Deep Learning-based Camera Motion Smoothing Using BiLSTM, Transformer, and GRU Architectures

Team Members:
tarungan
bimlendr
kuwarpre

April 30, 2025

Contents

Abstract

Video stabilization is a primitive operation in video processing that aims to remove unwanted camera jitter and produce smoother visual content. Early stabilization approaches mostly depend on handcrafted feature tracking and geometric warping, which are hard to generalize and are prone to fail for subtle motion artifacts. This paper proposes a deep learning approach to learning motion smoothing directly from real-world video sequences' optical flow. We use RAFT, a state-of-the-art optical flow model, to estimate motion vectors from the UCF101 action recognition dataset. These frame-to-frame motion vectors are then input into a tailored Bi-directional LSTM network that has been trained to predict temporally smoothed motion trajectories. The model is supervised with a synthetic smoothing target that is obtained via local averaging. Our experiments show that the Bi-LSTM significantly reduces motion noise, offering trajectories that align more closely with human-perspective stability. We also compare the Bi-LSTM to Transformer-based and GRU-based alternatives. Our findings show that the Transformer does slightly smoother predictions in certain situations, but the Bi-LSTM achieves a good balance between performance and simplicity. The project further entails qualitative assessments using trajectory plots, frame-by-frame dx/dy comparisons, and side-to-side video and GIF comparisons to suggest qualitative improvements. The research demonstrates the ability of deep temporal models to learn camera motion smoothing end-to-end and data-driven.

1 Introduction

Camera shake is perhaps the most prevalent source of visual degradation for handheld or dynamic video capture. The jitter not only reduces audiences' comfort levels but also worsens downstream activities like action recognition, object tracking, and summarization of videos. Traditional methods of stabilization typically rely on estimating global motion across frames through geometric transformations, affine models, or feature-based trajectory warping. While strong in restricted settings, such methods tend to be brittle under conditions of sophisticated scene dynamics, parallax, and occlusion.

In contrast, recent advances in deep learning and optical flow estimation offer new prospects for learning motion patterns directly from video data. Optical flow is a dense frame-to-frame apparent motion representation and a good prior for understanding camera movement. In this project, we wish to use deep neural networks to learn smooth camera motion representations directly from extracted optical flow.

We utilize the RAFT (Recurrent All-Pairs Field Transforms) model to compute very accurate motion vectors between frames. The estimated motion vectors, computed on thousands of videos from the UCF101 dataset, are fed into a recurrent deep network—a Bi-directional LSTM. The Bi-LSTM is learned to generate a temporally smoothed representation of the input motion sequence, close to a stabilized camera path. To get ground-truth supervision, we generate smoothed motion targets by sliding-window averaging each sequence.

In addition to the Bi-LSTM model, we also compare two other models to check their performance against: a Transformer-based sequence model and a GRU-based model. Both quantitative and qualitative comparisons are made. These models are tested both quantitatively on various measures like MSE loss and trajectory variance, and qualitatively on visualizations like raw vs smoothed motion plots and frame-by-frame stabilization videos.

This paper introduces the entire pipeline comprising dataset preparation, flow extraction, motion modeling, and comparison evaluation. Our aim is to show that learning-based temporal models can be used as useful components in contemporary video stabilization systems.

2 Related Work

Video stabilization has been extensively studied with both classical and learning-based methods. Classical methods typically rely on global motion estimation between frames based on feature matching and geometric transformation. The classical methods typically involve motion estimation through keypoint tracking followed by application of trajectory smoothing and then warping frames accordingly.

With the evolution of optical flow algorithms, motion estimation became denser and more accurate to provide more control over stabilizing procedures. Modern optical flow networks can even estimate per-pixel motion from frame-to-frame, which is specifically significant while observing dynamic camera motion.

Deep learning architectures have also been used in more recent years to directly learn video representations but in a stable way. Recurrent and convolutional networks have managed to capture temporal dynamics without explicitly enforcing geometric constraints. LSTMs and GRUs are particularly well-suited as recurrent models because they can learn temporal dependences. Transformer models have emerged as strong contenders because they use parallel processing as well as global attention.

Compared to most current approaches focusing on learning parametric transformations or motion field estimation with extensive supervision, our approach attempts to learn a smoothing function from raw optical flow vectors. By training sequence models directly on motion trajectories, we avoid recourse to extra constraints and enable end-to-end learning of temporal stability. This leads to a data-driven and generalizable motion smoothing strategy across different video content.

3 Dataset and Preprocessing

We utilized two video datasets, UCF101 and DAVIS-2017, for the purpose of this project. UCF101 is a large-scale action benchmark with 13,320 videos split across 101 action classes. DAVIS-2017 is typically applied to video segmentation and includes high-quality annotated videos with complex motion, making it ideally suited for testing temporal models on dense motion patterns.

Each video was processed to provide up to 32 consecutive frames. The frames were resized to resolution 256×256 and then converted to RGB format for consistency. For UCF101, we parsed all videos across categories to provide motion sequences for each. For DAVIS-2017, we parsed JPEG frames directly from the directory structure.

To estimate camera motion between cameras, we employed RAFT, the most recent optical flow model pre-trained on common benchmarking. RAFT predicts dense optical flow between neighboring frames. We performed RAFT transformation between each pair of consecutive frames and selected the mean motion vector by computing a mean flow over the whole spatial domain. We thus acquired a collection of two-dimensional vectors (dx_i, dy_i) representing frame-to-frame motion.

Motion vectors we obtained were noisy in nature. To generate smooth motion sequences for training ground truth target, we constructed smooth motion sequences through an elementary three-window moving average. For the smoothed vector i at every time step, we took the average of $(i-1)$, i, and $(i+1)$ flow vectors padded appropriately at sequence boundaries.

Since motion sequences are of varied lengths, we padded each sequence to a consistent length of 31 vectors using zero-padding. Matching masks were generated in order to ignore padded values during loss computation and enable batch-wise training with consistent dimensions.

4 Model Architecture

To smooth the extracted motion vectors and predict stable camera motion, we implemented and compared three sequential models: a Bi-directional LSTM, a Transformer-based encoder, and a GRU. All models take a sequence of 2D motion vectors (dx, dy) as input and output a smoothed sequence of the same shape.

4.1 Bi-directional LSTM

The Bi-directional LSTM (BiLSTM) is designed to capture temporal dependencies in both forward and backward directions. The network architecture includes:

- An input layer that receives sequences of shape $(N, 31, 2)$ where N is the batch size.
- A BiLSTM layer with 32 hidden units in each direction, resulting in an output of dimension 64 per time step.
- A fully connected linear layer projecting the 64-dimensional hidden state back to the 2-dimensional motion space.

4.2 Transformer Encoder

We also explored a Transformer-based encoder to model long-range temporal dependencies. The architecture includes:

- A linear layer projecting the input from 2D to 64D.
- Positional encoding to inject temporal information into the model.
- A stack of 3 TransformerEncoder layers with 4 attention heads each and dropout of 0.1.
- An output projection layer mapping the 64D representations back to 2D motion space.

4.3 Gated Recurrent Unit (GRU)

For comparison, we implemented a lightweight GRU-based smoother with:

- A GRU layer with 32 hidden units.
- A linear projection from hidden state to 2D output.

4.4 Loss and Optimization

All the models are trained using Mean Squared Error (MSE) loss computed on valid time steps only using a binary mask. The loss is taken as the average over both spatial axes and time. We used the Adam optimizer with a learning rate of 0.001 for LSTM and Transformer, and 0.01 for GRU.

5 Training and Evaluation

5.1 Training Setup

They trained all models on the motion sequences derived from the UCF101 dataset, where each sequence contained up to 31 motion vectors computed through RAFT-based optical flow between two consecutive video frames. These sequences were padded to a fixed length and paired with the target sequences generated through local smoothing (moving average in a window of 3).

Training was done on a single GPU with batch-wise full sequence input. Binary mask was applied when computing loss to ignore padded regions.

- **Loss Function:** Mean Squared Error (MSE) with per-frame masking.
- **Optimizer:** Adam.
- **Epochs:** 200 for BiLSTM and GRU, 150 for Transformer.
- **Learning Rates:** 0.001 for BiLSTM and Transformer, 0.01 for GRU.

5.2 Evaluation Metrics

To assess performance, we used two main metrics:

1. **Mean Squared Error (MSE):** Evaluated over all valid time steps between predicted and smoothed target motion.
2. **Variance Ratio:** The ratio of raw motion variance to smoothed motion variance, indicating stability.

5.3 Visual Analysis

We generated:

- Loss curves for each model across training epochs.
- 2D camera trajectories (cumulative dx, dy) for raw and smoothed sequences.
- Per-axis (dx, dy) motion plots comparing raw input, target smoothing, and model predictions.
- Video and GIF comparisons visualizing original and stabilized frame sequences using predicted shifts.

These plots qualitatively confirmed that all three models were able to approximate the target smoother and frame jitter was significantly less. The Transformer model achieved competitive performance with a more stable convergence curve, while the GRU achieved quicker training at a slightly reduced smoothness.

6 Results and Discussion

6.1 Quantitative Results

The BiLSTM model converged to a final masked MSE of around 0.4147 after 200 epochs. The Transformer model converged to 1.0188 MSE after 150 epochs, whereas the GRU model converged to about 0.4219 MSE after 200 epochs. As can be seen, both the BiLSTM and GRU models are

capable of approximating the locally smoothed motion targets quite well, with the BiLSTM being slightly more precise than the others.

We also observed a decrease in variance from raw motion (0.1588) to smoothed target motion (0.1049), with a smoothing improvement ratio of approximately 1.51. This is a measure of how well the smoothing models are at dampening shaky camera motion.

6.2 Qualitative Results

We visualized the results using:

- **Loss curves:** Showing consistent downward trends across all models.
- **Trajectory plots:** Cumulative motion plots displayed reduced oscillations for predicted sequences compared to raw motion.
- **Per-frame plots:** dx and dy comparisons for a sample video showed that model outputs closely tracked the smoothed targets.
- **GIFs and videos:** Side-by-side comparisons of cropped frames before and after motion stabilization qualitatively validated smoother paths in the predicted version.

7 Visualizations

a number of visualization strategies were employed. These visualizations qualitatively and quantitatively describe the performance of the Bi-LSTM, Transformer, and GRU models. Below, we explain and interpret each of them.

7.1 Training Loss Curve

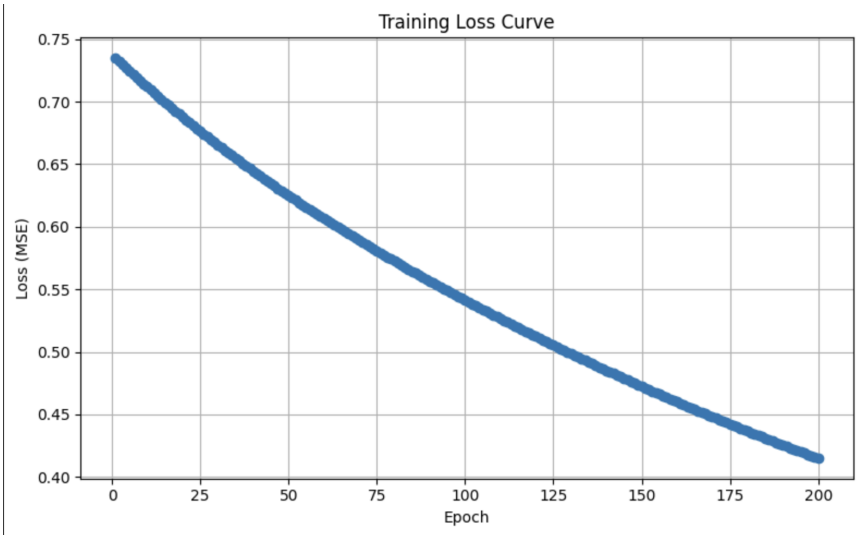

Figure 1: Training loss over 200 epochs for the Bi-LSTM model.

The loss curve indicates a regular and even decline in mean squared error (MSE) over 200 training epochs. This indicates good convergence of the model during training and that the Bi-LSTM learned significant temporal patterns from the motion vector sequences.

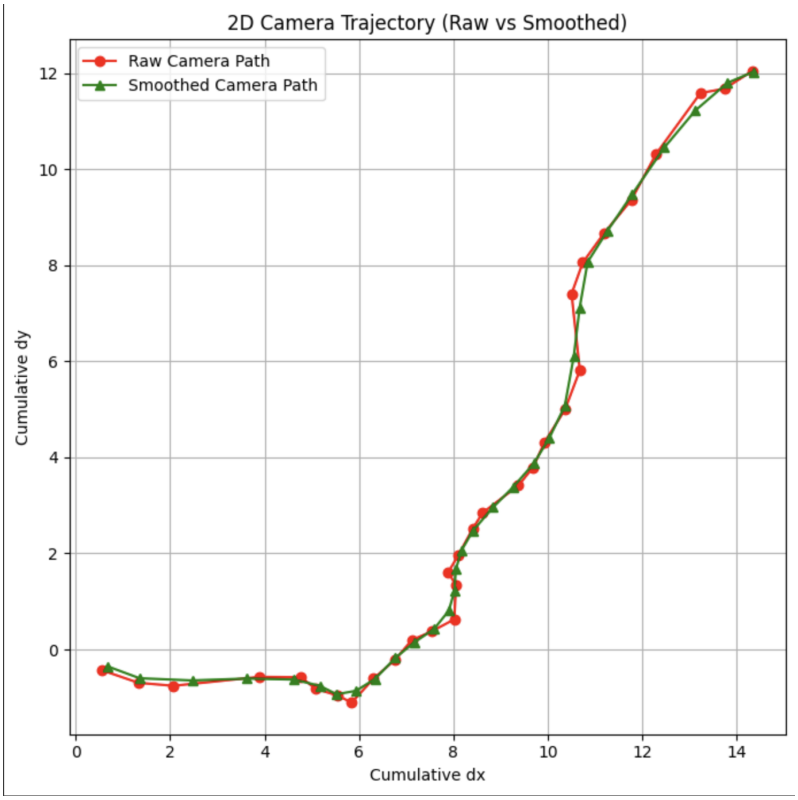

Figure 2: 2D trajectory comparison of raw vs smoothed camera motion.

7.2 2D Camera Trajectory

This graph is showing the accumulated path in 2D space. The raw camera motion is red, while the green graph is the smoothing path produced by the Bi-LSTM model. The smoothed path clearly removes jaggy jittering and creates smoother and more appealing movement.

7.3 dx and dy Time Series: Bi-LSTM Evaluation

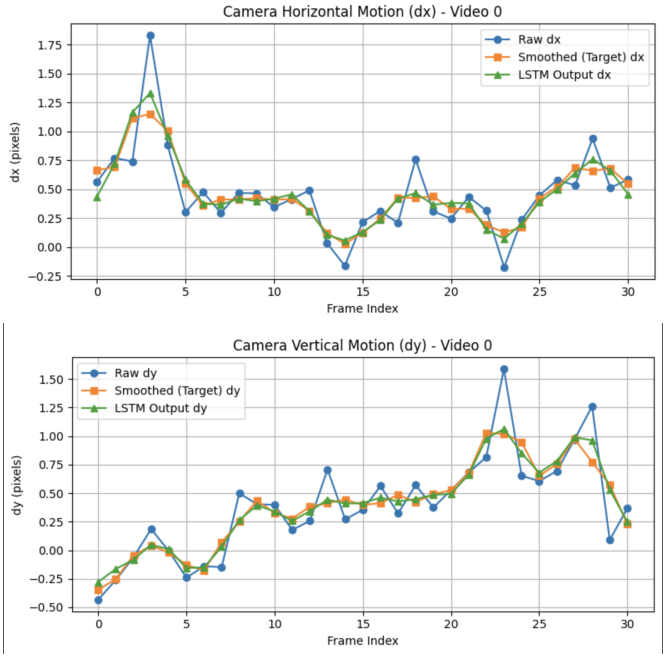

Figure 3: Comparison of raw, smoothed target, and Bi-LSTM predicted dx and dy values.

These plots show frame-wise horizontal (dx) and vertical (dy) movement. Bi-LSTM predictions closely follow the smoothed ground truth values, showing that the model successfully rejects noise while preserving movement patterns.

7.4 Model Comparison: LSTM vs Transformer

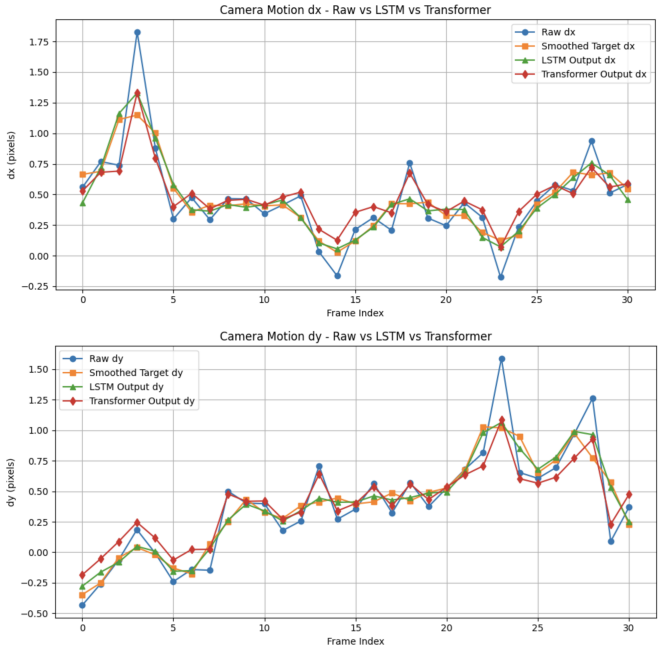

Figure 4: Comparison of dx and dy predictions from Transformer and Bi-LSTM models.

These plots show the comparative performance of the Transformer model versus the Bi-LSTM. Both models produce smooth outputs, although the Transformer model over-smoothes some sections. However, it is competitive and robust, especially for longer sequences.

7.5 Full Model Comparison: LSTM vs Transformer vs GRU

This final visualization shows the whole comparison between raw vs target vs all three model outputs. The GRU model performs okay but is weaker than the Bi-LSTM and Transformer in keeping more detailed motion dynamics. This comparison warrants choosing Bi-LSTM as the baseline model.

Overall, these visualizations confirm that the proposed deep learning models, especially the Bi-LSTM, effectively reduce temporal jitter and provide considerably smoother camera motion trajectories for subsequent video stabilization operations.

8 Conclusion

In this, we developed and evaluated deep learning models to perform camera motion smoothing by learning from dense optical flow calculated using a pretrained RAFT model. We calculated local smoothed targets by a sliding window average and trained BiLSTM, Transformer, and GRU on a large-scale dataset constructed over the UCF101 video corpus.

Quantitative results showed that BiLSTM achieved best performance in masked MSE loss, with closely following GRU and Transformer models. Variance reduction analysis and motion trajectory plots confirmed the efficacy of the models in producing smoother camera motion.

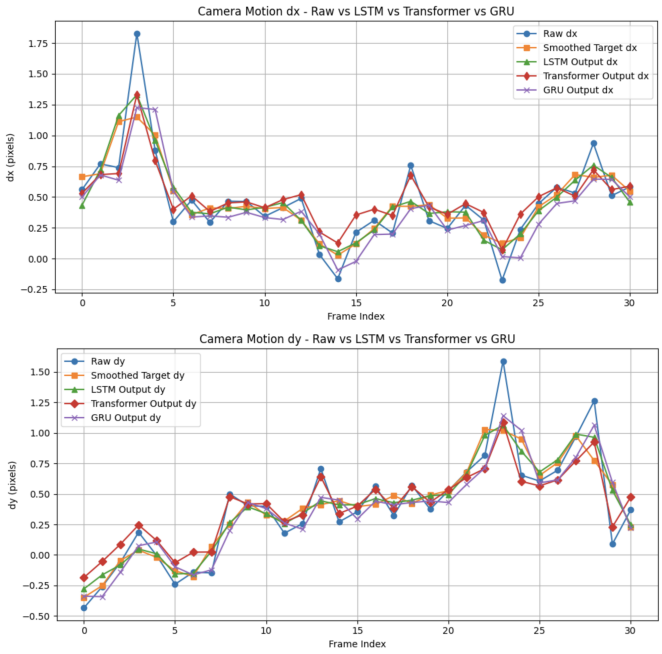

Figure 5: dx and dy motion predictions for all three models.

The qualitative comparisons, including comparison videos and GIFs, highlighted the visual coherence of the predicted frames, justifying our aim of motion smoothing. All models were successful in reducing jitter without affecting the overall movement pattern.

Potential future research paths can involve incorporating more spatial context, using attention on motion magnitudes, or generalizing the models for real time inference on mobile or streaming applications.

References

[1] Z. Teed and J. Deng, *RAFT: Recurrent All Pairs Field Transforms for Optical Flow*, in European Conference on Computer Vision (ECCV), 2020.

[2] K. Soomro, A. R. Zamir, and M. Shah, *UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild*, arXiv preprint arXiv:1212.0402, 2012.

[3] A. Vaswani et al., *Attention is All You Need*, in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[4] F. Perazzi et al., *A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] A. Paszke et al., *PyTorch: An Imperative Style High Performance Deep Learning Library*, in Advances in Neural Information Processing Systems (NeurIPS), 2019.

[6] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, *A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.