

LVI-SAM学习小组docker使用图文简洁介绍

前备条件：

执行步骤

- 1.拉取镜像
- 2.做好代码的隔离
- 3.启动docker镜像
- 4.编译之前的准备&&编译
- 5.运行launch文件
- 6.RVIZ和Bag

特别鸣谢

前备条件：

1. 安装docker
2. 安装ROS
3. 下载LVI-SAM源码
4. 下载LVI-SAM官方数据集

执行步骤

1.拉取镜像

```
1  docker pull liangjinli/slam-docker:v1.2
```

C++ | 复制代码

刘嘉林制作好了lvi-sam的docker镜像的环境配置，可以直接用这么命令从docker服务器拉取镜像
拉取大概过程如下，显示显示一些进度

```
92dc2a97ff99: Pull complete
be13a9d27eb8: Pull complete
c8299583700a: Pull complete
```

2. 做好代码的隔离

```
1  mkdir /home/lvi_sam_docker/catkin_ws/src
2  cd /home/lvi_sam_docker/catkin_ws/src
3  git clone git@github.com:TixiaoShan/LVI-SAM.git
```

这里开始，称呼docker外的环境叫**宿主机**，docker镜像内部叫**docker内**

上述步骤在宿主机内完成代码环境的分离

3. 启动docker镜像

```
1  docker run -v /home/lvi_sam_docker:/home/ --net=host -it
    liangjinli/slam-docker:v1.2 /bin/bash
```

执行成功之后，如下所示，这代表挂载是成功的

```
root@lvi_sam_docker:/home# l
catkin_ws/
root@lvi_sam_docker:/home# cd catkin_ws/
root@lvi_sam_docker:/home/catkin_ws# l
src/
```

4. 编译之前的准备&&编译

编译要在docker内进行，而此docker内没有配置可视化程序，所以需要注释掉run.launch文件里的一句话防止报错（宿主机和docker内的代码是连通的，在哪里注释都可以）

这个run.launch文件位置位于LVI-SAM代码文件里的launch文件夹里



```
4 <include file="$(find lvi_sam)/launch/include/module_robot_state_publisher.launch" />
5
6 <!-- Run Rviz -->
7 <include file="$(find lvi_sam)/launch/include/module_rviz.launch" />
8
```

```
5
6 <!-- Run Rviz
7 <include file="$(find lvi_sam)/launch/include/module_rviz.launch" />-->
8
```

完成上述注释后，执行编译操作，应该会顺利编译通过，因为docker内配置好了环境

```
[ 93%] Linking CXX executable /home/catkin_ws/devel/lib/lvi_sam/lvi_sam_imuPreintegration
[ 93%] Built target lvi_sam_imuPreintegration
[ 95%] Linking CXX executable /home/catkin_ws/devel/lib/lvi_sam/lvi_sam_visual_odometry
[ 95%] Built target lvi_sam_visual_odometry
[ 96%] Linking CXX executable /home/catkin_ws/devel/lib/lvi_sam/lvi_sam_mapOptimization
[ 96%] Built target lvi_sam_mapOptimization
[ 98%] Linking CXX executable /home/catkin_ws/devel/lib/lvi_sam/lvi_sam_visual_feature
[100%] Linking CXX executable /home/catkin_ws/devel/lib/lvi_sam/lvi_sam_visual_loop
[100%] Built target lvi_sam_visual_loop
[100%] Built target lvi_sam_visual_feature
root@lvi-sam-docker:/home/catkin_ws#
```

5.运行launch文件

在运行launch命令之前，在命令行里source一下（source仅在此命令行有效）

```
1 source devel/setup.bash
```

然后按照LVI-SAM的官网教程运行launch

```
1 roslaunch lvi_sam run.launch
```

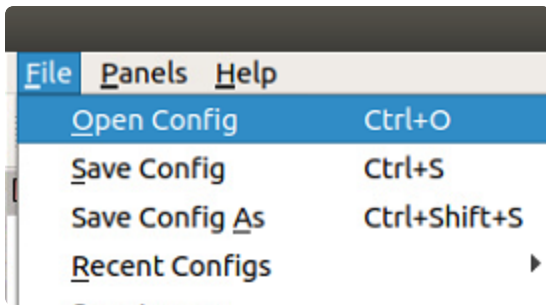
可得到如下显示

```
setting /run_id to a0560f92-de9f-11ec-8aca-18c04d40a34e
process[rosout-1]: started with pid [1426]
started core service [/rosout]
process[lvi_sam_robot_state_publisher-2]: started with pid [1433]
process[lvi_sam_imuPreintegration-3]: started with pid [1434]
[ INFO] [1653753772.820051901]: ----> Lidar IMU Preintegration Started.
process[lvi_sam_imageProjection-4]: started with pid [1441]
[ INFO] [1653753773.002211548]: ----> Lidar Cloud Deskew Started.
process[lvi_sam_featureExtraction-5]: started with pid [1447]
[ INFO] [1653753773.290497947]: ----> Lidar Feature Extraction Started.
process[lvi_sam_mapOptmization-6]: started with pid [1456]
[ INFO] [1653753773.455612392]: ----> Lidar Map Optimization Started.
process[lvi_sam_visual_feature-7]: started with pid [1462]
[ INFO] [1653753773.678975473]: ----> Visual Feature Tracker Started.
process[lvi_sam_visual_odometry-8]: started with pid [1470]
[ INFO] [1653753773.705884357]: ----> Visual Odometry Estimator Started.
process[lvi_sam_visual_loop-9]: started with pid [1476]
[ INFO] [1653753773.938345060]: ----> Visual Loop Detection Started.
process[lvi_sam_republish-10]: started with pid [1490]
```

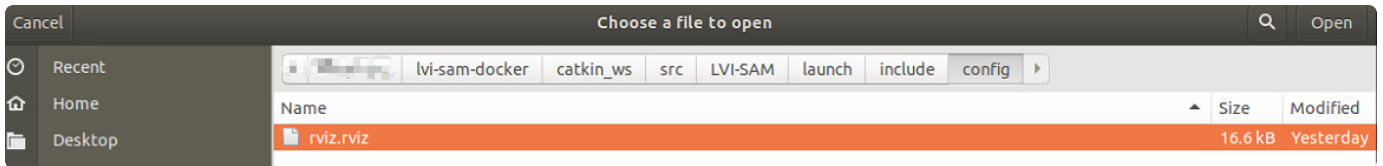
6.RVIZ和Bag

在宿主机内调用RVIZ

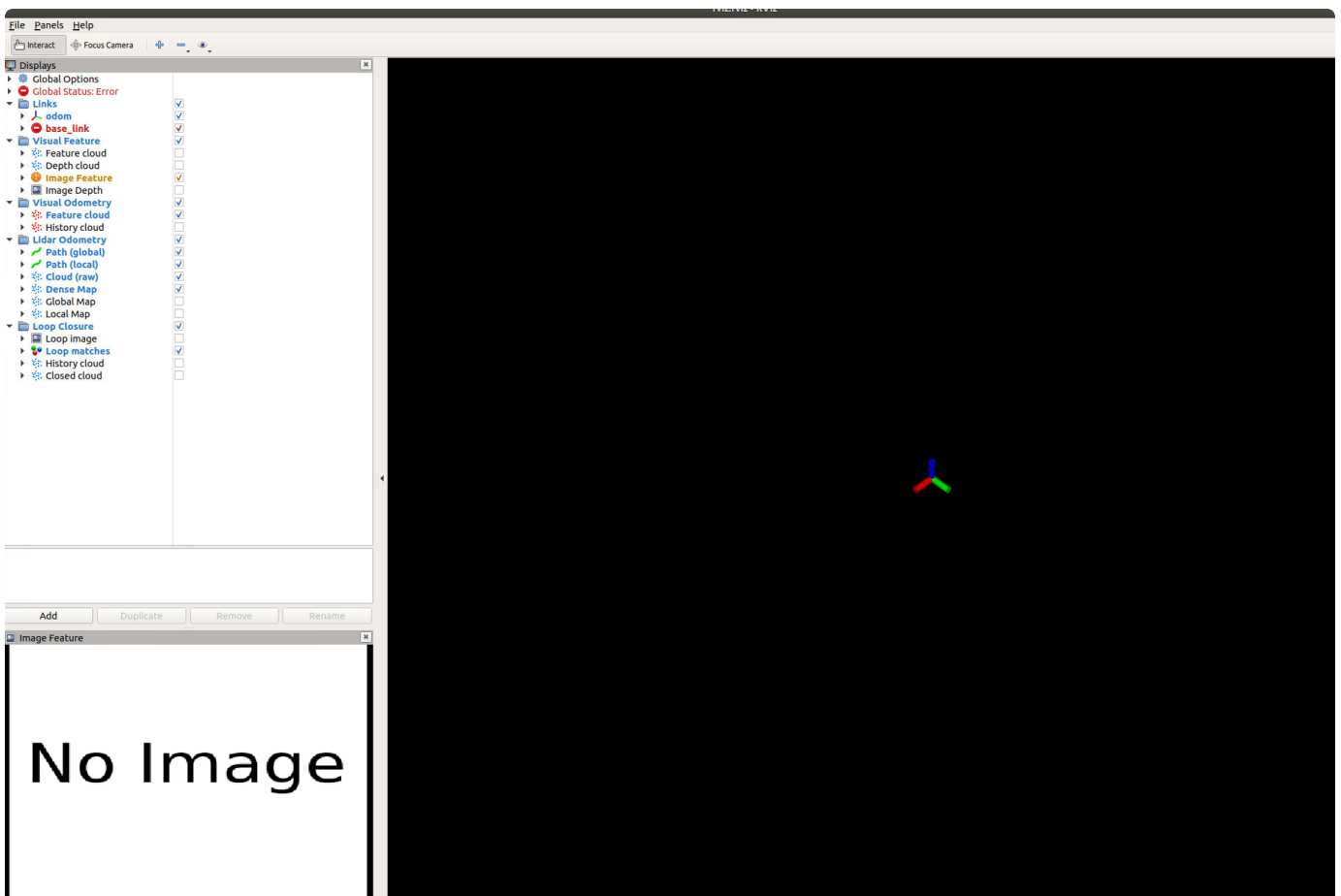
在左上角File选项加载open config



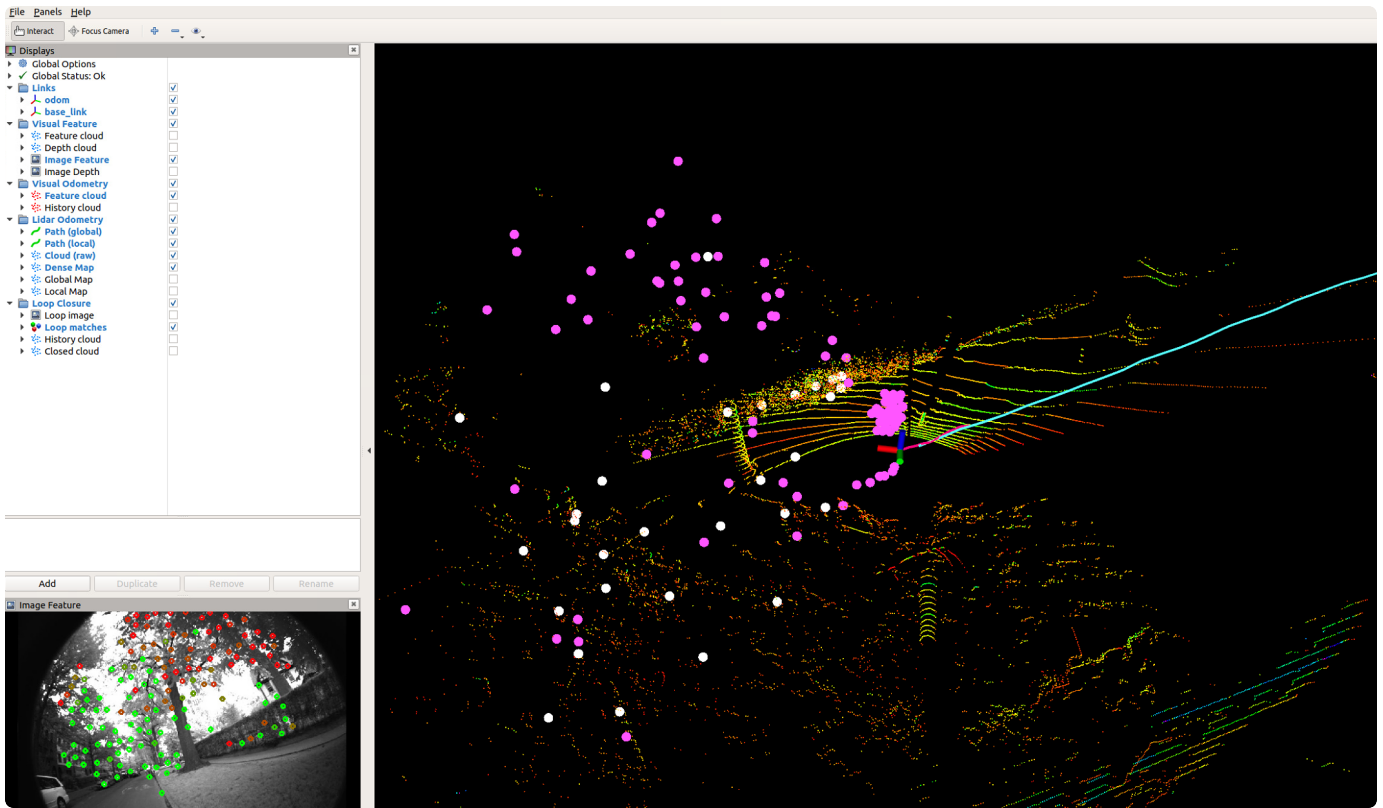
选择如下路径加载rviz.rviz



加载成功如下图所示



下载官方github放在google网盘上的bag包并播放，可得到如图



完毕，docker就顺利使用了！

特别鸣谢

教程联合创作者：汪寿安、刘嘉林

docker镜像创作者：刘嘉林