

杭州电子科技大学信息工程学院

《QT 图形界面编程》

软件设计报告

题 目	基于 QT 的数字图像增强软件
系	计算机系
专 业	计算机科学与技术
学 期	2019-2020-1
学 号	17905827
姓 名	唐铃沐
任课教师	孙志海
完成日期	2019 年 12 月

目录

一、软件设计任务概述	4
1.应用背景	4
2.任务	4
3.开发环境搭建	4
二、可行性研究、需求分析及分工	4
1.可行性	4
2.需求分析	4
3.分工	4
三、软件设计的基本原理和采用的主要方法(算法)与技术	4
1.设计原理	4
2.主要算法	4
(1) Gamma 变换	4
(2) 图像反转	5
(3) 灰度直方图均值化	5
(4) 高斯滤波	5
(5) Sobel 边缘检测	5
四、实现的过程与步骤	6
1.界面设计	6
2.几何变换	6
3.灰度处理	7
4.图像增强	7
5.颜色模型	7
6.输出窗口	7
7.属性窗口	7
8.绘图	7
9.其他功能	7
10.视频处理	7
11.发布名称、图标设置	7
12.跨平台	7

五、遇到的困难与获得的主要成果	8
1. 困难	8
六、测试与运行记录	8
七、结果分析与小结	13
1. 结果分析	13
2. 总结	13

自编图像处理软件 myPs

一、 软件设计任务概述

应用背景：myPs 是一个带有几何变换、灰度变换、图像增强、绘图等功能的图像编辑软件。可以广泛应用于生活、工作中，对于有平面设计、图形、绘图等需求的人群也可以选择。

任务：实现图像处理软件的基本功能，其中包括，

几何变换：图像大小变换、图像缩放、图像旋转、图像翻转；

灰度变换：线性变换、指数变换、伽马变换、二值图、灰度图、图像反转，直方图均值化；

图像增强：可调模板大小的均值滤波、高斯滤波、中值滤波，sobe 边缘检测、laplacian 检测；

RGB 模型：R 分量图，G 分量图，B 分量图；

绘画：可以通过画笔、直线和各种形状（圆、三角、长方形等）在图像上进行绘画并保存，可以调节画笔线宽和颜色；

输出窗口：显示操作记录

属性窗口：图像基本信息；

其他：各种窗口部件的隐藏和显示、全屏、关于、设置、皮肤切换、语言转换、跨平台等；

开发环境搭建：Qt5.6

二、 可行性研究、需求分析及分工

可行性：生活中越来越多的人想要让自己的照片看起来更漂亮，而主流的 Ps 一类软件较大，不轻便，对于新手不友好，难以掌握其用法；

需求：急需一款跨平台的轻便、简单的图像处理软件；

分工：开发人员 唐铃沐；

三、 软件设计的基本原理和采用的主要方法（算法）与技术

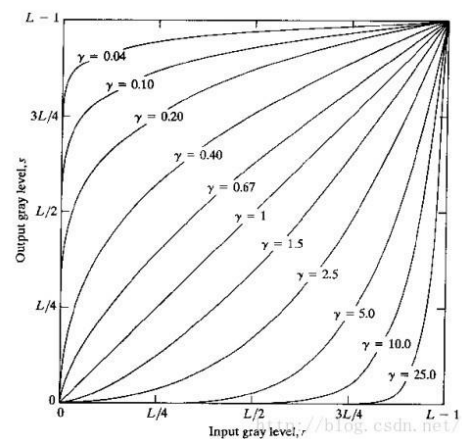
设计原理：着重于软件的轻便、容易上手、在主窗口能完成所有基本操作、图像处理准确无误；

主要算法：

伽马变换

变换公式就是对原图像上每一个像素点 RGB 值做乘积运算：

$$s = cr^{\gamma} \quad r \in [0,1]$$

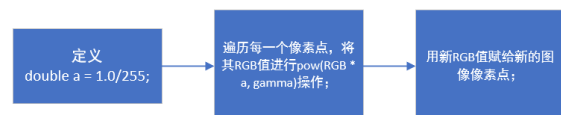


所以在实现中需要对像素值进行归一化，即乘上 $1/255$ ，C 默认 1；

实现：

1. 定义 $\text{double } a = 1.0 / 255.0$;
2. 遍历每一个像素点，分别对其 RGB 值 $\text{pow}(\text{RGB} * a, \text{gamma})$ 操作；
3. 将新值赋给图像；

流程图：



图像反转：

对每个像素点 RGB 值进行取反或者 255-RGB 值，再赋给原图像；

$$y = \sim x$$

或者

$$y = 255 - x$$

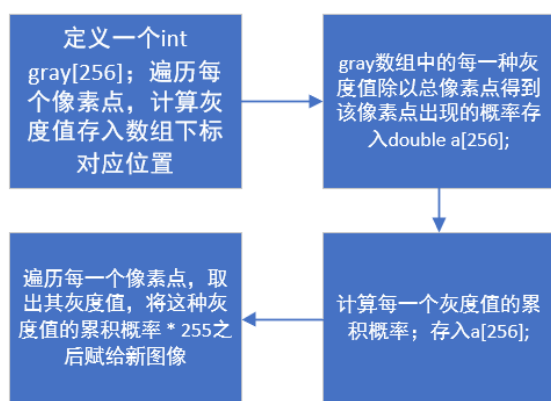
实现：

1. 遍历每一个像素点；
2. 取出 RGB 值分别取反；
3. 将取反后的值赋给新图像；

直方图均值化

1. 首先绘制图像的灰度直方图
2. 计算出每一种灰度值出现的概率；灰度值出现概率 = 这一种灰度值出现的次数 / 总像素点个数
3. 计算累计概率；灰度值的累计概率 = 这一种灰度值出现的概率与这一种灰度值之前出现的概率之和；
4. 计算各个像素点的灰度值；灰度值 = 原灰度值 * 原灰度值的累计概率

流程图：



高斯滤波：

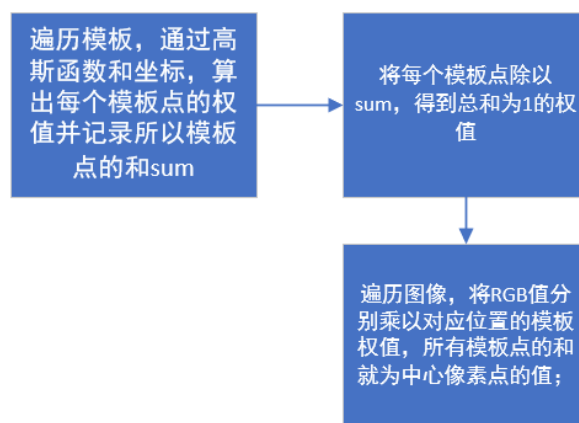
(-1,1)	(0,1)	(1,1)
(-1,0)	(0,0)	(1,0)
(-1,-1)	(0,-1)	(1,-1)

1. 选择模板大小，将模板中心点设为 (0,0)，通过高斯二维函数：

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- 计算出每个模板点的权值；
2. 此时模板中各点权值和不为1，因此将每个模板点权值除以所有模板点权值和，使它们和为一；
3. 计算像素点的 RGB 值，每个模板点分别乘以上一步计算出的权值，总和就为中心点 (0,0) 的值；

流程图：



Sobel 算子：

1. 计算每个像素点的灰度值

$$\text{Gray} = (R+G+B) / 3;$$

2. 计算梯度

(-1,1)	(0,1)	(1,1)
(-1,0)	(0,0)	(1,0)
(-1,-1)	(0,-1)	(1,-1)

$G_x = \text{Gray}(-1,1) - \text{Gray}(-1,-1) + 2 * (\text{Gray}(0,1) - \text{Gray}(0,-1)) + \text{Gray}(1,1) - \text{Gray}(1,-1);$

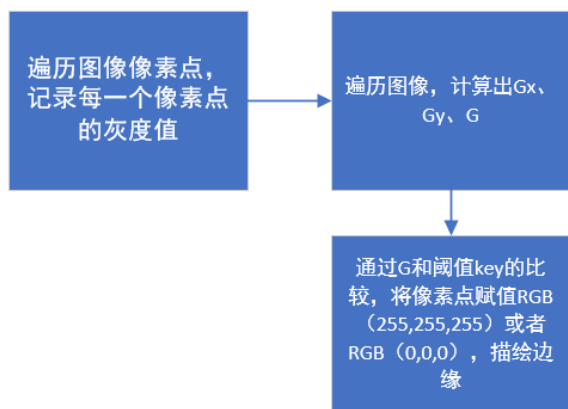
$G_y = \text{Gray}(-1,1) - \text{Gray}(1,1) + 2 * (\text{Gray}(-1,0) - \text{Gray}(1,0)) + \text{Gray}(-1,-1) - \text{Gray}(1,-1);$

$$G = \sqrt{G_x^2 + G_y^2}$$

有时为了简便计算 $G = |G_x| + |G_y|;$

3. 最后通过 G 与阈值 key 比较，确定为 RGB (0, 0, 0) 或者 RGB (255, 255, 255) 描绘边缘;

流程图:



四、实现的过程与步骤

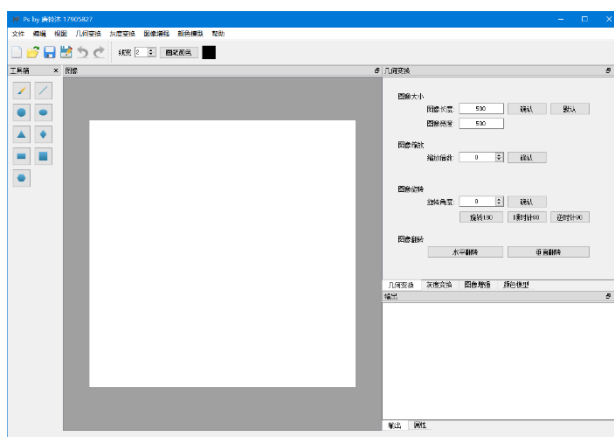
1. 界面设计

主界面由菜单栏，工具栏，以及工具箱、图像、几何变换、灰度变换、图像增强、颜色

模型、输出、属性九个 Dock 窗口；其中图像窗口为主窗口且不可关闭，几何变换、灰度变换、图像增强、颜色模型合并，默认几何窗口显示最前端。输出、属性窗口合并，默认输出窗口显示在最前端；

工具栏包括新建、打开、保存、另存为、撤销、重做、画笔宽度、画笔颜色等

菜单栏包括文件、编辑、视图、几何变换、灰度变换、图像增强、颜色模型、帮助；



2. 几何变换

图像大小：设置宽高，通过 `scaled(int width, int height)` 完成；

图像缩放：倍数放大缩小，通过 `scaled(int width, int height)` 完成；

图像旋转：旋转的角度，`QMatrix` 对象 `rotate(qreal degrees)` 完成；

图像翻转：将图像水平或竖直翻转，`Tlm` 类中 `symmetry(QImage img, bool t);`

3. 灰度变换

线性变换：通过 $y = kx + b$ 对像素点处理，`Tlm` 类中 `linearImage(QImage img, double k, int b)` 完成；

指数变换：通过 $y = x^a$ 对像素点处理，`Tlm` 类中 `expImage(QImage img, double t)` 完成

伽马变换：通过 $y = cr^{\text{gamma}}$ ， r 范围

为 [0,1] 对像素点处理，Tlm 类中 gammaImage(QImage img,double gamma)完成；

二值图像：像素点灰度值和阈值比较，赋值白或黑，Tlm 类中 TowValue(QImage img, int value)完成；

灰度图像：通过 $Gray = (R+G+B)/3$ ，赋值给像素点，Tlm 类中 Graylevel(QImage src)完成；

反转变换：通过对像素点取反对像素点进行处理，Tlm 类中 reverseImage(QImage img)完成

直方图均值化：见主要算法一栏；

4. 图像增强

图像平滑：

均值滤波：将模板像素点取平均，赋给中心点；Tlm 类中 averageImage(QImage src, int length)完成；

高斯滤波：将模板像素点通过二维高斯函数加权计算后赋给中心点，Tlm 类中 gaussImage(QImage image, int length, double sigma)完成；

中值滤波：将模板像素点排序，取中间值赋给中间点，Tlm 类中 medianImage(QImage src, int length)完成；

图像锐化：

Sobel 边缘检测：通过 sobel 算子对图像进行边缘检测，Tlm 类中 sobelImage(QImage image, int key)完成；（详细过程见“主要算法”一栏）

Laplacian 边缘检测：用 Laplacian 算子对图像进行边缘检测，Tlm 类中 laplaImage(QImage img, int key)完成；

5. 颜色模型

RGB 模型：通过仅保留 R、G 或者 B 值，提取 R、G、B 分量图，Tlm 类中 splitBGR(QImage src, int color)完成；

6. 输出窗口

通过 QLabel 的 append 方法，在每次图像处理后进行输出操作；

7. 属性窗口

每次显示图片后更新图片信息；

8. 绘图

通过 paint 类实现基本绘图功能；

9. 其他功能（全屏、窗口部件隐藏显示、关于、设置、换皮肤、中英文转换）

全屏：通过 showFullScreen() 和 showNormal()实现全屏和退出全屏；

关于：自定义窗口，显示软件信息和作者信息；

设置：自定义窗口，可进行皮肤切换、语言切换；

10. 视频处理

创建视频播放 Dock 窗口和视频控制 Dock 窗口。播放窗口进行视频正常播放，可以停止、暂停、调节亮度、对比度、色相、饱和度；控制窗口是控制图片帧在 label 上的显示，可以通过暂停对这一帧进行上述图像各种处理，将修改后的图片帧保存，预览按钮可以播放刚刚修改过的视频，此外还能输入帧数进行查看。

11. 发布名称、图标设置

略

12. 跨平台

在 Ubuntu 虚拟机下载 Qt 以及配置插件等，对项目进行编译。即可运行

五、遇到的困难与获得的主要成果

困难：其实不管是界面模块、图像处理的功能模块还是视频处理模块，都没有那么顺畅的完成，大大小小很多可能印象不是很深了，所以列举一些印象深一点的困难。

Sobel 边缘检测：这一类还包括高斯滤波、直方图均值化等需要浮点数类型的处理。在方法实现中稍微没有控制好或者考虑到数据类型转换问题，就会精度下降或者图片全白全黑等问题。

中值滤波：我实现的滤波基本都是四层循环，外面两层遍历像素点，里面两层遍历模板，而中值滤波需要对模板点排序，所以我又实现了一个冒泡排序，导致一共六层循环，复杂度很高。在 3x3 等小一点的模板示例图片速度还行，但是当模板变大一点，循环次数是爆炸性的增长，所以处理时间也会非常长。

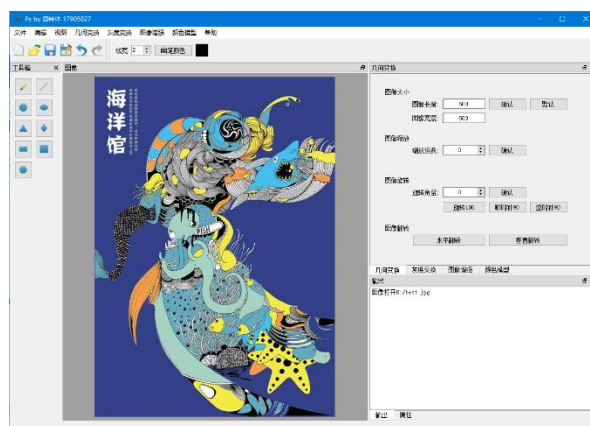
视频处理：在对播放帧的处理是，将视频转化为一帧帧图片，通过我显示图像的 Label 显示，同时存入我的 QImage 数组，我想暂停后对这一帧进行图片处理后再存入 QImage 数组，但是怎么也实现不了，因为处理完成之前，图片已经存入了，你修改后在 label 显示的图片只会被后面帧的图片冲掉。最后我的解决方案是定义 `int counters[100]` 和 `QImage editImageVideo[100]` 两个数组存放修改的帧数和这一帧修改后的图像，当我点击预览的时候，将这些修改的帧插入存视频帧的 QImage 数组后，再播放，实现了播放修改后的视频；

视频修改后预览：我是遍历 QImage 数

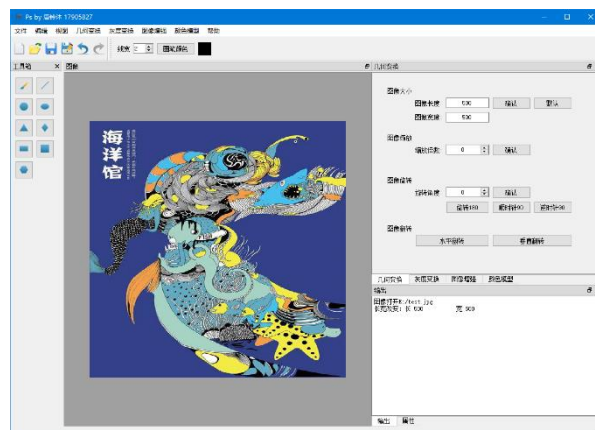
组里面的图像信息，显示到 Label，这样界面的情况是会直接显示最后一张图片，原因是因为遍历数组太快了。所以我在遍历中用到计时器，通过获取视频总时长除以总帧数得到每一帧的间隔，在遍历时加上这一间隔的时间，最后能达到还原播放的效果；

六、测试与运行记录

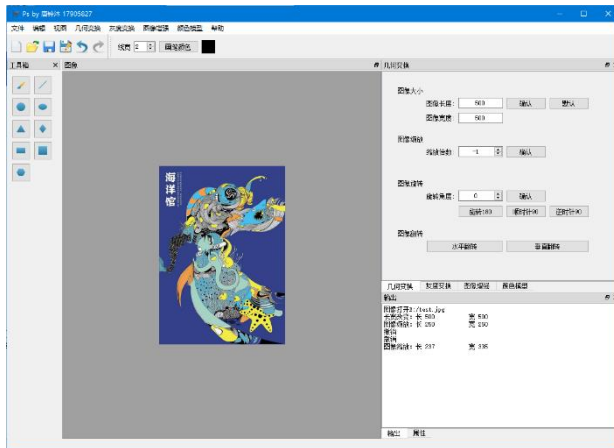
原图：



图像大小：长 500，宽 500



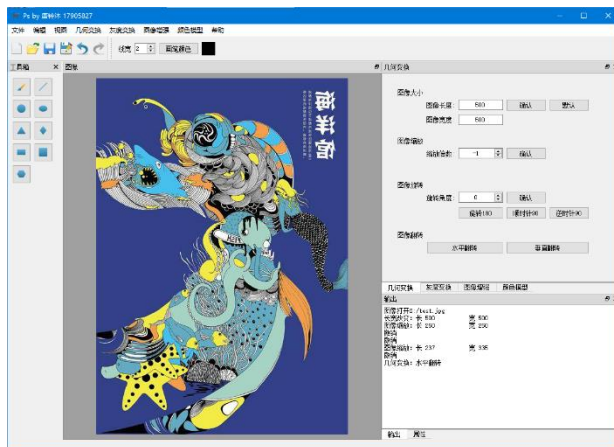
图像缩放：缩放倍数 -1



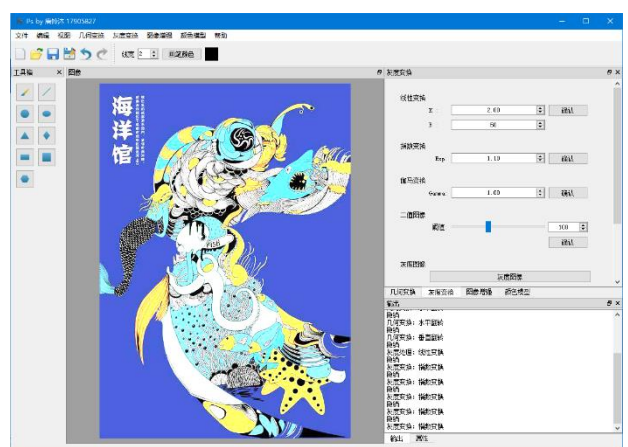
线性变换：K 2.00, B 50



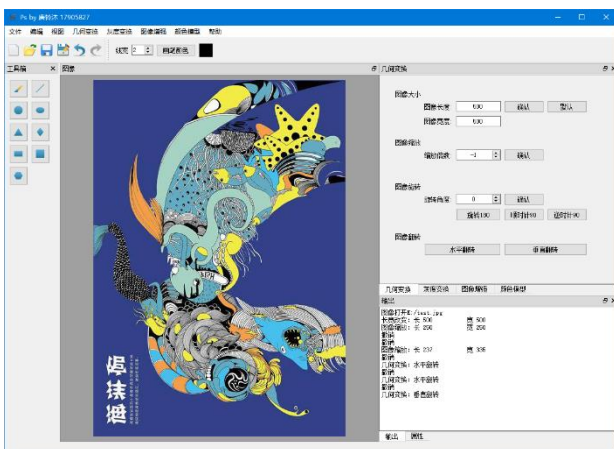
图像翻转：水平翻转



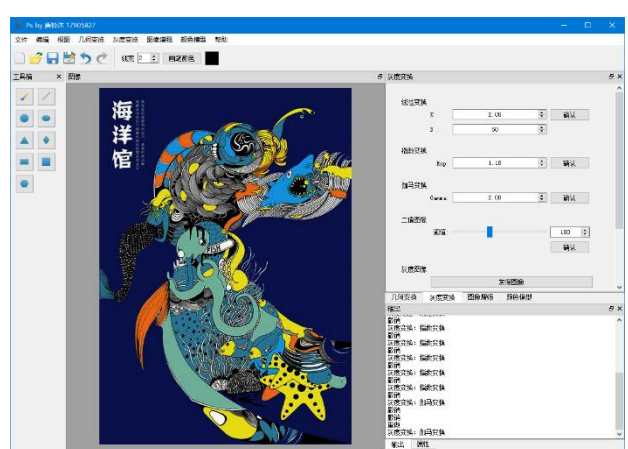
指数变换：Exp 1.10



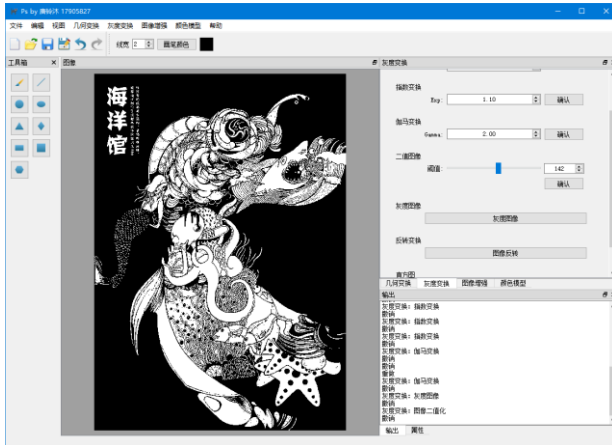
竖直翻转：



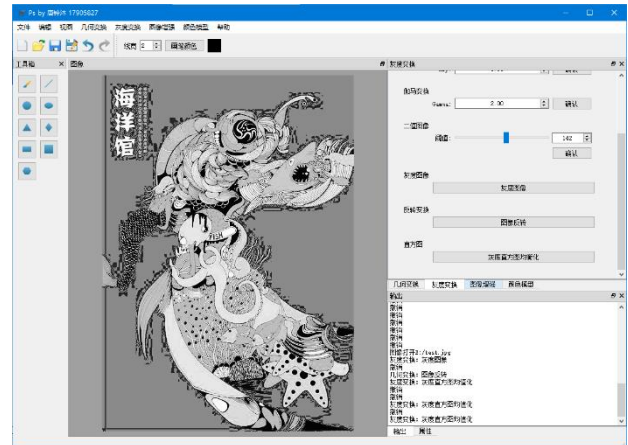
伽马变换：Gamma 2.00



二值图像：阈值 142



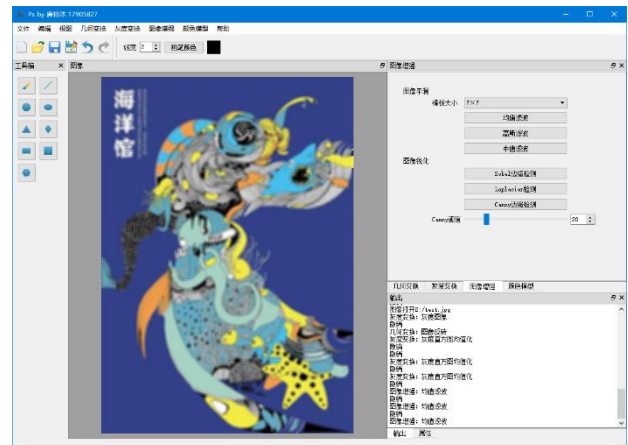
灰度直方图均值化：



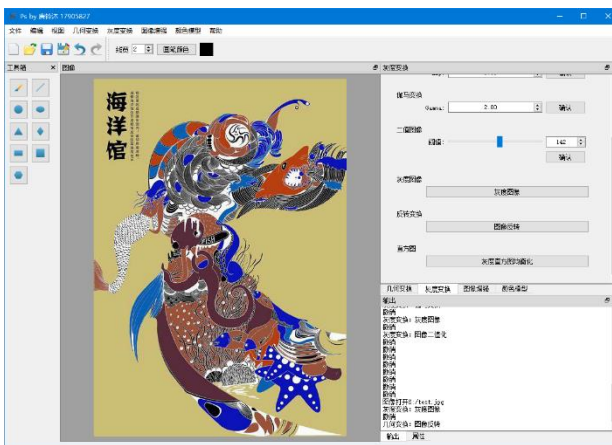
灰度图像：



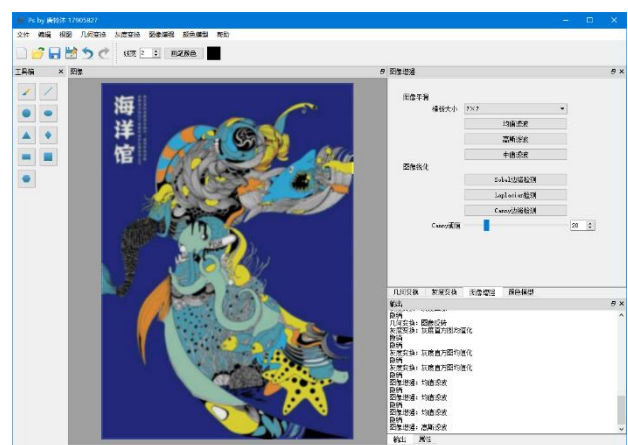
均值滤波：模板 7x7



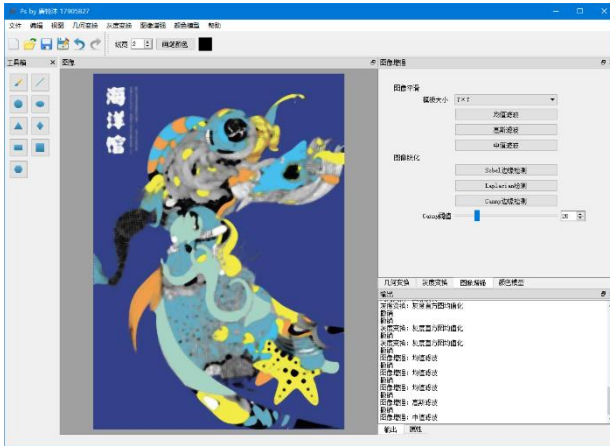
图像反转：



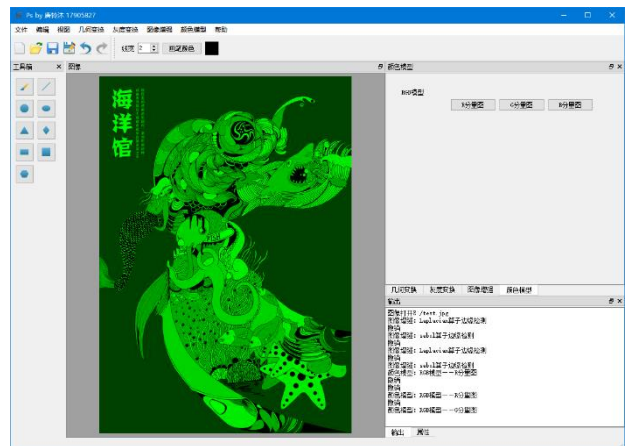
高斯滤波：模板 7x7



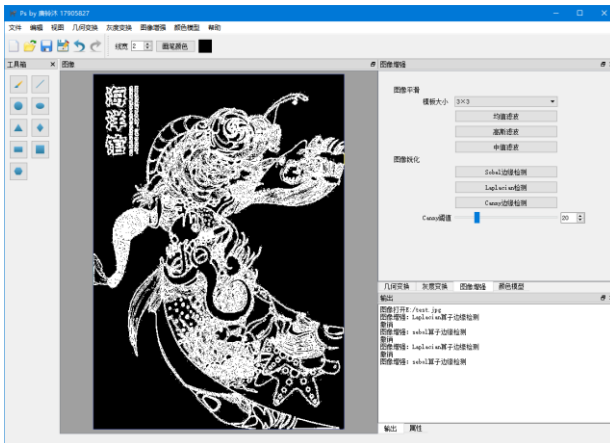
中值滤波：模板 7x7



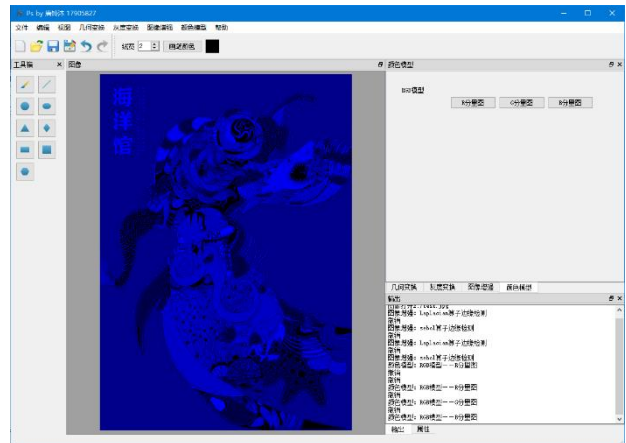
G 分量图：



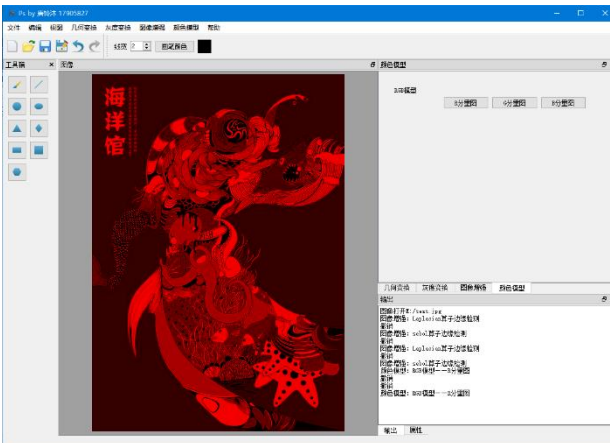
Sebel 边缘检测：



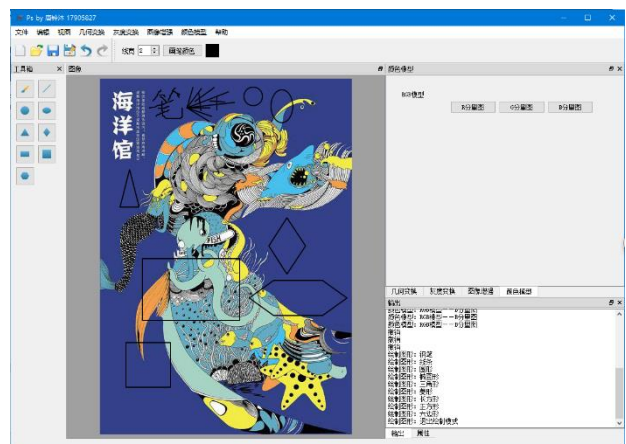
B 分量图：



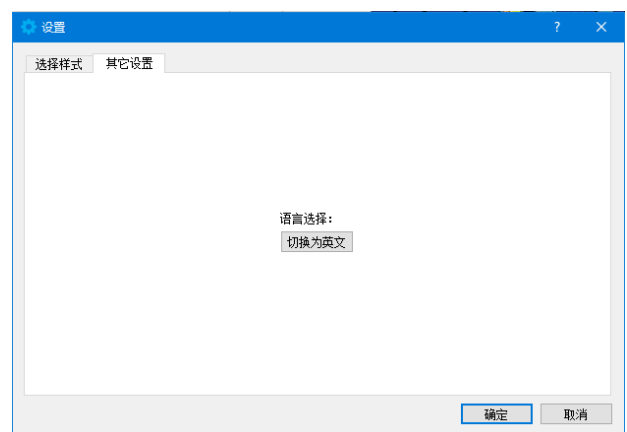
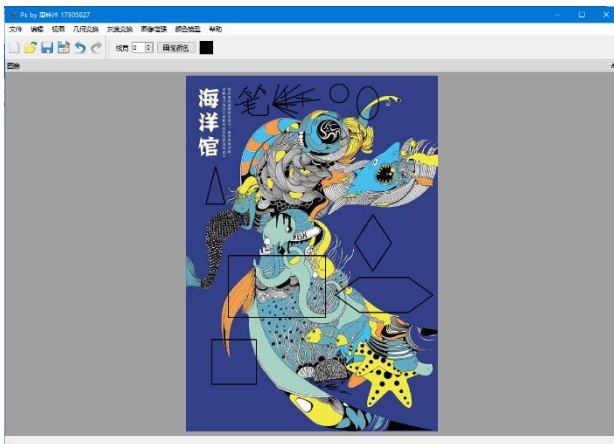
R 分量图：



绘图：画笔、直线、圆形、三角形、长方形、六边形



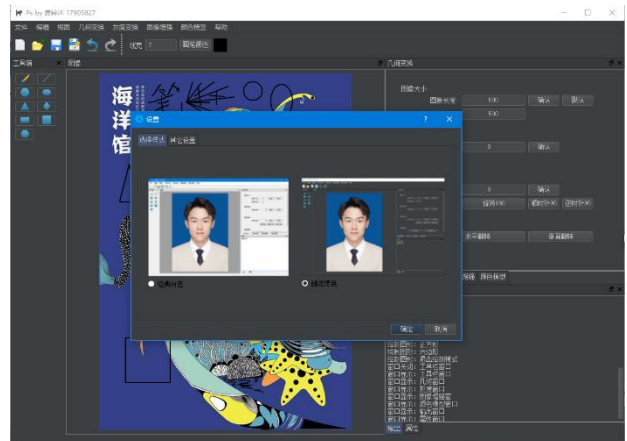
视图：所有可隐藏窗口窗口隐藏（可在视图菜单栏打开）



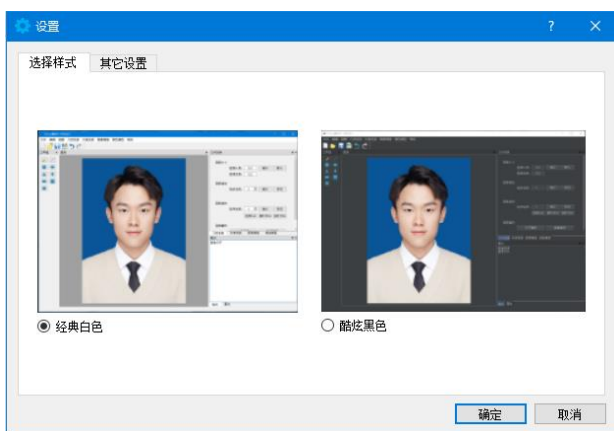
关于窗口



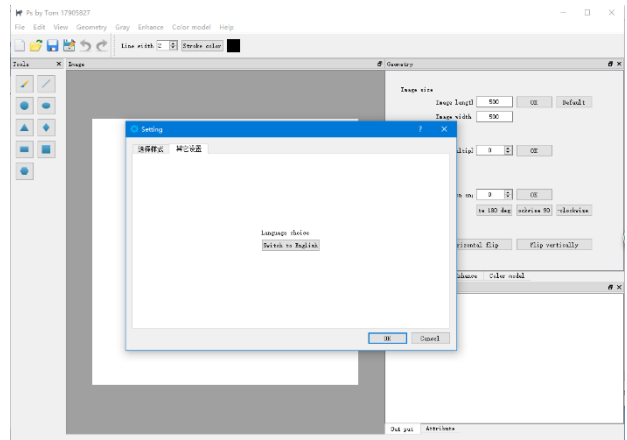
皮肤切换:



设置窗口:



切换英文:



七、结果分析与小结

结果分析：

全部实现了考核任务，并在此基础上还添加了许多额外功能，如绘图、换皮肤、灰度直方图均值化、RGB 分量图等等。但是细心的话可以看到在一些图像处理上面还不够细致，如滤波边缘我采用保留原值的方法，在多次滤波或者改用较大的模板时，边缘会有明显的层次感；又如边缘检测，还是会有少数噪声干扰；高斯滤波响应速度慢；视频取帧等很多操作流程还待完善；视频的存储等很多问题。

总结：

其实每学期期末的节奏都是很紧的，自己也多少有点适应这种情况了。这次大作业大概加班加点也有两周时间，我也很庆幸有这次大作业形式的考核。其中除了熟悉和新

学到很多关于 Qt 编程的知识，更让我兴奋的是重新认识了一遍图形图像的处理。当初上图像图像这门课也学的不是很好，这次通过 Qt 大作业的形式重新翻看了之前的书，并且把大部分图像算法都实现了一遍，有了很深的认识。此外，我最近对人脸识别等类似的基于图像识别的东西也比较感兴趣，算是丰富了一下知识储备。最后，这次图像处理软件的制作也有非常多的难题，在与同学交流和查阅资料中大多也都能想到一些解决方案，也有一小部分甚至到现在都还没解决的。所以我也深知，还有很多东西等着我去了解和学习，我也希望能更加脚踏实地充满信心的走下去。