Monocular Visual–Inertial Odometry for Outdoor Localization on Martian-like Terrain

Administrative Info

Semester: Spring 2026
Project: Monocular Visual–Inertial Odometry for Outdoor Localization on Martian-like Terrain
Responsibles: Arno Laurie (Software Systems Engineer) & Giovanni Ranieri (Vice-President Research)
Contact: arno.laurie.@epfl-xplore.ch & giovanni.ranieri@epfl-xplore.ch

Motivation

EPFL Xplore is a student-led swiss robotics association, part of the MAKE Initiative at EPFL. It primary builds a Martian-like Rover each year to compete in the European Rover Challenge (ERC) in Krakow, against other universities. The team is composed of around 50 students (management & engineering). In its rank, the software team of EPFL Xplore is composed of navigation, robotic arm and control station teams, each one developing efficient solutions to bring the Rover alive.

Reliable localization is a core challenge for autonomous navigation in outdoor, unstructured environments. Visual–Inertial Odometry (VIO) is a key enabling technology for such systems, as it allows state estimation using only a monocular camera and an IMU, without reliance on GNSS. This project aims to develop a **filter-based VIO pipeline** integrated in ROS 2, suitable for deployment on a rover operating in Martian-like terrain.

The Multi-State Constraint Kalman Filter (MSCKF) is a filter-based approach to Visual–Inertial Odometry that estimates the motion of a platform by fusing high-rate inertial measurements with visual observations from a monocular camera. In MSCKF, the system state consists of the inertial state (position, velocity, orientation, and IMU biases) augmented with a sliding window of past camera poses. The inertial measurements are used to propagate the state forward in time using motion models, while visual feature tracks observed across multiple camera frames impose geometric constraints between these camera poses.
By combining accurate inertial propagation with visual constraints, MSCKF provides metric-scale odometry using only a monocular camera and an IMU, making it well suited for real-time robotic applications with limited computational resources.

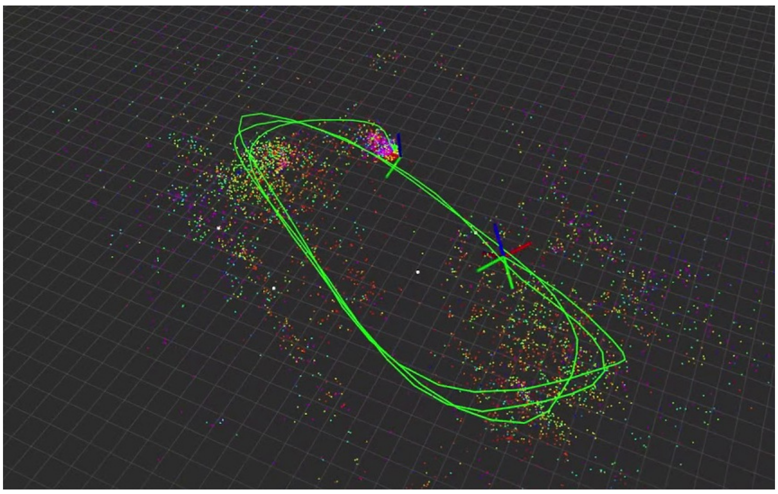

Figure 1: Example MSCKF-based visual–inertial odometry implementation.https://www.youtube.com/watch?v=vaia7iPaRW8

Objectives

The main objective is to design and implement a complete **monocular VIO pipeline based on the Multi-State Constraint Kalman Filter (MSCKF)**.

At the end of the project, the student will:

1. Implement a real-time ROS 2 VIO system using a monocular RGB camera and a high-rate 9-axis IMU.
2. Validate the system on public datasets and real outdoor experiments.
3. Analyze estimation accuracy, drift, and robustness.
4. Deliver a final report documenting the design, implementation, and experimental results.

Requirements

EPFL Xplore is looking for a motivated student that wants to tackle real software challenges in a well-established association. For this project, we are particularly interested in someone with the following qualifications:

- Basic familiarity with ROS2 and OpenCV.
- Strong programming skills in C++.
- Comfort with Linux-based development.
- Interest in state estimation, robotics, and sensor fusion
- Good analytical mindset, attention to detail, and the ability to communicate results clearly.
- Clear written & verbal communication skills, ready to document your design decisions.
- Ability to work both independently and collaboratively within a student team.

Hardware Recommendations

1. Monocular RGB camera (USB, 1080p recommended).
2. $\geq$200 Hz 9-axis IMU (gyroscope + accelerometer + magnetometer).
3. Microcontroller with USB, UART or SPI interface for IMU data streaming to a PC.

Software Stack Recommendations

1. C++
2. ROS 2 Humble
3. Eigen3
4. OpenCV
5. Sophus

Resources

- ROS2 Introduction: `https://docs.ros.org/en/humble/index.html`
- Intro to Autonomous Robots: `https://rpg.ifi.uzh.ch/docs/teaching/2025/Ch4_AMRobots.pdf`
- VIO Learning Resources: `https://rpg.ifi.uzh.ch/teaching.html`
- IMU Calibration: `https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model`
- Datasets for testing: `https://ethz-asl.github.io/datasets/euroc-mav/`

- MSCKF Vision Inertial Navigation Paper : https://intra.ece.ucr.edu/~mourikis/papers/MourikisRoumeliotis-IC pdf
- Implementation example: https://github.com/rpng/open_vins/