

# 华侨大学

## 毕业设计（论文）

题目：激光散斑成像系统 CCD 相机数据采集  
与控制软件设计

院（系） 工学院

专 业 光电信息科学与工程

届 别 2017 届

学 号 1395121001

姓 名 艾贺

指导老师 杜永兆 老师

华侨大学教务处印制

2017 年 5 月

# 激光散斑成像系统 CCD 相机数据采集与控制软件设计

华侨大学工学院

光电信息科学与工程 2017 届 艾贺 1395121001

**摘要** 激光散斑成像(Laser Speckle Imaging, LSI)技术在工业测量和生命科学领域都有广泛的应用。它利用定向散斑或散斑的多次曝光作为信息存储方法,使用调制斑纹图样的光学处理来研究物体的信息。在信息采集这一必不可少的步骤中,由于相机种类繁多,一些相机采集软件并不能完全符合我们的需求,因此自主研发一款相机数据采集软件对我们有很大的帮助。本次 LSI 系统 CCD 数据采集与控制软件设计,采用了面向对象的编程方法,结合型号为 Basler acA640-120uc 的 CCD 相机、Pylon SDK (Software Development Kit) 5.0.5、OpenCV 3.2.0 和 Qt 5.4,实现了可在 Mac 和 Windows 上运行的激光散斑成像系统 CCD 图像数据采集与控制软件的设计。软件可自动检测插入电脑的 USB 接口相机;预览相机采集到的图片信息;同时进行激光散斑空间衬比分析法处理,并实时显示处理结果;还可控制相机的帧速率、曝光时间、采集数量、保存路径以及其他各种参数来进行图片单张采集和连续采集,为今后的数据采集工作提供便利。

**关键词** 激光散斑成像系统 CCD 图像采集 Qt SDK

# **Design of CCD Camera Data Acquisition and Control Software for Laser Speckle Imaging System**

Ai He

1395121001, Major in Optoelectronic Information Science and Engineering, 2013

College of Engineering, Huaqiao University

**Abstract :** Laser speckle imaging (LSI) technology is widely used in industrial measurement and life sciences. It uses the multiple exposure of speckle or directional speckle as the information storage method, and uses the optical processing of the modulated markings to study the information of the object. In the essential steps of the information acquisition, due to the various of acquisition software of different cameras does not fully meet our needs, so develop a camera data acquisition software will provide convenience for us. In this LSI system design, I use the object-oriented programming methods, combined with the model for the Basler acA640-120uc CCD camera, Pylon SDK (Software Development Kit) 5.0.5, OpenCV 3.2.0 and Qt 5.4, to accomplish the purpose to run this software in the Mac and Windows .The software can automatically detect the camera which plugin the computer's USB interface 、 preview the image the camera is capturing, and collect the image. At the previewing time, the software can real-time display the LSI space contrast algorithm result. We can also control the camera's frame rate, exposure time, save path and other parameters to collect the image in single capture or continuous capture. This design will provide the convenience in our future work.

**Key words:** Laser speckle imaging system; CCD image acquisition; Qt; SDK

## 目 录

第 1 章 绪论.....	1
1.1 研究背景.....	1
1.2 论文主要工作和研究内容.....	2
第 2 章 CCD 数据采集与控制系统设计理论基础.....	3
2.1 硬件组成.....	3
2.1.1 Basler 相机.....	3
2.1.2 RICOH 镜头.....	4
2.1.3 Pylon SDK.....	5
2.2 Qt 软件介绍.....	6
2.3 OpenCV 介绍.....	7
第 3 章 LSI 系统 CCD 数据采集与控制软件设计 .....	8
3.1 项目需求.....	8
3.2 项目流程.....	8
3.2.1 相机检测模块.....	9
3.2.2 预览模块.....	9
3.2.3 算法处理模块.....	9
3.2.4 相机设置模块.....	10
3.2.5 采集模块.....	10
3.3 软件的系统配置.....	11
3.4 界面制作.....	12
3.5 系统的关键程序.....	13
3.5.1 枚举插入电脑的相机.....	13
3.5.3 保存图片.....	14
3.5.4 相机参数保存和加载.....	15
3.5.5 相机参数设置.....	15

3.5.6 图像格式转换.....	15
3.6 本章小结.....	15
第 4 章 LSI 系统 CCD 数据采集与控制软件的测试结果.....	16
4.1 LSI 系统 CCD 数据采集与控制软件在 OSX 系统上的测试.....	16
4.2 LSI 系统 CCD 数据采集与控制软件在 Windows 系统上的测试.....	20
4.3 本章小结.....	23
第 5 章 结论与展望.....	24
4.1 全文总结.....	24
4.2 未来展望.....	24
参考文献.....	26
附 录：LSI 系统 CCD 相机数据采集与控制软件程序.....	28

## 第 1 章 绪论

### 1.1 研究背景

20 世纪 60 年代，随着激光器的使用和发展，人们发现了激光散斑<sup>[1]</sup>。20 世纪 70 年代以后，人们逐渐发现散斑的形态能够反映物体的信息，例如表面粗糙度测量、生物组织活性等。激光开始作为一种十分具有价值的研究对象<sup>[2]</sup>。其后，激光散斑成像技术开始信息处理、生物医学等领域广泛应用，如工业涂料干燥过程监测<sup>[3]</sup>、人体微循环血流研究<sup>[4]</sup>、种子活性<sup>[5]</sup>等。特别在医学领域上，激光散斑成像具有非接触无创伤、快速成像等优点，越来越受到研究工作者的青睐。

1981 年，Fercher 和 Briers 开始提出了利用相机拍摄曝光时间后的散斑图像，用来监测血流的速度<sup>[4]</sup>。激光散斑成像系统的原理可以解释为，当激光照射在物体表面发生散射，通过相机记录曝光时间，相机的采集过程由计算机相机控制软件控制，最终采集的散斑图像在计算机中经过算法处理，得到便于研究人员观察的图像。激光散斑成像系统的组成如图 1.1 所示。

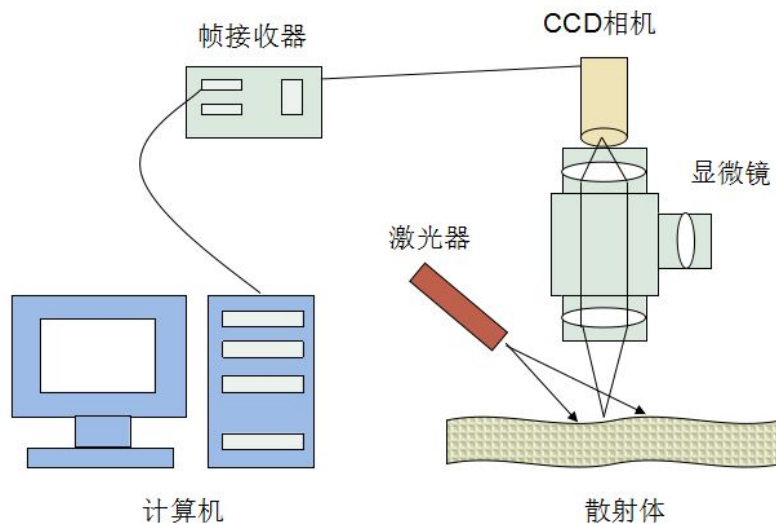


图 1.1 激光散斑成像系统组成

位于图 1.1 左侧计算机端的激光散斑成像 CCD 相机采集与控制系统，是本次设计的主要内容，属于的信息采集与预处理部分。

由图 1.1 可以看出，激光散斑成像系统在一定程度上依赖于计算机的处理，一些市面的有关 CCD 的相机软件价格昂贵，或功能冗余，或功能不足，不符合我们的预期。挑选了合适的 CCD 相机之后，相机自带了一些预留接口与开发文档，可以直接使用。鉴于市场上相机可能存在的问题，此次设计，我们自主研发一款激光散斑成像 CCD 相机数据采集与控制软件。这款软件将很大的方便我们今后的科研工作。

## 1.2 论文主要工作和研究内容

本次工作主要为 LSI 系统 CCD 相机数据采集与控制软件的设计，涉及到的知识点有 Qt 跨平台开发的适配，集成使用相机 SDK，OpenCV 图像算法处理等。本文共分为四章，各章的内容具体安排如下：

第 1 章 绪论，主要介绍了研发此激光散斑成像系统控制软件的背景与目前软件研发现状，问题与背景等。

第 2 章 阐述开发此 CCD 系统控制软件所需要的基本理论与概念。

第 3 章 说明开发此 CCD 系统控制软件的总体过程，从项目需求，到寻找技术解决方案，查阅资料，到最终实现功能，做出产品。并进行测试。

第 4 章 总结与展望，对本文研究的主要内容做出总结，提出本文需要改善的地方，并对今后的可能进行展望。

## 第 2 章 CCD 数据采集与控制系统设计理论基础

### 2.1 硬件组成

CCD 数据采集与控制系统的硬件由 Basler 相机、RICOH 镜头、Pylon SDK 组成。以下为硬件部分的具体描述。

#### 2.1.1 Basler 相机

CCD 工业相机是 LSI 系统中至关重要的部件,它负责将光信号转变成特定的电信号,完成对目标物体的图像采集,对图像信息进行编码处理,如果是 USB 接口的工业相机,则通过 USB 将图像传递给计算机进行信息处理,若是 IP 相机,则可通过 TCP/IP 协议传递图像数据,图像处理软件再根据识别或检测的需要,向相机传递控制信息,修改相机的参数设置。

根据 LSI 系统的要求选择合适的 CCD 工业相机是一个重要步骤,CCD 工业相机的类型直接影响整个系统的采集效果与运行模式,决定了采集图像的质量和分辨率等重要内容。

在选择 CCD 工业相机类型时,需要清楚采集任务,包括静态采集图像还是动态采集、图像采集频率、物体的尺寸,远近,要求等。对于动态采集图像,需要清楚被测物体的移动速度,根据这一参数来确定相机的最小曝光时间,选择线阵相机或者面阵相机:动态采集图像还需要考虑相机的帧率,即每秒采集图像的次数。在普通的 LSI 系统采集任务中,一般选用价格比较低廉的模拟相机和图像采集卡结合,工业级的应用中一般选用可靠性比较高的数字相机<sup>[6]</sup>。

此次我们选择的是 Basler acA640-120uc 型号的 USB 相机,其外观如图 2.1 所示<sup>[7]</sup>。



图 2.1 Basler acA640-120uc 相机



另外，相机规格如表 2.1 所述。

表 2.1 LSI 系统 Basler acA640-120uc 相机规格参数<sup>[7]</sup>

感光芯片		相机数据		设计	
感光芯片供应商	Sony	接口	USB 3.0	设计	Box
感光芯片	ICX618	视频输出格式	Mono 8、YCbCr 422_8、Bayer BG 8、Bayer BG 12、Bayer BG 12p (Bayer BG 12 Packed)、RGB 8、BGR 8	外壳尺寸 (L x W x H)	29.3 mm x 29 mm x 29 mm
快门	Global Shutter	像素位深	12 bits	镜头接口	C-mount/CS-mount
靶面尺寸	1/4"	同步	Software external trigger free-run	外壳温度	0 - 50° C
感光芯片类型	CCD	曝光控制	programmable via the camera API external trigger signal	重量（典型）	80g
感光芯片尺寸	3.7 mm x 2.8 mm	数字输入	1	符合标准	CE、RoHS、GenICam、IP30、USB3 Vision、UL、FCC
水平/垂直分辨率	658 px x 492 px	数字输出	1		
分辨率	VGA	通用 I/O	2		
水平/垂直像素尺寸	5.6 μm x 5.6 μm	电源要求	Via USB 3.0 interface		
帧速率	120 fps	功率（典型）	2.4 W		
黑白/彩色	Color				

2.1.2 RICOH 镜头

光学镜头是本次 LSI 系统中必不可少的部件，直接影响成像的质量，好的工业相机光学镜头可以提高 LSI 系统的工作效率<sup>[8]</sup>。光学镜头从焦距上可分为短焦镜头、中焦镜头，长焦镜头；从视场大小分有广角、标准，远摄镜头；结构上分有固定光圈定焦镜头，手动光圈定焦镜头，自动光圈定焦镜头，手动变焦镜头、自动变焦镜头，自动光圈电动变焦镜头，电动三可变（光圈、焦距、聚焦均可变）镜头等<sup>[8]</sup>，种类繁多，具备专业的知识之人才能选择好的光学镜头。在本次选择 LSI 系统的镜头时，我们选用的是 Ricoh Lens FL-CC1614-2M 镜头，镜头外观如

图 2.2 所示，镜头参数如表 2.2 所述。



图 2.2 LSI 系统 Ricoh Lens FL-CC1614-2M 镜头<sup>[9]</sup>

表 2.2 LSI 系统 FL-CC1614A-2M 镜头参数<sup>[9]</sup>

参数名称	参数值	参数名称	参数值
型号	FL-CC1614A-2M	水平视角	1/3 型 17.10°
画面尺寸	2/3 型		1/2 型 22.6°
焦距	16 mm		1/1.8 型 25.3°
最大口径比	1:1.4		2/3 型 30.7°
光圈范围	1.4~16	最小物距	0.1 m
安装方式	C	背景焦点	10.7 mm
外形尺寸	29.5×32.2 mm	滤镜尺寸	27 P=0.5mm
重量	54 g	备注	带有安全锁螺丝附带锁杆

### 2.1.3 Pylon SDK

软件开发工具包（Software Development Kit，SDK），一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合<sup>[10]</sup>。SDK 一般可以理解为多个 API 的集合，在 SDK 中的 Basler pylon API 是用于 IEEE 1394，千兆以太网（GigE），USB 和 Camera Link 摄像头的应用程序编程接口（API），它提供了通用的统一编程接口，可以隐藏接口技术之间的大部分差异，允许从一个摄像机接口轻松迁移到另一个摄像机接口，而无需

主要代码更改。Basler tylon 5 配备了下一代 GenICam API，在 GenICam 性能和内存占用方面进行了大量改进。

通过 Basler 的开发文档可以看到，Pylon API 的应用 API 结构如图 2.3 所示。根据此 API，我们可以根据自己所需进行相机的控制。

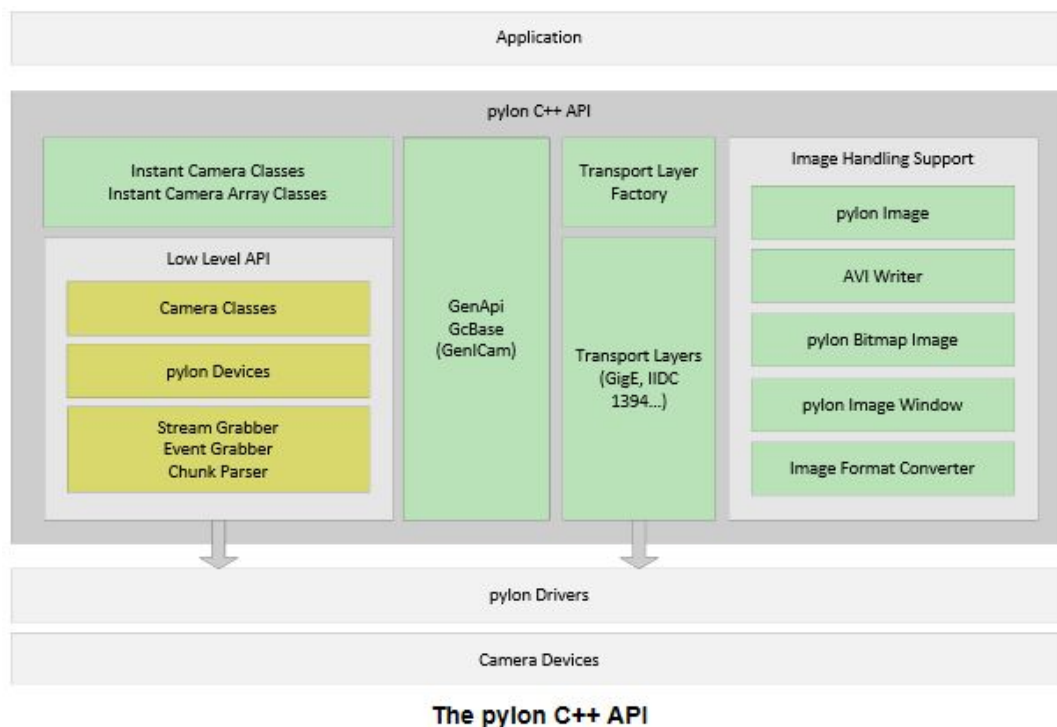


图 2.3 Pylon API 结构<sup>[10]</sup>

## 2.2 Qt 软件介绍

Qt 是 1991 年奇趣科技开发的一个跨平台的 C++ 图形用户界面应用程序框架<sup>[12]</sup>。Qt 的名字有一个有趣的故事，Qt 的两位创始人 Haavard Nord 和 Eirik Chambe-Eng 合作开发时，他们发现字母 Q 看起来非常漂亮；而字母 t 的灵感来源于另外一个工具 Xt(X toolkit)而取名<sup>[13]</sup>。Qt 软件有自己的软件开发包(SDK)，该 SDK 的组成主要是 Qt Creator 和 Qt Opensource，Qt SDK 能够完成一整套程序的开发。

Qt Creator 是一个跨平台的、完整的 Qt 集成开发环境，其中包含了 C++ 的代码编辑器，项目生成和配置工具，集成的上下文帮助系统，版本控制系统等。Qt Opensource 则是提供 Qt 软件开发中所必须的所有核心 API 接口。Qt 软件在

图形化界面开发存在着巨大的优势使得许多人采用其开发图形软件，这也使得 Qt 的用户文档丰富，便于初学者快速上手<sup>[12]</sup>

相比较传统的一些图形界面开发软件，Qt 具有以下的开发优势：

#### 1. 良好的跨平台特性

Qt 支持当今所有主流的操作系统：Windows、Unix、Linux、OSX、嵌入式操作平台（包括嵌入式 Linux、Windows CE）等。

#### 2. 面向对象开发

Qt 采用面向对象的方法进行开发，这使得用户复用很多已经在 Qt 中写好的组件进行开发，同时，Qt 特有的信号与槽模式特点又使得对象之间的相互响应操作的实现变得更加简单<sup>[14]</sup>。

#### 3. 应用程序接口十分丰富

Qt 的 API 十分丰富，图形界面，多媒体，网络，数据库，测试等各种组件都已集成好。

#### 4. 应用和开发文档丰富

Qt 不仅封装了很多好用的模块，在其官网上还有详细的案例指导我们如何使用其中的组件，有关 Qt 的书也是多的数不胜数。在开发的过程中，即使没有联网，内置的 Qt Creator 也有详细的文档供我们参考<sup>[15]</sup>。

## 2.3 OpenCV 介绍

Open CV 的全称是 Open Source Computer Vision Library，直译就是“开源计算机视觉库”。OpenCV 于 1999 年由 Intel 建立，如今由 Willow Garage 提供支持。它是一个基于开源行的卡平台计算机视觉库，可以运行在 Linux, Windows, MacOS, Android, IOS, FreeBSD, OpenBSD 等操作系统上。OpenCV 由一些 C 函数和 C++ 类构成，开发者可以自由的调用函数库中的相关处理函数来实现一些常用的图像处理及计算机视觉算法<sup>[16]</sup>。它具有良好的可移植性，Open CV 是在 IPL 基础上发展起来的，在 IPL 支持下，Open CV 主要用于对图像进行一些高级处理，比如说特征检测与跟踪运动分析，目标分割与识别以及 3D 重建等。本系统主要用到 OpenCV 的 HighGUI 库，High GUI 是图形界面接口库，它包括了载入/保存图像、视频 I/O 通用系统函数等。

## 第 3 章 LSI 系统 CCD 数据采集与控制软件设计

本课题开发的激光散斑成像系统 CCD 相机数据采集与控制软件主要在内部使用，为今后的试验提供便利。在后续的使用过程中，系统功能将会有所调整和增加，此次设计，要求系统开发具备设计清晰、模块性强、易于扩展等特点。

### 3.1 项目需求

本次设计主要基于 Mac pro, Qt 软件平台上实现编程开发。Mac 是基于 Unix 的一种系统，类 Linux，但是比 Linux 具有更加漂亮的图形界面，更便捷的操作。Qt 和 OpenCV 在基础部分已经有所提及。本次设计相机型号：底座型号 Basler acA640-120uc。

在进行 CCD 相机软件的程序时，首先要明确该系统的需求，具体包括：

1. 能对插入的 CCD 相机 USB 接口进行识别；
2. 预览区域的图像实时显示；
3. 实现单张采集和连续采集；
4. 实现 CCD 相机的参数进行设定，如曝光时间、帧速率、采集数量、图像分辨率等关键参数；
5. 实现对 CCD 相机采集到的图像数据进行保存，可设定文件名、文件路径、文件格式等；
6. 扩展：对 CCD 相机实时采集到的图像数据进行算法处理，并对算法处理结果进行实时显示。

### 3.2 项目流程

LSI 系统 CCD 相机数据采集与控制软件的设计主要可分为 5 个模块，分别为：相机检测模块、预览模块、算法处理模块、相机设置模块、采集模块。总流程图如图 3.1 所示。接下来将对每个模块进行详细描述。

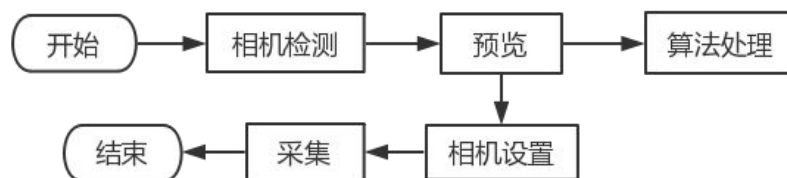


图 3.1 LSI 系统 CCD 软件设计总流程图

### 3.2.1 相机检测模块

刚打开 LSI 系统 CCD 相机数据采集与控制软件时，软件会自动检测是否有相机插入 USB 接口，如果没有插入相机，系统会弹出消息对话框，提示请插入相机。根据提示，重新插入相机后，点击工具栏的刷新按钮，可进入预览模式。相机检测模块流程如图 3.2 所示。

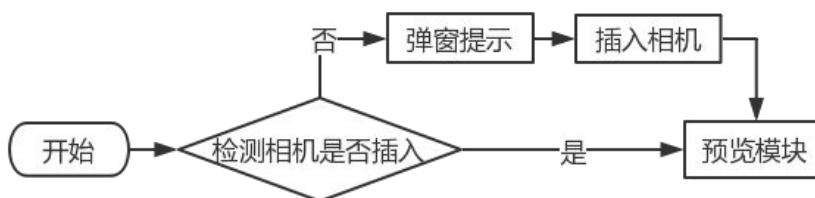


图 3.2 相机检测模块流程图

### 3.2.2 预览模块

相机进入预览模式，实时显示当前相机曝光的图像。在预览模式下，可以在相机设置模块进行相机参数的配置，然后再次预览，预览效果满意可以直接采集，采集完之后再次进入预览模式。预览流程如图 3.3 所示。



图 3.3 预览流程图

### 3.2.3 算法处理模块

根据激光散斑图像空间衬比分析的算法<sup>[17]</sup>，画出的程序框图如图 3.4 所示。

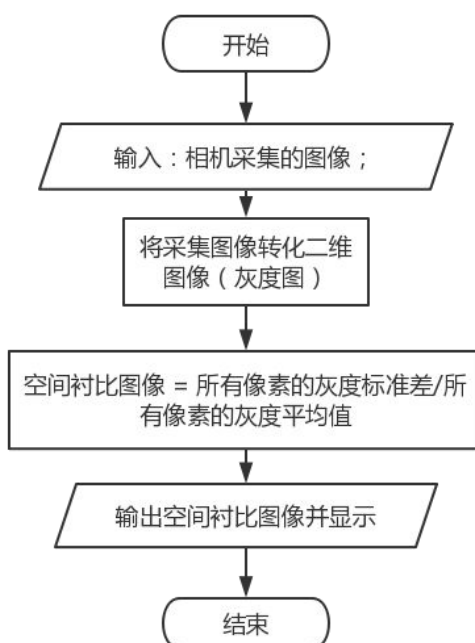


图 3.4 激光散斑空间衬比分析程序流程图

### 3.2.4 相机设置模块

该模块集成的相机设置参数包括：文件名前缀、文件类型、保存路径、分辨率、采集数量、帧频、曝光时间、图像大小（图片的宽和高）。在采集图像前，软件使用人员根据需求对相机参数的进行设置。

### 3.2.5 采集模块

在预览模式下，可以点击工具栏上的单张采集和多张采集。点击单张采集，图像直接保存在设置路径下的“single”文件夹里，并且按照设置的文件名排序命名；点击连续采集，图像保存在设置路径下的“multi”文件夹里，所有图像按“设置的文件名+序号”命名。

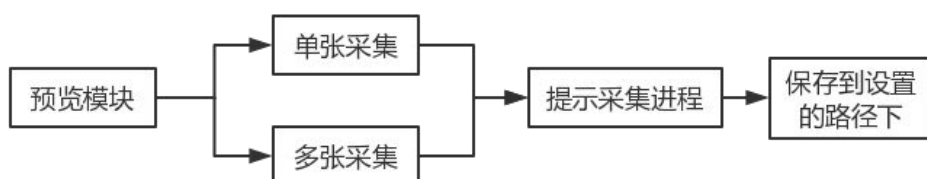


图 3.5 采集模块流程图

### 3.3 软件的系统配置

LSI 系统 CCD 相机软件同时在 Windows 和 Mac 上运行需要做一些配置,才能运行。

1. LSI 系统 CCD 相机数据采集与控制软件在 Windows 上的配置。需要从 OpenCV 官网下载最新的 Opencv 二进制安装包。另外,在 Windows 系统需将 Qt 转化为 VS 项目,并配置路径,程序如下:

```
qmake -spec win32-msvc2013 -tp vc
```

在 VS 中,打开项目属性->链接->通用->附加库目录->添加 D:\pylon5\Development\lib\x64 到库路径的最前面,如图 3.6 所示。

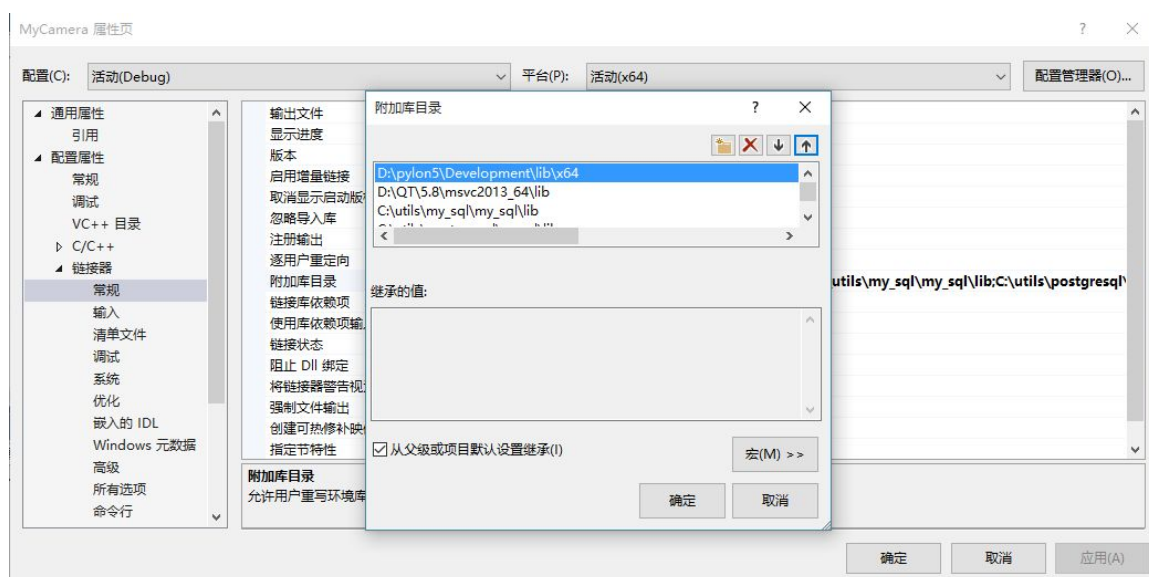


图 3.6 LSI 系统 CCD 相机软件在 Windows 上的 VS 配置

2. LSI 系统 CCD 相机数据采集与控制软件在 Mac 上的配置。使用 Macport 直接进行安装 OpenCV。在 Mac 上使用 Qt 进行开发时,还需要设置在环境变量中配置系统库路径,如图 3.7 所示。

使用 **系统环境变量** 和

设置 **PATH** 到 `/Users/aihe/Qt5.4.0/5.4/clang_64/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin`

设置 **PKG\_CONFIG\_PATH** 到 `/opt/local/lib/pkgconfig/`

图 3.7 Mac 环境变量配置



### 3.4 界面制作

在程序在处理逻辑之前,一般先进行界面的设计与开发,借用 Qt creator 的 Qt designer 我们可以首先通过拖拽原件的方式进行布局。布局的控件必须拖拽到主面板内才可生效。在 LSI 系统 CCD 相机数据采集与控制软件中,主要的界面包括:主界面、相机设置界面。

1.主界面制作。主界面主要考虑到用户只需要做一些点击操作即可,因此主要的功能放在了工具栏。主界面的设计如图 3.8 所示。

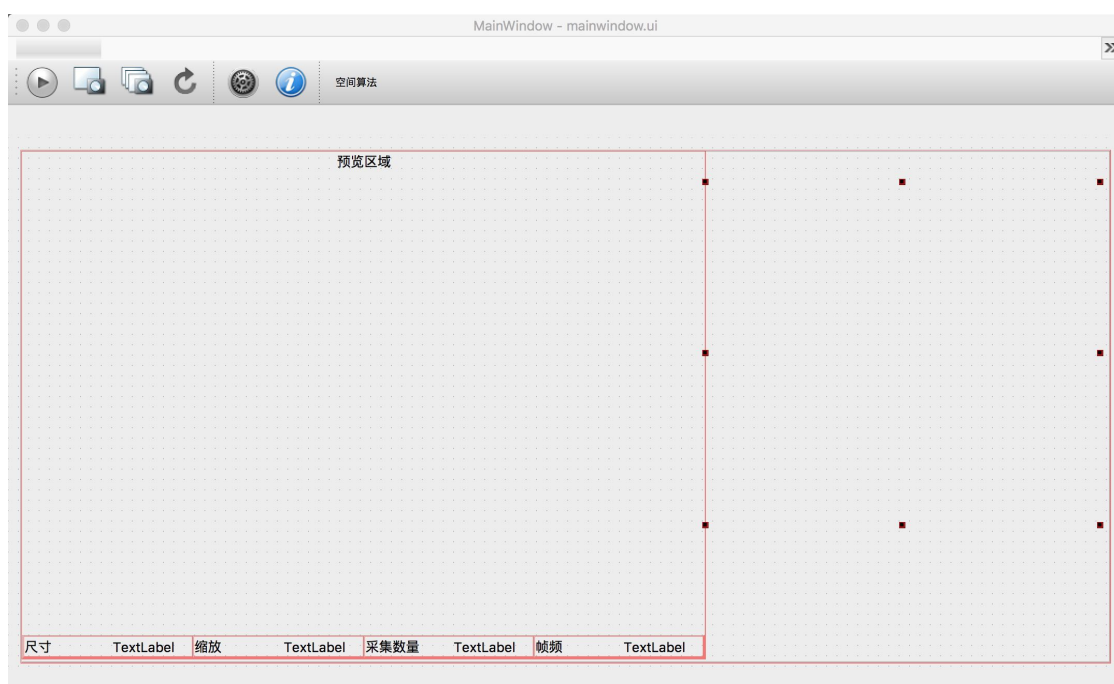


图 3.8 软件的主界面

2. 相机设置界面制作。相机有很多参数可以控制,考虑到我们常用的参数只有几个,设置界面只取了我们常用的参数控制。具体的控制参数如第 3 章 3.2.4 节所述。界面设计如图 3.9 所示。



图 3.9 相机参数设置界面

制作界面后，可使用命令 `uic` 编译设计好的界面文件，程序如下：

```
uic mainwindow.ui > ui_mainwindow.h    //
```

大于符号在 \*nix 系统下表示管道，能把字节流输入到新的 `ui_mainwindow.h` 文件里。

接下来可以在程序中引入 `ui_mainwindow.h` 文件，即可使用界面文件。

### 3.5 系统的关键程序

本次设置使用了 Mac pro, OSX 版本 10.12.4, OpenCV 3.2.0, Qt5.4 和 PylonSDK 5.0.5 的版本。

#### 3.5.1 枚举插入电脑的相机

当系统启动时，我们需要知道是否有相机插入电脑上，因此我们采用如下方法可枚举插入电脑的 USB 相机设备。

`CTIFactory::EnumerateDevices()` 能够列出所有可用的设备信息，并存放在类型为 `DeviceInfoList_t` 的列表里。样例代码如下：

```
CTIFactory& TIFactory = CTIFactory::GetInstance();

DeviceInfoList_t lstDevices;
```

```

TIFactory.EnumerateDevices( lstDevices );

if ( ! lstDevices.empty() ) {

    DeviceInfoList_t::const_iterator it;

    for ( it = lstDevices.begin(); it != lstDevices.end(); ++it )

        cout << it->GetFullName();

}

```

### 3.5.2 设置相机的采集方式

相机开始采集的代码如下，在调用采集函数的同时，设置采集策略，程序如下：

```

CInstantCamera.StartGrabbing(

    EGrabStrategy strategy=GrabStrategy_OneByOne,

    EGrabLoop grabLoopType=GrabLoop_ProvidedByUser

)

```

采集方式有以下四种：

1. GrabStrategy\_OneByOne :图片按照到达顺序进行处理，这是默认的采集策略；
2. GrabStrategy\_LatestImageOnly:只有最新采集的图片才会放在输出队列，其它的图片将被跳过，如果检索图片的时候没有找到图片，那么进程将会等待即将到来的图片；
3. GrabStrategy\_LatestImages:这个方式用来采集图片同时保存最新的图片，主要根据输出队列大小来控制，CInstantCamera::OutputQueueSize 参数用来控制多少张图片存在队列中，当输出队列设置为 1，它等价于 GrabStrategy\_LatestImageOnly 方式，当设置为 CInstantCamera::MaxNumBuffer，它等于 GrabStrategy\_OneByOne；
4. GrabStrategy\_UpcomingImage: 这种方式不适用与 USB 相机。

### 3.5.3 保存图片

1. 保存方式。软件设计使用 Pylon SDK 内置的保存方式，程序如下：

```

CImagePersistence::Save( ImageFileFormat_Png, "GrabbedImage.png", ptrGrab
Result);

```

2. 图片格式。Windows 上支持的图片格式有:Bmp、Tiff、Jpeg、Png。Mac 和\*nix 上支持的两种:Tiff、Png。

### 3.5.4 相机参数保存和加载

Pylon 相机特性要保存到的文件名字，相机特性众多，有时，我们想要保存相机特性或者从文件加载相机特性，可使用如下方式：

```
const char Filename[] = "NodeMap.pfs";

CFeaturePersistence::Save( Filename, &camera.GetNodeMap() );

CFeaturePersistence::Load( Filename, &camera.GetNodeMap(), true );
```

### 3.5.5 相机参数设置

参数设置可浏览 Pylon SDK 包中的参数。例如设置相机曝光时间，程序如下：

```
INodeMap& nodemap = camera.GetNodeMap();

CFloatPtr ExposureTime( nodemap.GetNode("ExposureTime") );

ExposureTime->FromString("8000");
```

### 3.5.6 图像格式转换

因为 Basler 摄像头采集到的图像数据类型是 CPylonImage 型，而 Opencv 处理的图片数据类型是 Mat 型矩阵，因此需要将其转换成 Mat 型，程序如下：

```
CImageFormatConverter fc;                //定义 Pylon 的图片转换对象

Mat openCvImage

fc.OutputPixelFormat = PixelType_BGR8packed;    //转换后的输出格式

if (ptrGrabResult->GrabSucceeded())

{

    fc.Convert(image, ptrGrabResult);        //进行转换

    openCvImage=cv::Mat(ptrGrabResult->GetHeight(),

        ptrGrabResult->GetWidth(), CV_8UC3, (uint8_t *) image.GetBuffer());

}
```

## 3.6 本章小结

本章从 5 个方面描述了 LSI 系统 CCD 相机数据采集与控制软件的设计过程，包括软件设计的需求、软件设计的思路 and 流程、软件系统配置、界面制作、关键程序。解决这些关键问题后，LSI 系统 CCD 相机数据采集与控制软件的开发基本完成。

## 第 4 章 LSI 系统 CCD 数据采集与控制软件的测试结果

在完成 LSI 系统 CCD 相机数据采集与控制软件设计后，本章将描述在实验室进行散斑图像采集并分别在 OSX 系统和 Windows 系统上测试软件的功能，然后，得出测试结果并分析。

### 4.1 LSI 系统 CCD 数据采集与控制软件在 OSX 系统上的测试

在 OSX 系统上的功能测试过程如下：

1. 我们可首先在 MAC 电脑下不插入设备，然后进入页面，页面如图 4.1 所示。

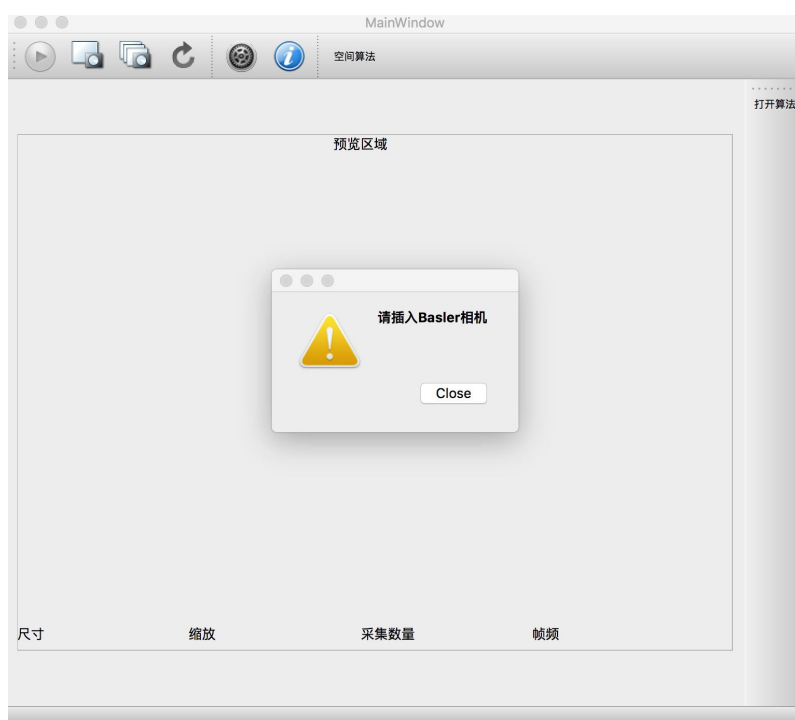
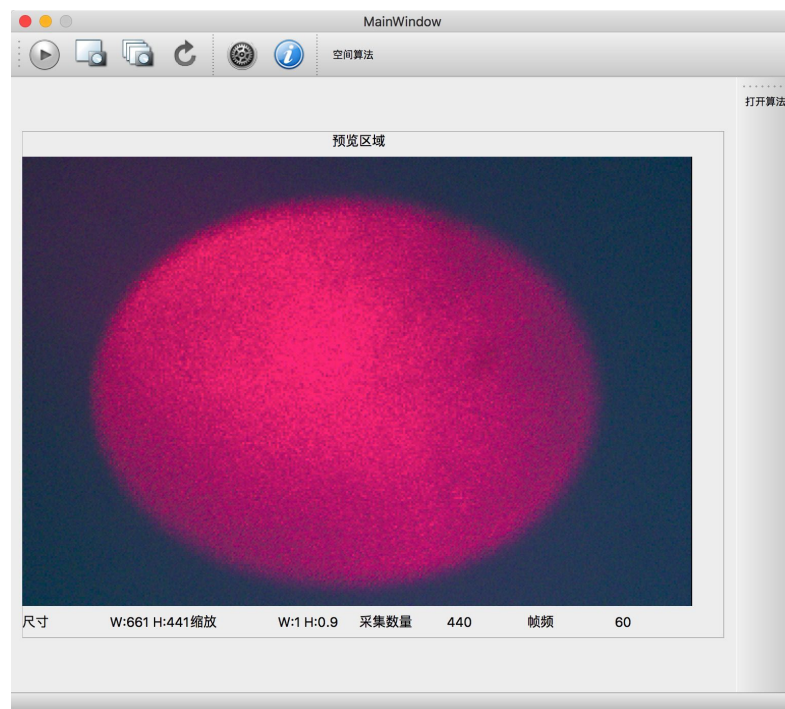


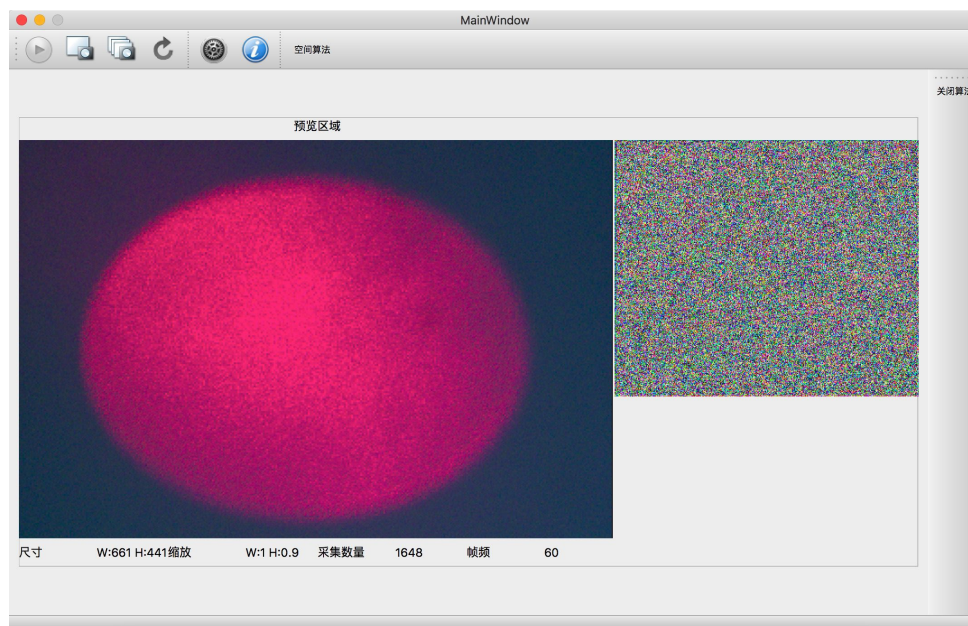
图 4.1 插入相机提示窗

2. 当我们插入设备，并点击刷新按钮，相机进入预览模式，并在预览区域的底部可以看到当前图像的尺寸、相机设置的采集数量和帧频等参数。相机预览如图 4.2 所示。



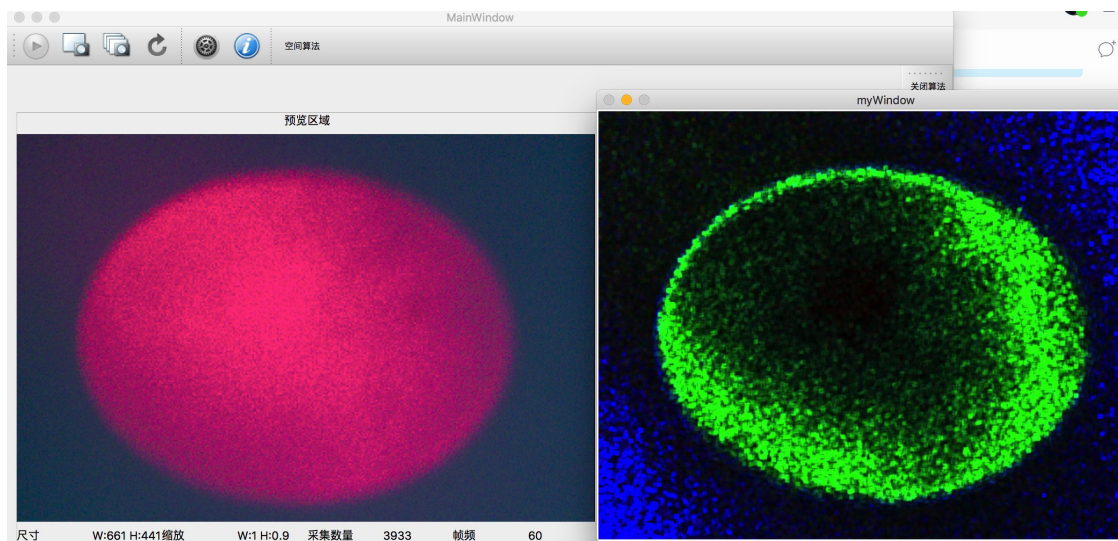
4.2 相机预览图

3. 预览的同时进行散斑像的空间衬比分析算法处理。点击侧边栏的打开算法，软件将会展开算法处理预览区域，如图 4.3 所示。图 4.3 (a) 采用的是 Qt 中的 QImage 显示，图 4.3 (b) OpenCV 内置的图像显示方式显示。两图左侧都为相机曝光的图像预览区域，右侧都为空间衬比算法实时处理和显示的区域。



(a) 采用 Qt 中的 QImage 显示





(b) 采用 OpenCV 内置的图像显示方式显示

图 4.3 相机预览同时算法处理：(a)采用 Qt 中的 QImage 显示；(b)采用 OpenCV 内置的图像显示方式显示

4. 配置相机的参数。点击工具栏上的设置图标，即可打开相机设置窗口，窗口如图 4.4 所示。

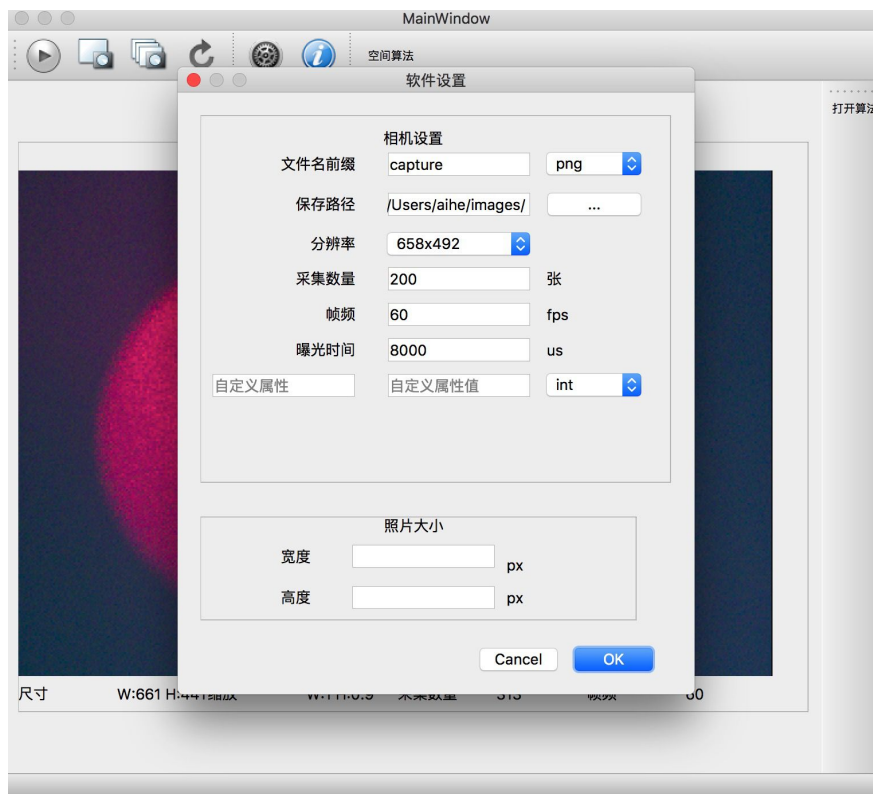
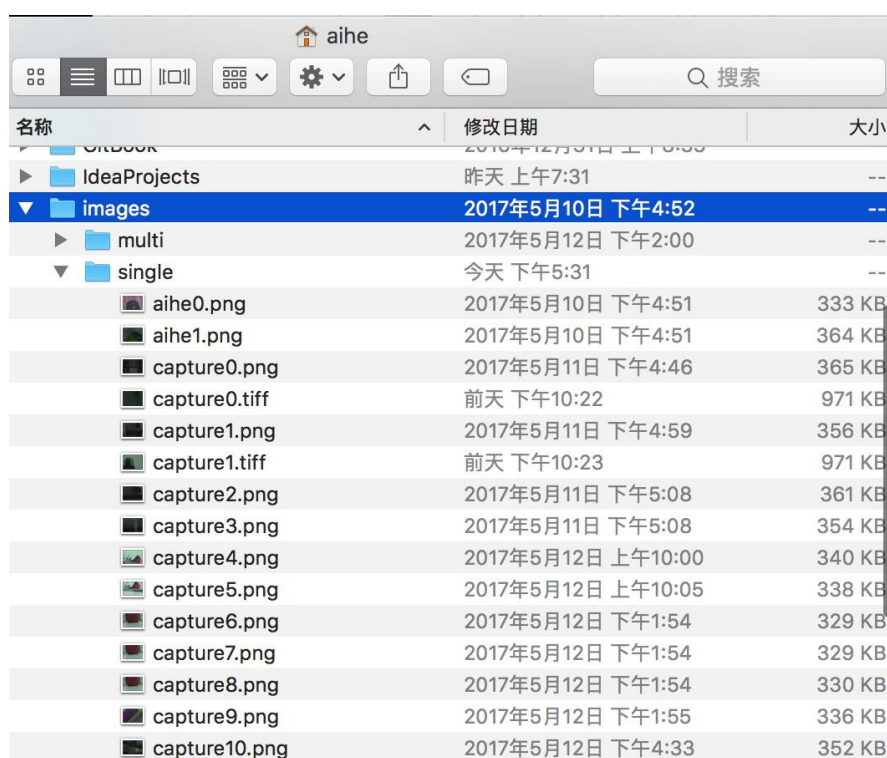


图 4.4 配置相机参数

相机参数配置窗口可以实现的功能包括：设置保存图片得到格式、文件名、保存路径、帧速率、曝光时间、以及一些自定义的属性。自定义的属性需要参考我们列出的文档。

5. 单张采集和多张采集。在预览的时候，我们可以点击单张采集或者连续采集，如果是单张采集，相机保存采集到的图片在设置路径文件夹下的“single”文件夹；如果是连续拍摄，则会保存在“multi”文件夹下，可以看到单张采集的图片。保存的图片根据文件名和先后顺序逐个排列。数据采集目录如图 4.5 所示。



名称	修改日期	大小
IDEBOOK	2016年12月31日 上午10:00	--
IdeaProjects	昨天 上午7:31	--
images	2017年5月10日 下午4:52	--
multi	2017年5月12日 下午2:00	--
single	今天 下午5:31	--
aihe0.png	2017年5月10日 下午4:51	333 KB
aihe1.png	2017年5月10日 下午4:51	364 KB
capture0.png	2017年5月11日 下午4:46	365 KB
capture0.tiff	前天 下午10:22	971 KB
capture1.png	2017年5月11日 下午4:59	356 KB
capture1.tiff	前天 下午10:23	971 KB
capture2.png	2017年5月11日 下午5:08	361 KB
capture3.png	2017年5月11日 下午5:08	354 KB
capture4.png	2017年5月12日 上午10:00	340 KB
capture5.png	2017年5月12日 上午10:05	338 KB
capture6.png	2017年5月12日 下午1:54	329 KB
capture7.png	2017年5月12日 下午1:54	329 KB
capture8.png	2017年5月12日 下午1:54	330 KB
capture9.png	2017年5月12日 下午1:55	336 KB
capture10.png	2017年5月12日 下午4:33	352 KB

图 4.5 数据采集目录

6. 弹窗提示采集进程。当点击连续采集时，系统弹出连续采集的窗口，提示我们正在采集，如图 4.6 所示。



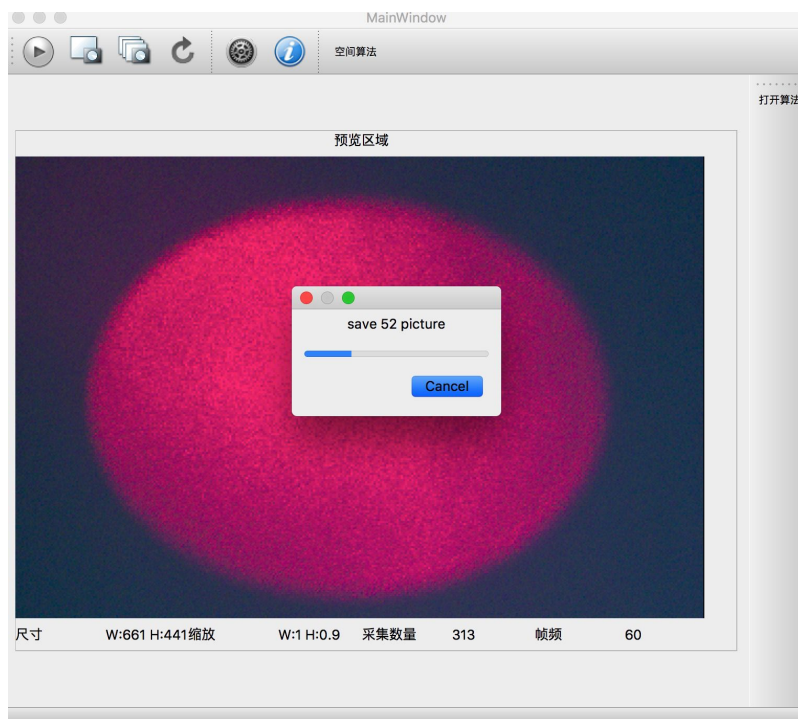
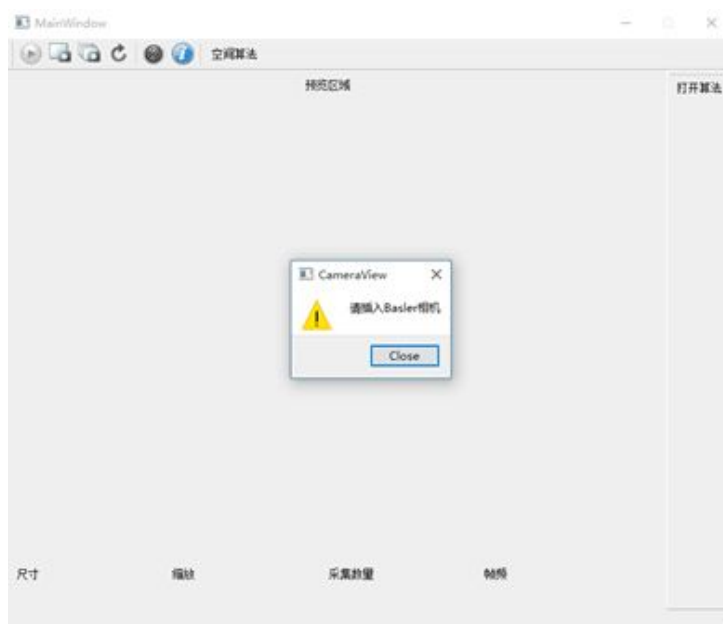


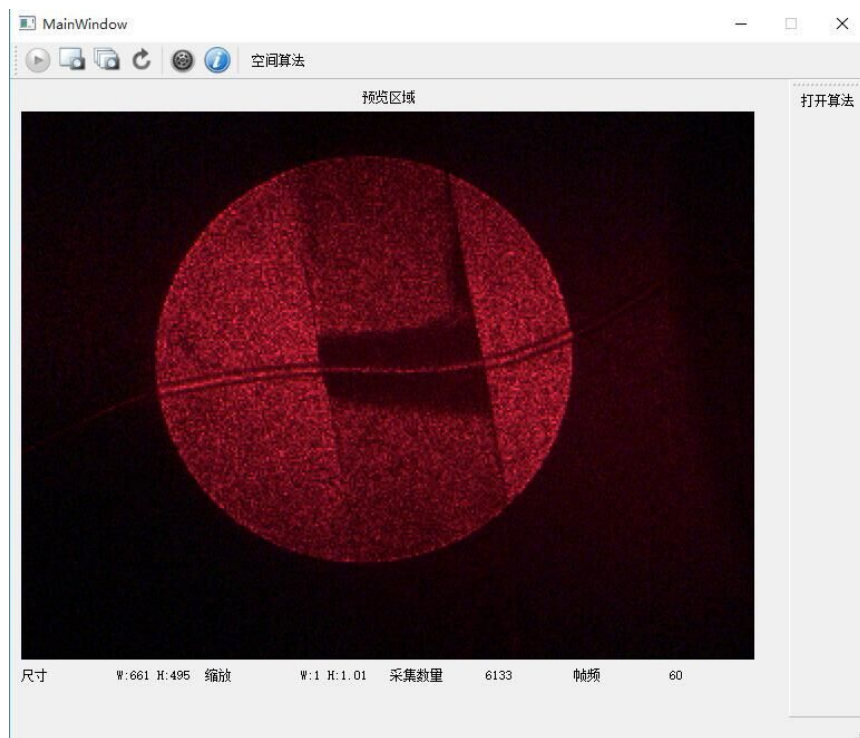
图 4.6 弹窗提示采集进程

## 4.2 LSI 系统 CCD 数据采集与控制软件在 Windows 系统上的测试

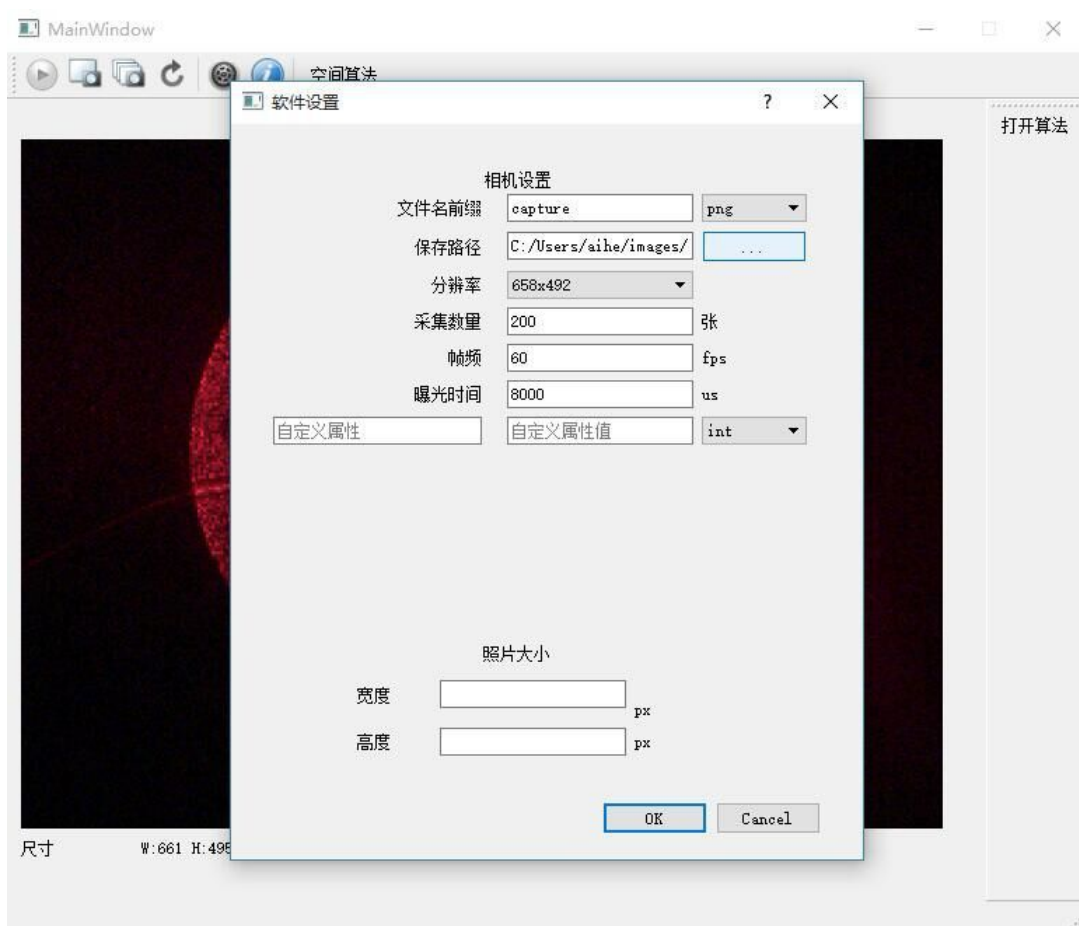
LSI 系统 CCD 数据采集与控制软件在 Windows 系统与 OSX 系统上具有相同的功能。在 Windows 系统上的测试结果如图 4.7 所示。



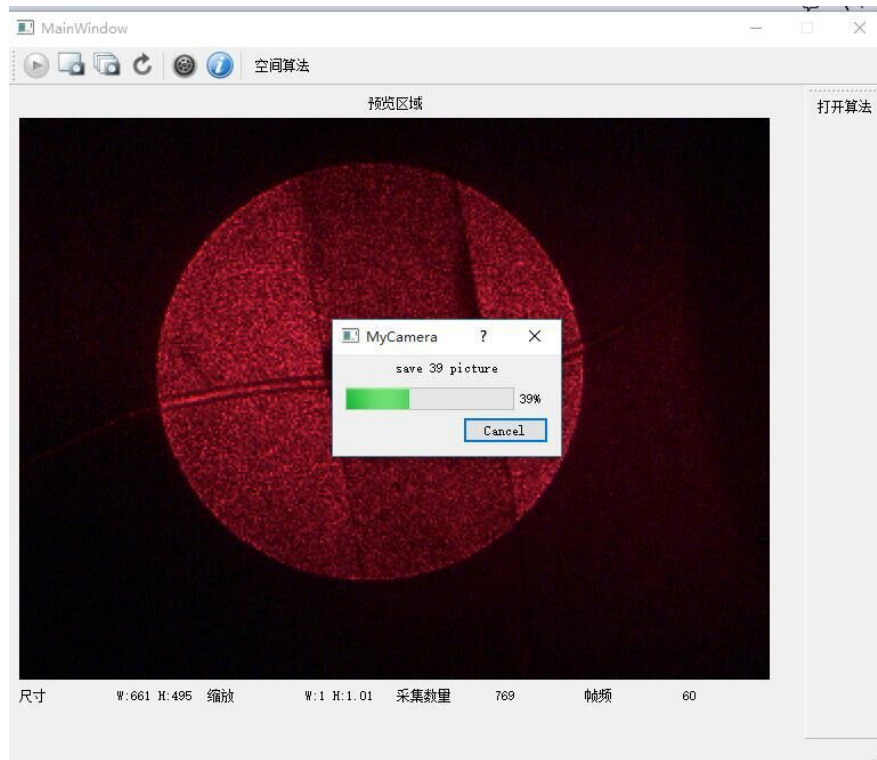
(a) 插入相机提示窗



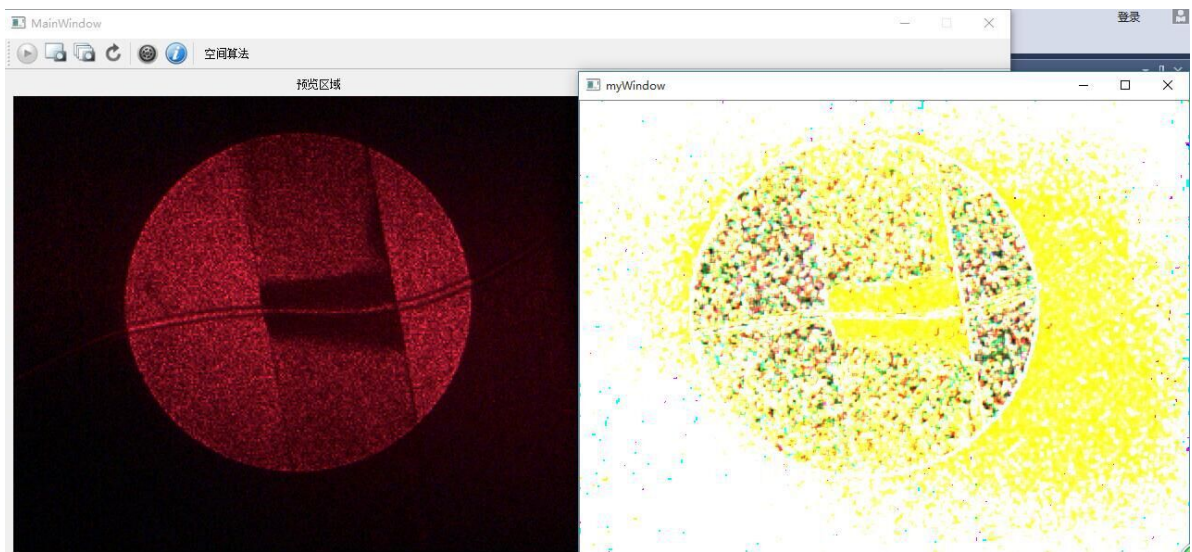
(b) 相机预览



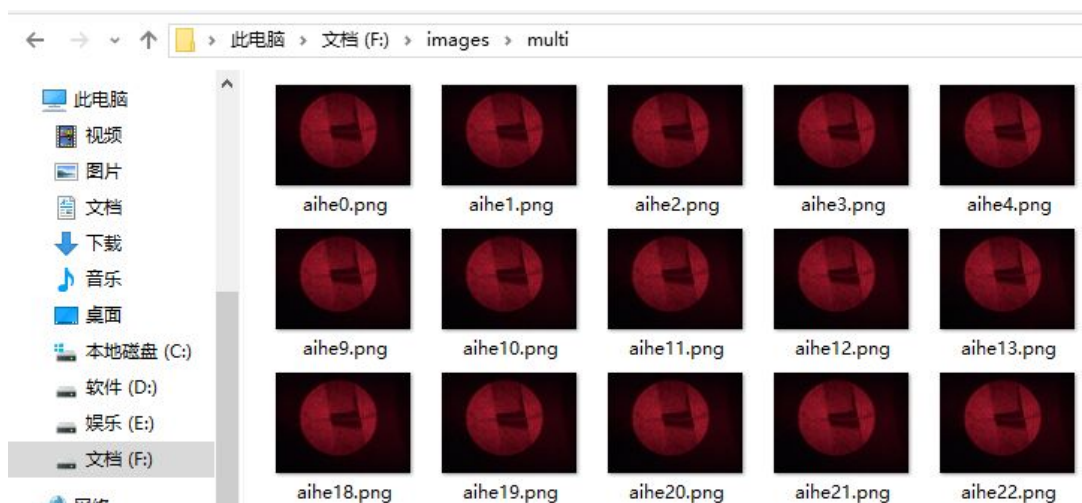
(c) 相机设置



(d) 采集进程



(e) 空间算法实时处理



(f) 图像保存目录示意图

图 4.7 软件在 Windows 系统上的测试结果图：（a）插入相机提示窗；（b）相机预览；（c）相机设置；（d）采集进程；（e）空间算法实时处理；（f）图像保存目录示意图

### 4.3 本章小结

本章主要在实验室环境下，采集工业涂料干燥过程的散斑图像，对 LSI 系统 CCD 相机控制软件的功能进行测试，按照采集过程演示了软件功能。可以看到软件的预览、图像采集、相机参数设置、算法实时处理和结果实时显示等功能运作正常。

## 第 5 章 结论与展望

### 4.1 全文总结

本文主要根据课题要求，采用面向对象编程方法，结合 Qt 与 OpenCV 对程序进行分析和设计。本文的主要工作总结如下：

1. 介绍了激光散斑成像技术的应用和软件开发的背景，并对文章研究的内容和布局做了具体安排与说明；
2. 介绍了 CCD 相机数据采集与控制软件设计的理论基础、相机选型、镜头选型、面向对象编程、Qt 和 OpenCV 的基础知识；
3. 在理论基础之上，分析课题需求，完成程序流程图，进行程序编码设计。完成 LSI 系统 CCD 相机数据采集与控制软件的开发，包括了图像预览、重要相机参数、单张采集、多张采集、保存数据、算法处理并实时显示算法处理结果等功能。

### 4.2 未来展望

LSI 系统 CCD 相机数据采集与控制软件的开发，已完成了基本功能，满足当前实验室的基本研究需求。考虑到软件的扩展性，今后还将实现的功能：

1. 适配所有遵守了欧洲相机联盟标准的相机；
2. 增加多线程并行数据采集；
3. 美化图形用户界面；
4. 完成相机采集之后，可增加更多需要的算法。

## 致 谢

经过一学期的努力，终于写完了这次毕业论文。在这次完成作业的过程中，感谢杜老师的悉心培养与教育。杜老师的每周抽查，让我没有因为生活中的各种事情干扰而放置毕业设计与一边，每周都会在毕设上有一些进展。在做毕设的过程中也有遇到难题，感谢杜老师给我精神上的鼓励，问题一定可以解决的。感谢学院四年来的收留，让我有个实验室可以学习，感谢学校给我提供了这样一个平台可以发挥自己的能力，去做自己喜欢做的东西。最后感谢自己通过自己的努力完成了此次毕业设计。

## 参考文献

- [1] 冯能云.激光散斑处理方法研究[D]. 武汉：华中科技大学, 2012.
- [2] Stetson K.A. A review of speckle photography and speckle interferometry [J]. Opt.Eng., 1975, 14(5):482~489.
- [3] 王凤鹏, 李亮, 尹真, 郭金水, 曾明生. 基于激光散斑的涂料干燥过程实时监测技术[J]. 光子学报, 2013,42(8) : 993~996.
- [4] Fercher, A.F., Briers, J.D., Flow visualization by means of single-exposure speckle photography [J]. Optics Commun, 1981, 37(5):326-330.
- [5] 王凤鹏, 曾全荣, 叶尚臣, 尹真. 基于激光散斑的大豆活力检测实验研究[J]. 应用激光, 2013,33(4) : 452~455.
- [6] 于向阳.机器视觉在仪表监控识别系统中的应用[D]. 济南：山东大学 2015
- [7] Basler. Basler ace acA640-120uc - Area Scan Camera [EB/OL]  
<https://www.baslerweb.com/cn/products/cameras/area-scan-cameras/ace/aca640-120uc/> , 2017.
- [8] 林宝和.基于机器视觉的 Cover glass 外观缺陷检测之设计与实现[D]. 厦门：厦门大学, 2012
- [9] Basler. Ricoh Lens FL-CC1614-2M F1.4 f16mm 2/3" - 镜头[EB/OL]  
<https://www.baslerweb.com/cn/products/vision-components/lenses/ricoh-lens-fl-c1614-2m-f1-4-f16mm-2-3/> , 2017.
- [10] 百度.sdk（软件开发工具包）\_百度百科[EB/OL]  
[http://baike.baidu.com/link?url=IXJs9vnBEYljN\\_nxqfaoTHhf3HUFOUFGxvGL0hNj6KnkGDTjT82ilogJ8fJNN5123-E2KUI0qqeIMRrJRGJqdK](http://baike.baidu.com/link?url=IXJs9vnBEYljN_nxqfaoTHhf3HUFOUFGxvGL0hNj6KnkGDTjT82ilogJ8fJNN5123-E2KUI0qqeIMRrJRGJqdK), 2017.
- [11] Basler. Programmer's Guide and API Reference for pylon[Z]. 2016.
- [12] 桑旻生.基于 QT+Linux 的实验室视频监控系统设计与实现[D] 西安：电子科技大学, 2015.
- [13] 霍亚飞.Qt5 编程入门[C]. 北京：北京航空航天大学出版社, 2015：前言 I
- [14] 梁建适.面向对象和组件技术在图像输出子系统中的应用[D]. 湖北：华中科

- 技大学, 2004.
- [15] 石沙.基于 Qt 的跨平台视频监控客户端的设计与实现[D]. 西安: 电子科技大学, 2013.
- [16] 毛星云.OpenCV3 编程入门[C]. 北京: 电子工业出版社 , 2015: 8-11.
- [17] 段怡婷, 李光宇, 高瞻. 生物散斑测量技术综述[J]. 激光与光电子学进展, 2013: 50, 020003.



## 附 录：LSI 系统 CCD 相机数据采集与控制软件程序

### 1. LSI 系统 CCD 相机数据采集与控制软件程序

见：github:<https://github.com/ai-hes/MyCamera>

### 2. LSI 系统 CCD 相机数据采集与控制软件的 Pro 文件

```
#-----
#
# Project created by QtCreator 2017-05-08T08:50:51
#
#-----
QT      += core gui
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
TARGET = MyCamera
TEMPLATE = app
SOURCES += main.cpp\
           mainwindow.cpp \
           myfram.cpp \
           settings.cpp \
           settingdialog.cpp \
           aboutdialog.cpp
HEADERS  += mainwindow.h \
           myfram.h \
           settings.h \
           settingdialog.h \
           aboutdialog.h

FORMS    += mainwindow.ui \
           myfram.ui \
           settingdialog.ui \
           aboutdialog.ui

RESOURCES += \
           images.qrc

DISTFILES += \
           config.json
macx {
    message("Using Macx configuration")
    QT_CONFIG -= no-pkg-config
    CONFIG    += link_pkgconfig
```

```

PKGCONFIG += opencv

INCLUDEPATH += /opt/local/include
LIBS += -L/opt/local/lib
LIBS += -lboost_thread-mt -lboost_system-mt

INCLUDEPATH                                     +=
/Library/Frameworks/pylon.framework/Headers/GenICam
LIBS += -F/Library/Frameworks -framework pylon
}
win32 {
    message("Using win32 configuration")
    INCLUDEPATH += D:\OpenCV2\opencv\build\include
    INCLUDEPATH += D:\OpenCV2\opencv\build\include\opencv
    INCLUDEPATH += D:\OpenCV2\opencv\build\include\opencv2
    INCLUDEPATH += D:\pylon5\Development\include
    LIBS += -L"D:\pylon5\Development\lib\x64"
    LIBS += -lGCBBase_MD_VC120_v3_0_Basler_pylon_v5_0 \
        -lGenApi_MD_VC120_v3_0_Basler_pylon_v5_0 \
        -lPylonBase_MD_VC120_v5_0 \
        -lPylonC_MD_VC120 \
        -lPylonGUI_MD_VC120_v5_0 \
        -lPylonUtility_MD_VC120_v5_0

    OPENCV_PATH = D:\OpenCV2\opencv
    LIBS_PATH = "$$OPENCV_PATH/build/x64/vc12/lib"
    LIBS += -L$$LIBS_PATH \
        -lopencv_core2413d \
        -lopencv_imgproc2413d \
        -lopencv_photo2413d \
        -lopencv_ml2413d \
        -lopencv_calib3d2413d \
        -lopencv_contrib2413d \
        -lopencv_features2d2413d \
        -lopencv_flann2413d \
        -lopencv_gpu2413d \
        -lopencv_legacy2413d \
        -lopencv_objdetect2413d \
        -lopencv_ts2413d \
        -lopencv_video2413d \
        -lopencv_nonfree2413d \
        -lopencv_ocl2413d \
        -lopencv_stitching2413d \
        -lopencv_superres2413d \

```

```
-lopencv_videostab2413d \
-lopencv_highgui2413d
}
```