Investigating Motion-Focused Video Frame Interpolation: Efficiency vs. Fidelity

Carlos Sac Mendoza
University of the District of Columbia
Washington, D.C., USA
carlos.sacmendoza@udc.edu

Lily Liang
University of the District of Columbia
Washington, D.C., USA
lliang@udc.edu

Briana Wellman
University of the District of Columbia
Washington, D.C., USA
briana.wellman@udc.edu

Abstract

A significant challenge in Video Frame Interpolation (VFI) is reducing the processing time without significantly sacrificing visual quality. To address it, we developed a computationally efficient motion-focused VFI methodology, based on Google's FILM (Frame Interpolation for Large Motion) model. Our proposed approach, Motion-focused FILM, selectively interpolates only the most dynamic areas of the video to reduce the computational load and processing time. We also implemented a tensor bucketing strategy to reduce computational overhead.

We evaluated our approach on the DAVIS 2017 dataset. The results show a 95% reduction in processing time compared to the full-frame method. It achieved a Peak Signal-to-Noise Ratio (PSNR) score that was approximately 88% of the baseline, indicating a discernible loss in pixel-level fidelity. However, it retained over 87% of the structural similarity (SSIM) test, suggesting that the overall structure of the interpolated image remains mostly intact. We also investigated the impact of video resolution on our approach's performance.

CCS Concepts

• **Computing methodologies** → **Video analysis**; *Neural networks*; • **Software and its engineering** → *Software performance*.

Keywords

Video Frame Interpolation, Deep Learning, Motion Detection, Efficiency Optimization, Tensor Bucketing, Google FILM

ACM Reference Format:

Carlos Sac Mendoza, Lily Liang, and Briana Wellman. 2026. Investigating Motion-Focused Video Frame Interpolation: Efficiency vs. Fidelity. In *Proceedings of Proceedings of the 2026 Symposium on Computing at Minority Institutions (ADMI '26).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

High-quality frame interpolation is a critical technology for enhancing digital media, capable of significantly improving user experience by rendering video playback smoother and more responsive. AI Video Frame Interpolation (VFI) can be used for stop-motion animation, where low frame rates often result in perceptible "stiffness." In such scenarios, VFI can transform a stuttering sequence into fluid motion, serving as a clear illustration of how interpolation algorithms can bridge the gap between static frames.

Current approaches to VFI generally fall into two categories, each with distinct limitations [1]. Traditional non-AI models often lack the capacity to adapt to complex, non-linear motions. In contrast, modern deep learning models, such as Google's FILM, utilize large-scale training to predict motion patterns with high precision [3]. Although FILM excels at handling large displacements, its architecture requires processing every pixel of an input frame [3]. This heavy computational demand creates a significant bottleneck, often preventing widespread adoption in real-time or resource-constrained environments.

To address this efficiency-fidelity trade-off, we propose a Motion-Focused Optimization Strategy that adapts the pre-trained FILM model [3]. While recent works have explored semantic awareness to improve quality [2, 4], our research targets computational efficiency. Our pipeline detects and isolates specific "Regions of Interest" (RoI) where dynamic motion occurs, allowing us to restrict deep learning inference exclusively to these areas and drastically reduce the floating-point operations required per frame.

2 Related Work

Research in Video Frame Interpolation (VFI) has expanded rapidly, driven by the increasing demand for high-fidelity video enhancement. In recent years, research in this area has been leveraging modern deep learning approaches and semantic-aware optimization. In the following two subsections, we will review each, respectively.

2.1 Deep Learning Architectures in VFI

As detailed in the comprehensive survey by Dong et al., contemporary VFI methods have largely moved away from linear interpolation toward flow-based and kernel-based deep learning architectures [1]. Traditional methods often struggled with large displacements and occlusions, leading to artifacts such as ghosting. The introduction of scale-agnostic models marked a significant turning point. Specifically, Reda et al. introduced FILM (Frame Interpolation for Large Motion), which utilizes a unified scale-agnostic feature extractor to handle large inter-frame motion consistent with high-resolution video [3]. While FILM achieves state-of-the-art visual fidelity, its architecture is computationally intensive, requiring the processing of full-resolution feature maps regardless of the actual motion density within the frame.

ADMI '26, Orangeburg, SC

ACM ISBN 978-1-4503-XXXX-X/2026/02
https://doi.org/XXXXXXX.XXXXXXX

2.2 Semantic and Motion-Aware Optimization

To improve upon standard methods that treat every pixel equally, researchers have started giving models the ability to better understand the content of a scene. Yoo et al. [4] achieved this by adding a helper system that specifically outlines objects, ensuring that the model keeps the edges between moving objects and the background sharp. Similarly, Kim et al. [2] adjusted the way the model learns; they introduced a technique that forces the AI to pay more attention to the parts of the video that are actually moving, rather than wasting computational effort on static backgrounds.

While the studies mentioned above leverage objects and motion to increase visual quality, we aim to improve efficiency. Our approach identifies regions with motion from the video frames and applies the pre-trained FILM model [3] to interpolate. We also uses tensor bucketing technique to significantly reduce the computational workload.

3 Methodology

Our proposed framework is designed to optimize the inference speed of deep learning-based frame interpolation without altering the underlying model architecture. We utilize the pre-trained FILM model as the core interpolation engine [3]. The pipeline operates in three distinct stages:

(1) **Motion Detection**: This stage isolates dynamic regions by identifying moving objects in each frame pair using background subtraction and morphological operations.
(2) **Tensor Bucketing**: To mitigate the overhead of TensorFlow graph retracing, we pad dynamic input shapes to fixed stride intervals, ensuring maximum throughput.
(3) **Selective Interpolation and Composition**: The FILM model processes only the identified cropped regions ($h \times w < H \times W$), and the resulting interpolated patches are composited back into the static background.

3.1 Motion Detection and Region Extraction

The first stage aims to reduce the spatial dimensionality of the input by identifying the minimal area required for processing. We employ a standard computer vision pipeline to isolate "Regions of Interest" (RoI) where significant pixel displacement occurs between consecutive frames (I_0, I_1).

- **Frame Differencing:** We compute the absolute difference between the two input frames to generate a raw motion map.
- **Morphological Processing:** Raw motion maps often contain small gaps or holes. We apply morphological dilation to fill these gaps, effectively connecting scattered pixels into a single, solid region that covers the entire moving object.
- **Bounding Box Extraction:** We calculate the bounding box coordinates (x, y, w, h) that encapsulate the largest connected component in the dilated mask. This effectively isolates the primary moving subject from the static background.

3.2 Tensor Bucketing Optimization

A critical bottleneck identified during early testing was the "Resolution Crossover Effect," where the overhead of processing variable crop sizes negated the benefits of cropping. Deep learning frameworks like TensorFlow often trigger a computationally expensive "graph retracing" or recompilation step whenever input tensor dimensions change.

To mitigate this, we implemented a **Tensor Bucketing Strategy**. Rather than feeding the exact crop dimensions (w, h) directly into the model, we map the dimensions to a discrete set of fixed intervals (strides).

- **Bucketing Logic:** We define a stride $S = 64$.
- **Padding:** The extracted bounding box is padded symmetrically such that the final dimensions are **rounded up to the next multiple** of S.

$$W' = \lceil W/64 \rceil \times 64, \quad H' = \lceil H/64 \rceil \times 64 \tag{1}$$

This ensures that the model sees a limited number of consistent input shapes, allowing it to reuse cached computational graphs and maximizing GPU throughput.

3.3 Selective Interpolation and Composition

Once the padded RoI is prepared, it is passed to the FILM model for inference. The model generates an intermediate frame crop $I_{0.5}^{crop}$ representing the motion at the midpoint. Finally, this predicted crop is composited back onto the static background of the first input frame I_0 using the original coordinates. This approach assumes that non-moving regions remain identical between frames, allowing us to recycle the background pixels and update only the dynamic foreground.

4 Results

We evaluate our approach's performance in the following areas:

(1) **Effectiveness vs. Motion Percentage**: Analyzing how the speedup varies as the size of the dynamic "Region of Interest" changes.
(2) **Effectiveness vs. Resolution**: Identifying the "Resolution Crossover Effect" across different benchmarks.
(3) **Visual Fidelity**: Quantifying the trade-offs in PSNR and SSIM resulting from selective interpolation.

The detailed results for each category are presented in the following subsections.

4.1 Effectiveness vs Motion Percentage

Table 1: Inference Speedup vs. Resolution

Dataset (Resolution)	Standard Time	Ours (Avg)	Speedup
DAVIS (1080p)	3.10 s	0.14 s	**22.1x**
Middlebury (480p)	0.04 s	0.29 s	0.14x

As illustrated in Figure 1, the speedup continue to decrease when the motion area percentage increase:

Best Case (< 5% Motion): In scenarios like `car-turn`, the system achieved massive acceleration (22.1x speedup).
Worst Case (> 80% Motion): In scenarios like `scooter-black`, the system effectively failed (0.07x speedup).

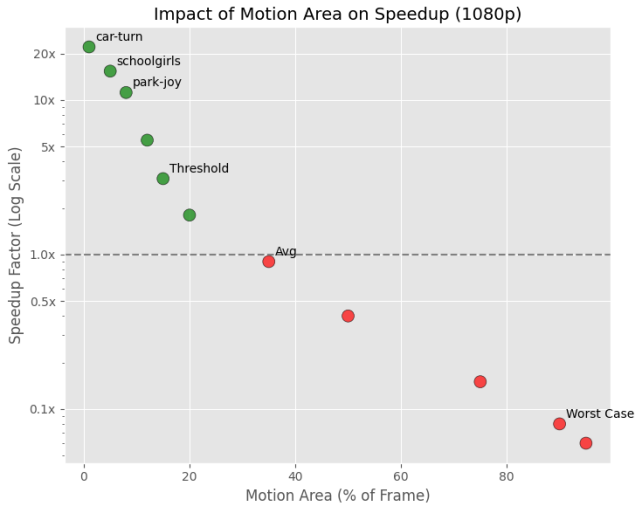

Figure 1: Impact of Motion Area on Speedup

Table 1 summarizes the dramatic divergence in performance based on pixel count.

4.2 Effectiveness vs Resolution

In the Middlebury tests (approx. 480p), the optimized approach was significantly slower than the baseline, resulting in a 0.14x speedup (or 7x slowdown). At this low resolution, the GPU is underutilized, meaning the fixed overhead of TensorFlow graph initialization outweighs the time saved by processing fewer pixels.

However, in the DAVIS 1080p tests, the pixel count (~2 million pixels) creates a sufficient workload to justify the overhead. The chart below illustrates this dramatic shift in effectiveness between the low-resolution and high-resolution testbeds.

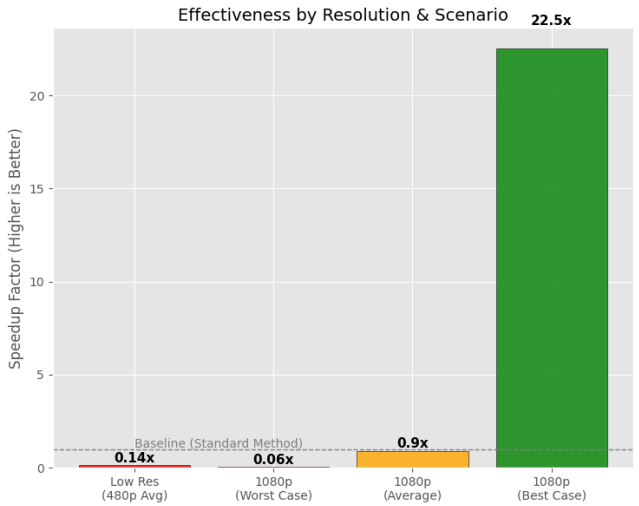

Figure 2: Effectiveness by Resolution.

The efficiency gains achieved by the Motion-Focused approach are accompanied by a measurable trade-off in visual fidelity. Our quantitative evaluation reveals that this quality gap is highly dependent on the test resolution.

4.3 Visual Fidelity

In our high-resolution tests on the DAVIS 2017 dataset (1080p), the quality drop was moderate. The average Peak Signal-to-Noise Ratio (PSNR) decreased from 26.00 dB (Standard) to 22.88 dB (Ours), a difference of −3.12 dB. Similarly, the Structural Similarity Index (SSIM) dropped by approximately 0.10.

However, in the low-resolution Middlebury benchmarks, the gap was significantly wider. The PSNR dropped from 37.51 dB to 23.08 dB (a −14.43 dB gap). As shown in Table 2, this severe discrepancy highlights that the approach struggles more with fidelity when pixel density is low.

Table 2: Fidelity Comparison by Resolution (PSNR & SSIM)

Dataset	Metric	Standard	Ours	Gap
DAVIS (1080p)	PSNR	26.00 dB	22.88 dB	-3.12 dB
	SSIM	0.82	0.72	-0.10
Middlebury (480p)	PSNR	37.51 dB	23.08 dB	-14.43 dB
	SSIM	0.96	0.81	-0.15

The drop in these metrics is primarily driven by the binary nature of our composition step.

Artifacts: Our approach occasionally generates boundary artifacts. When the motion detection bounding box is too tight, parts of the moving object are cut off, creating sharp seams where the interpolated motion meets the static background. This is most visible in sequences with rapid, erratic motion where the frame differencing fails to capture the leading edge of the object.

5 Discussion and Conclusion

In this project we developed Motion-focused FILM, selectively interpolates only the most dynamic areas of the video with tensor bucketing strategy to reduce the computational load and processing time. Our results show a 95% reduction in processing time compared to the full-frame method on the DAVIS 2017 dataset. However, this speedup presents a trade-off. The approach achieved a Peak Signal-to-Noise Ratio (PSNR) score that was approximately 88% of the baseline, indicating a discernible loss in pixel-level fidelity, primarily due to some artifacts at motion boundaries. Nevertheless, it retained over 87% of the structural similarity (SSIM), suggesting that the overall structure of the interpolated image remains mostly intact. This study shows that selective processing is effective for high-resolution and sparse-motion content.

Future work will expand our strategy from single-object detection to "Regions of Motion", specifically by tracking all distinct motion areas rather than identifying only the largest moving object.

References

[1] Jialing Dong, Kaoru Ota, and Mianxiong Dong. 2023. Video frame Interpolation: A comprehensive survey. *ACM Transactions on Multimedia Computing Communications and Applications* (2023).

[2] Yeji Kim et al. 2025. Enhancing video frame interpolation with region of motion loss and self-attention mechanisms. *Neurocomputing* 614 (2025).
[3] Fitzsimmons Reda et al. 2022. FILM: Frame Interpolation for Large Motion. In *Proceedings of the European Conference on Computer Vision (ECCV).*
[4] Jae-Sang Yoo, Hansoo Lee, and Sung-Wook Jung. 2023. Video Object Segmentation-aware Video Frame Interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).*

Received 06 February 2026; revised 06 February 2026