life.augmented

SLAM

Antonio BUEMI
System Research and Application

Agenda

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

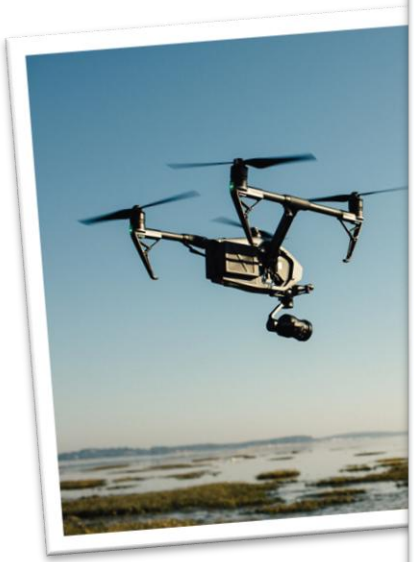
Definition

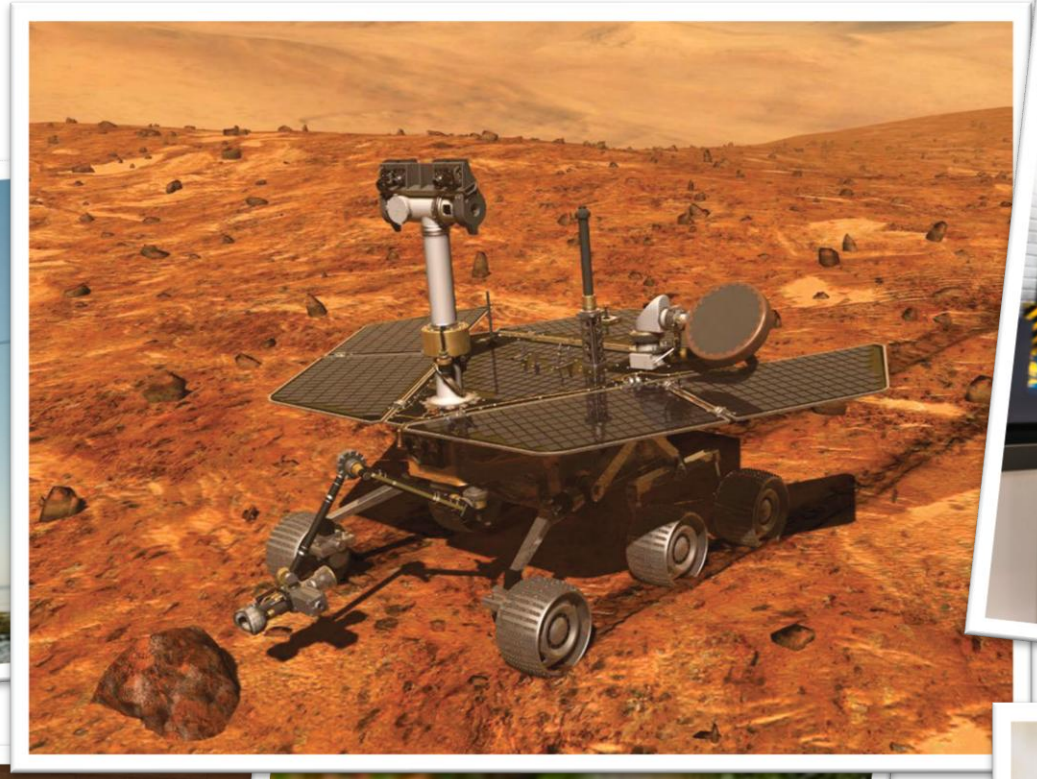
S **Simultaneous**

L **Localization**
Find "your" position in an (unknown) environment

A **And**

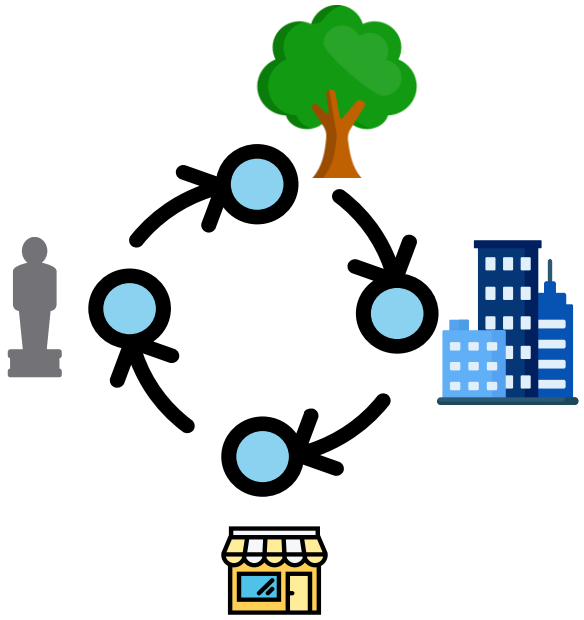
M **Mapping**
Build and update the map of the environment





life.augmented

SLAM building blocks



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.

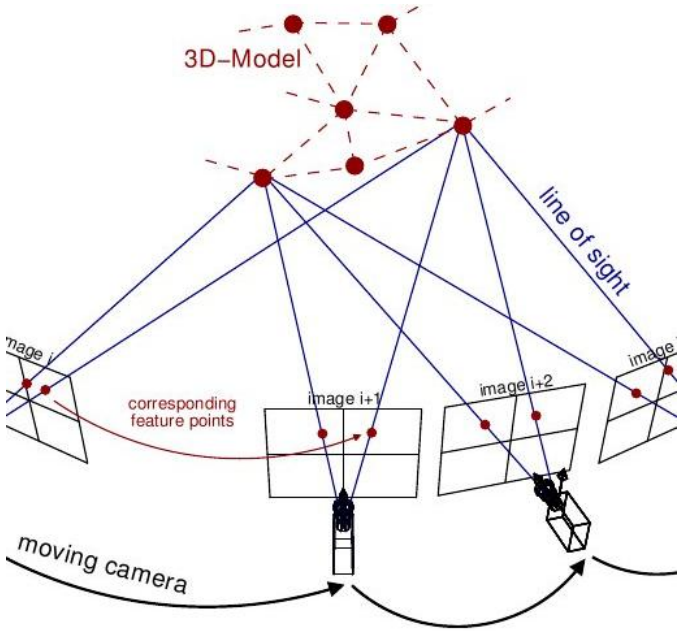
Adjusting the camera pose also means **correcting the map**, so a strategy to do this is required, too.

SLAM siblings

SfM

Struct from Motion

Reconstructs 3D struct from a set of 2D images

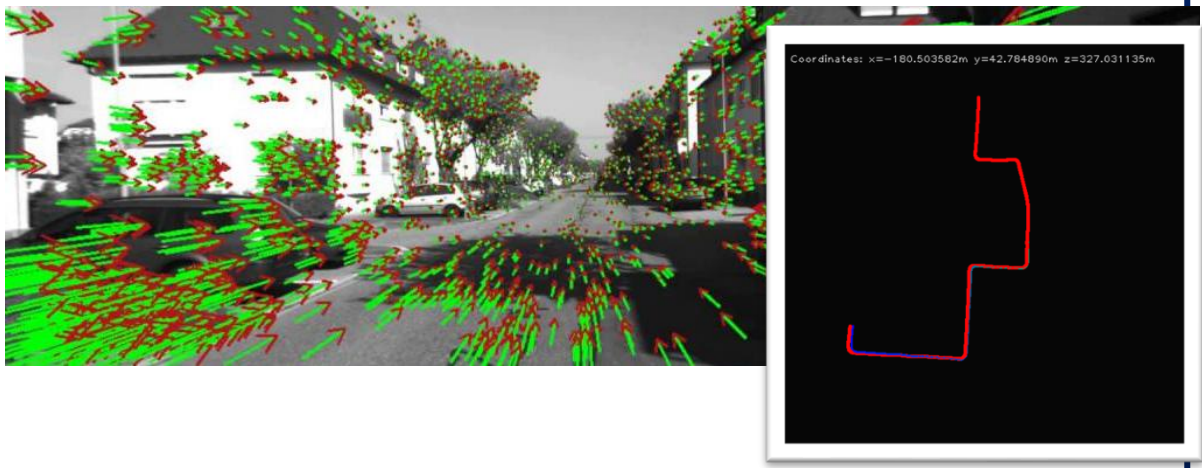


- NO odometry
- NO computational constraints
- NO real-time

VO

Visual Odometry

Estimate odometry computing differences in camera poses between two frame taken one after another

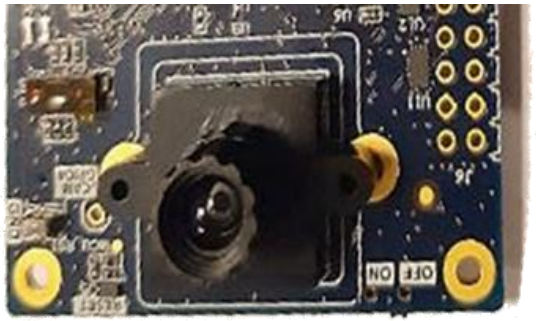


- NO Loop Closure
- NO global mapping
- Efficiency

life.augmented

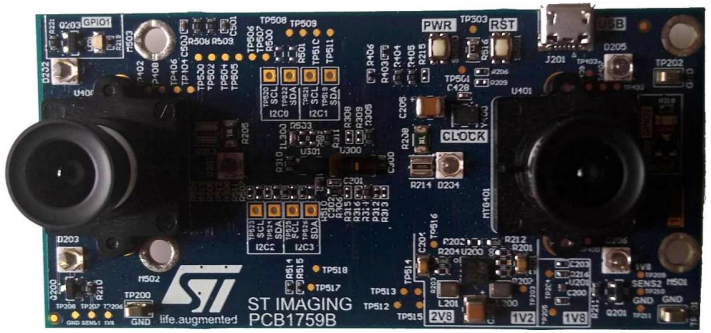
SLAM tools: how to get depth information

Monocular camera



Single image
- comparing frames taken at different camera positions

Stereo camera



Left and right images
- Correspondence between frame pairs + triangulation

Depth camera

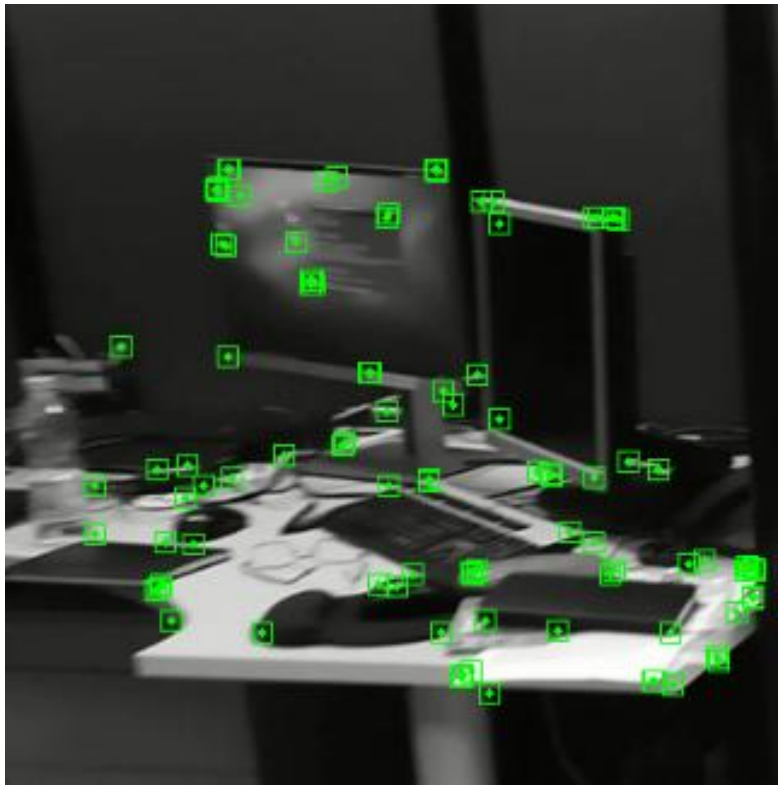
Depth information+RGB
- time of flight
- the pattern density of the Infra Red light emitted by the camera.

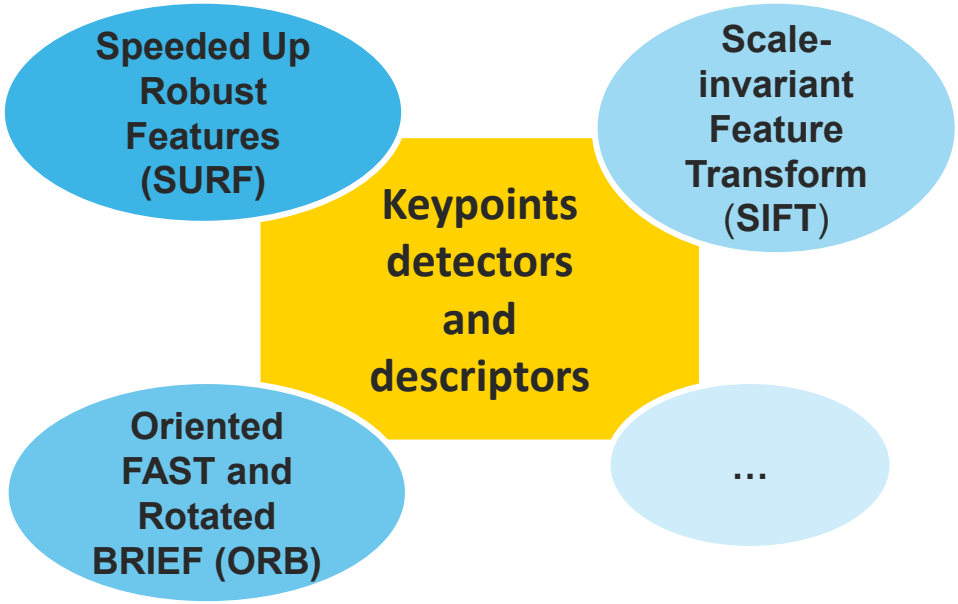
Lidar

Light **D**etection **a**nd **R**anging

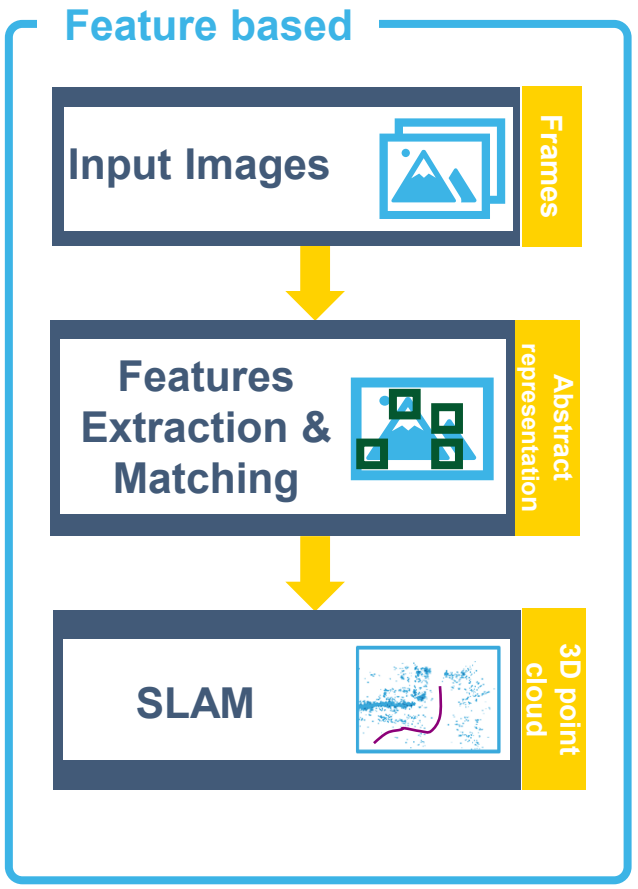
Depth information
- Laser

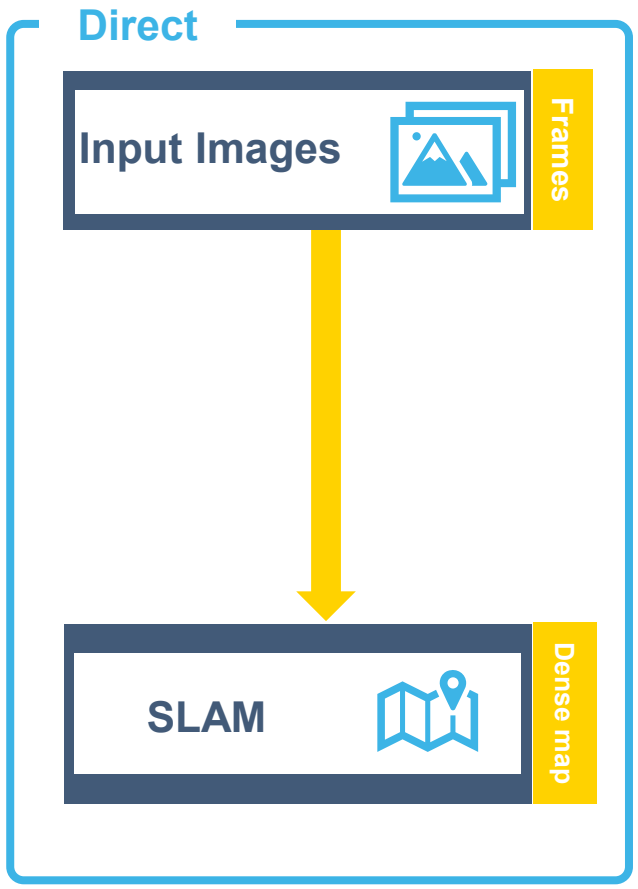
SLAM tools: descriptors





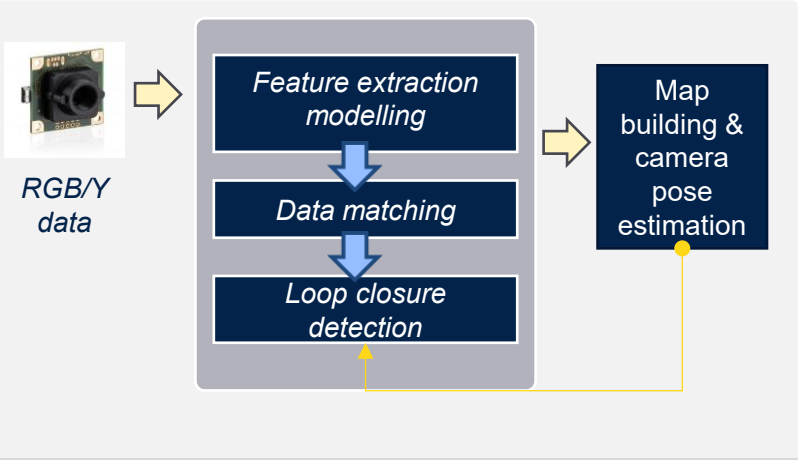
Feature based & Direct methods





SLAM: not only visual

Monocular

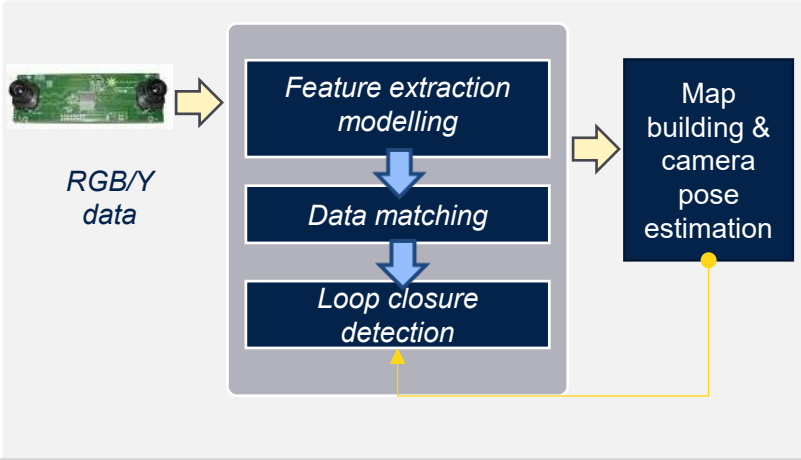


Cheap HW

No depth infos without camera motion

Absolute scale information lack

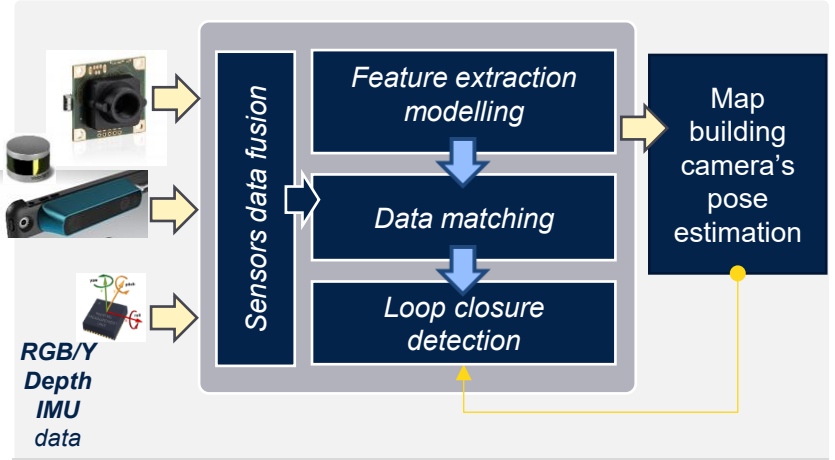
Stereo



Depth infos also with static cameras

Computational overhead

Multimodal

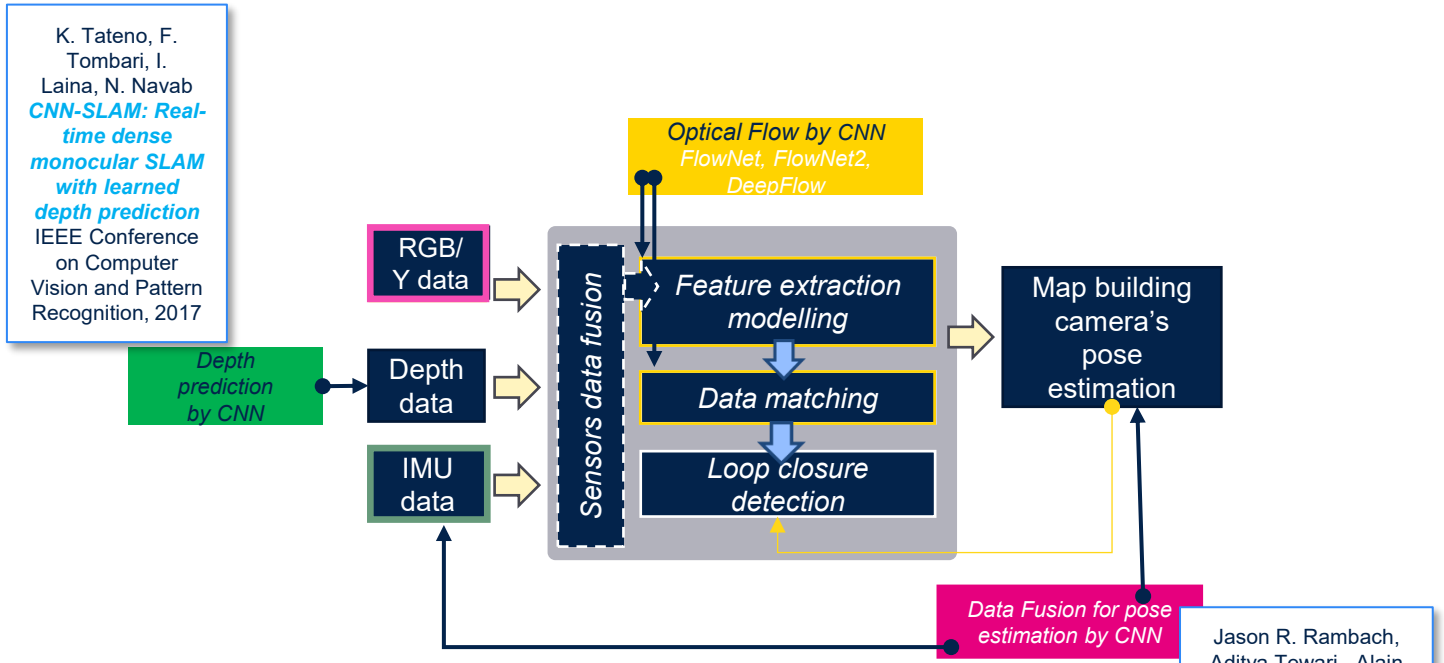


High accuracy

Sensors data fusion required

life.augmented

SLAM: where can we use Neural Networks?

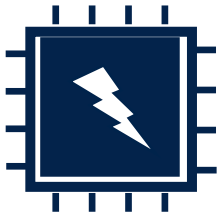


life.augmented

ORB-SLAM

ORB-SLAM2

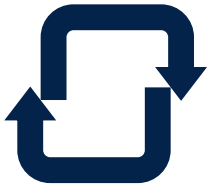
ORB-SLAM2 is a feature-based monocular SLAM system



real time
on CPU

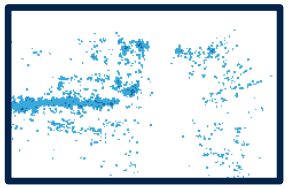
small & large
scale

indoor & outdoor
environment



loop closure

relocalization



sparse
3D map

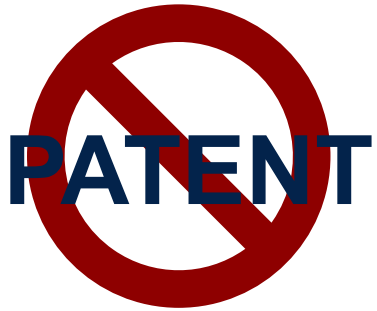
Main features

Raúl Mur-Artal and Juan D. Tardós
ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras
IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, 2017

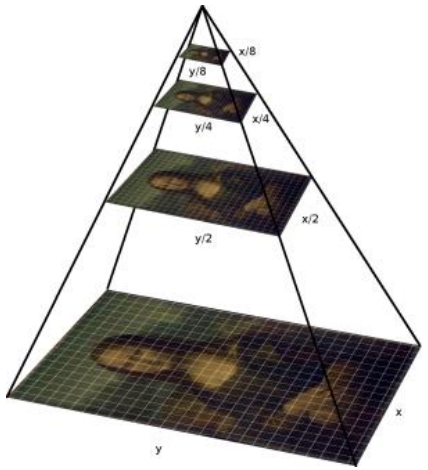
life.augmented

ORB: oriented FAST and rotated BRIEF

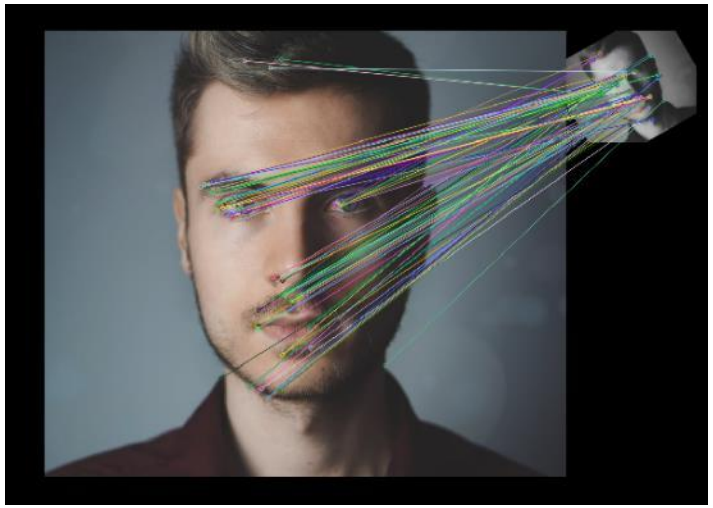
- ORB is a fusion of FAST keypoints detector and BRIEF descriptors
- It is a free alternative to SIFT and SURF and it overcomes them in computation cost and performance



free



partial scale invariant



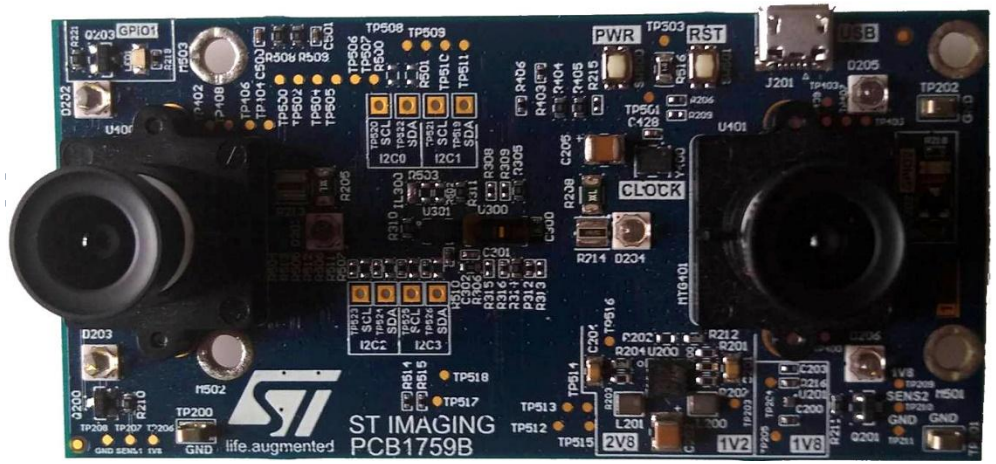
rotation invariant

STEREO-CAM board

- 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)

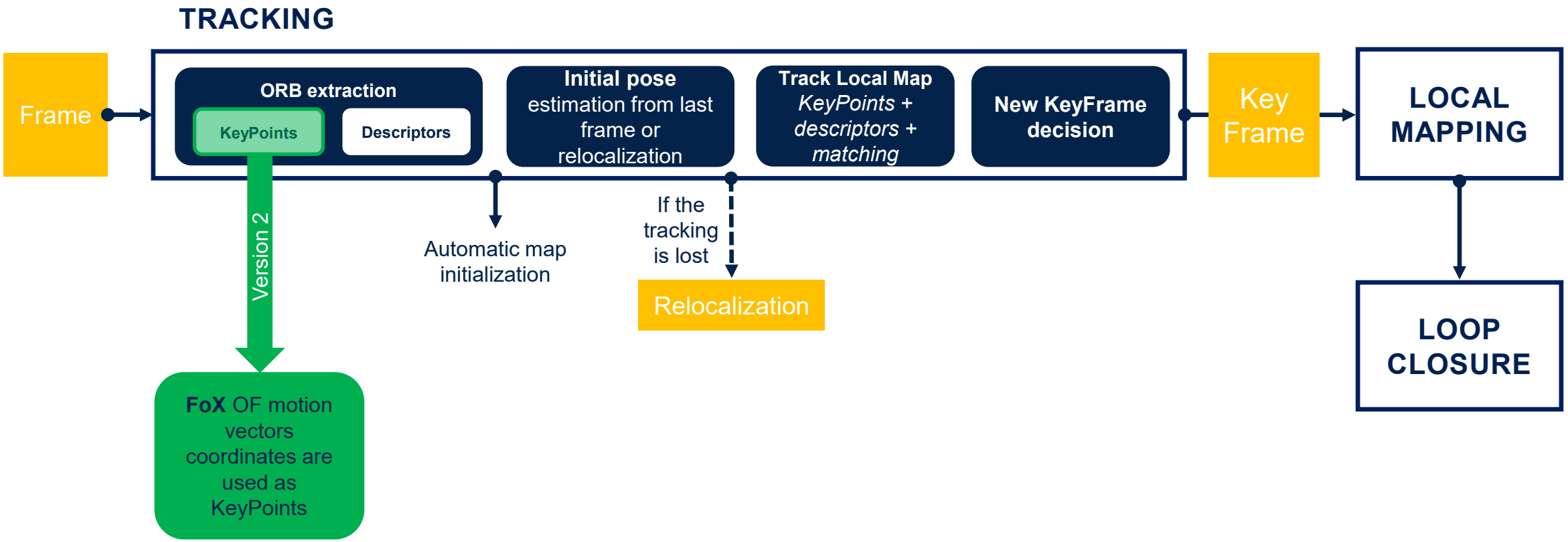


life.augmented

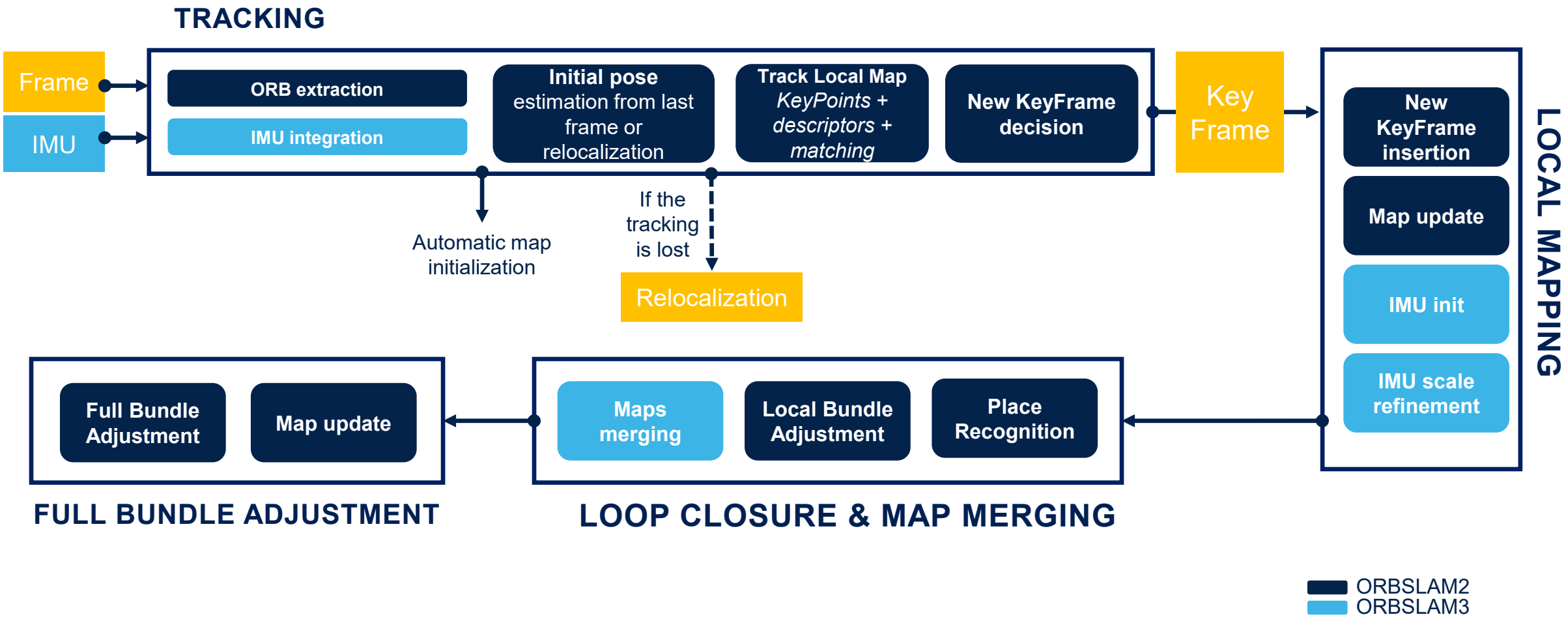
From Version 1 to Version 2

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

Version 2 exploit the Optical Flow (Keypoints+MVs) retrieved by FoX



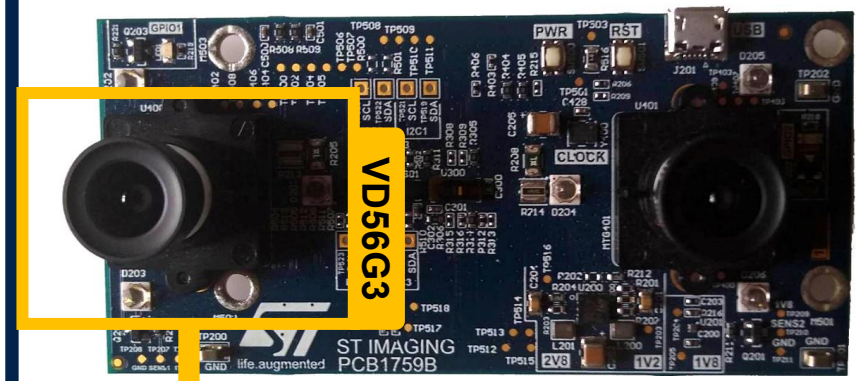
ORBSLAM2 and ORBSLAM3



life.augmented

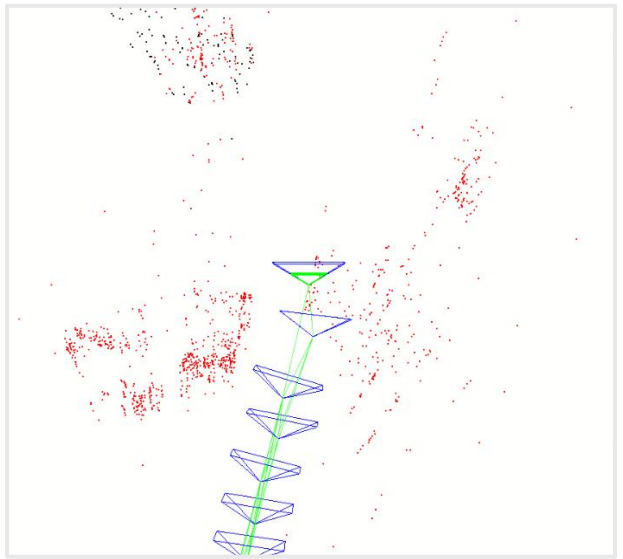
ORB-SLAM2 (standard) IN/OUT

STEREO-CAM



Near-InfraRed (NIR) sequence

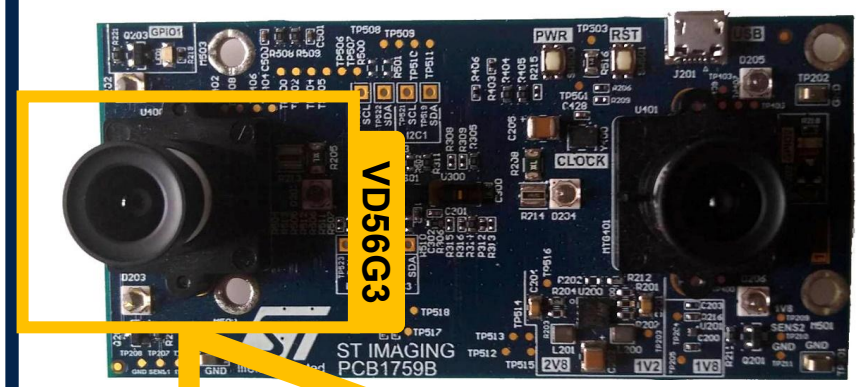
ORB-SLAM



life.augmented

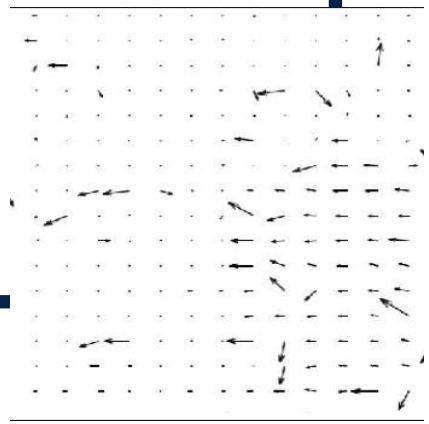
ORB-SLAM2 (OF-based) IN/OUT

STEREO-CAM

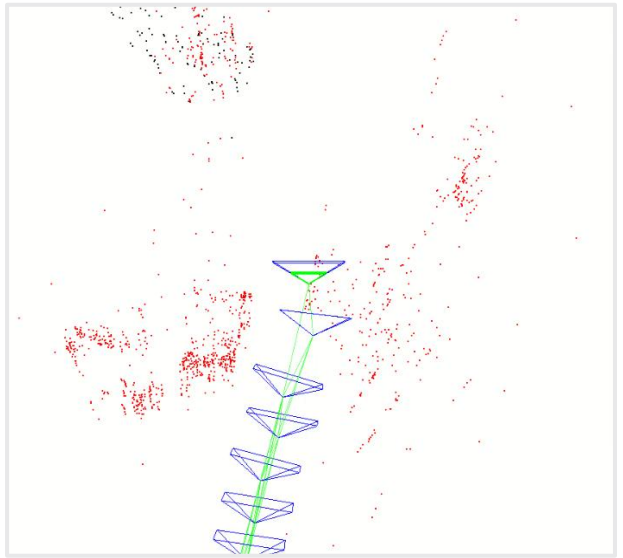


Near-InfraRed (NIR) sequence

Motion Vectors (MVs)



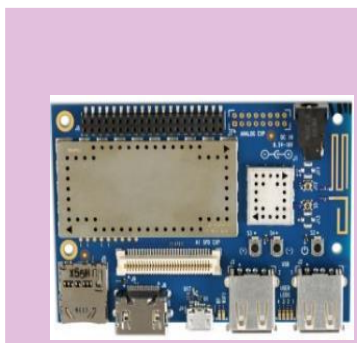
ORB-SLAM

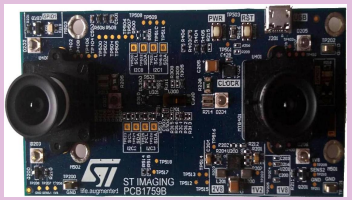


life.augmented

Platforms 1/2

ORB-SLAM2+DragonBoard

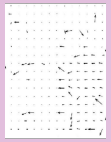




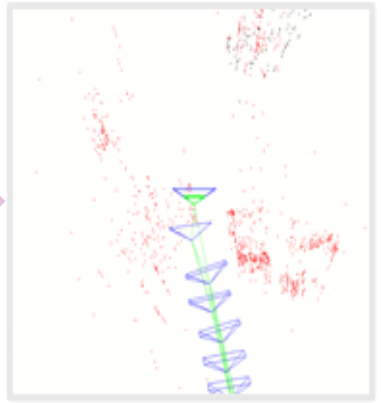
StereoCAM

Stereo NIR images

OF

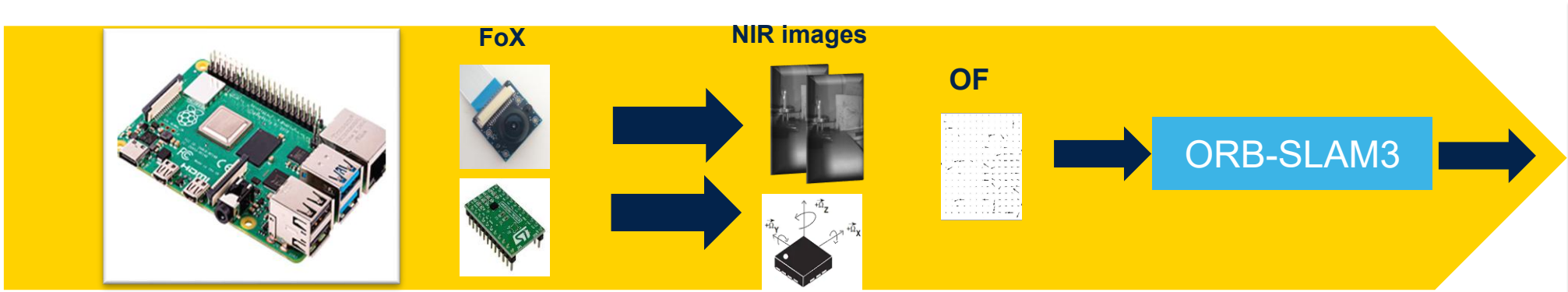


ORB-SLAM2



Platforms 2/2

ORB-SLAM3+Raspberry



Raspberry PI 4

STEVAL-MKI194V1
Ism6dso

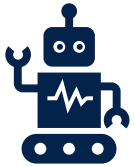
IMU

Conclusion

Is SLAM solved?

Short answer: *It depends on the application*

Context definition aspects



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

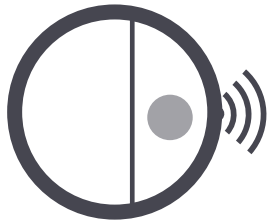
Estimation latency

Maximum size of the map

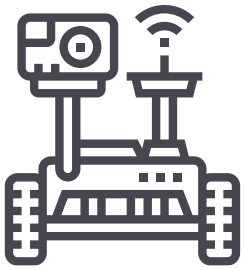
Maximum operation time

life.augmented

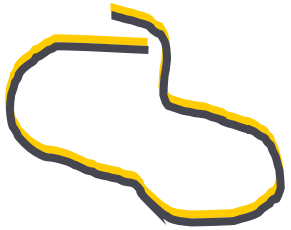
Examples of solved cases



2D mapping of indoor environment with robot equipped with wheels and laser scanner

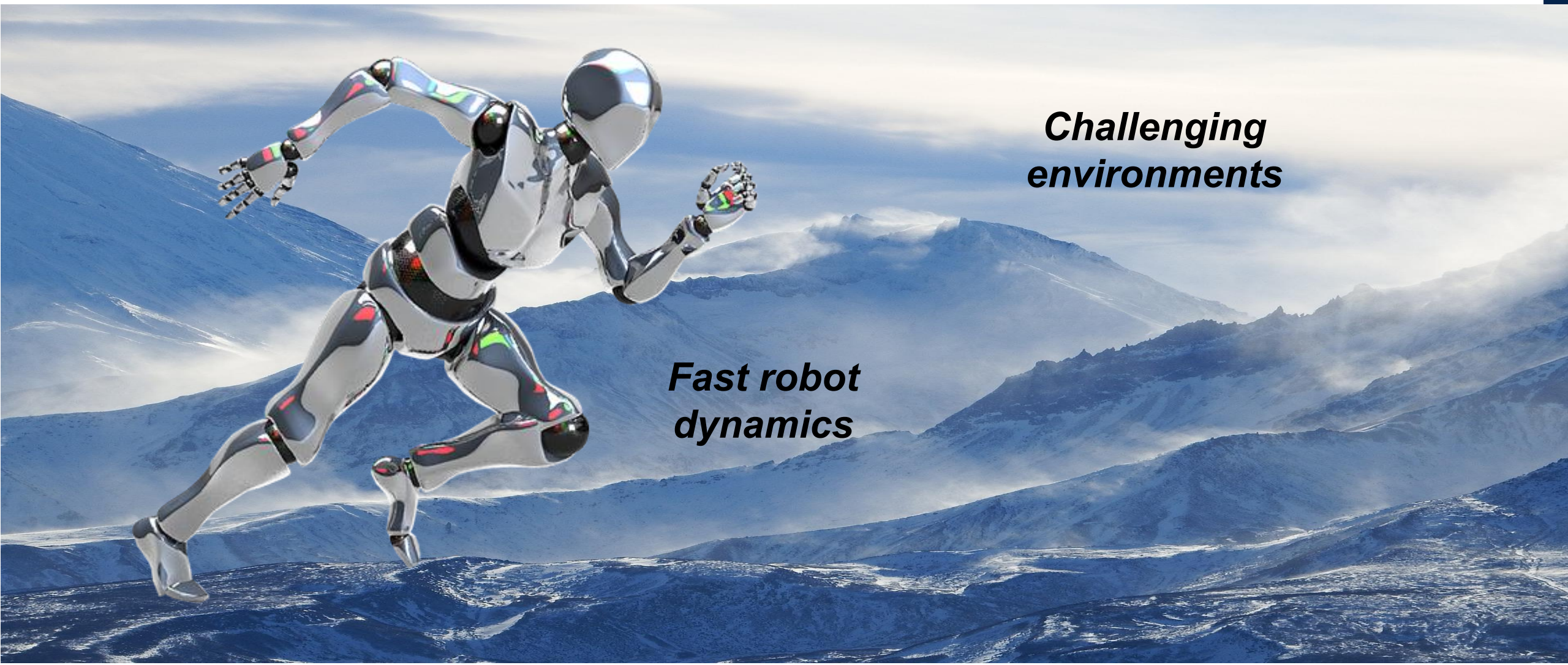


Vision-based SLAM using slowly moving robots



Visual-Inertial Odometry

Some open problem



Future SLAM key requirements

Robust performances

Reliablility
Low failure rate for an extendend time in a broad range of environments

Automomy
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

life.augmented

Thank you

life.augmented