

# LVI-SAM Docker使用图文简介V2.0

---

前备条件:

执行步骤

- 1.拉取镜像
- 2.做好代码的隔离
- 3.第一次启动docker镜像
- 4.编译之前的准备&&编译
- 5.运行launch文件
- 6.RVIZ和Bag
- 7.重新进入docker与关闭镜像

版本更新说明

番外篇: 用此docker运行vins\_mono算法

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
```

LVI-SAM的Docker镜像的环境配置已经制作好, 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 -p /YOUR_PATH/lvi-sam-docker/catkin_ws/src
2  cd /YOUR_PATH/lvi-sam-docker/catkin_ws/src
3  git clone git@github.com:TixiaoShan/LVI-SAM.git
4
5  # 这里也可以拉取课程所用的代码仓库E-LVI-SAM
6  git clone git@github.com:shuttworth/E-LVI-SAM.git
```

注意，第一行第二行代码里的“YOUR\_PATH”需要更改为你本机要存放代码的位置，并完成代码在src目录下的git clone

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.第一次启动docker镜像

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

代码里的“YOUR\_PATH”需要更改为你本机要存放代码的位置

执行成功之后，如下所示，这代表挂载是成功的

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

## 4.编译之前的准备&&编译

(如果拉取的是E-LVI-SAM代码，则跳过本阶段处理，仅原TixiaoShan/LVI-SAM需要)

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run.launch文件夹里第七行



注释前

```
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
```

注释后

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

完成上述注释后，进入/home/catkin\_ws 目录下

接着执行编译操作catkin\_make，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: /home/catkin_ws#
```

## 5.运行launch文件

在docker内的/home/catkin\_ws 目录下，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）

```
▼ Bash |
1 source devel/setup.bash
```

然后按照LVI-SAM的官网教程运行launch

```
▼ Bash |
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]
```

如果是使用的E-LVI-SAM仓库

执行命令有所改变

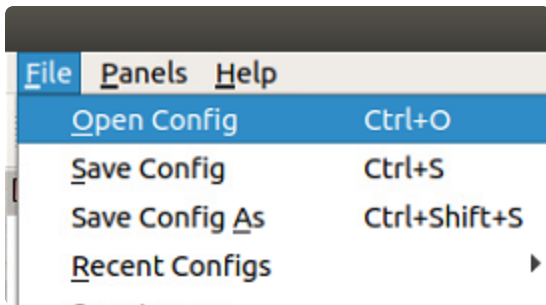
```
1 # 运行M2DGR的launch文件
2 roslaunch lvi_sam M2DGR.launch
3
4 # 播放M2DGR的bag包
5 rosbag play door_03.bag
```

bag包需自行下载，详见M2DGR的[数据集链接](#)

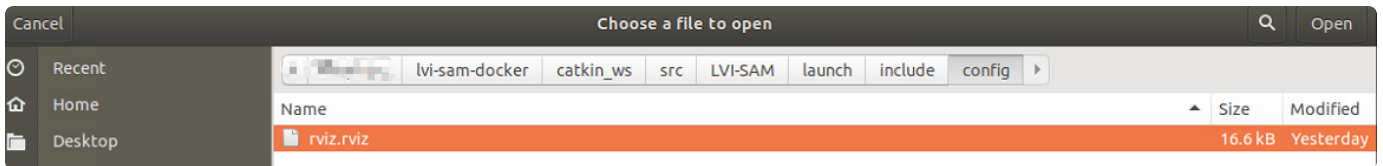
## 6.RVIZ和Bag

在宿主机内调用RVIZ

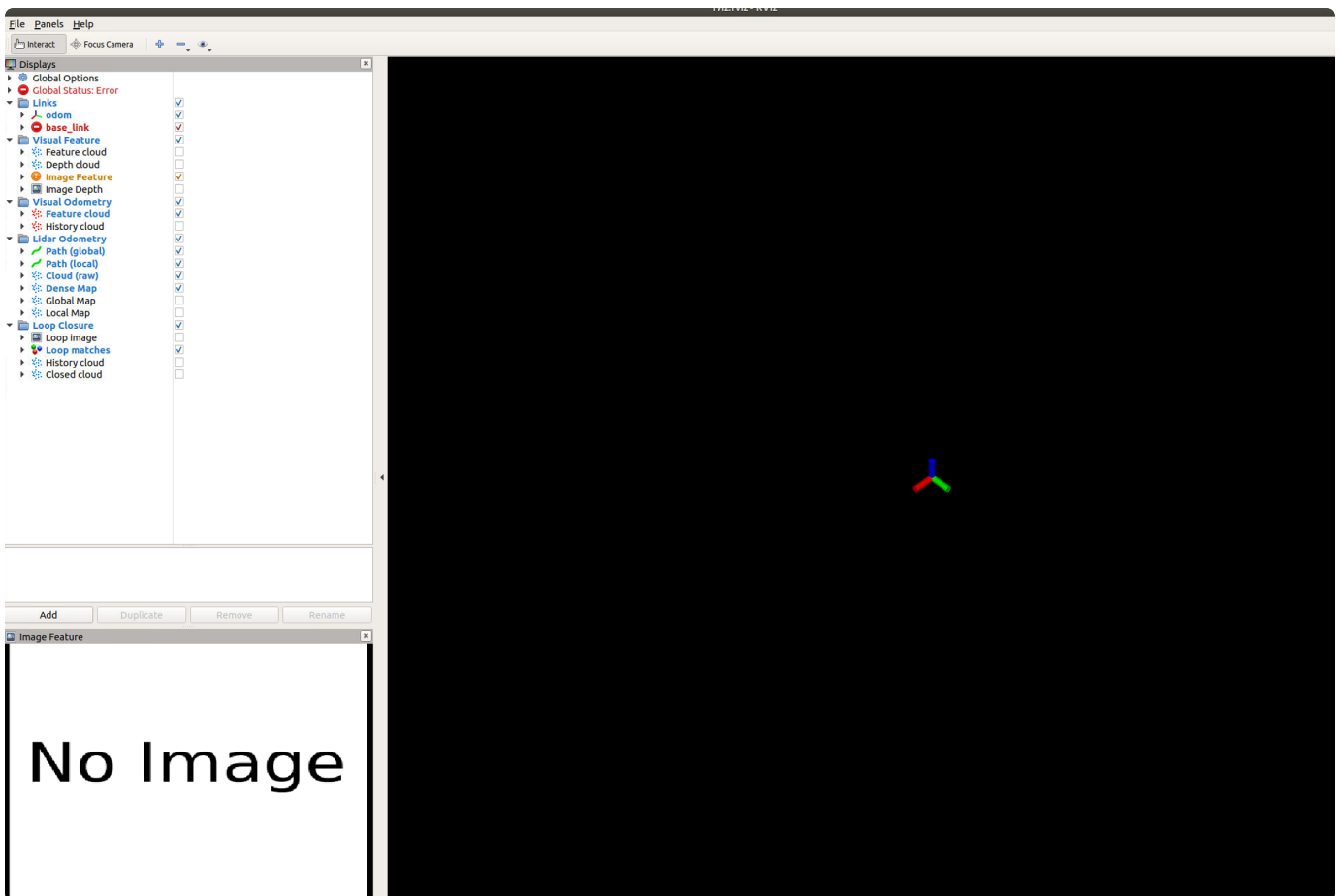
在左上角File选项加载open config



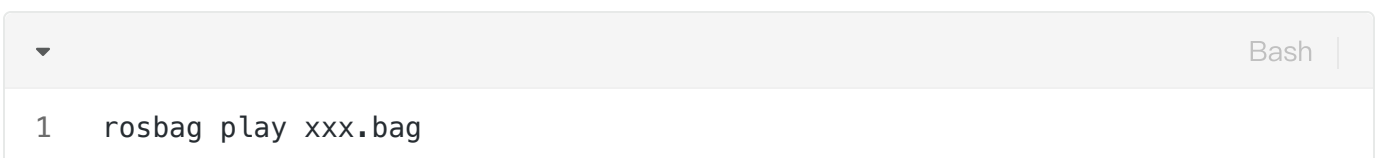
选择如下路径加载rviz.rviz

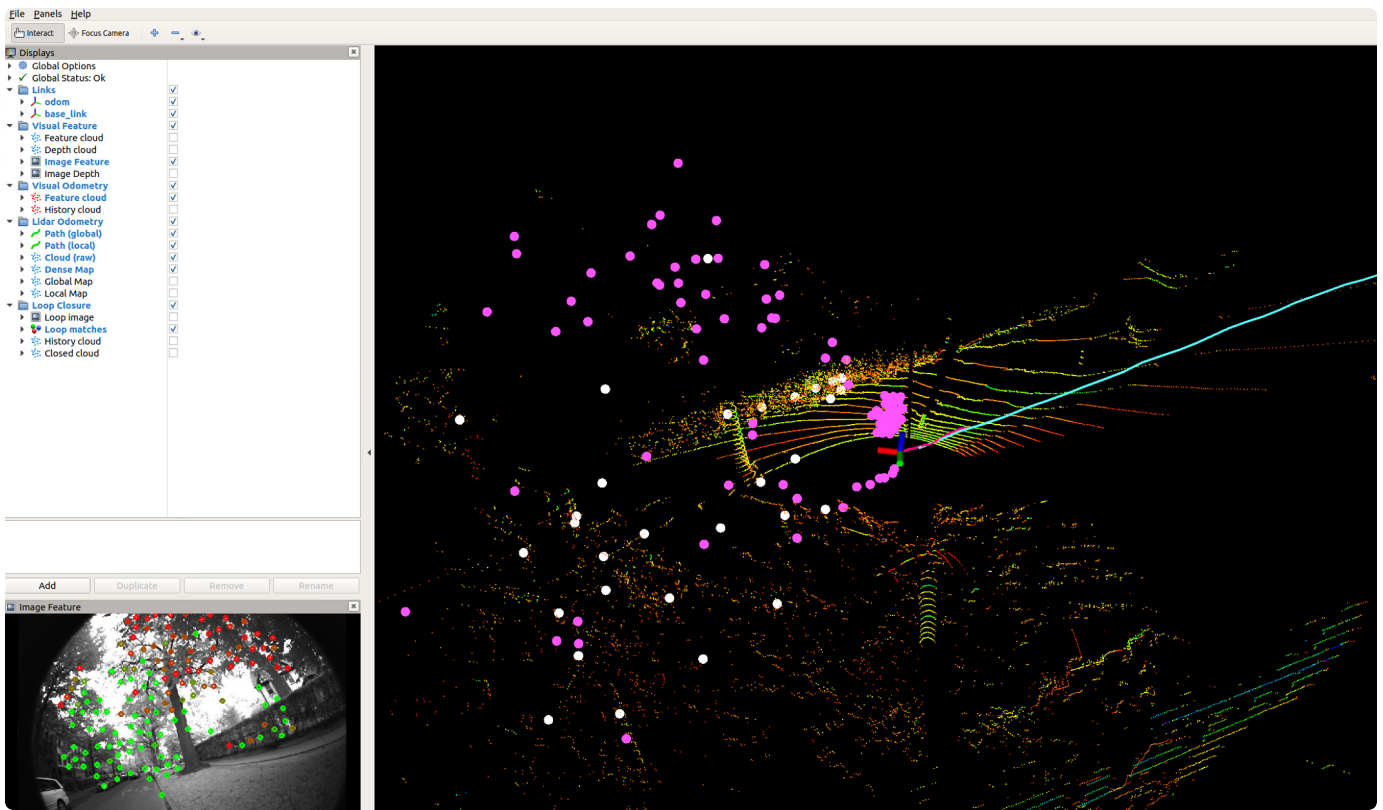


加载成功如下图所示，一键实现将所有需要的节点都订阅



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！

## 7.重新进入docker与关闭镜像

上述操作已经实现了docker镜像的第一次进入，后续再进入，需要先查看此docker镜像的ID和状态

```
1 docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e70dc7aa1878	liangqinli/slam-docker:vl.2	"/ros_entrypoint.sh ..."	18 minutes ago	Exited (0) 1 second ago		inspiring_almeida
f3ff45e319db		"/bin/bash"	2 hours ago	Up 2 hours		
92441934283e	carlasim/carla:latest	"/bin/bash"	4 weeks ago	Exited (0) 4 weeks ago		competent_edison
efa9bfcffee7	carlasim/carla	"/bin/sh -c '/bin/ba..."	4 weeks ago	Exited (1) 4 weeks ago		charming_proskuria
kova						
16dde6bd18dc	e2d52c46c575	"/bin/sh -c '/bin/ba..."	4 weeks ago	Exited (1) 4 weeks ago		keen_bouman

如果此镜像是exit状态，需要用最左边红框的CONTAINER\_ID启动（注意下方代码里的CONTAINER\_ID需要替换成自己实际的镜像ID）

```
1 docker start CONTAINER_ID
```

然后exec进入容器



C++

```
1 docker exec -it CONTAINER_ID /bin/bash
```

如需要关闭镜像，采用



C++

```
1 docker stop CONTAINER_ID
```

## 版本更新说明

2.0版增加了对E-LVI-SAM课程代码库的适配

## 番外篇：用此docker运行vins\_mono算法

1. 拉取镜像，操作相同
2. 代码隔离



Plain Text

```
1 mkdir -p /YOUR_PATH/vins_mono_docker/catkin_ws/src
2 cd /YOUR_PATH/vins_mono_docker/catkin_ws/src
3 git clone git@github.com:HKUST-Aerial-Robotics/VINS-Mono.git
```

3. 第一次启动docker镜像，路径替换即可
4. 此docker不支持可视化RVIZ，所以还是在宿主机内启动RVIZ，注意；启动rviz后，加载 config/vins\_rviz\_config.rviz得到原有的订阅节点
5. 保留原有的第1 3步，删去了原有的第2步 (roslaunch vins\_estimator vins\_rviz.launch)，转为在 docker外启动rviz



▼ bash

Plain Text |

```
1 // docker内执行
2 roslaunch vins_estimator euroc.launch
3
4 // 宿主机内执行
5 rviz
6
7 // docker内执行 or 宿主机内执行 都可
8 rosbag play YOUR_PATH_TO_DATASET/MH_01_easy.bag
```

## 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