Note: SLAM

Yanqing Wu

Update: November 3, 2022

Contents

1 Chap 1: Introduction

1.1 V-SLAM

SLAM = Simultaneous Localization and Mapping

- Localization = where am I?
- Map building = what is the surrounding environment like?
- e.g.: a moving rigid body, equipped with a specific sensor, estimates its motion and builds a model (certain kinds of description) of the surrounding environment, without a priori information. (In V-SLAM: sensor == camera)

Non-intrusive Sensors

- self-contained
- hard, but can adapt to different environments
- e.g. cameras, laser scanners, IMU

Intrusive sensors

- Depends on a prepared environments
- easy, but not transferable
- e.g. QR code, gliding rails

Cameras: monocular camera, stereo camera, RGB-D camera

Monocular camera

- + Captures photo (projection of 3D world to 2D plane)
- + Simple to use
- - no depth info
  - Soln: move the camera and estimate its motion, and distances + sizes of objects
  - Pixel disparity: the movement of objects on the photo as the camera moves
- - scale ambiguity

    - No clear solution
- Human beings can use common-sense to determine depth from photo

Stereo camera = two monocular camera

- + Has depth info
- + Applies to indoor & outdoor
- - Computational heavy
- - Measuring distance is limited by baseline (the distance between two cameras) & camera resolution
- - Needs special calibration and configuration

RGB-D camera (aka depth camera)

- + Has depth info (measure distance by light travel time)
- - narrow measurement range
- - noisy data
- - small field of view
- - susceptibility to sunlight interference
- - unable to measure transparent material

1.2 Classical VSLAM Framework

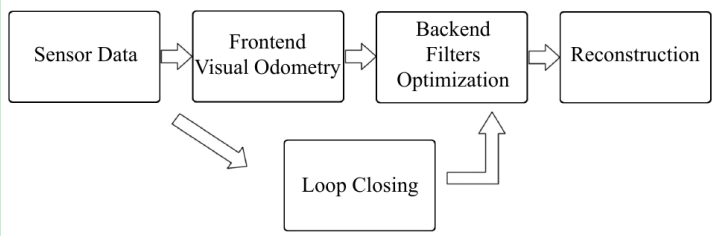

Figure 1: classic visual SLAM framework

- Sensor data acquisition: Camera images, motor encoders, IMU sensors etc
- Visual Odometry (VO)
    - Estimate the camera movement between adjacent frames and generate a rough local map
    - Wiki: visual odometry is the process of determining the position and

orientation of a robot by analyzing the associated camera images
    - Odometry = the use of data from motion sensors to estimate change in position over time
- Filtering / Optimization: Generate a fully optimized trajectory and map
- Loop Closing: Detect if the robot returned to previous position, to reduce the accumulated drift
- Reconstruction: It constructs a task-specific map based on the estimated camera trajectory

VO Area: Computer Vision (CV), Image feature extraction & matching

Prob: accumulative drift (propagated error)

Soln: backend optimization & loop closing

In A Comprehensive Survey of Visual SLAM Algorithms, the authors mention VSLAM differs from VO by 'the global consistency of the estimated trajectory and map' (rather than a belonging relationship in this book

Backend Optimization Area: State Estimation

Deal with noise

It studies (1) estimate camera movements; (2) noise in each estimation; (3) how noise is propagated; (4) how confident we have in the current estimation

Soln: Filtering, nonlinear optimization for estimating the mean and uncertainty (covariance) of the states

Loop Closing aka loop closure detection

- Solving drifting problem in VO (correcting drifts accumulated at the end of the robot's trajectory)
- E.g. of drifting: origin location != returned location
- Need to enable robot to identify the scenes it has visited before
- Intrusive methods: QR codes

- Non-intrusive methods: visual loop detection (detect similarities between images; inspired by humans), laser-based SLAM

Mapping the process of building a map (a description of the environment)

- Example of map: 2D grid map, 2D topological map, 3D point clouds, 3D meshes
- The kind of map is highly dependent on the application
- Metrical maps
  - emphasize the exact metrical locations of the objects in maps
  - Sparse metric map: store the scene into a compact form, a partial form of the scene (Sufficient for localization task)
  - Dense metric map: store all the things that are seen (Sufficient for localization and navigation task)
  - Still have some open problems (wall overlapping due to steering error)
- Topological maps
  - emphasize the relationships among map elements; consists of nodes & edges, only considering the connectivity between nodes
  - A more compact representation than metrical maps as we only consider connectivity
  - Open problems: (1) How to split a map to form nodes and edges; (2) How to use a topological map for navigation & path finding

1.3 Mathematical Formulation of SLAM Problems

- $\mathbf{x}_i$: location of the robot at time step i
- $\mathbf{x}_1, \ldots, \mathbf{x}_k$: trajectory of the robot
- y: landmark
- $y_1, \ldots, y_N$: total of N landmarks
- motion: $\mathbf{x}$ changes from step $k-1$ to step k
- motion equation: $\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$, $\mathbf{u}_k$ = input commands, $\mathbf{w}_k$ = noise
- observation: sees a landmark point $\mathbf{y}_j$ at $\mathbf{x}_k$ and generates an observation

data $\mathbf{z}_{k,j}$

- observation equation: $\mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k, \mathbf{v}_{k,j})$

- pose = position + rotation (in this book)
- The presence of noise makes the motion process stochastic: the robot will not move exactly as the command
- Basic SLAM problem: how to solve the estimate $\mathbf{x}$ (localization) and $\mathbf{y}$ (mapping) problem with the noisy control input $\mathbf{u}$ and the sensor reading $\mathbf{z}$ data?
- SLAM problem is modeled as a *state estimation problem*: How to estimate the internal, hidden state variables through the noisy measurement data?
- State estimation problem is classified based on (non)-linearity of the observation equation, and the (non)-Gaussian noise.
- Linear Gaussian (LG) system can be solved by Kalman Filter (KF)
- Non-linear non-Gaussian (NLNG) can be solved by (1) Extended Kalman Filter (EKF); and (2) nonlinear optimization
  - SOTA: graph optimization

2 Chap 1: 3D Rigid Body Motion