# 1编程框架

#include <iostream>

#include <opencv2\opencv.hpp>

using namespace std;

using namespace cv;

int main()

{

return 0;

}

# 2滑块创建

#include <iostream>

#include <opencv2\opencv.hpp>

using namespace cv;

using namespace std;

int i=1;//i表示滑块的当前值

int maxvalue=10;

void change(int ,void \*);//函数声明

int main()

{

namedWindow("show");

createTrackbar("数字","show",&i,maxvalue,change);

waitKey(0);

return 0;

}

void change(int ,void \*)

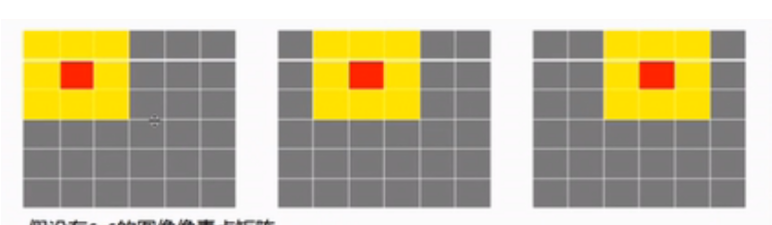
{

cout<<i<<endl;

}

# 3卷积学习

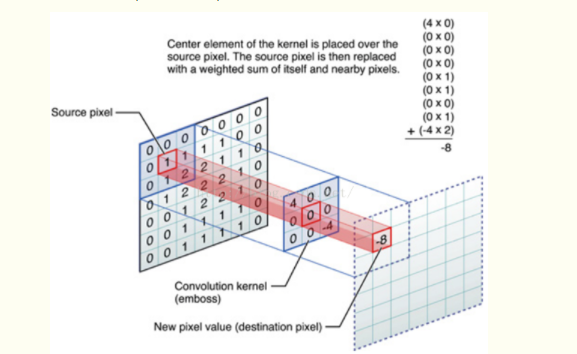
一个二维的滤波器矩阵（有个高大上的名字叫卷积核）和一个要处理的二维图像。然后，对于图像的每一个像素点，计算它的邻域像素和滤波器矩阵的对应元素的乘积，然后加起来，作为该像素位置的值。这样就完成了滤波过程。



假设有6\*6 的图像像素点矩阵，即灰色图像，色表示源图像 每个方格代表一个像素点。

黄色的矩阵表示卷积操作数矩阵，灰黄色的方格每次在X/Y前进一个像素方格，就会产生一个新的输出像素，图中正中间红色的代表要输出的像素方格，走完全部的像素方格，就得到了所有输出像素。

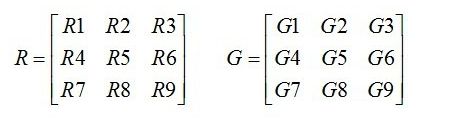
卷积过程：6\*6上面是个3\*3的窗口，从左向右，从上到下移动，黄色的每个像素点值之和取平均值赋给中心红色像素作为它卷积处理之后的新的像素值，每次移动一个像素格。



卷积示例：

3\*3的像素区域R与卷积核G的卷积运算(中心像素)

R5=R1G1+R2G2+R3G3+R4G4+R5G5+R6G6+R7G7+R8G8+R9G9;



对图像和滤波矩阵进行逐个元素相乘再求和的操作就相当于将一个二维的函数移动到另一个二维函数的所有位置，这个操作就叫卷积。

模板：矩阵方块，其数学含义是一种卷积运算；

卷积运算：是加权求和的过程；所有乘积之和作为区域中心像素的新值；

# 4学生学习内容：滤波

# 线性滤波：方框滤波、均值滤波、高斯滤波

## 方框滤波





### boxFilter函数功能及参数意义

void boxFilter( InputArray src, OutputArray dst, int ddepth,

Size ksize, Point anchor = Point(-1,-1),

bool normalize = true,

int borderType = BORDER\_DEFAULT );

## 均值滤波

均值滤波就是方框滤波的一个特殊情况。均值滤波的缺点就是不能很好地保护细节，在图像去燥的同时也破坏了图像的而细节部分，从而使图像变得模糊，不能很好的去除噪点。





#### blur函数功能及参数意义

void blur( InputArray src, OutputArray dst,

Size ksize, Point anchor = Point(-1,-1),

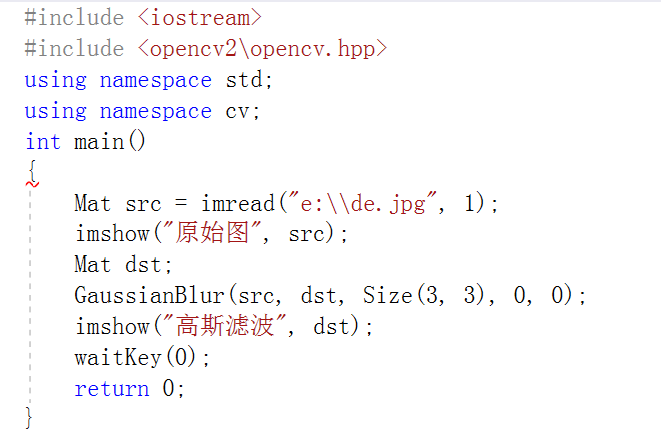
int borderType = BORDER\_DEFAULT );

## 高斯滤波

高斯滤波，可以消除高斯噪声，广泛应用于图像处理的减噪过程。



高斯滤波的而模糊感明显降低



#### GaussianBlur函数功能及参数意义

void GaussianBlur( InputArray src, OutputArray dst, Size ksize,

double sigmaX, double sigmaY = 0,

int borderType = BORDER\_DEFAULT );

# 三种线性函数对比：

明显在相同的kernel size的情况下，GaussianBlur()的结果对原图的失真比较少（当然还和sigmaX & sigmaY有关），而boxFilter & blur()得到的结果是相同的，因为都是均值滤波。

# 非线性滤波：中值滤波、双边滤波

## 中值滤波

基本思想就是用像素点的领域灰度的中值来代替该像素点的灰度值，**该方法在去除脉冲噪声、椒盐噪声的同时又能保留图像的细节（不会出现边缘模糊的情况）**。

中值滤波跟均值滤波的思想看起来很相似，只是一个取平均值，一个取中位数而已





#### medianBlur函数功能及参数意义

void medianBlur( InputArray src, OutputArray dst, int ksize );

### 中值滤波与均值滤波的比较

均值滤波中噪声成分会被加入到平均计算，所以输出是受到噪声的影响的。

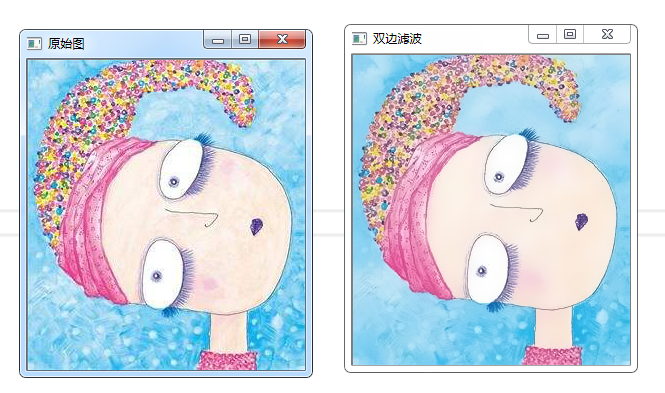
但是中值滤波中，由于噪声成分很难选上，所以基本不影响输出。当然好的性能也需要付出一点代价的； 中值滤波花费的时间是均值滤波的5倍以上。

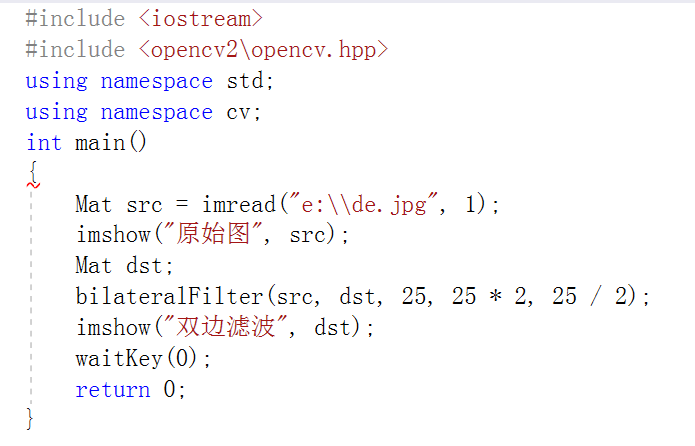
中值滤波一般采用**奇数**的卷积核。

中值滤波对一些细节多（特别是细、尖顶的）的图像不太适合

## 双边滤波

双边滤波是所有滤波中最清晰的。





#### bilateralFilter函数功能及参数意义

void bilateralFilter( InputArray src, OutputArray dst, int d,

double sigmaColor, double sigmaSpace,

int borderType = BORDER\_DEFAULT );