Mini-project: A Visual Odometry Pipeline!

Jérôme Bonvin 20-917-332
Nicola Irmiger 20-926-127
Liam Achenbach 20-940-268

Abstract

We implemented a Visual Odometry (VO) pipeline comprising two main stages: initialization and continuous operation. The initialization stage establishes a foundational 3D reconstruction using feature detection, matching, and triangulation, while the continuous operation phase focuses on frame-to-frame motion estimation and map updating. Our approach was tested on diverse datasets [1], including Parking, KITTI 05, and Malaga, to evaluate its performance in terms of trajectory estimation accuracy, robustness to scale drift, and computational efficiency. The results demonstrate the pipeline's strengths in detecting reliable features and maintaining consistency across frames, alongside challenges in addressing cumulative errors and scale consistency for extended sequences. [2]

1. Introduction

Visual Odometry (VO) is a critical component in robotics and computer vision, enabling accurate estimation of a camera's trajectory based solely on visual inputs. Its applications span autonomous driving, drone navigation, and augmented reality. Despite significant advancements, VO systems face challenges, such as handling scale ambiguity in monocular setups and mitigating cumulative errors over extended trajectories. These limitations highlight the need for robust methodologies capable of delivering accurate pose estimation in diverse and dynamic environments.

In this project, we implemented a two-stage VO pipeline designed to address these challenges. The initialization phase focuses on establishing a robust 3D reconstruction using feature detection, matching, and triangulation. The continuous operation phase extends this foundation by estimating frame-to-frame motion and updating the 3D map. To evaluate our pipeline's performance, we tested it on three benchmark datasets—Parking, KITTI 05, and Malaga—each presenting unique challenges in terms of trajectory complexity and environmental variability.

Our contributions are twofold: first, we provide a detailed implementation of a monocular VO pipeline, incorporating state-of-the-art techniques for feature detection and matching. Second, we analyze the strengths and limitations of our approach through quantitative and qualitative evaluations, offering insights into the trade-offs involved in VO system design.

2. Method

As suggested in the report, we adopted a two-stage pipeline consisting of the initialization phase and the continuous operation phase. This section outlines the methodology and implementation of these phases. In our pipeline, we replaced implementations from earlier exercises with optimized OpenCV counterparts, improving performance and computational efficiency.

2.1. Initialization

To initialize the Visual Odometry (VO) pipeline, it is necessary to establish a 3D reconstruction from the features detected in two initial frames. This involves feature detection, matching, and triangulation, as detailed below.

Feature Detection

Feature detection is a critical step to identify distinct points in the images for reliable matching. For this purpose, we employed the Scale Invariant Feature Transform (SIFT) [3], which offers robustness to scale and rotation variations. SIFT was chosen due to its superior performance in challenging scenarios compared to other detectors, such as the Harris Corner Detector and Shi-Tomasi Corner Detector, which were initially considered but ultimately not utilized.

Feature Matching

Once features are detected, the next step is to match them across the two initial frames. This was accomplished by comparing feature descriptors using the Sum of Squared Differences (SSD). We utilize the Brute Force matching implemented in OpenCV with K-nearest-neighbor matching, as we have also seen it during the exercises. To ensure robust matching and reduce ambiguity, we applied the distance ratio test, as proposed by Lowe [3]. This test ensures that a keypoint is confidently matched by enforcing that the

ratio of the closest match to the second-closest match remains below a predefined threshold.

Triangulation

Matched features from the two frames were then used for triangulation to estimate their 3D positions. This process leverages the camera intrinsics and the relative pose between the frames, obtained using the essential matrix decomposition. The resulting 3D points serve as the foundation for initializing the VO pipeline and are stored as landmarks in the state.

2.2. Continuous Operation

2.2.1 General structure

We use the structure recommended in the task statement by keeping track of keypoints, landmarks and candidate keypoints as the state of the pipeline. We implement this with a dict structure in Python for fast memory access.

2.2.2 Feature Detection

We initially choose SIFT for the Structure-from-motion (SFM) initialization because it offers invariance to various transforms as seen in the lecture. This allows us to initialize robustly in various datasets with a large baseline. We recognize that a corner detector such as Shi-Tomasi or Harris would probably increase the feature detection speed a lot during continuous operation, but during implementation of the full VO pipeline we observed some stability issues with them and ultimately opted for the slower but seemingly more stable SIFT detectors. With our current implementation one could switch the detection to Shi-Tomasi or Harris in one line of code, but we believe we didn't find the right parameters to get these running reliably.

2.2.3 Feature tracking

We employ the Kanade-Lucas-Tomasi (KLT) tracker to keep track of the keypoints from image I_k to I_{k+1}. We choose this method because it is direct and less computation than feature matching as also instructed in the task description. If a feature can not be tracked through the keyframes it is retired from the candidate points. Alternatively, tracking the feature through the frames is the precondition to adding new landmarks.

2.2.4 Motion Estimation

For motion estimation, we used a PnP algorithm with RANSAC from OpenCV to estimate the camera pose from 3D-2D correspondences. Given a set of valid 3D landmarks and their corresponding 2D projections in the current image, the algorithm estimates the rotation and translation of the camera while rejecting outliers through iterative sampling. The resulting inliers indicate consistent correspondences, ensuring robust pose estimation even in the presence of noise and mismatches. Initially, we experimented with a basic P3P solver but observed stability and accuracy issues on more challenging datasets. Additionally, we ran into issues after some intial time with the EPnP solver of cv2. Switching to the iterative solver (`cv2.SOLVEPNP_ITERATIVE`) significantly improved performance, offering more reliable pose estimates across varying conditions and complex scenes.

2.2.5 Adding new Landmarks

In the process of adding new landmarks, we identify candidate keypoints from the current and previous frames and attempt to triangulate their 3D positions. For each successfully tracked keypoint, we calculate the angle between the viewing rays corresponding to the first detection of the keypoint and its current observation. This angle must exceed a predefined threshold to ensure sufficient baseline disparity and reliable triangulation (CONT_VO - baseline angle threshold in 2). This helps mitigate numerical instability and poor triangulation quality. Keypoints that meet this criterion are then used to triangulate new 3D landmarks using the relative poses of the two frames and the camera intrinsic matrix. The triangulation itself is performed using OpenCV's triangulation function.

3. Results

To evaluate the performance of our Visual Odometry (VO) pipeline, we tested it on multiple datasets to analyze its strengths and weaknesses. Specifically, we used the Parking, KITTI 05, and Malaga datasets. The resulting camera poses were extracted and visualized in Figures 1, 2, and 3. For the Parking dataset, ground truth trajectory data was available, enabling us to calculate the drift introduced by our model. Since our pipeline uses monocular VO, it cannot estimate absolute scale but only relative changes. Therefore, we scaled the ground truth trajectory to match the length of the x-axis for comparison.

For the KITTI and Malaga datasets, the trajectories presented more complexity due to the presence of turns and extended sequences. These scenarios amplified the impact of small deviations in the local frame on the global trajectory. This effect is particularly evident in the first turn shown in Figure 3. A slight deviation to the left before the turn caused the entire trajectory to shift at an angle, demonstrating the compounding nature of errors in VO systems. Such behavior highlights the challenge of achieving robust performance in diverse environments with monocular VO.

Additionally, for both the KITTI and Malaga datasets,

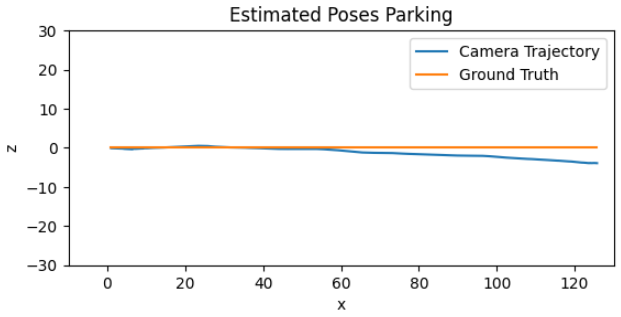

Figure 1. **Parking Pose Estimation**. The trajectory demonstrates the pipeline's ability to closely follow the ground truth with minimal drift.

we observed that after some time, the scale of the camera poses would collapse and become very small, giving the impression that the camera remained stationary. This behavior indicates a limitation in our pipeline that we were unable to resolve. We investigated several reasons on why we are getting stuck in the Malaga and the KITTI datasets. Intially we found a correlation between the number of keypoints being promoted to landmarks after encountering a feature dense object after some feature poor frames. This seemed to heavily bias the landmark spread to feature dense areas such as trees in the Malaga dataset. Noticing this behavior we were able to prevent it by limiting the feature promotion and the addition of new candidate points. This improved our pipeline but after some time we still tend to get stuck in certain scenarios. Unfortunately, we were not able to tune our parameters enough to prevent this from happening in the remaining time. Additionally, the inability to maintain a consistent scale over extended sequences is a known challenge in monocular VO, further emphasizing the difficulty of accurate trajectory estimation in these scenarios.

In addition to trajectory estimation, we evaluated the pipeline's feature detection capabilities by analyzing the distribution of keypoints before and after outlier filtering using RANSAC (Figure 4). The results indicate that the pipeline consistently detects a sufficient number of keypoints per frame. Moreover, the similarity in the shapes of the histograms before and after filtering suggests that the initial keypoints detected using SIFT are of high quality and largely reliable. This highlights the robustness of the feature detection component in our pipeline.

The computational efficiency of the pipeline is another critical metric, particularly for real-time applications. To assess this, we plotted the distribution of SIFT computation times across frames (Figure 5). The histogram reveals a consistent performance of the SIFT algorithm, with the majority of frames processed within a narrow time range. This consistency is essential for maintaining steady performance in time-sensitive applications.

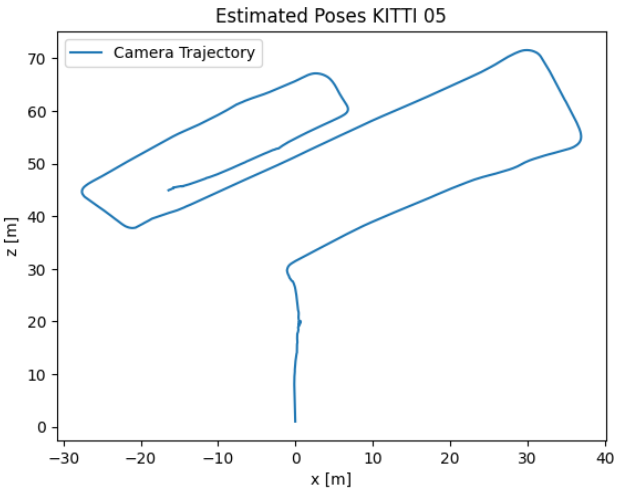

Figure 2. **KITTI 05 Pose Estimation**. The trajectory visualization highlights the pipeline's handling of complex paths with multiple turns.

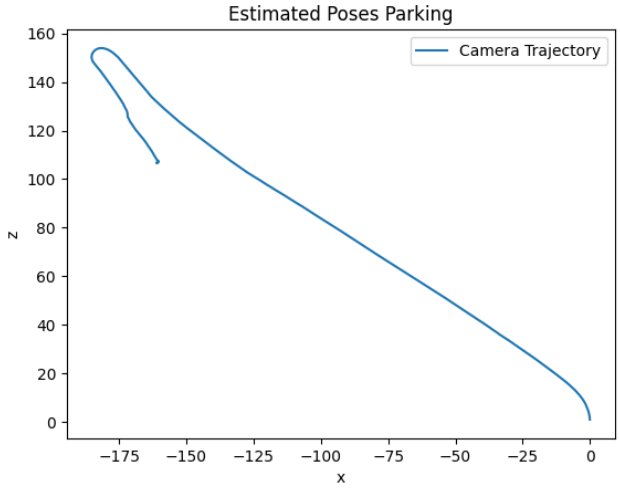

Figure 3. **Malaga Pose Estimation**. The trajectory showcases the effect of cumulative errors over extended sequences, particularly in turning scenarios.

Finally, we explored the relationship between SIFT computation time and the number of detected keypoints. As shown in Figure 6, no clear correlation was observed between these variables. This suggests that other factors, such as hardware performance or implementation details, may influence computation time independently of the number of keypoints detected.

4. Conclusion

The implementation of our Visual Odometry (VO) pipeline demonstrates the feasibility and challenges of monocular pose estimation in diverse environments. Our results on the Parking, KITTI 05, and Malaga datasets

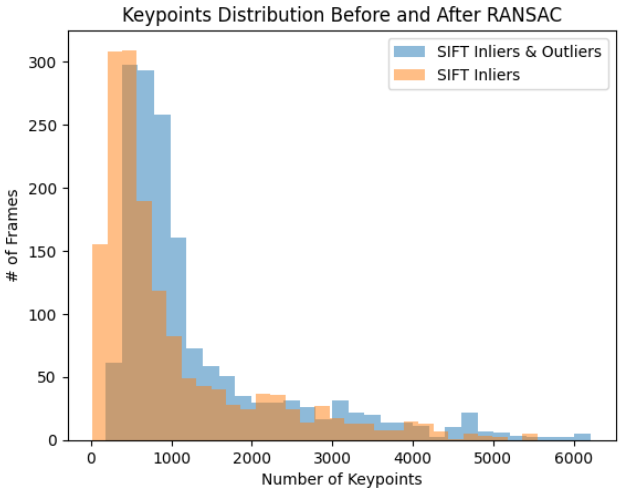

Figure 4. **Inliers vs. Outliers**. Comparison of detected keypoints before and after RANSAC filtering. The similarity in histogram shapes supports the reliability of the initial feature detection process.

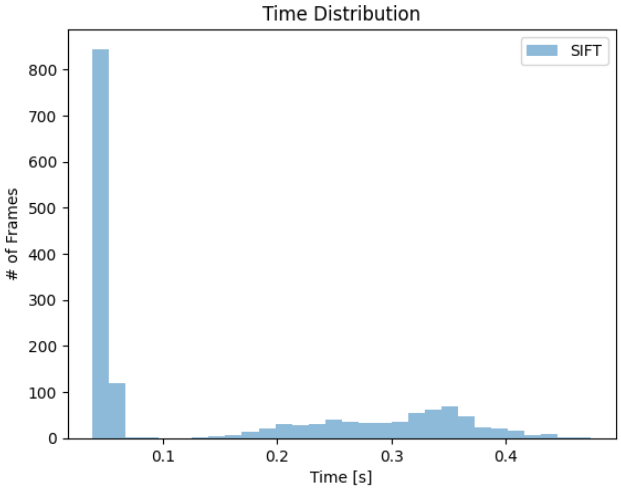

Figure 5. **Histogram of SIFT Computation Times**. The distribution illustrates consistent processing times, crucial for real-time VO applications.

illustrate the pipeline's ability to detect reliable features, maintain computational efficiency, and approximate trajectories with reasonable accuracy. However, the analysis also reveals limitations, particularly in maintaining consistent scale over extended sequences and mitigating the compounding effects of small local errors.

Key strengths of our pipeline include the robustness of SIFT-based feature detection and matching, as evidenced by consistent performance across datasets, and the computational efficiency achieved during real-time processing. Conversely, the inability to prevent scale collapse and trajectory deviations in complex scenarios underscores areas for future improvement. Integrating additional modules, such as

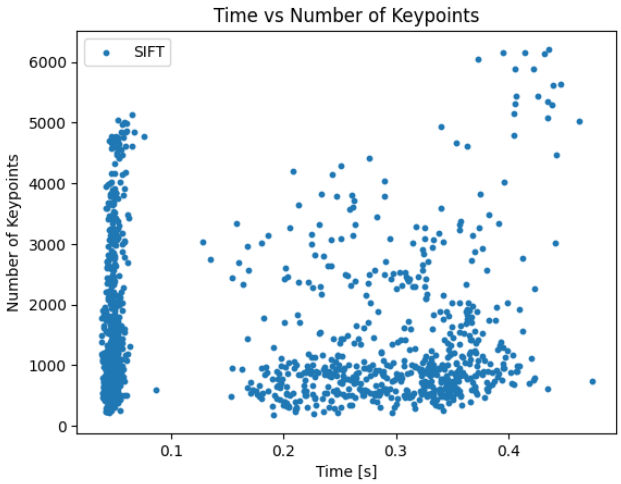

Figure 6. **SIFT Computation Time vs. Number of Keypoints**. Scatter plot illustrating the lack of correlation, suggesting that computation time is influenced by factors beyond keypoint count.

loop closure detection or multi-sensor fusion, could help address these limitations and enhance the overall robustness and accuracy of VO systems.

In summary, our project provides a solid foundation for further exploration and optimization of Visual Odometry techniques, with potential applications in autonomous systems and beyond.

5. Encountered Problems

Throughout the development and continuous operation of our visual odometry pipeline, we encountered several challenges that impacted its performance and accuracy. In this section, we highlight the most significant problems that were encountered during the implementation and explain the solutions we devised to mitigate or resolve them. Addressing these issues was crucial for ensuring the robustness and reliability of the visual odometry system in diverse environments.

During the continuous operation of the visual odometry pipeline, we observed an issue where triangulated points occasionally resulted in the creation of landmarks located behind the camera. This problem arose as the system computed 3D coordinates for new points based on stereo images or frame-to-frame correspondences. However, certain mismatches or inaccuracies in feature tracking led to erroneous triangulation results, causing landmarks to be placed in physically implausible positions, specifically behind the camera.

The presence of such invalid landmarks introduced significant challenges to the pipeline's accuracy and stability. Since the system relied on these landmarks to estimate both the position and orientation of the camera, having incorrect

3D points led to a drift in the camera's trajectory and a distorted understanding of its surroundings. This caused the visual odometry algorithm to become "confused" about the true location of the camera, impacting both localization and mapping accuracy.

To address this issue, we applied a physical constraint based on the assumption that the camera cannot observe points behind it, rejecting any triangulated landmarks with negative depth. This ensured that only points in front of the camera were considered valid, helping to eliminate the erroneous landmarks behind the camera and restoring the pipeline's stability. This solution proved effective in reducing this specific type of error.

Another challenge arose when large, textured objects appeared in the camera's view. In such cases, the system would detect and track a high number of keypoints, especially when the object contained rich visual features. As the system processed the images and reached the angle threshold for triangulation, it would attempt to add a significant number of new landmarks—sometimes almost 1000 in a single frame.

This sudden influx of landmarks did not cause computational bottlenecks, but rather disrupted the pose estimation process. A large number of features were concentrated in the same region of the image, leading to poor localization accuracy. Many of these features were prone to mismatching, which likely resulted in a high number of incorrect correspondences. As a consequence, the system's pose estimation became unreliable, causing instability in the pipeline. This issue was particularly pronounced in environments with dense, textured objects, such as trees, where the number of keypoints tracked by the system could be excessively high.

To alleviate this problem, we introduced an upper bound on the number of landmarks that could be added per frame (CONT_VO - max candidate points in 2). This limit prevented the system from becoming overloaded by a large number of new points, ensuring smoother operation and more stable performance during periods of high feature density.

6. Author Contributions

Task	Assigned Personnel
Initialization	Jerome, Nicola, Liam
Continuous Operation	
Matching keypoints to landmarks	Nicola, Liam
Estimating the current pose	Nicola, Liam
Triangulating new landmarks	Nicola, Liam
Visualization	Jerome
Report	Jerome, Nicola, Liam

Table 1. Task Assignment

References

[1] Youtube SwissJAB. Vo miniproject. https://www.youtube.com/playlist?list=PLYPB14LfGrkPhpV88stzWYthXyCQFvqRH, 2025. Accessed: January 5, 2025. 1

[2] Github SwissJAB. Vo_miniproject. https://github.com/SwissJAB/VO_miniproject, 2025. Accessed: January 2, 2025. 1

[3] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1

Appendix

Parameter	Parking	Malaga	KITTI
Feature Detector	SIFT	SIFT	SIFT
SIFT - number of features	no limit	no limit	no limit
SIFT - contrast threshold	0.04	0.04	0.04
SIFT - sigma	1.6	1.6	1.6
SIFT - number of octave layers	3	3	3
MATCHING - k	2	2	2
MATCHING - ratio	0.75	0.75	0.75
Luxas Kanade - window size	21	21	21
Lucas Kanade - max level	3	3	3
Lucas Kanade - critical count	30	30	30
Lucas Kanade - critical eps	0.01	0.01	0.01
RANSAC - probability	0.999	0.999	0.999
RANSAC - threshold	2.0	2.0	2.0
PNPRANSAC - iterations	1000	1000	1000
PNPRANSAC - max reprojection error	4.0	4.0	4.0
PNPRANSAC - probability	0.99	0.99	0.99
CONT_VO - baseline angle threshold	3	2	3
CONT_VO - kp dist threshold	4.0	12.0	4.0
CONT_VO - max candidate points	150	150	150

Table 2. Configuration Comparison for Parking, Malaga, and KITTI Datasets