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

Kun Zhou1,2 Wenbo Li3 Xiaoguang Han1 Jiangbo Lu2*
1SSE, CUHK-Shenzhen, 2SmartMore Corporation 3CUHK
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 $I_{t−1}$ and $I_t$, we conduct a user study of choosing its most possible position in the intermediate frame $I_o$. 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
parison of SepConv [33] and our model with/without TCL 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 in-
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.

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 $F_{-1}→0$ and $F_{1}→0$, 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 $\alpha$ is to balance the importance of the two items.

3. Methodology

In this section, we first give an overview of the proposed algorithm for video frame interpolation (VFI) in Sec. 3.1.

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.

3.1. 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 $F_{-1}→0$ and $F_{1}→0$, 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 $\alpha$ is to balance the importance of the two items.
3.2. Texture Consistency Loss

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

$$y^*, t^* = \arg\min_{y, t} L_2(\hat{f}_x, f_{y^*}^t),$$

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

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:

$$v_x(x + x_n) = \begin{cases} 0, & f_x(x) > f_x(x + x_n), \\ 1, & f_x(x) \leq f_x(x + x_n) \end{cases}, x_n \in \mathcal{R},$$

where $f_x(x)$ is the pixel value at central position $x$ and the patch field $x_n$ is defined as

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

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

$$\phi(x) = \{y \mid |y - x| \leq d\},$$

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

$$y^*, t^* = \arg\min_{y \in \phi(x), t \in [-1, 1]} L_2(\hat{v}_x, v_y^t),$$

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

$$L_p(\hat{I}_0, I_{-1}, I_1)(x) = L_1(\hat{f}_x, f_{y^*}^t).$$

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 pixelwise 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., $F_0^{0}$), 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

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

Then, this result serves 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 $F_{-1}^{2} \rightarrow 0$ and $F_1^{2}$. In detail, we first bilinearly upsample $F_{-1}^{2} \rightarrow 0$ by a factor of 2 to obtain $F_{-1}^{2 \times 2}$, and then calculate the window-based similarity maps between $F_{-1}^{2 \times 2}$ and $F_1^{1}$, yielding $C_{-1}^{2 \times 2}$. The fusion is finally carried out as

$$\hat{F}_{-1}^{1 \rightarrow 0} = \text{Fuse}(F_{-1}^{2 \times 2}, F_1^{1}, C_{-1}^{1 \times 2}),$$

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

$$F_{-1}^{1} = \text{Align}(\hat{F}_{-1}^{1 \rightarrow 0}, F_1^{1}).$$

22172
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

$$\tilde{F}_{l-1 \rightarrow 0} = F_{\text{use}}(F_{l-1 \rightarrow 0}, F_{l-1 \rightarrow 0}, F_{l-1}, C_{l-1}, C_{l-1}) \text{.} \quad (11)$$

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

The alignment block is zoomed up in Fig. 4b. In terms of the $l$-th level alignment for frame $I_{l-1}$, the block first concatenates the fused cross-scale feature $\tilde{F}_{l-1 \rightarrow 0}$ and $F_{l-1}$, and conducts a $3 \times 3$ convolution. Five sequential residual blocks and another convolution are used to predict a weight map $W^l_{l-1 \rightarrow 0}$ and an offset map $O^l_{l-1 \rightarrow 0}$. Finally, the aligned feature $F^l_{l-1 \rightarrow 0}(x)$ at position $x$ is calculated by

$$F^l_{l-1 \rightarrow 0}(x) = \sum_i \tilde{F}^l_{l-1 \rightarrow 0}(x + O^l_{l-1 \rightarrow 0,i}(x)) \ast W^l_{l-1 \rightarrow 0,i}(x), \quad (12)$$

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 $F^0_{l-1 \rightarrow 0}$ and $F^0_{l+1 \rightarrow 0}$. 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

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

Then, the final aggregated result $F_0$ is obtained by

$$F_0 = M \ast F^0_{l-1 \rightarrow 0} + (1 - M) \ast F^0_{l+1 \rightarrow 0} \text{.} \quad (14)$$

### 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 $5 \times 10^{-4}$ to $0$ by a cosine annealing strategy. We set the batch size to 64. The training lasts 600K iterations, during which we adopt random $64 \times 64$ cropping, vertical or horizontal flipping, and $90^\circ$
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 $256 \times 448$ 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 ($256 \times 256$) 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 ($\uparrow 1.57$dB 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 con-
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.

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

---

**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.

**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.

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

**Table 3.** Ablation studies of the proposed components.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.90</td>
<td>0.969</td>
</tr>
<tr>
<td>Baseline w/ TCL</td>
<td>36.21 (+0.31)</td>
<td>0.977 (+0.008)</td>
</tr>
<tr>
<td>Baseline w/ GCSPA</td>
<td>36.56 (+0.66)</td>
<td>0.976 (+0.007)</td>
</tr>
<tr>
<td>Full</td>
<td>36.85 (+0.95)</td>
<td>0.982 (+0.013)</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Methods</th>
<th># Param. Vimeo-Triplets-Test</th>
<th>Middlebury</th>
</tr>
</thead>
<tbody>
<tr>
<td>SepConv [33]</td>
<td>21.7M 30.42 32.21</td>
<td></td>
</tr>
<tr>
<td>FLAVR [18]</td>
<td>42.1M 31.14 32.90</td>
<td></td>
</tr>
<tr>
<td>VFI-T [38]</td>
<td>29.1M 31.18 33.60</td>
<td></td>
</tr>
<tr>
<td>SepConv w/TCL</td>
<td>21.7M 31.14 (↑0.72) 33.53 (↑1.32)</td>
<td></td>
</tr>
<tr>
<td>FLAVR w/TCL</td>
<td>42.1M 31.35 (↑0.21) 33.27 (↑0.37)</td>
<td></td>
</tr>
<tr>
<td>VFI-T w/TCL</td>
<td>29.1M 31.28 (↑0.10) 33.72 (↑0.12)</td>
<td></td>
</tr>
<tr>
<td>Ours-extra.</td>
<td>21.5M 32.16 34.85</td>
<td></td>
</tr>
</tbody>
</table>
demonstrate that our proposed TCL is able to further improve the performance of SOTA methods \cite{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.

**Hyperparameter $\alpha$ in Eq. 1.** The hyperparameter $\alpha$ 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 $\alpha$ 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 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Vimeo-Triplets-Test</th>
<th>Middlebury</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCL-RGB</td>
<td>36.85/0.982</td>
<td>38.85/0.989</td>
</tr>
<tr>
<td>TCL-CT</td>
<td>36.57/0.978</td>
<td>38.41/0.988</td>
</tr>
</tbody>
</table>

Table 6. Analysis of census 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 \cite{31}, PWDN \cite{5}. Reducing the model complexity is our future direction for real-time video frame interpolation/extrapolation applications. Also, unlike SoftSplat \cite{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-KCZY-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


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


