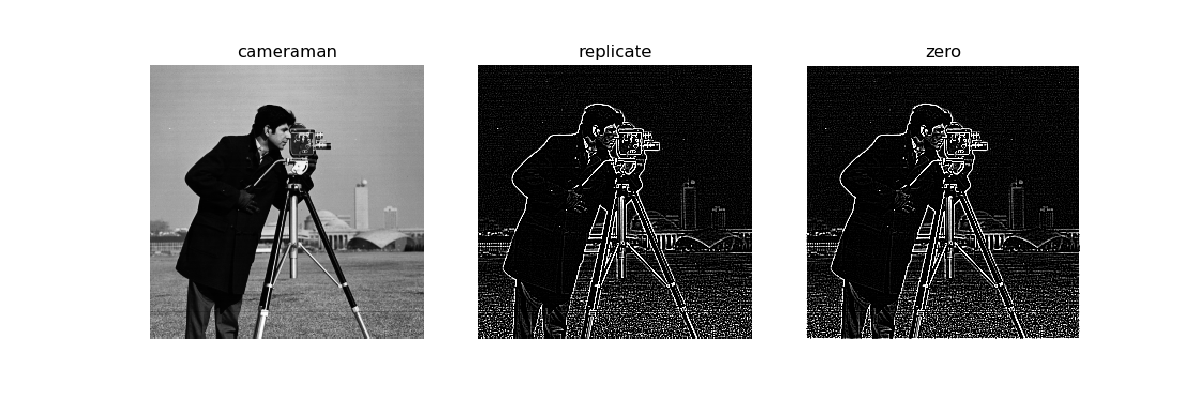
**问题1：**

（程序为q3.py）

m为核的尺寸，边界补充个数为a=（m-1）/2

**补0思想：**建一个比原先长宽均多2a的零矩阵，直接使其中间的元素等于原图像的元素。

**像素复制思想：**建一个比原先长宽均多2a的零矩阵，先判断最上面补充出来的，使其等于原图像的第零行元素，往下判断，左面补充出来的元素等于原图像第0列的元素，往右判断，元素等于原图像元素，右面补充出来的元素等于原图像最后一列的元素。再往下看，矩阵下面补充出来的元素等于原图像最后一行的元素。



**问题2：**

（程序为q4.py）

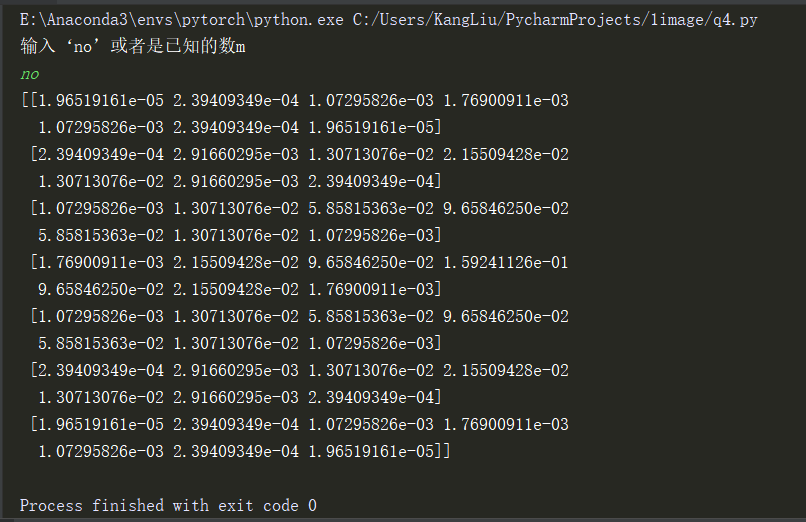
输入m，可以是no，即无输入，也可以是大于0的数。

no的话代表没有给m，所以m = math.ceil (3\*sig)\*2 + 1

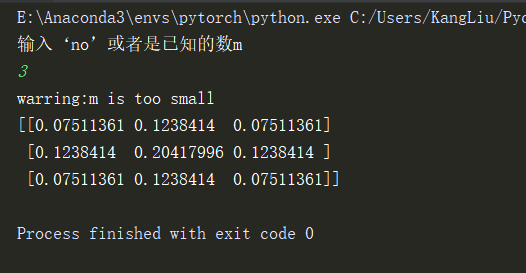
在写高斯核的时候，坐标是以中心为原点，由下式求得再归一化。

w=

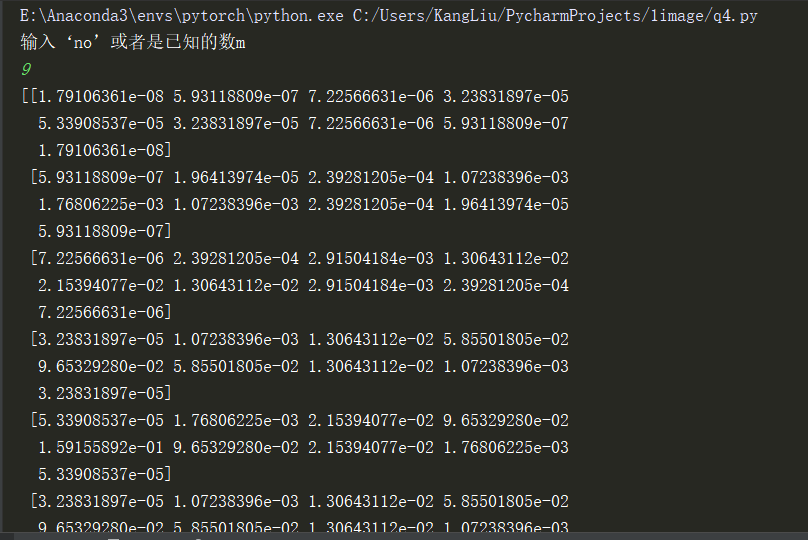
下图是输入为no时：



下图是输入为3时：



下图是输入为9时：



**问题3：**

（程序为q5.py）

把问题1、2的两个函数整合为一个函数

twodConv(f,sig,method='zero')

程序运行后会出现四张图对应如下：

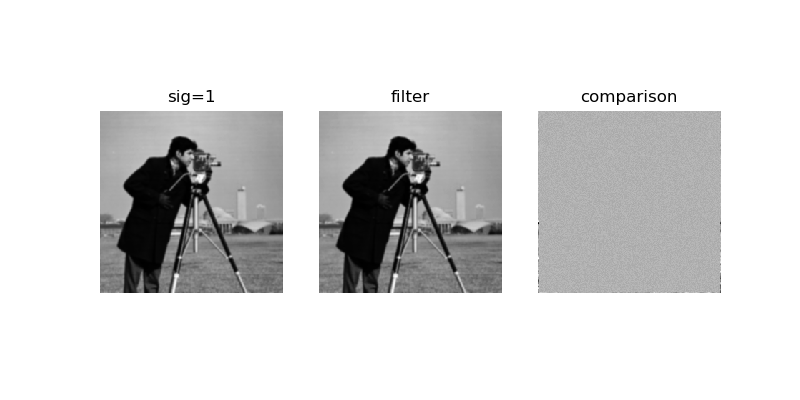
下图是一张图片在sig=1,2,3,5下的区别对比：



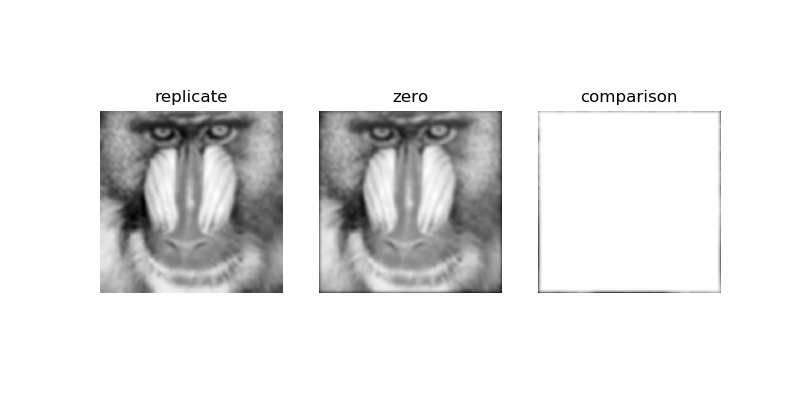
下图是分别采用sig=1,2,3,5的结果图：



下图分别对应sig=1时自己写的函数与直接调用GaussianBlur函数的比较以及两张图的差值图：



下图分别对应像素复制和补零差异以及他们的差值图（此时sig=3）：



两张图进行对比，明显看得到使用zero方法时图片的边缘有一圈黑色。我进行的差值是zero-replicate。可以看出两种方法效果基本一样，但replicate没有黑边。