山东大学 计算机科学与技术 学院

数字图像处理 课程实验报告

|  |  |  |
| --- | --- | --- |
| 学号：201705130113 | 姓名：黄瑞哲 | 班级：计科17.3 |
| 实验题目：图像滤波 | | |
| 实验内容：  实验4.1：高斯滤波    实验4.2 中值滤波  实现图像的中值滤波  实验4.3 快速均值滤波 | | |
| 实验过程中遇到和解决的问题：  （记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）  在使用镜像映射处理边界情况的时候，由于理解错误，我第一次将代码写成了循环映射，而不是镜像映射，但是由于测试图片蒙娜丽莎的左右两侧图像几乎一致，因此在结果中并没有看出什么差异来，经同学指正才发现并改正。  此外，在求像素和的时候，由于忘记添加括号，导致计算的优先级出了问题，导致计算出的图片是一堆乱点，经过很长时间的排查才发现这个问题。 | | |
| 实验步骤和代码：  实验4.1  1. 由于实验要求中告诉我们可以利用二维高斯函数的行列可分离性进行加速运算，先对每行进行一维高斯滤波，再对结果的每一列进行同样的一维高斯滤波。那么我们先根据一维高斯滤波和窗口大小计算出窗口中的高斯函数值，然后做归一化处理使得他们的和为1    2. 然后对于每一行，做列上的一维高斯滤波。方法是将当前点(i, j)作为窗口的中心点，然后计算其水平方向上窗口内的点与高斯函数的线性加权，作为该点的像素值。特别的对于边界情况，我们采用镜像映射的方式计算那些位于图像以外的点的像素值。    这样我们就可以正确的处理边界之外的像素点了。    然后对于每一列，做行上的一维高斯滤波，代码与上述基本一致，只需要交换一下内外层循环    实验4.2中值滤波  1. 中值滤波的原理是将像素点周围窗口中所有像素点的像素值的中位数作为该点的像素值，因此只需要将所有点的像素值加入一个线性表中，对其进行排序，然后去中间的那个数即可    同样的，对于边界情况，我采用了镜像映射的方式，计算边界之外的像素点的像素值。  实验4.3快速均值滤波  1.均值滤波的原理是将像素点周围窗口中的所有点的像素值求平均作为该像素点的像素值，由于该方法需要遍历窗口中的每一个点，因此它的复杂度与窗口大小的有关，当窗口过大时会造成程序效率低下的情况。因此我们采用积分图的方式来计算窗口中像素值的总和，再除以窗口大小就可以得到窗口的平均值。  2. 计算积分图本质上是二维矩阵的前缀和    由这个图我们看到对于(i,j)这个点来说，它的前缀和是红色+深蓝+浅蓝+绿色，而点(i, j-1)的前缀和是天蓝+红色，点(i-1,j)的前缀和是绿色+红色,点(i-1,j-1)的前缀和是红色，而蓝色则对于点(i,j)的值。因此我们可以看出，sum(i, j)=sum(i-1, j) + sum(i, j-1) - sum(i-1, j-1) + val(i, j)。由于将点(i-1, j)的前缀和和点(i, j-1)的前缀和相加时，红色区域被加了两次，因此我们在计算的时候需要将红色区域剪掉（容斥原理），由此我们得到了上述前缀和的计算公式。这样我们对这个图片遍历一遍就可以计算出它的积分图来。时间复杂度为O(宽度\*高度)  3.计算窗口像素值之和  当我们得到积分图以后，我们便可以在常数的时间复杂度内得到窗口范围内的像素值之和，而计算和的时候与求前缀和类似，需要应用容斥原理。假设中心点是(u, v)，窗口大小为w，那么计算公式为 O(u, v)=[sum(u+w, v+w)-sum(u-w-1, v+w)-sum(u+w, v-w-1)+sum(u-w-1, v-w-1)]/Z，其中Z是窗口中像素点的个数。  当然这里还存在一个问题，如果窗口中的一些像素点在原图像的边界之外，那么我们在编程的过程中，如果直接应用这个公式将会造成越界错误，以致程序不能正常运行。那么为了避免这种情况，我参照镜像映射的原理将原来u\*v的图片扩大到了(u+2\*w)\*(v+2\*w)中，即在原图像的上下左右、左上左下右上右下共八个方向扩充了镜像内容。    他们分别映射    这样以后对于整个大图求它的积分图，然后直接应用刚才说到的计算窗口像素值之和的公式即可，这样程序运行的效率与窗口大小无关。  代码如下    其中PIEXL(i, j)是计算这个点对应原图像素点的一个宏 | | |
| 实验结果：  左上：原图  右上：高斯滤波，sigma=2  左下：中值滤波，窗口直径=11  右下：快速均值滤波，窗口直径=11 | | |
| 实验总结：  在这个实验，我学会了如何除去原图像中的噪声点：利用图像滤波技术，就可以去除原图像中的噪点。图像滤波中不同的算法处理出来的图像表现也有不同：高斯滤波给人一种虚化的感觉，中值滤波像是蒙了一层纱，图像有些许变形，而均值滤波更像是近视的人不戴眼镜。  此外，我也学习到了如何去处理边界情况，一般常用的是镜像映射和常数填充，一般来说镜像映射会更合理一点，因为边界外的像素点与边界内的像素点是相关联的，而常数填充是不相关的，因此镜像映射处理出来的结果会更加自然一点。 | | |