

LinJK

太急没有故事，太缓没有人生。

公告

昵称：LinJK

园龄：1个月

粉丝：1

关注：1

+加关注

<

2016年12月

>

日	一	二	三	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

我的标签

Android反编译(1)

BLE(1)

boost(1)

CentOS(1)

Java(1)

mac(1)

NDK(1)

opencv(1)

Proguard(1)

React Native(1)

更多

随笔分类

Android(8)

C++(1)

Docker(1)

Mac平台下Opencv开发环境搭建

OpenCV(Open Source Computer Vision Library), 是一个开源的跨平台的计算机视觉库，它实现了图像处理 and 计算机视觉领域的很多通用算法，可以在多种计算机平台上运行，支持C++、C#、Python等语言，也提供了 Ruby、Matlab等语言的接口。OpenCV是基于BSD许可证发布的，因此它可以免费应用于科研或商业领域。

可以在这个地址了解更多以及下载OpenCV安装包：<http://opencv.org>, 其GitHub仓库地址为：<https://github.com/opencv/opencv>，那里有主仓库，拓展模块仓库和实验测试仓库，大家可以看上面的readme文件和代码来进行学习。

因为我使用的是Mac电脑，懒得每次都要打开windows虚拟机使用Visual Studio编程，所以，决定在os x系统使用xcode来进行编译运行，下面介绍一下整个流程步骤。

1. 安装

在OpenCV官网下载的安装包适用于Linux/Mac，因此需要使用CMake工具来进行编译安装，尝试了几次，都在最后的链接那里失败了，在StackOverFlow查了才发现，目前opencv只支持使用xcode6版本的编译链接口，我使用的是xcode7的，为了避免不必要地麻烦，就不采取编译安装方式了，去homebrew搜索一下，可以下载：

```
apple at MBP in ~  
$ brew search opencv  
homebrew/science/opencv
```

现在执行"brew install opencv"，等待即可安装完成，安装后路径是：/usr/local/opt/opencv3，我们可以设置一个OPENCV3_HOME环境变量，方便以后快速打开这个文件夹。该文件夹下有opencv的核心和扩展模块，目录结构如下：

在lib目录下是opencv的库，在include目录下包含了两个子目录：opencv和opencv2，opencv里面包含了opencv1.x的核心头文件，opencv2安装模块功能组织，包括核心模块、图像处理模块、2D功能模块、高层

http://www.cnblogs.com/linjk/p/6029306.html

第 1 页 (共 9 页)

Embedded System(1)

iOS(4)

Linux(7)

Machine Learning

OpenCV(2)

Windows

随笔档案

2016年12月 (10)

2016年11月 (12)

最新评论

1. Re:Mac平台下Opencv开发环境搭建

@(^_^)程序员在iOS端我是直接引用opencv.framework的, pc端没这样试过哦, 可以试下😁...

--LinJK

2. Re:Mac平台下Opencv开发环境搭建

我直接下载了 opencv.framework , 没有上述的安装过程, 能用吗? --(^_^)程序员

阅读排行榜

1. Mac平台下Opencv开发环境搭建(274)

2. Android平台下OpenCV移植与使用---基于C/C++(190)

3. Mac下Boost环境搭建(118)

4. Android Studio增加NDK代码编译支持--Mac环境(89)

5. 聊聊Android的APK反编译(49)

评论排行榜

1. Mac平台下Opencv开发环境搭建(2)

推荐排行榜

1. Android平台下OpenCV移植与使用---基于C/C++(1)

GUI图像用户界面模块、机器学习模块等。

注意，这里安装的是3.1.0版本的OpenCV，在官网可以看到，这个版本只支持64位编译器，如果在windows平台开发，建议选择现在还很常用的2.4.9版本，随着64位机基本普及，使用3.0版本更方便，不用再包含一堆的lib了，只要几个就够了。

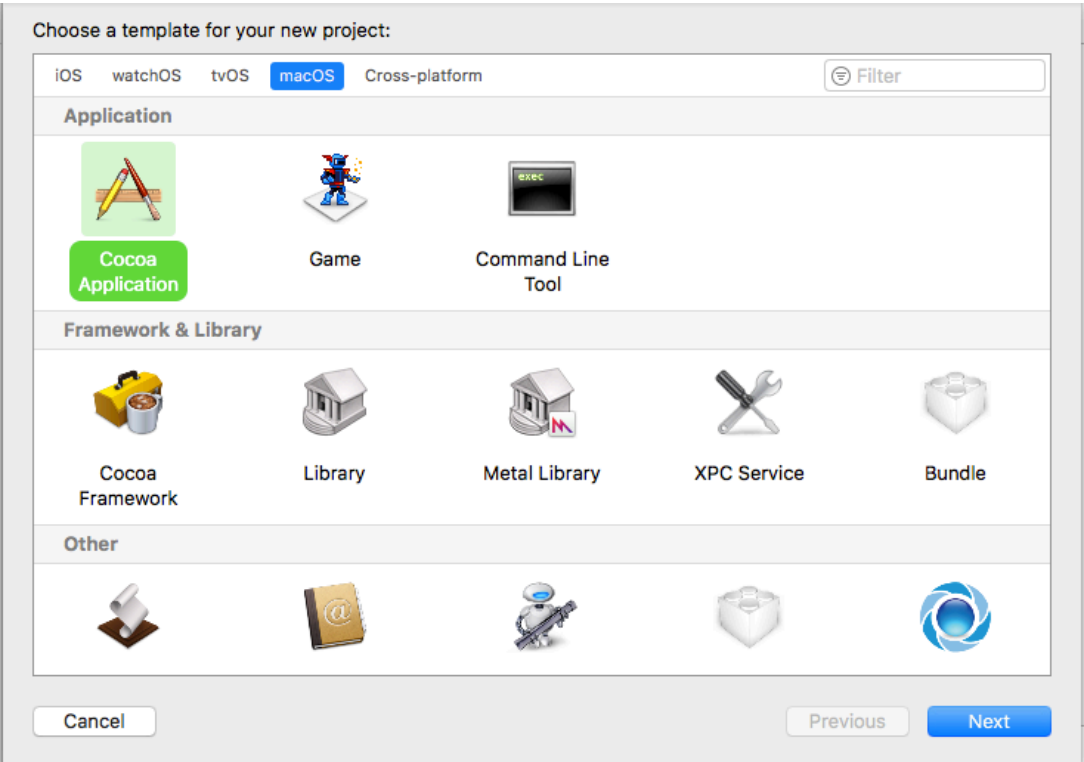
2. 环境配置

开发平台IDE的环境大同小异，基本都是选择编程语言后就确定了编译器，然后是头文件路径的设定和库文件路径的设定让编译器能找到就行，然后就可以进行编译链接成最终的可执行文件了，按照这个思想，就不难处理问题了。

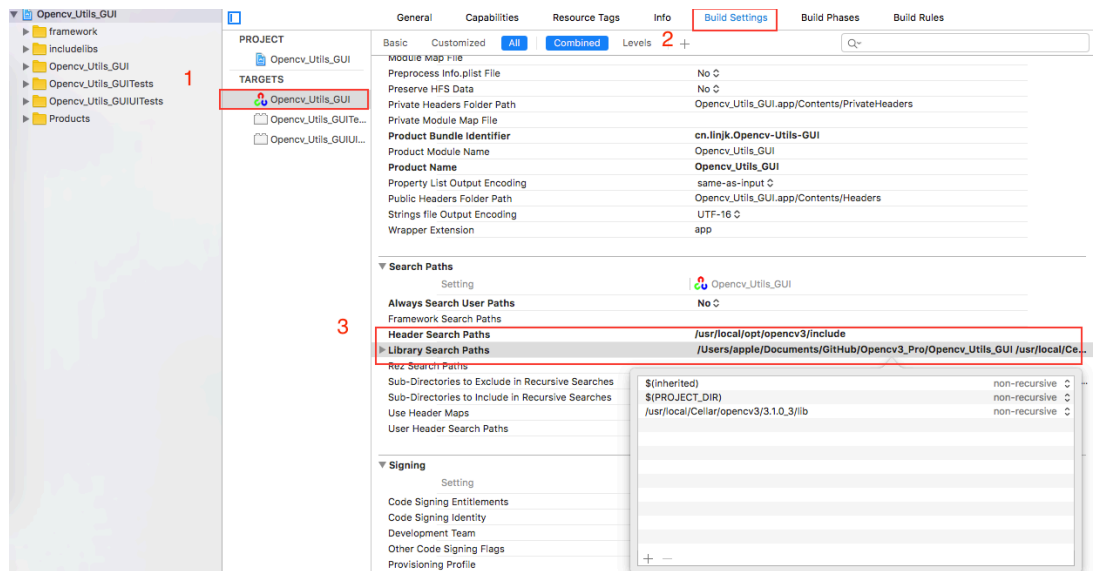
2.1 新建xcode项目

一般教程都会新建command line形式的项目，我不太喜欢这样的单个功能单个项目的形式，这样学下来就很多项目了，也麻烦，毕竟演示给别人看的时候，用gui软件形式比命令行软件形式更好看吧😁，这里新建工程就使用cocoa application形式吧。

打开XCode，新建工程，选择macOS模式，点击"cocoa application"，下一步，输入工程名和语言，这里选择object-c，下一步，选择工程存放路径，点击"done"即可。

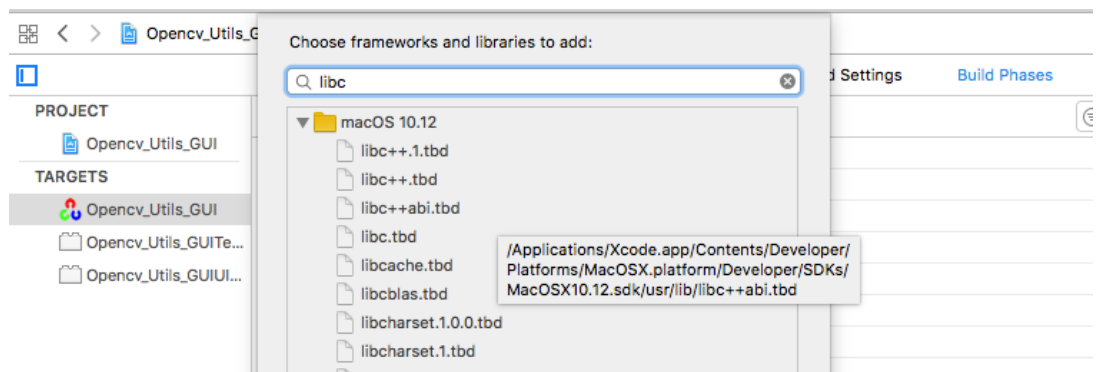


2.2 配置工程头文件和库文件搜索路径，如下图顺序所示：

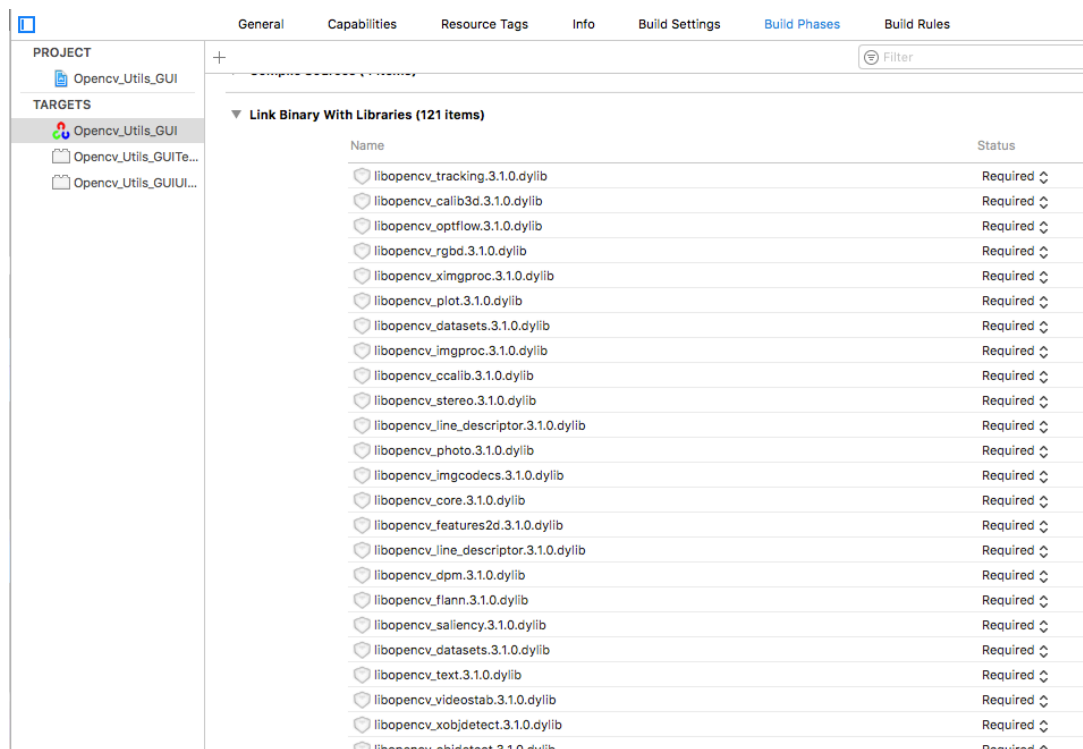


在Library Search Paths记得要加上“\$(PROJECT_DIR)”，这样在工程里添加的库框架也能搜索到，另外，因为OpenCV使用C/C++编写的，还需要引入C++动态链接库：

在Build Phase的Link Binary With Libraries下，添加时搜索：“libstdc++”，选中后点击“Add”即可：



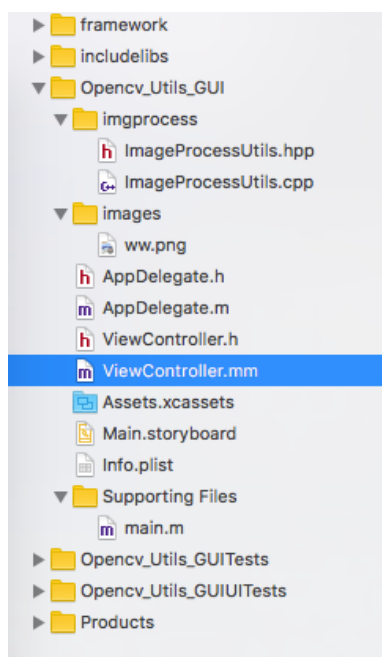
最后，再添加对opencv库的引用，添加完成后如下图所示：



到这里，开发环境就配置完成了，在visual studio也差不多，都是配置编译时搜索头文件和库文件的路径，这里就不重复了。

3. 测试opencv功能

这里使用这个工程，主要是为了测试opencv的功能函数，把公共的功能提取出来，然后方便以后移植到android和ios平台下，所有，工程目录组织如下：



在imgprocess目录下是通用的图像处理函数，images目录是默认实例图片，其他文件就是cocoa编程的文件了，重点是imgprocess目录的类文件。

我们再Main.storyboard里搭建大概的用户界面程序，如下：



ImageView和Button的IBOutlet和IBAction关联和iOS开发一样，这里就不多说了。

编写图像处理头文件，如下：

```
< > Opencv_Utils_GUI > Opencv_Utils_GUI > imgprocess > ImageProcessUtils.hpp > No Selection
// Created by apple on 9/7/16.
// Copyright © 2016 LinJK. All rights reserved.
//

#ifndef _IMAGEPROCESSUTILS_H_
#define _IMAGEPROCESSUTILS_H_

#ifdef check
#define OS_X_STUPID_CHECK_MACRO check
#undef check
#endif

#include <opencv2/opencv.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>

#ifdef OS_X_STUPID_CHECK_MACRO
#define check OS_X_STUPID_CHECK_MACRO
#undef OS_X_STUPID_CHECK_MACRO
#endif

using namespace cv;

class ImageProcessUtils{
public:
    /* 图像转换为灰度 */
    Mat imageToGray(String inputImagePath);
    /* 图像腐蚀
     - 用图像中暗色部分"腐蚀"掉图像中高亮部分
    */
    Mat imageErode(String inputImagePath);
    /* 图像均值滤波 */
    Mat imageBlur(String inputImagePath);
    /* 图像边缘检测 - Canny算子 */
    Mat imageBorderDetectWithCanny(String inputImagePath);
};

#endif
```

这里特别要注意圈住的地方，需要加上才能编译连接通过。

这里声明了一些图像处理函数，现在来看看cpp文件的实现：

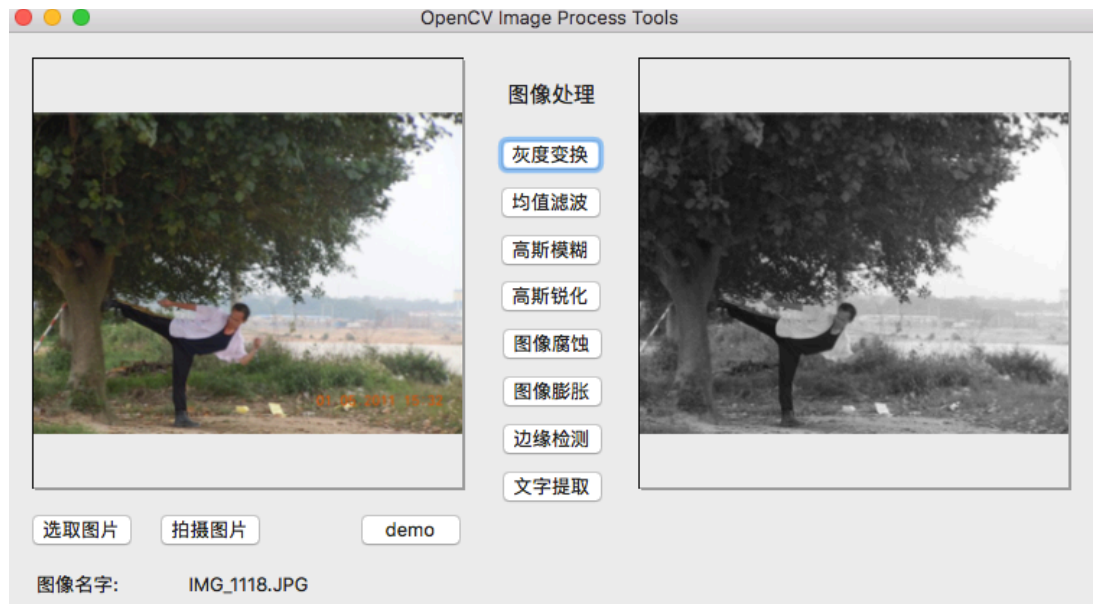
```
9  #import "ImageProcessUtils.hpp"
10
11  /** 图像转为灰度 */
12  Mat ImageProcessUtils::imageToGray(String inputImagePath) {
13      cv::Mat gray;
14
15      Mat input = imread(inputImagePath).clone();
16
17      cvtColor(input, gray, cv::COLOR_BGR2GRAY);
18
19      return gray;
20  }
21
22  /** 图像均值滤波 */
23  Mat ImageProcessUtils::imageBlur(String inputImagePath) {
24      Mat result;
25
26      Mat input = imread(inputImagePath).clone();
27
28      blur(input, result, Size(7, 7));
29
30      return result;
31  }
32
33  /** 图像腐蚀 */
34  Mat ImageProcessUtils::imageErode(String inputImagePath) {
```

图像处理部分函数准备好了，现在来验证一下，这里看看灰度变化怎么样：

函数如下：

```
9  #pragma mark - Actions
10  /**按钮 - 默认图片*/
11  -(IBAction)btnDemoImageTapped:(id)sender {
12      [_imageInput setImage:[UIImage imageNamed:@"ww.png"]];
13  }
14
15  /**按钮 - 选取图片*/
16  -(IBAction)btnSelectFileTapped:(id)sender {
17      NSOpenPanel *panel = [[NSOpenPanel alloc] init];
18      panel.delegate = self;
19
20      [panel beginSheetModalForWindow:[sender window] completionHandler:^(NSInteger result) {
21          //Do nothing ...
22      }];
23  }
24
25  /**按钮 - 灰度变换*/
26  -(IBAction)btnImageToGrayTapped:(id)sender {
27      NSLog(@"将进行二值化处理...");
28      Mat output = ImageProcessUtils().imageToGray([_curFullFilePath UTF8String]);
29      imwrite([PATH_IMAGE_OUTPUT_GRAY UTF8String], output);
30
31      [_imageOutput setImage:[UIImage alloc] initWithContentsOfFile:(NSString *)PATH_IMAGE_OUTPUT_GRAY];
32  }
```

执行结果：



功能没问题~~

接下去就是其他处理函数的编写和测试了，测试完成后，基本很小的修改就可以移植到android和ios平台了。

在打开照片文件选择图片那里，代码如下：

```
#pragma mark - Delegates
#pragma mark - NSOpenSavePanelDelegate
- (void)panel:(id)sender didChangeToDirectoryURL:(NSURL *)url {
    _curBaseFilePath = url.absoluteString;
    NSLog(@"当前文件夹路径 : %@", _curBaseFilePath);
}

- (BOOL)panel:(id)sender validateURL:(NSURL *)url error:(NSError **)outError {
    if (url.isFileURL) {
        NSString *stringFileFullPath = url.absoluteString;
        NSLog(@"文件全路径 : %@", stringFileFullPath);
        NSString *fileName = [stringFileFullPath substringFromIndex:[stringFileFullPath
            localizedStandardRangeOfString:_curBaseFilePath].length];
        NSLog(@"- 文件名: %@", fileName);
        [_tvImageName setStringValue:fileName];
        if ([fileName hasSuffix:@".png"] ||
            [fileName hasSuffix:@".jpg"] ||
            [fileName hasSuffix:@".JPG"]) {
            _curFullFilePath = [url.absoluteString substringFromIndex:7]; //去掉"file://"前缀
            UIImage *inputImage = [[UIImage alloc] initWithContentsOfURL:url];
            [_imageInput setImage:inputImage];

            return YES;
        }
        else {
            UIAlertView *alert = [[UIAlertView alloc] init];
            [alert setMessageText:@"请选择图片格式文件"];
            [alert runModal];
            return NO;
        }
    }

    return YES;
}

return NO;
}
```

好了，在mac平台搭建opencv开发环境到此结束，后面不断完善先，有时间再补充这部分。

GitHub: <https://github.com/linjk> LinJK: <https://www.linjk.cn:3298>

分类: [OpenCV](#)

标签: [mac](#), [opencv](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

LinJK

[关注 - 1](#)[粉丝 - 1](#)[+加关注](#)« 上一篇: [聊聊Android的APK反编译](#)» 下一篇: [LNMP平台搭建---Linux系统安装篇](#)

posted @ 2016-11-04 10:25 LinJK 阅读(274) 评论(2) 编辑 收藏

0

0

评论列表

#1楼 2016-12-20 14:56 (^_^)程序员

我直接下载了 opencv.framework , 没有上述的安装过程, 能用吗?

支持(0) 反对(0)

#2楼[楼主] 2016-12-21 09:31 LinJK

@ (^_^)程序员

在iOS端我是直接引用opencv.framework的, pc端没这样试过哦, 可以试下😏

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)**注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。**

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【福利】微软Azure给博客园的你专属双重好礼

【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍

【推荐】BPM免费下载



最新IT新闻:

- 知乎告微博大号侵权背后的细节告诉我们: 删帖子易, 破保护伞难
- 明年Google将发布两款智能手表, 支持Android Pay
- 免费办公软件LibreOffice 5.2.4发布下载
- 法国建世界首条太阳能公路: 一公里3600万元
- VR项目坑太多? 这里有8个“血泪”贴士供参考

» 更多新闻...



最新知识库文章:

- 写给未来的程序媛
- 高质量的工程代码为什么难写
- 循序渐进地代码重构
- 技术的正宗与野路子
- 陈皓：什么是工程师文化？
- » 更多知识库文章...

Copyright ©2016 LinJK