第二次作业：空间域与频率域滤波器

姓名：丁保荣　　**学号**：171860509

2019 年 10 月 31 日

1　从空间域滤波器生成频域滤波器

1.1　大致实现原理

sobel.m 中包含了 sobel 算子的空间域滤波和频率域滤波。

average.m 中包含了一个 3*3 的均值算子的空间域滤波和频率域滤波。

prewitt.m 中包含了 prewitt 算子的空间域滤波和频率域滤波。

laplacian.m 中包含了一个 $\alpha = 0.2$ 的 laplacian 算子的空间域滤波和频率域滤波。

padding.m 是用来实现补零操作的。

test.m 是复现课程 ppt 中的代码的，做了一些改动。

频域滤波器: 首先对目标图像进行补零，中心化，然后进行傅里叶变换。对算子也进行类似操作。然后将傅里叶变换后的算子和傅里叶变换后的目标图像相乘，然后做傅里叶逆变换得到滤波后的图像。将新图像去中心化，取实部，并裁剪至原图像大小。我这里没有取 abs，原因在后面会说明。

空间域滤波器：将算子和目标图像进行补零，然后进行卷积，然后将新图像裁剪至原图像大小。

1.2　实现效果

原图如下所示：

图 1: 原图

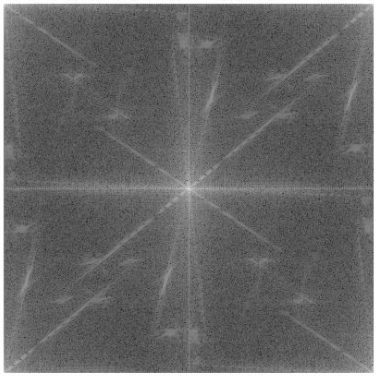

图 2: 原图的傅里叶谱

1.2.1 Sobel 算子

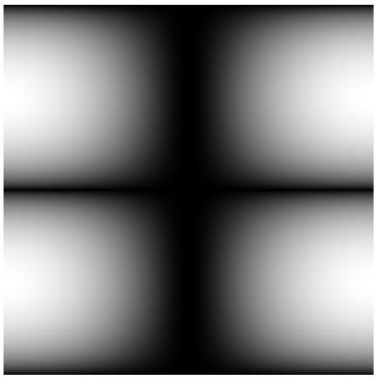

图 3: Sobel 算子傅里叶谱

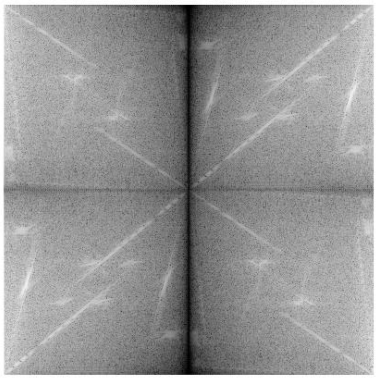

图 4: 原图经过 sobel 算子滤波后的傅里叶谱

图 5: Sobel 算子空间域滤波

图 6: Sobel 算子频域滤波

1.2.2 均值算子

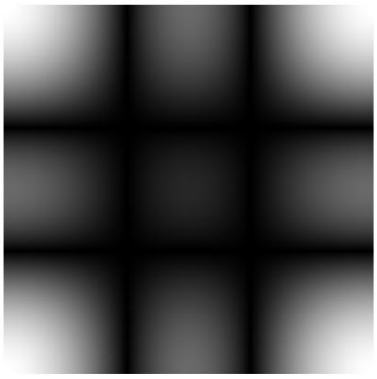

图 7: 均值算子傅里叶谱

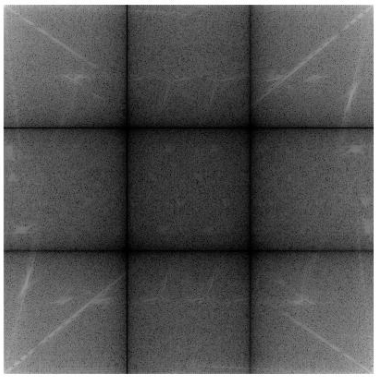

图 8: 原图经过均值算子滤波后的傅里叶谱

图 9: 3*3 的均值算子空间域滤波

图 10: 3*3 的均值算子频域滤波

1.2.3 prewitt 算子

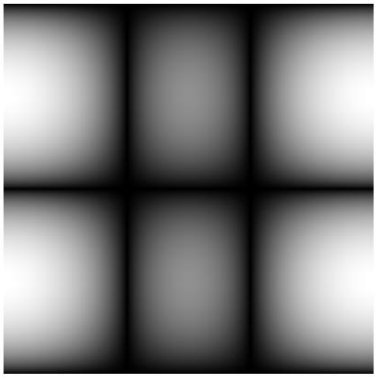

图 11: prewitt 算子傅里叶谱

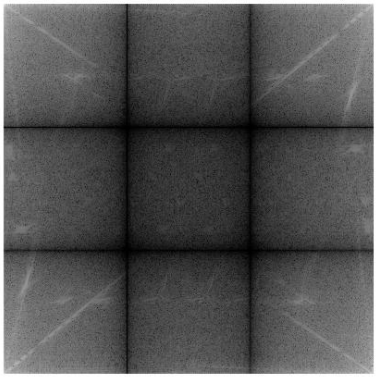

图 12: 原图经过 prewitt 算子滤波后的傅里叶谱

图 13: prewitt 算子空间域滤波

图 14: prewitt 算子频域滤波

1.2.4 laplacian 算子

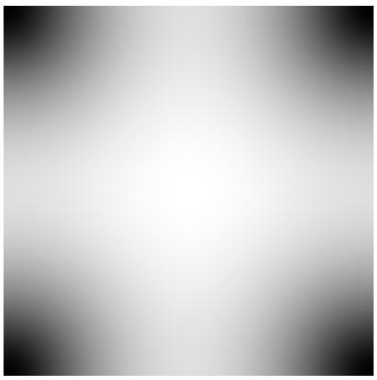

图 15: laplacian 算子傅里叶谱

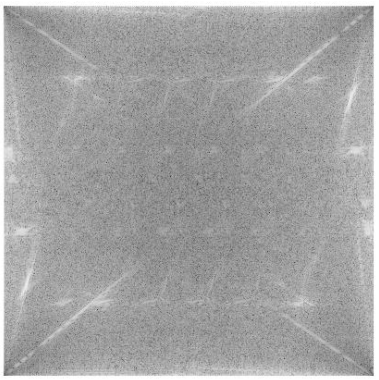

图 16: 原图经过 laplacian 算子滤波后的傅里叶谱

图 17: $\alpha = 0.2$ 的 laplacian 算子空间域滤波

图 18: $\alpha = 0.2$ 的 laplacian 算子频域滤波

1.3 经验或教训

1. 在测试课件上的代码时，发现效果不对，最后发现课件上的代码没有把图像还原成原来的大小。

2. 显示图像的时候，要转换成 uint8, 不然效果很奇怪。

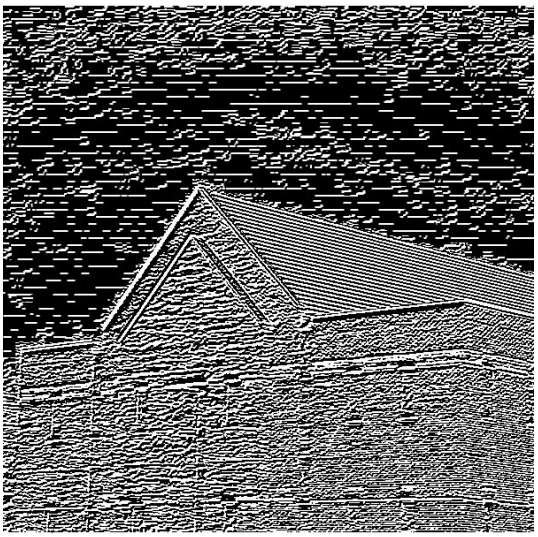

图 19: 未进行类型转换

图 20: 进行类型转换

3. 我发现最后显示图像前，如果对图像取 abs 会发现错误，效果如下所示。注意看屋顶。

图 21: 使用 abs 函数

图 22: 不使用 abs 函数

2 美颜软件

2.1 大致实现原理

log_convertion.m 文件里是一个对图像进行对数灰度变换的函数, 公式如下，可以用来实现美白功能。

$$Output(i,j) = \frac{\log\left(Input(i,j) * (\beta - 1) + 1\right)}{\log \beta}$$

Butterworth.m 文件中包含了巴特沃斯低通滤波器的实现。

GausLow.m 文件中包含了高斯低通滤波器的实现。

Ideal.m 文件中包含了理想低通滤波器的实现。

midFilter.m 文件中包含了中值滤波器的实现。

RF.m 文件中包含了递归域变换滤波器的实现，代码参考自网络，并做了小幅修改。

主要实现的点在于生成相应的滤波器。首先对大小为 $m \times n$ 的图像进行补零，形成大小为 $2m \times 2n$ 的图像，并进行中心化。所以我们设计的滤波器大小为 $2m \times 2n$, 且中心位于 $(m+1, n+1)$, 所以我们可以对滤波器矩阵中的每个点，根据它与中心的距离按照公式来赋值即可。

2.2 实现效果

2.3 理想低通滤波器

图 23: 原图

图 24: $D_0 = 130Hz$

图 25: 加入美白效果,$\beta = 5$

可以看出理想滤波器对眼角的皱纹稍微去除了一点，但整个图像也变得有些模糊了，尤其是脸颊和脖子处。美白效果较好。

2.4 巴特沃斯低通滤波器

图 26: 原图

图 27: $D_0 = 80Hz, N = 2$

图 28: $D_0 = 100Hz, N = 2$

图 29: 加入美白效果,$\beta = 5$

可以看出巴特沃斯低通滤波器的效果比理想滤波器的要好一点，$D_0 = 100Hz$ 的效果较好。

2.5 高斯低通滤波器

图 30: 原图

图 31: $D_0 = 80Hz$

图 32: $D_0 = 100Hz$

图 33: 加入美白效果,$\beta = 5$

2.6 中值滤波器

图 34: 原图

图 35: 大小为 5×5 的中值滤波器

图 36: 加入美白效果,$\beta = 5$

2.7 递归高斯滤波器

通过上面的几个滤波器，我们发现它们都会是图像变得很模糊，这是因为上面的几种滤波方法都都只考虑了空间邻近度，而忽视了像素之间的像素度。所以我在网上找到了另一类滤波叫做边缘保留滤波 (Edge, Preserving Filter, EPF). EPF 中有一个滤波叫做递归高斯滤波器 (RF). 用它实现的效果较好。

图 37: 原图

图 38: 透明度 =30

图 39: 透明度 =50

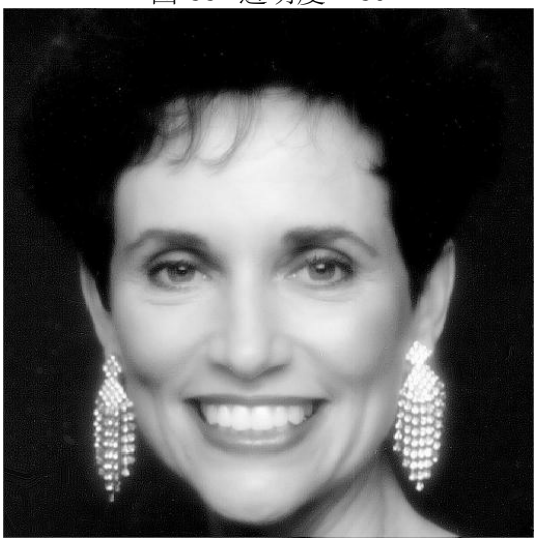

图 40: 加入美白效果,$\beta = 5$