《数字图象处理》小作业 3

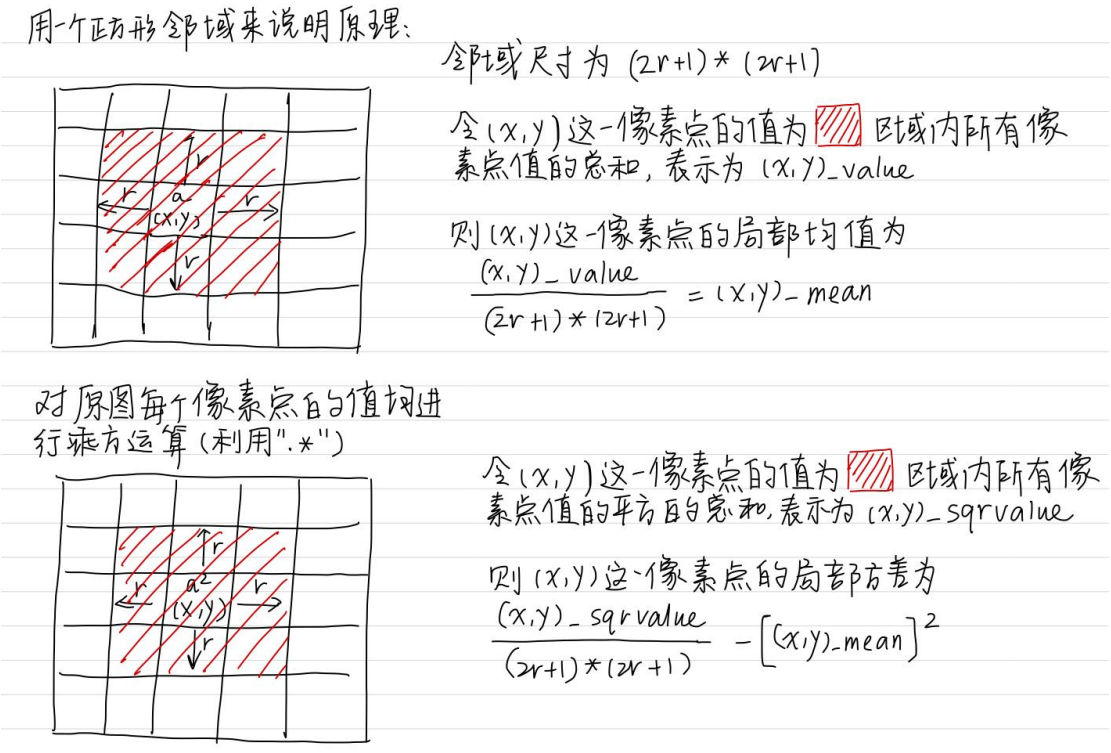
2017011010 杜澍洽 自71

一、 实验任务

1. （题目一）实现更高效的计算局部均值和局部方差的算法。
2. （题目二）对小光圈照片进行处理，使背景虚化，取得类似大光圈照片的效果。

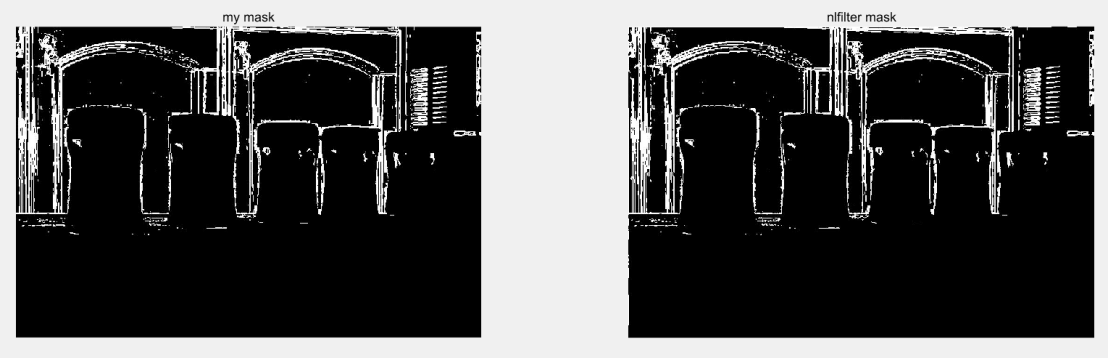
二、 题目一

计算思路如下图所示，根据此图我编写了两个函数，一个用来求取一个矩阵某尺寸的邻域内所有像素点值的总和（bsum（）），另一个用来求取给定矩阵给定邻域的局部均值和方差（getLocalVar（）），两个函数的具体内容见代码。



利用第二题所给的’0.jpg’，将其缩小后用于检验算法效果。此处选

定的邻域大小为 3*3，下图依次对比我的程序和 nlfilter 得到的 mean_local，std_local，mask，和最终的增强结果。



利用尺寸为 32*32 的邻域来检验速度，由下面的两张图可见我的函数速度更快。

探查摘要

基于performance时间于 27-Oct-2019 16:40:23 生成。

函数名称	调用次数	总时间	自用时间*	总时间图 (深色条带 = 自用时间)
myLocalEnhance	1	16.400 s	0.398 s	
getLocalVar	1	13.124 s	0.027 s	
bsum	3	13.097 s	13.097 s	
imshow	6	2.608 s	0.281 s	
initSize	6	2.029 s	0.033 s	
movegui	6	1.952 s	1.940 s	
imwrite	4	0.117 s	0.013 s	

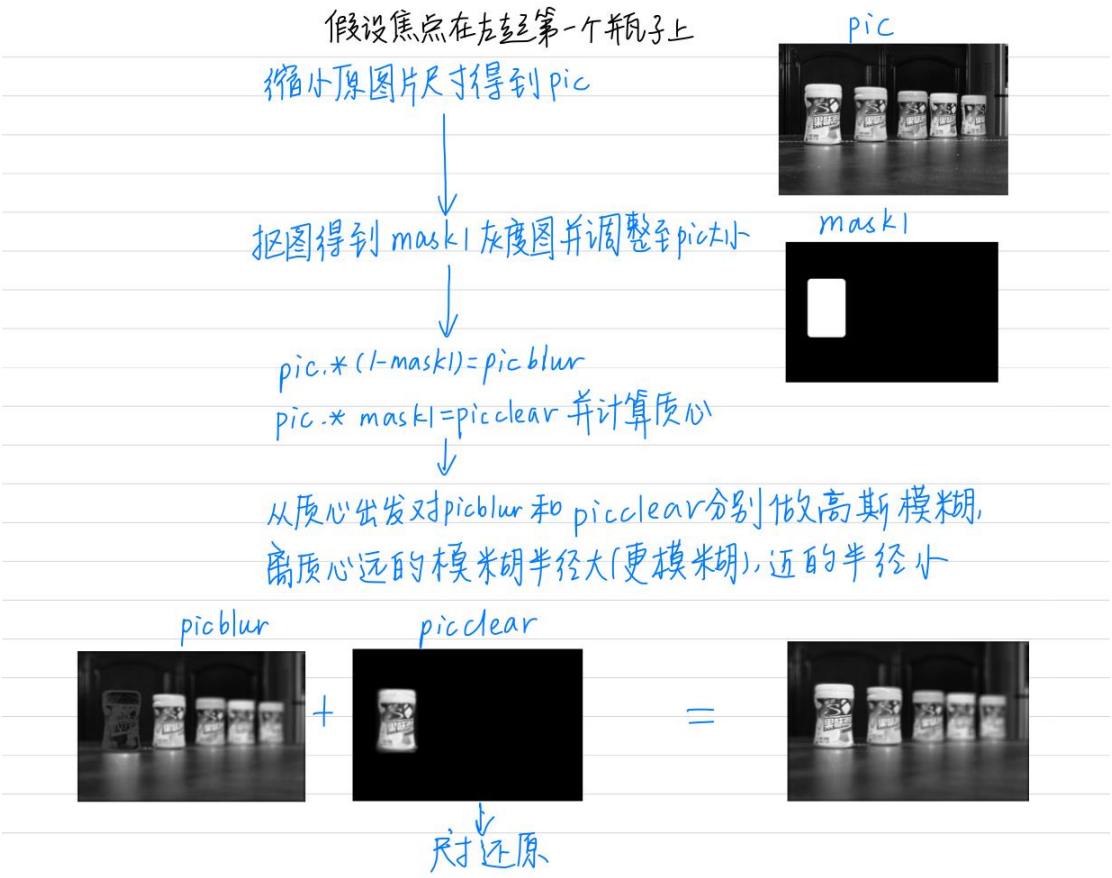
探查摘要

基于performance时间于 27-Oct-2019 16:43:42 生成。

函数名称	调用次数	总时间	自用时间*	总时间图 (深色条带 = 自用时间)
ex312_LocalEnhance	1	78.108 s	0.591 s	
nlfilter	2	75.293 s	16.707 s	
ex312_LocalEnhance>@(x)std2(x)	666001	48.348 s	3.989 s	
std2	666002	44.373 s	25.963 s	
std	666002	18.410 s	5.104 s	
var	666002	13.306 s	13.306 s	
ex312_LocalEnhance>@(x)mean2(x)	666001	7.286 s	3.176 s	
mean2	666002	4.113 s	4.113 s	
waitbar	1334	2.112 s	1.130 s	

三、 题目二

算法思路如下图所示，主要问题就在于如何实现可变半径的高斯滤波，下面对此进行详细说明。



生成不同半径高斯核的功能在函数 getGausskernel（）中实现了，正如算法思路中所述，高斯核的半径越大，各个点对应权重的差别越大，与图像卷积后也会使得图像更模糊。但如果直接用一个半径适中的高斯核与图像卷积，得到的将是均匀模糊的结果，因此考虑如何实现离焦点远的部分比离焦点近的更模糊。

首先要确定焦点，这里假设 picclear 的质心就是焦点，其位置的计算公式如下：

$$x_0 = \frac{\sum xf(x,y)}{\sum f(x,y)}$$

$$y_0 = \frac{\sum yf(x,y)}{\sum f(x,y)}$$

接下来需要衡量图像上的点到此质心的距离，我采用了曼哈顿距离来表示，并将其归一化，最后能够得到整张图像的距离矩阵（或者说是权重矩阵），这两个功能在函数 getWeightMatrix（）中实现。

在进行渐进模糊时，我的方法是手动画分距离区域，即距离小于 0.05 的部分用半径为 3 的高斯核去卷积，距离在 0.05 到 0.2 之间的则用半径为 9 的核。具体的参数设置在函数 blurImg（）中实现。

对图像的各个部分分别处理后再叠加就得到了最终结果，这一部分在函数 focusNum（）中。

四、 问题与不足

题目二的算法存在一个显著的问题，即在渐进模糊时，因为我是手动设置分区，不管如何调整参数，都不能避免因为高斯核的离散半径而造成的跳跃，反映在图像上就是曼哈顿距离框出的四边形边缘会显现出来（也尝试过欧氏距离，但会使速度明显变慢）。我没能解决这个问题，因此想请教一下助教有没有可能的方案。

五、 实验总结

我花了很多时间在本次小作业的题目二上，主要是用来调整分区的参数使图片看起来更自然一点，但我意识到这个方案从根本上无法规避上面提到的问题，这是我在设计算法之初没考虑到的。我后来也跟其他同学交流了方法，发现有实现起来更简单效果也更好的，今后的实践中我会更注意交流。