IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 November 2022, accepted 13 February 2023, date of publication 6 March 2023, date of current version 9 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252911

RESEARCH ARTICLE

Video Frame Interpolation Based on Symmetric and Asymmetric Motions

WHAN CHOI[1], (Student Member, IEEE), YEONG JUN KOH[2], (Member, IEEE), AND CHANG-SU KIM[1], (Fellow, IEEE)

[1]School of Electrical Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, South Korea

Corresponding author: Chang-Su Kim (changsukim@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant NRF-2021R1A4A1031864 and Grant NRF-2022R1A2B5B03002310, in part by the Basic Science Research Program through the NRF funded by the Ministry of Education under Grant NRF-2022R1I1A3069113, and in part by Samsung Electronics Company Ltd. under Grant IO201214-08156-01.

ABSTRACT Video frame interpolation is the task to synthesize intermediate frames between consecutive frames to increase the frame rate. Recently, various deep-learning techniques have been proposed to interpolate intermediate frames more reliably. However, many existing methods use either symmetric (linear) or asymmetric (non-linear) schemes only to estimate motions for the warping process, resulting in unreliable interpolation results. In this paper, we propose a novel video frame interpolation network based on both symmetric and asymmetric motion-based warping modules, which can deal with linear and non-linear motions, as well as occlusions, effectively. The symmetric warping module estimates symmetric motions to generate intermediate frames, while the asymmetric one predicts asymmetric motions to address non-linear motions and occlusion problems. We combine symmetric and asymmetric warping results to reconstruct intermediate frames more reliably. We also develop the frame synthesis network to refine the combined warping results. Experimental results demonstrate that the proposed network outperforms state-of-the-art video interpolation algorithms and that the two types of warping modules work effectively in a complementary manner on various benchmark datasets.

INDEX TERMS Video frame interpolation, convolutional neural network, symmetric motion, asymmetric motion, motion estimation, frame synthesis, kernel-based approach, deformable convolution.

I. INTRODUCTION

Video frame interpolation aims to increase the frame rate of a video sequence by synthesizing intermediate frames between adjacent real frames. A low frame rate (or temporal resolution) of a video may cause temporal jittering, aliasing, and motion blur artifacts, degrading video quality as well as visual satisfaction. Hence, the video frame interpolation task is important to enhance the temporal resolution and visual experience by creating videos with high frame rates. It is widely used in numerous applications, such as video compression [1], slow-motion generation [2], frame rate-up conversion [3], [4], video enhancement [5], frame recovery in video streaming [6], [7], and novel view synthesis [8]. In recent years, many deep learning approaches have been proposed to interpolate video frames, but it is still challenging to reconstruct high-quality intermediate frames because of motion blur and non-linear motions such as occlusion and disappearance.

Learning-based video frame interpolation algorithms can be classified into two approaches: flow-based methods and kernel-based methods. By employing deep-learning-based optical flow estimators [9], [10], flow-based methods [11], [12], [13], [14] warp and blend consecutive input frames to yield reliable intermediate frames. Many flow-based methods generate intermediate frames by halving motion vectors

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

based on the symmetric motion assumption between two successive frames. The symmetric motion assumption is valid in the cases of linear motions with constant velocities such as typical background movements. However, it cannot cope with occlusions and non-linear movements properly and may yield degraded interpolation results in those regions. Moreover, flow-based methods are sensitive to optical flow errors, caused by occlusions, large displacement, and motion blur. Such optical flow errors lead to wrong reference positions, resulting in poor intermediate frames.

Kernel-based methods have been developed to synthesize intermediate frames using dynamic kernel weights. Recent kernel-based algorithms [15], [16], [17] estimate kernel weights and offset vectors and then generate intermediate frames based on deformable convolution [18], where the offset vectors are used as pseudo-motion vectors for determining matching pixels. Without the symmetric motion constraint, they estimate asymmetric bilateral offsets to determine reference positions for interpolation in two consecutive frames. These kernel-based methods, however, may be ineffective for synthesizing intermediate frames clearly and reducing motion blur artifacts on linear motions, since their bilateral offsets are predicted asymmetrically.

In this paper, we propose a novel video interpolation network, composed of a symmetric motion-based warping module, an asymmetric motion-based warping module, and a frame synthesis network. The proposed network extracts multi-scale features to obtain multi-scale kernel weights and offset vectors for video frame interpolation. Then, the symmetric motion-based and asymmetric motion-based warping modules interpolate an intermediate frame between two input frames based on symmetric and asymmetric offsets, respectively. Finally, the frame synthesis network generates a residual frame to refine the interpolated frame from the warping modules. Experimental results demonstrate that the proposed algorithm outperforms state-of-the-art video interpolation algorithms on various benchmark datasets.

To summarize, this work has the following three main contributions:

- Unlike existing video interpolators that exploit either symmetric or asymmetric motion model only, the proposed warping module estimate both symmetric and asymmetric motions to deal with both linear and non-linear motions between successive frames effectively.
- We propose the frame synthesis network, which generates residual intermediate frames, to refine warping results more reliably.
- The proposed network outperforms the state-of-the-arts on benchmark datasets, which contain videos with both linear and non-linear motions at various resolutions.

This paper is organized as follows: Section II reviews related work. Section III describes the proposed algorithm, and Section IV discusses its experimental results. Finally, Section V concludes this paper.

II. RELATED WORKS

The conventional motion-compensated frame interpolation algorithms have focused on estimating exact motion vectors between two consecutive frames and then generating an intermediate frame by halving those motion vectors. Many motion vector refinement techniques [19], [20], [21], [22], [23] have been proposed to estimate more precise motion vectors for video frame interpolation. Huang and Nguyen [19] developed a multi-stage refinement algorithm, which classifies motion vectors according to their reliability levels and analyzes the distribution of residual energies to merge motion blocks near motion boundaries effectively. Jacobson et al. [20] applied saliency techniques and segmentation methods to refine motion vectors. Jeong et al. [21] generated multiple motion hypotheses and determined the best pixel-wise motion hypothesis through label optimization. Zhang et al. [22] formed pixel intensities, representing spontaneous transition along a motion trajectory, across adjacent frames and designed a motion estimation algorithm based on polynomial approximation. Choi et al. [23] developed a MAP-based refinement algorithm, which repeatedly updates the motion vector of each block.

Flow-based video frame interpolation methods [11], [12], [13] have been developed by employing off-the-shelf optical flow estimators [9], [10]. Niklaus and Liu [11] predicted bi-directional optical flow utilizing PWC-Net [10] to perform forward warping and modified GridNet [24] to handle occlusions and non-linear motions. Bao et al. [12] developed an adaptive warping layer, which blends pixels adaptively using optical flow information [9]. Also, Bao et al. [13] proposed a depth-aware flow projection layer, which uses depth information [25], to predict an intermediate flow vector by combining bi-directional flow vectors [10]. Niklaus and Liu [14] used optical flow [10] to forward-warp input frames based on softmax splatting. These flow-based methods are, however, vulnerable to optical flow errors, and some of them require additional training data for optical flow estimation and incur additional training complexity.

It is worth pointing out that, in addition to video frame interpolation, flow-based warping has been adopted also in various image processing and computer vision tasks, including video object detection [26], video super-resolution [27], [28], and video translation [29], to exploit high temporal correlation in natural video sequences.

Some flow-based methods, which simultaneously estimate optical flow and interpolate intermediate frames using the estimated flow information, have adopted an end-to-end video frame interpolation framework. Liu et al. [30] developed an end-to-end fully convolutional network to estimate optical flow by exploiting the information in a 3D spatiotemporal neighborhood of each output pixel. Jiang et al. [2] estimated bi-directional flow vectors to approximate intermediate flow vectors for frame interpolation. Liu et al. [31] proposed a cycle consistency loss to supervise their network to generate intermediate frames. Park et al. [32] designed a symmetric bilateral motion

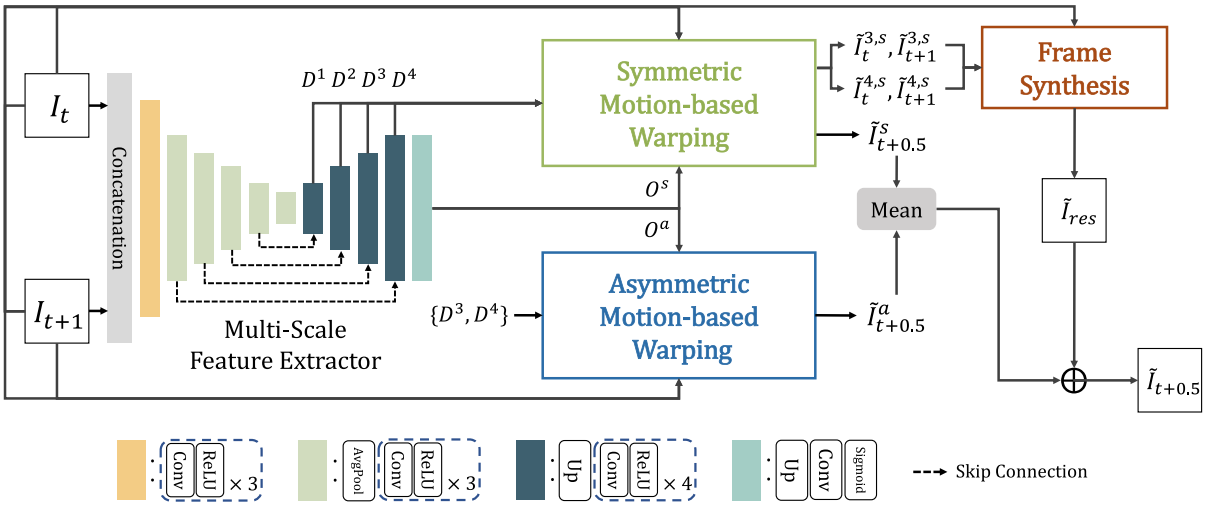


FIGURE 1. An overview of the proposed video frame interpolation network, which consists of 4 different modules: multi-scale feature extractor, symmetric and asymmetric motion-based warping modules, and frame synthesis network.

network to estimate intermediate motion vectors based on the linear motion constraint and used the bilateral motion vectors for intermediate frame interpolation.

The kernel-based approach has been developed to synthesize intermediate frames by convolving consecutive input frames with dynamic kernel weights. Niklaus et al. [33] estimated pixel-wise spatially adaptive 2D convolution kernels with large window sizes but required high memory complexity for the pixel-wise parameters. To reduce the high memory complexity, Niklaus et al. [34] employed adaptive separable convolution by separating 2D kernels into 1D horizontal and vertical kernels. Since these algorithms [33], [34] do not use motion information explicitly, they cannot deal with motions larger than pre-defined kernel sizes properly.

Another kernel-based approach has been attempted to predict kernel weights and motion information simultaneously [15], [16], [17], [35]. Peleg et al. [35] estimated offsets via motion classification and convolved input frames based on the estimated kernels and offsets. Recent kernel-based methods [15], [16], [17] have employed deformable convolution to estimate kernel weights and offsets to warp two consecutive frames. They regarded offsets as pseudo-motions to determine reference positions and convolved two input frames at the reference positions with the estimated kernel weights. Moreover, Choi et al. [17] used a multi-scale warping module to cope with both small and large motions. However, these kernel-based methods estimate only asymmetric offsets, even though symmetric offsets are effective for linear motions.

III. PROPOSED ALGORITHM

Figure 1 shows the structure of the proposed video frame interpolation network. The proposed network takes two consecutive input frames I_t and I_{t+1}, where t is a frame index, and produces an intermediate frame $\tilde{I}_{t+0.5}$. The proposed network consists of the feature extractor, the symmetric and asymmetric motion-based warping modules, and the frame synthesis network.

TABLE 1. Specification of the multi-scale feature extractor: $H \times W$ denote the spatial resolution of an input image.

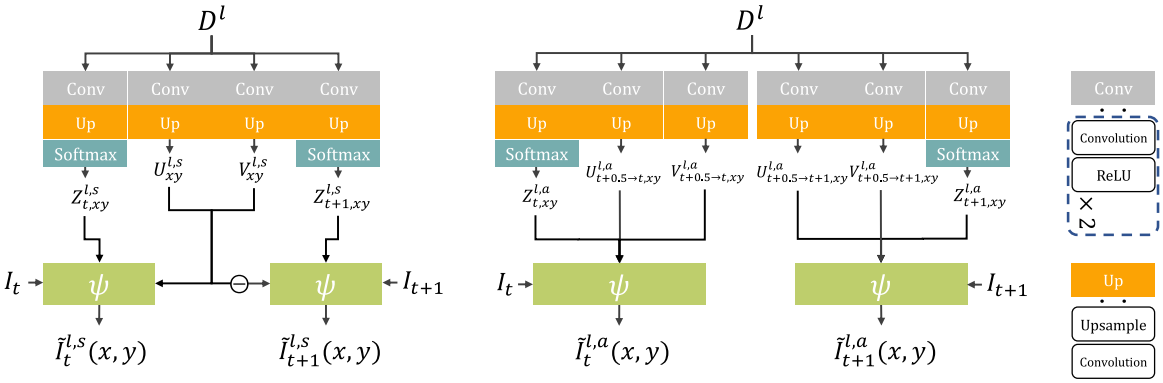
Operator	Output Resolution	Output Channel	Feature
Encoder	$H/32 \times W/32$	512	-
Up-sample block	$H/16 \times W/16$	256	D^1
Convolution block	$H/16 \times W/16$	256	-
Up-sample block	$H/8 \times W/8$	128	D^2
Convolution block	$H/8 \times W/8$	128	-
Up-sample block	$H/4 \times W/4$	64	D^3
Convolution block	$H/4 \times W/4$	64	-
Up-sample block	$H/2 \times W/2$	64	D^4
Convolution block	$H/2 \times W/2$	64	-

A. FEATURE EXTRACTOR

The feature extractor takes a concatenation of I_t and I_{t+1} and produces multi-scale features to perform symmetric and asymmetric motions-based warping. As shown in Figure 1, we use the U-net architecture [17], [36] as the backbone for the feature extractor, which consists of an encoder and a decoder with skip connections. From the output of the encoder, the decoder extracts multi-scale features and passes them to the symmetric motion-based and asymmetric motion-based warping modules. The decoder contains four blocks, each of which consists of an up-sample layer and four sets of a 3×3 convolution layer with the ReLU activation. The up-sample layer performs bilinear interpolation with a scale factor of 2. Then, from each of the decoder blocks, we extract multi-scale features $\mathcal{D} = \{D^1, \ldots, D^4\}$, where D^l is the output feature of the lth block. The specification of the multi-scale feature extractor is summarized in Table 1.

B. MOTION-BASED WARPING

The proposed network contains both symmetric and asymmetric warping modules to interpolate the intermediate frame between I_t and I_{t+1}. Both modules perform multi-scale



(a) Symmetric motion-based warping (b) Asymmetric motion-based warping

FIGURE 2. **(a) Symmetric and (b) asymmetric motion-based warping processes for pixel position (x, y) at scale l.**

warping based on deformable convolution. The feature D^l is converted to kernel weights and horizontal and vertical offsets for deformable convolution. Let $Z^l \in \mathbb{R}^{H^l \times W^l \times k^2}$ denote kernel weights and $U^l \in \mathbb{R}^{H \times W \times k^2}$ and $V^l \in \mathbb{R}^{H \times W \times k^2}$ denote horizontal and vertical offsets, respectively, extracted from D^l. Here, $H \times W$ is the spatial resolution, and the kernel size k is set to 5.

We use Z^l, U^l, and V^l for the warping of an image I (either I_t or I_{t+1}). The interpolation for pixel position (x, y) is obtained by the warping function ψ, which is defined as

$$\begin{aligned}\tilde{I}^l(x,y) &= \psi(I, Z^l_{xy}, U^l_{xy}, V^l_{xy}) \\ &= \sum_{i=-\frac{(k-1)}{2}}^{\frac{(k-1)}{2}} \sum_{j=-\frac{(k-1)}{2}}^{\frac{(k-1)}{2}} Z^l_{xy}(i,j) \\ &\quad \times I(x+i+U^l_{xy}(i,j), y+j+V^l_{xy}(i,j)) \end{aligned} \quad (1)$$

where $Z^l_{xy} \in \mathbb{R}^{k\times k}$, $U^l_{xy} \in \mathbb{R}^{k\times k}$, and $V^l_{xy} \in \mathbb{R}^{k\times k}$ are the kernel weights, horizontal offsets, and vertical offsets for pixel (x, y), respectively. To compute $\tilde{I}^l(x, y)$ in Eq. (1), we adopt deformable convolution in [18]. Also, considering that offset values, $U^l_{xy}(i, j)$ and $V^l_{xy}(i, j)$, may not be integers, we compute $I(x+i+U^l_{xy}(i,j), y+j+V^l_{xy}(i,j))$ using bilinear interpolation.

1) SYMMETRIC MOTION-BASED WARPING

The symmetric warping module estimates horizontal and vertical offsets symmetrically to consider linear movements between adjacent frames. As shown in Figure 2(a), both warping processes for I_t and I_{t+1} share offsets with the same absolute value but opposite signs. For each scale l, intermediate frames $\tilde{I}^{l,s}_t$ and $\tilde{I}^{l,s}_{t+1}$ for I_t and I_{t+1}, respectively, are defined as

$$\begin{aligned}\tilde{I}^{l,s}_t(x,y) &= \psi(I_t, Z^{l,s}_{t,xy}, U^{l,s}_{xy}, V^{l,s}_{xy}), \\ \tilde{I}^{l,s}_{t+1}(x,y) &= \psi(I_{t+1}, Z^{l,s}_{t+1,xy}, -U^{l,s}_{xy}, -V^{l,s}_{xy}). \end{aligned} \quad (2)$$

Also, to take advantage of both coarse-scale and fine-scale information, the symmetric warping module produces the warping results $\tilde{I}^s_t$ and $\tilde{I}^s_{t+1}$ with learnable adaptive weights $\{w^{l,s}_t\}^4_{l=1}$ and $\{w^{l,s}_{t+1}\}^4_{l=1}$, respectively, which are given by

$$\tilde{I}^s_t = \sum_{l=1}^{4} w^{l,s}_t \tilde{I}^{l,s}_t, \quad \tilde{I}^s_{t+1} = \sum_{l=1}^{4} w^{l,s}_{t+1} \tilde{I}^{l,s}_{t+1} \quad (3)$$

where $\sum^4_{l=1} w^{l,s}_t = 1$ and $\sum^4_{l=1} w^{l,s}_{t+1} = 1$.

We combine the forward warped frame $\tilde{I}^s_t$ and the backward warped frame $\tilde{I}^s_{t+1}$ with a learnable weight map $O^s \in \mathbb{R}^{H\times W}$ to obtain a symmetrically warped frame

$$\tilde{I}^s_{t+0.5} = O^s \odot \tilde{I}^s_t + (1 - O^s) \odot \tilde{I}^s_{t+1} \quad (4)$$

where $\odot$ denotes the Hadamard product. To obtain O^s, D^4 passes through one up-sample block and one 3×3 convolution layer with the sigmoid activation, satisfying the constraint $0 \le O^s(x, y) \le 1$, as done in [16] and [17].

2) ASYMMETRIC MOTION-BASED WARPING

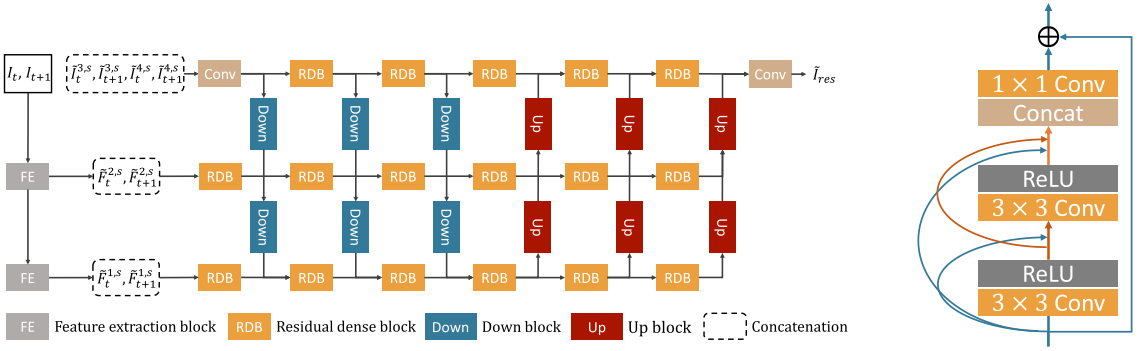
The asymmetric warping module predicts horizontal and vertical offsets asymmetrically to deal with occlusions and non-linear movements as done in other kernel-based methods [15], [16], [17]. To this end, two sets of horizontal and vertical offsets $(U^{l,a}_{t+0.5\to t,xy}, V^{l,a}_{t+0.5\to t,xy})$ and $(U^{l,a}_{t+0.5\to t+1,xy}, V^{l,a}_{t+0.5\to t+1,xy})$ are extracted from different sets of convolution and upsample blocks in Figure 2(b). To this end, asymmetric motion-based warping results $\tilde{I}^{l,a}_t$ and $\tilde{I}^{l,a}_{t+1}$ at scale l are obtained by

$$\begin{aligned}\tilde{I}^{l,a}_t(x,y) &= \psi(I_t, Z^{l,a}_{t,xy}, U^{l,a}_{t+0.5\to t,xy}, V^{l,a}_{t+0.5\to t,xy}), \\ \tilde{I}^{l,a}_{t+1}(x,y) &= \psi(I_{t+1}, Z^{l,a}_{t+1,xy}, U^{l,a}_{t+0.5\to t+1,xy}, V^{l,a}_{t+0.5\to t+1,xy}). \end{aligned} \quad (5)$$

For the asymmetric motion-based warping, we use only two scale features D^3 and D^4 to focus on fine-scale information. As in the symmetric warping, we combine warping results with trainable weights to obtain asymmetric warping results

$$\tilde{I}^a_t = \sum_{l=3}^{4} w^{l,a}_t \tilde{I}^{l,a}_t, \quad \tilde{I}^a_{t+1} = \sum_{l=3}^{4} w^{l,a}_{t+1} \tilde{I}^{l,a}_{t+1} \quad (6)$$

where $\sum^4_{l=3} w^{l,a}_t = 1$ and $\sum^4_{l=3} w^{l,a}_{t+1} = 1$. Also, we obtain an asymmetrically warped frame $\tilde{I}^a_{t+0.5}$ by combining $\tilde{I}^a_t$ and



(a) Frame synthesis network (b) Residual dense block

FIGURE 3. The detailed architecture of (a) the frame synthesis network and (b) the residual dense block. The frame synthesis network generates a residual frame by processing multi-scale warping features and multi-scale warping frames.

$\tilde{I}^a_{t+1}$, which is given by

$$\tilde{I}^a_{t+0.5} = O^a \odot \tilde{I}^a_t + (1 - O^a) \odot \tilde{I}^a_{t+1}. \tag{7}$$

A learnable weight map O^a is obtained similarly to O^s in (4).

C. FRAME SYNTHESIS

We adopt the GridNet structure [24] for the frame synthesis network. Figure 3(a) represents the architecture of the frame synthesis network. The frame synthesis network contains several residual dense blocks (RDBs) [37], each of which has two sets of a 3×3 convolution layer with the ReLU activation, as shown in Figure 3(b). Also, the frame synthesis network takes multi-scale warping features and multi-scale warping frames to provide a residual intermediate frame. The warping features $\tilde{F}^{l,s}_t$ and $\tilde{F}^{l,s}_{t+1}$ at scale l are defined as

$$\begin{aligned}\tilde{F}^{l,s}_t(x,y) &= \psi(F^l_t, Z^{l,s}_{t,xy}, U^{l,s}_{xy}, V^{l,s}_{xy}),\\ \tilde{F}^{l,s}_{t+1}(x,y) &= \psi(F^l_{t+1}, Z^{l,s}_{t+1,xy}, -U^{l,s}_{xy}, -V^{l,s}_{xy})\end{aligned} \tag{8}$$

where F^l_t and F^l_{t+1} are frame features, extracted from I_t and I_{t+1} through two feature extraction blocks. Each feature extraction block has two sets of a 3×3 convolution layer with the PReLU activation. Then, for multi-scale warping features, the concatenation of $\tilde{F}^{1,s}_t$ and $\tilde{F}^{1,s}_{t+1}$ and the concatenation of $\tilde{F}^{2,s}_t$ and $\tilde{F}^{2,s}_{t+1}$ are separately input to RDBs. Also, the concatenation of $\tilde{I}^{3,s}_t$, $\tilde{I}^{3,s}_{t+1}$, $\tilde{I}^{4,s}_t$, and $\tilde{I}^{4,s}_{t+1}$ is fed into a convolution layer as input. To mix these inputs, down-sampling and up-sampling are performed to equalize the spatial resolution. The frame synthesis network yields the residual $\tilde{I}_{\text{res}}$. Finally, the intermediate frame $\tilde{I}_{t+0.5}$ is synthesized by

$$\tilde{I}_{t+0.5} = \frac{\tilde{I}^s_{t+0.5} + \tilde{I}^a_{t+0.5}}{2} + \tilde{I}_{\text{res}}. \tag{9}$$

D. IMPLEMENTATION DETAILS

The proposed network is trained in an end-to-end manner based on a loss function, given by

$$\mathcal{L} = \mathcal{L}_r + 0.01\mathcal{L}_s \tag{10}$$

where $\mathcal{L}_r$ is the reconstruction loss and $\mathcal{L}_s$ is the smoothness loss. $\mathcal{L}_r$ is defined as the Charbonnier loss [38] between the predicted intermediate frame $\tilde{I}_{\text{out}}$ and the ground-truth I_{gt}, which is given by

$$\mathcal{L}_r = \rho(\tilde{I}_{\text{out}} - I_{\text{gt}}) \tag{11}$$

where $\rho(x) = (x^2+\epsilon^2)^{1/2}$ and $\epsilon = 0.001$. Also, $\mathcal{L}_s$ is the loss to encourage neighboring pixels to have similar motions,

$$\begin{aligned}\mathcal{L}_s = \sum_{l=1}^{4} \{&\rho(\nabla_x(\mathcal{P}_{\text{avg}}(Z^l \odot U^l)))\\ &+ \rho(\nabla_y(\mathcal{P}_{\text{avg}}(Z^l \odot U^l)))\\ &+ \rho(\nabla_x(\mathcal{P}_{\text{avg}}(Z^l \odot V^l)))\\ &+ \rho(\nabla_y(\mathcal{P}_{\text{avg}}(Z^l \odot V^l)))\}\end{aligned} \tag{12}$$

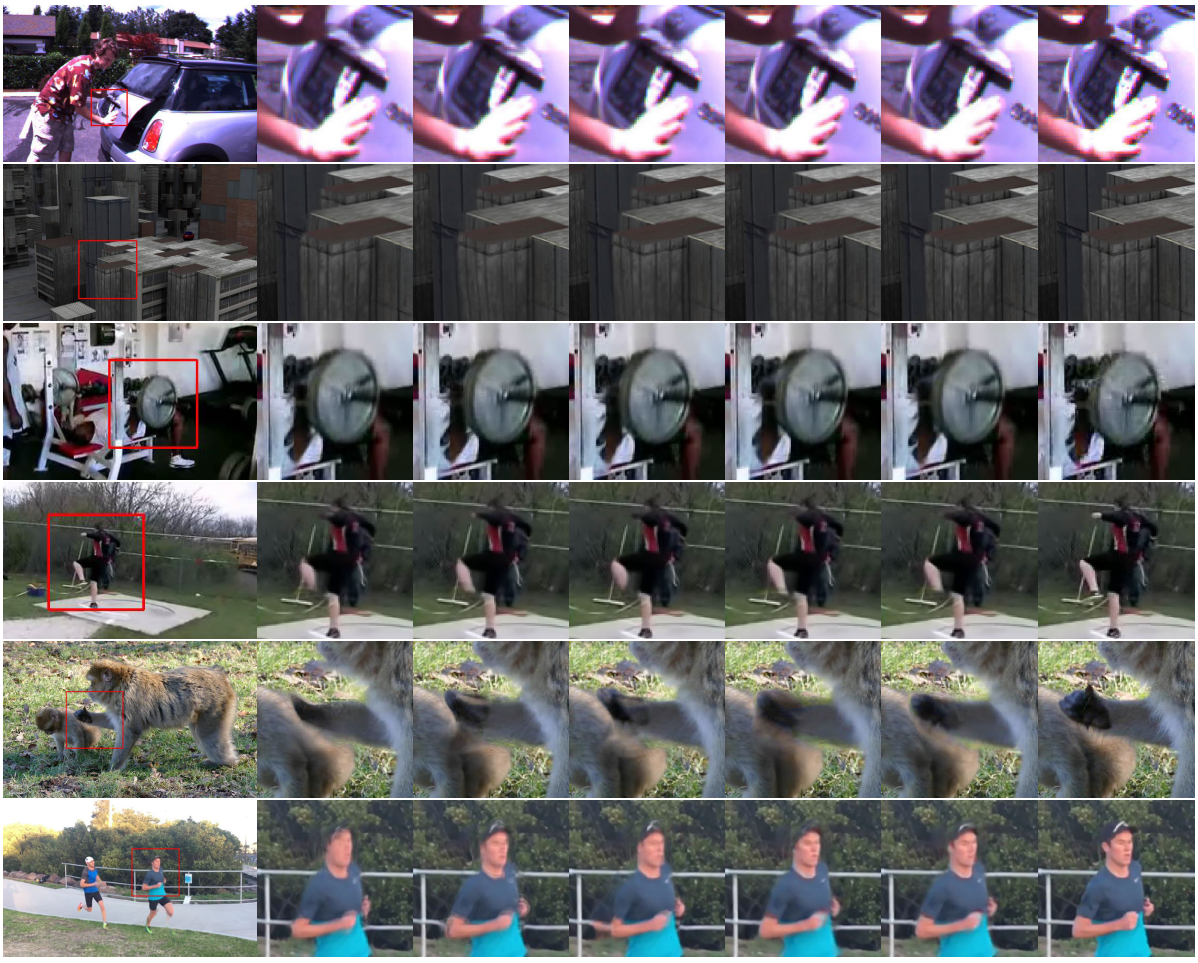
where $\mathcal{P}_{\text{avg}}$ is the average pooling function along the channel axis. Also, ∇_x and ∇_y denote partial derivatives in the horizontal and vertical directions, respectively.

We train the proposed network using the AdaMax optimizer [39]. We use hyper-parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initialized to 0.001 and then halved every 20 epochs. The training is iterated for 80 epochs with an RTX 2080Ti GPU.

IV. EXPERIMENTAL RESULTS

A. DATASETS AND METRICS

For training, we use the Vimeo90K [5] dataset, which is widely used for assessing various video frame interpolation methods [2], [16], [17], [32]. From Vimeo90K, we select the same 73,171 triplets of frames of resolution 448 × 256 as done in [16] and [17] for a fair comparison. For data augmentation, the triplets are randomly cropped with a 256 × 256 size and then randomly flipped horizontally or vertically. For evaluation, we use the same test sets as [16] and [17]: the 12 sequences in Middlebury [13] and randomly sampled sequences from the UCF101 and DAVIS datasets. These test sets include consecutive frames of various resolutions, which experience both linear and non-linear motions. For the quantitative assessment of video frame interpolation, we use the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) metrics.



(a) V1 (b) V2 (c) V3 (d) V4 (e) V5 (proposed) (f) GT

FIGURE 4. Qualitative comparison of settings V1∼V4 with the proposed algorithm (setting V5).

B. ABLATION STUDIES

First, we analyze the efficacy of three components of the proposed network: symmetric motion-based warping (SMW), asymmetric motion-based warping (AMW), and frame synthesis network (FSN). Table 2 summarizes the video frame interpolation results in five settings. In settings V1 and V2, either symmetric motion-based warping or asymmetric motion-based warping is employed only, respectively. In setting V3, both symmetric motion and asymmetric motion-based warping processes are performed without the frame synthesis network. Setting V4 adopts the symmetric motion-based warping only, but uses the frame synthesis network to refine intermediate frames. Finally, setting V5 employs all components in the proposed network.

As listed in Table 2, the proposed network (V5) provides the best performance on all datasets with large margins — 0.69dB on Middlebury, 0.26dB on UCF101 and 0.58dB on DAVIS — against setting V1. By comparing V1 and V2, we see that the asymmetric motion-based warping module interpolates middle frames more precisely on UCF101 and DAVIS, but not on Middlebury, which experiences fewer occlusions. This confirms that asymmetric motion-based warping is more robust to occlusions. By employing both symmetric and asymmetric warping modules, V3 outperforms both V1 and V2 on Middlebury and DAVIS. Settings V4 and V5 achieve the second-best and best results, respectively, using the frame synthesis network, which indicates that the frame synthesis network provides residual intermediate frames effectively to refine warping results.

Figure 4 shows qualitative comparison results of the proposed network with settings V1∼V4 on the Middlebury, UCF101, and DAVIS datasets. V1∼V3 yield blurry interpolation results on detailed patterns, such as the numbers on the license plate in the 1st row and the lines of the building in

TABLE 2. Ablation studies for the proposed components.

				Middlebury	UCF101	DAVIS
Setting	SMW	AMW	FSN	PSNR(↑)	PSNR(↑)	PSNR(↑)
V1	✓			36.19	35.04	26.89
V2		✓		35.96	35.13	27.06
V3	✓	✓		36.31	35.12	27.08
V4	✓		✓	<u>36.80</u>	<u>35.18</u>	<u>27.30</u>
V5 (Proposed)	✓	✓	✓	**36.88**	**35.30**	**27.47**

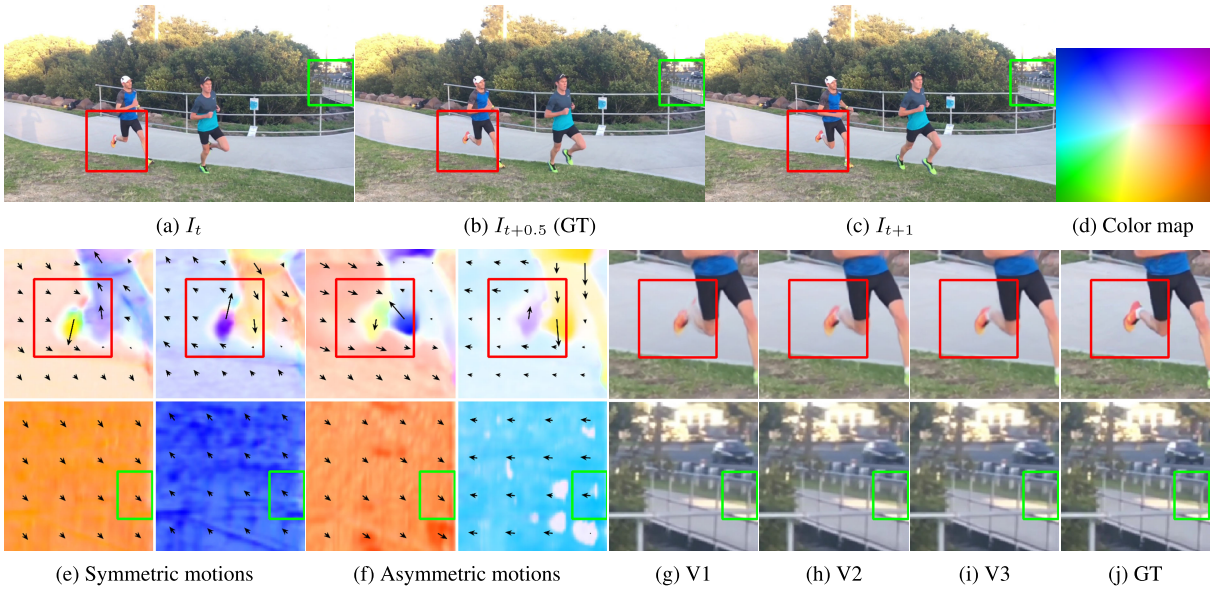


FIGURE 5. Qualitative comparison of settings V1∼V3 with the visualization of motion offsets.

the 2nd row. Also, V1 fails to reconstruct fast-moving objects faithfully, for example, the athlete's arm in the 3rd row, the person in the 4th row, the monkey's hand in the 5th row, and the runner's upper body in the 6th row. In contrast, V4 and the proposed network reconstruct visually more clear and more precise interpolation results on those details. However, V4 has the limitation of producing blurry results on fast and large moving objects in the 5th and 6th rows. These results show that all components of the proposed network are essential for reconstructing fine details faithfully and dealing with fast and large movements of objects reliably.

Figure 5 visualizes the motion offsets for interpolating two regions (red and green boxes in Figure 5(b)) to demonstrate the effectiveness of combining symmetric and asymmetric motions. The red box includes a runner's legs with non-linear motions, while the green box contains a background fence with linear motions. Note that settings V1 and V2 are based on the symmetric and asymmetric motion-based warping, respectively. For the motion visualization in Figure 5(e) and (f), we use offsets $(U^{4,s}, V^{4,s})$ and $(-U^{4,s}, -V^{4,s})$ obtained by setting V1 for symmetric motions, while $(U^{4,a}_{t+0.5\to t}, V^{4,a}_{t+0.5\to t})$ and $(U^{4,a}_{t+0.5\to t+1}, V^{4,a}_{t+0.5\to t+1})$ by setting V2 for asymmetric motions. It can be observed that the symmetric motions in Figure 5(e) represent the linear motions of the fence more reliably, while the asymmetric ones in Figure 5(f) do the non-linear motions of the legs more accurately.

In Figure 5(g), setting V1 yields blur artifacts on the leg, since its symmetric warping cannot deal with non-linear motions properly. In contrast, in Figure 5(h), setting V2 provides a more faithful warping result on the leg by estimating asymmetric offsets. For the fence in the green box, setting V1 yields better results, more similar to GT in terms of the fence thickness and sharpness, than setting V2 does. This is because the symmetric warping provides more robust results on regions with linear motions. Also, note that setting V3, which adopts both symmetric and asymmetric warping, provides high-quality results on both the leg and the fence, which indicates that the joint usage of symmetric and asymmetric motions helps to improve the video frame interpolation performance.

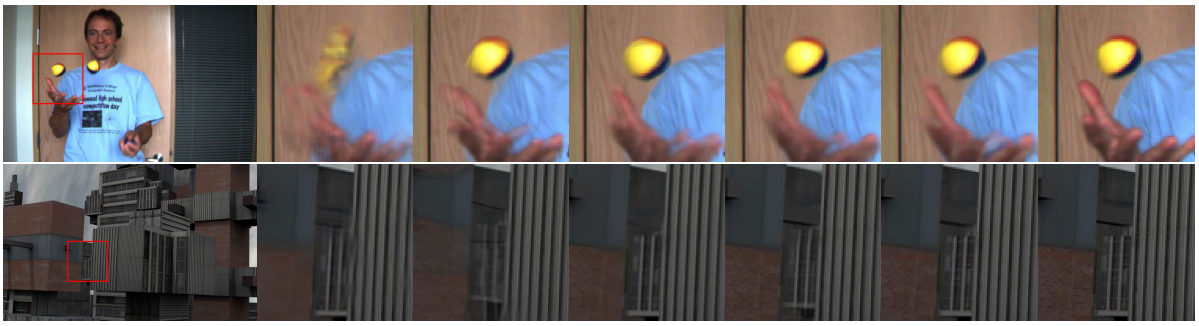
TABLE 3. Comparison of the proposed algorithm with conventional video frame interpolation algorithms.

	Middlebury		UCF101		DAVIS	
	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)
Overlapping	27.97	0.879	30.45	0.935	21.92	0.740
Phase [40]	31.12	0.933	32.45	0.953	23.47	0.800
MIND [41]	31.37	0.943	32.44	0.963	25.57	0.852
SepConv-$\mathcal{L}_1$ [34]	35.52	0.977	34.74	0.973	26.26	0.861
DVF [30]	34.24	0.971	34.47	0.972	25.88	0.858
SuperSlomo [2]	34.23	0.972	34.06	0.970	25.70	0.858
AdaCoF [16]	35.72	0.978	35.06	0.974	26.64	0.868
BMBC [32]	36.79	**0.984**	35.08	0.974	26.99	0.884
MSW [17]	36.12	0.980	35.20	0.974	27.09	0.877
Proposed	**36.88**	**0.984**	**35.30**	**0.974**	**27.47**	**0.887**

To summarize, we quantitatively and qualitatively verify that both symmetric and asymmetric warping modules contribute to synthesizing intermediate frames precisely. As shown in Table 2, V3 outperforms V1 and V2 by employing both symmetric and asymmetric motion-based warping modules. Also, Figure 5 provides a detailed qualitative analysis of the warping results for symmetric and asymmetric motions to demonstrate that the combination of symmetric and asymmetric motions is essential to deal with both linear and non-linear motions. Finally, settings V4 and V5 in Table 2 show that the proposed frame synthesis network refines warping results faithfully.

C. COMPARISON WITH STATE-OF-THE-ARTS

Table 3 compares the proposed network with existing video frame interpolation algorithms — Phase [40], MIND

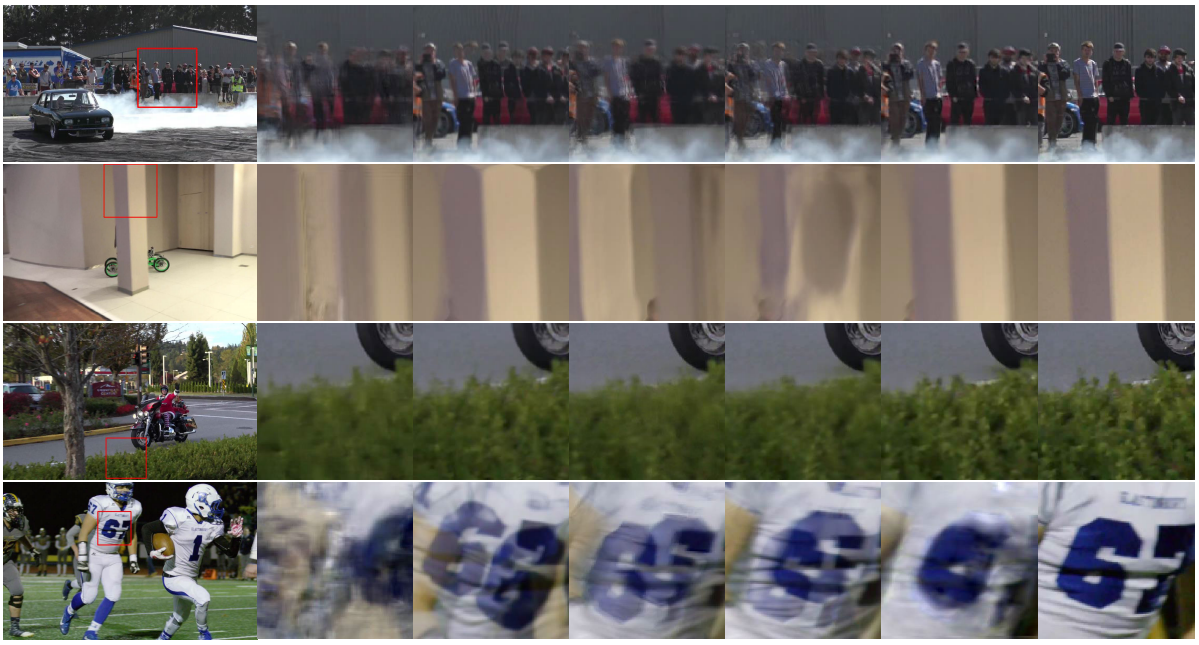


(a) SepConv (b) SuperSlomo (c) AdaCoF (d) MSW (e) Proposed (f) GT

FIGURE 6. Qualitative comparison of the proposed algorithm with the existing algorithms on the Middlebury dataset.

(a) SepConv (b) SuperSlomo (c) AdaCoF (d) MSW (e) Proposed (f) GT

FIGURE 7. Qualitative comparison of the proposed algorithm with the existing algorithms on the UCF101 dataset.



(a) SepConv (b) SuperSlomo (c) AdaCoF (d) MSW (e) Proposed (f) GT

FIGURE 8. Qualitative comparison of the proposed algorithm with the existing algorithms on the DAVIS dataset.

[41], SepConv-$\mathcal{L}_1$ [34], DVF [30], SuperSlomo [2], AdaCoF [16], BMBC [32], and MSW [17] — on the Middlebury, UCF101, and DAVIS datasets. The scores of the existing algorithms in Table 3 are from [17]. For the scores of BMBC, we use the official source codes, provided by its authors. The proposed network outperforms the existing algorithms with meaningful PSNR gains on all three datasets.

In Table 3, kernel-based methods, SepConv-$\mathcal{L}_1$, AdaCoF, and MSW, yield higher PSNR and SSIM scores than flow-based ones, DVF and SuperSlomo. This is because the kernel-based methods estimate asymmetric motions, which

can deal with non-linear motions or occlusions, whereas DVF and SuperSlomo depend on symmetric motions and are vulnerable to optical flow errors. In contrast, another flow-based method, BMBC, addresses optical flow errors based on the dynamic local blending filters, which compensate for motion inaccuracies, resulting in the second-best performance. Notice that the proposed network surpasses MSW with large margins (more than 0.3dB in PSNR) — 0.76dB on Middlebury and 0.38dB on DAVIS. Also, the proposed network outperforms BMBC, especially on the UCF101 and DAVIS datasets, which include video sequences containing more non-linear motions and occlusions than the Middlebury dataset. This indicates that the combination of the symmetric-based and asymmetric-based warping modules in the proposed network is effective. Also, the highest performance on Middlebury shows that the proposed frame synthesis network is essential for refining the intermediate results.

Figure 6 compares frame interpolation results of the proposed network with those of the existing algorithms on Middlebury qualitatively. The proposed network synthesizes the shapes of the left hand and the ball in the 1st row and the shape of the building in the 2nd row more faithfully than the other algorithms do. Figure 7 shows interpolation results on the UCF101 dataset. In the 1st row, the proposed network precisely reconstructs the movement of the pitcher's legs. Also, in the 2nd row, the proposed network synthesizes the athlete's body more faithfully. These results verify that the proposed algorithm deals with fast-moving objects robustly and effectively.

Finally, Figure 8 provides qualitative comparisons on the DAVIS dataset. From the 1st to the 3rd rows, videos contain fast-moving components, such as crowds, pillars, and bushes. In those regions, the proposed algorithm generates sharp interpolation results, whereas the other algorithms yield blurry or deformed interpolation results. Finally, in the 4th row, the proposed network reconstructs the uniform numbers of the fast-moving player with less blur than the other video frame interpolation algorithms do.

V. CONCLUSION

In this paper, we proposed a video interpolation network based on both symmetric and asymmetric motion-based warping modules, which can deal with linear and non-linear motions and occlusions effectively. The proposed network estimates both symmetric and asymmetric motions and performs motion-based warping to obtain symmetric and asymmetric warping results. It then combines two warping results and refines them using the frame synthesis network. Experimental results demonstrated that the proposed algorithm outperforms state-of-the-art video interpolation algorithms on various benchmark datasets. For future work, we will develop a video interpolation network for high-resolution video sequences such as 4K or higher-resolution videos. To interpolate such high-resolution sequences reliably and efficiently, we will design a lightweight network to perform symmetric and asymmetric warping jointly in a single module and also reduce the model size of the frame synthesis network.

REFERENCES

[1] G. Lu, X. Zhang, L. Chen, and Z. Gao, "Novel integration of frame rate up conversion and HEVC coding based on rate-distortion optimization," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 678–691, Feb. 2018.

[2] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[3] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers, "PhaseNet for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 498–507.

[4] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao, "High-order model and dynamic filtering for frame rate up-conversion," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3813–3826, Aug. 2018.

[5] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, 2019.

[6] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Modeling and optimization of high frame rate video transmission over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2713–2726, Apr. 2016.

[7] H. Choi and I. V. Bajić, "Deep frame prediction for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1843–1855, Jul. 2020.

[8] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "DeepStereo: Learning to predict new views from the world's imagery," in *Proc. CVPR*, 2016, pp. 5515–5524.

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[10] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[11] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1701–1710.

[12] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Mar. 2021.

[13] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3703–3712.

[14] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5437–5446.

[15] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10607–10614.

[16] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5316–5325.

[17] W. Choi, Y. J. Koh, and C.-S. Kim, "Multi-scale warping for video frame interpolation," *IEEE Access*, vol. 9, pp. 150470–150479, 2021.

[18] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[19] A.-M. Huang and T. Q. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 694–708, May 2008.

[20] N. Jacobson, Y.-L. Lee, V. Mahadevan, N. Vasconcelos, and T. Q. Nguyen, "A novel approach to FRUC using discriminant saliency and frame segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924–2934, Nov. 2010.

[21] S.-G. Jeong, C. Lee, and C.-S. Kim, "Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4497–4509, Nov. 2013.
[22] Y. Zhang, L. Xu, X. Ji, and Q. Dai, "A polynomial approximation motion estimation model for motion-compensated frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 8, pp. 1421–1432, Aug. 2016.
[23] D. Choi, W. Song, H. Choi, and T. Kim, "MAP-based motion refinement algorithm for block-based motion-compensated frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1789–1804, Oct. 2016.
[24] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," 2017, *arXiv:1707.07958*.
[25] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. NIPS*, 2016, pp. 730–738.
[26] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Single shot video object detector," *IEEE Trans. Multimedia*, vol. 23, pp. 846–858, 2021.
[27] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "BasicVSR: The search for essential components in video super-resolution and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4947–4956.
[28] K. C. K. Chan, S. Zhou, X. Xu, and C. C. Loy, "BasicVSR++: Improving video super-resolution with enhanced propagation and alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5972–5981.
[29] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-GAN: Unpaired video-to-video translation," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 647–655.
[30] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4463–4471.
[31] Y. L. Liu, Y. T. Liao, Y. Y. Lin, and Y. Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8794–8802.
[32] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 109–125.
[33] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 670–679.
[34] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 261–270.
[35] T. Peleg, P. Szekely, D. Sabo, and O. Sendik, "IM-Net for high resolution video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2398–2407.
[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Cham, Switzerland: Springer, 2015, pp. 234–241.
[37] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
[38] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 2. Sep. 1994, pp. 168–172.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
[40] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung, "Phase-based frame interpolation for video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1410–1418.
[41] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 434–450.

WHAN CHOI (Student Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include computer vision and machine learning, especially in the problems of video frame interpolation.

YEONG JUN KOH (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2011 and 2018, respectively. In March 2019, he joined the Department of Computer Science and Engineering, Chungnam National University, as an Assistant Professor. His research interests include computer vision and machine learning, especially in the problems of video object segmentation and image enhancement.

CHANG-SU KIM (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University (SNU), in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, USA. From 2001 to 2003, he coordinated the 3D Data Compression Group, National Research Laboratory for 3D Visual Information Processing, SNU. From 2003 to 2005, he was an Assistant Professor with the Department of Information Engineering, The Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is currently a Professor. He has published more than 320 journals and conference papers. His research interests include image processing, computer vision, and machine learning. He was a member of the Multimedia Systems and Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. He received the Distinguished Dissertation Award from SNU for his Ph.D. degree. He received the IEEK/IEEE Joint Award for Young IT Engineer of the Year, in 2009, and the Best Paper Award from *Journal of Visual Communication and Image Representation* (JVCI), in 2014. He served as an Editorial Board Member for *JVCI* and an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON MULTIMEDIA. He is a Senior Area Editor of *JVCI*. He was an APSIPA Distinguished Lecturer, from 2017 to 2018.