
SLAM Kit

发行版本 *1.0.1*

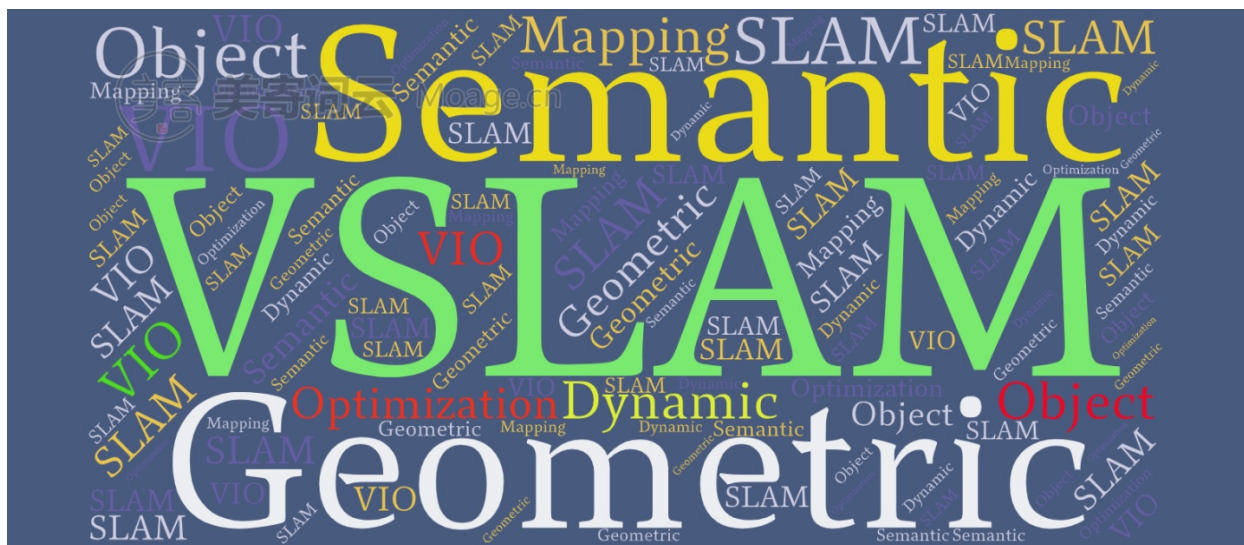
Gavin Gao

2023 年 01 月 09 日

Localization Mapping

1	Study Notes	3
2	SLAM Basis	5
3	Localization & Mapping	7
4	SLAM Benchmark	9
5	SLAM Applications	11
6	SLAM QA	13

- [Source Code \(GitHub\)](#)



CHAPTER 1

Study Notes

- [Simultaneous localization and mapping \(Wikipedia\)](#)
- [SLAM Blog \(CSDN\)](#)
- [SLAM Blog \(CGABC\)](#)
- [SLAM Notes \(LaTeX on Overleaf\)](#)

CHAPTER 2

SLAM Basis

- Mathematics & Scientific Computing
 - <https://sci.cgabc.xyz/>
- Kinematics and Dynamics
- Computer Vision
 - <https://cv.cgabc.xyz/>
- State Estimation
 - <https://est.cgabc.xyz/>
- Multi-Sensor Fusion
 - <https://msf.cgabc.xyz/>
- Robotics
 - <https://robot.cgabc.xyz/>

CHAPTER 3

Localization & Mapping

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

CHAPTER 4

SLAM Benchmark

- SLAM Benchmark
- SLAM Dataset
- SLAM Simulation

CHAPTER 5

SLAM Applications

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

- Experience of Running SLAM
- Engineering Tricks
- Challenge

6.1 SLAM Frameworks

6.1.1 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*
 - [libfovvis](#)
 - [fovvis_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

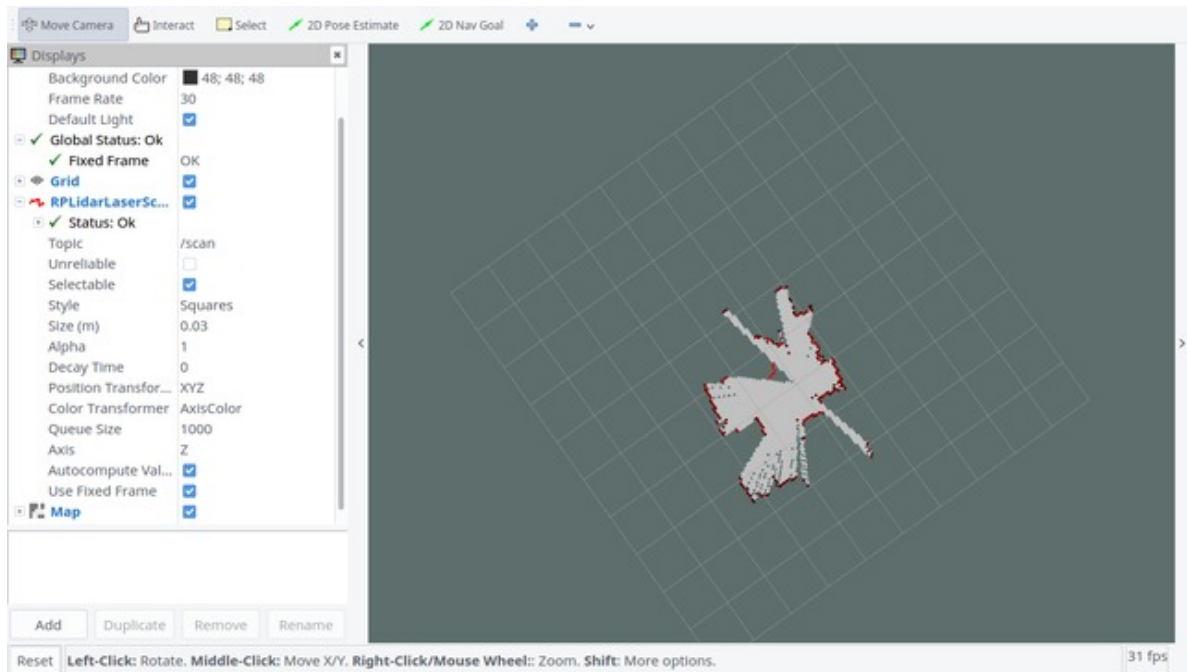
- `cggos/okvis_cg`
- `cggos/vins_mono_cg`
- `cggos/vins_fusion_cg`
- `cggos/msckf_cg`
- `cggos/rovio_cg`: Robust Visual Inertial Odometry

Laser SLAM

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

Example Run:

- 通过 **hector_slam** 建图，运行 `roslaunch rplidar_ros view_slam.launch`，效果如下



6.2 SLAM Odometry

6.2.1 Visual Odometry

[TOC]

Visual Tracking

Optical Flow

- LK

Pose Estimation

2D-2D: Epipolar Geometry

- Fundamental Matrix
- Essential Matrix
- Homography Matrix

3D-3D: ICP

- SVD
- 非线性优化

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（非线性优化）
- RANSAC

Direct Method VO

6.3 SLAM Loop Closure

6.3.1 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 extremely optimized version of the DBow2/DBow3 libraries.

FAB-MAP

- [FAB-MAP](#) is a system for **appearance-based** navigation or place recognition
- [arenglover/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
```

6.4 SLAM Mapping

- [Axis Maps](#)
 - [Map representations](#)
-

[TOC]

6.4.1 Metric Map

Grid Map

- [ANYbotics/grid_map](#): Universal grid map library for mobile robotic mapping
- [占据栅格地图 \(Occupancy Grid Map\)](#)

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

6.4.2 Topological Map

6.4.3 Semantic Map

6.4.4 Map Viewers

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

6.5 SLAM Benchmark

6.5.1 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`

6.5.2 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`

6.5.3 rpg_trajectory_evaluation

- https://github.com/uzh-rpg/rpg_trajectory_evaluation
- https://github.com/rpng/open_vins/tree/master/ov_eval

6.6 SLAM Dataset

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

[TOC]

6.6.1 TUM

- TUM Data

6.6.2 KITTI

- KITTI Vision Benchmark Suite

6.6.3 ETHZ ASL

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

6.6.4 UZH RPG

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

6.6.5 PennCOSYVIO

- PennCOSYVIO Dataset

6.6.6 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](#)

6.7 SLAM Simulation

6.7.1 VIO Simulator

- 思路 1: 指定轨迹方程, 求一阶导得到速度和角速度, 求二阶导得到加速度
 - [HeYijia/vio_data_simulation](#): Generate imu data and feature in camera frame
- 思路 2: 已有 pose 轨迹, 不知道方程, 利用 B-Spline 产生 IMU 数据
 - from OpenVINS (On manifold SE3 B-Spline)
 - * https://github.com/rpng/open_vins/tree/master/ov_msckf/src/sim
 - * <https://github.com/Edwinem/VI-Simulator>

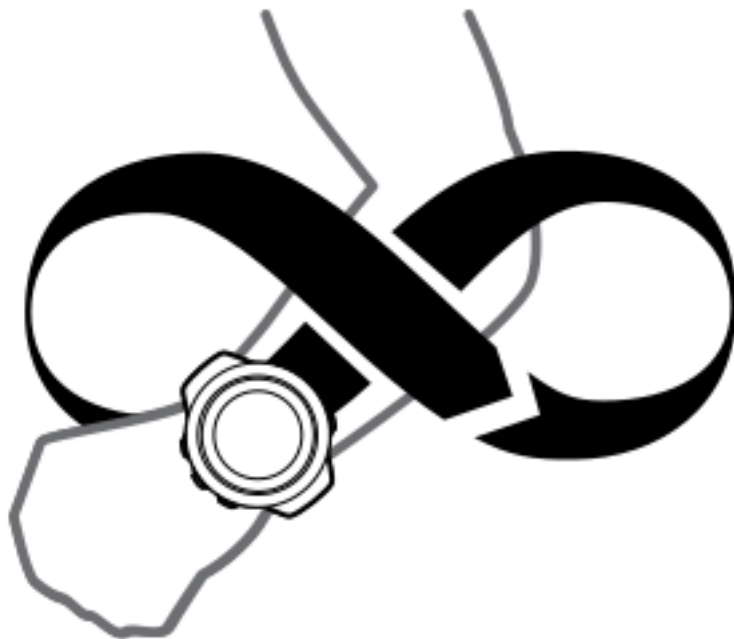
6.8 Experience of Running SLAM

6.8.1 HKUST VINS

- 启动运行即飞, 原因主要有两个:
 - 初始化操作有误
 - cam 和 imu 的外参不对
- VINS-Mono 和 VINS-Fusion 中 cam 和 imu 的外参为 **from camera frame to imu frame**, 不要写反

VINS-Mono

- 动态初始化，操作手法可以参考如下 **8 字法**（下图为 Suunto 腕表校准指南针的手法）



VINS-Fusion

- 静态初始化
- VINS-Fusion 即使开启在线外参估计选项 `estimate_extrinsic: 1`，对应的 `cam` 和 `imu` 的外参 `body_T_cam0` 和 `body_T_cam1` 最好也要给一个比较好的初始参数，否则（比如单位阵）容易飞