Windows10环境中CMake+VS2017编译OpenCV

1. 概述

目前OpenCV官方发布的最新版本是4.1.0，针对于C++开发者而言官方发布的版本只有x64的相应库文件及动态链接库，那么如果需要开发x86的软件，就需要自己进行编译x86的库文件及动态链接库。此外官方版本中的相应函数只是稳定的可靠的函数，那么有些测试中的函数并没有包括在其中，如果需要使用这些测试中的函数要么需要自己写要么需要将opencv 的extra模块中的内容编译进去，从而调用这些测试中的函数。

综上所述，对OpenCV进行编译是十分必要的，我们一般在Windows平台下借助于CMake及VS进行编译，下面做简单介绍。

1. 准备工作
2. 安装CMake

本人在Win10 x64平台下进行编译，此处采用的CMake是64位的版本，版本号是3.11.0。下载地址是<https://github.com/Kitware/CMake/releases/tag/v3.11.0>

图片包含 屏幕截图

描述已自动生成

下载之后安装到合适的目录中。

1. 安装Visual Studio

本人此处采用的是Visual Studio 2017 Community。注意的是，本人安装之后如下图所示出现一个Microsoft Visual Studio文件夹，一个Microsoft Visual Studio 14.0文件夹，在很多的教程中CMake的Configure中选择Visual Studio 15 2017或者Visual Studio 15 2017 Win64，但是本人碰到的情况是如果这样操作就识别不出VS，只能选择Visual Studio 14 2015或者Visual Studio 14 2015 Win64，也许是文件夹Microsoft Visual Studio 14.0的原因，因此此处建议后面选择的时候根据此处的具体情况进行选择。

图片包含 屏幕截图

描述已自动生成

图片包含 屏幕截图

描述已自动生成

1. 下载opencv及opencv\_contrib项目的源代码

Opencv的下载链接是

<https://github.com/opencv/opencv/tree/4.1.0>

Opencv\_contrib的下载链接是<https://github.com/opencv/opencv_contrib/tree/4.1.0>

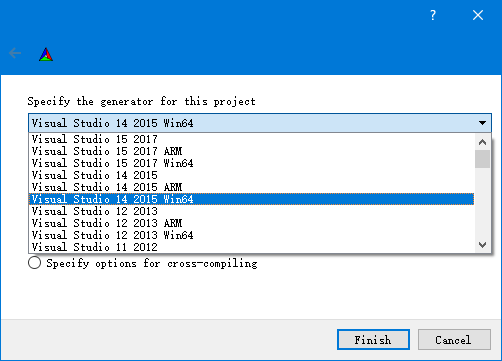
注意此处opencv的版本要和opencv\_extra的版本一致。

1. 操作步骤
2. 将下载的opencv和opencv\_extra源代码解压到非中文路径下。
3. 打开CMake。
4. 在CMake界面中导入opencv源代码所在路径及opencv编译后所在路径，如下图所示：

图片包含 屏幕截图

描述已自动生成

1. 点击Configure按钮，在弹出的界面中选择编译后的opencv库文件及动态链接库的版本，x86还是x64，如果要编译x86版本则选择Visual Studio 14 2015，如果要编译x64版本则选择Visual Studio 14 2015 Win64，如下图所示：



1. 点击Finish，处理过程如下所示：

图片包含 屏幕截图

描述已自动生成

1. 上一步处理之后的结果如下图所示：

图片包含 屏幕截图

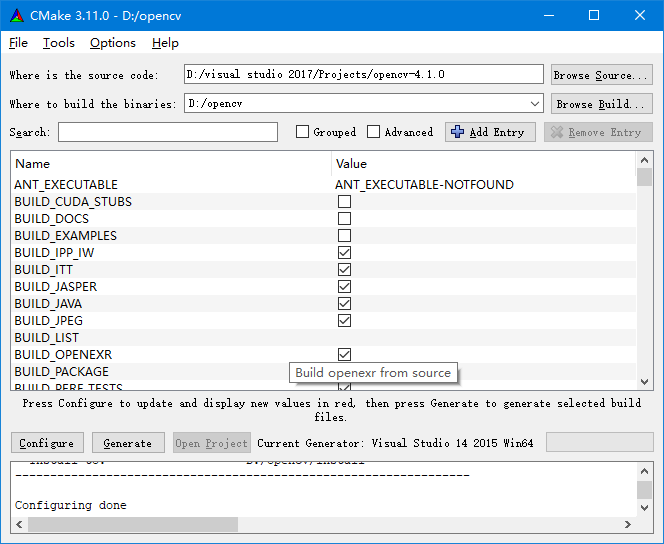
描述已自动生成

1. 再次点击Configure按钮，处理过程如下所示：

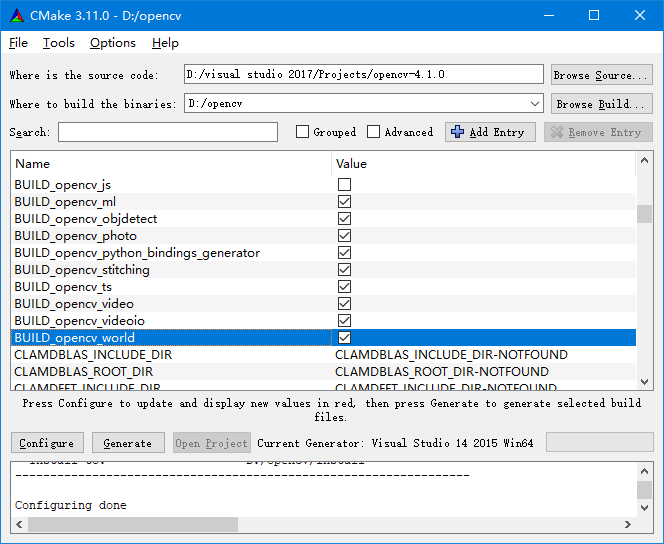
图片包含 屏幕截图

描述已自动生成

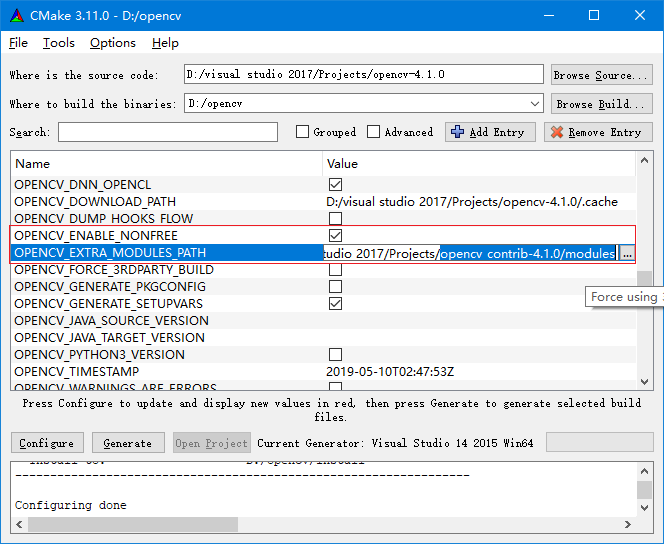
1. 上一步处理之后的结果如下图所示：



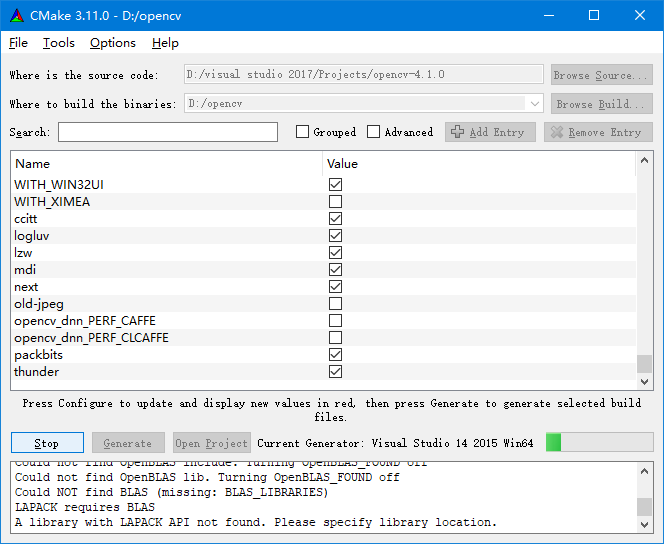
1. 将BUILD\_opencv\_world打勾，打勾之后生成的文件中所有的库文件及动态链接库都包括在一个标记为opencv\_world410d.lib和opencv\_world410d.dll文件中，上述是debug版本，如果是release版本则是opencv\_world410.lib和opencv\_world410.dll。如下图所示：



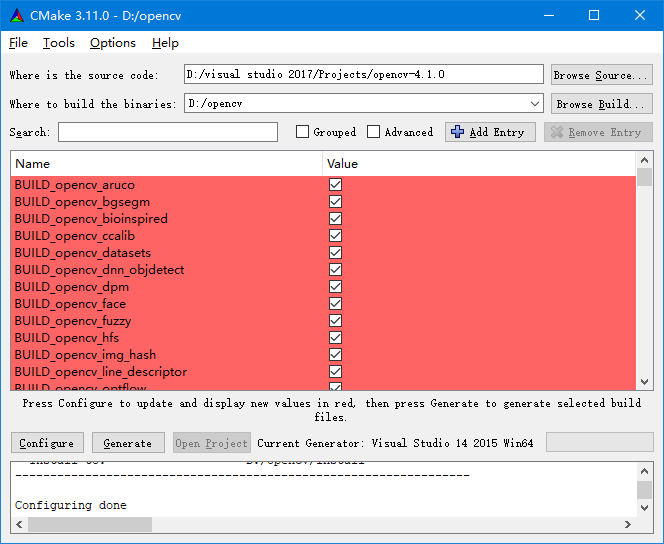
1. 将OPENCV\_ENABLE\_NONFREE打勾，打勾后就可以将一些非免费的算法编译进去，下面的OPENCV\_EXTRA\_MODULES\_PATH选择opencv\_contrib源代码中的modules文件夹所在的路径，如下图所示：



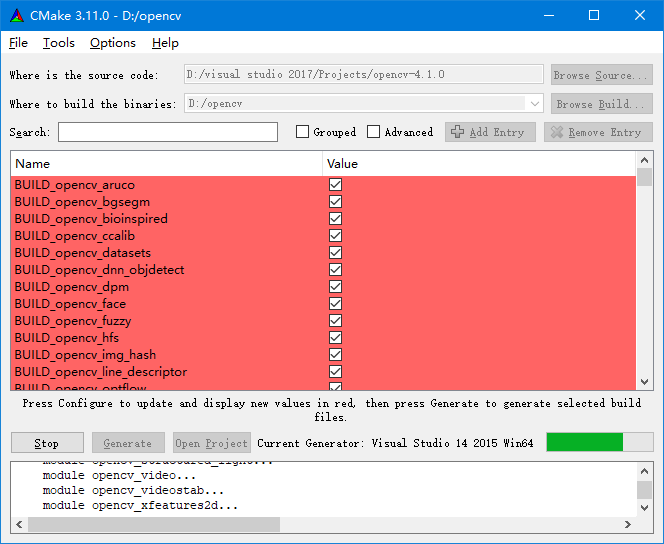
1. 上述步骤操作完成之后点击Configure按钮，处理过程如下图所示：



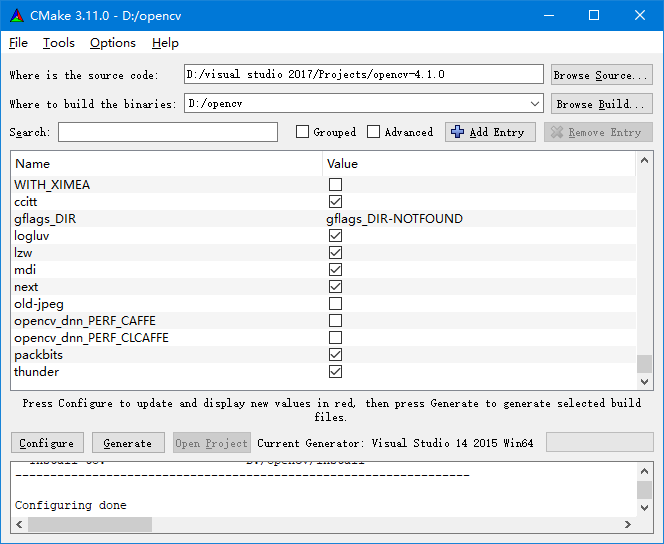
1. 处理之后的结果如下图所示：



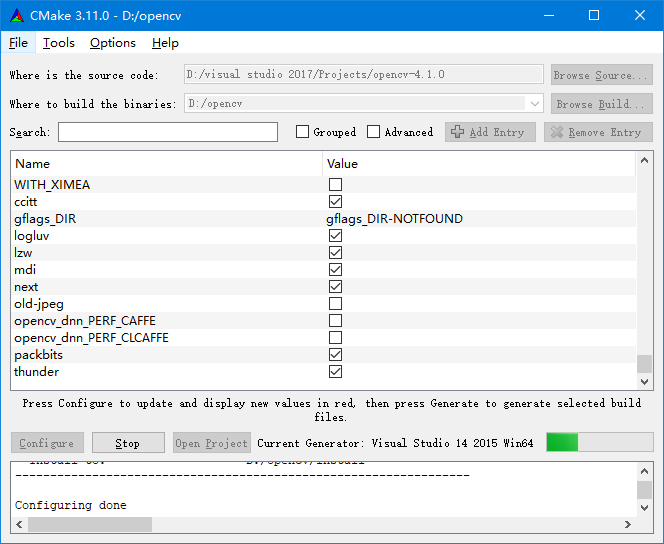
1. 上一步的操作完成之后，点击Configure，处理过程如下图所示：



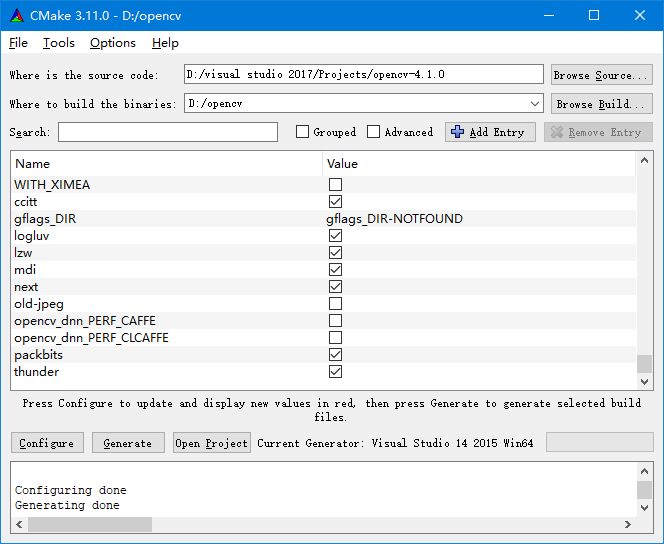
1. 处理完成之后的结果如下图所示：



1. 点击Generate按钮，生成解决方案，处理进度图如下所示：



1. 处理完毕如下图所示：



1. 很多教程在此步骤会直接点击Open Project或者通过VS打开项目进行编译，事实上这里有更便捷的办法，并且若生成Python库必须这样做（本人此步操作失败，依然没有生成Python库）。（也可以直接跳到18）进行操作，依然可以生成所需要的文件）

在生成项目的文件夹（如opencv）空白处，按住shift的同时点击鼠标右键，点击“在此处打开powershell窗口”，键入如下命令：

Release模式：

cmake.exe --build . --config Release --target INSTALL

Debug模式： debug模式需要去掉 INSTALL\_PYTHON\_EXAMPLES、BUILD\_opencv\_python3 和BUILD\_opencv\_python2 这三项的勾选，经测试若勾选会编译失败。

cmake.exe --build . --config Debug --target INSTALL

等待编译完成即可。

1. 点击Open Project按钮或者到opencv文件夹中找到opencv.sln并打开，于是就打开了生成的解决方案。（如果依照17）的方法进行操作下面的步骤可以不进行操作）
2. 通过切换Debug和Release分别编译成Debug版本或者Release版本，后面的x64不能切换，需要在CMake中的configure中重新生成，如下图所示：

图片包含 屏幕截图

描述已自动生成

1. 当打开了生成的opencv项目，可以选择想要生成的库文件的Debug版本或Release版本

默认是Debug版本

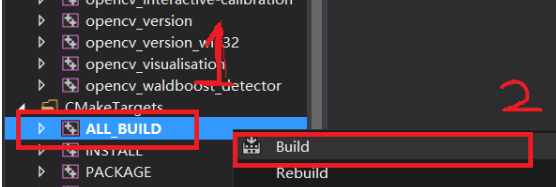
第一步：选择CMakeTargets中的 ALL\_BUILD，右键 ->生成，

第二步：生成结束如果没有失败的，再选择INSTALL→右键→仅当前项目→只生成INSTALL。

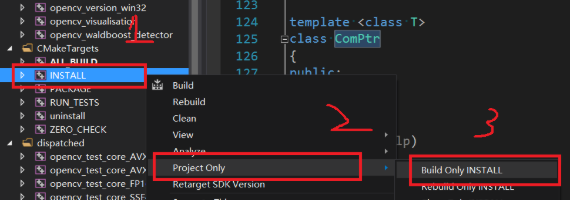
如果需要release版本，选择release版后重复第一步和第二步。

如下图所示：

第一步示意图：



第二步示意图



1. 当在VS2017完成了生成INSTALL，那么OpenCV的库就已经生成好了，在我们的opencv文件夹下的install文件夹中，包括了bin、etc、include、x64、x86文件夹和一些文件，这就和官网提供的OpenCV包里build文件夹下一样的结构了，只是没有java和python文件，将install拷贝到其他地方改个名,如OpenCV410就可以正常使用啦。

参考资料

<https://www.cnblogs.com/Brandon0807/p/10403872.html>

<https://www.cnblogs.com/lzhu/p/8198654.html>

<https://blog.csdn.net/Gordon_Wei/article/details/85775328>

<https://blog.csdn.net/howlclat/article/details/78783701>

<https://blog.csdn.net/LOVELESSYI/article/details/79946859>

<https://blog.csdn.net/poem_qianmo/article/details/21974023>

<https://blog.csdn.net/qq_36088602/article/details/83754403>

<https://blog.csdn.net/qq385105501/article/details/73770826>

<https://blog.csdn.net/suswulongyuan/article/details/86544180>

<https://blog.csdn.net/u012900686/article/details/69442406?locationNum=13&fps=1>