PCL交叉编译文档

> 引言：本文档是主要针对实现分割算法及其依赖库进行交叉编译，未涵盖所有模块的编译。

编译环境：

1. 编译平台：Ubuntu18.04
2. 目标平台：瑞芯微rk3399
3. 编译工具：交叉工具链 rockchip-cc-tools
4. 编译软件：cmake-gui
5. 源代码：
    - **pcl-1.9.1**
    - **boost**
    - **flann**
    - lz4

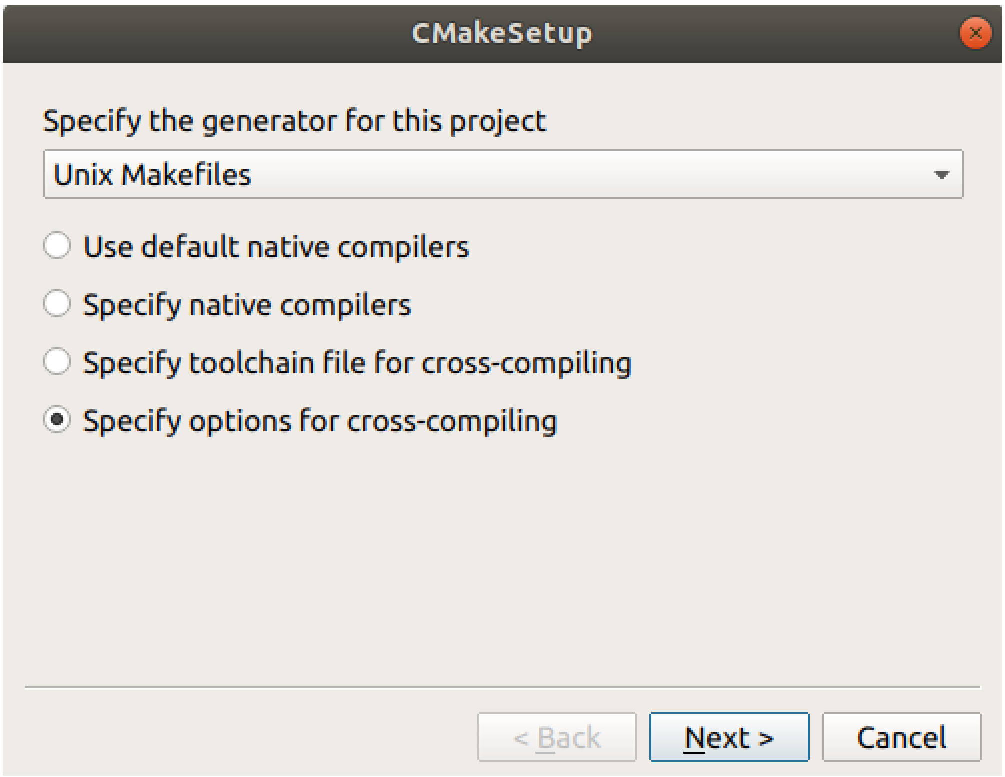
      说明：pcl依赖boost、flann、eigen   flann依赖lz4

编译步骤：

1. **按照一下形式构建目录**

    ```
    ├──── extern_libs
    │     ├──── boost_1_66_0
    │     ├──── boot_install
    │     ├──── flann
    │     ├──── flann_install
    │     ├──── lz4
    │     ├──── lz4_install
    ├──── install_pcl
    └──── pcl-pcl-1.9.1
    ```

2. **先编译PCL**
    - cd pcl-pcl-1.9.1&&mkdir build
    - configure弹出以下窗口


CMakeSetup

Specify the generator for this project

Unix Makefiles

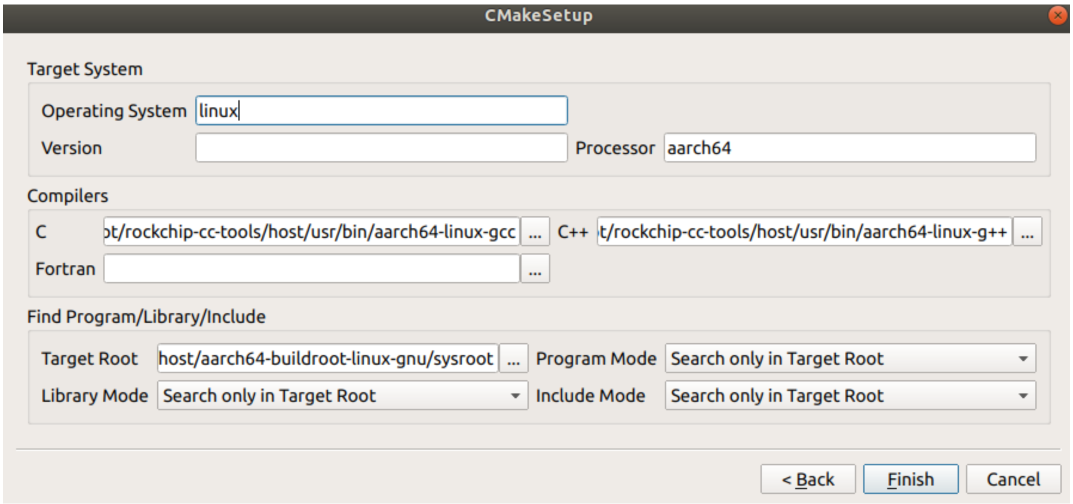
○ Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

● Specify options for cross-compiling

- 配置工具链信息


CMakeSetup

Target System

Operating System: linux

Version:

Processor: aarch64

Compilers

C: ɔt/rockchip-cc-tools/host/usr/bin/aarch64-linux-gcc

C++: t/rockchip-cc-tools/host/usr/bin/aarch64-linux-g++

Fortran:

Find Program/Library/Include

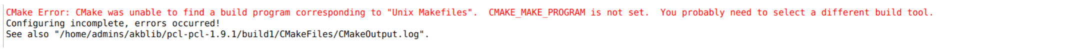
Target Root: host/aarch64-buildroot-linux-gnu/sysroot

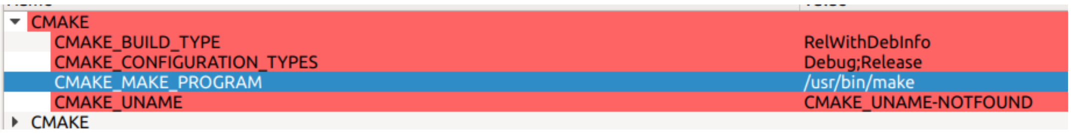
Program Mode: Search only in Target Root

Library Mode: Search only in Target Root

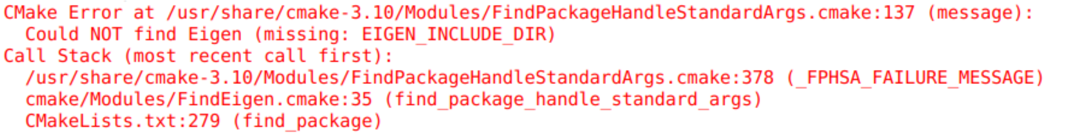
Include Mode: Search only in Target Root

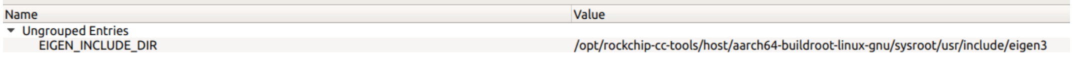
- 缺少CMAKE_MAKE_PROGRAM的值 Search搜索 填加值/usr/bin/make


```
CMake Error: CMake was unable to find a build program corresponding to "Unix Makefiles".  CMAKE_MAKE_PROGRAM is not set.  You probably need to select a different build tool.
Configuring incomplete, errors occurred!
See also "/home/admins/akblib/pcl-pcl-1.9.1/build1/CMakeFiles/CMakeOutput.log".
```


```
▾ CMAKE
    CMAKE_BUILD_TYPE                    RelWithDebInfo
    CMAKE_CONFIGURATION_TYPES           Debug;Release
    CMAKE_MAKE_PROGRAM                  /usr/bin/make
    CMAKE_UNAME                         CMAKE_UNAME-NOTFOUND
▸ CMAKE
```

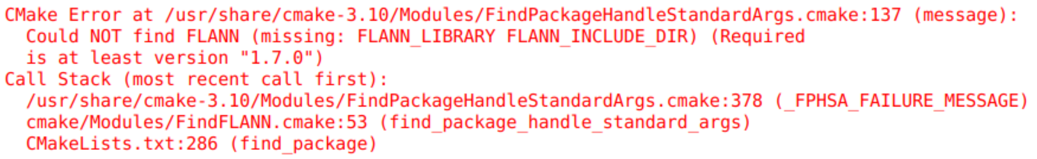
- 缺少eigen库 交叉编译链工具有 就不用编译 直接链接到目录即可


```
CMake Error at /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:137 (message):
  Could NOT find Eigen (missing: EIGEN_INCLUDE_DIR)
Call Stack (most recent call first):
  /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:378 (_FPHSA_FAILURE_MESSAGE)
  cmake/Modules/FindEigen.cmake:35 (find_package_handle_standard_args)
  CMakeLists.txt:279 (find_package)
```



Name	Value
▾ Ungrouped Entries	
EIGEN_INCLUDE_DIR	/opt/rockchip-cc-tools/host/aarch64-buildroot-linux-gnu/sysroot/usr/include/eigen3

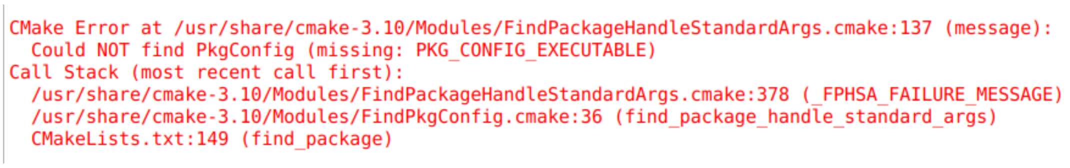
- 缺少flann 先编译flann再处理pcl



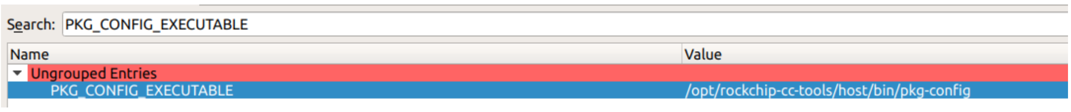
```
CMake Error at /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:137 (message):
  Could NOT find FLANN (missing: FLANN_LIBRARY FLANN_INCLUDE_DIR) (Required
  is at least version "1.7.0")
Call Stack (most recent call first):
  /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:378 (_FPHSA_FAILURE_MESSAGE)
  cmake/Modules/FindFLANN.cmake:53 (find_package_handle_standard_args)
  CMakeLists.txt:286 (find_package)
```

pcl

3. **编译flann**
   - 和前面一样 进入flann源码 mkdir build 配置工具链
   - missing: PKG_CONFIG_EXECUTABLE （再工具链里面找）



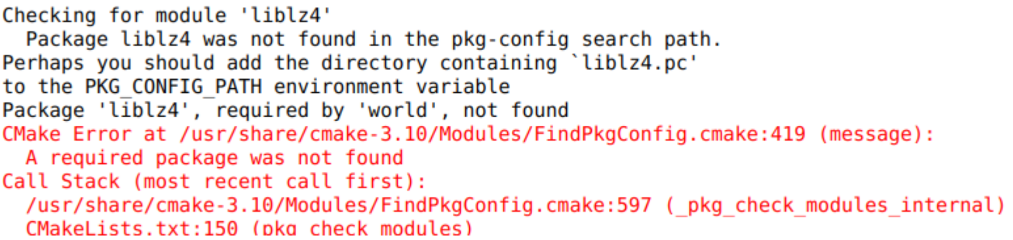
```
CMake Error at /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:137 (message):
  Could NOT find PkgConfig (missing: PKG_CONFIG_EXECUTABLE)
Call Stack (most recent call first):
  /usr/share/cmake-3.10/Modules/FindPackageHandleStandardArgs.cmake:378 (_FPHSA_FAILURE_MESSAGE)
  /usr/share/cmake-3.10/Modules/FindPkgConfig.cmake:36 (find_package_handle_standard_args)
  CMakeLists.txt:149 (find_package)
```



Search: PKG_CONFIG_EXECUTABLE

Name	Value
Ungrouped Entries	
PKG_CONFIG_EXECUTABLE	/opt/rockchip-cc-tools/host/bin/pkg-config

   - 如过缺少liblz4库 编译后放在工具链对应的include和lib文件夹里



```
Checking for module 'liblz4'
  Package liblz4 was not found in the pkg-config search path.
Perhaps you should add the directory containing `liblz4.pc'
to the PKG_CONFIG_PATH environment variable
Package 'liblz4', required by 'world', not found
CMake Error at /usr/share/cmake-3.10/Modules/FindPkgConfig.cmake:419 (message):
  A required package was not found
Call Stack (most recent call first):
  /usr/share/cmake-3.10/Modules/FindPkgConfig.cmake:597 (_pkg_check_modules_internal)
  CMakeLists.txt:150 (pkg_check_modules)
```

   - 把liblz4.pc /opt/rockchip-cc-tools/host/aarch64-buildroot-linux-gnu/sysroot/usr/lib/pkgconfig/liblz4.pc
   - 修改安装路径为源码同级的文件夹：flann_install
   - make -j8 && make install

4. **编译boost**
   - cd到boost 源目录
   - 编译配置，运行下面代码

     `./bootstrap.sh --without-libraries=atomic,chrono,context,coroutine,exception,graph,graph_parallel,mpi,python,wave --prefix=/home/admins/akblib/extern_libs/boot_install`

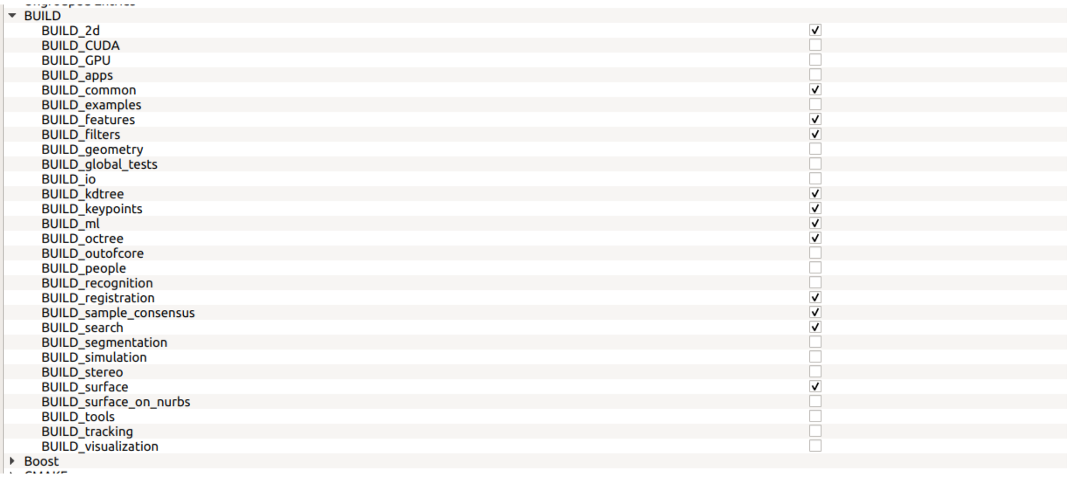
   - 更改生成的project-config.jam 文件，修改该文件的一行(指定自己的编译器)：

```
if ! gcc in [ feature.values <toolset> ]
{
   using gcc : : /opt/rockchip-cc-tools/host/usr/bin/aarch64-linux-gcc ;
}
```

   - 执行 ./bjam 即可，这是编译，创建的 lib 文件默认在 stage 文件夹
   - 编译完后，执行 ./bjam install 即进行安装，我的上面 --prefix=/.../ 指定的目录就是安装目录

5. **接着编译PCL**
   - 配置好上述boost和flann的路径 with里面 勾pcap
   - 关掉gl、vtk、cuda无关模块；

- build对象勾选如下图：



▾ BUILD

BUILD_2d	☑
BUILD_CUDA	□
BUILD_GPU	□
BUILD_apps	□
BUILD_common	☑
BUILD_examples	□
BUILD_features	☑
BUILD_filters	☑
BUILD_geometry	□
BUILD_global_tests	□
BUILD_io	□
BUILD_kdtree	☑
BUILD_keypoints	☑
BUILD_ml	☑
BUILD_octree	☑
BUILD_outofcore	□
BUILD_people	□
BUILD_recognition	□
BUILD_registration	☑
BUILD_sample_consensus	☑
BUILD_search	☑
BUILD_segmentation	□
BUILD_simulation	□
BUILD_stereo	□
BUILD_surface	☑
BUILD_surface_on_nurbs	□
BUILD_tools	□
BUILD_tracking	□
BUILD_visualization	□

▸ Boost

- CMAKE-BUILD-TYPE设置为release　减少耗时
- 修改安装路径为源码同级的文件夹：pcl_install
- make -j8 && make install

参考链接：

交叉编译流程

基于cmake的交叉编译工具链

boost库交叉编译