SlamPark Documentation

1.0.0

GaoHongchen

2020 09 20

SLAM Basis

- Simultaneous localization and mapping (wikipedia)]

SLAM Study Notes:

- SLAM Blog (CSDN)
- SLAM Notes (github.io)
- SLAM Notes (LaTeX on Overleaf)

CHAPTER 1

SLAM Theoretical Basis

- Mathematics
  - https://github.com/cggos/maths_cg
- Computer Vision
  - https://github.com/cggos/cgocv
  - https://cggos.github.io/categories.html#ComputerVision
- Kinematics and Dynamics
- State Estimation
  - https://github.com/cggos/state_estimation_cg
- Sensors
  - Sensor Types
  - Sensor Calibration
    - https://cggos.github.io/robotics/robot-calibration.html

CHAPTER 2

Localization & Mapping

- SLAM Frameworks
- SLAM Odometry
  - Visual Odometry/Visual Inertial Odometry
  - Laser Odometry
  - Wheel Odometry
- SLAM Loop Closure
  - Visual Vocabulary
- SLAM Mapping

CHAPTER 3

GNSS & INS

- INS
- GNSS
- GIS
- Geomagnetics

CHAPTER 4

Multiple Sensor Fusion

- https://github.com/cggos/sensor_fusion_cg

CHAPTER 5

SLAM Benchmark

- SLAM Benchmark
- SLAM Dataset
- SLAM Simulation

CHAPTER 6

SLAM Applications

- SLAM Modules
- AR (6 DoF)
- Drone ($\approx$ 4 DoF)
- Ground Mobile Robot (3 DoF)

CHAPTER 7

SLAM QA

- Experience of Running SLAM
- Engineering Tricks
- Challenge

7.1 Kinematics and Dynamics

Kinematics and Dynamics for Robotics and Computer Vision, etc.

- EuclideanSpace - Mathematics and Computing

7.1.1 Online Tools

- Quaternions Visualization
- 3D Rotation Converter

7.1.2 Libraries

- Eigen Space transformations
- tf2 (ROS) is the second generation of the transform library, which lets the user keep track of multiple coordinate frames over time
- Kindr - Kinematics and Dynamics for Robotics
- ethz-asl/minkindr: A minimal library for transformations, following the kindr interface. Uses active quaternions of rotation in **Hamilton notation**.
- RBDL: Rigid Body Dynamics Library

- orocos KDL: Kinematics and Dynamics Library
- Spatial Vectors and Rigid-Body Dynamics: Spatial vectors are 6D vectors that simplify the task of describing, analysing, and calculating rigid-body dynamics
- ethz-asl/geodetic_utils: Simple library for converting coordinates to/from several geodetic frames (lat/lon, ECEF, ENU, NED, etc.)
- posest: A C/C++ Library for Robust 6DoF Pose Estimation from 3D-2D Correspondences

7.1.3 Books

- ***Visualizing Quaternions***
- ***Rigid Body Dynamics Algorithms***

7.2 Sensors

7.2.1 Sensor Types

Camera

IMU

- IMU

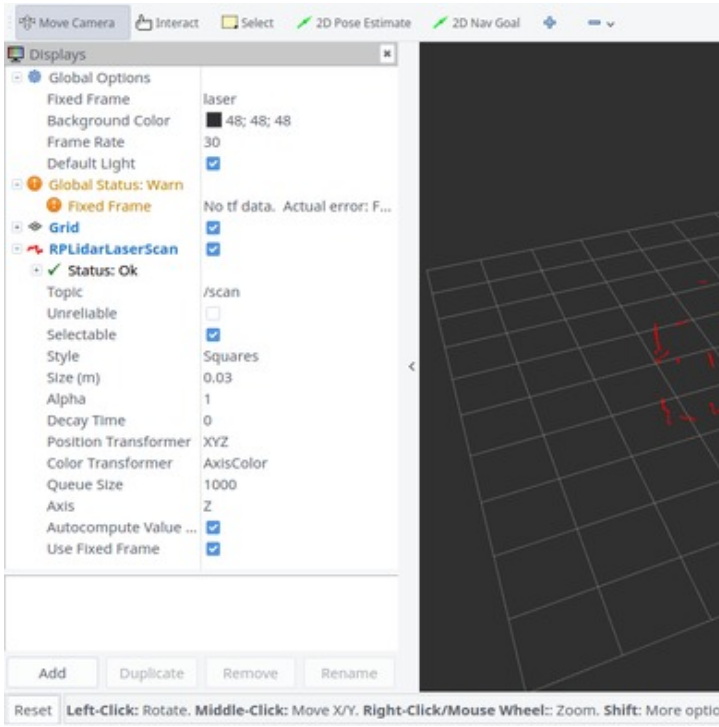
radar

- LiDARNews
- ?

rplidar

- rplidar (ROS Wiki)

- RPLIDAR A1 (slamtec)



- `roslaunch rplidar_ros view_rplidar.launch`

GPS

Sonar

7.2.2 Sensor Calibration

- Spatial Calibration
- Temporal Calibration

7.3 SLAM Frameworks

7.3.1 SLAM Frameworks Overview

[TOC]

Filter-SLAM

EKF-SLAM

- The Extended Kalman Filter - SLAM

- EKF Slam Example
- EKF-SLAM-on-Manifold
- damarquezg/SLAMTB: EKF-SLAM TOOLBOX FOR MATLAB

MonoSLAM

- cggos/SceneLib2
  - MonoSLAM open-source library
  - an open-source C++ library for SLAM originally designed and implemented by **Andrew Davison** and colleagues at the University of Oxford
- MATLAB Implementation of MonoSLAM
- EKF based Monocular SLAM
- Mono-Slam Implementation in ROS

EKFmonocularSLAM

- EKFmonocularSLAM
- 1-Point RANSAC Inverse Depth EKF Monocular SLAM

vSLAM

- Monocular SLAM
- mono_vo
- FOVIS: a visual odometry library that estimates the 3D motion of a camera using a source of depth information for each pixel.
  - Paper: *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*
  - libfovis
  - fovis_ros: **mono_depth_odometer** and **stereo_odometer**
- SOFT

Feature-based Method

- cggos/ptam_cg
- cggos/orbslam2_cg
- cggos/viso2_cg
- cggos/rgbdslam2_cg
- rubengooj/pl-slam: an algorithm to compute stereo visual SLAM by using both point and line segment features
- UcoSLAM is a library for SLAM using keypoints that able to operate with **monocular cameras, stereo cameras, rgbd cameras**

Semi-Direct Method

- cggos/svo_cg: a **Semi-direct Monocular Visual Odometry** pipeline
- HeYijia/svo_edgelet: A more robust SVO with edgelet feature

Direct Method

- DSO
- cggos/lsd_slam_cg

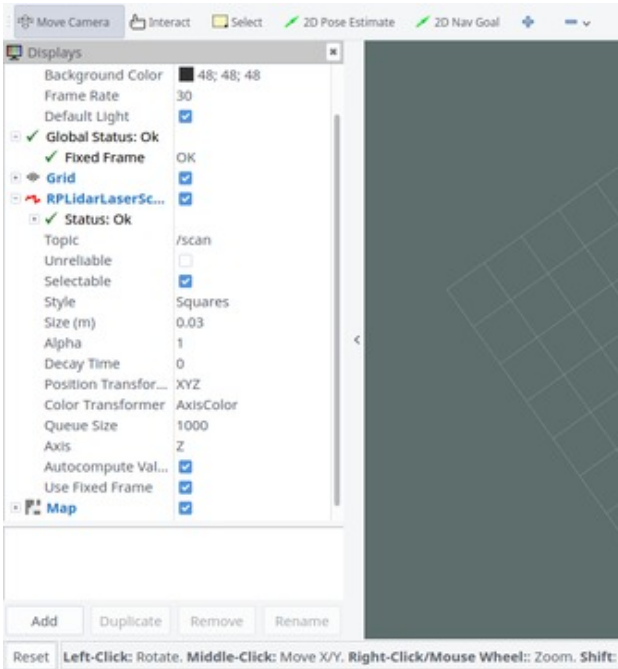
VI-SLAM/VIO

- cggos/okvis_cg
- cggos/vins_mono_cg
- cggos/vins_fusion_cg
- cggos/msckf_cg
- cggos/rovio_cg: Robust Visual Inertial Odometry

Laser SLAM

- rplidar_ros (branch `slam`)
- hector_slam
- openslam_gmapping
- slam_gmapping

Example Run:



- **hector_slam** `roslaunch rplidar_ros view_slam.launch`

7.4 SLAM Odometry

7.4.1 Visual Odometry

[TOC]

Visual Tracking

Optical Flow

- LK

Pose Estimation

2D-2D: Epipolar Geometry

- Fundamental Matrix
- Essential Matrix
- Homography Matrix

3D-3D: ICP

- SVD
-

3D-2D: PnP

The **Perspective-n-Point (PnP)** pose problem is the problem of estimating the relative pose – 3D position and orientation – between a calibrated perspective camera and a 3D object (or between the camera and the entire 3D scene) from a set of n visible 3D points with known (X,Y,Z) object (or scene) coordinates and their 2D projections with known (u,v) pixel coordinates.

The PnP problem is relevant in 3D object tracking and camera localization / tracking and is often used for example in Structure-from-Motion (SfM), Visual Odometry (VO), Simultaneous Localization and Mapping (SLAM) and image-based localization pipelines.

- Perspective-n-Point (Wikipedia)
- Code
  - midjji/pnp: A RANSAC and BA based pnp wrapper for the Lambdatwist p3p solver
  - ydsf16/PnP_Solver: Personal implementations of solvers for PnP problem, including DLT and EPnP

Solver

- DLT
- P3P
- EPnP
- UPnP
- Motion Only BA
- RANSAC

Direct Method VO

7.5 SLAM Loop Closure

7.5.1 Visual Vocabulary

- Simple bag-of-words loop closure for visual SLAM

[TOC]

DBoW

- dorian3d/DBoW: an open source C++ library for indexing and converting images into a bag-of-word representation (Note: out of date)
- dorian3d/DBoW2: an improved version of the DBow library
- rmsalinas/DBow3: an improved version of the DBow2 library

```
git clone https://github.com/rmsalinas/DBow3.git
cd DBow3
mkdir build & cd build
cmake .. & make
```

DLoopDetector

- dorian3d/DLoopDetector is an open source C++ library to detect loops in a sequence of images collected by a mobile robot

FBOW

- rmsalinas/fbow: FBOW (Fast Bag of Words) is an extremmely optimized version of the DBow2/DBow3 libraries.

FAB-MAP

- FAB-MAP is a system for **appearance-based** navigation or place recognition
- arrenglover/openfabmap: Open-source C++ code for the FAB-MAP visual place recognition algorithm
- OpenFABMAP (OpenCV) is an open and modifiable code-source which implements the **Fast Appearance-based Mapping algorithm (FAB-MAP)** developed by Mark Cummins and Paul Newman

Visual Vocabulary Example

- ORBvoc.txt.tar.gz

```
# location: https://github.com/raulmur/ORB_SLAM2/blob/master/Vocabulary/
wget https://raw.githubusercontent.com/raulmur/ORB_SLAM2/master/Vocabulary/ORBvoc.
↪txt.tar.gz
```

7.6 SLAM Mapping

- Axis Maps
- Map representations

[TOC]

7.6.1 Metric Map

Grid Map

- ANYbotics/grid_map: Universal grid map library for mobile robotic mapping
- Occupancy Grid Map

2D Occupancy Grid Map

- costmap_2d (ROS Wiki): provides an implementation of a 2D costmap that takes in sensor data from the world, builds a 2D or 3D occupancy grid of the data (depending on whether a voxel based implementation is used), and inflates costs in a 2D costmap based on the occupancy grid and a user specified inflation radius

3D Occupancy Grid Map

- OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, implements a 3D occupancy grid mapping approach, providing data structures and mapping algorithms in C++ particularly suited for robotics
- ANYbotics/elevation_mapping: Robot-centric elevation mapping for rough terrain navigation
- ethz-asl/volumetric_mapping: 3D volumetric (occupancy) maps, providing a generic interface for disparity map and pointcloud insertion, and support for custom sensor error models

7.6.2 Topological Map

7.6.3 Semantic Map

7.6.4 Map Viewers

- Octomap: `octovis octomap.bt`
- PCD: `pcl_viewer map.pcd`

7.7 INS

An **Inertial Navigation System (INS)** is a navigation device that uses a computer, motion sensors (**accelerometers**) and rotation sensors (**gyroscopes**) to continuously calculate by dead reckoning the position, the orientation, and the velocity (direction and speed of movement) of a moving object without the need for external references. Often the inertial sensors are supplemented by a **barometric altimeter** and occasionally by **magnetic sensors (magnetometers)** and/or speed measuring devices.

- Open Source Inertial Navigation Toolkit
- Inertial Navigation System (INS) Toolbox (matlab)
- Aided Inertial Navigation System (AINS) Toolbox for MatLab Software
- NaveGo: an open-source MATLAB/GNU Octave toolbox for processing integrated navigation systems and performing inertial sensors analysis
- Aceinna Navigation Studio: Simulate, Deploy, and Analyze Navigation Systems
- priseborough/InertialNav: Inertial Navigation Estimation Library
- RoNIN: Robust Neural Inertial Navigation

7.7.1 SINS

Strapdown Inertial Navigation System

Some basic processes for SINS based on **MARG (Magnetic, Angular Rate, and Gravity)** are implemented, including electromagnetic compass calibration, an **AHRS (attitude and heading reference system)** that use an extended Kalman filter (EKF), and trace tracking. Loosely coupling sensor fusion by using the minimum squared error (MSE) method is also implemented.

- Strapdown inertial navigation
- Strapdown Inertial Navigation Technology, 2nd Edition
-
- IMU
- IMU

7.7.2 AHRS

Attitude and Heading Reference System

An AHRS consists of sensors on three axes that provide attitude information for aircraft, including roll, pitch and yaw. These are sometimes referred to as **MARG (Magnetic, Angular Rate, and Gravity)** sensors and consist of either solid-state or microelectromechanical systems (MEMS) **gyroscopes, accelerometers and magnetometers**.

The key difference between an IMU and an AHRS is the addition of an on-board processing system in an AHRS, which provides attitude and heading information.

- Sensor Fusion Algorithms: There are a variety of sensor fusion algorithms out there, but the two most common in small embedded systems are the **Mahony and Madgwick filters**.
- Open source IMU and AHRS algorithms
- psiphi75/ahrs: AHRS (Attitude Heading Reference Systems) calculation for JavaScript

7.7.3 IMU Attitude Calculation

- IMU Data Fusing: Complementary, Kalman, and Mahony Filter
- cggos/imu_tools: ROS tools for IMU devices
- IMU-sensor-fusion-with-linear-Kalman-filter (mathworks): Reads IMU sensor (acceleration and velocity) wirelessly from the IOS app 'Sensor Stream' to a Simulink model and filters an orientation angle in degrees using a linear Kalman filter.

7.7.4 Dead Reckoning

In navigation, dead reckoning is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course.

7.8 GNSS

- An Introduction to GNSS
- Navipedia

GNSS Systems:

- GPS (United States)
- GLONASS (Russia)
- BeiDou (China)
- Galileo GNSS system (European Union)
- IRNSS regional navigation satellite system (India)
- QZSS regional navigation satellite system (Japan)

7.8.1 GNSS Coordinates

- ECEF
- ECI
- LLA LongitudeLatitude Altitude
- ENUNED

7.8.2 RTK

DGPS stands for Differential GPS. It is the GPS signal corrected by ground reference stations (which can estimate the ionosphere error on the GPS signal). Traditionally, DGPS accuracy can go down to sub-meter level. RTK technique allows DGPS accuracy to be at centimeter-level.

RTK stands for Real Time Kinematic. RTK algorithms make use of the carrier phase measurement to estimate accurately the distance from the receiver to the satellites. It allows an accuracy of 1-2 cm on position.

7.8.3 GNSS in inertial navigation

Inertial navigation systems can take advantage of additional constellation to use more satellites and further improve signal robustness in harsh environments such as urban canyons, forests, mountains.

- Aceinna/gnss-ins-sim: GNSS-INS-SIM is an GNSS/INS simulation project, which generates reference trajectories, IMU sensor output, GPS output, odometer output and magnetometer output.
- aaronboda24/Loose-GNSS-IMU
- karanchawla/GPS_IMU_Kalman_Filter

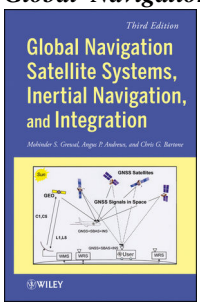
7.8.4 Software

- GeographicLib is a small set of C++ classes for performing conversions between geographic, UTM, UPS, MGRS, geocentric, and local cartesian coordinates, for gravity (e.g., EGM2008), geoid height, and geomagnetic field (e.g., WMM2010) calculations, and for solving geodesic problems.
- RTKLIB: An Open Source Program Package for GNSS Positioning
- GPSoftNav
- gpsd is a service daemon that monitors one or more GPSes or AIS receivers attached to a host computer through serial or USB ports, making all data on the location/course/velocity of the sensors available to be queried on TCP port 2947 of the host computer.

7.8.5 Books

- –

- ***Global Navigation Satellite Systems, Inertial Navigation, and Integration, 3rd Edition*** (link for code)



7.9 GIS

Geographic Information System

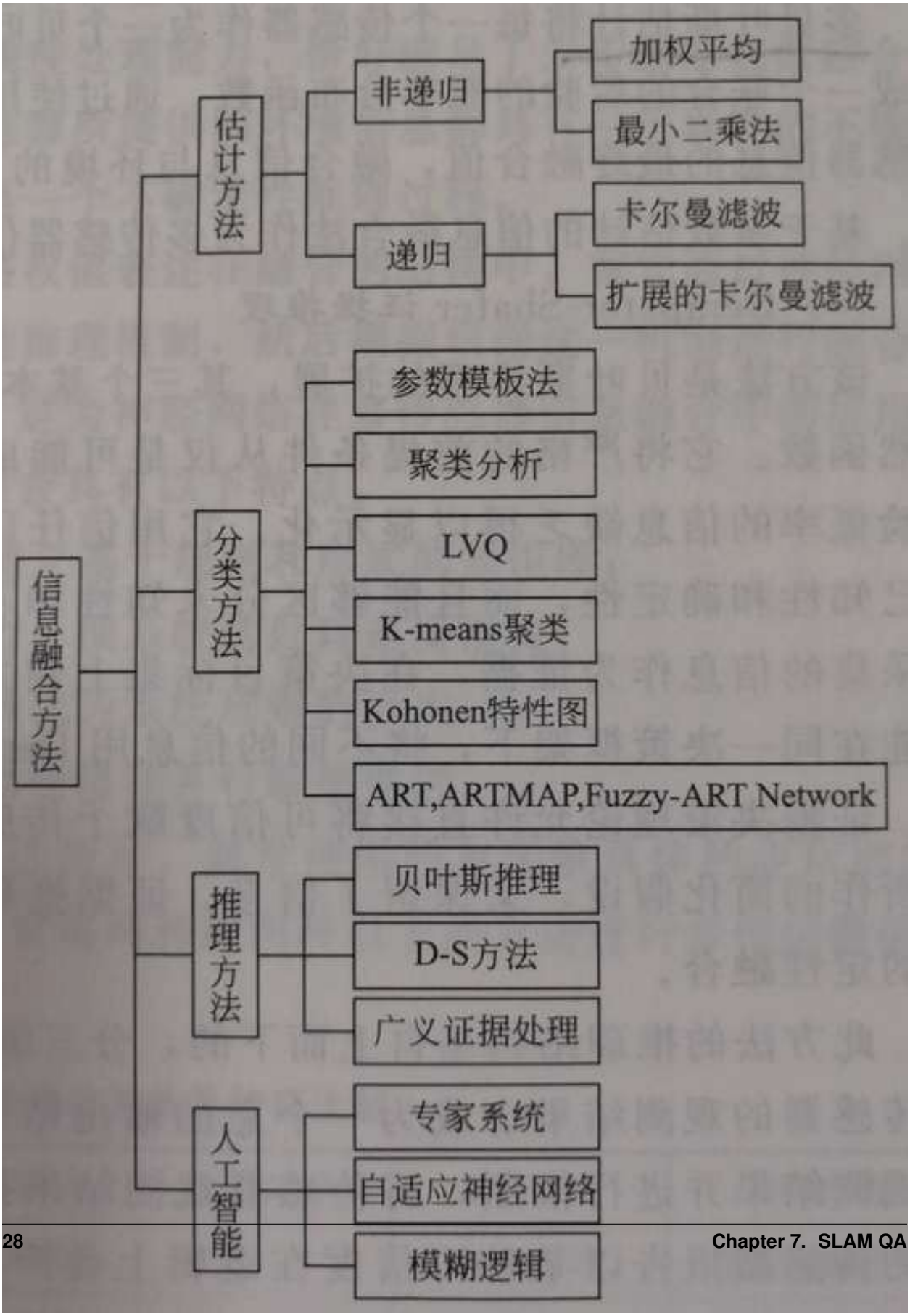
- GIS Geography
- Esri: GIS Mapping Software, Spatial Data Analytics, Location Intelligence
- AnalyticalGraphicsInc/3d-tiles: Specification for streaming massive heterogeneous 3D geospatial datasets
- OpenStreetMap Wiki

7.10 Geomagnetics

- 0.4-0.6 Gauss2426
- magnetic-declination: find the magnetic declination at your location

---

7.11 Multi Sensor Fusion



7.11.1 MSF Overview

- Sensor Fusion Tutorial
- WTF is Sensor Fusion? The good old Kalman filter
-

7.11.2 MSF Examples

- cggos/lidar_radar_ekf_fusion
- cggos/ethzasl_msf
  - ethzasl_sensor_fusion: Time delay compensated single and multi sensor fusion framework based on an EKF
- cggos/robot_pose_ekf
  - robot_pose_ekf (ROS Wiki)
  - A Beginner's Guide to the ROS robot_pose_ekf Package
  - ROS Kalman Filter for Sensor Fusion
  - RoboND-EKFLab
  - EXTENDED KALMAN FILTER: INCORPORATING GPS USING ROBOT_POSE_EKF
- robot_localization wiki
  - cra-ros-pkg/robot_localization
  - methylDragon/ros-sensor-fusion-tutorial: An in-depth step-by-step tutorial for implementing sensor fusion with robot_localization!
  - The Ros Robot_localization package
- locusrobotics/fuse: The fuse stack provides a general architecture for performing sensor fusion live on a robot. Some possible applications include state estimation, localization, mapping, and calibration.

7.11.3 MSF Toolbox

- Sensor Fusion and Tracking Toolbox (mathworks)

7.12 SLAM Benchmark

7.12.1 EVO

evo: Python package for the evaluation of odometry and SLAM

1. get the ground truth file, e.g. EuRoC **data.csv**

2. convert it to TUM format **data.tum**: `evo_traj euroc data.csv --save_as_tum`
3. evaluate `evo_ape tum data.tum pose_out.tum --align --plot`
4. or directly `evo_ape euroc data.csv pose_out.tum --align --plot`

7.12.2 rgbd-dataset tools

Useful tools for the RGB-D benchmark (TUM)

1. get the ground truth file, e.g. EuRoC **data.csv**
2. convert it to TUM format **data.tum**, we can use `evo`
3. evaluate: `./evaluate_ate.py data.tum pose_out.tum --plot result --verbose`

7.13 SLAM Dataset

dataset collections for cv, slam and robotics, also include some tools and code, etc.

[TOC]

7.13.1 TUM

- TUM Data

7.13.2 KITTI

- KITTI Vision Benchmark Suite

7.13.3 ETHZ ASL

- ETHZ ASL Datasets
- The EuRoC MAV Visual Inertial Datasets
  - Machine Hall 01
  - Machine Hall 03

7.13.4 UZH RPG

- UZH RPG Datasets
- The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM

7.13.5 PennCOSYVIO

- PennCOSYVIO Dataset

7.13.6 Others

Stereo Dataset

- Karlsruhe Dataset: Stereo Video Sequences + rough GPS Poses
- Middlebury Stereo Data

Laser Dataset

- LaFiDa - A Laserscanner Multi-Fisheye Camera Dataset
- Oxford New College Vision and Laser Data Set

Event Camera Dataset

- Multi Vehicle Stereo Event Camera Dataset

7.14 SLAM Simulation

7.14.1 Simulation Data

- HeYijia/vio_data_simulation: Generate imu data and feature in camera frame

7.15 SLAM Modules

7.15.1 DIY

- ROS CAMERA AND IMU SYNCHRONIZATION
- Visual inertial odometry on a budget!!

7.15.2 VI-Modules

- RealsenseT265
- PerceptInTrifo-VIO
- A-StarASLAM
- IndemindVi-SLAM

7.16 Experience of Running SLAM

7.16.1 HKUST VINS

- –
  - camimu
- VINS-Mono VINS-Fusion camimu **from camera frame to imu frame**

VINS-Mono

- **8** Suunto

VINS-Fusion

-
- VINS-Fusion `estimate_extrinsic: 1`camimu `body_T_cam0 body_T_cam1`