This WACV 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

High-quality Frame Interpolation via Tridirectional Inference

Jinsoo Choi
KAIST
Rep. of Korea
jinsc37@kaist.ac.kr

Jaesik Park
POSTECH
Rep. of Korea
jaesik.park@postech.ac.kr

In So Kweon
KAIST
Rep. of Korea
iskweon77@kaist.ac.kr

Abstract

Videos have recently become an omnipresent form of media, gathering much attention from industry as well as academia. In the video enhancement field, video frame interpolation is a long-studied topic that has dramatically improved due to the advancement of deep convolutional neural networks (CNN). However, conventional approaches utilizing two successive frames often exhibit ghosting or tearing artifacts for moving objects. We argue that this phenomenon comes from the lack of reliable information provided only by two frames. With this motivation, we propose a frame interpolation method by utilizing tridirectional information obtained from three input frames. Information extracted from triplet frames allows our model to learn rich and reliable inter-frame motion representations, including subtle nonlinear movement, which can be easily trained via any video frames in a self-supervised manner. We demonstrate that our method generalizes well to high-resolution content by evaluating on FHD resolution, and illustrates our approach's effectiveness via comparison to state-of-the-art methods on challenging video content.

1. Introduction

Videos have become a major media form in various domains, including entertainment, education, marketing, health, behavior analysis, etc. Due to such popularity, videos have gathered much attention from industry and academia. Thus, there has been considerable effort to enhance the quality of videos. In the field of video processing and enhancement, frame interpolation has been studied in the last few decades due to its applications to frame upsampling, visual effects, and video compression applicable to various display devices.

Recently, due to the significant advancement in deep neural networks, frame interpolation methods have significantly improved and have shown impressive results. Prior arts include kernel based [15, 16], optical flow based [13, 9], and phase based [11] methods which demonstrate promising results on benchmark datasets. Recently, Bao *et al*. [2] proposed a method utilizing depth information, and Niklaus *et al*. [14] proposed softmax splatting to improve performance further.

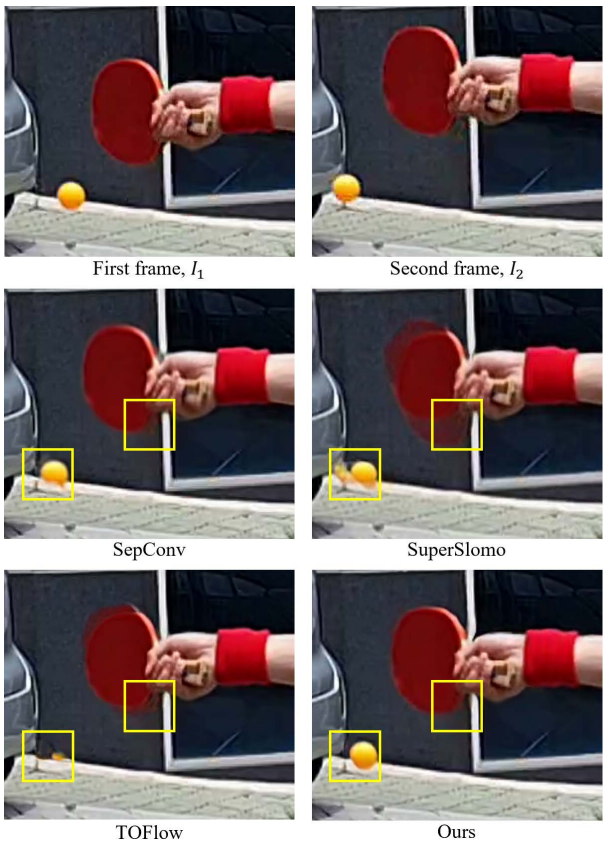

Figure 1. Comparison of interpolation quality with SepConv [16], SuperSlomo [7], TOFlow [24] and our method. Our approach does not show ghosting or tearing artifacts (yellow boxes) by using tridirectional inference and a new data augmentation scheme. Shown patches are cropped from the FHD images.

Many approaches utilize the bidirectional flow maps between two consecutive frames to infer motion information. Given only two frames, the interpolation position will most likely become the exact *mid-point* of motion, since it is the

maximum likelihood-based on the given information. However, this may not be optimal since it does not consider the underlying motion characteristics (e.g., change in direction and speed) of complex movement. Moreover, information obtained only from two frames may lack reliability of motion information due to existing subtle visual characteristics like disappearance and appearance (e.g., lights blinking), which perhaps requires another input frame for confirmation (i.e., whether it is indeed a moving entity). As a result, as shown in Fig 1, the interpolated frame exhibits ghosting or tearing artifacts, especially for moving objects.

In this work, we propose a high-quality frame interpolation method for videos by utilizing tridirectional information provided by input frame triplets, which demonstrates an effective way to extract complex and reliable motion information. In other words, our method makes use of the third frame in addition to the conventional two frame input to enhance the level of video understanding. This free source of information provides significant representational power for complex movements without the additional computational overhead. Since it is safe to assume that videos typically contain more than two frames, our approach does not impose any constraints on applicable videos.

We also propose a new data augmentation approach to facilitate robustness to challenging motion profiles by overlaying flying objects on top of the video frame data. Specifically, we segment objects from the PASCAL VOC dataset [4] and overlay them on top of the Vimeo90k [24] video frames with flying projectiles. We also incorporate an additional loss term dedicated to the interpolation of flying objects to facilitate robust learning. Due to our method's representational power of object motion, our method can extract reliable motion characteristics of small and fast-moving objects, as shown in Fig. 1.

Our method is easy to train since it is based on self-supervised learning that only requires video frames of any content (without the need for labeling efforts whatsoever). Our approach uses far fewer parameters while outperforming the state-of-the-art methods.

Our work contains the following contributions:

1. We propose a deep architecture capable of learning rich and reliable motion characteristics via inter-frame motion information among adjacent triplet frames.

2. Our method enables effective learning via self-supervised video data in addition to a novel data augmentation scheme and a dedicated loss term.

3. Our approach is light-weight with fewer parameters that can perform well with high-resolution videos and outperform state-of-the-art algorithms on complex scenes.

We evaluate our method's effectiveness and robustness via various high-definition videos against several state-of-the-art frame interpolation methods. Our approach demonstrates superior results in terms of quantitative and qualitative comparisons.

2. Related Work

Due to the success of CNNs and their application to numerous computer vision and graphics tasks, CNNs have also been applied to video frame interpolation. One of the earlier works on deep frame interpolation includes the work of Niklaus *et al*. [16] which proposed an architecture that takes two image patches and estimates convolution kernels for the pixel centered at each patch. The kernels are convolved with the input image patches to synthesize all of the output pixels. Although this work demonstrates an effective deep method for frame interpolation, it requires large convolutional kernels of size 41×41 to handle large displacements.

Due to its high computational cost, this work was extended [15] to a deep fully convolutional neural network (CNN) that takes two video frames as input to estimate four 1D kernels for all pixels. Each 1D kernel represents the horizontal and vertical kernels (to form a 2D kernel) for each input frame pixel. Another prominent work addressed the frame interpolation task via a phase-based approach [11]. Given two frames, the neural network architecture estimates the phase decomposition of the middle frame, which is then combined to generate the final output frame.

Recent frame interpolation methods utilize deep optical flow networks to compute bidirectional flow between the two input frames. Niklaus *et al*. [13] computed the bidirectional flow to warp the two input frames *halfway towards* each other as well as its context features to synthesize the middle frame. Similarly, Liu *et al*. [9] proposed a voxel flow layer given two consecutive input frames which estimates the interpolated motion vector field (IMVF) and an occlusion map to generate the output frame. Moreover, Xue *et al*. [24] proposed the task-oriented flow that emphasizes the role of optical flow on various video tasks such as frame interpolation, video denoising, deblocking, and video super-resolution. This work demonstrates that each video task requires a dedicated optical flow computation.

Another flow-based approach by Jiang *et al*. [7] coined as SuperSlomo, utilizes two U-Net [20] architectures first to estimate the bi-directional flow maps between the given two frames, then estimate the flow maps from the (to be generated) middle frame to each input frame. Via these estimated flow maps, the output frame is generated. A recent work by Bao *et al*. [2] utilized monocular depth information on top of optical flow, context features, and kernel methods to improve interpolation quality. However, since their model includes many modules, the number of parameters is significantly large (24 million) and thus limited in applicable

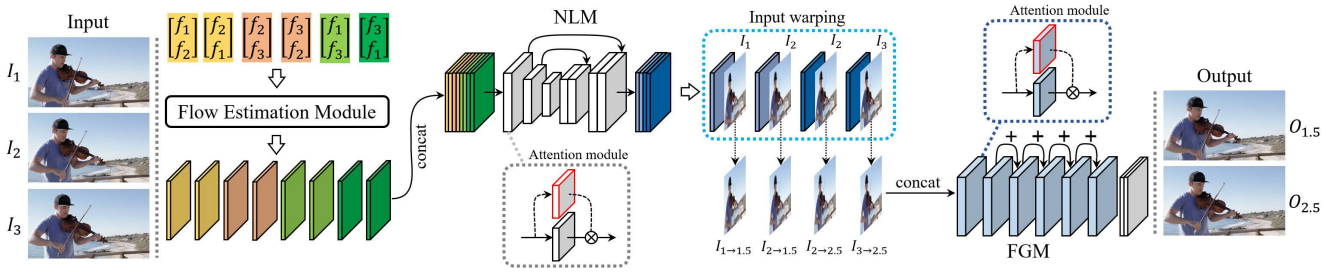

Figure 2. Overview of our network. Our network utilizes three frames I_1, I_2, and I_3 for nonlinear frame interpolation. The flow estimation module computes flow maps for all combinations of the input frames. The flow maps are concatenated and fed through the nonlinear motion estimation module (NLM) which outputs four nonlinear flow maps which are then used to warp the input frames to the intermediate positions between I_1, I_2 and I_2, I_3, (i.e., positions 1.5 and 2.5). Feeding the warped frames, the frame generation module (FGM) outputs $O_{1.5}$ and $O_{2.5}$ as the interpolated frames.

high-resolution videos. Another recent work by Niklaus *et al*. [14] utilized forward warping for motion compensation by means of softmax splatting, once again using bidirectional information. Meanwhile, Xu *et al*. [23] proposed using four frame inputs for estimating quadratic movement. Our work provides a more general learning approach, dealing with tridirectional motion, and shows that three frame inputs are powerful enough to express complex motion, in terms of theoretical and empirical analysis.

Until the work of Peleg *et al*. [18], previous methods have not addressed application to high-resolution videos. Their approach addresses high-resolution video frame interpolation via training on patches collected from high-resolution videos. Their method also estimates the interpolated motion vector field (IMVF) and an occlusion map to generate the output middle frame. However, instead of evaluating on *real-world* high-resolution videos, Peleg *et al*. evaluate their method on up-sampled video frames from the Vimeo90k dataset [24] using an off-the-shelf super-resolution (SR) algorithm.

In this work, we propose a robust frame interpolation method for complex object movement and visual characteristics. Instead of estimating the linear interpolation of pixels given two frames, we utilize an additional frame to estimate general and reliable nonlinear motion. Furthermore, since our approach is fully convolutional and lightweight, our method can generate interpolated frames for high-resolution videos such as FHD.

3. Proposed Method

3.1. Motivation

One of our approach's key ideas is to achieve robustness to nonlinear movement and reliability using a single additional frame. This concept can be explained in terms of vector equations. If the coordinates of the same object in each image are $\mathbf{p}_1$ and $\mathbf{p}_2$, we can draw a line $\mathbf{r}_L$ through the points:

$$\mathbf{r}_L(\lambda) = \mathbf{v}_1 + \lambda \mathbf{v}_{12}, \tag{1}$$

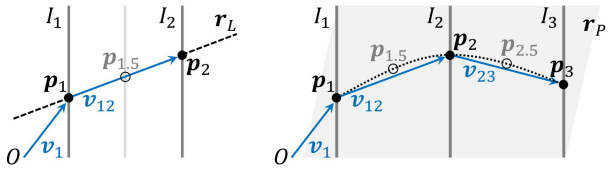

Figure 3. An illustration of the (a) two frame and (b) three frame interpolation. Given $\mathbf{p}_1$ and $\mathbf{p}_2$ on the frames I_1 and I_2 (viewed from side angle), interpolation is done on the line $\mathbf{r}_L$, while an additional point $\mathbf{p}_3$ from frame I_3 allows interpolation on the plane $\mathbf{r}_P$ with the necessary information of the nonlinear movement (dotted curve).

where $\mathbf{v}_1$ is the position vector of $\mathbf{p}_1$, $\mathbf{v}_{12}$ is the vector from $\mathbf{p}_1$ to $\mathbf{p}_2$, and λ is its coefficient, as shown in Fig. 3 (a). The frame synthesis network would most likely assign λ to be 0.5 with sufficient training data since it is the best option (maximum likelihood) to minimize the discrepancy between the synthesized image and training data. Even if the data contains nonlinear motion, it will not learn it but instead take on the maximum likelihood estimate (average position across all dataset samples), due to the representational limits of the two-frame assumption.

Meanwhile, given an additional point $\mathbf{p}_3$ from the next frame, any point on the plane $\mathbf{r}_P$ can be modeled using the two vectors:

$$\mathbf{r}_P(\lambda, \mu) = \mathbf{v}_1 + \lambda \mathbf{v}_{12} + \mu \mathbf{v}_{23}, \tag{2}$$

where $\mathbf{v}_{23}$ is the vector from point $\mathbf{p}_2$ to $\mathbf{p}_3$, and μ is its coefficient. This is illustrated in Fig. 3 (b). Thus, with sufficient training data, the interpolated position denoted as $\mathbf{p}_{1.5}$ and $\mathbf{p}_{2.5}$ can be reconstructed even if they are on a free-form curve. Since *any* point on the plane $\mathbf{r}_P$ can be estimated, theoretically, three frames are enough to estimate any nonlinear motion. The network can learn the best parameters for λ and μ by understanding scene context (such as velocity or physically reasonable path) using evidence (maximum likelihood) captured from the three frames.

3.2. Model Architecture

With the key motivation, we propose a frame interpolation network that utilizes three continuous frames effectively. Our proposed architecture consists of a flow estimation module, a nonlinear motion estimation module, and the frame generation module. The flow estimation module computes the bidirectional flow maps between each combination of three given frames I_1, I_2, and I_3, resulting in the tridirectional inference. The nonlinear motion estimation module combines the flow information to refine the flow maps representing any nonlinear motion. Then, the given three frames are warped via the refined nonlinear flow maps and fed through the frame generation module that outputs two interpolated frames $I_{1.5}$ and $I_{2.5}$ between I_1, I_2 and I_2, I_3. Using a third additional frame as input also provides the benefit of confirming any motion characteristics inferred from otherwise bidirectional input, making tridirectional inference reliable. This is significant information that supports or corrects any unsure inference done with only two frames, such as scenarios containing disappearing objects (e.g., blinking lights). The overview of our model is illustrated in Fig. 2.

Specifically, given three frames I_1, I_2, and I_3, the flow estimation module (FEM) computes the bidirectional flow between each combination of input frames as defined below (with a slight abuse of notation):

$$\begin{aligned} &FEM(I_1, I_2, I_3) \\ &= [\mathcal{F}\{(I_1, I_2), (I_2, I_1), (I_2, I_3), (I_3, I_2), (I_1, I_3), (I_3, I_1)\}] \\ &= [M_{1\to2}, M_{2\to1}, M_{2\to3}, M_{3\to2}, M_{1\to3}, M_{3\to1}], \end{aligned} \tag{3}$$

where $\mathcal{F}$ represents the optical flow estimation network taking two image pairs as input, $M_{1\to2}$ is the warping map obtained from I_1 to I_2 and so on.

Next, given $\mathcal{M} = [M_{1\to2}, ..., M_{3\to1}]$ from FEM, the nonlinear motion estimation module (NLM) can be expressed as follows:

$$NLM(\mathcal{M}) = [M_{1\to1.5}, M_{2\to1.5}, M_{2\to2.5}, M_{3\to2.5}], \tag{4}$$

where NLM represents the neural network architecture which is U-Net [20] inspired (due to its ability to utilize global and local information), outputting four sets of refined nonlinear flow maps. Essentially, the NLM is fully learned to estimate any such scaling and nonlinear combinations of the given input, for nonlinear motion. Note that the NLM architecture comprises of attention layers for each constituent CNN layer. This attention layer enables the architecture to learn proper scaling and nonlinear combinations of features.

Given the refined nonlinear flow maps $\{M_{1\to1.5}, ..., M_{3\to2.5}\}$ from NLM, input frames I_1, I_2, and I_3 are warped as expressed as follows:

$$\begin{aligned} \mathcal{I} &= \begin{bmatrix} \mathcal{W}(I_1, M_{1\to1.5}) \\ \mathcal{W}(I_2, M_{2\to1.5}) \\ \mathcal{W}(I_2, M_{2\to2.5}) \\ \mathcal{W}(I_3, M_{3\to2.5}) \end{bmatrix} \\ &= [I_{1\to1.5}, I_{2\to1.5}, I_{2\to2.5}, I_{3\to2.5}]^\top, \end{aligned} \tag{5}$$

where $\mathcal{I}$ is a set of four warped frames $I_{1\to1.5}$, $I_{2\to1.5}$, $I_{2\to2.5}$, and $I_{3\to2.5}$, while $\mathcal{W}$ represents the bicubic backward warping process. These warped frames represent frames warped towards the two intermediate positions between I_1, I_2, and I_3.

The warped frames $\mathcal{I}$ are concatenated and fed through the frame generation module (FGM), expressed as:

$$FGM(\mathcal{I}) = [O_{1.5}, O_{2.5}], \tag{6}$$

where it outputs the final intermediate frames $O_{1.5}$, $O_{2.5}$. The frame generation module consists of ResNet [6] blocks that are well suited for image generation. Note that this architecture also contains attention layers for each CNN layer, facilitating image generation robust to any residual nonlinear information.

3.3. Datasets for Training

Our approach utilizes three frames to enhance the level of video understanding. Therefore, it is not possible to apply the conventional training dataset that is designed for two successive frames. As a result, we propose two new *augmented* datasets *Vimeo90k tridirectional dataset* and *Vimeo90k flying objects dataset* on-the-fly for the network training.

3.3.1 Vimeo90k tridirectional dataset

To train our model, we need at least five consecutive video frames where three are used as input, and two are used as the ground-truth interpolation frames. Since most frame interpolation datasets only provide frame triplets, they cannot be used for our model (our linear network version using triplet data is separately introduced in the experiments section).

Thus, we utilize the SEPTUPLET dataset provided by the Vimeo90k dataset [24]. Among the provided seven frames, we randomly select five consecutive frames for training on-the-fly. The 1st, 3rd, and 5th frames are used as inputs I_1, I_2, and I_3, while the 2nd and 4th frames are used as ground truth frames of $I_{1.5}$ and $I_{2.5}$. Vimeo90k is an accessible dataset used to train numerous video tasks due to its various content and dynamic motion. Since the Vimeo90k dataset contains various nonlinear and complex motion examples, it is a good fit for training nonlinear motion representation.

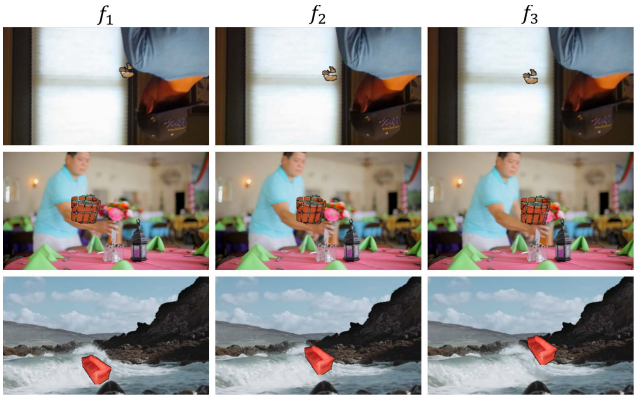

Figure 4. Samples of the flying objects dataset. The objects from the PASCAL VOC dataset are segmented and added on top of the Vimeo90k data samples on-the-fly.

3.3.2 Vimeo90k flying objects dataset

To the best of our knowledge, datasets explicitly containing small moving objects to facilitate training (and testing) do not exist. Therefore, to develop an approach robust to small object movement, we take a data-driven approach by augmenting the Vimeo90k tridirectional dataset. Our flying objects dataset may resemble the flying chairs dataset [10]. However, the flying chairs dataset was used for the sole purpose of training optical flow entirely with synthetic data, whereas our dataset is an additional tool for augmenting the Vimeo90k dataset for further robust inference.

Specifically, we segment 7,000 objects provided by the PASCAL VOC dataset [4]. Then, we select a random object and randomly resize it to either 64×64 or 32×32 resolution and overlay them on the five consecutive input frames. The objects are overlaid on the frames in a random direction, such that the objects have moved at least 32 pixels between frames, conveying a *flying object motion*. Thus, we term this augmented dataset as the Vimeo90k flying objects dataset. Some data examples are shown in Fig. 4. Our augmented flying objects dataset yields results that can express the interpolation of nonlinear motion as well as small object fast motion.

Please note that this augmented dataset was not used for reporting quantitative comparisons in the experiments using the Vimeo90k dataset for fair comparison.

3.4. Loss Function

3.4.1 Training with Vimeo90k tridirectional dataset

To train our model end-to-end, we use the L1 loss between the two output frames and two GT frames as follows:

$$L_g = \|I_{1.5} - O_{1.5}\|_1 + \|I_{2.5} - O_{2.5}\|_1 , \qquad (7)$$

where $O_{1.5}$ and $O_{2.5}$ are the estimated interpolated frames given inputs I_1, I_2, and I_3. This learning to reconstruct both of the intermediate frames can further facilitate the learning of nonlinear motion, rather than just either $O_{1.5}$ or $O_{2.5}$.

3.4.2 Training with Vimeo90k flying objects dataset

For utilizing the Vimeo90k flying objects dataset, we devise an additive loss term for the flying object local regions. This loss term is essentially the same as the L1 loss, but it is applied to the local image patch at which the flying object should be interpolated. Since we augment the Vimeo90k frames with flying objects, we have the information of the ground-truth object positions for every frame, thus we apply the L1 loss at these positions:

$$L_o = \|I^p_{1.5} - O^p_{1.5}\|_1 + \|I^p_{2.5} - O^p_{2.5}\|_1 , \qquad (8)$$

where $I^p_{1.5}$ denotes the local patch centered at the flying object position of $I_{1.5}$. The patch sizes were set to 64×64 for our implementation. Note that our network does not require such patches during testing. Our pipeline is fully convolutional and applicable to high-resolution videos. The final loss is the sum of the global and local object L1 losses with equal weights $L = L_g + L_o$.

3.5. Training Details

Our framework is implemented via the PyTorch library [17]. We train our network for approximately 5 days with four NVIDIA Titan Xp GPUs, using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, learning rate of 0.0001, mini-batch size of 4, for 50 epochs. The learning rate is set to decay to 10^{-5} after 20 epochs linearly. We utilize the 448×256 resolution images for training. To eliminate potential dataset bias, we also augment the training data on-the-fly by randomly reversing the frame order, applying horizontal and vertical flips to each frame.

We adopt the PWC-Net [22] as the backbone optical flow module. Since the flow module is initialized with the trained weights while the rest is initialized from scratch, training end-to-end from the start may propagate erroneous gradients to the optical flow module. Thus, we first train our model with a fixed optical flow module for the first epoch, then fine-tune for the rest of the epochs via end-to-end learning. This prevents the optical flow module from degradation during the early stages of training and allows a task-oriented flow learning [24] fit for nonlinear frame interpolation. Since a task-oriented flow learning approach is used, the optical flow module can be modified to optimize the network's performance or speed further.

4. Evaluation

With the trained models using the datasets we propose in Sec. 3.3, we test our network with various public and custom datasets. Although there are a handful of popular

Dataset	448 × 256		1344 × 768	
Metric	PSNR	SSIM	PSNR	SSIM
SepConv - L_f [16]	33.45	0.9509	31.81	0.9309
TOFlow [24]	**33.73**	0.9515	30.54	0.9190
IM-Net [18]	33.50	0.9473	33.11	**0.9436**
Ours	33.67	**0.9533**	**33.12**	0.9428

Table 1. Results on the original Vimeo90k (448 × 256) and super resolved (1344 × 768) versions (Best: red, runner-up: blue).

datasets including the Middlebury [1], Sintel [3], KITTI [5], UCF101 [21], and DAVIS [19], these datasets are mostly in low-resolution or fit to particular domains (e.g. driving scene) or synthetic. Thus, Peleg *et al*. [18] utilized the Vimeo90k [24] dataset for appropriate evaluation, while also creating a higher resolution version using an off-the-shelf SR algorithm [25] for further assessment. In our work, we adopt Vimeo90k, SMBV dataset [8], GoPro dataset [12], and our custom FHD dataset aiming for high resolution video interpolation. For quantitative experiments, we measure the interpolation quality of the outputs via the PSNR and SSIM metrics.

4.1. Vimeo90k Dataset

To demonstrate experiments on the same settings from Peleg *et al*. [18], we provide comparison results against state-of-the-art methods using the Vimeo90k dataset (448 × 256 pix.) as well as its high-resolution version (1344 × 768 pix.). Note that we test our model that is trained with *Vimeo90k tridirecional dataset* for fair comparison. The results are shown in Tab. 1.

We can see that our method outperforms the IM-Net [18] on the majority of metrics or at least comparable. Fig. 5 shows that our method can produce favorable or at least comparable results to the IM-Net. In particular, our method shows favorable reconstruction of the hand and sleeve patterns, while the IM-Net exhibits motion artifacts. We speculate that better reconstruction comes from our network processing the tridirectional information. That is, the third frame provides additional motion information in consensus with the otherwise bidirectional information, confirming the motion profile and thus leading to confidently construct the middle frame. The last row of Fig. 5 illustrates that bidirectional methods may fall short of challenging motion in high resolution, while our methods can reliably construct the middle frame.

However, the upsampled version of the Vimeo90k dataset is not a *real-world* high-resolution video, but instead can be thought of as a synthetic enhancement of the original data. Moreover, the SR algorithm of Yamanaka *et al*. [25] is a single image super-resolution (SISR) method which upsamples a single image without considering the temporal information of video frames. Thus, it may lead to discontinuities or temporal artifacts that are not consistent with real high-resolution video characteristics.

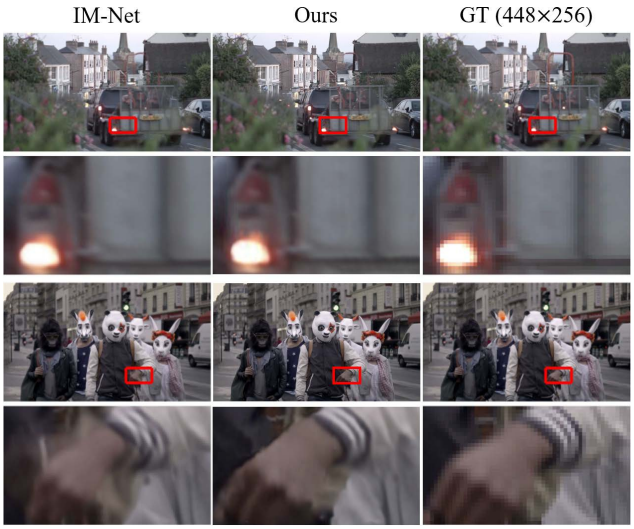

Figure 5. Visual comparison between IM-Net and our method. Our method is able to generate comparable or favorable results.

4.2. Real High-resolution Videos

Along with the results on Vimeo90k (and upsampled version), we conduct more extensive experiments on challenging sets of *real* high-resolution videos. Namely, we additionally conduct experiments on four test videos from the SMBV dataset provided by Jin *et al*. [8] (up to HD resolution), eleven videos from the GoPro dataset provided by Nah *et al*. [12] (all HD resolution), and several challenging FHD videos captured from a commercial camera. The custom FHD videos contain challenging scenarios with fast and nonlinear movement such as *table-tennis*, *water-balloon*, *candlelight*, and *tennis* scenes.

For the experiments with these datasets, we use the network trained with the *flying objects dataset* to confirm practicality of our proposed augmentation method. We also test our *Ours-Bidirectional* method that uses two frames as input to compare to our full version *Ours-Tridirectional* to demonstrate its benefits (explained in Fig. 3a).

4.2.1 The SMBV dataset

We present comparison results on the SMBV dataset in Tab. 2. It is worth noting that our method outperforms all state-of-the-art methods on average. Note that our method shows comparable results to the baseline methods on *Cars* and *Pedestrians* categories while outperforming significantly on *Basketball* and *Flag* categories. While *Cars* and *Pedestrians* categories convey mostly linear motion with constant speed, the *Basketball*, and *Flag* categories show complex nonlinear motion with abrupt changes in motion direction and speed (shown in Fig. 6 and 7). Thus, our

Category	Average		Cars		Pedestrians		Basketball		Flag	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SepConv - L_f [16]	32.3992	0.9512	36.8393	0.9777	32.1927	0.9573	32.2360	0.8776	30.5890	0.9628
SepConv - L_1 [16]	32.8667	0.9564	**37.3709**	**0.9809**	32.5070	**0.9596**	32.8521	0.8936	31.0434	0.9663
SuperSlomo [7]	30.9493	0.9108	31.3843	0.8787	29.5973	0.8803	31.4910	0.8430	31.1626	0.9625
TOFlow [24]	32.6144	0.9482	36.1871	0.9732	31.8370	0.9476	32.0790	0.8625	31.5702	0.9677
Ours-Bidirectional	32.6538	0.9553	36.0836	0.9760	32.0115	0.9539	32.7735	0.8937	31.3817	0.9685
Ours-Tridirectional	**33.4091**	**0.9593**	36.1567	0.9734	**32.7901**	0.9556	**34.0713**	**0.9086**	**32.2853**	**0.9718**

Table 2. Results on the SMBV dataset [8] (Best: red, runner-up: blue).

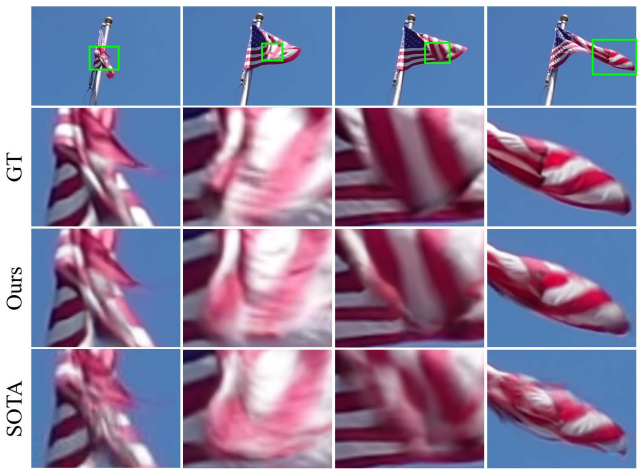

Figure 6. Visual comparison between state-of-the-art baselines and our method on the *Flag* category of SMBV dataset. For the bottom row, from left to right, results of SuperSlomo [7], TOFlow [24], SepConv - L_1, SepConv - L_f [16] are displayed.

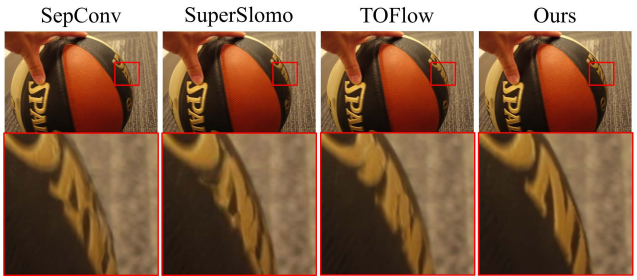

Figure 7. Frame interpolation results on a turning basketball. The results from SepConv - L_1 [16], SuperSlomo [7], TOFlow [24] and our approach are displayed.

method, especially for scenes containing complex movement, conveys a clear advantage over the baseline methods.

Although PSNR and SSIM are famous metrics to evaluate image reconstruction tasks, visual quality is essential to the frame interpolation task. Since we aim to produce quality frame interpolation for scenes containing complex movement, we provide extensive visual comparisons. Fig. 6 shows an example of a complex movement where a flag is fluttering in the wind. This is an extreme case of complex motion where linear motion compensation cannot adequately represent the underlying motion characteristics. Since our model explicitly takes the nonlinear motion into account, our model can convey such complex movements showing results with significant resemblance to the ground truth frame. Fig. 7 shows the interpolation results of a nonlinear motion of a turning basketball. Notice that the printing on the basketball is distorted for other baselines, whereas our method clearly reproduces the printing. For visual details, please refer to the supplementary video.

4.2.2 The GoPro dataset

We also conducted experiments on the GoPro dataset, as shown in Tab. 3. Our method outperforms the baselines in all metrics. It is worth noting that Ours-Bidirectional does not outperform the state-of-the-art methods, however, our full version containing the nonlinear flow estimation module is what gives our model the edge over the state-of-the-art methods, outperforming them.

4.2.3 A custom FHD dataset

Apart from using public datasets, to thoroughly evaluate our method on challenging video content, we collected several video clips with high resolution (FHD 1920×1080) containing complex movements and subtle visual phenomenon (e.g., fluid motion, abrupt deformation). For quantitative evaluation, we once again measure the PSNR and SSIM performances, as shown in Tab. 3. Despite the challenging videos, our method manages to outperform all methods in all metrics. Not only does our method demonstrate better performance in terms of the quantitative measure but also in terms of visual quality, as discussed next.

Fig. 8 shows the sequence photos (overlaid image of a small moving object) of a ping-pong ball flying fast. The figure overlays only the interpolated results obtained from the methods. This scene is also a challenging setting since a small object is moving across a non-homogeneous background. The baseline methods often fail to capture the correspondence between the small flying ping-pong ball, leav-

Dataset	GoPro [12]		FHD		#param.
Metric	PSNR	SSIM	PSNR	SSIM	(million)
SepConv - L_f [16]	37.2475	0.9792	33.3053	0.9583	21.6
SepConv - L_1 [16]	37.3856	0.9803	33.6792	0.9636	21.6
SuperSlomo [7]	36.3517	0.9674	32.0058	0.9455	19.8
TOFlow [24]	37.3925	0.9801	32.5766	0.9612	1.1
Ours-Bidirectional	37.3868	0.9802	33.5069	0.9640	10.3
Ours-Tridirectional	**37.5493**	**0.9804**	**34.5803**	**0.9671**	10.4

Table 3. Results on the GoPro [12] and our FHD datasets. We also provide the parameter count for the baselines.

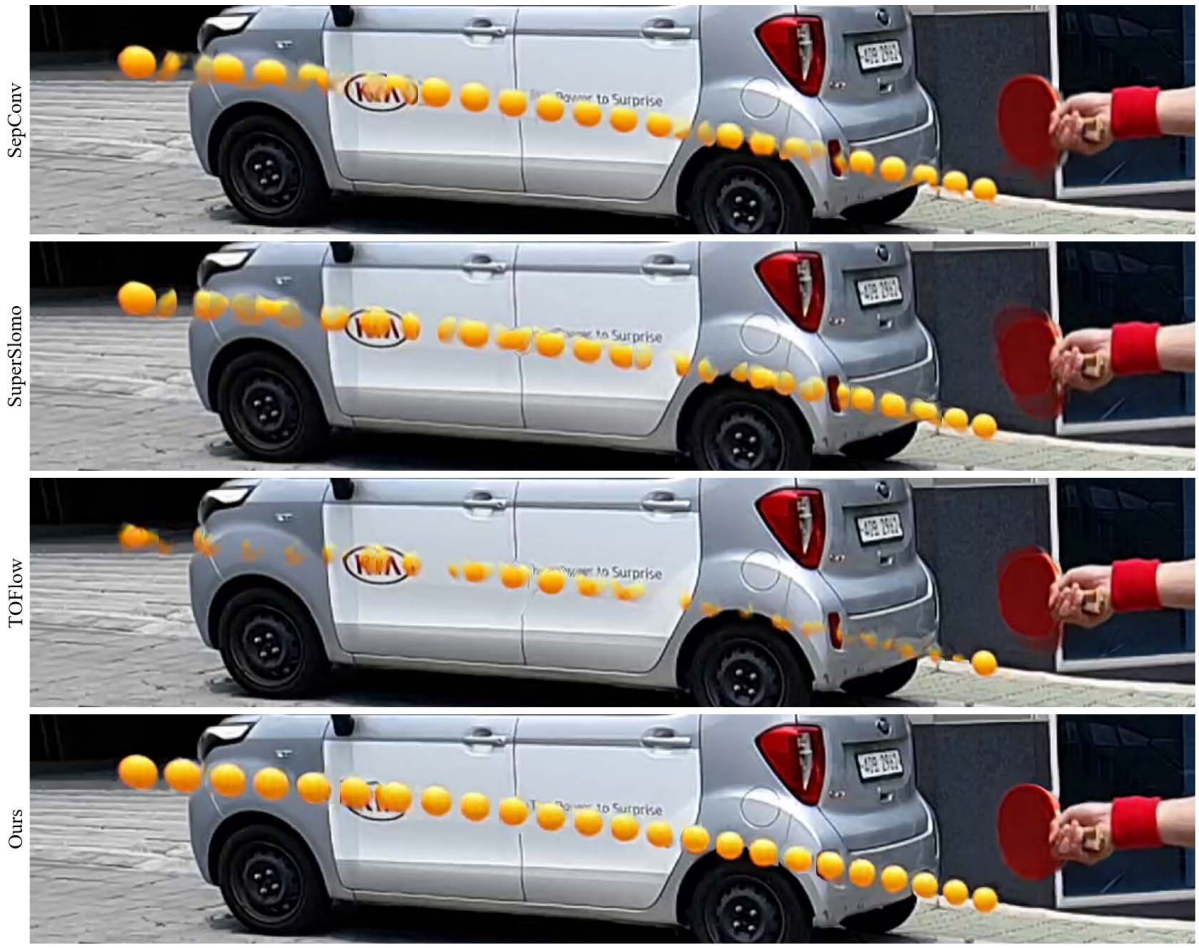

Figure 8. The sequence photo of interpolated frames on a fast moving ping-pong ball. Our results demonstrate successful reconstruction of the ping-pong ball trajectory while other baselines convey failed reconstruction.

ing a blank space at the supposed interpolation position and ghosting effects at the ball's given (input) positions. In contrast, our method successfully interpolates the ball for every consecutive frame without any ghosting artifacts.

5. Conclusion

Video frame interpolation has long been a classic video task that remains an active field of research due to its applicability to various video tasks. Although we have seen impressive advances in this field, we have only just begun exploring the inherent challenges of frame interpolation. This paper proposes a novel frame interpolation method which explicitly handles complex motion in videos via architecture design and a data-driven approach. Our method demonstrates superior interpolation quality for numerous challenging video content. We hope our approach is considered useful and contributes to solving the challenges of frame interpolation that lie ahead. We will release the dataset and code upon acceptance.

References

[1] 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 (IJCV)*, 92(1):1–31, 2011.

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3703–3712, 2019.

[3] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, pages 611–625. Springer, 2012.

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018.

[8] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 8112–8121, 2019.

[9] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4463–4471, 2017.

[10] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[11] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 498–507, 2018.

[12] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5437–5446, 2020.

[15] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[16] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[18] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2398–2407, 2019.

[19] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRVC*, 2012.

[22] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[23] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems*, pages 1645–1654, 2019.

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

[25] Jin Yamanaka, Shigesumi Kuwashima, and Takio Kurita. Fast and accurate image super resolution by deep cnn with skip connection and network in network. In *International Conference on Neural Information Processing*, pages 217–225. Springer, 2017.