**《生物医学图像处理实验》实验报告**

**（2021-2022学年第2学期）**

实验四：图像的二维傅里叶变换及变换域滤波

**课程号**：310025010 **课序号**：01 **任课教师**：林江莉 **成绩：**

**组长**：徐广玄 **小组成员：无**

**实验日期：**2023年4月30日

**一、实验内容：**

1. 掌握FFT和IFFT的定义，使用C语言（或基于matlab语句）编写FFT和IFFT的程序，能对一幅二维数字图象进行移中和变换。(注意不能直接调用matlab 的fft函数)
2. 编写理想滤波器和巴特沃斯低通滤波器函数（阶数可设），并对比两者滤波的效果。
3. 编写理想滤波器和巴特沃斯高通滤波器函数（阶数可设），并对比两者锐化图像的效果。
4. 编写同态滤波函数，并实现“PET\_image.bmp”和“factory.bmp”图像的同态滤波。

注意：逐渐实现**能对8位，16位，24位，32位图像进行操作。**

**二、编程软件：**

Python3.9，基于pycharm，不使用opencv

**三、实验步骤、结果及关键代码：**

1. **FFT和IFFT**

在这个实现中，我们首先定义了一个 ifft2d() 函数来计算二维灰度图像的逆 FFT。函数输入参数是一个二维的 Numpy 数组，表示一个灰度图像的 FFT 结果。函数首先对每一行进行一维的逆 FFT，然后对每一列进行一维的逆 FFT。我们使用 numpy.empty\_like() 函数创建一个空的 Numpy 数组来保存逆 FFT 的结果，使用 dtype=np.complex128 来指定数据类型为复数类型。在这个函数中，我们调用了 ifft() 函数来计算一维的逆 FFT。在 ifft() 函数中，我们使用了 Cooley-Tukey 算法来计算一维的逆 FFT。输入参数 x 是一个一维的 Numpy 数组，输出结果是逆 FFT 后的结果。这个实现与之前提供的一维 FFT 实现类似，使用递归方法将序列分成偶数和奇数部分，并将它们组合起来。函数使用了 Numpy 的广播机制来避免一些重复计算。最终，我们得到了一个二维的复数数组，表示输入图像的空域表示。我们将结果进行缩放并取实部，得到了灰度图像的逆 FFT 结果。

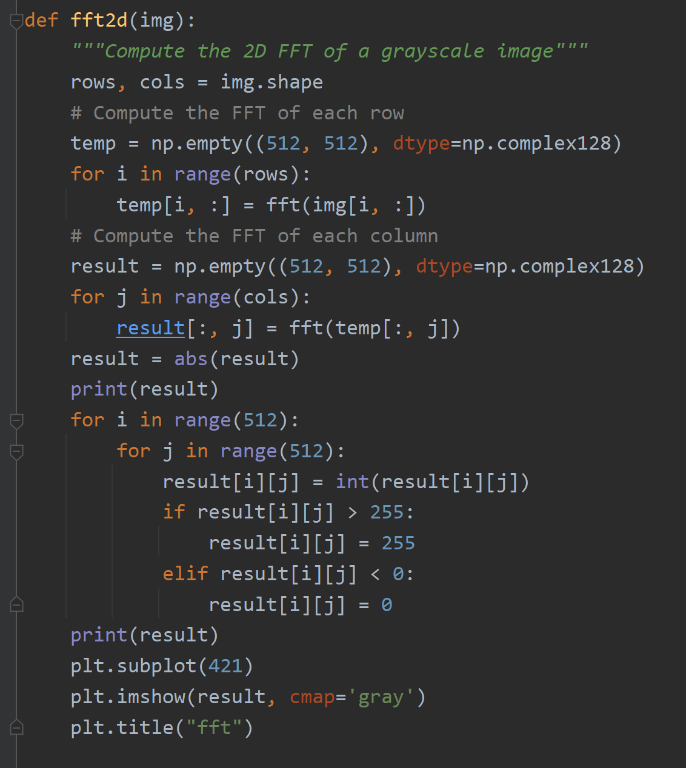


图1 二维fft实现

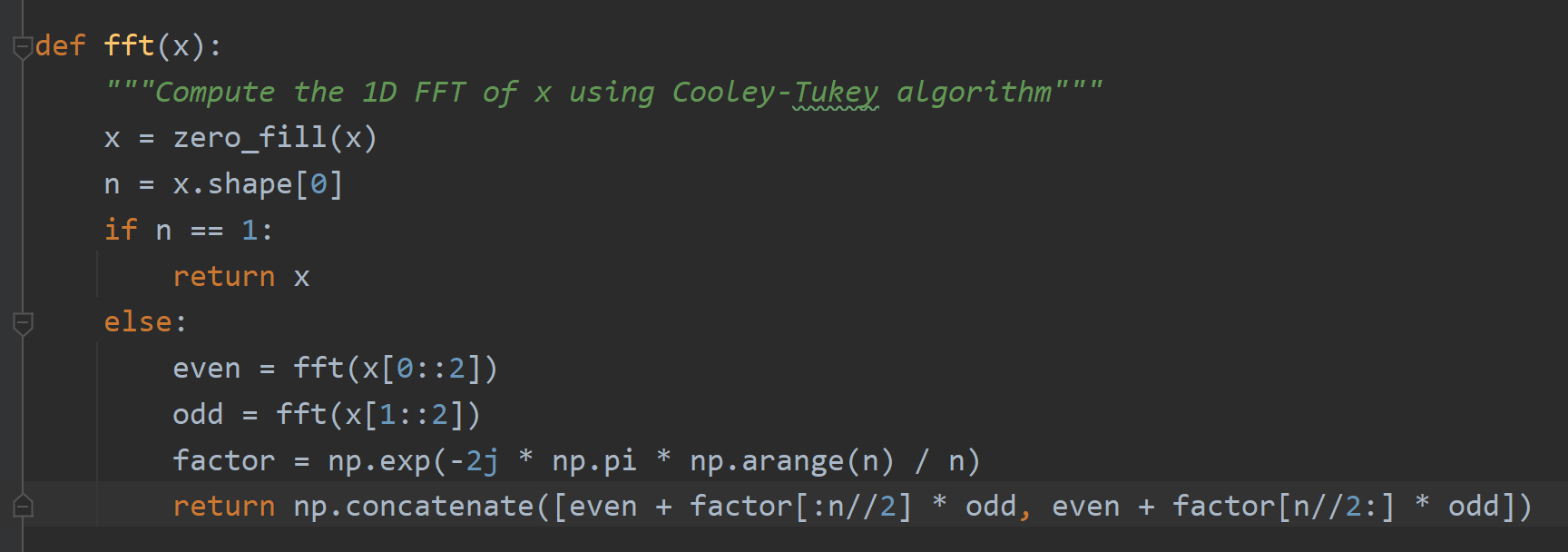


图2 一维fft实现

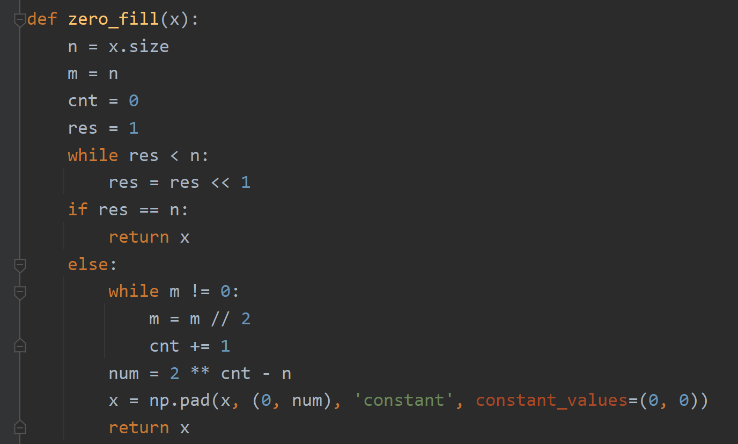


图3 fft补位

1. **理想滤波器和巴特沃斯低通滤波器函数**

**对于理想滤波器和**butterworth**滤波器来说，需要将时域图转换到频域，然后乘一个截断函数，再转回时域。**

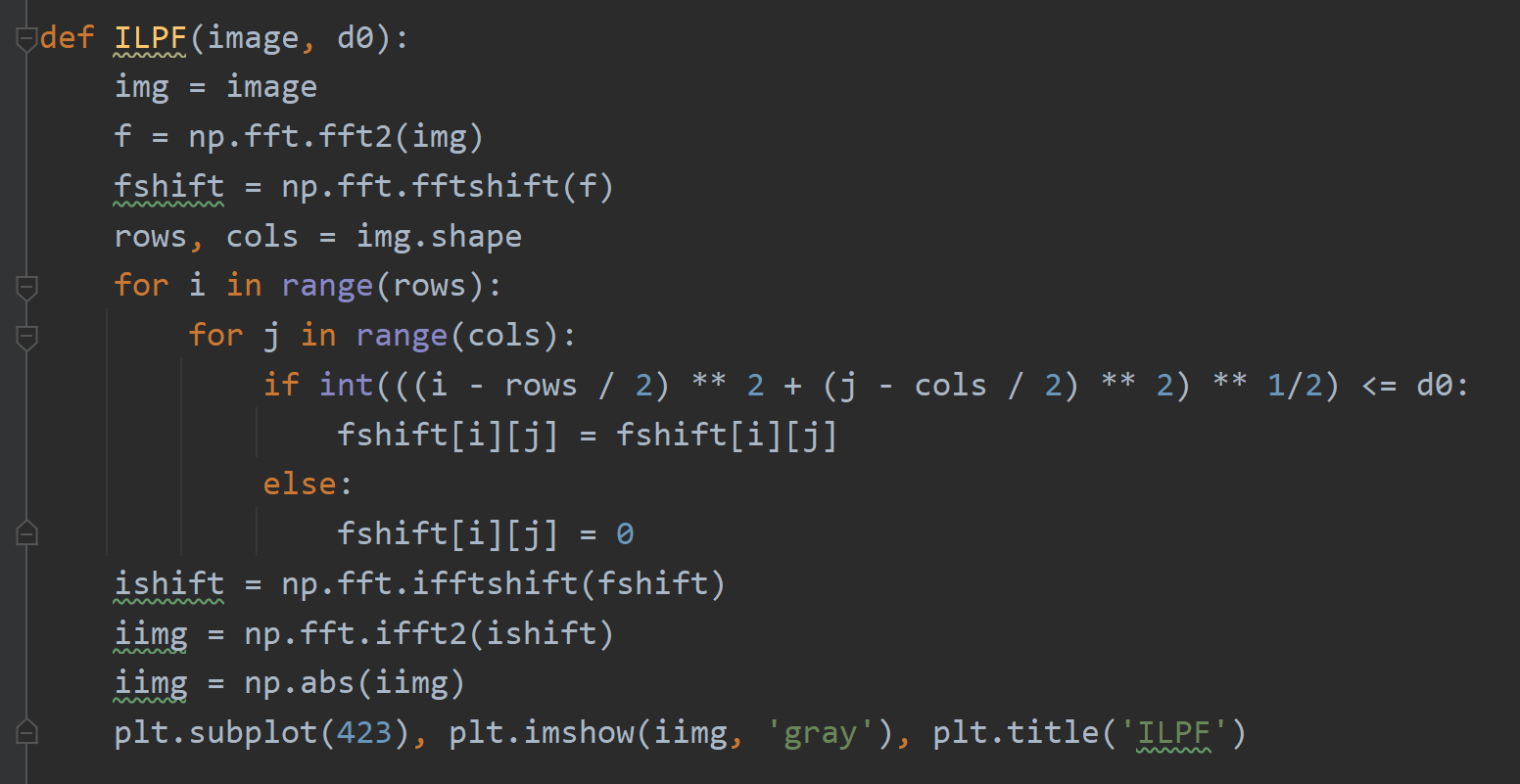
****

图4 理想滤波器实现

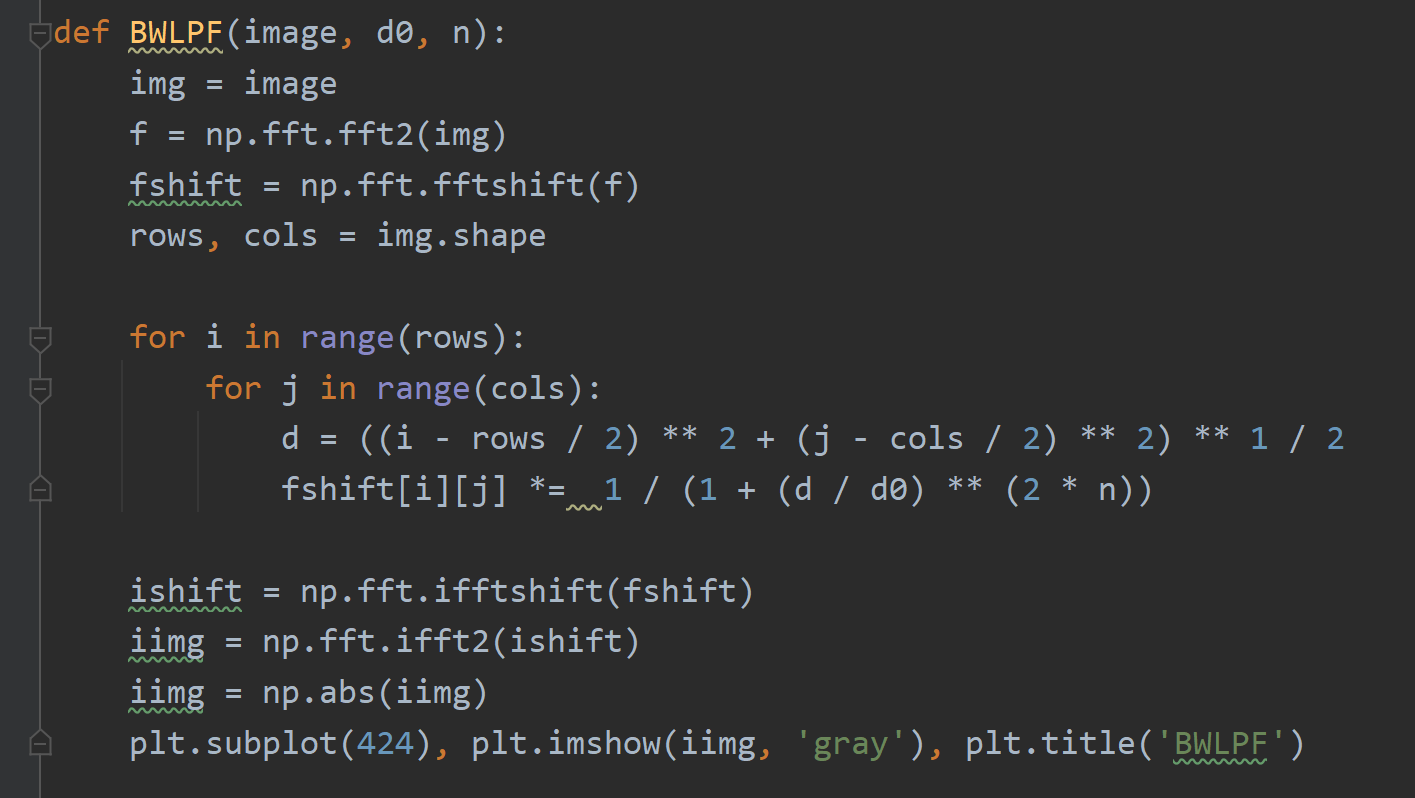
****

图5 butterworth滤波器实现

1. **编写理想滤波器和巴特沃斯高通滤波器函数**

**和2同理，仅需改变判断符号和d和d0的顺序即可。**

1. **编写同态滤波函数**

**对函数取以e为底的对数，然后进行类**butterworth**滤波器的操作，然后取以e为底的幂。**

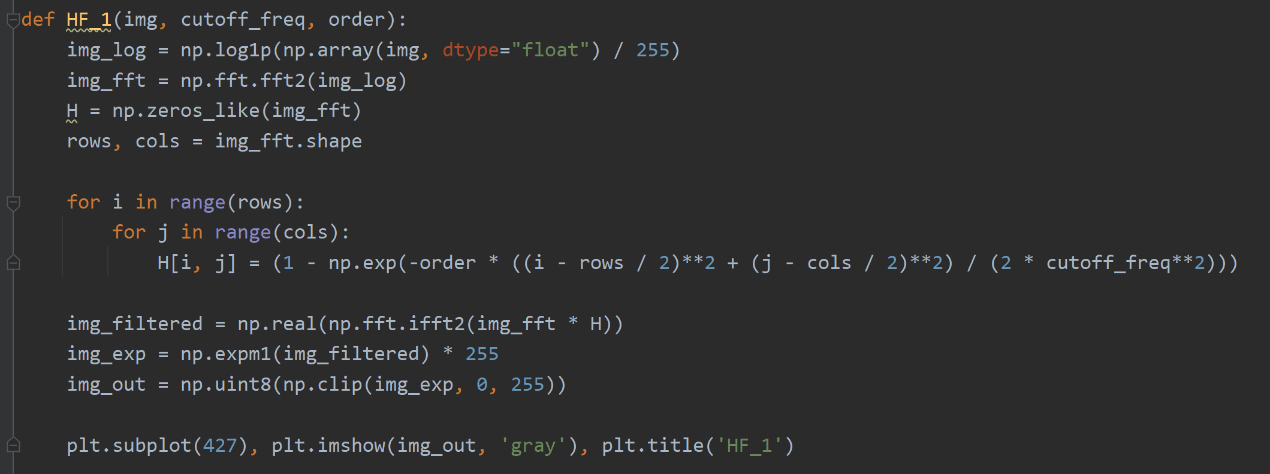
****

图6 同态滤波函数实现

**四、实验结果：**

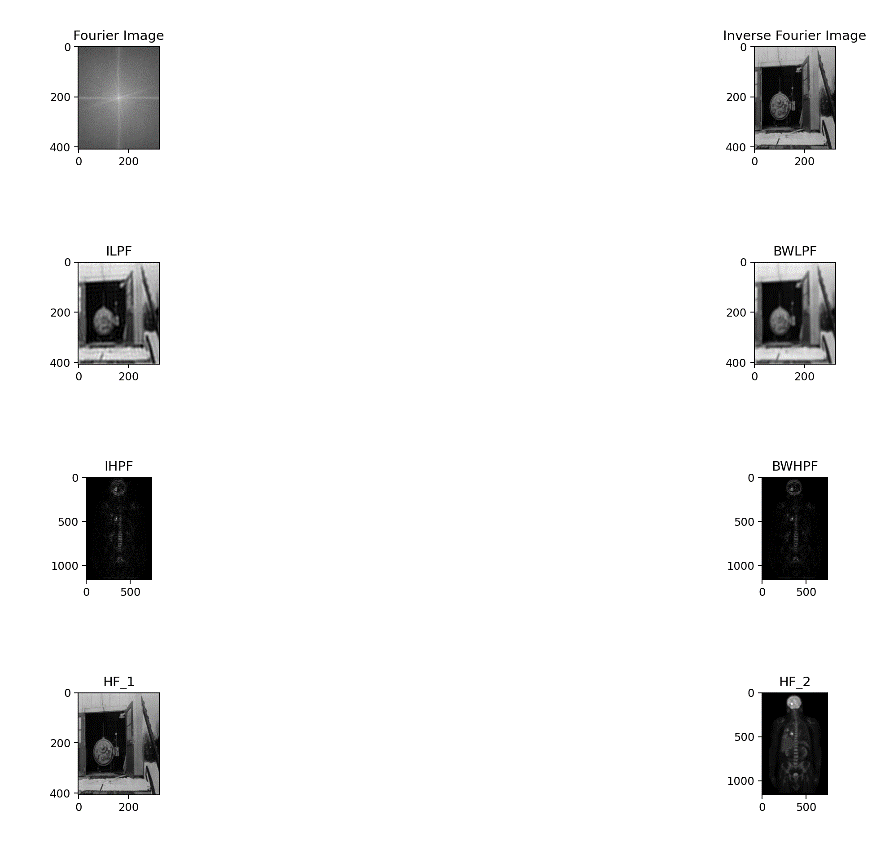
****

图7 实现结果

**如图可以看出可以看到fft之后的图像复原和之前的图像仍有细微区别；对理想和butterworth滤波器取不同的d0和参数都会对时域结果有一定的影响；使用同态滤波的带通函数使得图像增强效果最好。**

**五、结果分析及实验小结：**

编程中问题及其解决：fft实现时数组报错，解决方法：对数组末尾补零，使得其为二的幂。

编程中的注意事项：注意数组的大小

**六、小组成员分工**

**由徐广玄独立完成**