Raceline Extraction from Driver Camera Racing Videos

Niketh Bayya Mahesh

May 2025

1 Introduction

Our project aims to track the path a vehicle takes around a track using driver-mounted camera videos. The goal is to accurately extract the vehicle's trajectory, representing its movement along the track. This task combines techniques from robotics, machine learning, and computer vision. The project is highly relevant to robotics, especially in the context of autonomous vehicles and racing, as it leverages deep learning and computer vision for tracking and path estimation

2 Technical Details

Our solution involves using DeepVO to analyze sequential frames from driver-camera racing videos and reconstruct the vehicle's trajectory. DeepVO em-ploys a convolutional neural network (CNN) and a recurrent neural network (RNN) to effectively model spatial and temporal information. The model outputs relative pose changes between frames, which are aggregated to con-struct the global trajectory.

2.1 DeepVO Implementation

DeepVO is a state-of-the-art framework for visual odometry that leverages deep learning to estimate vehicle motion directly from video data. It uses a combination of convolutional and recurrent layers to process spatial and temporal information. The pipeline involves:

1. **Feature Extraction with CNNs**: The convolutional neural network (CNN) in DeepVO is responsible for extracting spatial features from individual video frames. It processes raw pixel data to create high-level feature maps, which capture essential details about the track and surroundings.

2. **Sequence Modeling with RNNs**: The extracted spatial features are then passed to a recurrent neural network (RNN), specifically an LSTM (Long Short-Term Memory) network, which captures temporal dependencies between consecutive frames. This enables the model to understand motion dynamics and estimate relative pose transformations effectively.

3. **Relative Pose Estimation**: The RNN outputs relative pose transformations, including both translation and rotation between frames. These pose changes are aggregated to reconstruct the global trajectory of the vehicle.

4. **End-to-End Training**: The model is trained end-to-end using a supervised learning approach, with ground truth pose data serving as labels. Loss functions such as mean squared error (MSE) are used to minimize the difference between predicted and true poses.

2.2 Recurrent Neural Networks (RNNs)

The RNN in DeepVO, specifically the LSTM module, plays a critical role in modeling temporal dependencies. Key functionalities include:

- **Sequential Dependency Modeling**: The LSTM captures relationships between consecutive frames, such as speed and direction changes, which are crucial for accurate trajectory estimation.

- **Handling Long Sequences**: Unlike traditional RNNs, LSTMs are designed to handle long sequences without suffering from vanishing gradients. This makes them suitable for processing extended video footage. We use training with sequence of 4 images, mainly limited due to memory limitations. It can work much better when trained on longer sequences.

- **Temporal Smoothing**: The RNN ensures temporal consistency in pose estimates by learning smooth transitions between frames, reducing noise and abrupt changes in the trajectory.

2.3 Convolutional Neural Networks (CNNs)

CNNs form the backbone of the DeepVO framework by extracting high-level features from video frames. Their role includes:

- **Spatial Feature Extraction**: The CNN processes raw image data to generate feature maps that highlight important elements such as edges, textures, and patterns in the track environment.
- **Robustness to Variations**: By learning hierarchical feature representations, CNNs can handle variations in lighting, camera angles, and environmental conditions, ensuring consistent performance across different datasets.
- **Integration with Optical Flow**: Features extracted by the CNN are augmented with motion cues from FlowNet, providing a comprehensive representation of spatial and motion information.

2.4 FlowNet

FlowNet is an essential component of the DeepVO framework, used for optical flow estimation. Optical flow refers to the apparent motion of objects between consecutive frames in a video, which is critical for understanding relative motion. Key details include:

- **Feature Integration**: The optical flow data generated by FlowNet is integrated into the CNN to enhance spatial feature extraction, making the model more robust to variations in lighting, texture, and motion blur.
- **Motion Cues**: FlowNet provides essential motion cues that help the RNN understand the dynamics of the vehicle's movement, improving overall pose estimation accuracy.

2.5 Preprocessing Images

Preprocessing is a critical step in ensuring the input data is suitable for DeepVO. The preprocessing pipeline includes:

1. **Frame Extraction**: Video frames are extracted from input racing videos at a consistent frame rate. This ensures temporal uniformity, which is essential for accurate pose estimation.

2. **Normalization and Resizing**: Frames are normalized to adjust pixel intensity values and resized to match the input dimensions required by the DeepVO model. This step ensures compatibility with the CNN architecture.

3. **Mean Subtraction**: Images are optionally preprocessed by subtracting mean RGB values to center the pixel distributions. While this technique can help in some scenarios, it did not significantly improve performance in our case, so its use is optional.

2.6 Poses Data Handling

The transformation data in DeepVO is initially provided in a 12 Degrees of Freedom (DoF) format, which includes both translation and rotation components. However, for the purpose of pose estimation, we convert this data into a 6 Degrees of Freedom (DoF) representation. This conversion simplifies the transformation data while preserving essential information for vehicle motion estimation. The 12 DoF data consists of:

- **Translation (3 DoF)**: The 3 translation values correspond to the vehicle's movement along the X, Y, and Z axes in 3D space.

- **Rotation (9 DoF)**: The 9 rotational values represent the vehicle's orientation using a combination of Euler angles (pitch, yaw, and roll) or a rotation matrix. However, in DeepVO, we simplify these to a 6 DoF representation.

To convert the 12 DoF data into 6 DoF, we focus on the most relevant components:

- **Translation (3 values)**: These values correspond to the vehicle's movement along the X, Y, and Z axes, representing its position in the 3D environment.

- **Rotation (3 values)**: We use a simplified representation of rotation, typically using a 3D rotation matrix or a quaternion, to capture the yaw, pitch, and roll of the vehicle. This reduced rotation representation is sufficient for DeepVO's pose estimation needs, which focus on trajectory reconstruction rather than full 3D orientation.

The conversion from 12 DoF to 6 DoF reduces computational complexity while retaining the critical pose information needed to estimate the vehicle's

motion and trajectory. This transformation ensures that the DeepVO model can operate efficiently on the pose data while maintaining high accuracy in tracking the vehicle's movement over time.

3 Results and Discussion

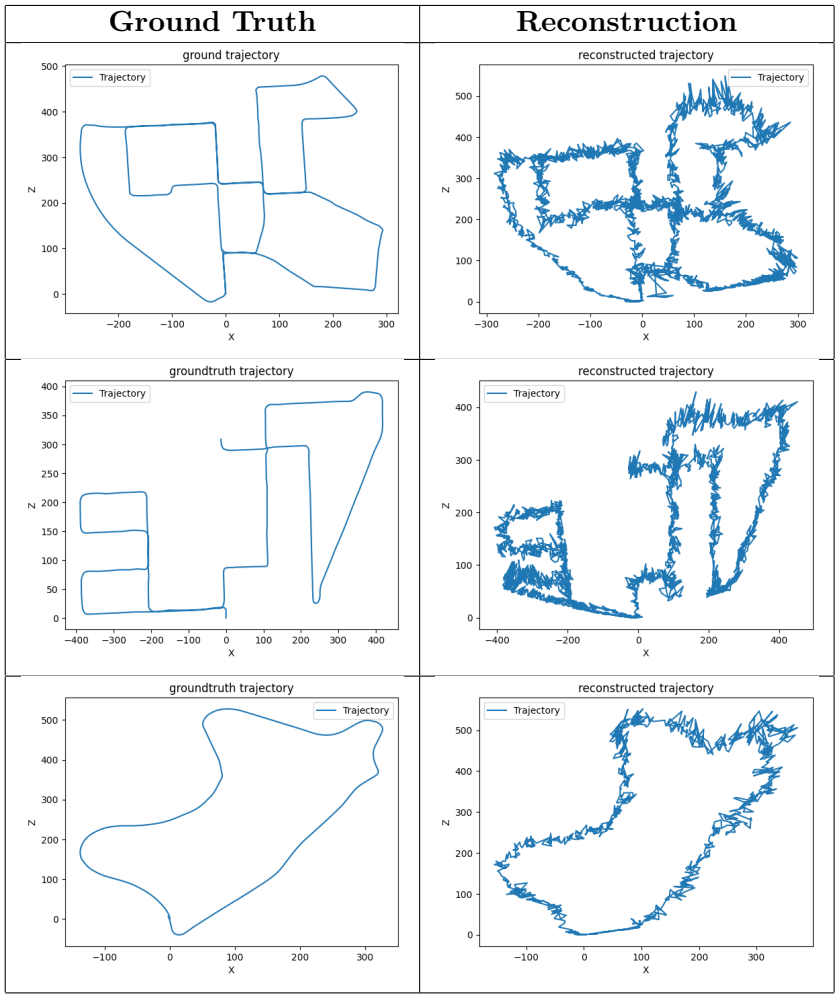

Table 1: Ground truth and corresponding reconstructions for the first sequence.

The results achieved in this project are not ideal. While the reconstructed trajectory captures the general path of the vehicle, the generated

raceline deviates significantly from the ground truth in certain sections of the track. This indicates that the model requires further fine-tuning to improve its accuracy.

3.1 Current Limitations

One primary limitation of this project was computational power. Resource constraints prevented full training of the model as described in the original DeepVO paper, restricting its ability to generalize across different video scenarios and leading to less accurate trajectory estimations and raceline outputs. Specifically, due to RAM memory insufficiency, we could only train the model with 4 sequences at a time, running for 200 epochs. This setup was executed on a Jetstream instance with the following configuration:

32 CPU cores 117 GB RAM 990 GB root disk

3.2 Future Improvements

Fully training the model as per the original methodology would likely result in more accurate trajectory predictions and a closer match to the ground truth raceline.

Incorporating depth perception into the model could enhance its understanding of the track environment, improving performance in complex scenarios, such as sharp turns or elevation changes. The model could be trained on a larger set of video/image sequences to improve generalization, as the current approach was limited by memory constraints. This would allow for better robustness and accuracy across different scenarios.

Implementing smoothing techniques could help reduce the coarse nature of the reconstructed trajectory, providing more precise and smoother raceline outputs. The addition of inertial data (e.g., accelerometer and gyroscope readings) could significantly enhance the model's ability to estimate motion and trajectory in complex environments.

With access to more memory, we could explore multimodal approaches by training the model with multiple stereo camera inputs and corresponding inertial data. This would provide a richer representation of the environment and further improve trajectory accuracy.

3.3 Conclusion

This project provided valuable insights into the application of DeepVO for raceline extraction. It emphasized the importance of high-quality data and effective preprocessing techniques in ensuring robust model performance.

Additionally, it highlighted the critical role of computational resources in fully training deep learning models to unlock their potential. The project also demonstrated that incorporating additional features, such as depth perception, significantly improves the model's accuracy and its applicability to real-world scenarios. One of the key advantages of using DeepVO is that it eliminates the need to create complex logic for physics-based simulations, which is often required in traditional methods, making the approach easier to implement. However, this method also has its limitations, such as sensitivity to video quality, lighting variations, and the need for substantial computational resources. Overall, the project underscored both the potential and the challenges of using visual odometry techniques, paving the way for more accurate and efficient raceline extraction methods in the future.

References

[1] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. pages 7286–7291, 2018.

[2] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks, September 2017. Accessed: 25 Sep 2017.

[3] Zhigang Wu and Yaohui Zhu. Swformer-vo: A monocular visual odometry model based on swin transformer. *IEEE Robotics and Automation Letters*, 9(5):4766–4773, 2024.

[4] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry, June 2020.