

# 【图像处理系统】

## [总体设计报告]

组长：林悦如

组员：刘涵宇、林宇翔

日期：2018.07.22

# 目录

1 引言 .....	3
1.1 项目背景 .....	3
2 系统配置 .....	3
3 项目架构 .....	4
3.1 MVC 框架.....	4
4 技术细节 .....	6
4.1 图像碎片化 Fragment .....	6
4.2 高斯模糊 Gaussian.....	6
4.3 老旧照片 Old Movie .....	6
4.4 图片素描化 Sketch .....	6
4.5 图片线条提取 Line .....	7
5 协作情况 .....	7
5.1 GitHub 的使用.....	7
5.2 团队工作流程.....	9

# 1 引言

## 1.1 项目背景

### 1.1.1 软件系统名称

图像处理系统

### 1.1.2 任务提出者：

浙江大学科研和工程中的 C++训练课程任课老师-袁昕

### 1.1.3 开发者

由浙江大学 2017-2018 学年短学期课程部分学生组成的项目组

### 1.1.4 相关背景介绍

为全面提高学生创新和实践能力，浙江大学科研和工程中的 C++训练课程分为课堂教学和综合性实验两部分。综合性实验采取分组形式完成，每 3 个学生为 1 组，设有组长，通过让学生开发一个实际的 C++项目，深入了解科研和工程中合作开发 C++项目基本原理、概念与方法，培养学生的应用开发能力与团队合作精神。 锻炼学生综合运用每个环节所学知识解决。

## 2 系统配置

- Windows 10
- Qt 5.10.0, Qt creator
- CMake 3.4.0
- Visual Studio 2017

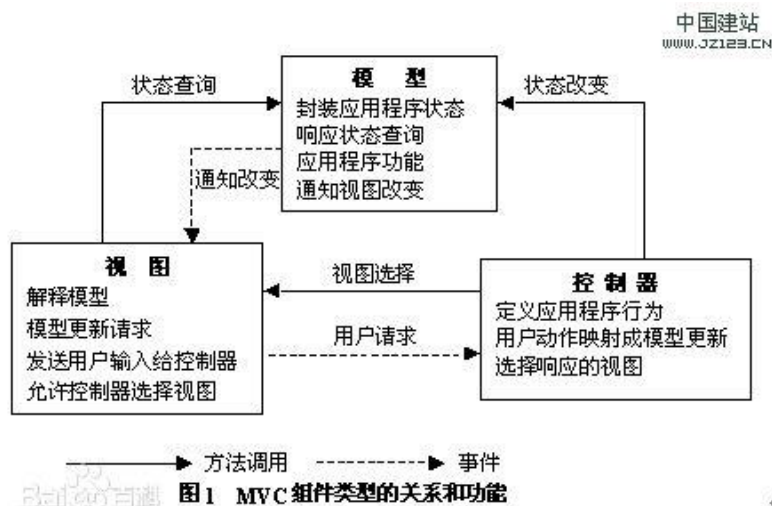
## 3 项目架构

### 3.1 MVC 框架

#### 3.1.1 框架介绍

MVC 全名是 Model View Controller，是模型(model) – 视图(view) – 控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 独特的发展起来被用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

#### 3.1.2 框架图示



#### 3.1.3 框架内容

MVC 指 MVC 模式的某种框架，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件：模型、视图、控制器。它们各

自处理自己的任务。

- 模型

模型表示企业数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。例如它可能用像 EJBs 和 ColdFusion Components 这样的构件对象来处理数据库，被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

- 视图

视图是用户看到并与之交互的界面。对老式的 Web 应用程序来说，视图就是由 HTML 元素组成的界面，在新式的 Web 应用程序中，HTML 依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括 Adobe Flash 和像 XHTML 等一些标识语言和 Web services。

- 控制器

控制器接受用户的输入并调用模型和视图去完成用户的需求，所以当单击 Web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

### 3.1.4 本项目中的框架实现

视图：QT 实现的基本 UI 界面。

控制器：QT 实现的按钮，本身不做任何处理，接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

模型：图像处理算法构件。

## 4 算法实现

### 4.1 图像碎片化 Fragment

#### 4.1.1 算法简介

本算法可以将图片进行碎片化，达到一种较为艺术化的图片处理效果，在工作上可以为设计者带来一些灵感或帮助。

### 4.2 高斯模糊 Gaussian

#### 4.2.1 算法简介

本算法可以将图片进行高斯模糊。是一种较为简易的图像处理算法。

### 4.3 老旧照片 Old Movie

#### 4.3.1 算法简介

算法可以将图片做旧，实现一种老电影式的图片效果

### 4.4 图片素描化 Sketch

#### 4.4.1 算法简介

算法可以将图片素描化，这是一种比较艺术化的处理效果

#### 4.4.2 算法效果

### 4.5 图片线条提取 Line

#### 4.5.1 算法简介

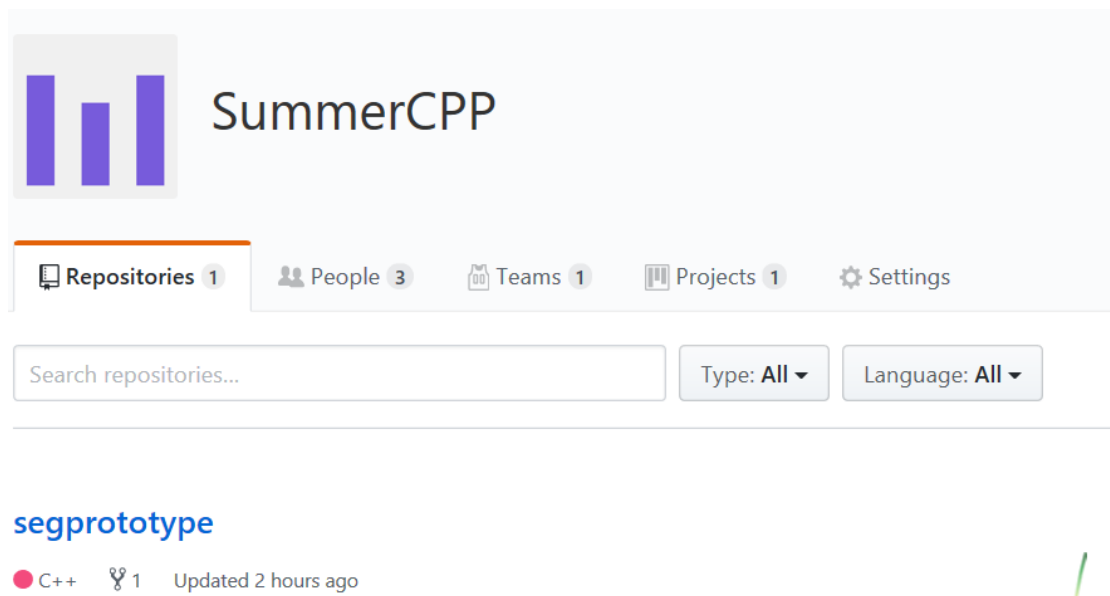
本算法可以提取图片的线条，为进一步设计打下基础

## 5 协作情况

### 5.1 GitHub 的使用

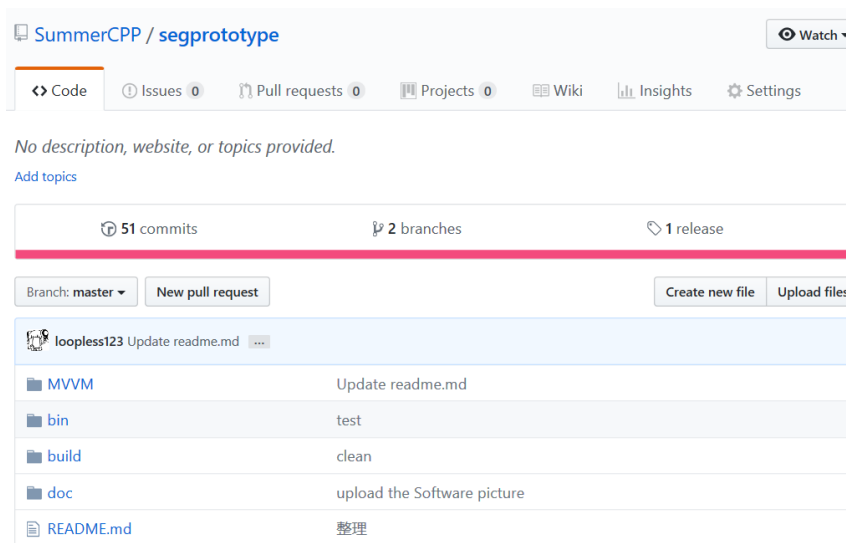
#### 5.1.1 团队的创建

创建了我们的个人团队 SummerCPP，并邀请两位队友加入团队以便于对库进行各种操作。



#### 5.1.2 公有库的创建

创建公有库 segprototype，团队成员在公有库上存放对工程做的更新与修改



### 5.1.3 工程的上传/更新/拉取/修改

本团队使用 git 对库进行一系列的操作

```

MINGW64:/d/Study/ShortTerm/Cpp/git
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SummerCPP/First.git
 16354c8..c531c15 master -> master

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First/doc (master)
$ cd ../

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First (master)
$ ls
appsrc/ build/ doc/ ext/ ide/ notice.md readme.md

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First (master)
$ cd build

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First/build (master)
$ ls
Data.exe* readme.md

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First/build (master)
$ git add "Data.exe"

hasee@Taki-Lin MINGW64 /d/Study/ShortTerm/Cpp/git/First/build (master)
$ git commit -am "upload the ExeFile"
[master a2187ec] upload the ExeFile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 build/Data.exe

```

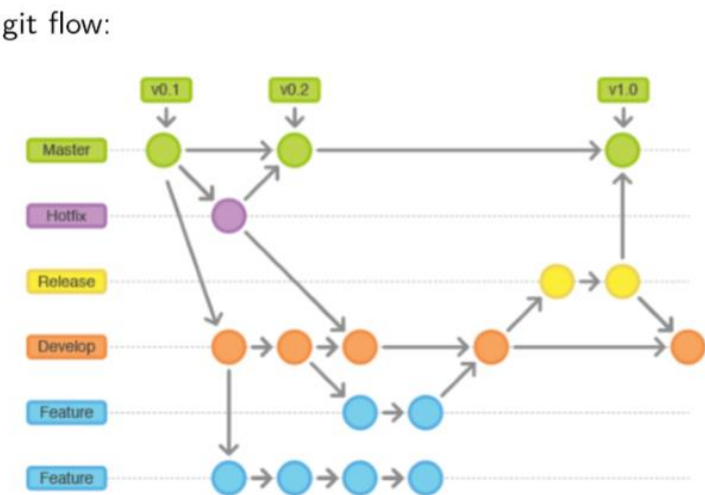
git clone    / git add    / git push    / git pull    / git remote

克隆        / 添加        / 推送        / 拉取        / 添加注释



5.2 团队工作流程

5.2.1 开发流程



5.2.2 角色分工

角色	分工
项目经理	产品设计和工程监督
架构师	软件架构设计
程序员	程序开发
测试员	软件测试

本次项目分多个迭代轮次，每一轮各小组组员分别担任不同的角色对项目进行开发。