

1.实验名称及目的

1.2实验名称

Ceres及opencv安装。

1.3实验目的

主要讲解如何通过脚本实现ceres以及opencv的快速安装。

1.4关键知识点

无

2.实验效果

通过脚本实现ceres以及opencv的快速安装。

3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\7.OpencvC++](#)

文件夹/文件名称	说明
ceres	ceres安装配置文件
opencv	Opencv安装配置文件
WslGUI.bat	一键运行脚本

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Ubuntu虚拟机。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：

<https://rflysim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 1台。

①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1 ceres的安装

首先，我们需要一台装有Ubuntu的电脑或虚拟机，

将该实验的文件夹整体进行拷贝到Ubuntu系统下，进入

到 `8.RflySimVision\0.ApiExps\2-DistributedSimAPI\4.0opencv++\ceres` 文件夹界面下，鼠标右键，选择在该界面下打开终端，运行ceres_install.sh文件进行ceres的安装，运行命令为 `./ceres_install.sh`。

代码中附有详细解释。

```

1 | \#!/bin/bash
2 |
3 | \# please refer to
4 |
5 | \# http://ceres-solver.org/installation.html\#linux
6 |
7 | \# https://blog.csdn.net/qq_62964142/article/details/132325484
8 |
9 | \# 卸载旧平台
10 |
11 | sudo rm -r /usr/local/lib/cmake/Ceres
12 |
13 | sudo rm -rf /usr/local/include/ceres /usr/local/lib/libceres.a
14 |
15 | \# 安装1.14.0版，注意，也能替换成2.2.0，不过改变较大，易出现兼容性问题
16 |
17 | sudo wget -O ceres-solver-1.14.0.tar.gz
18 | http://ceres-solver.org/ceres-solver-1.14.0.tar.gz --no-check-certificate
19 |
20 | tar zxf ceres-solver-1.14.0.tar.gz
21 |
22 | mkdir ceres-bin
23 |
24 | cd ceres-bin
25 |
26 | cmake ../ceres-solver-1.14.0
27 |
28 | make -j 16
29 |
30 | make test
31 |
32 | \# Optionally install Ceres, it can also be exported using CMake which
33 |
34 | \# allows Ceres to be used without requiring installation, see the documentation
35 |
36 | \# for the EXPORT_BUILD_DIR option for more information.
37 |
38 | sudo make install

```

该脚本默认安装的为ceres1.14版本，同样也可以通过对脚本中的版本号进行更改，来进行安装2.2.0版本，但是变化过大，可能出现兼容问题。

5.2 opencv的安装

将该实验的文件夹整体进行拷贝到Ubuntu系统下，进入到 `8.RflySimVision\0.ApiExps\2-DistributedSimAPI\4.OpencvC++\opencv` 文件夹界面下，鼠标右键，选择在该界面下打开终端，运行installOpencv.sh文件进行opencv的安装，运行命令为 `./installOpencv.sh`。

代码中附有详细解释。

```
1 | \#!/bin/bash
2 |
3 | \# 如果出现ceres报错,可参考如下链接
4 |
5 | \# https://blog.csdn.net/weixin\_46808875/article/details/126471302
6 |
7 | \# 1. Install minimal prerequisites, libgtk2.0-dev pkg-config 用来显示图像
8 |
9 | sudo apt update && sudo apt install -y cmake g++ wget unzip libgtk2.0-dev
10 | pkg-config
11 |
12 | if [ -e opencv.zip ]; then
13 |
14 | echo "File Exist.."
15 |
16 | rm -rf ./build
17 |
18 | else
19 |
20 | echo "File Download .."
21 |
22 | \# 2. Download and unpack sources
23 |
24 | sudo wget -O opencv.zip https://github.com/opencv/opencv/archive/4.2.0.zip
25 | \--no-check-certificate
26 |
27 | sudo wget -O opencv_contrib.zip
28 | https://github.com/opencv/opencv_contrib/archive/4.2.0.zip
29 | \--no-check-certificate
30 |
31 | unzip opencv.zip && unzip opencv_contrib.zip
32 |
33 | fi
34 |
35 | \# 3. Create build directory and switch into it
36 |
37 | mkdir build && cd build
38 |
39 | \# 4. Configure, build 目录内
40 |
41 | \# cmake -DCMAKE_BUILD_TYPE=Release -DOPENCV_GENERATE_PKGCONFIG=ON
42 | \-DCMAKE_INSTALL_PREFIX=/usr/local
43 | \-DOPENCV_EXTRA_MODULES_PATH=./opencv_contrib-4.8.0/modules ../opencv-4.8.0
44 |
45 | cmake -D CMAKE_BUILD_TYPE=RELEASE \
46 | \-D CMAKE_INSTALL_PREFIX=/usr/local \
47 | \-D INSTALL_C_EXAMPLES=ON \
48 | \-D INSTALL_PYTHON_EXAMPLES=ON \
```

```

53 \-D OPENCV_GENERATE_PKGCONFIG=ON \
54
55 \-D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib-4.2.0/modules \
56
57 \-D BUILD_EXAMPLES=ON \
58
59 ../opencv-4.2.0
60
61 \# 5. build 目录内
62
63 make -j 32
64
65 \# 安装opencv到路径
66
67 sudo make install
68
69 \# 添加动态链接库
70
71 sudo ldconfig
72
73 \# 确保 pkg-config --modversion opencv 能起作用
74
75 cd /usr/local/lib/pkgconfig/
76
77 cp opencv4.pc opencv.pc
78
79 pkg-config --modversion opencv
80
81 \# \# 删除压缩包
82
83 \# cd .. && sudo rm opencv.zip opencv_contrib.zip

```

5.3 Rosbridge的安装

同样，在4.2节中打开的opencv文件夹下打开终端，运行installRosBridge.sh文件进行Rosbridge协议的安装，运行命令为

```
./installRosBridge.sh
```

Rosbridge是一个在ROS（Robot Operating System，机器人操作系统）系统和其他系统之间起到桥梁作用的工具包。以下是关于Rosbridge的详细介绍：

功能：

- Rosbridge为ROS系统与非ROS系统之间的通信提供了可能。它允许非ROS程序通过JSONAPI与ROS进行交互。
- Rosbridge屏蔽了代理转发过程，使得开发者可以更加便捷地进行系统间的通信。

3. 文件目录

- Rosbridge的内容结构主要包括rosbridge protocol（协议）和rosbridge implementation（实现）。
- Rosbridge Protocol：规定了通信格式，即ROS中具体操作（如话题发布、订阅）所使用的JSON格式。
- Rosbridge Implementation：具体实现功能，包括rosbridge_library（负责获取json字符串并将命令发送到ROS）、rosbridge_server（提供传输功能，支持WebSocket连接）和rosapi（提供ROS服务，如获取和设置参数、获取主题列表等）。

应用场景：

- Rosbridge适用于浏览器和ROS系统交互的场景，通过WebSocket API实现。
- Rosbridge也适用于安卓平台和ROS系统交互的场景，采用C/S模式进行通信。

特点：

- 跨平台：Rosbridge支持各种操作系统和Web浏览器，具有跨平台性。
- 高度定制化：Rosbridge支持多种消息类型和服务调用，开发者可以根据需要自定义应用程序的行为。

使用：

- 要使用Rosbridge，开发者需要集成Rosbridge Suite（包含Rosbridge的元程序包）到ROS或Web应用程序中。这通常涉及将Python服务器添加到ROS工作区，并在Web应用程序中引入JavaScript客户端库。

与其他工具的比较：

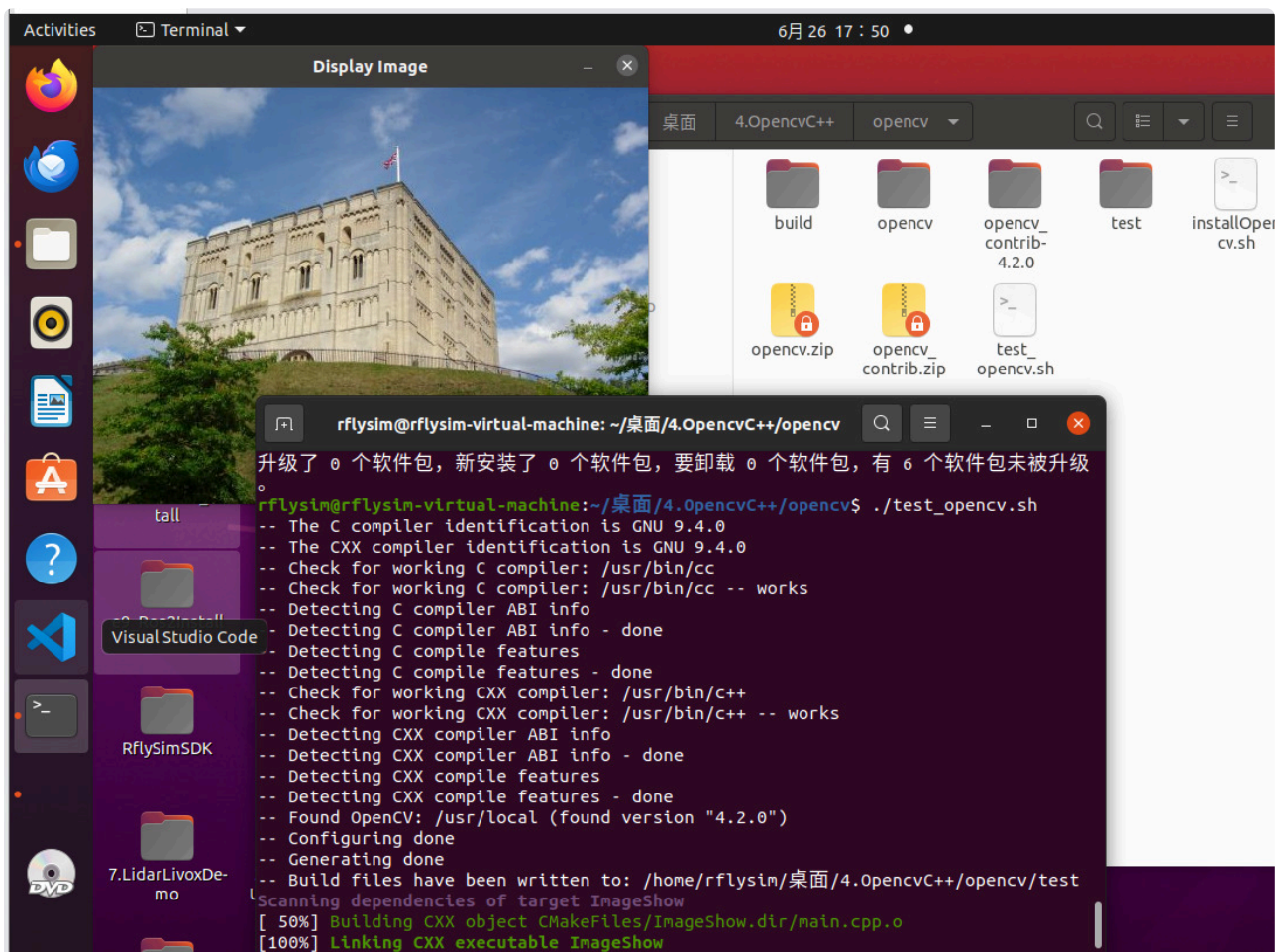
- 与rosjava相比，Rosbridge提供了更轻量级的ROS交互能力，适用于资源受限的移动平台或嵌入式系统。Rosjava虽然功能全面，但体积较大，集成到Android等平台可能较为复杂。

```
1 | \#!/bin/bash
2 |
3 | sudo apt-get install ros-noetic-cv-bridge
4 |
5 | sudo apt-get install ros-foxy-cv-bridge
```

```
rflysim@rflysim-virtual-machine:~/桌面/4.OpencvC++/opencv$ ./installRosBridge.sh
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
ros-noetic-cv-bridge 已经是最新版 (1.16.2-1focal.20230620.191830)。
ros-noetic-cv-bridge 已设置为手动安装。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 6 个软件包未被升级。
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
ros-foxy-cv-bridge 已经是最新版 (3.0.7-1focal.20230527.052309)。
ros-foxy-cv-bridge 已设置为手动安装。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 6 个软件包未被升级。
```

5.4 Opencv测试

在进行opencv的安装之后，我们可以在所打开的终端中进行测试，可以通过运行test_opencv.sh文件，来观察输出的结果，运行命令为 `./test_opencv.sh`。



6.参考资料

1. 无

7.常见问题

Q1: 无

A1: 无