个特征
值大小对图像点进
行分类：

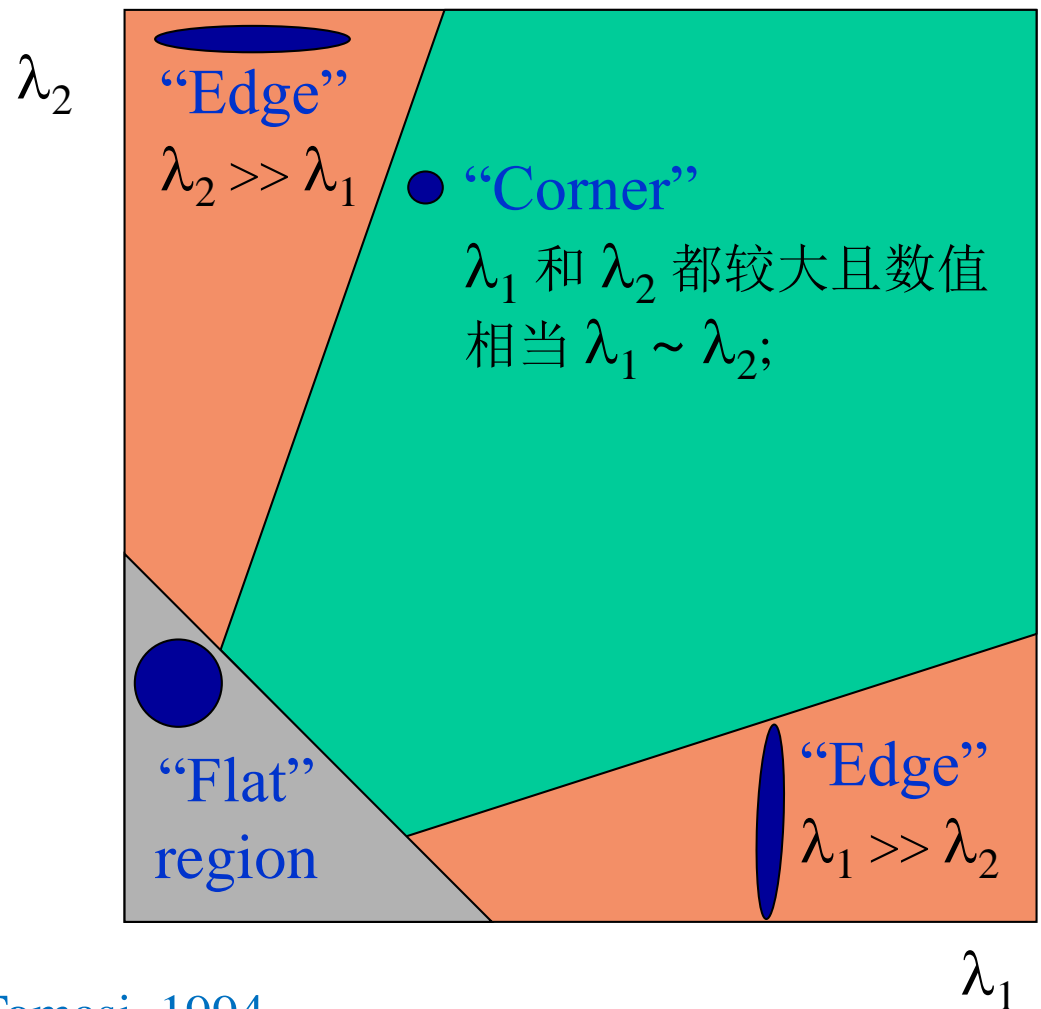
如果 λ_1 和 λ_2 都很小，
图像窗口在所有方向上
移动都无明显灰度变化



点特征检测—数学表达

角点应该满足的基本性质：最小特征值尽量大

角点响应 $R = \lambda_{\min}$



Good features to track, Shi and Tomasi, 1994

点特征检测—Harris角点检测

比 $R = \lambda_{\min}$ 更有效的角点响应函数：

$$R = \det(H) - \alpha \operatorname{trace}(H)^2 = \lambda_0 \lambda_1 - \alpha (\lambda_0 + \lambda_1)^2 \quad \alpha = 0.06$$

Harris and Stephens, 1988

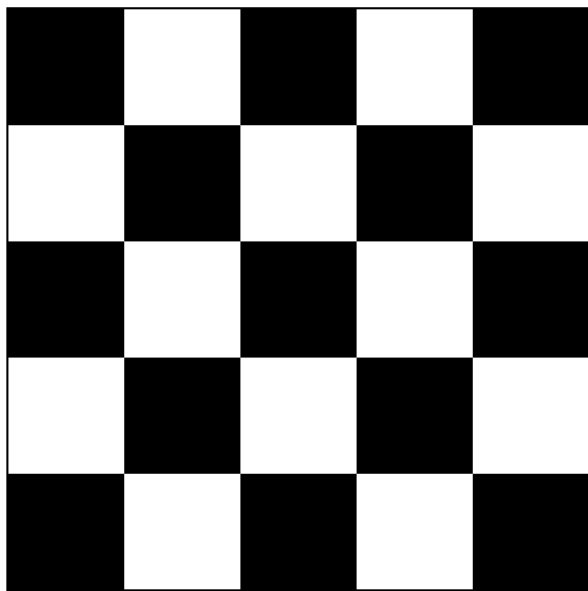
$$R = \lambda_0 - \alpha \lambda_1 \quad \alpha = 0.05 \quad \text{Triggs, 2004}$$

$$R = \frac{\det(H)}{\operatorname{trace}(H)} = \frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1} \quad \text{Brown, Szeliski and Winder, 2005}$$

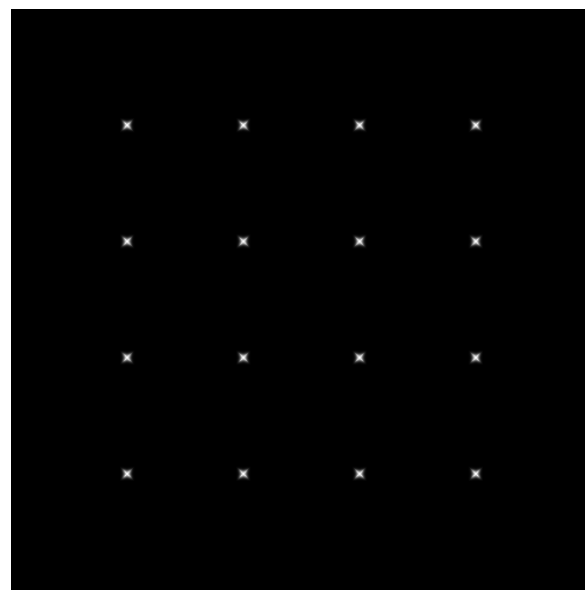
点特征检测—Harris角点检测

算法:

1. 将原图像 I 使用 $\omega(x, y)$ 进行卷积, 并计算图像梯度 I_x 与 I_y ;
2. 计算每一个图像点的自相关矩阵 H ;
3. 计算角点响应 $R = \det(H) - \alpha \text{trace}(H)^2$;
4. 选择 R 大于阈值且为局部极大值的点作为角点。



I

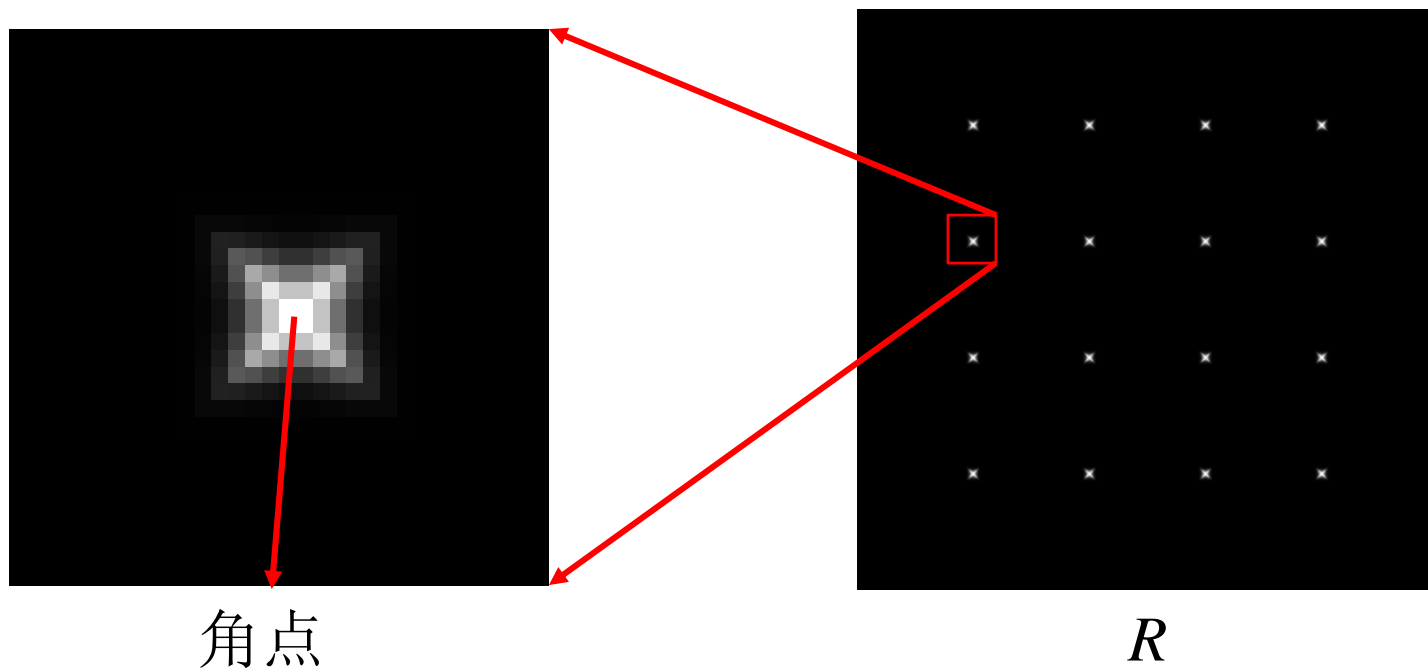


R

点特征检测—Harris角点检测

算法：

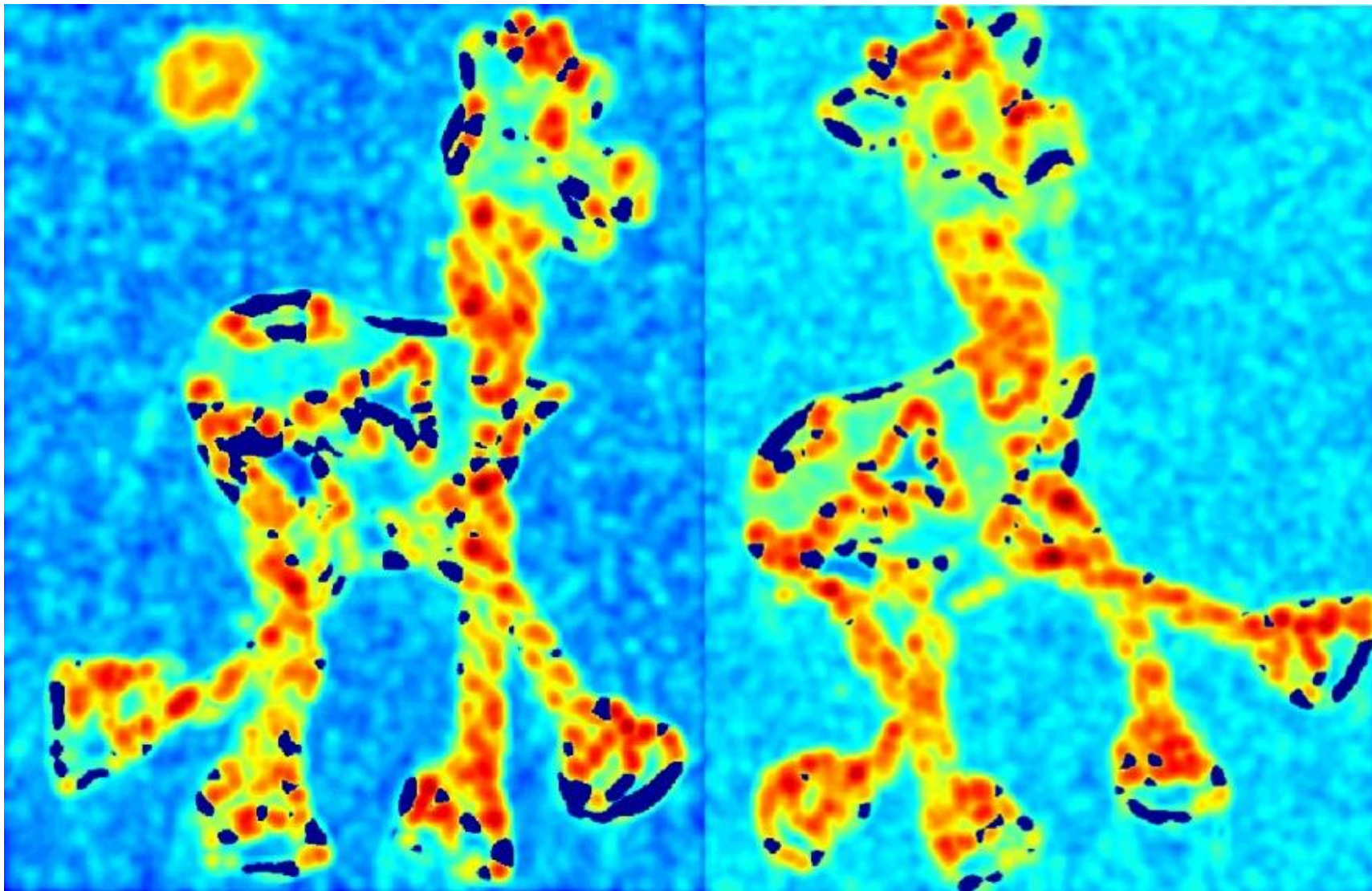
1. 将原图像 I 使用 $\omega(x, y)$ 进行卷积，并计算图像梯度 I_x 与 I_y ；
2. 计算每一个图像点的自相关矩阵 H ；
3. 计算角点响应 $R = \det(H) - \alpha \text{trace}(H)^2$ ；
4. 选择 R 大于阈值且为局部极大值的点作为角点。



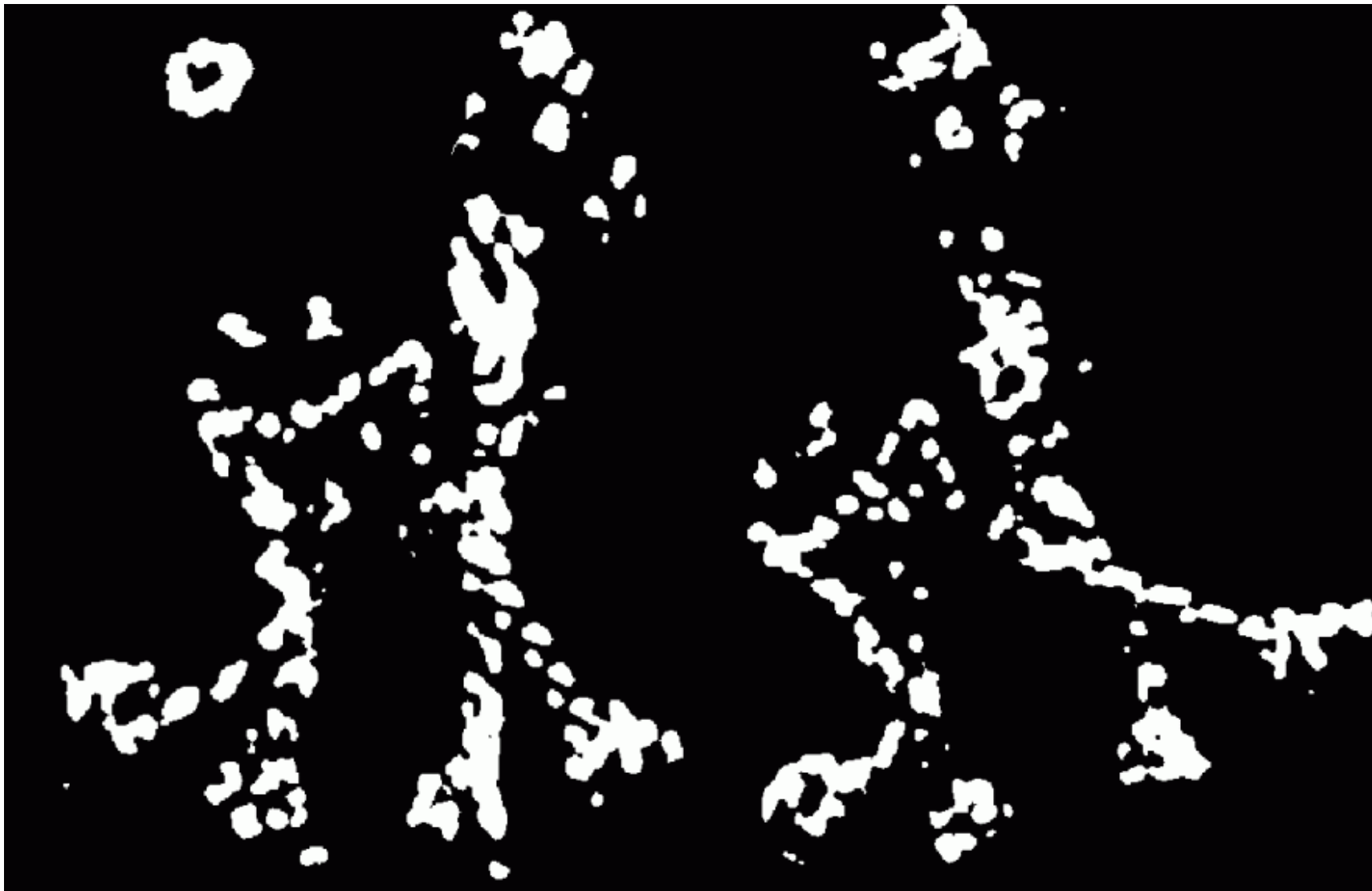
点特征检测—Harris角点检测



点特征检测—Harris角点检测



点特征检测—Harris角点检测



点特征检测—Harris角点检测



点特征检测—Harris角点检测

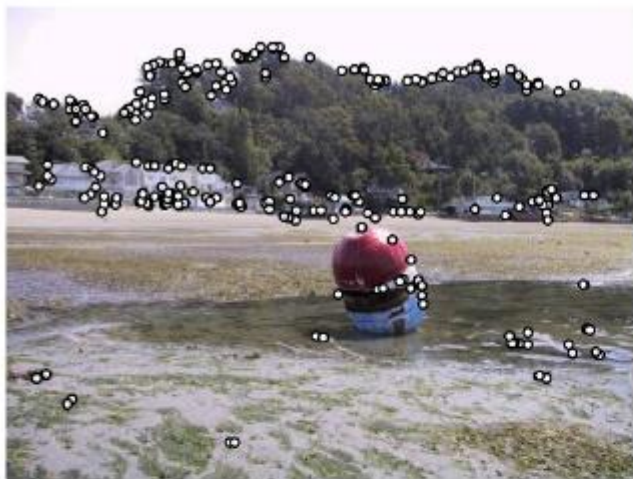


点特征检测—Harris角点改进

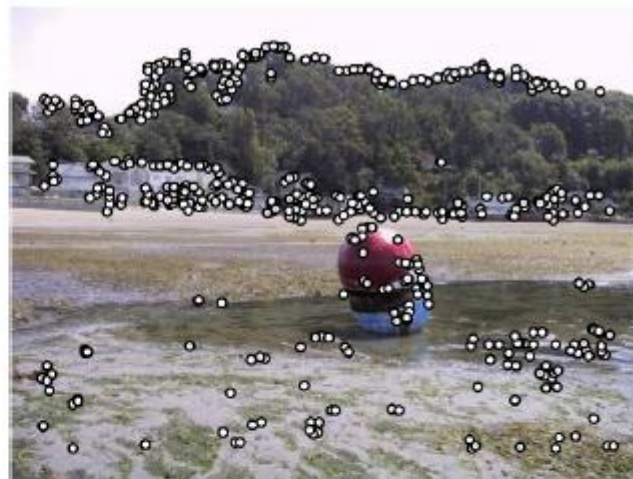
Harris检测子获得的角点可能在图像上分布不均匀（对比度高的区域角点多）

改进方法：Adaptive non-maximal suppression (ANMS)，只保留半径 r 内角点响应比其他点大10%的点作为角点。(Brown, Szeliski and Winder, 2005)

点特征检测—Harris角点改进



(a) Strongest 250



(b) Strongest 500



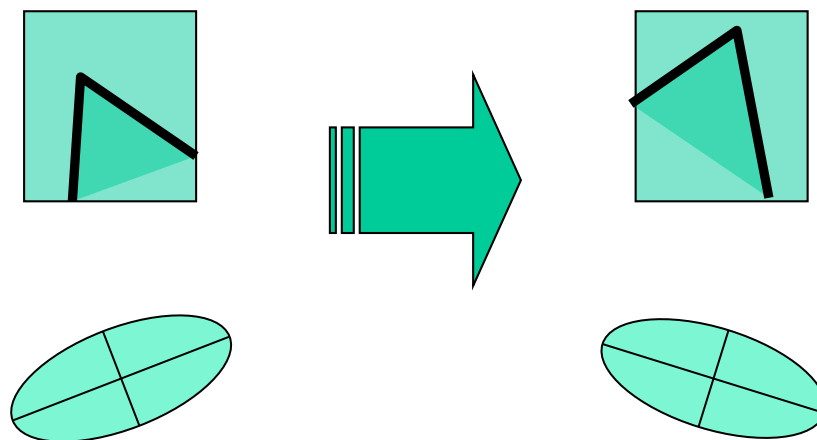
(c) ANMS 250, $r = 24$



(d) ANMS 500, $r = 16$

点特征检测—Harris角点性质

旋转不变:



椭圆转过一定角度但是其形状保持不变（特征值保持不变）

点特征检测—Harris角点性质

光照变化不变：



I



$I - 100$

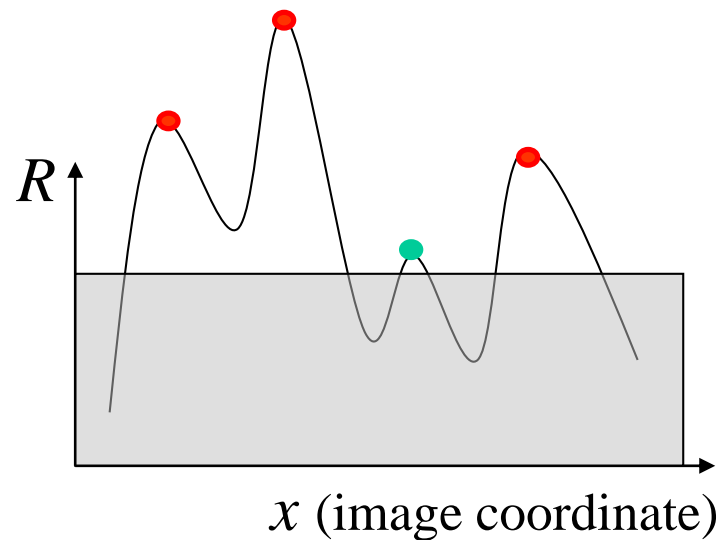
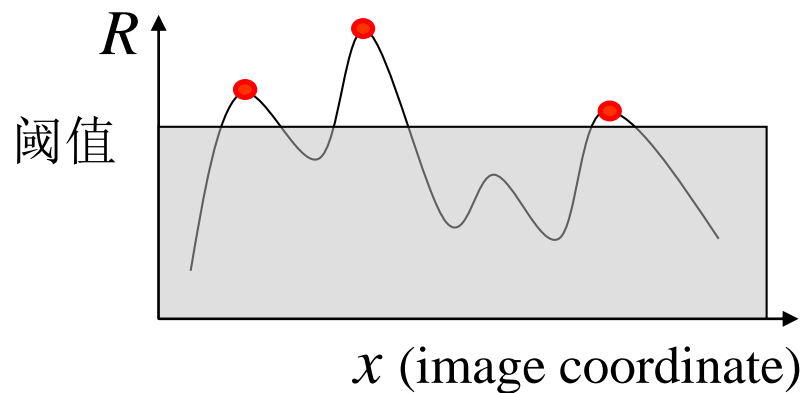


I_x

$H = \omega(x, y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$ 只使用了图像导数，对于光照线性变化不变。

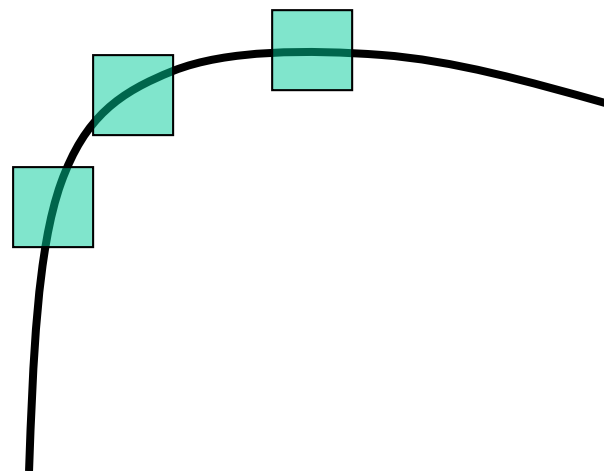
点特征检测—Harris角点性质

对比度变化部分不变：



点特征检测—Harris角点性质

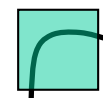
对于图像尺度变化不具有不变性：



这几个点被分类为边缘点



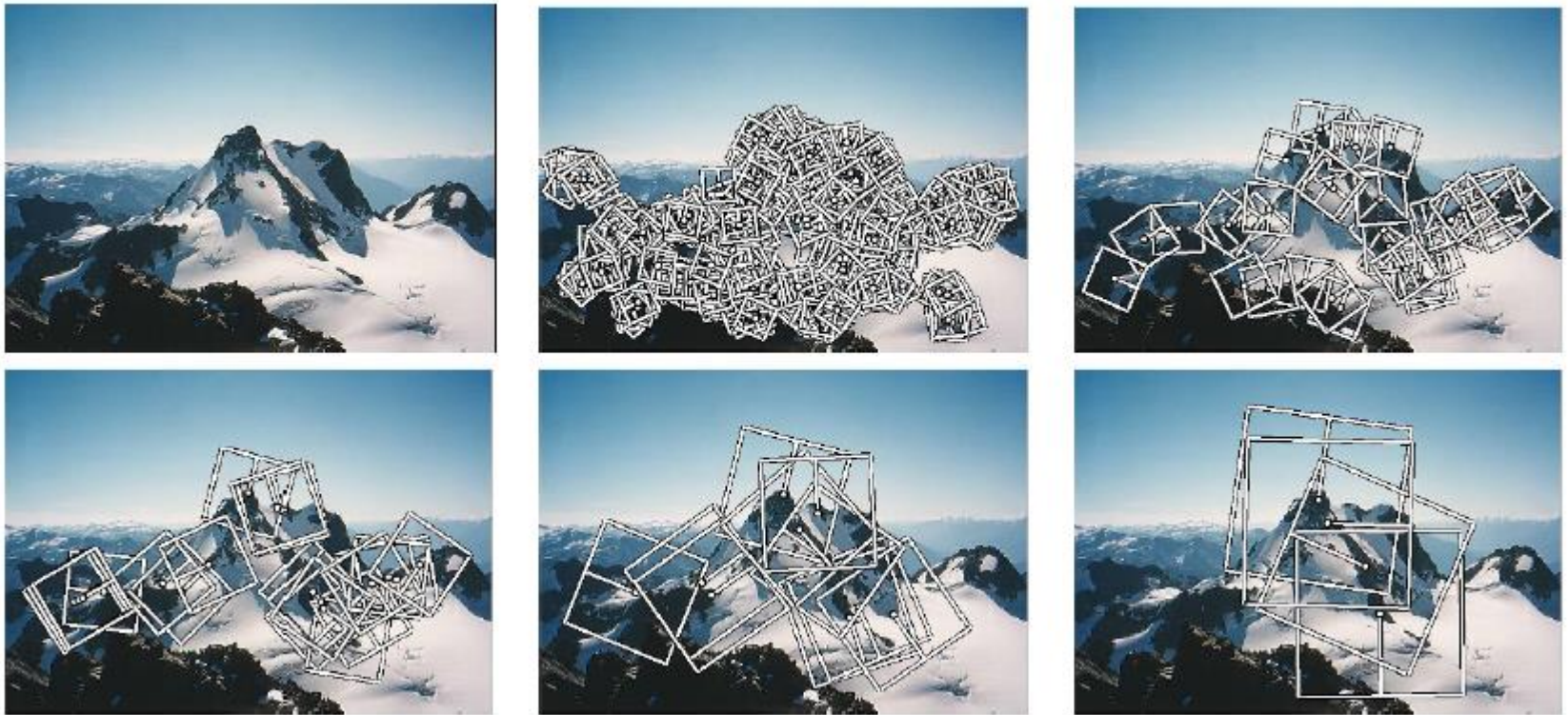
图像缩小



角点！

点特征检测—MOPS

尺度不变：在多层图像金字塔上检测角点，在同一层进行匹配

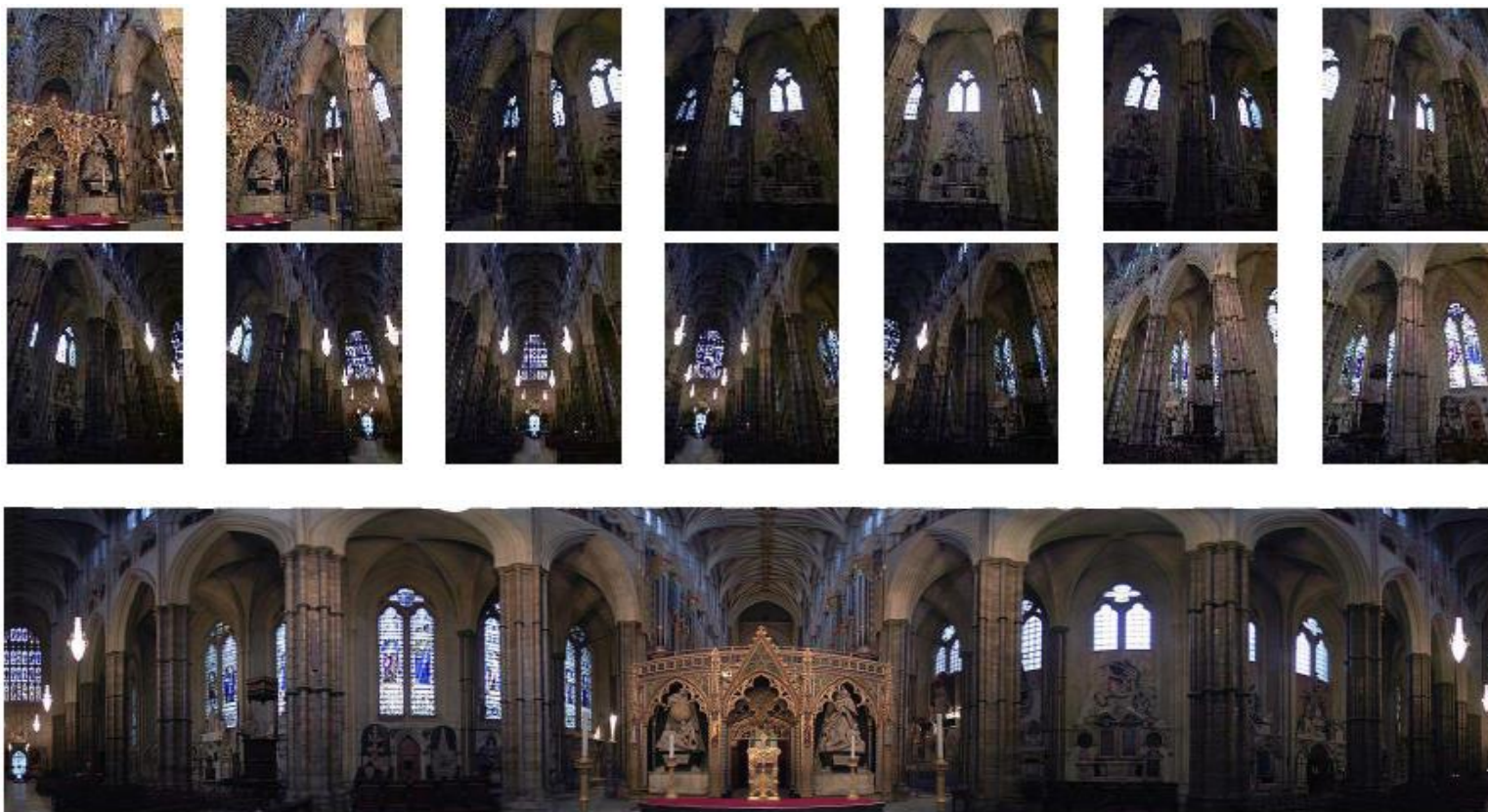


Multi-scale oriented patches (MOPS) (Brown, Szeliski and Winder, 2005)

点特征检测—MOPS

尺度不变：在多层图像金字塔上检测角点，在同一层进行匹配

MOPS的局限：待匹配的图像需要尺度近似



点特征检测—SIFT

Scale Invariant Feature Transform (SIFT) (Lowe, 2004)

- 不变性
 - 对图像的旋转和尺度变化具有不变性；
 - 对三维视角变化和光照变化具有很强的适应性；
 - 局部特征，在遮挡和场景杂乱时仍保持不变性；
- 辨别力强
 - 特征之间相互区分的能力强，有利于匹配；
- 数量较多
 - 一般 500×500 的图像能提取出约2000个特征点。

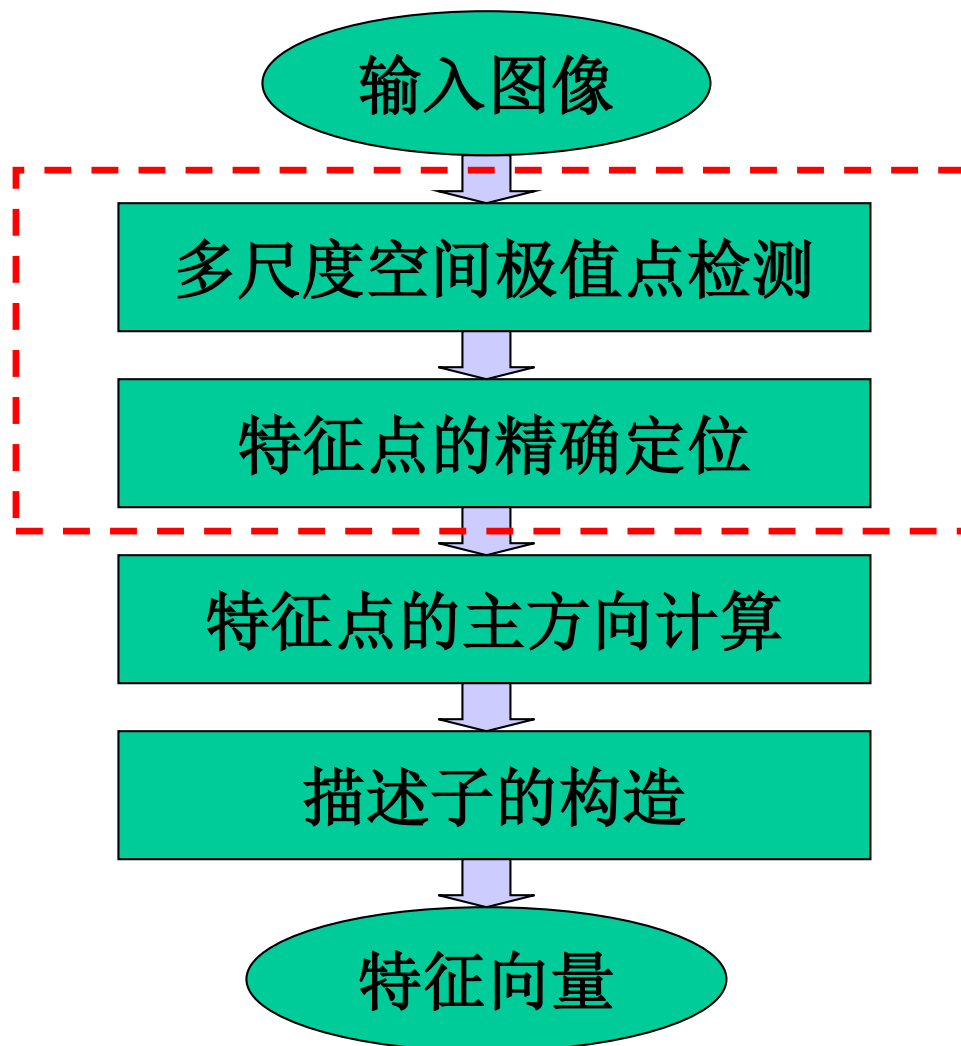
David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”
International Journal of Computer Vision 60(2): 91-110 (2004)

[23250 cites](#)

点特征检测—SIFT

Scale Invariant Feature Transform (SIFT) (Lowe, 2004)

在高斯差分(Difference of Gaussian, DOG)尺度空间中提取极值点并进行优化从而获取特征点。



点特征检测—SIFT

构建尺度空间：模拟图像数据的多尺度特征

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

其中 $G(x, y, \sigma)$ 是尺度可变高斯函数

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

尺度参数 σ 决定图像的平滑程度，大尺度对应图像的概貌特征，小尺度对应图像的细节特征

点特征检测—SIFT

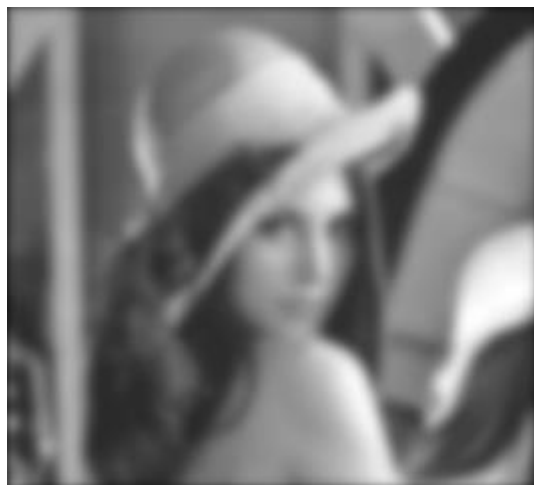
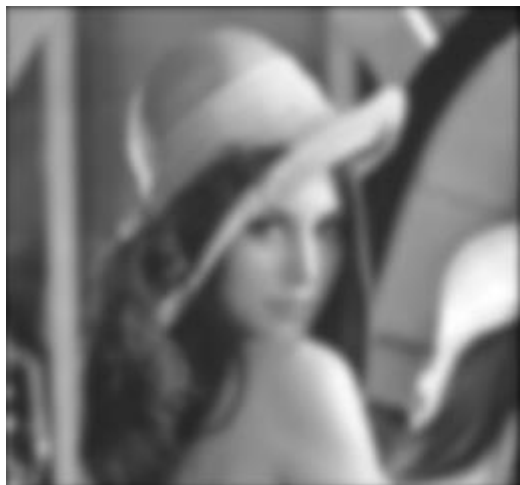
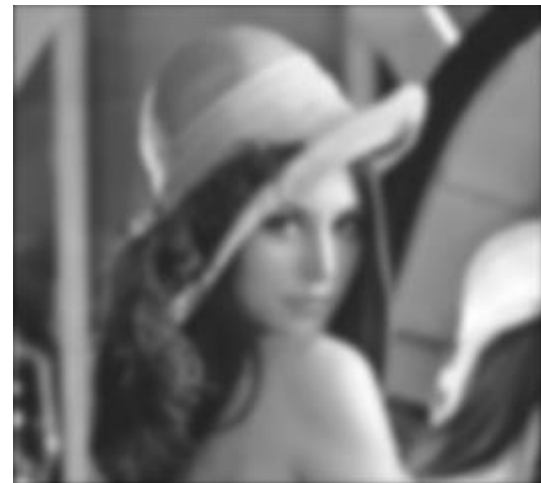
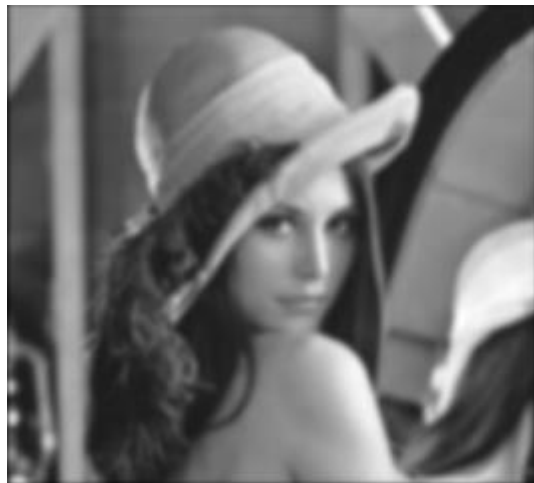
为了在尺度空间中检测稳定的关键点，构造高斯差分尺度空间（Difference of Gaussian, DOG）

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

使用DOG的几个理由：

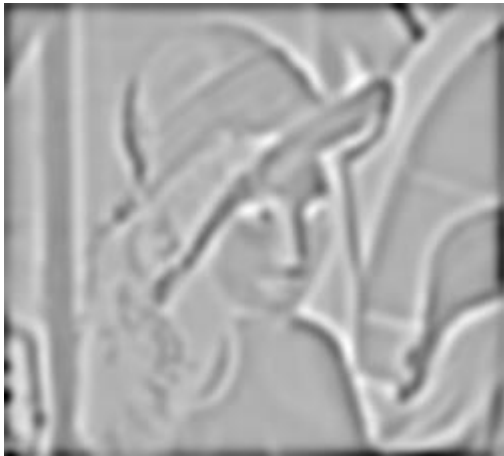
- 计算效率高：高斯卷积，减法；
- 高斯差分是对尺度归一化LoG（ $\sigma^2 \nabla^2 G$ ）的一个很好的近似，而尺度归一化的LoG空间具有真正的尺度不变性 (Lindeberg 1994)；
- 实验比较表明，从尺度归一化LoG空间中提取的图像特征的尺度稳定性最好，优于梯度、Hessian或Harris角点函数。

点特征检测—SIFT



高斯尺度空间

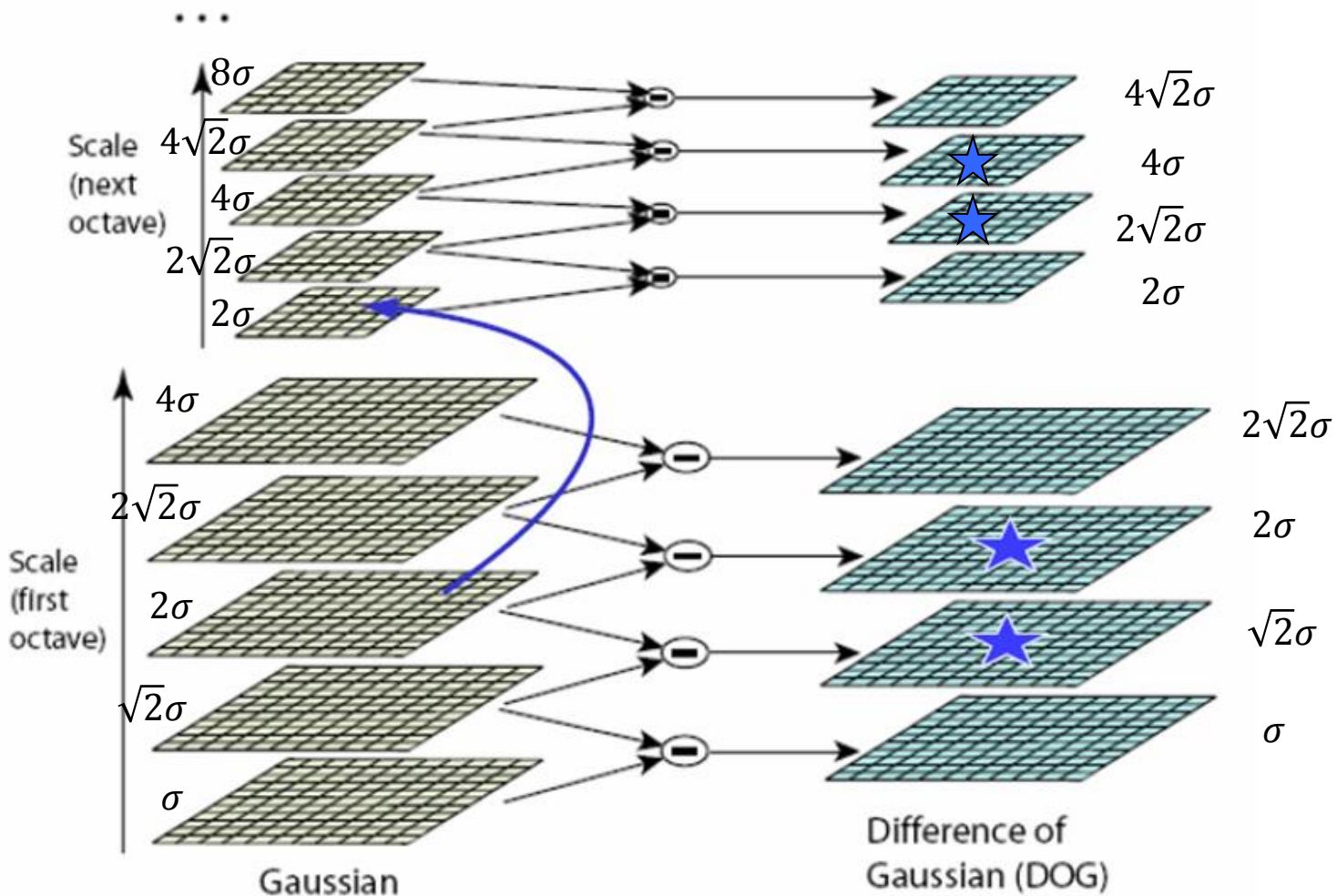
点特征检测—SIFT



DOG尺度空间

点特征检测—SIFT

DoG尺度空间, $D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$



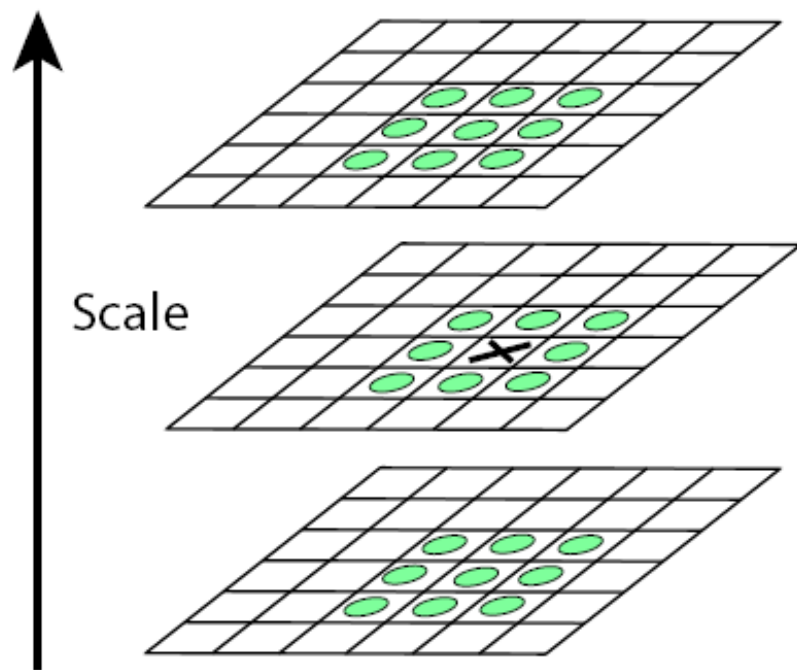
点特征检测—SIFT

DoG尺度空间极值点检测

一个点和它同尺度的8个相邻点以及上下相邻尺度对应的 9×2 个点共26个点比较，以确保在尺度空间和二维图像空间都检测到极值点。

一个点如果在DOG尺度空间的26个领域中是最大或最小值时，就认为该点是图像在该尺度下的一个特征点。

一个特征点是在三维尺度空间 $D(x, y, \sigma)$ 的局部极值点。



点特征检测—SIFT

特征点精确定位

一个特征点是在三维尺度空间 $D(x, y, \sigma)$ 的局部极值点。但 (x, y) 为整数像素， σ 为离散尺度，需要对DOG空间拟合进行特征点精确定位。

将 $D(x, y, \sigma)$ 在特征点 $\mathbf{x} = [x, y, \sigma]$ 处二阶Taylor展开：

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

对上式求导，并令其为0，得到精确的位置（偏移量）：

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}$$

若 $\hat{\mathbf{x}} = [\hat{x}, \hat{y}, \hat{\sigma}]$ 中的三个变量任意一个偏移量大于0.5，说明精确极值点更接近于另一个特征点，则更换特征点重复上述精确定位流程。

点特征检测—SIFT

去除不稳定特征点

去除对比度低的点：计算极值点取值 $D(\hat{\mathbf{x}}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}$

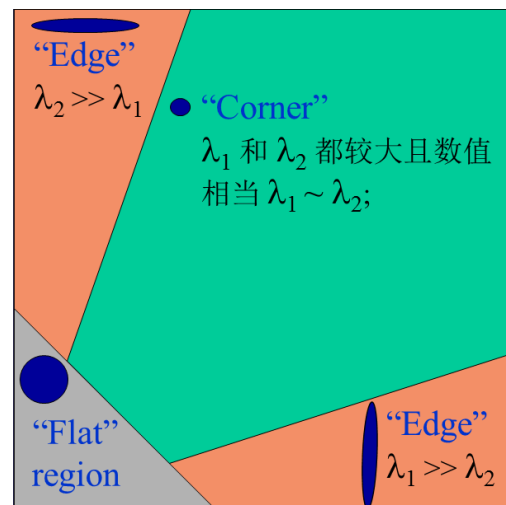
若 $|D(\hat{\mathbf{x}})| \geq 0.03$ ，则保留该特征点，否则丢弃。

去除边缘点：DoG算子会产生较强的边缘响应，利用Harris检测子判断。

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\text{若 } \frac{\text{Tr}^2(H)}{\text{Det}(H)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} \leq \frac{(10+1)^2}{10},$$

则保留该特征点，否则丢弃。



点特征检测—SIFT

一个例子



原始图像大小， 233×189 像素



多尺度DoG空间中的极值点，832个

点特征检测—SIFT

一个例子



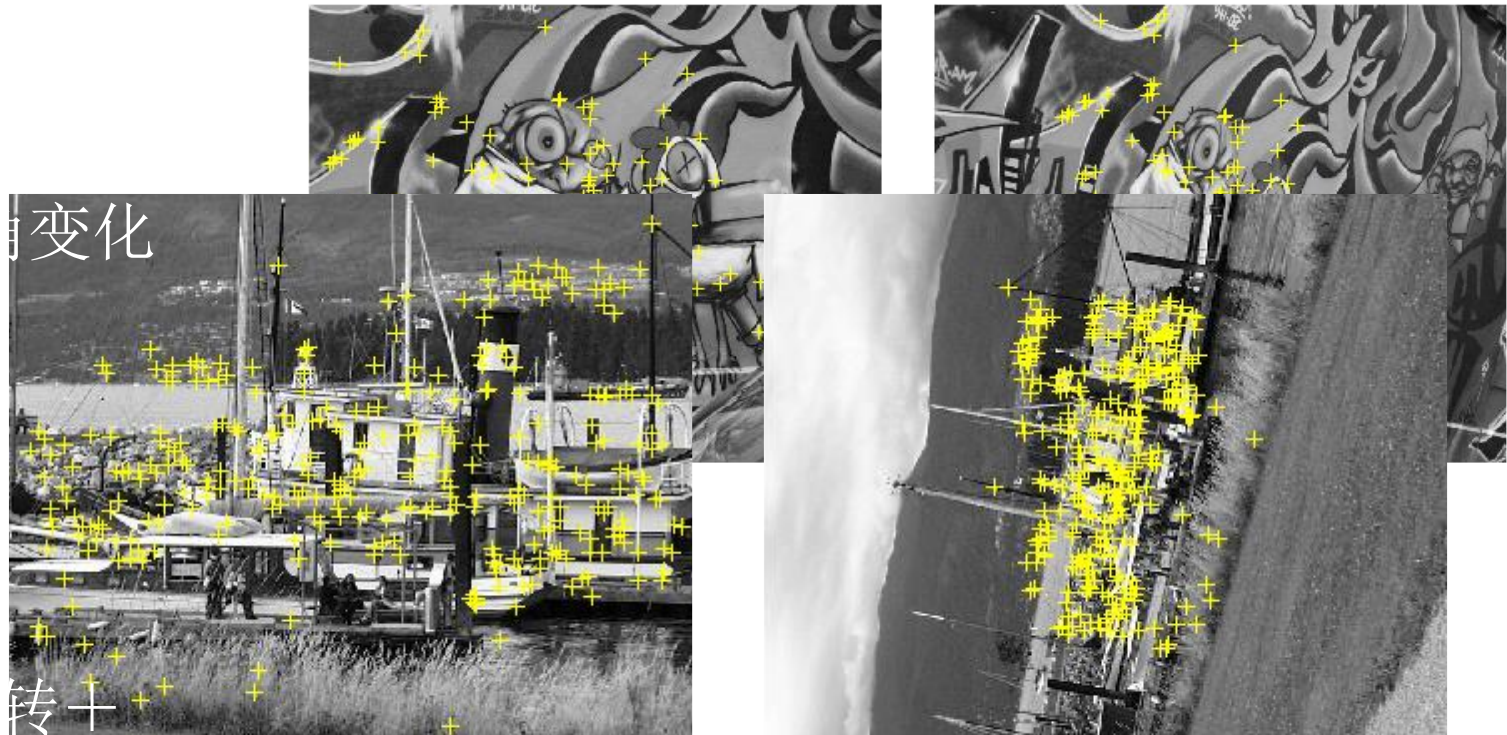
对比度阈值处理, $832 \rightarrow 729$



去除边缘上的点, $729 \rightarrow 536$

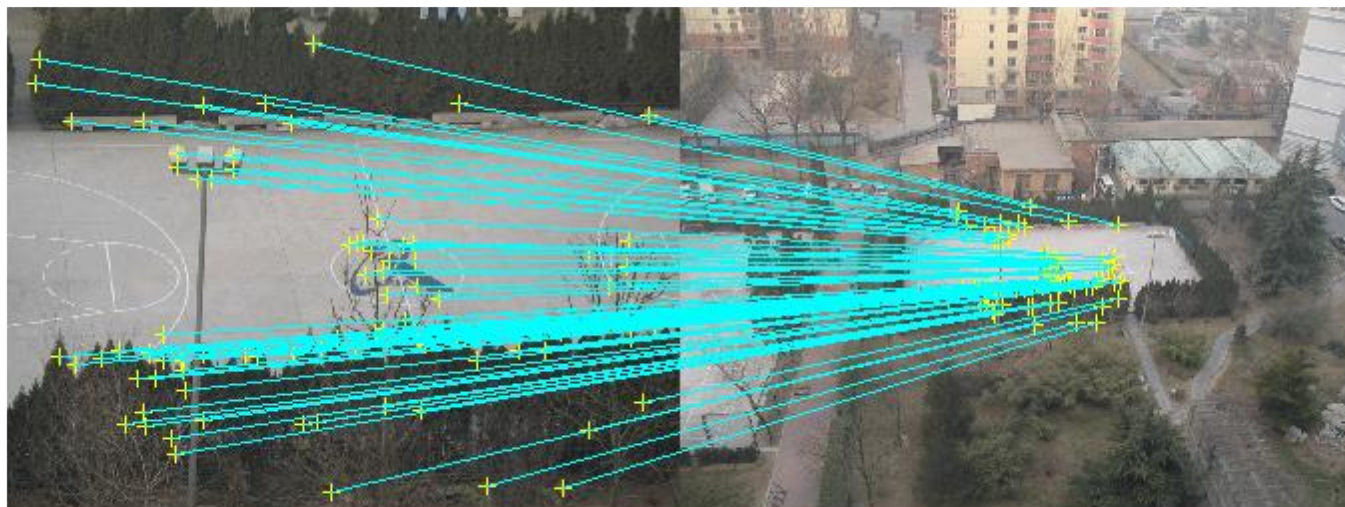
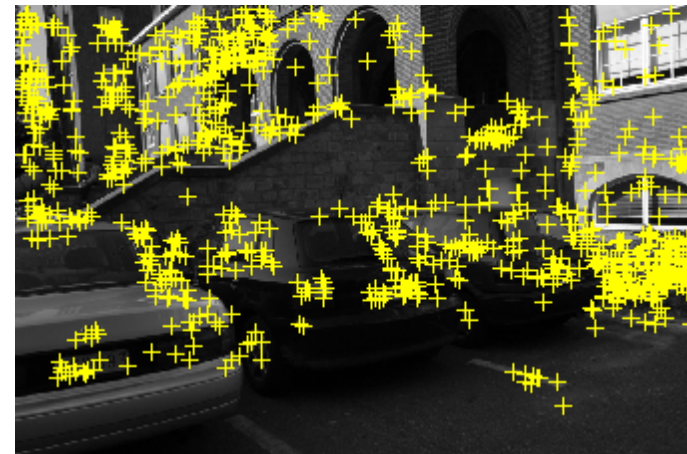
点特征检测—SIFT

SIFT点—视角和旋转变化



点特征检测—SIFT

SIFT点—光照和尺度变化



点特征检测—小结

- 根据自相关矩阵特征值检测角点(Harris);
Harris角点具有旋转、光照不变性，但不具有尺度不变性。
- 高斯差分尺度空间中检测尺度不变特征点(SIFT) ;
SIFT具有尺度不变性。
- Harris与SIFT的机理不同，因此可以联合使用，互为补充。

边缘检测

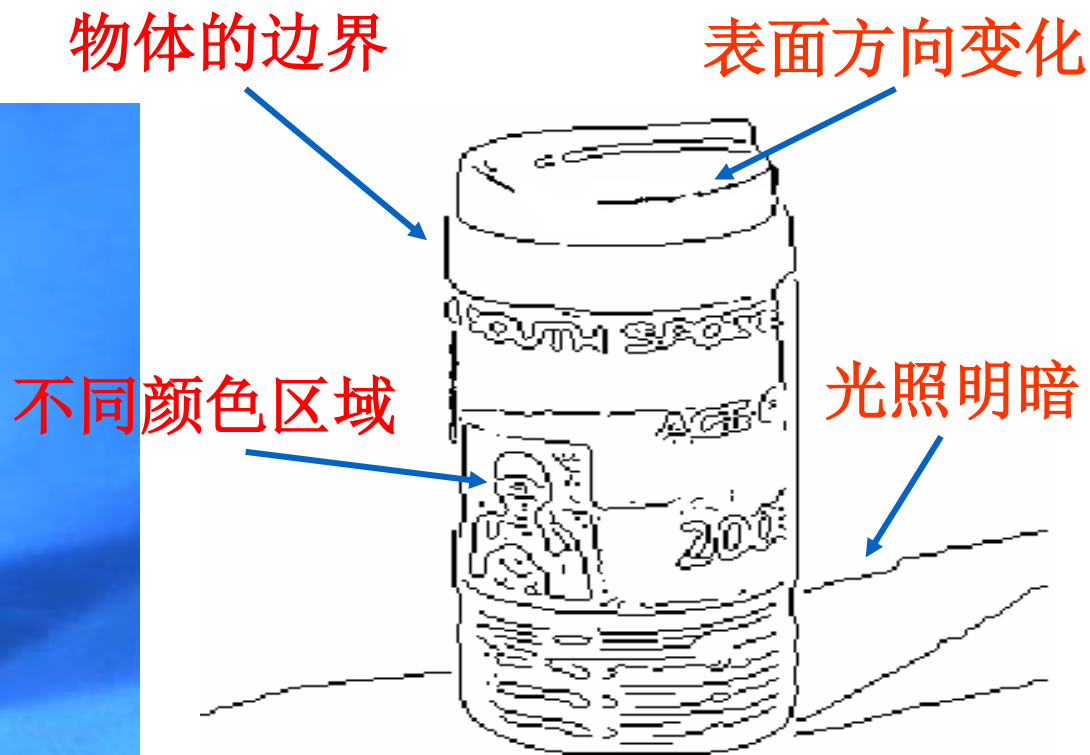
为什么要检测边缘？



基于边缘的图像编辑 (Elder and Goldberg 2001)

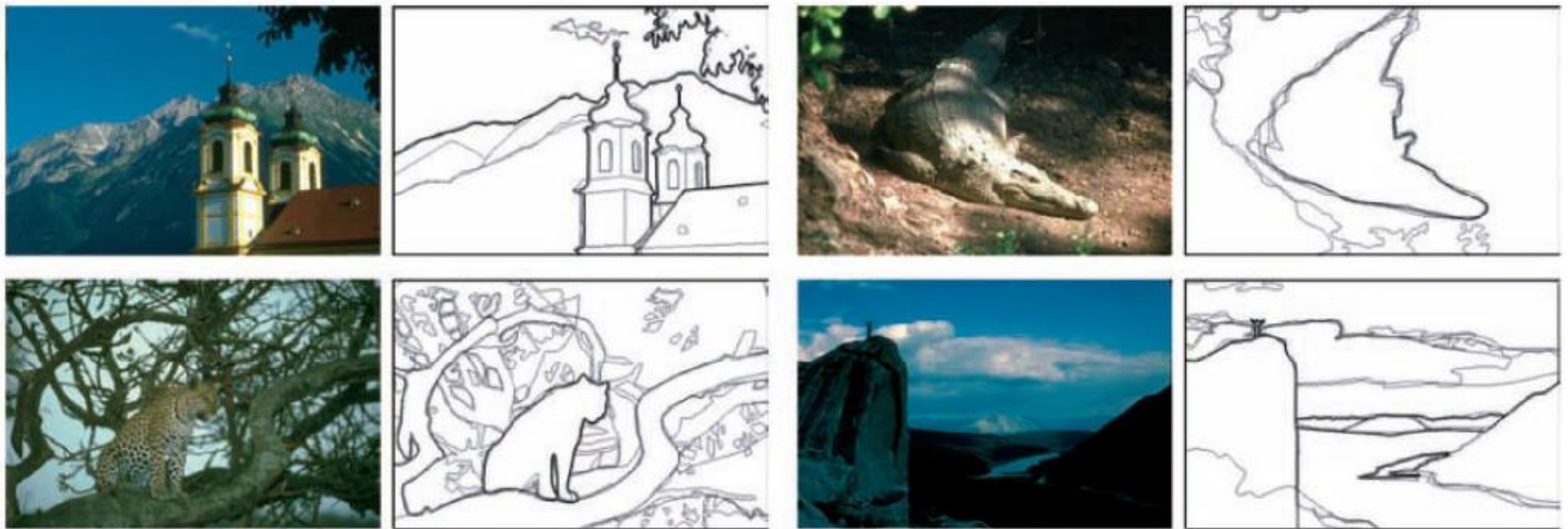
边缘检测

图像边缘的产生：物体的边界、表面方向的变化、不同的颜色、光照明暗的变化。



边缘检测

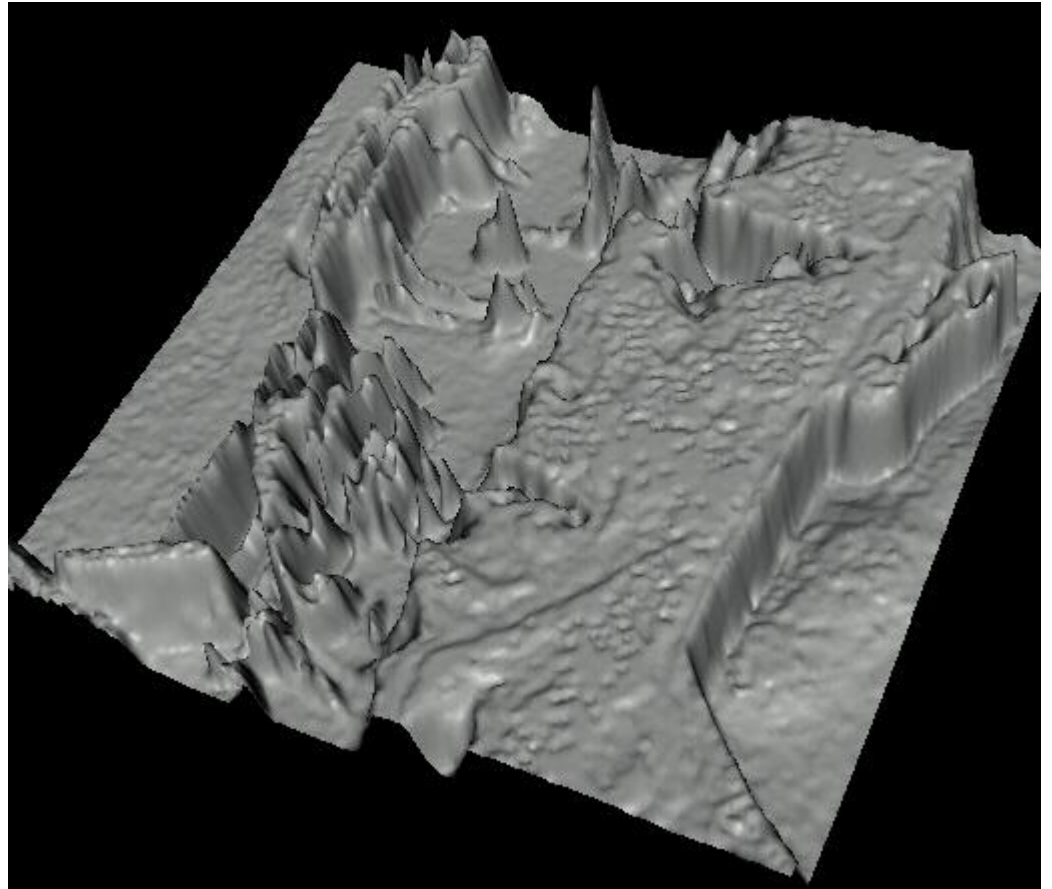
如何判断一个像素是否在边缘上？



Human boundary detection ([Martin et. al, 2004](#))

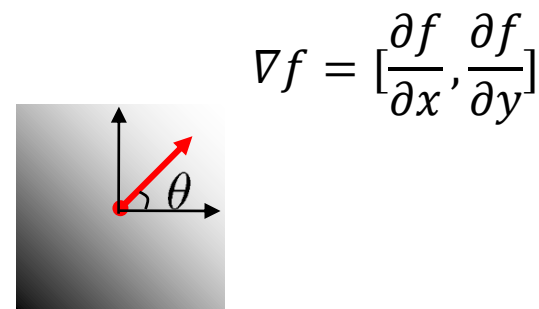
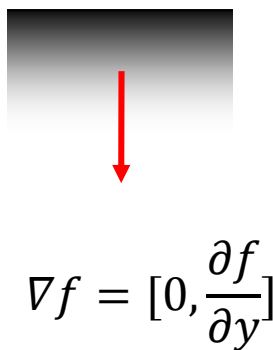
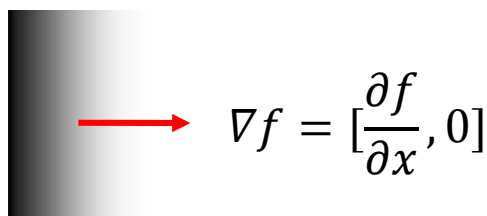
边缘检测

如果把图像看作二维函数，那么边缘就是悬崖峭壁



边缘检测—图像梯度

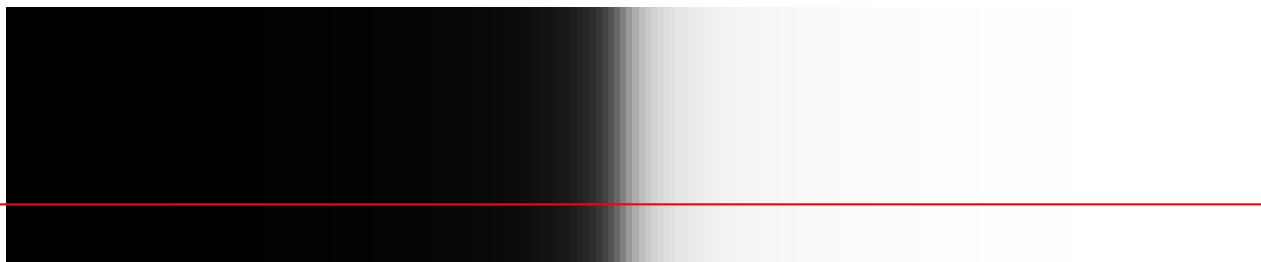
图像梯度的定义： $\nabla I = [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}]$



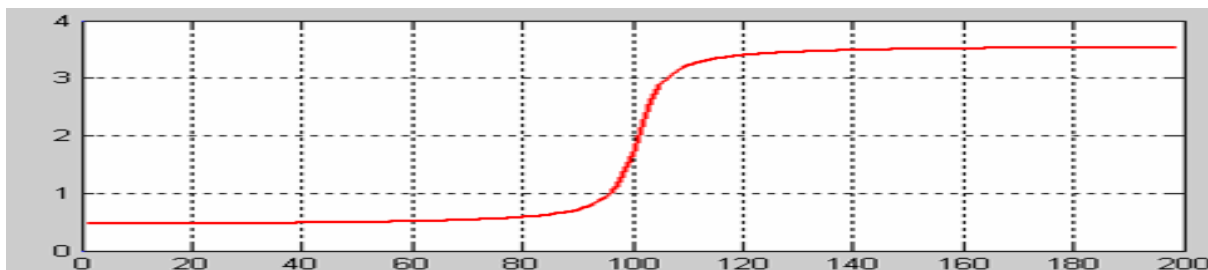
图像梯度指向灰度变化最快的方向， $\theta(x, y) = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$

梯度幅值表示边缘的强弱， $M(x, y) = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

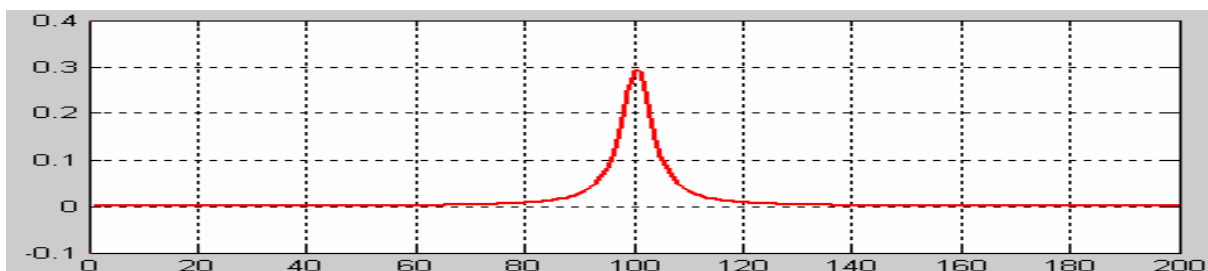
边缘检测—图像梯度



$$f(x)$$



$$\frac{d}{dx}f(x)$$



边缘是一阶倒数的极大值点

边缘检测——一阶微分算子

使用差分近似一阶微分算子：

$$\frac{\partial I}{\partial x} \approx \frac{I(x+1, y) - I(x-1, y)}{2} \longleftrightarrow \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline \end{array} \times 0.5$$

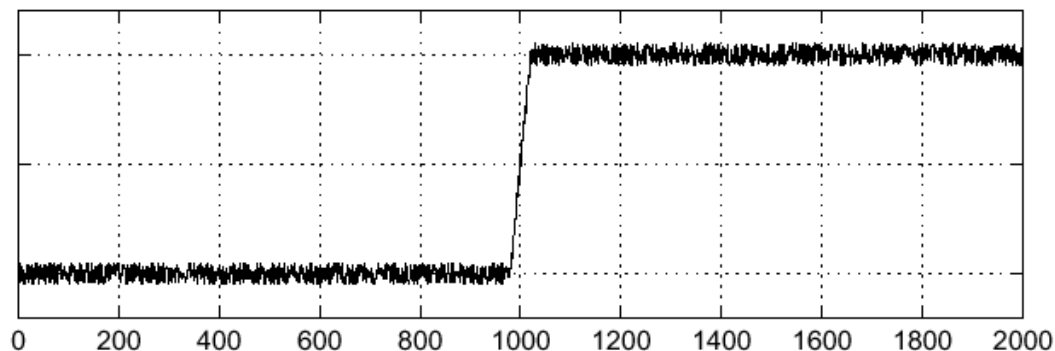
$$\frac{\partial I}{\partial y} \approx \frac{I(x, y+1) - I(x, y-1)}{2} \longleftrightarrow \begin{array}{|c|} \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array} \times 0.5$$

水平边缘

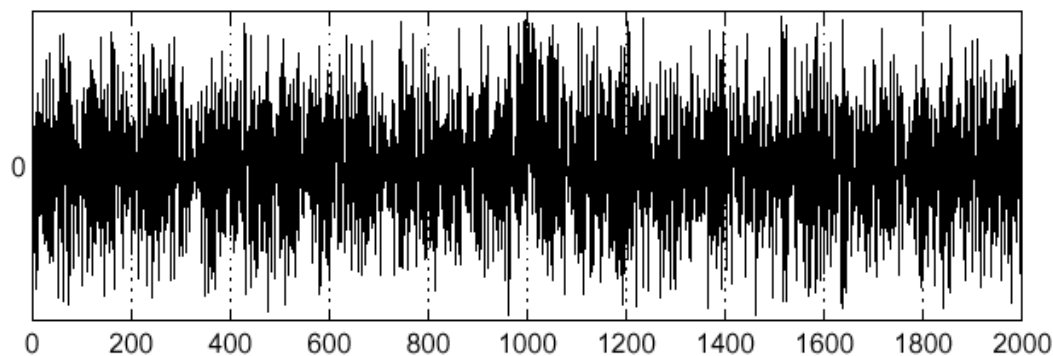
边缘检测——一阶微分算子

直接对图像使用差分容易受到噪声影响：

$f(x)$



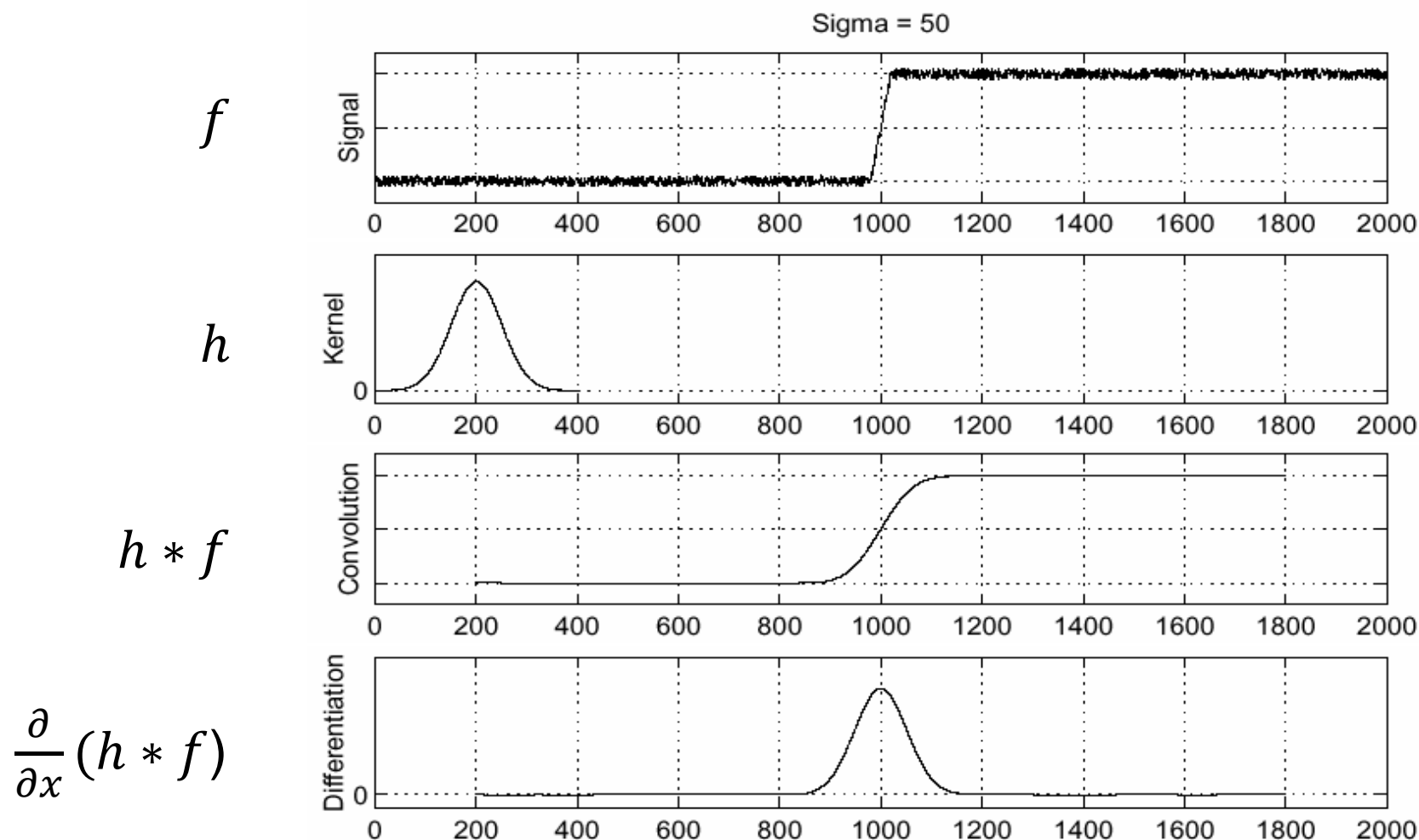
$\frac{d}{dx}f(x)$



边缘在哪里？

边缘检测——一阶微分算子

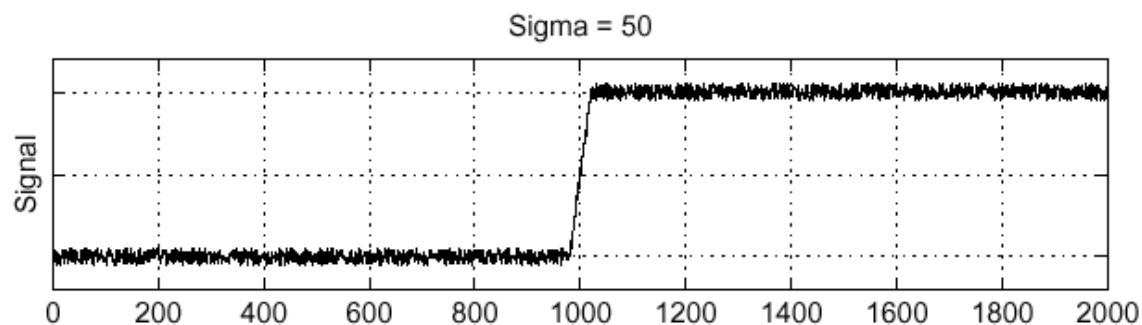
解决方法：先平滑，再微分



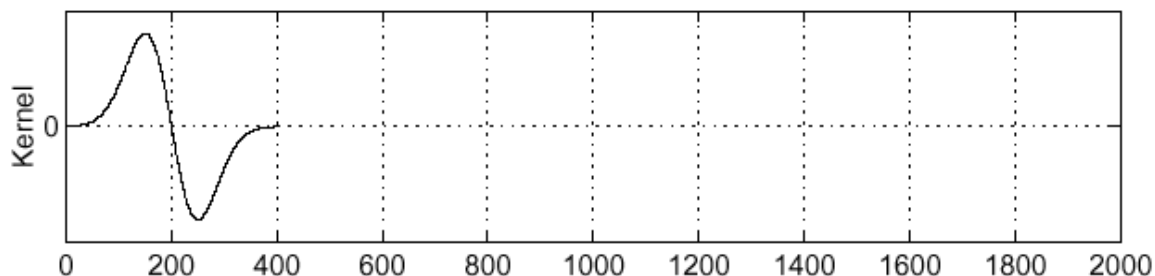
边缘检测——一阶微分算子

由 $\frac{\partial}{\partial x}(h * f) = (\frac{\partial}{\partial x}h) * f$ ，可将平滑和微分合为一个算子：

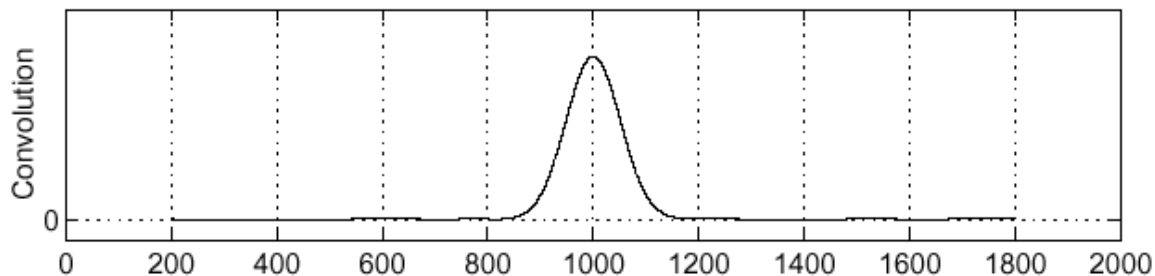
f



$\frac{\partial}{\partial x}h$

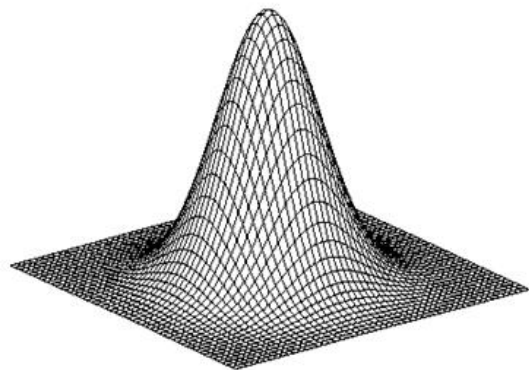


$(\frac{\partial}{\partial x}h) * f$



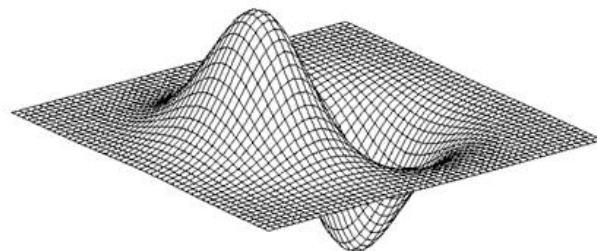
边缘检测——一阶微分算子

二维高斯微分



Gaussian

$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



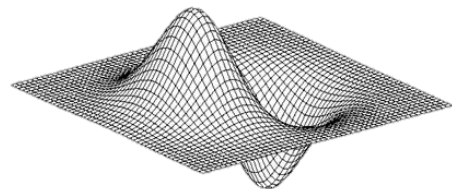
derivative of Gaussian

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

实际应用中：对二维高斯微分进行数值近似
Prewitt算子、Sobel算子

边缘检测——一阶微分算子

Prewitt算子：去噪 + 增强边缘



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

计算均值，
平滑噪声

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

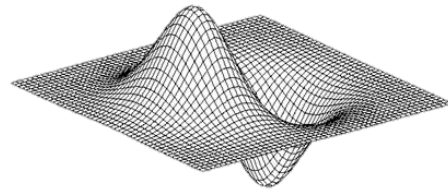
检测水
平边缘

检测竖直边缘

计算均值，
平滑噪声

边缘检测——一阶微分算子

Sobel算子：去噪 + 增强边缘，
给四邻域更大的权重



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

计算均值，
平滑噪声

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

检测水
平边缘

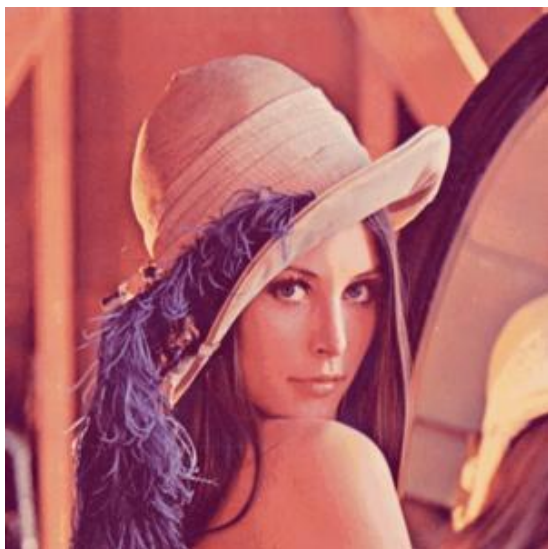
检测竖直边缘

计算均值，
平滑噪声

边缘检测——一阶微分算子

使用一阶微分算子提取边缘流程：

1. 使用Prewitt或Sobel算子对图像进行卷积；
2. 将梯度幅值大于阈值的点标记为边缘；
3. (optional)将边缘细化为一个像素宽度。



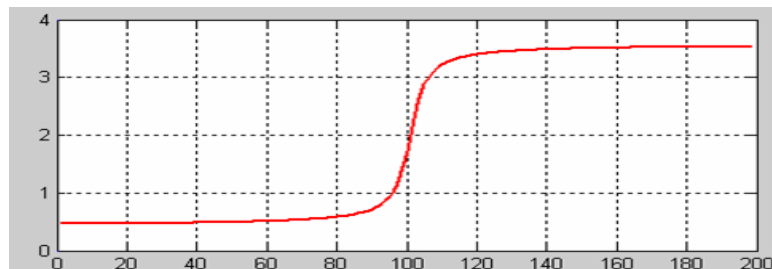
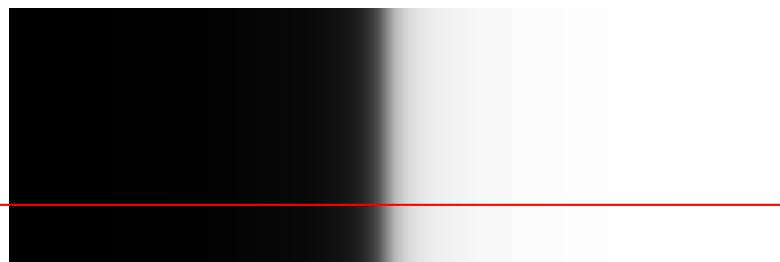
Sobel, no thinning



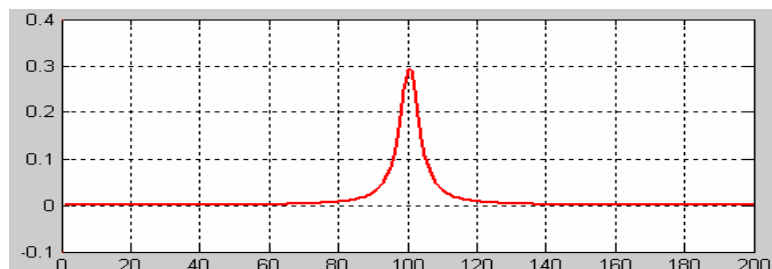
Sobel, thinning

边缘检测—二阶微分算子

如果不使用细化，如何获得单像素宽度边缘？

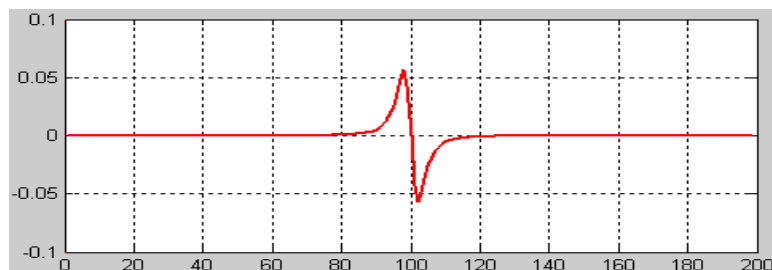


边缘是一阶倒数的极大值点



边缘是二阶倒数的过零点

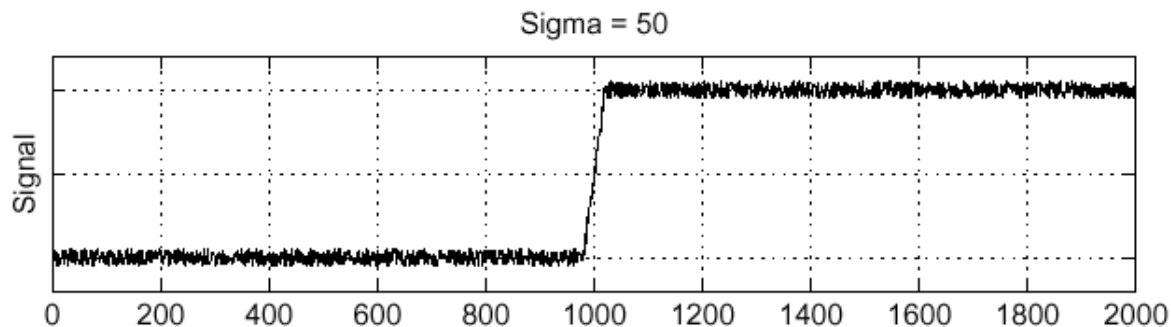
注意：仅仅等于0不够，常数函数也为0，必须存在符号改变



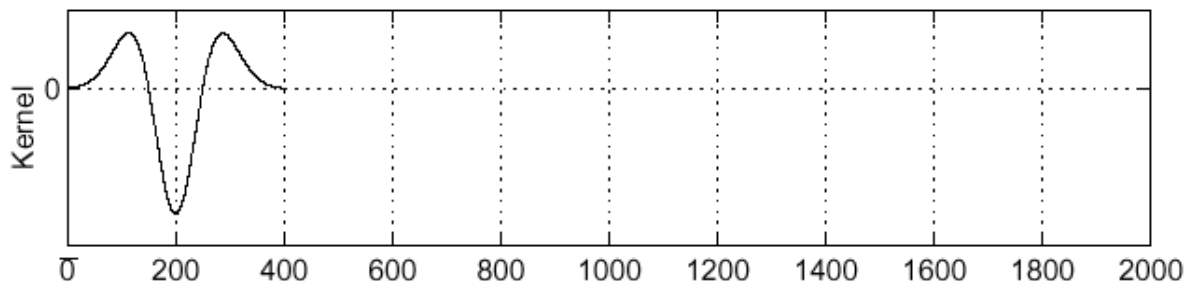
边缘检测—二阶微分算子

对平滑图像做二阶微分： $\frac{\partial^2}{\partial x^2} (h * f) = (\frac{\partial^2}{\partial x^2} h) * f$

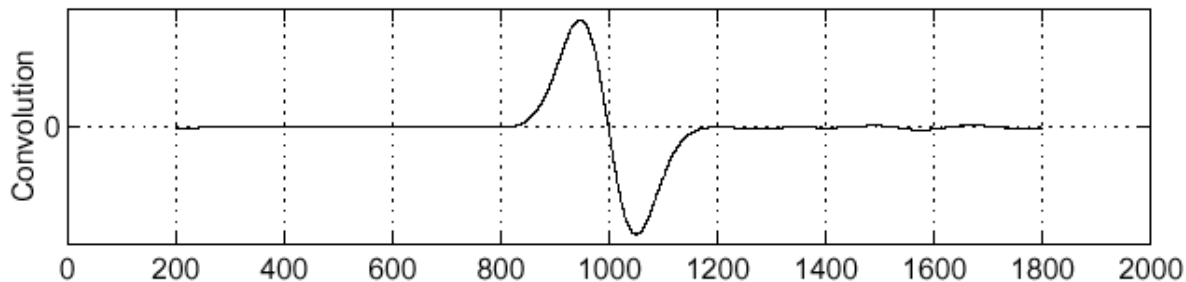
f



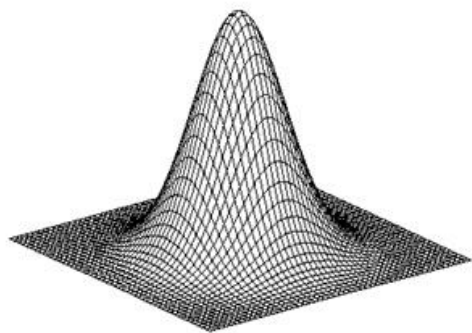
$\frac{\partial^2}{\partial x^2} h$



$(\frac{\partial^2}{\partial x^2} h) * f$

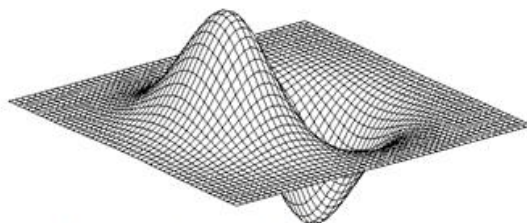


边缘检测—二阶微分算子



Gaussian

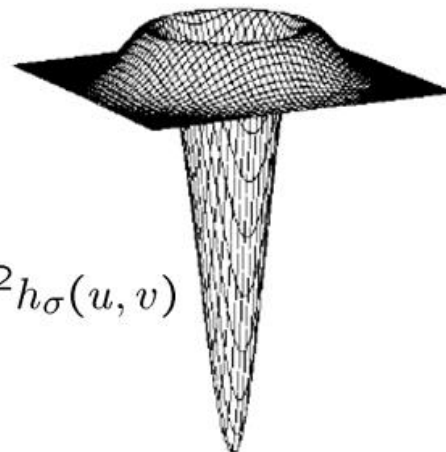
$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

Laplacian of Gaussian

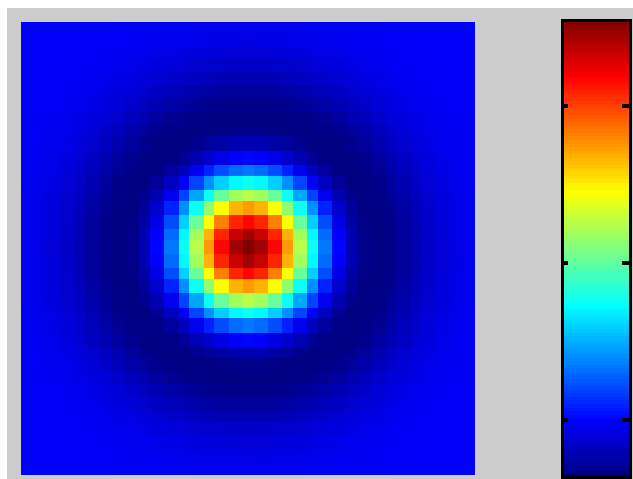
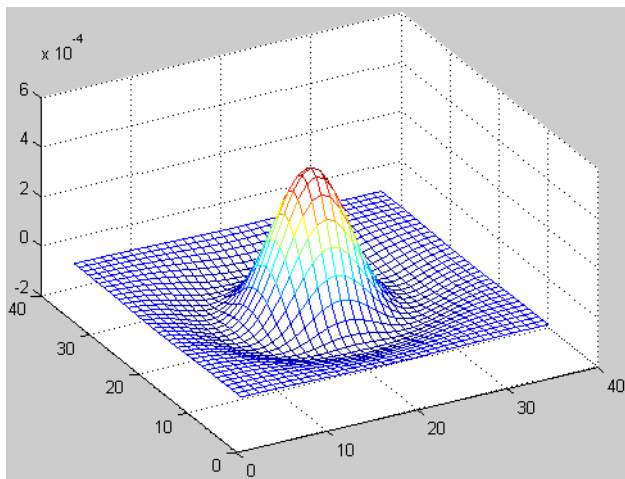


$$\nabla^2 h_{\sigma}(u, v)$$

Laplacian of Gaussian (LoG)算子：首先用Gauss函数对图像进行平滑，抑制噪声，然后对经过平滑的图像使用Laplacian算子

LoG算子等效于： Gaussian平滑 + Laplacian二阶微分

边缘检测—二阶微分算子



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

LoG 因其形状，也称为Mexican hat

边缘检测—二阶微分算子

LoG算子与一阶微分算子的比较



Sobel



LoG

LoG算子的特点：

- 正确检测到的边缘：单像素宽度，定位准确；
- 形成许多封闭的轮廓，这是一个主要问题；
- 需要更加复杂的算法检测过零点。

边缘检测—Canny算子

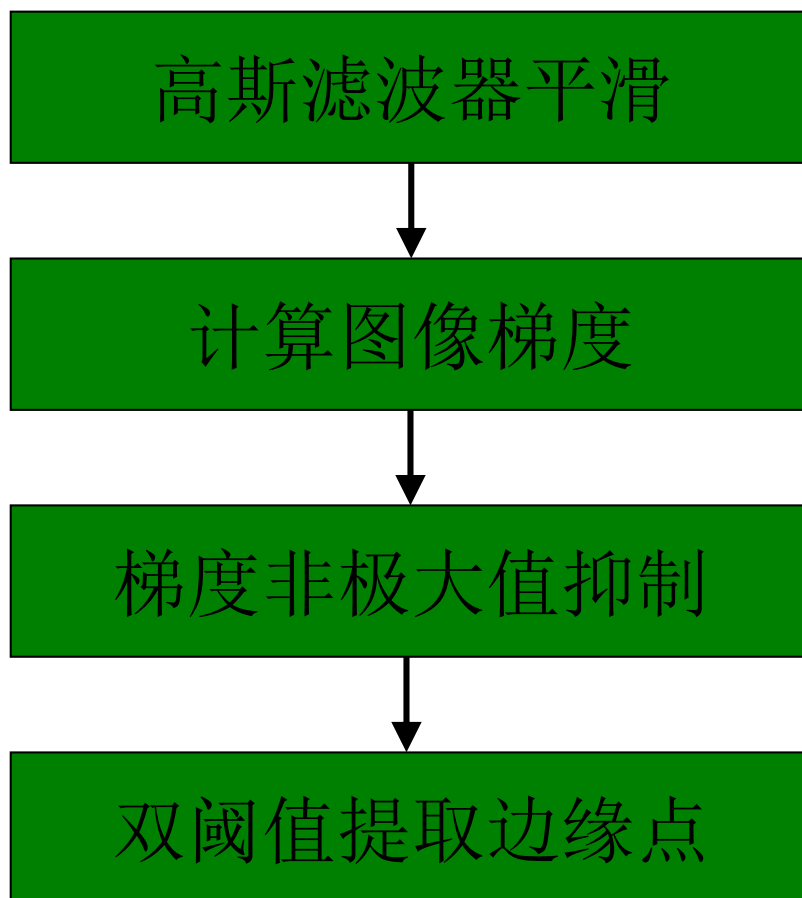
- Canny算子是最常用的边缘检测算子
- Canny算子是一阶微分算子，但是是一个优化的方案
 - 单像素宽度
 - 噪声抑制
 - 边缘增强
 - 边缘定位

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

18482 cites

边缘检测—Canny算子

Canny算子基本流程



边缘检测—Canny算子

高斯函数的一阶导数(Derivative of Gaussian)

(1) 高斯平滑滤波器卷积

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

(2) 使用一阶有限差分计算偏导数的两个阵列

$$D_x(x, y) \approx \frac{S(x, y + 1) - S(x, y) + S(x + 1, y + 1) - S(x + 1, y)}{2}$$

$$D_y(x, y) \approx \frac{S(x, y) - S(x + 1, y) + S(x, y + 1) - S(x + 1, y + 1)}{2}$$

相当于与模版进行卷积运算：

-1	1
-1	1

1	1
-1	-1

边缘检测—Canny算子

高斯函数的一阶导数(Derivative of Gaussian)

(1) 高斯平滑滤波器卷积

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

(2) 使用一阶有限差分计算偏导数的两个阵列

$$D_x(x, y) \approx \frac{S(x, y + 1) - S(x, y) + S(x + 1, y + 1) - S(x + 1, y)}{2}$$

$$D_y(x, y) \approx \frac{S(x, y) - S(x + 1, y) + S(x, y + 1) - S(x + 1, y + 1)}{2}$$

当然也可以使用高斯微分算子(Prewitt或Sobel)直接与 $I(x, y)$ 卷积计算 $D_x(x, y)$ 和 $D_y(x, y)$

边缘检测—Canny算子

高斯函数的一阶导数(Derivative of Gaussian)

(3) 边缘幅值和边缘方位角

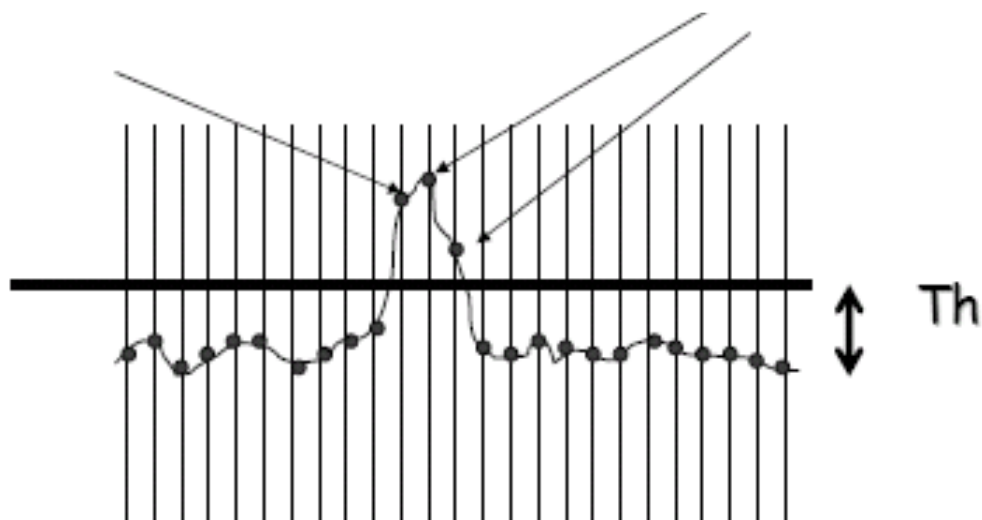
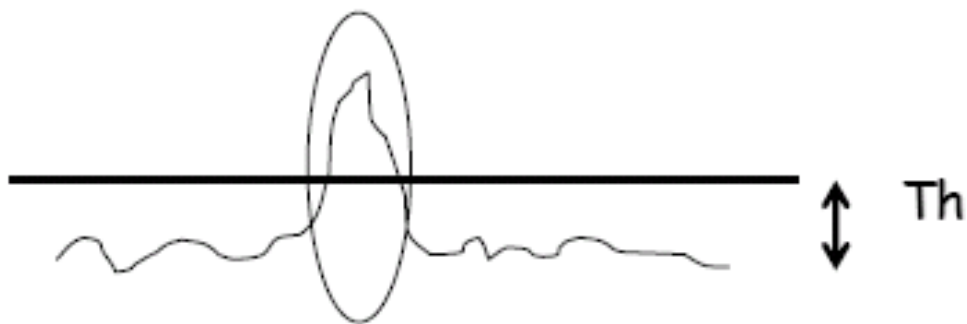
$$M(x, y) = \sqrt{D_x(x, y)^2 + D_y(x, y)^2}$$

$$\theta(x, y) = \tan^{-1}(D_y(x, y)/D_x(x, y))$$

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”。

边缘检测—Canny算子

如何检测边缘



局部极值周围存在
相近数值的点

边缘检测—Canny算子

非极大值抑制（NMS: Non-Maxima Suppression）

主要思想：

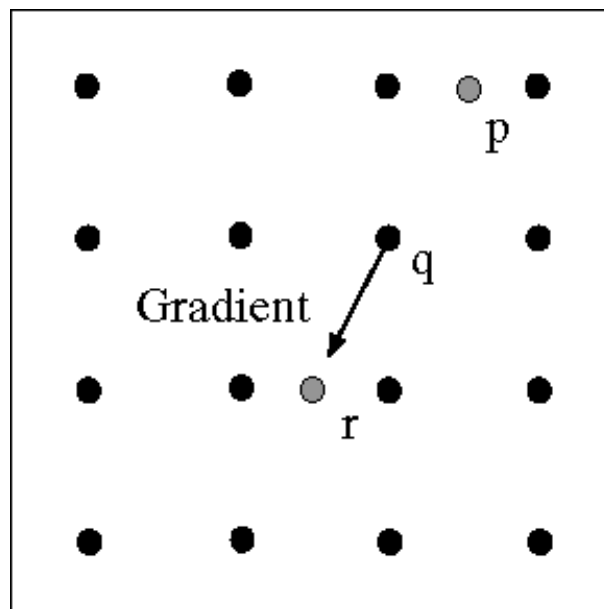
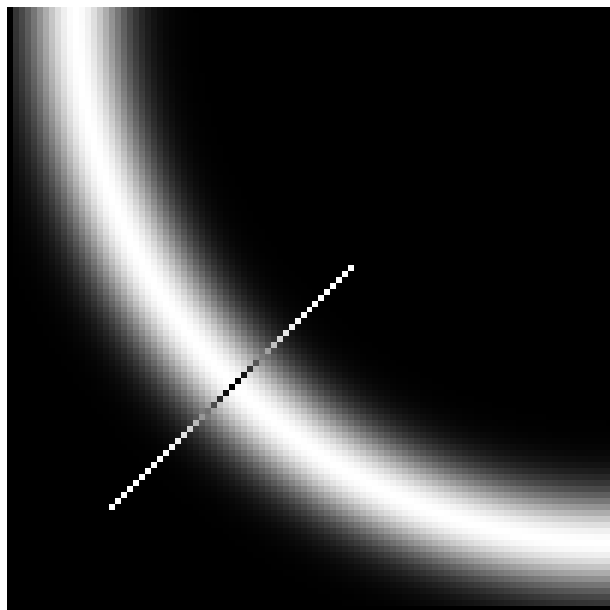
由梯度幅值图像 $M(x, y)$ ，仅保留极大值（严格地说，保留梯度方向上的极大值点）。

具体过程：

1. 初始化 $N(x, y) = M(x, y)$;
2. 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。

边缘检测—Canny算子

非极大值抑制（NMS: Non-Maxima Suppression）



- 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；

边缘检测—Canny算子

对NMS结果进行阈值二值化：

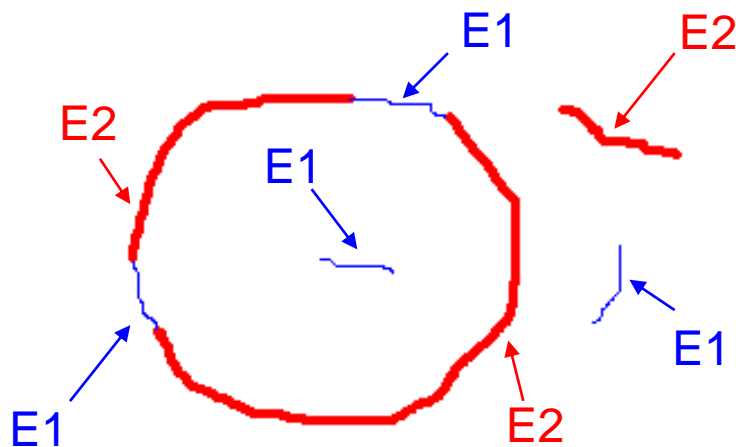
- 使用大的阈值，得到：
 - 少量的边缘点
 - 许多空隙
- 使用小的阈值，得到：
 - 大量的边缘点
 - 大量的错误检测

边缘检测—Canny算子

对NMS结果进行阈值二值化—双阈值检测：

两个阈值 $T1$ ， $T2$ ： $T2 \gg T1$

- 由 $T1$ 得到 $E1(x, y)$ ，低阈值边缘图：更大的误检测率
- 由 $T2$ 得到 $E2(x, y)$ ，高阈值边缘图：更加可靠

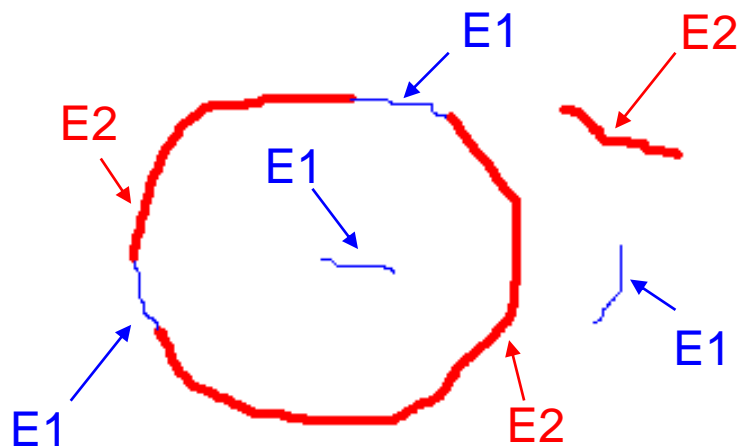


边缘检测—Canny算子

对NMS结果进行阈值二值化—双阈值检测：

边缘连接

1. 将 $E2(x, y)$ 中相连的边缘点输出为一幅边缘图像 $E(x, y)$;
2. 对于 $E(x, y)$ 中每条边，从端点出发在 $E1(x, y)$ 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 $E1(x, y)$ 中没有它延长的部分；
3. 将 $E(x, y)$ 作为结果输出。



边缘检测—Canny算子



原始图像



原始图像经过Gauss平滑

边缘检测—Canny算子



梯度幅值图像



梯度幅值经过非极大值抑制

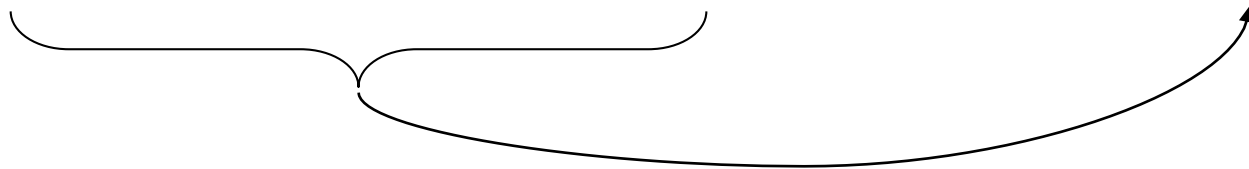
边缘检测—Canny算子



低阈值边缘图像

高阈值边缘图像

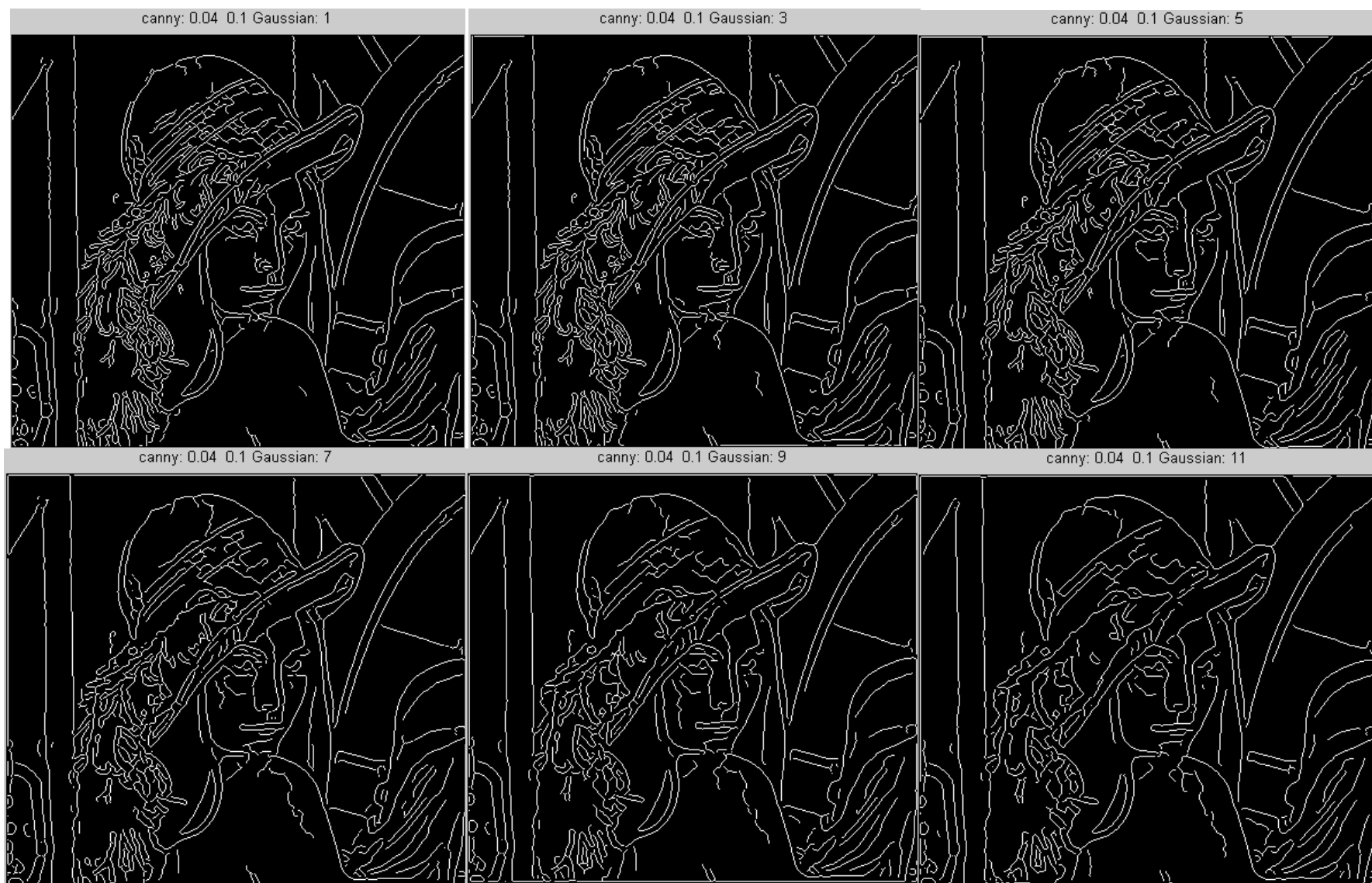
Canny输出边缘图像



边缘检测—Canny算子

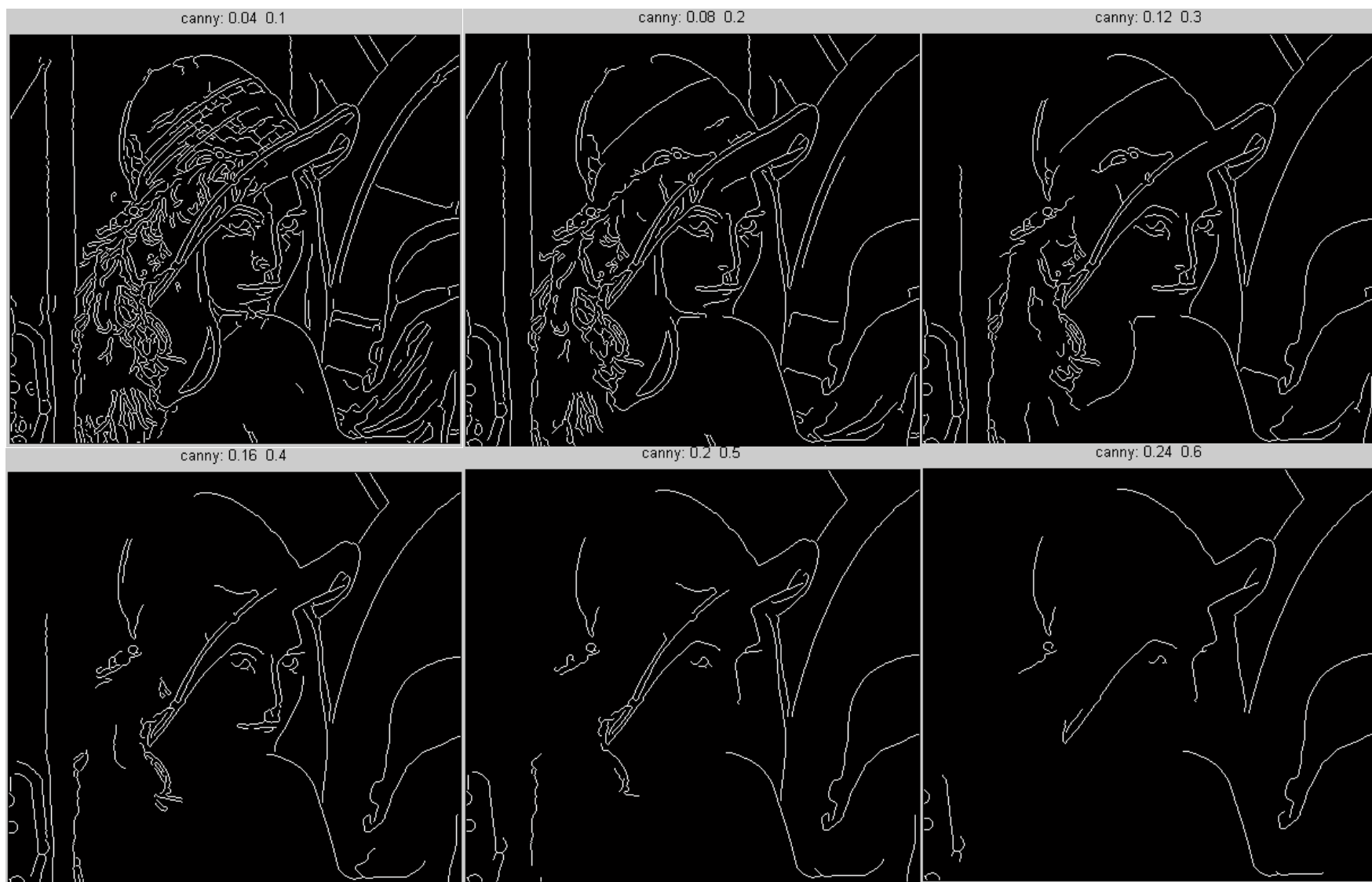
- Canny算子的优点：
 - 参数较少
 - 计算效率
 - 得到的边缘连续完整
- 参数的选择：
 - Gauss滤波的尺度
 - 双阈值的选择($LOW=HIGH*0.4$)

边缘检测—Canny算子



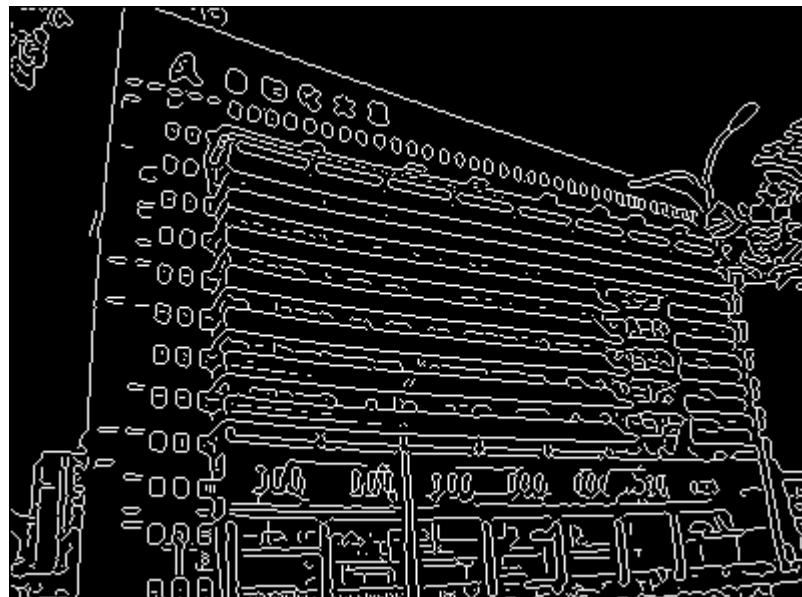
渐增高斯滤波模版的尺寸

边缘检测—Canny算子

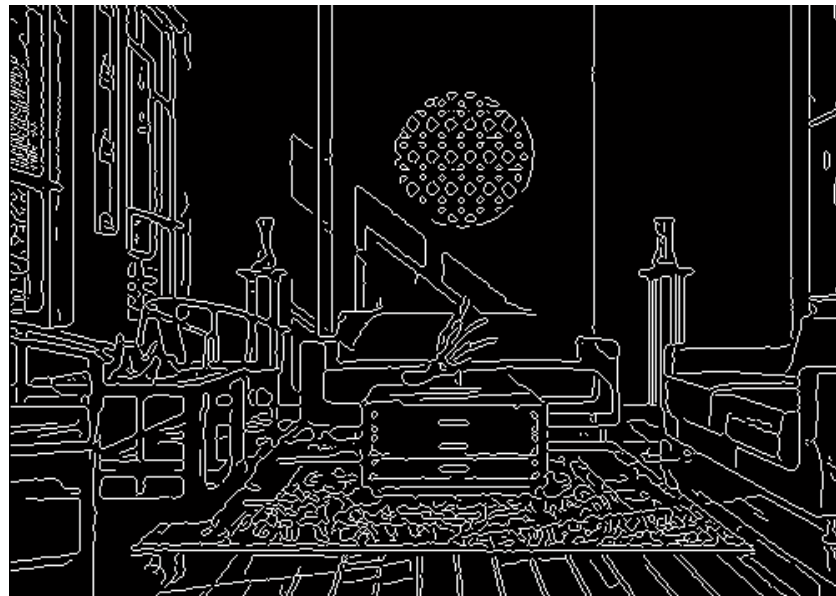
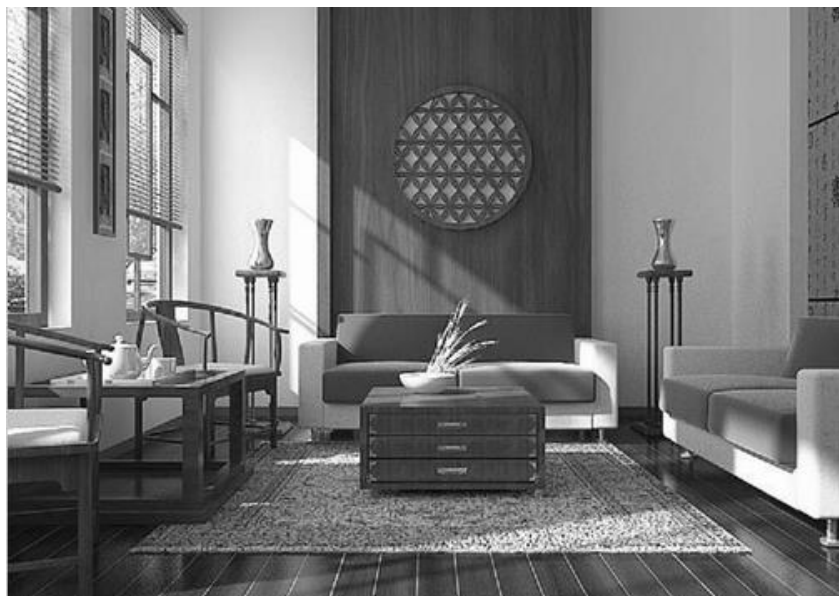


渐增双阈值的大小，保持 $\text{low} = \text{high} * 0.4$

边缘检测—Canny算子



边缘检测—Canny算子



总结

本节课内容回顾：

- 根据自相关矩阵特征值检测角点(Harris);
- 高斯差分尺度空间中检测尺度不变特征点(SIFT);
- 一阶高斯微分算子(Prewitt、Sobel)极值检测边缘;
- 二阶高斯微分算子(LoG)过零点检测边缘;
- 非极大值抑制+双阈值检测边缘(Canny)。

总结

特征检测参考文献:

- Harris, C. and Stephens, M. J. **A combined corner and edge detector**. In *Alvey Vision Conference*, 1988.
- Lowe, D. G. **Distinctive image features from scale-invariant keypoints**. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- Canny, J. **A computational approach to edge detection**. *IEEE Transactions on PAMI*, 8(6):679–698, 1986.
- Tuytelaars, T. and Mikolajczyk, K. **Local Invariant Feature Detectors: A Survey**. *Foundations and Trends in Computer Graphics and Vision*, 3(3): 177–280, 2007.