IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 7, 2021, accepted November 2, 2021, date of publication November 8, 2021,
date of current version November 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126593

Multi-Scale Warping for Video Frame Interpolation

WHAN CHOI[1], (Student Member, IEEE), YEONG JUN KOH[2], (Member, IEEE), AND CHANG-SU KIM[1], (Senior Member, IEEE)
[1]School of Electrical Engineering, Korea University, Seoul 136-701, 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 by the Korean Government through MSIT under Grant NRF-2018R1A2B3003896, Grant NRF-2019R1F1A1062907, and Grant NRF-2021R1A4A1031864; and in part by Samsung Electronics Company Ltd., under Grant IO201214-08156-01.

ABSTRACT A novel video interpolation network to improve the temporal resolutions of video sequences is proposed in this work. We develop a multi-scale warping module to interpolate intermediate frames robustly for both small and large motions. Specifically, the proposed multi-scale warping module deals with large motions between two consecutive frames using coarse-scale features, while estimating detailed local motions by exploring fine-scale features. To this end, it takes multi-scale features from the encoder and estimates kernel weights and offset vectors for each scale. Finally, it synthesizes multi-scale warping frames and combines them to obtain an intermediate frame. Extensive experimental results demonstrate that the proposed algorithm outperforms state-of-the-art video interpolation algorithms on various benchmark datasets.

INDEX TERMS Video frame interpolation, convolutional neural network, multi-scale feature, kernel-based approach, deformable convolution, adaptive convolution.

I. INTRODUCTION

The objective of video frame interpolation is to synthesize intermediate frames between two consecutive video frames. Video sequences with low temporal resolutions suffer from blur artifacts, temporal jittering, and motion aliasing, which provide unpleasant visual experience. Thus, the video frame interpolation task is essential for generating visually pleasant videos with high frame rates. Video frame interpolation can be applied to various fields, such as frame rate-up conversion [1], [2], slow motion generation [3], frame recovery in video streaming [4], [5], and novel view interpolation [6]. Many attempts have been made to interpolate intermediate frames, but it is still difficult to generate high-quality middle frames due to challenging factors such as occlusions and fast motions.

Most video frame interpolation algorithms include two processes: motion estimation and frame interpolation. They estimate motions to determine corresponding pixel positions in input frames and then interpolate an intermediate frame based on the correspondence matching information. In this regard, accurate motion estimation is required to obtain visually plausible intermediate frames. Recently, flow-based video interpolation methods [7]–[10] employ deep-learning-based optical flow techniques [11], [12] to yield reliable intermediate frames. However, these flow-based methods are vulnerable to optical flow errors. Moreover, they require large network parameters and additional training data to learn the optical flow networks.

Attempts have been made to obtain intermediate frames without explicit optical flow information. For example, some video frame interpolation algorithms [13], [14] derive pixel-wise adaptive convolution filters to perform the interpolation. Since they do not estimate motion vectors explicitly, they need large kernel sizes (*e.g.* 51×51) instead to cover large reference regions due to possible large motions. Thus, it requires high memory complexity and time consumption to obtain pixel-wise weights of these large kernels. Also, these algorithms may fail to find matching positions properly when motions are larger than the kernel size. Inspired by deformable convolution [15], recent video frame interpolation algorithms [16], [17] estimate kernel weights and offsets simultaneously. They use the offsets as pseudo-motions to determine matching locations, so they can reduce the

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

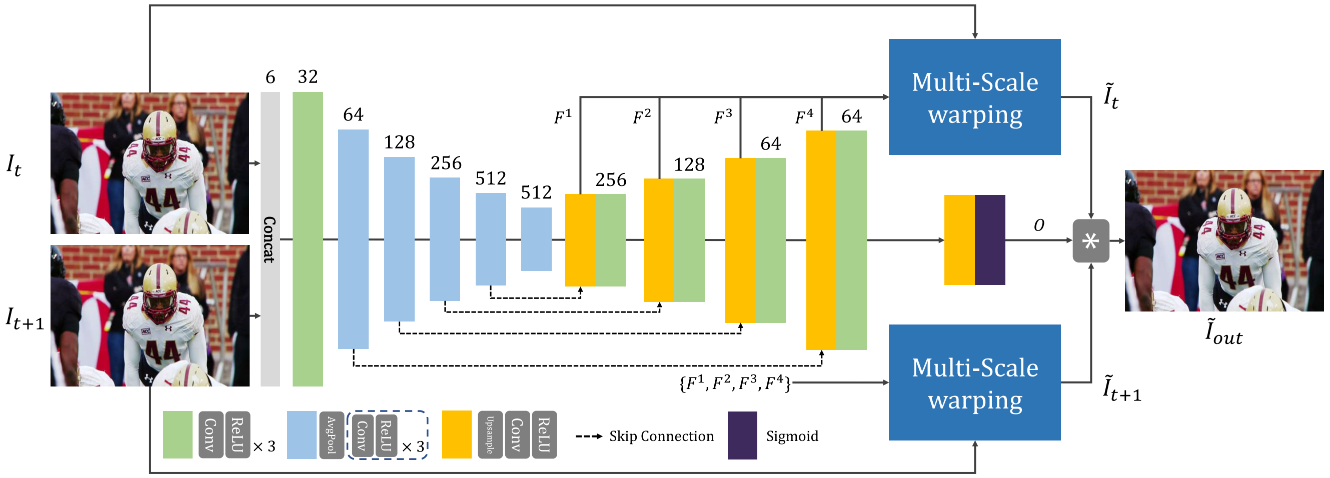


FIGURE 1. An overview of the proposed network.

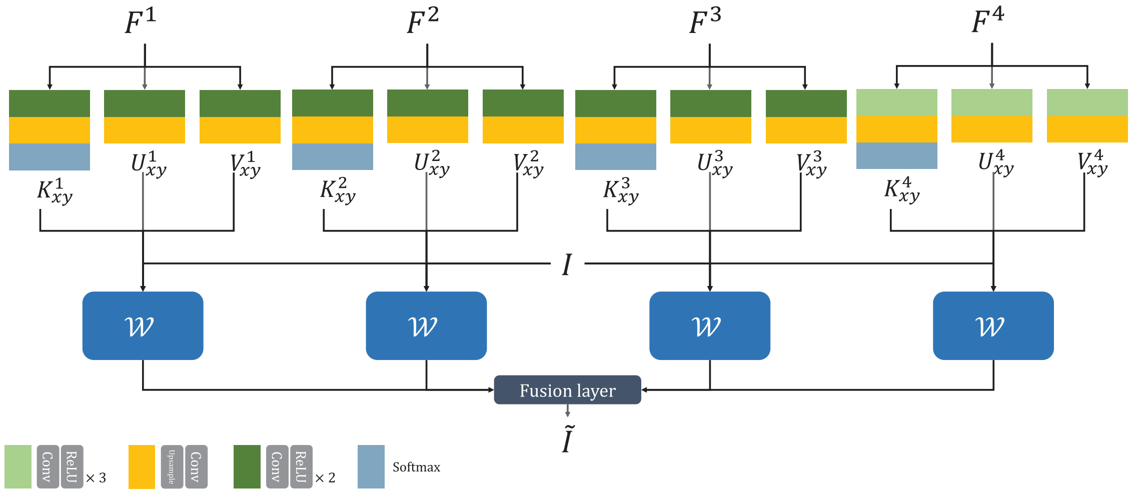


FIGURE 2. Architecture of the multi-scale warping module, where $\mathcal{W}$ represents the warping operation.

kernel size and exploit motion information without additional optical flow networks. However, these algorithms still have limitations on large motions, since they extract only fine-scale features to obtain kernel weights and offsets.

Multi-scale features have demonstrated the effectiveness for deep neural networks for various computer vision applications, including object detection [18], [19], image recognition [20], [21], and super resolution [22].

In this paper, we propose a novel video interpolation network, which consists of a multi-scale warping module based on deformable convolution. The proposed algorithm extracts coarse-scale features, as well as fine-scale features, to obtain multi-scale kernels and offsets for video frame interpolation. Then, the multi-scale warping module interpolates an intermediate frame between two input frames by performing deformable convolution based on those multi-scale kernels and offsets. The proposed multi-scale warping module deals with large motions between two consecutive frames using coarse-scale features while estimating detailed local motions by exploring fine-scale features. Experimental results demonstrate that the proposed algorithm outperforms state-of-the-art video interpolation algorithms on various benchmark datasets.

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

Many video frame interpolation algorithms use motion information to determine reference pixels for interpolation in two consecutive frames. In the past, conventional motion-compensated frame interpolation algorithms focused on estimating accurate motion vectors between two consecutive frames and then interpolated an intermediate frame by halving those motion vectors. To obtain reliable motion vectors for frame interpolation, motion vector refinement algorithms [23]–[28] have been developed. Choi *et al.* [23] used adaptive block sizes for motion estimation to represent complex motions around object boundaries. Huang and Nguyen [24] proposed a multistage motion vector refinement algorithm, which analyzes the distribution of residual energies to update unreliable motion vectors with varying block shapes and sizes. Jacobson *et al.* [25] used saliency and segmentation to refine motion vectors. Jeong *et al.* [26] formed multiple motion hypotheses for each pixel using various parameters such as block sizes and directions. They then solved a labeling problem to determine optimal parameters. Zhang *et al.* [27] modeled pixel intensities across neighboring frames as a differentiable function and developed a motion estimation scheme based on polynomial approximation. Choi *et al.* [28] proposed a MAP-based motion vector field refinement algorithm, which iteratively updates the motion vector of each block.

With the advance of deep-learning-based optical flow techniques, some flow-based video frame interpolation methods [7]–[9] employ existing optical flow techniques [11], [12] to warp two consecutive frames to generate intermediate frames. Niklaus and Liu [7] estimated bi-directional optical flow using PWC-Net [12] and performed the forward warping using the estimated optical flow to generate initial intermediate frames. MEMC-Net [8] proposed an adaptive warping layer, which uses optical flow [11] as offsets to decide matching positions. DAIN [9] refined optical flow [12] using depth information [29] to deal with occlusions. Niklaus and Liu [10] proposed softmax splatting to forward-warp frames based on the off-the-shelf optical flow method [12]. However, these flow-based methods are prone to optical flow errors, and some of them require additional training data for optical flow estimation and additional training time.

Instead of using off-the-shelf optical flow methods, some flow-based algorithms design end-to-end video frame interpolation networks, which extract motion information and perform motion-based frame warping jointly. DVF [30] is an encoder-decoder network to predict 3D optical flow across space and time in a video sequence. Liu *et al.* [31] proposed a cycle consistency loss for intermediate frame warping and improved the performance of DVF. Jiang *et al.* [3] developed an end-to-end convolutional neural network, which estimates bi-directional motions to interpolate intermediate frames. Also, BMBC [32] consists of a bilateral motion network to estimate intermediate motions and a dynamic filter generation network to obtain intermediate frames.

Instead of using motion information explicitly, the kernel-based approach interpolates intermediate frames by convolving input images with learnable kernel weights. AdaConv [14] uses pixel-wise convolution filters of a large size for frame interpolation, but it requires a large number of parameters to provide those pixel-wise coefficients. To reduce the memory usage, SepConv [13] performs separable convolution by dividing each 2D kernel into 1D horizontal and 1D vertical kernels. However, these algorithms [13], [14] cannot deal with motions larger than kernel sizes.

Recently, attempts have been made to estimate kernel weights and reference positions simultaneously [16], [17], [33]. Peleg *et al.* [33] formulated the motion estimation as a classification problem and performed convolutions with trained kernel weights based on the classified motions. Inspired by deformable convolution [15], DSepConv [16] and AdaCoF [17] adopt convolutional neural networks to produce offsets to decide reference positions in input images. With the estimated offsets, they can provide reliable interpolation results using a small kernel size.

III. PROPOSED ALGORITHM

Figure 1 shows the structure of the proposed network for video frame interpolation. The proposed network takes two successive video frames I_t and I_{t+1}, where t is a frame index, and produces an intermediate frame $\tilde{I}_{out}$. The feature extractor yields multi-scale features, and the two multi-scale warping modules generate warped results from I_t and I_{t+1}, respectively. Finally, the network combines the two warped results, $\tilde{I}_t$ and $\tilde{I}_{t+1}$, based on a pixel-wise learnable weights to obtain the intermediate frame $\tilde{I}_{out}$.

A. NETWORK ARCHITECTURE

1) FEATURE EXTRACTOR

The feature extractor takes a stack of I_t and I_{t+1} and produces multi-scale features for video frame interpolation. As shown in Figure 1, we design the feature extractor based on the U-net architecture [34], which is composed of an encoder, a decoder, and skip connections. The encoder contains six convolution blocks, each of which includes three sets of a 3 × 3 convolution layer with the ReLU activation. Also, each convolution block except the first one performs average pooling to extract high-level features. From the output of the encoder, the decoder yields multi-scale features for video interpolation. In the decoder, there are four sets of an up-sample block and a convolution block. Each up-sample block contains an up-sample layer with factor 2 and a 3 × 3 convolution layer with the ReLU activation. Then, from the four up-sample blocks, we extract multi-scale features, $\mathcal{F} = \{F^1, \ldots, F^4\}$, where F^l is the output feature of the lth up-sample block. The specification of the multi-scale feature extractor is summarized in Table 1.

TABLE 1. The 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	F^1
Convolution block	$H/16 \times W/16$	256	-
Up-sample block	$H/8 \times W/8$	128	F^2
Convolution block	$H/8 \times W/8$	128	-
Up-sample block	$H/4 \times W/4$	64	F^3
Convolution block	$H/4 \times W/4$	64	-
Up-sample block	$H/2 \times W/2$	64	F^4
Convolution block	$H/2 \times W/2$	64	-

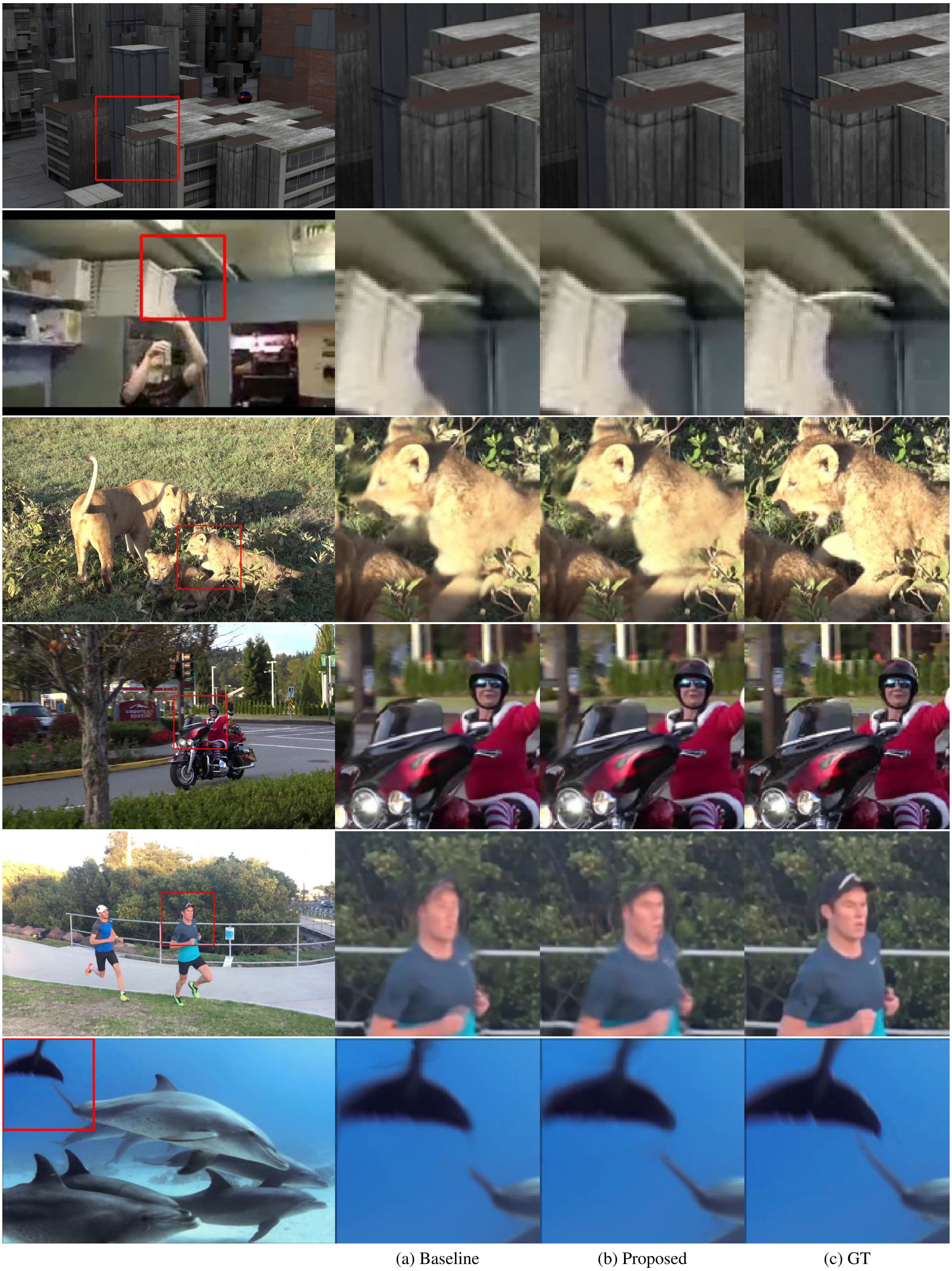


FIGURE 3. Qualitative comparison of the proposed algorithm with the baseline on Middlebury, UCF101, and DAVIS.

2) MULTI-SCALE WARPING MODULE

The multi-scale warping module warps an input image to obtain an intermediate frame. We formulate the warping process as the convolution with kernel weights. For reliable warping, motion information to determine reference (or matching) positions in the input image is required.

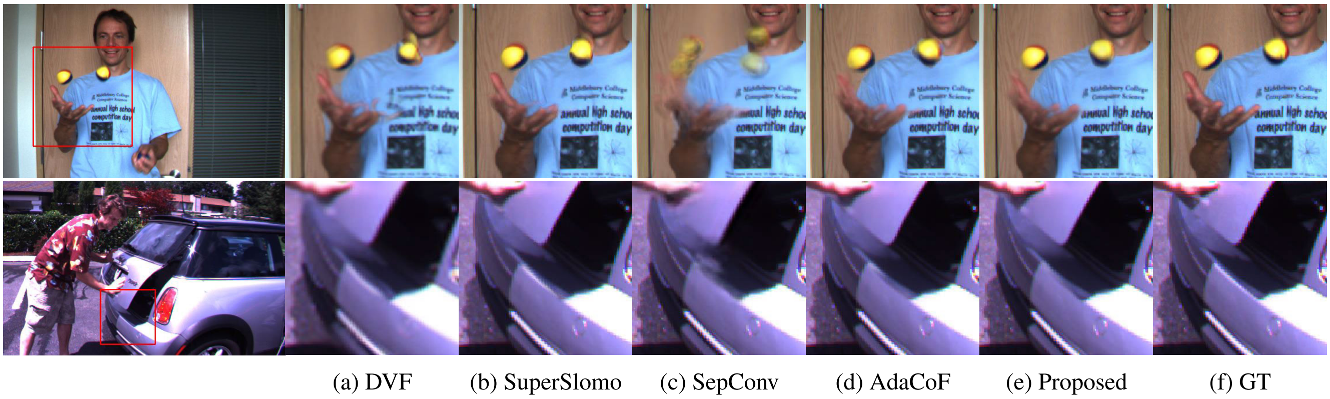


FIGURE 4. Qualitative comparison of the proposed algorithm with the existing algorithms on the Middlebury dataset.

Therefore, the proposed multi-scale warping module performs deformable convolution [15] to convolve the input image with kernel weights on reference positions.

Figure 2 shows the structure of the proposed multi-scale warping module, which contains 12 sub-networks to yield multi-scale kernel weights and offsets for the image warping. Each feature in $\mathcal{F}$ is fed into three sub-networks, which generate kernel weights, horizontal offsets, and vertical offsets, respectively. Using all multi-scale features in $\mathcal{F} = \{F^1, \ldots, F^4\}$, we obtain four sets of kernel weights, horizontal offsets, and vertical offsets. The offsets estimated at coarse scales can cover large motions, whereas small but detailed motions can be estimated at fine scales.

More specifically, for each feature F^l at the lth scale, three convolution blocks produce kernel weights $K^l \in \mathbb{R}^{H^l \times W^l \times k^2}$, horizontal offsets $U^l \in \mathbb{R}^{H^l \times W^l \times k^2}$, and vertical offsets $V^l \in \mathbb{R}^{H^l \times W^l \times k^2}$, respectively. Here, $H^l \times W^l$ is the spatial resolution of the lth scale, and the kernel size k is set to 5 in this work. Then, K^l, U^l, and V^l are up-sampled to have the same spatial resolution $H \times W$ as an input image via the up-sample blocks.

Let us consider the warping process of an input image I to compute the interpolation result at pixel (x, y) using K^l, U^l, and V^l. Specifically, let $K^l_{xy} \in \mathbb{R}^{k \times k}$, $U^l_{xy} \in \mathbb{R}^{k \times k}$, and $V^l_{xy} \in \mathbb{R}^{k \times k}$ denote the kernel weights, horizontal offsets, and vertical offsets for pixel (x, y), respectively. Then, the interpolation for pixel (x, y) at the lth scale is obtained by the warping process $\mathcal{W}$, which is expressed as

$$\begin{aligned}\tilde{I}^l(x,y) &= \mathcal{W}(I, K^l_{xy}, U^l_{xy}, V^l_{xy}) \\ &= \sum_{i=0}^{k-1}\sum_{j=0}^{k-1} K^l_{xy}(i,j) \\ &\quad \times I(x+i+U^l_{xy}(i,j), y+j+V^l_{xy}(i,j)). \end{aligned} \tag{1}$$

Note that we perform deformable convolution [15] to compute $\tilde{I}^l(x, y)$ in Eq. (1), as done in [16], [17]. Also, $I(x + i + U^l_{xy}(i,j), y + j + V^l_{xy}(i,j))$ is computed through bilinear interpolation, because those offset values, $U^l_{xy}(i,j)$ and $V^l_{xy}(i,j)$, may not be integers.

For each scale l, the multi-scale warping module obtains $\tilde{I}^l$ via Eq. (1). Warping results at fine scales tend to preserve detailed local motions, while those at coarse scales represent large motions between consecutive frames reliably. To take advantage of both coarse and fine scale information, the multi-scale warping module produces the warping result $\tilde{I}$ by combining $\{\tilde{I}^l\}_{l=1}^4$ with learnable adaptive weights $\{\alpha^l\}_{l=1}^4$ through a fusion layer as

$$\tilde{I} = \sum_l \alpha^l \tilde{I}^l \tag{2}$$

where the sum of the adaptive weights in $\{\alpha^l\}_{l=1}^4$ are constrained to 1. In other words, $\sum_{l=1}^4 \alpha^l = 1$.

B. VIDEO FRAME INTERPOLATION

As shown in Figure 1, the proposed algorithm employs two multi-scale warping modules. The first multi-scale warping module performs the warping forwardly from I_t to the intermediate frame. This forwardly warped frame is denoted by $\tilde{I}_t$. On the other hand, the second multi-scale warping module produces the backwardly warped frame $\tilde{I}_{t+1}$ from I_{t+1}. Note that the two multi-scale warping modules have different network parameters to estimate the kernel weights, horizontal offsets, and vertical offsets for performing the warping forwardly and backwardly, respectively.

Finally, we reconstruct the intermediate frame $\tilde{I}_{out}$ by combining the two warping results as

$$\tilde{I}_{out}(x,y) = O(x,y)\tilde{I}_t(x,y) + (1 - O(x,y))\tilde{I}_{t+1}(x,y) \tag{3}$$

where $O \in \mathbb{R}^{H \times W}$ is a learnable weight map for combining the two warping results effectively. Here, $0 \leq O(x, y) \leq 1$. A weight $O(x, y) > 0.5$ indicates that the warping result from I_t is more reliable than that from I_{t+1} at pixel (x, y). When a pixel is occluded in one of the two frames I_t and I_{t+1}, a higher weight can be assigned to the other frame through the weight map O. To obtain O, we add a sub-network to the 4th up-sample block in the decoder. This sub-network contains an up-sample block with the sigmoid activation to satisfy the constraint $0 \leq O(x, y) \leq 1$, as done in [17].

C. IMPLEMENTATION DETAILS

The proposed algorithm is trained in an end-to-end manner using a loss function

$$\mathcal{L}_M = \mathcal{L}_c + 0.01\mathcal{L}_s. \tag{4}$$

Here, $\mathcal{L}_c$ is the color loss between the predicted intermediate frame $\tilde{I}_{out}$ and the ground-truth I_{gt}. For the color loss, we use the Charbonnier function [35],

$$\mathcal{L}_c = \rho(\tilde{I}_{out} - I_{gt}) \tag{5}$$

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

$$\begin{aligned}\mathcal{L}_s = \sum_{l=1}^{4}\{&\rho(\nabla_x(\mathcal{P}_{\text{avg}}(K^l \odot U^l)))\\ &+\rho(\nabla_y(\mathcal{P}_{\text{avg}}(K^l \odot U^l)))\\ &+\rho(\nabla_x(\mathcal{P}_{\text{avg}}(K^l \odot V^l)))\\ &+\rho(\nabla_y(\mathcal{P}_{\text{avg}}(K^l \odot V^l)))\}\end{aligned} \tag{6}$$

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

In each convolution layer, ReLU is used as the activation function. The batch normalization is not employed, since we use a small batch size of 4. We train the proposed network using the AdaMax optimizer [36] with 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 70 epochs with an RTX 2080Ti GPU. The proposed network takes about 0.2 seconds to obtain an intermediate frame of size 1280 × 720.

IV. EXPERIMENTAL RESULTS

1) DATASETS AND METRICS

For training, we use Vimeo90K [37] as the training set, which contains 73,171 triplets of frames of resolution 448×256. The triplets in the training set are randomly cropped with a 256 × 256 size and then randomly flipped horizontally or vertically for data augmentation. For the evaluation, we use the same test sets as the state-of-the-art AdaCoF [17]: the 12 sequences in Middlebury [9] and randomly sampled sequences from the UCF101 and DAVIS datasets. For quantitative assessment of video frame interpolation, we use the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) metrics.

A. ABLATION STUDIES

First, we analyze the efficacy of multi-scale features for obtaining multi-scale kernel weights and offsets. In this test, we measure the video frame interpolation performances by modifying the number of multi-scale features as follows.

- 1) Warping with F^4 (baseline):
$$\tilde{I} = \tilde{I}^4$$
- 2) Warping with F^3 and F^4:
$$\tilde{I} = \alpha^3\tilde{I}^3 + \alpha^4\tilde{I}^4$$
- 3) Warping with F^2, F^3, and F^4:
$$\tilde{I} = \alpha^2\tilde{I}^2 + \alpha^3\tilde{I}^3 + \alpha^4\tilde{I}^4.$$

We set the first case as the baseline, since it uses only the finest-scale feature similarly to AdaCoF. As listed in Table 2, the warping with F^3 and F^4 provides a little performance gain on Middlebury and even worse performance on UCF101, as compared with the baseline. In contrast, the warping with three-scale features (F^2, F^3, and F^4) achieves performance gains on all three datasets. Finally, the proposed algorithm, which uses all four multi-scale features, provides the best performance on all datasets with large margins — 0.38dB on Middlebury and 0.45dB on DAVIS —- against the baseline. This indicates that the proposed multi-scale features are effective for the video frame interpolation task.

TABLE 2. Ablation study of multi-scale features.

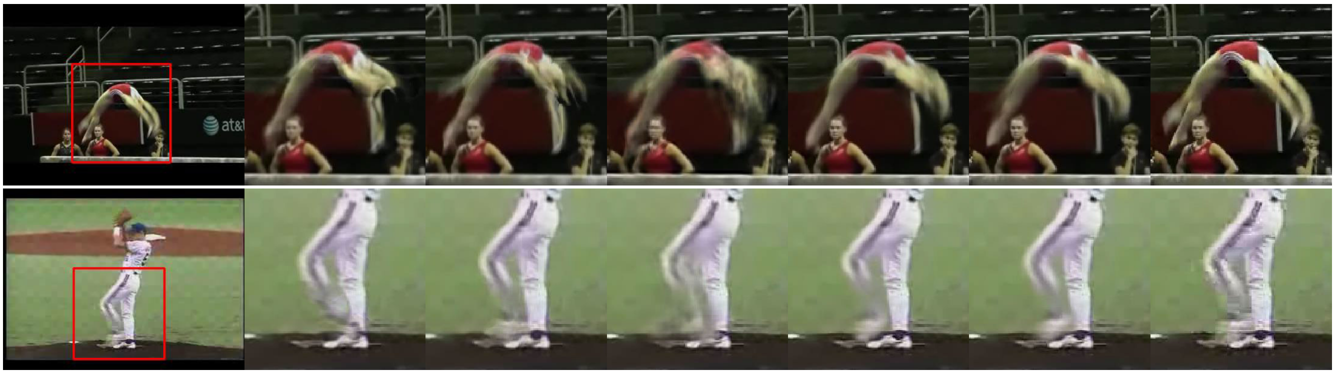
	Middlebury	UCF101	DAVIS
	PSNR(↑)	PSNR(↑)	PSNR(↑)
Warping with F^4 (baseline)	35.74	35.08	26.64
Warping with F^3 and F^4	35.96	35.06	26.99
Warping with F^2, F^3, and F^4	35.97	35.10	27.02
Proposed	**36.12**	**35.20**	**27.09**

Next, we conduct another ablation study by varying kernel sizes $k \in \{3, 5, 7\}$ in both training and test for predicting kernel weights and offset vectors. Table 3 shows that larger kernel sizes ($k = 5$ and $k = 7$) yield better performance than a small kernel size ($k = 3$). This is because a large kernel size can cover a large reference region in general. However, the best performance is achieved at $k = 5$, not at $k = 7$. This indicates that the proposed multi-scale warping module effectively deals with large motions even with $k = 5$ by exploiting features at coarse scales. We hence use the kernel size of 5 in the proposed algorithm.

TABLE 3. Ablation study of the kernel size k.

	Middlebury		UCF101		DAVIS	
	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)
$k = 3$	35.69	0.977	34.90	0.973	26.70	0.871
$k = 5$	**36.12**	**0.980**	**35.20**	**0.974**	**27.09**	**0.877**
$k = 7$	36.10	0.980	35.19	**0.974**	26.91	0.871

Figure 3 shows qualitative comparisons of the proposed algorithm and the baseline on the Middlebury, UCF101,



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

FIGURE 5. Qualitative comparison of the proposed algorithm with the existing algorithms on the UCF101 dataset.

and DAVIS datasets. The baseline yields blurry interpolation results in fast-moving objects, such as pizza dough in the 2nd row, runner in the 5th row, and dolphin tails in the 6th row. Also, the baseline fails to faithfully reconstruct fine details of objects, such as building in the 1st row, lion in the 3rd row, and helmet in the 4th row. In contrast, the proposed algorithm provides visually pleasing interpolation results on those fast-moving objects as well. These comparisons indicate that the proposed multi-scale warping module is capable of dealing with large motions effectively and reconstructing fine details between consecutive frames.

B. COMPARISON WITH STATE-OF-THE-ARTS

Table 4 compares the proposed algorithm with existing video frame interpolation algorithms — Phase [38], MIND [39], SepConv-$\mathcal{L}_1$ [13], DVF [30], SuperSlomo [3], and AdaCoF [17] — on the Middlebury, UCF101, and DAVIS datasets. The scores of the existing algorithms in Table 4 are from [17]. The proposed algorithm outperforms the existing algorithms with large PSNR gains on all three datasets. Notice that the proposed algorithm surpasses AdaCoF as well, which indicates that the proposed multi-level warping module is effective for video frame interpolation.

TABLE 4. Comparison of the proposed algorithm with the existing 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 [38]	31.12	0.933	32.45	0.953	23.47	0.800
MIND [39]	31.37	0.943	32.44	0.963	25.57	0.852
SepConv-$\mathcal{L}_1$ [13]	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 [3]	34.23	0.972	34.06	0.970	25.70	0.858
AdaCoF [17]	35.72	0.978	35.06	**0.974**	26.64	0.868
Proposed	**36.12**	**0.980**	**35.20**	**0.974**	**27.09**	**0.877**

Figure 4 compares the proposed algorithm with the existing algorithms on Middlebury qualitatively. The proposed algorithm reconstructs the shape of the right ball in the 1st row and the shadow of the trunk in the 2nd row more faithfully to the ground-truth than the other algorithms do. Figure 5 compares interpolation results on the UCF101 dataset. In the 1st row, the proposed algorithm and AdaCoF faithfully synthesize the arc shape of the athlete's movement. In the 2nd row, the proposed algorithm and SuperSlomo reconstruct the pitcher's legs more similarly to the ground-truth than the other algorithms. These results demonstrate that the proposed algorithm reconstructs fast-moving regions robustly. Finally, Figure 6 shows qualitative results on the DAVIS dataset. From the 1st to the 3rd rows, videos contain fast-moving objects, such as football player, wheel, and car. It is observed that the proposed algorithm yields visually pleasing interpolation results, especially in the helmet number and the shape of the car, while the other algorithms generate blurry interpolation results. Also, in the 4th row, the detailed texture of the reptile is faithfully reconstructed by the proposed algorithm.

C. VISUALIZATION OF MULTI-SCALE OFFSETS

The proposed multi-scale warping modules produce multi-scale kernel weights $\{K^l\}_{l=1}^4$ and multi-scale offsets $\{U^l, V^l\}_{l=1}^4$. We visualize these offsets to show the effectiveness of the multi-scale warping modules. For this purpose, we obtain a flow map $\mathcal{F}^l$ for each scale l by combining offsets with kernel weights. Specifically, a flow vector $\mathcal{F}^l(x, y)$ at pixel (x, y) is defined as

$$\mathcal{F}^l(x, y) = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1} K_{xy}^l(i, j)(U_{xy}^l(i, j), V_{xy}^l(i, j)). \tag{7}$$

Then, we combine all multi-scale offsets using the weights $\{\alpha^l\}_{l=1}^4$ in Eq. (2) to obtain a mixed flow map $\mathcal{F}_m$, which is given by

$$\mathcal{F}_m = \sum_{l=1}^{4} \alpha^l \mathcal{F}^l. \tag{8}$$

Figures 7(a) and (b) show two consecutive frames, I_t and I_{t+1}, and Figures 7(c) and (d) show the forward flows, $\mathcal{F}^4$

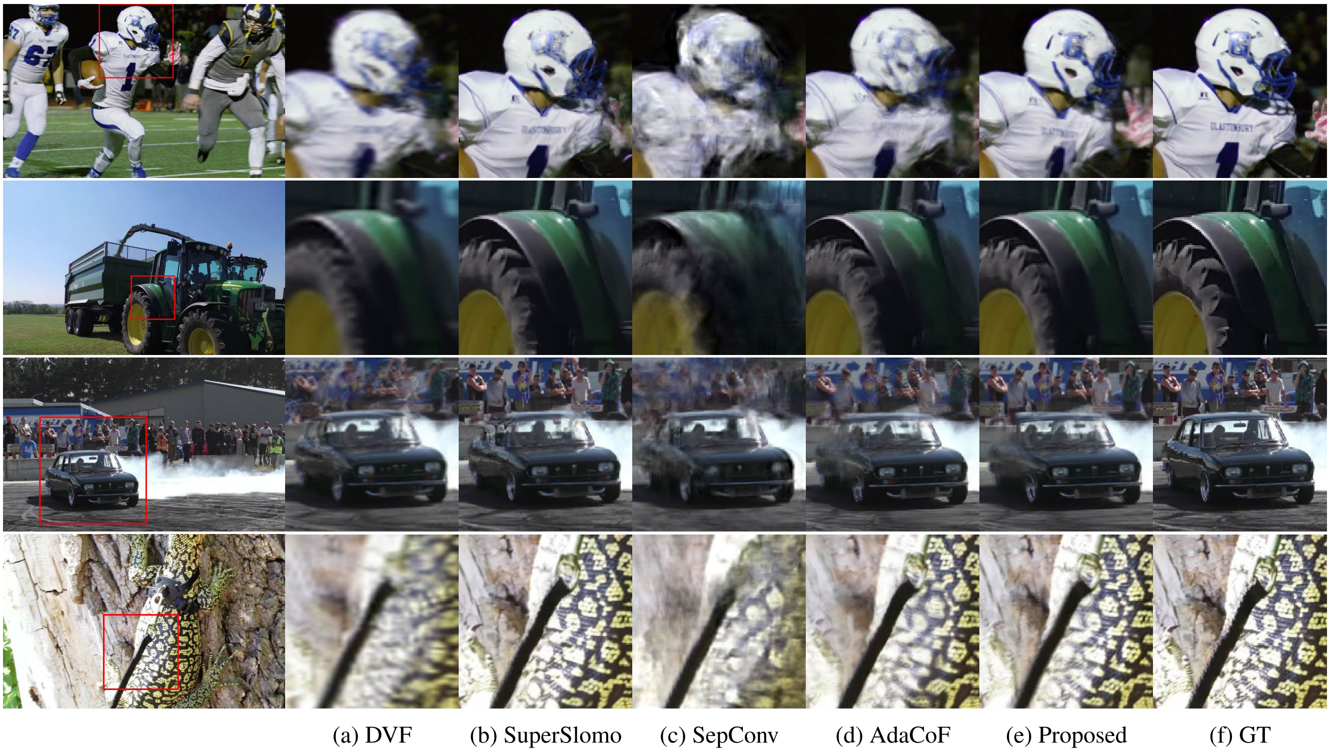


FIGURE 6. Qualitative comparison of the proposed algorithm with the existing algorithms on the DAVIS dataset.

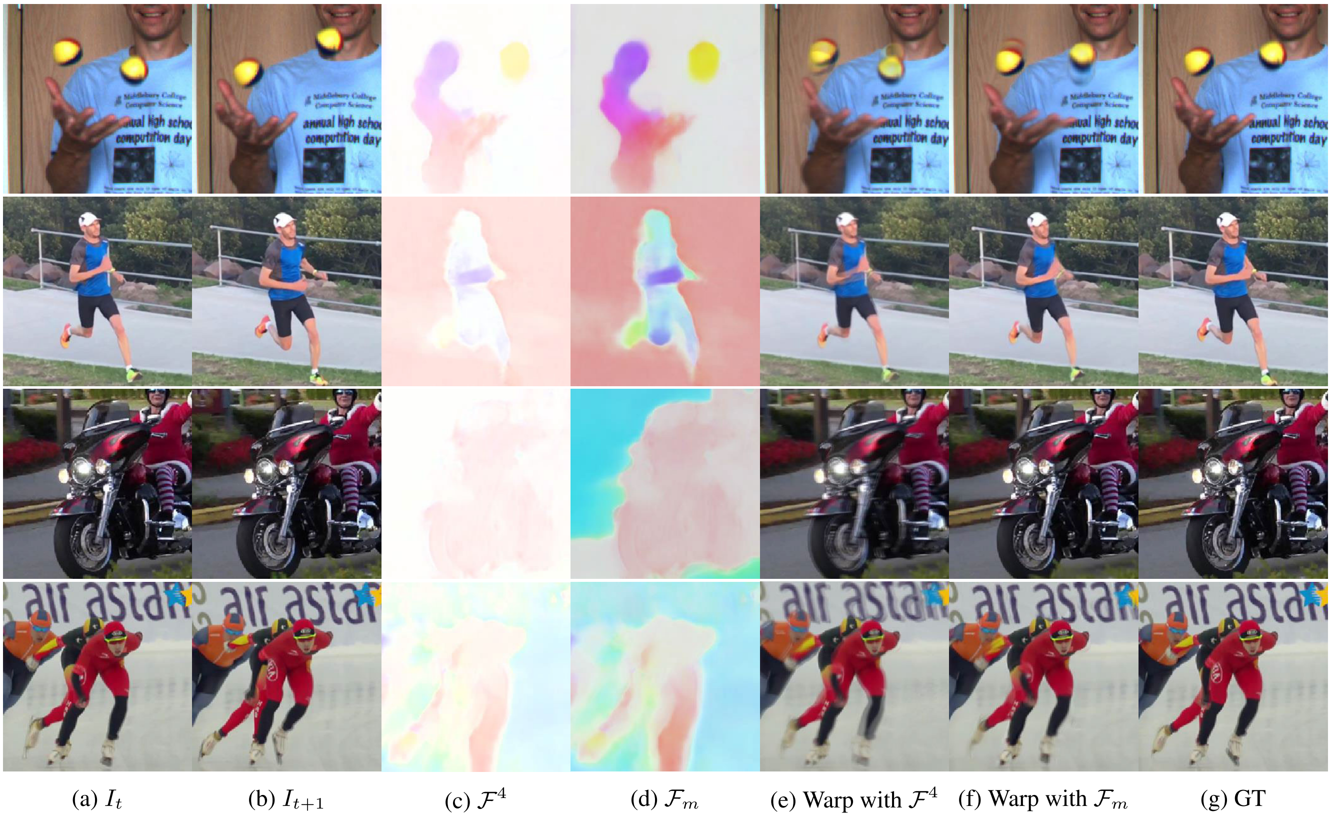


FIGURE 7. Visualization of offsets.

and $\mathcal{F}_m$, respectively. For simplicity, we show only the forward flow from I_t to the intermediate frame. In the 1st row, both left and right balls have large motions, but those ball regions have low flow values in $\mathcal{F}^4$ in Figure 7(c). On the contrary, $\mathcal{F}_m$ in Figure 7(d) have high responses on those regions. This indicates that multi-scale offsets represent large motions robustly. Also, by comparing two warping results in Figures 7(e) and (f), we observe that the warping result using all multi-scale features is more faithful than that using F^4 only. Similar results can be observed in a runner, a motorbike, and a skater in the 2nd, 3rd, and 4th rows, respectively.

V. CONCLUSION

In this paper, we developed a video interpolation network based on the multi-scale warping modules, which can deal with both large and small motions robustly. The proposed network extracts multi-scale features and estimates the set of kernel weights and offset vectors at each scale. It then performs the warping according to the scales and combines multi-scale warping results using learnable weights to obtain intermediate frames. Experimental results demonstrated that the proposed algorithm outperforms state-of-the-art video interpolation algorithms on various benchmark datasets.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep Stereo: Learning to predict new views from the world's imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5515–5524.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[19] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.

[20] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proc. ECCV*, Sep. 2018, pp. 805–821.

[21] X. Song, S. Jiang, and L. Herranz, "Multi-scale multi-feature context modeling for scene recognition in the semantic manifold," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2721–2735, Jun. 2017.

[22] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. ECCV*, Sep. 2018, pp. 517–532.

[23] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. NIPS*, 2016, pp. 730–738.

[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*. Springer, 2020, pp. 109–125.

[33] 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.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Springer, 2015, pp. 234–241.

[35] 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.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[37] 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.

[38] 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.

[39] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. ECCV*. 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 as an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University. His research interests include computer vision and machine learning, especially in the problems of video object segmentation and image enhancement.

CHANG-SU KIM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University, in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. 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 with the School of Electrical Engineering, Korea University, where he is currently a Professor. He has published more than 300 journals and conference papers. His research interests include image processing, computer vision, and machine learning. He is a member of the Multimedia Systems and Application Technical Committee (MSATC), IEEE Circuits and Systems Society. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from *Journal of Visual Communication and Image Representation* (JVCI). During his Ph.D. degree, he received a Distinguished Dissertation Award. He was an APSIPA Distinguished Lecturer, from 2017 to 2018. 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*.