空间滤波器

分为线性滤波器和非线性滤波器

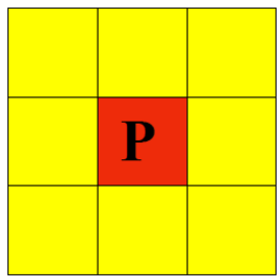
- 线性滤波器，f 是线性函数
- 非线性滤波器，f 是非线性函数

空间域滤波器 (Spatial Filter)

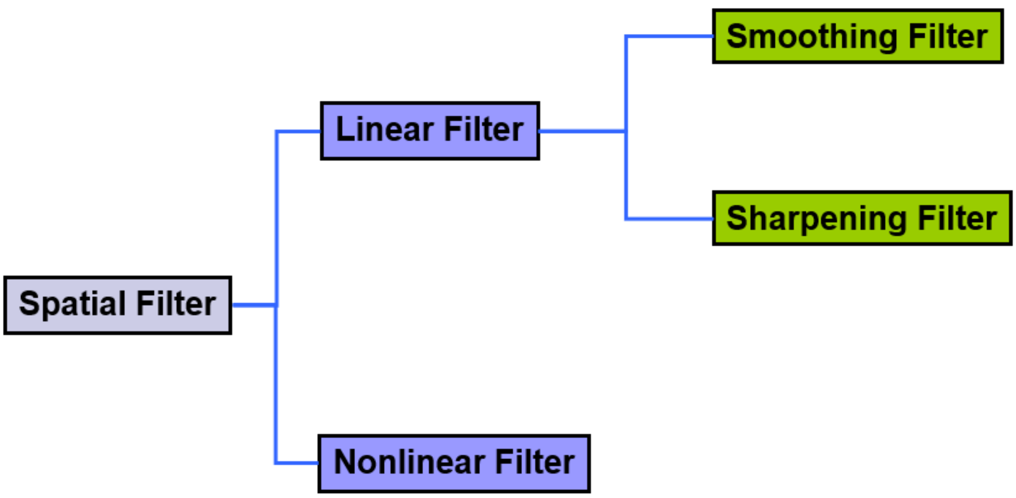
- **对任意像素 p**

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

N_p **为像素 p 的某个邻域像素集合**



$$\mathrm{N}_p^{3\times3} = \{p_0, p_1, \ldots, p_8\}$$



线性滤波器

- $w(s,t)$ 是滤波核
- 分为平滑滤波器和锐化滤波器
- 图像的线性滤波是对图像的线性变换，可以表示成对图像的矩阵变换的形式（对整个图像写一个滤波的矩阵，其中有些值为0）

线性滤波器 (Linear Filter)

- 邻域像素的加权平均

$$p' = \sum_i w_i p_i$$

$$\Downarrow$$

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

平滑滤波器（低通滤波器）

- 主要是加权平均
- 举例：均值滤波，高斯滤波

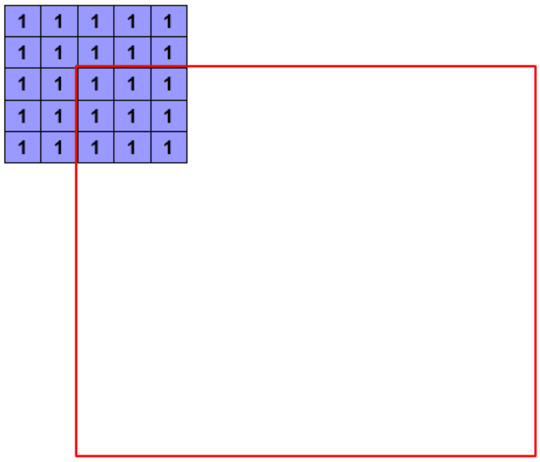
Smoothing Filter

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

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

边界处理

- 为什么要进行边界处理
  - 原因一：
    - 对边界附近的像素，滤波核的部分可能落在图像区域外。



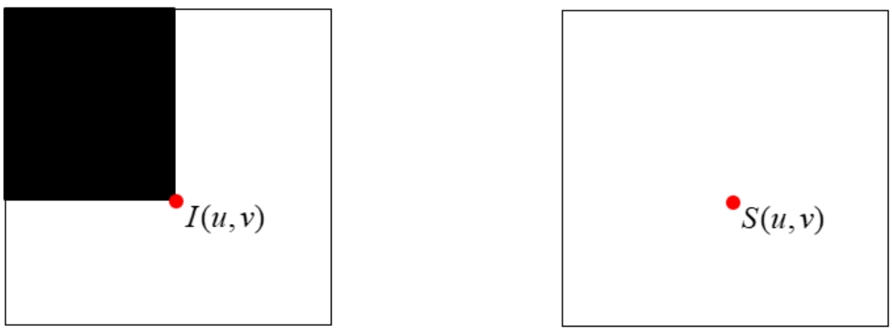
  - 原因二：如果不进行边界处理，不同像素点作为滤波核中心的次数不一样，有些点不会作为滤波核的中心出现
- 对边界外某个范围内的区域进行填充（常用方法）
  - 常数填充
  - 镜像填充
- 在边界附近调整滤波核的大小

快速均值滤波

首先这里需要了解一下积分图，积分图实际上就是对图像进行二维前缀和操作

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。



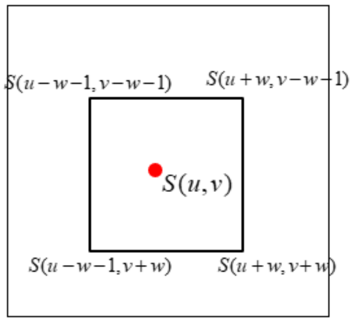
- 积分图可增量计算，只需对原图进行一遍扫描：

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

- 通过积分图进行快速均值滤波（右下角 + 左上角 - 另外两个角）

- 设滤波窗口大小为2w+1，滤波结果为图像O，则：

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

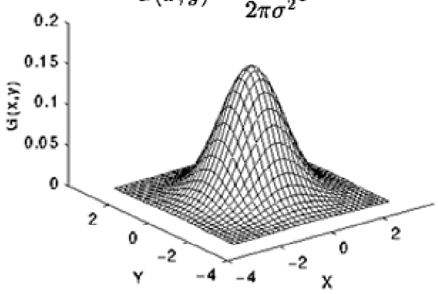


Z=(2w+1)*(2w+1)为像素个数；
中括号内即为滤波窗口覆盖的像素颜色值之和；

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

$\frac{1}{273}$

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

5*5卷积核（**注意归一化!**）

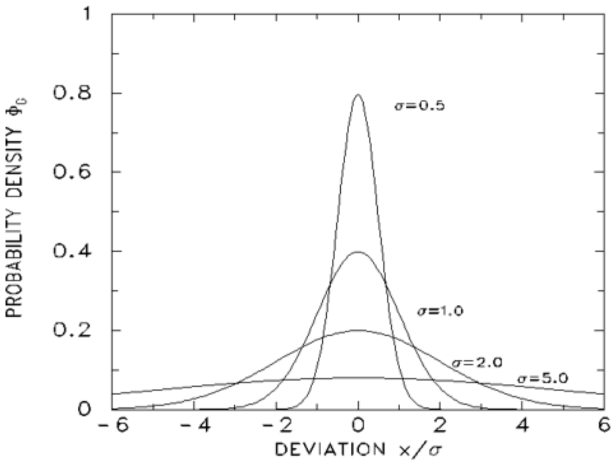
- 高斯滤波具有行列可分离性。也就是说，我们在对二维图像进行高斯滤波时，为了减少运算时间，可以先按行滤波，再将得到的结果按列滤波，或者反过来。总之就是将二维滤波转化成两个一维滤波

- **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

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

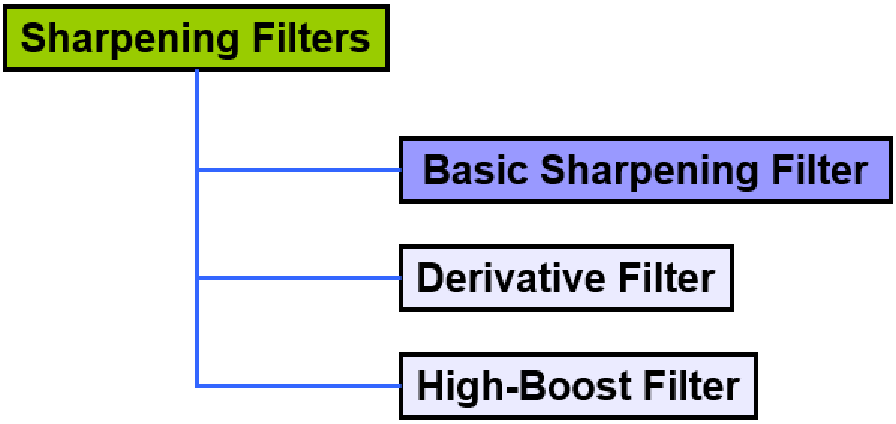
- 一般情况下，σ 越大，高斯滤波核的尺寸就越大，$size = 6\sigma - 1$

  - **核大小 与 σ 的关系?**
    - σ **越大，核应该越大**



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

锐化滤波器（高通滤波器）



基本高通滤波器：

- 滤波器中心有正系数，边缘上有负的系数，且总和为0
- 在平坦变化的区域很暗，在剧烈变化的区域很亮
- 可以用来进行边缘检测。滤波以后，只有边缘的区域是亮的

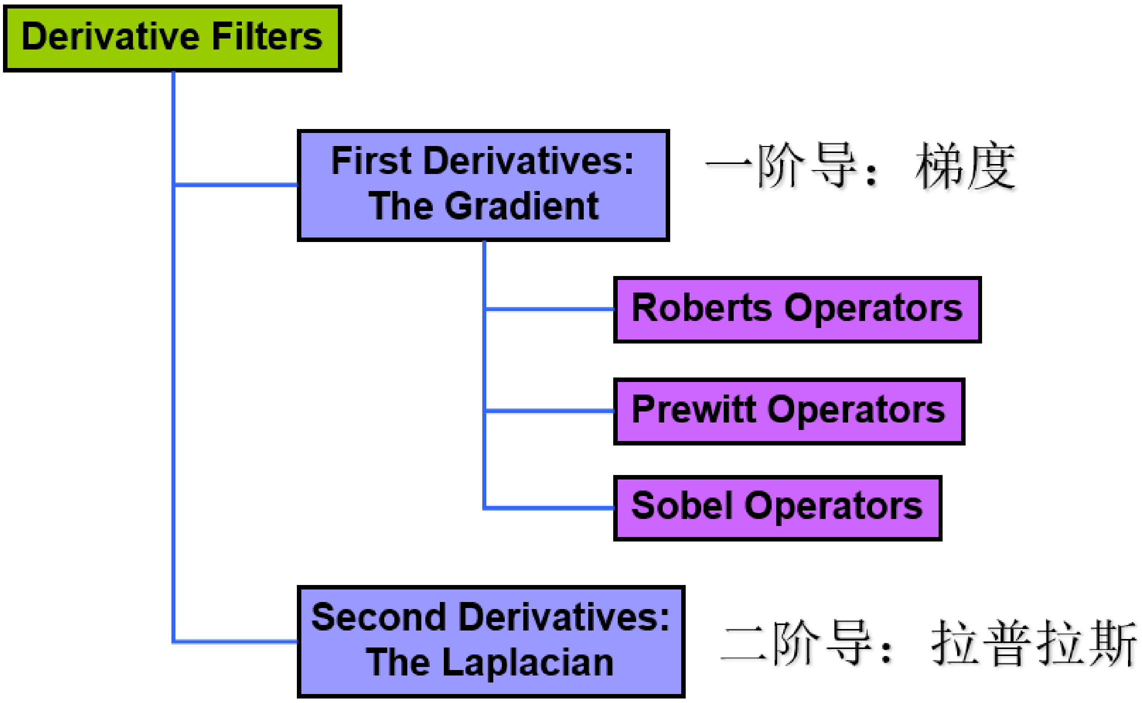
-1	-1	-1
-1	8	-1
-1	-1	-1

-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 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

导数滤波器



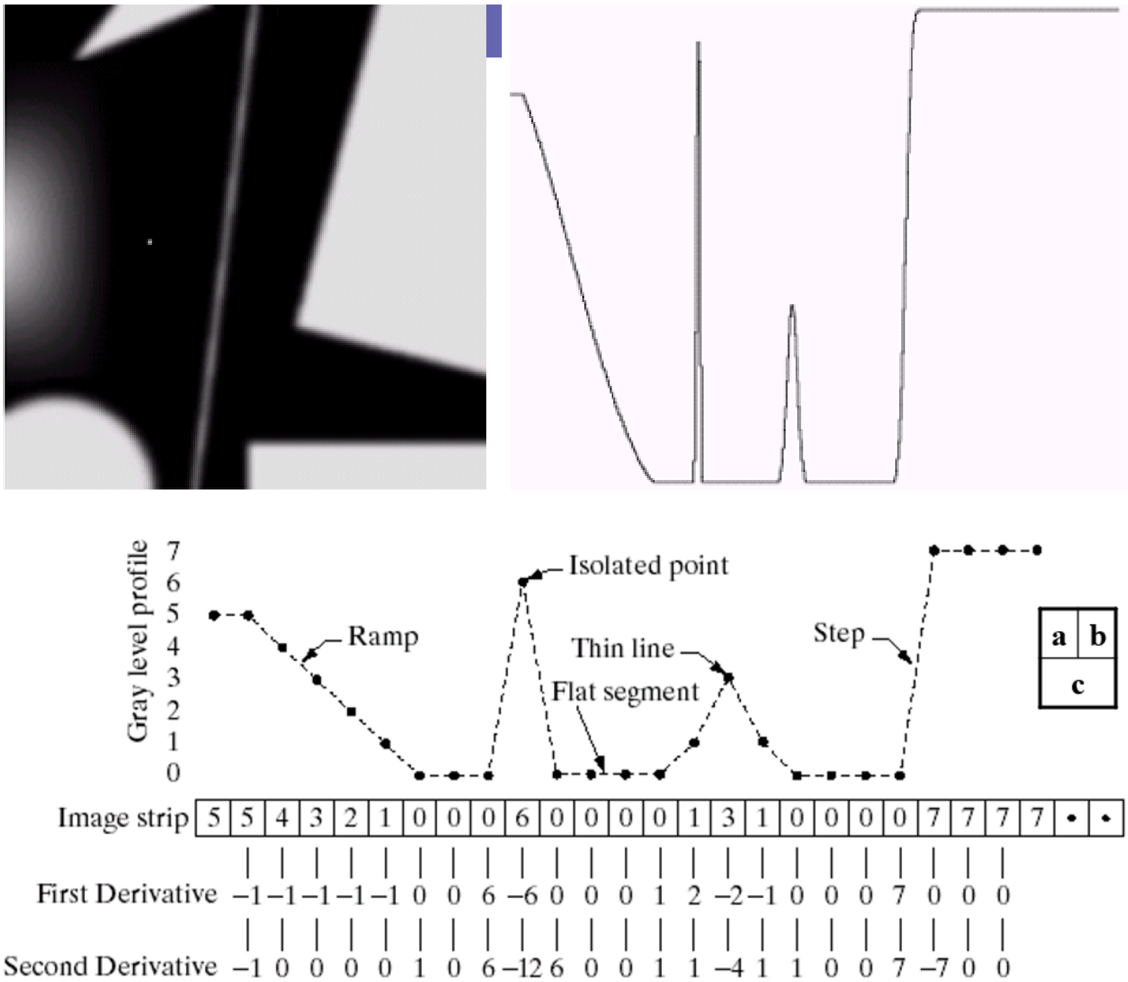
- 图像平均和图像差分（离散点的导数就是用差分计算的）的区别
  - 图像平均类似于积分，导致图像模糊
  - 图像差分则可能有相反的效果，导致图像锐化
  - 一阶导

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

  - 二阶导

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

- 一阶导和二阶导的区别（以下图为例，一阶导和二阶导都按照上述公式计算）
  - 左边的图中间有一个噪声白点，还有一条白线，右边的图是其沿着中间从左到右像素变化曲线



- 对于一阶导来说
  - 零：在常数值的区域
  - 非零：在斜坡的开始和斜坡上（斜坡结束点不是，比如上面第一个斜坡的结束位置，导数为 0）
- 对于二阶导来说
  - 零：在常数值的区域和斜坡上
  - 非零：在斜坡的两端
- 总的来说
  - 一阶导产生的边缘可以有较大的宽度，因为斜坡可以是持续的，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

一阶导的运用 - > 梯度

- 图像处理中最为常见的差分方法就是梯度

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

梯度是一个向量，指向灰度变化最大的方向，其长度为：

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

- 三种典型的梯度算子：$Roberts, Prewitt, Sobel$
  - $Roberts$算子
    - **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

  - $Prewitt$算子（下面减去上面，右边减去左边）
    - **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

  - $Sobel$算子（下面减去上面，右边减去左边，中间位置的数乘以 2）

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

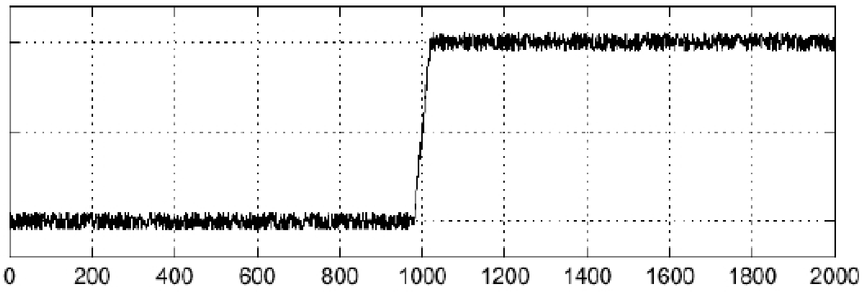
G_y

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

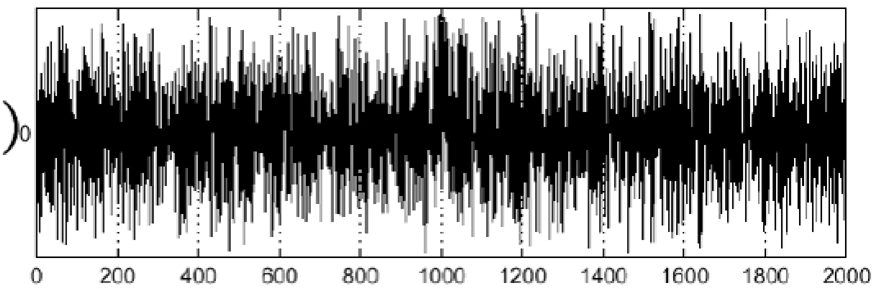
G_x

- 受噪声的影响
    - 这里看到，噪声点和边缘点的梯度实际上是差不多大的，因此我们很难分辨出边缘

$f(x)$



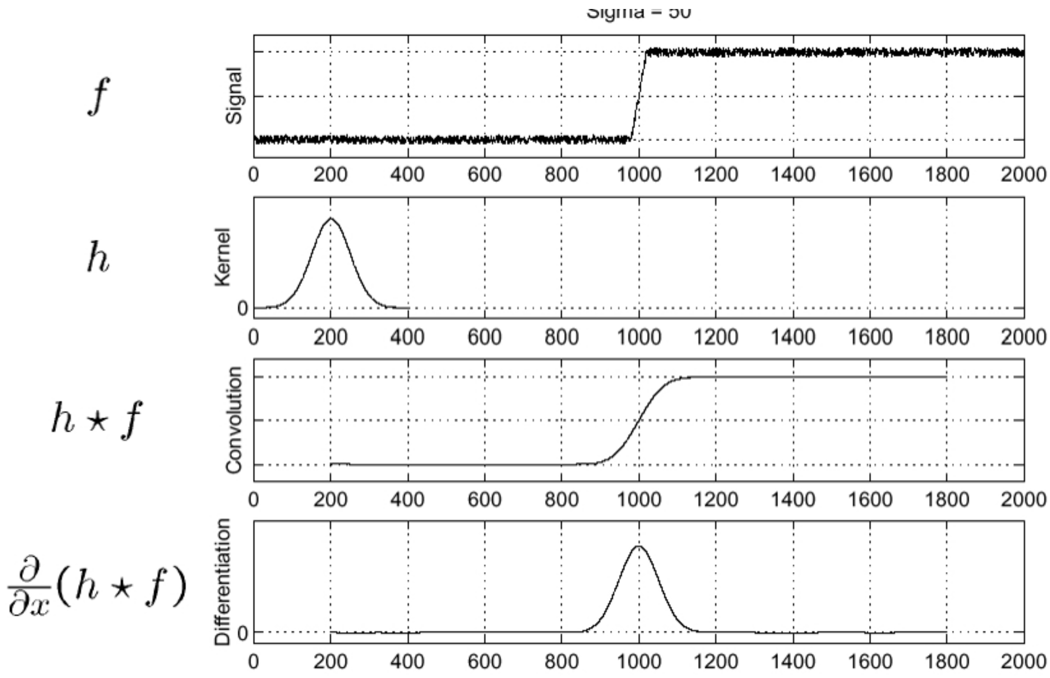
$\frac{d}{dx}f(x)$



    - 通常，会先对图像进行滤波

      这种情况下，进行高斯滤波，就是先对图像进行加权平均，噪声点滤波后的值会接近一个常数，而边缘处的值还在变化

      然后我们再利用梯度，就可以分辨出边缘



二阶导的运用 - > 拉普拉斯

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

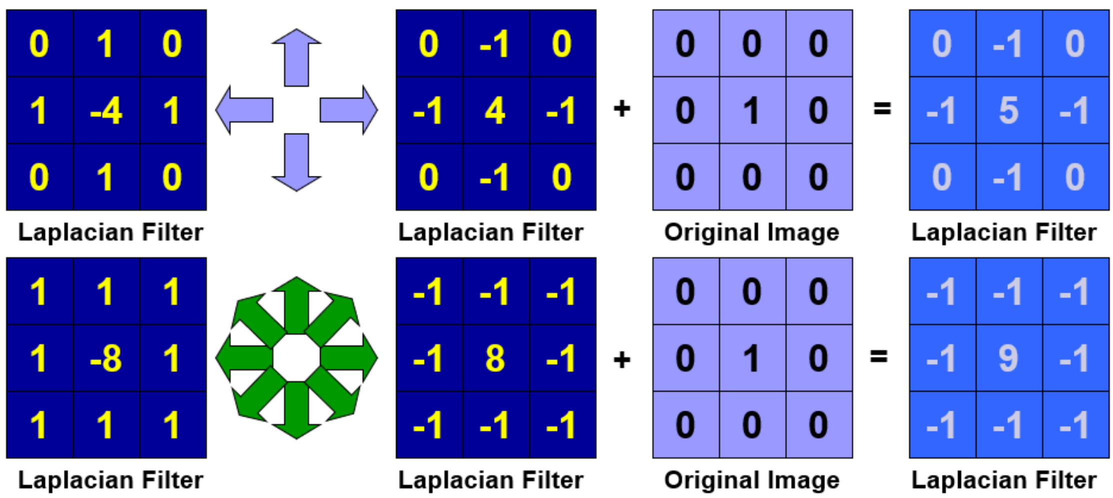
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

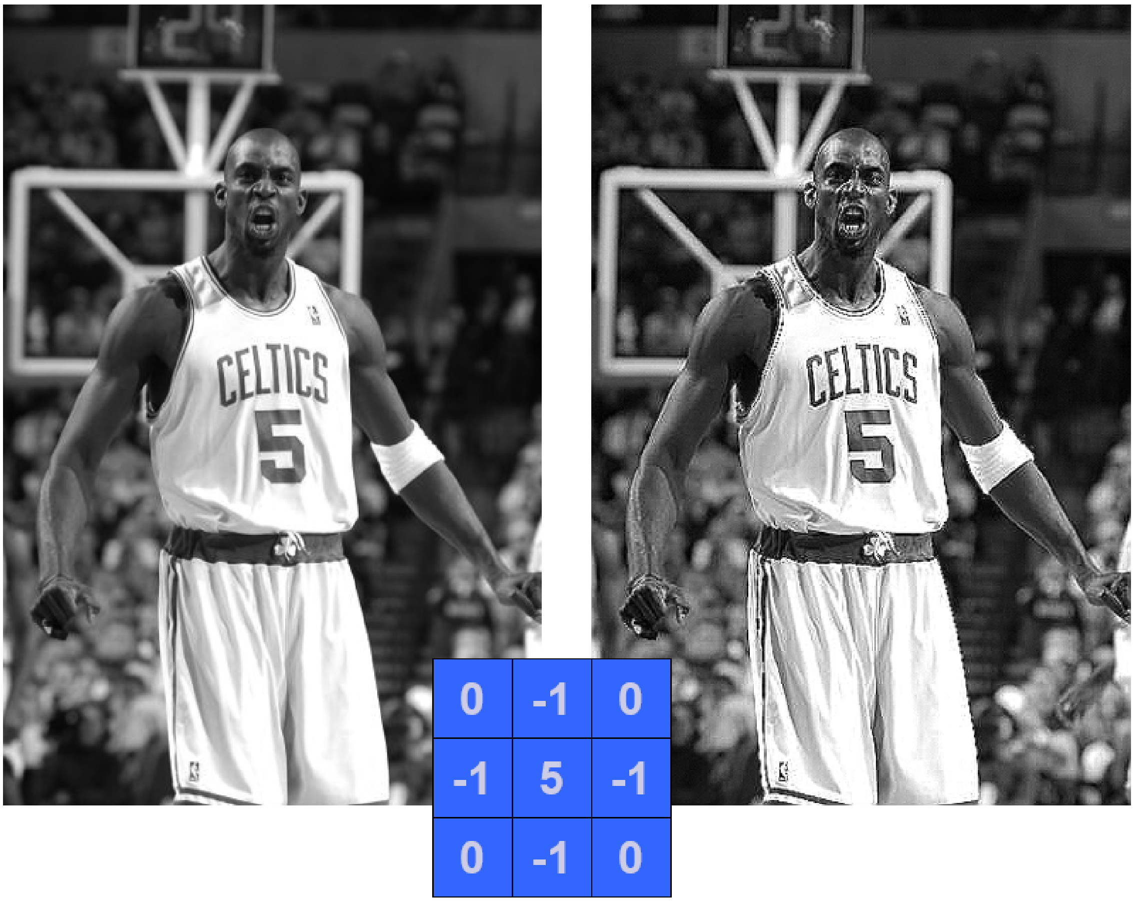
- **Discrete version 离散表示:**

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

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

- 拉普拉斯滤波核课以是中间为正数，周围是负数，也可以中间是负数，周围是正数。
- 可以通过将拉普拉斯检测出来的图像加上原图像来锐化原图像

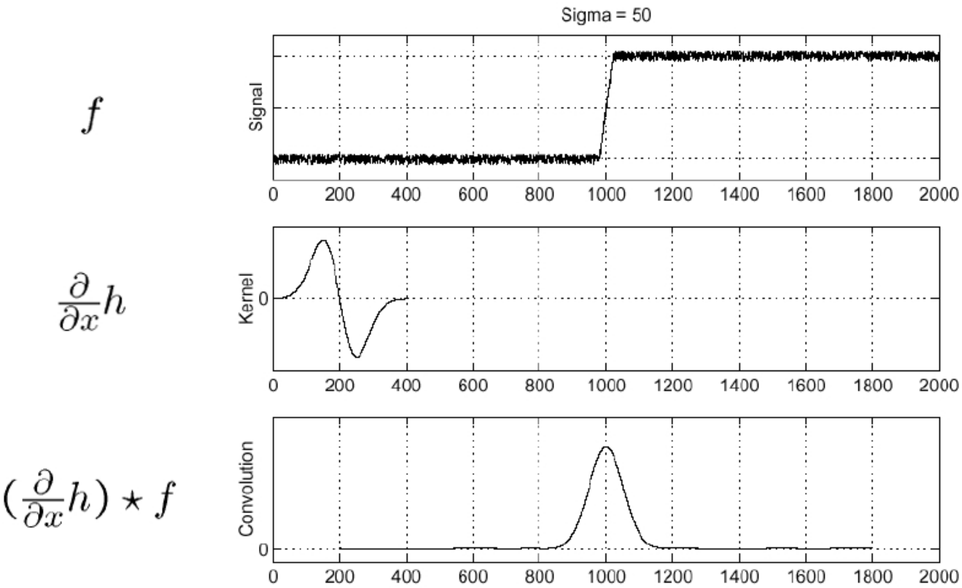




- LOG（Laplacian of Gaussian）
  - 在一阶导中，我们处理过噪音对图像的影响，就是先滤波。这里我们发现，先进行高斯滤波，再对看滤波的结果求一阶导数，和先对高斯函数求一阶导数，再滤波得到的结果一样

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- **Derivative of Gaussian.**



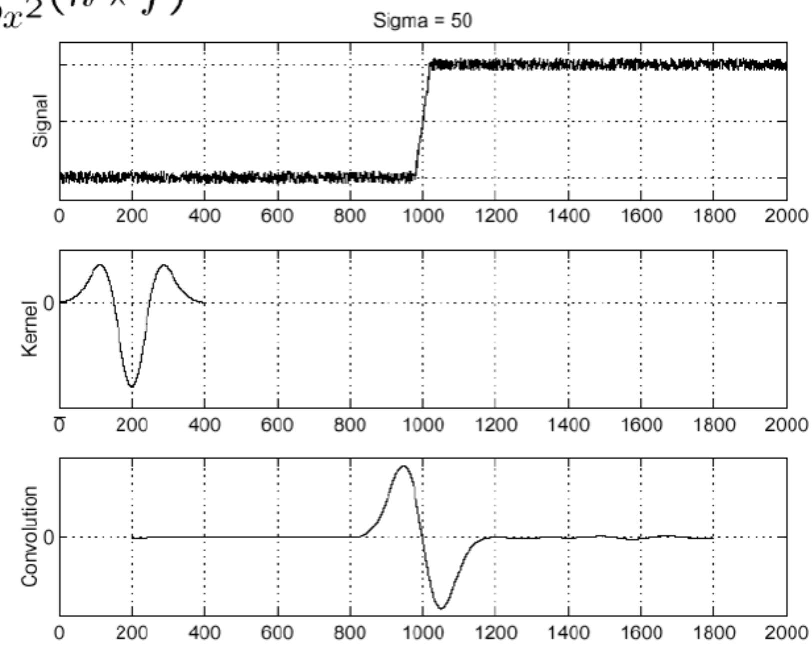
  - 在二阶导中，也有这个特性。高斯函数的二阶导就是 LOG

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

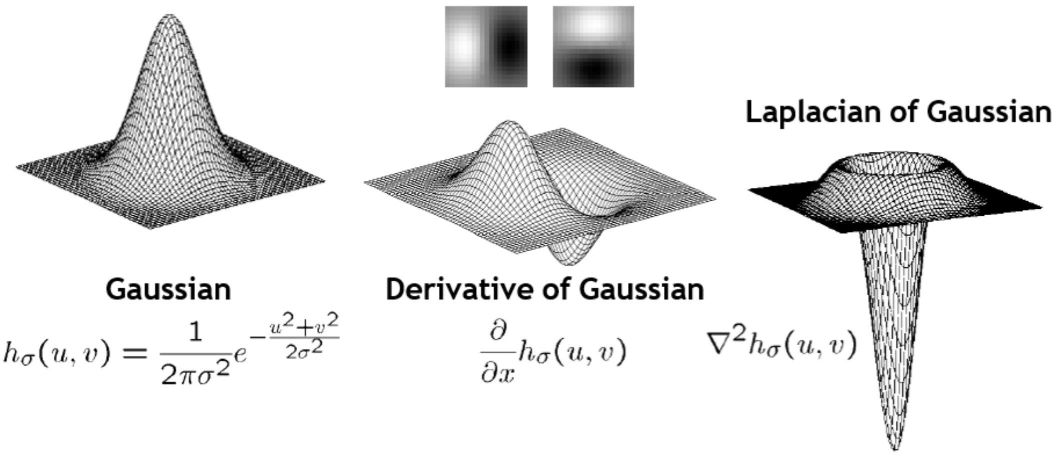
f

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

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



  - 我们可以通过图形进行直观感受



- ∇^2 **is the Laplacian operator:**

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

高频补偿滤波器

- 非锐化掩蔽

  顾名思义，就是把不是锐化（细节）的地方遮住，也就是说我们这个 mask 是用来描述图像细节的

■ **Unsharp masking:**

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

- 高频补偿滤波器

- **High-boost filtering:**

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

- 这里的 $f_s(x,y)$ 是上面的 unsharp masking
- 实际情况中，我们用矩阵表达

Original   lowpass

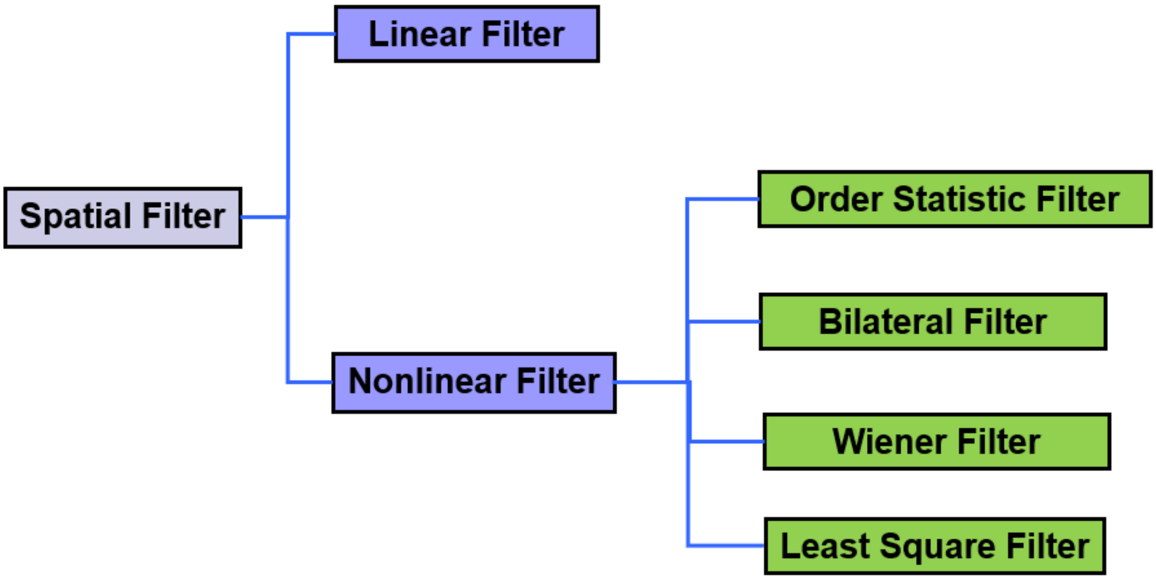
$$
A \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - 1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
$$

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

$$
= 1/9 \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad w = 9A - 1,\ A \geq 1
$$

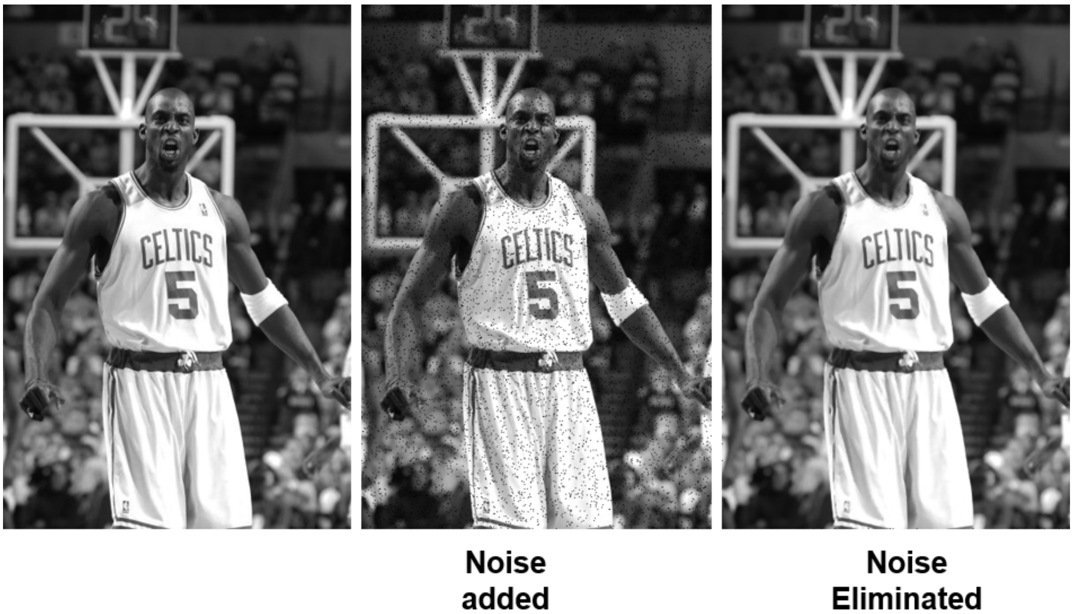
  - 可见，当$A=1$的时候，就是一个基本的高通滤波器
  - $A > 1$ 的时候，图像看起来更像原始图像，并且在一定程度上进行了细节的增强

非线性滤波器



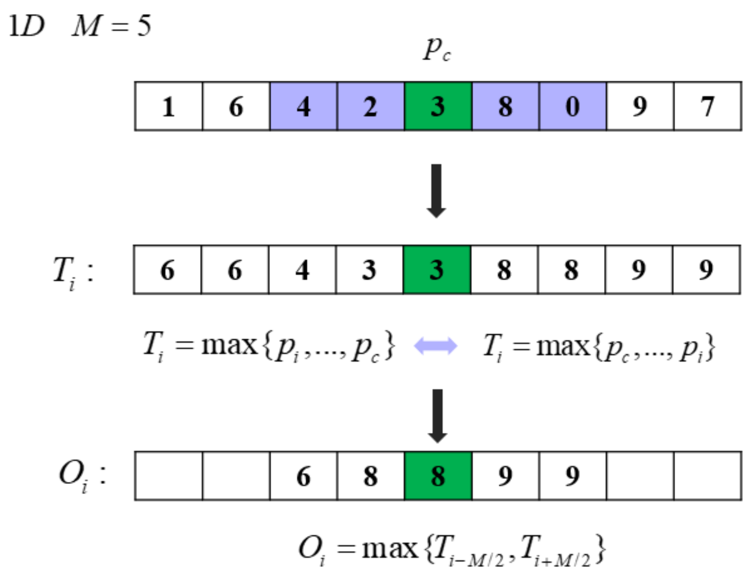
次序统计滤波器（Order Statistic Filter）

- 中值滤波：取邻域中灰度的中值作为该点的灰度值
  - 尺寸越大，图像越模糊
  - 可以用于消除椒盐噪声

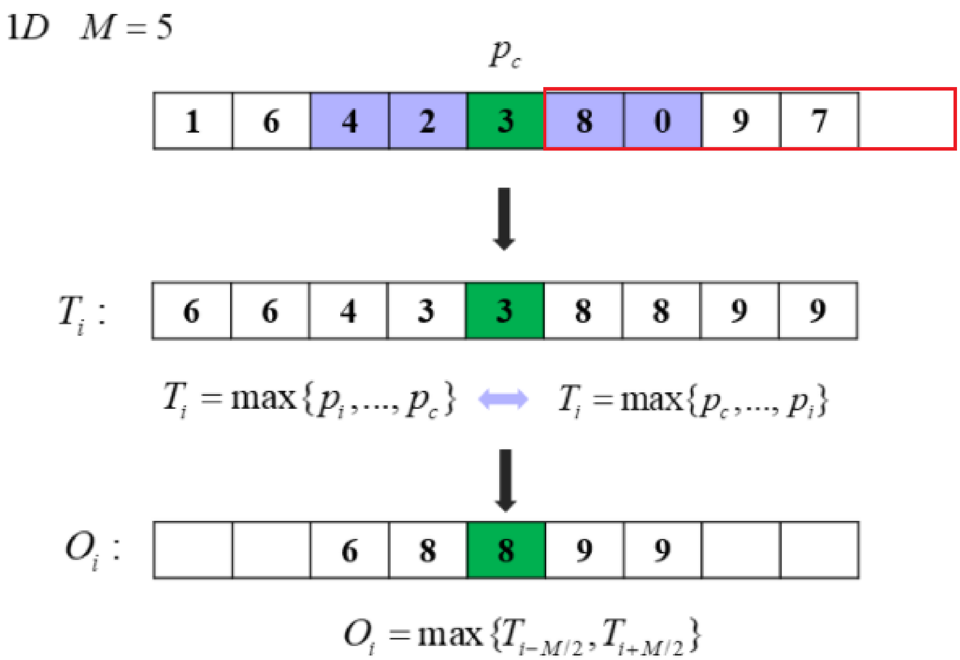


- 最大值滤波
- 最小值滤波
- 快速 最大/最小值滤波
  - 每次选取一个区域 M，将 M 的中间位置当做 c。
  - 对位置 i 来说，如果 i 在 c 左边，则位置 i 的值为从 i 到 c 的最大值。如果 i 在 c 的右边，则位置 i 的值为从 c 到 i 的最大值
  - 产生 T 后，从 c 往左往右都是递增的（单调非减）
  - 因此取最大值的时候，直接取两端中的一个

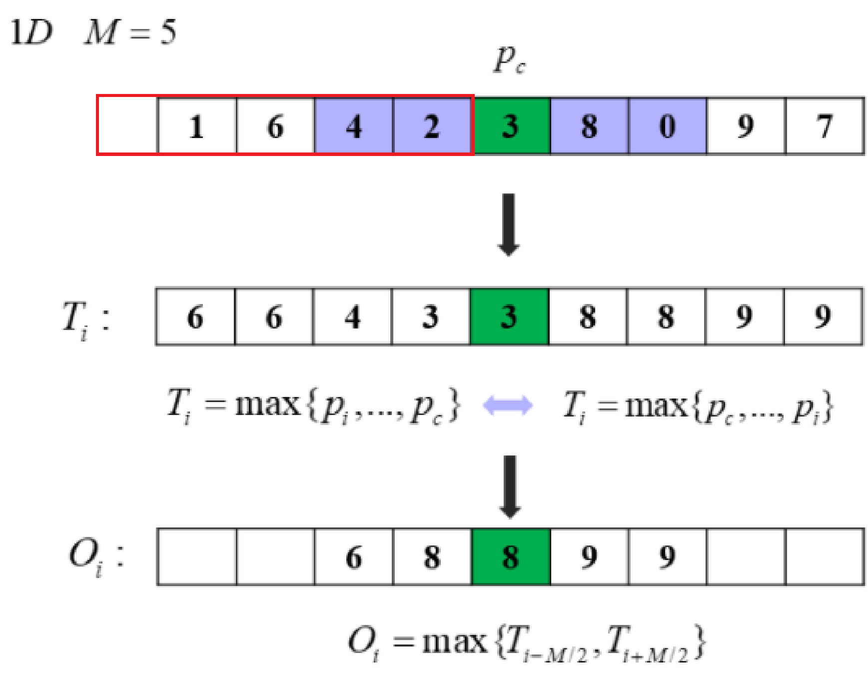
Fast Max/Min Filter



- 只能计算出紫色部分位置的最大值
  - 如果向右太多。红色部分的左右两端的最大值可能是绿色部分的值，但是这个值并不在红色框内



  - 如果向左太多，红色部分的左右两端的最大值可能是绿色部分的值，但是这个值并不在红色框内



双边滤波器（Bilateral Filter）：

- 高斯滤波的问题：在消除噪音的同时也会对图像中的边缘信息进行平滑，使图像变得模糊
    - 高斯滤波之所以会导致图像变得模糊，是因为它在滤波过程中只关注了位置信息。即在滤波窗口内，距离中心点越近的点的权重越大
    - 这种只关注距离的思想在某些情况下是可行的，例如在平坦的区域，距离越近的区域其像素分布也越相近，自然地，这些点的像素值对滤波中心点的像素值更有参考价值。但是在像素值出现跃变的边缘区域，这种方法会适得其反，损失掉有用的边缘信息
- 双边滤波：计算权重的同时考虑空间位置和像素颜色之差
    - 双边滤波的思想很简单，在高斯滤波的基础上加入了像素值权重项，也就是说既要考虑距离因素，也要考虑像素值差异的影响，像素值越相近，权重越大

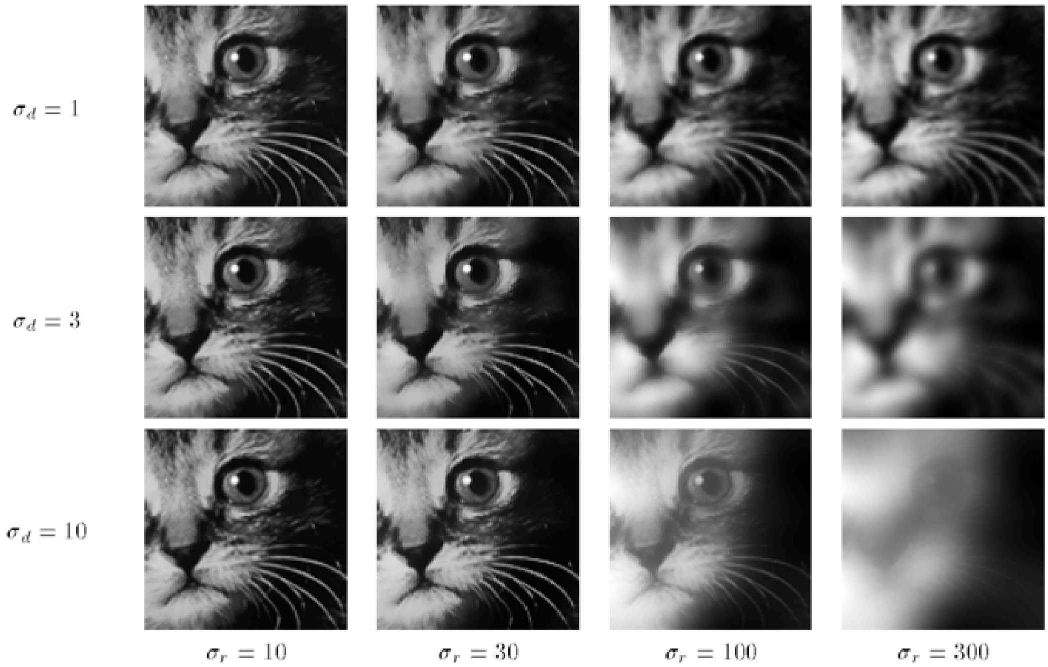
Gaussian:

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

$$\Downarrow$$

Bilateral:

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



对前半段公式来说，σ 越大，离得远的位置的权重会增大，离得近的位置的权重会减小

对后半段公式来说，σ 雨大，像素值相差大的位置的权重会增大，像素值相差小的位置的权重会减小

所以，两个 σ 越大，图像越模糊，越小，图像越清晰