



暨南大学
JINAN UNIVERSITY

课程实验报告

课程类别： 专业学位课

课程名称： 图像工程与处理

课程代码： **085400mb03**

任课教师： 彭青玉、张庆丰

《双边滤波器的模版设计与处理分析》

姓 名 李永鸿

学 号 202034261171

院 系 信息科学技术学院计算机科学系

专 业 计算机技术

电子邮件 707884308@qq.com

提交时间： 2021 年 05 月 10 日

1. 去噪算法的简介、双边滤波算法的背景介绍与说明

图像去噪是图像处理领域中一项基本而又关键的技术，一直是图像处理领域的难题。在图像的获取、传输和存储过程中总是不可避免地受到各种噪声的干扰。为了从图像中获取更准确的信息，图像去噪预处理算法的好坏成为后续处理的关键。常用的去噪模型有：高斯滤波器、各项异性扩散方程、双边滤波器、全变分模型、小波变换滤波器、非局部平均滤波器等。图像去噪包含两个方面的内容：（1）消除噪声；（2）增强图像特征。

但这两个目标在一定程度上是矛盾的。因为去除噪声意味着除去图像的高频部分，而图像的边界同样是高频部分，所以在去除噪声的同时，往往使得图像的边界变得模糊。如何解决好这一对矛盾是评价图像去噪模型好坏的一个重要标准。

去噪算法其中有高斯滤波，它的原理是邻域内像素值的加权平均，权值的大小取决于空间的邻近程度，距离越近，权值越大。这种邻域内的平均会使一些边界被平均掉而模糊图像。双边滤波正是为了解决边缘模糊而提出的，它不仅考虑空间的邻近性也考虑亮度的相似性，只有邻域内亮度相似的才被一起平均。双边滤波自提出以来在理论和应用方面都有了很大的改进和发展。

2. 开发环境、语言的介绍与库函数说明

操作系统：deepin20.2

语言：C/C++

编译器：g++

库函数说明：调用了 opencv 库，只是利用 opencv 库来读写图片文件。

3. 双边滤波的基本原理

如前文所说，双边滤波正是为了解决边缘模糊而提出的。因为它的核由两个函数生成：

1) **空间距离**：指的是邻域内某点与中心点的欧式距离。空间域高斯函数其数学形式为（这就是高斯滤波核）：

$$e^{-\frac{(x_i-x_c)^2+(y_i-y_c)^2}{2\sigma^2}} \quad (\text{公式 1})$$

其中 (x_i, y_i) 为邻域内某点位置， (x_c, y_c) 为中心点的位置， $\sigma(\text{space})$ 为空间域标准差。

2) **灰度距离**：指的是邻域内某点灰度与中心点灰度的差的绝对值。值域高斯函数其数学形式为：

$$e^{-\frac{(\text{gray}(x_i, y_i) - \text{gray}(x_c, y_c))^2}{2\sigma^2}} \quad (\text{公式 2})$$

其中 $\text{gray}(x_i, y_i)$ 为邻域内某点灰度值， $\text{gray}(x_c, y_c)$ 为中心点灰度值， $\sigma(\text{color})$ 为值域标准差。

双边滤波是以像素的加权平均定义的，不同的是双边滤波利用强度的变化来保存边缘信息，双边滤波依据两个像素空间上的邻近关系和灰度上的相似性。**BF** 作为双边滤波的符号，由下式定义：

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(I_p - I_q) I_q$$

其中 W_p 是一个标准量， σ_d 即 $\sigma(\text{space})$ ， σ_r 即为 $\sigma(\text{color})$ 。

$$W_p = \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(I_p - I_q)$$

对于高斯滤波，仅用空间距离的权值系数核与图像卷积后，确定中心点的灰度值。即认为离中心点越近的点，其权重系数越大。双边滤波中加入了对灰度信息的权重，即在邻域内，灰度值越接近中心点灰度值的点的权重更大，灰度值相差大的点权重越小。所以掩膜最终权重大小，则由空间域高斯核函数和值域高斯核函数共同确定。双边滤波器能较好的消除噪声，得到去噪的平滑图像，且能保持图像的边缘细节，解决了高斯滤波模糊图像的问题。

3) 参数:

双边滤波器是由两个参数 **sigma space** 和 **sigma color** 决定。**sigma space** 和 **sigma color** 一旦其中一个量接近 0 就会出现不平滑现象，只要 **sigma space** 的范围变化小于它的幅度增加空间 **sigma color**，就对边缘没有影响，**sigma color** 的变化比 **sigma space** 更能影响图像的细节。随着参数 **sigma color** 逐渐增加，双边滤波器逐渐接近高斯模糊，因为这一系列的高斯函数比较平坦，即在图像覆盖的强度间隔上几乎是个常量。增加空域参数 **sigma space** 可以光滑更多特征。双边滤波的一个重要的特点是两个量是相乘的，这意味着一旦其中一个量接近零就会出现不平滑现象。尽管滤波器有很大的空间范围，一个大的空域高斯函数和灰度域的高斯函数可以实现有限的高斯平滑，这样确保了对轮廓的严格保护。

4. 课程设计实现细节

获取值域高斯核模板（核心代码）

```
for (int i = 0; i < 256; ++i){
    double colordiff = exp(-(i*i) / (2 * colorSigma * colorSigma));
    colorMask.push_back(colordiff);
}
```

这里用查表的方式计算，因为灰度差的范围是固定的。这个过程只是上面公式的代码实现。表的范围是 0~255，在 OpenCV 中用的是 0~255*channels。此处处理彩色图像实现的方式和 OpenCV 的方式不同，此处每个通道分别计算其值域高斯核模板，所以灰度差异最大是 255，但是 OpenCV 是三个通道的像素值先加起来再相减的，所以差异最大是 255*channels。

获取空间域高斯核模板（高斯滤波模板）

```

for (int i = 0; i < h; ++i){
    y = pow(i - center_h, 2);
    double* Maskdate = Mask.ptr<double>(i);
    for (int j = 0; j < w; ++j){
        x = pow(j - center_w, 2);
        double g = exp(-(x + y) / (2 * spaceSigma * spaceSigma));
        Maskdate[j] = g;
        sum += g;
    }
}

```

使用的处理方法与 OpenCV 相类似，是对上述公式的代码实现。

5. 程序设计过程

主函数主要功能为调用 `opencv` 库函数来读取图像，然后等待用户输入图像的名称，双边滤波函数的两个参数的值，随后开始调用双边滤波函数，最后在显示设备现实原图与处理过的图并输出保存处理结果。流程图如图（1）所示。

双边滤波函数主要为用代码实现理论公式当中的计算，它的编程内容主要都是公式的实现，该函数的主要流程图如图（2）所示。

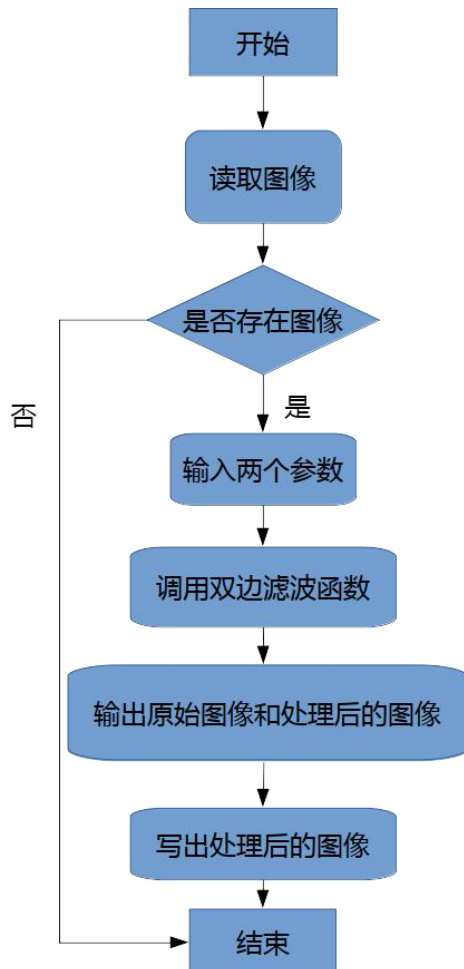


图 1

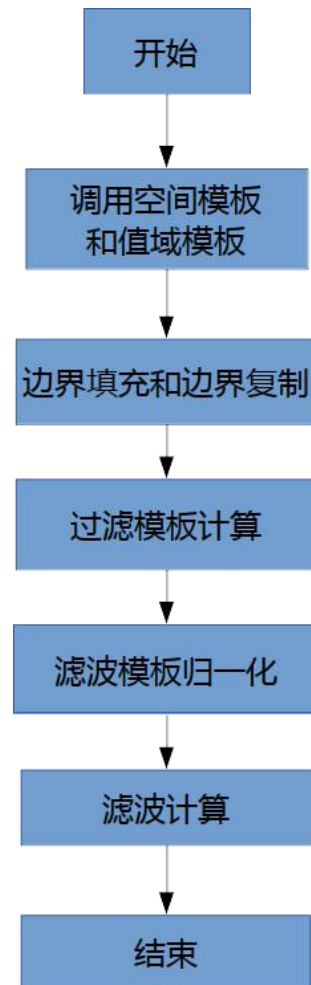


图 2

6. 实验结果与分析

双边滤波的去噪效果取决于两个参数 $\sigma_d \cdot \text{sigma}(\text{space})$ 和 $\sigma_r \cdot \text{sigma}(\text{color})$ ，式(1)中的 σ_d 和式(2)中的 σ_r 分别控制着空域和亮度域的特征。虽然参数的选择应该和噪声还有图像的特征联系起来，但是具体的选择标准还没有研究过。通过分析双边滤波的参数作为噪声水平的函数。基于一些模拟的结果，证明了 σ_r 的值比 σ_d 的值在在改变噪声水平方面更重要。

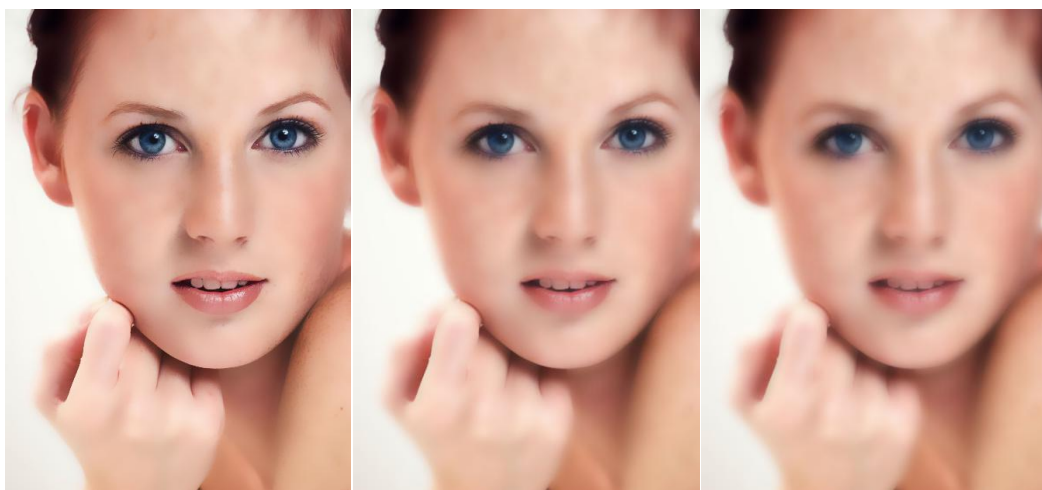
在下面的内容中，通过做大量实验对参数的选择做了研究，下面的 s10c10 表示 $\text{sigma}(\text{space})=10, \text{sigma}(\text{color})=10$ ，其他的例子皆是如此。



原图

s10c1

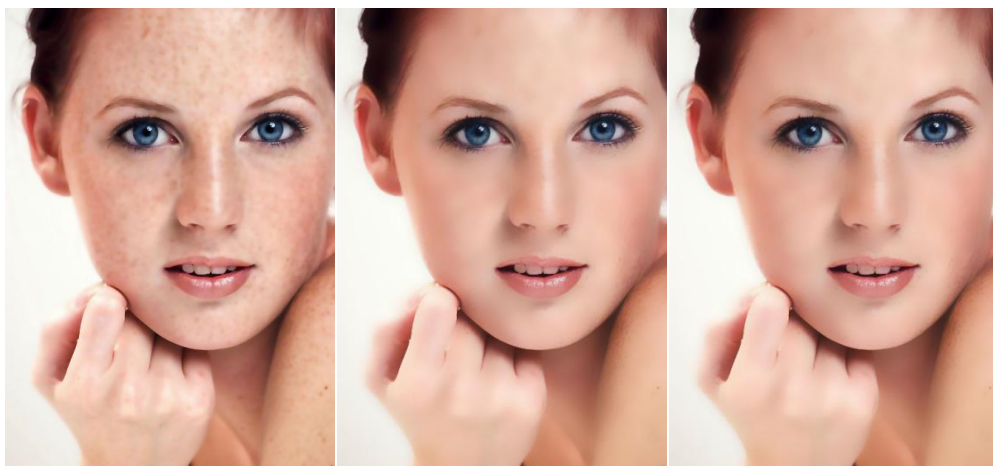
s10c2



s10c30

s10c60

s10c90



s2c30



s15c30



s30c30



s60c30



s90c30

以上实验表明了双边滤波参数选择的一般方法, $\sigma(\text{space})$ 的取值对图像去噪质量的影响并不明显, 而 $\sigma(\text{color})$ 要根据噪声的情况而定, 太大或太小时去噪效果都不好。

双边滤波器可以很好的保存图像边缘细节而滤除掉低频分量的噪音, 但是双边滤波器的效率不是太高, 花费的时间相较于其他滤波器而言也比较长。对于简单的滤波而言, 可以将两个 σ 值设置成相同的值, 如果值 < 10 , 则对滤波器影响很小, 如果值 > 150 则会对滤波器产生较大的影响, 会使图片看起来像卡通。

已知 $0 \leq \Delta g \leq 255$; 1) 假设 $\sigma = 255$, 当 $\Delta g = 255$ 时, 系数为 $\exp(-1) = 0.3679$, 当 $\Delta g = 0$ 时, 系数为 $\exp(-0) = 1$. 灰度最大点的系数与相差最小的灰度值系数之比为 0.3679 . 2) 假设 $\sigma = 122.5$, 当 $\Delta g = 255$ 时, 系数为 $\exp(-4) = 0.0183$, 当 $\Delta g = 0$ 时, 系数为 $\exp(-0) = 1$. 灰度差最大点的系数与相差最小的灰度值系数之比为 0.0183 . 结论: 因为 e^x 导数为 e^x , 其增长速度为指数增长。

当 σ 较大时, 灰度差最大值与最小值的系数在很小的一个范围之内, 其比值较大。及灰度差较大的点, 对于中心点也会有相应的较大的权值, 此与双边滤波的保留边缘的初衷相违背。

当 σ 较小时, 灰度差最大值与最小值的系数在较大的一个范围之内, 其比值很小, 及灰度差较大的点, 对应中心点仅有很小的权重。

综上所述可知:

Sigma 越大，边缘越模糊，极限情况为 sigma 无穷大，值域系数近似相等（忽略常数时，将近为 $\exp(0) = 1$ ），与高斯模板（空间域模板）相乘后可认为等效于高斯滤波。

Sigma 越小，边缘越清晰，极限情况为 sigma 无限接近 0，值域系数近似相等（接近 $\exp(-\infty) = 0$ ），与高斯模板（空间域模板）相乘后，可近似为系数皆相等，等效于源图像。

7. 本实验函数与 opencv 的函数效果对比

实验参数为 $\text{sigma}(\text{space})=10$, $\text{sigma}(\text{color})=30$ 。



原图

本实验函数

OpenCV 函数

可以看到 OpenCV 的函数平滑程度低一些。