

目录

实验报告	1
一、总体设计	1
1. 需求规定	1
2. 运行环境	1
二、结构	1
1. 基本思路	1
2. 系统框架	1
1. 模块设计说明	1
1-1. myWidget类	2
1-2. picForm类	2
2. 功能模块设计（picForm类内函数）	2
2-1. bmp读取	2
2-2. 非标准格式图片读取	2
2-3. 傅里叶变换与反变换	3
2-4. 直方图均衡化	3
三、测试结果	3
1. bmp图片显示	3
2. 非标准格式图片显示	4
3. 傅里叶变换、反变换	4
4. 直方图均衡化	5

实验报告

一、总体设计

1. 需求规定

程序按照用户需求, 执行读入非标准格式图像、读入(C++实现)bmp格式图像、傅里叶变换、傅里叶反变换、直方图均衡化等操作。

2. 运行环境

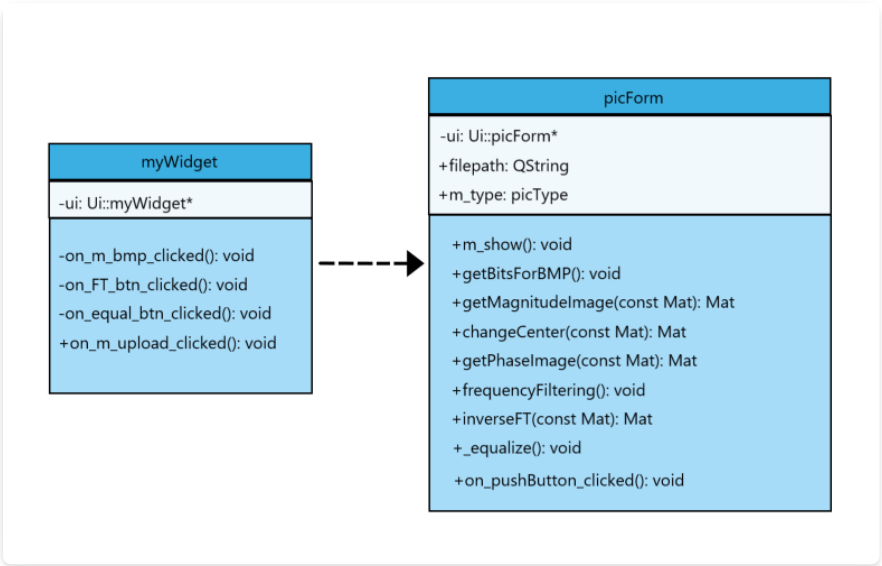
Qt(5.12.0)以及opencv(4.5.4)

二、结构

1. 基本思路

1. 由myWidget类实现与用户之间的交互;
2. 由picForm类实现相应功能的实现以及输出。

2. 系统框架



1. 模块设计说明

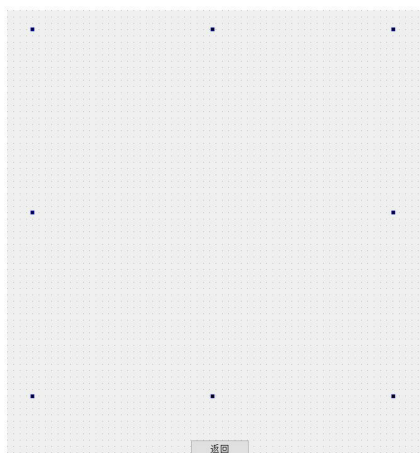
1-1. myWidget类

1. 主要负责实现了与用户之间的交互。
2. 输入的图像由用户指定, 具体实现哪一项功能由用户指定。
3. 界面设计:



1-2. picForm类

1. 功能的主要实现类。
2. 主要负责实现具体功能, 并将生成的图像(用户需要的)进行显示。
3. 界面显示:



2. 功能模块设计(picForm类内函数)

2-1. bmp读取

1. 分别读取图像的位图文件头和位图信息头(图像的宽、高)的数据。
2. 使用fread读取位图文件头和位图信息头, 用fwrite写文件, 每行的数据是宽度* 3个(像素分别由RGB三个分量组成)。
3. 读取的图片应为24位 的bmp图。

2-2. 非标准格式图片读取

1. 直接使用了Qt内的函数, 将获取到的.jpg或.png或.bmp类型的图片加载至窗口内的label中。

2-3. 傅里叶变换与反变换

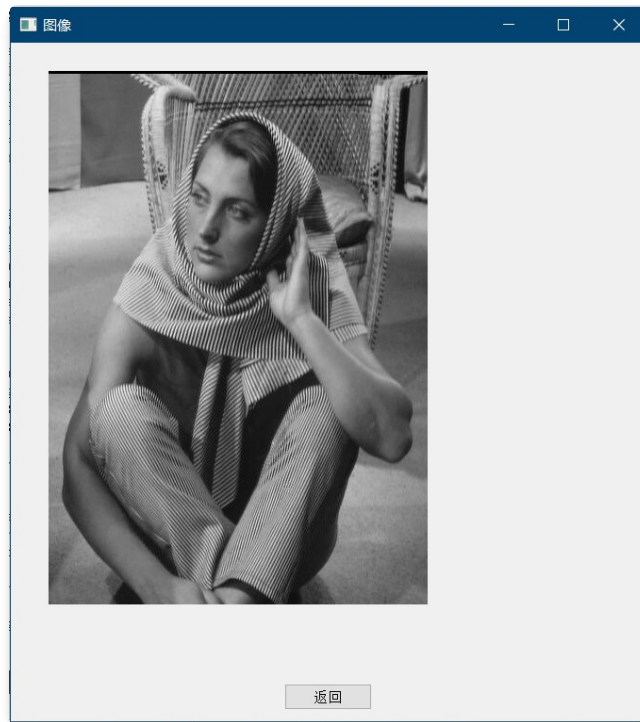
1. 该部分使用了opencv内置的相关函数dft和magnitude等。
2. 变换过程:
 1. 使用getOptimalDFTSize()获取DFT后结果的最优尺寸大小;
 2. 根据上述尺寸, 使用copyMakeBorder()填充Mat;
 3. 使用上一步生成的Mat, 通过merge()得到一个双通道的Mat;
 4. 使用dft()进行傅里叶变换;
3. 显示结果:
 1. 使用magnitude()得到结果幅度谱;
 2. 将幅度值转换到对数尺度, 使振幅较低的成分相对高振幅成分得以拉高;
 3. 进行归一化处理, 将矩阵转变为可视的图像格式;
 4. 显示结果。

2-4. 直方图均衡化

1. 该部分使用了opencv内的函数实现。
2. 步骤:
 1. 加载原图像;
 2. 转为灰度图;
 3. 对直方图均衡化(使用equalizeHist());
 4. 显示结果图像。

三、测试结果

1. bmp图片显示



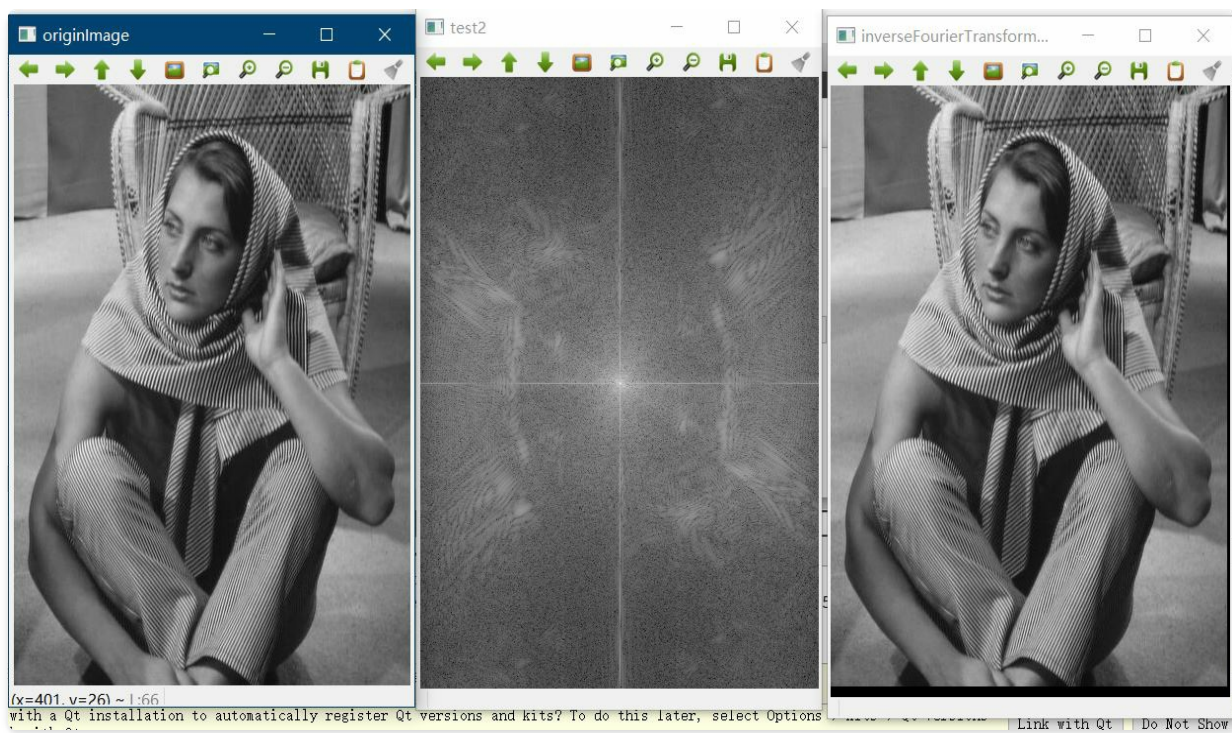
bmp格式

2. 非标准格式图片显示



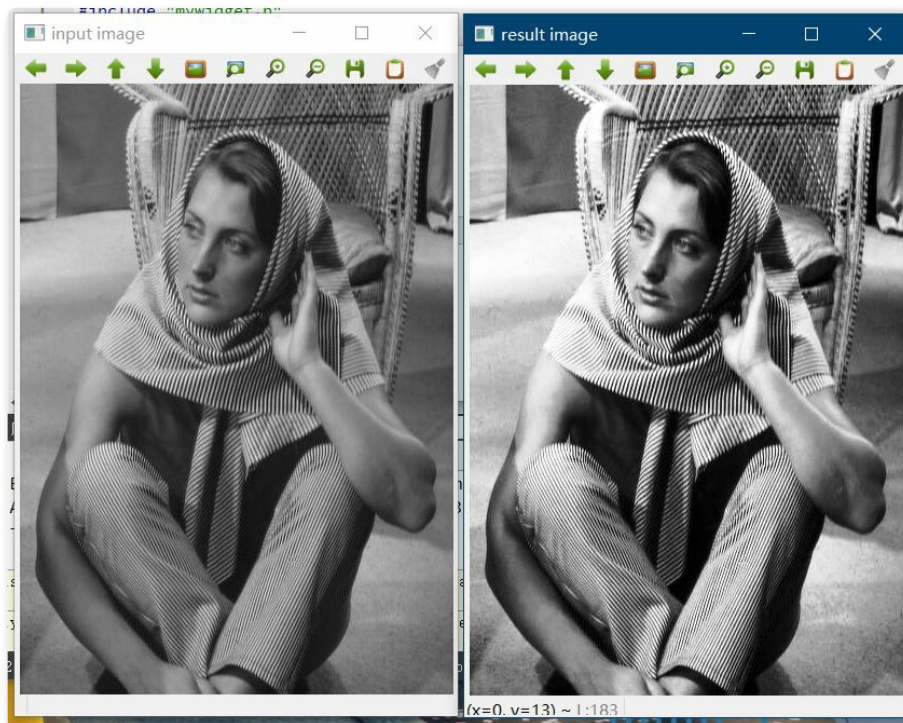
jpg格式图片

3. 傅里叶变换、反变换



FT、FFT

4. 直方图均衡化



直方图均衡化