作者: 293311923@qq.com
URL: https://github.com/snowflowersnowflake/lsd_slam

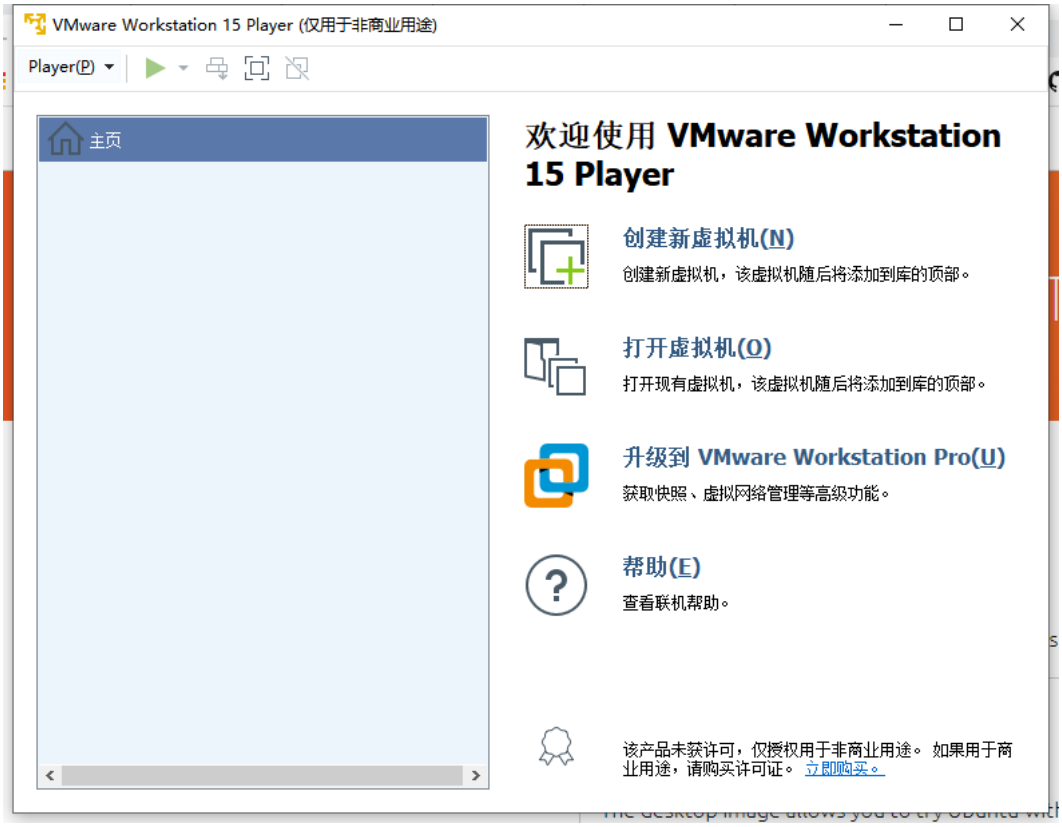
> 该教程是从 Ubuntu14 中安装 ros，在ros下安装lsdslam，因此也包括prework所需的教程

什么是lsdslam

LSD-SLAM是一种新型的方法用于实现实时单目视觉slam。这是种完全直接创建大型、半密集地图的方案，而不用依赖关键点或者特征，**是slam中直接法的实现**。你甚至能在笔记本上直接完成lsdslam。

在虚拟机上安装最新Ubuntu

- VM Workstations 15 Player:

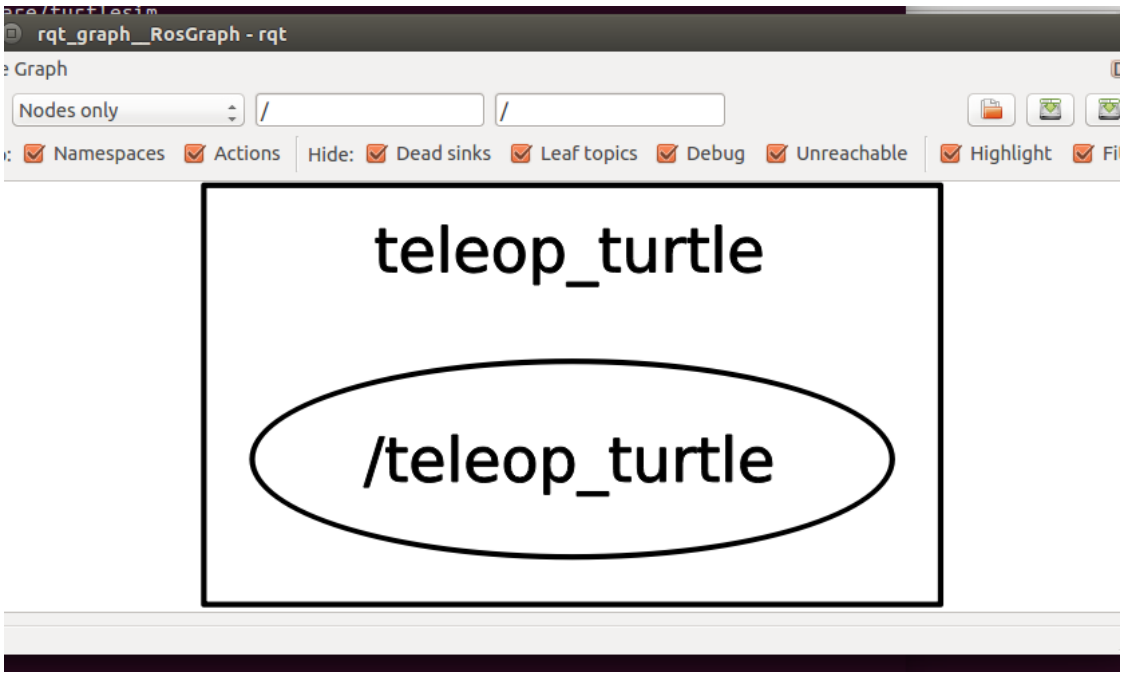


安装ROS

[安装第八代ros发行版本：ROS Indigo]

```
sudo apt install curl（curl用于安装ros）

参考官方网站（可以用Ubuntu的q-mail，在主机上发送邮件到自己的邮箱，然后在虚拟机上复制代码， linux命令行中复制粘贴是ctrl shift c/v）：

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

curl -sSL 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B4F42ED6FBAB17C654' | sudo apt-key add -

sudo apt-get update && sudo apt-get install dpkg

sudo apt-get install ros-indigo-desktop-full

到这里已经安装完了完整版本的ros-indigo，完整版（full）包括基本版（base）和桌面版（desktop）以及个性化定制包（ individual package ）

接下来初始化rosdep：

sudo rosdep init
rosdep update

安装环境

echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
source ~/.bashrc

安装ros所需要的命令行工具rosinstall
sudo apt-get install python-rosinstall

画个小乌龟测试效果
roscore
rosrun turtlesim turtlesim_node
```

```
rosrun turtlesim turtle_teleop_key
rosrun rqt_graph rqt_graph
```



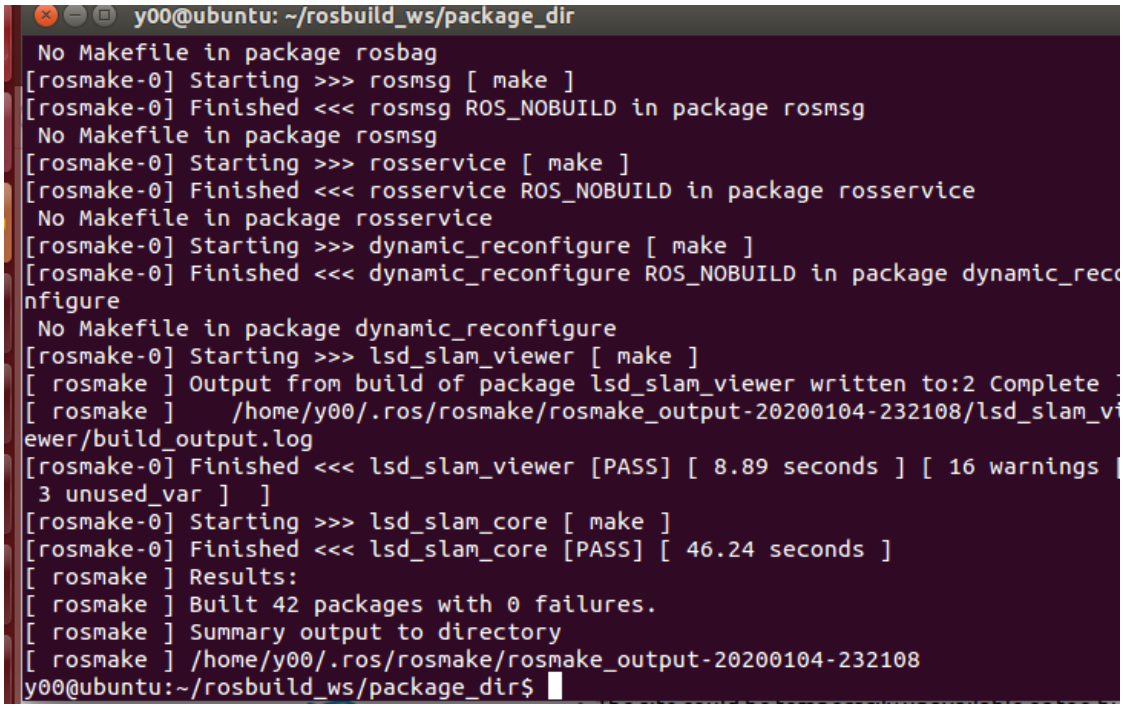
请注意，每次部署运行ros以及相关操作都要使用命令roscore

ROS下安装LSD-SLAM

```
创建工作空间：按顺序复制下面的命令，如果有提示选确认（Y）
sudo apt-get install python-rosinstall
mkdir ~/rosbuild_ws
cd ~/rosbuild_ws


rosws init . /opt/ros/indigo
mkdir package_dir
rosws set ~/rosbuild_ws/package_dir -t .
echo "source ~/rosbuild_ws/setup.bash" >> ~/.bashrc
bash


cd package_dir


在该目录下用apt安装所需要的依赖：
sudo apt-get install ros-indigo-libg2o ros-indigo-cv-bridge liblapack-dev
```

```
libblas-dev freeglut3-dev libqglviewer-dev libsuitesparse-dev libx11-dev

在package中从github clone需要的安装包：
git clone https://github.com/tum-vision/lsd_slam.git lsd_slam

在ros中进行make：
rosmake lsd_slam



报错：ERROR [gendeps] 1 Finding dependencies for
/home/cc/rosbuild_ws/package_dir/lsd_slam/lsd_slam_viewer/cfg/LSDSLAMViewerParams

分析：顾名思义，不应该在描述中出现引号（单引号或双引号），可以通过查看错误内
容得知需要修改的文件的名称。

解决方法：

1) lsd_slam_viewer/cfg/LSDSLAMViewerParams.cfg 第20行scaledDepthVarTH：单词
point's和keyframe's 中出现了单引号，删除单引号 第21行absDepthVarTH：单词
point's 第24行cutFirstNKf：单词 keyframe's

2) lsd_slam_core/cfg/LSDDebugParams.cfg 第11行plotStereoImages：单词 what's
第12行plotTracking：单词 what's 第48行continuousPCOutput：单词 Keyfram's
```

解决这个错误后 已经成功进行rosmake过程

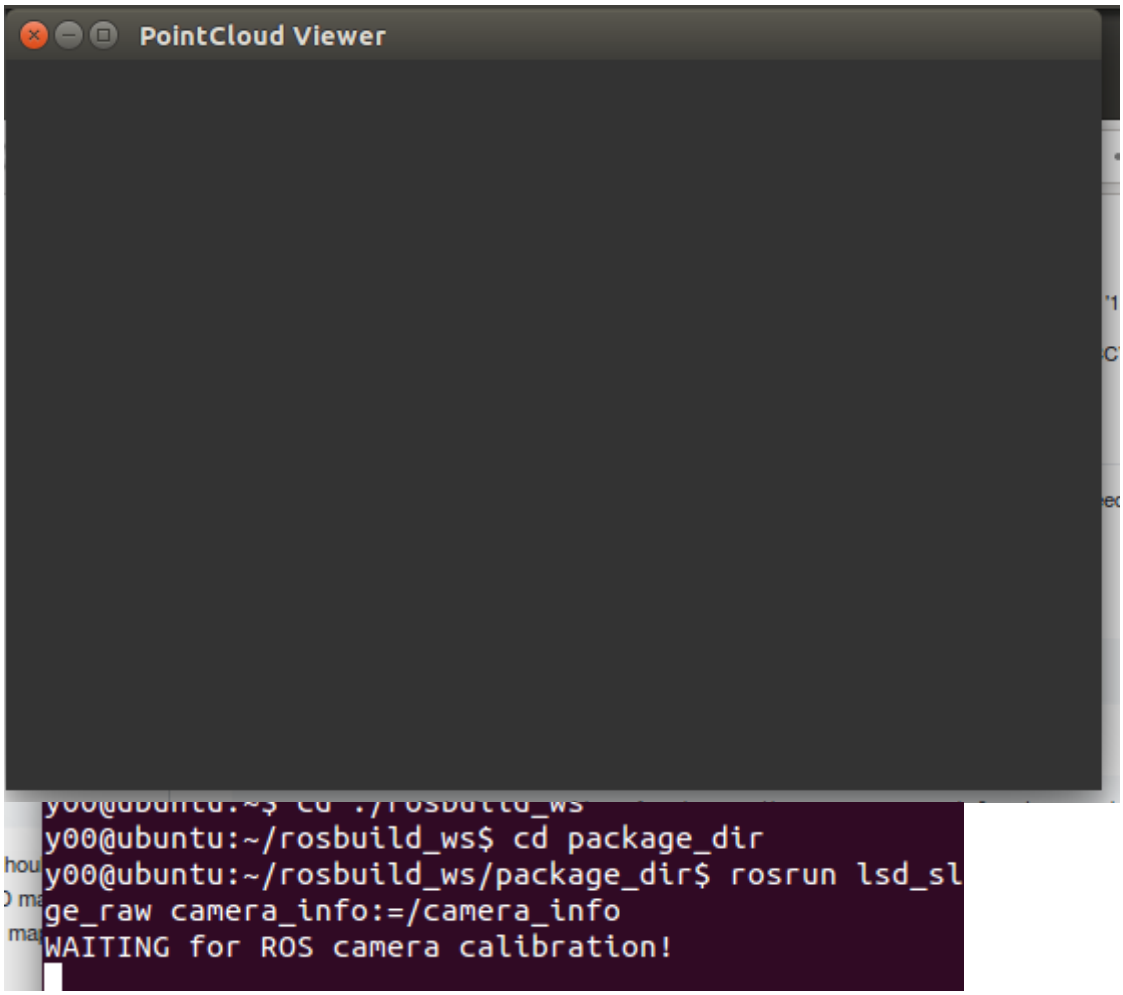


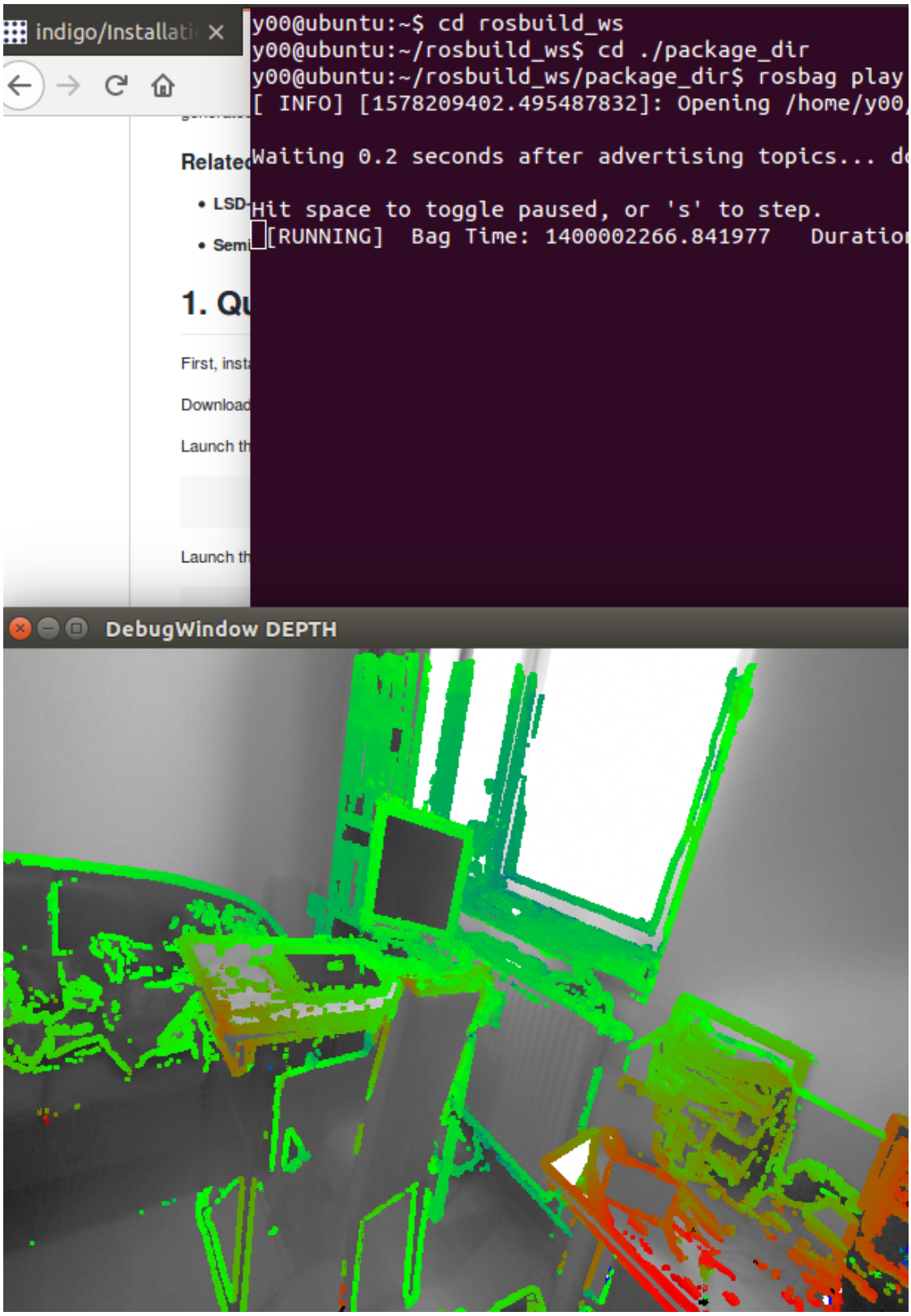
现在我们创建一个样例并且启动试试看

下载这个包并解压到Ubuntu，该样例是一个可以运行在lsdslam下的room的表示。

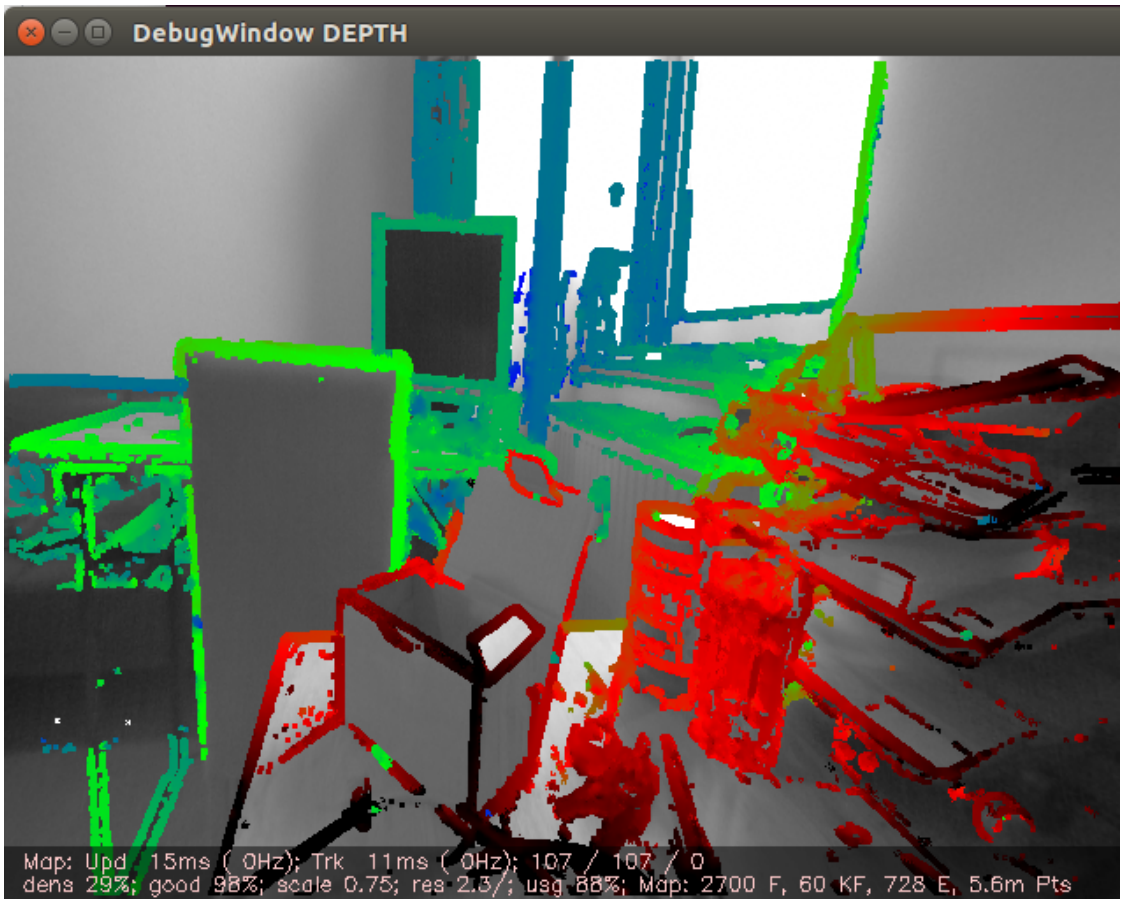
解压room.bag到**主目录**
打开新的命令行

```
rosrun lsd_slam_viewer viewer
生成点云窗口
rosrun lsd_slam_core live_slam image:=/image_raw camera_info:=/camera_info
标定相机内参和外参
rosbag play ~/LSD_room.bag
利用样例重建一个房间模型
```


PointCloud Viewer
y00@ubuntu:~/rosbuild_ws$ cd package_dir
y00@ubuntu:~/rosbuild_ws/package_dir$ rosrun lsd_sl
ge_raw camera_info:=/camera_info
WAITING for ROS camera calibration!
等待ros摄像机标定


indigo/Installati
Related
• LSD-
• Semi
1. Qu
First, insta
Download
Launch th
Launch th
y00@ubuntu:~$ cd rosbuild_ws
y00@ubuntu:~/rosbuild_ws$ cd ./package_dir
y00@ubuntu:~/rosbuild_ws/package_dir$ rosbag play
[INFO] [1578209402.495487832]: Opening /home/y00
Waiting 0.2 seconds after advertising topics... d
Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1400002266.841977 Duratio
DebugWindow DEPTH

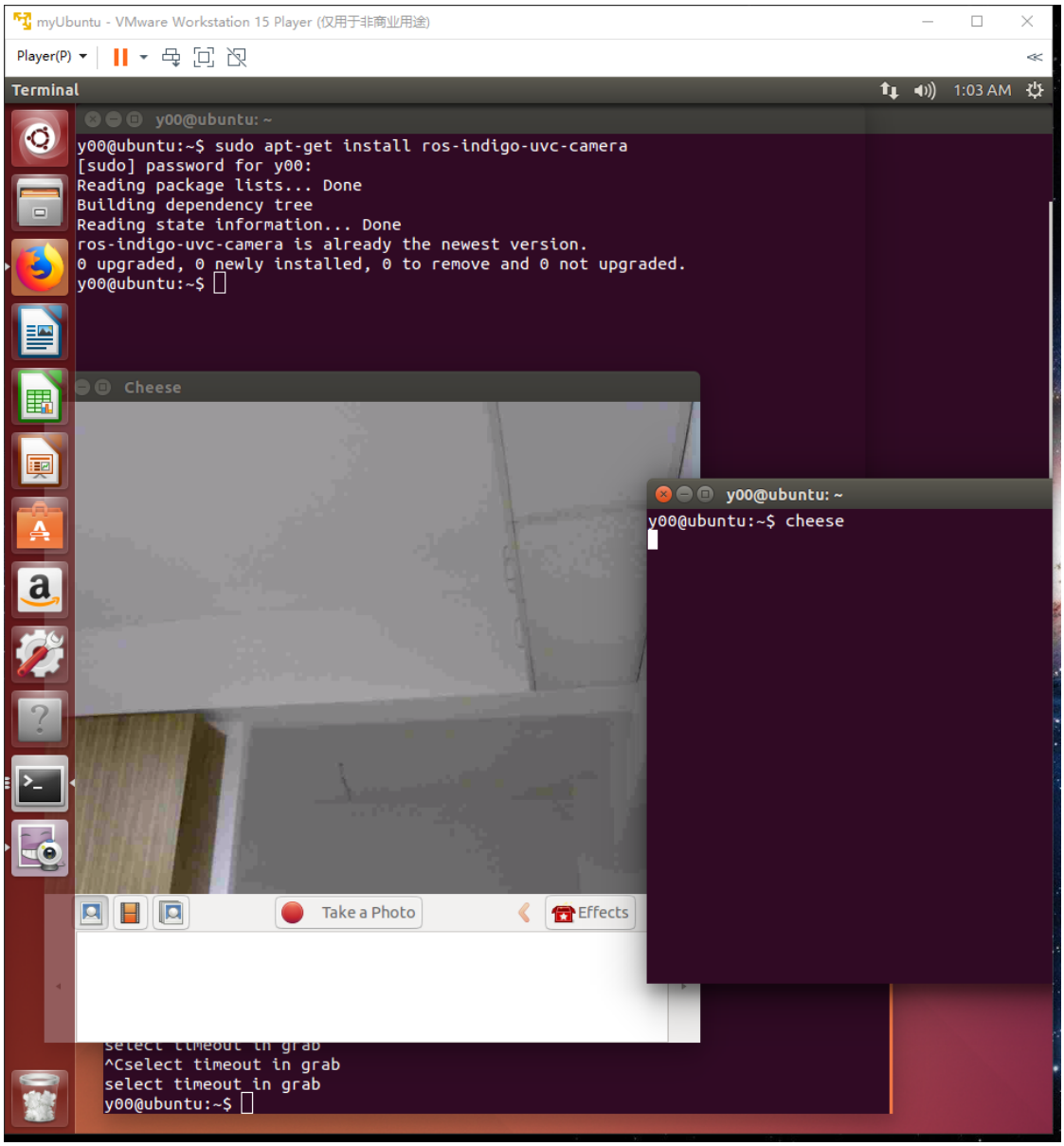
正在生成...



以上是最终效果

利用单目摄像头进行实时的lsdslam

接入摄像头后，通过apt安装cheese测试摄像头是否可用



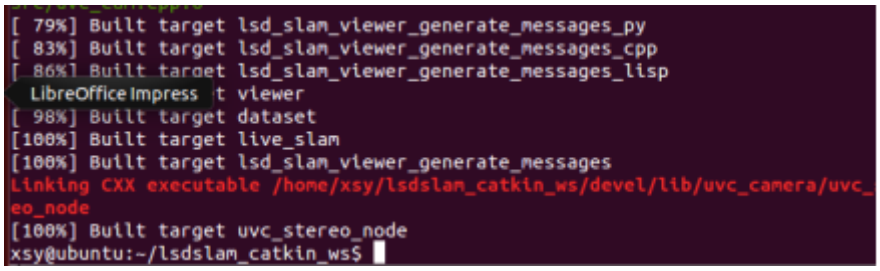
安装uvc_camera驱动：

  $ cd lsdslam_catkin_ws/src

  $ git clone https://github.com/ktossell/camera_umd

  $ cd ..

  $ catkin_make



  $ source devel/setup.bash

```
xsy@ubuntu:~/lsdslam_catkin_ws$ source devel/setup.bash
xsy@ubuntu:~/lsdslam_catkin_ws$
```

测试是否安装成功：

开启三个终端：

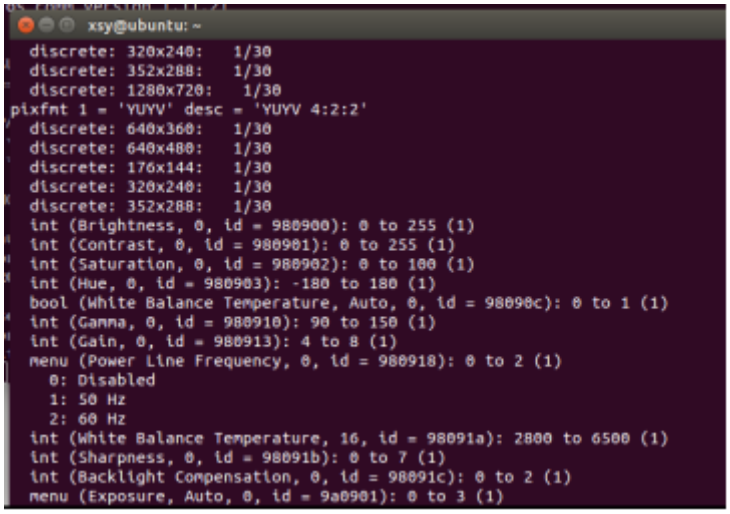
第一个启动ROS服务：

```
$ roscore
```

```
auto-starting new master
process[master]: started with pid [58518]
ROS_MASTER_URI=http://ubuntu:11311/
Ubuntu Software Center
process[rosout-1]: started with pid [58531]
started core service [/rosout]
```

第二个启动驱动：

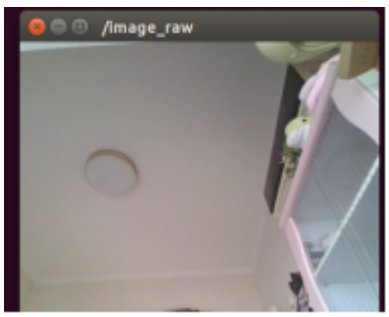
```
$ rosrun uvc_camera uvc_camera_node device:=/dev/video0
```



第三个启动视频窗口：

```
$ rosrun image_view image_view image:=/image_raw
```

如果正常显示摄像头视频即成功.



配置camera_node.launch文件
(在/lsdslam_catkin_ws/src/camera_umd/uvc_camera/launch中)，如：

```
<launch>
    <node pkg="uvc_camera" type="uvc_camera_node" name="uvc_camera"
output="screen">
        <param name="width" type="int" value="640" />
        <param name="height" type="int" value="480" />
        <param name="fps" type="int" value="30" />
        <param name="frame" type="string" value="wide_stereo" />
        <param name="auto_focus" type="bool" value="False" />
        <param name="focus_absolute" type="int" value="0" />
        <!-- other supported params: auto_exposure,
exposure_absolute, brightness, power_line_frequency -->
        <param name="device" type="string" value="/dev/video0" />
        <param name="camera_info_url" type="string"
value="file://$(find uvc_camera)/example.yaml" />
    </node>
</launch>
```

注意：官方程序的默认分辨率是640*480.

然后运行LSD-SLAM:

打开一个终端，运行：

```
$ roscore
```

打开另一个终端（原终端保留），运行：

```
$ cd lsdslam_catkin_ws/
$ source devel/setup.sh
$ rosrun lsd_slam_viewer viewer
```

打开另一个终端（前两个终端保留），运行：

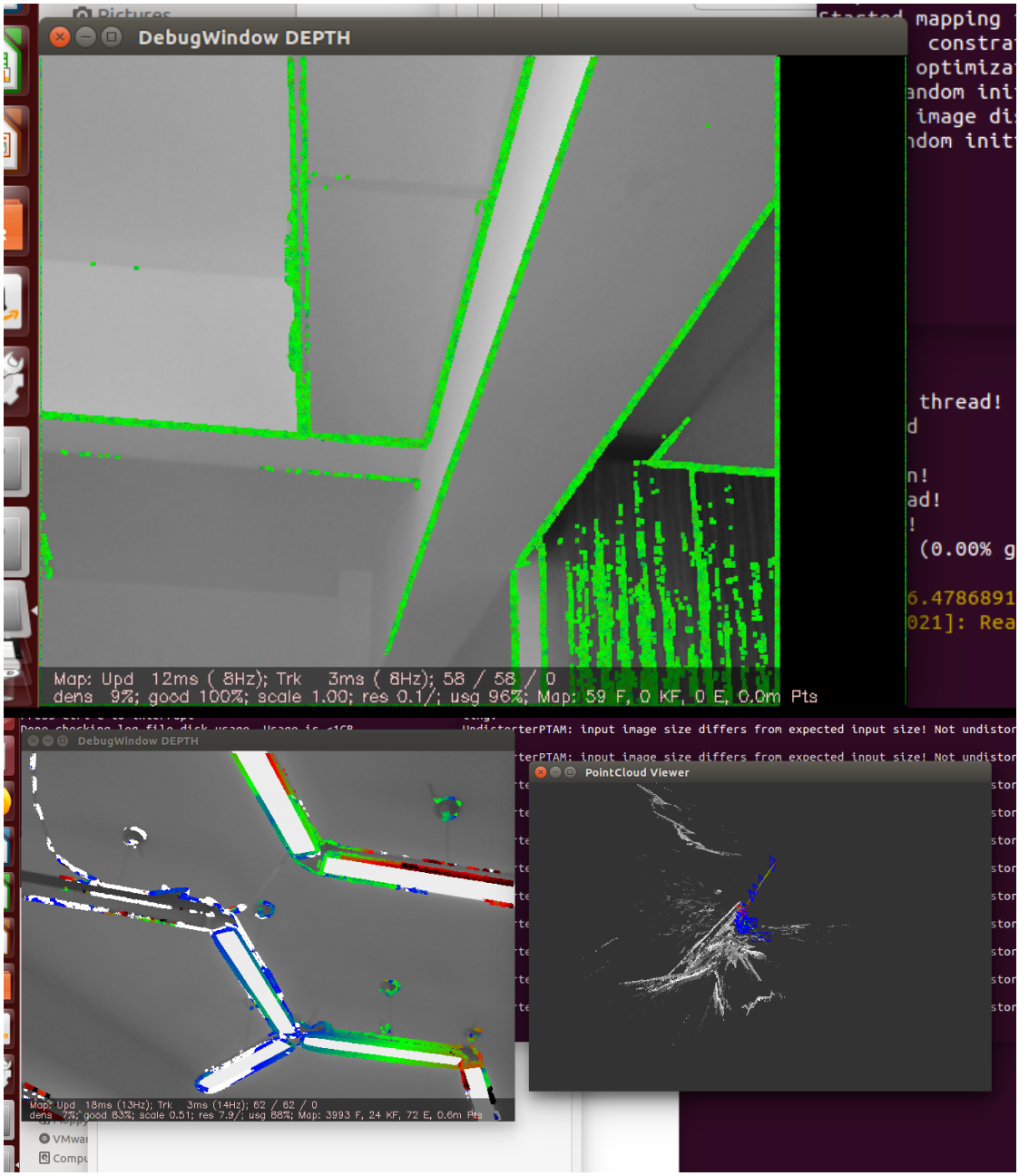
```
$ cd lsdslam_catkin_ws/
$ source devel/setup.sh
$ roslaunch uvc_camera camera_node.launch
```

再次打开一个终端（前三个终端保留），运行：

```
$ rosrun lsd_slam_core live_slam /image:=image_raw
_calib:=FOV_examle_calib.cfg
```

<calibration_file>三个可用的校准文件，任选一个
FOV_examle_calib.cfg 超广角镜头示例校准文件
OpenCV_example_calib.cfg opencv示例校准文件
pinhole_example_calib.cfg 小孔成像示例校准文件

在实验室环境测试：



将运行指令写成自动化测试脚本

> linux下的命令可以写成脚本，一次运行多命令，并且可以进行结构逻辑控制，比如if while等方法

可参考文档：[linux命令大全]

用gedit(或者其他编辑器)创建一个文件 aptC.sh
键入如下命令:

```
#!/bin/sh
echo "processing begin!"
sleep 1
gnome-terminal -t "1" -x bash -c "sh 1.sh;exec bash;"
sleep 3
gnome-terminal -t "2" -x bash -c "sh 2.sh;exec bash;"
sleep 3
gnome-terminal -t "3" -x bash -c "sh 3.sh;exec bash;"
sleep 5
gnome-terminal -t "4" -x bash -c "sh 4.sh;exec bash;"
echo "succ!!"
```

创建脚本1.2.3.4,编写下列代码

```
#!/bin/sh
echo "ros engining!"
roscore
echo "proc!succ!!"
```

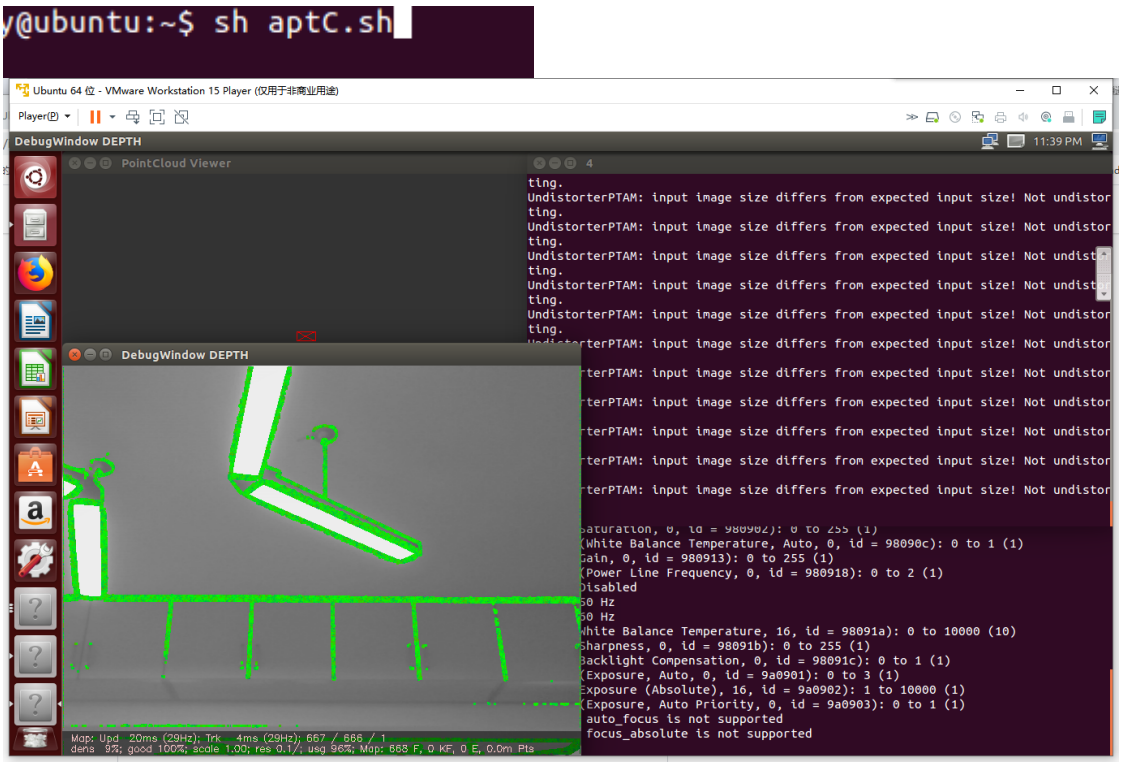
```
#!/bin/sh
echo "point viewer begin!"
cd lsdslam_catkin_ws/
source devel/setup.sh
rosrun lsd_slam_viewer viewer
echo "proc2!succ!!"
```

```
#!/bin/sh
echo "camera work!"
cd lsdslam_catkin_ws/
source devel/setup.sh
roslaunch uvc_camera camera_node.launch
echo "proc3!succ!!"
```

```
#!/bin/sh
echo "slam work!"
rosrun lsd_slam_core live_slam /image:=image_raw _calib:=FOV_examle_calib.cfg
echo "proc4!succ!!"
```

这表示按照顺序用sh打开脚本1,2,3,4，中间间隔一个sleep时间

运行：



reference

[1]LSD-SLAM: Large-Scale Direct Monocular SLAM, J. Engel, T. Schöps, D. Cremers, ECCV '14

[2]Semi-Dense Visual Odometry for a Monocular Camera, J. Engel, J. Sturm, D. Cremers, ICCV '13

[3]https://github.com/tum-vision/lsd_slam