Optimizing Monocular SLAM with Keyframe-Based Smoothing

AV-SLAM System using KITTI Dataset

Harshil Sharma, Harsh Patel, Dhyey Patel
Department of Computer Science
Saint Louis University, St. Louis, MO, USA

***Abstract*—We present a modular AV-SLAM system designed to extract, optimize, and visualize vehicle trajectories using monocular image sequences from the KITTI dataset. By leveraging ORB feature-based keyframe selection and Savitzky–Golay filtering, we enhance the accuracy and smoothness of 3D trajectory estimation. Our pipeline was validated on the `2001_09_26_drive_0051_sync` dataset using a custom Python implementation.**

***Index Terms*—SLAM, KITTI Dataset, ORB Features, Trajectory Optimization, Keyframe Selection, AV-SLAM**

I. Introduction

This project aims to design and deploy an effective and scalable AV-SLAM system that can precisely estimate the trajectory of a moving vehicle from monocular image sequences. Our proposed pipeline emphasizes keyframe selection and trajectory refinement to obtain a smooth and stable path.

We validated our approach using the KITTI dataset sequence `2001_09_26_drive_0051_sync`. The high-level methodology includes:

- Estimating 6-DoF camera poses from monocular input through motion estimation.
- Selecting keyframes based on ORB visual feature matching thresholds.
- Compositing the entire trajectory using pose chaining and applying Savitzky–Golay smoothing for final refinement.

II. System Design

A. Architecture Overview

Our architecture includes four major modules forming a linear processing pipeline:

- **Pose Extractor:** Parses raw transformation matrices from pose logs.
- **Trajectory Builder:** Chains relative poses to compute a global path.
- **Keyframe Selector:** Uses ORB feature descriptors to pick visually distinct frames.
- **Trajectory Smoother:** Applies filtering to produce a clean trajectory.

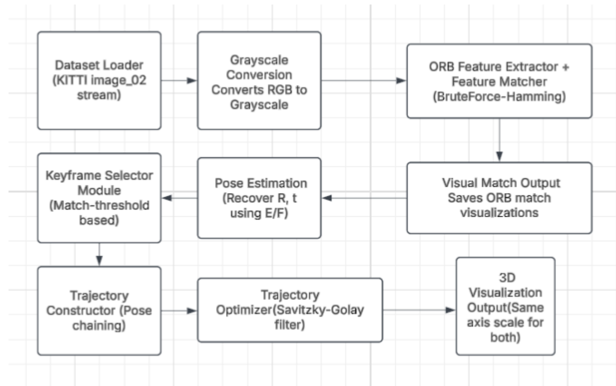

Fig. 1. System architecture from input initialization to optimized trajectory output.

B. ORB Keyframe Matching

ORB keyframe matching is performed using OpenCV's ORB detector. The brute-force Hamming distance is used to filter feature matches. Frames with fewer than 40 matches are skipped. Figure 2 shows a keyframe match between Frame 232 and 234.

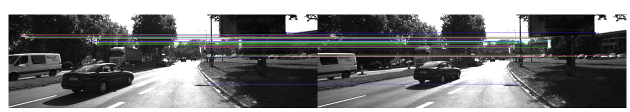

Fig. 2. Keyframe match between Frame 232 and 234 using ORB features.

C. Base Trajectory

The base trajectory is constructed by chaining transformation matrices from the pose logs. It often contains jitter and noise due to monocular estimation. An example is shown in Figure 3.

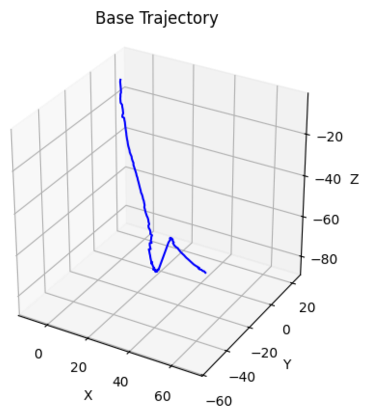

Fig. 3. Base trajectory from raw pose composition. Axes scaled uniformly.

D. Optimized Trajectory

To improve smoothness and accuracy, we select only keyframes and pass them through a Savitzky–Golay filter. This reduces noise and sudden deviations, yielding a smoother path (Figure 4).

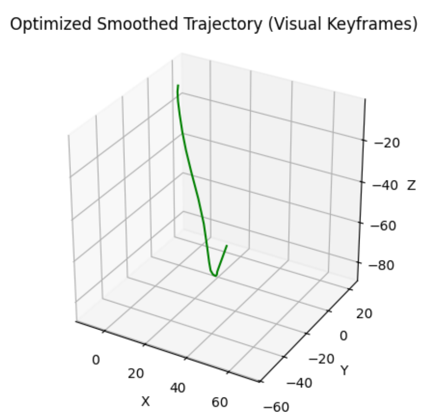

Fig. 4. Optimized trajectory after keyframe filtering and smoothing. Axes scaled uniformly.

III. Results

The optimized path better preserves the actual vehicle route with reduced noise. The smoothed trajectory improves visual quality and stability, especially along the Z-axis, when compared to the raw base trajectory.

Our modular approach also ensures computational efficiency by reducing the number of frames processed. Visual comparisons validate the effectiveness of both keyframe selection and smoothing.

IV. Conclusion

We deployed a visually optimized SLAM pipeline leveraging keyframe-based smoothing. The resulting system is robust, modular, and extensible. Future work could explore enhancements such as loop closure detection, depth integration, or bundle adjustment.

Acknowledgments

We used ChatGPT to assist with debugging, trajectory optimization, and documentation throughout the project. The full source code is available at: **AV-SLAM-SYSTEM REPO:** https://github.com/xxender13/AV-SLAM-SYSTEM

References

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, 2017.

[2] A. Geiger et al., "Vision meets Robotics: The KITTI Dataset," *IJRR*, 2013.

[3] OpenCV ORB Feature Matching: https://docs.opencv.org/