Guidelines to Implement OpenVINS

For capstone project - AAE14, 2023
(Only for academic use)
Created by SU Meiling - Final year student at Department of Aeronautical and Aviation Engineering of The Hong Kong Polytechnic University

You can download the project through this link:
https://drive.google.com/drive/folders/1XIMwdEy1i-Ckw92NI2cqpx8vd5vFhped?usp=share_link

1. Installation

1.1 Ubuntu

download ubuntu 18.04 amd64.iso on usb
install on computer
sudo apt update
sudo apt upgrade

1.2 Openvins

https://github.com/rpng/open_vins
get_started_installation guide:
https://docs.openvins.com/gs-installing.html

1.3 ROS

install ros for ubuntu 18.04 (follow the official website):
http://wiki.ros.org/melodic/Installation/Ubuntu
sudo apt-get install python-catkin-tools # ubuntu 16.04, 18.04
sudo apt-get install libeigen3-dev libboost-all-dev libceres-dev

echo "source /opt/ros/$ROS1_DISTRO/setup.bash" >> ~/.bashrc
source ~/.bashrc

1.4 clone openvins project

(follow instruction in the link above)
mkdir -p ~/workspace/catkin_ws_ov/src/
cd ~/workspace/catkin_ws_ov/src/
git clone https://github.com/rpng/open_vins/
cd ..
catkin build # ROS1

1.5 Opencv

check opencv installation (probably already there)
python
import cv2

1.6 explanation on OpenVins

ros1_serial_msckf.cpp is the file that shows how the data is processed
use serial.launch instead of subscribe.launch

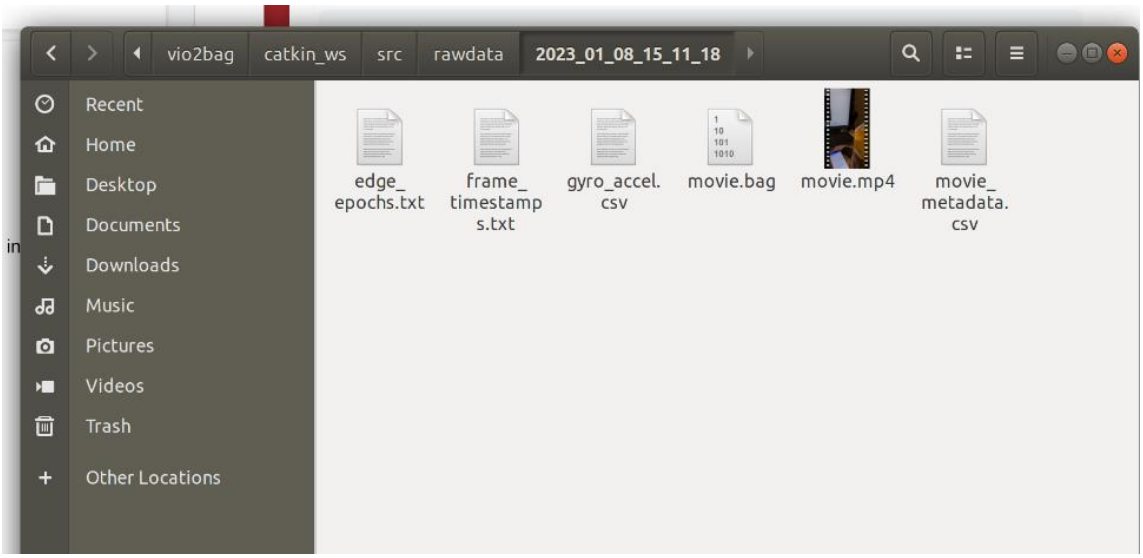
https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model
explains imu noise model clearly, can put in fyp:
A very interesting finding is that the result could vary a lot when the imu noise model is set differently, so what is the best way to solve this?

2. Data collection

2.1 create rosbag

2.1.1 collect data:
https://github.com/OSUPCVLab/mobile-ar-sensor-logger/releases/tag/v2.0-android
download marslogger app apk on phone (above link)
use marslogger to record
find the file in file manager, zip and upload to google drive
download the zip file on ubuntu system and extract to a folder

2.1.2 convert to rosbag
https://github.com/JzHuai0108/vio_common
build with ros
back to https://github.com/OSUPCVLab/mobile-ar-sensor-logger/wiki
follow “convert to rosbag” instruction and get rosbag (called movie.bag), which is stored in the same folder rawdata file
topics:
/cam0/image_raw 221 msgs : sensor_msgs/Image
/imu0 1621 msgs : sensor_msgs/Imu



Terminal commands:
cd ~/vio2bag/catkin_ws
source devel/setup.bash
cd ~/vio2bag/catkin_ws/src

BAG_PYTHON=~/vio2bag/catkin_ws/src/vio_common/python/kalibr_bagcreater.py
ANDROID_DATA_DIR=~/vio2bag/catkin_ws/src/rawdata/2023_01_08_15_11_18
sue@sue:~/vio2bag/catkin_ws/src$ python $BAG_PYTHON --video $ANDROID_DATA_DIR/movie.mp4 --imu $ANDROID_DATA_DIR/gyro_accel.csv --video_time_file $ANDROID_DATA_DIR/frame_timestamps.txt --output_bag $ANDROID_DATA_DIR/movie.bag

3. Calibration

3.1 camera-imu calibration

cd ~/kalibr_workspace/
source ~/kalibr_workspace/devel/setup.bash
cd /home/sue/kalibr_workspace/src/kalibr/aslam_offline_calibration/kalibr/python
provide the calibration rosbag, name it as dynamic.bag in the folder above
./kalibr_calibrate_cameras --target april_6x6_80x80cm.yaml --bag dynamic.bag --models pinhole-radtan --topics /cam0/image_raw #models choose pinhole-radtan instead of pinhole-equi
then get “dynamic-camchain.yaml” in the folder

3.2 imu calibration (IMU frequency is 221Hz)

more accurate:
follow this link: https://github.com/gaowenliang/imu_utils
(need to record 2 hours imu stationary data)

less accurate by manually modify:
imu_modified.yaml (modify in this file) parameter same as this link: https://github.com/ethz-asl/kalibr/issues/273

below is the command with manually modification (I used the modified one)
./kalibr_calibrate_imu_camera --target april_6x6_80x80cm.yaml --bag dynamic.bag --cam dynamic-camchain.yaml --imu imu_modified.yaml

finally get the file “dynamic-camchain-imucam.yaml”

4. estimator configuration

4.1 Config Yaml

then copy parameters in “dynamic-camchain-imucam.yaml” to “/home/sue/workspace/catkin_ws_ov/src/open_vins/config/euroc_mav/kalibr_imucam_chain.yaml”
then copy parameters in “dynamic-imu.yaml” to “/home/sue/workspace/catkin_ws_ov/src/open_vins/config/euroc_mav/kalibr_imu_chain.yaml”

init_window_time: 1.0,
init_imu_thresh: 0.2

4.2 IMU initialization:

#Need to try manually to set based on experiment conditions
init_window_time: 1.55 # how many seconds to collect initialization information (it determines

when the phone needs to start to move)
init_imu_thresh: 0.2 # threshold for variance of the accelerometer to detect a "jerk" in motion (can’t be too big or too small, need to try by myself)

4.3 Record to get the pose result
config file: record true

4.4 Use monocular instead of stereo:
inside the file of “open_vins/ov_msckf/launch/subscribe.launch”
search ‘stereo’
change
<arg name="max_cameras" default="2" /> to 1
<arg name="use_stereo" default="true" /> to false

inside file of “open_vins/ov_msckf/launch/subscribe.launch.py”
search ‘stereo’
change
name="use_stereo",
default_value="true" to false
name="max_cameras",
default_value="2", to 1

inside the file of
“/home/sue/workspace/catkin_ws_ov/src/open_vins/config/euroc_mav/estimator_config.yaml”
change
2 to 1, true to false

5. Implementation

cd ~/workspace/catkin_ws_ov/
source devel/setup.bash

then open 4 additional terminals (all need to source)

1. roscore
2. roslaunch ov_msckf subscribe.launch config:=euroc_mav
3. rviz -d src/open_vins/ov_msckf/launch/display.rviz
4. rosbag play /home/sue/vio2bag/catkin_ws/src/rawdata/2023_01_11_15_19_37/movie.bag
5. rqt_graph

6. Result

There are pose and covariance in the result txt
pose are the pose relative to the starting point (including tx ty tz, qx qy qz qw)

Individual result evaluation

convert to ENU coordinate system (Data_process file in github)

Record ground truth

How it is recorded:
using IMU + Lidar sensor

LC FGO

pose and covariance could be used as factors in FGO,
the covariance could represent how trustworthy is the sensor (such as which sensor should have more weighting)