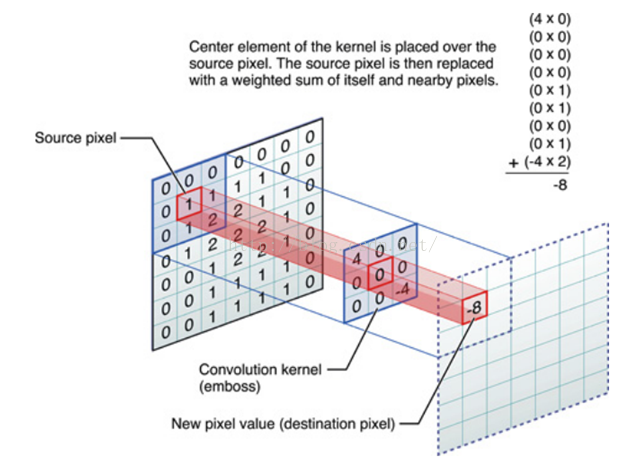
**OpenCV 入门 图像处理**

**图像处理在现实中的应用非常普遍，例如对物体图像进行识别的各种实际应用中，几乎都必须预先对图像进行各种处理（如滤波，开闭操作等），然后才能进行有效设别，所以深入学习图像的处理的基础，以及各种算法的原理和实现非常有必要。**

# 从卷积 - convolution开始

卷积一般用来线性的滤除信号，以下举个例子,用一个滤波矩阵，也叫卷积核，来对源图像进行卷积，得到一个新的图像。对于图像的每一个像素点，计算它的邻域像素和滤波器矩阵的对应元素的乘积，然后加起来，作为该像素位置的值。



# 线性滤波：方框滤波、均值滤波、高斯滤波

## 平滑处理

即模糊处理，例如减少图像上的噪点或失真，还有就时降低图像分辨率。

## 图像滤波与滤波器

**图像滤波相关的概念简介：**

图像滤波时对图像的噪声的抑制，但是尽量保留细节。

消除图像的噪声成分叫做图像的平滑或滤波操作。信号或是图像的能量大部分集中在幅度频的低中频段，在高频段，有用的信息经常被噪声淹没。

图像滤波的目的:

* 抽取对象的特征作为图像识别的特征模式
* 消除图像数字化时所混入的噪声

图像处理的要求：

* 不能损坏图像的轮廓及边缘等重要信息
* 使图像处理清晰视觉效果好

平滑滤波的目的：

* 模糊
* 消除噪声

空间域平滑滤波技术：

简单平均法，即求邻近的像元点的平均亮度值。

邻域的大小选择：越大平滑的效果越好，过大则会使边缘信息损失越大，输出图像变得模糊。

**滤波器种类简介：**

* 方框滤波 – BoxBlur函数
* 均值滤波（邻域平均滤波）- BoxBulr函数
* 高斯滤波 - GaussianBlur
* 中值滤波 – medianBlur 函数
* 双边滤波 – bilateraFilter 函数
* Gabor滤波器 （）

原理参考小魏博客: <http://blog.csdn.net/xiaowei_cqu/article/details/24745945>

* Schmid滤波器

原理参考小魏博客:<http://blog.csdn.net/xiaowei_cqu/article/details/25131473>

## 线性滤波器的简介

线性滤波器用于剔除输入信号中不想要的频率或者从许多频率中选择一个想要的频率。

* 低通滤波器：允许低频率通过
* 高通滤波器：允许高频率通过
* 带通滤波器：允许高频率通过
* 带阻滤波器：阻止一定范围频率通过
* 全通滤波器：允许所有频率通过并且允许其他频率通过
* 陷波滤波器（Band-Stop Filter）：阻止一个狭窄频率范围通过，是一种特殊带阻滤波器。

## 邻域算子与线性邻域滤波

邻域算子：利用给定像素周围的像素值决定此像素的最终输出值的一种算子。

线性邻域滤波 – 一种常用的邻域算子，像素的输出取决于输入像素的加权和。

邻域算子的作用：

* 局部色调调整
* 图像滤波

线性滤波处理的输出像素值是输入像素值的加权和：

其中就是“核”，也就时滤波器的加权系数，即滤波器的滤波系数。

## 方框滤波 – boxFilter函数

下面介绍OpenCV中的方框滤波的方法：

C++: void boxFilter(InputArray src, OutputArray dst, int ddepth, Size ksize, Point anchor=Point(-1,-1),  
bool normalize=true, int borderType=BORDER\_DEFAULT )

Parameters：src – input image.  
dst – output image of the same size and type as src.  
ddepth – the output image depth (-1 to use src.depth()).  
ksize – blurring kernel size.  
anchor – anchor point; default value Point(-1,-1) means that the anchor is at the kernel  
center.  
normalize – flag, specifying whether the kernel is normalized by its area or not.  
borderType – border mode used to extrapolate pixels outside of the image.  
The function smoothes an image using the kernel:

when normalize=true  
1 otherwise

Unnormalized (非归一化) box filter is useful for computing various integral characteristics(积性特性) over each pixel neighborhood, such as covariance matrices of image derivatives (图像导数的协方差矩阵) (used in dense optical flow algorithms – 密集光流算法, and so on). If you need to compute pixel sums over variable-size windows, use integral().

## 均值滤波 – blur函数

即输出图像的每一个像素时核窗口内输入图像对应像素的平均值 – 所有像素的加权系数相等。

均值滤波的理论基础：

它时线性滤波的一种，方法是邻域平均法。

就是用一片图像区域的各个像素的均值来代替原图像的各个像素值。

邻域平均法的具体实现：

对待处理的像素点（x，y），选择一个模板，模板由其邻近的若干个像素组成，求模板中所有像素的均值，再把该均值赋予当前像素点作为处理后图像在该点上的灰度点g（x，y）。

其中m为模板中包含当前像素在内的像素总和。

均值滤波的缺陷：

它不能很好的保护图像细节，去噪声的同时也破坏了部分细节。

函数原型：

blur函数内部对boxFilter函数进行了调用。

Blurs an image using the normalized box filter.  
void **blur**(InputArray **src**, OutputArray **dst**, Size **ksize**, Point **anchor**=Point(-1,-1), int **borderType**=BORDER\_DEFAULT )  
  
**Parameters  
src** – input image; it can have any number of channels, which are processed independently,  
but the depth should be CV\_8U, CV\_16U, CV\_16S, CV\_32F or CV\_64F.  
**dst** – output image of the same size and type as src.  
**ksize** – blurring kernel size.  
**anchor** – anchor point; default value Point(-1,-1) means that the anchor is at the kernel  
center.  
**borderType** – border mode used to extrapolate pixels outside of the image.  
The function smoothes an image using the kernel:

The call blur(src, dst, ksize, anchor, borderType) is equivalent to boxFilter(src, dst, src.type(),  
anchor, true, borderType) .

## 高斯滤波

高斯滤波也是线性滤波，消除高斯噪声，是应用范围较广的一种滤波器，比如在车牌设别的系统中运用高斯滤波。