激光+imu标定

（编译工具源码+标定过程）

源码地址

https://github.com/BHJX05/sensor_calibration/tree/master/lidar_imu_calibr

该仓库下代码已经修改好，无需修改

xavier安装配置并测试激光雷达与IMU标定算法lidar_align

代码地址：https://github.com/ethz-asl/lidar_align

开发平台：nvidia agx xavier +ubuntu18.04+ROS Melodic

1.源码安装编译：

注意，git clone时git本仓库下的src目录下的全部文件，放在catkin工程src目录下一起编译

```
mkdir -p lidar_align_ws/src
cd lidar_align_ws/src
git clone <https://github.com/ethz-sl/lidar_align>
cd ..
catkin_make
```

编译过程中，报错如下：

问题一：编译时出现Could not find NLOPTConfig.cmake

解决方法：下载 安装非线性优化库nlopt

```
sudo apt-get install libnlopt-dev
```

问题二：编译遇到lz4声明冲突：/usr/include/lz4.h conflict with /usr/include/flann/ext/lz4. 参考github上的issue：

```
sudo mv /usr/include/flann/ext/lz4.h /usr/include/flann/ext/lz4.h.bak
sudo mv /usr/include/flann/ext/lz4hc.h /usr/include/flann/ext/lz4.h.bak
sudo ln -s /usr/include/lz4.h /usr/include/flann/ext/lz4.h
sudo ln -s /usr/include/lz4hc.h /usr/include/flann/ext/lz4hc.h
```

2.录制传感器数据包

编译通过之后，录制激光雷达和IMU数据，录制过程中，需要注意以下几点：

1）在一个小空间里面精度会更高

2）空间中尽可能多一些规整的平面，比如墙壁，柜子，桌子椅子等，保证几何平面多一点

3）保证XYZ三轴都有较大的旋转，特别是Z轴，我这里的实验平台是turtlebot2机器人小车，为了Z轴方向有较大的的旋转，用纸壳制作了一个拱桥、斜坡，通过爬斜坡来实现Z轴旋转。

首先roslaunch激光雷达与IMU的传感器驱动，然后录制传感器数据话题，为了防止数据遗漏，这里选择录制所有话题。

```
roslaunch velodyne_pointcloud velodyne_vlp16.launch
roslaunch xsens_mti_driver xsens_mti_node.launch
cd ~/rosbag
rosbag record -a -o velo_xsens.bag
```

录制完数据后，

先改写lidar_align.launch文件，将录制好的bag文件路径写进参数里面

执行如下命令开始标定：

```
rosrun lidar_align.launch
```

标定过程中，会显示如下数据，

迭代236轮之后，结果输出到工程目录下的result目录下，结果如下所示：

```
Active Transformation Vector (x,y,z,rx,ry,rz) from the Pose Sensor Frame to  the Lidar Fra
[-0.0505898, -0.493581, 0.107833, 0.00608699, -0.0307165, 0.747116]
Active Transformation Matrix from the Pose Sensor Frame to  the Lidar Frame:
   0.733224   -0.679499   -0.025763  -0.0505898
   0.679321    0.733656   -0.016485   -0.493581
  0.0301027 -0.00541422    0.999532    0.107833
          0           0           0           1
Active Translation Vector (x,y,z) from the Pose Sensor Frame to  the Lidar Frame:
[-0.0505898, -0.493581, 0.107833]
Active Hamiltonen Quaternion (w,x,y,z) the Pose Sensor Frame to  the Lidar Frame:
[0.930915, 0.00297308, -0.0150029, 0.364915]
Time offset that must be added to lidar timestamps in seconds:
-0.00638869
ROS Static TF Publisher: <node pkg="tf" type="static_transform_publisher" name="pose_lidar
```

3.IMU内参标定

IMU需要标定的参数主要是确定性误差和随机误差，确定性误差主要标定bias，scale和misalignment，随机误差主要标定noise（噪声）和random walk（随机游走误差），IMU随机误差的标定，通常采用Allan方差法，该方法是20世纪60年代由美国国家标准局的David Allan提出的,它是一种基于时域的分析方法。imu_utils是用于求取随机误差的开源工具。

安装编译imu_utils标定工具包，在保持传感器绝对静止的状态下，录制2小时imu数据，保存为.bag包，供标定算法使用。标定结束后，会输出imu内参结果如下所示：

流程：

imu标定流程在相机+imu标定中已经叙述。

```yaml
%YAML:1.0
---
type: IMU
name: xsens
Gyr:
   unit: " rad/s"
   avg-axis:
      gyr_n: 2.3704854385698929e-03
      gyr_w: 2.6685224155471368e-05
   x-axis:
      gyr_n: 2.3962741210552460e-03
      gyr_w: 2.7832590571207559e-05
   y-axis:
      gyr_n: 2.7594070648156717e-03
      gyr_w: 3.8714660427230775e-05
   z-axis:
      gyr_n: 1.9557751298387616e-03
      gyr_w: 1.3508421467975758e-05
Acc:
   unit: " m/s^2"
   avg-axis:
      acc_n: 1.9740799873834897e-02
      acc_w: 4.3015781912824339e-04
   x-axis:
      acc_n: 2.2776468152754685e-02
      acc_w: 6.4335594196542046e-04
   y-axis:
      acc_n: 1.2740355080011763e-02
      acc_w: 3.6570176663835329e-04
   z-axis:
      acc_n: 2.3705576388738239e-02
      acc_w: 2.8141574878095643e-04
```

标定完获得上述参数之后，只取如下数据，用于liosam的param.yaml中IMU内参配置

```yaml
rostopic: /mynteye/imu/data_raw
update_rate: 200.0 #Hz
accelerometer_noise_density: 1.9740799873834897e-02    #continous
accelerometer_random_walk: 4.3015781912824339e-04
gyroscope_noise_density: 2.3704854385698929e-03     #continous
gyroscope_random_walk: 2.6685224155471368e-05
```

3.运行LIOSAM算法测试

由于传感器之间的标定没有真值，因此采用运行liosam算法来验证标定结果有效性。

安装编译LIOSAM，这里就不赘述了。github链接： https://github.com/TixiaoShan/LIO-SAM

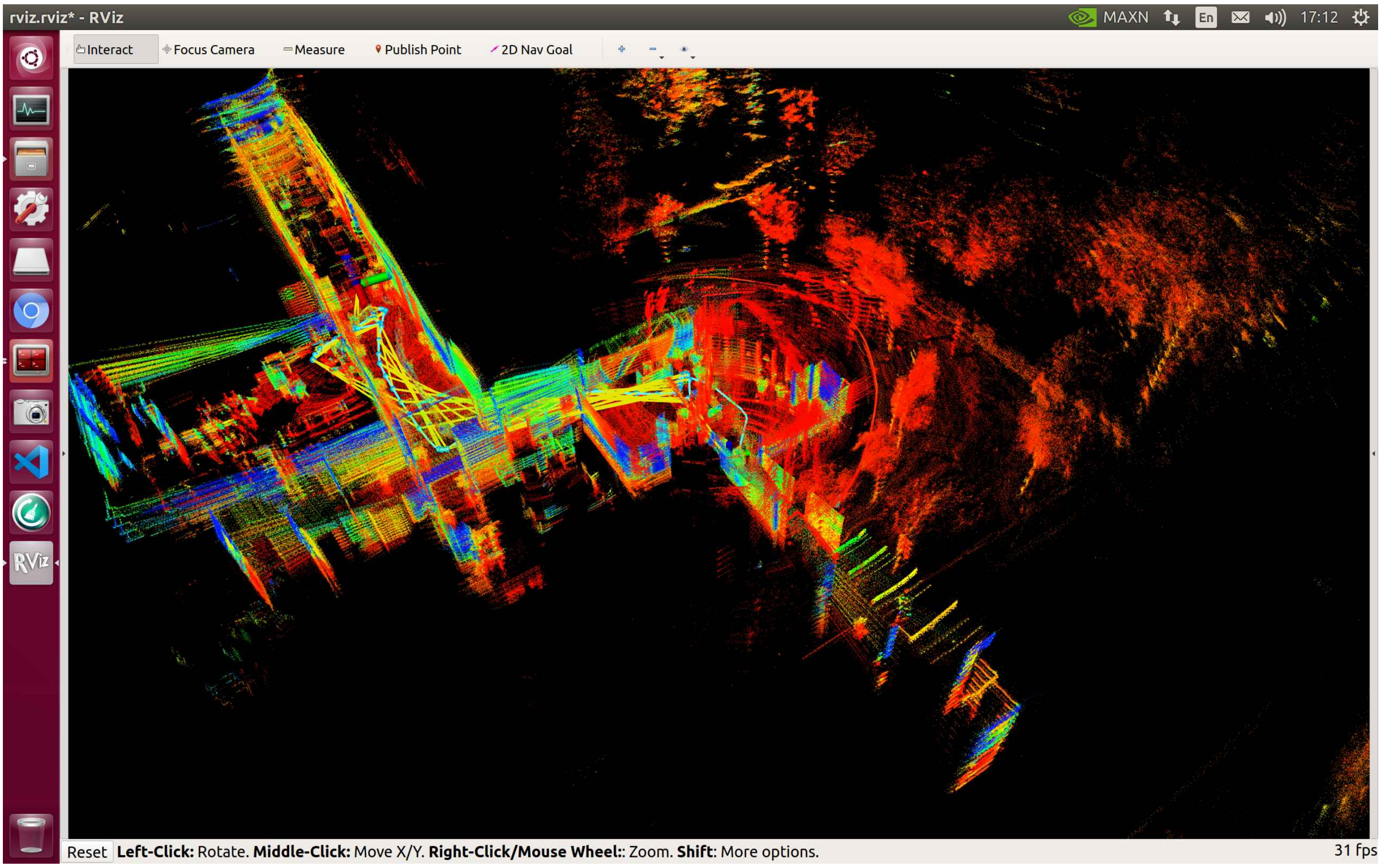
接上传感器实时运行liosam算法时，需要修改param.yaml文件中的相关参数，如激光雷达和IMU传感器的数据话题，IMU内参，lidar和imu的外参，根据我们标定的结果修改后的参数如下：

```yaml
lio_sam:
  # Topics
#  pointCloudTopic: "points_raw"                # Point cloud data
  pointCloudTopic: "velodyne_points"
  imuTopic: "imu/data"                          # IMU data
#  imuTopic: "imu_correct"
  odomTopic: "odometry/imu"                    # IMU pre-preintegration odometry, same fre
  gpsTopic: "odometry/gpsz"                    # GPS odometry topic from navsat, see modul
  # Frames
  lidarFrame: "base_link"
  baselinkFrame: "base_link"
  odometryFrame: "odom"
  mapFrame: "map"
  # GPS Settings
#  useImuHeadingInitialization: true            # if using GPS data, set to "true"
  useImuHeadingInitialization: false
  useGpsElevation: false                       # if GPS elevation is bad, set to "false"
  gpsCovThreshold: 2.0                         # m^2, threshold for using GPS data
  poseCovThreshold: 25.0                       # m^2, threshold for using GPS data
  # Export settings
#  savePCD: false                               # <https://github.com/TixiaoShan/LIO-SAM/i
  savePCD: true
  savePCDDirectory: "/dataset/LIO-SAM/xsens_velodyne"        # in your home folder, start
  # Sensor Settings
  sensor: velodyne                             # lidar sensor type, either 'velodyne' or '
  N_SCAN: 16                                   # number of lidar channel (i.e., 16, 32, 64
  Horizon_SCAN: 1800                           # lidar horizontal resolution (Velodyne:180
  downsampleRate: 1                            # default: 1. Downsample your data if too m
  lidarMinRange: 1.0                           # default: 1.0, minimum lidar range to be u
  lidarMaxRange: 1000.0                        # default: 1000.0, maximum lidar range to b
  #IMU Settings
  imuAccNoise: 1.9740799873834897e-02
  imuGyrNoise: 2.3704854385698929e-03
  imuAccBiasN: 4.3015781912824339e-04
  imuGyrBiasN: 2.6685224155471368e-05
  imuGravity: 9.80511
  imuRPYWeight: 0.01
#  imuAccNoise: 1.9238237446574064e-02
#  imuGyrNoise: 1.5385754496033436e-03
#  imuAccBiasN: 4.9615115224550062e-04
#  imuGyrBiasN: 5.0721205121154150e-06
#  imuGravity: 9.80511
#  imuRPYWeight: 0.01
  # Extrinsics (lidar -> IMU)
  extrinsicTrans: [-0.00201536, 0.00144471, -0.00145396]
#  extrinsicTrans: [0.0, 0.0, 0.0]
#  extrinsicRot: [-1, 0, 0,
#                  0, 1, 0,
#                  0, 0, -1]
#  extrinsicRPY: [0,  1, 0,
#                 -1, 0, 0,
```

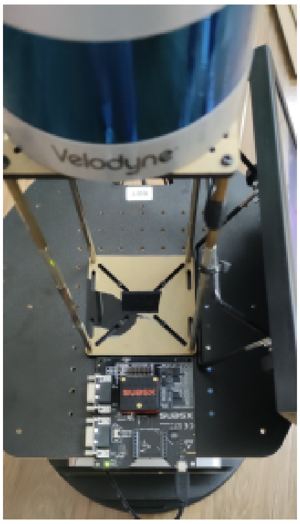
```
#                  0, 0, 1]
#  extrinsicRot: [1, 0, 0,
#                 0, 1, 0,
#                 0, 0, 1]
#  extrinsicRPY: [1, 0, 0,
#                 0, 1, 0,
#                 0, 0, 1]
  extrinsicRot:    [0.733224   -0.679499   -0.025763
   0.679321    0.733656   -0.016485
  0.0301027 -0.00541422    0.999532]
  extrinsicRPY:  [0.733224   -0.679499   -0.025763
   0.679321    0.733656   -0.016485
  0.0301027 -0.00541422    0.999532]
  # LOAM feature threshold
  edgeThreshold: 1.0
  surfThreshold: 0.1
  edgeFeatureMinValidNum: 10
  surfFeatureMinValidNum: 100
  # voxel filter paprams
  odometrySurfLeafSize: 0.4                     # default: 0.4 - outdoor, 0.2 - indoor
  mappingCornerLeafSize: 0.2                    # default: 0.2 - outdoor, 0.1 - indoor
  mappingSurfLeafSize: 0.4                      # default: 0.4 - outdoor, 0.2 - indoor
  # robot motion constraint (in case you are using a 2D robot)
  z_tollerance: 1000                            # meters
  rotation_tollerance: 1000                     # radians
  # CPU Params
  numberOfCores: 4                              # number of cores for mapping optimizatio
  mappingProcessInterval: 0.15                  # seconds, regulate mapping frequency
  # Surrounding map
  surroundingkeyframeAddingDistThreshold: 1.0   # meters, regulate keyframe adding thresh
  surroundingkeyframeAddingAngleThreshold: 0.2  # radians, regulate keyframe adding thres
  surroundingKeyframeDensity: 2.0               # meters, downsample surrounding keyframe
  surroundingKeyframeSearchRadius: 50.0         # meters, within n meters scan-to-map opt
  # Loop closure
  loopClosureEnableFlag: true
  loopClosureFrequency: 1.0                     # Hz, regulate loop closure constraint ad
  surroundingKeyframeSize: 50                   # submap size (when loop closure enabled)
  historyKeyframeSearchRadius: 15.0             # meters, key frame that is within n mete
  historyKeyframeSearchTimeDiff: 30.0           # seconds, key frame that is n seconds ol
  historyKeyframeSearchNum: 25                  # number of hostory key frames will be fu
  historyKeyframeFitnessScore: 0.3              # icp threshold, the smaller the better a
  # Visualization
  globalMapVisualizationSearchRadius: 1000.0    # meters, global map visualization radius
  globalMapVisualizationPoseDensity: 10.0       # meters, global map visualization keyfra
  globalMapVisualizationLeafSize: 1.0           # meters, global map visualization cloud
# Navsat (convert GPS coordinates to Cartesian)
navsat:
  frequency: 50
  wait_for_datum: false
  delay: 0.0
  magnetic_declination_radians: 0
  yaw_offset: 0
```

```
  zero_altitude: true
  broadcast_utm_transform: false
  broadcast_utm_transform_as_parent_frame: false
  publish_filtered_gps: false
# EKF for Navsat
ekf_gps:
  publish_tf: false
  map_frame: map
  odom_frame: odom
  base_link_frame: base_link
  world_frame: odom
#  frequency: 50
  frequency: 200
  two_d_mode: false
  sensor_timeout: 0.01
  # -------------------------------------
  # External IMU:
  # -------------------------------------
#  imu0: imu_correct
  imu0: imu/data
  # make sure the input is aligned with ROS REP105. "imu_correct" is manually transformed
  imu0_config: [false, false, false,
                true,  true,  true,
                false, false, false,
                false, false, true,
                true,  true,  true]
  imu0_differential: false
  imu0_queue_size: 50
  imu0_remove_gravitational_acceleration: true
source devel/setup.bash
roslaunch lio_sam run.launch
rosbag play your-bag
```

运行后的效果如下图所示：



传感器安装位置：



标定场地：