

Received 1 September 2023, accepted 15 September 2023, date of publication 20 September 2023,
date of current version 26 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317515

RESEARCH ARTICLE

A Motion Refinement Network With Local Compensation for Video Frame Interpolation

KAIQIAO WANG AND PENG LIU 

Hainan Acoustics Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Haikou 570105, China

Corresponding author: Peng Liu (liup@dsp.ac.cn)

This work was supported in part by the Youth Innovation Promotion Association, Chinese Academy of Sciences (CAS); in part by the South China Sea Nova Project of Hainan Province; in part by the Important Science and Technology Project of Hainan Province under Grant ZDKJ2020010; and in part by the Frontier Exploration Project Independently Deployed by the Institute of Acoustics, Chinese Academy of Sciences under Grant QYTS202015 and Grant QYTS202115.

ABSTRACT Video frame interpolation (VFI) is a challenging yet promising task that involves synthesizing intermediate frames from two given frames. State-of-the-art approaches have made significant progress by directly synthesizing images using forward optical flow. However, these methods often encounter issues such as occlusion and pixel blurring when handling large motion scenes. To address these challenges, this paper proposes a novel approach based on the concept of local compensation. By adopting this approach, more refined optical flow estimation can be obtained, leading to higher-quality video frame interpolation results. Specifically, we introduce two modules, namely the Comprehensive Contextual Feature Extraction (CCFE) module and Motion-Guided Feature Fusion (MGFF) module, to enable local compensation of optical flow estimation. The CCFE module is designed to be embedded in each layer of the image pyramid structure. It aims to encourage the model to extract clean and sufficiently rich contextual information from the input images. On the other hand, the MGFF can guide the multi-source features fusion based on motion features, making the feature fusion of moving objects more precise, thus providing local compensation for optical flow estimation. Extensive experimental results demonstrate that incorporating our proposed modules into the baseline network significantly enhances the performance of video frame interpolation.

INDEX TERMS Video frame interpolation, comprehensive contextual feature extraction, motion-guided feature fusion, local compensation, motion refinement network.

I. INTRODUCTION

Video frame interpolation is an application of computer vision in the field of video enhancement, which has attracted significant attention from scholars in recent years. The purpose of VFI is to improve the frame rate of the original video by inserting intermediate frames between adjacent frames. VFI technology has versatile applications, including the conversion of videos into higher frame rates and the enhancement of visual effects [4]. Additionally, VFI technology can be employed in video compression [4], video editing [5], generating training data to learn how to synthesize

motion blur [6], and serving as an auxiliary task for optical flow estimation [7], [8], etc.

The most recent studies on VFI predominantly leverage deep neural networks (DNNs) as their primary methodology. These studies can be divided into flow-based and kernel-based methods. Kernel-based methods [9], [10], [11] synthesize the target frame by predicting the interpolation kernel for each pixel, while flow-based methods [12], [13], [14] estimate optical flow to perform frame warping and then synthesize the target frame. Although kernel-based methods are effective, they are limited to interpolating frames at a fixed time step, and their runtime increases linearly with the expected number of output frames. Flow-based methods establish dense correspondences between frames and apply warping to render the intermediate pixels, which


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



FIGURE 1. The ground truth Images vs. images generated by M2M-PWC. (a) The wooden stake in front of the car is not distorted, the tennis racket has distinct grids, and there are speckles along the edges of the racket. (b) The wooden stake in front of the car is distorted, the tennis racket is blurred, and there is no obvious grid and speckles.

can effectively reduce the multi-frame interpolation time and allow arbitrary-time interpolation. Hence, the flow-based video frame interpolation method has emerged as the predominant approach for arbitrary-time interpolation.

The current flow-based methods have achieved promising results in generating images with authenticity and inter-frame consistency. However, these methods, such as ABME [16], QVI [13], FLAVR [15], and SoftSplat [14], often employ increasingly complex networks, resulting in a large number of model parameters and increased computational complexity. In comparison, M2M-PWC [29] reaches an outstanding accuracy for arbitrary-time interpolation with fewer model parameters and lower computational complexity. Nevertheless, there is still room for further improvement to the M2M-PWC model. For instance, the M2M-PWC has obvious defects when synthesizing objects in large motion scenes. As shown in Fig. 1, comparing the ground truth image with the one generated by M2M-PWC, several issues can be observed. The surrounding wooden stake of the car undergoing large motion in the ground truth image remains undistorted, while in the image generated by M2M-PWC, the

corresponding positions of the wooden stake are distorted. In addition, the ground truth image shows clear grid patterns and spotted edges of the high-speed moving racket, whereas the racket in the image generated by M2M-PWC appears blurry. We believe that if the above issues can be improved, the performance of video frame interpolation in large motion scenes will also be further improved.

We have observed that the quality of synthesized frames mainly relies on the synthesis of moving objects, which need to consider two factors: one is the separation between the boundary of moving object and the background, and the other is pixel fusion inside moving objects. The former requires global pixel information, while the latter necessitates local pixel details. Therefore, it is essential to extract comprehensive features from the source images, encompassing both global and local information. Furthermore, the fusion strategy of multi-source features not only requires effective separation of boundary for moving object but also demands the fusion of internal object details. To address this problem, we propose a novel approach based on local compensation for refining optical flow estimation and developing a corresponding VFI algorithm. The introduced CCFE module and MGFF module both are specifically designed for local compensation. Our main contributions in this paper can be summarized as follows:

- We propose a CCFE module that can be embedded in each layer of the image pyramid structure to extract comprehensive features, including global and local features, from the source image.
- We propose an MGFF module, which utilizes optical flow estimation to guide the fusion of multi-source of features. It enables fine-grained feature fusion and provides local compensation for optical flow estimation.
- Experimental results show that the baseline network equipped with our modules can boost the VFI performance.

II. RELATED WORK

The kernel-based methods usually generate intermediate frame at fixed times, typically between input images, which limit arbitrary-time interpolation and linearly increases the processing time for multiple-frame interpolation. Flow-based methods explicitly estimate inter-frame motion and use it to warp between frames, aiding subsequent intermediate frame estimation. This means that the flow-based methods can perform arbitrary temporal interpolation and effectively reduce runtime for multi-frame interpolation. Therefore, most methods that support arbitrary-time interpolation are based on optical flow estimation. In general, flow-based methods can be divided into backward optical flow-based methods and forward optical flow-based methods.

The motion estimation module of the former results in backward optical flow, which does not start from the input time but starts from the middle time. Liu et al. [17] first used a CNN to estimate the backward optical flow and then used the backward warping operator to synthesize

the intermediate frame. Park et al. [18] modified the flow design process of the optical flow estimation network PWCNet [19] to make it suitable for solving the problem of calculating the optical flow. Zhang et al. [21] drew on the design of Recurrent Residual Pyramid in the optical flow estimation field to improve the performance of the backward optical flow estimation. Chen et al. [22] used a multi-scale motion estimation module to improve the accuracy of deformation synthesis. Huang et al. [20] developed an iteratively optimized backward optical flow estimation module, which has the advantages of simple design and fast speed.

The latter can be further divided into two categories. The first category is known as the methods based on indirect forward optical flow. The paradigm is as follows. It first estimates the forward optical flow then calculates the backward optical flow using the forward warping operator, and finally synthesizes with object linear or higher-order motion assumptions. Jiang et al. [12] was the first to use this design for video interpolation. Under the assumption of linear motion of objects, it can quickly calculate the optical flow starting from any intermediate time point using the forward warping method, thus achieving fast multi-frame interpolation. Bao et al. [23] combined adaptive convolution with deformation and used a backward warping operator with adaptive weights for frame synthesis. Later, Bao et al. [24] used the reciprocal of the depth map as the weights combined with the forward warping operator to propose the highly acclaimed and visually effective DAIN. In addition, Xu et al. [13] based on the quadratic motion hypothesis, considering the one-sidedness of the linear motion hypothesis. Liu et al. [25] based on the cubic motion hypothesis. Zhang et al. [21] fully utilized convolutional networks to enhance nodes in the process with optimization space and has the advantages of the convolutional synthesis method. Sim et al. [26] focuses on solving the frame interpolation requirements of 4K resolution videos.

The second category is referred to as the methods based on direct forward optical flow. It directly synthesizes images through forward optical flow. These methods have a simple and reasonable process design, which also avoids the problem of ghosting introduced by backward warping. However, this type of method also needs to deeply consider the issues of voids and conflicts caused by forward warping. Niklaus and Liu [27] proposed a context-aware synthesis method, which can warp the input frames and their pixel context information and use them to interpolate high-quality intermediate frames. Then, Niklaus and Liu [14] proposed SoftSplat to seamlessly handle hole and conflict issues in the forward warping results and can achieve high-performance interpolation at any time point. Li et al. [28] solved the challenges of “lack of texture” and “nonlinear and large motion” in a coarse-to-fine manner. Hu et al. [29] proposed M2M-PWC based on SoftSplat, estimated multiple bidirectional flows, which can directly warp pixels to the desired time step and then fuse any overlapping pixels.

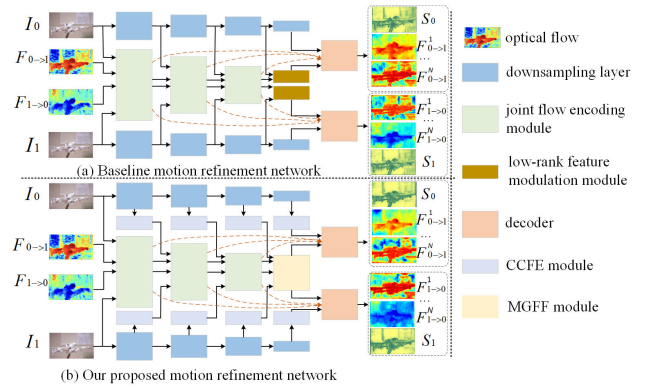


FIGURE 2. An overview of our plug-in modules. The details of our CCFE module and MGFF module are further illustrated in Fig. 3 and 4, respectively.

Overall, these above methods have advantages and disadvantages respectively. The main disadvantage of the methods based on backward optical flow is that they require abundant computation time for multi-frame interpolation. The methods based on indirect forward optical flow can solve this problem and achieve faster multi-frame interpolation. However, this type of methods requires bidirectional optical flow during the frame synthesis process, which doubles the computational load compared to other methods that only need to calculate unidirectional optical flow. Although these two types of methods can avoid pixel conflict issues (such as occlusion and blur), and achieve satisfactory performance in frame synthesis, the high computational complexity limits their application prospects in the real world. In contrast, methods based on direct forward optical flow struggle to easily achieve desirable frame synthesis results. However, due to their efficient computational approach, they have become one of the practical deployment solutions. Considering this, our proposed method is developed based on direct forward optical flow. While retaining the efficient computational characteristics, it enhances frame synthesis performance and holds promise for practical application. Detailed explanations will be provided in the following sections.

III. PROPOSED METHOD

A. OVERVIEW

Given two consecutive frames I_0 and I_1 of a video, the goal of VFI is to synthesize an intermediate frame I_t at a time between the given input frames and the expected time step $t \in (0, 1)$. To achieve this goal, we first use existing optical flow estimation methods to estimate the coarse forward optical flow $\{F_{0 \rightarrow 1}, F_{1 \rightarrow 0}\}$ between the two input frames. Then, we utilize a refined network to improve and enhance the original optical flow estimation via local compensation. This process generates the refined bidirectional motion fields $\{F_{0 \rightarrow 1}^n, F_{1 \rightarrow 0}^n\}_{n=1}^N$ at full resolution, as well as corresponding pixel reliability scores $\{S_0, S_1\}$.

$$\{F_{0 \rightarrow 1}^n, F_{1 \rightarrow 0}^n\}_{n=1}^N, \{S_0, S_1\} = \varphi(I_0, I_1, F_{0 \rightarrow 1}, F_{1 \rightarrow 0}) \quad (1)$$

under the assumption of linear motion, we scale the motion vector of each pixel according to the expected time step.

$$F_{0 \rightarrow t}^n(i_0) = t \cdot F_{0 \rightarrow 1}^n(i_0) \quad (2)$$

$$F_{1 \rightarrow t}^n(i_1) = (1 - t) \cdot F_{1 \rightarrow 0}^n(i_1) \quad (3)$$

here, i_0 and i_1 respectively indicate the i -th source pixel from I_0 and I_1 . Then, a source pixel i_s will be warped to $i_{s \rightarrow t}^n$ by its n -th motion vector $F_{s \rightarrow t}^n$ through a forward warping operation ϕ .

$$i_{s \rightarrow t}^n = \phi(i_s, F_{s \rightarrow t}^n) \quad (4)$$

where $s \in (0, 1)$ denotes the source frame. To address the issues of overlap and holes, we simulate the motion of each source pixel using multiple motion vectors. We use $N (N > 1)$ sub-motion vectors to forward warp each pixel in the source to time, resulting in a collection of warped pixels.

$$\hat{I}_{s \rightarrow t} = \bigcup_{n=1}^N \hat{I}_{s \rightarrow t}^n \quad (5)$$

Finally, using the pixel warping and fusion strategy Ψ , we obtain the final synthesized frame.

$$I_t = \Psi(\hat{I}_{s \rightarrow t}, S_s) \quad (6)$$

B. MOTION REFINEMENT NETWORK WITH LOCAL COMPENSATION

The optical flow-based VFI algorithm generates intermediate frames using two source frames and their corresponding optical flow estimations. The two source frames provide foreground and background information for the synthesized intermediate frame [30], [31]. Typically, moving objects between the two source frames are considered as foreground, while relatively static objects are considered as background. The optical flow estimation describes the relative motion between the foreground and background. Compared to the original optical flow estimation, the refined optical flow estimation obtained by our proposed motion refinement network effectively separates foreground and background information. Furthermore, it combines multiple motion fields in a meaningful way, resulting in accurate representations of the motion prototypes of foreground objects. These advances are primarily attributed to the implementation of two modules, which will be explained in the following section.

1) COMPREHENSIVE CONTEXTUAL FEATURE EXTRACTION MODULE

The CCFE module is highly modular, as shown in Fig. 2, which can be embedded after each down-sampling layer of the image pyramid to enhance the extraction of contextual features. Its details are shown in Fig. 3. It includes a multi-scale feature aggregation mechanism and a channel attention mechanism. The multi-scale feature aggregation mechanism enables the network to capture large motion information, accurately capturing both global and local

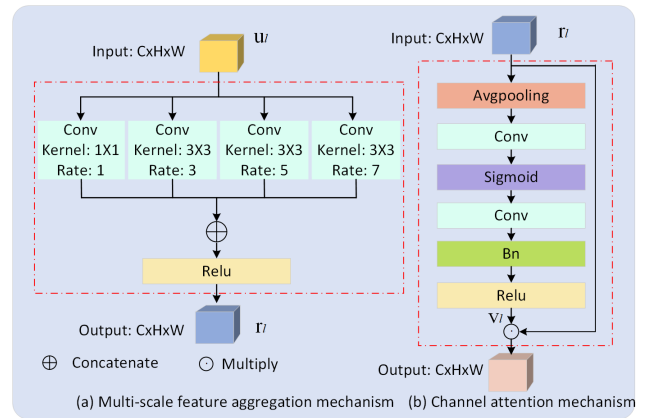


FIGURE 3. An illustration of CCFE module. (a) The multi-scale feature aggregation mechanism is mainly composed of four dilated convolution branches with different dilation rates. Their dilation rates are 1, 3, 5, and 7, respectively. (b) The channel attention is mainly used for feature recalibration.

information about moving objects. However, the extracted source frame information usually contains noise. To mitigate this noise interference, the channel attention mechanism is employed to guide the model to learn important information and suppress the noise information.

a: MULTI-SCALE FEATURE AGGREGATION MECHANISM

As shown in Fig. 3 (a), the multi-scale feature aggregation mechanism is mainly composed of four dilated convolution branches with different dilation rates. This structure can receive information from different receptive fields, thereby extending features to different scale-spaces. Define the input feature map of l -th layer as $u_l \in R^{C \times H \times W}$, where C represents the number of channels and $H \times W$ represents the size of the input feature map. First, the feature map u_l is simultaneously sent to four different branches to generate new feature maps. Then, we concatenate these features generated by different branches along the channel dimension and pass them through the ReLU activation function to obtain multi-scale information r_l .

The dilation rates of the four convolutions are 1, 3, 5, and 7, respectively. As a result of the smaller dilation rates can make the convolution kernel focus on local patterns, while larger dilation rates allow the convolution kernel to “see” larger areas of the input image. In this way, the model can capture not only the global information of the source frame but also the local information, which is essential for local compensation of refining optical flow estimation in large motion scenes.

b: CHANNEL ATTENTION MECHANISM

Generally, the feature map extracted by convolutional neural network will contain a certain amount of noise. Therefore, to suppress noise interference and select useful features for synthesizing intermediate frames, we apply channel attention to recalibrate the features after each multi-scale feature

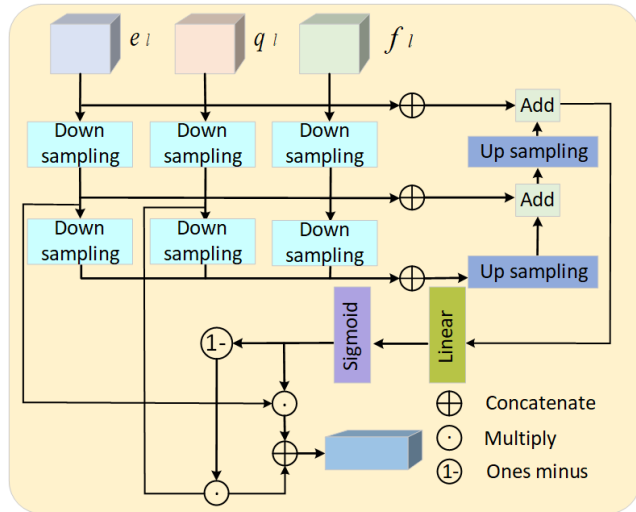


FIGURE 4. An illustration of motion-guided feature fusion module. e_l , q_l , and f_l are branch features, pyramid features, and optical flow estimation features, respectively. Down denotes the downsampling function implemented by 2D convolution, and Up denotes the upsampling function implemented by bilinear interpolation.

aggregation mechanism. As shown in Fig. 3 (b), the channel attention mainly consists of a global average pooling, two convolutional layers, and a product operation.

Assuming $r_l \in R^{C \times H \times W}$ is the feature of l -th layer obtained through multi-scale feature aggregation mechanism. Firstly, the mechanism performs the 2D global average pooling operation to compress the feature map along the spatial dimension into a feature vector, which theoretically has the global receptive field with input characteristics. Secondly, the feature vector will be transformed into learnable parameters v_l through convolutional computation. v_l is the learned coefficient, which represents the importance of each channel. Finally, the product operation is performed by weighting the original feature with the learned importance coefficient for each corresponding feature channel, thus achieving the rescaling of the original feature: $q_l = v_l \cdot r_l$

2) MOTION-GUIDED FEATURE FUSION MODULE

The MGFF module utilizes motion features to dynamically guide the fusion of source frame features and branch features. It enables the obtained fusion features to effectively fuse source frame information, branch mixed information, and motion information. As a result, more accurate optical flow estimation vectors can be obtained.

As shown in Fig. 2, the features on the branch are mixed features that fuse information from two input frames and original optical flow estimation. The image pyramid features are down-sampled features of the input image. To handle the fusion of the two under different motion patterns, we propose to fuse branch features and image pyramid features under the guidance of corresponding motion features. That is the motion-guided feature fusion module, which replaces the LFM module in M2M-PWC.

The structure of MGFF is shown in Fig. 4. MGFF dynamically fuses the features e_l on the branches and the image pyramid features q_l based on the motion pyramid features f_l of l -th layer, where $l = 4$. Firstly, MGFF obtains three types of multi-scale weighted features:

$$w_1 = \mathcal{F}_{l1}([e_l, q_l, f_l]) \quad (7)$$

$$w_2 = \mathcal{F}_{l2}([\mathcal{D}_2(e_l), \mathcal{D}_2(q_l), \mathcal{D}_2(f_l)]) \quad (8)$$

$$w_3 = \mathcal{F}_{l3}([\mathcal{D}_4(e_l), \mathcal{D}_4(q_l), \mathcal{D}_4(f_l)]) \quad (9)$$

where $\mathcal{F}(\cdot)$ is the linear layer, $[\cdot]$ represents concat features with channel dimension, $\mathcal{D}(\cdot)_x$ represents the downsampling function implemented by 2D convolution with stride 2, and x is the downsampling ratio. Then the weighted features are fused:

$$w_4 = w_2 + \mathcal{U}_2(w_3) \quad (10)$$

$$w_o = \sigma(\mathcal{F}_{l4}(w_1 + \mathcal{U}_2(w_4))) \quad (11)$$

where $\mathcal{U}(\cdot)_x$ is the upsampling function implemented by bilinear interpolation, x is the upsampling factor. Finally, we obtain the fused feature by:

$$e_l = e_l \cdot w_o + q_l \cdot (1 - w_o) \quad (12)$$

This design prompts the network to automatically adjust the weights based on the optical flow features, thereby dynamically fusing the mixed features obtained from the previous layer and the image features. As shown in Fig. 2, after this process, the decoder can be applied to obtain the enhanced optical flow estimation and its pixel reliability scores. Fig. 5 illustrates a compelling comparison between the optical flow estimations before and after incorporating local compensation. This visualization depicts that the optical flow estimation post-local compensation offers a more intricate and detailed depiction of motion.

C. PIXEL WARPING AND FUSION

We finally synthesize the intermediate frame based on $\{F_{0 \rightarrow 1}^n, F_{1 \rightarrow 0}^n\}_{n=1}^N$ and $\{S_0, S_1\}$ obtained above through forward warping. Since the synthesis strategy proposed in [30] directly operates on the pixel color domain based on the framework of multi-frame pixel fusion and learning-based pixel reliability scores, building upon the work of [14]. As a result, this model can efficiently synthesize high-quality intermediate frames. Therefore, this paper adopts its synthesis strategy as the subsequent synthesis method.

IV. EXPERIMENTS

A. TRAINING SETTINGS AND DETAILS

1) COMPARISON METHODS

We conducted comparative analysis to our proposed method and several recent approaches, including QVI [13], FLAVR [15], ABME [16], DAIN [17], SepConv [9], RIFE [20], SoftSplat [14], and M2M-PWC [29]. To ensure fairness, we reproduced the official publicly available codes of QVI, FLAVR, ABME, DAIN, SepConv, RIFE, and M2M-PWC. Although we encountered difficulties in accessing the

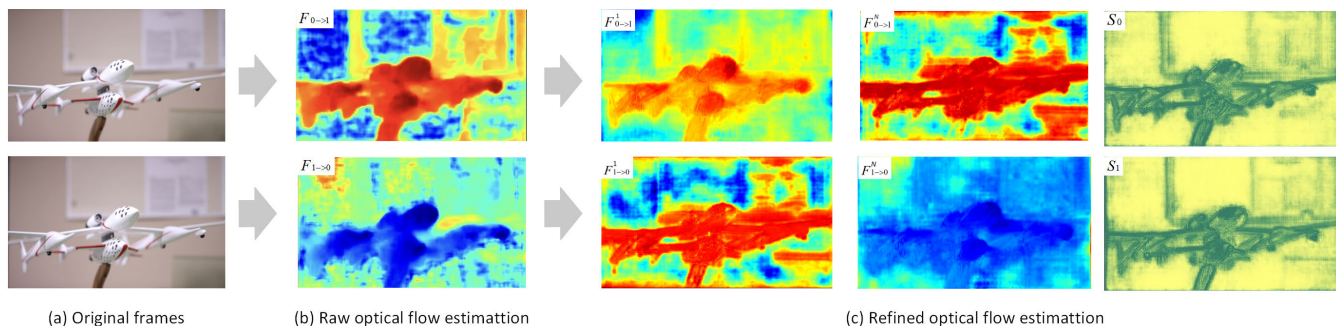


FIGURE 5. Visualization of optical flow estimation before and after refinement. (a) The original frame input into the model. (b) The raw optical flow estimation obtained through PWC-Net. (c) The refined optical flow estimation obtained by our proposed method.

code for SoftSplat, we made every effort to replicate it and compare its experimental results directly with those of our proposed method.

2) DATASETS

We adopted the same dataset partitioning approach as [29], where we trained our model on triplets extracted from the Vimeo90K [40] dataset. Subsequently, we evaluated the performance of our model on various datasets, which include:

- Vimeo90K: This dataset comprises 3782 triplets, with each image having a resolution of 448×256 .
- UCF101 [32]: Derived from a collection of human action videos, this dataset has been organized by [33] and consists of 379 triplets. Each image in this dataset has a resolution of 256×256 .
- ATD12K [33]: This dataset contains 2000 triplets extracted from various animated videos. The images in this dataset have a resolution of 960×480 .
- DAVIS [39]: Originating from a dynamic video, the Davis dataset consists of 92 different scenes, comprising a total of 6208 images. Each image has a resolution of either 854×480 or 910×480 .
- X-TEST [26]: Extracted from the X4K1000FPS dataset, X-TEST contains 15 scenes from 4K videos captured at 1000 FPS. The original resolution of this dataset is denoted as X-TEST(4K), while a downsampled version obtained through downsampling is referred to as X-TEST(2K) [26].

3) METRICS

We utilize the SSIM [34] and PSNR [35] metrics widely used in most VFI work to evaluate the performance of our proposed method. SSIM assesses image similarity based on brightness, contrast, and structural aspects. PSNR is the peak signal-to-noise ratio, which describes the relationship between the maximum signal and background noise. The higher the values of SSIM and PSNR, the better the quality of the synthesized image.

4) IMPLEMENT DETAILS

All experiments presented in this study were conducted using the PyTorch [36] framework and executed on an NVIDIA A100 Tensor Core GPU. For optimization, we employed the Adam optimizer with a weight decay of $1e-4$. The model was trained for 400,000 iterations, utilizing a batch size of 64. The training dataset comprised 51,312 triplets extracted from the Vimeo90K dataset and involved various random data augmentations, including spatial and temporal flipping, color jittering, and random cropping of 256×256 patches. End-to-end supervised training was performed using Charbonnier loss [37] and census loss [38].

B. QUANTITATIVE ANALYSIS

1) $\times 2$ FRAME INTERPOLATION

Table 1 presents a comparison of the performance between our proposed method and the other methods for $\times 2$ frame interpolation across multiple datasets. The methods are categorized into two groups based on their capability for arbitrary-time interpolation, where the upper half of the table represents methods that do not support arbitrary-time interpolation, while the lower half includes methods that do. To provide a holistic evaluation, we also include the corresponding parameter sizes and inference speeds for each method. The inference speed is calculated based on images from the Davis dataset, with each image cropped to a resolution of 480×854 . Notably, the text in bold font indicates that the method achieved the best results in the respective dataset, while the text with an underline indicates the second-best result.

In terms of the PSNR metric, the non-arbitrary-time interpolation methods generally demonstrate slightly better performance compared to the arbitrary-time interpolation methods. However, when considering the SSIM metric, the performance between these two categories is comparable. Upon comparing our proposed model with other arbitrary-time interpolation methods, we found that it achieved suboptimal results on the Vimeo90K, UCF101, and ATD12K datasets, while obtaining the best performance

TABLE 1. Quantitative comparisons of our proposed model with state-of-the-art VFI models.

Method	Parameter size (M)	Speed (ms/f)	Arbitrary	Vimeo90K		UCF101		ATD12K		Davis		X-TEST(2K)	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SepConv (2017) [9]	21.60	118	×	33.79	0.970	34.78	0.967	27.4	0.950	26.21	0.857	34.77	0.919
RIFE (2022) [20]	9.80	23	×	35.51	0.978	35.25	0.969	<u>28.59</u>	<u>0.953</u>	27.3	0.866	36.15	0.962
ABME (2021) [16]	18.10	428	×	<u>36.18</u>	0.981	35.38	<u>0.970</u>	28.71	0.959	27.48	0.875	<u>35.18</u>	<u>0.964</u>
FLAVR (2023) [15]	42.05	1232	×	36.25	0.975	<u>33.31</u>	0.971	26.62	0.942	<u>27.43</u>	<u>0.874</u>	36.81	0.982
DAIN (2017) [17]	24.00	898	✓	34.71	0.976	35.00	0.968	27.38	0.955	26.12	0.87	35.97	0.940
QVI (2019) [13]	29.23	282	✓	35.15	0.971	32.89	0.970	28.13	0.960	27.17	0.874	<u>35.76</u>	0.960
SoftSplat (2020) [14]	12.09	153	✓	36.10	0.98	35.39	0.970	28.22	0.957	27.36	0.877	36.62	0.944
M2M-PWC (2022) [29]	7.61	41	✓	35.40	0.978	35.17	0.970	29.03	0.959	<u>27.28</u>	<u>0.879</u>	36.45	<u>0.967</u>
Ours	<u>7.94</u>	<u>46</u>	✓	<u>35.52</u>	<u>0.978</u>	<u>35.31</u>	<u>0.970</u>	<u>28.77</u>	<u>0.959</u>	27.41	0.881	36.78	0.970

on the Davis and X-TEST datasets. These five datasets encompass diverse resolutions and FPSs, indicating that our proposed model can adapt effectively to various real-world scenarios and exhibit strong generalization abilities.

Notably, among the arbitrary-time interpolation methods, SoftSplat performs well, achieving the best results on the Vimeo90K and UCF101 datasets. However, this high performance comes at the cost of a larger parameter size and slower inference speed. Specifically, the parameter size of SoftSplat is approximately 1.59 times that of M2M-PWC, and its inference speed is roughly 3.7 times slower than M2M-PWC. In contrast, our proposed model only incurs a modest increase of 0.33 M parameters compared to M2M-PWC, resulting in a 5 ms slowdown in inference speed. It can be seen that our proposed model is still lightweight, retaining the advantages of direct forward optical flow-based methods while better addressing the issues of existing VFI, thereby further improving frame synthesis performance.

2) $\times 8$ FRAME INTERPOLATION

Arbitrary-time VFI is important in frame-rate conversion. We present the performance of our proposed method and the comparison methods for $\times 8$ frame interpolation in Table 2. In this experiment, we specifically focused on high-resolution datasets, namely X-TEST (2K) and X-TEST (4K), which pose more significant challenges compared to low-resolution datasets. The reported inference speed in Table 2 was obtained by training the model on the Vimeo90K dataset and testing it on the X-TEST (2K) dataset. Specifically, the model takes two frames with a resolution of 1080×2048 from the X-TEST (2K) dataset as input and generates 7 intermediate frames. It is evident that our proposed method has significant speed advantages and outperforms other comparison methods on the X-TEST (2K) and X-TEST (4K) datasets. Although M2M-PWC demonstrates slightly faster speed than our proposed model, its PSNR and SSIM metrics are not as impressive. On the X-TEST (2K) dataset, these two metrics are 0.31 and 0.018 lower than our proposed model, respectively, while on the X-TEST (4K) dataset, they are 0.04 and 0.025 lower than our proposed model, respectively. And this disadvantage can also be apparently reflected in visualization, which will be shown in the next subsection. On the other hand, QVI achieves comparable

TABLE 2. Quantitative analysis results for $\times 8$ frame interpolation on the X-TEST dataset. The inference speed was obtained by testing models on the X-TEST (2K) dataset.

Method	Speed (ms/f)	X-TEST(2K)		X-TEST(4K)	
		PSNR	SSIM	PSNR	SSIM
SepConv (2017) [9]	1638	25.70	0.800	23.94	0.794
DAIN (2017) [17]	5250	29.33	0.910	26.78	0.807
QVI (2019) [13]	873	31.57	<u>0.938</u>	28.42	<u>0.934</u>
RIFE (2022) [20]	234	27.49	0.806	24.67	0.797
ABME (2021) [16]	4426	30.65	0.912	30.16	0.879
FLAVR (2023) [15]	3489	29.89	0.884	28.25	0.924
SoftSplat (2020) [14]	664	29.73	0.824	25.48	0.725
M2M-PWC (2022) [29]	115	<u>32.07</u>	0.923	<u>30.81</u>	0.912
Ours	<u>136</u>	32.38	0.941	30.85	0.937

SSIM performance to our model but at the expense of being 6.4 times slower in terms of speed. In contrast, our model achieves an acceptable trade-off between accuracy and speed in the arbitrary-time frame interpolation task through the two proposed modules.

3) INTERPOLATION WITH DIFFERENT TIME STEPS

We assessed the robustness of our proposed model by expanding the $\times 8$ frame interpolation experiments and evaluating its interpolation performance with different time steps. The experiments were conducted on the X-TEST (2K) dataset, and the PSNR values of all models were calculated when inserting frames at various time steps. Fig. 6 illustrates the experimental results, indicating that our proposed method achieves optimal performance at each time step.

Furthermore, in general, all models tend to exhibit better frame interpolation performance at the edge time steps, while demonstrating relatively inferior performance at the middle time step. Compared with models represented by tortuous curves such as ABME and DAIN, the curves related to our method are smooth. So to speak, our model stably delivers excellent frame interpolation performance across different time steps, showcasing its temporal consistency and reliability.

C. QUALITATIVE ANALYSIS

1) HANDLING OF OCCLUSION ISSUE

Fig. 7 provides a visualization comparison between M2M-PWC and our proposed model. As previously

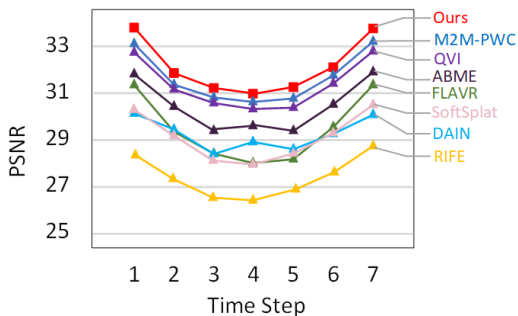


FIGURE 6. Evaluating multi-frame interpolation. Per-frame accuracy for $\times 8$ interpolation on X-TEXT(2K). Best viewed in red color.

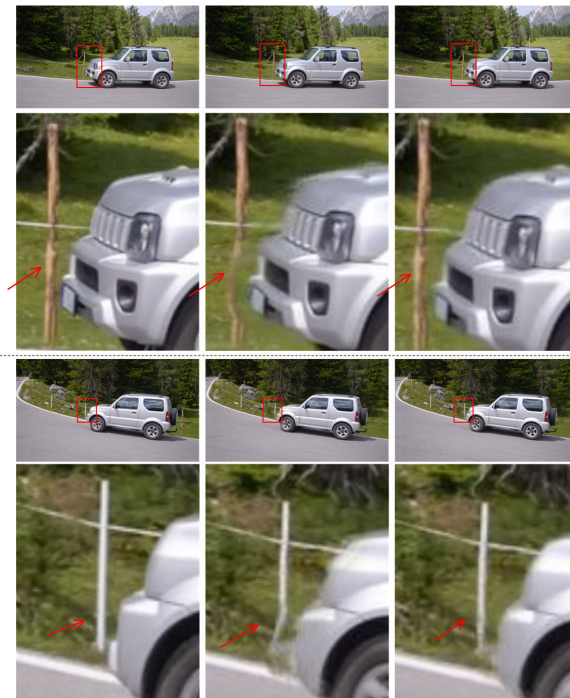
mentioned, M2M-PWC is currently recognized as one of the best VFI methods with arbitrary-time interpolation, exhibiting an overall good performance in VFI. However, upon closer examination of the comparison details, differences can be observed in the handling of occlusion between the two models. In the second row of Fig. 7, the images synthesized by M2M-PWC distort the originally undistorted wooden stake. This is a typical occlusion problem that exists in the methods based on direct forward optical flow. In contrast, the image (c) synthesized by our proposed model closely resembles the source frame, with the wooden stake in corresponding positions remaining undistorted. Similarly, in the synthesized image by M2M-PWC in the fourth row of Fig. 7, the white stake that the car is about to pass through is distorted, while the white stake we synthesized remains undistorted.

Based on these observations, we can conclude that our proposed model accurately captures pixel movement details between frames through local compensation, resulting in refined optical flow estimation. This enables our model to better handle occlusion issues and achieve high-quality video interpolation.

2) HANDLING OF PIXEL BLUR ISSUE

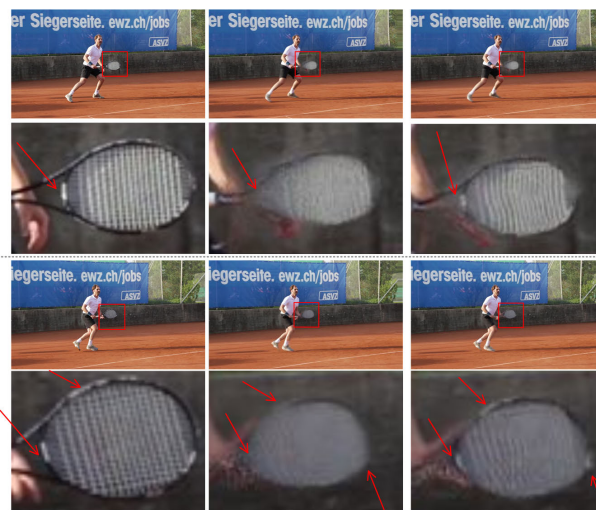
As mentioned earlier, methods based on direct forward optical flow, although efficient in achieving multi-frame interpolation, often encounter the issue of blur. Solving this blur problem through algorithm design has been a continuous research focus. As shown in Fig. 8, we demonstrate the performance of our model in handling this problem with some challenging pixel synthesis examples.

Fig. 8 (a) is the source frame, in which a man is holding a Tennis racquet for a large movement with large pixel displacement. The original tennis racquet has square patterns, and the edge of the racquet exhibits white speckles. Fig. 8 (b) is the image generated by M2M-PWC. It is apparent that the edges and squares of the tennis racquet appear blurred and lack clarity. However, in Fig. 8 (c), generated by our proposed model, the squares and edge speckles of the tennis racquet are well-preserved. It is apparent that synthesizing complex-shaped objects in scenes with large motion remains a challenge for methods based on direct forward optical



(a) Ground truth (b) M2M-PWC (c) Ours

FIGURE 7. Comparison between M2M-PWC and our model in handling occlusion issue. Our proposed model exhibits superior performance in accurately separating large motion objects from the background. The images generated by our model show no distortion in both the wooden stake and the white stake.



(a) Ground truth (b) M2M-PWC (c) Ours

FIGURE 8. Comparison between M2M-PWC and our model in handling pixel blur issue. The images generated by our model retain some of the grid patterns and speckles of the tennis racket, whereas M2M-PWC does not.

flow. Our proposed model, however, has made further improvements based on the base model. This improvement can be attributed to the inclusion of the MGFF and CCFF modules in our proposed model. These modules effectively

TABLE 3. Results of ablation experiment.

Method	Vimeo90K		UCF101		ATD12K		Davis		X-TEST(2K)		Avg	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
#1	33.81	0.951	33.79	0.928	26.93	0.936	25.75	0.858	34.56	0.941	30.97	0.923
#2	35.13	0.969	34.98	0.958	28.27	0.948	27.17	0.874	35.98	0.967	32.31	0.943
#3	35.27	0.976	35.16	0.965	28.44	0.947	27.24	0.878	36.11	0.966	32.44	0.946
Ours	35.52	0.978	35.31	0.970	28.77	0.956	27.41	0.881	36.78	0.970	32.76	0.951

address the issue of pixel blur, resulting in the generation of frames with more realistic details.

D. ABLATION EXPERIMENT

To investigate the contributions of the CCFE module and MGFF module to our model, we conducted ablation experiments, and the results are presented in Table 3. In the table, #1 represents the model with the CCFE module removed and the MGFF module replaced with a convolution operation, #2 represents the model with the CCFE module removed, and #3 represents the model with the MGFF module replaced with a convolution operation. The experiments conducted on five datasets demonstrate that both the CCFE and MGFF modules contribute to the performance indicators of our proposed model. Specifically, the average contributions of CCFE+MGFF to PSNR and SSIM across the five datasets are 1.79 and 0.29, respectively. Furthermore, from the last three rows of the table, it can be observed that the CCFE module enhances the two indicators of the model by 0.45 and 0.008, respectively, while the MGFF module improves the two indicators by 0.32 and 0.005, respectively. These results highlight the importance and effectiveness of both the CCFE and MGFF modules in enhancing the performance of our model.

V. CONCLUSION

In this paper, we introduce the CCFE module and MGFF module and demonstrate their effectiveness in frame interpolation applications through extensive experiments. The model equipped with these two modules can obtain more accurate optical flow estimation, enabling better handling of occlusion and blurring issues in large motion scenes. Nevertheless, there is room for refinement in terms of image generation clarity and model inference speed. Future research endeavors will focus on enhancing these aspects to optimize our proposed model further.

REFERENCES

- [1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.
- [2] D. Rufenacht, R. Mathew, and D. Taubman, "Occlusion-aware temporal frame interpolation in a highly scalable video coding setting," *APSIPA Trans. Signal Inf. Process.*, vol. 5, no. 1, p. e8, 2016.
- [3] M. Ogaki, T. Matsumura, K. Nii, M. Miyama, K. Imamura, and Y. Matsuda, "Frame rate up-conversion using hoe (hierarchical optical flow estimation) based bidirectional optical flow estimation," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 6, p. 52, 2012.
- [4] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 416–431.
- [5] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," 2018, *arXiv:1808.03232*.
- [6] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6840–6848.
- [7] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands: Springer, Oct. 2016.
- [8] J. Wulff and M. J. Black, "Temporal interpolation as an unsupervised pretraining task for optical flow estimation," in *Proc. 40th German Conf. Pattern Recognit. (GCPR)*, Stuttgart, Germany. Springer, 2019, pp. 567–582.
- [9] 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.
- [10] 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.
- [11] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 10663–10671, 2020.
- [12] 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.
- [13] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1645–1654.
- [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. 5436–5445.
- [15] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, "FLAVR: Flow-agnostic video representations for fast frame interpolation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 2070–2081.
- [16] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14519–14528.
- [17] 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. 4473–4481.
- [18] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Springer, Aug. 2020, pp. 109–125.
- [19] 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.
- [20] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2022*, pp. 624–642.
- [21] H. Zhang, Z. Yang, and R. Wang, "A flexible recurrent residual pyramid network for video frame interpolation," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020*, pp. 474–491.
- [22] Z. Chen, R. Wang, H. Liu, and Y. Wang, "PDWN: Pyramid deformable warping network for video interpolation," *IEEE Open J. Signal Process.*, vol. 2, pp. 413–424, 2021.

- [23] 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.
- [24] 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.
- [25] Y. Liu, L. Xie, L. Siyao, W. Sun, Y. Qiao, and C. Dong, "Enhanced quadratic video interpolation," in *Computer Vision—ECCV 2020 Workshops*, Glasgow, U.K. Springer, Aug. 2020, pp. 41–56.
- [26] H. Sim, J. Oh, and M. Kim, "XVFI: Extreme video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14469–14478.
- [27] 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.
- [28] L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. C. Loy, and Z. Liu, "Deep animation video interpolation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6587–6595.
- [29] P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, "Many-to-many splatting for efficient video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3543–3552.
- [30] C. Vondrick, P. Hamed, and T. Antonio, "Generating videos with scene dynamics," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 613–621.
- [31] M. Sun, W. Wang, X. Zhu, and J. Liu, "MOSO: Decomposing motion, scene and object for video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 18727–18737.
- [32] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
- [33] H. Li, Y. Yuan, and Q. Wang, "Video frame interpolation via residue refinement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2613–2617.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
- [36] *PyTorch*. Accessed: Mar. 8, 2019. [Online]. Available: <http://http://pytorch.org/>
- [37] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. 1st Int. Conf. Image Process.*, 1994, pp. 168–172.
- [38] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI*, 2018, pp. 7251–7259.
- [39] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.
- [40] 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, Aug. 2019.



KAIQIAO WANG received the master's degree in management science and engineering from the Dalian University of Technology, Dalian, China, in 2021. She is currently with the Hainan Acoustics Laboratory, Institute of Acoustics, Chinese Academy of Sciences. Her research interests include computer vision and multimedia signal processing.



PENG LIU received the B.S. degree in communication engineering from Hainan University, Haikou, China, in 2011, and the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2016.

He is currently with the Hainan Acoustics Laboratory, Institute of Acoustics, Chinese Academy of Sciences, and has been an Associate Professor, since 2018. His research interests include computer vision, multimedia signal processing, and information forensics.

• • •