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

- 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

vslam_framework.png

Figure 1: classic visual SLAM framework

- 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 hrefhttps://www.mdpi.com/2218-6581/11/1/24A 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