

程序运行说明文档

编写人：贝元琛 2018053293 计算机科学与技术

程序测试运行环境：

操作系统：Windows10, 64 位（在满足所需库环境的条件下, Linux 系统下也可运行）

处理器：Inter(R) Core(TM) i7-8550U CPU

编程语言：Python 3.8

所需安装的库函数：

- (1) numpy 1.20.2, 用于进行算法涉及到的相关矩阵运算
- (2) PIL 8.1.2, 用于进行图片的读取、显示与保存
- (3) matplotlib 3.4.0, 用于进行实验结果图表的绘制、显示与保存

实验程序整体文件结构：

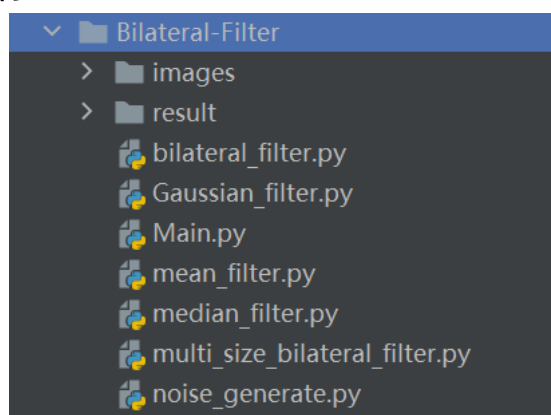


图 1：实验程序整体文件结构

其中：images 中存放了所要处理的原始图像和加噪声后的图像，原始图像均可以从开源网站 <http://www.eecs.qmul.ac.uk/~phao/IP/Images/> 中下载。

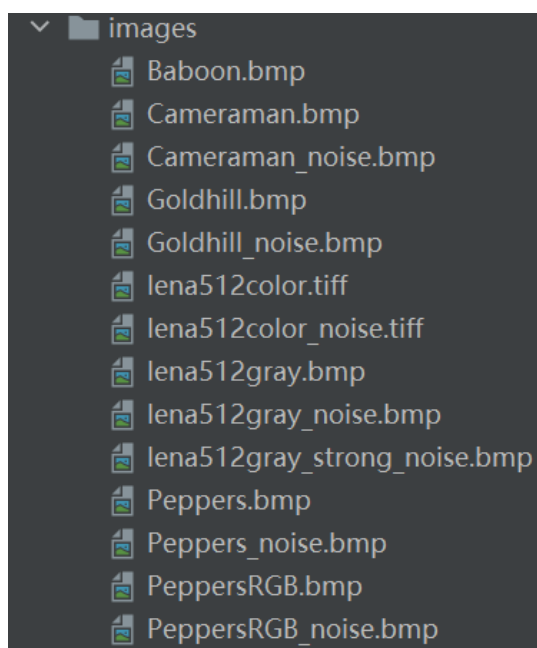


图 2：images 文件夹内容

result 文件夹中存放了处理结果图像：

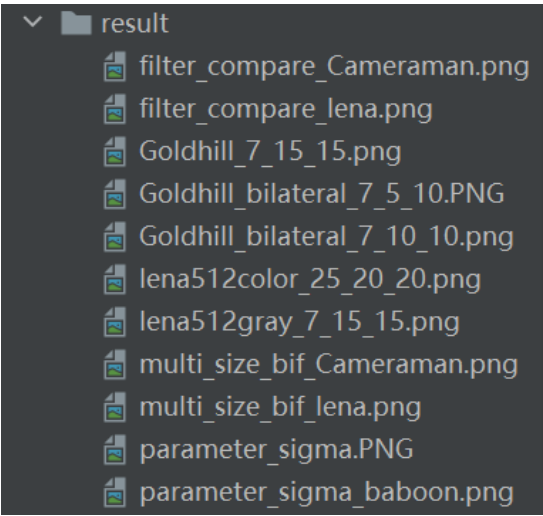


图 3: result 文件夹内容

编写的各功能模块的作用如下表所示：

模块名	作用
Main.py	调用其他函数模块中的各功能函数，实现对应的实验内容
bilateral_filter.py	包含处理灰度图的双边滤波器函数和处理彩色图的双边滤波器函数及相关所需的功能函数
Gaussian_filter.py	包含高斯滤波算法函数，用于对于不同去噪滤波算法的对比探索
mean_filter.py	包含均值滤波算法函数，用于对于不同去噪滤波算法的对比探索
median_filter.py	包含线性中值滤波算法函数，用于对于不同去噪滤波算法的对比探索
multi_size_bilateral_filter.py	包含所实现的多粒度双边滤波算法函数，用于对双边滤波算法的改进进行探索
noise_generate.py	包含为原始图像生成噪声点的函数，用于生成带噪声的图像

表 1: 编写的各个模块的作用


其中：

- (1) 每个模块都在模块最后的 main 函数中单独编写了相应的测试代码（如下示意图所示），因此各模块可单独运行测试。

```
138 ▶ if __name__ == '__main__':
139     #test
140     init_image_path = "./images/Cameraman_noise.bmp"
141     kernel_size = [5, 5]
142     sigma = [10, 10]
143     filtered_image = bilateral_filtering_gray(init_image_path, kernel_size, sigma)
144     filtered_image.show()
```

图 4: 模块 main 函数测试代码示意图

- (2) Main.py 中调用了编写的其他各功能模块, 编写了 4 个核心实验模块的实验代码, 每个编写的实验模块测试代码都通过注释进行了相应的实验说明。

- 
1. 实现双边滤波算法并记录实验效果
 2. 探索双边滤波模板的不同参数对图像的处理效果变化 (探索1)
 3. 不同去噪滤波算法的处理效果对比 (探索2)
 4. 双边滤波算法改进: 多粒度双边滤波算法 (探索3)

- (3) 在拥有 python 编译器如 pycharm, spyder 等的环境下, 可直接在编译器中点击相应的运行按键来运行代码。

在满足环境配置的 linux 下也可以通过命令行进行代码的运行:

运行代码命令示例 (在 Bilateral-Filter 文件下):

```
python ./Main.py
python ./bilateral_filter.py
python ./Gaussian_filter.py
python ./mean_filter.py
python ./median_filter.py
python ./multi_size_bilateral_filter.py
python ./noise_generate.py
```