Environment Mapping for Autonomous Cars Using ORB-SLAM3 with Visual-Inertial Data

Group Number - 9

Team Members

- Alay Shah
- Akshaj Raut
- Varun Raghavendra
- Priyanshu Ranka

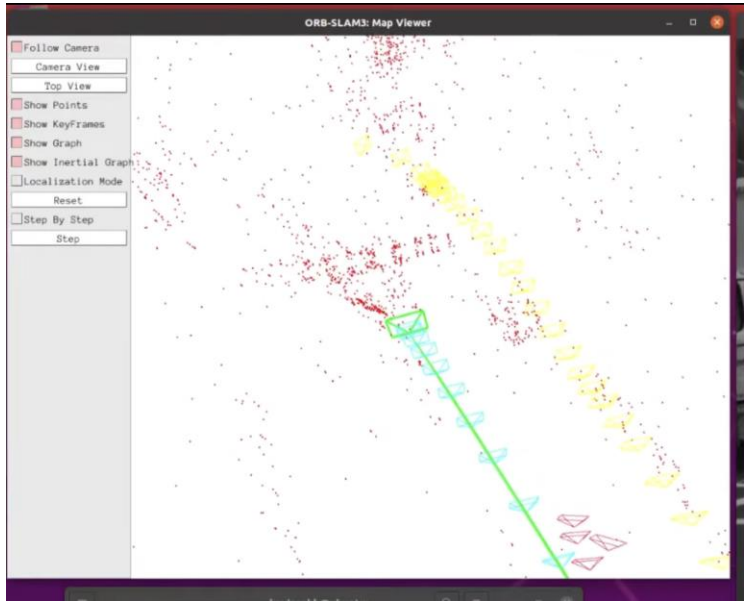
Index

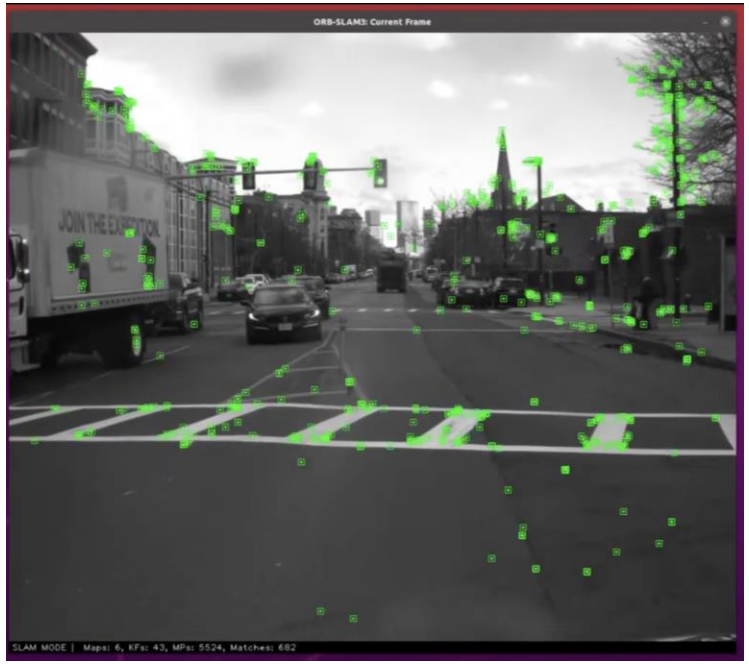
Intro to Visual-Inertial SLAM

Cameras are the closest sensors available that resembles human perception. Moreover, it beats its counterpart sensors such as LiDAR and RADAR with weight and cost, which makes it a very attractive option for autonomous systems.

Visual Inertial SLAM enhances robustness and accuracy by combining :

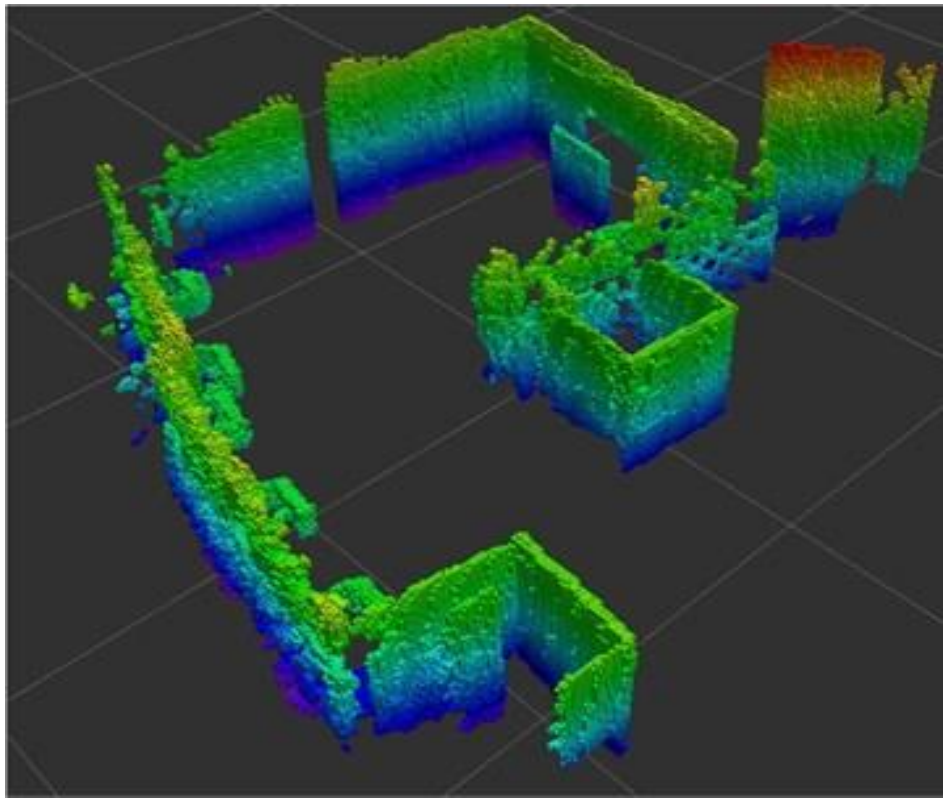
- Camera data for visual features.
- IMU readings for scale and motion estimation.
- Quickly estimates scale, gravity, and biases using MAP estimation.
- Enables robust tracking in fast motion or poor visibility.





ORB-SLAM3

- **What is ORB-SLAM3?**
  - A **feature-based SLAM system** supporting:
    - Monocular, Stereo, RGB-D cameras.
    - Pin-hole and fisheye lens models.
  - First SLAM system for **visual**, **visual-inertial**, and **multi-map** SLAM.



Key Features of ORB-SLAM3

- **Tightly-Integrated Visual-Inertial SLAM**:
- Based on **Maximum-a-Posteriori (MAP)** estimation framework.
- Operates robustly even during **IMU initialization**.
- **Multiple Map System**
- Relies on **improved place recognition** for better recall.
- Seamless merging of maps when revisiting previously mapped areas.
- **Keyframe-Based Bundle Adjustment**:
- Reuses widely separated keyframes for **high-parallax observations**.
- Enhances accuracy by leveraging **all mapping data**.

How ORB-SLAM3 Works?

Tracking:

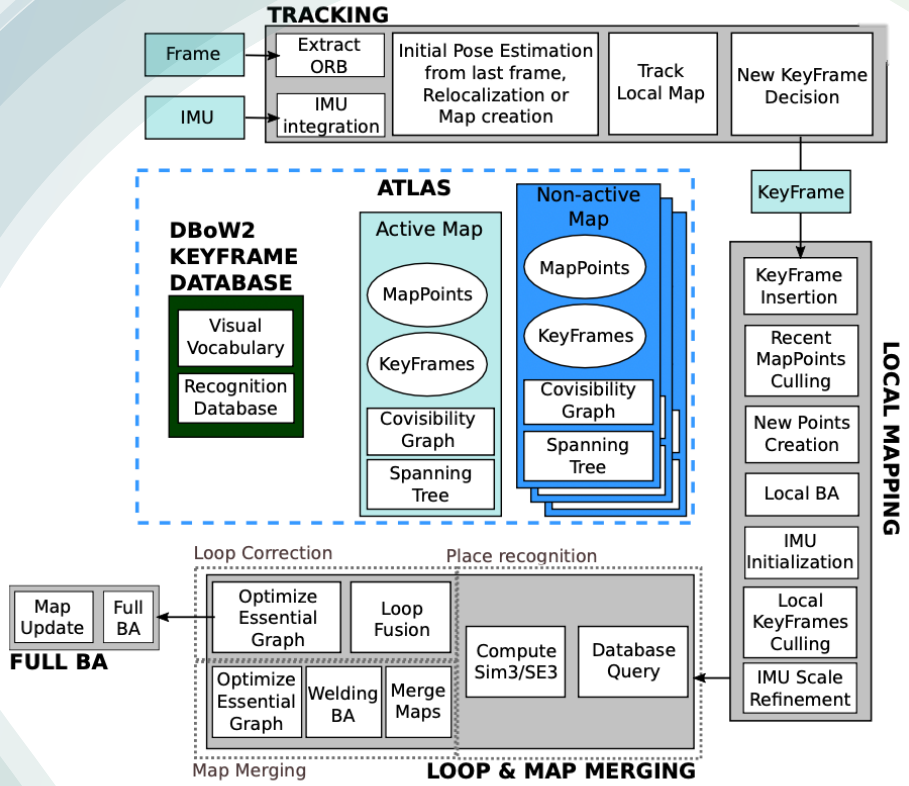
- Estimates camera pose using reprojection error.
- Handles localization when tracking is lost.

Local Mapping:

- Adds and optimizes keyframes and map points.
- Uses Local Bundle Adjustment (BA) for refinement.

Loop Closure and Map Merging:

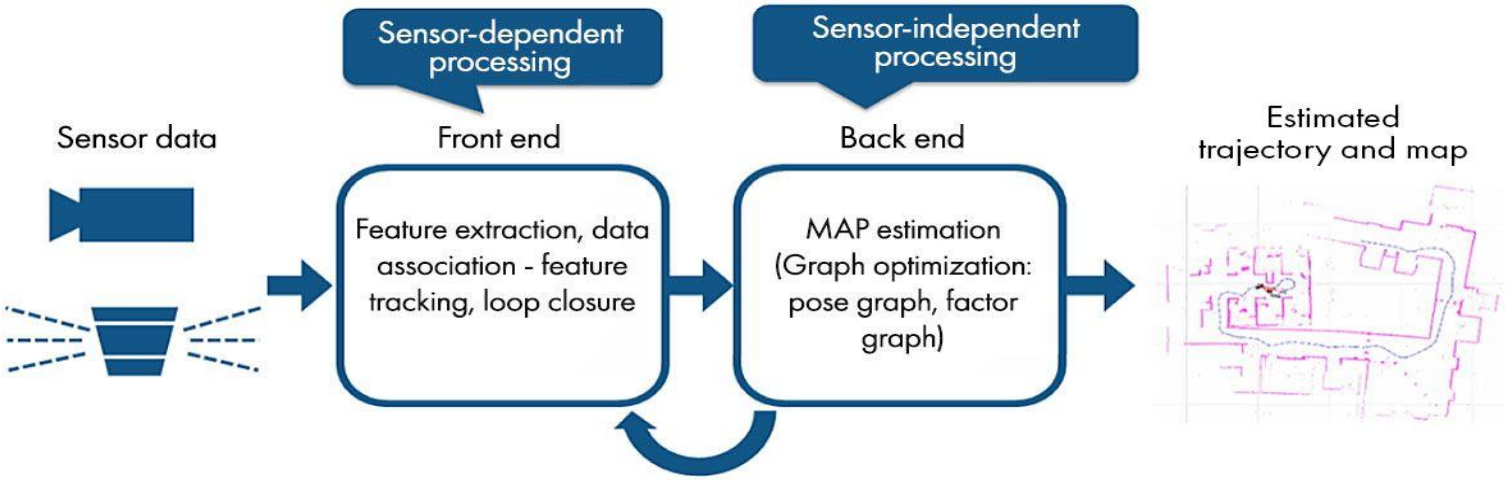
- Detects revisited areas using place recognition (DBoW2).
- Performs map merging for seamless integration.

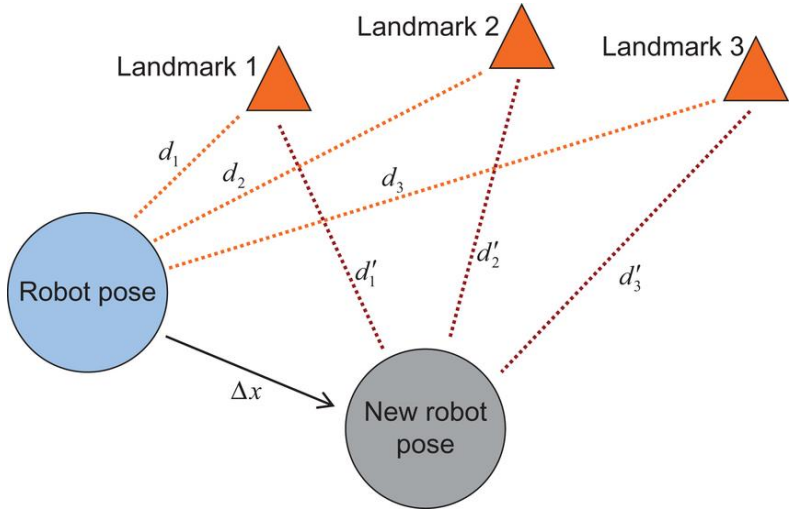


Main system components of ORB-SLAM3

Algorithm

1. **Initialization :** System initialization, camera calibration .
2. **Data Collection**: Capture environment data using cameras & IMUs and first frame processing.
3. **Landmark Identification**: Feature detection and feature matching. Detect distinct features as reference points.
4. **Pose Estimation**: Calculate position and use matched features to estimate the camera pose relative to the previous frame or the map using Perspective-n-Point (PnP) algorithm.
5. **Map Construction**: Update map with identified landmarks.
6. **Localization**: Compare sensor data to the map to determine position.
7. **Exploration**: Expand the map by identifying new landmarks and getting rid of redundant landmarks.
8. **Loop Closure**: Correct map errors by recognizing visited areas.
9. **Output**: Generate a detailed map and real-time location data.





Standard ORB SLAM 3 Performance Metrics

RMS ATE : Root Mean Squared Absolute Trajectory Error
Total Tracking Time : Pose Prediction Time, Landmark Tracking Time, New Keyframe Decision, IMU Integration
Total Mapping Time : Keyframe Insertion, Map Point Culling/Creation, Local Bundle Adjustment, Keyframe Culling
Map Size

	System	ORB-SLAM2	ORB-SLAM3	ORB-SLAM3	ORB-SLAM3	ORB-SLAM3
Settings	Sensor	Stereo	Monocular	Stereo	Mono-Inertial	Stereo-Inertial
	Resolution	752×480	752×480	752×480	752×480	752×480
	Cam. FPS	20Hz	20Hz	20Hz	20Hz	20Hz
	IMU	-	-	-	200Hz	200HZ
	ORB Feat.	1200	1000	1200	1000	1200
	RMS ATE	0.035	0.029	0.028	0.021	0.014
Tracking	Stereo rect.	3.07±0.80	-	1.32±0.43	-	1.60±0.74
	ORB extract	11.20±2.00	12.40±5.10	15.68±4.74	11.98±4.78	15.22±4.37
	Stereo match	10.38±2.57	-	3.35±0.92	-	3.38±1.07
	IMU integr.	-	-	-	0.18±0.11	0.22±0.20
	Pose pred	2.20±0.72	1.87±0.68	2.69±0.85	0.09±0.41	0.15±0.71
	LM Track	9.89±4.95	4.98±1.65	6.31±2.85	8.22±2.52	11.51±3.33
	New KF dec	0.20±0.43	0.04±0.03	0.12±0.19	0.05±0.03	0.18±0.25
	Total	37.87±7.49	21.52±6.45	31.48±5.80	23.22±14.98	33.05±9.29
Mapping	KF Insert	8.72±3.60	9.25±4.62	8.03±2.96	13.17±7.43	8.53±2.17
	MP Culling	0.25±0.09	0.09±0.04	0.32±0.15	0.07±0.04	0.24±0.24
	MP Creation	36.88±14.53	22.78±8.80	18.23±9.84	30.19±12.95	23.88±9.97
	LBA	139.61±124.92	216.95±188.77	134.60±136.28	121.09±44.81	152.70±38.37
	KF Culling	4.37±4.73	18.88±12.217	5.49±5.09	26.25±17.08	11.15±7.67
	Total	173.81±139.07	266.61±207.80	158.84±147.84	191.50±80.54	196.61±54.52
Map Size	KFs	278	272	259	332	135
	MPs	14593	9686	14245	10306	9761

Tested on EuRoC MAV Dataset in the original ORB SLAM 3 Paper

Our Experimental Setup

System Setup to run ORB-SLAM3 :

Operating System:

- Ubuntu 20.04 for ROS1 Noetic
- C++ 14
- CMake>=3.9
- Python 2.7
- Virtual Machine

Hardware:

- 8 GB RAM

ORB-SLAM3 creates a **map** consisting of keyframes and 3D points.
Keyframes store image data, camera pose, and feature associations, all of which consume memory.

- A quad-core CPU

ORB-SLAM uses a **multi-threaded design**

- Optional: GPU for accelerated computation (not required by default).

Dependencies:

- CUDA (For Accelerated Feature Detection and Matching)
- Eigen3 (for linear algebra).
- Pangolin (for visualization).
- OpenCV 4.2 and OpenCV 3.2 both were built from source
  features2d - feature detection and descriptor computation
  imgproc - resizing, grayscale conversion
- **g2o**: A general framework for optimizing graph-based nonlinear error functions.

Sensor	Monocular	Monocular + Inertial	Stereo	Stereo + Inertial
Camera Resolution (pixels)	1280x720	1280x720	1280x720	1280x720
Camera Frame Rate (FPS)	30–60	30–60	30–60	30–60
Baseline Distance (cm)	Depth is inferred through motion and triangulation, resulting in scale ambiguity.	-	10-20	10-20
IMU Data Rate (Hz)		200	-	200

Our Experimental Setup

Folder Structure

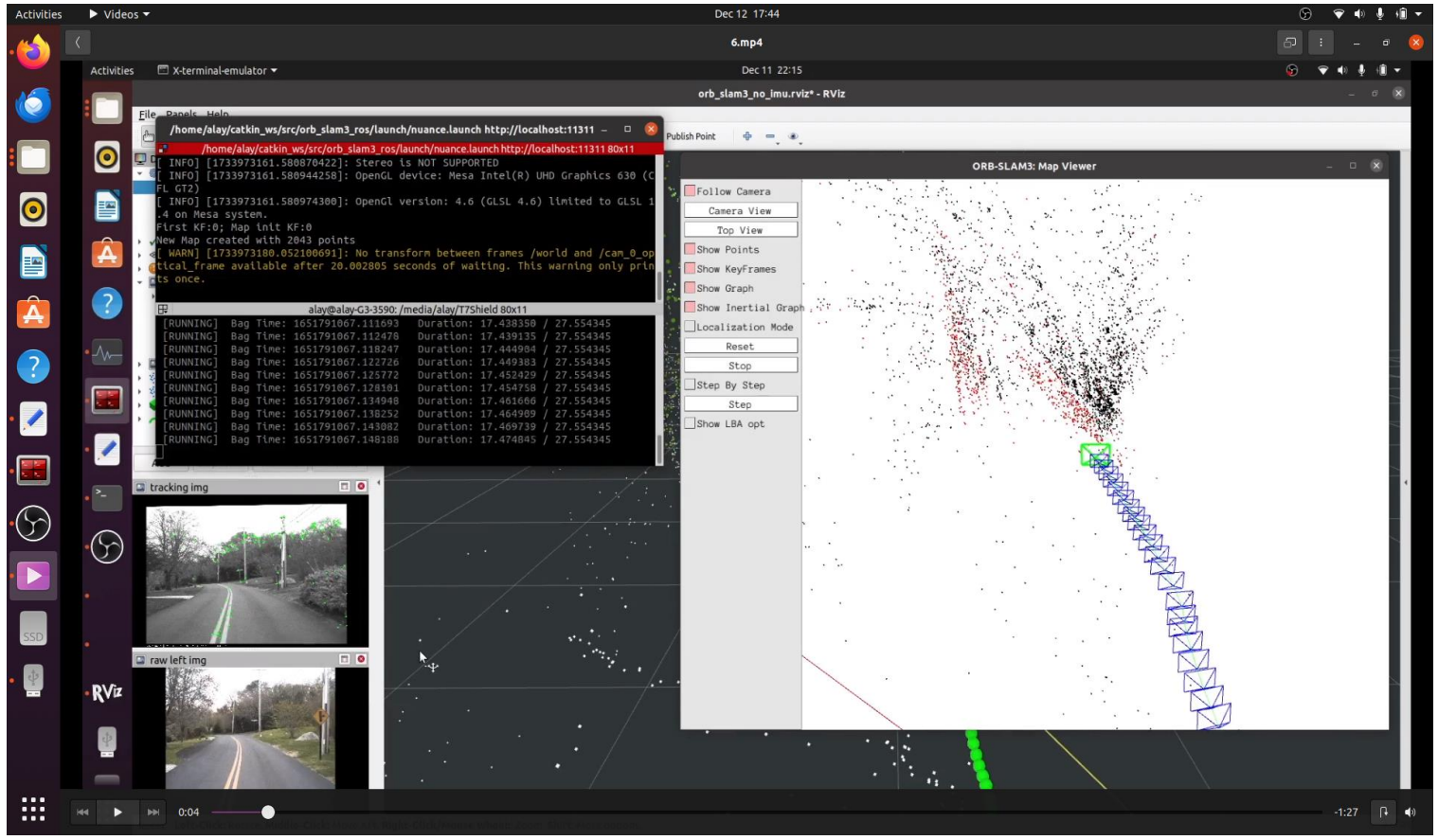
```
home
|___ ubuntu
        |__ ros2_ws
                |__ OpenCV
                |__ Pangolin
                |__ orb_slam3_ros
```

Some key folders in the orb_slam3_ros repository :

- **config**: Contains configuration files for camera parameters and system settings.
- **evaluation**: Holds scripts and data for evaluating the performance of the SLAM system.
- **include**: Contains header files for the ORB-SLAM3 library.
- **launch**: Contains ROS launch files for starting various ORB-SLAM3 configurations.
- **Examples**: Provides example scripts and code to demonstrate how to use ORB-SLAM3.
- **Vocabulary**: Contains pre-trained vocabulary files used for feature matching and visual SLAM.
- **Thirdparty**: Includes third-party libraries or dependencies used by ORB-SLAM3, such as DBoW, Sophus, g20

All screen recordings of our trials are logged here
[Videos]

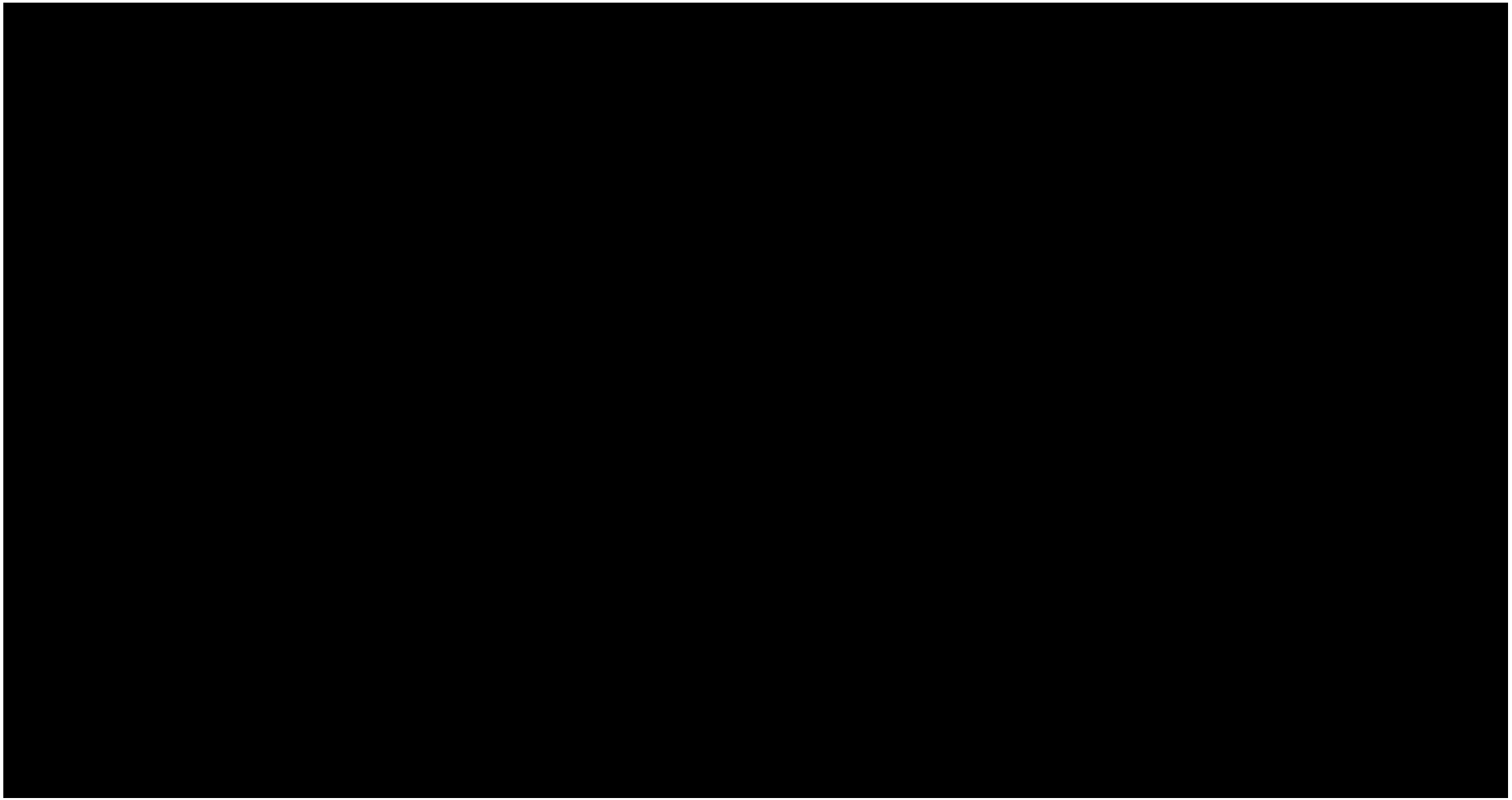
Nuance Dataset



ORB-SLAM3 on NUance Stereo Data: Performance Overview

- **General Results**: ORB-SLAM3 provided **decent results** for most parts of the dataset, with good feature tracking and map construction in areas with sufficient texture.
- **Challenges**:
  - **Feature Loss in Low-Texture Areas**: In regions with minimal texture (e.g., open spaces or areas with few distinguishing features), the system struggled to extract and track features.
  - **Speed-Related Issues**: When the car was driving faster, the lower frame rate (8 Hz) led to **feature loss** and reduced tracking accuracy.
- **ORB Parameters Used**:
  - **Number of Features**: 25,000
  - **Scale Pyramid Levels**: 12
  - **Scale Factor**: 1.2
  - **Initial FAST Threshold**: 20
  - **Minimum FAST Threshold**: 7
- **Conclusion**: While the system performed well in textured environments, adjustments like higher frame rates or enhanced feature extraction techniques may be needed for challenging scenarios.

Our Results on NUance Dataset



Test Dataset

Details of dataset

Euroc MH01 (Easy Machine Hall) dataset is commonly used for evaluating the performance of SLAM algorithms. This dataset contains a set of stereo images and IMU data, designed for testing visual-inertial SLAM algorithms. When running **ORB-SLAM3** on this dataset, it performs the tracking, mapping, and loop closure effectively, with each mode exhibiting different strengths. Ref

The sensor configuration is as follows :

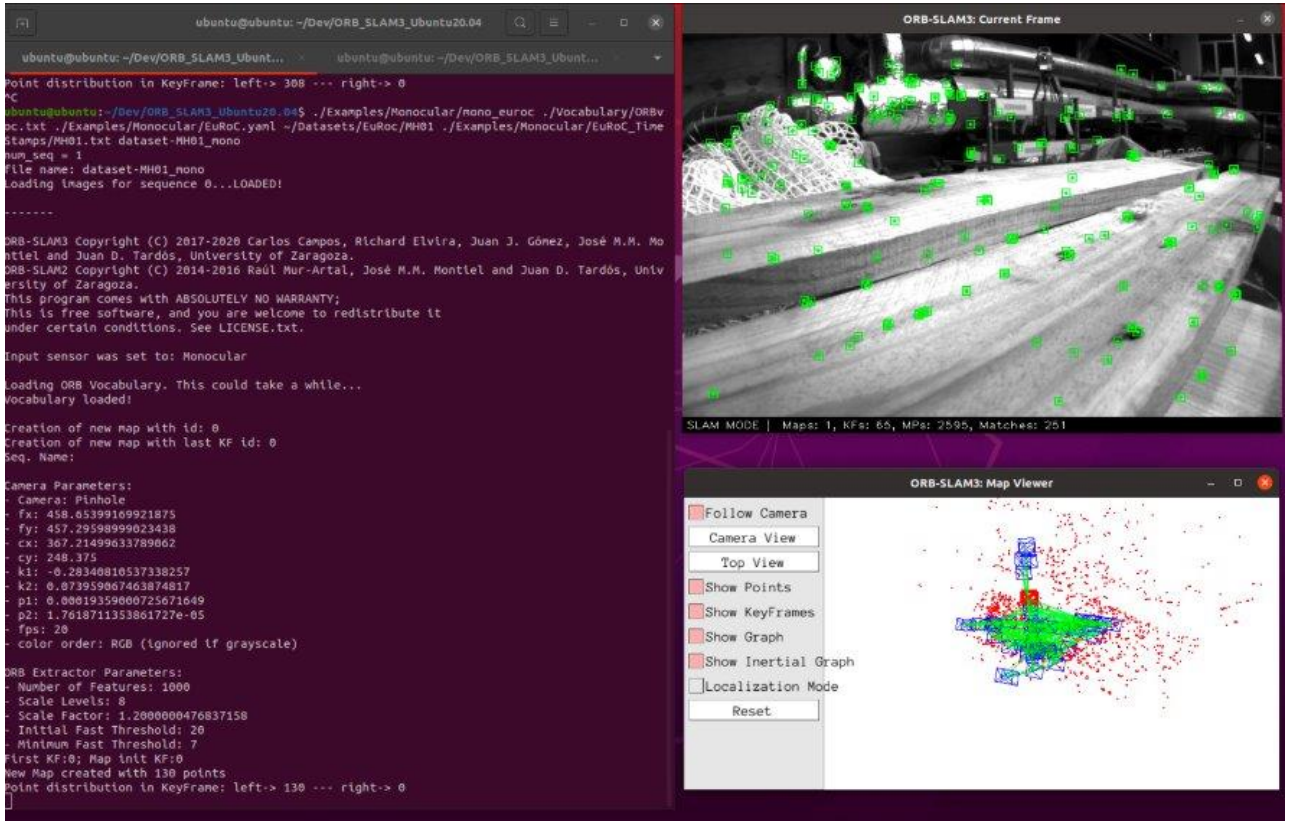
Stereo Images (Aptina MT9V034 global shutter, WVGA monochrome, 2×20 FPS)
MEMS IMU (ADIS16448, angular rate and acceleration, 200 Hz)

We ran the following modes on the aforementioned dataset :

Monocular
Monocular + Inertial
Stereo
Stereo + Inertial

ORB Extractor Parameters for EuRoC Dataset : -

Number of Features: 1000 -
Scale Levels: 8 -
Scale Factor: 1.2000000476837158 -
Initial Fast Threshold: 20 -
Minimum Fast Threshold: 7
First KF:0;
Map init KF:0
New Map created with 308 points



Snapshot of Monocular ORB-SLAM3 on our system

Key Inferences and Conclusion on ORB-SLAM3

Mono Mode

- **Features Required**:
  - Relies on **visual features** (ORB features) extracted from a single camera view.
  - **Keyframe selection** and tracking are crucial, requiring robust visual features in the environment.
- **Performance**:
  - Works well in moderate motion but may struggle with **drift** over long sequences, especially if the camera faces repetitive or textureless areas.
- **Key Characteristics**:
  - Simpler to set up (no IMU required).
  - Performance degrades without good feature texture or when moving too fast, leading to drift in long sequences.

Mono + Inertial Mode

- **Features Required**:
  - Combines **monocular vision** with **IMU data** for increased robustness.
  - IMU compensates for lack of depth information from a single camera and helps track high-speed motion or quick rotations.
- **Performance**:
  - This mode works significantly better than the standard **Mono** configuration, especially for **fast movements** or **quick rotations**. IMU data helps minimize visual drift and improves **real-time tracking**.
- **Key Characteristics**:
  - IMU aids in short-term pose tracking.
  - Provides **stabilized tracking** in challenging environments where visual-only SLAM may fail.

Key Inferences and Conclusion on ORB-SLAM3

Stereo Mode

- **Features Required**:
  - Uses **stereo cameras**, which provide direct depth estimation.
  - **Keypoints** are extracted from both left and right camera views, and **triangulation** is used to calculate the depth of these features.
- **Performance**:
  - The **Stereo** mode is more accurate than **Mono** because depth is computed directly from stereo pairs, reducing **drift** and providing better **feature matching** over long sequences.
  - However, stereo setups can fail in **low-texture** environments (like empty or highly repetitive rooms).
- **Key Characteristics**:
  - Highly accurate in well-textured environments with good feature points.
  - **Stereo cameras** handle motion well, but **fast rotations** or **high-speed movements** might still lead to drift over time.

Stereo + Inertial Mode

- **Features Required**:
  - Combines **stereo cameras** for depth and **IMU** data for motion compensation.
  - **IMU** enhances feature matching and helps track the camera pose during rapid motion or when visual features are scarce.
- **Performance**:
  - **Stereo + Inertial** is the most **robust** configuration, especially for **dynamic environments** or **high-speed motion**.
  - IMU helps to recover from short-term tracking failures due to motion blur or fast rotations, while stereo gives accurate depth information for the scene geometry.
- **Key Characteristics**:
  - **Highly accurate and robust** for long sequences.
  - Less drift compared to other configurations, as **IMU** compensates for lack of visual information in fast movements, improving **real-time pose estimation**.
  - Performs the best in **challenging conditions** (rapid motion, low-texture areas, and rotations).

References

- https://github.com/UZ-SLAMLab/ORB_SLAM3
- https://arxiv.org/pdf/2007.11898

ORB-SLAM3

V1.0, December 22th, 2021

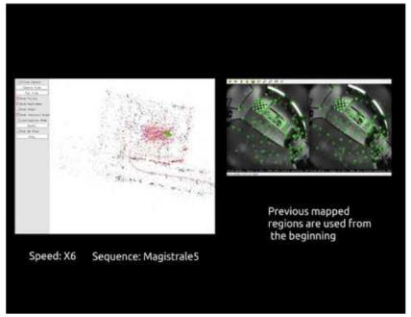
Authors: Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, Juan D. Tardos.

The Changelog describes the features of each version.

ORB-SLAM3 is the first real-time SLAM library able to perform **Visual, Visual-Inertial and Multi-Map SLAM** with **monocular, stereo and RGB-D** cameras, using **pin-hole and fisheye** lens models. In all sensor configurations, ORB-SLAM3 is as robust as the best systems available in the literature, and significantly more accurate.

We provide examples to run ORB-SLAM3 in the EuRoC dataset using stereo or monocular, with or without IMU, and in the TUM-VI dataset using fisheye stereo or monocular, with or without IMU. Videos of some example executions can be found at ORB-SLAM3 channel.

This software is based on ORB-SLAM2 developed by Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez (DBoW2).



ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

Carlos Campos*, Richard Elvira*, Juan J. Gómez Rodríguez, José M.M. Montiel and Juan D. Tardós

***Abstract*—This paper presents ORB-SLAM3, the first system able to perform visual, visual-inertial and multi-map SLAM with monocular, stereo and RGB-D cameras, using pin-hole and fisheye lens models.**

The first main novelty is a feature-based tightly-integrated visual-inertial SLAM system that fully relies on Maximum-a-Posteriori (MAP) estimation, even during the IMU initialization phase. The result is a system that operates robustly in real time, in small and large, indoor and outdoor environments, and is two to ten times more accurate than previous approaches.

The second main novelty is a multiple map system that relies on a new place recognition method with improved recall. Thanks to it, ORB-SLAM3 is able to survive to long periods of poor visual information: when it gets lost, it starts a new map that will be seamlessly merged with previous maps when revisiting mapped areas. Compared with visual odometry systems that only use information from the last few seconds, ORB-SLAM3 is the first system able to reuse in all the algorithm stages all previous information. This allows to include in bundle adjustment co-visible keyframes, that provide high parallax observations boosting accuracy, even if they are widely separated in time or if they come from a previous mapping session.

Our experiments show that, in all sensor configurations, ORB-SLAM3 is as robust as the best systems available in the literature, and significantly more accurate. Notably, our stereo-inertial SLAM achieves an average accuracy of 3.5 cm in the EuRoC drone and 9 mm under quick hand-held motions in the room of TUM-VI dataset, a setting representative of AR/VR scenarios. For the benefit of the community we make public the source code.

I. Introduction

Intense research on Visual Simultaneous Localization and Mapping systems (SLAM) and Visual Odometry (VO), using cameras either alone or in combination with inertial sensors, has produced during the last two decades excellent systems, with increasing accuracy and robustness. Modern systems rely on Maximum a Posteriori (MAP) estimation, which in the case of visual sensors corresponds to Bundle Adjustment (BA), either geometric BA that minimizes feature reprojection error, in feature-based methods, or photometric BA that minimizes the photometric error of a set of selected pixels, in direct methods.

With the recent emergence of VO systems that integrate loop closing techniques, the frontier between VO and SLAM is more diffuse. The goal of Visual SLAM is to use the sensors on-board a mobile agent to build a map of the environment and compute in real-time the pose of the agent in that map. In contrast, VO systems put their focus on computing the agent's ego-motion, not on building a map. The big advantage of a SLAM map is that it allows matching and using in BA previous observations performing three types of data association (extending the terminology used in [1]):

- **Short-term data association**, matching map elements obtained during the last few seconds. This is the only data association type used by most VO systems, that forget environment elements once they get out of view, resulting in continuous estimation drift even when the system moves in the same area.
- **Mid-term data association**, matching map elements that are close to the camera whose accumulated drift is still small. These can be matched and used in BA in the same way than short-term observations and allow to reach zero drift when the systems moves in mapped areas. They are the key of the better accuracy obtained by our system compared against VO systems with loop detection.
- **Long-term data association**, matching observations with elements in previously visited areas using a place recognition technique, regardless of the accumulated drift (loop detection), the current area being previously mapped in a disconnected map (map merging), or the tracking being lost (relocalization). Long-term matching allows to reset the drift and to correct the map using pose-graph (PG) optimization, or more accurately, using BA. This is the key of SLAM accuracy in medium and large loopy environments.

In this work we build on ORB-SLAM [2], [3] and ORB-SLAM Visual-Inertial [4], the first visual and visual-inertial systems able to take full profit of short-term, mid-term and long-term data association, reaching zero drift in mapped areas. Here we go one step further providing **multi-map data association**, which allows us to match and use in BA map elements coming from previous mapping sessions, achieving the true goal of a SLAM system: building a map that can be used later to provide accurate localization.

This is essentially a system paper, whose most important contribution is the ORB-SLAM3 library itself [5], the most complete and accurate visual, visual-inertial and multi-map SLAM system to date (see table I). The main novelties of ORB-SLAM3 are:

- **A monocular and stereo visual-inertial SLAM system** that fully relies on Maximum-a-Posteriori (MAP) estimation, even during the IMU (Inertial Measurement Unit)

* Both authors contributed equally to this work.
The authors are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain. E-mail: {campos, richard, jjgomez, josemari, tardos}@unizar.es.
This work was supported in part by the Spanish government under grants PGC2018-096367-B-I00 and DPI2017-91104-EXP, and by Aragón government under grant DGA_T45-17R.