gtsam_fusion

Estimates pose, velocity, and accelerometer / gyroscope biases by fusing GPS position and/or 6DOF pose with IMU data. The fusion is done using GTSAM's sparse nonlinear incremental optimization (ISAM2). The ROS (rospy) node is implemented using GTSAM's python3 inteface.

Contents

`gtsam_fusion_core.py`: Contains the core functionality related to the sensor fusion done using GTSAM ISAM2 (incremental smoothing and mapping using the bayes tree) without any dependency to ROS.

`gtsam_fusion_ros.py`: ROS node to run the GTSAM FUSION. The pose estimation is done in IMU frame and IMU messages are always required as one of the input. Additionally, either GPS (NavSatFix) or 6DOF pose (PoseWithCovarianceStamped) messages are required to constrain the drift in the IMU preintegration. The output is published as PoseWithCovarianceStamped messages with the frame rate defined by the IMU.

`plots.py`: Contains code to plot input and output of the GTSAM FUSION using matplotlib.

Installation (ROS Melodic)

As this module is implementd using python3 and ROS Melodic uses python2 by default, few steps need to be taken to be able to run python3 code. This installation assumes that ROS Melodic is already installed on a PC running Ubuntu 18.04LTS

Install the required dependencies

```
sudo apt-get install python3-pip python3-yaml python-catkin-tools python3-dev python3-numpy
```

Install the required python modules

```
pip3 install rospkg catkin_pkg empy pycryptodomex gnupg
```

Install already the modules required by gtsam slam:

```
pip3 install gtsam utm matplotlib
```

Create a new (separate) workspace for ROS packages using Python3

```
mkdir ~/python3_ws && cd ~/python3_ws
catkin config -DPYTHON_EXECUTABLE=/usr/bin/python3 -
DPYTHON_INCLUDE_DIR=/usr/include/python3.6m -
DPYTHON_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython3.6m.so
catkin config --install
```

Create a folder for the source and clone the two problematic ROS modules (tf2, vision_opencv, and ros_comm due to rosbag API)

```
mkdir ~/python3_ws/src && cd ~/python3_ws/src
git clone -b melodic https://github.com/ros-perception/vision_opencv.git
git clone -b melodic-devel https://github.com/ros/geometry2.git
git clone -b melodic-devel https://github.com/ros/ros_comm.git
```

Compile and source the workspace

```
cd ~/python3_ws
catkin build
source install/setup.bash --extend
```

In order to test that everything works, open the python3 intepreter

```
python3
```

Type the following import commands. If no errors show up, everything is setup correctly.

```
import rospy
import tf2_ros
import cv_bridge
import rosbag
import gtsam
```

Remember to source the python3_ws before using nodes from it:

```
source ~/python3_ws/install/setup.zsh --extend
```

Running

GPS & IMU FUSION offline via rosbag API

Download the sample bag file from: gtsam_fusion.bag

Launch the node giving the bag file path as input argument:

```
roslaunch gtsam_fusion rosbag_sample_gps_imu.launch
```

The bag file will be processed via rosbag API and the fusion results will be plotted at the end.

6DOF POSE & IMU FUSION offline via rosbag API

Download the sample bag file from: gtsam_fusion.bag

Launch the node giving the bag file path as input argument:

```
roslaunch gtsam_fusion rosbag_sample_gps_pose.launch
```

The bag file will be processed via rosbag API and the fusion results will be plotted at the end.

Online usage (GPS + IMU):

To run GTSAM_FUSION offline or online by subscribing to input topics:

```
rosrun gtsam_fusion gtsam_fusion_ros.py _imu_topic:=<imu_topic>
_gps_topic:=<gps_topic> _use_pose:=false _use_gps:=true
```

Start nodes that publish <gps_topic> and <imu_topic> and additionally adjust the other parameters defined in `gtsam_fusion_ros.py`.

The fused pose will be published in topic `/base_pose_in_map_100hz` (remap the topic name if needed).