**$一、本周工作**

**一、实验平台搭建步骤撰写**

**1．ubuntu18.04系统安装**

1.官网下载链接：<https://releases.ubuntu.com/18.04/>

2.安装教程：[https://blog.csdn.net/whatiscode/article/details/109153474?ops\_request\_misc=&request\_id=&biz\_id=102&utm\_term=ubuntu18.04%E5%AE%89%E8%A3%85%E6%95%99%E7%A8%8B&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduweb~default-3-109153474.142^v73^control,201^v4^add\_ask,239^v2^insert\_chatgpt&spm=1018.2226.3001.4187](https://blog.csdn.net/whatiscode/article/details/109153474?ops_request_misc=&request_id=&biz_id=102&utm_term=ubuntu18.04%E5%AE%89%E8%A3%85%E6%95%99%E7%A8%8B&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-109153474.142%5Ev73%5Econtrol,201%5Ev4%5Eadd_ask,239%5Ev2%5Einsert_chatgpt&spm=1018.2226.3001.4187)

**2、安装UBUNTU毫米波环境**

整个流程参考：<https://dev.ti.com/tirex/explore/node?node=A__AFIEVaaBoBCgo.VCtJenZQ__radar_toolbox__1AslXXD__LATEST>

**（一） ROS\_Melodic安装——ubuntu18.04**

1.ubuntu16.04与ros\_kinetic匹配， ubuntu18.04与 ros\_melodic匹配

2.按照官网链接安装melodic：<http://wiki.ros.org/melodic/Installation/Ubuntu>

3.报错解决：

（1）

sudo apt install curl # if you haven't already installed curl

curl -s <https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc> | sudo apt-key add

报错解决：[https://blog.csdn.net/lsyhaoshuai/article/details/128076106?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522167756925516800225592064%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=167756925516800225592064&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v1~rank\_v31\_ecpm-1-128076106-null-null.142^v73^control,201^v4^add\_ask,239^v2^insert\_chatgpt&utm\_term=ubuntu18.04%20ros%E5%AE%89%E8%A3%85gpg%3A%20%E6%89%BE%E4%B8%8D%E5%88%B0%E6%9C%89%E6%95%88%E7%9A%84%20OpenPGP%20%E6%95%B0%E6%8D%AE%E3%80%82&spm=1018.2226.3001.4187](https://blog.csdn.net/lsyhaoshuai/article/details/128076106?ops_request_misc=%7B%22request%5Fid%22%3A%22167756925516800225592064%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=167756925516800225592064&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-1-128076106-null-null.142%5ev73%5econtrol,201%5ev4%5eadd_ask,239%5ev2%5einsert_chatgpt&utm_term=ubuntu18.04%20ros安装gpg%3A%20找不到有效的%20OpenPGP%20数据。&spm=1018.2226.3001.4187)

（2）sudo rosdep init 报错解决： <https://blog.csdn.net/wjjwangde/article/details/109808371>

（3）rosdep update 报错解决：<https://blog.csdn.net/weixin_44023934/article/details/121242176>

**（二）ti\_mmwave\_rospkg ROS驱动安装和使用**

1.报错解决 遇到的问题主要出现在Running the driver 部分，对于TI IWR6843ISK-ODS，对应的启动命令为：

roslaunch ti\_mmwave\_rospkg 6843\_multi\_3d\_0.launch

在每次启动时候，都需要使用如下三条命令设置bash，但是直接运行上述命令出现如下报错：

source /opt/ros/melodic/setup.bash  
  
source workspace\_dir/mmwave\_ti\_ros/ros\_driver/devel/setup.bash #workspace\_dir 换成自己工作目录  
  
roslaunch ti\_mmwave\_rospkg 6843\_multi\_3d\_0.launch

报错如下：

RLException: [6843\_multi\_3d\_0.launch] is neither a launch file in package [ti\_mmwave\_rospkg] nor is [ti\_mmwave\_rospkg] a launch file name  
  
The traceback for the exception was written to the log file

解决方法：<https://blog.csdn.net/skye_fly/article/details/108091954>

将source /opt/ros/melodic/setup.bash和source workspace\_dir/mmwave\_ti\_ros/ros\_driver/devel/setup.bash 写入bashrc文件中。 之后每次启动，直接运行roslaunch ti\_mmwave\_rospkg 6843\_multi\_3d\_0.launch即可。

**3.Azure-Kinect-Sensor-SDK安装——支持的操作系统为Ubuntu18.04**

1.使用apt方式安装 （看到有的博客是用源码编译的方式进行安装的）

1.1 先配置Microsoft的包存储库

官网链接：<https://docs.microsoft.com/zh-cn/windows-server/administration/linux-package-repository-for-microsoft-software>

运行：

curl -sSL https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -  
​  
sudo apt-add-repository https://packages.microsoft.com/ubuntu/18.04/prod  
​  
sudo apt-get update

1.2 安装SDK 官网链接：<https://docs.microsoft.com/zh-cn/azure/Kinect-dk/sensor-sdk-download> 运行：

sudo apt install k4a-tools  
​  
sudo apt install libk4a1.1-dev  
​  
sudo apt install libk4a1.4-dev

回到主目录

git clone -b v1.4.0 https://github.com/microsoft/Azure-Kinect-Sensor-SDK.git  
​  
下载依赖项：在~/Azure-Kinect-Sensor-SDK/scripts/docker 路径下找到setup-ubuntu.sh文件，右击->属性->权限->

勾选允许作为程序执行文件

cd Azure-Kinect-Sensor-SDK/scripts/docker  
​  
sudo chmod 777 setup-ubuntu.sh  
​  
sudo sh setup-ubuntu.sh  
​  
出现了一个Syntax error: “(” unexpected 的报错，通过以下命令解决sudo dpkg-reconfigure dash   选择NO就行

1.3 运行k4aiewer进行安装测试

sudo k4aviewer

1.4 取消root权限启动

将Azure-Kinect-Sensor-SDK/scripts/99-k4a.rules文件，复制到/etc/udev/rules.d/ 目录中

cd ~/Azure-Kinect-Sensor-SDK

sudo cp scripts/99-k4a.rules /etc/udev/rules.d/.

**4. Azure\_Kinect\_ROS\_Driver安装和使用**

1.创建工作空间

mkdir -p ~/catkin\_ws/src  
​  
cd ~/catkin\_ws/src  
​  
catkin\_init\_workspace  
​  
cd ..  
​  
catkin\_make  
​  
echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc

1. 下载 package

cd catkin\_ws/src

git clone <https://github.com/microsoft/Azure_Kinect_ROS_Driver.git>

3.修改 k4a\_ros\_device.h

路径：Azure\_Kinect\_ROS\_Driver/include/azure\_kinect\_ros\_driver/k4a\_ros\_device.h

cd ~/catkin\_ws/src/Azure\_Kinect\_ROS\_Driver/include/azure\_kinect\_ros\_driver/

gedit k4a\_ros\_device.h

将下面-的内容替换为+：

// Last capture timestamp for synchronizing playback capture and imu thread

  -   std::atomic\_int64\_t last\_capture\_time\_usec\_;  
    
  +   std::atomic<int64\_t> last\_capture\_time\_usec\_;

// Last imu timestamp for synchronizing playback capture and imu thread

  -   std::atomic\_uint64\_t last\_imu\_time\_usec\_;  
    
  -   std::atomic\_bool imu\_stream\_end\_of\_file\_;  
    
  +   std::atomic<uint64\_t> last\_imu\_time\_usec\_;  
    
  +   std::atomic<bool> imu\_stream\_end\_of\_file\_;

4.编译

cd catkin\_ws  
​  
catkin\_make

5.测试

#启动  
​  
roslaunch azure\_kinect\_ros\_driver driver.launch  
​  
# Global Fixed Frame: rgb\_camera\_link  
​  
#新建终端控制台，输入下面的命令，出现ros界面  
​  
rviz  
​  
#查看topic  
​  
rostopic list

**二、数据采集流程**

1.将雷达和Kinect使用数据线接入PC机

2.启动

#新建终端，启动毫米波雷达命令  
roslaunch ti\_mmwave\_rospkg 6843\_multi\_3d\_0.launch  
#新建终端，启动Kinect命令  
roslaunch azure\_kinect\_ros\_driver driver.launch

3.启动成功后，可以在RVIZ界面看到毫米波雷达的点云图像，依次点击【Add】—>【By display type】—>【Image】，之后在界面左侧【Displays】区域中将增加【Image】选项，展开该选项，点击【Image Topic】设置要显示的topic，若成功则在下方image窗口出现相应topic的图像。

4.录制topic数据

新建终端，使用如下命令查看当前所有的ROS Topic

rostopic list

对于点云数据，对应的topic为：/ti\_mmwave/radar\_scan\_pcl\_0，对于kinect数据，存在多个topic，可根据需要录取相应的Topic。示例命令如下：

#录制30s的/ti\_mmwave/radar\_scan\_pcl\_0和/rgb/image\_raw数据，并重命名为test，最终得到test.bag文件  
rosbag record /ti\_mmwave/radar\_scan\_pcl\_0 /rgb/image\_raw --duration=30 -O test

之后，使用点云数据处理代码，生成matlab数据文件，数据处理完毕。

**三、雷达标定**

* 关于时空同步的问题：使用同一电脑采雷达和Kinect数据，是不是不需要用到时间同步？
* 关于空间同步，目前正在实验中，这周手动标定感觉做的不好，打算尝试下自动标定的方法，以下是找到的资料，作为本周的任务：

程序使用Kinect Azure DK和IWR6843ISK雷达的数据进行标定，并生成标定参数文件。

首先，将自动标定程序的软件包下载到您的工作空间，然后使用catkin\_make构建该软件包：

bash

cd ~/catkin\_ws/src

git clone <https://github.com/RobustFieldAutonomyLab/spatial-calibration.git>

cd ..

catkin\_make

然后，启动自动标定程序节点：

roslaunch spatial\_calibration automatic\_calibration.launch

在启动程序后，将Kinect Azure DK和IWR6843ISK雷达对准彼此，并确保它们可以看到相同的场景。程序将运行数秒钟，然后输出标定参数文件的路径。您可以将该文件用于将Kinect Azure DK和IWR6843ISK雷达的点云数据对齐。

注意：在运行程序之前，需要将雷达和Kinect Azure DK的位置和朝向进行粗略估计，以确保它们在同一平面上。

**其他工作**

协助梅博士改写综述；

**$二、下周工作**

* 进行雷达空间标定
* 和天顺师兄采数据
* 进一步学习数据预处理代码