Published: 23 May 2025
Online AM: 18 March 2025
Accepted: 28 February 2025
Revised: 22 January 2025
Received: 19 June 2024

RESEARCH-ARTICLE

Video Frame Interpolation via Fast Bidirectional 3D Correlation Volume

DENGYONG ZHANG, Changsha University of Science and Technology, Changsha, Hunan, China

RUNQI LOU, Changsha University of Science and Technology, Changsha, Hunan, China

JIAXIN CHEN, Changsha University of Science and Technology, Changsha, Hunan, China

XIANGLING DING, Hunan University of Science and Technology, Xiangtan, Hunan, China

XIN LIAO, Hunan University, Changsha, Hunan, China

GAOBO YANG, Hunan University, Changsha, Hunan, China

Video Frame Interpolation via Fast Bidirectional 3D Correlation Volume

DENGYONG ZHANG, RUNQI LOU, and JIAXIN CHEN, School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, China
XIANGLING DING, Hunan University of Science and Technology, Xiangtan, China
XIN LIAO and GAOBO YANG, Hunan University, Changsha, China

Recently, there has been a growing demand for flow-based video frame interpolation methods, which introduce correlation volumes to supervise the correlation of bidirectional optical flows. However, they often overlook the symmetry of the bidirectional motion field by consuming substantial computational cost, which is reflected in the fact that these methods often require a long runtime. To address these issues, in this article, we propose a bidirectional 3D correlation volume which is suitable for video frame interpolation. By decomposing the 4D correlation volume into two 3D correlation volumes in the horizontal and vertical directions, we significantly enhance the model's inference speed with a minor sacrifice compared to our baseline. Additionally, when handling 2K video frames, our method achieves several-fold improvement in inference speed compared to other methods which implied correlation volume. The code is available at https://github.com/famt0531.

CCS Concepts: • **Computing methodologies** → **Reconstruction**;

Additional Key Words and Phrases: Video Frame Interpolation, Convolutional Neural Network, Flow-based, Correlation Volume, Attention

ACM Reference format:
Dengyong Zhang, Runqi Lou, Jiaxin Chen, Xiangling Ding, Xin Liao, and Gaobo Yang. 2025. Video Frame Interpolation via Fast Bidirectional 3D Correlation Volume. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 5, Article 148 (May 2025), 22 pages.
https://doi.org/10.1145/3724123

This work is supported by the Hunan Provincial Postgraduate Scientific Research Innovation Project under Grant CSLGCX23089; the China National Natural Science Foundation under Grant 62172059, 62272160, 62402062, and U22A2030; the China National Key Research and Development Program under Grant 2022YFB3103500 and 2024YFF0618800; the Hunan Provincial Key Research and Development Program under Grant 2024AQ2027; the Hunan Province Natural Science Foundation, China, under Grant 2025JJ60415 and 2025JJ50370; and the Changsha City Natural Science Foundation, China, under Grant kq2402031.
Authors' Contact Information: Dengyong Zhang, School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, China; e-mail: zhdy@csust.edu.cn; Runqi Lou, School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, China; e-mail: lourunqi@stu.csust.edu.cn; Jiaxin Chen, School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, China; e-mail: chenjiaxin@hnu.edu.cn; Xiangling Ding (corresponding author), Hunan University of Science and Technology, Xiangtan, China; e-mail: xianglingding@163.com; Xin Liao, Hunan University, Changsha, China; e-mail: xinliao@hnu.edu.cn; Gaobo Yang, Hunan University, Changsha, China; e-mail: yanggaobo@hnu.edu.cn.

ACM 1551-6865/2025/5-ART148
https://doi.org/10.1145/3724123

1 Introduction

Video frame interpolation (VFI) is a task in computer vision focused on creating intermediate frames in a video sequence by utilizing appearance and motion information from multiple frames, thereby increasing the video's frame rate [4]. VFI is widely used in video frame rate conversion, slow-motion generation [2, 17], and view synthesis [38, 55]. Despite recent advancements in deep learning, VFI still struggles with challenges such as large motions and occlusions in real video frames.

Recent research has established flow-based VFI methods [13, 16, 17, 20, 26, 28, 32, 47] as mainstream in the field. They achieved promising results, but neglected the correlation between bidirectional optical flows. There are some methods [24, 35, 36, 54] which address this issue by incorporating a 4D ($H \times W \times H \times W$) correlation volume during the optical flow estimation stage, enhancing bidirectional optical flow correlation supervision and achieving promising results. However, constructing the correlation volume often involves using a coarse-to-fine warping framework to create multiple partial cost volumes [41], or employing a pyramid structure to capture both large and small motions simultaneously [42]. This leads to the formation of massive 4D cost volumes, significantly increasing computational time required for reconstructing intermediate frames. This issue is particularly pronounced when dealing with high-resolution video frames.

To balance accuracy and inference time, we were inspired by the work of Xu et al. [48], proposing a bidirectional 3D correlation volume for VFI task. In VFI region, accurately capturing intermediate frame motion requires estimating the bidirectional motion field between two frames. Since bidirectional motion is often asymmetric, unidirectional optical flow from traditional optical flow methods is unsuitable for VFI. Thus, we constructed a bidirectional correlation volume rather than an unidirectional one and scaled multi-scale feature pairs and searched within the bidirectional 3D correlation volume(referring to the lookup layer in Figure 2), addressing feature mismatches caused by occlusions, inspired by AMT [24]. To determine the optimal search radius for matching, we conducted extensive ablation experiments and used quantitative metrics to identify a radius that effectively capturing both large and small motions, see Section 4.4.1.

Moreover, to refine the motion field further, we introduce **bidirectional gated recurrent units (Bi-GRU)** [23] implemented by depth-wise separable convolutions during the optical flow update stage to fuse motion information and intermediate features from forward and backward direction. In the synthesis stage, we propose a multi-scale context-aware synthesis module to estimate the final result.

Our method demonstrates a more significant speed advantage for large-scale input images, as shown in Figure 1. The inference time of models grows exponentially as the input image resolution increases, whereas our model consistently maintains rapid inference speed, especially when the resolution reaches 2K. Compared to our baseline, we achieved a $\times 2$ speed improvement, and compared to other methods utilizing correlation volumes, the improvement in speed is even more pronounced. FBTCV significantly reduces inference time at high resolutions by decomposing the bidirectional 4D correlation volume in two directions. Meanwhile, the Bi-GRU and multi-scale context-aware interpolation modules help the model better fit the pixel values of the intermediate frames. Compared to other efficiency-focused methods, such as RIFE [16] and FLDR [34], which exhibit comparable speeds, our approach also achieves higher accuracy. This remains true even in interpolation tasks with standard resolutions, demonstrating the robustness of FBTCV, see details in Section 4.3 .

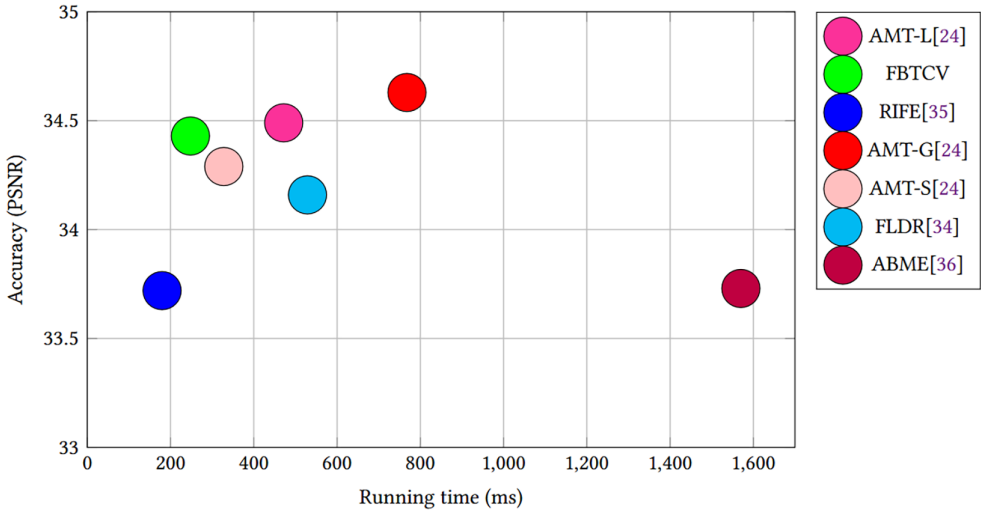


Fig. 1. Comparison of inference speeds and accuracy with our baseline with its other variants and exhibited efficiency at 2K resolutions.

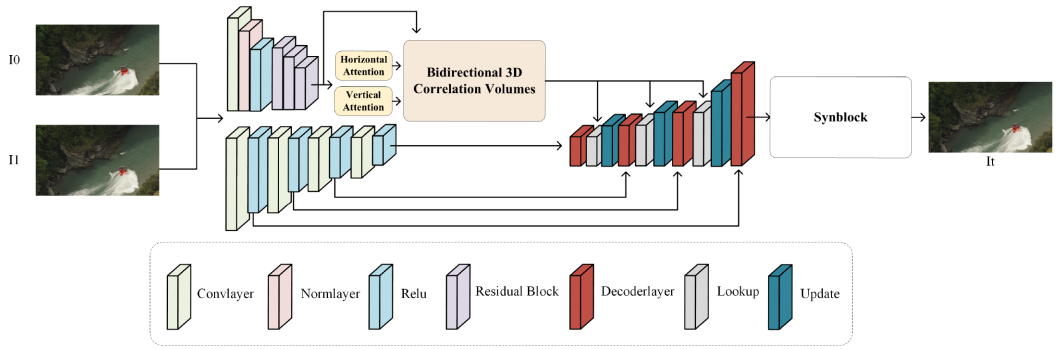


Fig. 2. Overview of the proposed method. It includes the following main components: an encoder combining CNN and Transformer, a pure CNN encoder, the corresponding decoder, bidirectional 3D cost volumes, and a synthesis module.

Our contributions are summarized as follows:

—We propose a bidirectional 3D correlation volume suitable for VFI task and construct a novel network for VFI called FBTCV, which models bidirectional correlation of feature pairs in a novel way, significantly promoting the computational speed while maintaining accuracy.
—FBTCV achieved highly competitive performance across various benchmark tests. In our VFI model, to better obtain bidirectional flow fields and intermediate frames, we designed an optical flow update module with Bi-GRU and a multi-scale context-aware synthesis block.

2 Related Work

2.1 VFI

VFI poses a challenging low-level task in computer vision. The advancement of deep learning has prompted researchers to explore numerous impressive methodologies, broadly categorized into

three main classes: kernel-based methods [8, 9, 11, 22, 33, 50, 52], flow-based methods [16, 20, 24, 32, 35], and hybrid methods [2, 3, 51].

Kernel-based methods create interpolated pixel values by using local convolution on input frames, predicting a spatially adaptive convolution kernel for each pixel based on two consecutive frames. Each pixel in the interpolated frame has its own kernel, which is then convolved with the input frames to generate the interpolated frame. Unlike traditional methods that separate motion estimation and resampling, convolution kernels combine these steps.

Flow-based methods use **convolutional neural networks (CNN)** [21] to estimate dense bidirectional optical flow [46], capturing pixel correspondences between consecutive images. This estimated flow guides a warping process that transforms and blends input images into interpolated frames while preserving motion coherence. Kong et al. [20] proposed IFRNet, an efficient video interpolation network with a single encoder-decoder structure. It extracts feature pyramids from two input frames, refines bidirectional intermediate optical flow fields, and restores the desired output at the input resolution. Huang et al. [16] introduced IFNet, which estimates intermediate flows efficiently from coarse to fine, enhancing speed. They also developed a privileged distillation scheme for training, significantly boosting performance. Jin et al. [18] created a compact model that estimates bidirectional motion simultaneously using a flexible pyramid recurrent framework, fine-tuning components within optical flow research for improved performance.

Hybrid methods achieve high-quality frame interpolation by combining various strategies, which usually combine the properties of optical-flow-based and kernel-based methods, and construct specific network structures and loss functions to accomplish the modeling of inter-frame motion information. Bao et al. [3] estimated the motion-compensated kernel and optical flow information used to interpolate frames via a CNN and introduced an adaptive warping layer that combines the motion-compensated kernel and the optical flow information to generate a specific pixel for each position. Yang and Oh [51] proposed a cascade network to maximize the benefits of the optical flow and motion estimation kernels, which consists of a network of three autoencoders to handle the initial frame interpolation and its refinement.

2.2 Correlation Volume

In computer vision, correlation volumes play a crucial role in assessing similarity between images by calculating the similarity scores for each pixel across two images. This process identifies the most similar target pixel for each reference pixel, essential for tasks like VFI where accurate motion estimation is paramount. Several methods in VFI have integrated correlation volumes into bidirectional optical flow estimation.

BMBC [35] extends the capabilities of PWC-Net [41] which replace the concatenation of sub-networks with multi-scale features by constructing bidirectional correlation volumes. Meanwhile, EBME [18] addresses bidirectional motion estimation using a unified correlation volume approach, leveraging CNN features from two warped frames to compensate for the estimated motion. Despite these advancements, methods based on PWC-Net often struggle with asymmetrical bidirectional motion, particularly evident in scenarios involving large motions.

DQBC [54] takes a novel approach by directly modeling correlations at a single high resolution in one step, thus avoiding iterative low-resolution modeling. This method aims to mitigate asymmetry in bidirectional motion effectively. On the other hand, AMT [24] introduces a method to calculate similarity between real frames by adapting the multi-scale pyramid structure of correlation volumes within the RAFT [42] framework. Both DQBC and AMT successfully address asymmetrical bidirectional motion fields, albeit at the cost of increased computational inference time due to their correlation volume construction complexities.

In summary, while methods like DQBC and AMT advance bidirectional motion estimation by handling asymmetry, there remains a tradeoff between accuracy and computational efficiency in correlation volume-based approaches for VFI tasks in computer vision. Our research aims to optimize these methods to achieve both accurate motion estimation and efficient computation especially for high resolutions in practical applications.

2.3 Attention in Computer Vision

The self-attention mechanism [44] was originally extensively applied in natural language processing. Its fundamental principle involves computing attention weights between features to measure their degree of association. These weights are calculated based on factors such as feature similarity, spatial position, or other inter-feature correlations. Specifically, the self-attention mechanism computes attention scores by performing dot products between each feature and all other features in the feature space, normalized using a softmax function to obtain attention weights. These weights are then multiplied with the input features to generate weighted feature representations, enabling more effective utilization of these features in subsequent processing steps.

In recent years, the self-attention mechanism has played a crucial role in computer vision tasks [5, 12, 27, 43, 45, 56] as well. For instance, in image classification tasks, the Transformer architecture segments images into patches and treats them as sequential data, effectively leveraging self-attention to learn global feature correlations. Vision Transformer exemplifies this application, demonstrating outstanding performance on large-scale datasets. Moreover, self-attention has made significant strides in tasks such as object detection and image segmentation, facilitating the transmission of attentional information across different feature layers to capture finer semantic details and spatial relationships.

Furthermore, the cross-attention mechanism [15] has been employed in computer vision, particularly in addressing inter-image relations. This mechanism dynamically adjusts attention levels across different images by establishing attentional connections between them, enhancing the model's understanding of semantic associations between images. For instance, in cross-modal tasks like image captioning or image matching, cross-attention helps capture detailed semantic associations between different images, thereby improving model performance in complex visual tasks and opening new avenues for multi-modal data fusion.

3 Our Approach

In this section, we first introduce the fundamental purpose of the model and its overall structure. Then, starting from the principles, we will provide a detailed explanation of the implementation details of each module of our model.

3.1 Problem Description

Given two real frames I^0 and I^1, the goal of the VFI task is to infer their intermediate frame I^t based on the input real frames, where $t \in [0, 1]$ represents the timestep and is typically set to 0.5 by default. Our model adopts a dual-encoder-single-decoder structure. The dual encoders consist of one pure CNN encoder and another encoder combining CNN and Transformer. The decoder comprises decoding layers, a lookup layer, and an update module. The dual-encoder structure has been applied in several computer vision fields. For the VFI task, L^2BC^2 [52] utilizes two pure CNN encoders, while the combination of CNN and Transformer benefits the modeling of both local and global image information [7, 37].

In FBTCV, the encoder combining CNN and Transformer is used to capture the correlation between the two input frames which contains residual blocks, horizontal and vertical attention, while the other CNN encoder is employed to extract the texture features of the two real frames.

The decoder's decoding layer decodes the texture feature information, the lookup layer searches for features within the constructed correlation volumes, and the update module generates residuals based on the outputs of the previous two layers to update motion and content information.

3.2 Bidirectional 3D Correlation Volume

3.2.1 Horizontal Attention. After getting the real frames I^0 and I^1, where $I^i(i = 0, 1) \in \mathbb{R}^{C\times H\times W}$, the feature extractor encodes them using several convolutional layers with a residual structure, and resulting in $Feat^0 \in \mathbb{R}^{C'\times \frac{H}{8}\times \frac{W}{8}}$ and $Feat^1 \in \mathbb{R}^{C'\times \frac{H}{8}\times \frac{W}{8}}$. C and C' represent the channel dimension of the input video frame and the number of channels mapped to a high-dimensional space after feature extraction, while H and W denote the height and width of the video frame, respectively. This cross-attention is bidirectional, we illustrate the computation of horizontal cross-attention from $Feat^0$ to $Feat^1$. First, we compute horizontal self-attention on $Feat^0$, ensuring that each pixel of $Feat^0$ encodes all pixel information within its corresponding row. To introduce more non-linear transformations, we utilize 2D convolution with a kernel size of 1 instead of linear layers for $Feat^0$ encoding, generating Q, K, and V, which are then combined to produce $Feat^0_{SA}$:

$$TexQ, K, V = Conv_{1\times 1}(Feat^0) \tag{1}$$

$$Feat^0_{SA} = Softmax\left(\frac{Q \times (K)^T}{\sqrt{C'}}\right) \times V, \tag{2}$$

where $Q \in \mathbb{R}^{\frac{H}{8}\times \frac{W}{8}\times C'}$, $K \in \mathbb{R}^{\frac{H}{8}\times C'\times \frac{W}{8}}$, and $V \in \mathbb{R}^{\frac{H}{8}\times \frac{W}{8}\times C'}$ denote the query, key, and value, $Conv_{1\times 1}$ represents the 2D convolutional layer with a kernel size of 1, *Softmax* is the softmax units, $(\cdot)^T$ is the transpose operation.

After obtaining $Feat^0_{SA}$, we compute horizontal cross-attention with $Feat^1$. Specifically, we first obtain the positional encoding of $Feat^0$. Then, we use a convolutional layer with a kernel size of 1 to convolve $Feat^0_{SA}$ and add it to the sine positional encoding [44] to obtain $Q^h \in \mathbb{R}^{C'\times \frac{H}{8}\times \frac{W}{8}}$. Similarly, we compute $K^h \in \mathbb{R}^{C'\times \frac{H}{8}\times \frac{W}{8}}$ for $Feat^1$ using the same method, while $Feat^1$ serves as V directly. To obtain the attention feature map in the horizontal direction, we re-encode $Q^h \in \mathbb{R}^{\frac{H}{8}\times \frac{W}{8}\times C'}$ and $K^h \in \mathbb{R}^{\frac{H}{8}\times C'\times \frac{W}{8}}$ and perform a matmul operation with Q^h to obtain the attention scores $Scores^h \in \mathbb{R}^{\frac{H}{8}\times \frac{W}{8}\times \frac{W}{8}}$. Finally, we multiply the attention scores with $V^h \in \mathbb{R}^{\frac{H}{8}\times \frac{W}{8}\times C'}$ to obtain the feature map in the horizontal direction $HA^{01} \in \mathbb{R}^{C'\times \frac{H}{8}\times \frac{W}{8}}$, as shown in Equation (5). After encoding each pixel of $Feat^0_{SA}$ with all pixel information within its corresponding row, through the cross-attention computation between $Feat^0_{SA}$ and $Feat^1$, we have completed the encoding of pixel information in $Feat^0$ for any pixel with its corresponding row's pixel information in $Feat^1$.

$$Q^h = Conv_{1\times 1}(Feat^0_{SA}) + P \tag{3}$$

$$K^h = Conv_{1\times 1}(Feat^1) + P \tag{4}$$

$$HA^{01} = Softmax\left(\frac{Q^h \times (K^h)^T}{\sqrt{C'}}\right) \times V^h, \tag{5}$$

where P denotes the position information.

Taking into account the asymmetry of bidirectional motion fields, we choose to construct bidirectional horizontal attention, ensuring that each pixel in $Feat^0$ and $Feat^1$ encodes all pixels in the corresponding row of the other feature. For the horizontal direction attention from $Feat^1$ to $Feat^0$, HA^{10}, we obtain it using the same operations as described above, with $Feat^1$ as the source feature and $Feat^0$ as the target feature.

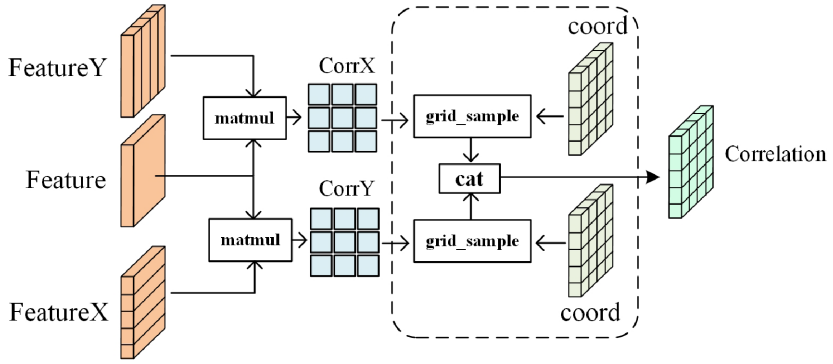


Fig. 3. The process of constructing a unidirectional 3D correlation volume.

3.2.2 Vertical Attention. Vertical attention calculation is similar to horizontal attention calculation. Here we illustrate the unidirectional vertical attention from $Feat^0$ to $Feat^1$. First, we compute self-attention vertically on the source feature $Feat^0$, allowing each pixel in Feat0 to encode all pixels in its column. Subsequently, the result of self-attention calculation is vertically cross-attended with Feat1. Unlike horizontal attention calculation, we encode Q, K, and V differently. For vertical attention calculation, we encode them as $Q^v \in \mathbb{R}^{\frac{W}{8}\times\frac{H}{8}\times C'}$, $K^v in \mathbb{R}^{\frac{W}{8}\times C'\times\frac{H}{8}}$, and $V^v \in \mathbb{R}^{\frac{W}{8}\times\frac{H}{8}\times C'}$, so that the model focuses on other pixels in the same column as each pixel and other pixels in the same column of the corresponding pixel in the target feature. We multiply Q^v, K^v, and V^v to obtain the vertical cross-attention [25], denoted as $VA^{01} \in \mathbb{R}^{C'\times\frac{H}{8}\times\frac{W}{8}}$:

$$VA^{01} = Softmax\left(\frac{Q^v \times (K^v)^T}{\sqrt{C'}}\right) \times V^v. \tag{6}$$

We exchange the source feature and the target feature and obtain the vertical cross-attention VA^{10} from $Feat^1$ to $Feat^0$ in the same way.

3.2.3 Constructing 3D Correlation Volume. After obtaining the bidirectional cross-attention feature maps of $Feat^0$ and $Feat^1$ in both horizontal and vertical directions, we utilize them along with the original feature maps to construct our 3D correlation volume. This correlation volume is bidirectional, simulating the asymmetry of the bidirectional motion field. We illustrate our 3D correlation volume taking the example of the unidirectional motion field from I^0 to I^1, as shown in Figure 3.

We multiply the original features with the horizontal and vertical attention feature maps separately to compute the feature matching relationship between $Feat^0$ and $Feat^1$ in these two directions, resulting in $CorrX \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times\frac{W}{8}}$ and $CorrY \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times\frac{H}{8}}$. Subsequently, based on the current motion information, we perform a simple 1D search in both horizontal and vertical directions to construct two 3D correlation volumes $Correlation1, Correlation2 \in \mathbb{R}^{H\times W\times(2R+1)}$, where R represents the search radius. We concatenate the 3D correlation volumes constructed from the two 1D searches along the channel dimension to achieve 2D modeling effect.

3.3 Bidirectional GRU Update Block

We incorporate a motion information update block with Bi-GRU units to generate residuals for bidirectional optical flow and intermediate features. Our block takes intermediate features, bidirectional optical flow from the previous stage, and motion features from bidirectional correlation volume matching as inputs, inspired by RAFT [42]. While RAFT employs unidirectional GRU

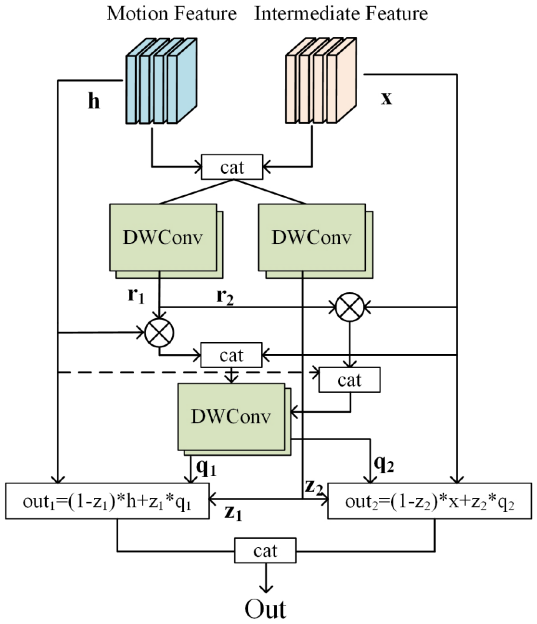


Fig. 4. The structure of Bi-GRU.

units implemented by spatial separable convolutions to model long-range dependencies in motion information, AMT [24] inherits this structure and replaces GRU units with multiple convolutional layers for computational efficiency and faster inference. In contrast, we use Bi-GRU implemented by depth-wise separable convolutions, capturing bidirectional dependencies of motion information (forward and backward) without significantly increasing computation or inference time.

As shown in Figure 4, we define the motion feature as the current state h and the intermediate feature as the hidden state x. We concatenate these two states along the channel dimension. Then, we pass them through reset gate and update gate implemented by depth-wise separable convolutions to generate $r1$ and $z1$, respectively. Subsequently, the candidate hidden state $q1$ is generated by the following equation and finally produces the modeling result:

$$r_1 = DWConv(Cat(h, x)) \tag{7}$$

$$q_1 = tanh(DWConv(Cat(r_1 \times h, x)) \tag{8}$$

$$out_1 = (1 - z_1) \times h + z_1 \times q_1, \tag{9}$$

where $DWConv$ is the depth-wise separable convolutional layers, Cat is the concat operation, $tanh$ is the tanh function.

Thus, we have achieved forward long-range dependency modeling for motion features and intermediate features. For backward long-range dependency modeling for both, we use motion features as h and intermediate features as x. By applying the same operations, we obtain out_2. We concatenate out_1 and out_2 along the channel dimension to capture bidirectional dependencies of motion information.

3.4 Multi-Scale Context-Aware Synthesis Block

AMT [24] generates multi-field optical flows, mask, and residual based on its estimated motion information and finally selects the most suitable intermediate frame through convolutional layers.

This approach achieves satisfactory results at a low cost, but the final interpolation result heavily relies on the multiple optical flows. We believe that multi-scale contextual information is crucial for the accuracy of the flow field [31]. It can provide rich edge information for flow estimation and further eliminate artifacts and blur in the interpolation result caused by inaccurate optical flow.

Specifically, we warp the multi-scale contextual information generated by the context encoder in Figure 2 with our estimated bidirectional flow fields to the desired timestep. Then, we use bilinear interpolation to upsample to a uniform scale and concatenate it with the intermediate features and bidirectional flow fields along the channel dimension. Through convolutional layers, residual structures, and transpose convolutions, we generate multi-field optical flows, mask, and residual.

4 Experiments

In this section, we will introduce the dataset we used and the implementation details. We have designed a series of comparative and ablation experiments to demonstrate the superiority of the model and the effectiveness of its various modules. Additionally, we have visualized and compared the model's inference results to more intuitively showcase the model's performance in reconstructing intermediate frames.

4.1 Datasets

4.1.1 Training Dataset. We train FBTCV on the Vimeo-90K [49] training set, similarly to most other VFI methods. Vimeo-90K comprises a total of 51,312 triples with a resolution of 448 × 256. For evaluation purposes, the test set consists of 3,782 triples, while the remaining triples are allocated for training our model. To minimize hardware resource requirements during training, we crop patches from the original images which is fetched from Vimeo-90K training set [49] with a resolution of 224 × 224. Additionally, to enrich our training set, we randomly augment it by performing horizontal and vertical flipping, chronological flipping, and rotating by 90 degrees, similar to other methods [16, 22, 33].

In addition, following the setup of other methods [34, 39], we further trained our model on X-Train [39] and conducted testing on the Xiph [30] dataset. We utilized the loss functions L_{char} and L_{cen} along with their corresponding weights λ_{ch} and λ_{cen}, as described in Section 4.2.1. The initial learning rate was set to 10^{-4} and eventually decreased to 10^{-5}. The entire training process lasted for 200 epochs with a batch size of 8. X-Train consists of 408 sets, each comprising 65 sequential frames at a resolution of 768 × 768, extracted from 4K video content. To enhance data diversity, various augmentation techniques are applied, including random flips, rotations, sequence reversals, and the extraction of 512 × 512 patches.

4.1.2 Evaluation Datasets. We test the performance of FBTCV on the Vimeo-90K [49] test set, UCF101 [40] dataset, Middlebury(M.B.) [1] dataset, SNU-FILM [10] dataset, and Xiph dataset [30]. For all the tables in this section, the best performance is labeled by us in bold font.

UCF101: The UCF101 dataset contains videos with a large variety of human actions. There are 379 triplets with a resolution of 256 × 256.

M.B.: The Middlebury benchmark is widely used to evaluate VFI methods. The image resolution in this dataset is around 640 × 480. We report the average **interpolation error (IE)** of the Middlebury-OTHER set.

SNU-FILM: SNU-FILM dataset contains 1,240 frame triplets, whose width ranges from 368 to 720 and height ranges from 384 to 1,280. With respect to motion magnitude, it is partitioned into four exclusive parts, namely Easy, Medium, Hard, and Extreme.

Xiph: The Xiph dataset consists of 8 video clips with a resolution of 4K, encompassing diverse scenes such as indoor settings, outdoor environments, crowds, natural landscapes, and urban street

views [24]. For experimentation, to reduce computational costs, the 4K video frames were resized to a 2K resolution, forming the "Xiph-2K," which evaluates the model's ability to recover image details from resolutions lower than the original. Additionally, cropping was used to extract local regions at a 2K resolution, creating the "Xiph-4K," allowing observation of the model's handling of specific features within small areas. The former preserves all texture details of the original video frames at a lower resolution, while the latter retains the local motion variations present in the original 4K resolution.

4.2 Implementation Details

4.2.1 Loss Function. We use three loss functions for end-to-end training which follows the design of AMT [24], which including Charbonnier loss [6] L_{char}, the census loss [29] L_{cen} and optical flow loss L_{flow}. The total loss L is

$$L = \lambda_{char} L_{char} + \lambda_{cen} L_{cen} + \lambda_{flow} L_{flow}. \tag{10}$$

Where λ_{char}, λ_{cen}, and λ_{flow} are weights for each loss, we set these three weights as 1.0, 1.0, and 0.002.

4.2.2 Training Details. FBTCV was implemented using the Pytorch framework and trained on a single NVIDIA A30 Tensor Core GPU. We used AdamW [19] as the optimizer with a weight decay of 10^{-3}. The initial learning rate is 3×10^{-4}, which is decayed to 3×10^{-6} by cosine annealing strategy during the training process, and the whole training process requires a total of 300 epochs with a batch size set to 32.

4.3 Performance Evaluation

4.3.1 Objective Results. We compare FBTCV to current SOTA methods including IFRNet [20], AdaCoF [22], BMBC [35], ABME [36], RIFE [16], DQBC [54], M2M [14], EMA-VFI [53], FLDR [34], AMT [24], and DAIN [2]. We measure IE for M.B. [1] dataset, and for other dataset, we measure **peak signal-to-noise ratio (PSNR)**, **structural similarity (SSIM)**. We also measure the model's size by its number of parameters and inference speed. To ensure fair comparison results, the inference speed of all comparing methods were tested on the NVIDIA A30 Tensor Core. For other performance metrics that we compared, we use the data reported in their paper directly. For metrics that they didn't test, we used their pre-trained weights to test in the same environment.

In the comparison between FBTCV and AMT [24], the main factor affecting inference time is the computation method of the correlation volume. Since we added relevant components to construct a bidirectional 3D correlation volume and reduced the number of channels in the matching results of the correlation volume, the parameter count of FBTCV is similar to that of AMT, while ours speed is faster.

Tables 1, 2, and 3 present the objective metrics comparing our model with other SOTA methods, where † indicates that the method was trained on the Vimeo-90K dataset, while ‡ indicates training on the X-Train [39] dataset. It can be observed that FBTCV achieves the best performance on the UCF101 [38] and M.B. [1] datasets. Compared to our baseline AMT [24], our model achieves similar fitting results while using half of its inference time. M2M-PWC [14], AdaCoF [22], and RIFE [16] achieve faster speeds, but FBTCV demonstrates advantages in model robustness. Both BMBC [35] and ABME [36] derive optical flow from correlation volumes and approximate intermediate flows in a two-stage manner, which incurs a considerable amount of inference time and performs lower than FBTCV. Overall, FBTCV, as an efficient VFI framework, strikes a comprehensive balance between performance and inference speed, demonstrating robustness.

Table 1. The Evaluation of Various Interpolation Methods on Vimeo-90K Test Set, Middlebury-OTHER Dataset, and UCF101 Dataset

Method	Vimeo-90K [49]		M.B. [1]	UCF101 [40]		Parameters	Runtime
	PSNR	SSIM	IE	PSNR	SSIM	(Million)	(ms)
IFRNet† [20]	35.80	0.979	1.95	35.29	0.969	**5.0**	**11**
AdaCoF† [22]	32.0	0.971	2.31	35.08	0.966	21.8	18
BMBC† [35]	35.01	0.976	2.04	35.15	0.969	11.0	854
ABME† [36]	36.18	0.980	2.01	35.38	0.969	17.5	386
RIFE† [16]	35.61	0.978	1.96	35.28	0.969	9.8	28
M2M-PWC† [14]	35.49	0.978	2.09	35.32	0.970	7.6	29
EMA-VFI† [53]	36.07	0.979	1.94	35.34	0.969	22.0	50
DQBC† [54]	**36.37**	0.981	1.86	35.35	0.969	18.3	39
AMT-L† [24]	36.35	**0.982**	1.87	**35.42**	0.970	14.38	57
Ours†	36.33	**0.982**	**1.85**	**35.42**	**0.971**	14.44	38

The runtime is measured at a resolution of 640 × 480.

Table 2. The Evaluation of Various Interpolation Methods on SNU-FILM Dataset

Method	SNU-FILM [10]				Parameters	Runtime
	Easy	Medium	Hard	Extreme	(MB)	(ms)
IFRNet† [20]	40.03/0.990	35.94/0.979	30.41/0.935	25.05/0.858	**5.0**	**26**
Adacof† [22]	39.80/0.990	35.05/0.975	29.46/0.924	24.31/0.843	21.8	52
BMBC† [35]	39.90/0.990	35.31/0.977	29.33/0.927	23.92/0.843	11.0	2234
ABME† [36]	39.59/0.990	35.77/0.978	30.58/0.936	25.42/0.863	17.5	765
RIFE† [16]	40.06/0.991	35.75/0.979	30.10/0.933	24.84/0.853	9.8	78
M2M-PWC† [14]	39.66/**0.991**	35.74/0.980	30.32/0.936	25.07/0.860	7.6	77
EMA-VFI† [53]	39.81/0.990	35.88/0.979	30.69/0.937	**25.47**/0.862	22.0	116
DQBC† [54]	**40.15**/0.990	**36.10**/0.979	**30.78**/0.937	25.41/0.862	18.3	122
AMT-L† [24]	39.95/**0.991**	36.09/**0.981**	30.75/**0.938**	25.41/**0.864**	14.38	172
Ours†	39.84/0.990	35.96/0.980	30.63/0.937	25.36/0.862	14.44	104

The runtime was tested at a resolution of 1280 × 720, which is the resolution of the SNU-FILM test set.

The 4D correlation volume is an all-pixel feature matching which is computationally expensive but has good accuracy. Since the bidirectional 3D correlation volume is matched using every horizontal and vertical pixel of the two reference frames and the SNU-FILM [10] test set includes numerous instances of dramatic lighting changes and non-linear motion, there are situations that can cause some of the motion information to be incorrectly matched on the two reference frames. This is especially noticeable on large resolution videos, due to the fact that the span of motion becomes larger on high resolution video frames, and incorrect matching of pixel features can give incorrect results for the estimation of the optical flow. Therefore, the performance of FBTCV on SNU-FILM [10] is worse than on other datasets.

The speed advantage of FBTCV becomes increasingly apparent when handling 2K video frames. The Xiph [30] test set features complex texture structures and subtle variations in detail. In high-resolution video frames, the changes in image details and texture structures become even more

Table 3. The Evaluation of Various Interpolation Methods on Xiph Dataset

Method	Xiph-2K [30]		Xiph-4K [30]		Parameters	Runtime
	PSNR	SSIM	PSNR	SSIM	(Million)	(ms)
ABME† [36]	**36.53**	0.964	33.73	0.901	17.5	1,570
BMBC† [35]	32.82	0.928	31.19	0.880	11.0	6,566
DAIN† [2]	35.95	0.940	33.49	0.895	24.0	2,132
AMT-L† [24]	36.27	0.940	**34.49**	0.903	14.38	472
FLDR‡ [34]	33.0	0.894	34.16	0.913	**0.9**	529
RIFE† [16]	36.13	0.938	33.72	0.894	9.8	180
IFRNet† [20]	36.00	0.936	33.99	0.893	5.0	**79**
M2M-PWC† [14]	36.45	**0.967**	33.93	0.945	7.6	154
Ours†	36.43	0.965	34.43	**0.947**	14.44	248
Ours‡	34.31	0.954	33.01	0.940	14.44	248

The runtime is measured on 2K resolutions.

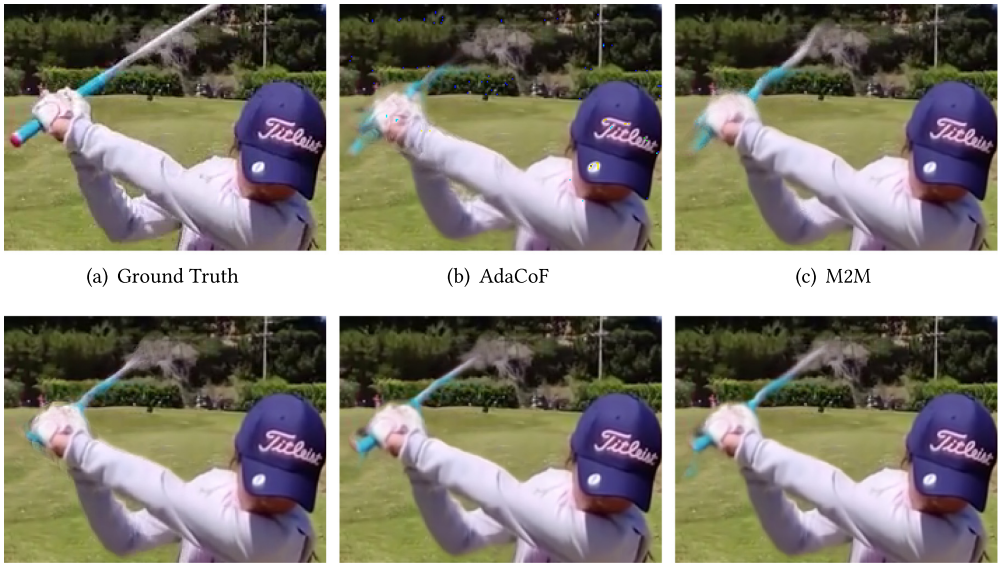
pronounced. On the Xiph-2K [30] and Xiph-4K [30] test sets, FBTCV achieved the second-highest or third-highest PSNR and the highest SSIM. Notably, on Xiph-2K [30], it outperformed our baseline while nearly reducing the inference time by half. Although ABME [36] delivered results comparable to FBTCV, its inference time was approximately six times longer than ours. This indicates that while FBTCV's matching mechanism introduces slight pixel-level differences to some extent, it preserves most of the texture structures in the video frames with significantly less time compared to other methods utilizing correlation volume.

Efficient methods such as RIFE [16], IFRNet [20], and M2M-PWC [14] have shown impressive accuracy in handling motion on Xiph-2K [30]. However, when 4K video frames are cropped to a 2K resolution, the motion of objects within the video frames is further amplified, their performance falls short of FBTCV. This is primarily due to their lack of attention to the correlation of optical flow or their disregard for the symmetry inherent in optical flow.

Although X-Train [39] is a high-resolution, high-frame-rate video dataset, whereas Vimeo-90K [49] has a comparatively lower resolution, Vimeo-90K [49] contains a more diverse range of scenes and object motion patterns compared to X-Train [39]. It encompasses a variety of scenarios, including indoor, outdoor, static backgrounds, and dynamic backgrounds, as well as small-scale local motion and large-scale global motion. This diversity helps enhance the generalization capability of our model. In addition, both the Xiph [30] and Vimeo-90K [49] datasets have a default timestep of 0.5, whereas the timestep for video frames selected for training in X-Train [39] is random. As a result, the version trained on X-Train [39] performs slightly worse on the Xiph [30] test set compared to the version trained on Vimeo-90K [49].

4.3.2 Subjective Results. We conducted visual comparisons between FBTCV and several other methods with similarly fast inference times, including AdaCoF [22], IFRNet [20], M2M [14], and AMT [24], and we selected challenging video frame triplets from the test set for visual comparison with the aforementioned representative methods.

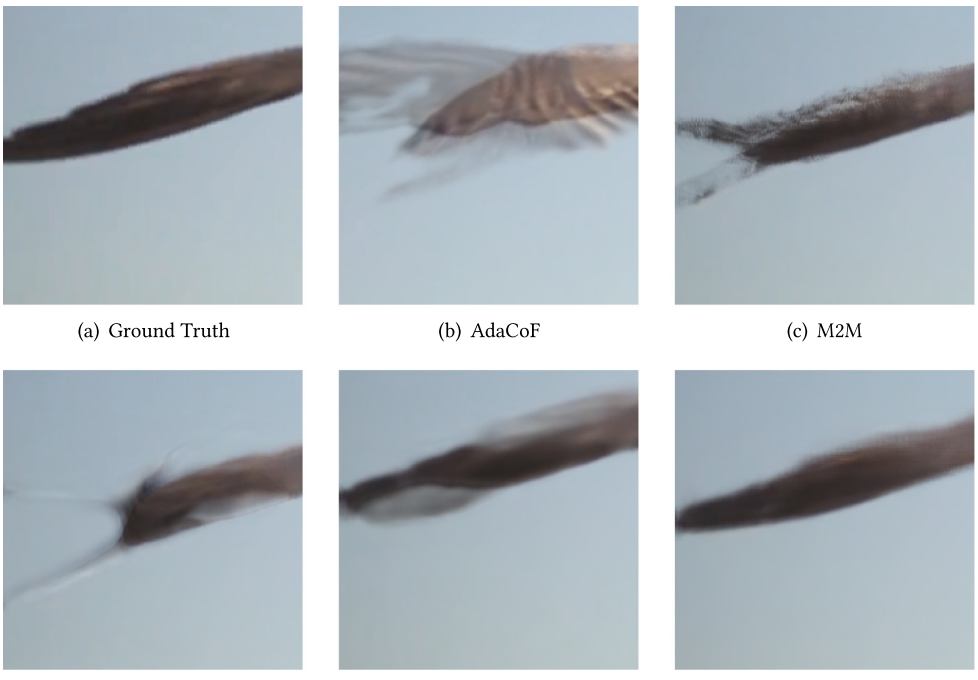
As depicted in Figures 5 and 6, we showcase the reconstruction quality of FBTCV on the two most challenging test sets within the SNU-FILM [10] dataset. These test sets present scenarios where motion spans significant temporal and spatial ranges rather than occurring within closely related time sequences, and the resolution is notably high. In such challenging conditions, several existing methods encounter issues such as distortion, loss of features, and blurring.



(a) Ground Truth (b) AdaCoF (c) M2M

(d) IFRNet (e) AMT (f) FBTCV

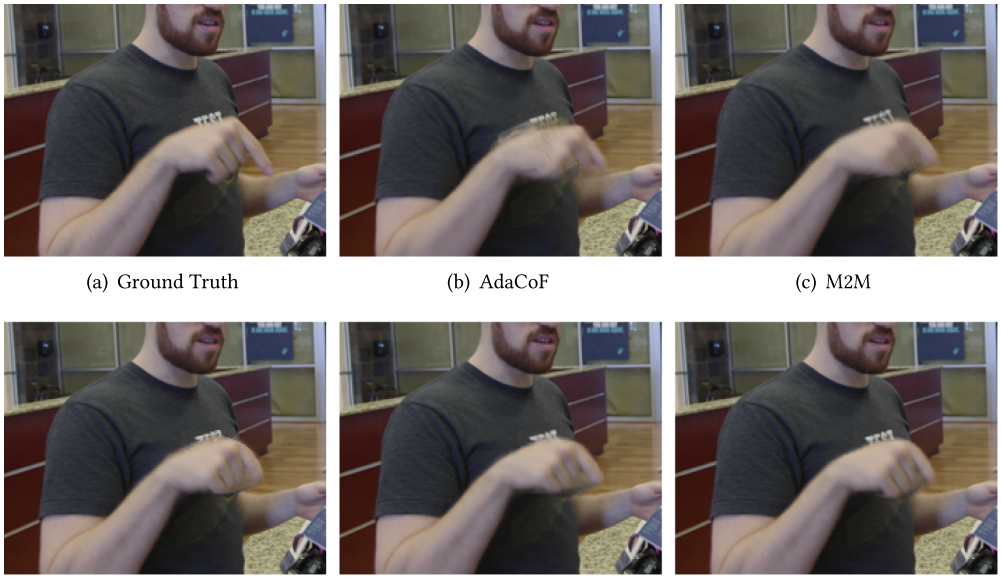
Fig. 5. Comparison of visualization results on the "Extreme" set of SNU-FILM [10] dataset.



(a) Ground Truth (b) AdaCoF (c) M2M

(d) IFRNet (e) AMT (f) FBTCV

Fig. 6. Comparison of visualization results on the "Hard" set of SNU-FILM [10] dataset.



(a) Ground Truth (b) AdaCoF (c) M2M

(d) IFRNet (e) AMT (f) FBTCV

Fig. 7. Comparison of visualization results on small but fast motion.

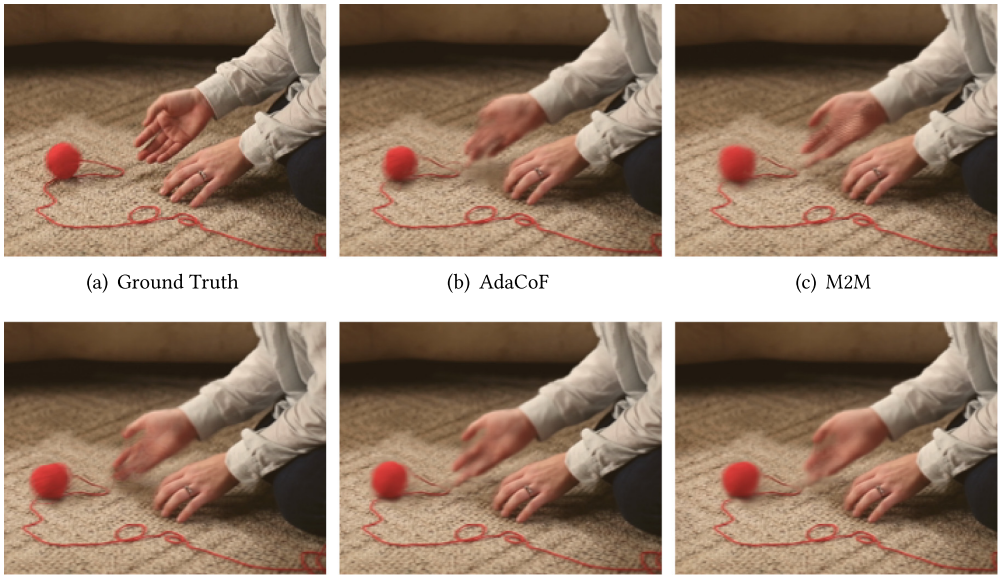
In contrast, FBTCV demonstrates its capability by effectively restoring a substantial amount of features and fine textures, thus successfully addressing motion distortion. Despite operating at large-scale resolutions, FBTCV achieves a high inference speed while maintaining a certain level of accuracy.

FBTCV also excels in producing visually appealing interpolation results when handling small and fast-moving objects, as demonstrated in Figure 7. In scenarios involving such objects, many models struggle to maintain clarity due to the inherent challenges in capturing key features amidst rapid and often non-linear motion. This difficulty complicates optical flow prediction.

However, FBTCV effectively addresses these challenges through its innovative optical flow update module utilizing Bi-GRU. This module enhances the model's ability to retain critical features, even in deeper network layers. Additionally, FBTCV integrates a fusion module that leverages multi-scale contextual information, further improving the accuracy and robustness of feature preservation during interpolation.

Handling large motion is a common challenge in VFI tasks. Large motion requires the optical flow to perform search and matching over a wide range, which reduces the accuracy of flow estimation. Additionally, in VFI tasks, the reconstructed video frames should remain consistent with the real frames in terms of timing and content. The drastic changes associated with large motion can lead to incorrect motion estimation in the predicted intermediate frames. As shown in Figure 8, FBTCV addresses this issue by expanding the search radius within the bidirectional correlation volume, enhancing the ability to reconstruct large motions. This expanded search radius allows the optical flow to perform search and matching over a larger pixel area, improving the accuracy of handling significant motion changes.

In video frames where motion occlusion occurs, certain pixels lack corresponding relations between preceding and following frames, resulting in discontinuous motion fields and a deficiency of interpolation information for synthesizing intermediate frames. FBTCV exhibits impressive performance in addressing such occlusion scenarios. Despite the challenges posed by missing



(a) Ground Truth (b) AdaCoF (c) M2M

(d) IFRNet (e) AMT (f) FBTCV

Fig. 8. Comparison of visualization results on large motion.

information, FBTCV excels in accurately estimating bidirectional motion fields by leveraging dense optical flow information.

This capability underscores the robustness of our model in handling complex motion scenarios, as depicted in Figure 9. By effectively integrating dense optical flow information, FBTCV overcomes the limitations imposed by motion occlusion, ensuring reliable and high-quality interpolation results even in challenging video sequences.

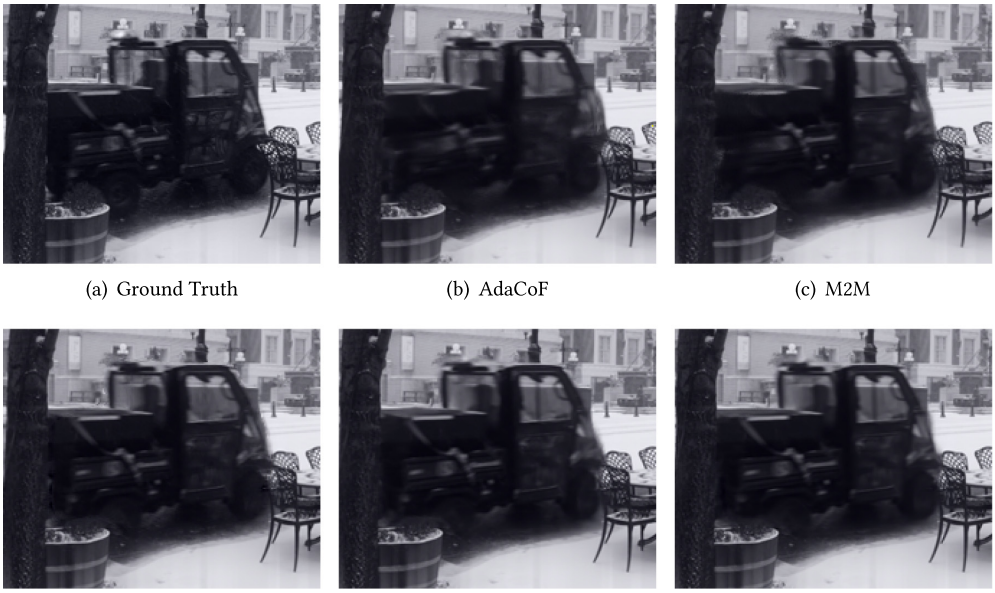
When people are engaged in boxing, their arms can be considered fast-moving objects that may experience occlusion, as illustrated in Figure 10. In high-resolution video frames, non-linear and fast-moving objects become more challenging to handle, while the details and textures in the images are further magnified. AMT [24] exhibits noticeable arm distortion and blurring, whereas IFRNet [20] demonstrates insufficient prediction for occluded objects. In contrast, FBTCV demonstrates its capability by effectively restoring a substantial amount of features and fine textures, thus addressing motion distortion for high-resolution video frames.

We visualized the bidirectional optical flow generated by FBTCV and superimposed it on the original ground truth frames to demonstrate the model's capability in fitting motion information, as shown in Figure 11. FBTCV accurately captures the motion between frames while preserving rich texture details. For challenging motion boundaries, our bidirectional optical flow also aligns well, maintaining clear and precise boundaries.

4.4 Ablation Study

In order to systematically illustrate the effectiveness of each module in our proposed method, we conduct a series of experiments with several metrics. In this section, the runtime is measured with a resolution of 256×448.

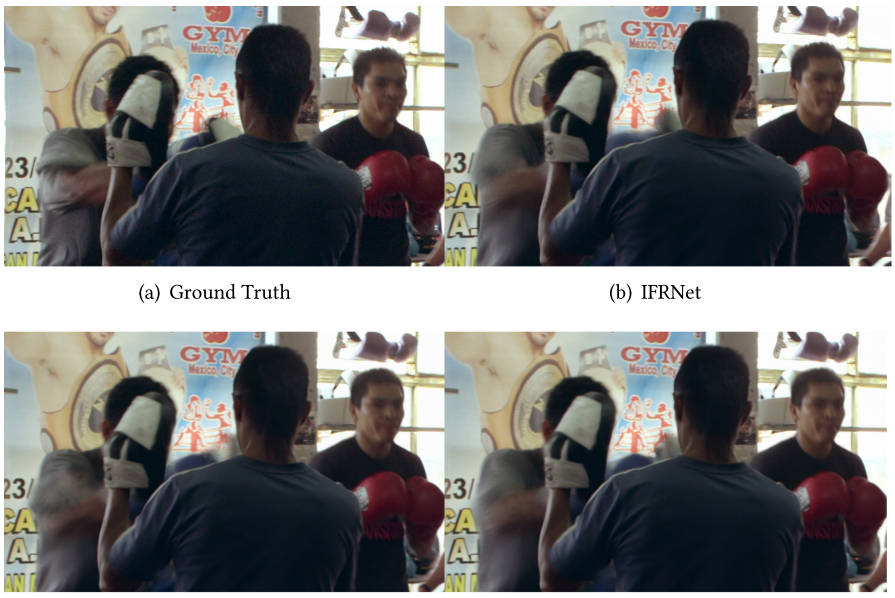
4.4.1 Radius Ablation Experiment. We conducted ablation experiments on the search radius for pixel-wise feature matching, as shown in Table 4, where we use R to denote the radius. In AMT



(a) Ground Truth (b) AdaCoF (c) M2M

(d) IFRNet (e) AMT (f) FBTCV

Fig. 9. Comparison of visualization results on occluded motion.



(a) Ground Truth (b) IFRNet

(c) AMT (d) FBTCV

Fig. 10. Comparison of visualization results on Xiph-4K [30] test set.

[24], the search area is a 7×7 window, resulting in a search radius of 3. In RAFT [42], the search radius is 4. Since we did not build a pyramid structure for the correlation volumes, we doubled our search radius from R = 4 to simultaneously address small displacements and large motions. When

(a) Overlap (b) $Flow_{0\to1}$ (c) $Flow_{1\to0}$ (d) Bidirectional Flow

Fig. 11. Comparison of visualization of optical flow and overlapped frames.

Table 4. Validity of Different Radius of Our Model

Setting	Vimeo-90K	SNU-FILM			
		Easy	Medium	Hard	Extreme
	PSNR	PSNR	PSNR	PSNR	PSNR
R = 3	36.30	39.95	36.01	30.55	25.28
R = 4	36.31	39.93	35.96	30.57	25.29
R = 8	36.30	**40.0**	36.01	30.56	25.32
R = 16	**36.33**	39.84	35.96	**30.63**	**25.36**
R = 32	36.27	39.77	35.94	30.62	**25.36**

constructing our correlation volumes with radii of 3 and 4, the model fits well with smaller motions, as reflected in its performance on the Easy and Medium test sets of the SNU-FILM [10] dataset. This is because pixels corresponding to small displacements often lie in adjacent regions, and enlarging the search radius may lead to matching errors and errors in motion fitting. However, as we increased the search radius, the model's fitting performance improved for larger motions, as evidenced by its performance on the Hard and Extreme test sets of the SNU-FILM [10] dataset. Nevertheless, the model's fitting performance deteriorated for smaller motions. Through experimental comparison, we ultimately selected R = 16 as the search radius for our correlation volumes.

4.4.2 Update Ablation Experiment. In the optical flow update stage, RAFT [42] adopts a GRU implemented by SepConv, while in AMT [24], convolutional layers are directly used instead of GRU. We believe that neither of these approaches considers updating motion information from both forward and backward directions. Therefore, we propose a Bi-GRU implemented by depth-wise separable convolutions and compare it with the above two methods. As shown in Table 5 and Figure 12, where *Convlayer* is the method that AMT used and $SepConv - GRU$ represents the design in RAFT, our method achieves performance improvement and avoids phenomena such as blurring and artifacts compared to the other two methods. Additionally, due to the use of depth-wise separable convolutions, our method does not introduce significant computational burden.

4.4.3 Synthesis Ablation Experiment. For the synthesis module, we compared the synthesis methods used in AMT [24], commonly used methods such as the context-aware synthesis method used in RIFE [16], and our improved method, as shown in Table 6 and Figure 13. Our method

Table 5. Validity of Different GRU of Our Model

Setting	Vimeo-90K	SNU-FILM		Runtime
		Hard	Extreme	
	PSNR	PSNR	PSNR	(ms)
Convlayer	36.28	30.56	25.23	**22**
SepConv-GRU	36.30	30.54	25.27	26
Bi-GRU	**36.33**	**30.63**	**25.36**	28

(a) Ground Truth

(b) Convlayer

(c) SepConv-GRU

(d) Bi-GRU

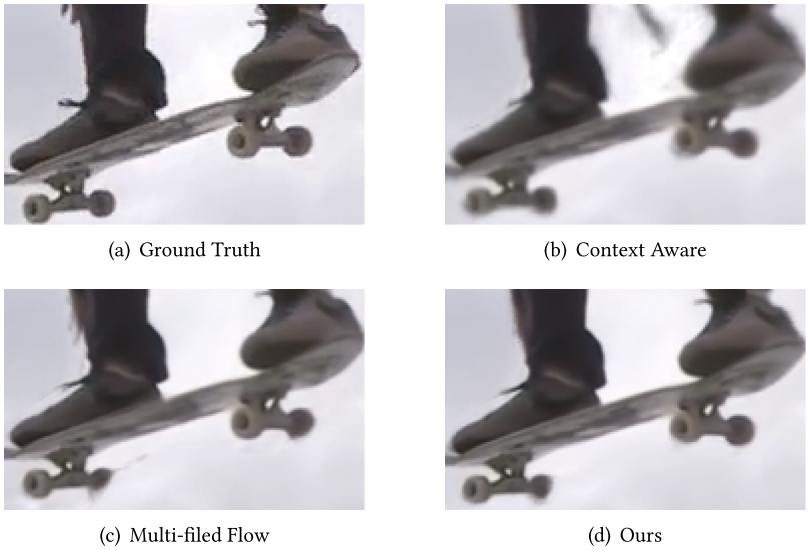
Fig. 12. Comparison of visualization results on different update strategy.

Table 6. Validity of Different Synthesis Block of Our Model

Setting	Vimeo-90K	SNU-FILM		Runtime
		Hard	Extreme	
	PSNR	PSNR	PSNR	(ms)
Multi-filed flow	36.29	30.58	25.30	**27**
Context aware	36.18	30.48	25.21	31
Ours	**36.33**	**30.63**	**25.36**	28

integrates multi-scale contextual information on the basis of the multi-field optical flow interpolation method proposed in AMT, achieving performance improvement with almost no additional computational resources.

4.4.4 Attention Strategy Ablation Experiment. We conducted ablation experiments on the strategy used to compute horizontal and vertical attention feature maps, as shown in Table 7. Here, *Pos* indicates sine positional encoding [44], and *DA* indicates dual attention. When we did not use



(a) Ground Truth (b) Context Aware

(c) Multi-filed Flow (d) Ours

Fig. 13. Comparison of visualization results on different synthesis strategy.

Table 7. Validity of Different Components of Attention Strategy

Setting	Middlebury-OTHER	UCF101	Runtime
	IE	PSNR	(ms)
FBTCV w/o Pos	1.87	35.41	28
FBTCV w/o DA	1.87	35.40	**22**
FBTCV	**1.85**	**35.42**	28

positional encoding, the model's performance slightly decreased. This demonstrates that positional encoding strengthens the model's perception of positional information in the image, helping the model better establish long-term feature connections when computing attention feature maps. When we directly computed horizontal and vertical attention feature maps from the original feature maps, i.e., without using dual attention, the model's performance decreased. This is because without dual attention, it is difficult for the model to learn proper aggregation since the same columns or rows in the source features may not contain corresponding pixels.

4.4.5 Correlation Volume Ablation Experiment. We conducted a series of ablation experiments to evaluate the effectiveness of our bidirectional 3D correlation volumes. The results include both quantitative metrics and inference times. In these experiments, we replaced our bidirectional 3D correlation volumes with the partial correlation volume from PWC-Net [41] and the dense bidirectional correlation volume from AMT [24].

As shown in Table 8, our bidirectional 3D correlation volumes effectively capture the approximate correspondences between frames in the VFI tasks, and they require less inference time. Although the partial correlation volumes in PWC-Net demand less inference time either, they focus solely on unidirectional frame-to-frame correspondence while ignoring the symmetry of optical flow, leading to inferior performance.

Table 8. Validity of Different Correlation Volume

Setting	Vimeo-90K	Runtime
	PSNR	(ms)
PWC	36.04	**19**
BFCV	**36.37**	46
BTCV	36.33	28

PWC is the design of PWC-Net, BFCV is the bidirectional 4D correlation volume, and BTCV represents the correlation volume of ours.

5 Conclusions

In this article, we address the computational time issue caused by the use of bidirectional 4D correlation volumes to capture the correlation of bidirectional flow fields in existing methods and propose a novel VFI framework, named FBTCV. With the increasing prevalence of videos in everyday life, the efficiency of FBTCV greatly meets the demands of real-world production environments. As mentioned in our introduction, our goal is to optimize the slow inference speed of current methods using correlation volumes when processing high-resolution video frames, thereby accelerating inference while maintaining reasonable accuracy. The introduction of bidirectional 3D correlation volumes may result in matching errors when dealing with large motions, which are further amplified in high-resolution video frames. Consequently, the current version of FBTCV experiences some accuracy loss when handling inputs containing intense non-linear motion and severe lighting changes. In the future, our objective is to make targeted improvements to address this issue.

References

[1] Simon Baker, Stefan Roth, Daniel Scharstein, Michael J. Black, J. P. Lewis, and Richard Szeliski. 2007. A database and evaluation methodology for optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*.

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. 2019. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3703–3712.

[3] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. 2019. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 3 (2019), 933–948.

[4] Wenbo Bao, Xiaoyun Zhang, Li Chen, Lianghui Ding, and Zhiyong Gao. 2018. High-order model and dynamic filtering for frame rate up-conversion. *IEEE Transactions on Image Processing* 27, 8 (2018), 3813–3826.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*. Springer, 213–229.

[6] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing, Vol. 2*, IEEE, 168–172.

[7] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. 2021. Transunet: Transformers make strong encoders for medical image segmentation. arXiv:2102.04306. Retrieved from http://arxiv.org/abs/2102.04306

[8] Xianhang Cheng and Zhenzhong Chen. 2020. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 10607–10614.

[9] Xianhang Cheng and Zhenzhong Chen. 2021. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2021), 7029–7045.

[10] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 2020. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 10663–10671.

[11] Xiangling Ding, Pu Huang, Dengyong Zhang, Wei Liang, Feng Li, Gaobo Yang, Xin Liao, and Yue Li. 2024. MSEConv: A unified warping framework for video frame interpolation. *ACM Transactions on Asian and Low-Resource Language Information Processing* (2024), 1–21. DOI: https://doi.org/10.1145/3648364

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929. Retrieved from http://arxiv.org/abs/2010.11929

[13] Mengshun Hu, Kui Jiang, Zhihang Zhong, Zheng Wang, and Yinqiang Zheng. 2024. IQ-VFI: Implicit quadratic motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6410–6419.

[14] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. 2022. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3553–3562.

[15] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. 2019. Ccnet: Criss-cross attention mantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 603–612.

[16] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. 2022. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*. Springer, 624–642.

[17] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9000–9008.

[18] Xin Jin, Longhai Wu, Guotao Shen, Youxin Chen, Jie Chen, Jayoon Koo, and Cheul-hee Hahm. 2023. Enhanced bi-directional motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5049–5057.

[19] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from http://arxiv.org/abs/arXiv:1412.6980

[20] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. 2022. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1969–1978.

[21] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1, 4 (1989), 541–551.

[22] Hyeongmin Lee, Taeoh Kim, Tae Young Chung, Daehyun Pak, and Sangyoun Lee. 2020. AdaCoF: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] Pengpeng Li, An Luo, Jiping Liu, Yong Wang, Jun Zhu, Yue Deng, and Junjie Zhang. 2020. Bidirectional gated recurrent unit neural network for Chinese address element segmentation. *ISPRS International Journal of Geo-Information* 9, 11 (2020), 635.

[24] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. 2023. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9801–9810.

[25] Hezheng Lin, Xing Cheng, Xiangyu Wu, and Dong Shen. 2022. Cat: Cross attention in vision transformer. In *Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME). IEEE*, 1–6.

[26] Chunxu Liu, Guozhen Zhang, Rui Zhao, and Limin Wang. 2024. Sparse global matching for video frame interpolation with large motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19125–19134.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

[28] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. 2022. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3532–3542.

[29] Simon Meister, Junhwa Hur, and Stefan Roth. 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[30] Christopher Montgomery and H. Lars. 1994. Xiph. org video test media (Derf's collection). Retrieved from https://media.xiph.org/video/derf

[31] Simon Niklaus and Feng Liu. 2018. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1710.

[32] Simon Niklaus and Feng Liu. 2020. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5437–5446.

[33] Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision*, 261–270.

[34] Moritz Nottebaum, Stefan Roth, and Simone Schaub-Meyer. 2022. Efficient feature extraction for high-resolution video frame interpolation. arXiv:2211.14005. Retrieved from http://arxiv.org/abs/2211.14005

[35] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. 2020. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Proceedings of the 16th European Conference on Computer Vision (ECCV'20)*. Springer, 109–125.

[36] Junheum Park, Chul Lee, and Chang-Su Kim. 2021. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14539–14548.

[37] Yuxu Peng, Xin Yi, Dengyong Zhang, Lebing Zhang, Yuehong Tian, and Zhifeng Zhou. 2024. ConvMedSegNet: A multi-receptive field depthwise convolutional neural network for medical image segmentation. *Computers in Biology and Medicine* 176 (2024), 108559.

[38] Zhihao Shi, Xiaohong Liu, Chengqi Li, Linhui Dai, Jun Chen, Timothy N. Davidson, and Jiying Zhao. 2021. Learning for unconstrained space-time video super-resolution. *IEEE Transactions on Broadcasting* 68, 2 (2021), 345–358.

[39] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. 2021. Xvfi: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14489–14498.

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402. Retrieved from http://arxiv.org/abs/arXiv:1212.0402

[41] Deqing Sun, Xiaodong Yang, Ming Yu Liu, and Jan Kautz. 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[42] Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the 16th European Conference on Computer Vision (ECCV'20)*. Springer, 402–419.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*. PMLR, 10347–10357.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[45] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 568–578.

[46] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. 2023. Accflow: Backward accumulation for long-range optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12119–12128.

[47] Guangyang Wu, Xin Tao, Changlin Li, Wenyi Wang, Xiaohong Liu, and Qingqing Zheng. 2024. Perception-oriented video frame interpolation via asymmetric blending. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2753–2762.

[48] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. 2021. High-resolution optical flow from 1d attention and correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10498–10507.

[49] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision* 127 (2019), 1106–1125.

[50] Xiaohui Yang, Haoran Zhang, Zhe Qu, Zhiquan Feng, and Jinglan Tian. 2023. Video frame interpolation via residual blocks and feature pyramid networks. *IET Image Processing* 17, 4 (2023), 1060–1070.

[51] Yoonmo Yang and Byung Tae Oh. 2020. Video frame interpolation using deep cascaded network structure. *Signal Processing: Image Communication* 89 (2020), 115982.

[52] Dengyong Zhang, Pu Huang, Xiangling Ding, Feng Li, Wenjie Zhu, Yun Song, and Gaobo Yang. 2023a. L2BEC2: Local lightweight bidirectional encoding and channel attention cascade for video frame interpolation. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 2 (2023), 1–19.

[53] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. 2023b. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5682–5692.

[54] Chang Zhou, Jie Liu, Jie Tang, and Gangshan Wu. 2023. Video frame interpolation with densely queried bilateral correlation. arXiv:2304.13596. Retrieved from http://arxiv.org/abs/2304.13596

[55] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In *Proceedings of the 14th European Conference on Computer Vision (ECCV '16)*. Springer, 286–301.

[56] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159. Retrieved from http://arxiv.org/abs/2010.04159

Received 19 June 2024; revised 22 January 2025; accepted 28 February 2025