目录

实验报告

一、总体设计

1. 需求规定

实现灰度图像的基本几何变换(旋转、缩放)。
要求:

1. 采用人机交互方式读取灰度图像文件;
2. 选择人机交互方式确定基本几何变换参数, 包括旋转角度和缩放比例参数;
3. 使用双线性插值完成输出图像像素灰度值的估计;
4. 图像的旋转默认图像的几何中心进行, 允许选择人机交互方式确认旋转中心;
5. 缩放后图像尺寸超过显示窗口尺寸时, 应具备人机交互调整显示图像区域的功能, 不允许自动缩小图像尺寸以适应窗口尺寸;
6. 程序运行时, 可以通过人机交互多次完成图像的基本几何变换。

2. 运行环境

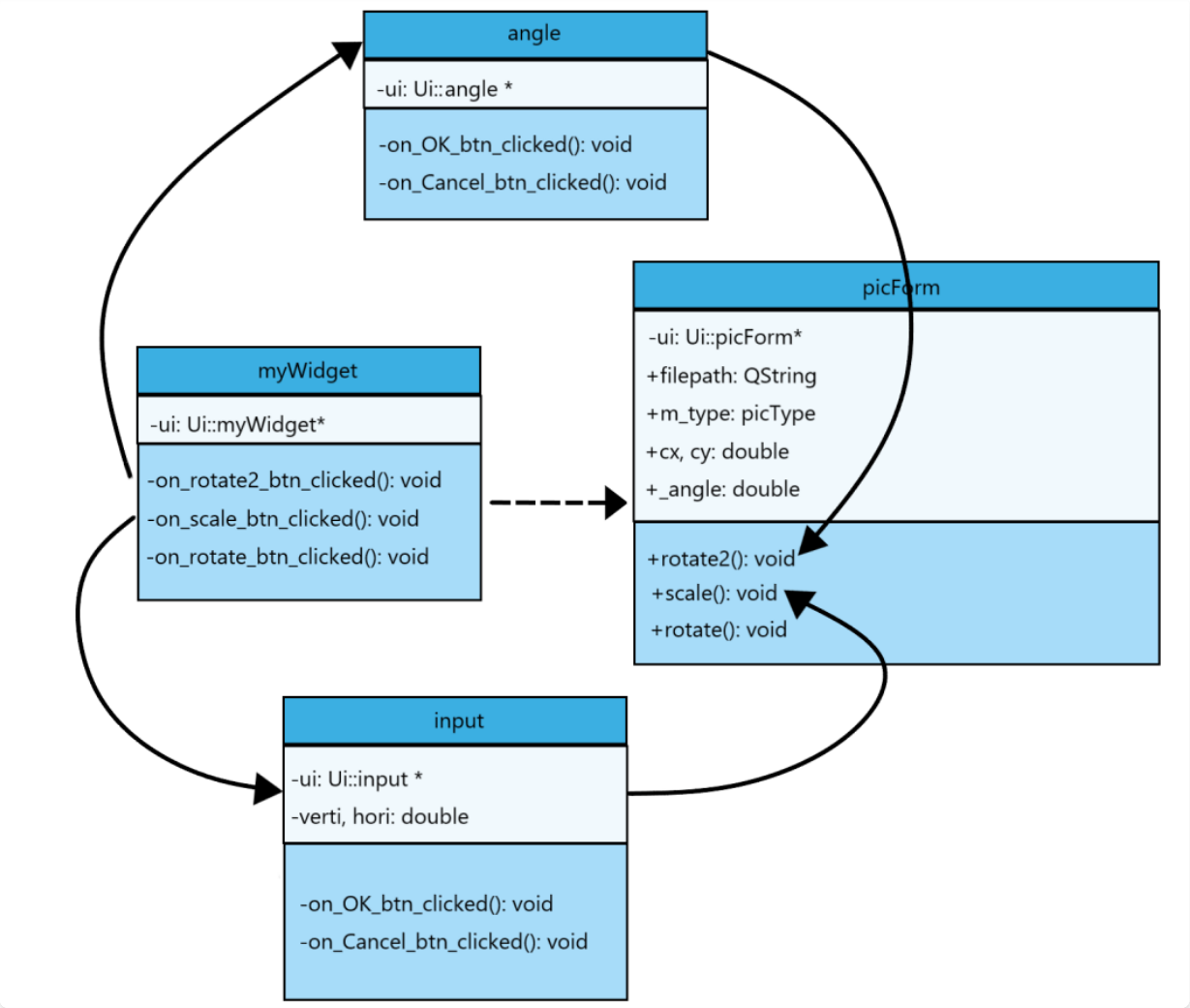
Qt(5.12.0)以及opencv(4.5.4)

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



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. 说明:
    1. 上面四个按钮均为第一次实验实现的方法, 因为本次实验延用上次实验的框架, 第一次实验的函数并未进行删除;
    2. 第二次实验下的四个按钮分别为:
        1. .bmp按钮:
            1. 在第一次实验时进行实现, 本次实验复用的一个按钮, 主要实现纯代码显示bmp格式的图像;
            2. 图像的路径由用户给出, 程序无默认路径。
        2. rotate按钮:
            1. 默认围绕图像的几何中心进行旋转的方法, 显示旋转后的全部图像;

2. 旋转角度由用户进行输入, 由Qt的QInputDialog实现与用户的交互;
3. 图像的路径由用户给出, 程序无默认路径。

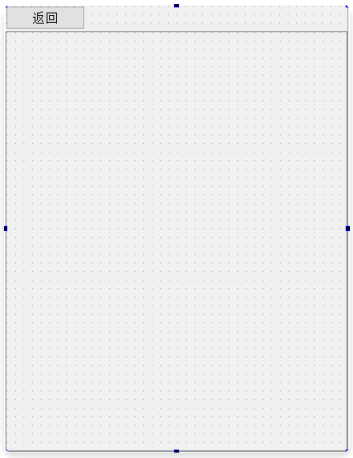
3. rotate2按钮:
    1. 围绕用户自定义的旋转中心以及交互进行旋转的方法, 只显示原图像大小的画布上的旋转后的图像;
    2. 旋转角度、旋转中心由angle类的ui进行显示并进行交互, 是自定义类;
    3. 图像的路径由用户给出, 程序无默认路径。
4. scale按钮:
    1. 按照用户给定的水平以及垂直缩放尺度进行缩放的方法, 当图像大小超过默认画布时显示滚轮, 便于用户查看局部图像, 也可以通过鼠标拖拽的方式进行显示窗口的缩放以便于查看更大面积的输出图像;
    2. 水平缩放尺度、垂直缩放尺度由input类的ui进行显示并进行交互, 是自定义类;
    3. 图像的路径由用户给出, 程序无默认路径。

4. 界面设计:

myWidget

1-2. picForm类

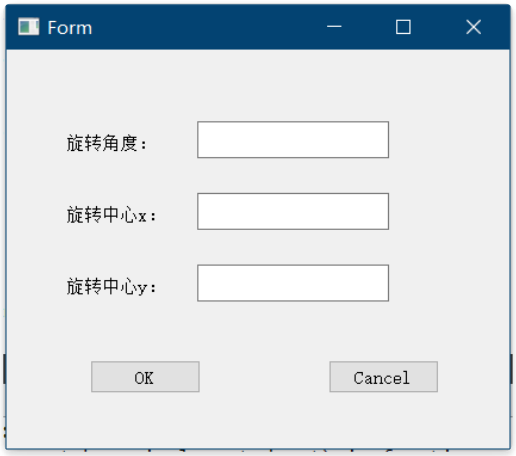
1. 功能的主要实现类。
2. 主要负责实现具体功能, 并将生成的图像(用户需要的)进行显示。
    1. 同时使用了Sroll Area实现了对大图像(超过图像显示范围的图像)的包含滚轮的局部显示, 即实现了图像查看部分的人机交互。
    2. 返回按钮返回myWidget界面, 实现了用户可以在一次程序运行过程中多次进行不同功能的查看。
3. 界面显示:



picForm

1-3. angle类

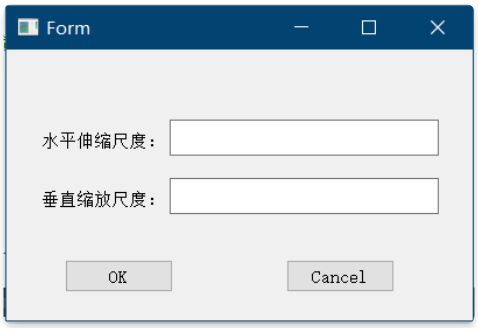
1. 旋转函数(自定义旋转中心、画布固定)的用户输入窗口, 负责实现人机交互。
2. 主要负责从用户获取有关的信息, 包括:旋转中心的x值和y值、以及旋转角度, 并将参数传至picForm中。
3. 界面显示:



angle

1-4. input类

1. 缩放函数的用户输入窗口。
2. 主要负责从用户获取有关的信息, 包括:水平的缩放参数, 垂直的缩放擦书, 并将参数传至picForm中。
3. 界面显示:



input

2. 功能模块设计(picForm类内函数)

2-1. 旋转

1. 旋转的步骤:
    1. 将输入原图图像坐标转换为笛卡尔坐标系;
        1. 图像坐标为:原点位于图片左上角, 水平向右为x轴, 垂直向下为y轴。
    2. 进行旋转计算;
    3. 将旋转后的笛卡尔坐标转换回图像坐标。
        1. 采用双线性插值算法进行插值:
            1. 为了更好的效果, 采取了原图像和目标图像几何中心对齐、将浮点运算转换成整数运算的方式。
2. 变换采用变换矩阵的方式, 即

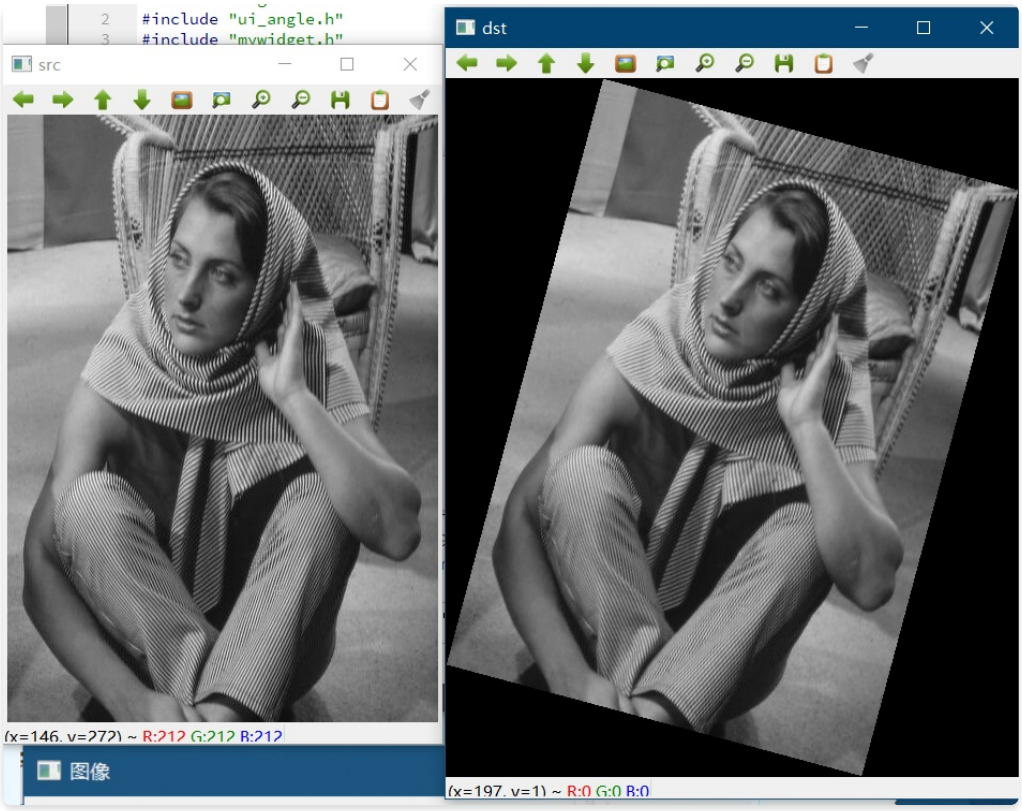
$$[x\ y\ 1] = [v\ w\ 1]\mathbf{T} = [v\ w\ 1]\begin{bmatrix} t11 & t12 & 0 \\ t21 & t22 & 0 \\ t31 & t32 & 1 \end{bmatrix}$$

2-2. 缩放

1. 图像缩放是二维的尺度伸缩变换, 基本原理就是根据原图像的像素通过一定的规则计算得到目标图像的像素值。
2. 依旧采用双线性插值算法进行求解。

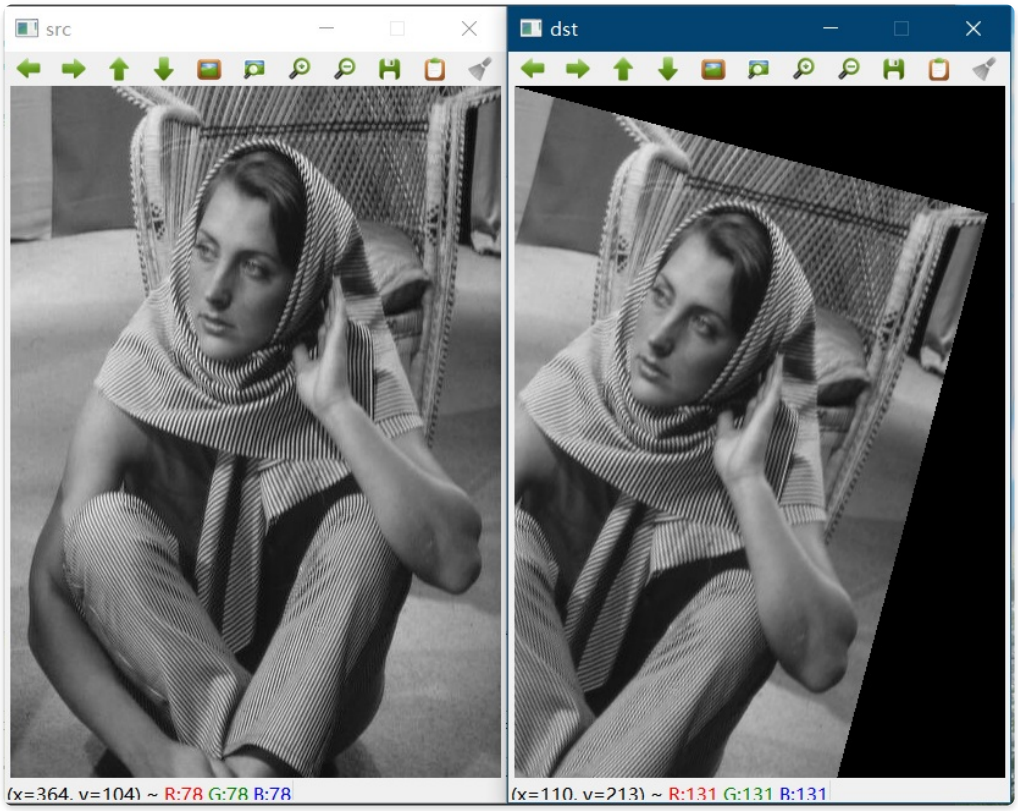
三、测试结果

1. 图片的旋转(默认旋转中心)



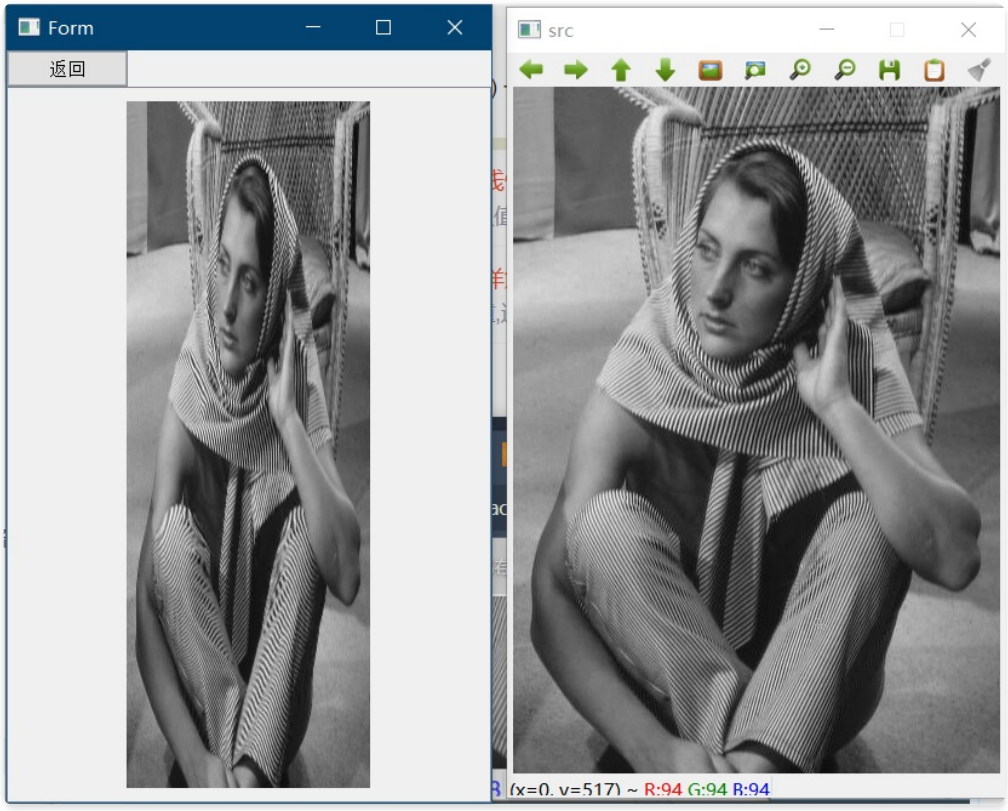
默认旋转中心, 角度=15°

2. 图像的旋转(自定义旋转中心)

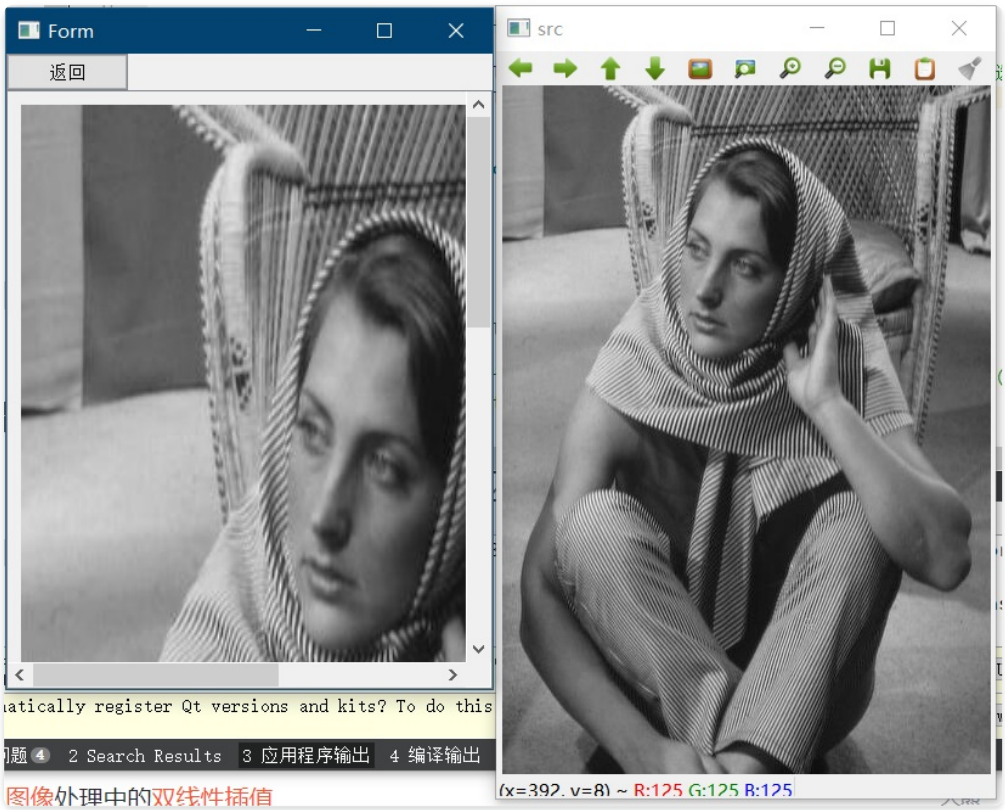


旋转中心(0 ,0), 角度=15°

3. 图像的缩小



水平缩放尺度=0.5，垂直缩放尺度=1.0



水平缩放尺度=1.5，垂直缩放尺度=2.0

3-1. 分析

1. 如下图, 双线性插值考虑了采样点周围4个直接相邻对采样点的影响, 但依旧存在不足:
    1. 首先, 显而易见地相对于临近插值法的计算量有所增大;
    2. 同时, 由于仅考虑了直接临近点的灰度值的影响, 二不是各临近点间灰度值变化率的影响, 所以一定程度上具有低通滤波器的性质, 使缩放后图像的高频分量受到损失, 图像的轮廓变得很模糊, 即存在精度下降的问题。

水平缩放尺度:10.0, 垂直缩放尺度:10.0 的图像的一部分