SLAM

Antonio BUEMI
System Research and Application
Agenda

1. Introduction to SLAM
2. SLAM building blocks
3. ORB-SLAM
4. Conclusion
Definition

**SLAM**

**Simultaneous**
Find “your” position in an (unknown) environment

**Localization**
And

**Mapping**
Build and update the map of the environment
Noise adds errors to the pose estimate and map.

The **Loop Detection** (also called Place Recognition) is the ability of the system to recognize the case in which one is passing in a place already visited.

The **Loop Closure** is the process of correcting the map using that information and the fact that the landmarks are supposed to be stationary.

A mechanism **to adjust the pose** estimation is required.

Adjusting the camera pose also means **correcting the map**, so a strategy to do this is required, too.
SLAM siblings

**SfM**

**Struct from Motion**
Reconstructs 3D structure from a set of 2D images

- NO odometry
- NO computational constraints
- NO real-time

**VO**

**Visual Odometry**
Estimate odometry computing differences in camera poses between two frames taken one after another

- NO Loop Closure
- NO global mapping
- Efficiency
**SLAM tools: how to get depth information**

**Monocular camera**
- Single image
  - comparing frames taken at different camera positions

**Stereo camera**
- Left and right images
  - Correspondence between frame pairs + triangulation

**Depth camera**
- Depth information + RGB
  - time of flight
  - the pattern density of the Infra Red light emitted by the camera.

**Lidar**
- Light Detection and Ranging
  - Depth information
    - Laser
SLAM tools: descriptors

- Speeded Up Robust Features (SURF)
- Scale-invariant Feature Transform (SIFT)
- Oriented FAST and Rotated BRIEF (ORB)
- ...
Feature based & Direct methods

Feature based:
- Input Images
- Features Extraction & Matching
- SLAM

Direct:
- Input Images
- SLAM

SLAM Methods:
- Feature based & Direct methods
- SLAM
- 3D point cloud
- Dense map
SLAM: not only visual

**Monocular**
- Feature extraction
- Modelling
- Map building & camera pose estimation
- Data matching
- Loop closure detection

**Stereo**
- Feature extraction
- Modelling
- Map building & camera pose estimation
- Data matching
- Loop closure detection

**Multimodal**
- Feature extraction
- Modelling
- Sensors data fusion
- Data matching
- Loop closure detection

- RGB/Y
- Depth IMU

**Cheap HW**
- No depth infos without camera motion
- Absolute scale information lack

**Depth infos also with static cameras**
Computational overhead

**High accuracy**
Sensors data fusion required
SLAM: where can we use Neural Networks?

Patrick van der Smagt, Daniel Cremers, Thomas Brox
*FlowNet: Learning Optical Flow with Convolutional Networks*
IEEE International Conference on Computer Vision (ICCV) 2015

Eddy Ilg, Nikolaus Mayer, Tommoy Saikia, Margret Keuper, Alexey Dosovitskiy, Thomas Brox
*FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, Cordelia Schmid
*DeepFlow: Large displacement optical flow with deep matching*
IEEE International Conference on Computer Vision (ICCV), 2013

K. Tateno, F. Tombari, I. Laina, N. Navab
*CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction*
IEEE Conference on Computer Vision and Pattern Recognition, 2017

Jason R. Rambach, Aditya Tewari, Alain Pagani, Didier Stricker
*Learning to Fuse: A Deep Learning Approach to Visual-Inertial Camera Pose Estimation*
IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2016

Mark D. Gross, Alexander Morel
*MPG & STH: High-fidelity, efficient, and robust visual SLAM in the real world*
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015
The stereo images used in the training phase allow to recover the scale information of the environment.

Note that this limits the generalization: the scenes used in the testing must be similar to ones used in training.
Deep SLAM: testing

The testing framework takes monocular color images as input and produces depth maps, poses and point clouds as outputs by using Mapping-Net, Tracking-Net, and Loop-Net.

Ruihao Li, Sen Wang
*DeepSLAM: A Robust Monocular SLAM System With Unsupervised Deep Learning*
IEEE Transactions on Industrial Electronics, Vol. 68, No. 4, April 2021
ORB-SLAM2 is a feature-based monocular SLAM system

Main features

- Real time on CPU
- Small & large scale
- Indoor & outdoor environment
- Loop closure
- Relocalization
- Sparse 3D map

Raúl Mur-Artal and Juan D. Tardós
ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras
ORB: oriented FAST and rotated BRIEF

- ORB is a fusion of FAST keypoints detector and BRIEF descriptors
- It is a free alternative to SIFT and SURF and it overcomes them in computation cost and performance

free  partial scale invariant  rotation invariant
STEREO-CAM (mezzanine board) is a board designed in ST

It contains:
- No microcontroller
- 96connector to be connected to any 96board
- 2x VD56G3 OLGA80 (FoX NIR imaging sensor with HW Optical Flow accelerator)
- 2x LSM6DSRXTR (3D accelerometer, 3D gyroscope)
- 1x Ewok MZ Module V2

Two 96boards are used (for drivers availability):
- DragonBoard 410C
  - Qualcomm Snapdragon, 4 cores @ 1.4GHz
  - Avenger96 (STM32MP157)
From Version 1 to Version 2

Version 1 ORB-SLAM2 uses Oriented FAST+Rotated BRIEF.

Version 2 exploits the Optical Flow (Keypoints+MVs) retrieved by FoX.
ORBSLAM2 and ORBSLAM3

**Tracking**
- Frame
  - ORB extraction
  - IMU integration
- IMU
  - IMU integration

**Local Mapping**
- New KeyFrame insertion
  - Map update
  - IMU init
  - IMU scale refinement

**Full Bundle Adjustment**
- Map update

**Loop Closure & Map Merging**
- Maps merging
- Local Bundle Adjustment
- Place Recognition

**Key Frame**
- Initial pose estimation from last frame or relocalization
- Track Local Map KeyPoints + descriptors + matching
- New KeyFrame decision

If the tracking is lost, automatic map initialization and relocalization occur.
ORB-SLAM2 (standard) IN/OUT

STEREO-CAM

Near-InfraRed (NIR) sequence

ORB-SLAM
ORB-SLAM2 (OF-based) IN/OUT

STEREO-CAM

Near-InfraRed (NIR) sequence

Motion Vectors (MVs)
ORB-SLAM2+DragonBoard

StereoCAM → Stereo NIR images → OF → ORB-SLAM2
Platforms 2/2

ORB-SLAM3 + Raspberry

- Raspberry PI 4
- STEVAL-MKI194V1
- Ism6dso
- IMU
- FoX
- NIR images
- OF
- ORB-SLAM3
Results

**iToF**

- Image
- Depth map
- Resolution: 402x336
- StereoCAM+DragonBoard 410C
- ORB-SLAM2
- 3D points cloud + trajectory
- 6.55 fps

**Eys3D**

- Image
- Depth map
- Resolution: 1104x848
- StereoCAM+DragonBoard 410C
- ORB-SLAM2
- 3D points cloud + trajectory
- 3.78 fps
Is SLAM solved?

Short answer: *It depends on the application*
Robots

- Motion limits
  *e.g.* max speed, dynamics
- Sensors features
  *e.g.* resolution, sampling rate
- Computational resources
  *e.g.* CPU features, RAM

Environment

- 2D or 3D map?
- Presence of landmarks
  *natural or artificial*
- Amount of dynamic elements

Performances

- Target accuracy
- Success rate
- Estimation latency
- Maximum size of the map
- Maximum operation time
Examples of solved cases

1. 2D mapping of indoor environment with robot equipped with wheels and laser scanner
2. Vision-based SLAM using slowly moving robots
3. Visual-Inertial Odometry
Some open problem

Challenging environments

Fast robot dynamics
Future SLAM key requirements

**Robust performances**
- Reliability: Low failure rate for an extended time in a broad range of environments
- Autonomy: Fail-safe mechanism, Self-tuning capabilities
- Flexibility: System parameters adaptivity to the scenario

**High-level understanding**
- Geometry: Beyond basic geometry reconstruction to understand the environment
- Semantic: Additional Artificial Intelligence modules

**Resource Awareness**
- Smart system: Monitoring of the available sensing and computational resource to adjust accordingly the computational load

**Task Driven Perception**
- Selectivity: Filter irrelevant sensor data, in order to support the task
- Scalability: Adaptive map representation whose complexity may vary depending on the task
Thank you