PCL完全编译手册

由于使用PCL时采用CPU执行模式处理点云数据非常缓慢，所以想编译一个支持CUDA的PCL版本。经过好多天的痛苦煎熬，终于出来了一个结果。虽然不是太满意，但是记录下来，以待以后改进。

安装过PCL的AllInOne版本就会知道，PCL依赖于众多的第三方库，当我们自己编译PCL时会发现，它不仅仅依赖于这些库，而且其他好多库都隐含在背后，没有显露出来，当我们使用CMake的时候就会一一呈现，我没有安装好所有的库，只是安装了必要的库。

下面我将整个操作流程记录一下，不太详细，但是大体的关键步骤和细节记录下来了。

# 安装环境

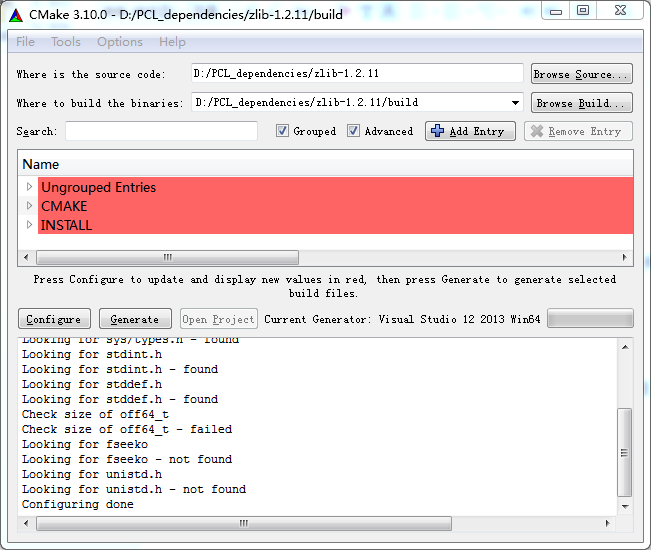
1. 安装Visual Studio 2013 ，没有特殊处理，一步一步安装即可。
2. 安装Qt vs2013 64位，没有特殊处理，一步一步安装即可。
3. 安装CUDA 8.0 for win10，我用的带固态硬盘的win10，显卡是GTX 1060。
4. 安装MPI。
5. 安装CMake。
6. 安装OpenNI2安装包。

# 开始编译

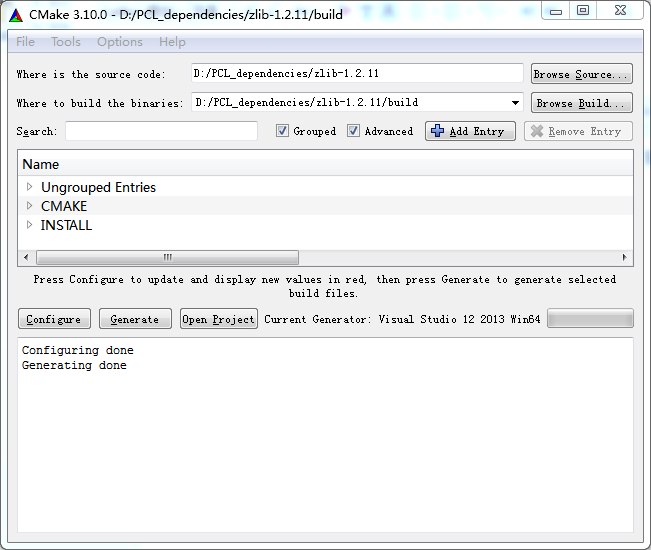
希望按照我的顺序来进行，不按照顺序也没关系，反正迟早还得有先后依赖的。

## Zlib

CMake配置和编译，选中目录，创建build目录，然后开始配置。如下所示：



别管它，再点击一下配置就好了，然后点击生成。

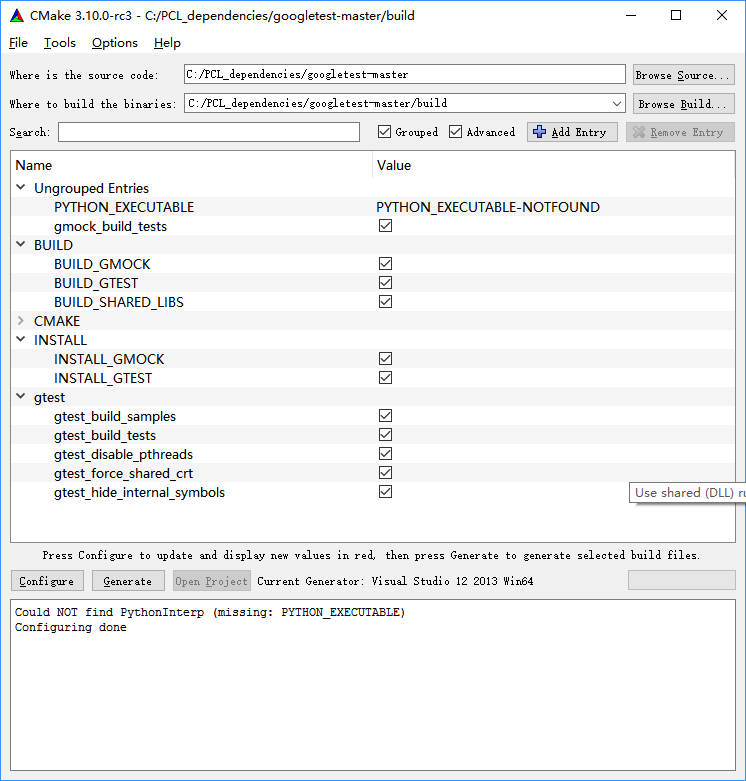


生成完毕后，用VS2013的管理员模式打开，编译INSTALL就会自动安装到指定位置。

Debug和Release都生成一下。

## Gtest

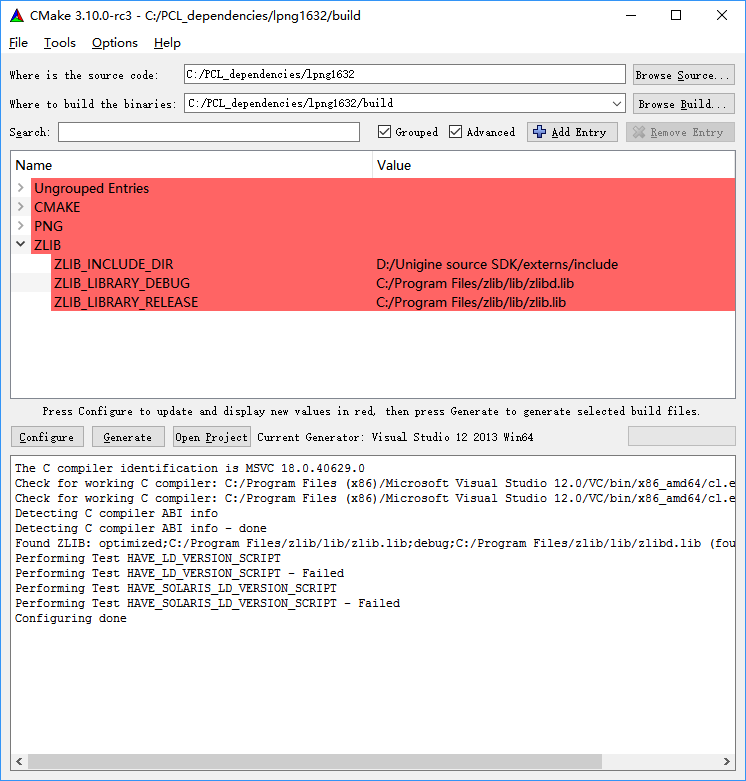
操作和zlib相同。唯一不同的在于能打的界面上打钩多一些，尽量多打一些，反正也无所谓。



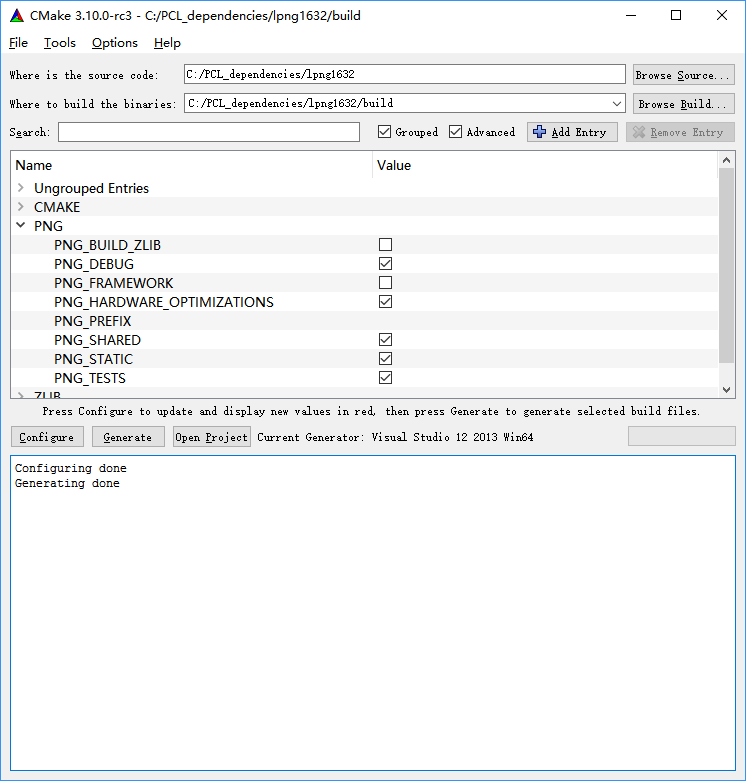
生成，编译安装。

## LibPng

和Zlib相同的步骤，注意事项为：LibPng用到zlib，所以要注意是否配置正确。



ZLIB\_INCLUDE\_DIR配置就不正确，修改为：C:/Program Files/zlib/include，然后PNG部分再多选几项。



然后生成，编译同上。

## Glew

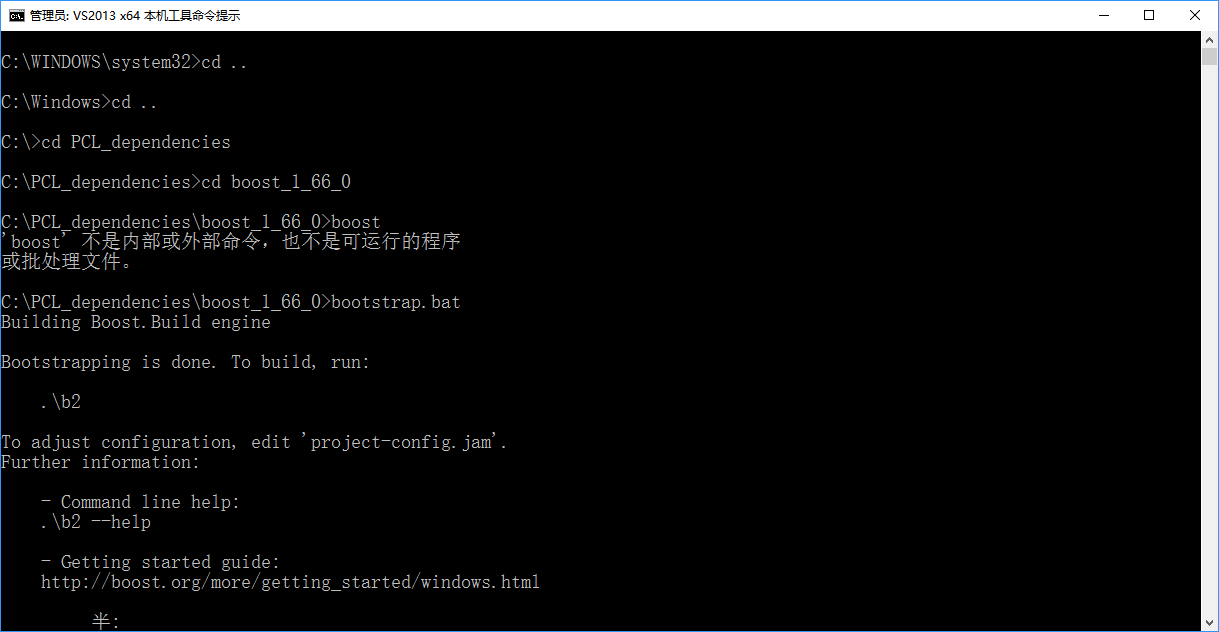
直接解压到C:/Program Files/目录下。

## FreeGlut

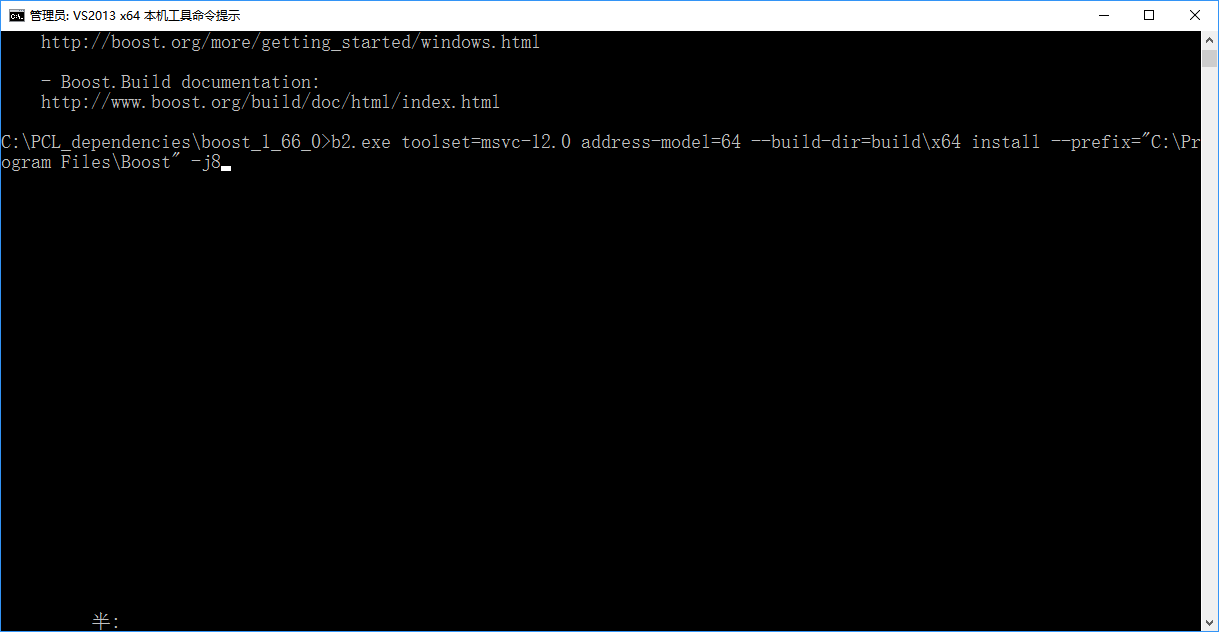
操作如Zlib，注意事项：不用管Debug模式了，直接Release安装。

## Boost

管理员方式打开VS2013 x64工具软件，执行bootstrap.bat

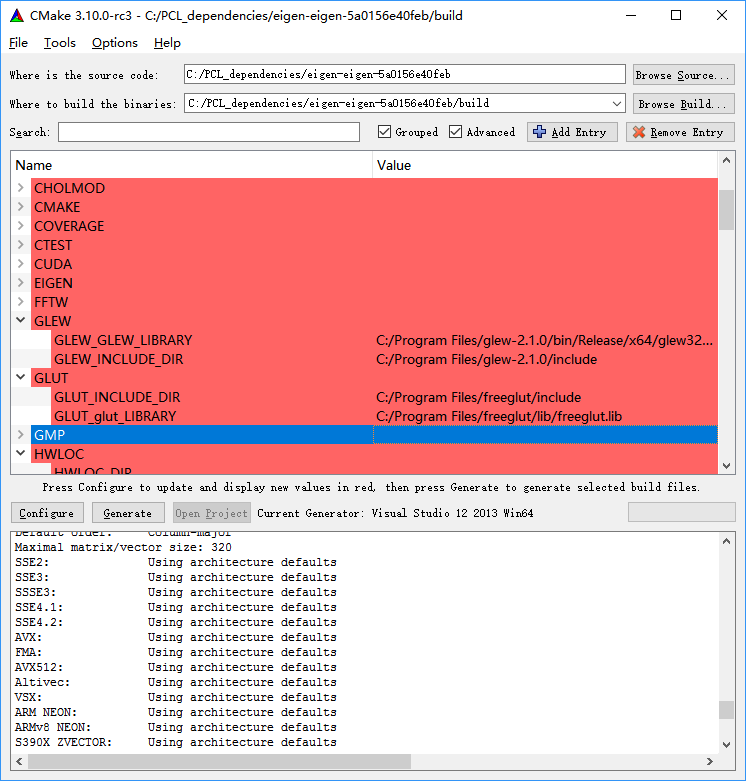


然后执行如下命令：执行64位编译b2.exe toolset=msvc-12.0 address-model=64 --build-dir=build\x64 install --prefix="C:\Program Files\Boost" -j8 --buildtype=complete或者执行32位编译b2.exe toolset=msvc-12.0 address-model=32 --build-dir=build\x86 install --prefix="C:\Program Files (x86)\Boost" -j8 --buildtype=complete



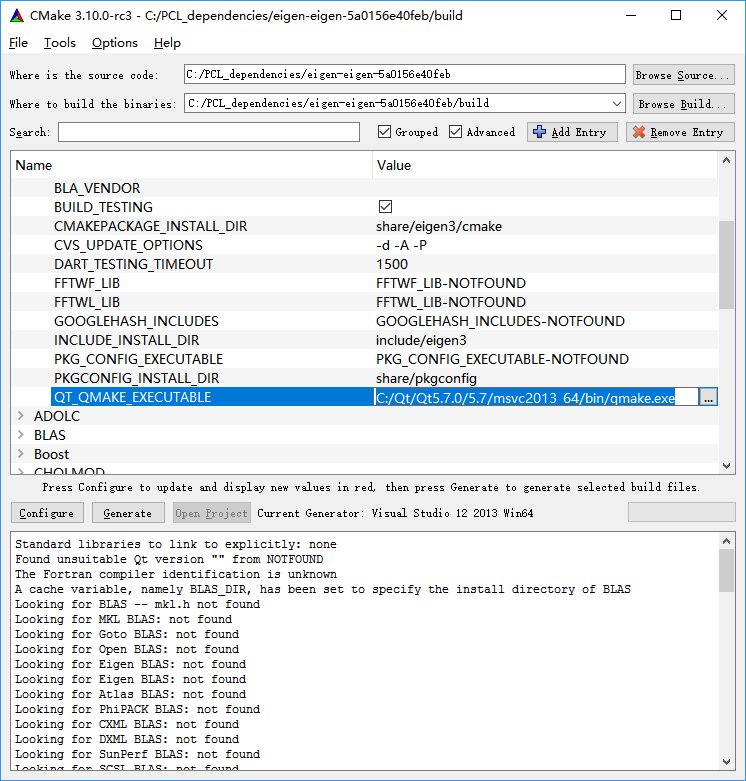
## Eigen

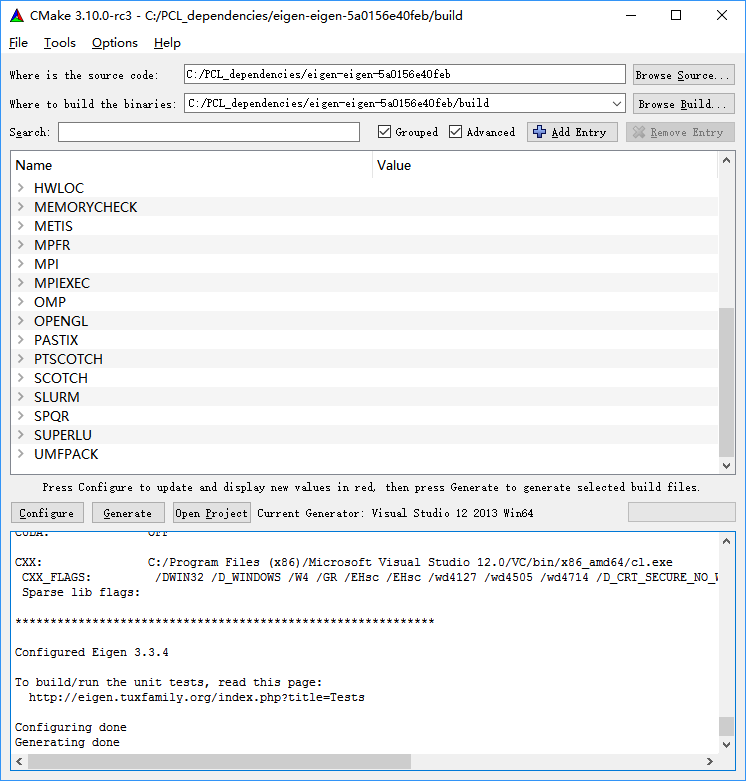
操作如Zlib，不同之处在于需要手动配置以下项目：



配置好Glew和Glut，然后配置好Qmake.exe，然后点击配置

。





## Qhull

直接配置和编译，没有任何问题。

## flann

编译的时候，在serialization.h文件的93行添加以下内容：

#ifdef \_MSC\_VER

BASIC\_TYPE\_SERIALIZER(unsigned \_\_int64);

#endif

## Vtk

默认配置，生成和编译。

## Pcl

配置界面如下，编译时间较长，占用cpu较大，建议晚上下班编译，然后第二天早上起来搞定。

