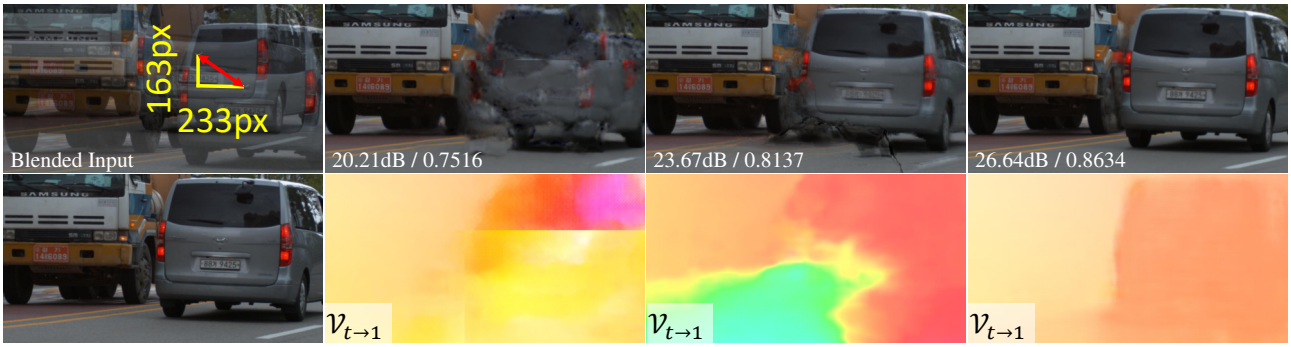
BiFormer: Learning Bilateral Motion Estimation via Bilateral Transformer for 4K Video Frame Interpolation

Junheum Park
Korea University
jhpark@mcl.korea.ac.kr

Jintae Kim
Korea University
jtkim@mcl.korea.ac.kr

Chang-Su Kim*
Korea University
changsukim@korea.ac.kr



Ground-Truth ABME XVFI BiFormer (Proposed)

Figure 1. Examples of 4K video frame interpolation results, obtained by ABME [1], XVFI [2], and the proposed BiFormer. 4K video frame interpolation is challenging due to large motion magnitudes, *e.g.* hundreds of pixels. PSNR/SSIM scores are presented within the interpolation results, and the estimated motion fields $\mathcal{V}_{t\to1}$ are at the bottom row.

Abstract

A novel 4K video frame interpolator based on bilateral transformer (BiFormer) is proposed in this paper, which performs three steps: global motion estimation, local motion refinement, and frame synthesis. First, in global motion estimation, we predict symmetric bilateral motion fields at a coarse scale. To this end, we propose BiFormer, the first transformer-based bilateral motion estimator. Second, we refine the global motion fields efficiently using blockwise bilateral cost volumes (BBCVs). Third, we warp the input frames using the refined motion fields and blend them to synthesize an intermediate frame. Extensive experiments demonstrate that the proposed BiFormer algorithm achieves excellent interpolation performance on 4K datasets. The source codes are available at https://github.com/JunHeum/BiFormer.

1. Introduction

Video frame interpolation (VFI) is a low-level vision task to increase the frame rate of a video, in which two (or more) successive input frames are used to interpolate intermediate frames. Its applications include video enhancement [3], video compression [4, 5], slow-motion generation [6], and view synthesis [7, 8]. Attempts have been made to develop effective VFI methods [1, 2, 6, 9–27]. Especially, with the advances in optical flow estimation [28–37], motion-based VFI methods provide remarkable performances. But, VFI for high-resolution videos, *e.g.* 4K videos, remains challenging due to diverse factors, such as large motions and small objects, hindering accurate optical flow estimation.

Most of these VFI methods are optimized for the Vimeo90K dataset [3] of a low spatial resolution (448×256), so they tend to yield poor results on 4K videos [2]. It is important to develop effective VFI techniques for 4K videos, which are widely used nowadays. 4K videos are, however, difficult to interpolate, for they contain large motions as in Figure 1. To cope with large motions, many optical flow estimators adopt coarse-to-fine strategies [31–33]. At a coarse scale, large motions can be handled more efficiently. But, motion errors at the coarse scale may propagate to a finer scale, making fine-scale results unreliable. To reduce such errors, the transformer can be a powerful solution, as demonstrated by recent optical flow estimators [36, 37]. However, these estimators cannot be directly used for VFI, in which the motion fields from an intermediate frame I_t, $0 < t < 1$, to input frames I_0 and I_1 should be estimated. For such bilateral motion estima-

*Corresponding author.

tion [1, 2, 24, 38, 39], a novel technique is required to adopt the transformer because the source frame I_t is not available.

In this paper, we propose a novel 4K VFI algorithm using the bilateral transformer (BiFormer) based on bilateral cross attention. First, we estimate global motion fields at a coarse scale via BiFormer. Second, we refine these global motion fields into final motion fields at a fine scale, by employing a motion upsampler recurrently. Last, we warp the two input frames using the final motion fields, respectively, and blend the two warped frames to synthesize an intermediate frame. Experimental results demonstrate that the proposed BiFormer algorithm provides the best performance on 4K benchmark datasets.

The work has the following major contributions:

- We propose the first transformer-based bilateral motion estimator, called BiFormer, for VFI.
- We develop blockwise bilateral cost volumes (BBCVs) to refine motion fields at 4K resolution efficiently.
- The proposed BiFormer algorithm outperforms the state-of-the-art VFI methods [1, 2, 14, 22–24, 40] on three 4K benchmark datasets [2, 41, 42].

2. Related Work

2.1. Motion-Based VFI

Motion-based frame warping for VFI has made great progress. An intermediate frame I_t, $0 < t < 1$, between two input frames I_0 and I_1 can be approximated by forward warping I_0 with a motion field $\mathcal{V}_{0\to t}$,

$$\hat{I}_t = \phi_{\mathrm{F}}(I_0, \mathcal{V}_{0\to t}) \tag{1}$$

where ϕ_{F} is the forward warping operator [43]. The required motion field $\mathcal{V}_{0\to t}$ can be obtained by scaling a motion field $\mathcal{V}_{0\to 1}$ between the input frames [16, 19, 25], given by

$$\mathcal{V}_{0\to t} = t \times \mathcal{V}_{0\to 1}. \tag{2}$$

However, no motion vector in $\mathcal{V}_{0\to t}$ may pass through a certain pixel, causing a hole in the warped frame. To alleviate the hole problem, Niklaus and Liu [16] predicted another warped frame $\phi_{\mathrm{F}}(I_1, \mathcal{V}_{1\to t})$, where $\mathcal{V}_{1\to t} = (1-t) \times \mathcal{V}_{1\to 0}$, and combine the two warped frames to synthesize I_t. On the other hand, multiple motion vectors may pass through the same pixel. To handle this collision, Niklaus and Liu [19] introduced the softmax splatting. Hu *et al*. [25] estimated reliability scores of motion vectors for better splatting.

In contrast, most motion-based VFI methods [1–3, 6, 12, 15, 17, 18, 24, 44] adopt backward warping [45],

$$\hat{I}_t = \phi_{\mathrm{B}}(\mathcal{V}_{t\to 0}, I_0). \tag{3}$$

Unlike $\mathcal{V}_{0\to t}$ in (2), it is not straightforward to determine the motion field $\mathcal{V}_{t\to 0}$ in (3) because the intermediate frame I_t is not available. Some algorithms [6, 44] assume that neighboring pixels have similar motion vectors and use motion vectors in $\mathcal{V}_{1\to 0}$ to approximate $\mathcal{V}_{t\to 0}$. Alternatively, the flow projection in [15, 18] aggregates multiple nearby motion vectors in $\mathcal{V}_{1\to 0}$ to approximate each vector in $\mathcal{V}_{t\to 0}$. To this end, Bao *et al*. [18] exploited depth information to determine aggregation weights adaptively. These approximate schemes, however, may cause visual artifacts near motion boundaries in warped frames.

Instead of approximation, Park *et al*. [24] estimated symmetric bilateral motion vectors directly, assuming motion trajectories between I_0 and I_1 are linear. For matched pixel pairs between I_0 and I_1, symmetric bilateral motion vectors are reliable in general. However, when a pixel in I_t is occluded in either I_0 or I_1, there is no matching pair, breaking the symmetry. Hence, Park *et al*. [1] refined symmetric vectors by loosening the linear motion constraint. The resultant bilateral motion vectors become asymmetric.

Sim *et al*. [2] employed both forward and backward warping techniques. They first predicted motion fields between input frames and then forward warped these fields using themselves, $\hat{\mathcal{V}}_{t\to 0} = \phi_{\mathrm{F}}(-\mathcal{V}_{0\to t}, \mathcal{V}_{0\to t})$. Then, they reconstructed $\hat{I}_t = \phi_{\mathrm{B}}(\hat{\mathcal{V}}_{t\to 0}, I_0)$ via backward warping.

2.2. Transformer

Vaswani *et al*. [46] proposed the transformer based on stacked self-attention layers. With its success in NLP, the transformer has been recently employed in many vision tasks as well. Dosovitskiy *et al*. [47] partitioned an image into patches and used them as tokens in the transformer. Based on global attention, the transformer is powerful but demands high complexity. To reduce the complexity, Liu *et al*. [48] developed the Swin transformer based on local attention and window shifting, which has been successfully adopted in dense prediction tasks.

Attempts have been made to adopt the transformer for VFI [26, 27]. Note that VFI algorithms can be classified into kernel-based or motion-based ones [1]. Shi *et al*. [27] extended spatial attention to spatiotemporal attention for kernel-based VFI. Lu *et al*. [26] developed convolutional networks to determine motion fields for motion-based VFI, but they adopted the transformer for frame synthesis. In contrast, in this work, we first use the transformer to estimate bilateral motion fields for motion-based VFI.

3. Proposed Algorithm

The performance of a motion-based VFI algorithm depends on the accuracy and reliability of motion estimation. However, as shown in Figure 1, motion magnitudes are as big as hundreds of pixels in a 4K video, but such a large search window for motion vectors is impractical at the original resolution. Hence, we propose BiFormer to estimate global motion fields at a coarse scale. But, at the coarse

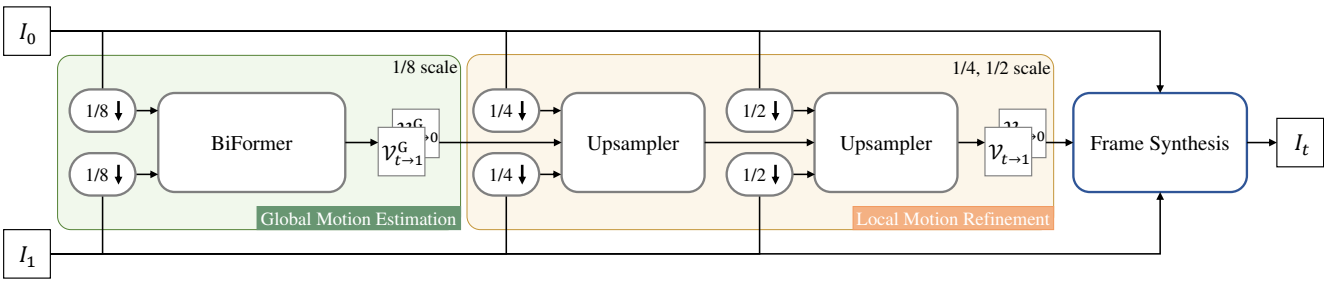


Figure 2. An overview of the proposed algorithm.

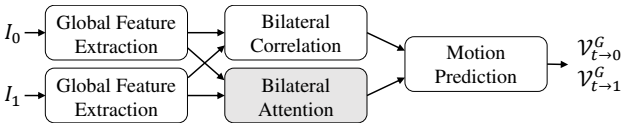


Figure 3. The architecture of BiFormer.

scale, small objects and detailed texture may be lost, and their motions may be unreliable. Thus, we also develop an upsampling module to upsample and refine the global motion fields recurrently.

Figure 2 is an overview of the proposed algorithm. We first downsample input frames I_0 and I_1 and then predict global motion fields $\mathcal{V}^{\mathrm{G}}_{t\to 0}$ and $\mathcal{V}^{\mathrm{G}}_{t\to 1}$ via BiFormer. Then, we upsample and refine the global fields using the upsampler twice to obtain final motion fields $\mathcal{V}_{t\to 0}$ and $\mathcal{V}_{t\to 1}$. Last, we synthesize an intermediate frame I_t using $\mathcal{V}_{t\to 0}$ and $\mathcal{V}_{t\to 1}$. For both $\{\mathcal{V}^{\mathrm{G}}_{t\to 0}, \mathcal{V}^{\mathrm{G}}_{t\to 1}\}$ and $\{\mathcal{V}_{t\to 0}, \mathcal{V}_{t\to 1}\}$, we use the symmetric bilateral motion model [24].

In Figure 2, the global estimation is performed at $1/8$ scale, while the local refinement is at $1/4$ or $1/2$ scale. Thus, input frames I_0 and I_1 are downsampled accordingly, but the downsampled images are also denoted by I_0 and I_1 for convenience. Also, it is assumed that the middle frame I_t, $t = \frac{1}{2}$, is interpolated from I_0 and I_1. All equations are derived for the case $t = \frac{1}{2}$, but the intermediate frame is denoted by I_t for notational convenience. Note that the equations can be straigtforwardly extended for a general t $(0 < t < 1)$.

3.1. Global Motion Estimation: BiFormer

To cope with large motions in 4K videos, transformer networks are more suitable than CNNs because of their longer-range connectivity. Hence, for global motion estimation, we propose BiFormer, which is the first transformer-based bilateral motion estimator. Figure 3 shows the architecture of BiFormer, consisting of global feature extraction, bilateral correlation, bilateral attention, and motion prediction modules. Let us describe these modules subsequently.

Global feature extraction: A transformer encoder extracts global feature maps $\mathcal{F}_0$ and $\mathcal{F}_1$ from I_0 and I_1, respectively. As the transformer encoder, we adopt the Twins architecture

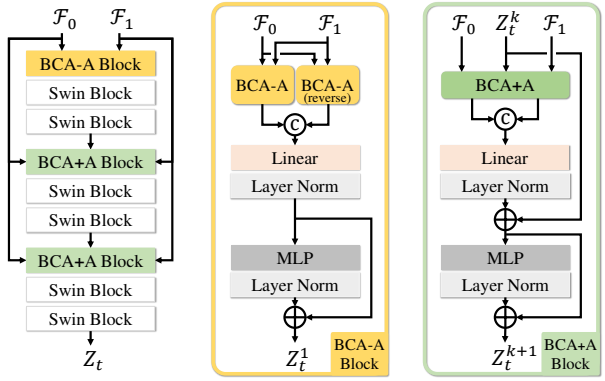


Figure 4. The architecture of the bilateral attention module.

[49]. The encoder reduces the spatial resolution by a factor of 8. Hence, compared with the original 4K frames, $\mathcal{F}_0$ and $\mathcal{F}_1$ are at $1/64$ scale.

Bilateral correlation: Many optical flow methods compute matching costs between input frames [1, 24, 28, 29, 31, 35]. Similarly, we compute matching costs between the global feature maps using the bilateral correlation module [24],

$$\mathcal{C}_t(\mathbf{x}, \mathbf{d}) = \mathcal{F}_0(\mathbf{x} - \mathbf{d})^T \mathcal{F}_1(\mathbf{x} + \mathbf{d}), \ \mathbf{d} \in \mathcal{D} \qquad (4)$$

where $\mathcal{D} = \{\, \mathbf{d} = (d_x, d_y) \,|\, -r \le d_x \le r, \ -r \le d_y \le r \,\}$ is a local window. We set r to 15, so motion magnitudes are considered up to 960 $(= 15 \times 64)$ pixels vertically and horizontally at 4K resolution. In (4), for each pixel $\mathbf{x}$ in I_t, a symmetrically matched pair of pixels, $\mathbf{x}-\mathbf{d}$ in I_0 and $\mathbf{x}+\mathbf{d}$ in I_1, are used to compute the cost $\mathcal{C}_t(\mathbf{x}, \mathbf{d})$.

Bilateral attention: In Figure 4, the bilateral attention module consists of three types of attention blocks. First, we use a bilateral cross attention without anchor (BCA-A) block to yield Z^1_t, where superscript 1 means the output of the first attention block. Then, we use two Swin blocks [50] with the shifted windowing to yield Z^3_t. Next, using Z^3_t as an anchor, a bilateral cross attention with anchor (BCA+A) block is used to yield Z^4_t. In this manner, we obtain the bilateral feature map Z^9_t through nine attention blocks in total. In Figure 4, the final output Z^9_t is denoted by Z_t.

1) BCA-A: Cross attention [46] is used to attend two differ-

Figure 5. Two types of bilateral cross attention blocks.

ent types of features, by employing one for query and the other for key and value. In particular, in optical flow estimation, query features are extracted from a source frame, while key and value features are from a target frame [36]. However, in bilateral motion estimation, the source (or anchor) frame I_t — which we aim to interpolate — is unavailable, while two target frames I_0 and I_1 are given. Thus, we develop the BCA-A block for bilateral motion estimation.

In the BCA-A block in Figure 5(a), the global feature maps $\mathcal{F}_0$ and $\mathcal{F}_1$ are given. We extract query features from $\mathcal{F}_0$ and key and value features from $\mathcal{F}_1$,

$$Q_0 = W_Q \mathcal{F}_0, \quad K_1 = W_K \mathcal{F}_1, \quad V_1 = W_V \mathcal{F}_1, \tag{5}$$

where W_Q, W_K, and W_V are projection matrices. Then, we compute the attention matrix using a sliding window [51], given by

$$A_t(\mathbf{x}, \mathbf{d}) = Q_0(\mathbf{x} - \mathbf{d})^T K_1(\mathbf{x} + \mathbf{d}), \ \mathbf{d} \in \mathcal{D} \tag{6}$$

With a learnable position bias P [50], we normalize A_t by

$$\bar{A}_t = \text{SoftMax}(A_t + P), \tag{7}$$

so that $\sum_{\mathbf{d}\in\mathcal{D}} \bar{A}_t(\mathbf{x}, \mathbf{d}) = 1$ for each $\mathbf{x}$. Thus, $\bar{A}_t(\mathbf{x}, \mathbf{d})$ represents the similarity between $\mathbf{x} - \mathbf{d}$ in I_0 and $\mathbf{x} - \mathbf{d}$ in I_1, which are symmetrically located with respect to $\mathbf{x}$ in I_t as shown in Figure 5(a). Finally, based on the similarities, we obtain the attended value feature,

$$Z_{1\to t}(\mathbf{x}) = \textstyle\sum_{\mathbf{d}\in\mathcal{D}} \bar{A}_t(\mathbf{x}, \mathbf{d})^T \, V_1(\mathbf{x} + \mathbf{d}). \tag{8}$$

Note that $Z_{1\to t}$ uses the information in I_1 to approximately represent I_t based on the symmetric bilateral matching information in (6).

Similarly, we reverse the roles of $\mathcal{F}_0$ and $\mathcal{F}_1$ in (5) and then perform the same BCA-A process to obtain another attended value feature $Z_{0\to t}$. Then, as shown in Figure 4, we concatenate $Z_{0\to t}$ and $Z_{1\to t}$ and process the result through linear, layer normalization, and MLP layers to yield Z_t^1.

2) BCA+A: After the first BCA-A block, the anchor information Z_t^k at time t is available. We can aggregate more informative features for I_t by exploiting Z_t^k. As shown in Figure 5(b), if pixel $\mathbf{x}$ in Z_t^k has a constant velocity, its trajectory formed by $\mathbf{x} - \mathbf{d}$ in $\mathcal{F}_0$, $\mathbf{x}$ in Z_t^k, and $\mathbf{x} + \mathbf{d}$ in $\mathcal{F}_1$ is linear. Hence, we develop the BCA+A block to exploit mutual connections among the triplet $(\mathbf{x} - \mathbf{d}, \mathbf{x}, \mathbf{x} + \mathbf{d})$.

First, we extract queries from the anchor Z_t^k and keys and values from the two target feature maps $\mathcal{F}_0$ and $\mathcal{F}_1$,

$$\begin{aligned} Q_t = W_Q Z_t^k, \quad K_0 = W_K \mathcal{F}_0, \quad V_0 = W_V \mathcal{F}_0, \\ K_1 = W_K \mathcal{F}_1, \quad V_1 = W_V \mathcal{F}_1. \end{aligned} \tag{9}$$

Then, we compute the anchor-aware attention matrix,

$$B_t(\mathbf{x}, \mathbf{d}) = Q_t(\mathbf{x})^T K_0(\mathbf{x} - \mathbf{d}) + Q_t(\mathbf{x})^T K_1(\mathbf{x} + \mathbf{d}) \tag{10}$$

where $\mathbf{d} \in \mathcal{D}$. We convert B_t into $\bar{B}_t$ similarly to (7). Finally, we obtain the anchor-aware attended value features,

$$Z_{0\to t}^{\text{anch}}(\mathbf{x}) = \textstyle\sum_{\mathbf{d}\in\mathcal{D}} \bar{B}_t(\mathbf{x}, \mathbf{d})^T V_0(\mathbf{x} - \mathbf{d}), \tag{11}$$

$$Z_{1\to t}^{\text{anch}}(\mathbf{x}) = \textstyle\sum_{\mathbf{d}\in\mathcal{D}} \bar{B}_t(\mathbf{x}, \mathbf{d})^T V_1(\mathbf{x} + \mathbf{d}). \tag{12}$$

As done in the BCA-A block, these two features are concatenated and then processed to yield Z_t^{k+1}.

Motion prediction: To predict global motion fields $\mathcal{V}_{t\to 0}^{\text{G}}$ and $\mathcal{V}_{t\to 1}^{\text{G}}$, we concatenate the cost volume $\mathcal{C}_t$ in (4) and the bilateral feature map Z_t^9 from the bilateral attention module. Then, the concatenated features are processed by convolution layers to yield the global bilateral motion field $\mathcal{V}_{t\to 1}^{\text{G}}$ from I_t to I_1. Also, because of the symmetric bilateral motion constraint [24], we have

$$\mathcal{V}_{t\to 0}^{\text{G}} = -\mathcal{V}_{t\to 1}^{\text{G}}. \tag{13}$$

More details are presented in the supplement.

3.2. Local Motion Refinement: Upsampler

At 1/8 scale, BiFormer predicts large global motions such as camera panning and rigid movements of large objects effectively. However, the global motion fields may be unreliable, especially in regions for small objects or near sharp object boundaries. Hence, we develop the upsampler to refine the global motion fields. As shown in Figure 2, we use the upsampler twice to refine the global motion fields $\mathcal{V}_{t\to 1}^{\text{G}}$ and $\mathcal{V}_{t\to 0}^{\text{G}}$ at 1/8 scale into the final motion fields $\mathcal{V}_{t\to 1}$ and $\mathcal{V}_{t\to 0}$ at 1/2 scale. Since the upsampler operates at the fine scales, we implement it based on convolution layers, instead of transformer blocks. This is because the transformer demands a huge number of parameters to process high-resolution input in general.

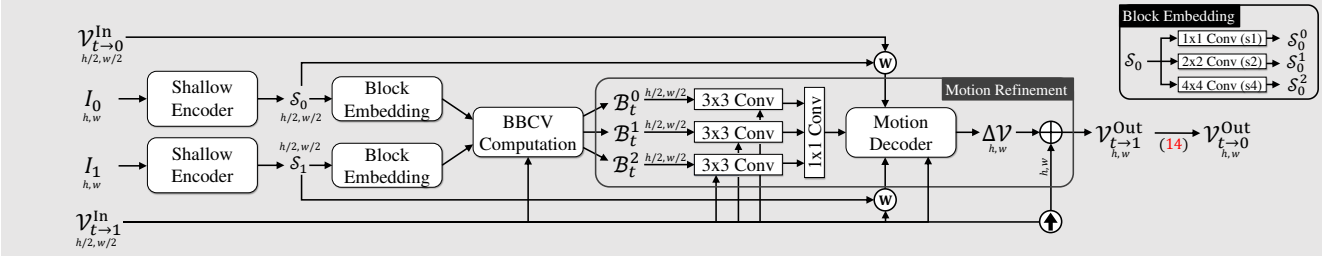


Figure 6. The network structure of the upsampler. In the block embedding layer, (sN) denotes that the convolution layer has stride N.

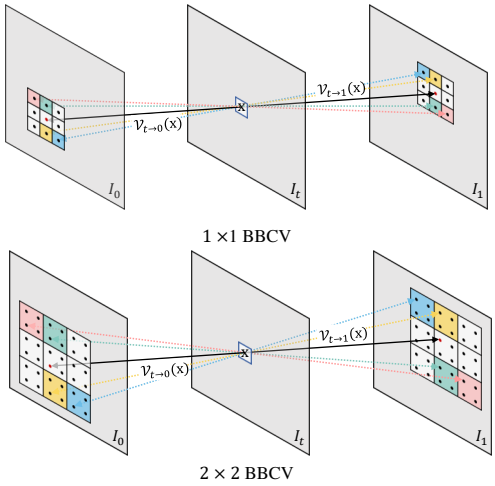


Figure 7. Illustration of a 1×1 BBCV $\mathcal{B}_t^0$ and a 2×2 BBCV $\mathcal{B}_t^1$.

Figure 6 shows the network structure of the upsampler, which takes a motion field $\mathcal{V}_{t\to1}^{\text{In}}$ as input and yields an upsampled field $\mathcal{V}_{t\to1}^{\text{Out}}$ as output. Note that

$$\mathcal{V}_{t\to0}^{\text{In}} = -\mathcal{V}_{t\to1}^{\text{In}} \quad \text{and} \quad \mathcal{V}_{t\to0}^{\text{Out}} = -\mathcal{V}_{t\to1}^{\text{Out}}. \tag{14}$$

Feature extraction: Given I_0 and I_1, we extract the feature maps using the shallow encoder, which yields local feature maps $\mathcal{S}_0$ and $\mathcal{S}_1$. Note that the local motion refinement is performed at fine scales of 1/4 and 1/2. Thus, to refine large motions, the search range should be sufficiently large as well. We hence perform block embedding using three convolution layers with kernel sizes of 1×1, 2×2, and 4×4, respectively. They process $\mathcal{S}_0$ and $\mathcal{S}_1$ to generate $\mathcal{S}_0^k$ and $\mathcal{S}_1^k$, where $k \in \{0, 1, 2\}$ is the block size index. Note that the block embedding layer is depicted in detail at the top right corner of Figure 6.

BBCVs: In the optical flow estimator in [35], blockwise cost volumes are used to increase the search range. Those volumes, however, cannot be used in this work, since the source frame I_t is unavailable and should be interpolated. We hence develop BBCVs.

Figure 7 illustrates BBCVs. Given the bilateral motion fields $\mathcal{V}_{t\to0}^{\text{In}}$ and $\mathcal{V}_{t\to1}^{\text{In}}$, which are symmetric with respect to I_t, the bilateral search windows for pixel $\mathbf{x}$ have two center points (red points in Figure 7), given by

$$\begin{aligned} \mathbf{x}_0'^k &= (\mathbf{x} + \mathcal{V}_{t\to0}^{\text{In}}(\mathbf{x}))/2^k, \\ \mathbf{x}_1'^k &= (\mathbf{x} + \mathcal{V}_{t\to1}^{\text{In}}(\mathbf{x}))/2^k, \end{aligned} \tag{15}$$

where k is the block size index. Then, we define a search window $\{\mathbf{d} = (d_x, d_y) \mid -r \le d_x \le r,\ -r \le d_y \le r\}$ with $r = 2$ and compute three BBCVs $\{\mathcal{B}_t^0, \mathcal{B}_t^1, \mathcal{B}_t^2\}$,

$$\mathcal{B}_t^k(\mathbf{x}, \mathbf{d}) = \mathcal{S}_0^k(\mathbf{x}_0'^k - \mathbf{d})^T \mathcal{S}_1^k(\mathbf{x}_1'^k + \mathbf{d}),\ \ k = 0, 1, 2. \tag{16}$$

Since the source pixel $\mathbf{x}$ in $\mathcal{B}_t^k$ in (16) is at the high resolution, the motion of even a small object can be predicted precisely. On the other hand, the size of the search window is $(2r+1)^2$ blocks, which corresponds to $((2r+1) \times 2^k)^2$ in the pixel unit. Thus, a large motion can be refined by employing BBCVs.

Motion refinement: For each $k \in \{0, 1, 2\}$, the BBCV $\mathcal{B}_t^k$ and the motion field $\mathcal{V}_{t\to1}^{\text{In}}$ are concatenated and processed by a 3×3 convolution layer. Then, the three results are aggregated by a 1×1 convolution layer, yielding the matching feature map.

Using $\mathcal{V}_{t\to0}^{\text{In}}$ and $\mathcal{V}_{t\to1}^{\text{In}}$, we warp the local feature maps $\mathcal{S}_0$ and $\mathcal{S}_1$. Then, by taking the warped feature maps, the matching feature map, and $\mathcal{V}_{t\to1}^{\text{In}}$ as input, the motion decoder predicts a residual motion field $\Delta\mathcal{V}$. Finally, we generate an upsampled, refined motion field given by

$$\mathcal{V}_{t\to1}^{\text{Out}} = \tilde{\mathcal{V}}_{t\to1}^{\text{In}} + \Delta\mathcal{V} \tag{17}$$

where $\tilde{\mathcal{V}}_{t\to1}^{\text{In}}$ is the bilinearly upsampled $\mathcal{V}_{t\to1}^{\text{In}}$.

3.3. Frame Synthesis

We develop a simple frame synthesis network, composed of an encoder, three skip connections with warping, and a decoder. The encoder processes I_0 and I_1 to extract multi-scale feature maps $\mathcal{G}_0^l$ and $\mathcal{G}_1^l$, where $l \in \{0, 1, 2\}$ is the scale index. At level l, we downsample the refined motion fields $\mathcal{V}_{t\to0}$ and $\mathcal{V}_{t\to1}$ to yield $\mathcal{V}_{t\to0}^l$ and $\mathcal{V}_{t\to1}^l$. Using them, we warp $\mathcal{G}_0^l$ and $\mathcal{G}_1^l$, which are then passed to the decoder via a skip connection to the first layer of the decoder at the lth level. Finally, the decoder synthesizes the intermediate frame I_t. The details are presented in the supplement.

Table 1. Quantitative comparison of VFI results. For each test, the best result is **boldfaced**, while the second-best is underlined.

	X4K1000FPS		Xiph-4K		BVI-DVC-4K		2K Runtime (seconds)	#Parameters (millions)
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM		
SepConv [14]	24.34	0.7420	32.61	0.8801	26.57	0.8485	0.41	21.6
CAIN [22]	24.50	0.7522	33.07	0.8896	27.06	0.8582	0.14	42.8
AdaCoF [23]	24.13	0.7338	32.72	0.8805	26.35	0.8402	0.28	22.9
BMBC [24]	22.86	0.7269	31.27	0.8804	25.41	0.8384	6.45	11.0
CDFI [40]	24.49	0.7419	33.01	0.8720	26.84	0.8496	1.24	5.0
XVFI [2]	30.12	0.8704	34.06	0.8946	29.17	0.8956	0.36	5.6
ABME [1]	30.16	0.8793	33.81	0.9030	28.28	0.8905	1.16	18.1
VFIformer [26]	24.58	0.8054	33.69	0.9252	27.45	0.9049	–	24.2
M2M-PWC [25]	30.81	0.9120	34.46	0.9252	**29.77**	0.9274	0.07	7.6
BiFormer (Proposed)	**31.32**	**0.9212**	**34.48**	**0.9268**	29.67	**0.9296**	0.53	11.2

4. Experiments

4.1. Training

We first train the global motion estimator, BiFormer. Then, we freeze BiFormer and train the local motion refinement network and the synthesis network together.

BiFormer: To train BiFormer, we define the photometric loss between the ground-truth I_t^{GT} and two warped frames:

$$\mathcal{L}_{\text{pho}} = \rho(I_t^{\text{GT}} - \phi_{\text{B}}(\mathcal{V}_{t\to 0}^{\text{G}}, I_0)) + \rho(I_t^{\text{GT}} - \phi_{\text{B}}(\mathcal{V}_{t\to 1}^{\text{G}}, I_1)) + \mathcal{L}_{\text{cen}}(I_t^{\text{GT}}, \phi_{\text{B}}(\mathcal{V}_{t\to 0}^{\text{G}}, I_0)) + \mathcal{L}_{\text{cen}}(I_t^{\text{GT}}, \phi_{\text{B}}(\mathcal{V}_{t\to 1}^{\text{G}}, I_1)) \tag{18}$$

where $\rho(x) = (x^2 + \epsilon^2)^\alpha$ is the Charbonnier function [52] and $\mathcal{L}_{\text{cen}}$ is the census loss [53–55]. The parameters are set to $\alpha = 0.5$ and $\epsilon = 10^{-3}$. We use only the Vimeo90K training set [3] to train BiFormer. It is composed of 51,312 triplets of resolution 448×256. Thus, in inference, for the global motion estimation, we downsample input frames to 448×256.

Local motion refinement with synthesis: To train the local motion refinement network and the synthesis network together, we define the synthesis loss between the ground-truth I_t^{GT} and a synthesized frame I_t as

$$\mathcal{L}_{\text{syn}} = \rho(I_t^{\text{GT}} - I_t) + \mathcal{L}_{\text{cen}}(I_t^{\text{GT}}, I_t). \tag{19}$$

We use the training dataset of X4K1000FPS [2], containing 4,408 sets of 65 successive frames of resolution 768×768, cropped from 4K videos. It is augmented by random flipping, rotating, order reversing, and cropping of 512×512 patches. Moreover, to learn the motion refinement at various resolutions, we downsample input frames to a random size from 96×96 to 256×256. We use the Adam optimizer [56] with a learning rate of $\eta = 10^{-4}$ until 0.1M iterations and then halve η after every 0.05M iterations. We use a batch size of 4 for 0.2M iterations in total.

4.2. Test Datasets

As mentioned above, we use the Vimeo90K training set for BiFormer and then the X4K1000FPS training set for the whole algorithm. The proposed algorithm is designed for 4K videos, so we use three 4K test datasets.

X4K1000FPS [2]: Its test set, called X-TEST, contains 15 clips of 33 successive 4K frames of frame rate 1000 fps. These clips contain diverse motions, including rotation, panning, rigid movement, zoom-in, and zoom-out.

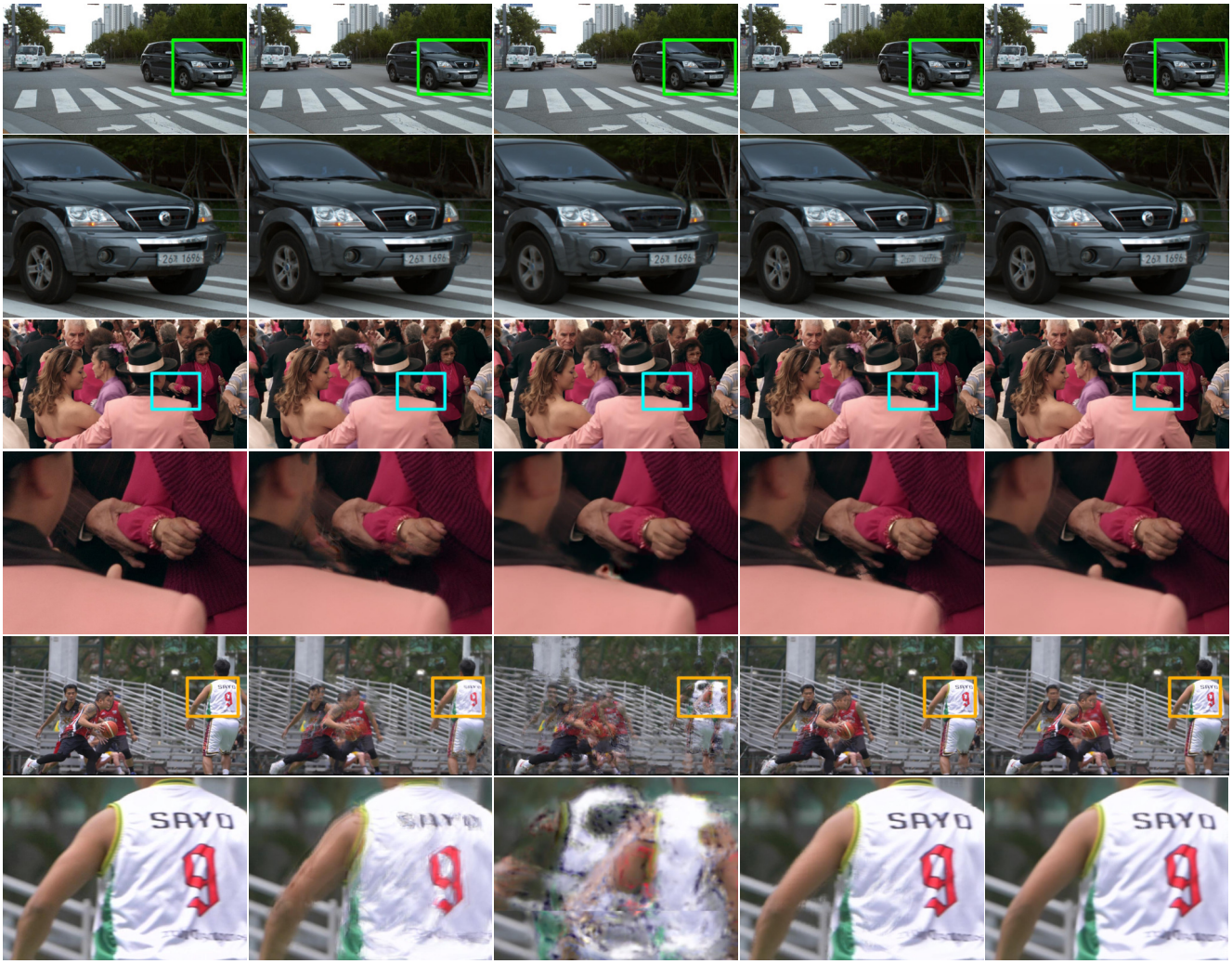
Xiph-4K [41]: It contains 19 raw video sequences for testing video codecs. Each sequence is composed of 31 successive 4K frames. Even-indexed frames are used as input frames, while odd-indexed frames are the ground-truth for intermediate frames. Thus, there are 285 triplets in total.

BVI-DVC-4K [42]: It contains 200 raw sequences. From each sequence, three successive 4K frames are extracted. Thus, there are 200 triplets in total. These sequences capture complex texture and cluttered scenes.

4.3. Comparison with the State-of-the-Arts

We compare the proposed BiFormer algorithm with nine conventional algorithms: SepConv [14], CAIN [22], AdaCoF [23], BMBC [24], CDFI [40], XVFI [2], ABME [1], VFIformer [26], and M2M-PWC [25]. Table 1 compares the average PSNR and SSIM scores on X4K1000FPS, Xiph-4K, and BVI-DVC-4K. Except for the PSNR metric on BVI-DVC-4K, BiFormer provides the best PSNR and SSIM scores. Especially, on X4K1000FPS, compared to the second-best M2M-PWC, BiFormer improves PSNR by more than 0.5dB. It is worth pointing out that, whereas M2M-PWC uses PWC-Net trained on an optical flow dataset, BiFormer achieves better results without employing such additional datasets.

Table 1 also lists the runtimes for interpolating an intermediate frame in a 2K (1920×1080) sequence using an RTX 3090 GPU. Note that, for a 4K sequence, BMBC, CDFI, VFIformer, and ABME cannot process entire frames



(a) Ground-Truth (b) XVFI (c) ABME (d) M2M-PWC (e) BiFormer (Proposed)

Figure 8. Qualitative comparison of interpolated frames. The top, middle, and bottom examples are from X4K1000FPS, Xiph-4K, and BVI-DVC-4K, respectively. The proposed BiFormer in (e) interpolates the frames faithfully to the ground truth in (a).

at once due to the lack of memory. Thus, they should divide input frames into patches, infer an intermediate frame patch-wise, and merge the interpolated patches. This patch-wise processing increases runtimes significantly. On the contrary, BiFormer can interpolate a 4K frame at once. Thus, for a fair comparison with the conventional methods, we compare the runtimes at 2K resolution, instead of 4K resolution.

Figure 8 shows interpolation results qualitatively. Due to large motions in the 4K frames, the conventional algorithms fail to predict the fingers in the middle image and the player in the bottom image properly, yielding ghost and deformation artifacts. Moreover, they cannot handle the fast motions of a car in the top image, causing blurry artifacts and missing object parts. In contrast, BiFormer reconstructs them more faithfully without noticeable artifacts.

4.4. Analysis

Let us analyze the VFI performance of the proposed algorithm on X4K1000FPS.

Bilateral process: In the bilateral attention module in Figure 4, there are six Swin blocks and three BCA blocks (one BCA-A block and two BCA+A blocks). Table 2 lists the performances when the BCA-A or BCA+A blocks are removed while the six Swin blocks are maintained. We see that the performance improves as more BCA blocks are used, which confirms the effectiveness of the BCA blocks.

Local motion refinement: Table 3 analyzes the impacts of two hyper-parameters of BBCVs in the local motion refinement: the size r of the search range and the maximum block size. First, the performance degrades when r is smaller than the default value 2. A bigger r (*e.g.* 4) increases mem-

Table 2. Ablation studies for the global motion estimation.

6 swin	BCA block			X4K1000FPS	
	BCA-A	BCA+A^{1st}	BCA+A^{2nd}	PSNR	SSIM
✓				30.94	0.8867
✓	✓			31.20	0.9086
✓	✓	✓		31.26	0.9158
✓	✓	✓	✓	31.32	0.9212

Table 3. Ablation studies for the local motion refinement.

	Setting	X4K1000FPS	
		PSNR	SSIM
Search range r	2 (default)	31.32	0.9212
	1	31.20	0.9037
Maximum block size	4×4 (default)	31.32	0.9212
	2×2	31.27	0.9161
	1×1	31.16	0.8914
Refinement scales	1/4, 1/2 (default)	31.32	0.9212
	1/2	31.15	0.8895
	1/4	30.88	0.8859

Table 4. Complexity analysis of the proposed algorithm.

	BiFormer	Upsampler	Synthesis	Total
#Parameters (millions)	9.68	0.78	0.72	11.2
4K Runtime (seconds)	0.02	0.47	1.74	2.23

ory complexity and makes it impossible to interpolate a 4K frame at once. Second, the maximum block size 4×4 means that the three BBCVs for 1×1, 2×2, and 4×4 block sizes are employed. The performance improves as more BBCVs are used, for a bigger block size enables the cost volume to consider larger motions more efficiently.

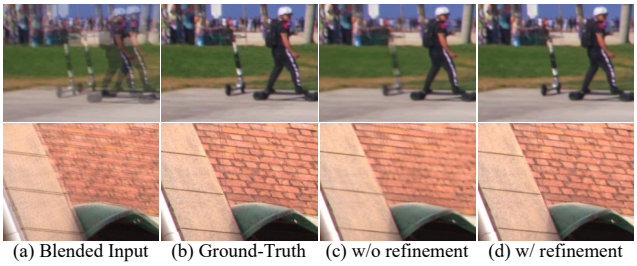
Table 3 also analyzes the impacts of refinement scales. In the default mode, we use the upsampler twice to refine the motion fields at $1/4$ and $1/2$ scales sequentially. If we refine them at the coarse scale of $1/4$ only, the performance degrades severely because fine motions are less accurately represented. If we instead do the refinement at the fine scale of $1/2$ directly, the performance gets better but is worse than the sequential refinement. This is because the sequential refinement takes advantage of both the reliability at the coarse scale and the accuracy at the fine scale.

Figure 9 illustrates that refined motion fields are more accurate than global motion fields. Also, Figure 10 shows that the local motion refinement improves VFI results by correcting errors in the global motion estimation. We see that motion errors on complicated texture and small objects in Figure 10(c) are reduced by the local motion refinement in Figure 10(d).

Computational complexity: Table 4 analyzes the complexity of the proposed algorithm, which consists of BiFormer, upsampler, and synthesis network. BiFormer uses 86% percent of parameters in the entire network. However, it operates at $1/8$ scale, so it takes only 0.02 seconds to estimate global motion fields. On the other hand, the upsampler works at $1/4$ and $1/2$ scales and requires a longer processing time, although it is implemented efficiently with only 0.78M parameters. The frame synthesis network demands the longest processing time to synthesize a 4K frame. It is implemented with 0.72M parameters.

(a) Blended Input (b) Global Motion (c) Refined Motion

Figure 9. Visualization of global and refined motion fields.



(a) Blended Input (b) Ground-Truth (c) w/o refinement (d) w/ refinement

Figure 10. Comparison of VFI results without and with the local motion refinement.

5. Conclusions

We proposed an effective 4K VFI algorithm based on BiFormer. First, we developed BiFormer to estimate global motion fields at a coarse scale. Second, we employed the upsampler to refine these global motion fields into final motion fields at a fine scale. Last, we used the synthesis network to warp the two input frames using the final motion fields, respectively, and synthesize an intermediate frame. It was shown that the proposed BiFormer algorithm provides excellent performance on 4K benchmark datasets.

Acknowledgments

This work was conducted by CARAI grant funded by DAPA and ADD (UD190031RD), supported partly by Samsung Electronics Co., Ltd. (No. IO201214-08156-01), and supported partly by the NRF grants funded by the Korea government (MSIT) (No. NRF-2021R1A4A1031864 and No. NRF-2022R1A2B5B03002310).

References

[1] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *ICCV*, pp. 14539–14548, Oct. 2021. 1, 2, 3, 6

[2] H. Sim, J. Oh, and M. Kim, "XVFI: Extreme video frame interpolation," in *ICCV*, pp. 14489–14498, Oct. 2021. 1, 2, 6

[3] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, pp. 1106–1125, Feb. 2019. 1, 2, 6

[4] 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, pp. 678–691, Feb. 2018. 1

[5] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *ECCV*, pp. 416–431, Sept. 2018. 1

[6] 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 *CVPR*, pp. 9000–9008, June 2018. 1, 2

[7] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "DeepStereo: Learning to predict new views from the world's imagery.," in *CVPR*, pp. 5515–5524, June 2016. 1

[8] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–10, 2016. 1

[9] 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. Circuit Syst. Video Technol.*, vol. 17, pp. 407–416, Apr. 2007. 1

[10] 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, pp. 4497–4509, Nov. 2013. 1

[11] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *ECCV*, pp. 434–450, Oct. 2016. 1

[12] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *ICCV*, pp. 4463–4471, Oct. 2017. 1, 2

[13] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *CVPR*, pp. 670–679, July 2017. 1

[14] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *ICCV*, pp. 261–270, Oct. 2017. 1, 2, 6

[15] 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, pp. 933–948, Mar. 2021. 1, 2

[16] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *CVPR*, pp. 1701–1710, June 2018. 1, 2

[17] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *AAAI*, pp. 8794–8802, Jan. 2019. 1, 2

[18] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *CVPR*, pp. 3703–3712, June 2019. 1, 2

[19] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *CVPR*, pp. 5437–5446, June 2020. 1, 2

[20] S. Gui, C. Wang, Q. Chen, and D. Tao, "FeatureFlow: Robust video interpolation via structure-to-texture generation," in *CVPR*, pp. 14004–14013, June 2020. 1

[21] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *AAAI*, p. 10607–10614, Feb. 2020. 1

[22] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *AAAI*, pp. 10663–10671, Feb. 2020. 1, 2, 6

[23] H. Lee, T. Kim, T.-y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *CVPR*, pp. 5316–5325, June 2020. 1, 2, 6

[24] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *ECCV*, pp. 109–125, Aug. 2020. 1, 2, 3, 4, 6

[25] P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, "Many-to-many splatting for efficient video frame interpolation," in *CVPR*, pp. 3553–3562, June 2022. 1, 2, 6

[26] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jiaya, "Video frame interpolation with transformer," in *CVPR*, pp. 3532–3542, June 2022. 1, 2, 6

[27] Z. Shi, X. Xu, X. Liu, J. Chen, and Y. Ming-Hsuan, "Video frame interpolation transformer," in *CVPR*, pp. 17482–17491, June 2022. 1, 2

[28] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *ICCV*, pp. 2758–2766, Dec. 2015. 1, 3

[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, pp. 2462–2470, July 2017. 1, 3

[30] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *CVPR*, pp. 4161–4170, July 2017. 1

[31] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *CVPR*, pp. 8934–8943, June 2018. 1, 3

[32] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *CVPR*, pp. 6489–6498, June 2020. 1

[33] S. Zhao, Y. Sheng, Y. Dong, E. I.-C. Chang, and Y. Xu, "MaskFlowNet: Asymmetric feature matching with learnable occlusion mask," in *CVPR*, pp. 6278–6287, June 2020. 1

[34] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," in *ECCV*, pp. 557–572, Aug. 2020. 1

[35] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *ECCV*, pp. 402–419, Aug. 2020. 1, 3, 5

[36] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: learning optical flow via global matching," in *CVPR*, pp. 8121–8130, June 2022. 1, 4

[37] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "FlowFormer: A transformer architecture for optical flow," in *ECCV*, p. 668–685, Oct. 2022. 1

[38] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *ECCV*, pp. 624–642, Oct. 2022. 2

[39] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless, "FILM: Frame interpolation for large motion," in *ECCV*, Oct. 2022. 2

[40] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "CDFI: Compression-driven network design for frame interpolation," in *CVPR*, pp. 8001–8011, June 2021. 2, 6

[41] C. Montgomery, "Xiph.org Video Test Media (derf's collection), the Xiph Open Source Community," *Online, https://media.xiph.org/video/derf*, 1994. 2, 6

[42] D. Ma, F. Zhang, and D. Bull, "BVI-DVC: A training database for deep video compression," *TMM*, pp. 1–1, Sept. 2021. 2, 6

[43] K. M. Fant, "A nonaliasing, real-time spatial transform technique," *IEEE Comput. Graph. Appl.*, vol. 6, pp. 71–80, Jan. 1986. 2

[44] F. A. Reda, D. Sun, A. Dundar, M. Shoeybi, G. Liu, K. J. Shih, A. Tao, J. Kautz, and B. Catanzaro, "Unsupervised video interpolation using cycle consistency," in *ICCV*, pp. 892–900, Oct. 2019. 2

[45] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press, 1990. 2

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inform. Process. Syst.*, pp. 5998–6008, Dec. 2017. 2, 3

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, May 2021. 2

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, pp. 10012–10022, Oct. 2021. 2

[49] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Adv. Neural Inform. Process. Syst.*, 2021. 3

[50] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," in *CVPR*, pp. 12009–12019, June 2022. 3, 4

[51] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Adv. Neural Inform. Process. Syst.*, p. 68–80, Dec. 2019. 4

[52] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *ICIP*, pp. 168–172, Nov. 1994. 6

[53] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *AAAI*, Feb. 2018. 6

[54] Y. Zhong, P. Ji, J. Wang, Y. Dai, and H. Li, "Unsupervised deep epipolar flow for stationary or dynamic scenes," in *CVPR*, pp. 12095–12104, June 2019. 6

[55] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *ECCV*, pp. 36–53, Sept. 2018. 6

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, May 2015. 6