
编译更加强大的 opencv (opencv_contrib b 编译方法)

摘 要

Opencv 在 3.0 之后版本的 Opencv 对 Sift、Sufer 等新算法没有提供直接编译好的动态库来加载使用，因此必须自己编译。这是因为 Sift、Sufer 算法专利现在是属于哥伦比亚大学，因此 opencv 对这部分内容及“所谓的”不稳定模块全都放到 opencv_contrib 中。且只在它的 Github 中能够找到，opencv 的官网上是没有的。因此对于想使用 Sift、Surf、双目立体匹配算法、追踪、结构光等新的算法及有专利权算法的同学需要自行编译。

编译 contrib 会有一些麻烦，其中有两个最大的坑，一个是 opencv 和 contrib 的版本一定要完全一致，另一个是使用 CMake 的时候自动下载必要依赖项的时候经常会下载失败，文档中会有解决方案。

杭州电子科技大学
机器视觉技术与应用
课程资料

目 录

1	环境与下载.....	3
1.1	操作环境	3
1.2	下载 opencv 和 opencv_contrib 源码.....	3
2	编译源码.....	6
2.1	第一次编译	6
2.2	第二次编译	9
2.3	用 vs 生成 release 和 debug 动态链接库.....	11
2.4	配置并生成属性表	12
3	测试程序.....	15
3.1	生成测试程序	15
3.2	加载属性文件表 。	错误!未定义书签。
	参考文献.....	17
	附录.....	18
	版本.....	19

1 环境与下载

Opencv 里面一些比较新的算法, 如一些跟踪算法、SIFT、SURF、3D 匹配等等非常多的算法被单独提取出来放在 opencv_contrib 模块中, 官方的说法是这些算法的稳定性没有得到验证, 所以没有加入 opencv 的正式版中, 实际上可能涉及到一些版权问题, 如 Sift 特征点是有专利的, 专利权属于哥伦比亚大学。这些算法的代码在 opencv 的官网上是没有的, 在 github 的主页上可以下载到, 并且下载的源码是没有编译过的, 需要自己编译。**在编译的时候需要 Opencv 的源码和 Opencv_contrib 的源码, 两个的版本号务必要一致, 不然会报错, 这里以 3.4.2 为例。**

1.1 操作环境

系统: win7 64 位 (win10)

工具: VS2015 、 CMake 3.12.0

源码: Opencv 3.4.2

Opencv_contrib 3.4.2

1.2 下载 opencv 和 opencv_contrib 源码

打开 opencv 在 github 的主页: <https://github.com/opencv>

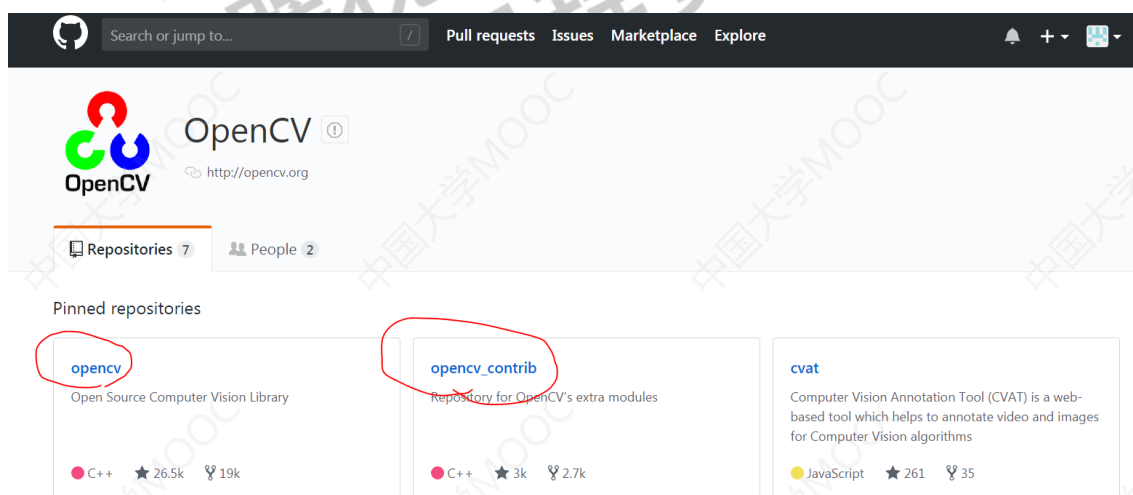


图 1-1 opencv 的 github 主页。

可以看到左下的“opencv”和“opencv_contrib”的链接。

首先进入 opencv 的页面

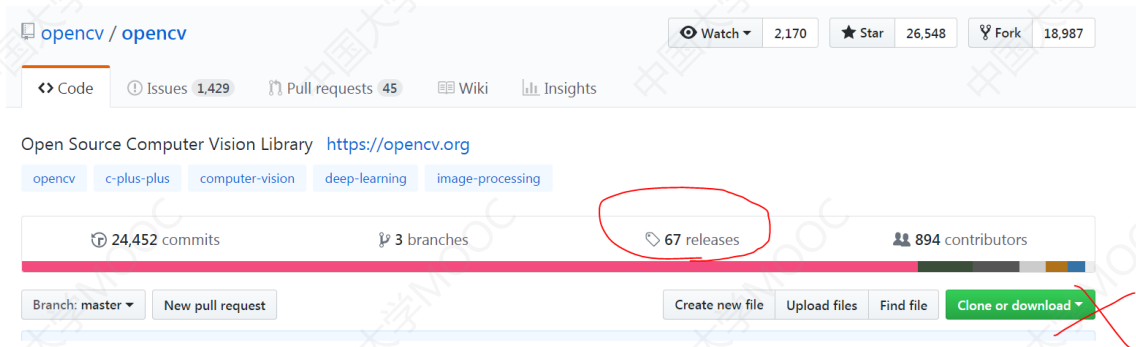


图 1-2 下载 opencv 的源码。

这里不要点击右下角的“Clone or download”下载，因为不一定会下载到你要的版本。可以点击“releases”下载。

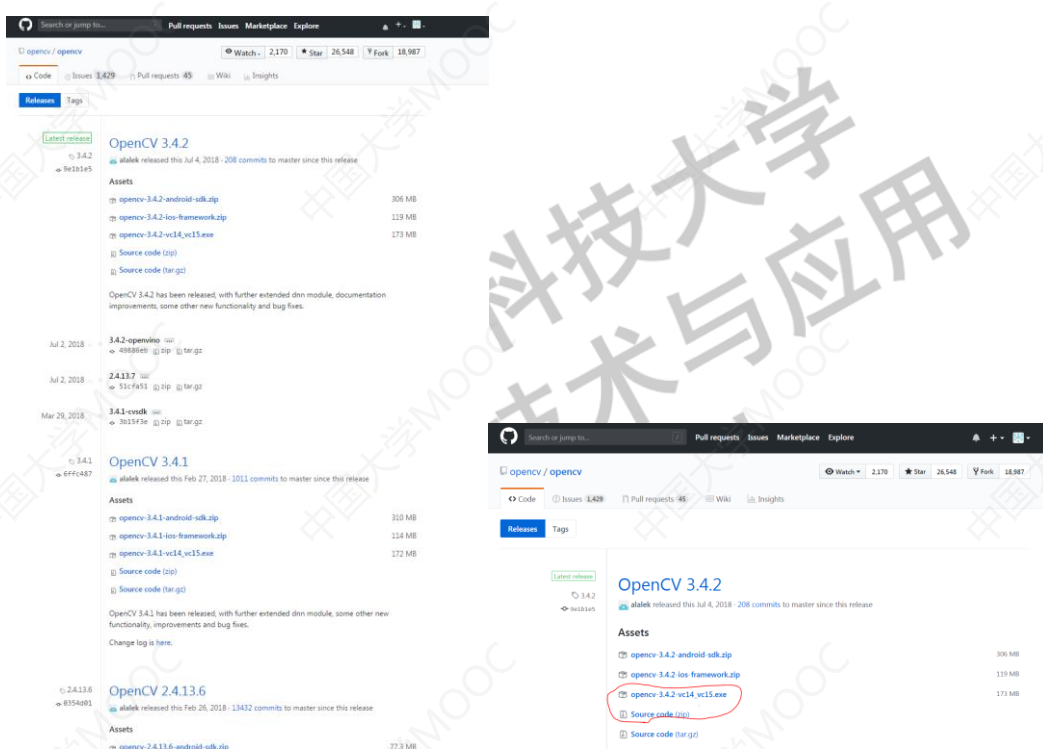


图 1-3 找到自己想要的版本的源码。

点击“opencv-3.4.2-vc14_vc15.exe”或者“Source code”都可以，一个是自解压文件，一个是压缩包。

然后下载“opencv_contrib”

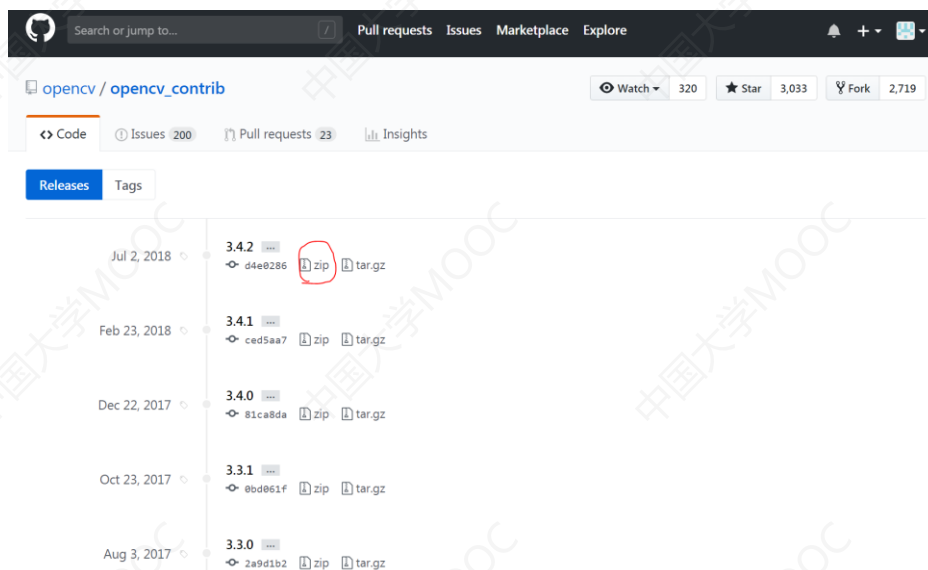


图 1-4 下载需要的 contrib 版本。

下载编译工具 CMake，网址：<https://cmake.org/download/>

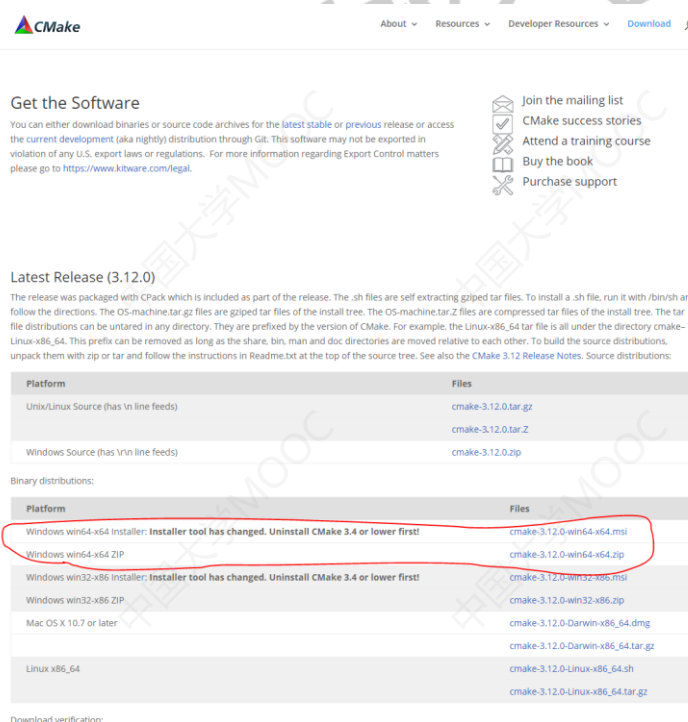


图 1-5 下载 CMake。

下载和自己电脑相对应的版本。我这里下载的是“[cmake-3.12.0-win64-x64.msi](#)”
下载完成后，点击安装。



图 1-6 CMake 快捷方式。

2 编译源码

2.1 第一次编译

在硬盘上合适位置建立文件夹进行编译。我这里是"D:\opencv_contrib_342\"

把"opencv"和"opencv_contrib"拷到文件夹内并解压，并建立"build"文件夹作为编译结果的保存位置。

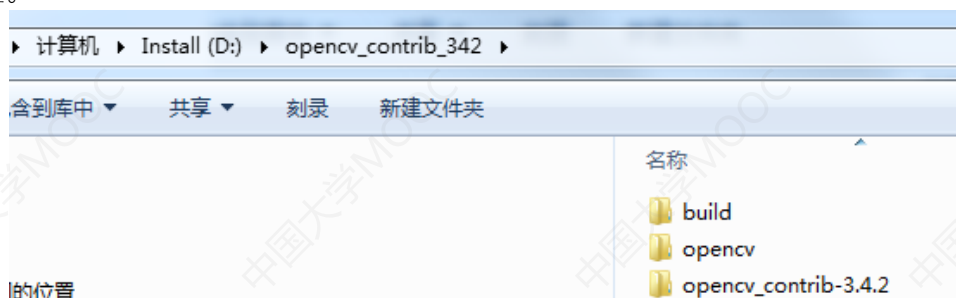


图 2-1 建立编译用的文件夹。

打开 CMake，选择路径。Where is the source code 是 opencv 源码路径，即“源路径”，Where to build the binaries 是要生成 opencv 的 vs 工程路径，即“目的路径”。我们分别如图 2-2 设置。

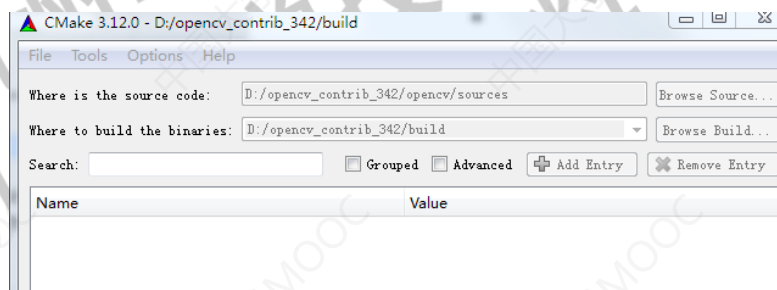


图 2-2 选择编译路径。

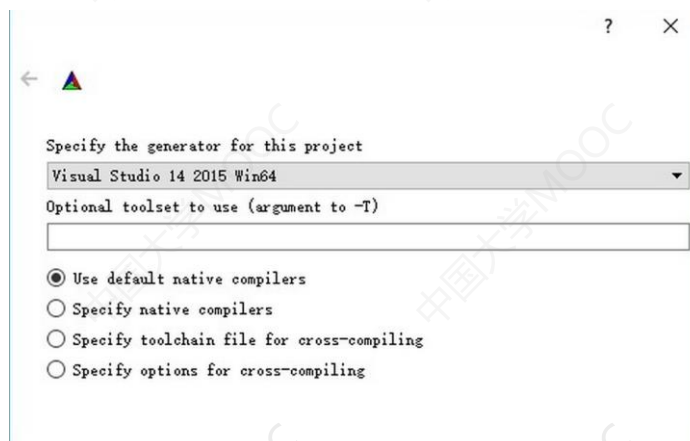


图 2-3 选择编译平台。

点击“Configure”，弹出对话框，根据自己的实际情况选择编译工程的平台。我选的是 64 位的 VS2015。然后点击“finish”。即开始配置。这需要一定的时间，期间要下载一些依赖项，此时通畅的网络就起了关键作用，如果下载不全，可通过查看日志文件 CMakeDownloadLog.txt 手动下载。

下载的文件会被放在“D:\opencv_contrib_342\opencv\sources\cache”文件夹下。

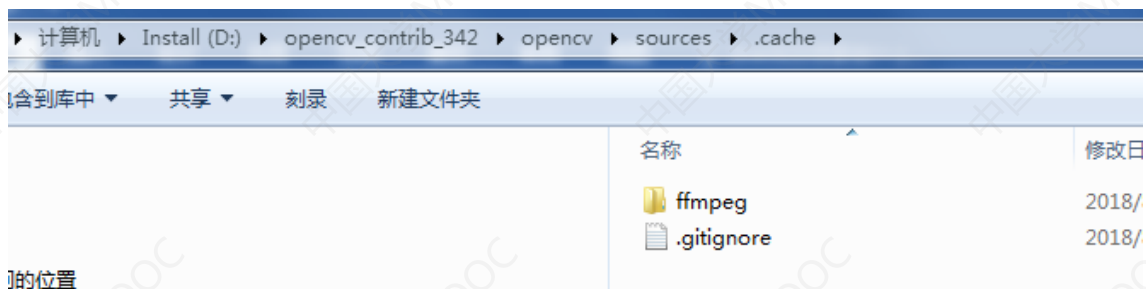


图 2-4 第一次编译下载文件的临时文件夹。

这里“opencv_ffmpeg.dll”、“opencv_ffmpeg_64.dll”、“ffmpeg_version.cmake”和“ippicv” 文件下载失败。如图 2-5，就是下载“opencv_ffmpeg.dll”的时候链接超时，导致下载失败。

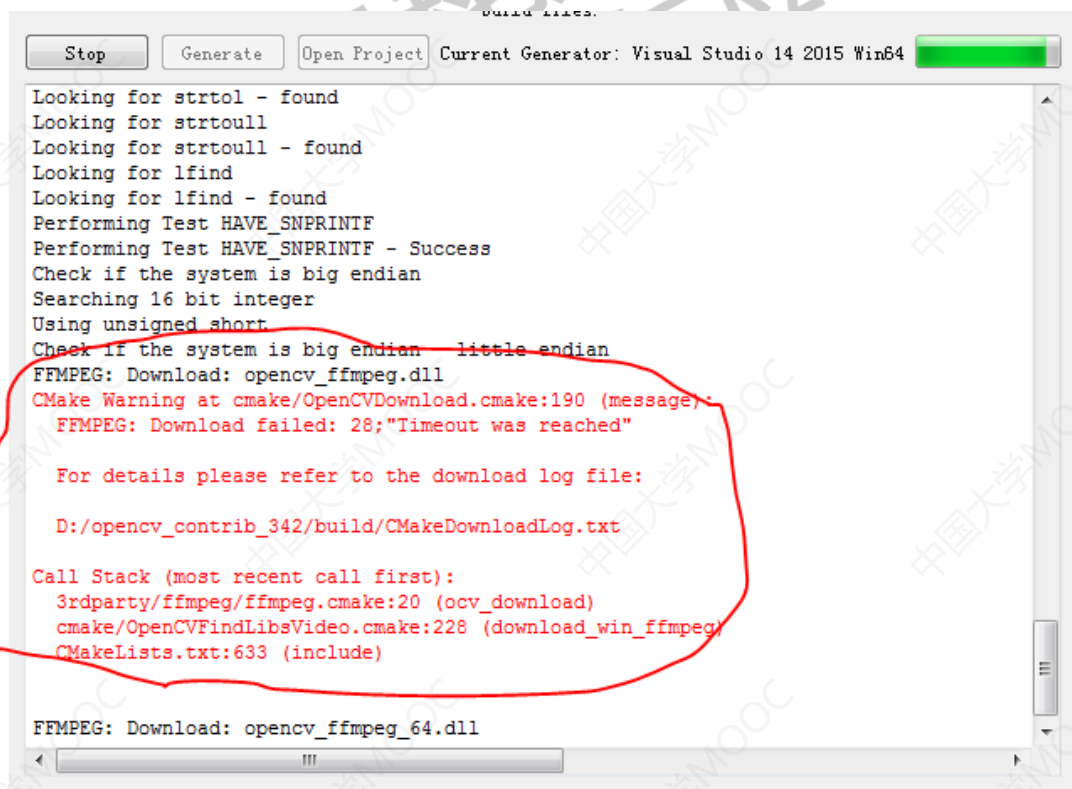


图 2-5 “opencv_ffmpeg.dll”下载失败。

这时候在“.cache”文件夹下有可能会看到“opencv_ffmpeg.dll”文件，如图 2-6，但是注意后面的文件大小。如果下载完全的话，文件应该是 16.3MB，图中只有 4M 说明没有下载成功。

For details please refer to the download log file:

D:/opencv_contrib_342/build/CMakeDownloadLog.txt

Call Stack (most recent call first):

```
3rdparty/ippicv/ippicv.cmake:42 (ocv_download)
cmake/OpenCVFindIPP.cmake:243 (download_ippicv)
cmake/OpenCVFindLibsPerf.cmake:12 (include)
CMakeLists.txt:634 (include)
```

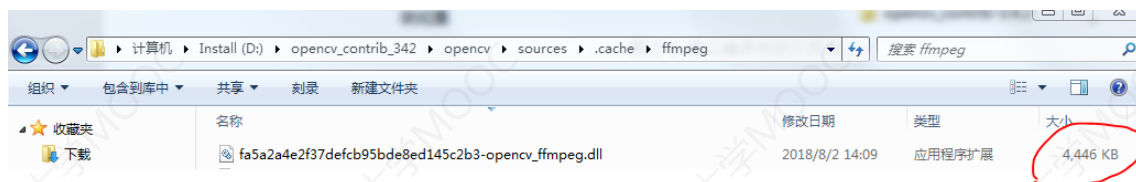


图 2-6 “opencv_ffmpeg.dll”的文件大小不对，说明下载中断导致失败。

等一会以后第一次有很多红色选项，是编译报错。另外在 CMake 下面的输出信息框里会有输出的错误信息。这里最常见的错误就是“opencv_ffmpeg.dll”、“opencv_ffmpeg_64.dll”、“ffmpeg_version.cmake”和“ippicv_2017u3_win_intel64_general_20180518.zip” 文件下载失败。



图 2-7 “opencv_ffmpeg.dll”的文件大小不对，说明下载中断导致失败。

解决方法：可以打开下载的 Log 文件如“CMakeDownloadLog.txt”，如图 2-7 所示。在这里路径是“D:/opencv_contrib_342/build/CMakeDownloadLog.txt”。找到对应的下载链接，用迅雷之类的工具自行下载，然后将下载的文件放到相应的目录中。再次点 Configure,直到不出现红字错误为止。



图 2-8 下载地址。

下载完以后，修改文件名，替代掉原文件。

“opencv_ffmpeg.dll”和“opencv_ffmpeg_64.dll”两个文件改为“.cache\ffmpeg”文件夹里面的名字，我这里是“D:\opencv_contrib_342\opencv\sources\cache\ffmpeg”。其实就是在前面加上哈希值再加上一个半角“-”。然后同样修改“ippicv_2017u3_win_intel64_general_20180518.zip”文件，位置在“.cache\ippicv”。然后再按一次“Configure”。这时候应该很快完成，并且没有红色报错信息。

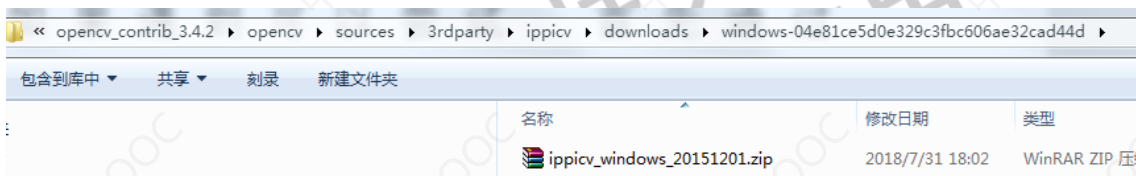


图 2-9 ippicv 的放置位置。

2.2 第二次编译

修改一个地方。

找到 OPENCV_EXTRA_MODULES_PATH 项，如下图所示，设置 contrib 库的 modules 目录，如图 2-5 我这里是“D:\opencv_contrib_3.4.2\opencv_contrib-3.4.2\modules”。这样会把 contrib 里面的内容一起编译进去。然后点“Configure”进行第二次配置。

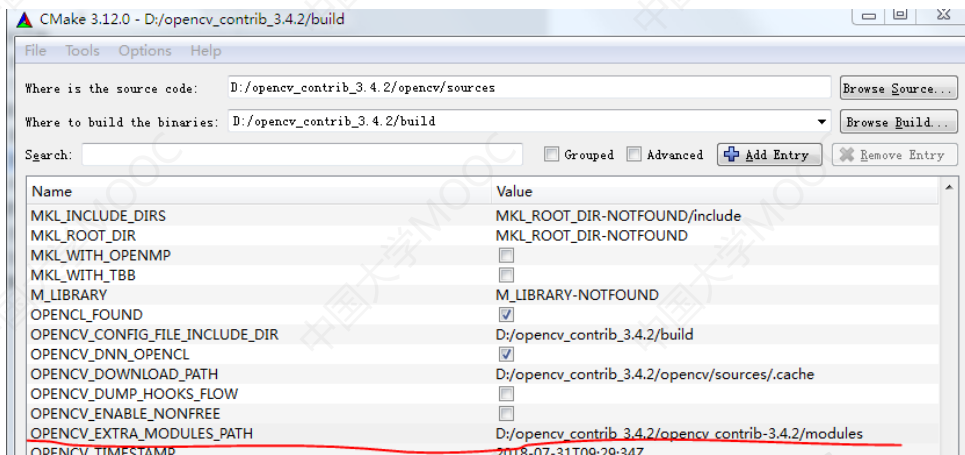


图 2-10 选择 modules 路径。

然后点击“Configure”，这个时候 CMake 还会下一些需要的依赖项。看网络情况，有可能会有一些下载失败。这些文件也会被下载到“D:\opencv_contrib_342\opencv\sources\cache\”下载。

同样，打开“D:/opencv_contrib_342/build/CMakeDownloadLog.txt”。找到对应的下载链接，用迅雷之类的工具自行下载，然后将下载的文件改名后放到相应的目录中。实际上你进入相应的文件夹会看到下载文件是存在的但是文件尺寸不对，也是下载到一半连接超时导致下载失败。

如图 2-11 所示，这次编译中，“face_landmark_model.dat”文件下载失败。找到下载地址，下载后保存在 log 文件中显示的存放位置。



图 2-11 “face_landmark_model.dat”文件下载地址。

在实际编译中，有时候还会遇到过“D:\opencv_contrib_342\build\downloads\features2d”中的一些文件下载失败。如图 2-12 所示，也和前述同样操作。**一定要把没有下载成功全部下载下来放在相应位置**，直到没有报错才行。不然后面编译会出错。网上有些教程说只要参数那栏没有红色覆盖就可以，是不对的，有可能再次 Configure 以后，红色会消失，但是只要下面的消息栏中有报错信息就一定要解决，不然后面编译会有很多奇怪的错误。

```
Tesseract: NO
xfeatures2d/boostdesc: Download: boostdesc_bgm.i
xfeatures2d/boostdesc: Download: boostdesc_bgm_bi.i
xfeatures2d/boostdesc: Download: boostdesc_bgm_hd.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_064.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_128.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_256.i
xfeatures2d/boostdesc: Download: boostdesc_lbgm.i
xfeatures2d/vgg: Download: vgg_generated_48.i
xfeatures2d/vgg: Download: vgg_generated_64.i
xfeatures2d/vgg: Download: vgg_generated_80.i
xfeatures2d/vgg: Download: vgg_generated_120.i
```

图 2-12 xfeatures2d 下的一些文件。

文件全部准备好后再次“Configure”，如果不缺文件的话会看到速度很快，完成并没有错误之后点击“Generate”生成编译。如果看到“Configuring done”和“Generating done”说明编译成功。

网上有些教程说只要看到两个 done 就说明编译成功，这也是不对的，一定要如前述把所有确实文件补齐。

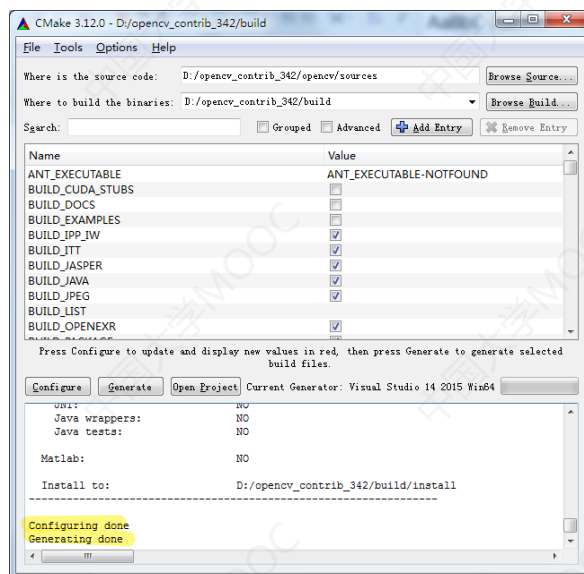


图 2-13 编译成功。

2.3 用 vs 生成 release 和 debug 动态链接库

上一步结束后，点击 cmake 中的“Open Project”这会自动打开 vs，也可以到“build”文件夹下找到 OpenCV.sln 打开。

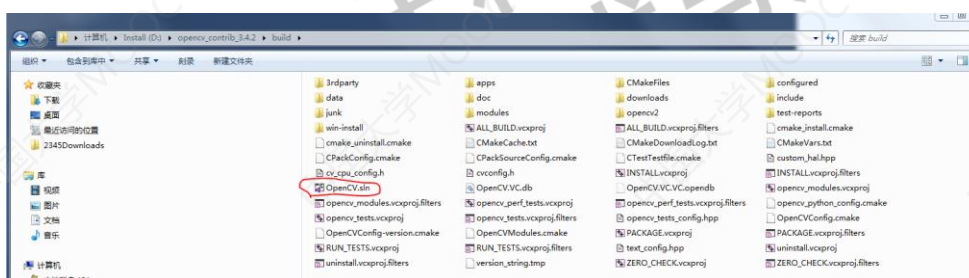


图 2-14 生成的工程文件。

打开 vs 后一般可以直接看到“解决方案资源管理器”，可以看到 OpenCV 的解决方案。选择“Release”和你的平台，我的是“x64”。右键解决方案→“重新生成解决方案”。

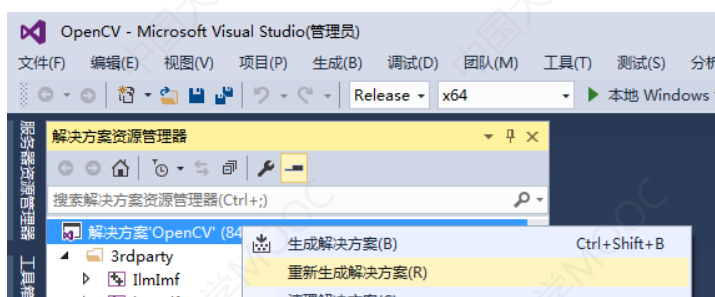


图 2-15 重新生成解决方案。

基本上只要前面不缺少文件，编译都不会有什么问题，有些奇奇怪怪的问题，网上有类似某个位置命名有该文件，但是编译的时候还是提示缺少该文件，很有可能是文件根本就没有下载成功，下载了一般连接失败，这样是不会通过 CMake 的哈希验证，还是认为缺少这个

文件。编译完成后如果报“失败 0 项”就说明全部生成成功。

最后在解决方案管理器中，右键“INSTALL”按如图所示选择即可（Debug 下编译生成一次得到尾号带 d 的 dll,Release 下编译生成一次得到不带 d 的 dll，可分别生成一次。然后 build 文件夹下的 install 即成了新的包含了 contrib 模块的 opencv 库。编译后文件夹很大，会有几个 G，只要保存 install 里面的文件就可以了。之后便可以进行环境配置。

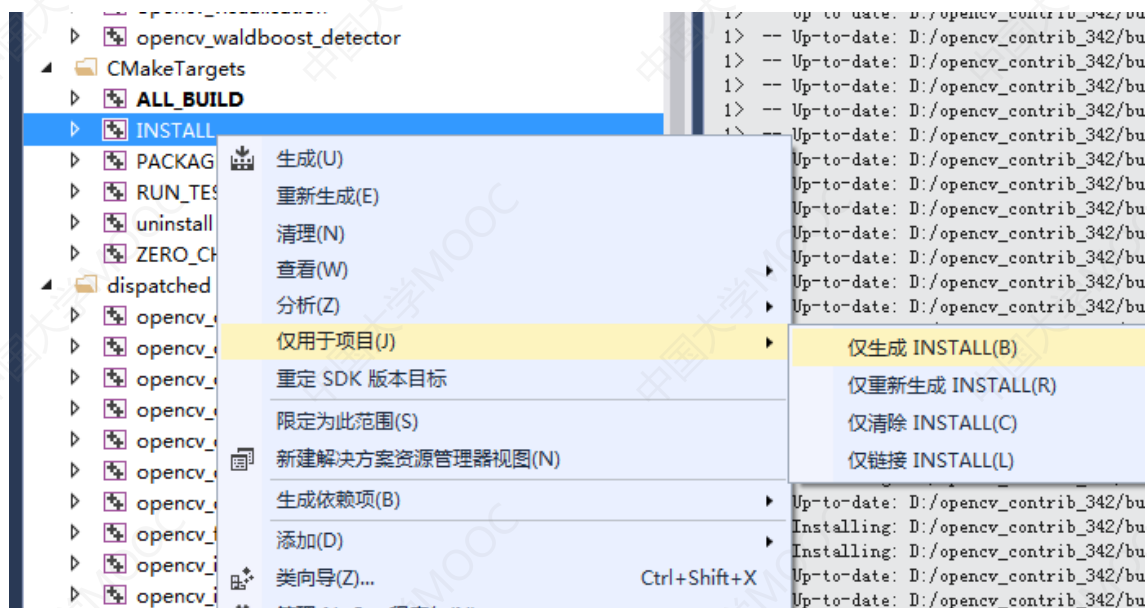


图 2-16 生成最终的库文件。

2.4 配置并生成属性表

这里基本上和普通配置 opencv 一样，具体的参考文档“VS2015 下的 OpenCV 的配置 (C++)”。首先把生成的新的的库路径添加到系统变量 Path 中，我把 install 下的内容移到了 “D:\opencv_contrib_342_install\”下，所以把 “D:\opencv_contrib_342_install\x64\vc14\bin” 添加到系统变量 Path 中。

“计算机”→右键“属性”→“高级系统设置”→“高级”→“环境变量”→“Path”。

然后建立属性表，在属性表中做如下配置，实际目录根据自己的目录修改

① 包含目录：

D:\opencv_contrib_342_install\include

D:\opencv_contrib_342_install\include\opencv

D:\opencv_contrib_342_install\include\opencv2

② 库目录：

D:\opencv_contrib_342_install\x64\vc14\lib

D:\opencv_contrib_342_install\x64\vc14\staticlib

③ 附加依赖项:

Debug 模式:

opencv_aruco342d.lib

opencv_bgsegm342d.lib

opencv_calib3d342d.lib

opencv_ccalib342d.lib

opencv_core342d.lib

opencv_datasets342d.lib

opencv_dnn342d.lib

opencv_dpm342d.lib

opencv_face342d.lib

opencv_features2d342d.lib

opencv_flann342d.lib

opencv_fuzzy342d.lib

opencv_hdf342d.lib

opencv_highgui342d.lib

opencv_imgcodecs342d.lib

opencv_imgproc342d.lib

opencv_line_descriptor342d.lib

opencv_ml342d.lib

opencv_objdetect342d.lib

opencv_optflow342d.lib

opencv_photo342d.lib

opencv_plot342d.lib

opencv_reg342d.lib

opencv_rgbd342d.lib

opencv_saliency342d.lib

opencv_shape342d.lib

opencv_stereo342d.lib

opencv_stitching342d.lib
opencv_structured_light342d.lib
opencv_superres342d.lib
opencv_surface_matching342d.lib
opencv_text342d.lib
opencv_tracking342d.lib
opencv_ts342d.lib
opencv_video342d.lib
opencv_videoio342d.lib
opencv_videostab342d.lib
opencv_xfeatures2d342d.lib
opencv_ximgproc342d.lib
opencv_xobjdetect342d.lib
opencv_xphoto342d.lib
ippicvmt.lib

Release 模式:

名称都相同，只是把后面 d 全部去掉。

因为我用的是 3.4.2 版本，所以 lib 文件名称后缀为 342，如果你是其他版本要改成相应的版本，如 3.4.0 就改成 340。添加完后，如果编译的时候出现某 lib 文件无法打开，很有可能是前面编译的时候跳过了，可以重新编译，或者暂时不用这个 lib 文件里面的功能的话，可以在依赖项中先去掉。

然后保存属性表，以后建立新项目的时候直接加载属性表就可以了。

3 测试程序

3.1 生成测试程序

新建一个程序，测试 contrib 里面的 tracking 模块。新建 win32 控制台程序，输入如下代码

```
// contrib_test.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"
#include <opencv2\opencv.hpp>
#include <opencv2\tracking.hpp>

using namespace cv;

int main()
{
    cv::VideoCapture capture;

    capture.open("G:\\1.avi");

    //创建tracker
    Ptr<Tracker> tracker =TrackerMIL::create();

    //ROI选取模版
    Rect2d roi;
    Mat frame;

    capture >> frame;
    roi = selectROI("tracker", frame);

    //如果没有选择ROI则跳出
    if (roi.width == 0 || roi.height == 0)
        return 0;

    // 初始化tracker
    tracker->init(frame, roi);
    int cnt = 0;
    while (true) {

        capture >> frame;

        // 读取失败跳出程序
        if (frame.empty())break;

        // update
        tracker->update(frame, roi);
        // 描画结果
        rectangle(frame, roi, Scalar(255, 0, 0), 2, 1);
        // 显示
        imshow("frame", frame);
        cnt++;
    }
}
```

```
// 按 ESC 键退出
if (waitKey(1) == 27)break;

}

return 0;
}
```

杭州电子科技大学
机器视觉技术与应用
课程资料

参考文献

1. 解决 opencv310 ffmpeg、ippicv_windows_20151201.zip 无法下载问题:

https://blog.csdn.net/Kelvin_Yan/article/details/53635641

杭州电子科技大学
机器视觉技术与应用
课程资料

附录

杭州电子科技大学 机器视觉技术与应用 课程资料

版本

时间	版本	作者	备注
2018/08/03	1.0	李竹	初版完成

杭州电子科技大学
机器视觉技术与应用
课程资料