

## Makefile\_VS\_CMake

最近想写一段关于 BA 的程序，要用到 G2O 库。之前看高博的 SLAM 十四讲都是用的 CMakeLists，自己当时也是用的 CMakeLists，但后来在公司实习一直用的 Makefile，感觉后者更好用一些，就想着用 Makefile 编译包含 G2O 的程序。

原来的 CMakeLists.txt 如下：

```
cmake_minimum_required( VERSION 2.8 )  
project( directMethod )  
set( CMAKE_BUILD_TYPE Release )  
set( CMAKE_CXX_FLAGS "-std=c++11 -O3" )  
list( APPEND CMAKE_MODULE_PATH ${PROJECT_SOURCE_DIR}/cmake_modules )  
find_package( OpenCV )  
include_directories( ${OpenCV_INCLUDE_DIRS} )  
find_package( G2O )  
include_directories( ${G2O_INCLUDE_DIRS} )  
include_directories( "/usr/include/eigen3" )  
set( G2O_LIBS    g2o_core g2o_types_sba g2o_solver_csparse g2o_stuff g2o_csparse_extension )  
add_executable( direct_sparse direct_sparse.cpp )
```

```
target_link_libraries( direct_sparse ${OpenCV_LIBS} ${G2O_LIBS} )
```

最终生效可以编译的 makefile 文件如下:

```
INCLUDES = $(shell pkg-config --cflags opencv)
```

```
INCLUDES_EIGEN3 = -I /usr/include/eigen3
```

```
INLCUDES_G2O = -I /usr/local/include/g2o
```

```
LIBDIRS = $(shell pkg-config --libs opencv)
```

```
G2O_LIBS = /usr/local/lib/libg2o_core.so /usr/local/lib/libg2o_types_sba.so \  
           /usr/local/lib/libg2o_solver_csparse.so /usr/local/lib/libg2o_csparse_extension.so \  
           /usr/local/lib/libg2o_stuff.so
```

```
all:Feature_Direct_VO
```

```
Feature_Direct_VO:Feature_Direct_VO.o
```

```
g++ -std=c++11 -o Feature_Direct_VO Feature_Direct_VO.o $(G2O_LIBS) $(LIBDIRS)
```

```
Feature_Direct_VO.o:Feature_Direct_VO.cpp
```

```
g++ -std=c++11 -c Feature_Direct_VO.cpp $(INCLUDES) $(INCLUDES_EIGEN3) $(INCLUDE_G2O)
```

```
clean:
```

```
rm -f *.o Feature_Direct_VO
```

其实原理挺简单，当用到一个第三方软件函数时，既要包含头文件，也要包含库文件。



		<a href="#">/usr/lib/libvtkso.5.8.0</a>
	头文件： <a href="#">find_package( PCL REQUIRED COMPONENT common io )</a> <a href="#">include_directories( \${PCL_INCLUDE_DIRS} )</a> <a href="#">add_definitions( \${PCL_DEFINITIONS} )</a>	头文件： <a href="#">INCLUDE_PCL = -I /usr/include/pcl-1.7</a> <a href="#">INCLUDE_OPENNI = -I /usr/include/ni</a> <a href="#">INCLUDE_VTK = -I /usr/include/vtk-5.8</a>

注意：

1. g2o 的库文件不止上面的一个，要根据具体需要进行 include，所有的库均在 /usr/local/include/ 目录下；
2. eigen3 是一个纯用头文件搭起来的库，所以只需要包含头文件（高博的 SLAM 十四讲）；
3. 再后来帮别人在 windows 下用 eigen 时，看到一篇博文，写到，“eigen 官网下载的是个 code 包，无需安装”，也就是说，我们只需要下载这个包就行，不用去安装它，使用它时，include 它的头文件即可

## 2018/1/7 PCL 点云库

后来又用到 PCL 点云库，其依赖项比较多，在编写其 Makefile 文件时一直出现库链接不到的问题，网上也没有通过 Makefile 编译 PCL 程序的，弄了半天也没弄好。后来我想，既然能用 cmake 编译成功就一定能用 Makefile 编译成功啊，因为 cmake 就是用来产生 Makefile 的，那 cmake 的中间文件里是不是也有 Makefile 的相关依赖呢？于是我就翻看 cmake 的一些中间文件，果然让我找到了。

在执行“cmake ..”后，产生了如下的中间文件目录：

名称	修改日期	类型	大小
CMakeFiles	2018/1/7 11:04	文件夹	
cmake_install.cmake	2017/10/8 16:40	CMAKE 文件	2 KB
CMakeCache	2017/10/8 16:40	TXT 文件	23 KB
joinMap	2017/10/8 16:43	文件	89 KB
Makefile	2017/10/8 16:40	文件	5 KB

一番寻找之后，重要找到了我们需要的东西，分别是：

`\build\CMakeFiles\joinMap.dir\flags.cmake` #存放所要包含的头文件路径，我们在执行 `g++ -c` 时，把这些头文件路径包含进去

```
1 # CMAKE generated file: DO NOT EDIT!
2 # Generated by "Unix Makefiles" Generator, CMake Version 2.8
3
4 # compile CXX with /usr/bin/c++
5 CXX_FLAGS = -std=c++11 -O3 -Wno-deprecated -O3 -DNDEBUG -I/usr/include/opencv -I/usr/include/eigen3 -I/usr/include/vtk-5.8 -I/usr/include/pcl-1.7 -I/usr/include/ni
6
7 CXX_DEFINES = -DEIGEN_USE_NEW_STDVECTOR -DEIGEN_YES_I_KNOW_SPARSE_MODULE_IS_NOT_STABLE_YET
8
9
```

`\build\CMakeFiles\joinMap.dir\link.txt` #存放所连接的库，我们将这些分类整理一下，在执行 `g++ -o` 时，将这些库文件链接进去

```

1 /usr/bin/c++ -std=c++11 -O3 -Wno-deprecated -O3 -DNDEBUG CMakeFiles/joinMap.dir/joinMap.cpp.o -o joinMap -rdynamic
2 /usr/lib/x86_64-linux-gnu/libopencv_videostab.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_video.so.2.4.8
3 /usr/lib/x86_64-linux-gnu/libopencv_ts.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_superres.so.2.4.8
4 /usr/lib/x86_64-linux-gnu/libopencv_stitching.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_photo.so.2.4.8
5 /usr/lib/x86_64-linux-gnu/libopencv_ocl.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_objdetect.so.2.4.8
6 /usr/lib/x86_64-linux-gnu/libopencv_ml.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_legacy.so.2.4.8
7 /usr/lib/x86_64-linux-gnu/libopencv_imgproc.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_highgui.so.2.4.8
8 /usr/lib/x86_64-linux-gnu/libopencv_gpu.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_flann.so.2.4.8
9 /usr/lib/x86_64-linux-gnu/libopencv_features2d.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_core.so.2.4.8
10 /usr/lib/x86_64-linux-gnu/libopencv_contrib.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_calib3d.so.2.4.8
11 -lboost_system -lboost_filesystem -lboost_thread -lboost_date_time -lboost_iostreams -lboost_serialization
12 -lpthread -lpcl_common -lpcl_octree -lOpenNI /usr/lib/libvtkCommon.so.5.8.0
13 /usr/lib/libvtkRendering.so.5.8.0 /usr/lib/libvtkHybrid.so.5.8.0 /usr/lib/libvtkCharts.so.5.8.0
14 -lpcl_io -lboost_system -lboost_filesystem -lboost_thread -lboost_date_time -lboost_iostreams -lboost_serialization
15 -lpthread -lOpenNI /usr/lib/libvtkCommon.so.5.8.0 /usr/lib/libvtkRendering.so.5.8.0
16 /usr/lib/libvtkHybrid.so.5.8.0 /usr/lib/libvtkCharts.so.5.8.0
17 /usr/lib/x86_64-linux-gnu/libopencv_photo.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_legacy.so.2.4.8
18 /usr/lib/x86_64-linux-gnu/libopencv_video.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_objdetect.so.2.4.8
19 /usr/lib/x86_64-linux-gnu/libopencv_ml.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_calib3d.so.2.4.8
20 /usr/lib/x86_64-linux-gnu/libopencv_features2d.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_highgui.so.2.4.8
21 /usr/lib/x86_64-linux-gnu/libopencv_imgproc.so.2.4.8 /usr/lib/x86_64-linux-gnu/libopencv_flann.so.2.4.8
22 /usr/lib/x86_64-linux-gnu/libopencv_core.so.2.4.8 -lpcl_common -lpcl_octree -lpcl_io /usr/lib/libvtkViews.so.5.8.0
23 /usr/lib/libvtkInfovis.so.5.8.0 /usr/lib/libvtkWidgets.so.5.8.0 /usr/lib/libvtkHybrid.so.5.8.0
24 /usr/lib/libvtkParallel.so.5.8.0 /usr/lib/libvtkVolumeRendering.so.5.8.0 /usr/lib/libvtkRendering.so.5.8.0
25 /usr/lib/libvtkGraphics.so.5.8.0 /usr/lib/libvtkImaging.so.5.8.0 /usr/lib/libvtkIO.so.5.8.0
26 /usr/lib/libvtkFiltering.so.5.8.0 /usr/lib/libvtkCommon.so.5.8.0 -lm /usr/lib/libvtkvga.so.5.8.0 -ldl
27

```

\build\CMakeFiles\joinMap.dir\

#所有要包含的头文件（其实写 Makefile 用上面两个就行了），这里面包含了所包含的所有的头文件，编写 Makefile 时没怎么参考

这是我最终编写并执行正确的 Makefile 文件：

```

1 INCLUDE_OPENCV = $(shell pkg-config --cflags opencv)
2 LIBDIR_OPENCV = $(shell pkg-config --libs opencv)
3
4 INCLUDE_EIGEN = -I /usr/include/eigen3
5
6 INCLUDE_PCL = -I /usr/include/pcl-1.7
7 LIBDIR_PCL = -lpcl_common -lpcl_octree -lpcl_io -lpcl_common -lpcl_octree
8
9 LIBDIR_BOOST = -lboost_thread -lboost_date_time -lboost_iostreams -lboost_serialization -lboost_system -lboost_filesystem
10
11 INCLUDE_OPENNI = -I /usr/include/ni
12 LIBDIR_OPENNI = -lOpenNI
13
14 INCLUDE_VTK = -I/usr/include/vtk-5.8
15 LIBDIR_VTK = /usr/lib/libvtkViews.so.5.8.0 /usr/lib/libvtkInfovis.so.5.8.0 \
16             /usr/lib/libvtkWidgets.so.5.8.0 /usr/lib/libvtkHybrid.so.5.8.0 \
17             /usr/lib/libvtkParallel.so.5.8.0 /usr/lib/libvtkVolumeRendering.so.5.8.0 \
18             /usr/lib/libvtkRendering.so.5.8.0 /usr/lib/libvtkGraphics.so.5.8.0 \
19             /usr/lib/libvtkImaging.so.5.8.0 /usr/lib/libvtkIO.so.5.8.0 \
20             /usr/lib/libvtkFiltering.so.5.8.0 /usr/lib/libvtkCommon.so.5.8.0 \
21             /usr/lib/libvtksys.so.5.8.0
22
23 INCLUDE_G2O = -I /usr/local/include/g2o
24 LIBDIR_G2O = /usr/local/lib/libg2o_core.so /usr/local/lib/libg2o_types_sba.so \
25             /usr/local/lib/libg2o_solver_csparse.so /usr/local/lib/libg2o_csparse_extension.so \
26             /usr/local/lib/libg2o_stuff.so
27
28 all:extract_plane
29
30 extract_plane:extract_plane.o
31     g++ -std=c++11 -Wno-deprecated -o extract_plane extract_plane.o $(LIBDIR_PCL) $(LIBDIR_OPENCV) $(LIBDIR_BOOST) $(LIBDIR_OPENNI) $(LIBDIR_VTK) $(LIBDIR_G2O)
32 extract_plane.o:extract_plane.cpp
33     g++ -std=c++11 -Wno-deprecated -c extract_plane.cpp $(INCLUDE_OPENCV) $(INCLUDE_EIGEN) $(INCLUDE_PCL) $(INCLUDE_VTK) $(INCLUDE_OPENNI) $(INCLUDE_G2O)
34 clean:
35     rm -f *.o extract_plane

```

如此一来，如果我们实在找不到用于编写 Makefile 时的依赖文件，就可以找寻 cmake 中的中间文件，借助这些文件来编写自己的 Makefile 文件。