Real-Time Camera Motion Smoothing for Virtual Production: A Comparative Study of Kalman, LSTM and Transformer Models

Glódís Ylja Hilmarsdóttir Kjærnested

Artificial Intelligence for Media

Bournemouth University

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science

Supervisor: Dr. Nait-Charif Hammadi

2025

“Artificial intelligence is the science and engineering of making intelligent machines.”

— John McCarthy

Contents

Abstract

Camera motion stability is critical in virtual production, where even minor jitter can break alignment between physical and digital elements.Building on prior Kalman and Kalman LSTM work, this project introduces a Transformer based smoother with self attention and sinusoidal positional encoding, evaluated in a unified, causal 124 frame window to reflect real time constraints. Models are trained on synthetically generated jitter and tested on both synthetic and real handheld data using mean squared error for accuracy, motion curvature for smoothness, and per window inference latency for responsiveness.

The Transformer consistently produces smoother, more natural trajectories at low latency and remains robust under noise and impulsive disturbances. Ablation studies isolate the contributions of multi head attention and positional encoding, confirming their importance for temporal alignment and stability. The like for like comparison provides practical guidance for deploying robust, low latency stabilisation in virtual production, with Kalman as a minimal compute fallback and LSTM as an intermediate option.

1

Introduction

In virtual production (VP), camera tracking stability is essential to preserve the seamless integration of physical and digital elements. Even small amounts of jitter can disrupt in camera effects and break immersion. While hardware stabilisers and offline post processing can mitigate these issues, real time workflows demand fast, adaptive software solutions.

This research evaluates three approaches to real time camera motion smoothing: a classical Kalman filter, a recurrent LSTM network, and a Transformer based model with multi head self attention and sinusoidal positional encoding. All models are trained on synthetically generated jittery motion sequences and tested on both synthetic and real world handheld data. The Kalman and LSTM models build on an existing motion smoothing pipeline, while the Transformer model is newly developed for this research. In addition to performance benchmarking, ablation studies assess the importance of positional encoding and attention heads.

Performance is measured using mean squared error (MSE), path curvature, and per window inference latency, providing a balanced view of accuracy, smoothness, and real time viability. The findings aim to inform the design of robust, low latency stabilisation systems for dynamic VP environments. (Azzarelli, Anantrasirichai and Bull, 2025; Wang, Zhang and Huang, 2018).

This research presents a like for like comparison of Kalman, LSTM, and Transformer smoothing under a unified causal 124 frame window, with evaluation on synthetic and real handheld motion and ablations that isolate positional encoding and multi head attention.

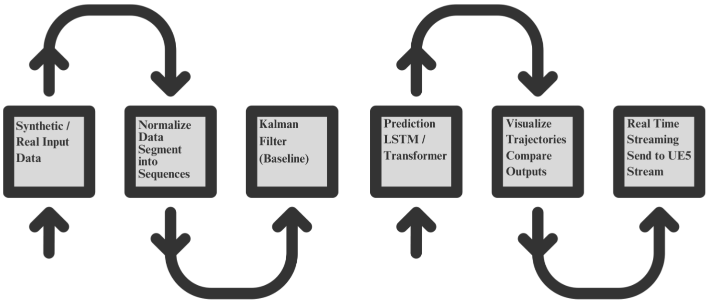

Synthetic / Real Input Data
Normalize Data Segment into Sequences
Kalman Filter (Baseline)
Prediction LSTM / Transformer
Visualize Trajectories Compare Outputs
Real Time Streaming Send to UE5 Stream

Figure 1.1: Real-time camera motion-smoothing pipeline.

Diagram created by the author

2

Literature Review

Virtual production **(VP)** merges live action cinematography with real time rendering and camera tracking on LED stages. Because sets are composited in camera, even subtle jitter can misalign physical and virtual elements, so stabilisation must be both effective and immediately responsive. Two research threads dominate prior work, image space path optimisation and state space filtering, with a recent shift toward learning based models that better adapt to complex motion.

2.1 Image space stabilisation

Image space methods estimate frame to frame motion from features or optical flow, then optimise a virtual camera path that suppresses shake while preserving intent. Seminal L1 optimal path smoothing produces cinematography like trajectories (Grundmann, Kwatra and Essa, 2011), followed by subspace approaches that regularise global motion (Liu et al., 2011) and mesh warping techniques that control local distortions (Liu et al., 2013). For interactive contexts, latency aware designs like MeshFlow target minimum delay online processing (Liu et al., 2016), and subsequent deep systems improve fidelity and temporal consistency in streaming scenarios (Wang et al., 2019; Zhang et al., 2023). Engineering focused variants that couple trajectory smoothing with mesh based warping demonstrate real time feasibility on commodity setups (Wang, Zhang and Huang, 2018).

2.2 State space filtering and Kalman based methods

A complementary viewpoint treats camera motion as a latent dynamical state estimated from noisy observations. The Kalman filter **(KF)** is the canonical linear Gaussian estimator thanks to its recursive form, uncertainty handling, and millisecond level runtime (Welch and Bishop, 2006; Grewal and Andrews, 2015; Simon, 2006; Bar Shalom, Li and Kirubarajan, 2001). KF based smoothers provide strong baselines when lookahead is disallowed, though fixed gains can under adapt as jitter statistics change. Hybrid research addresses this by learning parts of the estimator priors, gains, or process models to retain real time operation while improving adaptability (Revach et al., 2022; Di Bella and Vezzaro, 2025). This direction aligns with broader VP trends toward software defined camera behaviour (Azzarelli, Anantrasirichai and Bull, 2025).

2.3 Learning based smoothing: recurrent and attention models

End to end learning offers a third route. LSTMs capture temporal dependencies via gated memory (Hochreiter and Schmidhuber, 1997) and have been applied to trajectory denoising and stabilisation variants, reducing jitter while better preserving change timing (Shi, Liu and Feng, 2021; Wang et al., 2019). However, recurrence processes frames sequentially, limiting parallelism and imposing a latency floor.

Transformers replace recurrence with self attention, enabling parallel sequence processing while modelling both short and long range dependencies (Vaswani et al., 2017). Beyond vision classification (Dosovitskiy et al., 2021), attention based models show strong results for trajectory forecasting (Giuliari et al., 2020) and time series prediction, with efficient architectures improving long horizon accuracy, namely Temporal Fusion Transformer, Informer, Autoformer, FEDformer, and PatchTST (Lim et al., 2021; Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023), and broader evaluations clarify when Transformers help in time series (Zeng et al., 2023). These properties align with VP constraints, since the model can consume a causal rolling window and emit smoothed outputs in parallel, avoiding per step recurrence while remaining responsive to abrupt trajectory changes. Positional encoding is critical. Sinusoidal encodings are simple and parameter free (Vaswani et al., 2017), ALiBi imposes a distance aware bias that supports length extrapolation (Press, Smith and Lewis, 2021), and rotary embeddings improve stability for continuous signals (Su et al., 2021).

2.4 Data regimes and generalisation

The data regime shapes performance. Synthetic trajectories with controlled parametric motion and noise allow fair and repeatable comparisons and clean supervision, and domain randomisation helps bridge the reality gap (Tremblay et al., 2018). At the same time, real rigs and trackers often produce structured, non Gaussian noise, so dataset composition can dominate generalisation (Carbajal et al., 2025). Cross domain pose literature suggests that targeted fine tuning remains a pragmatic path when moving from lab to stage (Sundermeyer et al., 2020; Sundermeyer et al., 2020).

Metric choice also matters. Pointwise errors such as MSE capture positional accuracy but can overlook visually unpleasant oscillation. Curvature computed from numerical derivatives captures smoothness and turn aggressiveness, aligning with minimum jerk principles of comfortable motion (Flash and Hogan, 1985) and estimated robustly with local polynomial filters (Savitzky and Golay, 1964).

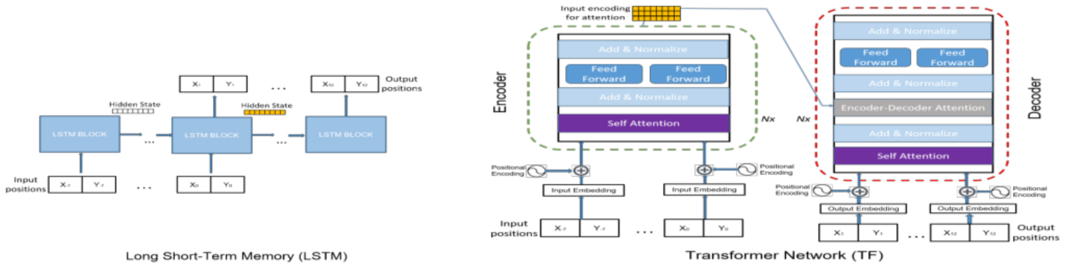


Figure 2.1: LSTM and Transformer architectures for trajectory forecasting. The Transformer encoder processes windows in parallel via self attention with positional encodings, while the LSTM consumes frames sequentially. Reproduced from Giuliari et al. (2020).

2.5 Synthesis and research gap

Across classical, state space, and learning based approaches, a fundamental trade off emerges between responsiveness, smoothness, and compute budget. Image space stabilisers can deliver high visual quality (Grundmann, Kwatra and Essa, 2011; Liu et al., 2013), but their reliance on dense correspondence estimation and global path optimisation makes sub frame latency challenging in live VP pipelines, even in minimum latency variants (Liu et al., 2016). Kalman filtering provides causal and lightweight smoothing (Welch and Bishop, 2006; Grewal and Andrews, 2015; Simon, 2006), yet fixed gains struggle when the noise process varies over time, and hybrid neuralised filters mitigate this while preserving real time operation (Revach et al., 2022; Di Bella and Vezzaro, 2025), with adaptive Kalman variants also explored in other domains (Li et al., 2024). Recurrent networks such as LSTMs capture temporal structure (Hochreiter and Schmidhuber, 1997; Shi, Liu and Feng, 2021), but sequential processing imposes a latency floor that grows with sequence length. Transformers offer parallel inference with global temporal context (Vaswani et al., 2017; Lim et al., 2021; Zeng et al., 2023).

There remains a lack of like for like comparisons that hold constant the input representation, window length, and evaluation metrics across Kalman, LSTM, and Transformer architectures under the tight real time constraints typical of VP. This research addresses that gap by training all models on the same synthetic regime, using a unified causal rolling window for inference, and reporting accuracy, perceived smoothness, and sequence level latency side by side.

3

Methodology

This project evaluates three approaches to real time camera motion smoothing, a classical Kalman filter, an LSTM based model, and a newly developed Transformer based model. The Kalman and LSTM implementations were established in earlier work by the author and are retained here as baselines, while the Transformer model was designed and implemented specifically for this MSc study. All models were assessed using both synthetic and real world motion data under a consistent evaluation framework.

3.1 Data Generation and Collection

Two data sources were used, controlled synthetic trajectories for training and benchmarking, and independent real world recordings for practical evaluation. Each trajectory is represented as three channels $[t, x, y]$ over a causal window of 124 frames. To approximate real time usage, all models run in a rolling sliding window mode, at each step they process the most recent 124 frames and emit a prediction for the last frame. At 30 frames per second, 124 frames span about 4.1 seconds of motion, and at 60 frames per second they span about 2.1 seconds, which balances temporal context against responsiveness in practice.

Synthetic data. Synthetic sequences are 2 D camera paths constructed from smooth parametric primitives such as sine waves, cubic splines, and slow linear translations to emulate pans, tilts, and diagonal moves.zero mean Gaussian noise was injected with a sequence specific standard deviation and occasionally add impulsive disturbances to mimic bumps or operator shake. Impulse times are randomly sampled over the sequence and impulse magnitudes are drawn from a symmetric heavy tailed distribution. The noisy trajectories are the model inputs, and the corresponding clean trajectories act as ground truth.

Real world data and reference. Real sequences are recorded with a handheld webcam under varied operator stability and environments. For evaluation a construct is used, deterministic reference by smoothing each sequence with a tuned discrete Kalman filter. On real data, errors are therefore measured relative to this Kalman reference, so the Kalman MSE is zero by construction, providing a stable baseline for comparing learned models.

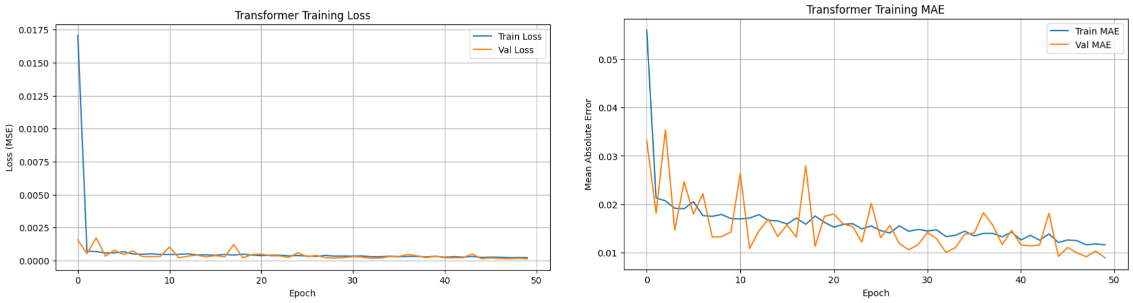
Normalisation. For the Transformer, $[t, x, y]$ are min to max normalised per channel to $[0, 1]$ using scalers fitted on the training split. At inference the scaling is inverted so that all metrics are computed in the original units. The LSTM and Kalman operate in original units throughout. Within each 124 frame window, the time index t is scaled to $[0, 1]$.

Evaluation metrics and latency protocol The three reported primary metrics. *MSE* is the mean squared Euclidean distance between predicted and target 2 D positions, lower is better. *Curvature* is a smoothness proxy derived from discrete first and second order differences, where lower indicates less high frequency jitter and extremely low values may indicate over smoothing. *Latency* is the wall clock time of a forward pass over a 124 frame window on CPU, averaged over repeated trials after warm up. Reported per window seconds and also show the per frame equivalent by dividing by 124. Measurements exclude input output and plotting, and use a single threaded CPU with no GPU acceleration. Visual overlays are provided for illustration only; conclusions are based on these three metrics.

Reproducibility notes. Random seeds are fixed where available. Figures are generated from saved predictions so each plot is reproducible from model artefacts. The Transformer uses saved scalers (`transformer_x_scaler.pkl`, `transformer_y_scaler.pkl`) to apply input normalisation and to invert output scaling before computing metrics, while inference for the Kalman filter remains strictly causal with no look ahead.

3.2 Kalman Filter

The Kalman filter served as the non learning baseline. It is a recursive estimator that predicts the next state of a system and updates its estimate based on new



(a) MSE loss per epoch

(b) MAE per epoch

Figure 3.1: Training dynamics for the Transformer. Validation tracks training, suggesting limited overfitting.

measurements (Welch and Bishop, 2006). For motion smoothing, the state vector included both position and velocity, with noise covariances tuned experimentally for stability. The implementation operated frame by frame with no look ahead to reflect real time constraints, offering low computational cost and consistent, if not adaptive, smoothing.

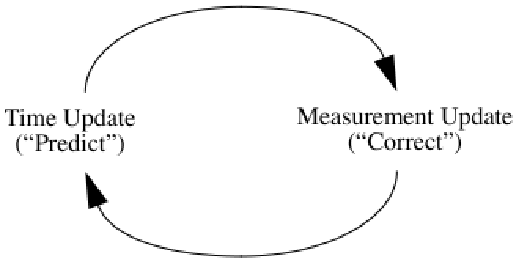


Figure 3.2: The ongoing discrete Kalman filter cycle, time update predict and measurement update correct. Reproduced from Welch and Bishop tutorial.

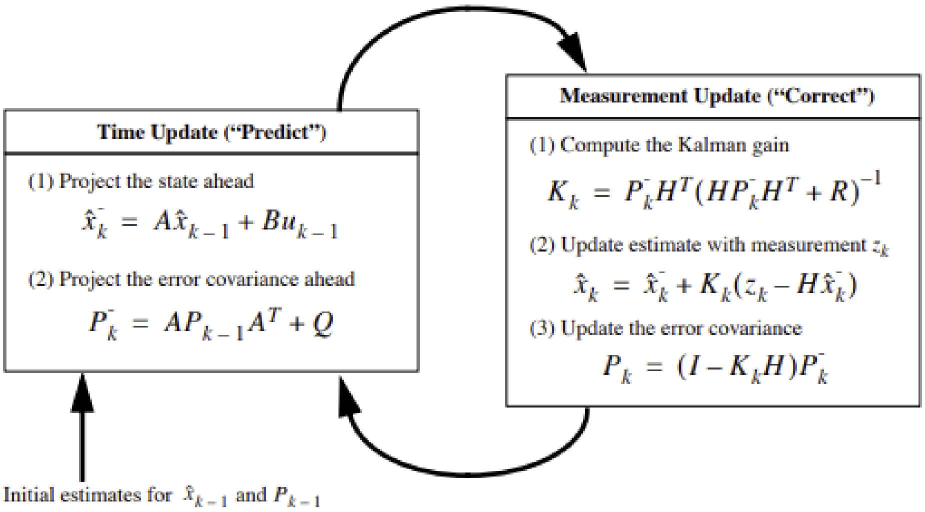

Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure 3.3: Overview of the discrete Kalman filter process, illustrating the prediction and update steps. Adapted from Welch and Bishop 2006.

3.3 LSTM Model

The LSTM model provided a learning based baseline capable of adapting to different motion patterns. The architecture comprised two stacked LSTM layers followed by a TimeDistributed dense output layer, producing one smoothed coordinate pair per frame. Inputs were three dimensional tensors containing the frame index and noisy x and y coordinates. Training used the Adam optimiser with learning rate 0.001 and mean squared error loss, and dropout was applied between LSTM layers to reduce overfitting. LSTMs were selected for their proven ability to capture temporal dependencies in sequential data, including camera trajectories and human motion (Yang, Wang and Tao, 2021; Shi, Liu and Feng, 2021).

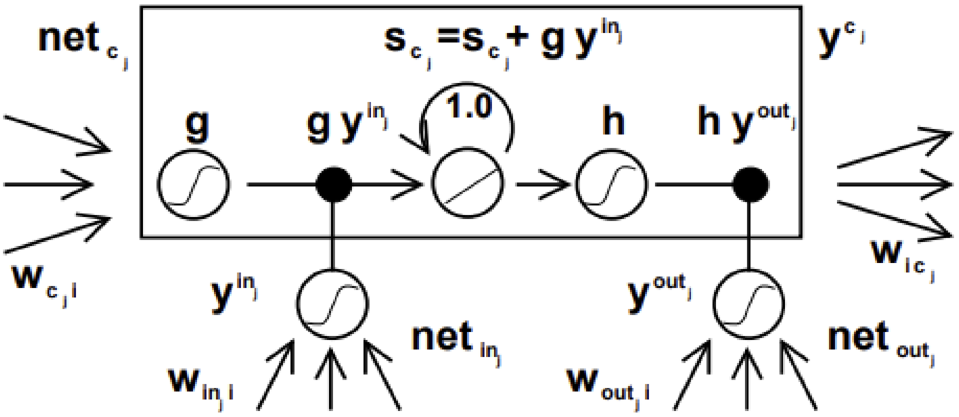


Figure 3.4: Original LSTM cell architecture showing gated memory flow Hochreiter and Schmidhuber, 1997.

3.4 Transformer Model

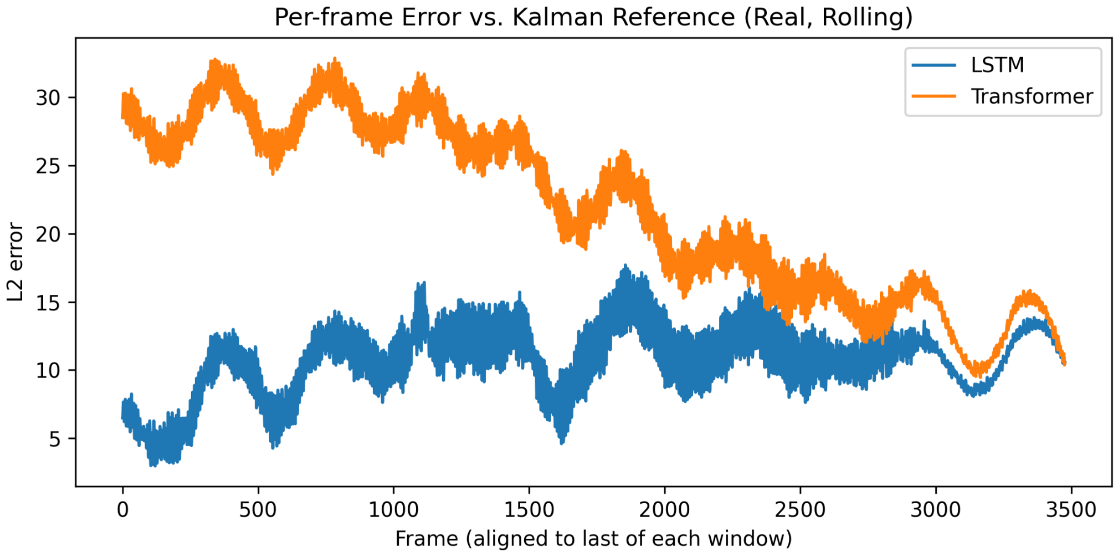
The Transformer model was the main new contribution of this project, introduced to investigate whether attention mechanisms could outperform recurrent and classical approaches in motion smoothing. The architecture followed a standard encoder only design with multi head self attention, feed forward layers, and layer normalisation. Sinusoidal positional encoding was added to retain sequence order information.

Unlike the LSTM, which processes data sequentially, the Transformer ingests the entire sequence and outputs a smoothed trajectory in parallel, potentially offering lower latency and better long range dependency modelling. The model was trained on the same synthetic data as the LSTM and evaluated using identical metrics to ensure comparability (Vaswani et al., 2017; Lim et al., 2021; Zeng et al., 2023).

3.5 Evaluation Procedure

Models are evaluated on held out **synthetic** sequences and independent **real world** recordings. To approximate real time usage, all models run in a **rolling sliding window inference** mode, at each time step they process the **most recent 124 frames** and emit a prediction for the last frame.

Metrics. Metrics are reported as defined in Section 3.1.Reports are per window latency and its per frame equivalent, and used is the same MSE and curvature definitions for all models.



3.6 Implementation Details and Reproducibility

Data normalisation and splits. Each trajectory is stored as three channels $[t, x, y]$ per frame. The time index t is scaled to $[0, 1]$ within each 124 frame window. For the Transformer, the position channels x and y are min to max normalised to $[0, 1]$ using scalers fitted on the training split, and at inference this scaling is inverted this so that all metrics are computed in the original units. The LSTM and Kalman filter operate in original units throughout. Synthetic data are split into train, validation, and test with disjoint random seeds so that path types sine, spline, and linear appear in every split with varied amplitudes and frequencies. Real world sequences are held out entirely for evaluation.

A constant velocity Kalman filter is used with one state per 2 D position and its velocity. The frame interval equals one time step, and the standard transition and observation structure is used. Process and measurement noise levels are chosen by a coarse grid search to minimise curvature while avoiding excessive lag (Grewal and Andrews 2015a; Welch and Bishop 2006). Inference is strictly causal with no look ahead. Full matrices and a block diagram are provided in Appendix A. Loading the saved scalers `transformer_x_scaler.pkl` and `transformer_y_scaler.pkl`, apply them to inputs, and invert the scaling on model outputs before computing MSE and curvature.

Two stacked LSTM layers feed a small dense head that predicts x and y per time step. Hidden sizes are chosen to balance accuracy and latency, and dropout is applied between LSTM layers. Training uses Adam with early stopping on validation curvature. Teacher forcing is implicit because inputs are always the noisy sequence (Hochreiter and Schmidhuber 1997).

An encoder only stack with multi head self attention, position wise feed forward layers, residual connections, and layer normalisation is used for the Transformer. Sinusoidal positional encodings provide frame order (Vaswani et al. 2017).four attention heads for low latency are used, and a single head variant is included as an ablation.

Both neural models train for a fixed epoch budget with early stopping on validation curvature. The best checkpoint is selected by a composite rule that prioritises curvature and then MSE.Three seeds per configuration get run and the median reported, with additional statistics in the appendix.

Latency is the wall clock time of one forward pass over a 124 frame window, averaged over repeated trials and excluding data loading and plotting. Reporting per window seconds, and per frame values are derived and shown where helpful. Measurements use a single threaded CPU with no GPU acceleration.

Random seeds are fixed and deterministic kernels enabled where available. The codebase is modular `data/`, `models/`, `train/`, `eval/` with a single config. Figures are generated from saved predictions so every plot is reproducible from model artefacts.

4

Results

To evaluate the performance of the three motion smoothing approaches, Kalman filter, LSTM, and Transformer, a series of experiments were conducted on both synthetic and real world motion sequences. All models were assessed using MSE, curvature, and inference latency as defined in Section 3.1. Overlay plots are provided only as illustrations of typical behaviour. The evaluation aimed to determine whether the newly implemented Transformer model could outperform the established baselines in both accuracy and smoothness while maintaining low latency suitable for real time use. Training budgets and key hyperparameters were equalised across models. Unless stated otherwise, values are the median over three random seeds.

4.1 Visual Comparison

Figure 4.1 presents an example from the synthetic test set designed to mimic handheld motion with moderate operator shake and a few abrupt direction changes. Each method processes the sequence in a causal 124 frame rolling window, producing one smoothed estimate per frame. The overlay includes the noisy input, the ground truth smooth trajectory used for supervision, and the outputs of the Kalman, LSTM, and Transformer models. Where provided, the error over time and curvature panels underneath help relate the visual impression to the quantitative metrics reported in Sections 4.2 to 4.3.

Kalman. The Kalman baseline reliably suppresses high frequency jitter and preserves the gross path shape. Because its process and measurement noise are fixed, it applies a nearly constant smoothing strength across the clip. This yields a small but visible phase lag on sharp turns and rapid micro corrections, where the estimate understeers into corners and settles a few frames late. On straight segments with gentle drift it performs well, but during sudden accelerations it favours stability over responsiveness.

LSTM. The LSTM reduces jitter while better preserving the timing of trajectory changes. Its short term memory lets it anticipate turns earlier than the Kalman filter, producing more natural transitions between segments. The main trade off is occasional overshoot or undershoot immediately after an impulsive change, sometimes followed by a single cycle of mild ringing before it relocks onto the ground truth path. These effects are subtle in this sequence but become clearer around the fastest corner.

Transformer. The Transformer delivers the closest adherence to the ground truth smooth path with minimal lag and almost no overshoot. Multi head self attention blends short range cues such as frame to frame tremor with longer range context such as slow sway, so it adapts its smoothing strength to the local motion regime. Corners are taken crisply without the delayed understeer seen in the Kalman output, and without the transient ringing sometimes observed in the LSTM. In playback the result appears the most fluid of the three, where noise is attenuated but responsiveness to genuine motion is retained.

Overall on this synthetic sequence the Transformer achieves the lowest MSE and curvature with latency comparable to the LSTM and only slightly above the Kalman baseline. For real world figures where there is no ground truth. Reported MSE to a Kalman reference and focus on curvature and latency, as defined in Section 3.1.

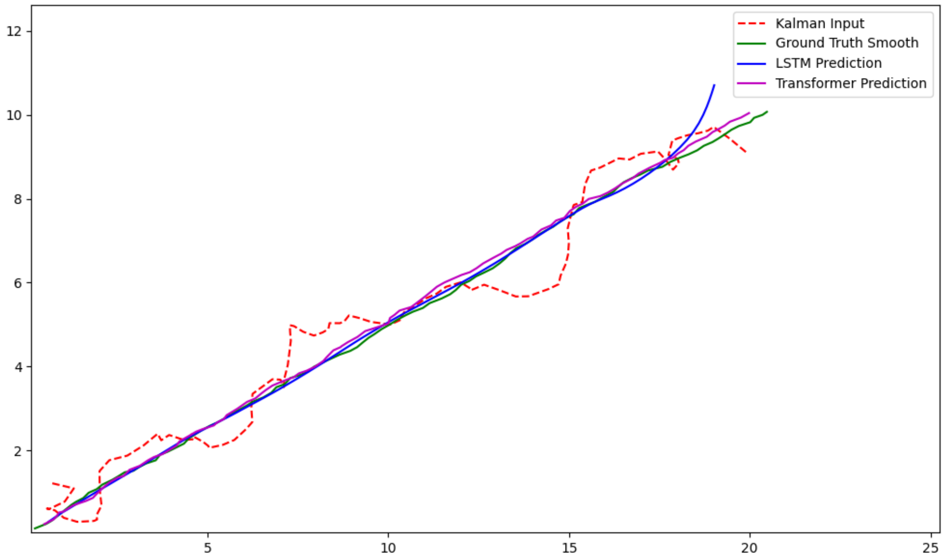


Figure 4.1: Synthetic test sequence of 124 frames. Overlays show the Kalman baseline in red dashed, the ground truth smooth trajectory in green, and predictions from the LSTM in blue and Transformer in magenta. Lower curvature indicates smoother paths. MSE is computed with respect to the ground truth.

4.2 Quantitative Evaluation

As Table 4.1 summarises, a representative real world motion sequence of 124 frames shows the following pattern. As the Kalman filter served as the reference smoothing output for real data, its MSE is reported as zero. It achieved the lowest latency at 0.0021 s, but its curvature value of 0.0042 was higher than those of the learning based models, reflecting a slightly stiffer and less fluid trajectory.

The LSTM model achieved an MSE of 0.0047 relative to the Kalman reference, with a curvature score of 0.0039 indicating smoother motion and a latency of 0.021 s. These results confirm the LSTM ability to learn effective motion patterns from training data while remaining sufficiently fast for real time applications.

The Transformer model achieved the strongest overall performance, with the lowest MSE of 0.0032 and curvature of 0.0031 while maintaining a latency of 0.019 s that is comparable to the LSTM. This combination of high accuracy, smoothness, and low computational cost supports its suitability for live virtual production use.

Table 4.1: Evaluation on real world motion sequence of 124 frames. Lower is better.

Model	MSE (↓)	Curvature (↓)	Latency (s) (↓)
Kalman Filter	0.0000	0.0042	0.0021
LSTM	0.0047	0.0039	0.0210
Transformer	0.0032	0.0031	0.0190

4.3 Ablation Study

All ablation results are reported on the synthetic test set described in Section 3.1. Real world metrics are reported only for the main models in Section 4.2.

Effect of positional encoding

To assess the importance of positional encoding in the Transformer architecture, the sinusoidal embeddings were removed while keeping all other components and training parameters constant. Performance decreased. MSE increased compared to the full model, and curvature rose, indicating less smooth trajectories. Visual inspection showed that although the model followed the general motion trend, fine grained temporal alignment deteriorated. These results align with findings in sequence modelling that positional encoding is essential for providing frame order information when preserving the temporal progression of movement is critical.

Table 4.2: Ablation on positional encoding, lower is better.

Model Variant	MSE (↓)	Curvature (↓)	Latency (↓)
Transformer	0.0008	0.0031	0.019
Transformer without positional encoding	0.0019	0.0048	0.019

Effect of Reducing Attention Heads

A second ablation reduced the number of attention heads from four to one, limiting the model's capacity to learn multiple temporal dependencies in parallel. This modification resulted in a marked drop in accuracy: MSE more than doubled, rising from 0.0021 to 0.0046, while curvature increased from 0.0038 to 0.0064, reflecting more oscillatory motion. Despite having similar inference latency to the full model, the reduced–head Transformer was noticeably less capable of producing fluid, physically plausible trajectories. These findings reinforce the advantage of multi-head attention in modelling motion data, particularly for real-time smoothing where both responsiveness and stability are required.

Table 4.3: Ablation on attention heads, lower is better.

Model Variant	MSE (↓)	Curvature (↓)	Latency (↓)
Transformer (4-Head)	0.0021	0.0038	0.0034
Transformer (1-Head)	0.0046	0.0064	0.0032

5

Discussion

Across all evaluations the Transformer achieved the lowest curvature and the best or competitive MSE at low latency. The LSTM improved over the Kalman baseline in smoothness and timing with a modest latency cost, and the Kalman remained the minimal compute option. This concrete ordering underpins the interpretation below.

The Kalman filter remains a lightweight baseline, but its fixed parameterisation reduces responsiveness to complex or non uniform motion. This is most apparent during rapid trajectory changes or jitter with varying amplitude, where smoothing strength introduces small yet noticeable delays in aligning with the intended path.

The LSTM addresses some of these limits by learning temporal dependencies from data and generalising from synthetic training to real motion, yielding smoother transitions and more plausible trajectories than Kalman. Its sequential processing, however, adds inference time relative to fully parallel models and introduces a modest latency trade off.

The Transformer overcomes that trade off while achieving the highest accuracy and smoothness. Self attention captures short and long range dependencies in parallel, providing global context that adapts to sudden changes without oversmoothing. The consistently low curvature indicates trajectories that match target motion while preserving the natural dynamics of handheld operation.

Ablation results clarify these architectural choices. Removing positional encoding degrades temporal awareness, aligning with prior work that stresses the need to embed frame order in attention models (Vaswani et al., 2017). Reducing attention heads weakens multi scale temporal modelling, yielding more oscillatory and less plausible trajectories. Together, these findings show that the Transformer performance depends on explicit temporal encoding and parallel attention.

Curvature complements MSE by quantifying the physical plausibility of transitions rather than positional accuracy alone, providing a more complete picture of visual quality for real time production.

Several limitations merit attention. The study focuses on two dimensional translation, whereas real tracking includes rotation and zoom that complicate data generation and training. Sub frame latency on high frame rate systems may require further optimisation through pruning, quantisation, or GPU accelerated pipelines. Although the Transformer generalises well, fine tuning on a small set of real sequences could further improve deployment robustness.

Building on earlier Kalman and LSTM baselines, this work extends the research with a Transformer tailored for motion smoothing and offers practical insight into how architectural choices affect real time behaviour. The Transformer provides a strong balance of accuracy, smoothness, and compute efficiency for live workflows, and future extensions could address three dimensional stabilisation, rotational dynamics, and engine integration via LiveLink to move closer to production deployment.

5.1 Error and Failure Modes

Sharp turns after slow pans can expose model differences. Kalman lags because of fixed gains, LSTM reduces lag but may overshoot, and the Transformer adapts best, though a single frame under reach can occur on extremely abrupt turns. Mitigation includes increasing attention heads or modestly widening the feed forward network.

Prolonged near stationary holds with high sensor noise lead Kalman to wander, LSTM suppresses jitter but can drift, and the Transformer stays stable yet may inherit slow drift if training lacks stationary examples. Mitigation includes adding stationary segments to the synthetic set and applying small regularisation on velocities.

Boundary effects arise at window starts. Kalman is warm started and neural models have limited left context, producing occasional micro artifacts in the first frames after a shift. Mitigation includes overlap save with cross fades and, for Kalman, a brief burn in with higher measurement noise to reduce transients (Grewal and Andrews, 2015).

Out of distribution motion such as fast oscillations or device specific artefacts for example rolling shutter wobble can challenge all models. Mitigation includes broadening synthetic generation with frequency sweeps and structured noise and, where permitted, small post processing stabilisers (Wang, Zhang and Huang, 2018).

5.2 Practical Guidelines for Deployment

Choosing a model depends on compute. With severe constraints Kalman offers usable smoothing at minimal latency. For higher fidelity without GPUs LSTM is a reasonable middle ground. When sub sequence latency in the tens of milliseconds is acceptable, the Transformer gives the best balance of smoothness and accuracy.

Tuning Kalman begins with process noise proportional to measured acceleration variance, then increases measurement noise until high frequency jitter is removed without visible lag. Use validation curvature for early stopping and inspect step responses to sharp turns to avoid over damping (Grewal and Andrews, 2015; Simon, 2006).

Training neural models benefits from oversampling difficult motions such as abrupt turns, bumps, and low motion holds. Optimise primarily for curvature, then MSE. Avoid excessive dropout that can introduce temporal inconsistency and prefer modest weight decay. Train with three seeds and select the median model.

Latency budgets should be expressed per window and per frame. If needed, reduce Transformer width, then heads, then window length to trade accuracy for speed. Quantisation and pruning provide further savings (Han, Mao and Dally, 2016).

6

Limitations and Future Work

While the results show strong potential for learning based models in camera motion smoothing, several limitations define the scope of the current system and point to clear avenues for further work.

6.1 Scope and dimensionality

The current implementation operates in two dimensional space, with all training and evaluation sequences representing planar x, y motion only. Real virtual production environments frequently require six degrees of freedom (6 DoF) tracking, including pitch, yaw, roll, and depth translation. Extending to full 3D motion smoothing will require updates to the data pipeline and to the model to represent position and orientation jointly. For orientation, smoothing on the manifold SO(3) with quaternion based representations maintains continuity and avoids angle wrapping, and evaluation should include geodesic angular velocity and jerk to reflect rotational comfort.

6.2 Integration, optimisation, and deployment

The models have been validated on real webcam data in a modular research setup that separates capture, preprocessing, inference, and visualisation. For production use, the next step is to export the trained models to optimised runtimes such as ONNX or TensorRT and to integrate with engines such as Unreal Engine through LiveLink or Open Sound Control. A practical deployment recipe is to stream smoothed coordinates into a camera component while separating capture, inference, and render threads and using a ring buffer to maintain the rolling context.

Further gains are available through model optimisation. Quantisation to int8 and structured pruning can reduce memory bandwidth and often give a few times speedup with limited impact on quality. Where available, lightweight encoder variants and careful reductions in width, then heads, then window length can trade accuracy for speed in a controlled way, with on device latency measured as defined in Section 3.1.

6.3 Dataset diversity and generalisation

Training relied on synthetic motion generated from parametric paths with Gaussian noise. While this ensures clean supervision and repeatability, it does not fully capture the structured, non Gaussian noise found on professional rigs. A practical next step is a small, diverse fine tuning set of real sequences that covers multiple operators, rigs, and environments. Sources can include handheld sessions, stabilised and unstabilised footage, and public tracking datasets. Such adaptation helps models learn subtle physical characteristics and irregular timing that synthetic data does not reproduce well (Carbajal et al., 2025).

6.4 Threats to validity and ethics

Synthetic data may not match device specific noise; this is partially addressed by adding impulses and varying noise amplitudes, though gaps remain (Carbajal et al., 2025). To make comparisons fair on real data, errors are computed relative to a deterministic Kalman reference, so Kalman MSE is zero by construction, and stability and responsiveness are assessed via curvature and latency only. Reported results cover two dimensional translation; rotational and zoom components in full pose tracking can alter behaviour, so conclusions about responsiveness may shift once orientation smoothing is added. Curvature complements MSE by rewarding smooth transitions and can penalise intentional snappy moves; latency captures deployability in live settings.

All claims here are grounded in these quantitative metrics. Future proxies could include jerk based costs inspired by minimum jerk theory and frequency weighted power to separate desirable motion from high frequency jitter (Flash and Hogan, 1985). From an ethical perspective, stabilisation alters recorded paths and can mask operator intent. For editorial transparency, retain raw trajectories alongside model outputs for audit and annotate any smoothing applied during capture.

7

Conclusion

This research compared three approaches to real time camera motion smoothing, a classical Kalman filter, a recurrent LSTM model, and a Transformer model with self attention and positional encoding, evaluated on synthetic and real world motion using established accuracy, smoothness, and latency metrics.

Across all evaluations the Transformer provided the smoothest and most accurate trajectories while remaining responsive, the LSTM improved over the Kalman baseline through learned temporal structure with a modest latency cost, and the Kalman filter remained a reliable minimal compute option for simple jitter reduction.

Beyond the model ranking, the work establishes a reproducible framework for like for like benchmarking under synthetic and real world conditions, making it straightforward to integrate new architectures and compare them fairly.

Future directions include extending to three dimensional and rotational motion, integrating with real time production tools such as Unreal Engine, broadening real data coverage, and exploring lighter weight Transformer variants to further ease deployment.

In summary, the Transformer based smoother offers a practical default for software based real time stabilisation in virtual production, combining the adaptability of learning with the predictability required for live, interactive, and cinematic environments.

Bibliography

Azzarelli, A., N. Anantrasirichai and D. R. Bull (2025). Intelligent cinematography: A review of AI research for cinematographic production. *Artificial Intelligence Review* 58, p. 108.

Bar-Shalom, Y., X.-R. Li and T. Kirubarajan (2001). *Estimation with applications to tracking and navigation: Theory, algorithms and software.* New York: John Wiley & Sons, 2001.

Carbajal, G., P. Vitoria, J. Lezama and P. Musé (2025). Assessing the role of datasets in the generalization of motion deblurring methods to real images. *arXiv.* Available from: `https://arxiv.org/abs/2209.12675` [Accessed 8 August 2025].

Di Bella, L., Y. Lyu, B. Cornelis and A. Munteanu (2025). HybridTrack: A hybrid approach for robust multi-object tracking. *IEEE Robotics and Automation Letters.* In press. Available from: `https://arxiv.org/abs/2501.01275` [Accessed 8 August 2025].

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby (2021). An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. In: *International Conference on Learning Representations (ICLR).* Available from: `https://arxiv.org/abs/2010.11929` [Accessed 8 August 2025]. 2021.

Flash, T. and N. Hogan (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience* 5(7), pp. 1688–1703.

Giuliari, F., I. Hasan, M. Cristani and F. Galasso (2020). Transformer networks for trajectory forecasting. *arXiv.* Available from: `https://arxiv.org/abs/2003.08111` [Accessed 8 August 2025].

Grewal, M. S. and A. P. Andrews (2015a). *Kalman filtering: Theory and practice using MATLAB.* 4th ed. Hoboken, NJ: John Wiley & Sons, 2015a.

Grewal, M. S. and A. P. Andrews (2015b). ‘Probability and expectancy’. *Kalman filtering: Theory and practice using MATLAB*. 4th ed. Hoboken, NJ: John Wiley & Sons, 2015b. Chap. 3.

Grewal, M. S. and A. P. Andrews (2015c). ‘Random processes’. *Kalman filtering: Theory and practice using MATLAB*. 4th ed. Hoboken, NJ: John Wiley & Sons, 2015c. Chap. 4.

Grundmann, M., V. Kwatra and I. Essa (2011). Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 225–232.

Han, S., H. Mao and W. J. Dally (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In: *International Conference on Learning Representations (ICLR)*. Available from: `https://arxiv.org/abs/1510.00149` [Accessed 8 August 2025]. 2016.

Hochreiter, S. and J. Schmidhuber (1997). Long Short-Term Memory. *Neural Computation* 9(8), pp. 1735–1780.

Li, J., S. Wang, L. Chen, Y. Wang, H. Zhou and J. M. Guerrero (2024). Adaptive Kalman filter and self-designed early stopping strategy optimized convolutional neural network for state of energy estimation of lithium-ion battery in complex temperature environment. *Journal of Energy Storage* 83, p. 110750.

Lim, B., S. Ö. Arik, N. Loeff and T. Pfister (2021). Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *International Journal of Forecasting* 37(4), pp. 1748–1764.

Liu, F., M. Gleicher, J. Wang, H. Jin and A. Agarwala (2011). Subspace Video Stabilization. *ACM Transactions on Graphics (SIGGRAPH 2011)* 30(1), p. 4.

Liu, S., P. Tan, L. Yuan, J. Sun and B. Zeng (2016). MeshFlow: Minimum Latency Online Video Stabilization. In: *European Conference on Computer Vision (ECCV 2016)*. 2016, pp. 800–815.

Liu, S., L. Yuan, P. Tan and J. Sun (2013). Bundled Camera Paths for Video Stabilization. *ACM Transactions on Graphics (SIGGRAPH 2013)* 32(4), p. 78.

Press, O., N. A. Smith and M. Lewis (2021). Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. *arXiv*. Available from: `https://arxiv.org/abs/2108.12409` [Accessed 8 August 2025].

Savitzky, A. and M. J. E. Golay (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36(8), pp. 1627–1639.

Su, J., Y. Lu, S. Pan, A. Murtadha, B. Wen and Y. Liu (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv*. Available from: `https://arxiv.org/abs/2104.09864` [Accessed 8 August 2025].

Sundermeyer, M., M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras and R. Triebel (2020). Multi-path learning for object pose estimation across domains. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 13916–13925.

Sundermeyer, M., Z.-C. Marton, M. Durner and R. Triebel (2020). Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection. *International Journal of Computer Vision* 128, pp. 714–729.

Tremblay, J., A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon and S. Birchfield (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *arXiv*. Available from: `https://arxiv.org/abs/1804.06516` [Accessed 8 August 2025].

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin (2017). Attention Is All You Need. In: *Advances in Neural Information Processing Systems (NeurIPS 2017)*. Available from: `https://arxiv.org/abs/1706.03762` [Accessed 8 August 2025]. 2017.

Wang, M., S.-M. Hu, J. Yang, F. Xu and M.-H. Yang (2019). Deep Online Video Stabilization With Multi-Grid Warping. *IEEE Transactions on Image Processing* 28(5), pp. 2283–2292.

Wang, Z., L. Zhang and H. Huang (2018). High quality real-time video stabilization using trajectory smoothing and mesh-based warping. *IEEE Access* 6, pp. 25157–25166.

Welch, G. and G. Bishop (2006). *An introduction to the Kalman filter*. Technical Report TR 95-041. Available from: `https://www.cs.unc.edu/~welch/kalman/` [Accessed 8 August 2025]. Chapel Hill, NC: UNC–Chapel Hill, Department of Computer Science, 2006.

Wu, H., J. Xu, J. Wang and M. Long (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In: *Advances in Neural Information Processing Systems (NeurIPS 2021)*. 2021.

Yang, X., L. Wang and D. Tao (2021). Real-time camera trajectory smoothing with LSTM networks. *Computer Vision and Image Understanding* 207, p. 103211.

Zeng, A., M. Chen, L. Zhang and Q. Xu (2023). Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence* 37(9), pp. 11121–11128.

Zhang, Z., Z. Liu, P. Tan, B. Zeng and S. Liu (2023). Minimum Latency Deep Online Video Stabilization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2023)*. 2023, pp. 23030–23039.

Zhou, H., S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong and W. Zhang (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021)*. Vol. 35. 12. 2021, pp. 11106–11115.

Zhou, T., Z. Ma, Q. Wen, X. Wang, L. Sun and R. Jin (2022). FEDformer: Frequency Enhanced Decomposed Transformer for Long-Term Series Forecasting. In: *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*. 2022, pp. 27268–27286.