

Realsense d435i 标定, imu utils, Kalibr 工具, 官网 IMU 标定方法

D435i:官方标定教程,标定 Imu 的 x,y,z 三个轴的 align matrix 和 bias

imu utils: 标定相机的 imu 随机游走,https://github.com/gaowenliang/imu_utils

Kalibr 工具:标定相机的外参和相机的联合标定(imu + camera)

具体步骤如下:

1 标定 IMU

1.1 按照官方教程标定相机的 bias 和三个轴的对齐:具体的标定数据可以参考官方文档

https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/RealSense_Depth_D435i_IMU_Calib.pdf

1.1.1 下载安装 librealsense 官方的

SDK:<https://github.com/IntelRealSense/librealsense>, 安装 pyrealsense2 安装包,和相应的依赖包.pip 安装的时候最好用清华源.

1.1.2 执行 python rs-imu-calibration.py,rs-imu-calibration.py 这个文件在 librealsense/tools/rs-imu-calibration 文件夹下

1.1.3 数据采集,并矫正 imu ,具体的步骤我是严格按照

<https://github.com/IntelRealSense/librealsense/tree/master/tools/rs-imu-calibration>,指示做的.结果为9.8046.最好弄个盒子,因为对6个面进行数据采集,还是需要点时间的.手持的话还是比较累,而且手容易动,导致要重新采集.弄个盒子还准一些.将标定的结果写入到相机.

Would you like to write the results to the camera?是否将标定的结果写入 IMU?
输入 Y, 表示将标定的结果是写入到 IMU。

1.2 标定随机游走

1.2.1 这里使用 imu_utils 标定 imu, 在安装 imu_utils 之前, 要先安装 ceres-solver 和 code_utils, ceres-solver 安装见 [Installation – Ceres Solver](#), 下面是 code_utils 的安装过程, 注意要安装 code_utils 后再下载 imu_utils

1.2.2 `cd ~/catkin_ws/src git clone https://github.com/gaowenliang/code\_utils cd ~/catkin_ws catkin_make`

编译时可能出现以下错误:

- error 1

fatal error: backward.hpp:No such file or directory

解决方法：将 `sumpixel_test.cpp` 中的 `#include "backward.hpp"` 修改为 `#include "code_utils/backward.hpp"` 即可

- error 2

`code_utils/backward.hpp:216:30: fatal error: elfutils/libdw.h: No such file or directory`

执行 `apt-get install libdw-dev` 安装对应的库即可解决

1.2.3 接下来安装 imu_utils

`cd ~/catkin_ws/src` git clone https://github.com/gaowenliang/imu_utils

`cd ~/catkin_ws`

`catkin_make`

1.2.4 安装完成后开始标定，我没有采用录制 imu 话题再回放 bag 文件的方式，而是直接现场发布 imu 的话题，同时运行标定的节点

1.2.4.1 编写一个标定的 launch 文件，命名 `my_calibration_imu_d435i.launch`，执行该 launch 文件，将 RealsenseD435i 静止摆放 1 个小时

```
<launch> <node pkg="imu_utils" type="imu_an" name="imu_an" output="screen">
<param name="imu_topic" type="string" value= "/camera/imu"/> #imu topic 的名字
<param name="imu_name" type="string" value= "d435i"/> <param
name="data_save_path" type="string" value= "${find imu_utils)/data/" /> <param
name="max_time_min" type="int" value= "60"/> #标定的时长 <param
name="max_cluster" type="int" value= "100"/> </node> </launch>
```

1.2.4.2 执行下面的命令可以发布带有 imu 的话题

```
roslaunch realsense2_camera rs_camera.launch
unite_imu_method:="linear_interpolation" roslaunch imu_utils
d435i_imu_calib.launch
```

会在 `~/catkin_ws/src/imu_utils/data/` 生成一文
件：`d435i_imu_calibration_imu_param.yaml` 内容如下：

```
ziju@ziju-MS-7A94:~/software/calib/imu/catkin_ws$ cat
src/imu_utils/data/d435i_imu_calibration_imu_param.yaml
```

%YAML:1.0

type: IMU

name: d435i_imu_calibration

Gyr:

unit: " rad/s"

avg-axis:

gyr_n: 4.1077629923569163e-03

gyr_w: 1.1022906135776512e-04

x-axis:

gyr_n: 2.4922772268248078e-03

gyr_w: 2.8455840527384030e-05

y-axis:

gyr_n: 6.8940750517066697e-03

gyr_w: 2.4181215767687077e-04

z-axis:

gyr_n: 2.9369366985392714e-03

gyr_w: 6.0419185869040582e-05

Acc:

unit: " m/s^2"

avg-axis:

acc_n: 2.7338232693468374e-02

acc_w: 8.4873152096528635e-04

x-axis:

acc_n: 2.1346669274695401e-02

acc_w: 6.4225369682846837e-04

y-axis:

acc_n: 2.8062624515000953e-02

acc_w: 6.3350006384460316e-04

z-axis:

acc_n: 3.2605404290708760e-02

acc_w: 1.2704408022227878e-03

其中 : Gyr:avg-axis 和 Acc:avg-axis:是将来联合标定的时候要用到(3.1 中).

2 标定 cam0 的内参

2.1. 相机内参标定

安装 Kalibr 工具

2.2 配置依赖项

```
sudo apt-get install python-setuptools sudo apt-get install python-setuptools
python-rosinstall ipython libeigen3-dev libboost-all-dev doxygen sudo apt-get install
ros-kinetic-vision-opencv ros-kinetic-image-transport-plugins
ros-kinetic-cmake-modules python-software-properties
software-properties-common libpoco-dev python-matplotlib python-scipy python-git
python-pip ipython libtbb-dev libblas-dev liblapack-dev python-catkin-tools
libv4l-dev
```

2.3 新建工作空间并初始化

```
mkdir -p ~/kalibr_ws/src cd ~/kalibr_ws/src catkin_init_workspace cd .. catkin_make
```

2.4 下载 Kalibr 包，放在新建工作空间的 src 文件夹下，然后编译

```
catkin_make
```

编译的时候需要联网，这个注意。编译的时候时间比较长，需耐心等待。

2.5 安装成功后，使用 Kalibr 生成标定版

从官网下载 Kalibr 标定纸: <https://github.com/ethz-asl/kalibr/wiki/downloads>.用 A4 纸打印,粘贴到墙上

2.6 编写 yaml 文件

生成或下载标定板之后, 需要自己编写 target.yaml 文件, 如下代码(注意配置文件中 targetCols 和 targetRows 是内角点的数量, 即使用语句生成标定板时候的 nx 和 ny), 由于 A4 纸大小限制, 我打印出来的每个正方形大小是 0.029m:

```
~/kalibr_ws/src/Kalibr/data$ gedit checkerboard.yaml #新建一个 yaml 文件,我用的,aprilgrid :size = 5.5cm ,spacing=1.65cm,6*6 ,缩放 40%后打印,参数入下
```

```
target_type: 'aprilgrid' #gridtype tagCols: 6
```

```
#number of apriltags tagRows: 6 #number of apriltags
```

```
tagSize: 0.022 #size of apriltag, edge to edge [m]
```

```
tagSpacing: 0.3 #ratio of space between tags to tagSize
```

2.7 录制 bag 包

2.7.1 运行 realsense 相机

```
roslaunch realsense2_camera rs_camera.launch
```

将话题重命名为 color, 并利用 throttle 工具降低录制 RGB 图像的频率至 4Hz

```
roslaunch topic_tools throttle messages /camera/color/image_raw 4.0 /color
```

1. 进入自己想要保存包的文件夹, 然后用 rosbag 录制, ctrl+c 结束录制
2. -O 为重命名包的名字, 这里命名为 camd435i :
3. rosbag record -O camd435i /color

注意: 录制的过程中调整相机的角度, 各个方向都采集一下, 但是拍到的棋盘格要完整(可以使用打开 rviz 便于查看相机捕获的图像), 否则在标定的时候会提示 **Detection failed ! **

2.7.2 进行标定

```
kalibr_calibrate_cameras --target ~/kalibr_ws/src/Kalibr/data/checkerboard.yaml  
--bag ~/kalibr_ws/src/Kalibr/data/camd435i.bag --models pinhole-equi --topics  
/color --show-extractio
```

但是最后报如下错误：Cameras are not connected through mutual observations, please check the dataset. Maybe adjust the approx. sync. tolerance.

参考解决办法：在 Kalibr 官网的 [issue](#) 中找到了一个解决办法，试一下看行不行。

***Comment this code on line 201 of
src/Kalibr/aslam_offline_calibration/kalibr/python/kalibr_calibrate_cameras to
solve your problem:***

```
if not graph.isGraphConnected(): obsdb.printTable() print "Cameras are not  
connected through mutual observations, please check the dataset. Maybe adjust the  
approx. sync. tolerance." graph.plotGraph() sys.exit(-1) 12345
```

把 kalibr_calibrate_cameras 文件中的以上几行注释掉 (**python 用#注释**)看样子是行了,得到的 yamI 文件就是相机的一些参数信息,这次标定的相机的参数如下(即 yamI 文件中的内容), 供大家参考:

```
cam0: cam_overlaps: [] camera_model: pinhole distortion_coeffs:  
[0.3569710914237566, 1.9053508914873958, -11.81782871637401,  
20.693085847996446] distortion_model: equidistant intrinsics: [612.7191756727211,  
614.9337597592461, 318.51127717080993, 238.7278753633164] resolution: [640,  
480] rostopic: /color
```

关于标定的结果

[Kalibr 标定 Intel D435i 相机](#)重投影误差的值越小越好, 我的是 **0.163** 和 **0.138**, 精度应该够了。

[Realsense d435i 驱动安装以及 kalibr 和 imu_utils 标定\[Ubuntu 16 LTS\]](#)这个帖子的最后也有一些出现的问题和解决办法, 我没有遇到, 大家有遇到的可以看看, 避免踩坑

3 相机和 IMU 的联合标定

3.1 编写 camchain.yaml 和 IMU.yaml 文件

将 intrinsics 和 distortion_coeffs 替换成你刚才标定好的相机参数, 我的 camchain.yaml 内容如下:

```
cam0: camera_model: pinhole intrinsics: [612.7191756727211, 614.9337597592461,  
318.51127717080993, 238.7278753633164] distortion_model: equidistant  
distortion_coeffs: [0.3569710914237566, 1.9053508914873958,
```

```
-11.81782871637401, 20.693085847996446] T_cam_imu:#先验,后期优化 -  
[0.01779318, 0.99967549,-0.01822936, 0.07008565] - [-0.9998017, 0.01795239,  
0.00860714,-0.01771023] - [0.00893160, 0.01807260, 0.99979678, 0.00399246] -  
[0.0, 0.0, 0.0, 1.0] timeshift_cam_imu: -8.121e-05 rostopic: /color resolution: [640,  
480]
```

编写 imu.yaml 文件,将 1.2 中生成的,d435i_imu_calibration_imu_param.yaml 文件中的随机游走和噪声,拷贝过来

```
rostopic: /imu update_rate: 200.0 #Hz accelerometer_noise_density:  
2.7338232693468374e-021 #continous accelerometer_random_walk:  
8.4873152096528635e-04 gyroscope_noise_density: 4.1077629923569163e-03  
#continous gyroscope_random_walk: 1.1022906135776512e-04
```

3.2 启动相机,插到电脑上

3.3 发布新话题和修改频率

参考: [Kalibr 标定 Intel D435i 相机](#) 步骤大致和相机标定的步骤一样,不过变成处理两个话题用 throttle 将图像频率降低为 20HZ, imu 频率设置为 200Hz,并发布新的话题

```
roslaunch topic_tools throttle messages /camera/color/image_raw 20.0 /color  
roslaunch realsense2_camera rs_camera.launch  
unite_imu_method:="linear_interpolation"
```

3.4 录制 rosbag 包

```
roslaunch rosbag record -O camimu /color /imu
```

3.5 标定:该步骤耗时有点长,耐心等待

```
kalibr_calibrate_imu_camera --target  
~/kalibr_ws/src/Kalibr/data/checkerboard.yaml --cam  
~/kalibr_ws/src/Kalibr/data/camchain.yaml --imu  
~/kalibr_ws/src/Kalibr/data/imu.yaml --bag ~/kalibr_ws/src/Kalibr/data/camimu.bag  
--show-extraction
```

error:

OverflowError: bad numeric conversion: positive overflow

解决办法: imu 产生了问题,从新录制一份