

## 【图像处理】-023 boxFilter

在学习了双边滤波之后，发现双边滤波由于涉及到两个高斯核函数的计算，其非线性特性，导致算法的计算量极大，难以实现实时处理的效果。于是，开始学习双边滤波的加速，在学习的过程中，发现很多加速算法都是对其中的两个高斯核函数进行改进或替换，之后使用诸如boxFilter、积分图像、积分直方图之类的手段进行加速。这里先学习一下boxfilter。

### 1 原理

### 2 OpenCV实现

## 1 原理

boxFilter又成为盒子滤波、方框滤波、均值滤波，它对邻域内所有的像素包括当前像素与相邻像素一视同仁，统一进行均值或累加处理。当进行均值处理时，指对结果进行了归一化处理，累加处理时，并不对结果进行归一化处理，不归一化处理的方框滤波在进行特征检测时有很多应用。

方框滤波的权值系数如下所示：

$$K = \alpha \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \quad (1)$$

其中，

$$\alpha = \begin{cases} \frac{1}{width*height} & \text{归一化} \\ 1 & \text{不归一化} \end{cases} \quad (2)$$

从权值矩阵可以看出，方框滤波涉及到的主要是对邻域内像素的求和和归一化的问题。最简单朴素的实现方法是直接对邻域内的像素按照要求一个一个的计算，得出计算结果作为滤波结果，但这无疑将是非常耗时的。目前，方框滤波比较常见的加速方法是通过行列分离的方式，通过额外的存储空间，先计算各行的累加结果，再对中间结果进行列处理，得到最终结果。

## 2 OpenCV实现

OpenCV中，方框滤波通过boxFilter实现。

函数原型如下：

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

参数解释：第一个参数，InputArray类型的src，输入图像，即源图像，填Mat类的对象即可。该函数对通道是独立处理的，且可以处理任意通道数的图片，但需要注意，待处理的图片深度应该为CV\_8U, CV\_16U, CV\_16S, CV\_32F 以及 CV\_64F之一。第二个参数，OutputArray类型的dst，即目标图像，需要和源图片有一样的尺寸和类型。第三个参数，int类型的ddepth，输出图像的深度，-1代表使用原图深度，即src.depth()。第四个参数，Size类型（对Size类型稍后有讲解）的ksize，内核的大小。一般这样写Size( w,h )来表示内核的大小( 其中，w 为像素宽度，h为像素高度)。Size (3,3) 就表示3x3的核大小，Size (5,5) 就表示5x5的核大小 第五个参数，Point类型的anchor，表示锚点（即被平滑的那个点），注意他有默认值Point(-1,-1)。如果这个点坐标是负值的话，就表示取核的中心为锚点，所以默认值Point(-1,-1)表示这个锚点在核的中心。第六个参数，bool类型的normalize，默认值为true，一个标识符，表示内核是否被其区域归一化（normalized）了。第七个参数，int类型的borderType，用于推断图像外部像素的某种边界模式。有默认值BORDER\_DEFAULT，我们一般不去管它。

OpenCV中，方框滤波是经过行列分离加速的。 OpenCV中，`blur` 图像模糊函数实际上内部调用的是归一化的方框滤波。