

写在之前：本文主要对基于ros和C++程序的调用做简要总结，第一部分简单分析了各文件分别什么作用，怎么更改每个文件的参数。第二部分演示了如何利用程序调出激光点云转化后的深度图，只讲如何调用和更改参数，不分析转换深度图过程中的代码含义。之后的其它程序的参数更改和调用依旧适用，可供参考。

首先对不同的文件作用及内部涉及参数进行讲解：

本文中“package”=“包”或“包名”

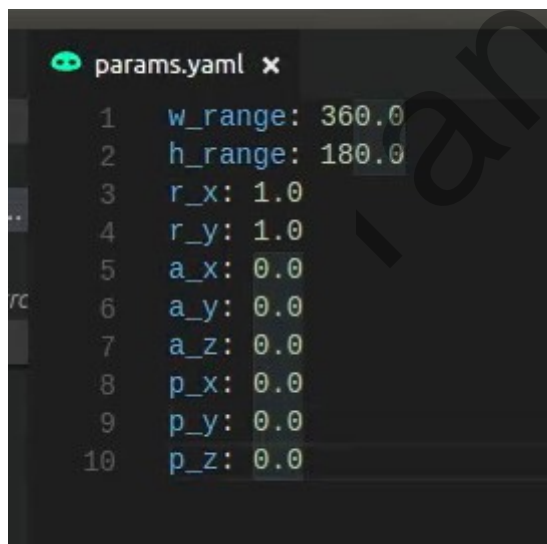
一、几种文件的介绍：

1、**cfg文件夹**：放置参数（param）文件包括“.cfg”文件和“.yaml”等文件，具体内容可以包括视野范围、分辨率、相机位姿等参数。

参数param有什么用

机器人工作时，我们需要对机器人的参数（如传感器参数、算法的参数）进行设置。有些参数（如机器人的轮廓、传感器的高度）在机器人启动时就设定好就行了；有些参数则需要动态改变（特别是在调试的时候）。ROS提供了参数服务器来满足这一需求。我们可以将参数设置在参数服务器，在需要用到参数的时候再从参数服务器中获取。

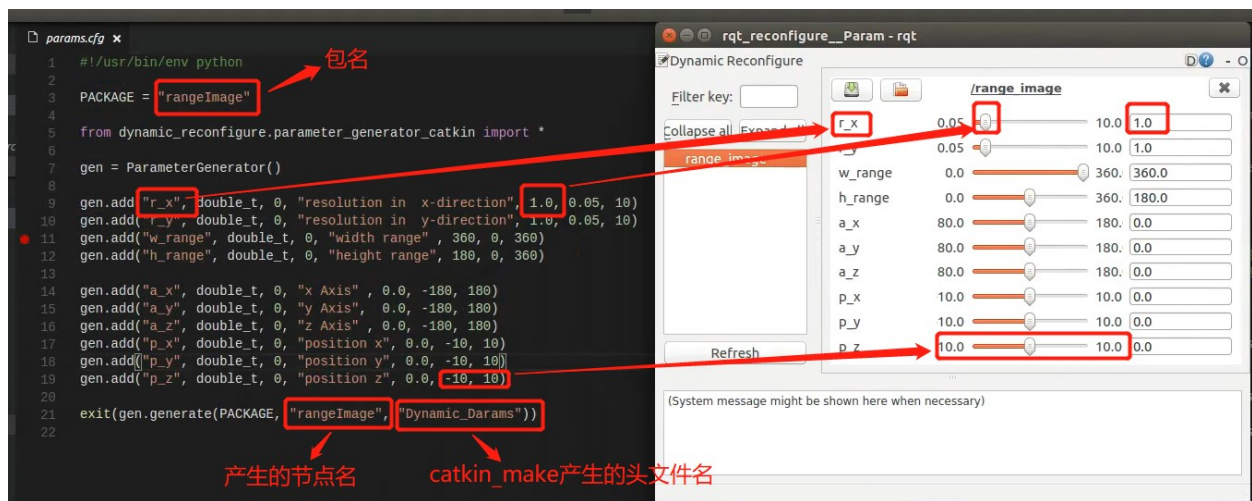
(1) “.yaml”文件是静态参数文件，其格式如图所示：



(2) “.cfg”文件是动态重配置参数文件，什么是动态重配置参数文件，它有什么作用？

动态重配置参数：就是动态地调整、设置正在运行的节点的参数值，通过动态重配置，可以更加高效地开发和测试节点，特别是机器人硬件调试的时候，使用动态重配置参数是一个不错的选择。

“.cfg”文件参数设置和控制界面相关，可以随时动态控制界面，目前只需要理解每部分什么意思，如何对应动态调整界面，其余稍后会详细讲解。



每部分含义：

```
#gen.add(参数名称, 参数类型, 位掩码, 参数描述, 默认值, 最小值, 最大值)
gen.add("int_param", int_t, 0, "An Integer parameter", 50, 0, 100)
gen.add("double_param", double_t, 0, "A double parameter", .5, 0, 1)
gen.add("str_param", str_t, 0, "A string parameter", "Hello World")
gen.add("bool_param", bool_t, 0, "A Boolean parameter", True)

size_enum = gen.enum([ gen.const("Small", int_t, 0, "A small constant"),
                        gen.const("Medium", int_t, 1, "A medium constant"),
                        gen.const("Large", int_t, 2, "A large constant"),
                        gen.const("ExtraLarge", int_t, 3, "An extra large constant")],
                        "An enum to set size")
```

2、“.launch”文件夹：放置“.launch”文件

“.launch”文件有什么用？——用“roslaunch”命令一个接一个启动很多命令会很麻烦，用launch文件可以把很多node都写进去，通过launch命令直接启动很多节点。

launch文件怎么写？——如下图，根据自己的文件，需要更改的就是黄字部分

02 编写launch文件并测试

ROS小课堂

launch文件内容

```
1 <!--
2 Copyright: 2016-2019 ROS小课堂 www.corvin.cn
3 Author: corvin
4 Description: This is a demo launch file.
5 -->
6 <launch>
7   <node pkg="ros_test_pkg" type="publish_node" name="test_publish_node" output="screen"/>
8
9   <node pkg="ros_test_pkg" type="subscribe_node" name="test_subscribe_node" output="screen"/>
10 </launch>
11 要启动节点所在的软件包 要启动的执行文件名 为当前节点定义一个名称
```

在launch文件前面可以使用<!-- -->标签,为本代码增加注释信息,这样就方便后面的代码维护。

会把ros::init 后的节点覆盖掉

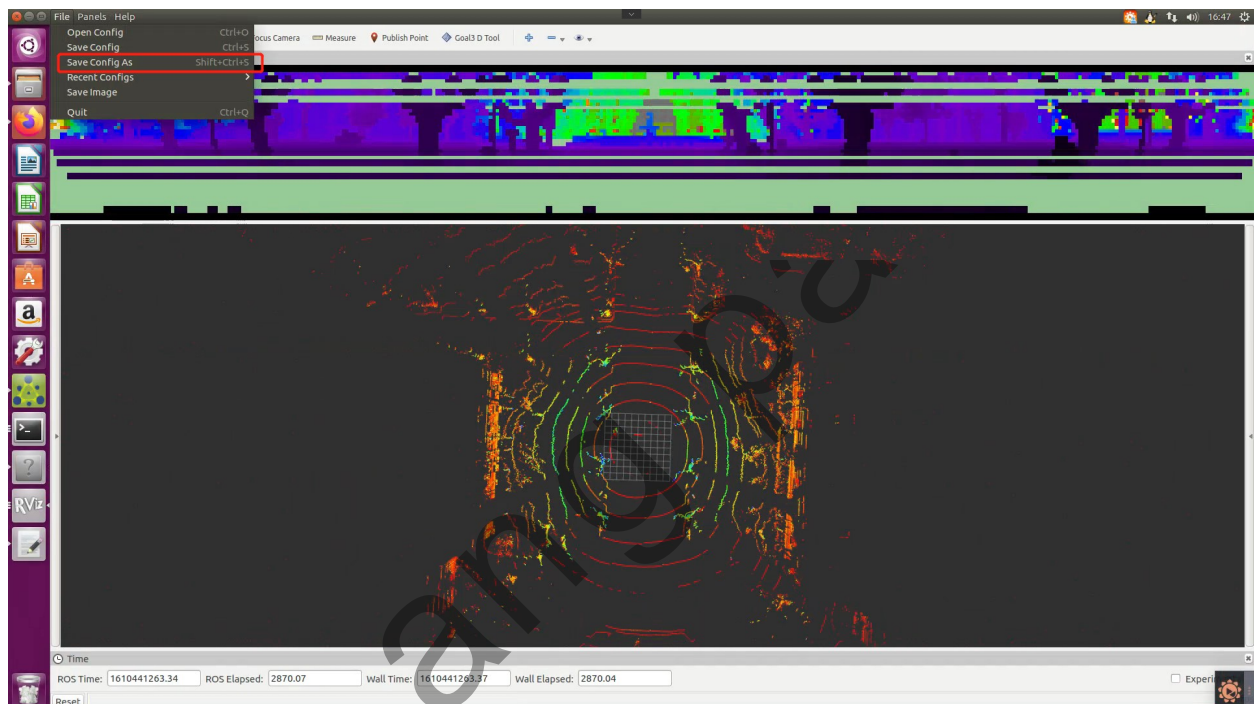
Ros小课堂

里边可以设置同时启动.cpp程序和rviz可视化界面如下图，需要根据自己各文件名更改的部分已用红线标出：

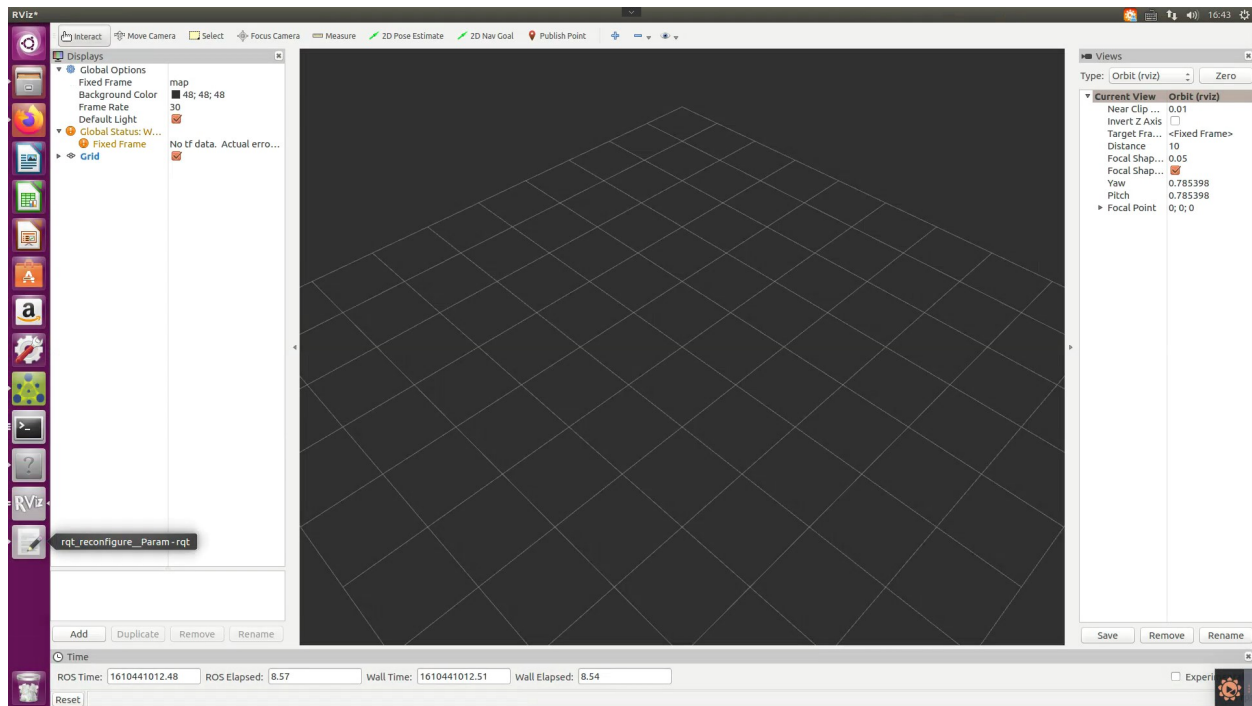
```
run.launch x
1 <?xml version="1.0"?>
2 <launch>
3   <node name="range_image" pkg="rangeimage" type="range_image" output="screen">
4     <rosparam file="$(find rangeimage)/cfg/params.yaml" command="load"/>
5   </node>
6   <node name="rvizvisualisation" pkg="rviz" type="rviz" output="log" args="-d $(find rangeimage)/rviz/range_image.rviz" />
7 </launch>
8
```

3、rviz文件夹：放置“.rviz”文件

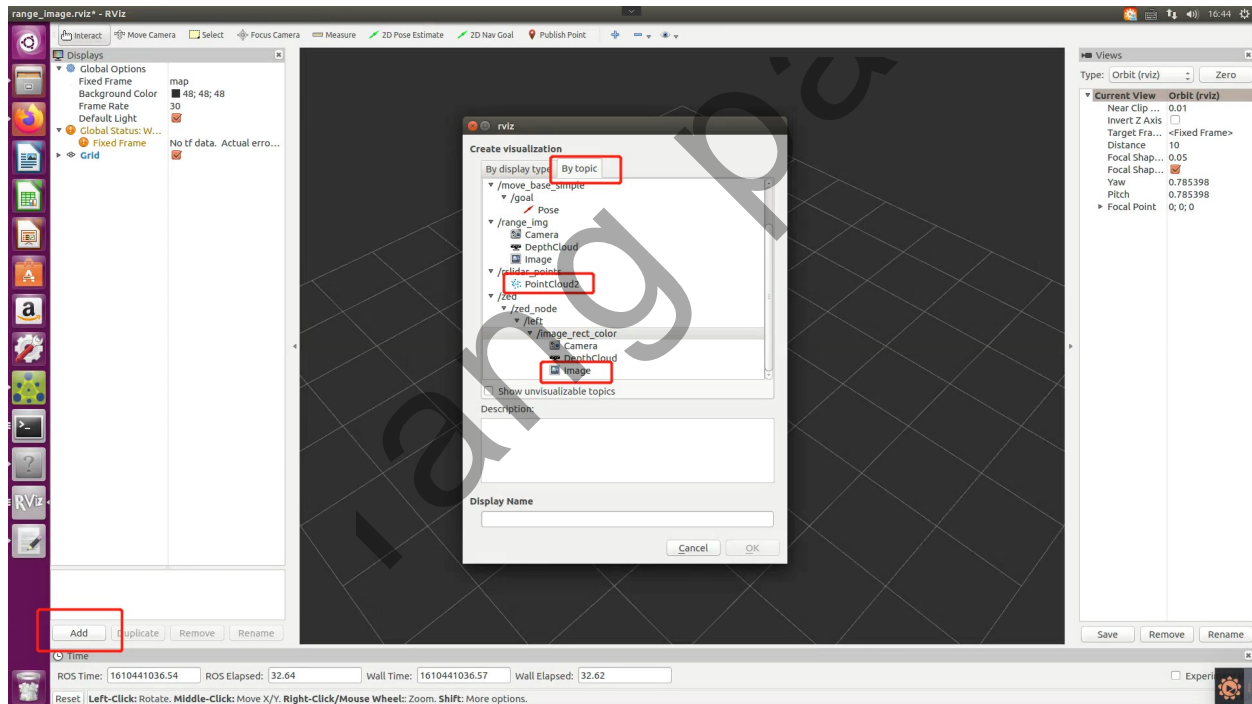
“.rviz”文件有什么用？——可以通过保存的“.rviz”文件直接打开之前配置好的界面（save configure as命令可以存储配置好的topic界面）如下图：



因为正常情况可以通过“.rviz”命令直接打开Rviz，但是打开的是一个空界面



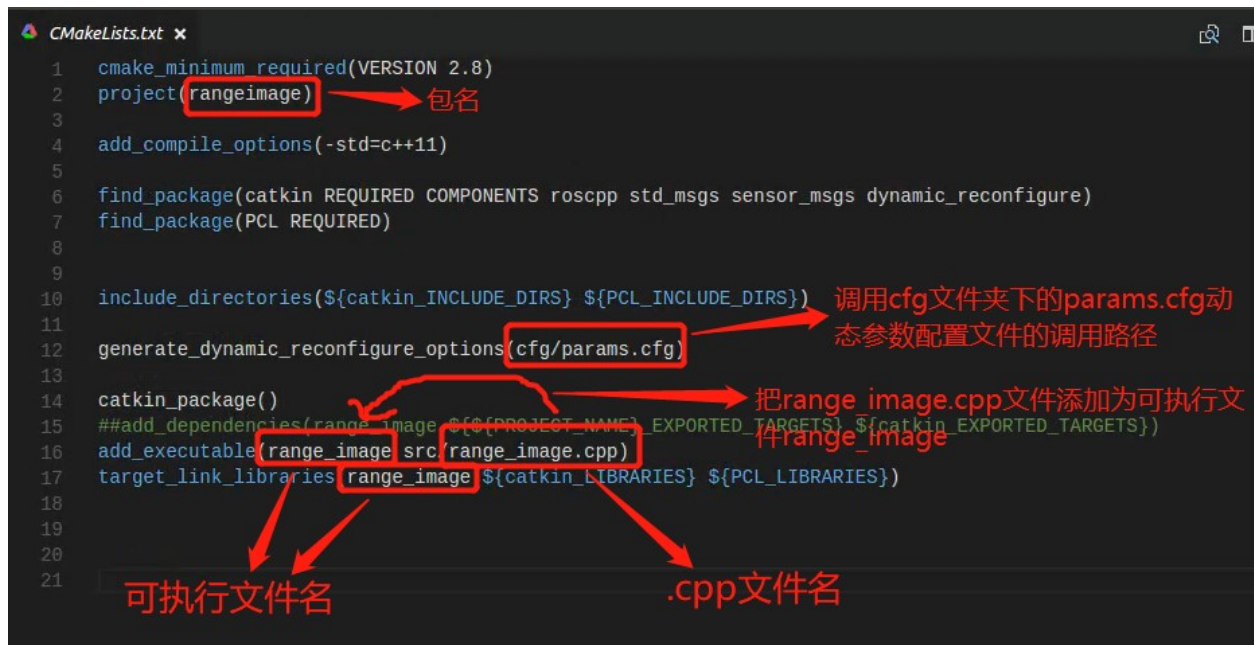
需要自己通过“By topic”去一个一个添加所需要的topic，如下图：



4、CMakeLists.txt文件

有什么用？——在ROS系统的功能包中 要包含 CMakeLists.txt 与 package.xml 文件 来编译功能包的内容。这个文件直接规定了这个package要依赖哪些package，要编译生成哪些目标，如何编译等等 流程。所以 CMakeLists.txt 非常重要，它指定了由源码到目标文件的规则，catkin编译系统在工作时首先会找到每个package下的 CMakeLists.txt，然后按照规则来编译构建。

怎么改？——如下图所示，需要更改的部分都已标出。



5、.xml文件

有什么用？——定义package的属性；例如：包名、版本号、作者、依赖等。

如何改？——如下图，一般主要需要自己定义更改为自己的包名，因为包名在其它文件比如“CMakeLists.txt”和“.launch”和“.cfg”文件中都会被调用，其它依赖部分根据具体情况来定：

```
package.xml x
1  <?xml version="1.0"?>
2  <package>
3      <name>rangeimage</name>
4      <version>0.0.0</version>
5      <description>rangeimage</description>
6      <maintainer email="11111111@qq.com">ccf</maintainer>
7      <license>TODO</license>
8
9      <buildtool_depend>catkin</buildtool_depend>
10
11     <build_depend>roscpp</build_depend>
12     <run_depend>roscpp</run_depend>
13
14     <build_depend>std_msgs</build_depend>
15     <run_depend>std_msgs</run_depend>
16
17     <build_depend>sensor_msgs</build_depend>
18     <run_depend>sensor_msgs</run_depend>
19
20
21
22     <export> </export>
23
24 </package>
25
```


二、调用程序可视化深度图：

1、更改各部分名字。根据作业要求修改为自己需要的包名、topic名，节点名等，要让上述第一部分所有文件中的相关名字保持一致，作业的要求和更改如下面两张图，其余部分自己按照作业要求统一修改。

提交说明：

1. 提交 ROS package 和深度图截图或 gif 或视频(<10Mb)，参数设置#1 前视视角 360 度范围，#2 后视视角 180 度范围。
2. package 命名为 aiimooc_姓名缩写，如 aiimooc_scz。
3. node 和 script 命名为 range_image，图片命名为姓名缩写_N,N=1,2,3..., 如 scz_1.gif, scz_2.png。

```
package.xml ●
1  <?xml version="1.0"?>
2  <package>
3      <name>aiimooc_tangpan</name>
4      <version>0.0.0</version>
5      <description>rangeimage</description>
6      <maintainer email="11111111@qq.com">ccf</maintainer>
7      <license>TODO</license>
8
9      <buildtool_depend>catkin</buildtool_depend>
10
11      <build_depend>roscpp</build_depend>
12      <run_depend>roscpp</run_depend>
13
14      <build_depend>std_msgs</build_depend>
15      <run_depend>std_msgs</run_depend>
16
17      <build_depend>sensor_msgs</build_depend>
18      <run_depend>sensor_msgs</run_depend>
19
20
21
22      <export> </export>
23
24  </package>
25
```



2、roslaunch命令启动launch文件。在工作空间根目录（我的根目录是catkin_ws）下使用roslaunch命令启动 launch 文件：roslaunch rangeimage run.launch（roslaunch+包名+launch文件名），roscore也会自动启动。

3、rosviz命令打开动态参数调整界面（GUI）。rosviz rqt_reconfigure rqt_reconfigure 启动GUI界面即可动态调节。

