

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	2020-2021-1
学 号	189050602
姓 名	黄湘烨
任课教师	孙志海
完成日期	2020 年 12 月

一、 软件设计任务概述（应用背景、任务、开发环境搭建）

1) 应用背景

实现一个能够进行图像处理与视频播放的软件，软件可跨平台并支持中英文显示。

2) 任务

1. 主要任务为实现图形处理与视频播放
2. 注意界面的美观和简洁，使软件拥有较好的布局
3. 实现友好的中英文界面
4. 实现跨平台（windows、mac、ubuntu 等）

3) 开发环境搭建

Windows + Qt 5.6.1 for MinGW 32 位

二、 可行性研究、需求分析及分工

首先将需求简单分为图形处理、视频播放和其他功能。

1) 图形处理

1. 通过文件对话框打开需要访问的图片(支持单选和多选功能)，打开的图片在软件中央显示
2. 对图像进行旋转、镜像翻转、亮度调节、饱和度调节、灰度化、二值化等基础操作
3. 自定义可打印的 ASCII 码（如 ABCD 等），实现 bmp 位图转指定的 ASCII 字符图，可以修改字体和字体大小以达到最佳的显示
4. 对图像进行 3×3 均值滤波、不同算子的边缘检测、边缘检测叠加原图等复杂功能
5. 保存、另存为图片（带有姓名手写水印）

2) 视频播放

1. 通过文件对话框打开需要访问的视频，打开的视频在软件在中央流畅播放
2. 支持大文件播放（如 2~3G 的电影）
3. 显示当前播放的进度时间，拖动进度条，可跳转至相应位置
4. 支持空格键控制视频的暂停与播放
5. 支持音量调节

3) 其他功能

1. 有菜单、工具条，支持快捷键
2. 软件支持中文、英文两种语言，用户可自由切换，设计并美化用户界面
3. 可执行文件有自己定义的名称、自己设计的图标
4. 利用消息提示对话框，含有“关于本软件对话框”
5. 软件可以分别在 Windows、Linux 或 Mac 系统上运行

三、 软件设计的基本原理和采用的主要方法(算法)与技术

1) 基本原理

图像操作主要基于 QImage 类和图形图像处理算法；视频播放主要基于 QMediaPlayer 类和 QVideoWidget 类

2) 主要算法

1. bmp 位图转 ASCII 编码算法：遍历图片，黑色像素输出 ASCII 码，最后返回整张图对应的字符串。

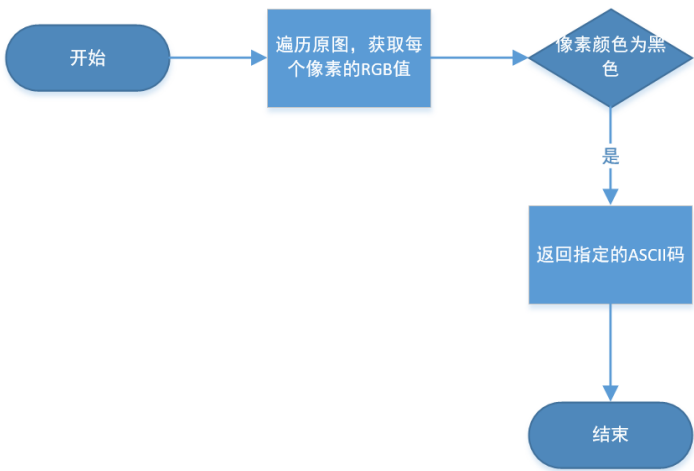


图 1bmp 位图转 ASCII 编码算法设计流程图

2. 灰度算法：遍历图片，RGB 平均值设为新 RGB。

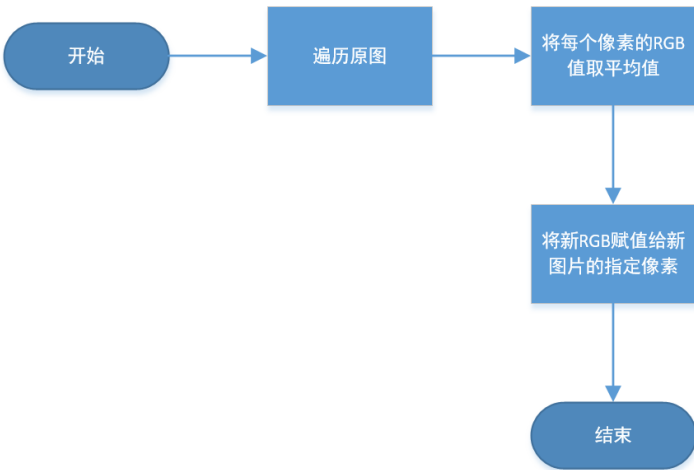


图 2 灰度算法设计流程图

3. 二值算法：遍历图片，RGB 平均值在阈值上则为白色，反之为黑色。

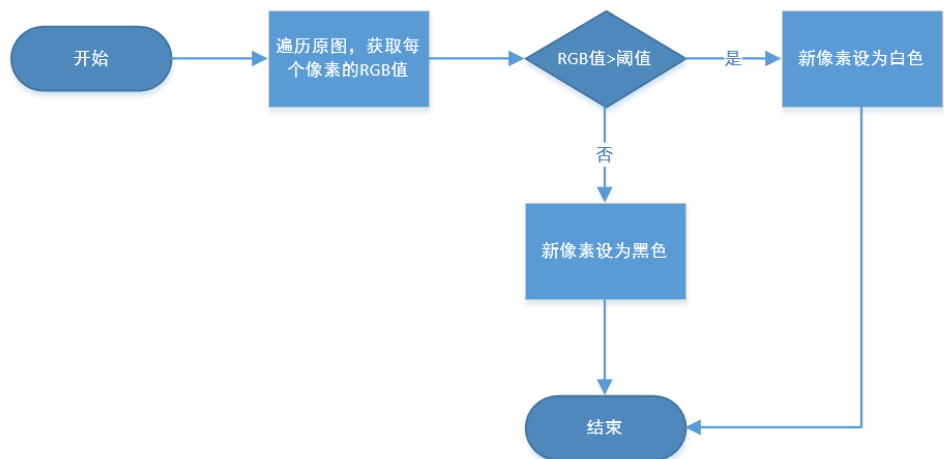


图 3 二值算法设计流程图

4. 调节亮度算法：RGB 值各加上一个数值（数值通过 horizontalSlider 滑块设置）。

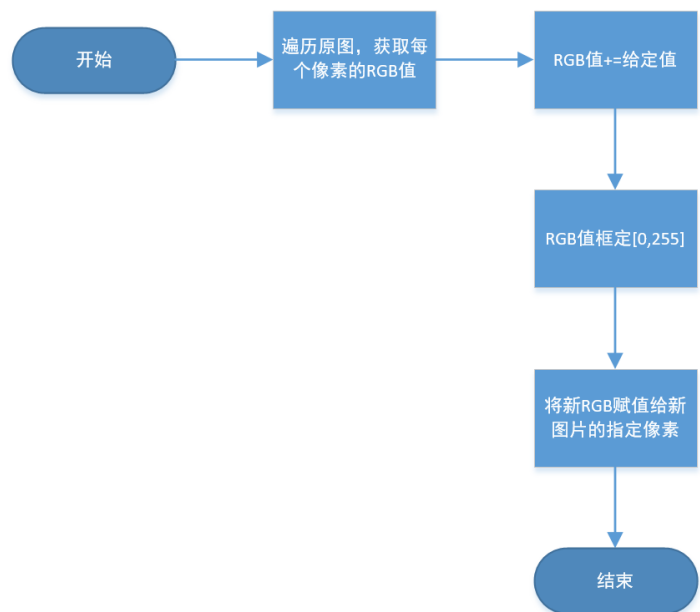


图 4 调节亮度算法设计流程图

5. 调节饱和度算法：修改 HSL 中的 S 值（数值通过 horizontalSlider 滑块设置）。

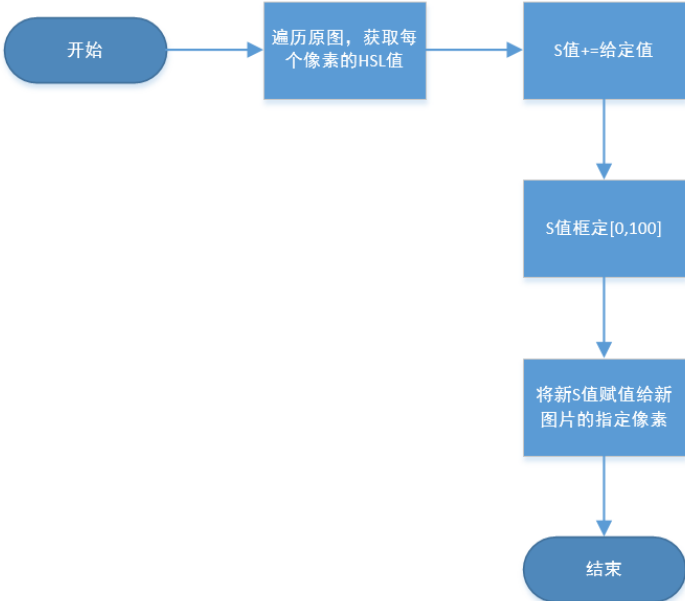


图 5 调节饱和度算法设计流程图

6. 均值滤波算法：以 3×3 像素为单位遍历图片，选择九个像素的 RGB 平均值作为中心像素的 RGB。

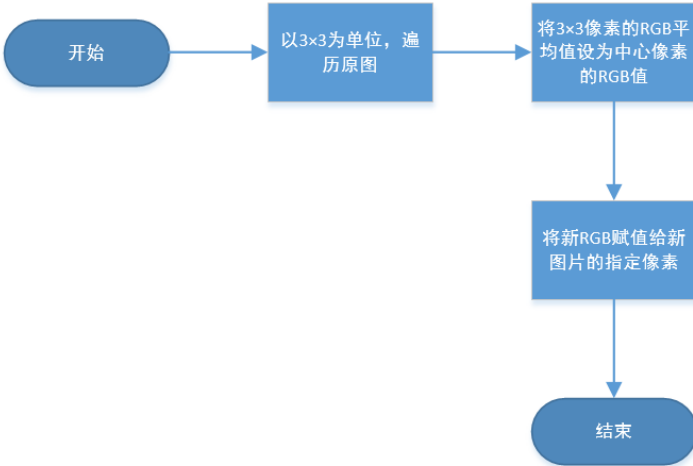


图 6 均值滤波算法设计流程图

7. roberts 算子、prewitt 算子、sobel 算子边缘检测算法：将图片转灰度图，roberts 算子（每个维度以 2×2 像素为单位）和 sobel 算子（每个维度以 3×3 像素为单位）均有 X、Y 两个维度，RGB 值乘上相应的权重，为减少计算，最后结果为 $|X \text{ 方向} + Y \text{ 方向}|$ ；prewitt 算子只计算一个维度，较简单。相较三种算子，sobel 算子效果较好。

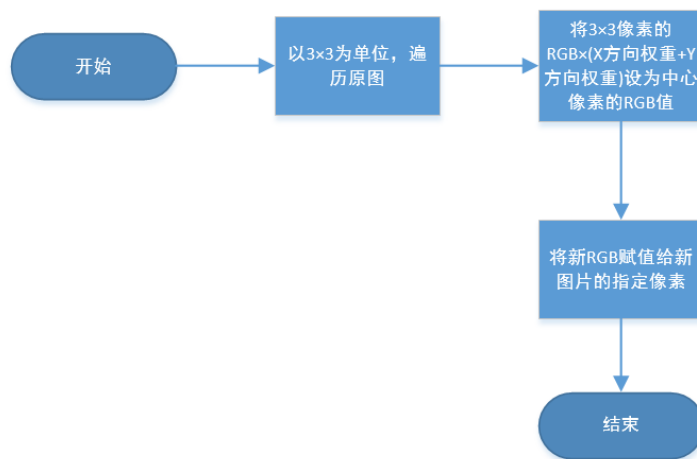


图 7sobel 算子边缘检测算法设计流程图

8. sobel 算子边缘检测与原图复合算法：修改原图的 α 值和 sobel 图的 α 值，满足两值相加为 1，可实现层叠效果。

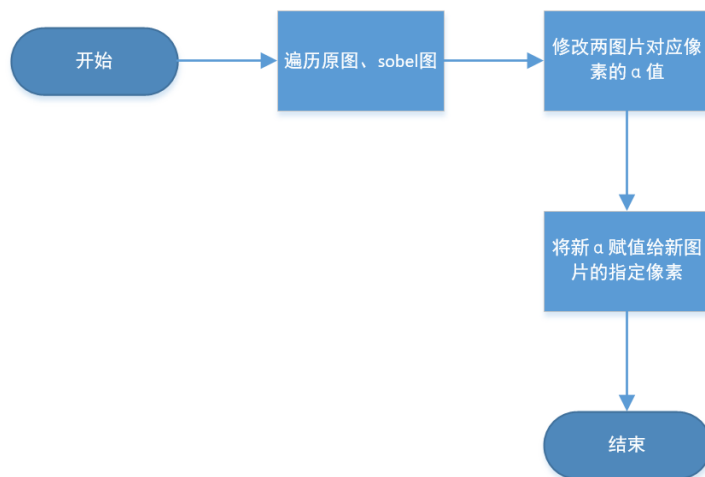


图 8sobel 算子边缘检测与原图复合算法设计流程图

9. 顺时针、逆时针旋转，水平、垂直翻转算法：使用 QImage 自带方法（速度更快）。

10. 图片加手写姓名水印算法：遍历姓名图片，找到姓名图片中的黑色像素点，然后做原图像素点的判断，如果原图像素点偏向白色（给定一个阈值来判断），则将黑色像素插入在原图上，反之插入白色像素点。

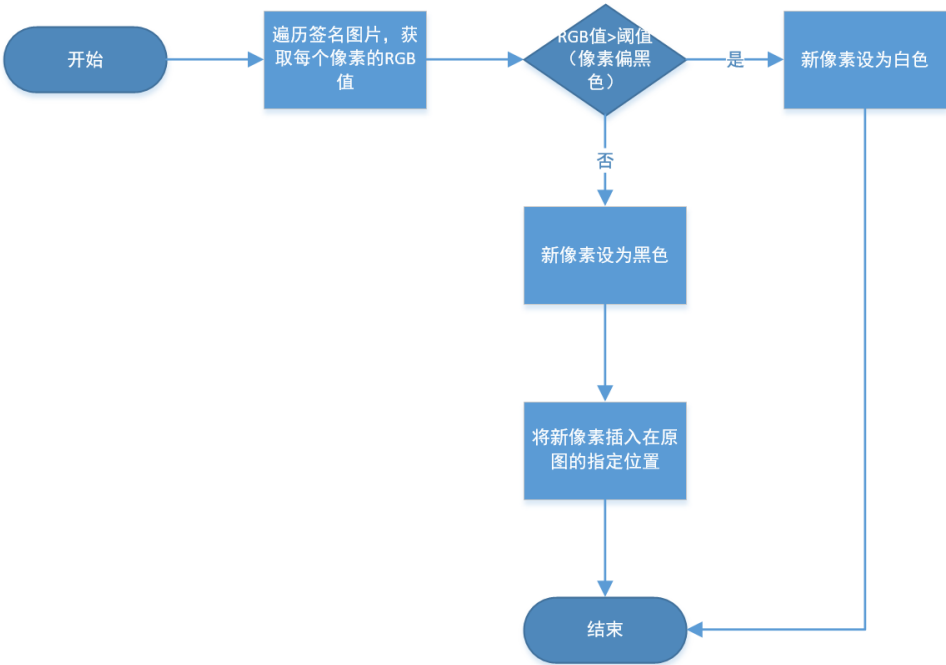


图 9 手写签名水印方式设计流程图

11. 视频当前进度时间、进度条 handler、播放内容的同步：使用定时器，选择适合的时间间隔（如 100ms，数值越小越流畅），视频开始播放时，打开定时器，每到时间间隔时，需要完成进度条 handler 移动、当前进度时间显示；视频暂停时，计时器停止计时。

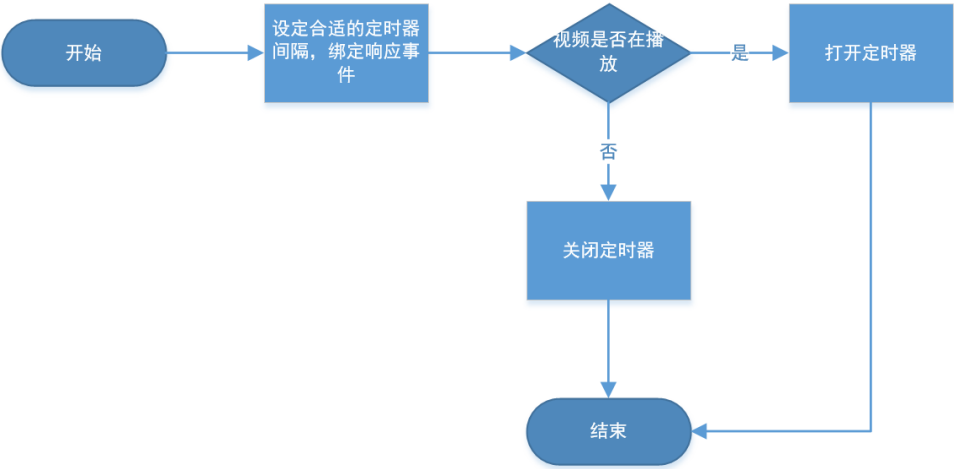


图 10 使用计时器同步内容算法设计流程图

四、 实现的过程与步骤

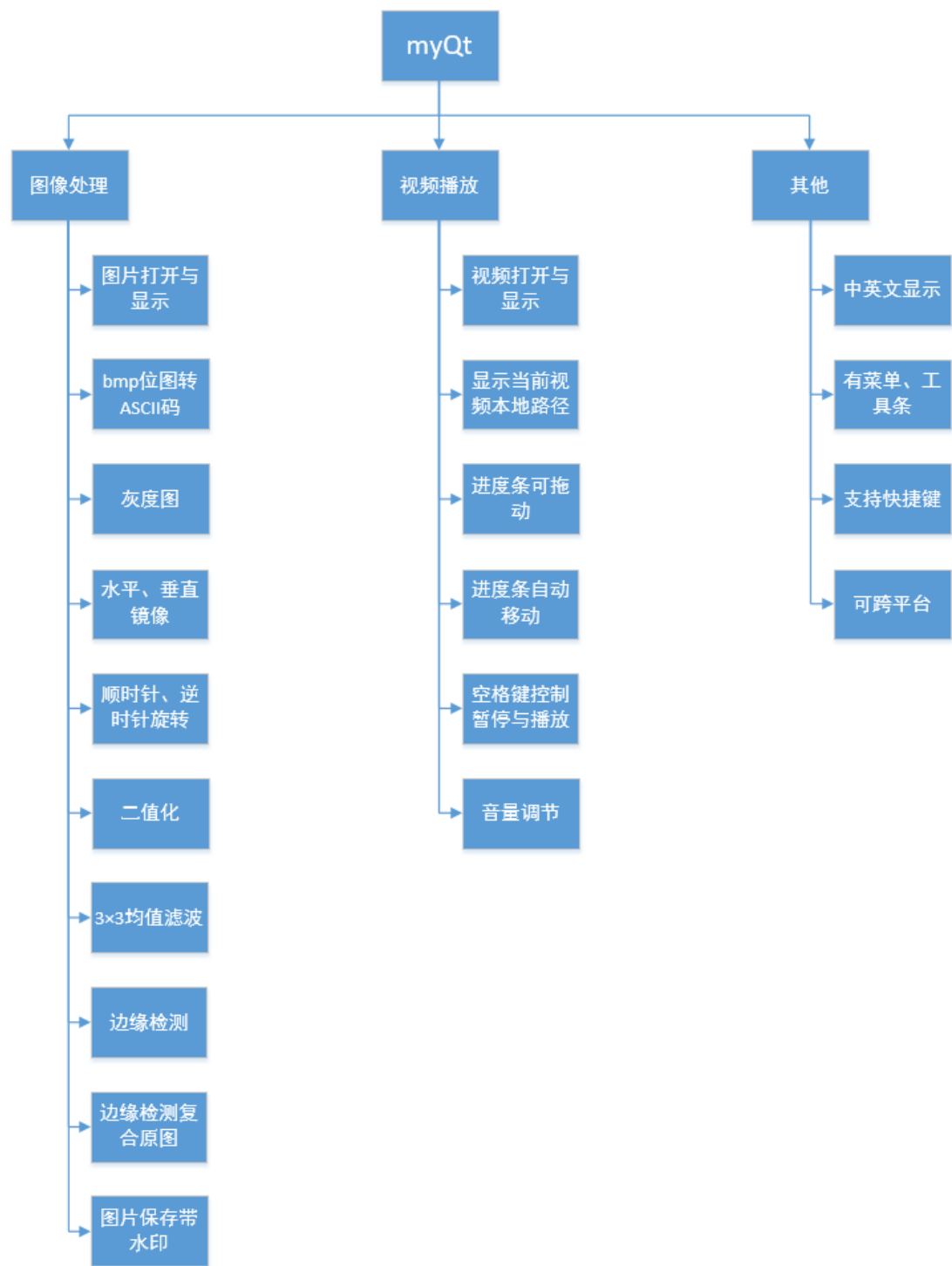


图 11 软件架构图

五、 遇到的困难与获得的主要成果

1) 遇到的困难

1. 刚开始实现 bmp 位图转 ASCII 编码时，没有采用 QImage 类，参照网上方法使用 C++ 去获取每个像素，虽然可行，但十分复杂，容易花费很多时间在 bmp 位图的格式上。使用 QImage 类来取像素、QColor 类来取像素的 RGB 值，方便很多。
2. 在图形边缘检测中，canny 算子相较于 sobel 算子肯定会有更好的效果，但同时相应的算法也复杂得多，参考网上方法后，由于 canny 算子实现的过程过于复杂，没有办法完全理解，遂没有采取。同时，下载网上 canny 算法后运行发现，软件响应的时间过长，先比之下 sobel 算法效率更高了。
3. 在实现图形水平、垂直翻转时，首先的思路是遍历图片，改变遍历时 X 或 Y 的方向，然后将新像素插入到新的图中，但运行时，系统警告图片有像素溢出，偶尔程序会跑崩。最后发现 QImage 有自带方法，QImage 确实不错。
4. 在实现给图片添加手写签名水印时，首先的思路是将签名图片和原图相叠，签名图片是透明底的即可，可完全没有效果，后将签名图片换成白色底时，可以叠加在原图上。然后更换思路，采用了上述第三点中所描述的图片加手写姓名水印方法。
5. 实现定时器，同步控制进度条、进度时间、播放内容时，涉及较多关于进度条的具体状态，如 released、pressed、moved、valuechanged 等，要区分这几个的区别。

2) 收获成果

源于实践，对图形图像处理有了更好的了解，尤其是实现边缘检测算法时，对算法原理产生了浓厚的学习兴趣；在实现视频播放器时，使用定时器、键盘监听、鼠标监听等实现了一些功能，很有成就感。

六、 测试与运行记录

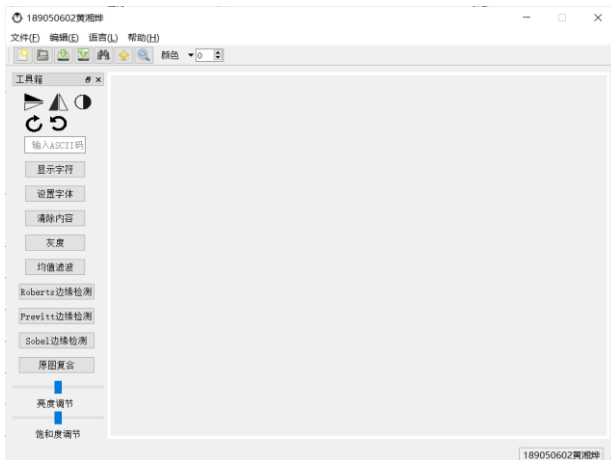


图 12 中文界面

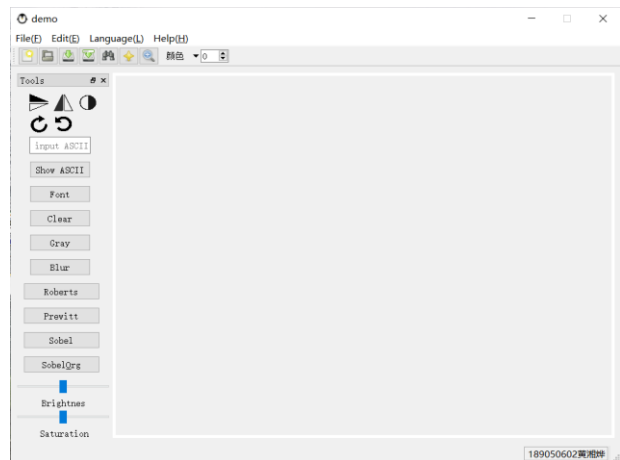


图 13 英文界面

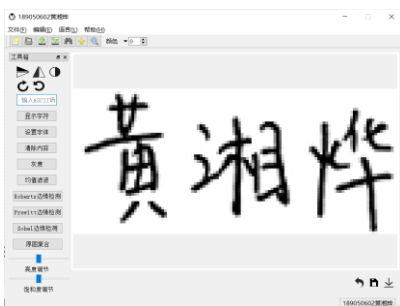


图 14 姓名原图

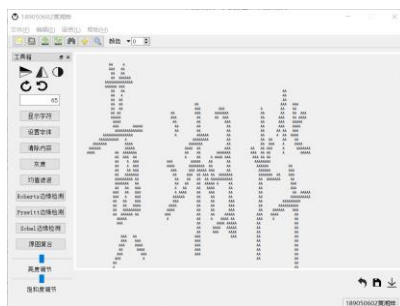


图 15 转 ASCII 码

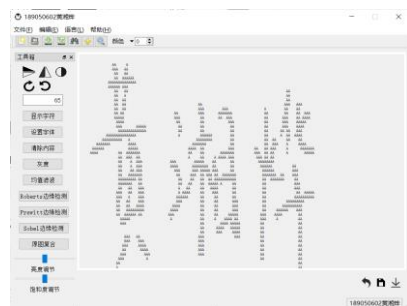


图 16 更换字体

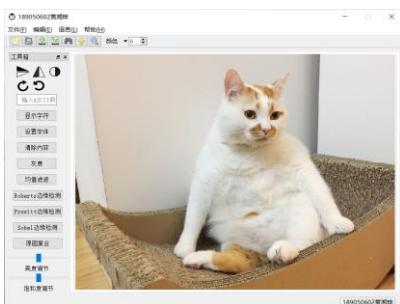


图 17 原图

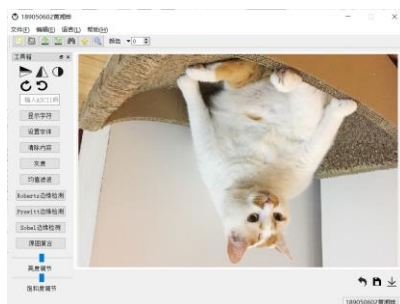


图 18 垂直镜像

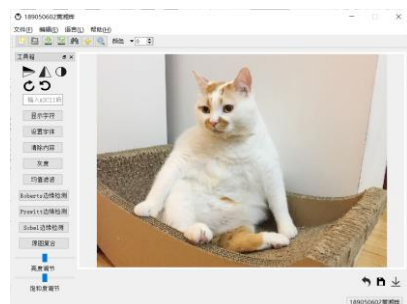


图 19 水平镜像

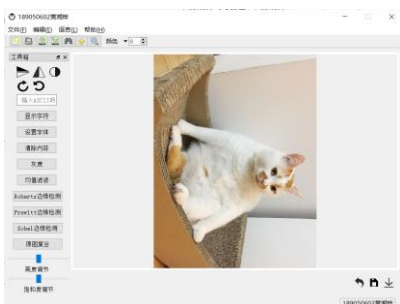


图 20 顺时针旋转

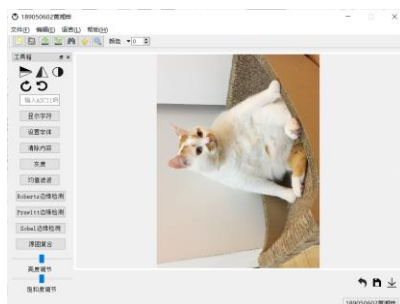


图 21 逆时针旋转

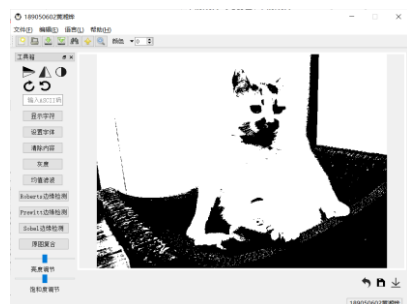
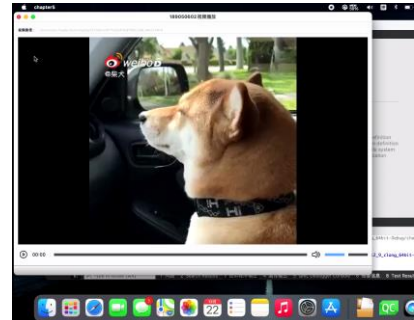
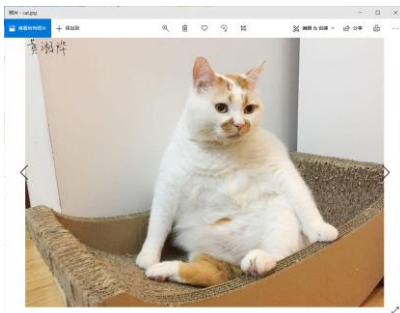
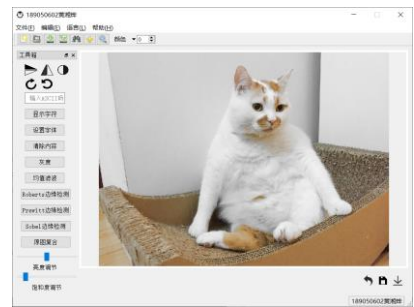
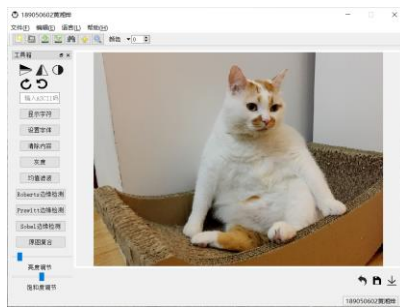
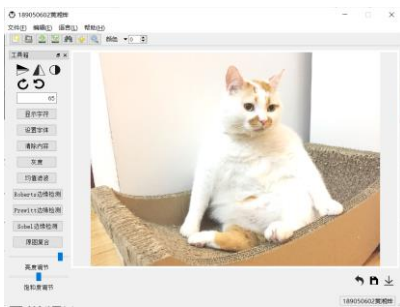
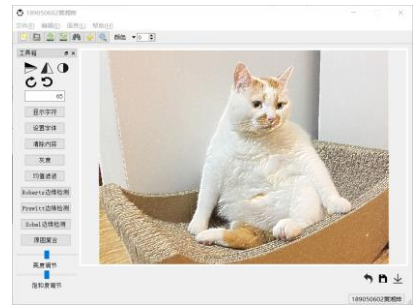
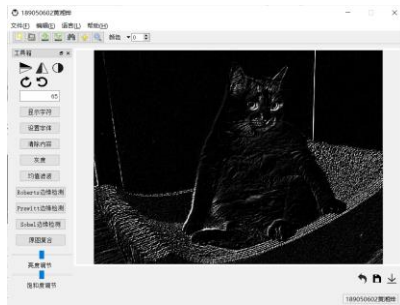
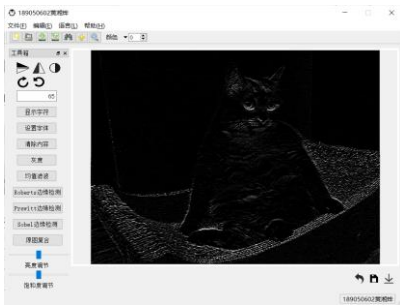
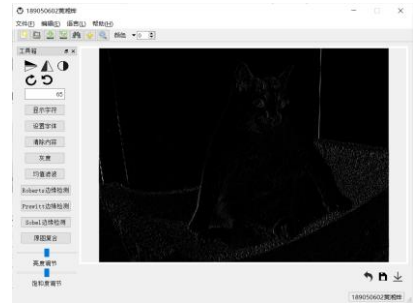
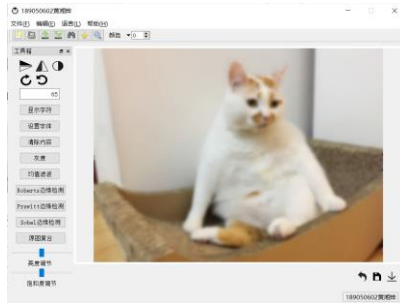
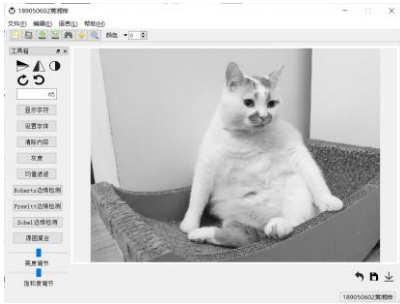


图 22 二值化



七、 结果分析与小结

1) 结果分析

1. 软件界面美观、简洁
2. 对图像进行处理方便、快速
3. 视频能够流畅播放、播放长视频时用户也能流畅拖动进度条
4. 软件拥有友好的中英文界面
5. 能较好地运行在 windows、mac、ubuntu 等平台

2) 小结

通过对图形图像的研究与具体实现，掌握了一些基础的图像处理算法，虽然这只是图形图像领域的冰山一角，但还是收获颇丰。在 Qt 使用 QImage 类能够很好的完成图形图像的处理，尤其是获得指定路径图片、取像素、取 rgba 值、设置像素保存图片至指定路径等等，让我在处理很多图片时，能够只去关注算法本身的实现和实现的原理。

通过完成整个软件的设计，觉得相较于以前，自己对问题处理更加有耐心，能在发现问题时积极地解决问题。同时，通过课堂上的学习和自主学习，学习到了很多知识，很有成就感。