JNMR: Joint Non-Linear Motion Regression for Video Frame Interpolation

Meiqin Liu, Chenming Xu, Chao Yao, Chunyu Lin, *Member, IEEE*, and Yao Zhao, *Fellow, IEEE*

***Abstract*— Video frame interpolation (VFI) aims to generate predictive frames by motion-warping from bidirectional references. Most examples of VFI utilize spatiotemporal semantic information to realize motion estimation and interpolation. However, due to variable acceleration, irregular movement trajectories, and camera movement in real-world cases, they can not be sufficient to deal with non-linear middle frame estimation. In this paper, we present a reformulation of the VFI as a joint non-linear motion regression (JNMR) strategy to model the complicated inter-frame motions. Specifically, the motion trajectory between the target frame and multiple reference frames is regressed by a temporal concatenation of multi-stage quadratic models. Then, a comprehensive joint distribution is constructed to connect all temporal motions. Moreover, to reserve more contextual details for joint regression, the feature learning network is devised to explore clarified feature expressions with dense skip-connection. Later, a coarse-to-fine synthesis enhancement module is utilized to learn visual dynamics at different resolutions with multi-scale textures. The experimental VFI results show the effectiveness and significant improvement of joint motion regression over the state-of-the-art methods. The code is available at https://github.com/ruhig6/JNMR.**

***Index Terms*— Video frame interpolation, multi-variable non-linear regression, motion estimation, interpolation modeling, deformable convolution.**

I. Introduction

THE purpose of video frame interpolation (VFI) is to generate new middle frames from existing reference frames. It is essential for various applications, such as slow-motion generation [1], frame compensation in video compression [2], [3], [4], [5], frame recovery [6], [7], etc. Typically, high-level global motions and subtle variations of the synthesis frame have the same importance. Therefore, it is challenging to accurately estimate the complicated motions for frame interpolation modeling.

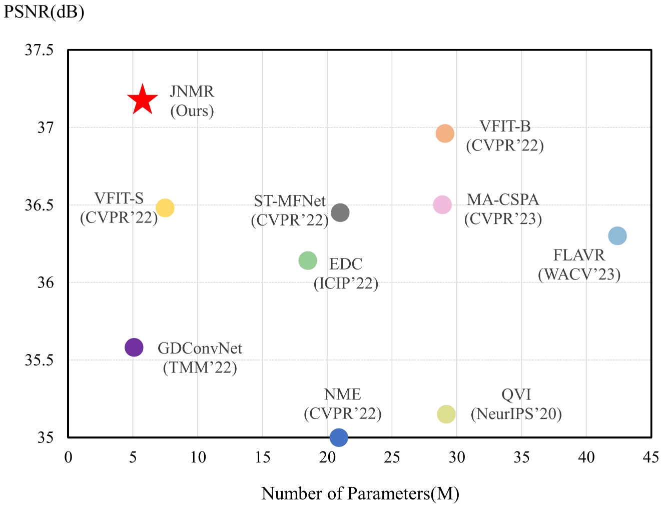


Fig. 1. Quantitative comparison of performance and model parameters with the Vimeo-90K Septuplet dataset. JNMR outperforms state-of-the-art methods with a competitive model size.

In general, many methods formulate complicated VFI motion estimation as seeking the warping correlation from historical consecutive frames. Optical flow-based solutions [8], [9] and kernel-based methods [10], [11] are commonly applied to realize the pipeline. Optical flow-based methods initially entail utilizing convolutional networks [12], [13] to estimate pixel-level motions. Whereas, the interpolation performance is limited by the accuracy of optical flow estimation. Therefore, some works [14], [15], [16], [17] propose to utilize exceptional features and supervision operations for improving the flow estimation accuracy. However, these methods suffer from high computational costs. Additionally, it is inevitable that the estimation precision of long-term variations and complex occlusions will be decreased by only exploring spatiotemporal dynamics in low-dimensional space. Kernel-based methods [11], [18], [19], [20] entail utilizing a deformable convolution operation for motion estimation, which can enable the adaption of the spatial dynamic scenes and the expansion of the receptive field of motion. However, the interpolation performance is still limited by the finite relations between inter-frame.

To explore temporal correlations, some approaches [21], [22], [23], [24], [25] entail extending the motion estimation methods to multiple reference frames interpolation. These methods primarily address the spatiotemporal correlations of complex motions by focusing on feature synthesis. Certain approaches [23], [24], [25] gradually divide temporal motions

Manuscript received 15 April 2023; revised 9 August 2023; accepted 7 September 2023. Date of publication 19 September 2023; date of current version 22 September 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0118001; and in part by the National Natural Science Foundation of China under Grant 62372036, Grant 61972028, Grant 62332017, and Grant 62120106009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lisimachos Paul Kondi. *(Meiqin Liu and Chenming Xu contributed equally to this work.) (Corresponding author: Chao Yao.)*

Meiqin Liu, Chenming Xu, Chunyu Lin, and Yao Zhao are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: mqliu@bjtu.edu.cn; chenming_xu@bjtu.edu.cn; cylin@bjtu.edu.cn; yzhao@bjtu.edu.cn).

Chao Yao is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: yaochao@ustb.edu.cn).

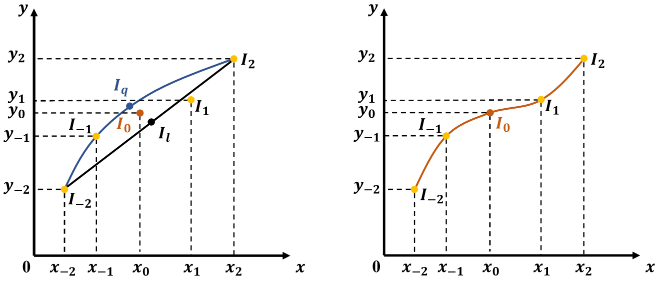
Digital Object Identifier 10.1109/TIP.2023.3315122

into smaller groups to enhance the accuracy of motion estimation. Other methods [21], [22] utilize 3D convolution to synthesize frames not only in the space dimension but also in the time dimension. However, the above methods mostly involve an assumption that there are uniform motions between consecutive frames according to a linear distribution (as with the black model in Fig.2(a)) in kinematics. This assumption fails to consider motion correlations in the temporal dimension of multi-variable regression. To address this problem, some works [26], [27], [28], [29] explore a quadratic interpolation model for multi-variable regression, allowing prediction with variable velocity and non-linear attributes. Furthermore, Dutta et al. [30] use the coefficients to adaptively select a linear or quadratic model for non-linear motion formulation. Shen et al. [31] introduce ConvLSTM to combine consecutive linear features as a quadratic regression. As illustrated in the blue model of Fig.2(a), these methods entail the prediction of intermediate motions with a curvilinear trajectory instead of the linear geometric center estimation.

Nevertheless, the non-linear movements in real-world scenes are always more complicated than previous assumptions due to variable acceleration, irregular movement trajectories, camera movement and shaking, etc. The linear and quadratic models cannot precisely describe non-linear situations. Therefore, in contrast to other methods, we utilize a multi-stage non-linear model to optimize the motion compensation. As shown in Fig.2(b), VFI is reformulated as the joint decoupled quadratic trajectories to refine the simulation of irregular movements.

Specifically, we propose a joint non-linear motion regression (JNMR) strategy to optimize the interpolation modeling, where both spatial deformations and temporal variations are considered. The movements can be regarded as a temporal concatenation of multi-stage quadratic models to address complicated motion reconstruction. ConvLSTM is implemented to initialize existing motions as a distribution in the temporal dimension. To preserve the moderate semantics of variations, we explore a hierarchical network structure before regression to retain the high-level variations in feature learning. Meanwhile, the skip-connection operation between the convolutional layers is utilized to compensate for the subtle variations in the final features. A coarse-to-fine synthesis enhancement module is designed for different spatial resolution feature synthesis with joint regression to further remove artifacts and unreliable pixels. Experimental results demonstrate the effectiveness of JNMR on video frame interpolation, outperforming state-of-the-art methods, as shown in Fig.1. In summary, our contributions are as follows:

- We propose a joint multi-variable non-linear motion regression strategy for motion compensation of VFI. The refined interpolation modeling can reconstruct the actual kinematic characteristics among large and complicated motions.
- We leverage ConvLSTM to realize the joint multi-stage quadratic model for JNMR, which enhances the temporal consistency in inter-frame prediction.
- We design the regression-driven feature learning module and coarse-to-fine enhancement module, separately optimizing the feature extraction with competitive parameters and pleasing visual qualities on different resolutions.



(a) Linear and Quadratic Model

(b) Multi-stage Non-linear Model

Fig. 2. Visualization of linear/quadratic estimation models and the proposed joint multi-stage non-linear regression model. (a) I_l denotes the symmetric midpoint referred to as a linear function, and I_q represents the middle motion of the quadratic trajectory. It can be observed that both the linear (black) and the quadratic (blue) models cannot precisely estimate some relatively complicated intermediate motions in the sequence $\{I_{-2}, I_{-1}, I_0, I_1, I_2\}$. (b) The interpolation is reformulated as decoupling the trajectory into individual regression models. The successive estimation with individual models can be regressed as a reintegration to predict I_0.

II. Related Works

Intermediate frames are generated by motion estimation and interpolation modeling. Motion estimation methods mainly incorporate optical flow-based methods and kernel-based methods. The non-linear interpolation strategy is proposed to match the curve trajectory. We briefly review related works in the following section.

A. Optical Flow-Based Methods

Optical flow-based methods aim to exploit the pixel-wise corresponding relationships of bidirectional reference frames as motion vectors for VFI. Xue et al. [13] first introduce bidirectional optical flows for motion estimation. Park et al. [32] recursively warp the optical flow by tracking the position with the greatest correlation. Cheng et al. [33] utilize a feature pyramid to achieve multi-scale optical flow estimation. Huang et al. [15] design a privileged distillation scheme optimized by the ground truth for precisely training intermediate flow models. Kong et al. [34] adopt a gradually refined intermediate feature to efficiently compensate for contextual details. Lu et al. [35] leverage Transformer to extend the receptive field of optical flow for long-term dependency optimization. To further improve the accuracy of flow estimation, some methods [14], [36], [37] have entailed inferring the additional information during the warping of original frames, such as with asymmetric fields and softmax splatting. Sim et al. [38] propose a recursive multi-scale structure for extreme VFI on 4K videos. In addition, many solutions have utilized multi-frame as input to enhance the logical continuity of optical flow. Specifically, Shen et al. [31] introduce ConvLSTM to estimate optical flow and combine temporal and spatial data. Park et al. [17] promote exceptional motion detection to improve the robustness of interpolation. These works have achieved state-of-the-art performance in realistic intermediate frame generation. However, the methods involved suffer from inaccurate pixel shifts, ghosting or blurry artifacts that are inevitably generated where the temporal dynamics dependencies are significant for the input frames.

B. Kernel-Based Methods

Instead of relying on optical flow, kernel-wise convolutional operations have been introduced to expand the receptive field of motion estimation with a lower computation burden. Niklaus et al. [11] are the first to provide useful insights on how to match the visual scenes and complicated motion for new frame interpolation through a dynamic network. To adapt to inter-frame motion properties, Lee et al. [19] design the AdaCoF model to complement the offset vector coefficient for kernel estimation. Cheng and Chen [39] propose an enhanced deformable separable convolution to estimate relatively complete kernel motions with a non-local neighborhood. To further extend the receptive field of kernel estimation, Shi et al. [24] and Tian et al. [40] explore a suitable distribution of reference points to effectively train the generalized deformable convolution for capturing the shape of the object. With regard to the improvement on the concrete network structure of the feature extraction, Ding et al. [20] deploy the feature pyramid and network compression in motion learning to strengthen the robustness with parameter reduction. Wijma et al. [41] implement spatial pyramids to increase the receptive field of the model to adaptively adjust the kernel size. Nevertheless, the limitation on temporal dependency still exists in kernel-level motion estimation. Thus, Choi et al. [42] attempt to utilize channel attention to distribute individual feature maps for motion extraction with spatiotemporal information. Kalluri et al. [21] and Danier et al. [22], [23] adopt multiscale 3D convolution to solve the spatiotemporal dependence with multiple frames. To integrate the advantages of the optical flow-based method in detailed motions, Bao et al. [43] propose the MEMC-Net to estimate motions by optical flow and deformable convolution. Hu et al. [44] propose a recurrent motion-enhanced interpolation network based on MEMC-Net to further capture spatiotemporal perspectives. Shi et al. [45] utilize the stacked optical flow estimator, trainable feature extraction and residual convolutional network to increase the quality of the interpolation view. Xiao et al. [46] propose a joint framework of flow and kernel estimation to achieve the space-time super-resolution of satellite videos. However, these methods use the compositions of convolutional layers to enable the learning of the spatiotemporal motions in complicated scenes without considering the consecutive variation recurrence. More importantly, both flow-based and kernel-based methods explicitly or implicitly assume linear motion among input frames, which leads to insufficient exploration of higher-order information in motion estimation.

C. Interpolation Modeling Methods

To further model long-term dynamics consistency from historical observations, some researchers have made innovations in terms of interpolation modeling based on existing motions. Typically, Zhou et al. [47] develop a texture consistency loss to ensure that interpolated content maintains structures similar to corresponding counterparts. Argaw et al. [48] conduct motions in the same direction as references for current interpolation when there is a large gap between input frames. In addition, some methods [49], [50], [51] propose the progressive non-linear motion estimation strategy, which aims to approximate complex motions. BasicVSR [49] and BasicVSR++ [50] use temporal alignment for long-term motion formulation in video super-resolution with all frames known, where the accumulation of errors may affect the final performance. Zhou et al. [51] propose an iterative alignment strategy that performs gradual refinement for shared sub-alignments. These methods focus on identifying how a finer motion for alignment can be learned with all frames known and on matching motion estimation in temporal dimensionality. To obtain reliable motions from a kinematic perspective, non-linear models [26], [27], [28], [29], [30], [52] have been utilized with the aim of approximating complex motion in the real world and overcoming the temporal limitation of camera sensors. Xu et al. [26] first define a quadratic model against existing linear models. To achieve precise motions, Liu et al. [27] adopt a rectified quadratic flow prediction formulation with a least squares function. Zhang et al. [28] further derive a general curvilinear motion trajectory formula on reference frames without temporal priors. Dutta et al. [30] utilize the predicted non-linear flow as coefficients to automatically select linear or quadratic models through space-time convolution networks. Tulyakov et al. [52] realize continuous non-linear motion estimation by combining both images and intermediate events. However, due to the incomplete consideration of complicated models in real scenes, the performance of VFI is still limited. Hence, we reformulate VFI as a joint regression strategy and further expand upon the existing models to adapt them to kinematic situations.

III. Methodology

A. Formulation

Multi-reference VFI entails the assumption that an input dynamic video sequence $\{I_n \mid n \in \{-\frac{N}{2}, \ldots, -1, 1, \ldots \frac{N}{2}\}\}$ consists of N frames along the time dimension, where each frame is recorded in a spatial region of $\mathbb{R}^{C\times H\times W}$. The goal is to predict the most likely frame I_0 for the ground truth intermediate frame I_{gt}.

Following the kernel-based motion estimation methods [19], the variations of the input sequence I_n can be modeled as the motions. Specifically, the input synthesized features can be decoupled into motion vectors using deformable convolution kernels, where the input frames I_n are refined to the warped frames $\hat{I}_n$, as:

$$\begin{aligned}&\hat{I}_n(i,j)\\&\quad=\sum_{p=0}^{K-1}\sum_{q=0}^{K-1}\mathbf{W}_{p,q}(i,j)I_n(i+dp+\boldsymbol{\alpha}_{p,q},j+dq+\boldsymbol{\beta}_{p,q})\end{aligned} \tag{1}$$

where $\hat{I}_n(i,j)$ denotes the target pixel at (i,j) in I_n, $d \in \{0,1,2,\ldots\}$ indicates the dilation of the motion vectors bias

(p, q), and K is the kernel size. $\boldsymbol{W}_{p,q}$ and $(\boldsymbol{\alpha}_{p,q}, \boldsymbol{\beta}_{p,q})$ denote the kernel weight and motion vectors between $I_n(i, j)$ and $\hat{I}_n(i, j)$.

Next, considering the occlusion between bidirectional resource frames, the interpolated frame $\hat{I}_0$ is formulated as:

$$\hat{I}_0 = O \cdot \sum_{t=-N/2}^{-1} \hat{I}_n + (1 - O) \cdot \sum_{t=1}^{N/2} \hat{I}_n \tag{2}$$

where $[\cdot]$ is the inner product between two matrices. $O \in [0, 1]$ indicates the occlusion generated by the deconvolutional layers with a sigmoid function. However, the spatial information modeled by the above formulation is limited to linear correlations, which neglects the essential information about temporal variable dynamics in real complicated kinematics.

To estimate the complicated motion of the intermediate frame, we first reformulate the VFI as a motion-time model:

$$\tilde{\boldsymbol{y}}_n = \boldsymbol{\omega}_n \boldsymbol{x}_n \tag{3}$$

where $\boldsymbol{x}_n$ denotes the temporal variables. $\tilde{\boldsymbol{y}}_n$ is the corresponding predicted motion. $\boldsymbol{\omega}_n$ is a regression coefficient related to temporal variables $\boldsymbol{M}_n$. Typically, $\boldsymbol{M}_n$ with the common component $(\boldsymbol{W}_n, \boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ is defined as an image wise motion from $I_n \rightarrow I_0$. To further concretize the correlations in Eq.3, the motions can be commonly understood as the distance between two frames I_0 and I_n.

Therefore, according to the general kinematic regularity, motions can be quantified by the instantaneous velocity $\boldsymbol{v}_0$ of I_n and acceleration $\boldsymbol{a}_t$ as a distance-time function:

$$\boldsymbol{M}_{n+1} - \boldsymbol{M}_n = \int_0^t (\boldsymbol{v}_0 + \int_0^k \boldsymbol{a}_t dt) dk \tag{4}$$

where $(\boldsymbol{M}_{n+1} - \boldsymbol{M}_n)$ is a variation vector and represents the distance between two frames. t is the differentiable variable between two adjacent motions. k is the intermediate variable used for integration. Since $\boldsymbol{v}_0$ and $\boldsymbol{a}_t$ are difficult to calculate, two adjacent motions cannot determine the kinematic regression model. Thus, an additional motion $\boldsymbol{M}_{n-1}$ is introduced, and the new condition can be formulated as:

$$\boldsymbol{M}_n - \boldsymbol{M}_{n-1} = \int_0^t (\boldsymbol{v}_0 - \int_0^k \boldsymbol{a}_t dt) dk \tag{5}$$

Since the instantaneous velocity $\boldsymbol{v}_0$ is constant, the kinematic model can be solved as the difference between Eq.4 and Eq.5:

$$(\boldsymbol{M}_{n+1} - \boldsymbol{M}_n) - (\boldsymbol{M}_n - \boldsymbol{M}_{n-1}) = 2 \int_0^t \boldsymbol{a}_t t dt \tag{6}$$

It is evident that the quadratic model can be determined by at least three consecutive motions $\{\boldsymbol{M}_{n-1}, \boldsymbol{M}_n, \boldsymbol{M}_{n+1}\}$. Obviously, this assumption is based on the prior that the movement follows a uniform acceleration. In real-world scenarios, objects do not always move regularly at a consistent velocity. It is inaccurate to describe the overall motion only by this dynamical model.

Consequently, we design a general kinematic model to capture complicated motions by combining consecutive independent quadratic models, which can be regarded as uniformly variable motion. The parameters of each individual quadratic model can be defined by the existing motions, and the empirical regression equation of the kinematic model can be determined as:

$$\hat{\boldsymbol{y}}_n = \hat{\boldsymbol{\omega}}_n \boldsymbol{x}_n \tag{7}$$

with

$$\begin{aligned} \hat{\boldsymbol{y}}_n &= \hat{\boldsymbol{M}}_n, \\ \hat{\boldsymbol{\omega}}_n &= [\boldsymbol{M}_n, \hat{\boldsymbol{v}}_n, \hat{\boldsymbol{a}}_n], \\ \boldsymbol{x}_n &= [1, t, t^2]^T \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{M}}_n$ denotes the individual regressed motions from $\boldsymbol{M}_n$. $\hat{\boldsymbol{v}}_n$ and $\hat{\boldsymbol{a}}_n$ are the initial velocity and acceleration of I_n.

As depicted in Fig.2(b), the individual quadratic model is inadequate to accurately capture the complicated non-linear motions with irregularity. To alleviate this limitation, we decompose the overall motion into a multi-stage quadratic model. Specifically, three consecutive frames are utilized to form a complete model after regressing the sub-distribution. Following this pipeline, the whole regression can be defined as the temporal-aware combination of multiple dependent quadratic models as follows:

$$H_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_n) = \boldsymbol{\theta} \otimes \boldsymbol{Y} \tag{9}$$

with

$$\begin{aligned} \boldsymbol{\theta} &= [\theta_{-\frac{N}{2}+1}, \cdots, \theta_{\frac{N}{2}-1}], \\ \boldsymbol{Y} &= [\hat{\boldsymbol{M}}_{-\frac{N}{2}+1}, \cdots, \hat{\boldsymbol{M}}_{\frac{N}{2}-1}]^T \end{aligned} \tag{10}$$

where $H_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_n)$ represents the temporal concatenation of quadratic models, which can be utilized to predict motions. $\boldsymbol{\theta}$ denotes the polynomial coefficient collection between different quadratic models. $\otimes$ represents the temporal concatenation instead of linear combination in general matrix multiplication. $\boldsymbol{Y}$ contains the individual regressed motions. It is noted that the above formulation releases the constraint of velocity and acceleration and achieves the connection of the multi-stage quadratic models. Furthermore, the empirical model could be trained with forward and backward regression:

$$H_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_n) = \hat{\boldsymbol{\theta}} \otimes \hat{\boldsymbol{Y}} \tag{11}$$

with

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= [\hat{\theta}_f, \hat{\theta}_b], \\ \hat{\boldsymbol{Y}} &= [\hat{\boldsymbol{M}}_f, \hat{\boldsymbol{M}}_b]^T \end{aligned} \tag{12}$$

where $\hat{\boldsymbol{\theta}}$ which contains $[\hat{\theta}_f, \hat{\theta}_b]$ denotes the collection of bidirectional regressed polynomial coefficients. $\hat{\boldsymbol{Y}}$ denotes the second-order regressed motion sequence. $\hat{\boldsymbol{M}}_f$ and $\hat{\boldsymbol{M}}_b$ denote the forward and backward regressed motions, respectively, in a minimal unilateral neighborhood of the intermediate moment. The intermediate instantaneous motion obtained in Eq.12 cannot be directly transformed into the visual location of the final frame. Consequently, the visual movement offset $\Delta \hat{I}_0$ needs to be inferred from the most adjacent frames I_{-1} and I_1, as follows:

$$\Delta \hat{I}_0 = \hat{\theta}_f \cdot \varphi(I_{-1}, \hat{\boldsymbol{M}}_f) + \hat{\theta}_b \cdot \varphi(I_1, \hat{\boldsymbol{M}}_b) \tag{13}$$

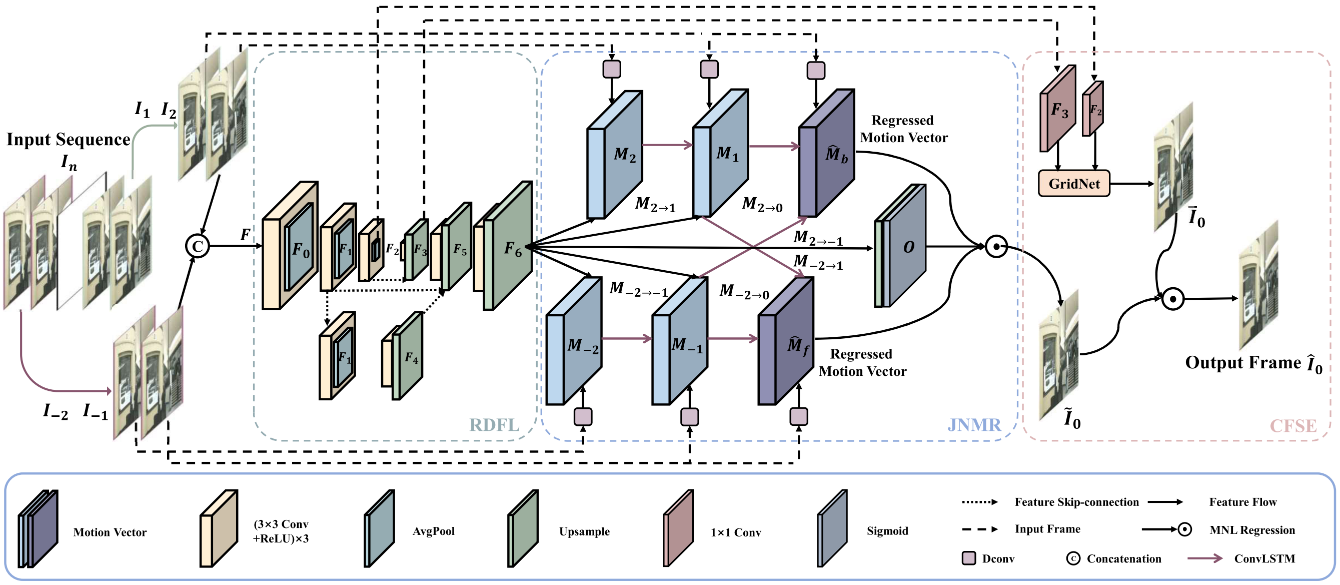


Fig. 3. Illustration of our pipeline for interpolation. In particular, we implement a concatenation of four reference frames $\{I_{-2}, I_{-1}, I_1, I_2\}$ as the input into the network for regression-driven feature learning. The fusion feature $\boldsymbol{F}_6$ is processed by deformable convolution to obtain the corresponding motion vectors. Then, a joint non-linear regression strategy is proposed to achieve robust interpolation modeling, considering the real kinematic model. The details are shown in Fig.4. Finally, coarse-to-fine synthesis enhancement is designed to overcome artifacts caused by complicated motion estimation.

where $\hat{\theta}_f$ and $\hat{\theta}_b$ denote the predicted polynomial coefficients for forward and backward motions. φ indicates the warping operation on the reference frame illustrated in Eq.1. The current predicted frame $\tilde{I}_0$ can be incorporated to make up the limitation of long-term dynamics dependency, as:

$$\tilde{I}_0 = \hat{I}_0 + \Delta\hat{I}_0 \tag{14}$$

where $\Delta\hat{I}_0$ indicates the visual movement offset, and $\hat{I}_0$ denotes the basic synthesis frame in Eq.2.

B. Regression-Driven Feature Learning

Feature learning module can obtain an aggregated feature from the concatenation of input sequences before regression. It is essential for each spatiotemporal dynamic feature to be learned separately in the temporal dimension from the input frame concatenation. Previously developed algorithms, such as CNN-based [19], [20], [21] and Transformer-based methods [25], [53], only formulate feature learning as a general feature filter without consideration given to the internal structure of fusion features. These stacked filter layers bring structural redundancy and result in the loss of local-global expressions. Therefore, we propose a regression-driven feature learning (RDFL) network, which is shown in Fig.3. The RDFL network not only compensates with multiple hierarchical spatial structures to exploit high-level motions with appropriate motion features for regression but also simultaneously reduces the number of parameters.

In detail, the concatenation feature $\boldsymbol{F}$ of the multiple candidate frames $\{I_{-2}, I_{-1}, I_1, I_2\}$ is input into three convolutional layers with the down-sampling operation to obtain the fundamental feature $\boldsymbol{F}_0$ for computation reduction. Then, three layers of the hierarchical spatial structure are utilized with $\boldsymbol{F}_0$ to extract regression-aware detailed features $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ with different resolutions, which can be expressed as:

$$\begin{aligned}\boldsymbol{F}_0 &= \downarrow(\phi_{conv}(\boldsymbol{F})),\\ \boldsymbol{F}_1 &= \downarrow(\phi_{conv}(\boldsymbol{F}_0)),\\ \boldsymbol{F}_2 &= \downarrow(\phi_{conv}(\boldsymbol{F}_1))\end{aligned} \tag{15}$$

where $\downarrow()$ denotes the down-sampling operation with average pooling and $\phi_{conv}()$ represents three consecutive convolutional layers.

Then, to further enhance the spatial feature expression, a multi-scale fusion strategy is implemented in the hierarchical spatial structures by skip-connection operation. The deconvolutions with the up-sampling operation are adopted to extract features $\boldsymbol{F}_3$, $\boldsymbol{F}_4$ and $\boldsymbol{F}_5$, as follows:

$$\begin{aligned}\boldsymbol{F}_3 &= \uparrow(\phi_{deconv}(\boldsymbol{F}_2)) + \boldsymbol{F}_1,\\ \boldsymbol{F}_4 &= \uparrow(\phi_{deconv}(\boldsymbol{F}_1)),\\ \boldsymbol{F}_5 &= \uparrow(\phi_{deconv}(\boldsymbol{F}_3)) + \boldsymbol{F}_0 + \boldsymbol{F}_4\end{aligned} \tag{16}$$

where $\uparrow()$ denotes the up-sampling operation by bilinear interpolation and $\phi_{deconv}()$ represents three consecutive deconvolutions. After that, $\boldsymbol{F}_6$ is sampled by $\boldsymbol{F}_5$, as follows:

$$\boldsymbol{F}_6 = \uparrow(\phi_{deconv}(\boldsymbol{F}_5)) \tag{17}$$

where the final feature $\boldsymbol{F}_6$ can be decoupled as the original motions $\{\boldsymbol{M}_{-2}, \boldsymbol{M}_{-1}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$ as in Eq.1.

C. Joint Non-Linear Motion Regression

After obtaining the appropriate motion features $\boldsymbol{F}_6$ from the RDFL network, as shown in Fig.3, joint non-linear motion regression (JNMR) is proposed to implement the aforementioned rationale in Sec.III-A with four reference motions

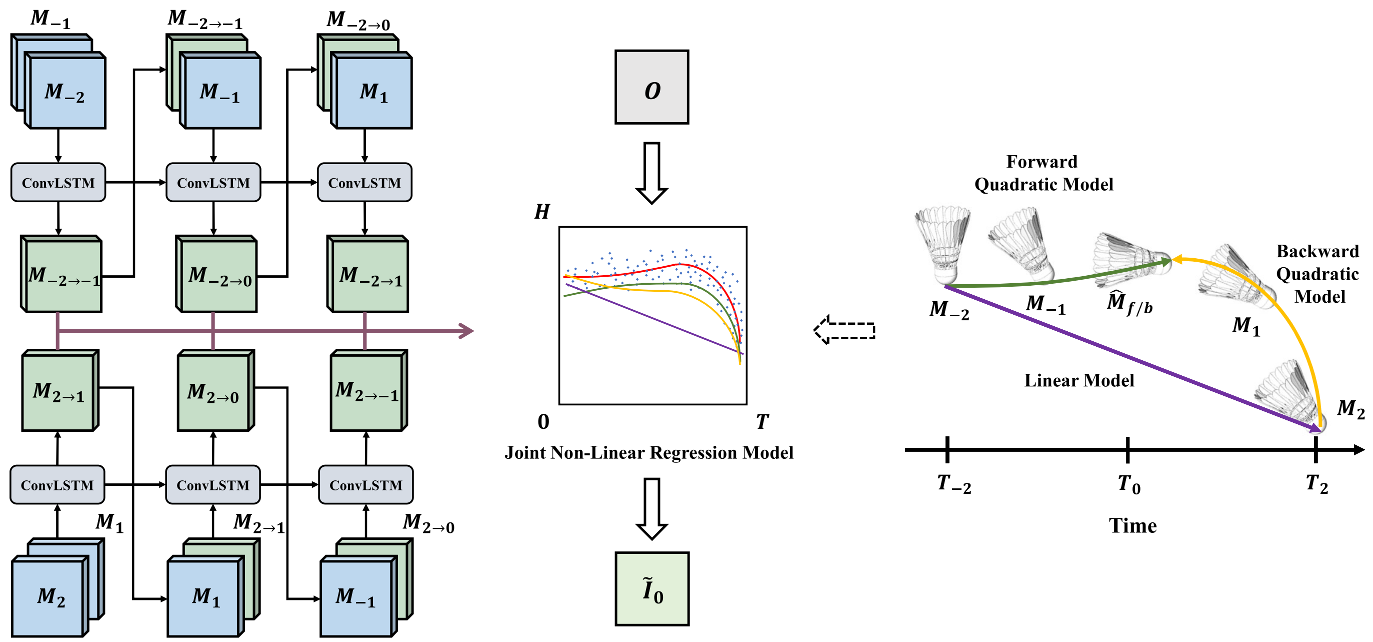


Fig. 4. Illustration of JNMR structure details. Here, we incorporate temporal-aware motion sequences $[\boldsymbol{M}_{-2}, \boldsymbol{M}_{-1}, \boldsymbol{M}_1]$ and $[\boldsymbol{M}_2, \boldsymbol{M}_1, \boldsymbol{M}_{-1}]$ to model different quadratic regressions for the middle motions $[\hat{\boldsymbol{M}}_f, \hat{\boldsymbol{M}}_b]$. Thus, the entire model is regressed by optimizing each individual sub-non-linear model.

$\{\boldsymbol{M}_{-2}, \boldsymbol{M}_{-1}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$. Following Eq.7, the individual bidirectional regressed motions can be expressed as:

$$\lim_{f\to0^-} \hat{\boldsymbol{M}}_f = \frac{(\boldsymbol{M}_1 - \boldsymbol{M}_{-1}) - 2(\boldsymbol{M}_{-1} - \boldsymbol{M}_{-2})}{3},$$
$$\lim_{b\to0^+} \hat{\boldsymbol{M}}_b = \frac{(\boldsymbol{M}_{-1} - \boldsymbol{M}_1) - 2(\boldsymbol{M}_1 - \boldsymbol{M}_2)}{3} \tag{18}$$

where $\hat{\boldsymbol{M}}_f$ and $\hat{\boldsymbol{M}}_b$ denote the forward and backward regressed motions in a minimal unilateral neighborhood of the intermediate moment, respectively. The variation vectors between two frames are temporally combined by the consecutive ConvLSTM elaborated in Fig.4. For example, $(\boldsymbol{M}_{-2}, \boldsymbol{M}_{-1})$ and $(\boldsymbol{M}_{-1}, \boldsymbol{M}_1)$ are input into ConvLSTM to explore relative variation for solving $\hat{\boldsymbol{M}}_f$ in Eq.18 of the forward temporal dimension. Later, $\hat{\boldsymbol{M}}_f$ and $\hat{\boldsymbol{M}}_b$ are applied in Eq.12 for independent regression to adaptively form a complete regression model. The regressed expression can be attained as:

$$\hat{\boldsymbol{\theta}}' = [\hat{\theta} \quad (1-\hat{\theta})],$$
$$\hat{\boldsymbol{Y}}' = [\hat{\boldsymbol{M}}_f \quad \hat{\boldsymbol{M}}_b]^T \tag{19}$$

where the regressed coefficient $\hat{\theta}$ is initialized by occlusion $O \in [0, 1]$. The visual movement offset $\Delta\hat{I}_0$ can be inferred by Eq.13, as:

$$\Delta\hat{I}_0 = \hat{\theta} \cdot \varphi(I_{-1}, \hat{\boldsymbol{M}}_f) + (1-\hat{\theta}) \cdot \varphi(I_1, \hat{\boldsymbol{M}}_b) \tag{20}$$

As illustrated in Eq.14, the basic synthesis frame $\hat{I}_0$ can be specifically expressed as:

$$\hat{I}_0 = O \cdot (\hat{I}_{-2} + \hat{I}_{-1}) + (1-O) \cdot (\hat{I}_1 + \hat{I}_2) \tag{21}$$

Consequently, the current predicted frame $\tilde{I}_0$ can be attained by the combination of $\hat{I}_0$ and $\Delta\hat{I}_0$, as:

$$\tilde{I}_0 = \hat{I}_0 + \Delta\hat{I}_0 \tag{22}$$

D. Coarse-to-Fine Synthesis Enhancement

After obtaining the motions $\{\boldsymbol{M}_{-2}, \boldsymbol{M}_{-1}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$, the interpolation frame $\tilde{I}_0$ is synthesized by JNMR, as illustrated in Sec.III-C. In general, occlusion affects the reconstructed details and decreases the visual quality of interpolation frames. Therefore, a coarse-to-fine synthesis enhancement (CFSE) module is proposed to further preserve the details of interpolated frames.

As described in Sec.III-C and Fig.3, the coarse features $\boldsymbol{F}_2$ and $\boldsymbol{F}_3$ are decoupled into motions, and $\tilde{I}_0$ is reconstructed at different scales. Following GridNet [54], the multi-scale reconstructed frames are concatenated to generate a coarse-to-fine interpolated frame $\overline{I}_0$. The final interpolation frame I_0 is then synthesized with $\tilde{I}_0$ and $\overline{I}_0$, as:

$$I_0 = \lambda \cdot \tilde{I}_0 + (1-\lambda) \cdot \overline{I}_0 \tag{23}$$

where λ denotes the weight coefficient initialized by occlusion.

E. Objective Function

For the end-to-end training, we utilize the objective, perceptual and deformation loss to measure the difference between the synthesized frame I_0 and its ground truth I_{gt}. Specifically, the ℓ_1 loss with the Charbonnier penalty [55] is introduced as:

$$\mathcal{L}_{Charbon} = \rho(I_0 - I_{gt}) \tag{24}$$

where $\rho(x) = (||x||_2^2 + \epsilon^2)^{\frac{1}{2}}$ and $\epsilon = 0.001$.

The perceptual loss $\mathcal{L}_{vgg}$ is expressed as:

$$\mathcal{L}_{vgg} = ||\Phi(I_0) - \Phi(I_{gt})||_2 \tag{25}$$

where $\Phi()$ is a feature extraction from conv4_3 of the pre-trained VGG16 [56].

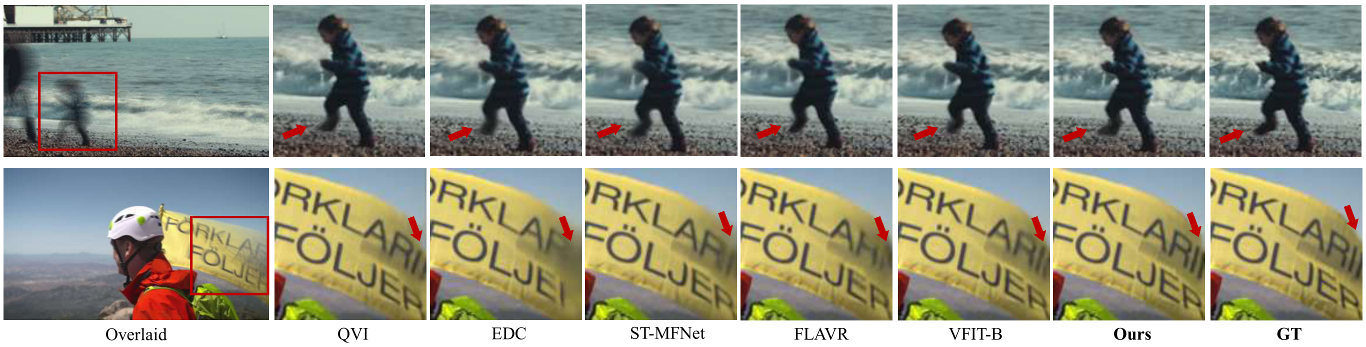


Fig. 5. Qualitative comparisons against the state-of-the-art multi-reference interpolation algorithms with the Vimeo-90K Septuplet dataset. Our method generates higher-quality frames with an exquisite visual appearance and few distortions.

The deformation loss $\mathcal{L}_d$ is implemented to measure the abnormal deformation distance of each directional motion vector at the kernel-level as follows:

$$\mathcal{L}_d = \sum_{i,j} ||(\boldsymbol{\alpha}, \boldsymbol{\beta})_{i,j+1} - (\boldsymbol{\alpha}, \boldsymbol{\beta})_{i,j}||_1 + ||(\boldsymbol{\alpha}, \boldsymbol{\beta})_{i+1,j} - (\boldsymbol{\alpha}, \boldsymbol{\beta})_{i,j}||_1 \tag{26}$$

where $(\alpha, \beta)_{i,j}$ represents the regressed kernel-level motion vectors in $\hat{\boldsymbol{M}}_n$ and bidirectional motions in $\hat{\boldsymbol{M}}_f$ and $\hat{\boldsymbol{M}}_b$.

Three loss functions are combined to optimize the whole network parameters for end-to-end training as follows:

$$\mathcal{L} = \mathcal{L}_{Charbon} + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_d\mathcal{L}_d \tag{27}$$

where λ_{vgg} and λ_d denote the weight coefficients of perceptual loss and deformation loss, respectively.

IV. Experiments

A. Implementation Details

1) Configuration: All experiments are implemented on two NVIDIA GeForce RTX 3090 GPUs with Intel(R) Xeon(R) Gold 6226R CPUs. We conduct 120 training epochs with the Vimeo-90K Septuplet [13] dataset and 30 fine-tuning epochs on BVI-DVC [57] dataset with a mini-batch size of 8. AdaMax [58] is the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set as $1e^{-3}$ to $1.5e^{-5}$ by half decay every 20 epochs. λ_{vgg} and λ_d are set to 0.005 and 0.01, respectively.

2) Training Datasets: We train our model using the Vimeo-90K Septuplet [13] dataset, which includes 64,612 and 7,824 seven-frame sequences with a resolution of 256 × 448. To further improve the performance for large motions, we use the BVI-DVC [57] dataset, which includes 17,600 quintuplets with a resolution of 256 × 256, to fine-tune for better evaluation. The middle frame of each septuplet and quintuplet is the interpolation target, and its adjacent four consecutive frames are used as the input frames in Fig.3. We also apply random horizontal, vertical flipping, and temporal order reversal to further enhance the training dataset.

3) Evaluation Datasets: The experimental model is evaluated not only on the validation set of the Vimeo-90K Septuplet but also on other commonly used benchmark datasets, such as DAVIS [59] and GoPro [60] as previously demonstrated in QVI [26]. Using the same sampling principle, we report PSNR and SSIM [61] with 2,849 quintuples generated from DAVIS and 3,012 quintuples with a resized resolution of 480 × 854 from GoPro.

B. Evaluation Against State-of-the-Art Methods

To verify the effectiveness of JNMR, we make a fair comparison with state-of-the-art methods using four reference frames, such as QVI [26], PRF$_4$ [31], EDC [22], ST-MFNet [23], GDConvNet [24], FLAVR [21], MA-CSPA [47], NME [30] and VFIT [25]. With regard to QVI, EDC, ST-MFNet, GDConvNet, FLAVR and VFIT, the pre-trained models are directly used with the same experimental setups for evaluation. In addition, we compare AdaCoF [19] and CDFI [20] using two reference frames with the Vimeo-90K Septuplet. Regarding other studies presenting methods without publicly available code,[1] we conduct a comparison with the results kindly provided by the study authors.

1) Quantitative Evaluation: As shown in TABLE I, our proposed JNMR has great superiority with the Vimeo-90K Septuplet, DAVIS, and GoPro benchmarks with exceptional performance in terms of model parameters and running time. Notably, JNMR achieves an interpolation performance beyond 37 dB with the Vimeo-90K Septuplet dataset. JNMR further improves the interpolation performance without complicated feature learning and synthesis by exploiting the temporal-aware acceleration information through joint regression. Moreover, after the fine-tuning process with the BVI-DVC dataset followed by the ST-MFNet, the experimental results in TABLE I show that JNMR has improved on different test datasets. With only 5.7M parameters, JNMR promotes significant improvements over state-of-the-art methods, such as 0.23 dB with the Vimeo-90K Septuplet and 1.16 dB with GoPro. In summary, our JNMR method achieves new state-of-the-art performance with strong generalization for different benchmarks with competitive parameters.

2) Qualitative Evaluation: We provide a qualitative comparison between our method and the latest state-of-the-art methods with the Vimeo-90K Septuplet and DAVIS datasets in Fig.5 and Fig.6. As shown in Fig.5, JNMR generates visually pleasing interpolation results with clear structures on

[1] The results are marked by † in TABLE I.

TABLE I

VIDEO FRAME INTERPOLATION: QUANTITATIVE COMPARISONS OF VIMEO-SEPTUPLET, DAVIS, AND GOPRO WITH STATE-OF-THE-ART METHODS. THE NUMBERS IN **BOLD** AND UNDERLINE INDICATE THE BEST AND THE SECOND BEST PSNR(DB)/SSIM RESULTS WITH PARAMETERS(#P.). THE RUN-TIME (RT) OF EACH MODEL IS REPORTED WITH AN INPUT SIZE OF 480P FRAMES. THE RESULTS MARKED WITH † ARE CLONED FROM THE AUTHORS

Methods	Venue	Training Dataset	Vimeo-Septuplet	DAVIS	GoPro	#P.(M)	RT(s)
AdaCoF [19]	CVPR'20	Vimeo-Septuplet	34.63 / 0.970	27.28 / 0.872	29.41 / 0.912	21.8	**0.04**
CDFI [20]	CVPR'21	Vimeo-Septuplet	35.38 / 0.972	27.17 / 0.863	29.59 / 0.915	**5.0**	0.17
QVI [26]	NeurIPS'19	GoPro	35.15 / 0.971	27.73 / 0.894	30.55 / 0.933	29.2	0.16
†PRF$_4$-Large [31]	TIP'20	Adobe240	-	-	31.06 / 0.907	11.4	-
EDC [22]	ICIP'22	Vimeo-Septuplet & BVI	36.14 / 0.974	27.63 / 0.878	30.93 / 0.936	18.5	0.35
ST-MFNet [23]	CVPR'22	Vimeo-Septuplet & BVI	36.45 / 0.976	28.29 / **0.895**	30.88 / 0.936	21.0	0.38
GDConvNet [24]	TMM'22	Vimeo-Septuplet	35.58 / 0.958	27.03 / 0.821	30.82 / 0.913	5.1	0.88
†NME [30]	CVPR'22	Vimeo-Septuplet	34.99 / 0.954	27.53 / 0.828	29.08 / 0.883	20.9	-
VFIT-S [25]	CVPR'22	Vimeo-Septuplet	36.48 / 0.976	27.92 / 0.885	30.55 / 0.939	7.5	0.20
VFIT-B [25]	CVPR'22	Vimeo-Septuplet	36.96 / 0.978	28.09 / 0.888	30.60 / 0.940	29.1	0.28
FLAVR [21]	WACV'23	Vimeo-Septuplet	36.30 / 0.975	27.44 / 0.874	31.31 / 0.940	42.4	0.30
†MA-CSPA [47]	CVPR'23	Vimeo-Septuplet	36.50 / 0.962	-	-	28.9	-
JNMR(Ours)	-	Vimeo-Septuplet	37.13 / 0.978	28.25 / 0.887	32.46 / **0.951**	5.7	0.38
JNMR(Ours)	-	Vimeo-Septuplet & BVI	**37.19** / **0.979**	**28.32** / 0.889	**32.47** / **0.951**	5.7	0.38

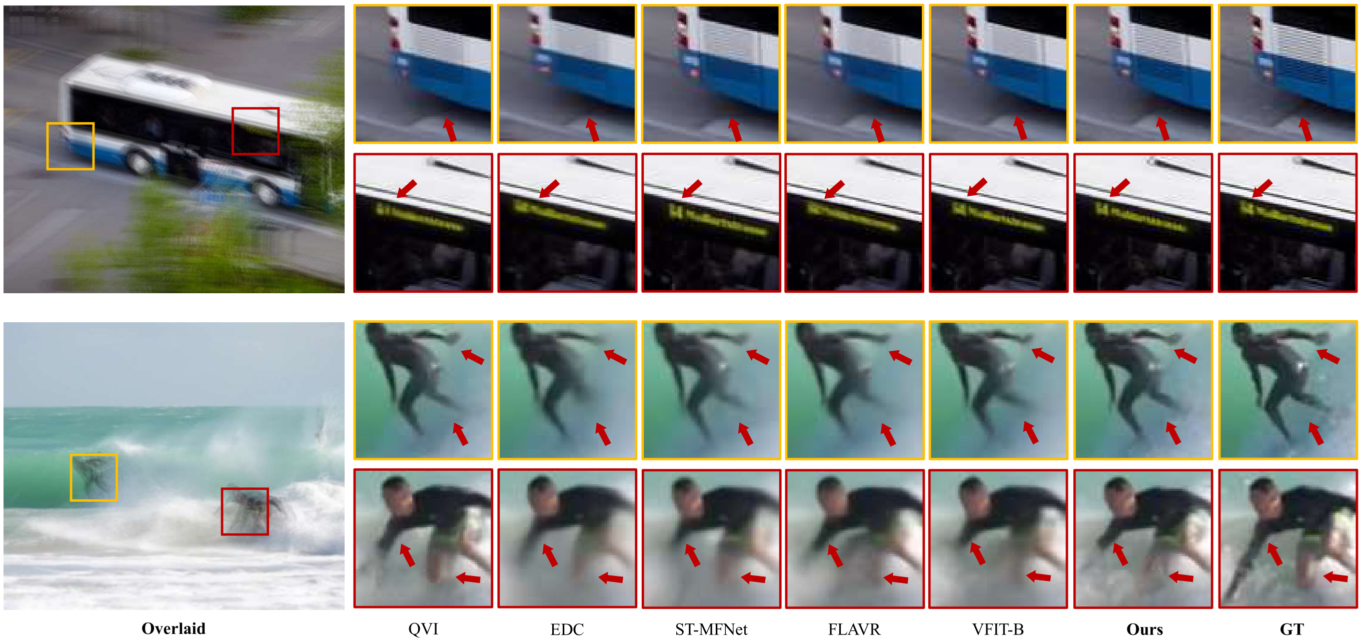


Fig. 6. Qualitative evaluation of the high-speed motion of the DAVIS dataset. JNMR not only outperforms other models in terms of texture reconstruction and artifact reduction but also generates a reasonable middle frame with structural consistency when dealing with large motions.

Vimeo-90K Septuplet. The two samples both contain complicated motion with low resolution. In particular, the second sample involves structural details such as characters. It is noted that other methods fail to restore the original appearance of the child's foot and the rightmost text. In contrast, our model achieves a sharp boundary and realistic texture without excessive smoothing due to the appropriate feature learning and motion regression. Furthermore, we demonstrate the temporal-aware performance and the reconstruction effect on high-speed movement in Fig.6. Although EDC [22], ST-MFNet [23], FLAVR [21] and VFIT-B [25] generate visually correct structures, there is some serious blurring due to the direct fusion of overlaid input. QVI [26] relies on the quadratic modeling of motions, but their interpolation frames usually contain notable artifacts because of inaccurate synthesis. In contrast, our method successfully handles complicated acceleration movement modeling and produces plausible structures with abundant details. In particular, the relative position between the cement marking line and the bus in the first sample indicates the temporal consistency in consecutive frames. JNMR can realize a close relation to the ground truth (GT) with clear edges and demonstrate effective

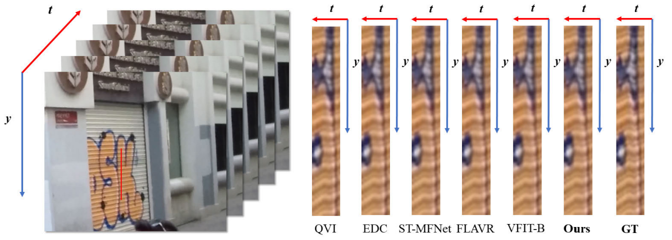


Fig. 7. The temporal profile on the GroPro dataset. The temporal profile is generated by stacking a single-pixel line (red light) among video sequences. The resource sequence contains fifteen consecutive frames, where the nine middle frames are generated by different models.

TABLE II

Quantitative Comparisons on the Slow, Medium and Fast Sub-Collection of the Vimeo-90K Septuplet With State-of-the-Art Methods. The Numbers in **Bold** and Underline Indicate the Best and the Second Best PSNR(dB)/SSIM Results, Respectively

Methods	Vimeo-90K Septuplet		
	Slow	Medium	Fast
AdaCoF [19]	37.54 / 0.982	34.55 / 0.970	31.11 / 0.946
CDFI [20]	38.27 / 0.984	35.23 / 0.973	32.19 / 0.951
QVI [26]	37.99 / 0.982	35.04 / 0.971	32.08 / 0.949
EDC [22]	38.71 / 0.987	36.08 / 0.975	32.99 / 0.952
ST-MFNet [23]	38.70 / 0.987	36.35 / 0.976	33.88 / 0.957
GDConvNet [24]	38.01 / 0.978	35.58 / 0.961	32.36 / 0.923
FLAVR [21]	38.72 / $\underline{0.988}$	36.25 / $\underline{0.977}$	33.34 / 0.953
VFIT-S [25]	$\underline{38.73}$ / 0.987	$\underline{36.40}$ / $\underline{0.977}$	$\underline{33.88}$ / $\underline{0.958}$
JNMR(Ours)	**39.47 / 0.989**	**37.12 / 0.978**	**34.45 / 0.959**

TABLE III

Ablation Results of Individual Sub-Component

Models	#P.(M)	PSNR(dB)	SSIM
Baseline	5.3	36.82	0.975
Baseline w/ RDFL	**4.0**	36.82	0.975
Baseline w/ JNMR	6.9	37.08(+0.26)	0.978(+0.003)
Baseline w/ CFSE	5.4	36.98(+0.16)	0.976(+0.001)
JNMR(Full)	5.7	**37.19**(+0.37)	**0.979**(+0.004)

TABLE IV

Ablation Results of Different Regression Models

Models	Illustrations	#P.(M)	PSNR(dB)	SSIM
Model 1	Linear	**4.1**	36.95	0.976
Model 2	Quadratic	5.7	37.04	0.977
Model 3	Linear combination of quadratic	5.7	37.04	0.977
Model 4	Unidirectional	5.7	37.10	0.978
Model 5	Second-order unidirectional	5.7	37.05	0.977
JNMR	Joint bidirectional	5.7	**37.19**	**0.979**

capabilities in handling temporal-aware high-speed motions. In addition, it is noted that our method can achieve clear text reconstruction and edge preservation of high-speed moving objects.

3) Temporal Consistency: We apply the temporal consistency [62] to measure the movement of pixel lines and further evaluate the structural continuity in interpolated videos. Methods that exhibit similar pixel tendencies as the ground truth are deemed to preserve the original temporal variations effectively. We present the visual results of the comparison methods on the GoPro dataset in Fig.7. The ground truth contains crooked horizontal lines that indicate camera platform movement. Our method reflects these dynamic scenes with a fine high-frequency context. Other methods produce overly smooth results and fail to capture the turning point of pixel direction under long-term dynamics. From the above evaluation, it can be seen that JNMR is an effective method for restoring subtle temporal variations using motion regression.

4) Evaluation on Large and Complicated Motions: To verify the superiority of JNMR on large and complicated motions, we evaluate our method with Vimeo-90K Septuplet sub-collections according to the methods of previous work [63]. The test sequences are stratified into slow, medium and fast sub-collections by estimated motion velocities. Compared to the other state-of-the-art methods with similar complexity, as shown in TABLE II, our method performs better performance on different motion velocities. Specifically, it can be seen that motion regression among multiple reference frames can effectively improve VFI performance compared with our baseline methods. As shown in Fig.8, the qualitative example on the fast sub-collection demonstrates that JNMR does not produce more motion artifacts when dealing with complicated and large motions, especially in the case of irregular camera movement in the first example.

C. Ablation Study

In this section, we present the results of the comprehensive ablation studies to evaluate the contribution of the JNMR strategy and other auxiliary sub-components with the Vimeo-90K Septuplet dataset. The quantitative evaluation results of individual components with the baseline model are shown in TABLE III.

1) Joint Non-Linear Motion Regression: The proposed JNMR is designed to model variable acceleration curve motion as a multi-stage quadratic movement. To verify the effectiveness of this strategy, we conduct different multi-variable regressions as shown in TABLE IV. We first compare the performance of the linear (Model 1) and quadratic (Model 2) models. Then, the validity of the temporal-aware combination using ConvLSTM illustrated in Sec.III-C is verified. We use a linear combination of quadratic models (Model 3) to derive $\hat{M}_f$ and $\hat{M}_b$ without ConvLSTM. Model 3 can also be seen as the natural cubic spline. Both Model 3 and natural cubic spline obtain a quadratic curve locally and sequentially process the video sequence. We also compare two different regression strategies, unidirectional regression and second-order unidirectional regression. In particular, unidirectional regression (Model 4) only uses forward/backward motions in Eq.19, such as $\hat{\boldsymbol{Y}}' = [\hat{\boldsymbol{M}}_f]^T$ or $\hat{\boldsymbol{Y}}' = [\hat{\boldsymbol{M}}_b]^T$. The second-order unidirectional regression (Model 5) makes a key ablation in terms of regression direction through serial repeating of the above pipeline with both directions.

The evaluation results demonstrate the robustness of our ratiocination as shown in TABLE IV and Fig.9. Our regression strategy successfully restores the correct occlusion and

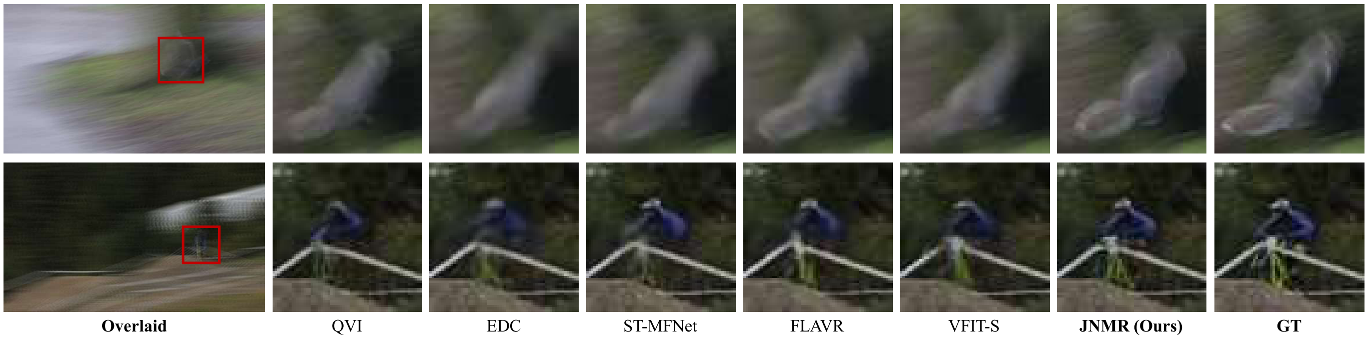


Fig. 8. Qualitative comparisons against the state-of-the-art multi-reference interpolation algorithms with the Vimeo-90K Septuplet dataset. The middle frames generated by JNMR have clear high-frequency structures and few artifacts compared with other models when dealing with large and complicated motions.

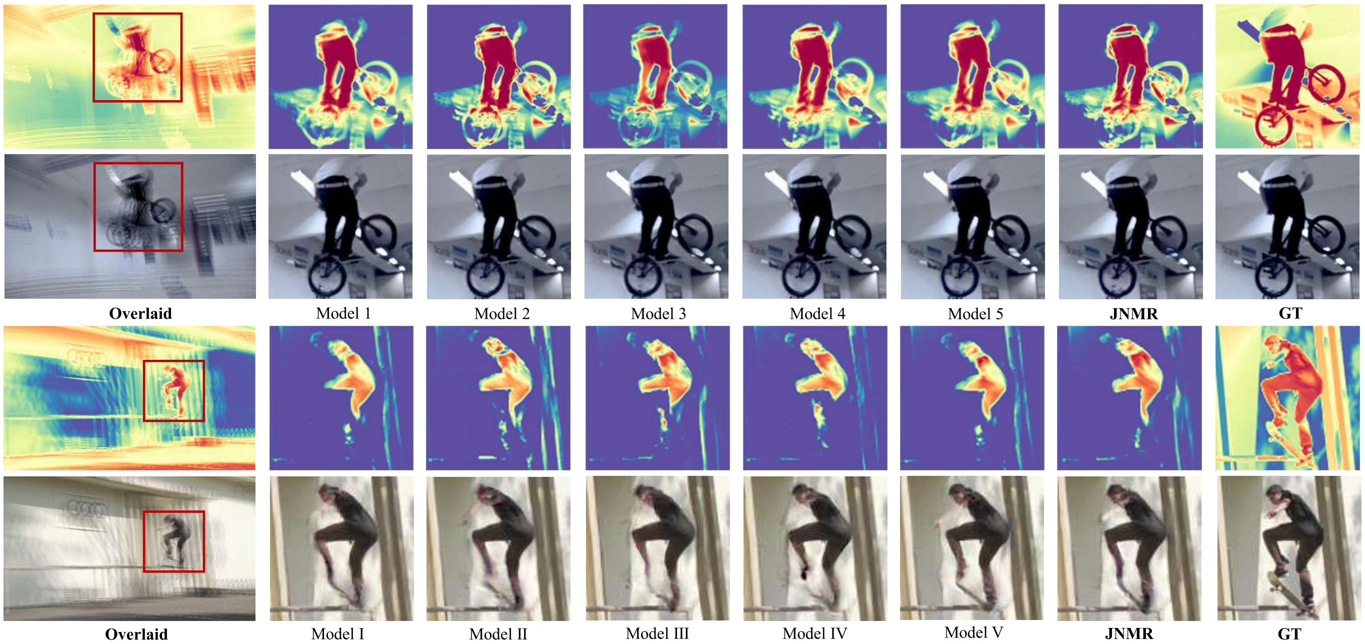


Fig. 9. Qualitative comparisons with the different regression models and reconstruction strategies with the Vimeo-90K Septuplet dataset. It is noted that JNMR generates reasonable motion with an exquisite visual appearance.

TABLE V

Ablation Results on Different Numbers of Hierarchical Structures With Multi-Stage Compensation

Models	Structures Number	Multi-stage Compensation	#P.(M)	PSNR(dB)	SSIM
Model I	5	✘	7.0	37.11	0.978
Model II	3	✘	**5.6**	37.05	0.978
JNMR	3	✔	5.7	**37.19**	**0.979**

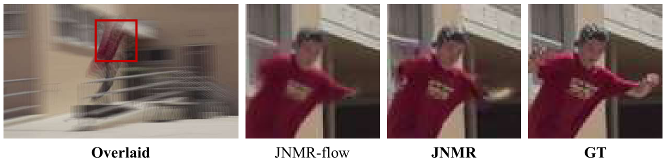


Fig. 10. Qualitative comparisons with different motion estimation methods with the Vimeo-90K Septuplet dataset. Flow-based motions are able to reduce artifacts but extract less complicated motion information.

consistent patterns in handling complicated motions of the sample. Moreover, the visualization of $\tilde{I}_0$ verifies that JNMR can interpolate motion with clear edge details and few artifacts. Notably, our reliable kinematic model is robust for different dynamic visual scenes.

2) Feature Learning and Frame Synthesis: As described in Sec.III, an appropriate network architecture is explored to retain semantic information favorable for motion regression. The coarse-to-fine synthesis enhancement module is implemented to preserve finer details, with the different resolution motions integrated into the final frame. TABLE V shows the performance of the network with different numbers of hierarchical spatial structures illustrated in Eq.16. Model I, which has five hierarchical spatial structures, does not outperform the JNMR. It indicates that more complex feature extraction structures are not very helpful for frame reconstruction and motion regression but also bring about an increase in parameters. In addition, the multi-stage compensation strategy with a few parameters improves the performance compared to Model II which has only up-sampling operations. To verify the efficiency of the coarse-to-fine synthesis enhancement module,

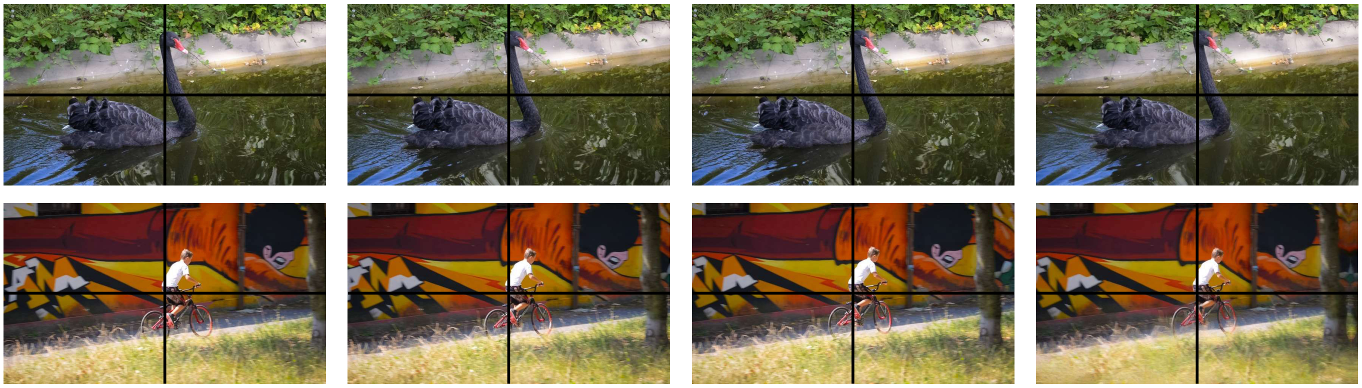


Fig. 11. The qualitative evaluation of extrapolation on the DAVIS dataset. The moving targets in the frame sequence are indicated by the calibration of the black cross.

TABLE VI

THE ABLATION RESULTS ON DIFFERENT SOURCE FEATURES FOR COARSE-TO-FINE SYNTHESIS ENHANCEMENT MODULE

Models	Source Features	GridNet	#P.(M)	PSNR(dB)	SSIM
Model III	-	✘	**5.6**	37.03	0.977
Model IV	$\boldsymbol{F}_1, \boldsymbol{F}_2$	✔	5.7	37.06	0.977
Model V	$\boldsymbol{F}_2, \boldsymbol{F}_3$	✘	**5.6**	37.06	0.977
JNMR	$\boldsymbol{F}_2, \boldsymbol{F}_3$	✔	5.7	**37.19**	**0.979**

TABLE VII

ABLATION RESULTS ON DIFFERENT MOTION ESTIMATION METHODS

Models	Motion Estimation Methods	#P.(M)	PSNR(dB)	SSIM
JNMR-flow	Optical Flow	9.5	36.96	0.976
JNMR	Deformable Convolution	**5.7**	**37.19**	**0.979**

TABLE VIII

ABLATION RESULTS ON THE DIFFERENT NUMBERS OF REFERENCE FRAMES WITH THE GOPRO DATASET

Models	Reference Number	#P.(M)	PSNR(dB)	SSIM
JNMR	4	5.7	32.47	0.951
JNMR-6f	6	6.0	32.52	0.952

we conduct an ablation study on source features of different resolutions. As described in TABLE VI, the multi-stage extraction influences the visual details laterally. Besides, GridNet has proven effective in multi-scale feature fusion through Model V. In addition to the advantages in quantitative evaluation, our method can also restore comprehensive structures, in contrast to the other ablation methods shown in Fig.9.

3) Flow-Based Motion Estimation: To further verify the universality of our method, the pre-trained LiteFlowNet [64] is utilized to generate optical flow motions instead of deformable convolution denoted as JNMR-flow. JNMR-flow can also achieve good VFI performance, but uses more model parameters, as illustrated in TABLE VII. However, the whole network is mainly designed toward the feature-level, which leads to the lower interpolation performance of JNMR-flow compared to the kernel-based JNMR. In addition, Fig.10 indicates that the pixel-level optical flow of JNMR-flow is still inferior to the feature-level offset map of JNMR in catching large motions.

TABLE IX

VIDEO FRAME EXTRAPOLATION: QUANTITATIVE COMPARISONS WITH THE VIMEO-TRIPLET AND ADOBE240 WITH STATE-OF-THE-ART METHODS. THE NUMBERS IN **BOLD** AND UNDERLINE INDICATE THE BEST AND THE SECOND BEST PSNR(DB) AND SSIM RESULTS WITH PARAMETERS(#P.). THE RESULTS OF OTHER METHODS ARE CLONED FROM [47] AND [53]

Methods	#P.	Vimeo-Triplet		Adobe240	
	(M)	PSNR	SSIM	PSNR	SSIM
Convtransformer [53]	-	30.52	0.941	30.42	0.946
DVF [55]	**3.8**	27.08	0.907	28.74	0.925
MCNet [65]	-	28.62	0.873	28.21	0.880
Sepconv [11]	21.7	30.42	0.917	-	-
FLAVR [21]	42.1	31.14	0.927	-	-
MA-CSPA [47]	22.4	**32.05**	0.940	-	-
JNMR (Ours)	4.5	31.55	**0.947**	**31.33**	**0.952**

4) The Number of Reference Frames: To validate the influence of the reference frame number, we input 6 reference frames to interpolate the intermediate frame, denoted as JNMR-6f. Particularly, the first and last frames of 4-frame input and 6-frame input from GoPro dataset are constant in different models. Due to the increase in the number of reference frames, the intermediate motion estimation is more accurate. Therefore, as illustrated in TABLE VIII, the results of JNMR-6f are slightly higher than those of the 4 reference frames (JNMR). The two models remain consistent overall and have similar performance, demonstrating the generalization of JNMR on long sequences.

D. Extension for Extrapolation

To further explore the extension of the motion regression, we follow the work DVF [55] to conduct JNMR in the video frame extrapolation to generate future frames with several reference frames. In detail, we predict the next frame utilizing two consecutive frames and make a quantitative evaluation with the Vimeo-Triplet and Adobe240 [66] datasets. As shown in TABLE IX, JNMR achieves improvements on most evaluation indicators with fewer parameters. The qualitative evaluation results on the DAVIS dataset are shown in Fig.11. The predicted objects in JNMR have the same relative position compared to ground truth, which demonstrates the effectiveness of JNMR in maintaining the temporal consistency of moving objects.

V. Conclusion

In this paper, we conduct an analysis on the importance of long-term dynamics in the task of video frame interpolation. To overcome the challenge of large and complicated motion synthesis, a joint non-linear motion regression (JNMR) strategy is designed to introduce multi-variate non-linear regression for interpolation. Our method formulates the kinematic trajectory as joint multi-stage quadratic models and achieves accurate and consistent motion prediction. Furthermore, regression-driven feature learning and coarse-to-fine synthesis enhancement modules are explored to maintain global structures and complement details for regression. The experimental results demonstrate the superior performance and robustness of JNMR compared to other state-of-the-art methods.

References

[1] 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.

[2] Z. Hu, G. Lu, and D. Xu, "FVC: A new framework towards deep video compression in feature space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1502–1511.

[3] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10998–11007.

[4] R. Pourreza and T. Cohen, "Extending neural P-frame codecs for B-frame coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6660–6669.

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

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

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

[8] 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.

[9] T.-C. Wang et al., "Video-to-video synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1152–1164.

[10] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman, "Visual dynamics: Stochastic future generation via layered cross convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2236–2250, Sep. 2019.

[11] 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.

[12] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[13] 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.

[14] 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.

[15] 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.*, 2022, pp. 624–642.

[16] 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. 3698–3707.

[17] M. Park, H. G. Kim, S. Lee, and Y. M. Ro, "Robust video frame interpolation with exceptional motion map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 754–764, Feb. 2021.

[18] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10607–10614.

[19] 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. 5315–5324.

[20] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "CDFI: Compression-driven network design for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7997–8007.

[21] 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.

[22] D. Danier, F. Zhang, and D. Bull, "Enhancing deformable convolution based video frame interpolation with coarse-to-fine 3D CNN," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 1396–1400.

[23] D. Danier, F. Zhang, and D. Bull, "ST-MFNet: A spatio-temporal multi-flow network for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3521–3531.

[24] Z. Shi, X. Liu, K. Shi, L. Dai, and J. Chen, "Video frame interpolation via generalized deformable convolution," *IEEE Trans. Multimedia*, vol. 24, pp. 426–439, 2022.

[25] Z. Shi, X. Xu, X. Liu, J. Chen, and M.-H. Yang, "Video frame interpolation transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 17482–17491.

[26] 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. 1–10.

[27] Y. Liu, L. Xie, L. Siyao, W. Sun, Y. Qiao, and C. Dong, "Enhanced quadratic video interpolation," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2020, pp. 41–56.

[28] Y. Zhang, C. Wang, and D. Tao, "Video frame interpolation without temporal priors," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 13308–13318.

[29] J. Xing, W. Hu, Y. Zhang, and T.-T. Wong, "Flow-aware synthesis: A generic motion model for video frame interpolation," *Comput. Vis. Media*, vol. 7, no. 3, pp. 393–405, Sep. 2021.

[30] S. Dutta, A. Subramaniam, and A. Mittal, "Non-linear motion estimation for video frame interpolation using space-time convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1725–1730.

[31] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Video frame interpolation and enhancement via pyramid recurrent framework," *IEEE Trans. Image Process.*, vol. 30, pp. 277–292, 2021.

[32] J. Park, K. Ko, C. Lee, and C. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 109–125.

[33] X. Cheng and Z. Chen, "A multi-scale position feature transform network for video frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3968–3981, Nov. 2020.

[34] L. Kong et al., "IFRNet: Intermediate feature refine network for efficient frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1969–1978.

[35] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3522–3532.

[36] 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.

[37] 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.

[38] 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.

[39] X. Cheng and Z. Chen, "Multiple video frame interpolation via enhanced deformable separable convolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7029–7045, Oct. 2022.

[40] H. Tian, P. Gao, and X. Peng, "Video frame interpolation based on deformable kernel region," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1349–1355.

[41] R. Wijma, S. You, and Y. Li, "Multi-level adaptive separable convolution for large-motion video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1127–1135.

[42] 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.*, 2020, pp. 10663–10671.

[43] 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.

[44] M. Hu et al., "Capturing small, fast-moving objects: Frame interpolation via recurrent motion enhancement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3390–3406, Jun. 2022.

[45] J. Shi, X. Jiang, and C. Guillemot, "Deep residual architecture using pixel and feature cues for view synthesis and temporal interpolation," *IEEE Trans. Comput. Imag.*, vol. 8, pp. 246–259, 2022.

[46] Y. Xiao et al., "Space-time super-resolution for satellite video: A joint framework based on multi-scale spatial–temporal transformer," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 108, Apr. 2022, Art. no. 102731.

[47] K. Zhou, W. Li, X. Han, and J. Lu, "Exploring motion ambiguity and alignment for high-quality video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22169–22179.

[48] D. M. Argaw and I. S. Kweon, "Long-term video frame interpolation via feature propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3533–3542.

[49] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "BasicVSR: The search for essential components in video super-resolution and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4945–4954.

[50] K. C. K. Chan, S. Zhou, X. Xu, and C. C. Loy, "BasicVSR++: Improving video super-resolution with enhanced propagation and alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5962–5971.

[51] K. Zhou, W. Li, L. Lu, X. Han, and J. Lu, "Revisiting temporal alignment for video restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6043–6052.

[52] S. Tulyakov, A. Bochicchio, D. Gehrig, S. Georgoulis, Y. Li, and D. Scaramuzza, "Time lens++: Event-based frame interpolation with parametric nonlinear flow and multi-scale fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17734–17743.

[53] Z. Liu et al., "ConvTransformer: A convolutional transformer network for video frame synthesis," 2020, *arXiv:2011.10185*.

[54] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[55] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4463–4471.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[57] D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: A training database for deep video compression," *IEEE Trans. Multimedia*, vol. 24, pp. 3847–3858, 2022.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–13.

[59] 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.

[60] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265.

[61] 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.

[62] H. Song, W. Xu, D. Liu, B. Liu, Q. Liu, and D. N. Metaxas, "Multi-stage feature fusion network for video super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 2923–2934, 2021.

[63] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3892–3901.

[64] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8981–8989.

[65] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–22.

[66] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1279–1288.