Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation

Kun Zhou[1,2] Wenbo Li[3] Xiaoguang Han[1] Jiangbo Lu[2*]
[1]SSE, CUHK-Shenzhen, [2]SmartMore Corporation [3]CUHK
kunzhou@link.cuhk.edu.cn, wenboli@cse.cuhk.edu.hk
hanxiaoguang@cuhk.edu.cn,jiangbo.lu@gmail.com

Abstract

For video frame interpolation (VFI), existing deep-learning-based approaches strongly rely on the ground-truth (GT) intermediate frames, which sometimes ignore the non-unique nature of motion judging from the given adjacent frames. As a result, these methods tend to produce averaged solutions that are not clear enough. To alleviate this issue, we propose to relax the requirement of reconstructing an intermediate frame as close to the GT as possible. Towards this end, we develop a texture consistency loss (TCL) upon the assumption that the interpolated content should maintain similar structures with their counterparts in the given frames. Predictions satisfying this constraint are encouraged, though they may differ from the predefined GT. Without the bells and whistles, our plug-and-play TCL is capable of improving the performance of existing VFI frameworks consistently. On the other hand, previous methods usually adopt the cost volume or correlation map to achieve more accurate image or feature warping. However, the $O(N^2)$ (N refers to the pixel count) computational complexity makes it infeasible for high-resolution cases. In this work, we design a simple, efficient $O(N)$ yet powerful guided cross-scale pyramid alignment (GCSPA) module, where multi-scale information is highly exploited. Extensive experiments justify the efficiency and effectiveness of the proposed strategy.

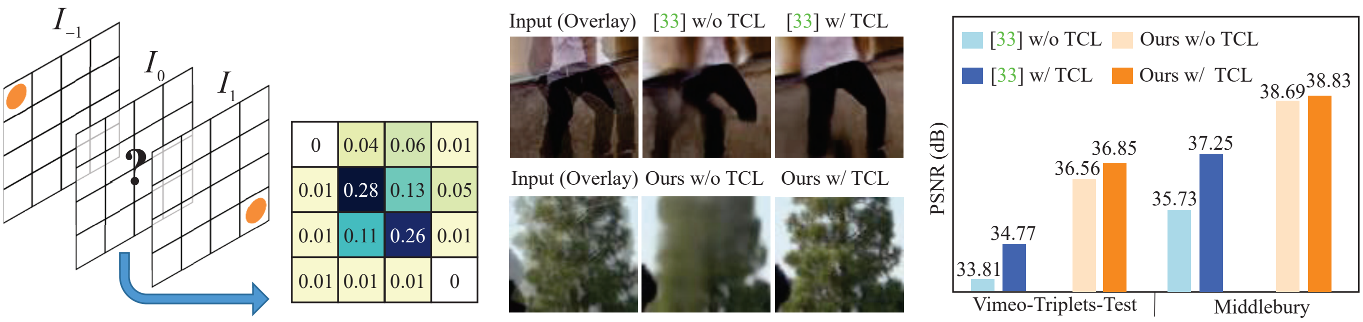
1. Introduction

Video frame interpolation (VFI) plays a critical role in computer vision with numerous applications, such as video editing and novel view synthesis. Unlike other vision tasks that heavily rely on human annotations, VFI benefits from the abundant off-the-shelf videos to generate high-quality training data. The recent years have witnessed the rapid development of VFI empowered by the success of deep neural networks. The popular approaches can be roughly divided into two categories: 1) optical-flow-based methods [1, 8, 15–17, 20, 26–28, 30, 31, 39, 44–46, 49, 51, 53] and 2) kernel-regression -based algorithms [4–6, 22, 32, 33, 37].

The optical-flow-based methods typically warp the images/features based on a linear or quadratic motion model and then complete the interpolation by fusing the warped results. Nevertheless, it is not flexible enough to model the real-world motion under the linear or quadratic assumption, especially for cases with long-range correspondence or complex motion. Besides, occlusion reasoning is a challenging problem for pixel-wise optical flow estimation. Without the prerequisites above, the kernel-based methods handle the reasoning and aggregation in an implicit way, which adaptively aggregate neighboring pixels from the images/features to generate the target pixel. However, this line stands the chance of failing to tackle the high-resolution frame interpolation or large motion due to the limited receptive field. Thereafter, deformable convolutional networks, a variant of kernel-based methods, are adopted to aggregate the long-term correspondence [5, 7, 22], achieving better performance. Despite many attempts, some challenging issues remain unresolved.

First, the deep-learning-based VFI works focus on learning the predefined ground truth (GT) and ignore the inherent motion diversity across a sequence of frames. As illustrated in Fig. 1 (a), given the positions of a ball in frames I_{-1} and I_1, we conduct a user study of choosing its most possible position in the intermediate frame I_0. The obtained probability distribution map clearly clarifies the phenomenon of motion ambiguity in VFI. Without considering this point, existing methods that adopt the pixel-wise L1 or L2 supervision possibly generate blurry results, as shown in Fig. 1 (b). To resolve this problem, we propose a novel texture consistency loss (TCL) that relaxes the rigid supervision of GT while ensuring texture consistency across adjacent frames. Specifically, for an estimated patch, apart from the predefined GT, we look for another texture-matched patch from the input frames as a pseudo label to jointly optimize the network. In this case, predictions satisfying the texture consistency are also encouraged. From the visualization com-

[*] Corresponding author



(a) (b) (c)

Figure 1. Analysis of motion ambiguity in VFI. (a) User study of querying the location of a ball in the intermediate frame I_0 with the observed two input frames $\{I_{-1}, I_1\}$. The results are visualized in a probability distribution map. (b) Visual comparison between SepConv [33] and our method with/without the proposed texture consistency loss (TCL). (c) Quantitative evaluation of the two methods with/without TCL loss on Vimeo-Triplets [47] and Middlebury [2] benchmarks.

parison of SepConv [33] and our model with/without TCL [1] in Fig. 1 (b), we observe that the proposed TCL leads to clearer results. Besides, as shown in Fig. 1 (c), it is seen that our TCL brings about considerable PSNR improvement on Vimeo-Triplets [47] and Middlebury [2] benchmarks for both two methods. More visual examples are available in our supplementary materials.

Second, the cross-scale aggregation during alignment is not fully exploited in VFI. For example, PDWN [5] conducts an image-level warping using the gradually refined offsets. However, the single-level alignment may not take full advantage of the cross-scale information, which has been proven useful in many low-level tasks [23, 25, 52]. To address this issue, some recent works [5, 13, 31] have considered multi-scale representations for VFI. Feflow [31] adopts a PCD alignment proposed by EDVR [42] that performs a coarse-to-fine aggregation for long-range motion estimation. Specifically, the fusion of image features is conducted at two adjacent levels, without considering distant cross-scale aggregation. In this work, we propose a novel guided cross-scale pyramid alignment (GCSPA) module, which performs bidirectional temporal alignment from low-resolution stages to higher ones. In each step, the previously aligned low-scale features are regarded as a guidance for the current-level warping. To aggregate the multi-scale information, we design an efficient fusion strategy rather than building the time-consuming cost volume or correlation map. Extensive quantitative and qualitative experiments verify the effectiveness and efficiency of the proposed method.

In a nutshell, our contributions are three-fold:

- **Texture consistency loss**: Inspired by the motion ambiguity in VFI, we design a novel texture consistency loss to allow the diversity of interpolated content, producing clearer results.

- **Guided cross-scale pyramid alignment**: The proposed alignment strategy utilizes the multi-scale information to conduct a more accurate and robust motion compensation while requiring few computational resources.

- **State-of-the-art performance**: The extensive experiments including frame interpolation and extrapolation have demonstrated the superior performance of the proposed algorithm.

2. Related Works

Optical-flow-based methods. A large group of methods utilize optical flow to build pixel-wise correspondences, thereafter, they warp the given neighboring frames to the target frame. For example, TOFlow [47] designs a task-oriented optical flow module and achieves favorable results compared with approaches using off-the-shelf optical flow. In [30], Niklaus and Liu present a context-aware frame interpolation approach by introducing additional warped deep features to provide rich contextual information. Huang *et al.* [15] devise a lightweight sub-module named IFNet to predict the optical flow and train it in a supervised way. Choi *et al.* [9] propose a tridirectional motion estimation method to obtain more accurate optical flow fields. To resolve the conflict of mapping multiple pixels to the same target location in the forward mapping, Niklaus *et al.* [31] develop a differential softmax splatting method achieving a new state of the art. However, these optical flow-based methods generally have a poor performance when facing some challenging cases, such as large occlusion and complex motion.

Kernel-regression-based methods. In addition to optical-flow-based methods above, learning adaptive gathering kernels [5, 7, 22, 24, 32, 33, 36] has also received intensive attention. Niklaus *et al.* [32] regard the video frame inter-

[1] The four models are trained on Vimeo-Triplets [47] dataset.

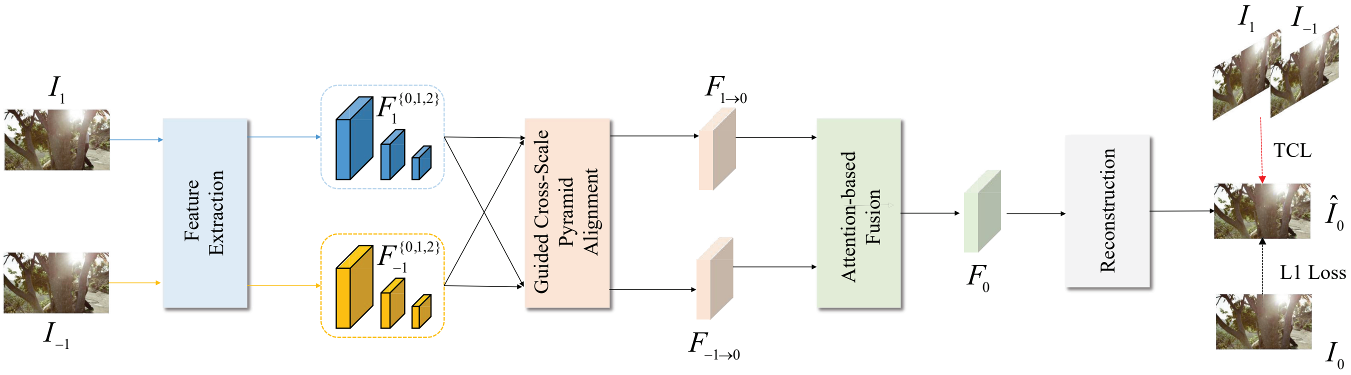


Figure 2. Overview of the proposed VFI architecture. There are four components including a feature extraction module, a guided cross-scale pyramid alignment module, an attention-based fusion module, and a reconstruction module. In addition to the L1 loss for supervision, we propose a texture consistency loss (TCL) to encourage the diversity of objects' motion.

polation as a local convolution over the input frames. A U-Net is designed to regress a pair of kernels that are applied on the input frames to handle the alignments and occlusions simultaneously. To reduce the model parameters while maintaining a comparable receptive field, Niklaus *et al.* [33] propose another separable convolution network by combining two 1D kernels into a 2D adaptive kernel. However, both of these methods obtain limited performance when dealing with large displacements due to the restricted kernel size. Recently, deformable convolution networks (DCN) [7, 22, 55] have shown a great success in the field of video frame interpolation. In PDWN [5], the authors design a pyramid deformable network to warp the contents of input frames to the target frame. While these kernel-based VFI approaches show high flexibility and good performance, they neglect the essential cross-scale information from input frames. In this work, we devise a guided cross-scale pyramid alignment to fuse multiple features in different resolutions, achieving better performance.

Temporal consistency. Some previous studies [12, 21, 50] have exploited temporal consistency for deep-learning models. Lai *et al.* [21] present a short- and long-term temporal loss to enforce the model to learn consistent results over time. Recently, Zhang *et al.* [50] conduct extensive experiments to study spatial-temporal tradeoff for video super-resolution. Dwibedi *et al.* [12] utilize a temporal cycle-consistency loss for self-supervised representation learning. In this work, we propose a novel temporal supervision which improves the quality of frame interpolation by considering adjacent frames. Following [50], we manually search for a balancing factor to achieve appropriate spatial-temporal tradeoff.

3. Methodology

In this section, we first give an overview of the proposed algorithm for video frame interpolation (VFI) in Sec. 3.1. In Sec. 3.2, we explain our texture consistency loss for supervision. At last, we elaborate on the cross-scale pyramid alignment and adaptive fusion in Sec. 3.3.

3.1. Overview

Frame interpolation aims at synthesizing an intermediate frame (e.g., I_0) in the middle of two adjacent frames (e.g., I_{-1} and I_1). As illustrated in Fig. 2, our framework completes the interpolation in a four-step process. First, we obtain the feature pyramids $F_{-1}^{\{0,1,2\}}$ and $F_1^{\{0,1,2\}}$ of frames I_{-1} and I_1 using a feature extraction module. After that, the extracted features are passed through a cross-scale pyramid alignment module to perform a bidirectional alignment towards the middle point in time. Then, we develop an attention-based fusion module to fuse aligned features $F_{-1\to0}$ and $F_{1\to0}$, resulting in F_0. Finally, a sequence of residual blocks are applied on F_0 to synthesize the intermediate frame $\hat{I}_0$.

The existing methods usually strongly penalize the predicted frame $\hat{I}_0$ when it does not exactly match the predefined ground truth (GT) I_0. However, due to the non-uniqueness of movement between I_{-1} and I_1, there may exist many plausible solutions in terms of I_0. Relaxing the rigid requirement of synthesizing the intermediate frame as close as possible to GT I_0, we allow the prediction to be supervised by not only the GT but also the corresponding patterns in I_{-1} and I_1. In this case, our learning target is formulated as

$$\hat{I}_0 = \arg\min_{\hat{I}_0}(\ L_1(\hat{I}_0, I_0) + \alpha L_p(\hat{I}_0, I_{-1}, I_1)), \quad (1)$$

where $L_1(\hat{I}_0, I_0)$ is the commonly adopted data term and $L_p(\hat{I}_0, I_{-1}, I_1)$ is the proposed texture consistency loss detailed in Sec. 3.2. The scaling parameter α is to balance the importance of the two items.

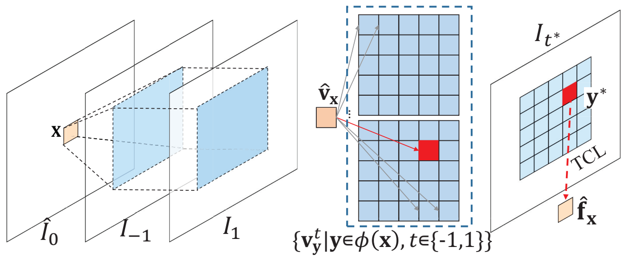

x
$\hat{I}_0$
I_{-1}
I_1
$\hat{\mathbf{v}}_{\mathbf{x}}$
$\{\mathbf{v}_{\mathbf{y}}^t|\mathbf{y}\in\phi(\mathbf{x}), t\in\{-1,1\}\}$
I_{t^*}
$\mathbf{y}^*$
TCL
$\hat{\mathbf{f}}_{\mathbf{x}}$

Figure 3. Overview of our texture consistency loss (TCL). The best matched $\mathbf{f}_{\mathbf{y}^*}^{t^*}$ is served as a pseudo label for training.

3.2. Texture Consistency Loss

The proposed texture consistency loss is illustrated in Fig. 3. For the patch $\hat{\mathbf{f}}_{\mathbf{x}}$ centrally located at position $\mathbf{x}$ on the predicted frame $\hat{I}_0$, we first seek for its best matching $\mathbf{f}_{\mathbf{y}^*}^{t^*}$ from the input frames $\{I_{-1}, I_1\}$, where $\mathbf{y}^*$ and t^* are obtained from

$$\mathbf{y}^*, t^* = \mathop{\arg\min}_{\mathbf{y},t} \; L2(\hat{\mathbf{f}}_{\mathbf{x}}, \mathbf{f}_{\mathbf{y}}^t), \tag{2}$$

where $\mathbf{y}^*$ and $t^* \in \{-1, 1\}$ refer to the optimal position and the optimal frame index, respectively. Then $\mathbf{f}_{\mathbf{y}^*}^{t^*}$ is adopted as an additional pseudo label for estimation $\hat{\mathbf{f}}_{\mathbf{x}}$.

In our implementation, to avoid the interference of illumination in the RGB space across frames, we first apply a census transform [48] to the query and the matching candidates before matching:

$$\mathbf{v}_{\mathbf{x}}(\mathbf{x}+\mathbf{x}_n) = \begin{cases} 0, & \mathbf{f}_{\mathbf{x}}(\mathbf{x}) > \mathbf{f}_{\mathbf{x}}(\mathbf{x}+\mathbf{x}_n) \\ 1, & \mathbf{f}_{\mathbf{x}}(\mathbf{x}) \le \mathbf{f}_{\mathbf{x}}(\mathbf{x}+\mathbf{x}_n) \end{cases}, \mathbf{x}_n \in \mathcal{R}, \tag{3}$$

where $\mathbf{f}_{\mathbf{x}}(\mathbf{x})$ is the pixel value at centeral position $\mathbf{x}$ and the patch field $\mathbf{x}_n$ is defined as

$$\mathcal{R} = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\}. \tag{4}$$

To accelerate the matching process and maintain a reasonable receptive field, we further define the searching area as

$$\phi(\mathbf{x}) = \{\mathbf{y} | \, |\mathbf{y}-\mathbf{x}| \le \mathbf{d}\}, \tag{5}$$

where $\mathbf{y}$ is the position of patch candidates and $\mathbf{d}$ indicates the maximum displacement. In this case, the matching process is represented as

$$\mathbf{y}^*, t^* = \mathop{\arg\min}_{\mathbf{y}\in\phi(\mathbf{x}), t\in\{-1,1\}} \; L2(\hat{\mathbf{v}}_{\mathbf{x}}, \mathbf{v}_{\mathbf{y}}^t), \tag{6}$$

where $\hat{\mathbf{v}}_{\mathbf{x}}$ and $\mathbf{v}_{\mathbf{y}}^t$ are the representations of patches $\hat{\mathbf{f}}_{\mathbf{x}}$ and $\mathbf{f}_{\mathbf{y}}^t$ after census transform. Noticing that the operation of census transform is non-differentiable, our TCL is performed on the original RGB space as

$$L_p(\hat{I}_0, I_{-1}, I_1)(\mathbf{x}) = L1(\hat{\mathbf{f}}_{\mathbf{x}}, \mathbf{f}_{\mathbf{y}^*}^{t^*}). \tag{7}$$

3.3. Guided Cross-Scale Pyramid Alignment

As aforementioned in Sec. 1, most VFI methods utilize the optical flow to perform a two-step synthesis, image-level alignment and deep-learning-based interpolation. However, these approaches face challenges in handling occluded or textureless areas. Consequently, the inaccurate alignment may degrade the performance of the latter processing phases. By contrast, kernel-based works formulate the interpolation as an adaptive convolution over input frames, which typically use a deep network to regress a pair of pixel-wise kernels and apply them on the input frames. However, this single-scale aggregation at the image level may not make full use of information of input frames. To cope with this problem, some approaches have exploited multi-scale aggregation strategies by building dense correlation maps. Nevertheless, the computational complexity increases dramatically with the growth of image resolution.

In this work, we develop a guided cross-scale pyramid alignment (GCSPA) at the feature level aided by deformable convolution networks [11, 56]. Compared with the previous multi-scale aggregation strategies, GCSPA has the following advantages: (1) the previous aligned low-resolution results are regarded as a guidance for the alignment of higher-resolution features, which ensures more accurate warping; (2) aggregating cross-scale information is beneficial to restoring more details; (3) without constructing a cost volume or correlation map, our GCSPA is more computationally efficient.

In detail, the feature pyramids $F_{-1}^{\{0,1,2\}}$ and $F_1^{\{0,1,2\}}$ of frames I_{-1} and I_1 are aligned in a bidirectional way. Taking the direction of $I_{-1} \rightarrow I_0$ for example, we gradually align $F_{-1}^{\{0,1,2\}}$ from low resolution (i.e., F_{-1}^2) to high resolution (i.e., F_{-1}^0), as illustrated in Fig. 4. At first, referring to the other endpoint F_1^2, the alignment of $F_{-1}^2$is conducted to handle the large motion as

$$F_{-1\rightarrow 0}^2 = Align(F_{-1}^2, F_1^2). \tag{8}$$

Then, this result servers as a guidance for next higher-resolution alignment. To this end, we propose a guided cross-scale fusion module ("GCSF" in Fig. 4a) to aggregate cross-scale information from $F_{-1\rightarrow 0}^2$ and F_{-1}^1. In detail, we first bilinearly upsample $F_{-1\rightarrow 0}^2$ by a factor of 2 to obtain $F_{-1\rightarrow 0}^{2,\uparrow 2}$, and then calculate the window-based similarity maps between $F_{-1\rightarrow 0}^{2,\uparrow 2}$ and F_{-1}^1, yielding $C_{-1}^{1,2}$. The fusion is finally carried out as

$$\tilde{F}_{-1\rightarrow 0}^1 = Fuse(F_{-1\rightarrow 0}^{2,\uparrow 2}, F_{-1}^1, C_{-1}^{1,2}), \tag{9}$$

where $Fuse$ is implemented by a single deformable convolution layer. Later on, we feed the fusion $\tilde{F}_{-1\rightarrow 0}^1$ and F_1^1 at the other endpoint into the alignment block and obtain the aligned result:

$$F_{-1\rightarrow 0}^1 = Align(\tilde{F}_{-1\rightarrow 0}^1, F_1^1). \tag{10}$$

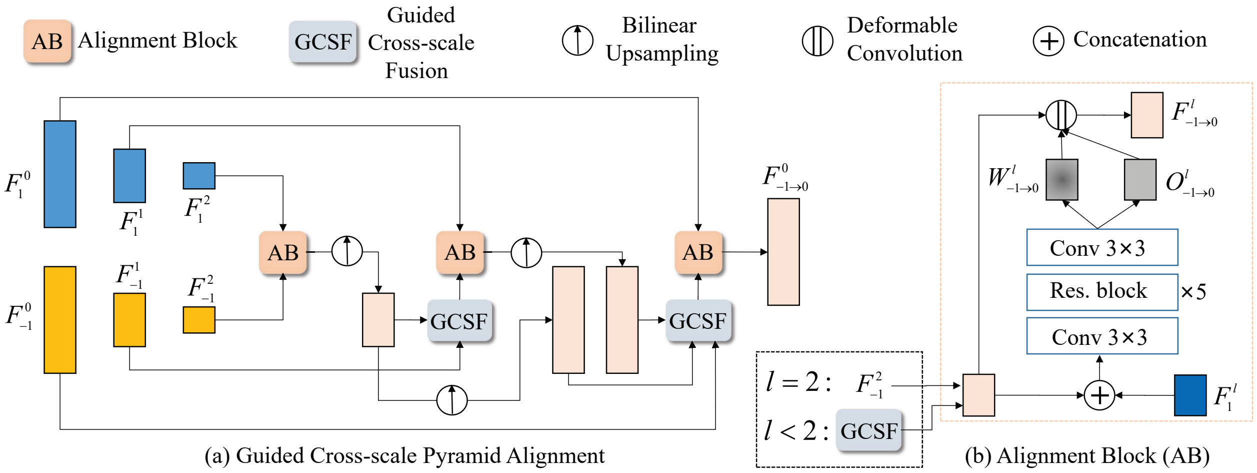


Figure 4. The framework of the guided cross-scale pyramid alignment (GCSPA) module. (a) The 3-level pyramid alignment from I_{-1} to I_0. (b) The detailed structure of the alignment block at the l-th level. The source feature (in light yellow) and the other endpoint feature F^l_1 (in blue) are fed through the alignment block to generate an aligned feature $F^l_{-1\to0}$. The source feature is F^2_{-1} when $l=2$, while representing the fused result of GCSF for other cases. More details can be found in Section 3.3.

Following the same pipeline, we perform the alignment at the highest resolution level to handle subtle motion. Specifically, the cross-scale fusion module takes three-level inputs as

$$\tilde{F}^0_{-1\to0} = Fuse(F^{2,\uparrow4}_{-1\to0}, F^{1,\uparrow2}_{-1\to0}, F^0_{-1}, C^{0,2}_{-1}, C^{0,1}_{-1}). \quad (11)$$

It is noted that the alignment of $I_1 \to I_0$ is completed symmetrically.

The alignment block is zoomed up in Fig. 4b. In terms of the l-th level alignment for frame I_{-1}, the block first concatenates the fused cross-scale feature $\tilde{F}^l_{-1\to0}$ and F^l_1, and conducts a 3×3 convolution. Five sequential residual blocks and another convolution are used to predict a weight map $W^l_{-1\to0}$ and an offset map $O^l_{-1\to0}$. Finally, the aligned feature $F^l_{-1\to0}(\mathbf{x})$ at position $\mathbf{x}$ is calculated by

$$F^l_{-1\to0}(\mathbf{x}) = \sum_i \tilde{F}^l_{-1\to0}(\mathbf{x}+O^l_{-1\to0,i}(\mathbf{x})) * W^l_{-1\to0,i}(\mathbf{x}), \quad (12)$$

where the subscript i means the i-th element in the receptive field of convolution.

3.4. Attention-Based Fusion

After the bidirectional alignment, we obtain a pair of aligned features $F^0_{-1\to0}$ and $F^0_{1\to0}$. In order to determine whether the information is useful or not in a spatially variant way, we employ an attention mechanism to aggregate these two features. First, the attention map is calculated by a convolution followed by a sigmoid operation as

$$M = Sigmoid(Conv(F^0_{-1\to0}, F^0_{1\to0})). \quad (13)$$

Then, the final aggregated result F_0 is obtained by

$$F_0 = M * F^0_{-1\to0} + (1-M) * F^0_{1\to0}. \quad (14)$$

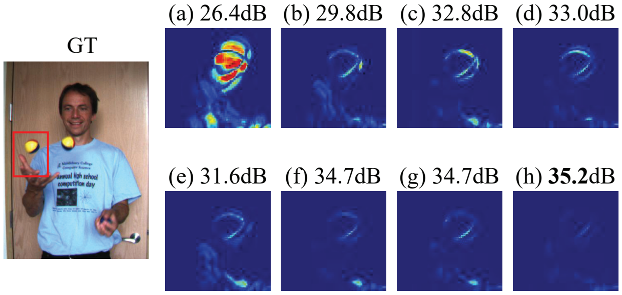


Figure 5. Visualized results for VFI on Middlebury [2]. Visualization of error maps for different video frame interpolation methods on Middlebury. The left figure is the ground truth and the right figure (a-h) represent inputs(overlay), SepConv [33], CtxSyn [30], RIFE-L [15], SepConv++ [34], Softmax-Splatting [31], VFI-Former [28] and ours. The best results are highlighted in **bold**.

4. Experiments

4.1. Implementation Details

All experiments are conducted on the NVIDIA GeForce RTX 2080Ti GPUs. We use two adjacent frames to interpolate the middle frame. An Adam optimizer is adopted and the learning rate decays from 5×10^{-4} to 0 by a cosine annealing strategy. We set the batch size to 64. The training lasts 600K iterations, during which we adopt random 64×64 cropping, vertical or horizontal flipping, and 90°

Method	training	# Parameters	Runtime	Vimeo-Triplets-Test		Middlebury		UCF101	
	dataset	(Million)	(ms)	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SepConv [33]	proprietary	21.6	51	33.79	0.970	35.73	0.959	34.78	0.967
SoftSplat [31]	Vimeo-Triplets-Train	7.7	135	36.10	0.980	38.42	0.971	35.39	0.970
DAIN [3]	Vimeo-Triplets-Train	24.0	130	34.71	0.976	36.70	0.965	35.00	0.968
CAIN [10]	Vimeo-Triplets-Train	42.8	38	34.65	0.973	35.11	0.974	34.98	0.969
EDSC [7]	Vimeo-Triplets-Train	8.9	46	34.84	0.975	36.80	0.983	35.13	0.968
PWDN [5]	Vimeo-Triplets-Train	7.8	-	35.44	-	37.20	0.967	35.00	-
FeFlow [13]	Vimeo-Triplets-Train	133.6	-	35.28	-	36.61	0.965	35.08	0.957
MEMC-Net [4]	Vimeo-Triplets-Train	70.3	120	34.40	0.970	36.48	0.964	35.01	0.968
RIFE-L [15]	Vimeo-Triplets-Train	20.9	72	36.10	0.980	37.64	0.985	35.29	0.969
M2M-PWC [14]	Vimeo-Triplets-Train	-	-	35.40	0.978	-	-	35.38	0.969
EA-Net [54]	Vimeo-Triplets-Train	-	-	34.39	0.975	-	-	34.97	0.968
IFRNet-L [19]	Vimeo-Triplets-Train	19.7	-	36.20	0.981	37.50	0.968	35.42	0.970
Splat-VFI [29]	Vimeo-Triplets-Train	-	-	35.00	-	38.42	0.971	36.63	-
VFIFormer [28]	Vimeo-Triplets-Train	24.2	1431	36.50	0.982	38.43	0.987	35.43	0.970
DKR-VFI [41]	Vimeo-Triplets-Train	31.2	-	34.52	0.961	-	-	35.50	0.965
Ours-triplets w/o TCL	Vimeo-Triplets-Train	28.9	292	36.56	0.981	38.64	0.970	35.37	0.969
Ours-triplets	Vimeo-Triplets-Train	28.9	292	36.85	0.982	38.83	0.989	35.43	0.979

Table 1. Quantitative comparison of single-frame VFI algorithms. The numbers in red and blue refer to the best and second-best PSNR(dB)/SSIM results. Runtime of each model is also reported with an input size of 2 × 480 × 640 using an RTX 2080Ti.

rotation augmentations. The detailed framework architecture is illustrated in our supplementary materials.

4.2. Datasets and Evaluation Metrics

Vimeo-Triplets [47]. It contains 51,312 and 3,782 triplet frames with a resolution of 256 × 448 for training and testing, respectively. Following the most commonly used protocols [3, 4, 6, 7, 10, 15, 31, 35, 47], we train a model for VFI on the training split (Vimeo-Triplets-Train) while evaluating the results on the testing part (Vimeo-Triplets-Test).

Middlebury [2] & UCF101 [40]. Both of the two testing are used for evaluation only. In Middlebury [2], there are 12 challenging cases where each of them contains three video frames. The central frame serves as the ground truth while the others are used as the input. UCF101 [40] contains 379 triplets (256 × 256) for video frame interpolation evaluation. Unlike the aforementioned datasets, UCF101 [40] has heavy compression noises. Following recent methods [4, 6, 7, 10, 15, 31, 35, 47], we assess our method on this benchmark without finetuning.

Metrics. We adopt PSNR/SSIM [43] as the evaluation metrics. The higher values indicate the better results.

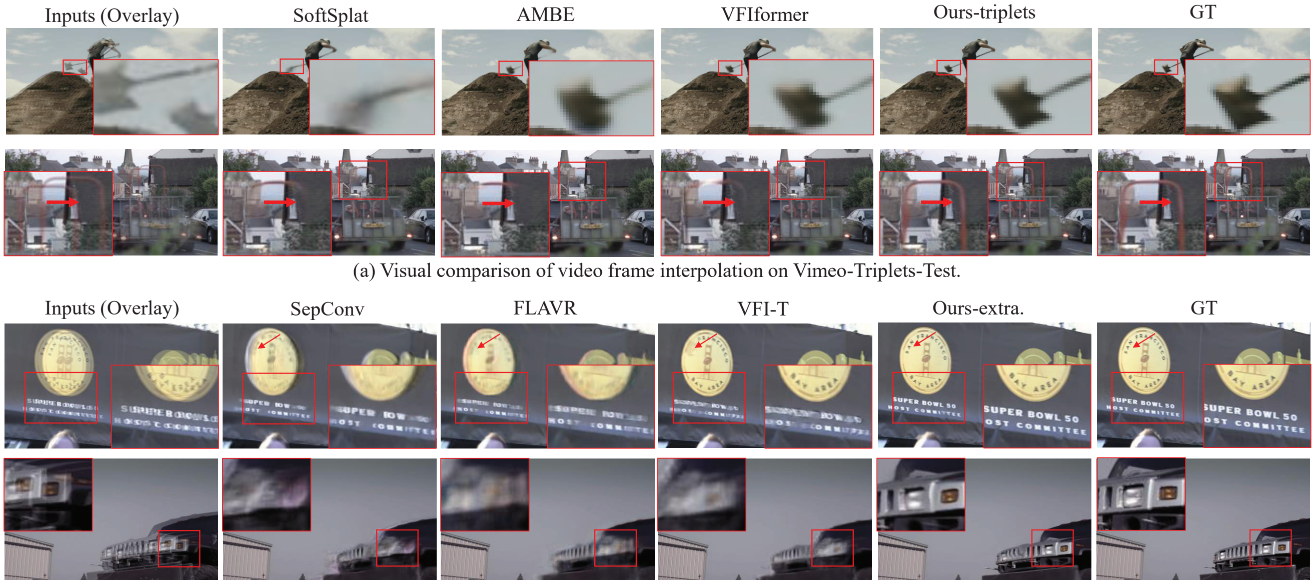
4.3. Comparison with SOTA Methods

To verify the effectiveness of the proposed method, we make a comparison with state-of-the-art methods under video frame interpolation/extrapolation settings.

Video frame interpolation. As illustrated in Table 1, it is clear that our model achieves a new state of the art on all benchmarks. While some methods may incorporate additional information (e.g., optical flow, depth), our method still stands out as the best. For example, SoftSplat [31] relies on accurate bidirectional optical flow, RIFE-L [15] and VFIformer [28] require a optical flow supervision, and DAIN [3] utilizes additional depth information. Besides, compared with FeFlow [13] that adopts multi-scale aggregation, our model with cross-scale aggregation achieves superior performance (↑ 1.57dB on Vimeo) with much fewer parameters. Meanwhile, due to the heavy compression nature of images in UCF101, our model obtains comparable performance with previous SOTA approaches, indicating that the domain gap is a critical issue in the VFI task.

We also show some visual examples in Fig. 6 and Fig. 5. Compared with other methods, our model successfully handles complicated motion and produces more plausible structures. In terms of the third example in Fig. 6(a), all other methods fail to restore the right structure of the fast moving objects, while ours interpolates the hand that is closest to the ground truth. Also, as illustrated in Fig. 5, the error maps show that our model achieves the best reconstruction result among all approaches [15, 28, 30, 31, 33, 34], further demonstrating the effectiveness of our method.

Video frame extrapolation. In addition, we also evaluate our method on the video frame extrapolation task. Unlike video frame interpolation, extrapolation aims to synthesize the future frames based on the observed historical frames. All the flow-based methods are heavily dependent on the pre-defined displacements, making them unsuitable for extrapolation. In this case, we compare our method with SepConv [33], FLAVR [18] and VFI-T [38] as they do not require optical-flow information. SepConv adopts a U-Net to regress a pair of separable 1D kernels to perform convo-



(a) Visual comparison of video frame interpolation on Vimeo-Triplets-Test.

(b) Visual comparison of video frame extrapolation on Vimeo-Triplets-Test.

Figure 6. Visual comparison of state-of-the-art algorithms. (a-b) refer to the qualitative results of video frame interpolation/extrapolation, respectively. Our method outperforms other state-of-the-art approaches with finer details and fewer artifacts.

Methods	# Param.	Vimeo-Triplets-Test	Middlebury
SepConv [33]	21.7M	30.42	32.21
FLAVR [18]	42.1M	31.14	32.90
VFI-T [38]	29.1M	31.18	33.60
SepConv w /TCL	21.7M	31.14 (↑ 0.72)	33.53 (↑ 1.32)
FLAVR w /TCL	42.1M	31.35 (↑ 0.21)	33.27 (↑ 0.37)
VFI-T w /TCL	29.1M	31.28 (↑ 0.10)	33.72 (↑ 0.12)
Ours-extra.	**21.5M**	**32.16**	**34.85**

Table 2. Quantitative comparison of SepConv, FLAVR, VFI-T and our method for video frame extrapolation. The models are trained on the Vimeo-Triplets-Train dataset and evaluated on the Vimeo-Triplets-Test set and Middlebury benchmarks. The best PSNR(dB) results are highlighted in **bold**.

lutional operations on the two input frames. FLAVR proposes an efficient 3D convolutional neural network for reconstruction. For a fair comparison, we retrain the three models from scratch under the same experimental setting on Vimeo-Triplets. More specifically, we need to predict a future frame I_3 from historical $\{I_1, I_2\}$. We use fewer residual blocks (18 residual blocks) in the reconstruction module, which makes our model has comparable parameters (21.5M) with the SepConv (21.7M) and much fewer parameters than FLAVR (42.1M).

As shown in Table 2, compared with SepConv and FLAVR, our model boosts PSNR by 1.74dB and 1.02dB, respectively. In addition, our model is nearly **2** times smaller than the FLAVR. Notably, our proposed TCL is plug-and-play and capable of significantly improving the performance of existing approaches. For example, SepConv supervised by TCL achieve much better results, demonstrating the effectiveness of our design. As depicted in Fig. 6(b), our method produces sharper edges and fewer artifacts. Especially in the first image, our model can produce recognizable characters. In a nutshell, both the quantitative and qualitative results demonstrate that our model is capable of generating high-quality extrapolated frames. More results are provided in our supplementary materials.

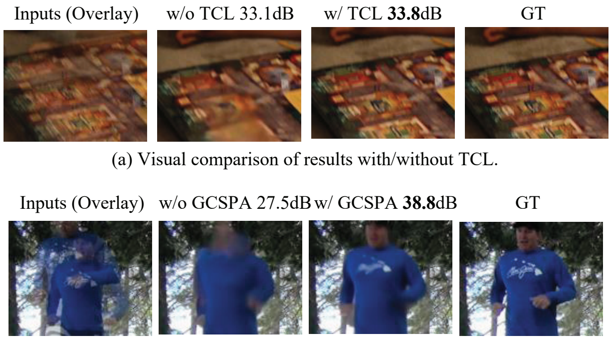
Method	PSNR (dB)	SSIM
Baseline	35.90	0.969
Baseline w/ TCL	36.21(+0.31)	0.977(+0.008)
Baseline w/ GCSPA	36.56(+0.66)	0.976(+0.007)
Full	36.85(+0.95)	0.982(+0.013)

Table 3. Ablation studies of the proposed components.

4.4. Ablation Study

Here, we make analysis of the contribution of each proposed component under the VFI setting. The Vimeo-Triplets-Test is adopted as the evaluation benchmark.

Texture consistency loss (TCL). The proposed TCL is designed to alleviate the over-constrained issue of the predefined ground truth, which actually is just one of many possible solutions given observed input frames. To verify this claim, we compare the conventional L1 loss and the proposed TCL. As illustrated in Table 3, the baseline trained with the additional TCL achieves a better performance in terms of PSNR and SSIM compared to the baseline. We also give a visual example to illustrate the impact of the proposed TCL in Fig. 7(a). The model trained with TCL is able to preserve the structures of interpolated contents. Moreover, the extensive empirical results in Fig. 1 and Tab. 2



(a) Visual comparison of results with/without TCL.

(b) Visual comparison of results with/without GCSPA.

Figure 7. Effects of the proposed TCL and GCSPA.

α	0	0.1	0.5	1.0	2.0	10.0
PSNR (dB)	36.56	**36.85**	36.69	36.69	36.54	-
SSIM	0.976	**0.982**	0.979	0.979	0.978	-

Table 4. Analysis of α in Eq. 1.

K	3	5	7	9
PSNR (dB)	**36.85**	36.64	36.56	36.50
SSIM	**0.982**	0.979	0.978	0.978

Table 5. Analysis of different patch sizes in TCL.

demonstrate that our proposed TCL is able to further improve the performance of SOTA methods [18, 33, 38] on both video frame interpolation/extrapolation, significantly.

Guided cross-scale pyramid alignment. Different from existing works that apply temporal alignment on a specific scale or multiple scales individually, we propose a guided cross-scale pyramid alignment (GCSPA) that enables a more accurate alignment. As shown in Table 3, the model with the proposed GCSPA leads to a 0.66dB improvement on PSNR compared with the baseline. Furthermore, we also give a visual example for qualitative evaluation in Fig. 7(b). The GCSPA benefits our model in restoring the structure of the human face and patterns on the clothes more clearly.

Hyperparamter α in Eq. 1. The hyperparameter α is used to balance the predefined ground truth and our proposed pseudo label. From Table 4, we notice that $\alpha = 0.1$ is the best setting in our experiments (may not be optimal), and a large α harms the performance of TCL. Especially, the model trained with $\alpha = 10.0$ fails to converge. We think of the proposed pseudo label better serving as auxiliary supervision apart from the L1 loss.

Patch size K of TCL. We explore the influence of the patch size $K \in \{3, 5, 7, 9\}$ used in the TCL. As shown in Table 5, a larger patch size may degrade the performance of the model. It is reasonable since the increase in patch size brings more difficulties in matching correctly to the candidates on neighboring frames and the inaccurate supervision signals bring negative impacts during training.

Census transform. We analyze the effect of adopting census transform in our TCL. We train another model by performing patch matching in the RGB space directly (denoted as "TCL-RGB"). The results are described in Table 6. It is observed that "TCL-RGB" leads to a lower interpolation quality in terms of PSNR and SSIM, which supports the claim that census transform is useful in eliminating the interference of illumination in Sec. 3.2.

Method	Vimeo-Triplets-Test	Middlebury
TCL-RGB	36.57/0.978	38.41/0.988
TCL-CT	**36.85/0.982**	**38.85/0.989**

Table 6. Analysis of cencus transform in TCL. We adopt PSNR(dB) and SSIM as the evaluation metrics.

Limitation. Despite significant improvements on interpolation quality, our model has a larger model capacity than some lightweight models, e.g., SoftSplat [31], PWDN [5]. Reducing the model complexity is our future direction for real-time video frame interpolation/extrapolation applications. Also, unlike SoftSplat [31] that can explicitly generate frame at arbitrary time step $t \in (0, 1)$, our method is only supposed to synthetic an intermediate frame at $t = 0.5$. We will improve the model flexibility along this line.

5. Conclusion

We present a novel and effective video frame interpolation/extrapolation approach. The proposed texture consistency loss relaxes the strict constraint of the pre-defined ground truth and the guided cross-scale pyramid alignment is able to make better use of multi-scale information, making it possible to generate much clearer details. Comprehensive experiments have demonstrated the effectiveness of our method to interpolate/extrapolate high-quality frames. In the future, we plan to study the potential of our method on other video restoration problems, such as video super-resolution, video deblurring and video denoising.

Acknowledgments. This work is partially supported by Shenzhen Science and Technology Program KQTD20210811090149095, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone. It was also supported in part by NSFC-62172348 and Shenzhen General Project with No. JCYJ20220530143604010, the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund 202002, by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), and by Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055).

References

[1] Dawit Mureja Argaw and In So Kweon. Long-term video frame interpolation via feature propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3543–3552, 2022. 1

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. 2, 5, 6

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 6

[4] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 6

[5] Zhiqi Chen, Ran Wang, Haojie Liu, and Yao Wang. Pdwn: Pyramid deformable warping network for video interpolation. *IEEE Open Journal of Signal Processing*, pages 1–1, 2021. 1, 2, 3, 6, 8

[6] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10607–10614, 2020. 1, 6

[7] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2, 3, 6

[8] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 107–123. Springer, 2020. 1

[9] Jinsoo Choi, Jaesik Park, and In So Kweon. High-quality frame interpolation via tridirectional inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 596–604, 2021. 2

[10] 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. 6

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4

[12] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 3

[13] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14004–14013, 2020. 2, 6

[14] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3553–3562, 2022. 6

[15] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 5, 6

[16] Tejas Jayashankar, Pierre Moulin, Thierry Blu, and Chris Gilliam. Lap-based video frame interpolation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4195–4199. IEEE, 2019. 1

[17] 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

[18] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512*, 2020. 6, 7, 8

[19] 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. 6

[20] Malwina Kubas and Grzegorz Sarwas. Fastrife: Optimization of real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2105.13482*, 2021. 1

[21] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 3

[22] 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

[23] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *European Conference on Computer Vision*, pages 335–351. Springer, 2020. 2

[24] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information*

Processing Systems, volume 33, pages 20343–20355. Curran Associates, Inc., 2020. 2

[25] Wenbo Li, Kun Zhou, Lu Qi, Liying Lu, and Jiangbo Lu. Best-buddy gans for highly detailed image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1412–1420, 2022. 2

[26] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *European Conference on Computer Vision*, pages 41–56. Springer, 2020. 1

[27] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 1

[28] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022. 1, 5, 6

[29] Simon Niklaus, Ping Hu, and Jiawen Chen. Splatting-based synthesis for video frame interpolation. *arXiv preprint arXiv:2201.10075*, 2022. 6

[30] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 1, 2, 5, 6

[31] 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, 5, 6, 8

[32] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 1, 2

[33] 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, 3, 5, 6, 7, 8

[34] Simon Niklaus, Long Mai, and Oliver Wang. Revisiting adaptive convolutions for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1099–1109, 2021. 5, 6

[35] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. *arXiv preprint arXiv:2007.12622*, 2020. 6

[36] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019. 2

[37] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. *IEEE Transactions on Multimedia*, 2021. 1

[38] 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. 6, 7, 8

[39] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6587–6595, 2021. 1

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11), 2012. 6

[41] Haoyue Tian, Pan Gao, and Xiaojiang Peng. Video frame interpolation based on deformable kernel region. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1349–1355. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track. 6

[42] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[44] Jinbo Xing, Wenbo Hu, Yuechen Zhang, and Tien-Tsin Wong. Flow-aware synthesis: A generic motion model for video frame interpolation. *Computational Visual Media*, pages 1–13, 2021. 1

[45] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[46] Fanyong Xue, Jie Li, Jiannan Liu, and Chentao Wu. Bwin: A bilateral warping method for video frame interpolation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 1

[47] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 2, 6

[48] Ramin Zabih and John Woodfill. A non-parametric approach to visual correspondence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Citeseer, 1996. 4

[49] Jiefu Zhai, Keman Yu, Jiang Li, and Shipeng Li. A low complexity motion compensated frame interpolation method. In *2005 IEEE International Symposium on Circuits and Systems*, pages 4927–4930. IEEE, 2005. 1

[50] Haochen Zhang, Dong Liu, and Zhiwei Xiong. Is there tradeoff between spatial and temporal in video super-resolution? *arXiv preprint arXiv:2003.06141*, 2020. 3

[51] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. In *European Conference on Computer Vision*, pages 474–491. Springer, 2020. 1

[52] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale

cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1590–1597, 2014. 2

[53] Bin Zhao and Xuelong Li. Ea-net: Edge-aware network for flow-based video frame interpolation. *arXiv preprint arXiv:2105.07673*, 2021. 1

[54] Bin Zhao and Xuelong Li. Edge-aware network for flow-based video frame interpolation. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 6

[55] Kun Zhou, Wenbo Li, Liying Lu, Xiaoguang Han, and Jiangbo Lu. Revisiting temporal alignment for video restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6053–6062, 2022. 3

[56] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 4