

# 机器视觉与图像处理

## 第4讲 空间域图像增强

汪凯巍  
2019-03-19

部分资料取自互联网，版权归原作者所有

## 回顾

### 第1讲 绪论

机器视觉的定义、系统组成、“机器视觉是很有用的”

### 第2讲 图像的获取

图像传感器、镜头、光照，“好的图像成功一半”

### 第3讲 图像的基础变换

点处理及灰度直方图、代数变换、几何变换

### 第4讲 图像的空间域增强

## 空间域图像增强

《数字图像处理与机器视觉》第5章

### □ 4.1 图像增强基础

### □ 4.2 空间域滤波

### □ 4.3 图像平滑

### □ 4.4 中值滤波

### □ 4.5 图像锐化

关键词：滤波、掩膜、核、模板、算子、窗口、拉普拉斯、Sobel、中值

## 图像增强基础

### □为什么要图像增强

- ✓ 突出需要的信息，削弱或去除不需要的信息
- ✓ 从图像中提取信息（纹理、边缘、特征点...）
- ✓ 检测图案（模板匹配）
- ✓ 处理结果更容易被人观察或机器进行后续处理

### □空间域增强和频率域增强

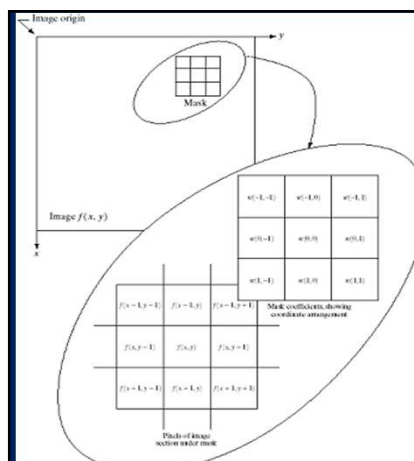
## 空间域滤波（邻域处理）

□邻域处理是指操作邻域内的图像像素值以及与邻域有相同维数的子图像的值。

□这些子图像被称为滤波器、核、模板、掩膜和窗口。

□滤波器子图像中的值是系数值，而不是像素值。

□有别于上一章的点运算等基础变换（邻域无关）以及后面的频域滤波。



空间滤波的  
机理

## 空间域滤波

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

在某一点  $(x, y)$  处，滤波器在该点的响应是滤波器的系数与滤波模板经过区域的相应像素值的乘积之和。

如图所示为  $3 \times 3$  的模板，在图像中的  $(x, y)$  点，用该模板进行线性滤波的响应  $R$  为：

$$R = w(-1, -1)f(x-1, y-1) + w(-1, 0)f(x-1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 0)f(x+1, y) + w(1, 1)f(x+1, y+1)$$

## 空间域滤波

若模板为下图所示的  $3 \times 3$  的模板

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

图像中任意一点  $(x, y)$  的响应  $R$  为：

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

## 空间域滤波

若模板为  $m \times n$  的模板，图像中任意一点  $(x, y)$  的响应  $R$  为：

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i$$

其中，通常  $m=2a+1$  且  $n=2b+1$  ( $a, b$  为非负整数)。即，模板的长和宽都为奇数，其有意义的最小尺寸为  $3 \times 3$ 。

## 空间域滤波

一般来说，在  $M \times N$  的图像  $f$  上，用  $m \times n$  大小的滤波器模板进行线性滤波由下式给出：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)$$

其中， $a=(m-1)/2$  且  $b=(n-1)/2$ 。

为了得到一幅完整的经过滤波处理的图像，必须对  $x=0, 1, 2, \dots, M-1$  和  $y=0, 1, 2, \dots, N-1$  依次应用公式。

## 空间域滤波

空间滤波3步：

- (1) **对齐**：将模板在图像中移动，模板中心与图像某个像素位置重合；
- (2) **积分**（矩阵相乘后求和）：将模板系数与模板下对应像素相乘；
- (3) **替换**：（即模板的输出响应）赋给图像中对应模板中心位置的像素。

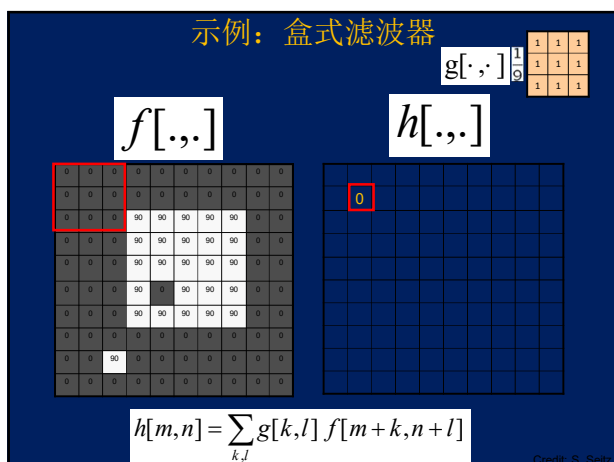
优点：

- 计算效率高，而且可以对多个像素并行处理。
- 程序统一，稍加修改即可适应不同需求。

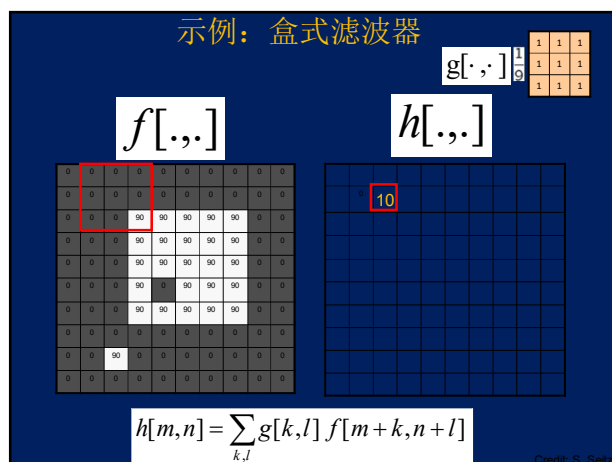
## 示例：盒式滤波器

	$g[\cdot, \cdot]$		
	1	1	1
1	1	1	1
9	1	1	1

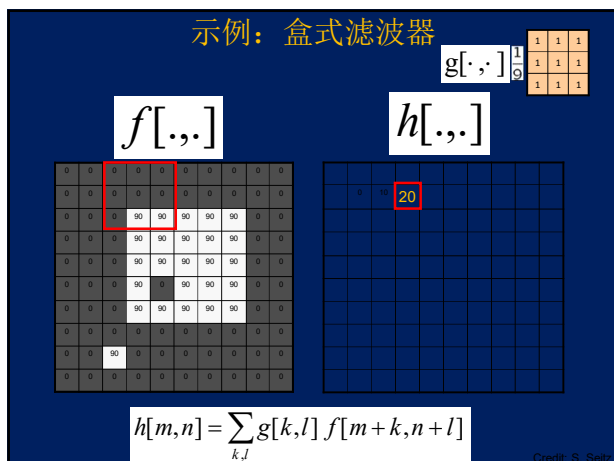
示例：盒式滤波器



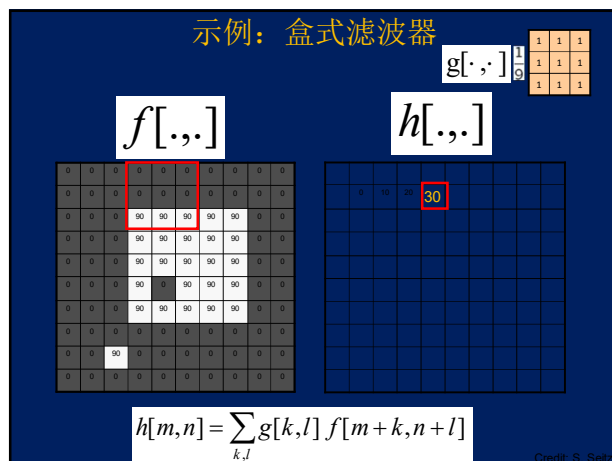
示例：盒式滤波器



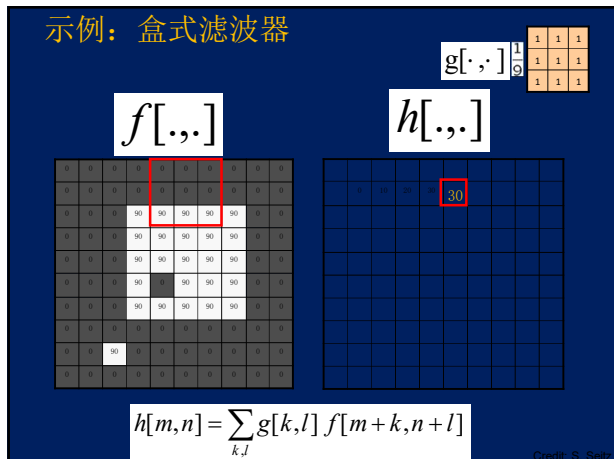
示例：盒式滤波器



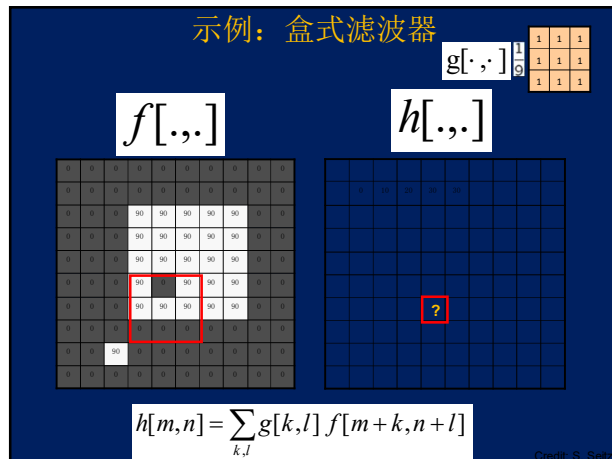
示例：盒式滤波器



示例：盒式滤波器

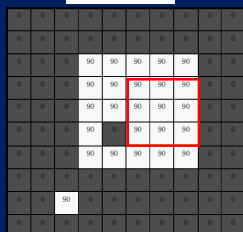
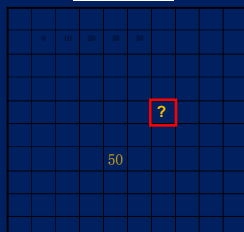


示例：盒式滤波器



## 示例：盒式滤波器

$$g[\cdot, \cdot] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

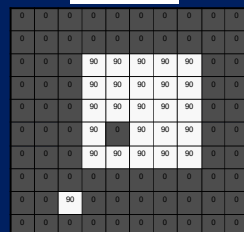
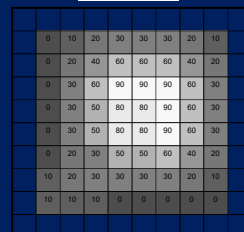
 $f[\cdot, \cdot]$ 

 $h[\cdot, \cdot]$ 


$$h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]$$

Credit: S. Seitz

## 示例：盒式滤波器

$$g[\cdot, \cdot] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

 $f[\cdot, \cdot]$ 

 $h[\cdot, \cdot]$ 


$$h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]$$

Credit: S. Seitz

## 盒式滤波器

## 作用

- 用周围像素灰度值的平局值代替当前像素灰度
- 获得平滑效果（消除陡变特征）

$$g[\cdot, \cdot]$$

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

Slide credit: David Lowe (UBC)

## 盒式滤波器用于图像平滑



## 线性滤波器练习



原图

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

核

?

Source: D. Lowe

## Practice with linear filters



Original

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

Filtered  
(no change)

Source: D. Lowe

## Practice with linear filters



Original

0	0	0
0	0	1
0	0	0

?

Source: D. Lowe

## Practice with linear filters



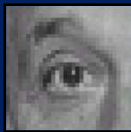
Original

0	0	0
0	0	1
0	0	0

Shifted left  
By 1 pixel

Source: D. Lowe

## Practice with linear filters



Original

0	0	0
0	2	0
0	0	0

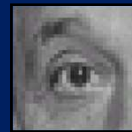
(Note that filter sums to 1)

1	1	1
1	1	1
1	1	1

?

Source: D. Lowe

## Practice with linear filters



Original

0	0	0
0	2	0
0	0	0

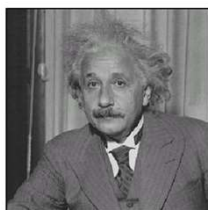
1	1	1
1	1	1
1	1	1

**Sharpening filter**

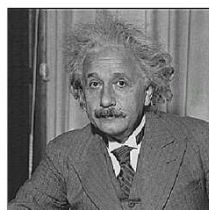
- Accentuates differences with local average

Source: D. Lowe

## Sharpening



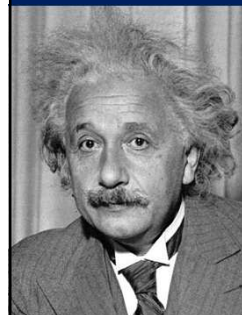
before



after

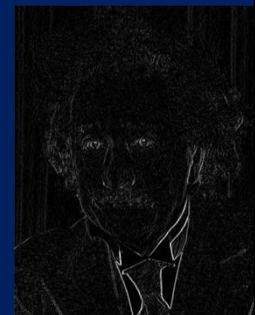
Source: D. Lowe

## Other filters

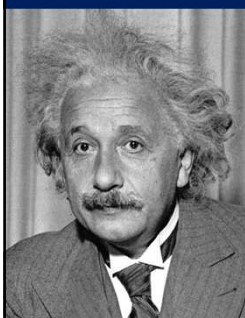


1	0	-1
2	0	-2
1	0	-1

Sobel

Vertical Edge  
(absolute value)

## Other filters



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

Sobel

Horizontal Edge  
(absolute value)

## 空间滤波的问题1——边界问题

在模板的加权或卷积运算中，存在的一些具体问题：

## ➤边界问题

如一个大小为 $n \times n$ 方形模板，当模板的中心离图像边缘为 $(n-1)/2$ 个像素时，该模板至少有一条边与图像的轮廓相重合，如果模板的中心继续向图像边缘靠近，那么模板的行或列就会处于图像平面之外。

## 空间滤波的问题1——边界问题

在模板的加权或卷积运算中，存在的一些具体问题：

## 解决方法：

- 将模板中心点的移动范围限制在距离图像边缘不小于 $(n-1)/2$ 个像素处。

例：设原始图像为 

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

，模板为 

1	1	1
1	1	1
1	1	1

，模板处理后

“—”表示无法进行模板操作的像素点。

## 空间滤波的问题1——边界问题

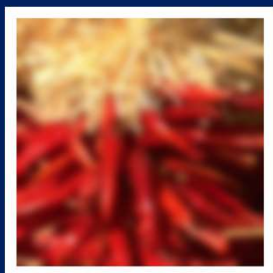
如果要求处理后的输出图像与原始图像一样大，所采用的经典方法是：

- 在图像边缘的像素带用包含于图像中的部分模板进行滤波处理。
- 在图像边缘以外再补上 $(n-1)/2$ 行或 $(n-1)/2$ 列灰度为零的像素点，或者将边缘复制补在图像之外。补上的那部分经处理后去除。

注意：但是补在靠近图像边缘的部分会带来不良影响，这种影响随着模板尺寸的增加而增大。

## Practical matters

- What about near the edge?
  - the filter window falls off the edge of the image
  - need to extrapolate
  - methods:
    - clip filter (black): 补零
    - wrap around: 缠绕
    - copy edge: 补边缘
    - reflect across edge: 边线镜像



Source: S. Marschner

## 算法

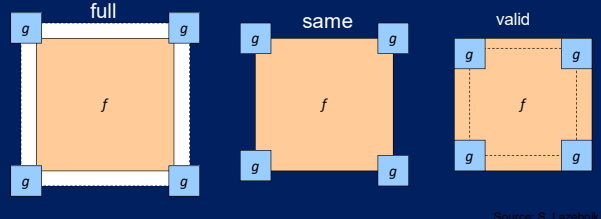
— methods (MATLAB):

- clip filter (black): `imfilter(f, g, 0)`
- wrap around: `imfilter(f, g, 'circular')`
- copy edge: `imfilter(f, g, 'replicate')`
- reflect across edge: `imfilter(f, g, 'symmetric')`

Source: S. Marschner

## 边界问题

- 输出的尺寸是多少？
- MATLAB: `filter2(g, f, shape)`，该操作总是边界补零。
  - `shape = 'full'`: output size is sum of sizes of f and g
  - `shape = 'same'`: output size is same as f
  - `shape = 'valid'`: output size is difference of sizes of f and g



Source: S. Lazebnik

空间滤波的问题2：计算出来的像素值的动态范围  
(处理后像素值的大小超出0到255的范围)

### 解决方法：

1. 如像素值的最小可能取值为-255，最大可能取值为255。对每个像素值再加255然后除2。

优点：快速简单

不足：

- 无法保证像素的取值可以覆盖0~255的全部8比特范围，即整个显示范围没有得到充分利用。
- 在除2过程中固有的结尾误差通常将导致精确度的损失。

空间滤波的问题2：计算出来的像素值的动态范围  
(处理后像素值的大小超出0到255的范围)

2. 首先提取最小值，并且把它的负值加到所有图像的像素中。然后，通过用  $255 / \text{Max}$  (其中Max为改进后图像的最大像素值) 值去乘每个像素。这样图像中的所有像素标定到0到255的范围中。

优点：具有更高的精度并使像素覆盖整个8比特的范围。

不足：比前一种更为复杂而且难以实现。

## 4.3 图像平滑

- 一幅图像可能存在着各种寄生效应。这些寄生效应可能在图像生成和传输过程中产生，也可能在量化等处理过程中产生。
- 如敏感元器件的内部噪声、感光材料的颗粒噪声、热噪声、电气机械运动产生的抖动噪声、传输信道的干扰噪声、量化噪声等。
- 原本均匀和连续变化的灰度突然变大或变小，形成虚假的物体边缘或轮廓。恶化了图像质量，使图像模糊，甚至淹没特征，给分析带来困难。
- 抑制或消除这类噪声而改善图像质量的过程称为**图像的平滑**。

## 4.3 图像平滑

目的：

- 减少图像中的噪声，以改善图像质量，有利于抽取对象特征进行分析。
- 在提取较大的目标前去除太小的细节或将目标内的小间断连接起来。

/线性平滑滤波器

/空域法

图像平滑方法

\非线性平滑滤波器

\频域法

### (一) 多幅图像平均法

如果一幅图像包含有加性噪声，这些噪声对于每个坐标点是不相关的。并且其平均值为零，在这种情况下就可能采用多图像平均法来达到去噪的目的。

设  $g(x,y)$  有噪声的图像， $\eta(x,y)$  为噪声， $f(x,y)$  为原始图像，可表示为：

$$g(x, y) = f(x, y) + \eta(x, y)$$

多图像平均法：就是把一系列有噪声的图像  $\{g_i(x,y)\}$  叠加起来，然后再取平均值以达到平滑的目的。

具体做法如下：

如果噪声符合上述限制，取 $K$ 幅内容相同但含有不同噪声的图像，将它们叠加起来，然后平均计算，如下式所示：

$$\bar{g}(x,y) = \frac{1}{K} \sum_{i=1}^K g_i(x,y)$$

则： $E\{\bar{g}(x,y)\} = f(x,y)$

$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{K} \sigma_{\eta(x,y)}^2$$

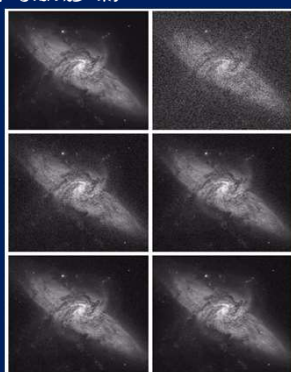
在平均图像中任意一点的标准差为：

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)}$$

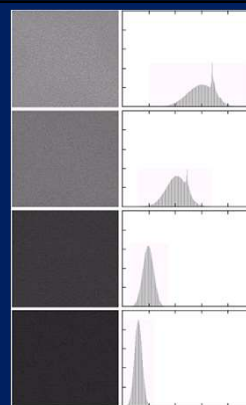
由上述二式可知，当 $K$ 增加时，像素值的方差减小，即噪声在每个像素位置 $(x,y)$ 的影响逐渐减小。

这就意味着，当作平均处理的噪声图像数量增加时，其统计平均值就越接近原始无噪声图像。

例：多图像平均法减少噪声



(a) 星系图，(b) 由0均值和64级灰度标准差的加性噪声污染了的图像，  
(c) 和(f) 是 $K=8, 16, 64$ 和128的带噪声图像取平均的结果。



(a) 从上到下：上图(a)与(c)到(f)4幅图像间的差值图像，(b) 相应的直方图

结论：

随着 $K$ 的增加差值图像的均值和标准差减小了。均值图像越来越趋近于原始图像，差值图像的灰度级越低（越暗）。

## (二) 平滑线性滤波器——邻域平均法

用模板确定的邻域内像素的平均灰度值来代替图像的每个像素点的值，因此也称**均值滤波器**。

以 $3 \times 3$ 的平滑滤波器为例

	1	1	1
1	1	1	1
9	1	1	1

则图像中任意一点的响应为：

$$R = \frac{1}{9} \sum_{i=1}^9 z_i$$

如是 $m \times n$ 的滤波器，则图像中任意一点的响应为：

$$R = \frac{1}{mn} \sum_{i=1}^{mn} z_i$$

有时也把所有系数都相等的空间均值滤波器称为**盒式滤波器**。

常见的应用：

- **减少噪声**。由于典型的随机噪声由灰度级的“尖锐”变化组成。
- **模糊处理**。在提取大目标前去除图像中一些琐碎的不相干的细节。去除细节的多少与滤波器的尺寸有关。
- **负面效应**：边缘模糊。图像边缘也是由于图像灰度的尖锐变化带来的特征，也是我们所希望有的特征。



例：图像的平滑效果与滤波器尺寸的关系



(a) 大小为500×500像素的原图像，  
(b) 用大小为n=3, 5, 9, 15和35的  
方形均值滤波模糊平滑的结果

注意：

在这处理后的图中有明显的**边界现象**。这是因为为了使处理后图像大小不变，在原始图像的边界以外补0，经处理后再去除添加区域的结果。而且滤波器越大边界越宽。

结论：

空间均值处理是为得到感兴趣物体的一个粗略的描述而模糊一幅图像。较小物体与背景混合在一起，较大物体变得像“斑点”而易于检测。而模板的大小由那些将融入背景中去的物体的尺寸决定。

### 超限邻域平均法

采用下列准则形成平滑图像，

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(m, n) \in S} f(m, n) & |f(x, y) - \frac{1}{M} \sum_{(m, n) \in S} f(m, n)| > T \\ f(x, y) & \text{其他} \end{cases}$$

其中， $S$ 表示以 $(x, y)$ 为中心的邻域点的集合， $M$ 是 $S$ 中像素的总数。 $T$ 是一个规定的非负阈值。

□ 当一个点和它的邻域内的点的灰度的平均值的差不超过阈值 $T$ 时，就仍然保留与灰度值不变

□ 如果大于阈值 $T$ 时就用它们的平均值来代替该点的灰度值。

盒式滤波器将邻域等同看待、与实际情况相符吗？

### 加权平均法

根据参与平均像素的特点赋予不同权值的方法称为**加权平均法**。

一般可以按照下列原则来确定参与平均的各像素的权值：

- (1) 待处理像素赋予较大的权值，其他像素的权值均较小。
- (2) 按照距离待处理像素的远近确定权值，距离待处理像素较近的像素赋予较大的权值。
- (3) 按照与待处理像素的灰度接近程度确定权值，与待处理像素灰度较接近的像素赋予较大的权值。

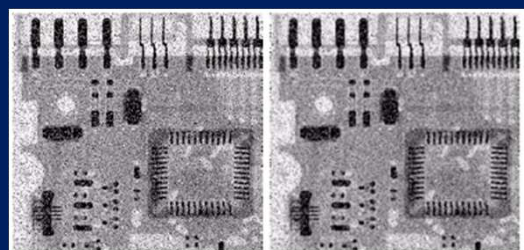
$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

为了便于计算机实现，模板中的所有系数之和通常取2的整数次幂。

一幅 $M \times N$ 的图像经过一个 $m \times n$  ( $m$ 和 $n$ 为奇数)的加权均值滤波器滤波的过程可由下式给出：

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

### 中值滤波的引出



受椒盐噪声污染的  
电路板的X光图像

经过3×3均值滤波  
器处理后的结果

还有噪声的影响！

### 中值滤波—统计排序滤波器

其响应基于图像滤波器包围的图像区域中像素灰度值的排列，然后由统计排序结果决定的值代替中心像素。

**优点：**

- 在一定条件下，可以克服线性滤波器（如邻域平滑滤波等）所带来的图像细节模糊
- 对滤除脉冲干扰及图像扫描噪声最为有效。

**不足：**

对一些细节多，特别是点、线、尖顶细节多的图像不宜采用中值滤波。

### 中值滤波3步法：

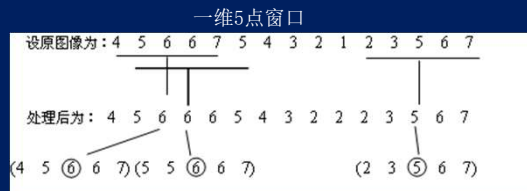
- ① 先将预求的像素及其邻域内的像素灰度值排序
- ② 确定出中值
- ③ 并将中值赋予该像素点。

**例：**选择滤波用的窗口W如图，是一个一维的窗口，待处理像素的灰度取这个模板中灰度的中值，滤波过程为：

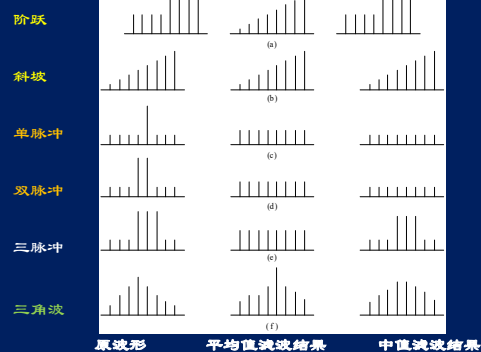
m-2	m-1	m	m+1	m+2
-----	-----	---	-----	-----

### 中值滤波3步法：

- ① 先将预求的像素及其邻域内的像素灰度值排序
- ② 确定出中值
- ③ 并将中值赋予该像素点。

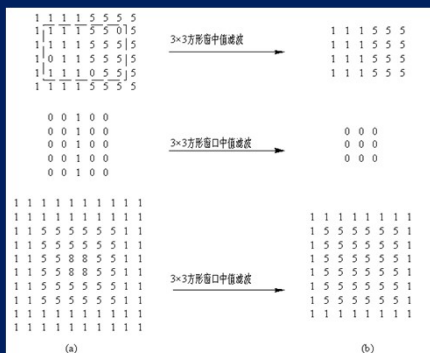


### 例：中值滤波和平均值滤波比较(模板为5个像素)



中值滤波器不影响阶跃函数和斜坡函数。周期小于窗口的一半的脉冲受到抑制，三角函数的顶部变平。

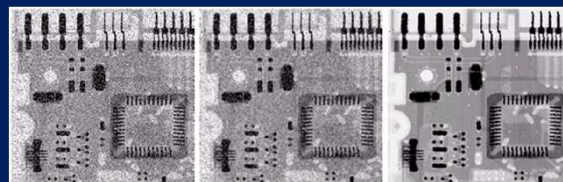
### 例：二维图像的中值滤波



原始图像

中值滤波输出

### 例：利用中值滤波器去噪



受椒盐噪声污染的电路板的X光图像

经过3×3均值滤波器处理后的结果

经过3×3中值滤波器处理后的结果

**比较：**

中值滤波器比均值滤波器更适合去除椒盐（脉冲）噪声。

结论：

- 如果图像中点、线、尖角**细节较多**，则不宜采用中值滤波。
- 可用 $n \times n$ 的中值滤波器去除那些相对于其邻域像素更亮或更暗的区域，并且其区域是小于 $n^2/2$ （滤波器区域的一半）的孤立像素集。
- 特别对处理**椒盐（脉冲）噪声**非常有效。

中值滤波器的使用非常普遍，这是因为对于一定类型的随机噪声，它有很强的去噪能力。

## 4.5 图像锐化

在图像获取、传输及处理过程中有许多因素会使图像变得模糊。

消减图像模糊的图像增强方法称为**图像锐化**。

**目的：**突出图像中的细节或增强被模糊了的细节。

**注意：**进行锐化处理的图像必须要有较高的信噪比，否则，图像进行锐化以后，信噪比更低图像质量急剧下降。另外，由于锐化将使噪声受到比信号还强的增强，故必须小心处理。一般都是先进行图像平滑，**去除或减轻图像中的干扰噪声**，然后才能进行锐化处理。

## 4.5 图像锐化

**基于一阶和二阶微分的细节锐化滤波器：**

我们最感兴趣的微分性质是在**恒定灰度区域**（平坦段）、**突变的开头与结尾**（阶梯和斜坡突变）及**沿着灰度级斜坡处**的特性。

这些类型的突变可以用来对应图像中的噪声点、细线与边缘模型化。所以，在向（或从）这些图像特性过渡时的微分性质很重要。

对于一阶和二阶微分的任何定义都必须满足以下几点：

- (1) 在平坦段（灰度不变的区域）微分值为零；
- (2) 在灰度阶梯或斜坡的起点处微分值非零；
- (3) 沿斜坡面微分值非零。

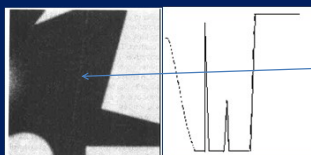
对于一元函数  $f(x)$  的一阶微分的差分形式：

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

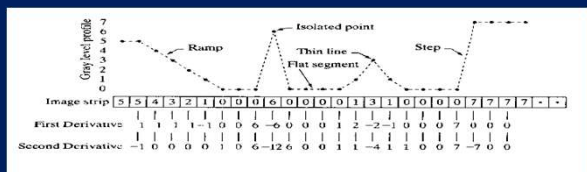
类似地，二阶微分的差分形式：

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

例：图像处理中一阶和二阶微分间的相同及不同点



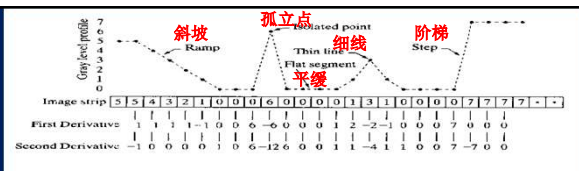
说明：斜坡的过渡包含四个像素，噪声点是一个单一像素，线有三个像素粗，而灰度阶梯的过渡变化在相邻像素间发生。灰度及数目简化为只有8个等级。



(a) 一幅简单图像，

(b) 沿图像中心并且包含孤立噪声点的一维水平灰度剖面图，

(c) 简化的剖面图(用虚线连接点以简化描述)



通过比较一阶微分处理与二阶微分处理的响应，得出以下结论：

- (1) 一阶微分处理通常会产生较宽的边缘；
- (2) 二阶微分处理对**细节**有较强的响应，如细线和孤立点；
- (3) 一阶微分处理一般对**灰度阶梯（突变）**有较强的响应；
- (4) 二阶微分处理对**灰度级阶梯变化**产生双响应。

大多数应用中，对图像增强来说，二阶微分处理比一阶微分好一些，因为形成增强细节的能力好一些。一阶微分在图像处理中主要用于边缘提取，但在图像增强中也起着很大作用。

### 4.5 图像锐化

由于需要锐化的图像的边界或线条可能是任意走向的，所以期望采用的算子应该是各向同性的。

**各向同性**：指无论边界或线条走向如何，只要幅度相等，算子就给出相同的输出。

#### (一) 基于一阶微分的图像增强—梯度法

对于图像函数 $f(x, y)$ ，它在点 $(x, y)$ 处的梯度是一个矢量，定义为：

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

### 4.5 图像锐化

梯度的两个重要性质：

- (1) 梯度的方向在函数 $f(x, y)$ 最大变化率的方向上。
- (2) 梯度的幅度（或模）定义为：

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

等于 $f(x, y)$ 在其最大变化率方向上的单位距离所增加的量。

### 4.5 图像锐化

□ 偏导不是各向同性的，而 $|\nabla f|$ 永远是正值，是各向同性的。

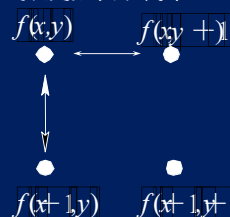
□ 在后续讨论中将笼统地将梯度的幅度称为梯度。

对于数字图像而言，微分运算一般用差分来代替，即

$$|\nabla f| = \{ [f(x, y) - f(x+1, y)]^2 + [f(x, y) - f(x, y+1)]^2 \}^{\frac{1}{2}}$$

$$|\nabla f| = \{ [f(x, y) - f(x+1, y)]^2 + [f(x, y) - f(x, y+1)]^2 \}^{\frac{1}{2}}$$

式中各像素的位置如下图所示：

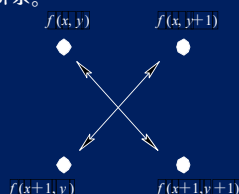


用计算机计算梯度时，通常用绝对值运算代替上式，上式可简化为

$$|\nabla f| = |f(x, y) - f(x+1, y)| + |f(x, y) - f(x, y+1)|$$

这种梯度法又称为**水平垂直差分法**。

另一种梯度法叫做**罗伯特梯度法** (Robert Gradient)，它是一种**交叉差分** (Robert cross Gradient operator) 计算法，如下图所示。



其数学表达式为

$$|\nabla f| = \{ [f(x, y) - f(x+1, y+1)]^2 + [f(x+1, y) - f(x, y+1)]^2 \}^{\frac{1}{2}}$$

同样可近似为

$$|\nabla f| = |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$

### 4.5 图像锐化

由梯度的计算可知，图像中灰度变化较大的边缘区域其梯度值大，在灰度变化平缓的区域其梯度值较小，而在灰度均匀区域其梯度值为零。

例：



(a) 二值图像； (b) 采用水平垂直差分法锐化的结果

梯度法可以表示成模板的形式

$$\nabla f = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \text{ 可用绝对值代替, 即}$$

$$\nabla f = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

$$\text{式中: } \frac{\partial f}{\partial x} = f(x+1, y+1) - f(x, y) \quad \frac{\partial f}{\partial y} = f(x+1, y) - f(x, y+1)$$

可通过下面的模板实现:

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

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

——Roberts交叉梯度算子

奇数尺寸的模板通常更受欢迎-Sobel 算子

Sobel算子法的基本原理 (使用奇数尺寸3×3模板):

$$\begin{array}{ccc} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{array}$$

Sobel算子图像坐标

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

第三行与第一行间的差值接近于x方向上的微分

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

奇数尺寸的模板通常更受欢迎-Sobel 算子

Sobel算子法的基本原理 (使用奇数尺寸3×3模板):

$$\begin{array}{ccc} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{array}$$

Sobel算子图像坐标

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

第三列与第一列间的差值接近于y方向上的微分。

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

可通过下面的模板实现:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

——Sobel算子

Sobel算子不像普通梯度算子那样用两个像素的差值, 具有以下四个优点:

- (1) 由于引入了平均因素, 因而对图像中的随机噪声有一定的平滑作用。
- (2) 由于它是相隔两行或两列之差, 故边缘两侧元素得到了增强, 边缘显得粗而亮。
- (3) 权重为2, 通过突出中心点的作用而达到平滑目的。
- (4) 系数总和为0, 表明灰度恒定区域的响应为0。

采用梯度微分锐化图像, 同时会使噪声、条纹等得到增强, Sobel算子则在一定程度上克服了这个问题。

## 梯度增强图像

梯度计算完之后, 可以根据需要生成不同的**梯度增强图像或锐化后图像**:

·**第一种**是使各点的灰度 $g(x, y)$ 等于该点的梯度幅度, 即

$$g(x, y) = |\nabla f(x, y)|$$

缺点是增强的图像仅显示灰度变化比较陡的边缘轮廓, 而灰度变化平缓的区域则呈黑色。

·**第二种**增强的图像是使

$$g(x, y) = \begin{cases} \nabla f(x, y) & \nabla f(x, y) \geq T \\ f(x, y) & \text{其他} \end{cases}$$

式中:  $T$ 是一个非负的阈值。

适当选取 $T$ , 即可使明显的边缘轮廓得到突出, 又不会破坏原灰度变化比较平缓的背景。

·**第三种**增强图像是使

$$g(x, y) = \begin{cases} L_G & \nabla f(x, y) \geq T \\ f(x, y) & \text{其他} \end{cases}$$

式中:  $L_G$  是根据需要指定的一个灰度级。

它将明显边缘用一个固定的灰度级 $L_G$ 来表征。

·**第四种**增强图像是使

$$g(x, y) = \begin{cases} \nabla f(x, y) & \nabla f(x, y) \geq T \\ L_B & \text{其他} \end{cases}$$

此法将背景用一个固定灰度级 $L_B$ 来实现, 便于研究边缘灰度的变化。

第五种增强图像是使

$$g(x, y) = \begin{cases} L_G & \nabla f(x, y) \geq T \\ L_b & \text{其他} \end{cases}$$

此法将背景和边缘用二值图像表示，便于研究边缘所在位置。

## (二) 基于二阶微分的图像增强——拉普拉斯算子

拉普拉斯 (Laplacian) 算子是常用的线性二次微分边缘增强算子，具有各向同性。其边缘增强程度与边缘的方向无关，能突出增强图像中的孤立点、孤立线或线端点。

对于连续图像  $f(x, y)$ ，其拉普拉斯算子为

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

对数字图像  $f(x, y)$ ，其拉普拉斯算子为

$$\nabla^2 f = \Delta_x^2 f(x, y) + \Delta_y^2 f(x, y)$$

式中，

$$\begin{aligned} \frac{\partial^2 f(x, y)}{\partial x^2} &= \nabla_x [\nabla_x f(x, y)] = \nabla_x f(x+1, y) - \nabla_x f(x, y) \\ &= [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y) \end{aligned}$$

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

则拉普拉斯算子  $\nabla^2 f$  为

$$\begin{aligned} \nabla^2 f &= \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \\ &= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \end{aligned}$$

可见，数字图像在  $(x, y)$  点的拉普拉斯算子，可以由  $(x, y)$  点灰度值减去该点邻域平均灰度值来求得。

## 拉普拉斯算子的成模板形式

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

拉普拉斯模板

例：



(a) 二值图像； (b) 拉普拉斯锐化结果

- 拉普拉斯是一种微分算子，强调图像中灰度的突变及降低灰度慢变化的区域。
- 产生一幅把图像中的浅灰色边线和突变点叠加到暗背景中的图像。

## 梯度算子与拉普拉斯算子比较

- 同梯度算子进行锐化一样，拉普拉斯算子也增强了图像的噪声。
- 但与梯度法相比，拉普拉斯算子对噪声的作用较梯度法强。
- 故用拉普拉斯算子进行边缘检测时，有必要先对图像进行平滑处理。

## 拉普拉斯图像增强

将原始图像和拉普拉斯图像叠加在一起，可以保护拉普拉斯锐化处理的效果，同时又能复原背景信息。

表示为下式：

$$g = f \pm \nabla^2 f$$

式中： $f$ 、 $g$ 分别为锐化前后的图像。

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

即首先计算拉普拉斯过滤图像，然后从原始图像减掉该图像的过程是两种线性处理之差。可通过一个模板来一次实现。

相应的模板为

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

实际中还常用到如下的拉普拉斯模板：

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

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

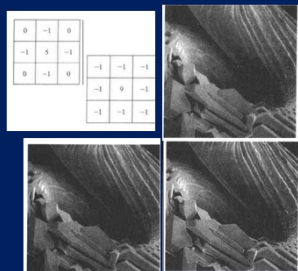
例：拉普拉斯算子的图像锐化



(a) 月球北面的图像, (b) 拉普拉斯滤波后的图像  
(c) 叠加后的拉普拉斯图像, (d) 叠加原图像后的图像

- ☐ 将原始图像加到拉普拉斯的处理结果中去就可以使图像中的各灰度是得到复原
- ☐ 通过拉普拉斯变换增强了图像中灰度突变处的对比度。
- ☐ 图像中小的细节部分得到增强并良好保留了图像的背景色调。

例：使用复合拉普拉斯模板锐化图像



(a) 损坏的钩丝的扫描电子显微镜图像(放大250倍),  
(b)和(c)为分别应用两种合成掩模滤波的结果

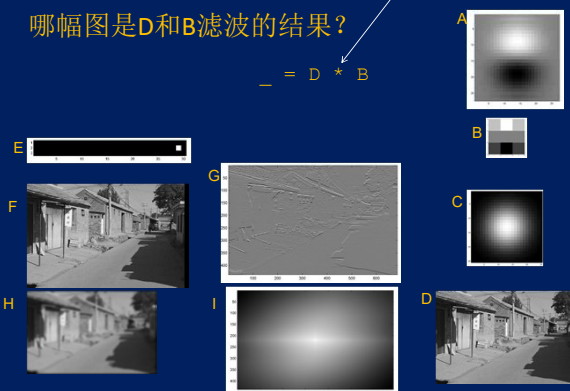
与原始图像比较,可以看出这两种模板都产生了有效的增强效果,但图(c)的结果锐化的更明显。

练习

哪幅图是D和B滤波的结果?

$$- = D * B$$

Filtering Operator

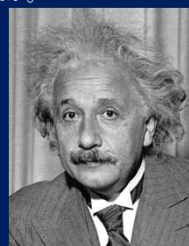


讨论

1. 分别应用 $n=23$ 、25和45的方形均值掩膜处理下面一幅图像。结果发现当 $n=23$ 、45时,处理后图像中左下角的垂直竖条被模糊了,但是竖条与竖条之间的分割仍然很清楚。当 $n=25$ 时,竖条却已经融入了整幅图像,尽管产生这幅图像的掩膜比45小得多,请解释这一现象。



图1注:垂直线段是5个像素宽,100个像素高;它们的间隔是20个像素。

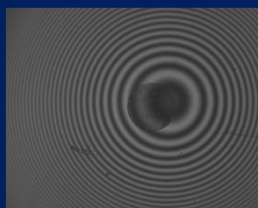


2. 能否用空间滤波的方式检测图像中的眼睛所在位置?

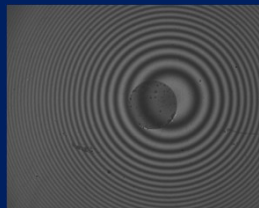


## 第一次作业

1. 以下为两幅相差为 $180^\circ$  的干涉图，用计算机找出中心低反射圆环（光纤）的中心坐标

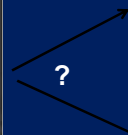


$$I_0 = a + b\cos(\varphi)$$



$$I_2 = a + b\cos(\varphi + \pi)$$

2. 以真实图像为“原材料”，用计算机生成一副有趣奇怪的图像



Slide: Hoiem