1. 空间滤波

之前说过，空间域即为图像像素的域，是对图像像素直接进行处理的，空间滤波即通过一个邻域对每个图像像素进行处理。

邻域处理包括:

(1)定义中心点(x,y);

(2)仅对预先定义的以(x, y)为中心点的邻域内的像素进行运算;

(3)令运算结果为该点处处理的响应;

(4)对图像中的每一-点重复此步骤。移动中心点会产生新的邻城，而每个邻域对应于输入图像上的一个像素。

对图像像素进行邻域处理就是空间滤波

1. 线性空间滤波

若对邻域中像素的计算为线性计算，则此运算称为线性空间滤波（空间卷积）。

线性空间滤波的运算为将邻域中每个像素与相应的系数相乘，然后将结果进行累加，从而

得到点(x. y)处的响应。

若邻域的大小为m x n,则总共需要,mn个系数。这些系数排列为一个矩阵，我们称其为滤波器、掩模、滤波掩模、核、模板或窗口。

掩膜一般为奇数的原因：奇数掩膜只有一个中心点。

相关是把掩膜旋转180°，再进行线性滤波操作，

函数 imfilter 实现线性空间滤波

g = imfilter(f, w, filtering\_mode, boundary\_options, size\_options)

f 输入图像

w 滤波掩膜

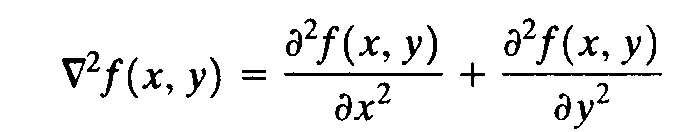
filtering\_mode 相关“corr”还是卷积“conv”

boundary\_options 处理边界零填充问题，”same”或”full”

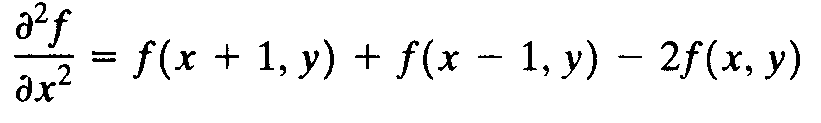
1. 非线性空间滤波

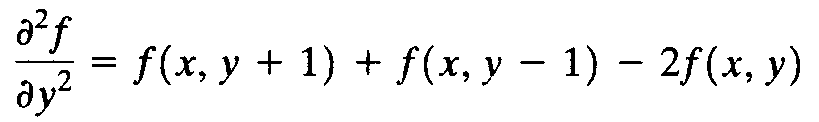
即不是用线性计算邻域内的像素，而是非线性运算，如中值滤波，取邻域内的中值灰度作为中间像素的响应值。

1. 线性空间滤波举例：拉普拉斯滤波器

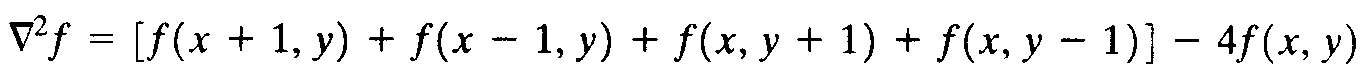


该二阶导数的通用数字近似为：

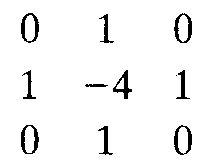




因而有



即可求得空间掩膜为：



另一种更常见的拉普拉斯算子掩膜为:

