实验4：图像滤波

钟 凡

zhongfan@sdu.edu.cn

实验4.1：高斯滤波

- 实现图像的高斯滤波：
  - 通过调整高斯函数的标准差(sigma)来控制平滑程度；

    void Gaussian(const MyImage &input, MyImage &output, double sigma);

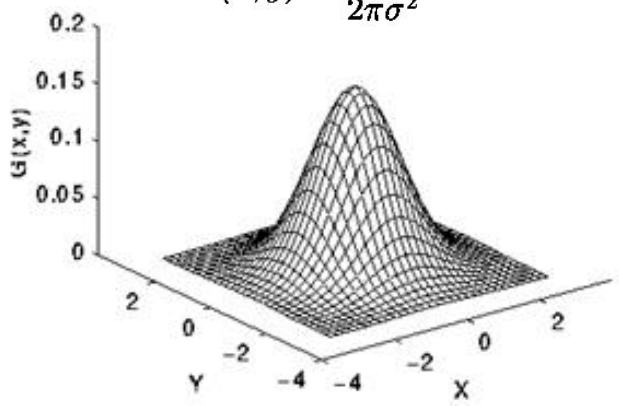
  - 滤波窗口大小取为[6*sigma-1]，[.]表示取整；

  - 利用二维高斯函数的行列可分离性进行加速；
    - 先对每行进行一维高斯滤波，再对结果的每列进行同样的一维高斯滤波；

高斯滤波

- 空间滤波=图像卷积
- 高斯滤波=以高斯函数为卷积核的图像卷积

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



二维高斯函数

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

5*5卷积核（**注意归一化!**）

实验4.2 快速均值滤波

- 实现图像的均值滤波
  - 滤波窗口大小通过参数来指定：

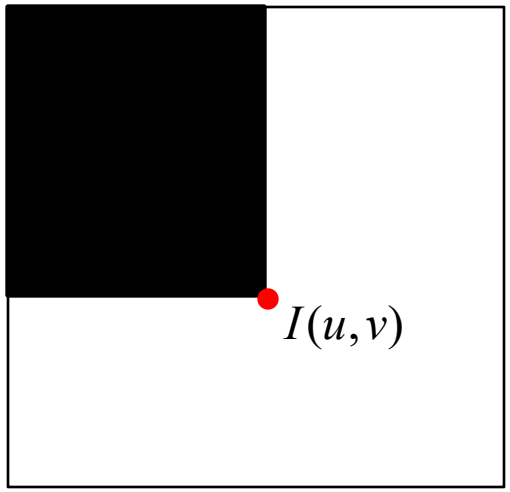
    ```
    void MeanFilter(const MyImage &input, MyImage &output, int window_size);
    ```

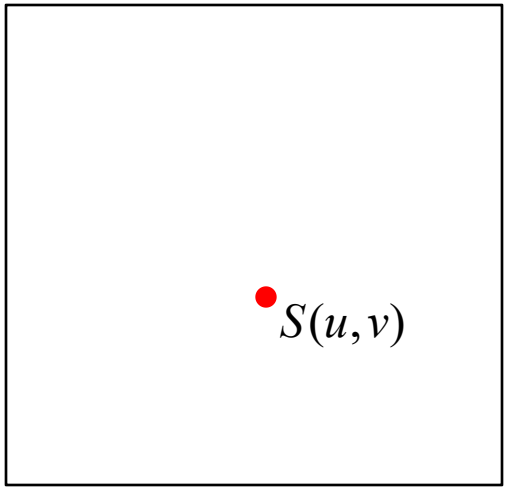
  - 采用积分图进行加速，实现与滤波窗口大小无关的效率；

- 与OpenCV的boxFilter函数比较计算速度，并分析导致速度差异的原因。

积分图

- 图像I的积分图S是与其大小相同的图像，S的每一像素S(u,v)存贮的是I(u,v)左上角所有像素的颜色值之和。





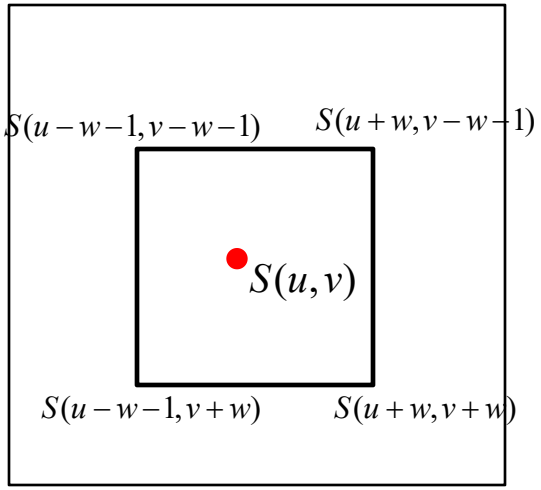
- 积分图可增量计算，只需对原图进行一遍扫描：

$$S(u,v) = S(u,v-1) + sum(I[1:u,v])$$

基于积分图的快速均值滤波

- 设滤波窗口大小为2w+1，滤波结果为图像O，则：

$$O(u,v)=\frac{1}{Z}[S(u+w,v+w)+S(u-w-1,v-w-1)-$$
$$S(u+w,v-w-1)-S(u-w-1,v+w)]$$



Z=(2w+1)*(2w+1)为像素个数；
中括号内即为滤波窗口覆盖的像素颜色值之和；