Supplementary Material - Explicit Guidance for Robust Video Frame Interpolation against Discontinuous Motions

Jaehyun Park[1] Nam Ik Cho[1,2]
[1]IPAI, Seoul National University, Korea
[2]Department of Electrical and Computer Engineering, INMC, Seoul National University, Korea
{jaep0805, nicho}@snu.ac.kr

Table 1. Comparison of trainable parameter count.

$\mathcal{F}$	Framework	Parameters (M)
AdaCoF [6]	[7]	2.6
	Ours	0.043
CAIN [1]	[7]	4.7
	Ours	0.043
VFIT [13]	[7]	3.0
	Ours	0.043

1. Additional Experiment Results

1.1. Training Parameters

Our framework enjoys a simple 'plug-and-play' design that only requires the training of auxiliary D-map estimator $\mathcal{E}$ to make existing VFI networks $\mathcal{F}$ robust to discontinuities. This is in comparison to [7] which requires the training of both $\mathcal{E}$ and $\mathcal{F}$. Table 1 show the difference in trainable parameters of the two frameworks when applied to various $\mathcal{F}$. Regardless of the $\mathcal{F}$ used, our framework retains an equal parameter count across all $\mathcal{F}$. Furthermore, compared to [7], our framework employs a lightweight D-map estimator design to drastically reduce the number of trainable parameters and hence, resulting in a shorter training time.

1.2. Computational Complexity of the D-map Estimator

Due to the design of our D-map estimator $\mathcal{E}$, the application of our framework greatly increases the computation overhead. As seen in Table 2, the addition of the D-map estimator generally increases the total computation by around four-folds across all datasets. To reduce the computation cost, we propose using down-scaled inputs for the ECG within the D-map estimator $\mathcal{E}$. The role of the ECG in $\mathcal{E}$ is to provide supervision based on the coherence properties across the four input frames. The coherence map M_c, extracted from ECG, offers explicit guidance to the $\mathcal{E}$ in discerning discontinuous areas. This coherence map primarily determines values based on segmented components in the scene, making it less dependent on pixel-level accuracy. Therefore, adequate guidance can still be achieved at a lower resolution. This is highlighted through the results in Table 2 where down-scaled inputs for the ECG have minor impact on the framework's performance. The framework's performance remain robust even when the inputs are down-scaled to 1/8 its original size for the ECG. On the other hand, reducing the spatial resolution drops the computation cost by large margins. At $\times 8$ down-scaled inputs, the computation cost is reduced by 1/2 of its original total cost, making this a favorable trade-off.

1.3. Quantitative Comparison

We compare against baseline versions of AdaCof [6], CAIN [1], EMA-VFI [16], VFIT-B [13], as well as previous VFI methods such as DVF [17], SuperSlomo [3], Sep-Conv [10], Softsplat [9], ABME [11], RIFE [2], IFR-Net [5], CBMNet [4] and SGM-VFI [8]. For [7] and Ours, we report the scores using VFIT-B for all comparisons below. Experiment results in Table 3 show that VFIT-B enhanced with our framework displays unrivaled results in the GDM dataset, achieving state-of-the-art results. Notably, our method outperforms [7] by a large margin, which has also been trained against discontinuous motions. For datasets concerning continuous motion, our framework is able to retain the strong performance of the VFIT-B baseline in interpolating continuous motion. Unlike [7], freezing the employed VFI network allows us to take advantage of the rich motion prior of high-capacity models like VFIT-B. On the other hand, [7] re-trains the VFI network, resulting in dropped performances across real-world datasets, shown in Table 1 of the main paper. This is further verified through the visualization of inaccurate D-map estimations in our supplementary materials. The experiment results highlight the design benefits of our framework against both continuous and discontinuous motion.

Table 2. Computational Complexity and Performance trade-off through down-scaled inputs for the ECG in the D-map estimator

Vimeo-90K [15]

Scale	PSNR	SSIM	LPIPS	FLOPs (T) $\mathcal{F}$	$\mathcal{E}$	Total
×1	33.472	0.9381	0.0590		0.47	0.63
×1.5	33.442	0.9380	0.0591		0.32	0.48
×2	33.422	0.9380	0.0591	0.16	0.23	0.39
×4	33.453	0.9381	0.0590		0.18	0.33
×8	33.443	0.9380	0.0591		0.16	0.32

UCF101 [14]

Scale	PSNR	SSIM	LPIPS	FLOPs (T) $\mathcal{F}$	$\mathcal{E}$	Total
×1	33.824	0.9408	0.0404		0.33	0.44
×1.5	33.825	0.9408	0.0404		0.22	0.34
×2	33.819	0.9408	0.0404	0.11	0.16	0.27
×4	33.822	0.9408	0.0405		0.12	0.23
×8	33.821	0.9407	0.0405		0.11	0.22

DAVIS [12]

Scale	PSNR	SSIM	LPIPS	FLOPs (T) $\mathcal{F}$	$\mathcal{E}$	Total
×1	26.249	0.8173	0.2044		8.63	11.50
×1.5	26.240	0.8172	0.2046		5.49	8.35
×2	26.236	0.8173	0.2046	2.86	4.33	7.20
×4	26.235	0.8167	0.2048		3.28	6.15
×8	26.202	0.8156	0.2049		3.01	5.88

GDM [7]

Scale	PSNR	SSIM	LPIPS	FLOPs (T) $\mathcal{F}$	$\mathcal{E}$	Total
×1	33.342	0.9486	0.0522		2.88	3.83
×1.5	33.334	0.9485	0.0523		1.83	2.78
×2	33.353	0.9485	0.0522	0.95	1.45	2.40
×4	33.309	0.9485	0.0523		1.10	2.05
×8	33.322	0.9486	0.0522		1.01	1.96

Table 3. Comparison against previous works on test datasets [7, 12, 14, 15]. Labels (2) and (4) refer to the input frame number. Red and Blue indicates the best and runner-up, respectively. VFIT-B(4) + *Ours* shows the state-of-the-art performance, especially on GDM [7].

	Vimeo-90K [15]			**UCF101** [14]			**DAVIS** [12]			**GDM** [7]		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
DVF (2) [17]	32.792	0.9359	0.0395	32.333	0.9397	0.0340	24.087	0.7852	0.1588	28.709	0.9118	0.0753
SuperSlomo (2) [3]	30.812	0.9291	0.0482	28.500	0.9228	0.0564	26.259	0.8303	0.1206	27.651	0.8911	0.1117
SepConv (2) [10]	33.729	0.9454	0.0335	33.075	0.9419	0.0333	26.550	0.8376	0.1478	29.696	0.9082	0.1037
AdaCoF (2) [6]	34.103	0.9459	0.0427	33.320	0.9438	0.0353	26.791	0.8353	0.1643	29.980	0.9227	0.0803
Softsplat (2) [9]	33.723	0.9452	0.0336	33.112	0.9419	0.0332	26.542	0.8376	0.1479	29.667	0.9086	0.1039
CAIN (2) [1]	34.699	0.9514	0.0421	33.306	0.9444	0.0373	27.449	0.8511	0.1855	30.238	0.9284	0.0807
ABME (2) [11]	35.846	0.9584	0.0309	33.542	0.9458	0.0383	27.661	0.8601	0.1320	29.472	0.9209	0.0958
RIFE (2) [2]	34.048	0.9449	0.0233	33.184	0.9412	0.0284	27.246	0.8471	0.0925	30.085	0.9088	0.0801
IFRNet (2) [5]	35.837	0.9597	0.0274	33.451	0.9450	0.0330	27.467	0.8596	0.1261	30.239	0.9277	0.0706
EMA-VFI (2) [16]	36.042	0.9725	0.0312	33.814	0.9456	0.0317	26.477	0.8893	0.1465	29.711	0.9427	0.0892
CBMNet (2) [4]	36.127	0.9634	0.0329	33.801	0.9482	0.0315	27.828	0.8409	0.1356	30.616	0.9254	0.0754
SGM-VFI (2) [8]	35.387	0.9623	0.0338	33.657	0.9466	0.0321	26.732	0.8943	0.1395	29.564	0.9366	0.0731
VFIT-B(4) [13]	36.743	0.9638	0.0318	33.769	0.9472	0.0363	28.090	0.8640	0.1442	30.019	0.9280	0.0736
VFIT-B(4) + [7]	36.671	0.9631	0.0324	33.823	0.9475	0.0370	28.056	0.8625	0.1507	30.921	0.9371	0.0645
VFIT-B(4) + *Ours*	36.731	0.9641	0.0312	33.858	0.9496	0.0420	28.134	0.8645	0.1443	32.969	0.9480	0.0795

1.4. Additional D-map Comparisons

Because the GDM dataset is mostly comprised of discontinuities, it is hard to verify the accuracy of the extracted D-map. Hence, we also examine D-maps in Figure 1 from the real-world test datasets to verify its effectiveness in the large presence of continuous motion. Contrary to the D-maps in [7], the D-maps extracted by our proposed framework are precise, activated only in regions of discontinuity. Given the camera movements, the background is not static in these real-world datasets. The D-map of our proposed framework is only activated for discontinuities, compared to [7] where the D-maps are rather naive, incorrectly predicting many continuous regions as discontinuities. This is also directly reflected onto the interpolation results, where our proposed framework shows better interpolation results with less artifacts.

1.5. Continuous motion

Our framework is designed to enhance the robustness of existing VFI models without the need for re-training. Instead, a frozen pre-trained model is employed, retaining the most optimal performance for continuous motions. This advantage of this design is can also be seen in Figure 1. Compared to [7], which re-trains the VFI model, our framework better retains the shape and structure of objects under extreme motions (1st and 2nd row) and also better preserves the texture of continuous regions, producing higher-

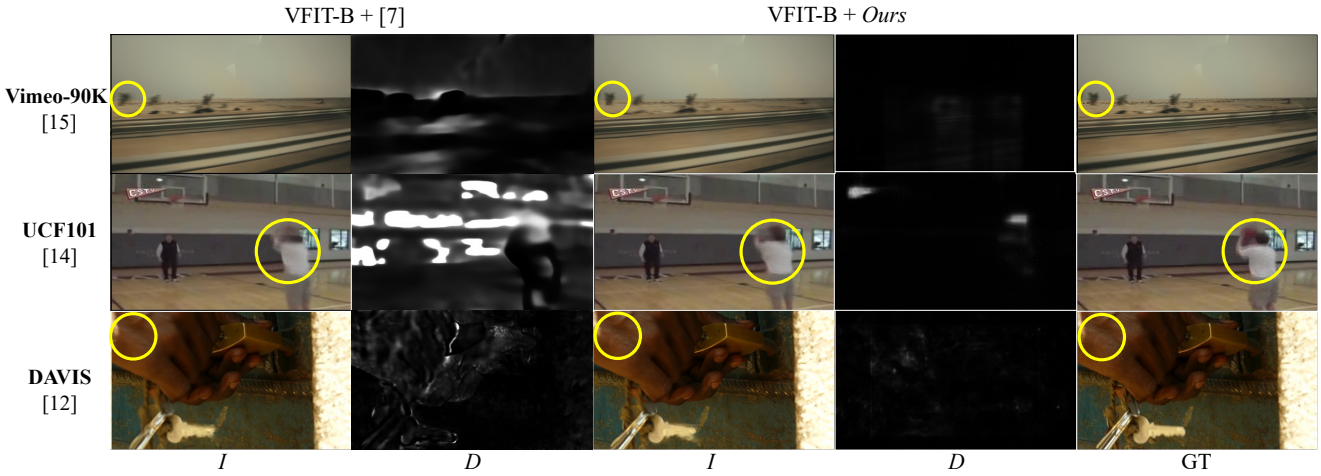

Figure 1. Visual comparison of continuous motion on real-world test dataset [12, 14, 15]

Table 4. Comparison on various augmentation methods. - indicate no augmentation used

Augmentation	Vimeo-90K [15]			UCF101 [14]			DAVIS [12]			GDM [7]		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
-	33.556	0.9385	0.0592	32.879	0.9321	0.0518	26.261	0.8171	0.2088	29.073	0.9206	0.1200
FTM	33.338	0.9377	0.0593	32.797	0.9315	0.0516	26.039	0.8077	0.2074	30.884	0.9290	0.1015
FTM+ (*Ours*)	33.472	0.9381	0.0590	33.824	0.9408	0.0404	26.249	0.8173	0.2044	33.342	0.9486	0.0522

quality interpolation results. This is further consolidated in the extracted D-maps where [7] correctly predicts discontinuity (1st row, yellow circle) yet shows incorrect interpolation, indicating a reduction in performance against continuous motion due to re-training. Meanwhile, less accurate D has also led to discontinuous interpolation of continuous regions, resulting in blurs (2nd row) and over-smoothed textures (1st row).

1.6. FTM and FTM+ comparison

We also evaluate the benefits of our expanded augmentation method FTM+ compared to the original FTM augmentation method. Table 4 show comparison of the AdaCoF [6] model trained with our framework using no augmentations, FTM and FTM+. From the baseline framework which uses no augmentation, the use of FTM augmentation definitely enhances the model's robustness to discontinuities at the slight cost of performance on real-world datasets [12, 14, 15]. Further expanding on the original FTM, we add transparency, fill configurations as well as irregular and scene change discontinuities to better replicate various discontinuities. As a result, FTM+ outperforms FTM on the GDM dataset [7] by a large margin, showing strong robustness to synthetic discontinuities. Moreover, we attribute the improvement across real-world datasets to the added scene change augmentation as previous augmentation methods inaccurately predicted intermediate frames for scene changes by a large degree.

2. Limitations

Although our proposed framework established a big step forward in addressing discontinuities in VFI, it also has its limitations. First, our framework has a strong dependence on the performance of the employed VFI model. Our proposed guidance techniques are established based on optimal interpolation by the employed VFI model on continuous motion. Therefore, the performance of the framework largely varies depending on the choice and performance of the employed model. Furthermore, the D-map approach of blending I_1 and I_c to synthesize I may be detrimental when the spatial information between I_1 and I_c has not been sufficiently leveraged by the D-map estimator $\mathcal{E}$. To address these limitations, a possible next step in research could be designing and training an end-to-end model without an explicit D-map from the ground-up to ensure problems do not arise from the interconnection of separate modules.

References

[1] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020. 1, 2

[2] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pages 624–642. Springer, 2022. 1, 2

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018. 1, 2

[4] Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18032–18042, 2023. 1, 2

[5] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022. 1, 2

[6] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 1, 2, 3

[7] Sangjin Lee, Hyeongmin Lee, Chajin Shin, Hanbin Son, and Sangyoun Lee. Exploring discontinuity for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9791–9800, 2023. 1, 2, 3

[8] Chunxu Liu, Guozhen Zhang, Rui Zhao, and Limin Wang. Sparse global matching for video frame interpolation with large motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19125–19134, 2024. 1, 2

[9] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 1, 2

[10] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*, pages 261–270, 2017. 1, 2

[11] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14539–14548, 2021. 1, 2

[12] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018. 2, 3

[13] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17482–17491, 2022. 1, 2

[14] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 3

[15] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019. 2, 3

[16] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023. 1, 2

[17] Zhifeng Zhang, Li Chen, Rong Xie, and Li Song. Frame interpolation via refined deep voxel flow. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1473–1477. IEEE, 2018. 1, 2