

在ROS中重新编译和配置OpenCV 4.2版本

背景问题

1. GTK的依赖与ORB中的GTK依赖有冲突。
2. 如果需要新增加模块且不希望系统中有多份OpenCV，可以按照以下方法重新编译OpenCV并重置系统中的OpenCV。

步骤

1. 下载OpenCV源码和OpenCV-contrib源码

- OpenCV: [opencv 4.2.0](#) 或者 [GitHub](#)
- OpenCV-contrib: [GitHub](#)

2. 编译OpenCV和contrib库

推荐安装cmake-gui:

```
▶ sudo apt-get install cmake-gui
```

2.1 更改CMakeList

- `opencv/3rdparty/carotene/CMakeList.txt` 将以下内容添加到文件中:

```
set_source_files_properties(${carotene_sources} COMPILE_FLAGS "--param ipa-cp-unit-growth=100000 -
```

- `opencv_contrib/modules/sfm/src/libmv_light/libmv/multiview/CMakeLists.txt` 将以下内容添加到文件中:

```
set(CMAKE_CXX_STANDARD 14)
set(CMAKE_CXX_STANDARD_REQUIRED ON)
``
```

2.2 配置CMake编译选项

选择安装模块和安装路径，注意路径不要设置为系统路径，后面会手动拷贝到系统目录中:

```
▶ cmake -DOPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules-4.2.0/modules \
-DOPENCV_GENERATE_PKGCONFIG=TRUE \
```

```
-DCMAKE_INSTALL_PREFIX=/home/autolabor/ThirdParty/opencv-build \  
-DBUILD_TESTS=OFF \  
-DCMAKE_CXX_STANDARD=11  
-DCMAKE_CXX_STANDARD_REQUIRED=ON..
```

或者通过cmake-gui选择这些编译选项

2.3 编译和安装

▶ `sudo make install -j8`

3. 覆盖系统OpenCV库

3.1 拷贝文件夹内容

分别将右侧的 `bin`、`include`、`lib`、`share` 中的内容拷贝到:

▶ `sudo cp -r /home/autolabor/ThirdParty/opencv-build/build/bin/* /usr/bin/`
`sudo cp -r /home/autolabor/ThirdParty/opencv-build/build/include/opencv4 /usr/include/`
`sudo cp -r /home/autolabor/ThirdParty/opencv-build/build/share/opencv4 /usr/share/`
`sudo cp /home/autolabor/ThirdParty/opencv-build/build/lib/*.4.2.0 /usr/lib/aarch64-linux-gnu/`

3.2 创建软链接脚本

执行 `create_link.sh` 脚本, 添加so库的链接 `*.so -> *.so.4.2`, `*.so.4.2 -> *.so.4.2.0`

3.3 配置pkgconfig

拷贝 `opencv4.pc` 文件到 `/usr/lib/aarch64-linux-gnu/pkgconfig/` 中:

▶ `sudo cp /home/autolabor/ThirdParty/opencv-build/build/lib/pkgconfig/opencv4.pc /usr/lib/aarch64-linux-`

使用 `pkg-config --modversion opencv4`, 检查是否配置成功, 正常能看到4.2.0的输出。

3.4 配置CMake文件

- 拷贝OpenCv的CMake文件到 `/usr/lib/aarch64-linux-gnu/cmake/` 中:

▶ `sudo cp -r /home/autolabor/ThirdParty/opencv-build/build/lib/cmake/opencv4 /usr/lib/aarch64-linux-`

- 修改 `OpenCVConfig.cmake` 文件, 将 `OpenCV_INSTALL_PATH` 目录多添加一层 `../`, 即指向 `/usr`。
- 修改 `OpenCVModules-release.cmake` 文件, 将所有的 `/lib/` 替换为 `/aarch64-linux-gnu/`。

这样就完成了CMake的配置, 在调用 `find_package(OpenCV)` 时, 会调用这几个CMake文件。

3.5 配置Python

拷贝python3.8和python2.7调用的动态连接库：

```
➤ sudo cp /home/autolabor/ThirdParty/opencv-build/build/lib/python3.8/dist-packages/cv2/python3.8/*.so ,  
sudo cp /home/autolabor/ThirdParty/opencv-build/build/lib/python2.7/dist-packages/cv2/python2.7/*.so ,
```

此时，就可以在Python中调用 `import cv2` 了。