计算机视觉
Computer Vision

-- Spatial Filtering

钟 凡

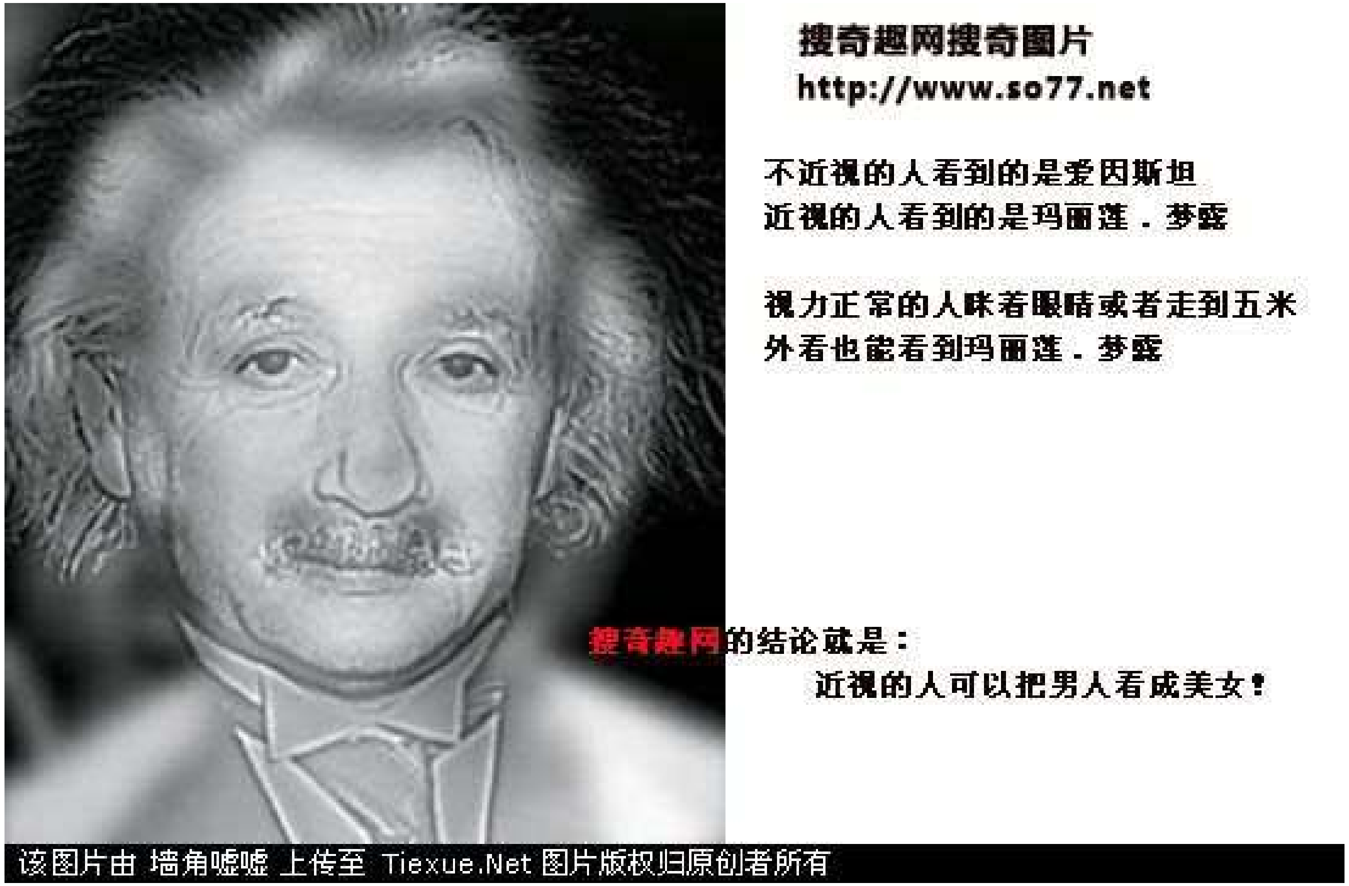
zhongfan@sdu.edu.cn

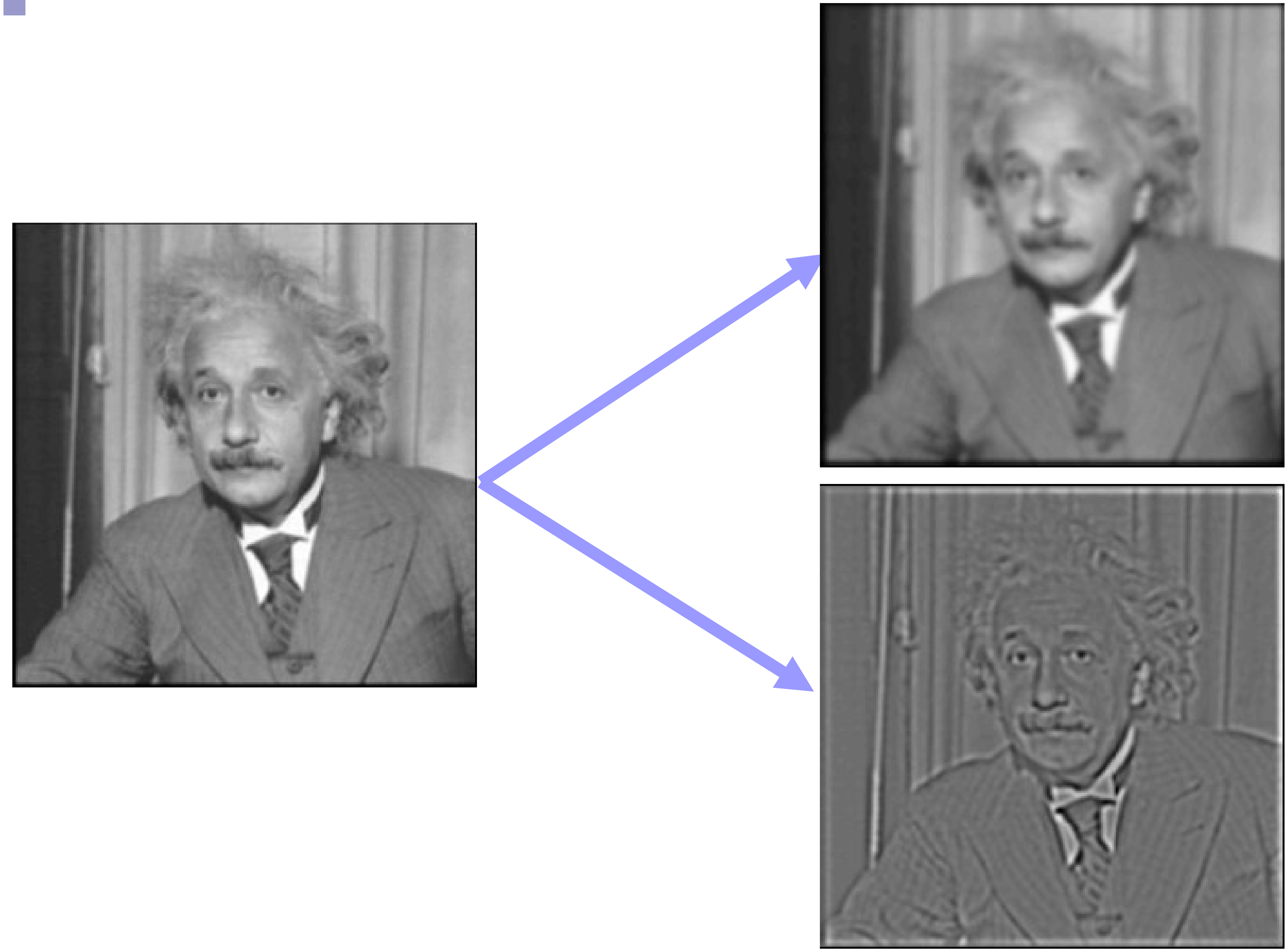
Filter (过滤/过滤器/滤波器）

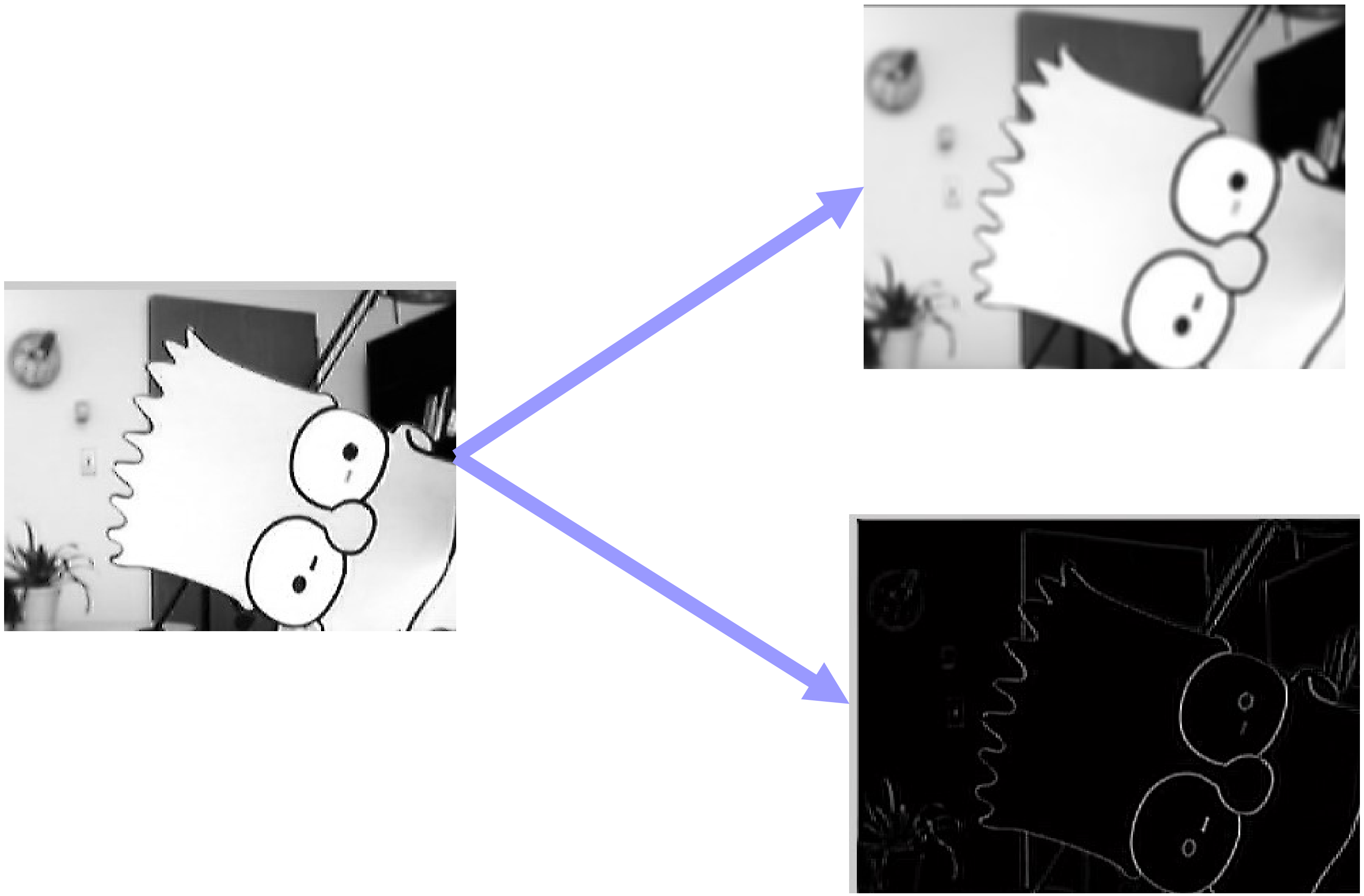
- 从混合物中提取某特定的成份

图像的成份

- 不同尺度（频率）的内容，以及噪音，角点，边缘等





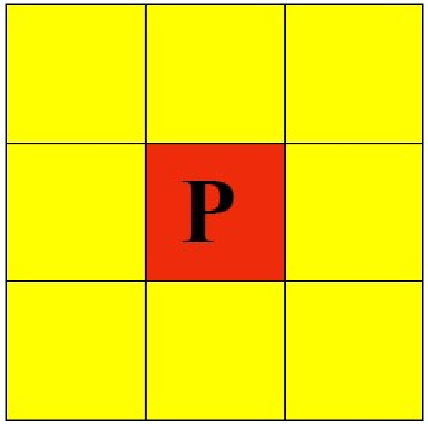


空间域滤波器 (Spatial Filter)

- **对任意像素 *p***

$$p' = f(\mathcal{N}_p)$$

$\mathcal{N}_p$ **为像素 *p* 的某个邻域像素集合**



$$\mathcal{N}_p^{3\times 3} = \{p_0, p_1, \ldots, p_8\}$$

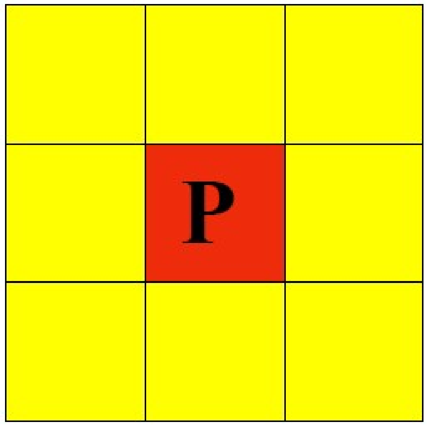
空间域滤波器 (Spatial Filter)

- **均值滤波器：*f=average***

$$p' = \frac{1}{9}(p_0 + p_1 + p_2 + \ldots + p_8)$$

- **最大值滤波器：*f=max***

$$p' = \max\{p_0, p_1, p_2 \ldots p_8\}$$



线性滤波器 (Linear Filter)

- 邻域像素的加权平均

$$p' = \sum_i w_i p_i$$

$$\Downarrow$$

$$p'(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, p(x+s, y+t)$$

滤波核

- **kernel / mask / template**

$$p'(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)\,p(x+s,y+t)$$

1/9 X

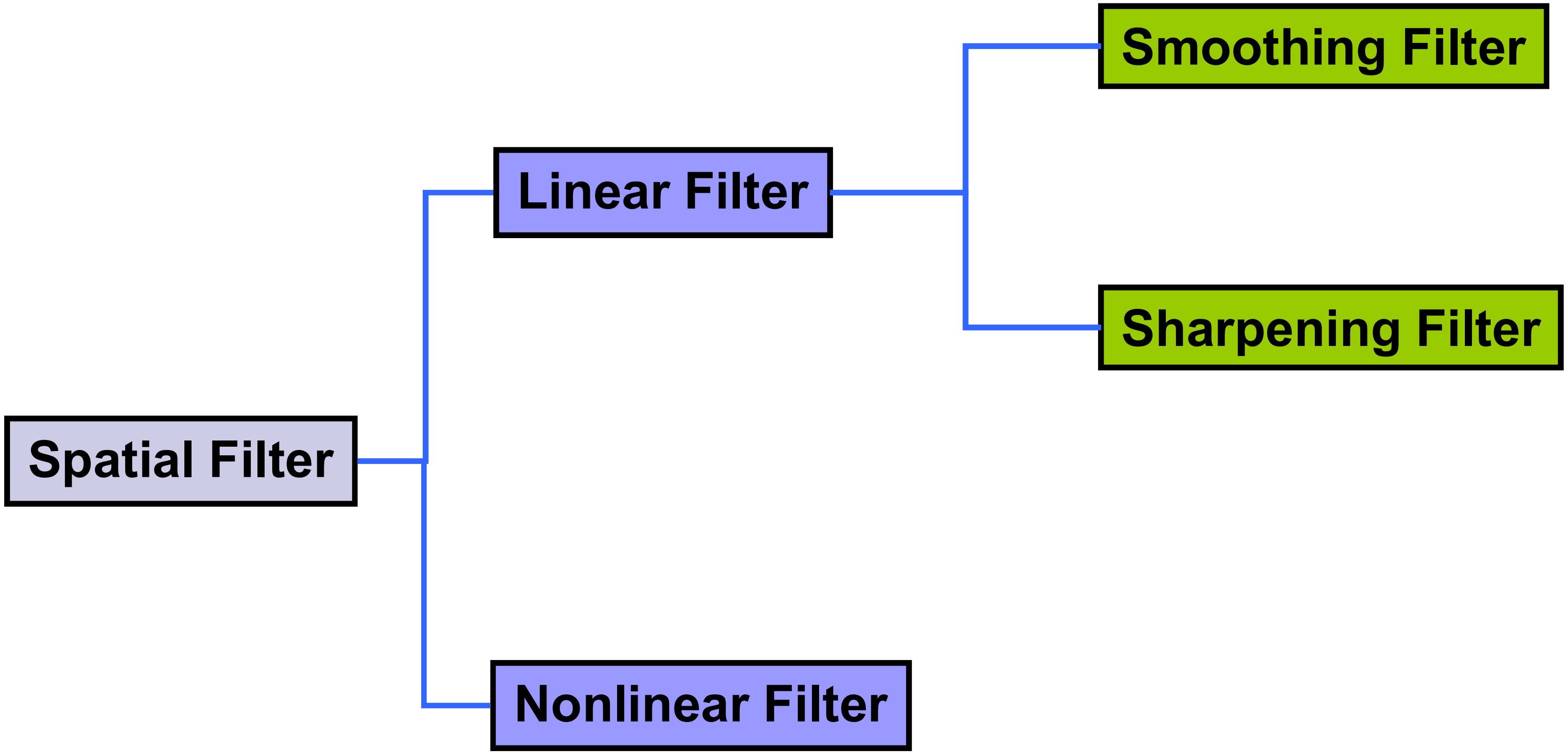
1	1	1
1	1	1
1	1	1

1/17 X

0.5	0.5	0.5	0.5	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	0.5	0.5	0.5	0.5

非线性滤波器（Nonlinear Filters）

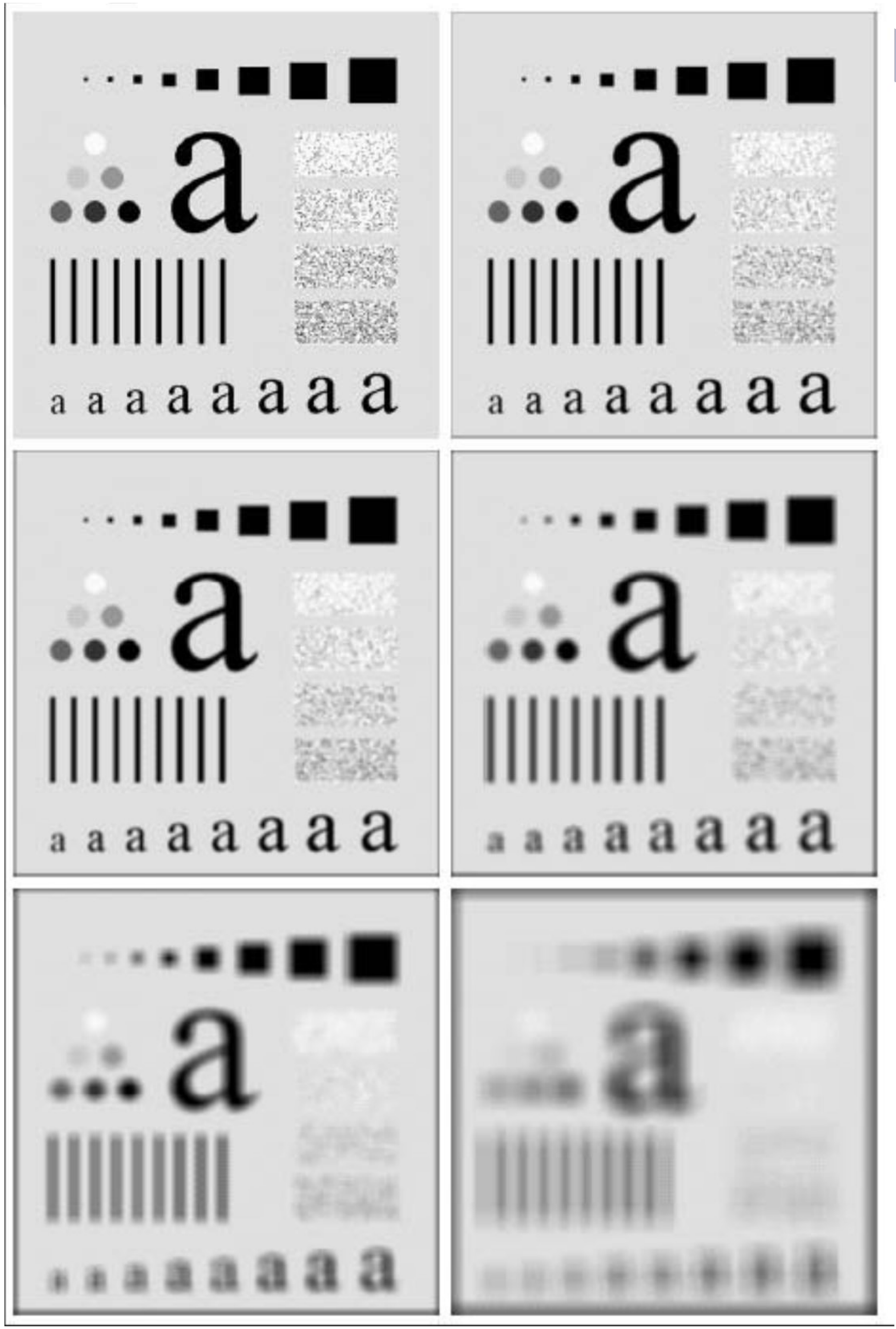
- ***f*** **is nonlinear functions…**
- **Median filter**
- **Max filter**
- **Min filter**


Spatial Filter
Linear Filter
Nonlinear Filter
Smoothing Filter
Sharpening Filter

Smoothing Filter

- Low-pass (低通) filtering
  - Neighborhood averaging
  - Weighted average

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Filter mask sizes:

3x3, 5x5, 9x9, 15x15, 35x35.

Mean Filtering

11x11 Mask

Gauss Filtering

11x11 Mask

1/9 X

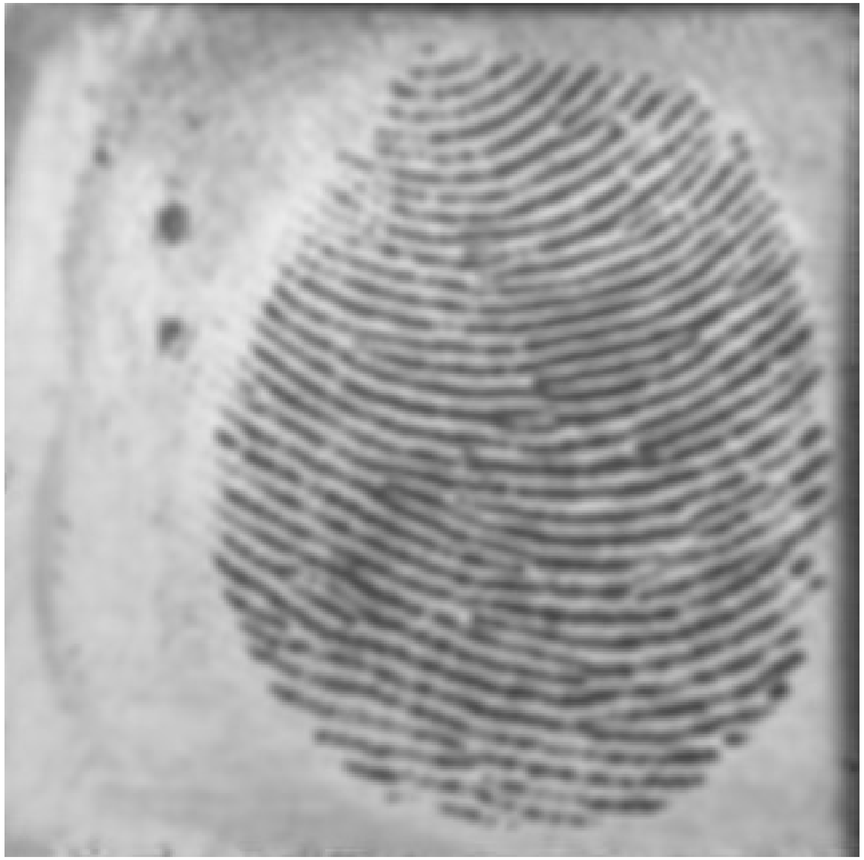
1	1	1
1	1	1
1	1	1

1/16 X

1	2	1
2	4	2
1	2	1

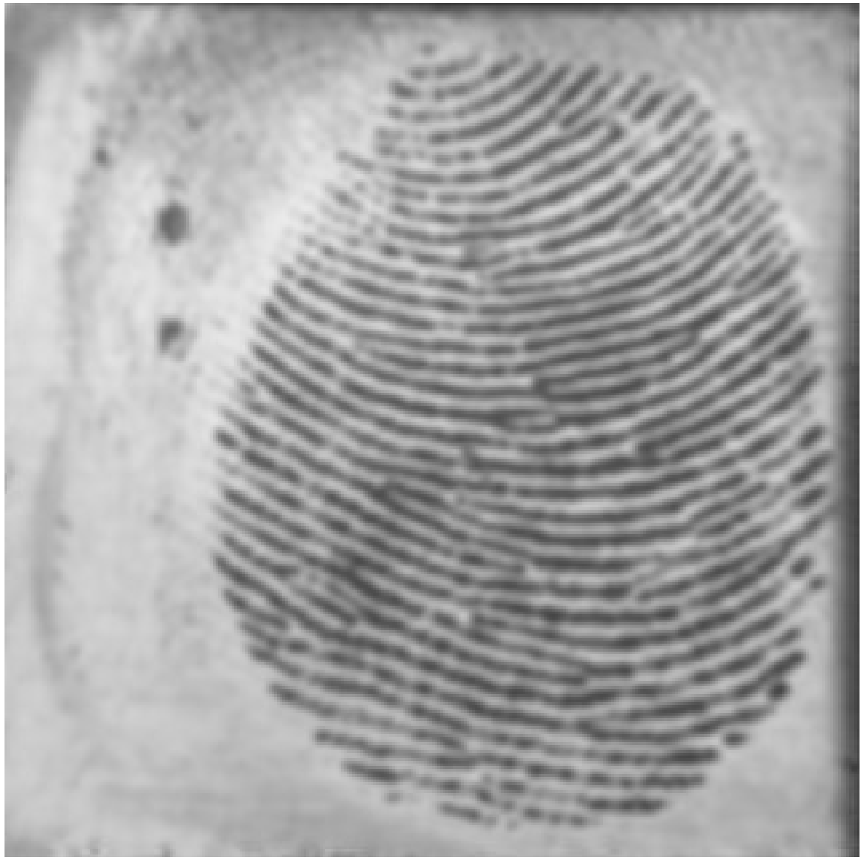
1/25 X

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



1/17 X

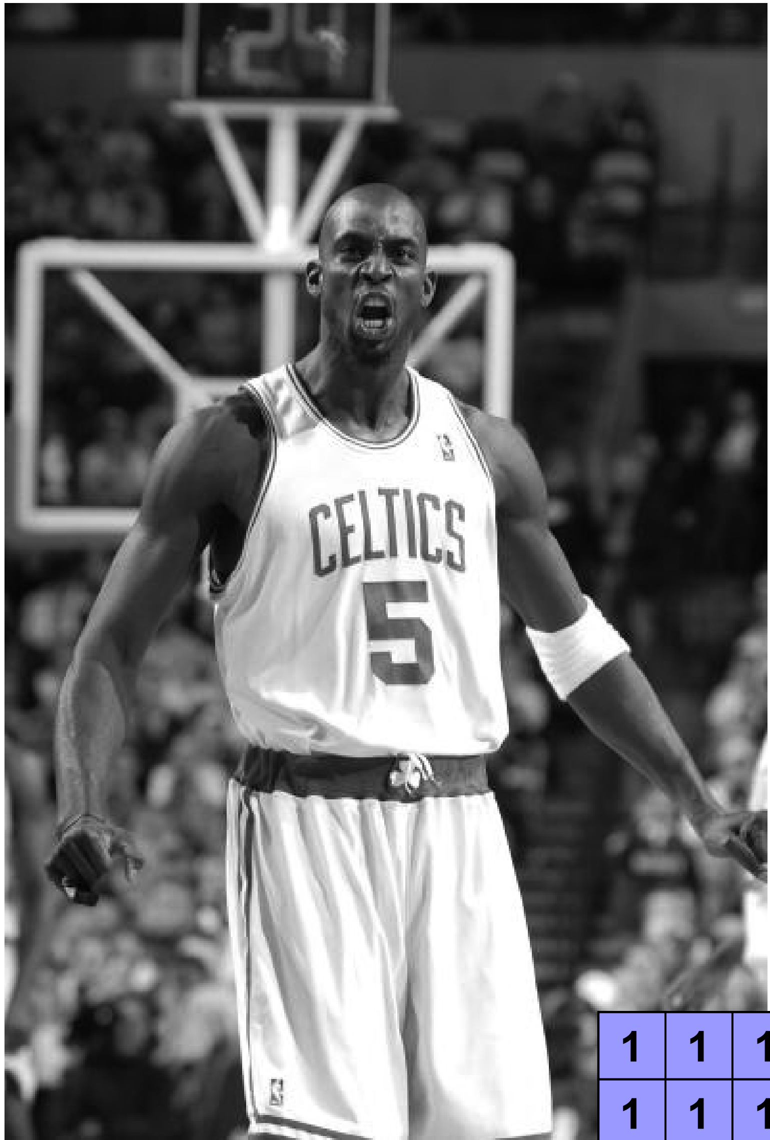
0.5	0.5	0.5	0.5	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	0.5	0.5	0.5	0.5



$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

1/17 X

0.5	0.5	0.5	0.5	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	1	1	1	0.5
0.5	0.5	0.5	0.5	0.5



1/25 X

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

$$1/9\ \text{X} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

1/25 X

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

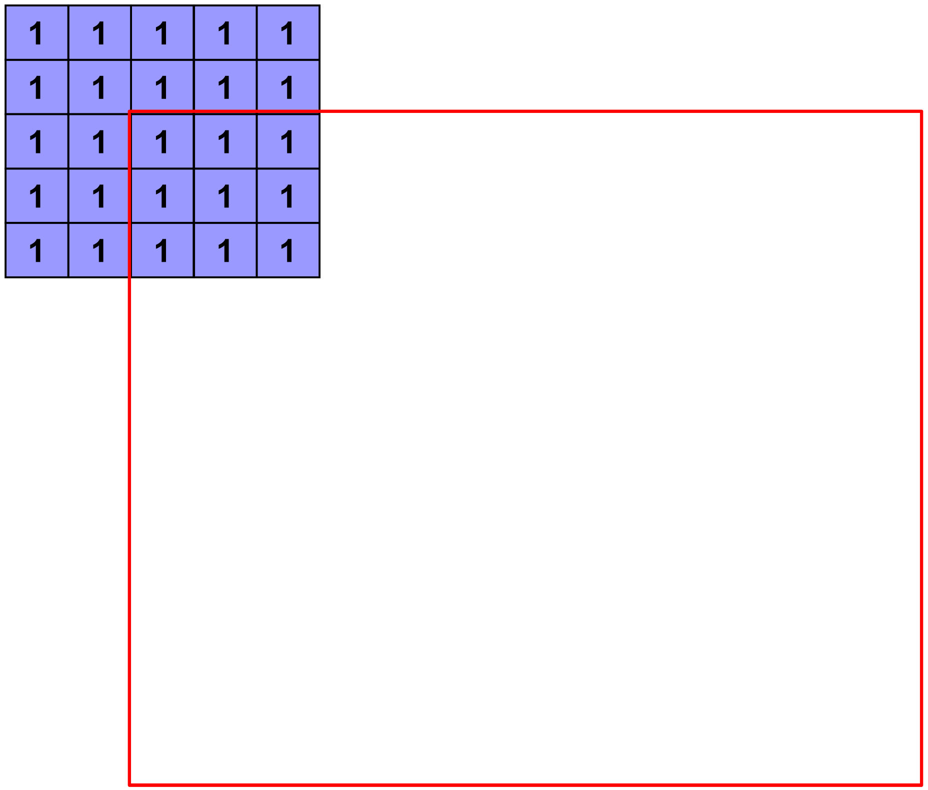
Mask size ↑ , blurring ↑

实现

- 均值滤波 (Mean Filter / Box Filter)
- 高斯滤波 (Gaussian Filter)

边界处理

- 对边界附近的像素，滤波核的部分可能落在图像区域外。



边界处理

- 对边界外某个范围内的区域进行填充
  - 常数填充
  - 镜像填充（以边界为轴，取图像内对称点的像素值填充）

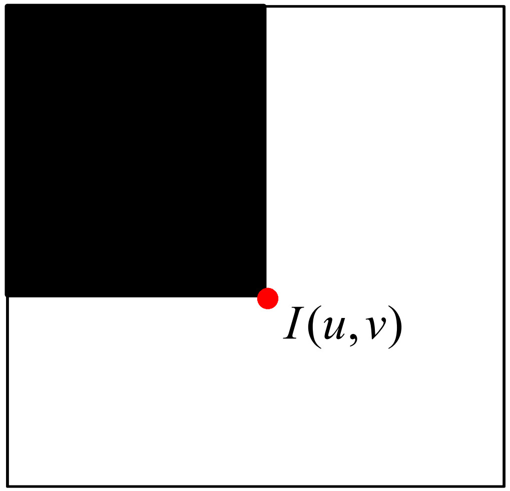
- 在边界附近调整滤波核的大小

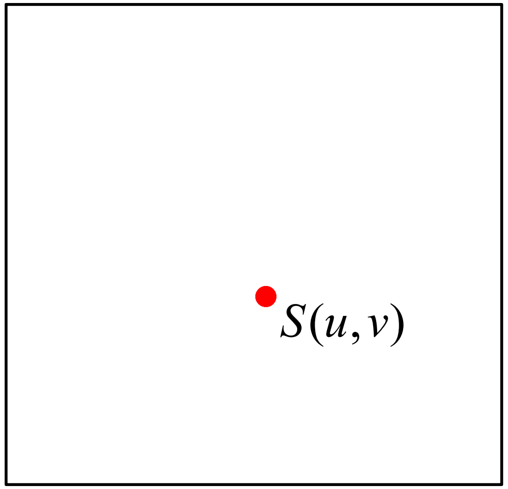
快速均值滤波

- ….

积分图

- 图像I的积分图S是与其大小相同的图像，S的每一像素S(u,v)存贮的是I(u,v)左上角所有像素的颜色值之和。





- 积分图可增量计算，只需对原图进行一遍扫描：

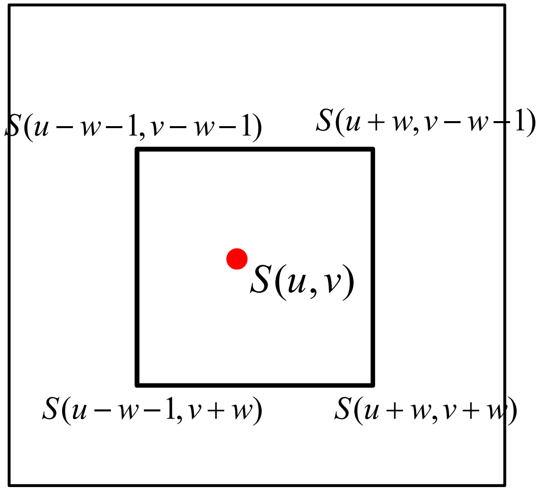
$$S(u,v) = S(u,v-1) + sum(I[1:u,v])$$

```
void  integral_image(const  uchar *src, int width, int height, int sstride, int *pint, int istride)
{
        int *prow=new int[width];

        memset(prow,0,sizeof(int )*width);

        for(int yi=0; yi<height; ++yi, src+=sstride, pint+=istride)
        {
            prow[0]+=src[0];  pint[0]=prow[0];   //for the first pixel

           for(int xi=1; xi<width; ++xi)
           {
                  prow[xi]+=src[xi];
                  pint[xi]=pint[xi-1]+prow[xi];
           }
       }

     delete[]prow;
}
```

基于积分图的快速均值滤波

- **设滤波窗口大小为2w+1，滤波结果为图像O，则：**

$$O(u,v)=\frac{1}{Z}[S(u+w,v+w)+S(u-w-1,v-w-1)-S(u+w,v-w-1)-S(u-w-1,v+w)]$$



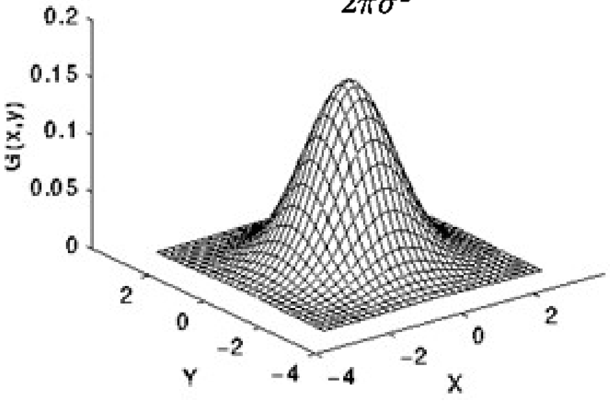
Z=(2w+1)*(2w+1)为像素个数；

中括号内即为滤波窗口覆盖的像素颜色值之和；

高斯滤波

- 高斯滤波=以高斯函数为滤波核

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



二维高斯函数

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

5*5卷积核（注意归一化!）

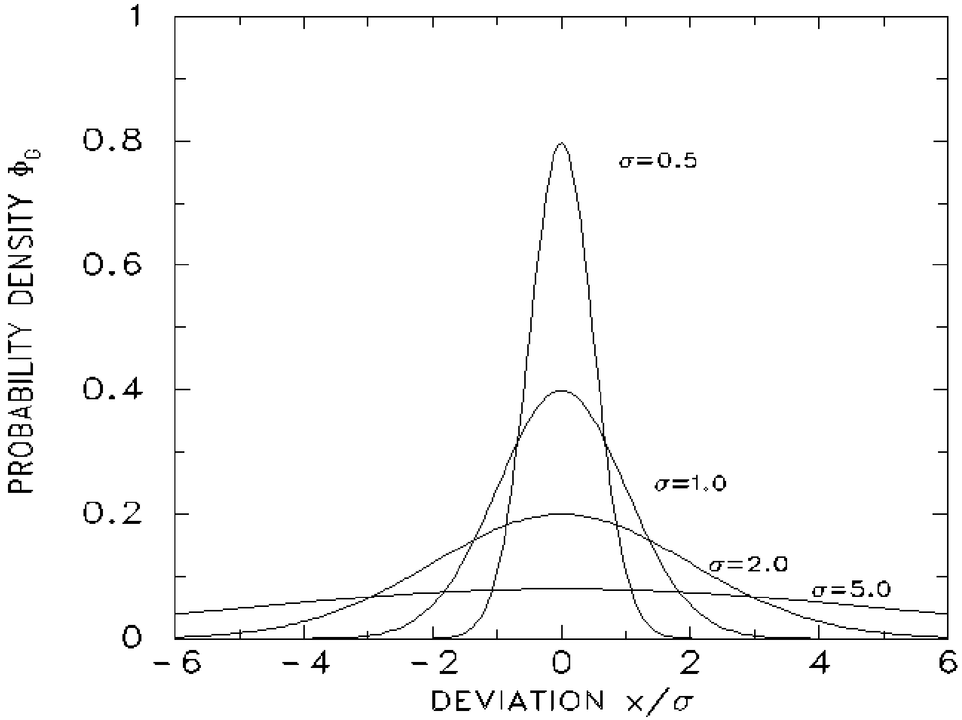
高斯滤波

- **行列可分离性:** 核大小为M的二维图像高斯滤波，等价于同样核大小的一维高斯滤波在行列方向的叠加

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

高斯滤波

- 核大小 与 σ 的关系?
  - σ 越大，核应该越大



$$M = [6\sigma - 1]$$

- **Sharpening Filters**
  - **Basic Sharpening Filter**
  - **Derivative Filter**
  - **High-Boost Filter**

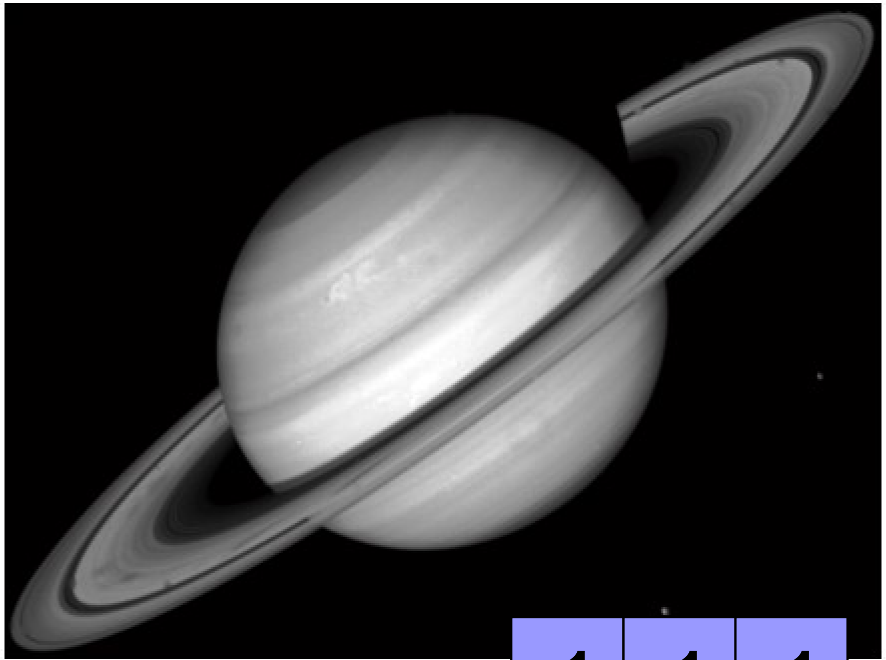
Basic High-Pass Filtering

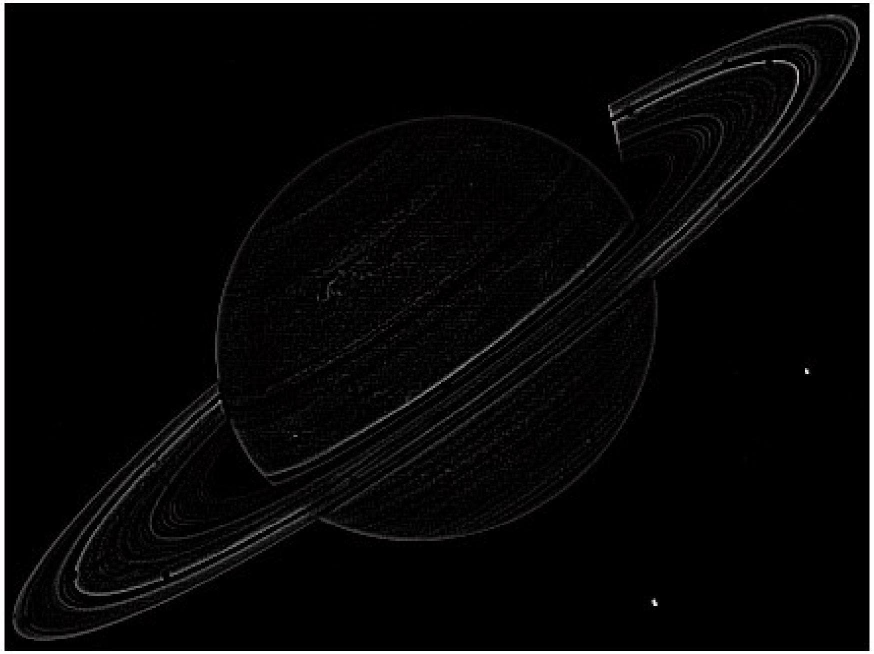
- 模版的特点:
  - 滤波器应该在中心有正系数，而在边缘上有负的系数
  - The filter should have positive coefficients near the center and negative in the outer periphery.
  - 总和为零，sum = 0

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

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

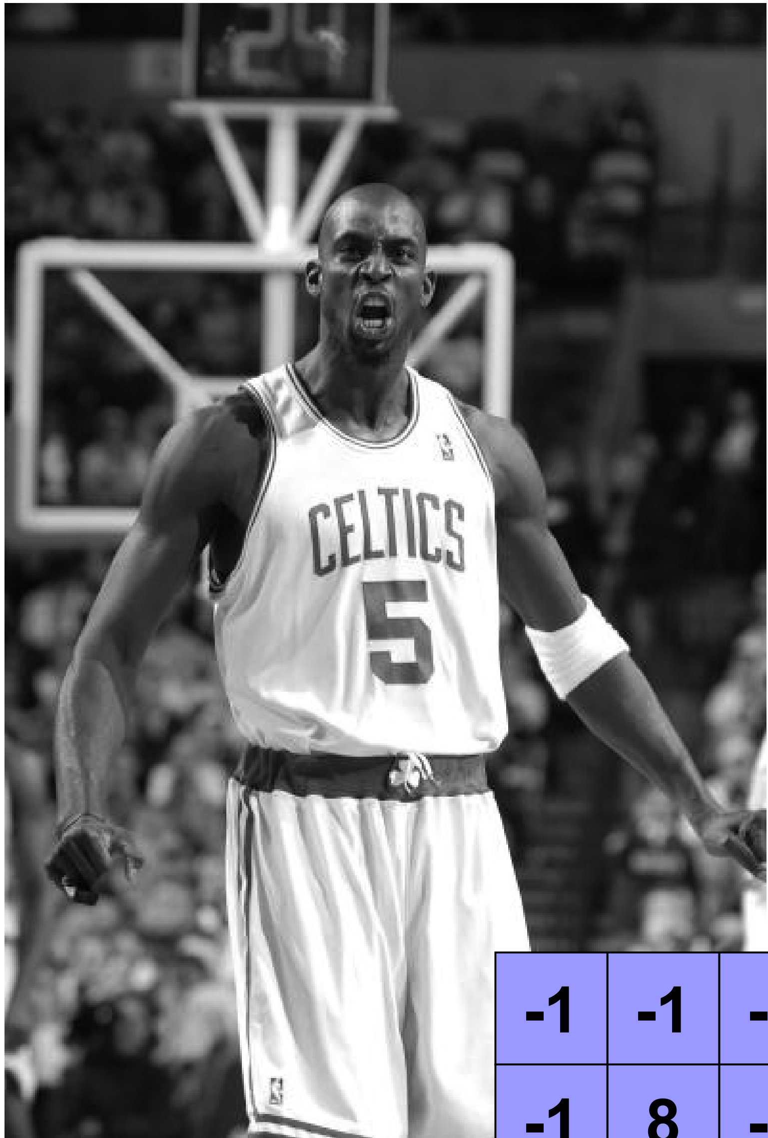
Basic High-Pass Filtering

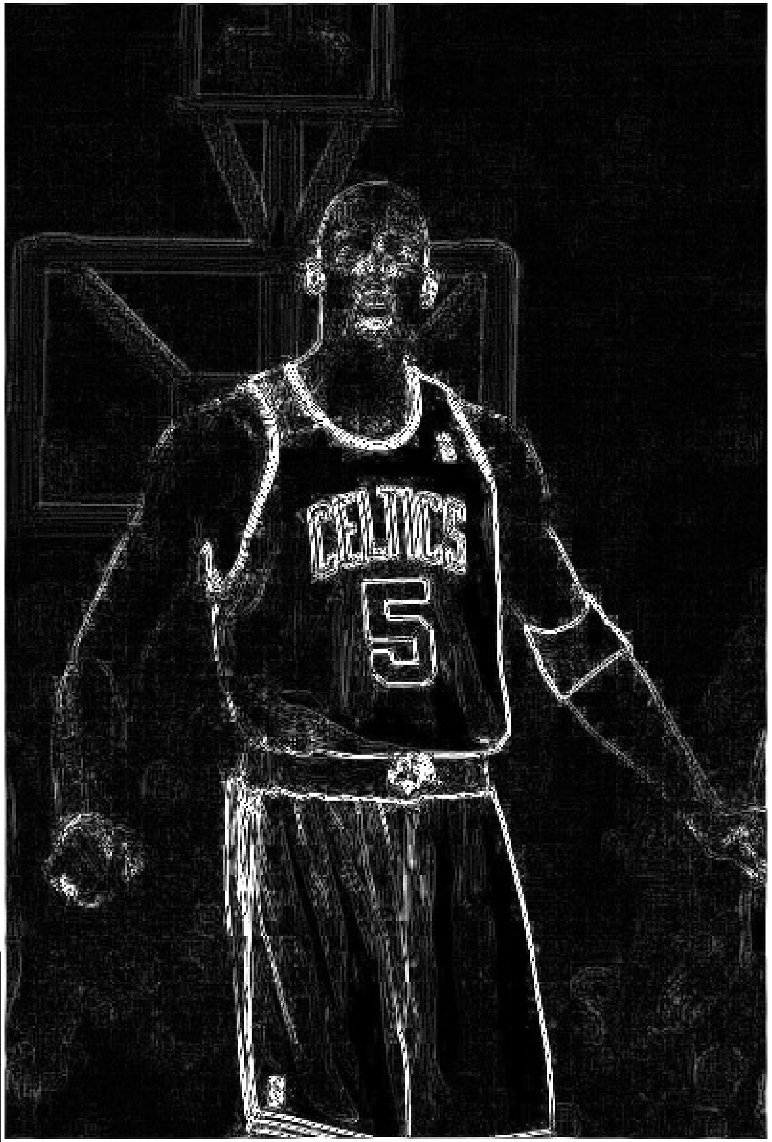




-1	-1	-1
-1	8	-1
-1	-1	-1

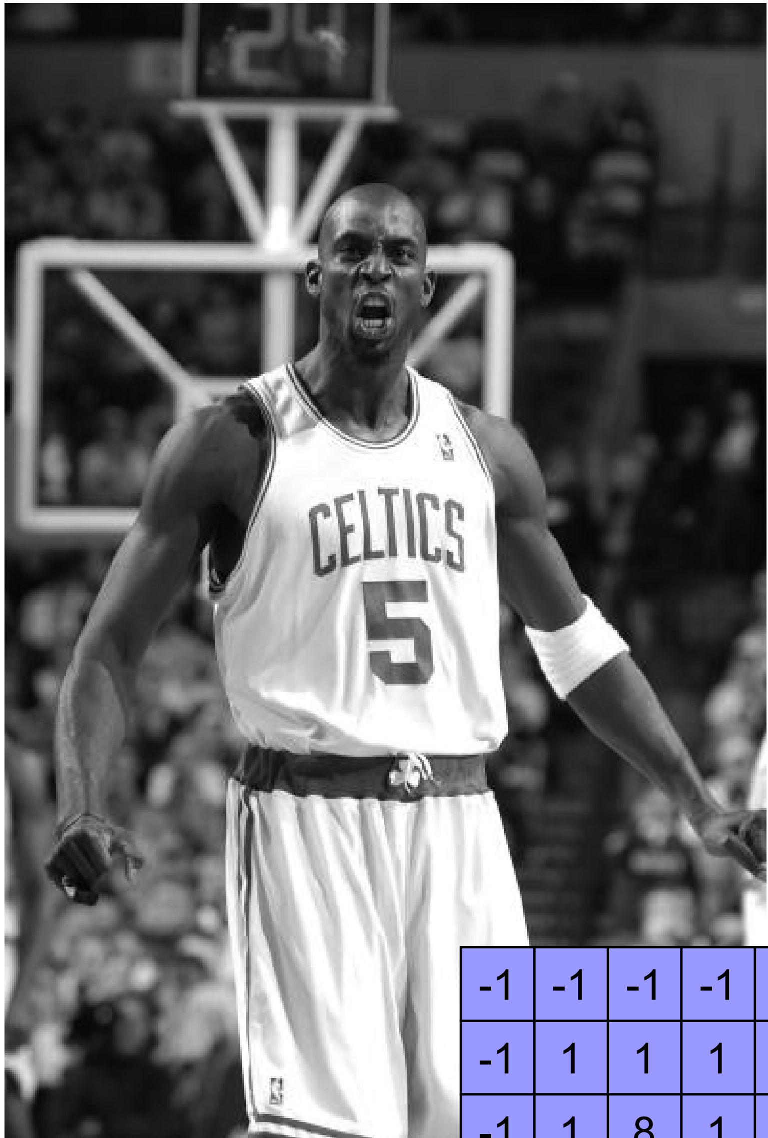
在平坦变化的区域很暗
在剧烈变化的区域很亮

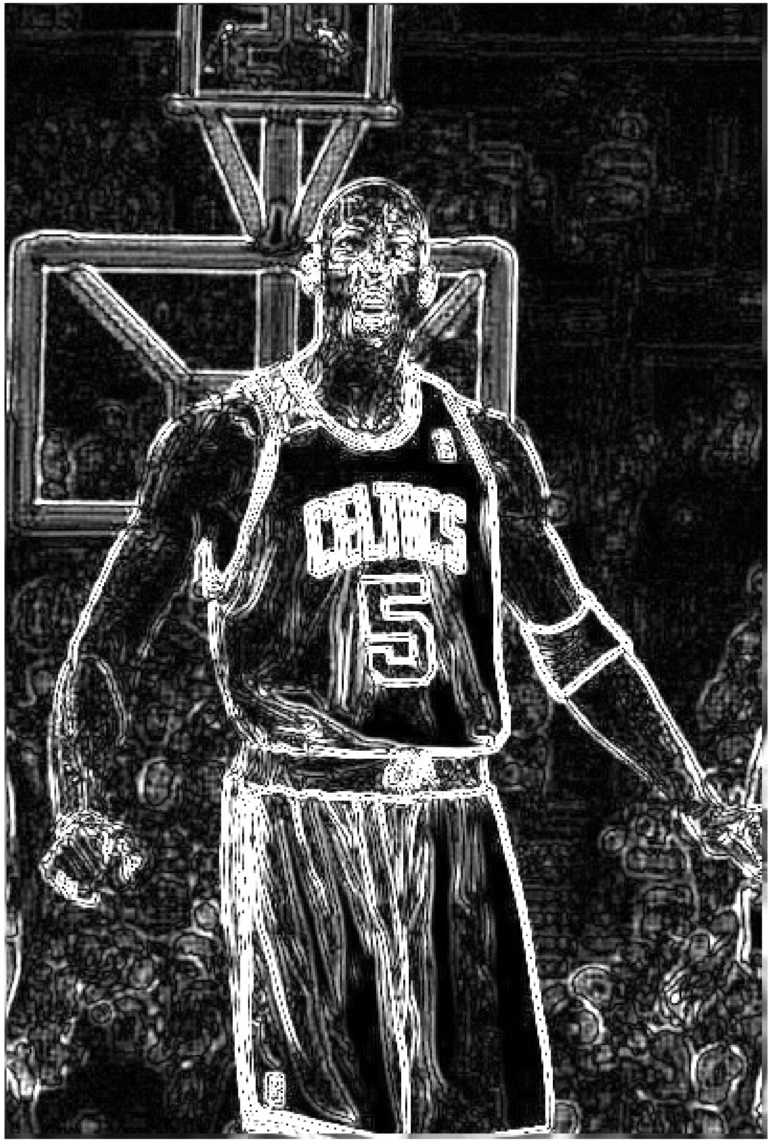




-1	-1	-1
-1	8	-1
-1	-1	-1

采用3x3的模板





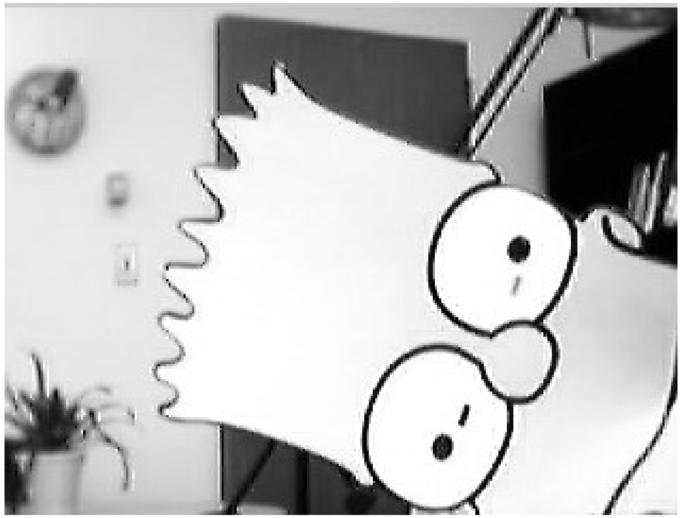
-1	-1	-1	-1	-1
-1	1	1	1	-1
-1	1	8	1	-1
-1	1	1	1	-1
-1	-1	-1	-1	-1

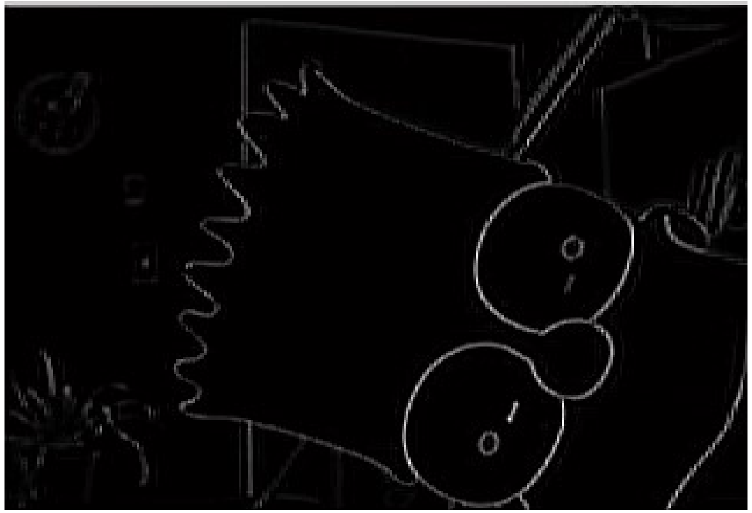
采用5x5的模板，边缘较宽

Basic High-Pass Filtering 基本高通滤波

- 滤波后往往出现负值，或者很暗，因此常常需要对其进行比例变换scaling或者截断clipping
- 体现图像的梯度属性，即显示图像的变化情况
  - 一般的模版是对称的；当模版非对称时，会使图像的边缘具有偏向
  - 由于模版总和为零，在颜色的平坦区域，滤波后的结果会趋于零，使得整幅图像比较暗
  - 由于其突出显示了变化大的区域，因此整幅图像的连贯性降低

Application: Edge Detection 边缘检测





滤波以后，仅有边缘的区域是亮的，而平坦区域则接近于零

- **Sharpening Spatial Filters**
  - **Basic High-Pass Spatial Filtering**
  - **Derivative Filters**
  - **High-Boost Filtering**

导数滤波器

- **Derivative Filters**
  - **First Derivatives: The Gradient** 一阶导：梯度
    - **Roberts Operators**
    - **Prewitt Operators**
    - **Sobel Operators**
  - **Second Derivatives: The Laplacian** 二阶导：拉普拉斯

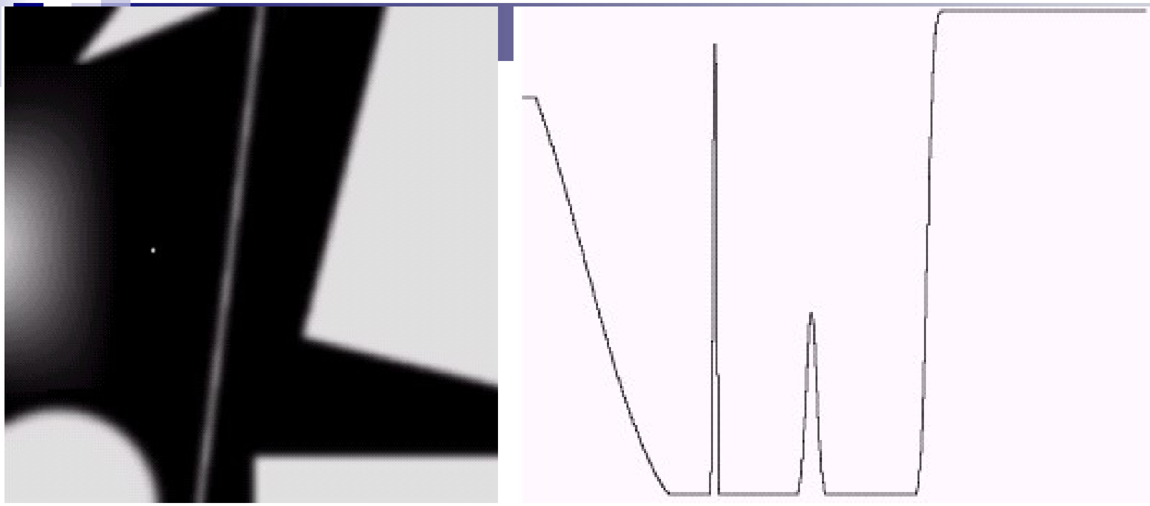
导数滤波器

- 图像平均类似于积分，导致图像模糊
- 图像差分则可能有相反的效果，导致图像锐化
- 一阶导

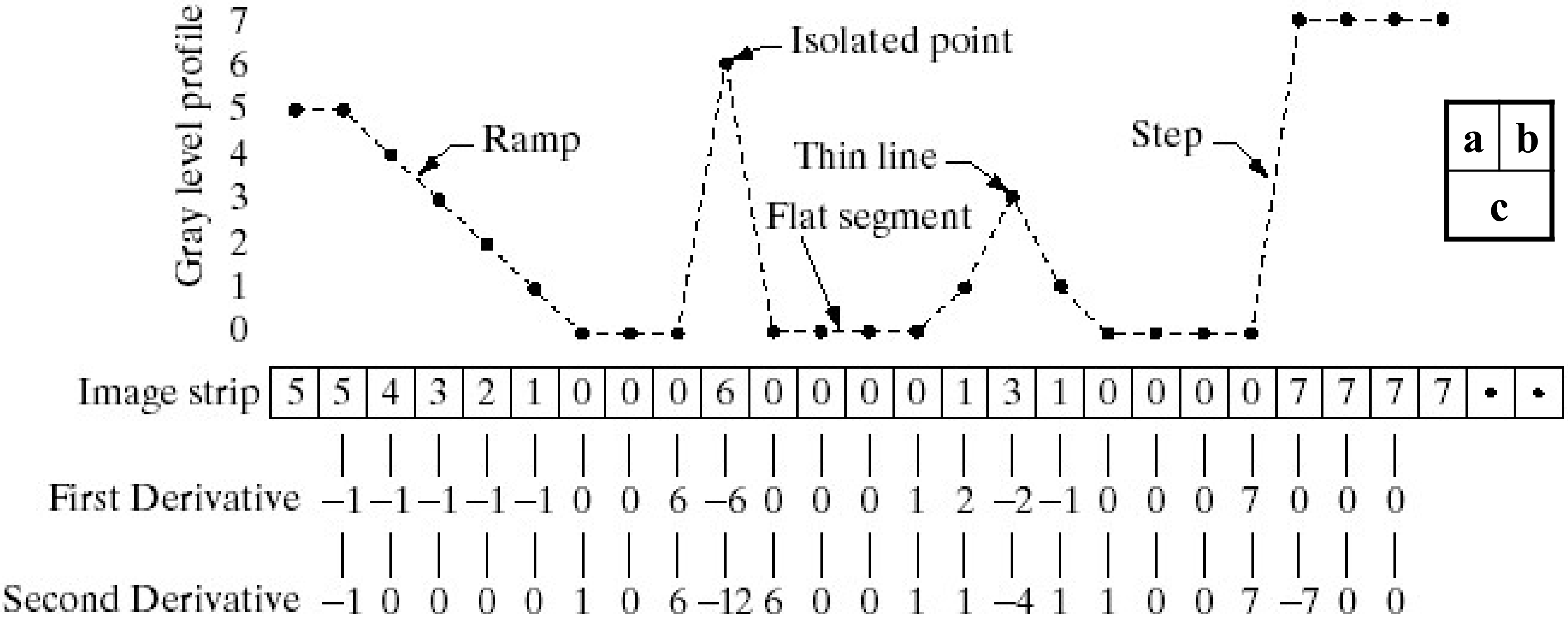
$$f'(x) = f(x+1) - f(x)$$

- 二阶导

$$f''(x) = f(x+1) + f(x-1) - 2f(x)$$



(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to Simplify interpretation).

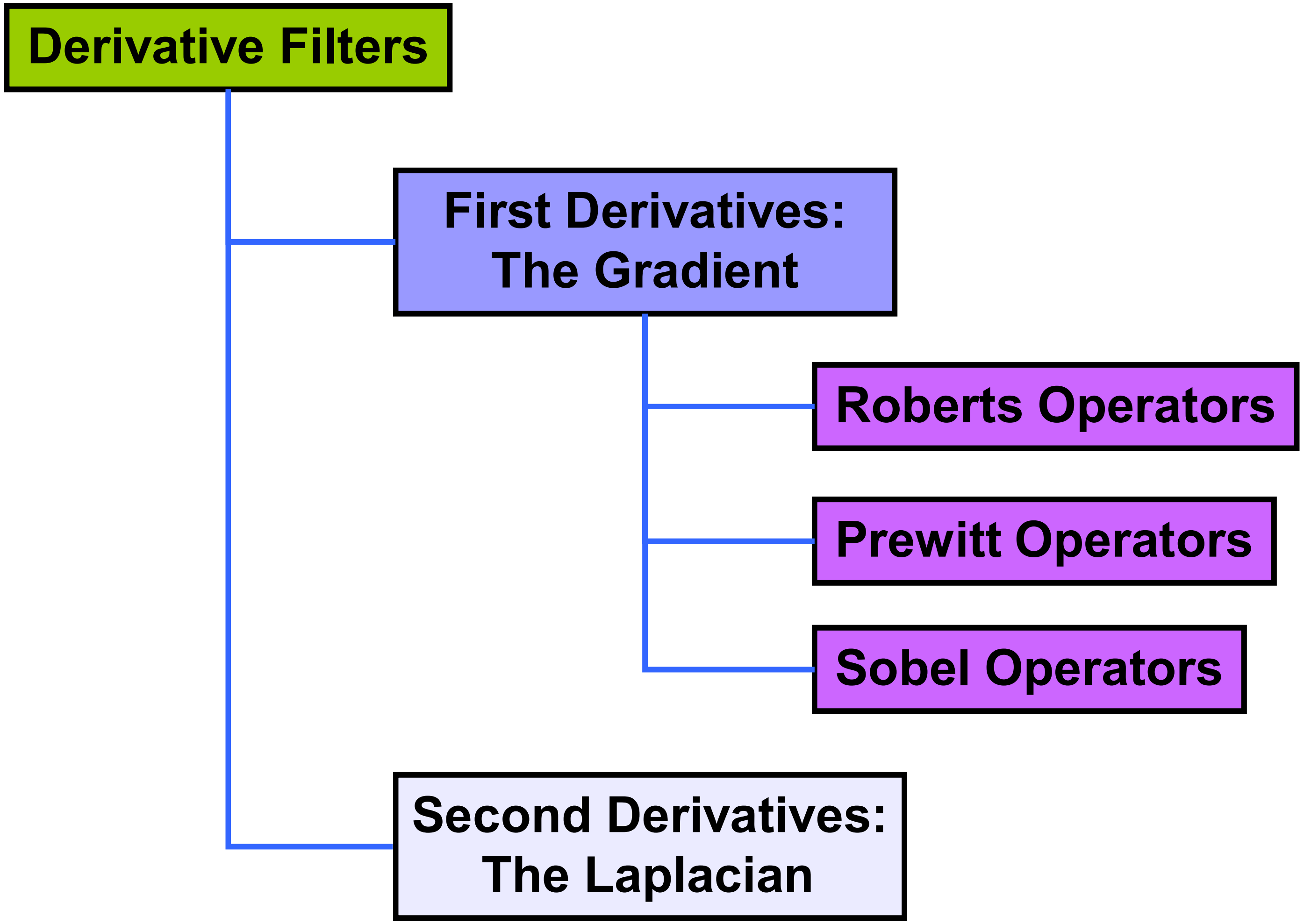


数字函数的导数 Digital Function Derivatives

- 一阶导 First derivative :
  - 零：在常数值区域　0 in constant gray segments
  - 非零：在斜坡的开始　Non-zero at the begin of steps or ramps
  - 非零: 在斜坡上　Non-zero along ramps
- 二阶导 Second derivative:
  - 零：在常数值区域　0 in constant gray segments
  - 非零：在斜坡的两端　Non-zero at the begin and end of steps or ramps
  - 零：在斜坡上　0 along ramps of constant slope

Observations

- 一阶导产生的边缘可以有较大的宽度，如斜坡具有持续的亮度，这对边缘位置的确定是不利的，因为在寻找边缘的时候，我们希望确定准确的边缘位置
- 二阶导在边缘处过零点，这是确定边缘的简单判别方式
- 二阶导对噪声敏感，因此一般的图像处理中的二阶导施行之前，需要对图像进行高斯滤波，先行去除噪声


Derivative Filters
First Derivatives: The Gradient
Roberts Operators
Prewitt Operators
Sobel Operators
Second Derivatives: The Laplacian

Use of First Derivatives - The Gradient

- 图像处理中最为常见的差分方法即梯度
- The most common method of differentiation in Image Processing is the gradient:

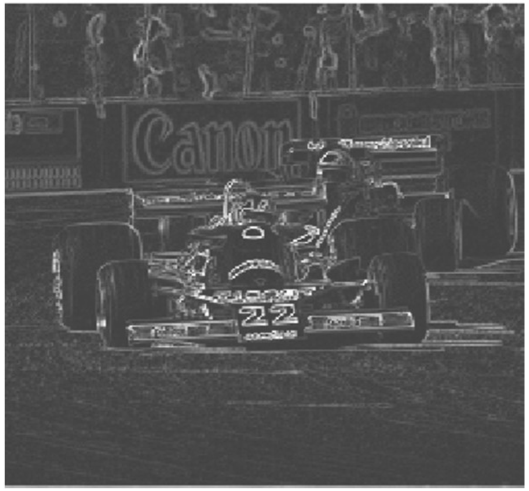
$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \text{at (x,y)}$$

梯度是一个向量，指向灰度变化最大的方向，其长度为:

$$mag(\nabla f) = [G_x^2 + G_y^2]^{\frac{1}{2}} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

梯度

Original image


Gradient magnitude

梯度

- 简约的计算公式 Computations:

$$\nabla f \approx |G_x| + |G_y|$$

- 三种典型的梯度算子
  - Roberts Operators
  - Prewitt Operators
  - Sobel Operators

Roberts Operators

- **Roberts算子的定义:**

$$G_x = (z_9 - z_5)$$

$$G_y = (z_8 - z_6)$$

- **Roberts梯度的近似计算**
  - **Approximation (Roberts Cross-Gradient Operators):**

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0
0	1

G_x

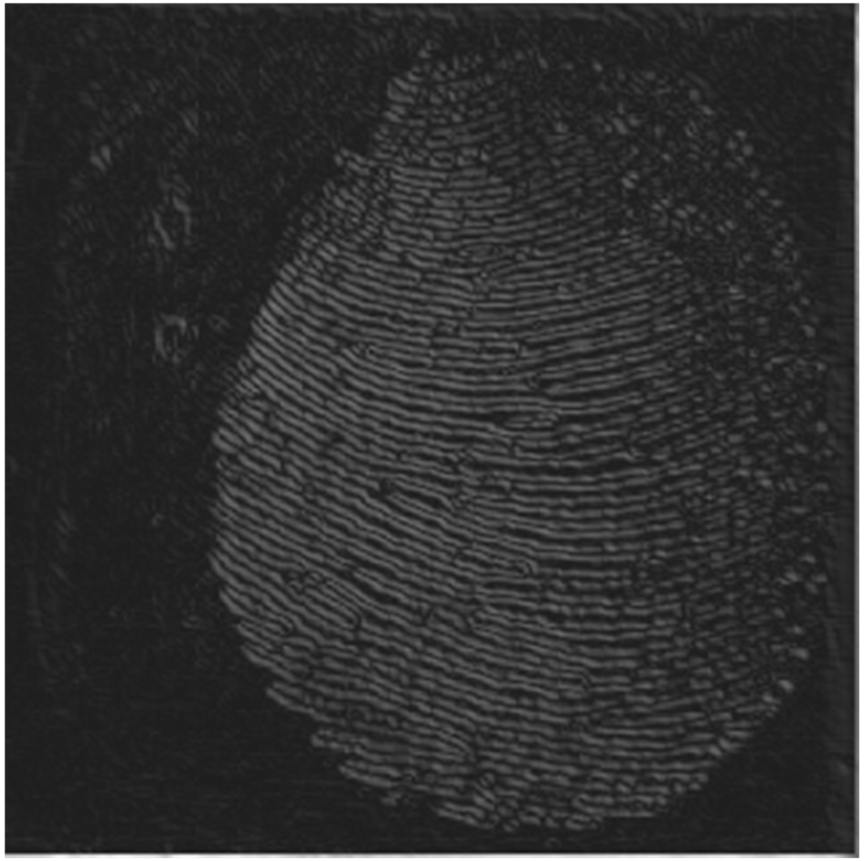
0	-1
1	0

G_y

-1	0
0	1

Right fraction

0	-1
1	0



Left fraction

Prewitt Operators

- **Approximation (Prewitt Cross-Gradient Operators):**

$$\nabla f = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

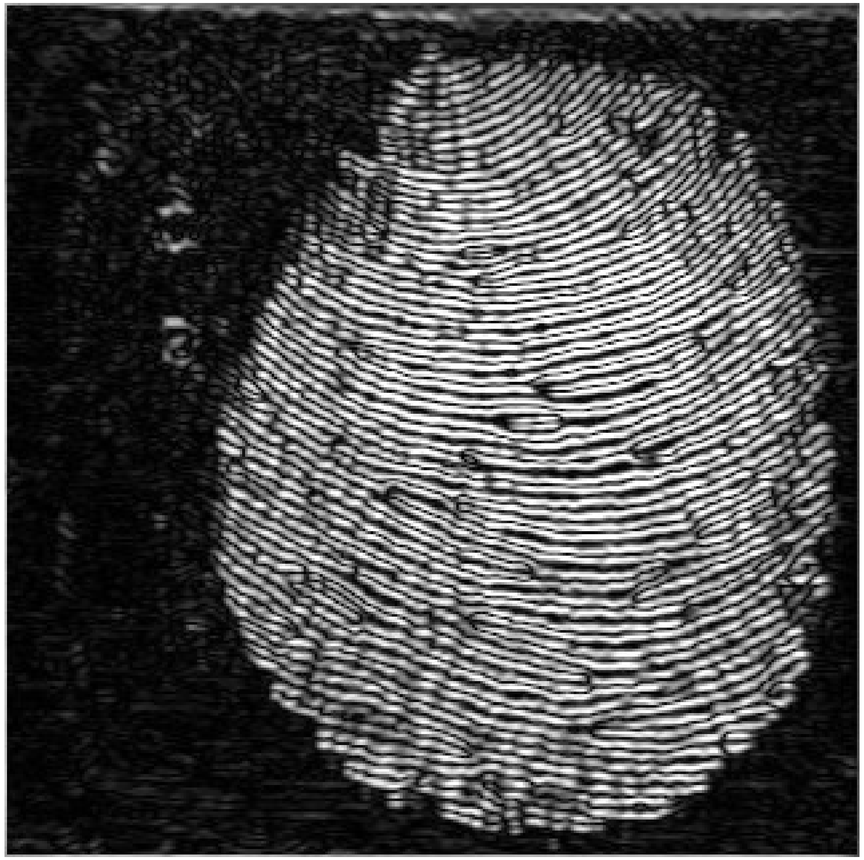
-1	-1	-1
0	0	0
1	1	1

G_y

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

G_x

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



Row fraction

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

Column fraction

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

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

Sobel Operators

- **Approximation (Sobel Cross-Gradient Operators):**

$$\nabla f = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$
$$+ |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

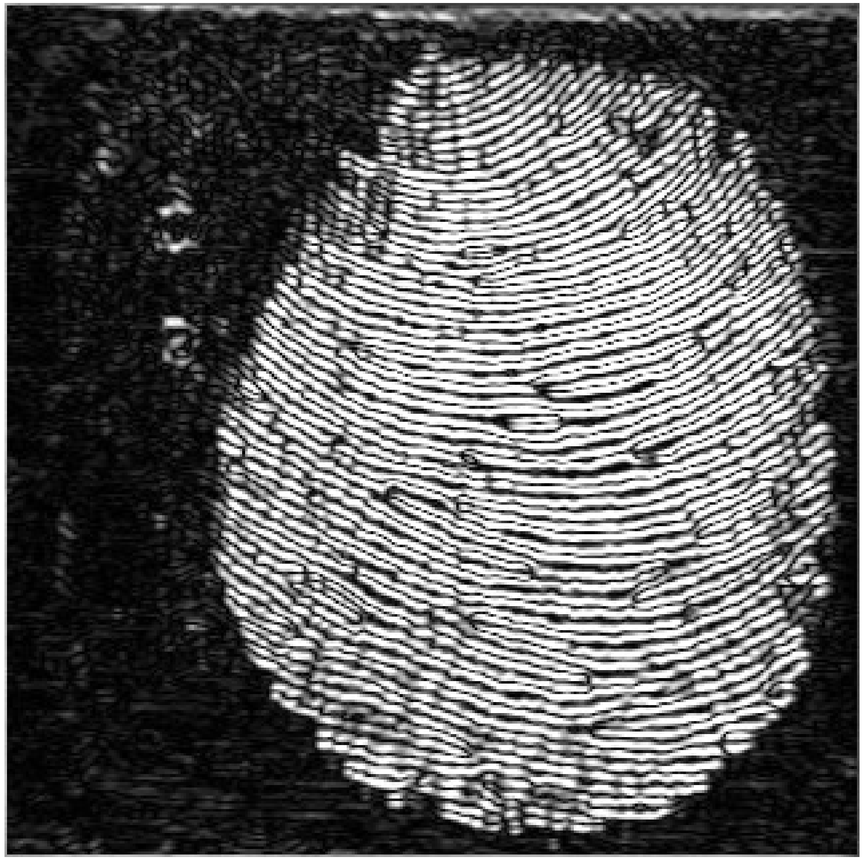
-1	-2	-1
0	0	0
1	2	1

G_x

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

G_y

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



Row fraction

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

Column fraction

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

纵向梯度很弱

横向梯度很弱

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

- Derivative Filters
  - First Derivatives: The Gradient
    - Roberts Operators
    - Prewitt Operators
    - Sobel Operators
  - Second Derivatives: The Laplacian

Use of Second Derivatives - The Laplacian

Original image

Laplacian filtered image

Use of Second Derivatives - The Laplacian

- **Laplacian (linear operator) 定义为x,y方向上的二阶导的和:**

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

- **Discrete version 离散表示:**

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

Laplacian

- **Laplacian的两种定义:**

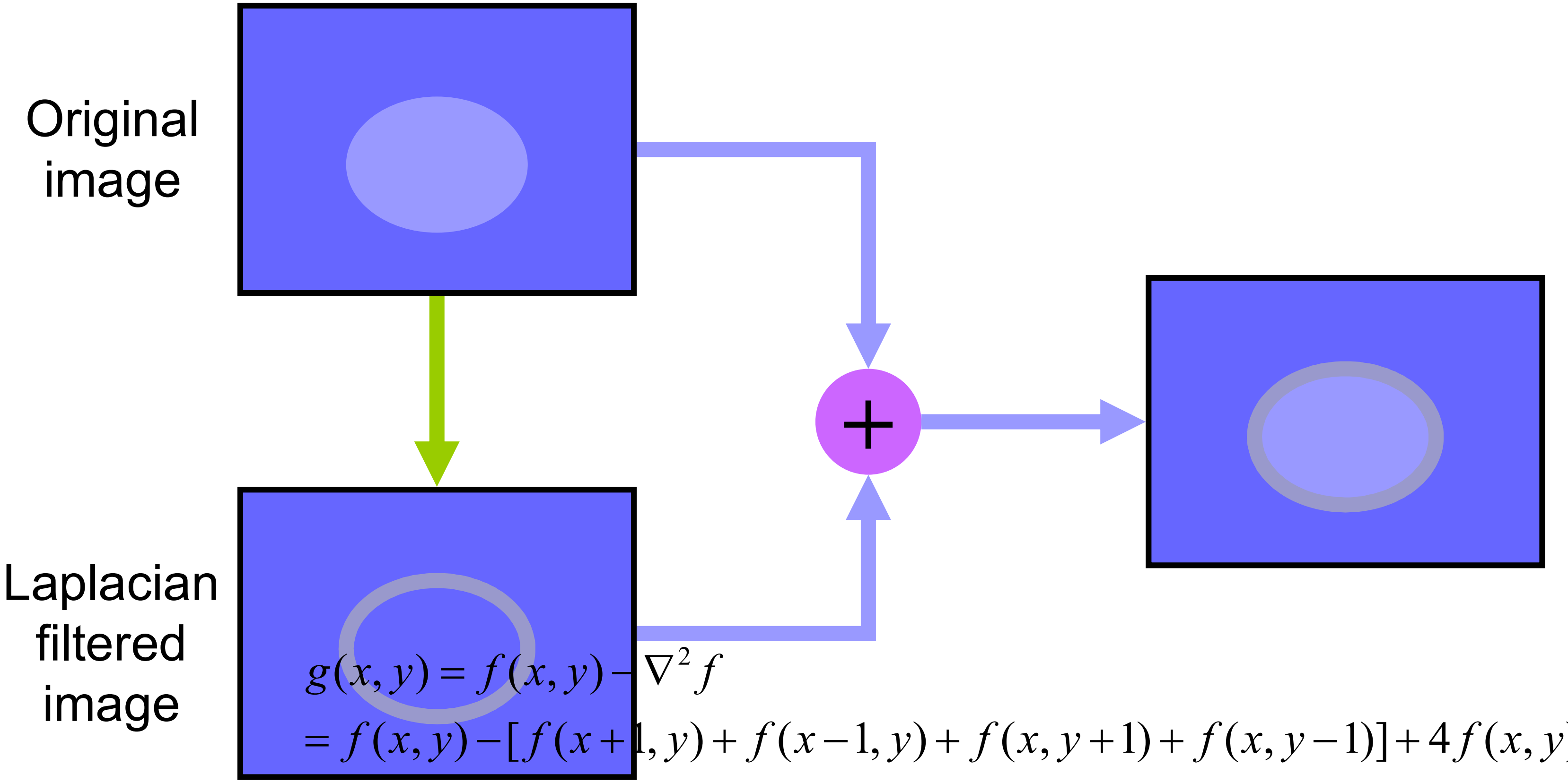
$$\nabla^2 f = [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y) \quad (I)$$

$$\nabla^2 f = 4f(x,y)-[f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)] \quad (II)$$

- **两种Laplacian: 互相反号**

Laplacian Filtering

- 可用Laplacian滤波检测到的边缘来锐化图像
  - Edges detected by the Laplacian can be used to sharpen the image!

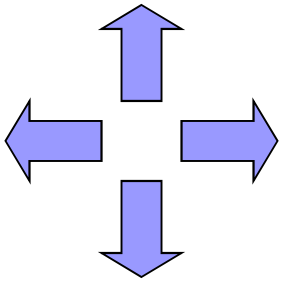


$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y) \\
&= 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

0	1	0
1	-4	1
0	1	0

Laplacian Filter



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

Laplacian Filter

+

0	0	0
0	1	0
0	0	0

Original Image

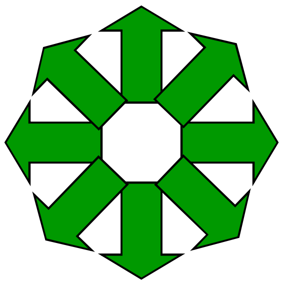
=

0	-1	0
-1	5	-1
0	-1	0

Laplacian Filter

1	1	1
1	-8	1
1	1	1

Laplacian Filter



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

Laplacian Filter

+

0	0	0
0	1	0
0	0	0

Original Image

=

-1	-1	-1
-1	9	-1
-1	-1	-1

Laplacian Filter

Example

原图像

Laplacian图像

注意噪声也被放大了

Example

By adding the Laplacian image and the original, a sharper image is achieved.

Original image

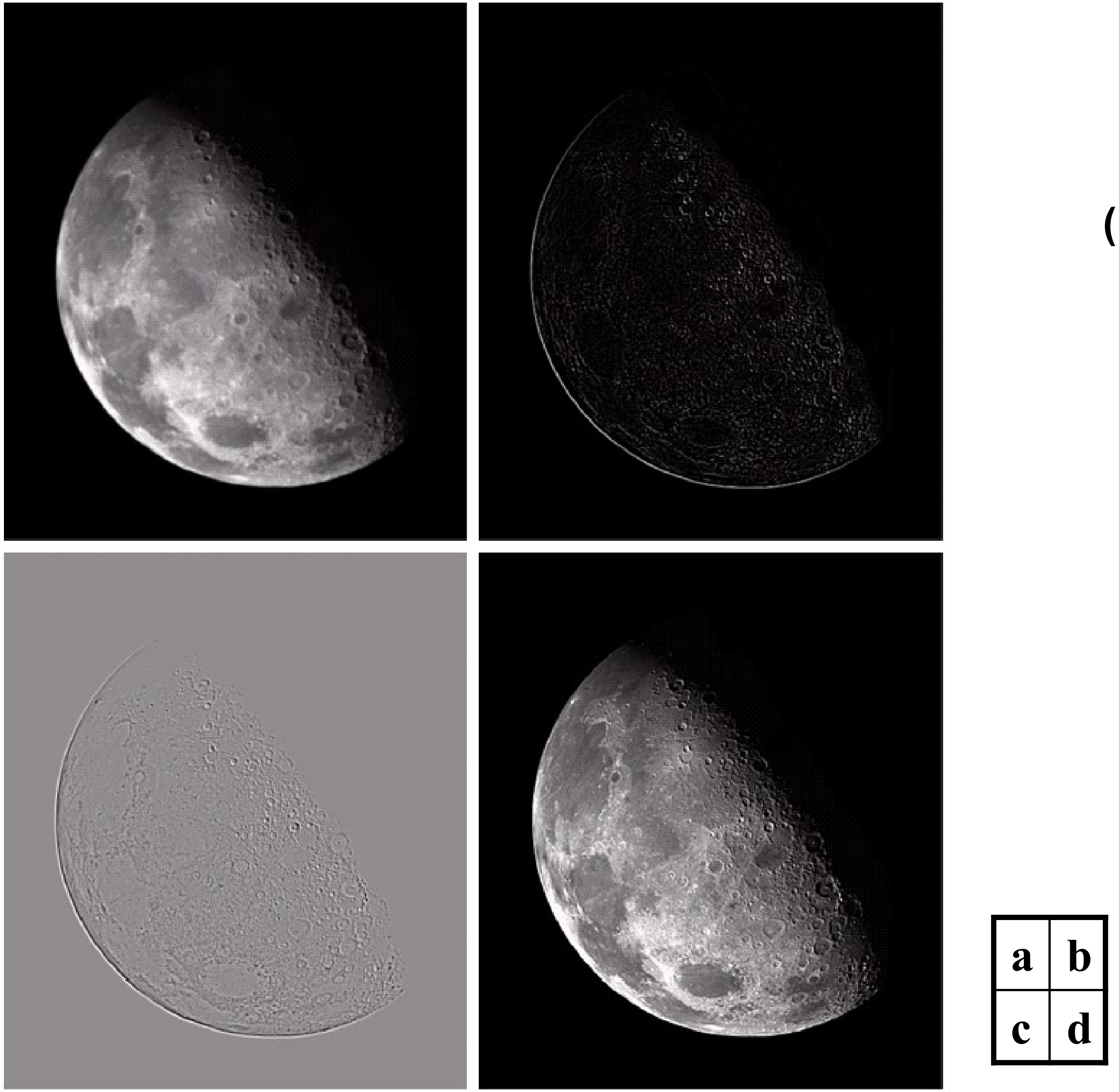
+

=

Blurry Image

Laplacian

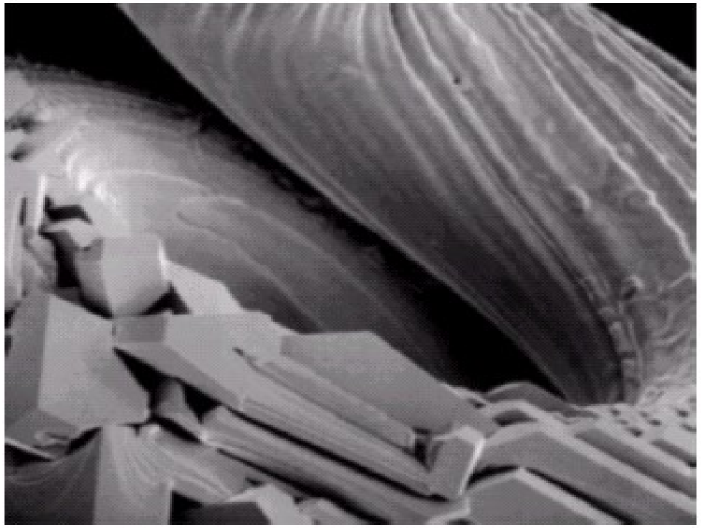
Sharpened Image

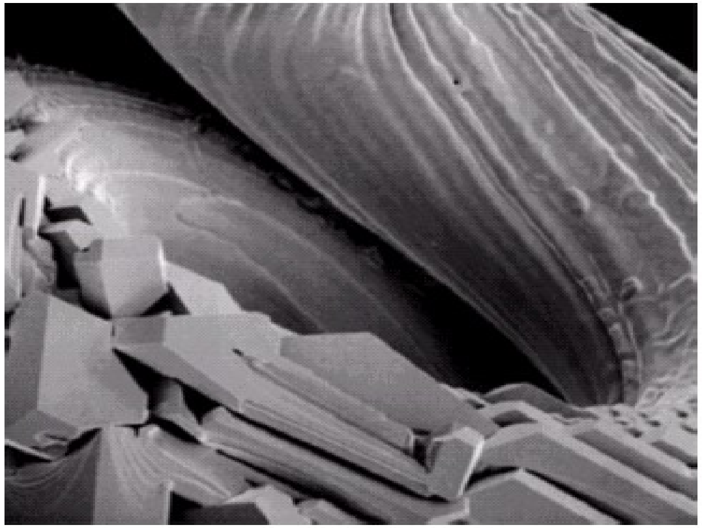


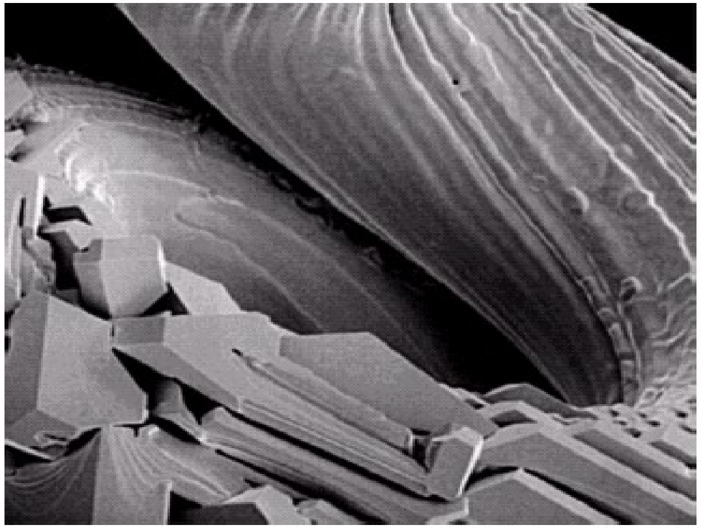
(a) Image of the North Pole of the moon.
(b) Laplacian filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1

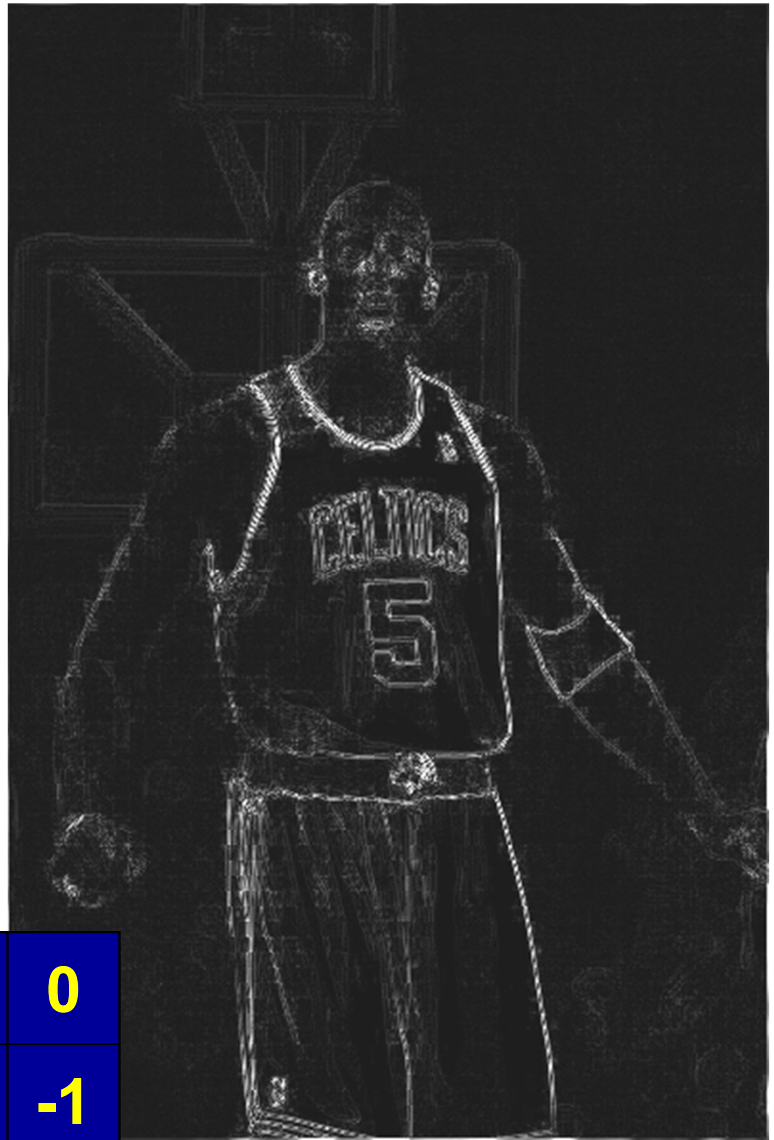




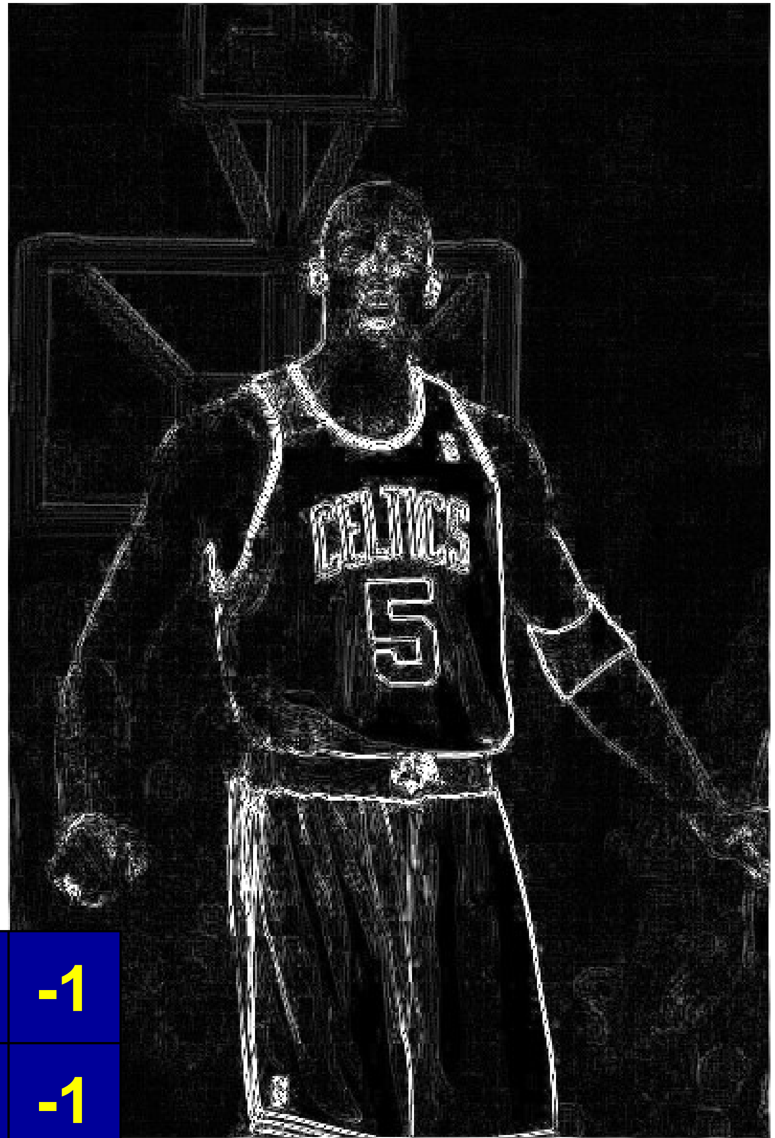


a	b	c
d		e

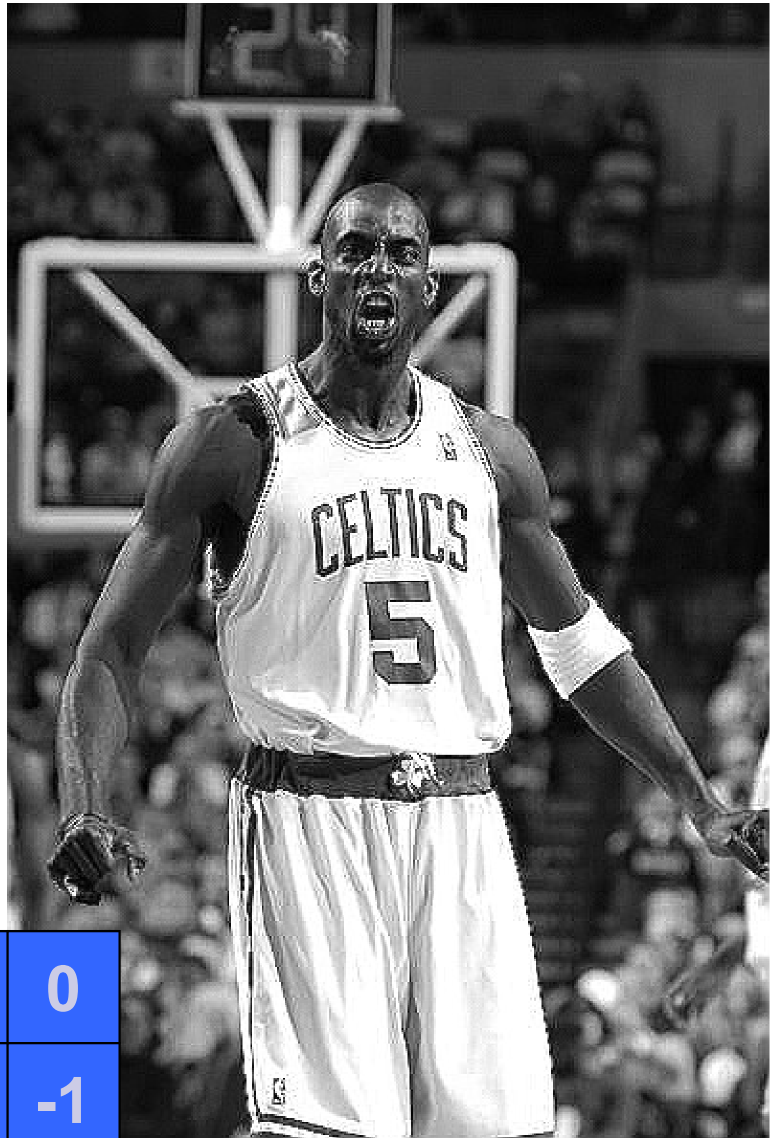
(a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, ugene.)



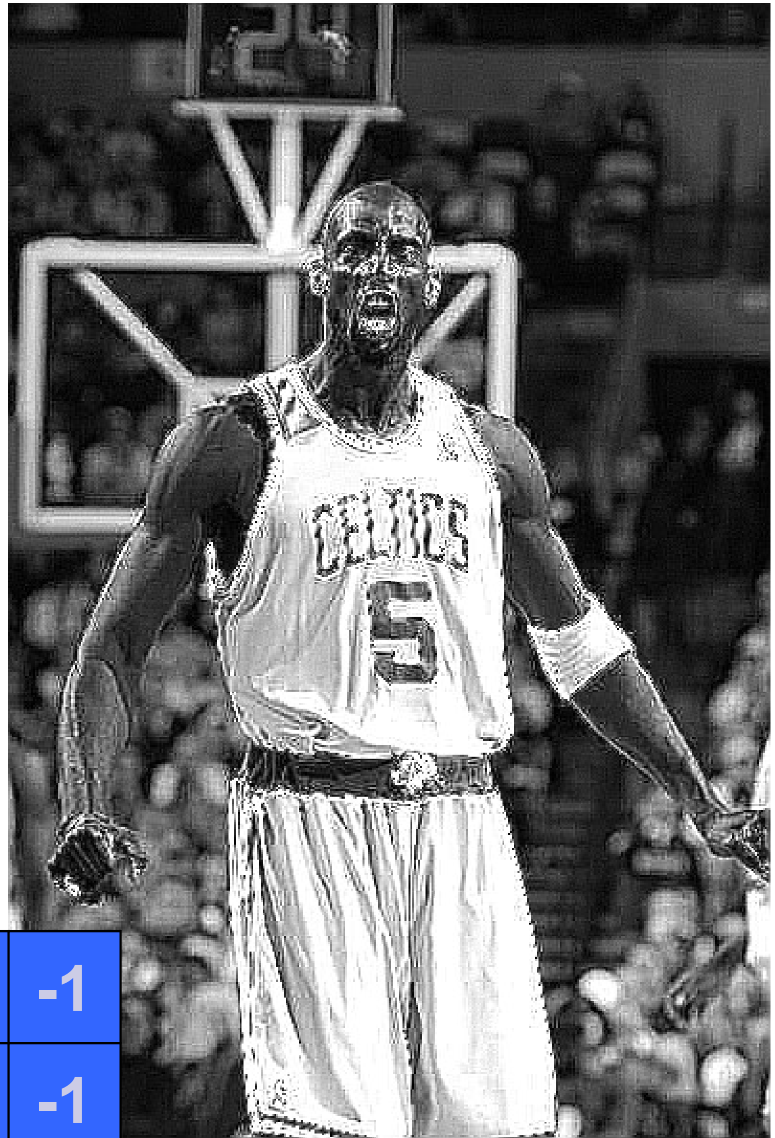
0	-1	0
-1	4	-1
0	-1	0



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

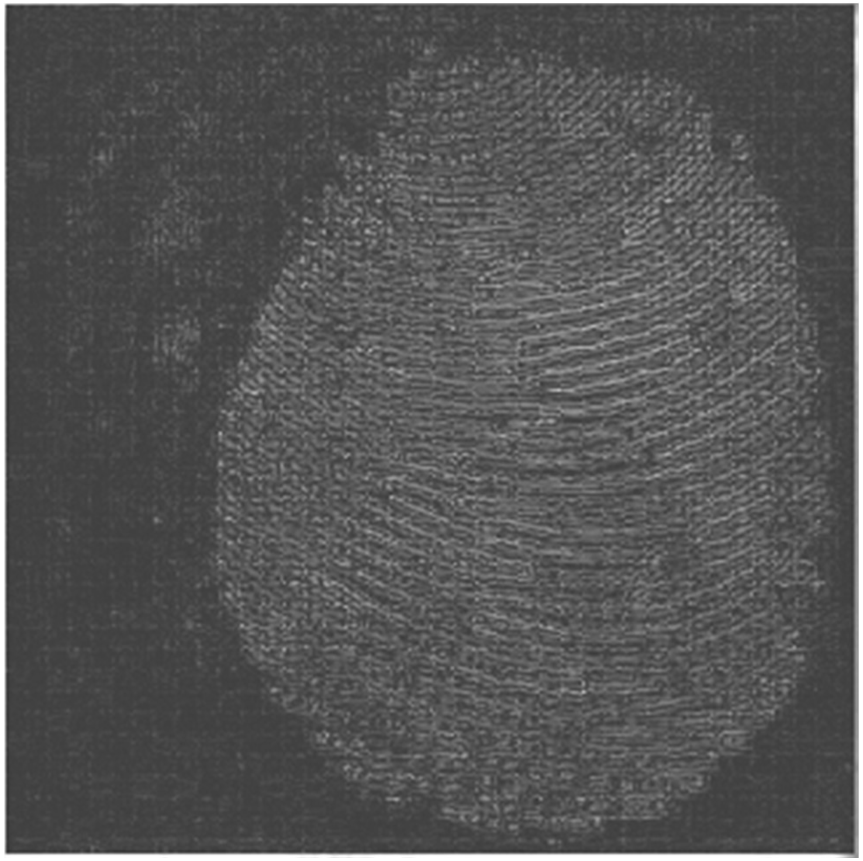


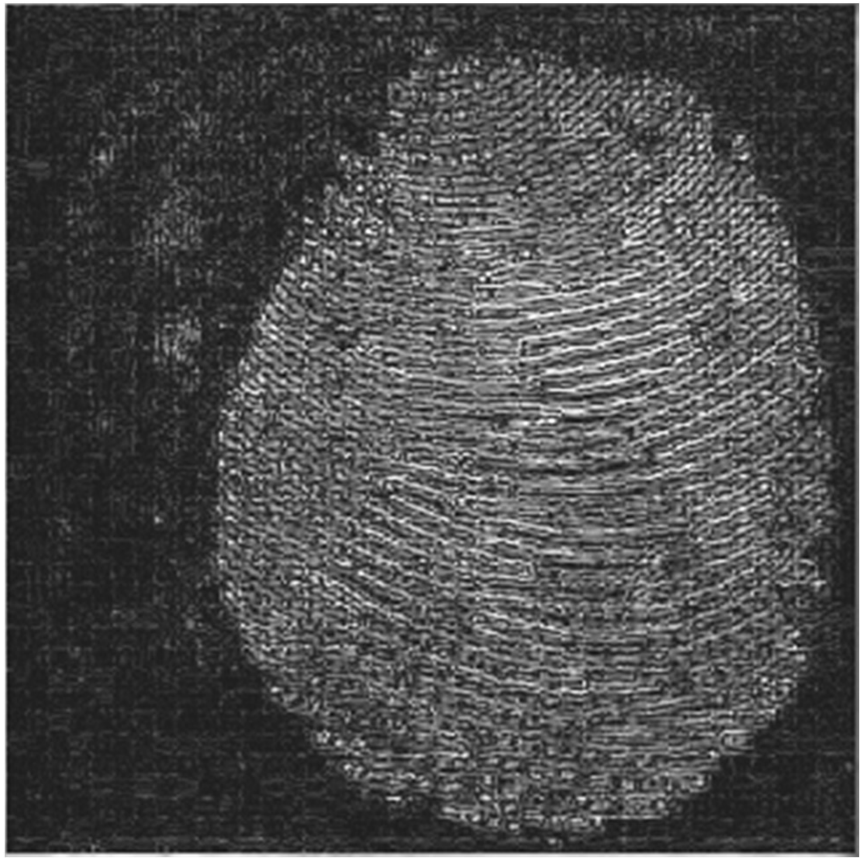
0	-1	0
-1	5	-1
0	-1	0



-1	-1	-1
-1	9	-1
-1	-1	-1

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$





$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

- Sharpening Spatial Filters
  - Basic High-Pass Spatial Filtering
  - Derivative Filters
  - High-Boost Filtering

High-Boost Filtering 高频补偿滤波器

- **Unsharp masking:**

$$f_s(x,y) = f(x,y) - \overline{f}(x,y)$$

- **High-boost filtering:**

$$\begin{aligned} f_s(x,y) &= Af(x,y) - \overline{f}(x,y) \\ &= (A-1)f(x,y) + f(x,y) - \overline{f}(x,y) \\ &= (A-1)f(x,y) + f_s(x,y) \end{aligned}$$

Original $\qquad$ lowpass

$$A \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - 1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= 1/9 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9A & 0 \\ 0 & 0 & 0 \end{bmatrix} - 1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= 1/9 \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

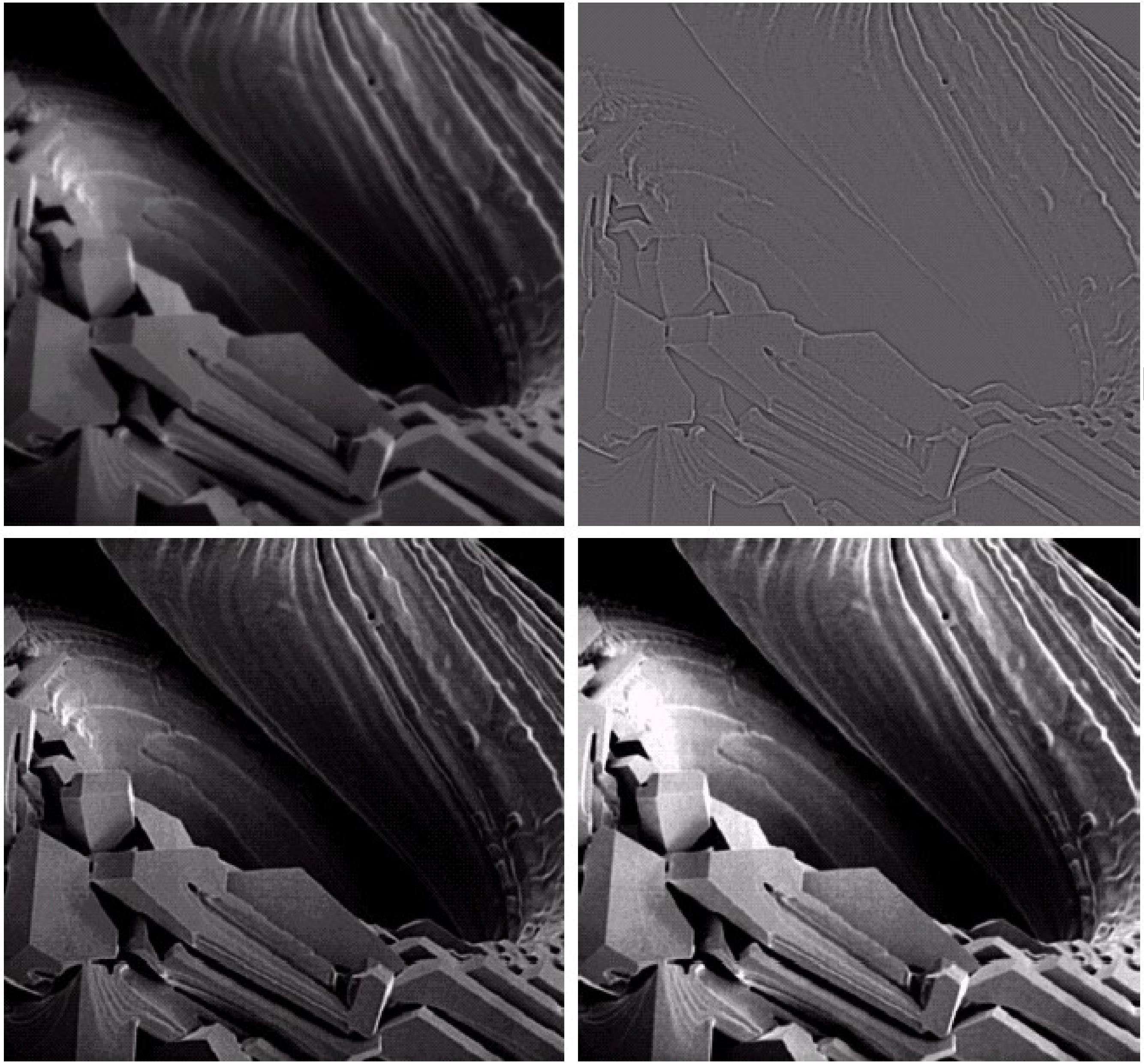
$w = 9A - 1,\ A \geq 1$

High-Boost Filtering 高频补偿滤波器

- A = 1: standard highpass result
- A > 1: the high-boost image looks more like the original with a degree of edge enhancement, depending on the value of A.
- Empirical study → 3 X 3 mask size

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & w & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

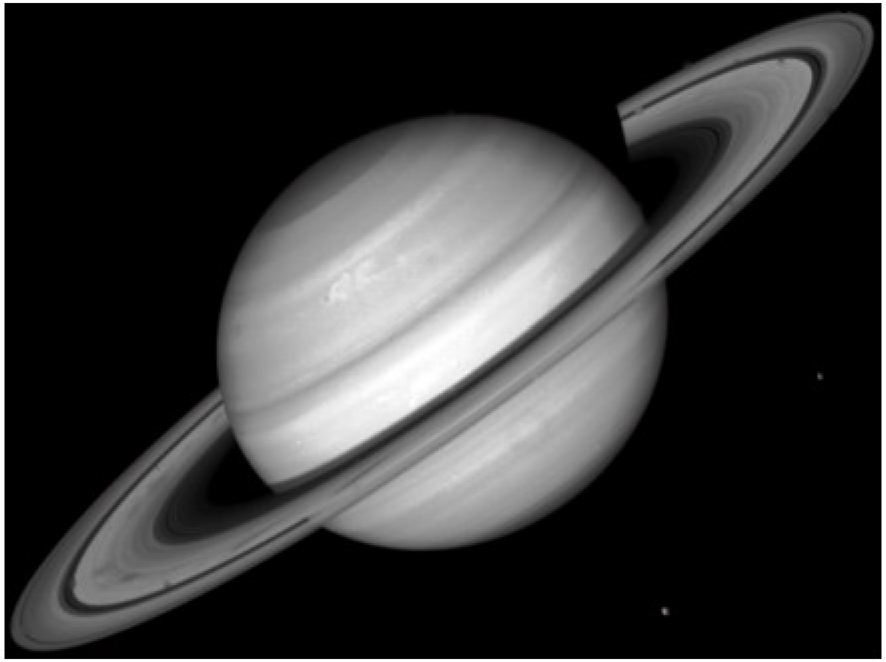
$w = 9A - 1,\ A \geq 1$



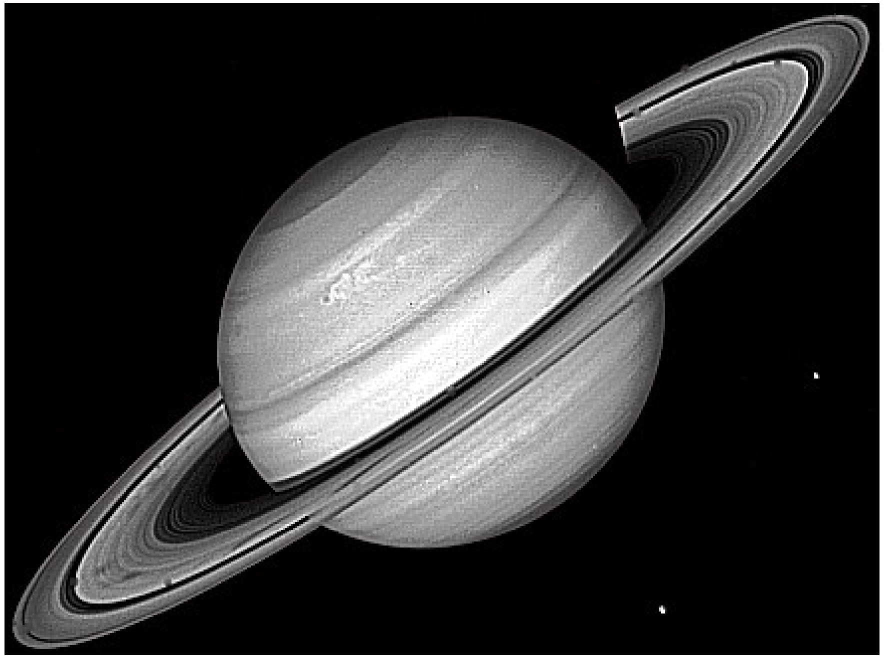
(a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using A=0. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with A=1. (d) Same as (c), but using A=1.7.

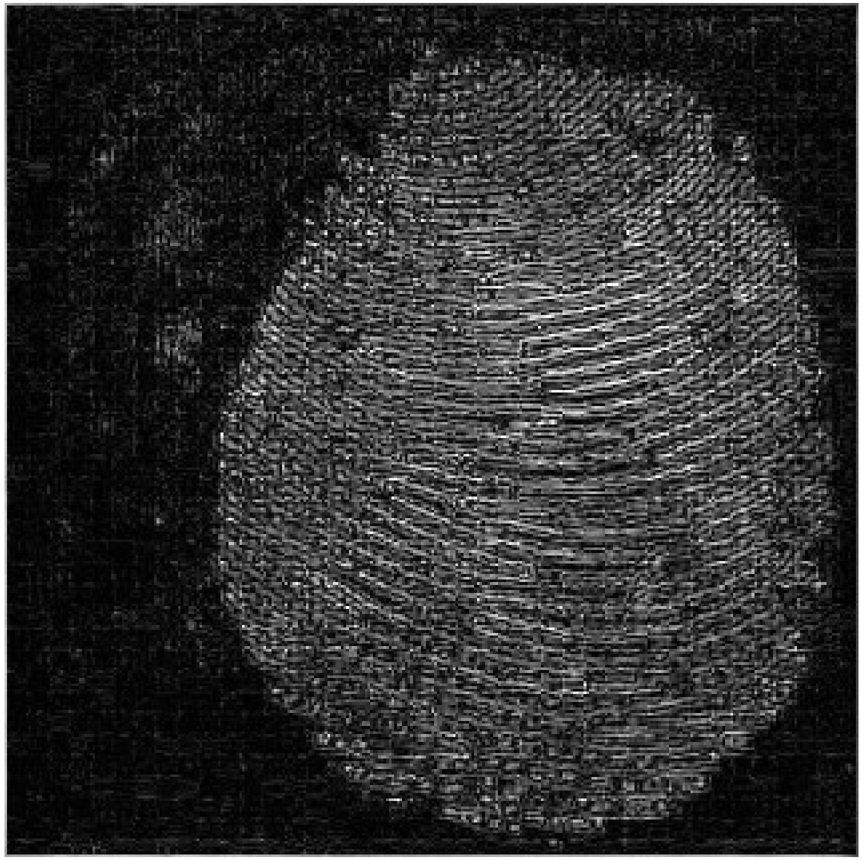
a	b
c	d

High-Boost Filtering



-1	-1	-1
-1	9	-1
-1	-1	-1

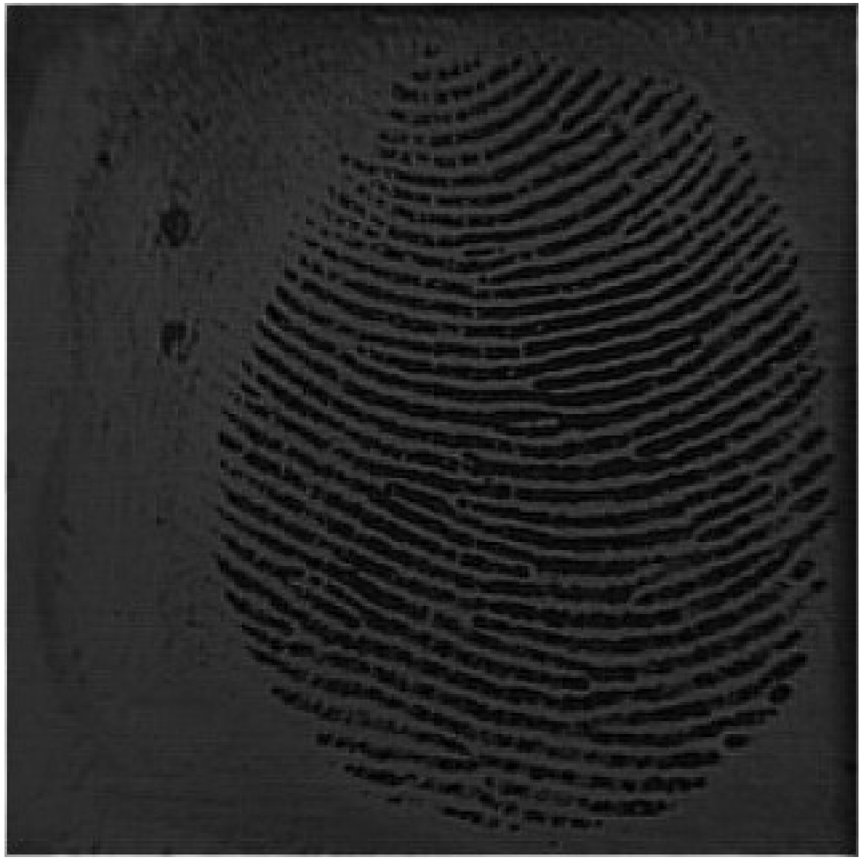




-1	-1	-1
-1	8	-1
-1	-1	-1

A = 1

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

1/9 X

-1	-1	-1
-1	13	-1
-1	-1	-1

1/9 X

-1	-1	-1
-1	17	-1
-1	-1	-1

A = 2

1/9 X

-1	-1	-1
-1	19	-1
-1	-1	-1

1/9 X

-1	-1	-1
-1	21	-1
-1	-1	-1

1/9 X

-1	-1	-1
-1	26	-1
-1	-1	-1

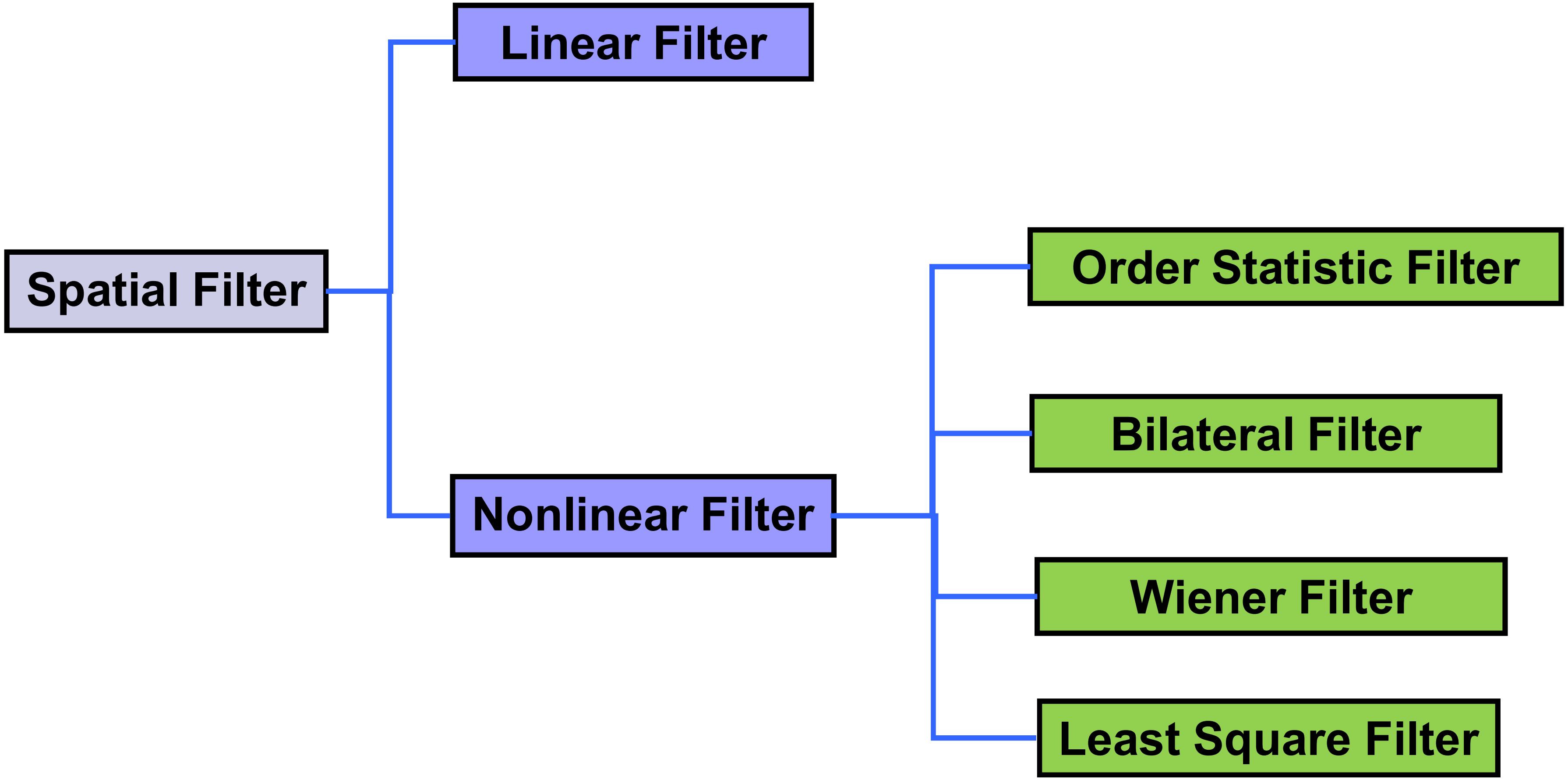
A = 3

1/9 X

-1	-1	-1
-1	35	-1
-1	-1	-1

A = 4

- **Linear Filter**
  - **Smoothing Filter**
  - **Sharpening Filter**
    - ➢Basic High-Pass Spatial Filtering
    - ➢Derivative Filters
      - •First Derivatives: The Gradient
      - •Second Derivatives: The Laplacian
    - ➢High-Boost Filtering


Spatial Filter
Linear Filter
Nonlinear Filter
Order Statistic Filter
Bilateral Filter
Wiener Filter
Least Square Filter

Order Statistics Filters

- **Sort pixels and select the one at specified position as output**
  - **Median (中值） filter**
  - **Max filter**
  - **Min filter**

Median Filtering

- **Median filtering (nonlinear)**
  - **The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel**
    - **instead of by the weighted average as before**
  - **Used primarily for noise reduction**
    - **Eliminates salt-and-pepper noise**
  - **Weighted Median Filter …**

Unfiltered values

6	2	0
3	97	4
19	3	10

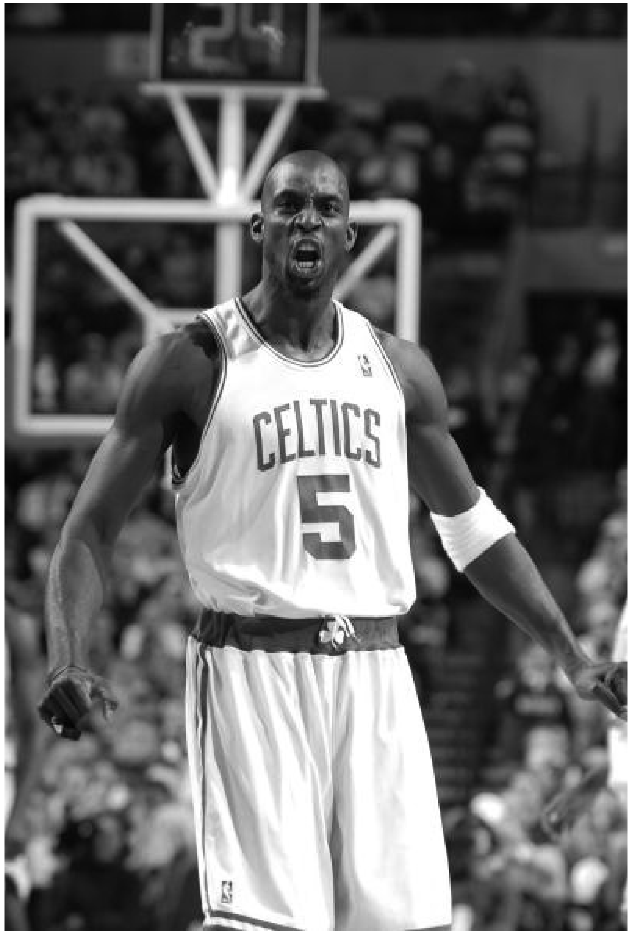
In order

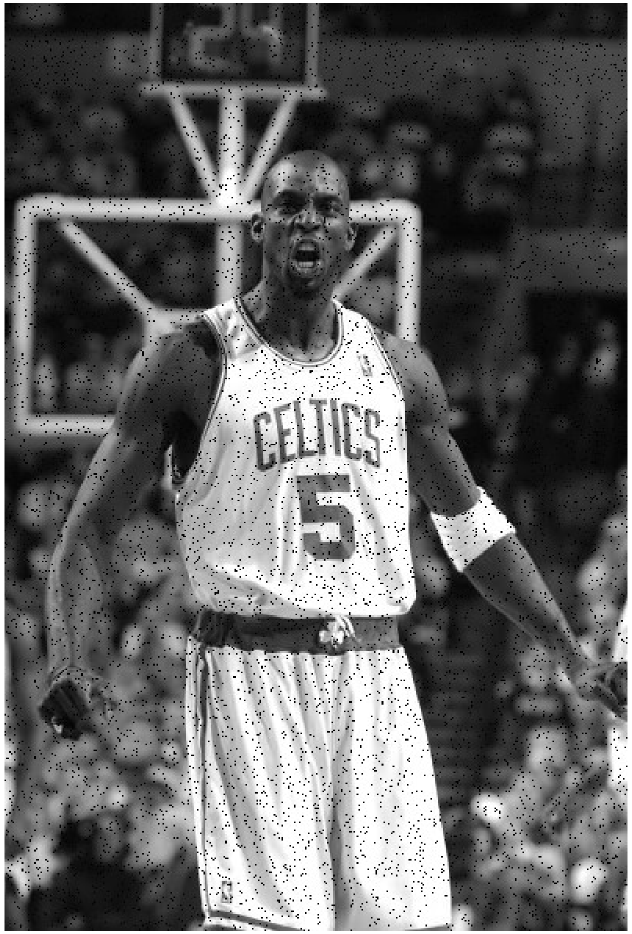
0, 2, 3, 3, 4, 6, 10, 19, 97

Median filtered

6	2	0
3	4	4
19	3	10

Salt&Pepper Noise





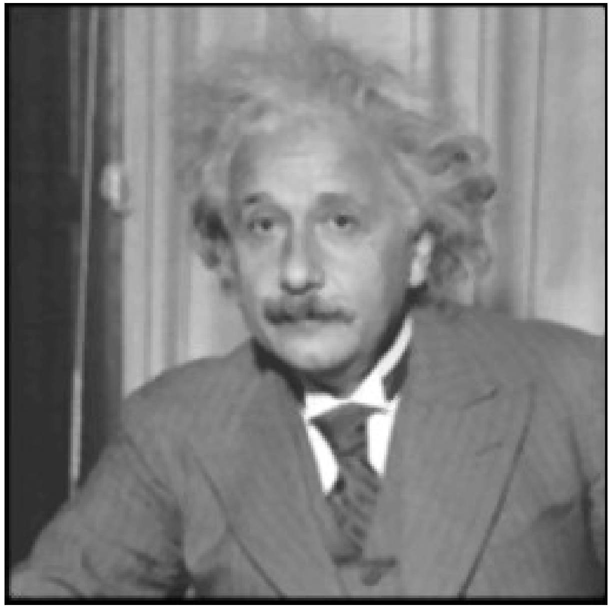
Noise added

Noise Eliminated

3x3 median filtering

Median Filtering - Different Mask Size

Salt & pepper noise image



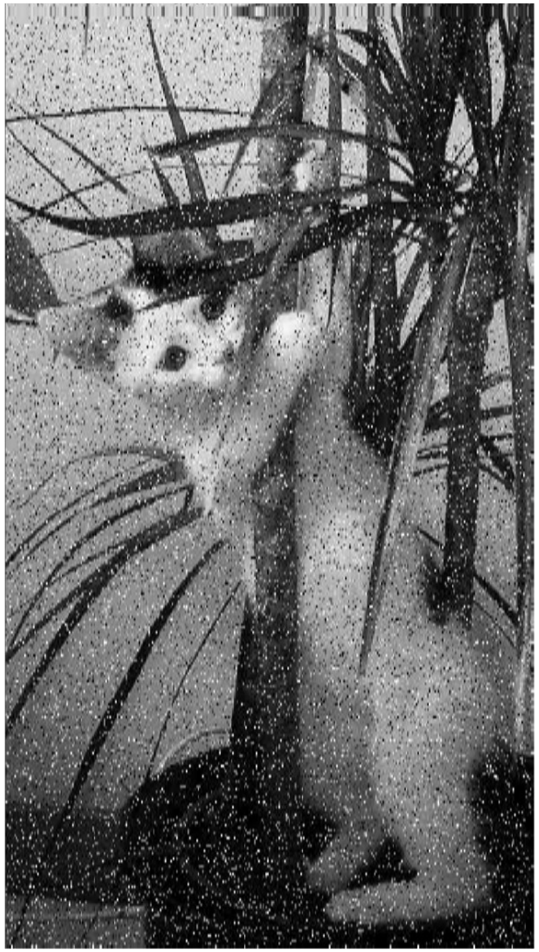
3x3 Median filtering

15x15 Median filtering

Mask size↑ blurring ↑

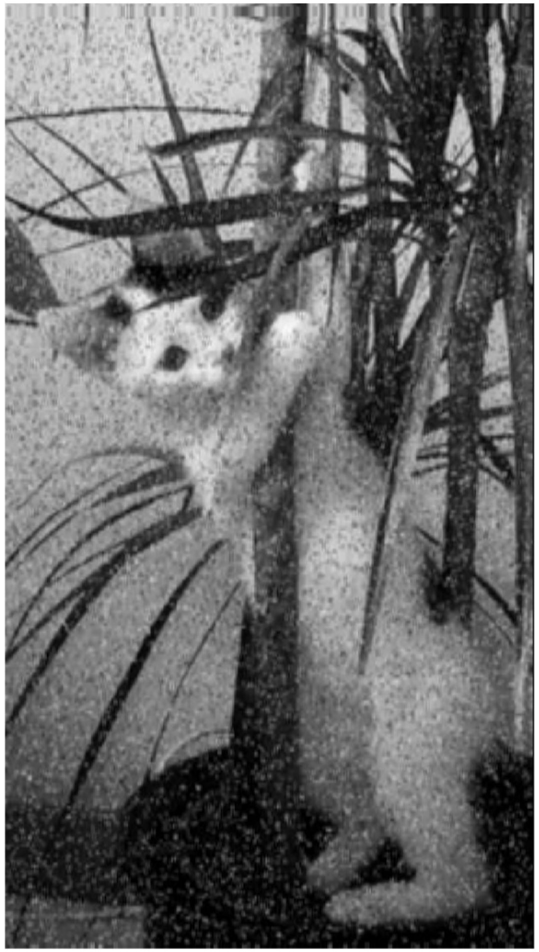
Median Filtering & Mean Filtering

Impulse noise added

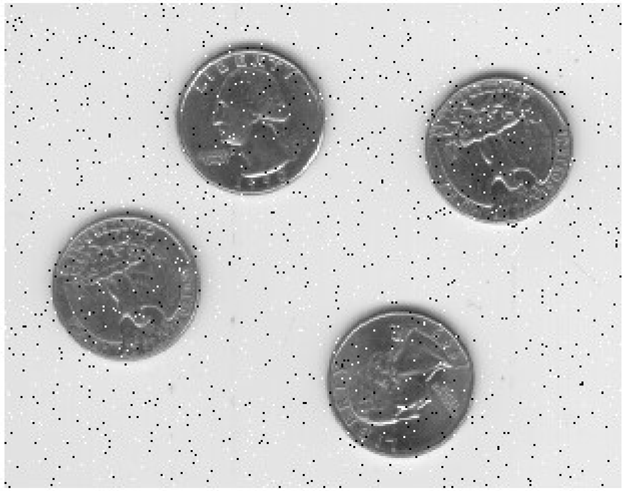


Median Filtered image

Mean Filtered image

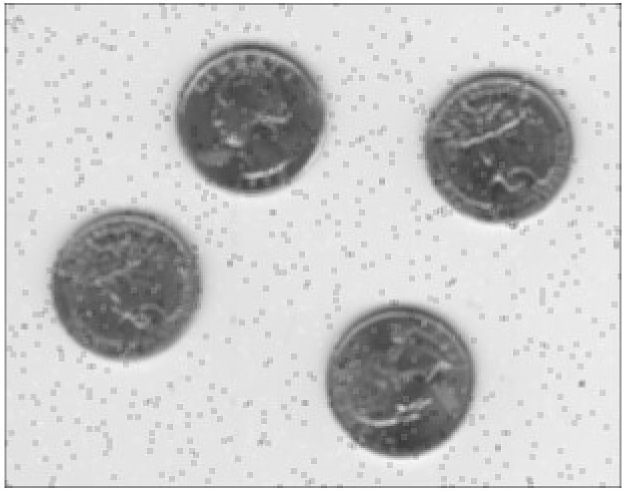


Original

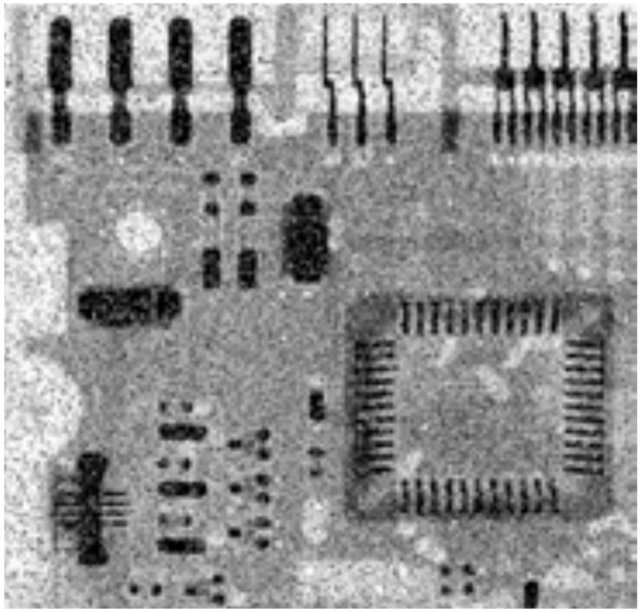


Salt and Pepper noise added

3x3 mean filter

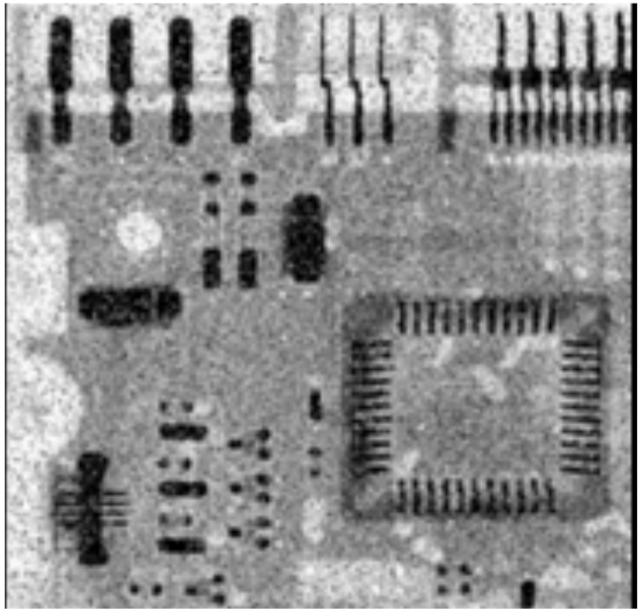


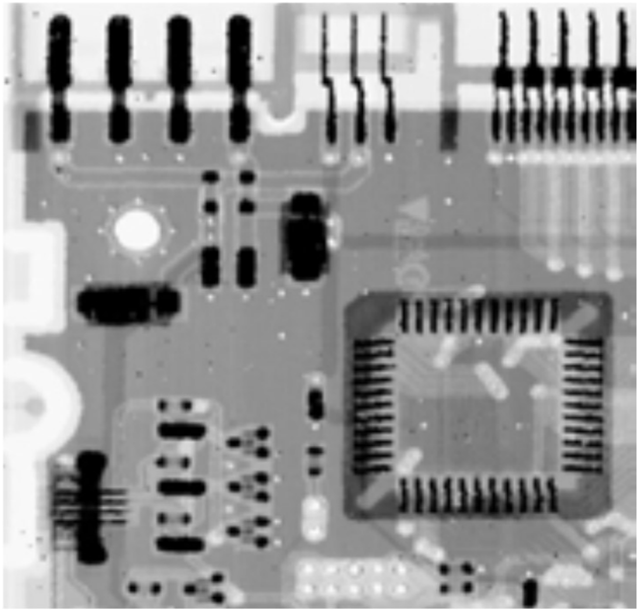
3x3 median filter



a	
b	c

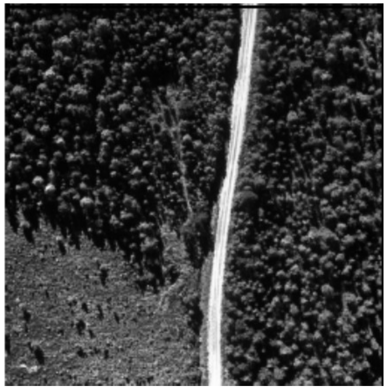
(a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3*3 averaging mask. (c) Noise reduction with a 3*3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



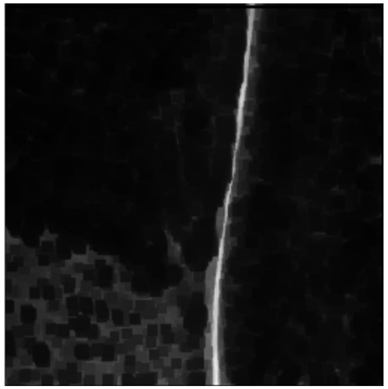


Order Statistics Filters

Original image

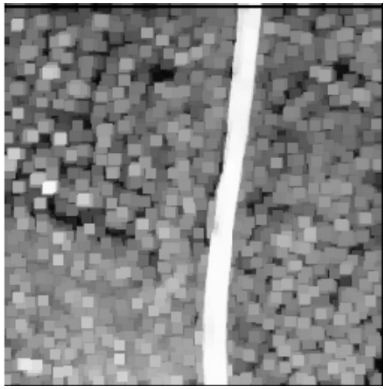


7x7 min



7x7 median

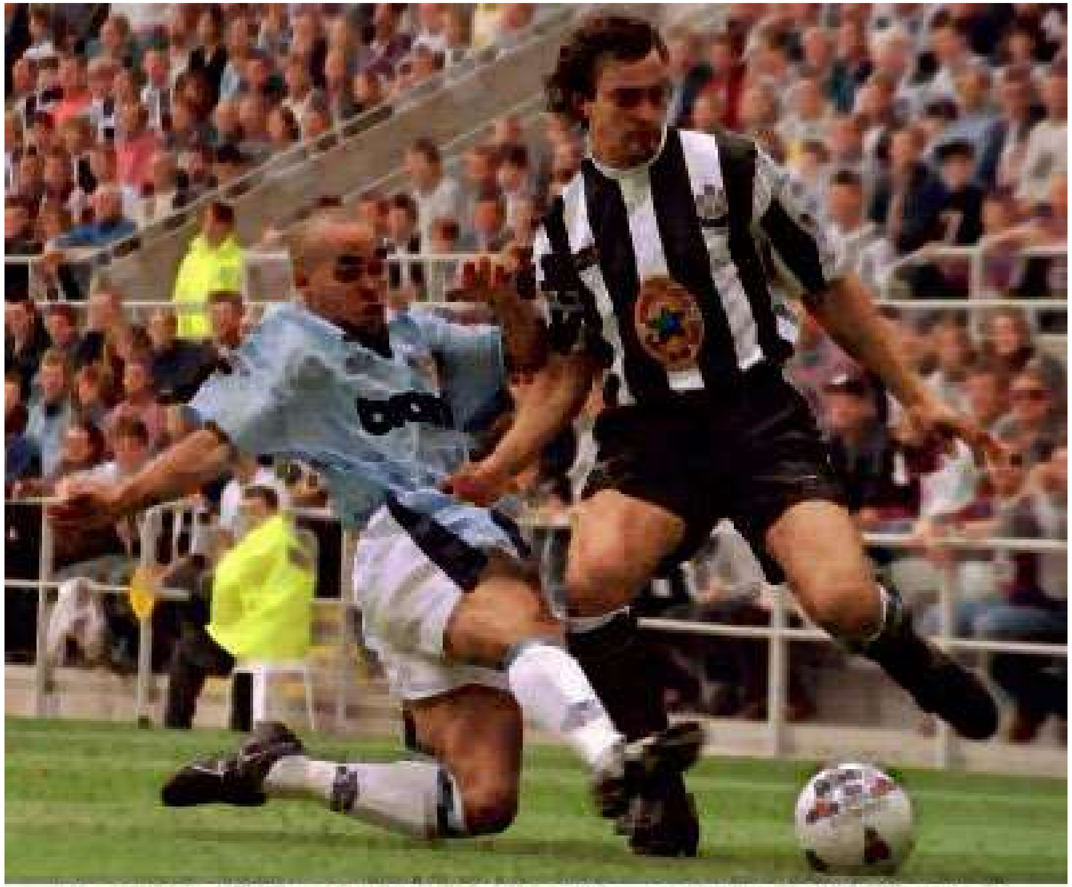
7x7 max



Max Filtering

Min Filtering

Man City's Richard Edghill tackles Newcastle's David Ginola at St James Park. City lost the match 3-1 Picture: REUTERS



Man City's Richard Edghill tackles Newcastle's David Ginola at St James Park. City lost the match 3-1 Picture: REUTERS

Fast Max/Min Filter

- …

Fast Max/Min Filter

$1D \quad M = 5$

p_c

1	**6**	**4**	**2**	**3**	**8**	**0**	**9**	**7**

Fast Max/Min Filter

$1D \quad M = 5$

p_c

1	**6**	**4**	**2**	**3**	**8**	**0**	**9**	**7**

T_i :

6	**6**	**4**	**3**	**3**	**8**	**8**	**9**	**9**

$$T_i = \max\{p_i, \ldots, p_c\} \Leftrightarrow T_i = \max\{p_c, \ldots, p_i\}$$

Fast Max/Min Filter

$1D \quad M = 5$

p_c

1	6	4	2	3	8	0	9	7

T_i :

6	6	4	3	3	8	8	9	9

$T_i = \max\{p_i, \ldots, p_c\} \Leftrightarrow T_i = \max\{p_c, \ldots, p_i\}$

O_i :

		6	8	8	9	9		

$$O_i = \max\{T_{i-M/2}, T_{i+M/2}\}$$

Fast Max/Min Filter

- 2D ?

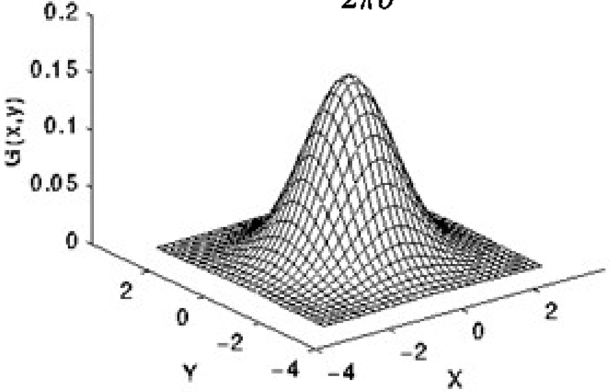
Fast Median Filter

- **Perreault S, Hébert P. Median filtering in constant time[J]. IEEE Transactions on Image Processing, 2007, 16(9): 2389-2394.**

Bilateral Filter (双边滤波）

- **高斯滤波：消除噪音的同时也使边缘变得模糊**

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



二维高斯函数

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

5*5卷积核

Bilateral Filter (双边滤波）

- 双边滤波：计算权重时同时考虑空间位置和像素颜色之差

Gaussian:

$$G(x, y) \propto e^{-\frac{x^2+y^2}{2\sigma^2}}$$

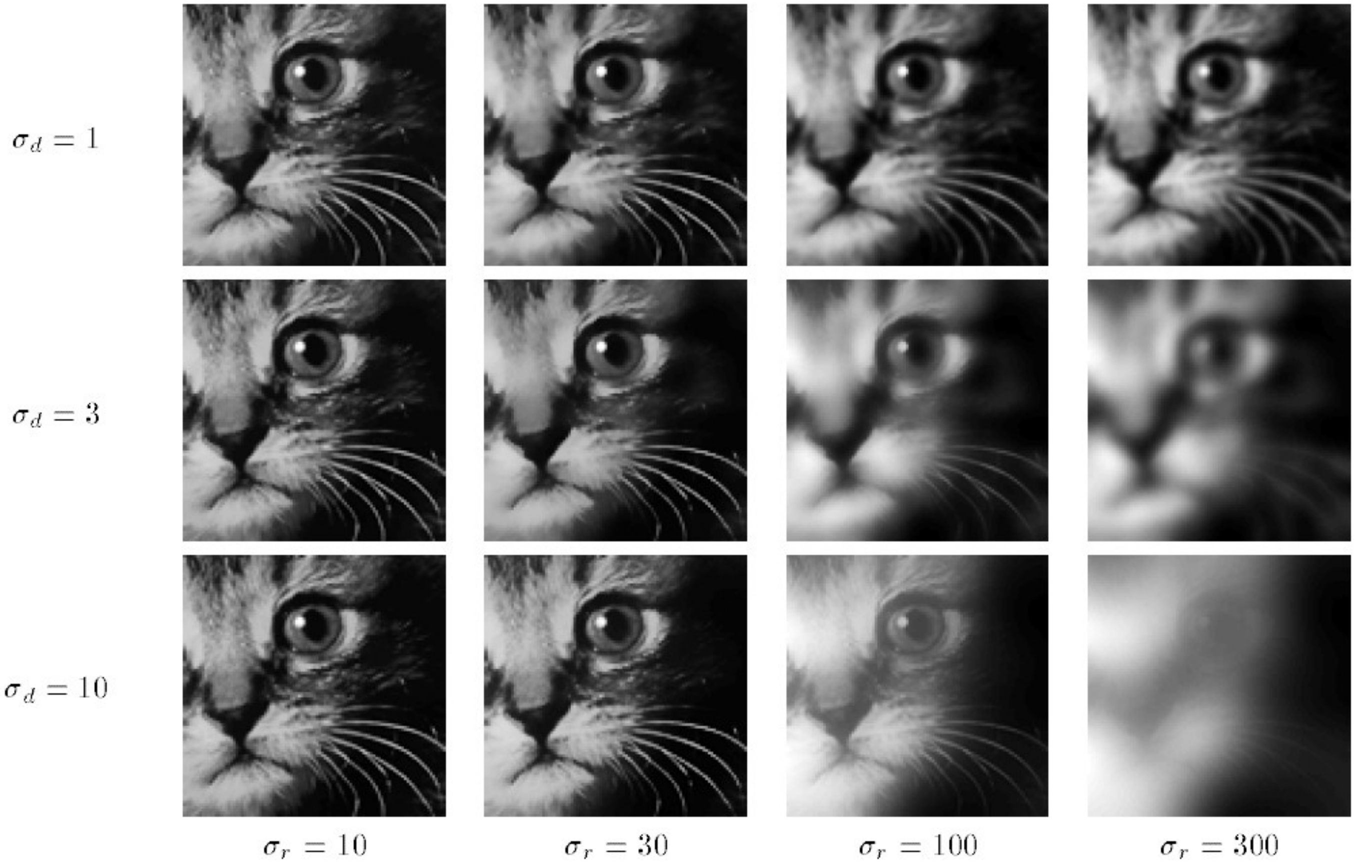
⬇

Bilateral:

$$G(x, y) \propto e^{-\frac{x^2+y^2}{2\sigma_d^2}} e^{-\frac{\|I(x,y)-I(0,0)\|^2}{2\sigma_r^2}}$$

Bilateral Filter (双边滤波）

- Tomasi C, Manduchi R. Bilateral filtering for gray and color images[C]//Computer Vision, 1998. Sixth International Conference on. IEEE, 1998: 839-846.



- 设计5*5的Sobel梯度算子

- 根据Laplacian函数的定义，推导如下标准的Laplacian算子

0	1	0
1	-4	1
0	1	0

- 图像的线性滤波是否是对图像的线性变换？可否表示成对图像的矩阵变换的形式?