# 配置 Opencv 开发环境

作者: Youngjam

- 原文链接: https://ld246.com/article/1680922707813
- 来源网站:链滴
- 许可协议:署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

#### 作 者: yangjian 创建日期: 2023-02-23 22:32:50 版权说明: Copyright (c) 2023 yangjian. All rights reserved.

Tags: #opencv#

**References:** 

## windows 环境

### 通过 vcpkg 安装

● 安装

#### Opencv

• vcpkg install opencv4: x64-windwos

```
set(OpenCV_DIR "${VCPKG_INSTALLED_DIR}/x64-windows/share/opencv4")
find_package(OpenCV REQUIRED)
```

● 在 vscode 中通过 cmake 引用

```
# 该值需要设置为全局,在project之前设置
set(CMAKE_TOOLCHAIN_FILE E:/vcpkg/scripts/buildsystems/vcpkg.cmake)
set(OpenCV_DIR "${VCPKG_INSTALLED_DIR}/x64-windows/share/opencv4")
```

```
# 查找OpenCV包
find_package(OpenCV REQUIRED)
if(OpenCV_FOUND)
    message("OpenCV found!")
    # 加入头文件路径
    include_directories(${OpenCV_INCLUDE_DIRS})
    # 添加OpenCV链接库
    link_libraries(${OpenCV_LIBS})
endif()
```

## Linux

#### 源码编译安装

```
1. oepncv 的 github 主页下载 opencv 和 opencv_contrib 源码,本文编译版本为 4.6.0
```

2. 分别创建 build 目录和 install 目录, build 目录用于存放编译的中间文件, install 目录用于存放最 生成的 lib 和 include 文件

3. 在 build 目录下执行 cmake-gui

- 1. 选择源码路径和 build 路径,最好将 Ground 和 Advanced 勾选上,方便查看需要配置的编译选项
  - 2. 选择编译工具链,可使用默认的本地工具链或指定交叉编译工具链
  - 3. 配置 lib 和 include 文件的安装路径

Search: pre	✓ Grouped ✓ Advance
Name	Value
CMAKE CMAKE_INSTALL_PREFIX	/home/jam/Libs/opencv
4. 只使用 c++ 语言开发,可去掉 java 支持,减少编译	<b>鲜的问</b>
Where to build the binaries: //home/jam/Downloads/op	encv_build
S <u>e</u> arch: java	

ſ	Name			Value
F	- BL	JILD		
		BUILD_JAVA		
		BUILD_opencv_java Enable Java support		
	<u>ها ام</u>		和田台社子。」。	

5. 为了可以正常加载多媒体文件需要增加 ffmpeg 支持,如果安装了 pkg-config,且配置了 ffmp g 的 pkconfig 目录,勾选后会自动搜索到相关 lib 路径

• 添加 ffmpeg 后编译时出现了,找不到相关定义的问题,原因是 ffmpeg 从 4.4 的版本开始 口出现了较大变化,导致 opencv 编译失败,需要重新编译低版本的 ffmpeg,实测 ffmepg4.3.5 适 opencv4.6.0

vnere t	o pulla	the	pinaries:	II.	/nome/	iam.	Dow	mo	ads/	0	pency	pulla
	o bana		Dillion icor		/	Jenny	2011	1110	4407	~	Perior_	_wanta

earch: ffmpeg	✓ Grou
Name	Value
- WITH	
WITH_FFMPEG	✓
✓ pkgcfg	
pkgcfg_lib_FFMPEG_avcodec	/home/jam/Libs/
pkgcfg_lib_FFMPEG_avformat	/home/jam/Libs/
pkgcfg_lib_FFMPEG_avutil	/home/jam/Libs/
pkgcfg_lib_FFMPEG_swscale	/home/jam/Libs/

6. 如果安装了已经安装了 QT,可以开启 QT 支持,配置以下两个路径后, cmake 会自动搜索其 库的路径

- 配置 qmake 目录,我安装的是 6.4.1,所以路径为 ".../6.4.1/gcc\_64/bin/qmake"
- 配置 QT5 或 QT6 目录,路径为 ".../6.4.1/gcc\_64/lib/cmake/QT6"

QT_ADDITIONAL_HOST_PACKAGES_PREFIX_PATH		
QT_ADDITIONAL_PACKAGES_PREFIX_PATH QT_QMAKE_EXECUTABLE	/home/jam/Tools/Qt/6.4.1/gcc_64/bin/qmake	
- WITH		
WITH_QT	v	•

earch: qt	Grouped V Advanced Add Entry
lame	Value
<ul> <li>Ungrouped Entries         Qt5_DIR         Qt6_DIR      </li> <li>7. 此外还可以开启 vulkan 和 opengl 支持,方         </li> <li>8. 为了减少编译时间,可以关闭所有 test         </li> <li>9. 为了方便引用 opencv 库,可以生成 pkg-cce         </li> <li>Where to build the binaries: /home/jam/Downloads/     </li> </ul>	Qt5_DIR-NOTFOUND /home/jam/Tools/Qt/6.4.1/gcc_64/lib/cmake/Qt6 法同上 onfig 文件
Search: pkg	✓ Grouped
Name	Value
<ul> <li>▼ OPENCV OPENCV_GENERATE_PKGCONFIG</li> <li>▼ PKG</li> <li>■ PKG_CONFIG_ARGN</li> <li>10. 此外还可以通过配置</li> <li>BUILD_opencv_w</li> <li>ED_LIBS 来决定生成动态库或静态库</li> </ul>	✓ vorld 将所有库打包为一个文件,配置 BUILD_SHA
11. 如果需要使用人脸检测等功能,需要添加 o	epncv_contrib,配置如下路径
● 如果需要编译 contrib,最好关闭 QT 编译 配置为系统路径,且 QT 版本不适配会导致编译失!	译支持,实践发现需要将 QT 的 include 下的子目 败
Search: ext	Grouped 🗹 Advanced 🖶 Add Entry 🗱 Remove
Name	Value
OPENCV     OPENCV_EXTRA_MODULES_PATH	/home/jam/Downloads/opencv_contrib-4.6.0/modules

X11 X11 Xext INCLUDE PATH

X11\_Xext\_INCLUDE\_PATH //usr/include 12. 配置完后选择 configure,检查输出信息是否有找不到的库或文件,如果出现红色错误或警告 最好解决后再进行编译

1. 找不到库就手动安装, 或者去掉不需要的编译项

2. 网络原因导致无法下载 github.raw 上的数据文件,可以手动下载后替换,具体方案可自行 索

13. 然后 generate, 没有错误后, 退出 cmake-gui

4. 在 build 目录下执行 make

5. 编译成功后,再执行 make install 即可

6. 如果生成的动态库,需要配置动态库加载路径,或者编译时指定 rpath