Video Frame Interpolation: A Comprehensive Survey

JIONG DONG, KAORU OTA, and MIANXIONG DONG, Muroran Institute of Technology, Japan

Video Frame Interpolation (VFI) is a fascinating and challenging problem in the computer vision (CV) field, aiming to generate non-existing frames between two consecutive video frames. In recent years, many algorithms based on optical flow, kernel, or phase information have been proposed. In this article, we provide a comprehensive review of recent developments in the VFI technique. We first introduce the history of VFI algorithms' development, the evaluation metrics, and publicly available datasets. We then compare each algorithm in detail, point out their advantages and disadvantages, and compare their interpolation performance and speed on different remarkable datasets. VFI technology has drawn continuous attention in the CV community, some video processing applications based on VFI are also mentioned in this survey, such as slow-motion generation, video compression, video restoration. Finally, we outline the bottleneck faced by the current video frame interpolation technology and discuss future research work.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**; **Computer vision tasks**;;

Additional Key Words and Phrases: Video Frame Interpolation, deep learning, convolutional neural network

ACM Reference format:
Jiong Dong, Kaoru Ota, and Mianxiong Dong. 2023. Video Frame Interpolation: A Comprehensive Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 2s, Article 78 (May 2023), 31 pages.
https://doi.org/10.1145/3556544

1 INTRODUCTION

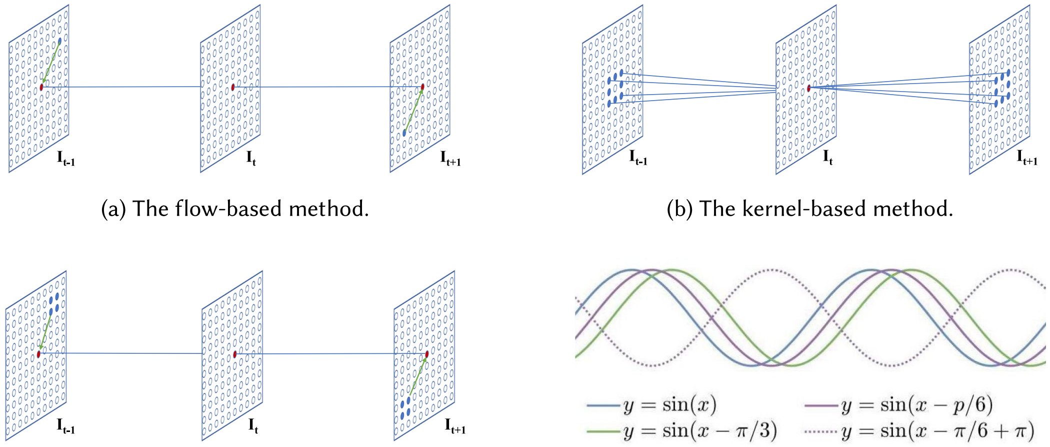
Video Frame Interpolation (VFI) is a longstanding research topic in the video processing field, which refers to synthesize non-existing frames between two successive video frames. The technology enjoys various real-world applications in the area of video processing, such as slow-motion generation [4, 50], frame rate up-conversion [6, 10], video compression [135], novel view synthesis [33], video restoration [57, 117, 127, 134], and intra-prediction in video coding [19, 136].

Traditional solutions for VFI have two steps: first estimate motion information between frames, typically based on optical flow, and then pixel synthesis [3]. However, there is a well-known problem with this kind of method: in some challenging conditions (e.g., abrupt brightness change, large motion, occlusion, illumination), the optical flow is usually tricky to estimate and may cause artifacts in the interpolated video frames.

This work is partially supported by JSPS KAKENHI Grant Numbers JP19K20250, JP20H04174, JP22K11989, Leading Initiative for Excellent Young Researchers (LEADER), MEXT, Japan, and JST, PRESTO Grant Number JPMJPR21P3, Japan. This work was also supported by the China Scholarship Council (CSC) under Grant 202108050244.

Authors' address: J. Dong, K. Ota, and M. Dong (corresponding author), Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran, Hokkaido, Japan; emails: 20096511@mmm.muroran-it.ac.jp, {ota, mx.dong}@csse.muroran-it.ac.jp.

1551-6857/2023/05-ART78 $15.00
https://doi.org/10.1145/3556544



(a) The flow-based method.

(b) The kernel-based method.

(c) The flow and kernel combined method.

(d) The phase-based method [78].

Fig. 1. The comprehensive description of the principle mechanism of VFI methods. In Figures 1(a), 1(b), and 1(c), the blue parts represent the reference points for generating the target pixel (red point).

To address the problem mentioned above, with the advancement of optical flow estimation methods based on deep **convolutional neural networks (CNNs)** [30, 46, 96, 113], some VFI algorithms adopt these models or their variations as sub-networks to interpolate frames in an end-to-end manner directly. As illustrated in Figure 1(a), the flow-based methods estimate the flow vector directly pointing to the reference location for each output pixel.

Another major trend methods for VFI are kernel-based (Figure 1(b)) [16, 26, 61, 84, 85]. Optical flow estimation is viewed as an intermediary phase that may be bypassed with a single convolution operation in these approaches. The limitation of these methods is how to balance the interpolation speed and the interpolation quality. The size of the motion that could be handled depends on the size of the kernel. The interpolation process will need more computation resources if the size of kernel is bigger; however, if the size is smaller, the method cannot resolve big motion between input frames resulting in blurry or shadows in intermediate frame. In addition, some algorithms combine the advantages of the flow-based and kernel-based approaches to improve the performance of interpolation (Figure 1(c)) [5, 143]. However, this method is closer to the flow-based method as it employs fewer reference points than the kernel-based method. Moreover, in some models, phase information is applied to address the VFI problem (Figure 1(d)). The phase-based approaches [78, 79] depict motion in individual pixel phase shifts, allowing in-between pictures to be created with simple per-pixel phase adjustment.

With technology development, the VFI technology has been applied in VR games, the smartphone camera and attracts more and more attention. At present, there is only one published paper about the overview on VFI tasks at the writing time, Parihar et al. [89] paid more attention to the advanced deep learning-based methods. Unlike their works, we attempt to provide a comprehensive overview of VFI research in this survey, we introduce all kinds of VFI methods, and then we highlight some critical issues for future research in this area. We aim to assist readers with a detailed overview of the current state-of-the-art approaches in VFI, as well as future potential research directions for researchers in this field. The following are the main contributions of our work:

- We give a comprehensive overview of VFI, including benchmark datasets, performance metrics, the systematic comparison of existing advanced methods, and the applications based on VFI.

- We summarize the performance of existing methods on some public benchmark datasets. Moreover, we also go through some of the crucial factors that influence VFI efficiency.
- We outline the challenges and future trends in VFI research. We aim to give an insightful guide to this community.

The remainder of this survey is laid out as follows: In Section 2, we first present the basic contents of VFI including the problem definition, the related benchmark datasets, and the performance metrics. Second, in Section 3, we present the existing methods for VFI problems. Section 4 focuses on the performance comparison between these methods. In Section 5, we introduce a series of applications based on VFI already proposed in different areas, including video compression, video restoration, slow motion generation, and novel view synthesis. In Section 6, we summarize some challenges and future trends about VFI. Finally, our conclusion is presented in Section 7.

2 BASIC CONCEPTS ON VIDEO FRAME INTERPOLATION

2.1 Problem Definition

According to the number and position of frame interpolation, VFI problems can be defined into two categories: single-frame interpolation and multi-frame interpolation.

Single-Frame Interpolation. As the name suggests, single-frame interpolation methods focus on synthesizing one frame between two input frames or at any arbitrary position between two frames. Given two video frames $I_0\,(x,y)$ and $I_1\,(x,y)$, these VFI algorithms aim to interpolate a non-existing frame $I_t\,(x,y)$, $t = 0.5$ or $t \in [0,1]$. The optical flow based VFI methods first estimate the bi-direction optical flow, denoted by $F_{t-1 \longrightarrow t+1}$ and $F_{t+1 \longrightarrow t-1}$, and then apply the forward or backward warping strategies to synthesize the intermediate frame. The kernel-based VFI methods estimate a pair of 2D convolution kernels $K_1\,(x,y)$ and $K_2\,(x,y)$ to convolve with $I_{t-1}\,(x,y)$ and $I_{t+1}\,(x,y)$ to compute the color of the output pixel. However, when computing large motion, the 2D kernels consume more computing resources, to address this problem, some methods estimate a pair of 1D kernels instead of 2D kernels.

Multi-Frame Interpolation. Multi-frame interpolation methods aim at interpolating multi-frames between two input frames. Given four input frames (I_{-1}, I_0, I_1, I_2), In [17], Chi et al. aimed to generate seven frames between I_0 and I_1, these seven frames are synthesized at a specific position: $I_{t_i}, t_i = \frac{i}{8}, i \in [1, 2, \ldots, 7]$. And the method Super-SloMo [50] aims to interpolate multi frames at any arbitrary time step between two frames, the approach GDCN [106] focuses on a four-frame interpolation case.

2.2 Benchmark Datasets

We present some commonly used benchmark datasets for training and evaluating VFI approaches in this subsection, including UCF101 [111], Vimeo90K [143], Middlebury [3], and other datasets used in specific papers.

UFC101. This dataset contains 13,320 videos with a large variety of human actions and a resolution of 320 × 240, it consists of 101 action classes. The DVF [73], CyclicGen [71], and CBOF-Net [37] VFI methods have been trained using the UCF101 dataset.

Vimeo90K. Vimeo90K dataset contains 4,278 videos with 89,800 independent shots downloading form the Vimeo video sharing website, and the authors resized all frames with a resolution of 448 × 256. Many VFI methods have applied Vimeo90k dataset to train including MEMC-Net [5], DSepConv [15], AdaCoF [61], CAIN [22], due to the motion of objects is much larger.

Middlebury. The Middlebury benchmark consists of an *Evaluation* set and an *Other* set. The *Other* set contains 12 examples in total, with maximum resolution of 640 × 480. Most VFI studies have reported and compared the performance of their methods on Middlebury benchmark, such as [79], [84], and [85].

Table 1. Some Widely-Used Datasets for Video Frame Interpolation

Dataset	Year	Website	Numbers of video	Resolution
Middlebury [3]	2011	Link	24	640 × 480
UCF101 [111]	2012	Link	13,320	320 × 240
SJTU 4K [110]	2013	Link	15	3840 × 2160
Ultra Video [60]	2014	Link	13	1,080 × 720, 1,980 × 1,080 3,840 × 2,160
DAVIS [93]	2016	Link	50	1,080 × 720
GOPRO [81]	2017	Link	33	1,280 × 720
Adobe240 [112]	2017	Link	71	1,280 × 720
Vimeo90K [143]	2019	Link	4,278	448 × 256
HD [5]	2019	Link	7	1280 × 720
Sony Dataset [51]	2019	Link	40	1,920 × 1,080
SNU-FILM [22]	2020	Link	31	1,280 × 720

HD. The HD dataset was proposed in [5]. Bao et al. collected seven high-definition videos from the Xiph website with high resolution from 1,280 × 544 to 1,920 × 1,080. The motions in this dataset are larger than other datasets.

YouTube240. In [50], Jiang et al. collected some 240 fps videos from YouTube to train their model. In total, the dataset consists of 1,132 video clips and 376K individual video frames. This dataset contains a wide range of scenes, including daily activities, professional sports, indoor and outdoor scenes, and capturing with static or moving cameras.

Adobe240. There are 71 videos at 240 fps with a resolution of 1,280 × 720 in the Adobe240 [112] dataset. The dataset was originally used for video deblur, and all the videos are captured with different mobile phones and consumer cameras.

DAVIS. The **DAVIS (Densely Annotated VIdeo Segmentation)** [93] dataset is consisted of 50 high-quality, full HD 1,080 video sequences and initially proposed for object segmentation in the video. Since the dataset includes both static and dynamic scenes with complex motion, Xu et al. [141] apply this dataset to evaluate their quadratic VFI algorithm.

GOPRO. The GOPRO [81] dataset was generated by a hand-held GOPRO4 Hero Black camera and composed of 3,214 pairs of blurry and sharp images at 1,280 × 720 resolution. Furthermore, this dataset contains complex motion from indoor and outdoor situations, which is difficult for some current VFI approaches.

Sony Dataset. This dataset [51] consists of 40 high-quality videos, and every video contains 1,000 frames at 1,080P, which was captured with a Sony RX V camera at 250 fps.

SNU-FILM. To assess the capabilities of the VFI approaches in terms of motion amount, Choi et al. [22] created a more comprehensive benchmark dataset called **SNU-FILM (SNU Frame Interpolation with Large Motion)** containing large motion and occlusion. The dataset consists of 31 videos of 240 fps, including 11 videos from the GOPRO dataset's test set and 20 videos collected from YouTube.

For a more practical and reliable experiment, in [1], the authors applied high resolution videos: Ultra Video [60] and SJTU 4K Video [110] datasets with 2160p resolution. In Table 1, we list the download link and comparison about all the aforementioned datasets.

2.3 Performance Metrics

The Peak Signal-to-Noise Ratio (PSNR), Structual Similarity Index (SSIM) [129], interpolation error (IE) [3], and normalized interpolation error (NIE) [3] are the most often utilized performance measures for VFI algorithms.

For VFI, given the ground truth frame $I^{GT}(x,y)$ and the interpolated frame $\hat{I}(x,y)$, the PSNR is defined as follow:

$$PSNR = 10 \cdot \log_{10}\left(\frac{L^2}{\frac{1}{N}\sum_{x,y}^{N}\left(I^{GT}(x,y) - \hat{I}(x,y)\right)^2}\right), \tag{1}$$

where L is the maximum pixel value, usually equals to 255, N is the number of pixels. The greater the value of PSNR, the better the performance of frame interpolation.

SSIM is used for measuring the similarity between two images. The range of structural similarity is −1 to 1. When the two images are exactly the same, the value of SSIM is equal to 1. Given the ground truth frame I^{GT} and the interpolated frame $\hat{I}$, SSIM is defined as:

$$SSIM = \frac{\left(2\mu_{\hat{I}}\mu_{I^{GT}} + c_1\right)\left(2\sigma_{\hat{I}I^{GT}} + c_2\right)}{\left(\mu_{\hat{I}}^2 + \mu_{I^{GT}}^2 + c_1\right)\left(\sigma_{\hat{I}}^2 + \sigma_{I^{GT}}^2 + c_2\right)}, \tag{2}$$

where $\mu_{\hat{I}}$ and $\mu_{I^{GT}}$ are the mean of $\hat{I}$ and I^{GT} respectively, and $\sigma_{\hat{I}}^2$ is the variance of $\hat{I}$, $\sigma_{I^{GT}}^2$ is the variance of I^{GT}, $\sigma_{\hat{I}I^{GT}}$ is the covariance between $\hat{I}$ and I^{GT}, c_1, c_2 are constants for avoiding instability and defined as:

$$c_1 = (k_1P)^2 \tag{3}$$

$$c_2 = (k_2P)^2 \tag{4}$$

where $k_1 = 0.01$, $k_2 = 0.03$, and P is the dynamic range of pixel values.

IE is the **root-mean-square (RMS)** difference between the ground-truth frame $I^{GT}(x,y)$ and the estimated interpolated frame $\hat{I}(x,y)$. IE is defined as:

$$IE = \left[\frac{1}{N}\sum_{(x,y)}\left(\hat{I}(x,y) - I^{GT}(x,y)\right)^2\right]^{\frac{1}{2}} \tag{5}$$

where N is the number of pixels.

The NIE between an interpolated frame $\hat{I}(x,y)$ and a ground-truth frame $I^{GT}(x,y)$ is given by:

$$NIE = \left[\frac{1}{N}\sum_{(x,y)}\frac{\left(\hat{I}(x,y) - I^{GT}(x,y)\right)^2}{\|\nabla I^{GT}(x,y)\|^2 + \mu}\right]^{\frac{1}{2}} \tag{6}$$

where N is the number of pixels, the arbitrary scaling constant is set to be $\mu = 1.0$. Lower IE or NIE indicates better performance.

In SoftSplat [83], Nicklaus and Liu also additionally incorporated the **LPIPS (Learned Perceptual Image Patch Similarity)** metric [153] which strives to measure perceptual similarity, the lower value of LPIPS means better performance of interpolation. Yang et al. [146] proposed a new dedicated quality metric for VFI used in video compression: **Perceptual Frame Interpolation Quality Metric (PFIQM)**. They considered the blocking artifacts and possible regions of quality degradation to overcome these disadvantages in other widely used metrics. Unlike existing **video quality assessment (VQA)** algorithms that rarely use motion information directly, Seshadrinathan et al. [102] proposed a motion-based VQA algorithm named MOVIE for evaluating dynamic video fidelity. Vu et al. also took into account visual perception of motion artifacts and presented spatiotemporal Most Apparent Distortion (ST-MAD) algorithm for VQA. Zhang et al. [152] proposed a visual saliency-based image quality assessment (IQA) algorithm, namely VSI. They first used visual saliency as a feature to compute the local quality map of the distorted image, then they used visual saliency as a weighting function to reflect the importance of a local region.

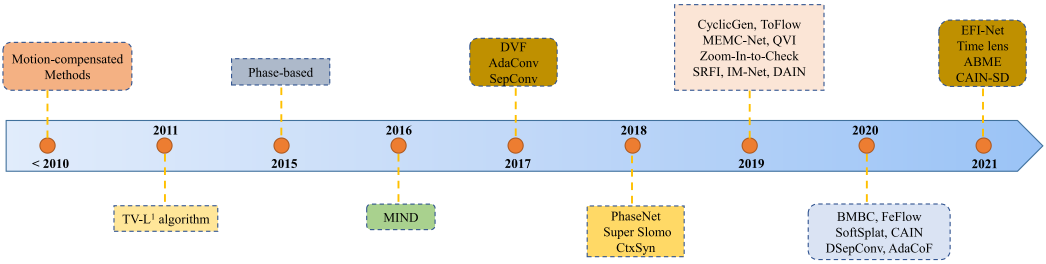


Fig. 2. The timeline of video frame interpolation algorithms development. Only representative methods are listed on this timeline.

The performance of VSI is better than SSIM; however, its computational cost is 5.6 times that of SSIM. Wang et al. proposed a multi-scale SSIM method for IQA, which provides more flexibility than single-scale approach. Men et al. [77] evaluated twelve quality assessment metrics for interpolating slow-motion videos, including four VQA methods (e.g., MOVIE, ST-MAD) and eight IQA methods (e.g., VSI, MS-SSIM, SSIM). However, in the field of VFI, PSNR, SSIM, IE, and NIE are the most frequently used.

3 THE VIDEO FRAME INTERPOLATION METHODS

In this section, we will go through the various VFI techniques that are currently available. We first introduce the brief history of VFI problem. Since optical flow estimation is a core technology of many VFI methods, we then provide a brief overview of optical flow estimation history. We will then systematically present VFI algorithms categorized into four types: flow-based methods, kernel-based methods, flow and kernel combined methods, and phased-based methods. Table 2 ∼ Table 5 summarize the advantages and disadvantages of the main methods.

3.1 Brief History

VFI is a longstanding research topic in the video processing field. Before 2010, most algorithms concentrated on motion-compensated [18, 40, 43, 44, 56, 69, 125]. Motion-compensated VFI methods generally consist of two phases: motion estimation and motion-compensated frame interpolation. These methods interpolate new frames along the motion trajectories after estimating the motion trajectories between neighboring frames. The precision of the motion trajectories and the performance of interpolation method determine the quality of the generated frames. A typical algorithm uses unidirectional motion vectors to interpolate frames, which often causes overlaps or holes. To handle these problems, some researchers pay attention to using bi-directional motion fields [18, 56]. Huang et al. [43] proposed a correlation-based motion vector processing approach to identify and rectify such unstable motion vectors in order to prevent visual errors.

With the development of optical flow estimation methods, more and more algorithms apply optical flow estimation between two consecutive input frames. Werlberger et al. [134] proposed an optical flow driven TV-L^1 denoising algorithm for video interpolation problem. The emergence of CNNs and deep learning has promoted more efficient optical flow analysis methods, such as FlowNet [30], PWC-Net [113]; many VFI methods apply them as base flow estimation algorithm to analyze the flow between input frames [4, 82, 90, 141]. With the advancement of CNNs, CNNs are also directly used to design end-to-end VFI algorithms. Long et al. [74] proposed a pioneer of the CNN-based method, Liu et al. [73] proposed an end-to-end and self-supervised fully differentiable network: Deep Voxel Flow (DVF). In the same year, AdaConv [84] was presented using the fully deep CNNs to estimate a spatially-adaptive convolution kernel for each pixel. CtxSyn [82] adopt

Table 2. Overview of Flow-Based Methods

Methods	Contribution	Limitation
DVF [73]	Applying flowing pixel values from existing video frames to interpolate intermediate frame.	Only synthesize a single frame in-between two input frames and cannot handle large motion.
Super-SloMo [50]	Could interpolate multi-frames between two input frames with an end-to-end trainable CNN.	Could not achieve real-time performance.
CtxSyn [82]	Combined input frames and contextual information to synthesize new frame at any temporal position.	Only synthesize a single frame in-between two input frames.
CyclicGen [71]	A novel cycle consistency loss was presented that can be combined with current VFI approaches in an end-to-end training manner.	Cannot handle large motion; would cause interpolation errors when videos are at very low frame rates, cannot interpolate arbitrary frame rate.
QVI [141]	Proposed a quadratic VFI model that using the acceleration information in videos.	Cannot handle large and complex motion.
DAIN [4]	Applying depth maps and contextual features as auxiliary information.	The depth maps are not always estimated well.
Zoom-In-to-Check [149]	The first instance-level adversarial learning framework.	Only synthesize a single frame in-between two input frames and cannot handle large non-rigid body movements.
PoSNet [147]	Constructed two separate models to interpolate three intermediate frames between two input frames.	Only targets 4 × interpolation.
EQVI [70]	Proposed an enhanced quadratic video interpolation method.	The performance of interpolation depends more on motion estimation accuracy.
BMBC [90]	Using bilateral motion estimation.	Sometimes, could not correctly find motions in frames with occlusion.
FeFlow [38]	Synthesized the intermediate frame through blending deep features.	Only synthesize a single frame in-between two input frames.
SoftSplat [83]	Proposed softmax splatting to forward-warp the frames.	Could not achieve real-time performance.

context maps to warp the input frame with pixel-wise contextual information for synthesizing intermediate video frame, and DAIN [4] utilized the depth information, FeFlow [38] paid attention to feature flow in-between corresponding deep features, CAIN [22] combined channel attention to interpolate high-quality frames.

On the other hand, Meyer et al. paid attention to phase information and propose phase-based VFI methods [78, 79]. We illustrate a timeline to introduce these milestones for the existing VFI methods in Figure 2, and the details of these methods will be introduced in next. In addition to these approaches, there are also some methods based on the Generative Adversarial Networks (GAN) model, event camera, and meta learning, we introduce these methods in Section 3.7

Table 3. Overview of Kernel-Based Methods

Methods	Contribution	Limitation
AdaConv [84]	A milestone method of kernel-based method using fully deep CNNs.	Cannot perfectly handle motion larger than the kernel size (41 pixels); Only synthesizes a single frame in between two input frames.
SepConv [85]	Using a pair of 1D kernels (one horizontal and one vertical) instead of 2D kernels.	Cannot perfectly handle motion larger than the kernel size (51 pixels); Only synthesizes a single frame in between two input frames.
SRFI [26]	Self-reproducing mechanism.	Cannot handle large motions.
IM-Net [92]	Modify SepConv [85] into a multi-scale architecture and formulate interpolated motion estimation as classification by calculating the center-of-mass of the convolution kernels.	Cannot perfectly handle motion larger than 25 pixels.
CAIN [22]	Employing channel attention mechanism and PixelShuffle [105] to directly interpolate frames.	Only synthesizes a single frame in between two input frames.
DSepConv [15]	Applying smaller kernel size and relevant features for resolving large-scale motions.	Only synthesizes a single frame in between two input frames.
AdaCoF [61]	Could handle a wide variety of complex motions.	Only synthesizes a single frame in between two input frames.
FLAVR [55]	FLAVR is 384x faster than QVI [141] and 23x faster than CAIN [22] due to its simplicity.	For each interpolation factor k, FLAVR still has to be retrained.
EDSC [16]	Could interpolate multiple frames.	The generated kernel size is limited to 5.
PDWN [13]	Applying the gradually refined offsets to conduct an image-level warping	The cross-scale information may not be fully utilized by the single-level alignment.

Table 4. Overview of Flow and Kernel Combined Methods

Methods	Contribution	Limitation
ToFlow [143]	Designed a model to learn the task-oriented flow with two components: motion estimation and video processing.	Only synthesizes a single frame in between two input frames.
MEMC-Net [5]	Proposed a motion estimation and compensation guided model.	Only synthesizes a single frame in between two input frames.
GDCN [106]	Could handle a wide range of motions.	The quality of the training dataset is important to the success of this method.

Table 5. Overview of Phase-Based Methods

Methods	Contribution	Limitation
Phase [79]	Combined phase information across oriented multi-scale pyramid levels.	Cannot handle large motions of high-frequency content; Uses hand-tuned parameters.
PhaseNet [78]	Combine the phase-based approach with a neural network decoder; Can handle a larger range of motion than [79].	Cannot preserve high-frequency details in videos with large temporal changes.

3.2 Optical Flow Estimetion Methods

In the field of CV, optical flow estimation is a long-standing research hotspot. In the past three decades, there has been an enormous development of technologies applied for various aspects of optical flow estimation since the pioneering method of [75] has been published in 1981.

In traditional methods, SimpleFlow [115] uses only local evidence to represent the motion flow without resorting to global optimization. SepConv [85] applies SimpleFLow to measure the average optical flow between the first and the last patch of whole frames. DeepFlow [132] is one of the top-performing handcrafted algorithms, with a convolutional framework that resembles deep learning models but no learned parameters. FlowNet [30] is a milestone method in the optical flow estimation scenario based on a U-Net architecture [98], the paper proposes two network architectures FlowNetS (FlowNet Simple) and FlowNetC (FlowNet Correlation), shows the advantages of using CNN models. To overcome the drawbacks of FlowNet and achieve better performance, Ilg et al. [46] designed a larger network FlowNet2 using FlowNetS and FlowNetC as building blocks. Moreover, the author further improve the performance of FlowNet2 by removing the small displacement network and explicit brightness error, and added residual connections in the stack, then propose FlowNet3 [47].

SpyNet [96] is the first coarse-to-fine approach, LiteFlowNet [45] is a lightweight optical flow estimation method, it is 30 times smaller and 1.36 times faster than FlowNet2. PWC-Net [113] outperforms all other optical flow estimation algorithms at the time of writing, which is a pyramidal coarse-to-fine CNNs based method. PWC-Net is used as a basis network in many VFI algorithms to predict optical flow between successive input frames, such as QVI [141], CtxSyn [82], DAIN [4], BMBC [90], and SoftSplat [83]. Recently, Teed et al. [116] presented a new end-to-end trainable network for predicting optical flow named RAFT, which uses a recurrent unit to repeatedly update a flow field and achieves a major achievement in this field. Inspired by RAFT, Li et al. [108] designed a transformer-like architecture to recurrently refine the piece-wise flow, and propose an effective framework named AnimeInterp[1] for animation frame interpolation.

The unsupervised method is another major development in this field. The main idea is to utilize various real datasets to train the optical flow model, which will be unaffected by the mismatch between training and test datasets. Meister et al. [76] presented an end-to-end unsupervised model to enable practical training of FlowNet networks on large datasets without optical flow ground truth. Jonschkowski et al. [53] studied what affects the performance of unsupervised optical flow estimation and use a systematic approach to compare, evaluate, and improve a series of important components.

There are some works about the overview on optical flow estimation field [34, 100, 120], more detail about the history of optical flow could be explored.

[1] AnimeInterp: https://github.com/lisiyao21/AnimeInterp/.

3.3 Flow-Based Methods

The most traditional and popular approach to resolve the frame interpolation problem is utilizing optical flow. Optical flow can perceive motion information in continuous frames and capture dense pixel correspondence. The optical flow directs the warping process, which converts the input frames to the interpolation frame's appropriate position and then constructively blends them. The methods based on optical flow estimation were proposed earlier than the kernel-based methods. Werlberger et al. [134] proposed an optical-flow-driven TV-L^1 denoising algorithm for the VFI problem, the approach is very reliant on the quality of the optical flow utilized. In [95], Raket et al. introduced a motion-compensated VFI approach based on interpolation along the motion vectors, which demonstrated competitive results applying a simple TV-L^1 optical flow algorithm as a prototype. Yu et al. [148] presented a multi-level VFI scheme based on block-level, pixel-level and sequence level. These traditional solutions analyze optical flow between consecutive frames and then interpolate along optical flow vectors. They work well when the optical flow is accurate; however, the situation is always challenging, and this will produce significant artifacts.

In recent years, the development of CNNs has promoted the solution of many CV problems, such as target recognition, image processing, and video processing. As one of the first to use CNN-based approaches, Long et al. [74] presented an unsupervised approach, which trains and applies CNNs for frame interpolation and then inverted CNN to predict the optical flow. However, the main disadvantage of this method is that suffering from severe blurriness. Liu et al. [73] introduced an end-to-end and self-supervised fully differentiable network, **Deep Voxel Flow (DVF)**, to interpolate no-existing frame by flowing pixel values from successive frames. However, although this method can reduce the blur of the new frame, it cannot handle large-scale motions.

These CNN-based methods focus on single-frame interpolation and are not well-suited for multi-frame interpolation [4, 17, 50, 140], which is the core technology for slow-motion applications. Jiang et al. [50] aimed to address this problem and they proposed Super-SloMo[2] to interpolate variable-length multiple frames between two input frames. They first apply a U-Net [98] network to calculate bi-directional optical flow between the input frames, and then they employ another U-Net to improve the flow and predict soft visibility maps. Dutta et al. [31] improved the performance of Super-SloMo by residual refinement. Wu et al. [137] proposed a dual refinement network including flow refinement, frame synthesis, and Haar refinement for the VFI problem, and the network includes three shallow U-Ntes, resulting in a much smaller number of parameters than many previous solutions. In addition, the DAIN [4][3] method uses the depth information to explicitly detect occlusion. They adopt PWC-Net [113] to estimate flow, U-Net [98] architecture to estimate kernels, and the hourglass architecture [12] as the depth estimation network. DAIN can interpolate multi-frames between input frames. However, the interpolation result mainly relies on the depth maps, and the approach tends to generate blurry and ambiguous boundaries when the depth maps are not accurately measured in the challenging cases.

Niklaus et al. proposed the CtxSyn [82] algorithm, and Figure 3 shows the overview of the method, which adopts context maps to warp the input frame and their pixel-wise contextual information for synthesizing intermediate frames. They used PWC-Net [113] to estimate the bi-direction flow and extract pixel-wise context maps, and then applied GridNet [35] to generate the final interpolation.

Liu et al. introduced a new loss function, *cycle consistency loss*, to enhance the interpolation results. CyclicGen [71][4] is built on the top of DVF [73] and shares the same setting ; however, this

[2] SuperSlomo: https://github.com/avinashpaliwal/Super-SloMo.

[3] DAIN: https://github.com/baowenbo/DAIN.

[4] CyclicGen: https://github.com/alex04072000/CyclicGen.

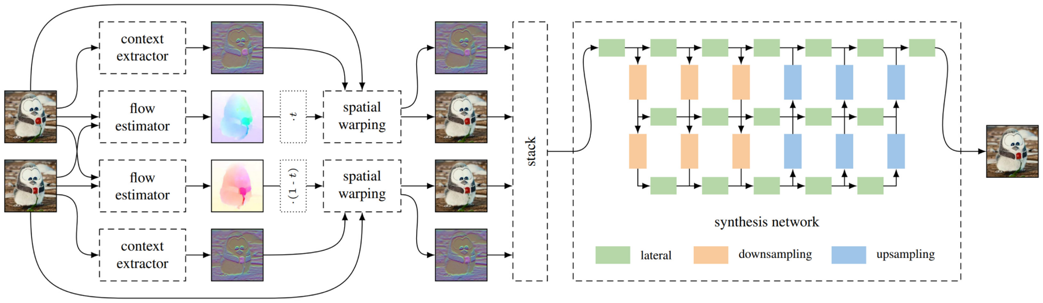


Fig. 3. The architecture of CtxSyn method [82]. The approach extracts per-pixel context mappings after estimating bidirectional flow between two input frames.

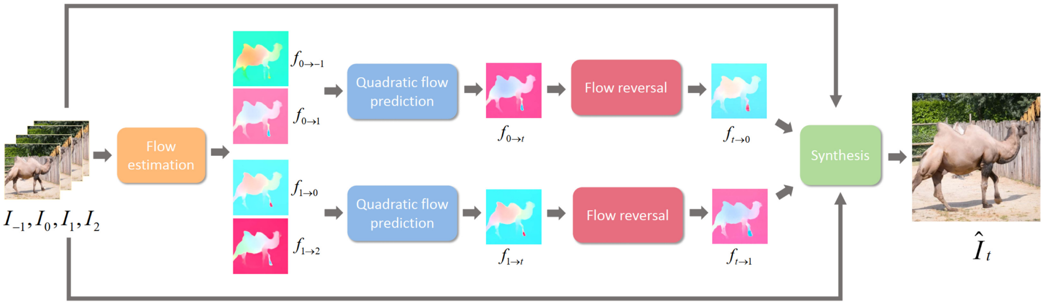


Fig. 4. The overview of QVI method [141].

method cannot handle large motions well. Reda et al. [97] attempted to learn from only low-frame rate original videos in a completely unsupervised way and use cycle consistency constraints to synthesize arbitrary high-frame rate videos. This approach is based on the Super Slo-mo model, it can predict intermediate frames at any timestamp, while CyclicGen is specifically trained to synthesize the middle frame. Moreover, Park et al. proposed a novel deep-learning-based VFI algorithm based on the bilateral motion estimation: **BMBC** [90],[5] which can interpolate a new frame at arbitrary time $t \in (0, 1)$. BMBC is made up of a bilateral motion network and a dynamic filter generation network, the former is used to estimate bilateral motions accurately, and the latter is used to warp the two input frames and feed them to learn filter coefficients. Finally, the new frame is synthesized by superimposing the warped frames with the generated blending filters.

Xu et al. presented a **quadratic video interpolation (QVI)** [141][6] algorithm, which takes advantage of the acceleration information from consecutive frames of videos. The overview of QVI approach is shown in Figure 4. First, the authors apply PWC-Net to estimate optical flow between the input frames (I_{-1}, I_0, I_1, I_2). Then the quadratic flow prediction module predicts the intermediate flow map $f_{0\longrightarrow t}$ using $f_{0\longrightarrow -1}$ and $f_{0\longrightarrow 1}$. The backward flow $f_{t\longrightarrow 0}$ is reversed by the forward flow $f_{0\longrightarrow t}$, and $f_{t\longrightarrow 1}$ can be computed similarly. Finally, the in-between frame is synthesized by warping and using the input frames with $f_{t\longrightarrow 0}$ and $f_{t\longrightarrow 1}$. However, the intermediate frames generated by this algorithm still contain some ghosts and artifacts, especially when there are large-scale and complex motions between the input frames. To address the large motion problem, Liu et al. proposed an **enhanced QVI (EQVI)** [70][7] model, and won first place

[5] BMBC: https://github.com/JunHeum/BMBC.
[6] QVI: https://sites.google.com/view/xiangyuxu/qvi_nips19.
[7] EQVI: https://github.com/lyh-18/EQVI.

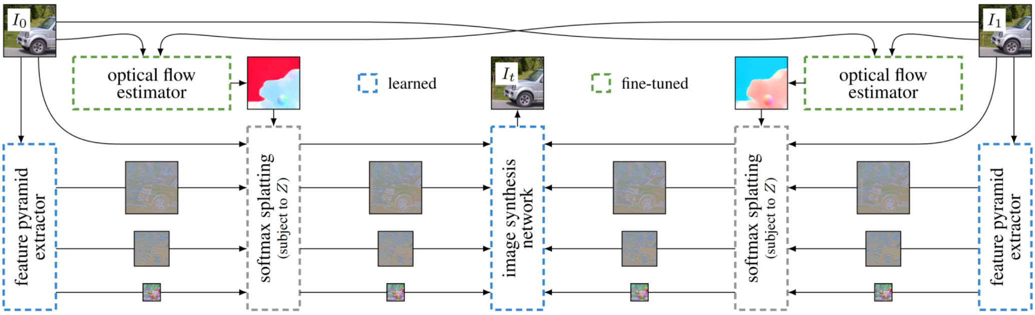

I_0
optical flow estimator
learned
I_t
fine-tuned
optical flow estimator
I_1
feature pyramid extractor
softmax splatting (subject to Z)
image synthesis network
softmax splatting (subject to Z)
feature pyramid extractor

Fig. 5. The architecture of SoftSplat method [83].

in AIM2020 **VTSR (video temporal super-resolution)** challenge [109]. They further improve the performance of QVI from three aspects: (1) They used the least square method to correct the original quadratic flow prediction module to increase the accuracy of the interpolated optical flow; (2) They proposed a **Residual Context Synthesis Network (RCSN)** to incorporate context information from pre-extracted high-dimensional features, thereby reducing the problem of inaccurate motion estimation and object occlusion; and (3) They proposed a novel and learnable enhancement multi-scale fusion network to improve performance. Zhang et al. [155] presented a general quadratic VFI method without knowing temporal priors, which combines with a restoration network to extract the temporal unambiguous sharp content from blurry frames.

On the other hand, Yan et al. [145] presented a novel **fine-grained motion estimation method (FGME)** to address the large motion problem for the VFI task, which mainly contains two strategies: (1) gradually refine the optical flow and weight map; (2) generate multiple optical flows and weight map to offer fine-grained motion characteristics. The method can handle motions of different scales, including small and large motions. Zhao et al. [156] introduced an edge-aware network integrating the edge information into the VFI issues to reduce the image blur. Gui et al. [38] paid attention to feature flow (FeFlow)[8] in-between corresponding deep features and design a structure-to-texture generation model for VFI. The interpolation process was divided into two steps: structure-guided interpolation and texture refinement. First, they devised a multi-flow multi-attention generator to estimate feature flow between two input frames. And then, they designed a frame texture compensator to synthesize the missed texture features of the new frame.

Park et al. [91][9] proposed a VFI method based on backward warping, composed of the **asymmetric bilateral motion estimation (ABME)** and the frame interpolation module. Unlike the common backward warping, the SoftSplat [83][10] model pays attention to forward warping. Figure 5 shows the architecture of SoftSplat method. Specifically, SoftSplat is based on the optical flow estimation applying softmax splitting, forward-warping the two input frames and their feature pyramid representations, and then using a synthetic network to interpolate new frames from the warped representations. However, both forward and backward warping only utilize the first frame. Xue et al. [142] introduced bilateral warping to make full use of optical flows and issue this limitation. Choi et al. [24] proposed a multi-scale warping module to interpolate frames robustly for both small and large motions.

The methods introduced above are all focused on interpolation performance. Most algorithms are trained and tested on GPUs. At present, with the increase of edge devices, the speed of the

[8] FeFlow: https://github.com/CM-BF/FeatureFlow.
[9] ABME: https://github.com/JunHeum/ABME.
[10] SoftSplat: https://github.com/sniklaus/softmax-splatting.

VFI algorithm has also attracted more attention, some lightweight algorithms were brought out. Yuan et al. [149] proposed a lightweight VFI framework Zoom-In-to-Check, Li et al. [65] introduced a lightweight model FI-Net, which accepts two frames in arbitrary size as inputs. FI-Net computes optical flow at feature level instead of image-level and has a small model size. Yu et al. proposed a 4x VFI model that aims to convert 15 fps to 60 fps videos using position-specific flow (PoSNet [147]).[11] To enhance the quality of the intermediate frame, they employed two input frames and flow maps as additional information in the model.

3.4 Kernel-Based Methods

The interpolation results of the optical flow-based methods are affected by the optical flow estimation quality, which is easily influenced by abrupt brightness change and illumination change. In the field of VFI, another commonly used method is based on kernel.

As a milestone method in the VFI field, AdaConv [84] pioneers the use of fully deep CNNs to estimate a spatially-adaptive convolution kernel for each pixel. AdaConv combines the two-step interpolation method (motion estimation and pixel synthesis) into one convolution process with two input frames. However, the performance of AdaConv is restricted by the convolution kernel size and requires large memory, the authors design the kernel size is 41 × 41, which will make the interpolate frame blurred if the large motion beyond 41 pixels. To solve this issue, Niklaus et al. then proposed SepConv [85][12] using pairs of 1D kernels (one horizontal and one vertical) instead of 2D kernels, which require 1.27 GB instead of 26 GB of memory for a 1080p video frame. Furthermore, this approach applies a deep fully CNN and could be trained end-to-end using publicly accessible benchmark datasets. However, there are still two disadvantages of SepConv. First, the amount of motion that SepConv can handle is limited by kernel size, which is 51 pixels, even larger than the AdaConv method, if two input frames have large motion than 51 pixels, the method still produces ghosting artifacts. And it is likewise wasteful to estimate tiny motions smaller than the kernel size. Secondly, SepConv and AdaConv interpolate a frame in the middle temporal position of the two input frames and cannot synthesize a frame at arbitrary temporal position. In [86], Niklaus et al. revisited the adaptive separable convolutions and optimize the individual parts of SepConv by a set of intuitive improvements, such as delayed padding, input and kernel normalization, contextual training. The improvements allow the proposed SepConv++ method ranks fourth among all available approaches on the Middlebury dataset.

Deng et al. [26] presented a novel self-reproducing mechanism called **Self-Reproducing Frame Interpolation (SRFI)** to effectively enhance the consistency and performance of VFI, which can be combined with and improved on state-of-the-art algorithms. Inspired by deformable convolution, Chen et al. [13] designed a **Pyramid Deformable Warping Network (PDWN) u**sing a pyramid network to synthesize unknown middle frame.

Peleg et al. [92] proposed an **interpolated motion neural network (IM-Net)**, aimed at high resolutions and real-time reference time. IM-Net consists of three modules: Feature Extraction, Encoder-Decoder, and Estimation. Feature Extraction module processes each of the six input frames and sends the extracted features to the next module, and then the Encoder-Decoder module passes three merged outputs to three parallel Estimation paths. The authors compared the performance and reference time with ToFlow [143] and SepConv [85] on low- and high resolution benchmark datasets. The three approaches are equivalent in quality at low resolution, but when using high-resolution videos (1376 × 768), IM-Net performs the best among the three algorithms. And on a single GPU (Nvidia Titan X), IM-Net is 16 times faster, and run time is only 30 msec

[11] PoSNet: https://github.com/SonghyunYu/PoSNet.

[12] SepConv: https://github.com/sniklaus/sepconv-slomo.

for HD resolution. Ahn et al. [1] also paid attention to high-resolution VFI. They present hybrid task-based CNNs for a fast and accurate 4K VFI task. The method consists of a temporal interpolation network that interpolates intermediate frames, and a spatial interpolation network that reconstructs the original-scale frames from the synthesized frames. Compared to SepConv and SuperSlomo, on Titan X (Pascal) GPU, the proposed algorithm only runs 620 ms to interpolate a 4K frame, which is 2.69x faster than SepConv, and also outperformed the two methods in the PSNR and SSIM values.

Cheng et al. [15] proposed DSepConv, inspired by the success of deformable convolution networks [25, 162] and aimed to solve large motion problem. DSepConv can handle large motion using small kernel size by learning deformable offsets, masks, and spatially-adaptive separable convolution kernels. Like other kernel-based methods, the limitation of DSepConv is that it can only interpolate a single frame. The authors proposed a novel method called Enhanced Deformable Separable Convolution (EDSC) [16][13] to resolve this issue. This model is not constrained by the kernel size and is able to handle large motion. The authors designed different estimators that involved temporal information as a control. EDSC could directly interpolate a frame at any arbitrary temporal position without using a recursive method. Another VFI method inspired by deformable convolution network is Adaptive Collaboration of Flows (AdaCoF) [61],[14] which refers to any number of pixels and any position in the consecutive frames. This approach has more Degrees of Freedom (DoF) because the sizes and shapes of the kernels are arbitary.

To solve optical flow drawbacks, Choi et al. proposed a novel method (CAIN) [22][15] that replaces optical flow with simple feature map transformations, referred to as PixelShuffle [105], and combines channel attention [154] to synthesize high-quality frames without explicit estimation motion. They also constructed a more comprehensive benchmark dataset called SNU-FILM to evaluate the available VFI methods on challenging motion and occlusion, and the CAIN approach achieves outstanding performance compared to DAIN [4], CyclicGen [71] on HD resolution (1280 × 720) frames. Based on CAIN method, Choi et al. [23] presented a motion-aware dynamic architecture to calculate amounts of computation for different regions of frame. The static regions pass through a smaller number of layers, while the regions with larger motion are downscaled for better motion reasoning. This method can significantly reduce the computation cost (FLOPs). Kalluri et al. proposed an optical flow-free VFI method called FLAVR [55],[16] which is able to interpolate multi-intermediate frames between two input frames. FLAVR is an efficient 3D CNN architecture that replaces all the 2D convolutions with 3D convolutions in the encoder and decoder to accurately model the temporal dynamics between the input frames, resulting in better interpolation quality even when no external inputs such as optical flow or depth maps. FLAVR is 384x faster than QVI [141] and 23x faster than CAIN [22] due to its simplicity.

3.5 Flow and Kernel Combined Methods

To compensate for each other's limitations, the approaches of integrating kernel-based and flow-based mechanism have recently been presented. They multiply the kernels with the flow vector's indicated position. As a result, they can refer to any position as well as certain nearby pixels.

Xue et al. proposed a self-supervised method called **ToFlow (task-oriented flow)** [143][17] to learn motion representation. On standard benchmark datasets, ToFlow outperforms traditional

[13] EDSC: https://github.com/Xianhang/EDSC-pytorch.

[14] AdaCoF: https://github.com/HyeongminLEE/AdaCoF-pytorch.

[15] CAIN: https://github.com/myungsub/CAIN.

[16] FLAVR: https://tarun005.github.io/FLAVR/.

[17] ToFlow: https://github.com/anchen1011/toflow.

optical flow methods in the VFI task. Bao et al. presented a **motion-estimation- and motion-compensation-guided neural network (MEMC-Net)** [5][18] for VFI task. Shi et al. proposed GDCN [106][19] method based on generalized deformable convolution mechanism. GDCN is capable of handling a wide range of motions because it effectively learns motion information in a data-driven manner and freely selects sampling points in space-time.

3.6 Phase-Based Methods

The other research direction uses phase information to interpolate in between frames to handle the challenging scenarios over optical-flow-based methods containing motion blur or lighting changes. As the Fourier theory states, the image can be expressed as a sum of a series of sinusoidal functions. For example, assuming we have two sinusoidal functions, which are defined as $y_1 = \sin(x)$ and $y_2 = \sin(x - \pi/3)$, as shown in Figure 1(d), y_1, y_2 are the same sinusoidal function and y_1 translates by $\pi/3$ to get y_2. The translation, i.e., the motion, can be represented by the phase difference of $\pi/3$. The phase-based methods indicate motion as the phase difference between individual pixels. In Figure 1(d), the two sinusoidal curves would correspond to the two input frames. An in-between curve would then represent the synthesized intermediate frame. But due to the 2π-ambiguity of phase values (i.e., $y = \sin(x-\pi/3) = \sin(x-\pi/3+2\pi)$, there exists two valid solutions, namely $y_3 = \sin(x-\pi/6)$ and $y_4 = \sin(x-\pi/6+\pi)$. The challenge with phase-based VFI methods is determining which option is the correct solution. Meyer et al. [79] introduced a phase-based VFI algorithm that combines phase information across oriented multi-scale pyramid levels using a novel bounded shift correction strategy. Such a method performs well and runs faster than any other flow-based methods at the time. However, this phase-based method cannot represent a large motion of high-frequency content, even still blurring in areas with small motion. To improve the performance, the authors proposed another new phase-based approach, PhaseNet [78]. It comprises of a neural network decoder that directly calculates the phase information of the input frames. The phase and amplitude values of the intermediate frame are predicted level by level. At different levels, the final frame is reconstructed from these predictions. PhaseNet can process larger motion than [79] and reduce parameters because of sharing weights across channels and pyramid levels. This method is suitable for scenarios with brightness changes and motion blur. However, it still does not achieve the same performence as methods that explicitly match and warp pixels.

3.7 Others

The flow-based and kernel-based methods mentioned above are often used in this VFI field. In addition to these types of methods, there are other methods using flow-based or kernel-based network as core architecture and combining other information to solve VFI problems. Blurry VFI [103][20] approach intends to tackle the problem of joint video enhancement by reducing blur and up-converting frame rate. Kwon et al. [59] presented direct VFI method with multiple latent encoders for 360-degree videos. Choi et al. [20, 21][21] first employed a meta-learning algorithm to any VFI network and improve their performance.

In [13], [104], [150], and [151], the authors paid attention to the pyramid network for VFI. MPRN [150] introduces a unique coarse-to-fine pyramid framework to effectively excavate the motion and occlusion information between several frames. Shen et al. [104] presented a blurry VFI pyramid module to jointly reduce blur and up-convert the frame rate, and an inter-pyramid

[18] MEMC-Net: https://github.com/baowenbo/MEMC-Net.

[19] GDCN: https://github.com/zhshi0816/GDConvNet.

[20] BIN: https://github.com/laomao0/BIN.

[21] SAVFI: https://github.com/myungsub/meta-interpolation.

Table 6. Overview of other VFI Methods

Methods	Contribution	Limitation
FIGAN [123]	Proposed a multi-scale GAN to predict flow and interpolate frame.	Interpolation results are based on flow quality.
MPRN [150]	Using spatio-temporal information contained in multiple frames to generate frames.	Interpolation results are based on optical flow estimation results.
MS-PFT [14]	Proposed a position feature transform layer to reduce the sensitivity of optical flow.	Interpolation results are heavily dependent on optical flow estimation results.
RRPN [151]	Proposed a recurrent residual pyramid network that can handle large motion for VFI problem.	Computational efficiency is high on low-resolution images (640 × 480).
BIN [103]	Proposed a method to reduce motion blur and up-convert frame rate simultaneously.	The considered motion blurs are still small.
SAVFI [20]	The first VFI method integrated with meta-learning thchnology.	The performance is mainly based on the baseline interpolation network.
MAF-net [42]	Proposed a method that focus on the interpolation video frame involving small and fast-moving objects.	Interpolation results are heavily dependent on optical flow estimation results.
STAR-Net [41]	Designed a Flow Refinement (FR) module to improve the flow estimation result.	Interpolation results are dependent on flow images.
Time Lens [121]	Event-based approach that is robust to motion blur and non-linear motions.	It has to be fine-tuned with high-cost real event data to gain higher performance.
EFI-Net [88]	Combined data from a traditional frame camera with an event camera to decrease aliasing and severe motion artifacts.	The train data lacks real-world components (e.g., occlusions, multiple depths, varying motion types, etc.).

recurrent module to enforce temporal consistency among the produced frames. Zhang et al. [151] proposed a flexible recurrent residual pyramid network for VFI, see Table 6.

Cheng et al. [14] proposed a **Multi-Scale Position Feature Transform (MS-PFT)** network for VFI, The author designed two parallel prediction networks and an optimization network, the former predicts the deep features of interpolation frame individually using two input frames, and then the latter interpolates the final frame using the outputs of the two prediction network. They also applied a pre-trained neural network to extract contextual information from the input frames and feed them into the network [82] to improve the quality of the extracted features in the prediction process.

Many flow-based VFI methods apply PWC-Net as an optical flow estimation network. PWC-Net adopts the multi-scale pyramid to estimate the optical flow. The higher pyramid level can capture large-scale motion, but fail on the fast-moving small object. Because the small object would be easily lost during the downsampling process. As a result, halos or ghosts would appear around the small objects in generated images. To address this issue, Hu et al. [42] proposed a **Motion Feedback network (MAF-net)** to improve the estimation of large motions for small objects. Haris et al.

[41] proposed a space-time-aware multiple resolutions for video enhancement task; they designed a **Flow Refinement (FR)** module to improve the flow estimation result.

In recent years, Transformer [124] has been successfully applied to numerous **Natural Language Processing (NLP)** tasks, some researchers applied it in the field of CV, such as image classification [11, 29], object detection [9, 158], segmentation [126, 128], and Transformer-based VFI algorithm is also a potential research direction [72, 101]. Liu et al. [72] proposed ConvTransformer, an end-to-end VFI approach that simplifies video frame synthesis as an encoder and decoder problem. ConvTransformer collects the high-order motion information present in video sequences and uses it to synthesize the target interpolated frames using the multi-head convolutional self-attention mechanism. Schatz et al. [101] concentrated on human actions from novel views and proposed a recurrent transformer network. They employed the help of an appearance prior to synthesize a video with the same action performed from a novel view.

Generative Adversarial Networks (GAN). In recent years, GAN have shown their effiency in numerous image application. GAN network is also useful in the field of VFI. Koren et al. [58] proposed an end-to-end neural architecture for VFI. Amersfoort et al. [123] presented a multi-scale GAN for VFI includes a multi-scale residual module to predict flow and synthesise frame. Inspired by the VFI scheme, Santurkar et al. [99] proposed a GAN-based VFI method to enhance the video compression performance. Wen et al. [133] designed two concatenated GANs to improve interpolation quality, one capturing motions and the other generating frame details. Xue et al. [144] presented a Frame-GAN method to generate more gait frames for enhancing the recognition performance. Tran et al. [119] introduced a lightweight GAN model for VFI. In their work, there are two generators, the first generator generate the interpolated frame via two input frames, and then the interpolated frame is improved by the second generator with the refinement network.

Event-based: An event camera is a novel imaging sensor that responds to local brightness changes. Event cameras do not use shutters to capture images like traditional cameras. On the contrary, each pixel within the event camera works independently and asynchronously, reporting brightness changes when they happen, and silent otherwise. Lin et al. [68] proposed an event-driven video blurring and interpolation method based on deep CNN to solve high frame-rate video generation. There are four modules in their algorithm: residual estimation network, keyframe deblurring, frame interpolation, and frame fusion. Tulyakov et al. [121][22] presented Time Lens, a novel event-based VFI approach that leverages the advantages of flow-based and kernel-based approaches. As a result, it is robust to motion blur and non-linear motions. Paikin et al. [88] introduced an Event Frame Interpolation network (EFI-Net) that combines data from a triditional frame camera with an event camera to decrease aliasing and severe motion artifacts.

4 COMPARISON

In the section, we compare the performance of the aforementioned VFI methods according to the metrics introduced in Section 2.3. In Table 7, we provide the quantitative comparison results of various VFI methods, we compare the PSNR, SSIM values on UCF 101, Vimeo90K, Middlebury, and Adobe240 datasets, and also compare IE value on Middlebury dataset. On each dataset, the first, second, and third best approaches are marked in red, green, and blue text. On the UCF101 dataset, the SoftSplat method gets the best PSNR value, the GDCN method gets the second PSNR value and the third SSIM value, the BMBC method gets the third PSNR value and the second SSIM value, the FLAVR method gets the best SSIM value, the CAIN, DSepConv, AdaCof, and EDSC methods also get better SSIM value. On the Vimeo90K dataset, the FLAVR method gets the best PSNR value and the second SSIM value, the PDWN method gets the second PSNR value, the FeFlow method

[22] Time Lens: https://rpg.ifi.uzh.ch/TimeLens.html.

Table 7. Quantitative Comparison of Various VFI Methods on UCF101, Vimeo90K, Adobe240, and Middlebury Datasets

Method	**UCF101**		**Vimeo90K**		**Adobe240**		**Middlebury**		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	IE
Flow-based:									
DVF [73]	34.12	0.946	31.54	0.921	22.33	0.616	34.34	0.971	4.04
SuperSlomo [50]	33.14	0.938	34.75	0.968	31.94	0.926	34.23	0.977	2.28
CtxSyn [82]	34.01	0.941	33.76	0.955	/	/	35.95	0.959	/
CyclicGen [71]	35.11	0.950	32.10	0.923	/	/	33.46	0.931	2.86
QVI [141]	/	/	/	/	33.06	0.939	/	/	/
DAIN [4]	35.00	0.950	34.70	0.964	32.51	0.954	36.70	0.965	2.04
Zoom-In-to-Check [149]	30.08	0.876	/	/	/	/	/	/	/
PoSNet [147]	34.21	0.948	34.32	0.953	/	/	/	/	/
BMBC [90]	35.15	0.969	35.01	0.976	/	/	36.79	0.965	/
FeFlow [38]	/	/	35.28	0.976	32.66	0.955	/	/	/
SoftSplat [83]	35.54	0.967	/	/	/	/	37.55	0.965	/
Kernel-based:									
AdaCoF [61]	34.91	0.968	34.27	0.971	33.17	0.931	35.72	0.978	2.31
SepConv [85]	34.69	0.945	33.45	0.951	33.41	0.935	35.73	0.959	2.44
CAIN [22]	34.91	0.969	34.65	0.973	/	/	35.11	0.951	2.28
DSepConv [15]	35.08	0.969	34.73	0.974	/	/	/	/	2.06
EDSC [16]	35.13	0.968	34.84	0.975	/	/	/	/	**2.02**
PDWN [13]	35.00	0.950	35.44	0.966	/	/	37.20	0.967	1.98
FLAVR [55]	33.33	0.971	36.30	0.975	32.20	0.957	/	/	/
Flow and Kernel Combined:									
MEMC-Net [5]	34.96	0.948	34.29	0.956	32.42	0.954	/	/	2.10
ToFlow [143]	34.58	0.967	33.73	0.968	33.73	0.987	35.29	0.956	2.15
GDCN [106]	35.16	0.968	34.99	0.975	34.53	0.945	/	/	2.03
Phase-based:									
Phase [79]	/	/	30.52	0.885	31.20	0.893	31.12	0.933	/
Others:									
MIND [74]	33.93	0.966	33.50	0.942	/	/	31.35	0.943	3.35
FGME [145]	35.12	0.950	34.91	0.964	/	/	36.73	0.966	/
BIN [103]	/	/	/	/	32.51	0.928	/	/	/
MS-PFT [14]	34.70	0.967	34.26	0.971	/	/	/	/	5.00
STAR-Net [41]	35.07	0.967	35.11	0.976	/	/	27.12	0.827	1.95

On each dataset, the first, second, and third best methods are marked in red, green, and blue text.

gets the third PSNR value and the first SSIM value, the BMBC method also gets the first SSIM value, the GDCN and EDSC methods also get the second SSIM values, the DSepConv method gets the third SSIM value. On the Adobe240 dataset, the GDCN and SepConv methods get the best and third PSNR values, the ToFlow method gets the second PSNR value and the best SSIM value, and the FLAVR and FeFlow method get the second and third SSIM values, respectively. On the Middlebury dataset, the STAR-Net, PDWN, and EDSC methods get the best, second, and third IE values, respectively. To sum up, the GDCN algorithm has achieved good results on these four datasets, and the FLAVR algorithm has also achieved good results on the first three datasets.

Table 8. The IE Evaluations on Middlebury Benchmark

Methods	Mequon			Schefflera			Urban			Teddy			Backyard			Basketball			Dumptruck			Evergreen		
	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.
Flow-based:																								
SoftSplat [83]	2.06	3.06	1.14	2.80	3.91	1.24	1.99	2.73	1.21	3.84	4.64	2.69	8.10	10.00	2.96	4.10	7.53	1.98	5.49	12.1	1.39	5.40	8.33	1.50
BMBC [90]	2.30	3.40	1.20	3.07	4.25	1.41	3.17	4.19	1.66	4.24	5.28	2.85	7.79	9.62	3.14	4.08	7.47	2.02	5.63	12.4	1.40	5.55	8.58	1.61
CtxSyn [82]	2.24	3.72	1.04	2.96	4.16	1.35	4.32	3.42	3.18	4.21	5.46	3.00	9.59	11.9	3.46	5.22	9.76	2.22	7.02	15.4	1.58	6.66	10.2	1.69
FeFlow [38]	2.28	3.73	1.18	3.50	4.78	2.09	2.82	3.13	1.66	4.75	5.78	3.72	7.62	9.40	3.04	4.74	8.88	2.03	6.07	13.1	1.59	6.78	10.5	1.65
CyclicGen [71]	2.26	3.32	1.42	3.19	4.01	2.21	2.76	4.05	1.62	4.97	5.92	3.79	8.00	9.84	3.13	3.36	5.65	2.17	4.55	9.68	1.42	4.48	6.84	1.52
DAIN [4]	2.38	4.05	1.26	3.28	4.53	1.79	3.32	3.77	2.05	4.65	5.88	3.41	7.88	9.74	3.04	4.73	8.90	2.04	6.36	14.3	1.51	6.25	9.68	1.54
SuperSlomo [50]	2.51	4.32	1.25	3.66	5.06	1.93	2.91	4.00	1.41	5.05	6.27	3.66	9.56	11.9	3.30	5.37	10.20	2.24	6.69	15.00	1.53	6.73	10.4	1.66
Kernel-based:																								
SepConv++ [86]	2.39	4.17	1.20	2.98	4.21	1.28	3.34	3.23	2.20	4.49	5.81	2.87	7.64	9.42	2.97	3.77	6.80	1.96	5.26	11.6	1.36	5.71	8.86	1.45
EDSC [16]	2.32	3.90	1.16	3.10	4.38	1.51	2.98	3.54	1.36	4.49	5.74	3.16	8.05	9.96	3.08	4.89	9.28	2.02	5.55	12.3	1,41	6.42	9.99	1.55
AdaCoF [61]	2.41	4.10	1.26	3.10	4.32	1.43	3.48	3.31	1.78	4.84	5.94	2.93	8.68	10.8	3.14	4.13	7.59	1.97	5.77	12.90	1.37	5.60	8.67	1.48
DSepConv [15]	2.47	4.39	1.21	3.32	4.60	1.72	3.28	3.66	1.50	5.11	6.36	3.23	7.85	9.69	3.11	4.68	8.78	2.04	5.65	12.5	1.44	6.54	10.2	1.58
SepConv [85]	2.52	4.83	1.11	3.56	5.04	1.90	4.17	4.15	2.86	5.41	6.81	3.88	10.2	12.8	3.37	5.47	10.4	2.21	6.88	15.6	1.72	6.63	10.3	1.62
AdaConv [84]	3.57	6.88	1.41	4.34	5.67	2.52	5.00	5.86	2.98	6.91	8.89	4.89	10.2	12.8	3.21	5.33	10.1	2.27	7.30	16.6	1.92	6.94	10.8	1.67
FLAVR [55]	3.02	4.65	1.34	3.70	4.49	1.71	3.52	4.19	1.68	8.08	9.60	3.65	7.35	9.04	2.94	4.20	7.73	1.96	6.17	13.0	1.62	5.53	8.40	1.57
Flow and Kernel Combined:																								
MEMC-Net [5]	2.39	3.92	1.28	3.36	4.52	2.07	3.37	3.86	2.20	4.84	5.93	3.72	8.55	10.6	3.14	4.70	8.81	2.03	6.40	14.2	1.58	6.37	9.87	1.57
ToFlow [143]	2.54	4.35	1.16	3.70	5.19	1.88	3.43	3.89	1.93	5.05	6.43	3.39	9.84	12.3	3.42	5.34	10.0	2.28	6.88	15.2	1.61	7.14	11.0	1.69
GDCN [106]	2.31	3.98	1.10	3.80	5.17	1.54	2.92	3.78	1.43	5.59	6.01	3.24	9.02	11.3	3.10	4.66	8.75	2.08	5.75	12.7	1.42	6.40	9.98	1.53
Others:																								
FGME [145]	2.08	3.34	0.98	3.32	4.43	1.63	2.46	3.28	1.41	4.08	4.85	3.05	7.36	9.08	3.03	4.17	7.62	2.06	4.95	10.7	1.44	5.45	8.41	1.57
STAR-Net [41]	2.18	3.37	1.21	3.46	4.88	1.47	3.04	3.53	1.58	4.41	5.44	2.76	7.51	9.27	2.98	4.65	8.72	1.99	6.21	13.4	1.41	6.17	9.45	1.49
MAF-Net [42]	2.23	3.84	1.08	3.53	4.85	1.78	2.83	3.70	1.58	4.83	5.88	3.31	9.44	11.8	3.27	5.27	10.0	2.15	6.30	14.2	1.54	6.38	9.90	1.63
MPRN [150]	2.53	4.43	1.21	3.78	4.97	1.57	3.39	5.49	1.28	5.03	6.58	3.19	9.53	11.9	3.31	5.25	9.92	2.22	6.87	15.5	1.49	6.72	10.4	1.60
MS-PFT [14]	2.53	4.35	1.16	3.61	5.03	1.69	3.30	4.25	1.77	5.13	6.55	3.19	7.94	9.81	3.21	4.49	8.24	2.22	6.55	13.9	1.79	6.42	9.89	1.69
MPRN [150]	2.53	4.43	1.21	3.78	4.97	1.57	3.39	5.49	1.28	5.03	6.58	3.19	9.53	11.9	3.31	5.25	9.92	2.22	6.87	15.5	1.49	6.72	10.4	1.60

The first, second, and third best methods are marked in red, green, and blue text. *disc.*: regions with discontinuous motion, and *unt.*: textureless regions.

Table 8 describes the detailed IE evaluation of different VFI approaches on the Middlebury benchmark, we get the latest results' indexes from the website.[23] At the writing time of this survey, the SoftSplat [83] method won the first place in the list. As we can see in the table, SoftSplat performs the best of the former four sequences, and performs poorly in the last four sequences. The last four sequences contain real images and have more complex motion blur, which means SoftSplat cannot handle complex motion. The algorithm CyclicGen [71] gets the best records for the last three sequences, which indicates that CyclicGen performs exceptionally well on real-scene sequences. FLAVR approach aims at fast frame interpolation and performs best on the *Backyard* sequence. The phase-based methods did not squeeze into the leader board.

Table 9 shows the comparison of the inference speed of these VFI methods. As we all know, the inference speed of the frame interpolation method relies heavily on computation resources. At present, all the algorithms that have been proposed are trained and tested on GPU devices, generally Nvidia graphic cards. With the improvement of the graphics card and the development of neural networks, the algorithm proposed in the past two years can achieve real-time frame interpolation on low-resolution videos (640 × 480), but it is still unable for high-resolution videos (1920 × 1080) to attain real-time performance. There is still a long way to go in terms of achieving real-time performance for embedded devices, which is also the direction of future research in the VFI field.

In this section, we presented a detailed analysis of a number of models for the VFI task. We compared the performances of these methods based on PSNR, SSIM values on UCF101, Vimeo-90k, Middlebury, Adobe240 datasets and IE value on the Middlebury dataset. As shown in Table 7, there is no perfect algorithm that achieves optimal results on all datasets. In general, GDCN and FLAVR methods can achieve better results. As shown in Table 8, the SoftSplat method obtain the lowest IE values in most sequences. However, the videos in the Middlebury dataset only have

[23] https://vision.middlebury.edu/flow/eval/results/results-i1.php.

Table 9. The Comparison of Running Time on Different VFI Methods

Methods	**Device**	**Runtime (Resolution)**
Flow-based:		
CtxSyn [82]	Nvidia Titan X (Pascal)	0.77 sec (1920 × 1080); 0.36 sec (1280 × 720)
DAIN [4]	Nvidia Titan X (Pascal)	0.125 sec (640 × 480)
Zoom-In-to-Check [149]	/	0.36 sec (2048 × 1024)
SoftSplat [83]	Nvidia Titan X	0.357 sec (1280 × 720); 0.807 sec (1920 × 1080)
PoSNet [147]	Nvidia Titan V	0.06 sec
Kernel-based:		
AdaConv [84]	Nvidia Titan X	2.8 sec (640 × 480); 9.1 sec (1280 ×720); 21.6 sec (1920 × 1080)
SepConv [85]	Nvidia Titan X (Pascal)	0.5 sec (1280 × 720); 0.9 sec (1920 × 1080)
IM-Net [92]	Nvidia Titan X	0.030 sec (1280 × 720); 0.055 sec (1920 × 1080)
CAIN [22]	Titan Xp	0.064 sec (1280 × 720)
AdaCoF [61]	RTX 2080 Ti	0.21 sec (1280 × 720)
EDSC [16]	Nvidia Titan X	0.120 sec (448 × 256); 0.557 sec (640 × 480); 1.707 seconds (1280 × 720)
PDWN [13]	NVIDIA RTX 8000	0.0086 sec (640 × 480)
Flow and Kernel Combined:		
ToFlow [143]	Nvidia Titan X	0.2 sec (256 × 448)
MEMC-Net [5]	NVIDIA Titan X (Pascal)	0.06 sec (640 × 480); 0.20 sec (1280 × 720); 0.41sec (1920 × 1080)
Phase-based:		
PhaseNet [78]	Nvidia Titan X (Pascal)	1.5 sec (2048 × 1024)
Others:		
MPRN [150]	Nvidia Tesla V100	0.15 sec (640 × 480)
FGME [145]	Nvidia GTX 1080 Ti	0.095 sec (640 × 480)
MS-PFT [14]	Nvidia GTX 1080	0.44 sec (640 × 480)
RRPN [151]	Tesla V100	0.12 sec (640 × 480)

640 × 480 resolution, and in recent years, the demand of HD videos is growing rapidly, more and more researches focus on the VFI algorithm of high-definition video, such as GDCN, FLAVR. The running time is also a key factor in VFI algorithms, which often trade off frame interpolation quality and speed. Compared with the EDSC and SoftSplat methods, they all run on Nvidia Titan X device, the SoftSplat method is nearly five times to EDSC method for 1280 × 720 resolution videos. Fortunately, with the advancement of hardware technology, we can focus more research efforts on algorithm performance. Sometimes, the visual comparison is also needed because the quantitative comparison cannot show the effect of frame interpolation very intuitively, visual comparison can clearly expose defects caused by frame insertion such as shadows, ghosts, or distortions.

5 APPLICATIONS

VFI technology has garnered considerable attention in the CV community with the digital video industry's prosperity and has become a new upsurge. The technology enjoys real-world video processing applications, such as slow-motion generation, frame rate up-conversion, video compression, novel view synthesis, video restoration, and intra-prediction in video coding. We'll go through some of the more exciting applications presented so far in the following subsections.

5.1 Slow-Motion Generation

Many unforgettable scenes exist in daily life that we would like to catch with a slow-motion camera because they are impossible to view clearly with our eyes: blooming fireworks, fish leaping from the water, galloping horses on the grassland. The VFI methods are also efficient to create high-quality slow-motion videos from low-resolution videos [4, 50, 139]. Jiang et al. proposed Super Slomo [50] algorithm to interpolate multi intermediate frames between two input frames. They used more than 1.1K 240 fps video clips to train the network to generate seven intermediate frames. Bao et al. [4] proposed a depth-aware VFI method, which combined the depth maps and contextual features to produce the new frames.

5.2 Frame Rate Up-conversion

Frame rate up-conversion (FRUC) is one of the most challenging issues in recent decades, aims to improve the visual quality. Before 2010, the motion-compensated frame interpolation methods were widely used to interpolate intermediate frames [10, 18, 56]. Jeon et al. [48] introduced a coarse-to-fine method using pyramid structure. Bao et al. [6] presented a novel algorithm based on a high-order model and dynamic filtering, using pixel-wise intensity variation and motion trajectory to convert frame rate.

5.3 Video Compression

Nowadays, nearly 75% traffic of Internet is in the form of video, and it is a significant challenge for transmission and storage as a result of its explosive growth, which needs strong video compression technique. In [135], Wu et al. presented an end-to-end deep video codec, relying on the repeated image interpolation. In [7], Begaint et al. introduce a deep VFI network for video compression to handle complex non-translational motions. Ding et al. [28] presented a learning dual-dream fusion CNN for the detection of Deep VFI in compression domain. Intra- and Inter-prediction also play a crucial role in achieving high efficiency in video compression. Choi and Bajic [19] and Zhao et al. [157], are inspired by the work of SepConv [85]. Zhao et al. [157] proposed a high-efficient inter-prediction method based on the virtual reference frame, and SepConv network and weights are used directly to generate the virtual reference frame. However, it cannot predict an ubi-directional frame using the previous frame for low-delay coding scenarios. Choi and Bajic [19] presented a novel frame prediction approach applying deep CNNs for video coding, which supports both uni- and bi-directional prediction and can handle reference frames varying distances from the predicted frame.

5.4 Video Restoration

Video restoration research, including super-resolution [5, 57, 117] and frame deblurring [2, 68, 104, 127], is drawing increasing attention in the CV field. In [134], Werlberger et al. presented a variant of TV-L^1 denoising algorithm for video restoration and interpolation. Tian et al. [117] designed the **Temporally-Deformable Alignment Network (TDAN)** to overcome the limitation of optical flow based methods for video super-resolution task. Wang et al. [127] introduced a novel video restoration framework based on the Enhanced Deformable convolutional networks to handle the large motions and blur challenges. Kim et al. [57] proposed a joint VFI and super-resolution framework for upscaling the spatio-temporal resolution of videos from 2K resolution at 30 fps to 4K resolution at 60 fps. Bao et al. [5] proposed MEMC-Net for super-resolution and denoising tasks. Tran et al. [118] focused on face video deblurring and proposed FineNet to deblur face videos. Some researchers focus on resolving the joint **video super resolution (VSR)** and VFI problem. Zhou et al. [159] presented a weights-shared interaction structure between VSR and VFI modules. This approach effectively improve their performance. Dutta et al. [32] explored a lightweight method; they first applied quadratic modeling to interpolate in low-resolution space. Then, the low-resolution frames are inputed a state-of-the-art VSR method for super-resolving.

Shen et al. proposed a blurry VFI method [103] that includes a pyramid network and an inter-pyramid recurrent module to reduce the blurry motion and up-convert rate at the same time. On the contrary, Brooks et al. [8] introduced a method to synthesize motion-blurred images from two input unblurred images. They applied VFI techniques to generate a large-scale synthetic dataset for training the motion blur interpolation network. In [104], Shen et al. concentrated on synthesizing high-frame-rate and high-quality frames from low-frame-rate blurry frames. Specifically, they designed a VFI method with a pyramid network to synthesize high-quality interpolation frames cyclically. In [87],[24] Oh and Kim introduced a novel framework called *DeMFI,* which joint deblurring and multi-frame interpolation are based on a **flow-guided attentive-correlation-based feature bolstering (FAC-FB) module** and on **recursive boosting (RB).** Zhu et al. [163] proposed an inter-prediction method for blurry video coding based on VFI to generate more credible reference frames.

5.5 Novel View Synthesis

Kalantari et al. [54] proposed a machine learning-based approach to synthesize the novel view. By interpolating between given viewpoints for light field cameras, they applied two sequential CNNs to model disparity and color estimation components, and trained these two networks simultaneously. Flynn et al. [33] introduced a deep learning based DeepStereo method for generating new views. DeepStereo is an end-to-end system and can synthesize the unseen views by interpolating from a set of surrounding pixels. Jin et al. [52] designed a separable convolution network to interpolate new sequence images between remote-sensing time-series data. Sim et al. [107] first presented an extreme VFI network for 4K videos with large motion.

5.6 Frame Interpolation for Medical Images

Applying frame interpolation technology in the field of medical images is a promising research direction, such as **Image-Guided Surgery (IGS)**, 3D reconstruction, and medical image segmentation [39, 67, 80, 114, 130, 138]. Guo et al. [39] studied dynamic medical images interpolation. They designed a **spatio-temporal volumetric interpolation network (SVIN)**[25] for 4D dynamic medical images. Wu et al. [138] introduced a new data augmentation approach based on frame interpolation to boost medical image segmentation accuracy. Wang et al. [130, 131] applied frame interpolation technology to **electron microscope (EM)** slices, and they presented a sparse self-attention aggregation network to synthesize pixels following the continuity of biological tissue. Liang et al. [67] compared six more popular video interpolation algorithms (including SepConv [85], Super SloMo [50], DAIN [4], BMBC [90], AdaCoF [61], RRIN [66]) on a coronary angiography image group dataset, and found a better method to interpolate continuous and clear high frame rate coronary angiography videos. Physicians can greatly minimize the frequency and intensity of X-ray exposure during coronary angiography with this technology. Jiang et al. [49] proposed a **Wireless Capsule Endoscopy (WCE)** VFI method for generating more consistent and detail-rich gastrointestinal (GI) track video.

[24] DeMFI: https://github.com/JihyongOh/DeMFI.

[25] SVIN: https://github.com/guoyu-niubility/SVIN.

5.7 The Commercialized Applications

Nowadays, some real-world applications apply the VFI techniques to generate high frame rate videos. **Smooth Video Project (SVP)**[26] is an advanced media tool that uses frame interpolation techniques to increase the frame rate by generating intermediate frames between original ones and making movie play more smoothly. This tool could convert frame rate up to 60 fps, 120 fps, or more than 144 fps from 24 fps. The users could install SVP on their computer. It is suitable for any up-to-date computer, and SVP makes GPU acceleration possible.

Dain-App [94][27] is a video interpolation application developed on top of the source code of DAIN [4]. This application only works on a computer with NVIDIA graphic cards supporting CUDA 5.0 or bigger.

Morpho Frame Interpolator (DFI)[28] was proposed by the Japanese company Morpho as a video processing technology in their video processing solution package. The Interpolator supports not only 2K video but also 4K video and high frame-rate video of 240 fps, and could generate high frame-rate video and smooth slow video at high-speed processing.

6 CHALLENGES AND FUTURE DIRECTIONS

The current VFI methods have achieved good results on a series of benchmark datasets, including UCF101, Vimeo-90K, and Middlebury. The blooming of deep learning algorithms in the CV area has significantly inspired great VFI approaches in recent years. However, there are still some challenges in the VFI field. For instance, high power consumption, more extensive storage requirements, and long train and reference time, cannot meet real-time requirements. In this section, we will outline these challenges and discuss the research direction in the future.

Computational Inefficiency. The optical flow-based VFI methods based on optical flow and pixel-level warping operations are inefficient in both training and inference, making them unsuitable for a wide range of applications. As shown in Table 9, we compare the reference time of most VFI methods. The fastest VFI method is FLAVR, which could get nearly 80 FPS. However, this runtime is tested on GPU, and in recent years, with the popularity of cloud services, end-users could utilize the cloud framework to compute and storage aspects. Augmented reality applications, cloud games, and virtual reality games have bloomed. Placing video or game data in the loud and interpolating frames locally improve game quality, effectively reducing transmission bandwidth and transmission time. Therefore, it is crucial to design a video interpolation algorithm for edge devices to meet real-time performance requirements. Usman et al. [122] proposed a fast and quality-oriented frame interpolation considering parallel processing for mobile end-user devices requiring low processing power and hardware resources. Ding et al. [27] designed a compression-driven network for VFI. They applied fine-grained pruning [161] based on sparsity-inducing optimization to compress AdaCoF and showed that a 10x compressed AdaCoF can still perform similarly as before. With the development of GPU cloud computing, GPU-accelerated services have been used in many multimedia applications. In [64], Li et al. presented an optimal pricing strategy of GPU-accelerated multimedia processing services. Meantime, some researchers have focused on applying deep learning in the **Internet of Things (IoT)** environments, and they propose many optimization algorithms [36, 62, 63, 160], which will inspire designing real-time video frame interpolation algorithms on edge devices. Real-time performance of VFI on embedded devices will be one of the exceting future research direction, and incorporating with other video enhancement techniques can bloom up a slew of practical and valuable applications, such as VR, AR games, and telemedicine.

[26] SVP: https://www.svp-team.com/.
[27] Dain-App: https://grisk.itch.io/dain-app.
[28] DFI: https://www.morphoinc.com/en/technology/frc.

Large Motion and Occlusion. There are many large motions or occlusions between frames in the natural world scenes, such as in sport, dance scenes, which is hard to handle in VFI methods. Optical flow-based VFI methods use an optical flow map to generate each pixel's value in the target intermediate frame. However, in some cases with complex motions, whether using traditional methods or deep-learning-based methods, it is different to receive an accurate optical flow map. The Kernel-based methods apply spatially-adaptive convolution kernels to synthesize the intermediate frames. The size of the kernel limits the performance of these methods.

On the other hand, one method may choose a very large kernel size to handle large motions, which needs enormous computational resources and is highly inefficient. Lee et al. [61] proposed the AdaCoF approach to estimate both kernel weights and offset vectors for each target pixel to generate the intermediate frames. Although this method removes the constraint on the kernel shape in the spatial domain, it does not fully exploit the degrees of freedom available in whole space-time. Expertly designing a VFI method to deal with large motion and occlusion is still a big challenge.

Event Cameras in VFI Field. Event cameras are a new type of sensor, which is different from traditional frame cameras: with pixels which respond asynchronously to changes in illumination, and output a stream of events that encode the time, location, and sign of the brightness changes. When compared to traditional cameras, event cameras have a variety of benefits, including low-latency, high temporal resolution, high dynamic range, low power, and sparse data output. This technology has recently attracted a lot of interest from academia and industry. Such as object tracking, surveillance and monitoring, and object/gesture recognition, and optical flow estimation, to name a few. In the VFI field, the optical flow between input frames is easily influenced by the change of illumination. Event-camera-based algorithm could overcome this disadvantage. In [68], [88], and [121], the authors had did some exploration,and got better results compared with the traditional flow-based VFI methods. Event-based VFI methods are one of future research directions for VFI scenario.

7 CONCLUSION

In this article, we present a comprehensive review of the VFI technique and systematically divided all the methods into three significant categories: flow-based, kernel-based, and phase-based. We first review the history of the development of VFI algorithms, the evaluation metrics, and publicly available datasets. We then compare each algorithm in detail, point out their advantages and disadvantages, and compare their interpolation performance and speed on different remarkable datasets. VFI technology has attracted continuous attention in the CV community. Some video processing applications based on VFI are also mentioned in this survey, such as slow-motion generation, video compression, video restoration. Finally, we outline the bottleneck faced by the current video frame interpolation technology, and discuss the future work, real-time performance on VR&AR or other embedded devices will be the focus of future research direction.

REFERENCES

[1] Ha-Eun Ahn, Jinwoo Jeong, and Je Woo Kim. 2019. A fast 4K video frame interpolation using a hybrid task-based convolutional neural network. *Symmetry* 11, 5 (2019), 619.

[2] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, and In So Kweon. 2021. Motion-blurred video interpolation and extrapolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[3] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision* 92, 1 (2011), 1–31.

[4] 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 Conference on Computer Vision and Pattern Recognition*. 3703–3712.

[5] 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* (2019).

[6] 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.

[7] Jean Bégaint, Franck Galpin, Philippe Guillotel, and Christine Guillemot. 2019. Deep frame interpolation for video compression. In *Proceedings of the DCC 2019-Data Compression Conference.* IEEE, 1–10.

[8] Tim Brooks and Jonathan T. Barron. 2019. Learning to synthesize motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6840–6848.

[9] 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.

[10] Roberto Castagno, Petri Haavisto, and Giovanni Ramponi. 1996. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on Circuits and Systems for Video Technology* 6, 5 (1996), 436–446.

[11] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning. (PMLR)*, 1691–1703.

[12] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems.* 730–738.

[13] Zhiqi Chen, Ran Wang, Haojie Liu, and Yao Wang. 2021. PDWN: Pyramid deformable warping network for video interpolation. *IEEE Open Journal of Signal Processing* 2 (2021), 413–424.

[14] Xianhang Cheng and Zhenzhong Chen. 2019. A multi-scale position feature transform network for video frame interpolation. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 11 (2019), 3968–3981.

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

[16] Xianhang Cheng and Zhenzhong Chen. 2021. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[17] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N. Plataniotis. 2020. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *European Conference on Computer Vision.* Springer, 107–123.

[18] Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim, and Sung-Jea Ko. 2007. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology* 17, 4 (2007), 407–416.

[19] Hyomin Choi and Ivan V. Bajić. 2019. Deep frame prediction for video coding. *IEEE Transactions on Circuits and Systems for Video Technology* (2019).

[20] Myungsub Choi, Janghoon Choi, Sungyong Baik, Tae Hyun Kim, and Kyoung Mu Lee. 2020. Scene-adaptive video frame interpolation via meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 9444–9453.

[21] Myungsub Choi, Janghoon Choi, Sungyong Baik, Tae Hyun Kim, and Kyoung Mu Lee. 2021. Test-time adaptation for video frame interpolation via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1.

[22] 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.* 10663–10671.

[23] Myungsub Choi, Suyoung Lee, Heewon Kim, and Kyoung Mu Lee. 2021. Motion-aware dynamic architecture for efficient frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 13839–13848.

[24] Whan Choi, Yeong Jun Koh, and Chang-Su Kim. 2021. Multi-scale warping for video frame interpolation. *IEEE Access* 9 (2021), 150470–150479.

[25] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision.* 764–773.

[26] Jiajun Deng, Haichao Yu, Zhangyang Wang, Xinchao Wang, and Thomas Huang. 2019. Self-reproducing video frame interpolation. In *Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).* IEEE, 193–198.

[27] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. 2021. CDFI: Compression-driven network design for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8001–8011.

[28] Xiangling Ding, Yifeng Pan, Qing Gu, Jiyou Chen, Gaobo Yang, and Yimao Xiong. 2021. Detection of deep video frame interpolation via learning dual-stream fusion CNN in the compression domain. In *Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME).* 1–6.

[29] 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 preprint arXiv:2010.11929* (2020).

[30] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision.* 2758–2766.

[31] Saikat Dutta and Anurag Mittal. 2021. ReFIn: A refinement approach for video frame interpolation. In *Proceedings of the NeurIPS 2021 Workshop on Deep Learning and Inverse Problems.*

[32] Saikat Dutta, Nisarg A. Shah, and Anurag Mittal. 2021. Efficient space-time video super resolution using low-resolution flow and mask upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 314–323.

[33] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5515–5524.

[34] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. 2015. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding* 134 (2015), 1–21.

[35] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. 2017. Residual conv-deconv grid network for semantic segmentation. *arXiv preprint arXiv:1707.07958* (2017).

[36] Chao Gong, Fuhong Lin, Xiaowen Gong, and Yueming Lu. 2020. Intelligent cooperative edge computing in internet of things. *IEEE Internet of Things Journal* 7, 10 (2020), 9372–9382.

[37] Donghao Gu, ZhaoJing Wen, Wenxue Cui, Rui Wang, Feng Jiang, and Shaohui Liu. 2019. Continuous bidirectional optical flow for video frame sequence interpolation. In *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME).* IEEE, 1768–1773.

[38] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. 2020. FeatureFlow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14004–14013.

[39] Yuyu Guo, Lei Bi, Euijoon Ahn, Dagan Feng, Qian Wang, and Jinman Kim. 2020. A spatiotemporal volumetric interpolation network for 4D dynamic medical image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4726–4735.

[40] Taehyeun Ha, Seongjoo Lee, and Jaeseok Kim. 2004. Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Transactions on Consumer Electronics* 50, 2 (2004), 752–759.

[41] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. 2020. Space-time-aware multi-resolution video enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2859–2868.

[42] Mengshun Hu, Liang Liao, Jing Xiao, Lin Gu, and Shin'ichi Satoh. 2020. Motion feedback design for video frame interpolation. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020).* IEEE, 4347–4351.

[43] Ai-Mei Huang and Truong Nguyen. 2009. Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation. *IEEE Transactions on Image Processing* 18, 4 (2009), 740–752.

[44] Ai-Mei Huang and Truong Q. Nguyen. 2008. A multistage motion vector processing method for motion-compensated frame interpolation. *IEEE Transactions on Image Processing* 17, 5 (2008), 694–708.

[45] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 8981–8989.

[46] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2462–2470.

[47] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. 2018. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proceedings of the European Conference on Computer Vision (ECCV).* 614–630.

[48] Bo-Won Jeon, Gun-Ill Lee, Sung-Hee Lee, and Rae-Hong Park. 2003. Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. *IEEE Transactions on Consumer Electronics* 49, 3 (2003), 499–508.

[49] Bing Jiang, Yuyao Zhang, Minye Wu, Ji Li, and Jingyi Yu. 2021. Consistent WCE video frame interpolation based on endoscopy image motion estimation. In *Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI).* 334–338.

[50] 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.

[51] Meiguang Jin, Zhe Hu, and Paolo Favaro. 2019. Learning to extract flawless slow motion from blurry videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 8112–8121.

[52] Xing Jin, Ping Tang, Thomas Houet, Thomas Corpetti, Emilien Gence Alvarez-Vanhard, and Zheng Zhang. 2021. Sequence image interpolation via separable convolution network. *Remote Sensing* 13, 2 (2021), 296.

[53] Rico Jonschkowski, Austin Stone, Jonathan T. Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. 2020. What matters in unsupervised optical flow. In *Proceedings of the European Conference on Computer Vision*. Springer, 557–572.

[54] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–10.

[55] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. 2020. FLAVR: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512* (2020).

[56] Suk-Ju Kang, Kyoung-Rok Cho, and Young Hwan Kim. 2007. Motion compensated frame rate up-conversion using extended bilateral motion estimation. *IEEE Transactions on Consumer Electronics* 53, 4 (2007), 1759–1767.

[57] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. 2020. FISR: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *Proceedings of the AAAI.* 11278–11286.

[58] Mark Koren, Kunal Menda, and Apoorva Sharma. 2017. Frame interpolation using generative adversarial networks.

[59] Yong-Hoon Kwon, Ju Hong Yoon, and Min-Gyu Park. 2021. Direct video frame interpolation with multiple latent encoders. *IEEE Access* 9 (2021), 32457–32466.

[60] Jean Le Feuvre, Jean-Marc Thiesse, Matthieu Parmentier, Mickael Raulet, and Christophe Daguet. 2014. Ultra high definition HEVC DASH data set. In *Proceedings of the 5th ACM Multimedia Systems Conference.* 7–12.

[61] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. 2020. AdaCoF: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5316–5325.

[62] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. 2019. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications* 19, 1 (2019), 447–457.

[63] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Network* 32, 1 (2018), 96–101.

[64] He Li, Kaoru Ota, Mianxiong Dong, Athanasios Vasilakos, and Koji Nagano. 2017. Multimedia processing pricing strategy in GPU-accelerated cloud computing. *IEEE Transactions on Cloud Computing* (2017).

[65] Haopeng Li, Yuan Yuan, and Qi Wang. 2019. Fi-net: A lightweight video frame interpolation network using feature-level flow. *IEEE Access* 7 (2019), 118287–118296.

[66] Haopeng Li, Yuan Yuan, and Qi Wang. 2020. Video frame interpolation via residue refinement. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020).* IEEE, 2613–2617.

[67] Dong-xue Liang. 2021. Analysis of coronary angiography video interpolation methods to reduce x-ray exposure frequency based on deep learning. *Cardiovascular Innovations and Applications* (2021).

[68] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy S. J. Ren. 2020. Learning event-driven video deblurring and interpolation. In *Proceedings of the ECCV (8).* 695–710.

[69] Yang Ling, Jin Wang, Yunqiang Liu, and Wenjun Zhang. 2008. A novel spatial and temporal correlation integrated based motion-compensated interpolation for frame rate up-conversion. *IEEE Transactions on Consumer Electronics* 54, 2 (2008), 863–869.

[70] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. 2020. Enhanced quadratic video interpolation. In *Proceedings of the European Conference on Computer Vision.* Springer, 41–56.

[71] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. 2019. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33. 8794–8802.

[72] Zhouyong Liu, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Chunguo Li, and Luxi Yang. 2020. ConvTransformer: A convolutional transformer network for video frame synthesis. *arXiv preprint arXiv:2011.10185* (2020).

[73] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. 2017. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision.* 4463–4471.

[74] Gucan Long, Laurent Kneip, Jose M. Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. 2016. Learning image matching by simply watching video. In *Proceedings of the European Conference on Computer Vision.* Springer, 434–450.

[75] Bruce D. Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision In. *Proceedings of the IJCAI (IJCAI81)* (1981), 674–679.

[76] 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.

[77] Hui Men, Vlad Hosu, Hanhe Lin, Andrés Bruhn, and Dietmar Saupe. 2020. Visual quality assessment for interpolated slow-motion videos based on a novel database. In *Proceedings of the 2020 12th International Conference on Quality of Multimedia Experience (QoMEX).* 1–6.

[78] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. 2018. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 498–507.

[79] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. 2015. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1410–1418.

[80] Thiago Moraes, Paulo Amorim, Jorge Vicente Da Silva, and Helio Pedrini. 2020. Medical image interpolation based on 3D lanczos filtering. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 8, 3 (2020), 294–300.

[81] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3883–3891.

[82] 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.

[83] 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.

[84] Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 670–679.

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

[86] Simon Niklaus, Long Mai, and Oliver Wang. 2020. Revisiting adaptive convolutions for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1099–1109.

[87] Jihyong Oh and Munchurl Kim. 2021. DeMFI: Deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting. *arXiv preprint arXiv:2111.09985* (2021).

[88] Genady Paikin, Yotam Ater, Roy Shaul, and Evgeny Soloveichik. 2021. EFI-net: Video frame interpolation from fusion of events and frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1291–1301.

[89] Anil Singh Parihar, Disha Varshney, Kshitija Pandya, and Ashray Aggarwal. 2021. A comprehensive survey on video frame interpolation techniques. *The Visual Computer* (2021), 1–25.

[90] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. 2020. BRUBC: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Proceedings of the European Conference on Computer Vision*. Springer, 109–125.

[91] 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.

[92] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. 2019. IM-net for high resolution video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2398–2407.

[93] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 724–732.

[94] Gabriel Poetsch. 2020. Dain-app: Application for video interpolations. (2020).

[95] Lars Lau Rakêt, Lars Roholm, Andrés Bruhn, and Joachim Weickert. 2012. Motion compensated frame interpolation with a symmetric optical flow constraint. In *Proceedings of the International Symposium on Visual Computing*. Springer, 447–457.

[96] Anurag Ranjan and Michael J. Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4161–4170.

[97] Fitsum A. Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J. Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2019. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE International Conference on Computer Vision*. 892–900.

[98] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 234–241.

[99] Shibani Santurkar, David Budden, and Nir Shavit. 2018. Generative compression. In *Proceedings of the 2018 Picture Coding Symposium (PCS)*. 258–262.

[100] Stefano Savian, Mehdi Elahi, and Tammam Tillo. 2020. Optical flow estimation with deep learning, a survey on recent advances. In *Deep Biometrics*. Springer, 257–287.

[101] Kara Marie Schatz, Erik Quintanilla, Shruti Vyas, and Yogesh S. Rawat. 2020. A recurrent transformer network for novel view action synthesis. *ECCV (27)* (2020), 410–426.

[102] Kalpana Seshadrinathan and Alan Conrad Bovik. 2010. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Transactions on Image Processing* 19, 2 (2010), 335–350.

[103] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. 2020. Blurry video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5114–5123.

[104] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. 2020. Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Transactions on Image Processing* 30 (2020), 277–292.

[105] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1874–1883.

[106] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. 2021. Video frame interpolation via generalized deformable convolution. *IEEE Transactions on Multimedia* (2021).

[107] 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.

[108] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris N. Metaxas, Chen Change Loy, and Ziwei Liu. 2021. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 6587–6595.

[109] Sanghyun Son, Jaerin Lee, Seungjun Nah, Radu Timofte, Kyoung Mu Lee, Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, et al. 2020. AIM 2020 challenge on video temporal super-resolution. In *European Conference on Computer Vision.* Springer, 23–40.

[110] Li Song, Xun Tang, Wei Zhang, Xiaokang Yang, and Pingjian Xia. 2013. The SJTU 4K video sequence dataset. In *Proceedings of the 2013 5th International Workshop on Quality of Multimedia Experience (QoMEX).* IEEE, 34–35.

[111] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[112] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. 2017. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1279–1288.

[113] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. PWC-NET: CRNS for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 8934–8943.

[114] Na Sun and Huina Li. 2019. Super resolution reconstruction of images based on interpolation and full convolutional neural network and application in medical fields. *IEEE Access* 7 (2019), 186470–186479.

[115] Michael Tao, Jiamin Bai, Pushmeet Kohli, and Sylvain Paris. 2012. SimpleFlow: A non-iterative, sublinear optical flow algorithm. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 345–353.

[116] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision.* Springer, 402–419.

[117] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. 2020. TDAN: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3360–3369.

[118] Phong Tran, Anh Tran, Thao Nguyen, and Minh Hoai. 2021. FineNet: Frame interpolation and enhancement for face video deblurring. *arXiv preprint arXiv:2103.00871* (2021).

[119] Quang Nhat Tran and Shih-Hsuan Yang. 2020. Efficient video frame interpolation using generative adversarial networks. *Applied Sciences* 10, 18 (2020), 6245.

[120] Zhigang Tu, Wei Xie, Dejun Zhang, Ronald Poppe, Remco C. Veltkamp, Baoxin Li, and Junsong Yuan. 2019. A survey of variational and CNN-based optical flow techniques. *Signal Processing: Image Communication* 72 (2019), 9–24.

[121] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. 2021. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 16155–16164.

[122] Muhammad Usman, Xiangjian He, Kin-Man Lam, Min Xu, Syed Mohsin Matloob Bokhari, and Jinjun Chen. 2016. Frame interpolation for cloud-based mobile video streaming. *IEEE Transactions on Multimedia* 18, 5 (2016), 831–839.

[123] Joost van Amersfoort, Wenzhe Shi, Alejandro Acosta, Francisco Massa, Johannes Totz, Zehan Wang, and Jose Caballero. 2017. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *arXiv preprint arXiv:1711.06045* (2017).

[124] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[125] Demin Wang, Andre Vincent, Philip Blanchfield, and Robert Klepko. 2010. Motion-compensated frame rate up-conversion-Part II: New algorithms for frame interpolation. *IEEE Transactions on Broadcasting* 56, 2 (2010), 142–149.

[126] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2021. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5463–5474.

[127] Xintao Wang, Kelvin C. K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. 2019. EDVR: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[128] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. 2021. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8741–8750.

[129] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[130] Zejin Wang, Jing Liu, Xi Chen, Guoqing Li, and Hua Han. 2021. Sparse self-attention aggregation networks for neural sequence slice interpolation. *BioData Mining* 14, 1 (2021), 1–19.

[131] Zejin Wang, Guodong Sun, Lina Zhang, Guoqing Li, and Hua Han. 2021. Temporal spatial-adaptive interpolation with deformable refinement for electron microscopic images. *arXiv preprint arXiv:2101.06771* (2021).

[132] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. 2013. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*. 1385–1392.

[133] Shiping Wen, Weiwei Liu, Yin Yang, Tingwen Huang, and Zhigang Zeng. 2019. Generating realistic videos from keyframes with concatenated GANs. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 8 (2019), 2337–2348. https://doi.org/10.1109/TCSVT.2018.2867934

[134] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. 2011. Optical flow guided TV-L 1 video interpolation and restoration. In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 273–286.

[135] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. 2018. Video compression through image interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 416–431.

[136] Jiyan Wu, Chau Yuen, Ngai-Man Cheung, Junliang Chen, and Chang Wen Chen. 2015. Modeling and optimization of high frame rate video transmission over wireless networks. *IEEE Transactions on Wireless Communications* 15, 4 (2015), 2713–2726.

[137] Xuanyi Wu, Zhenkun Zhou, and Anup Basu. 2021. DRVI: Dual refinement for video interpolation. *IEEE Access* 9 (2021), 113566–113576.

[138] Zhaotao Wu, Jia Wei, Wenguang Yuan, Jiabing Wang, and Tolga Tasdizen. 2020. Inter-slice image augmentation based on frame interpolation for boosting medical image segmentation accuracy. *arXiv preprint arXiv:2001.11698* (2020).

[139] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. 2020. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3370–3379.

[140] Jinbo Xing, Wenbo Hu, Yuechen Zhang, and Tien-Tsin Wong. 2021. Flow-aware synthesis: A generic motion model for video frame interpolation. *Computational Visual Media* (2021), 1–13.

[141] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. 2019. Quadratic video interpolation. *Advances in Neural Information Processing Systems* 32 (2019), 1647–1656.

[142] Fanyong Xue, Jie Li, Jiannan Liu, and Chentao Wu. 2021. BWIN: A bilateral warping method for video frame interpolation. In *Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6.

[143] 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, 8 (2019), 1106–1125.

[144] Wei Xue, Hong Ai, Tianyu Sun, Chunfeng Song, Yan Huang, and Liang Wang. 2020. Frame-GAN: Increasing the frame rate of gait videos with generative adversarial networks. *Neurocomputing* 380 (2020), 95–104.

[145] Bo Yan, Weimin Tan, Chuming Lin, and Liquan Shen. 2020. Fine-grained motion estimation for video frame interpolation. *IEEE Transactions on Broadcasting* (2020).

[146] Kai-Chieh Yang, Ai-Mei Huang, Truong Q. Nguyen, Clark C. Guest, and Pankaj K. Das. 2008. A new objective quality metric for frame interpolation used in video compression. *IEEE Transactions on Broadcasting* 54, 3 (2008), 680–11.

[147] Songhyun Yu, Bumjun Park, and Jechang Jeong. 2019. PoSNet: 4x video frame interpolation using position-specific flow. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 3503–3511.

[148] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. 2013. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 7 (2013), 1235–1248.

[149] Liangzhe Yuan, Yibo Chen, Hantian Liu, Tao Kong, and Jianbo Shi. 2019. Zoom-in-to-check: Boosting video interpolation via instance-level discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 12183–12191.

[150] Haoxian Zhang, Ronggang Wang, and Yang Zhao. 2019. Multi-frame pyramid refinement network for video frame interpolation. *IEEE Access* 7 (2019), 130610–130621.

[151] Haoxian Zhang, Yang Zhao, and Ronggang Wang. 2020. A flexible recurrent residual pyramid network for video frame interpolation. In *European Conference on Computer Vision*. Springer, 474–491.

[152] Lin Zhang, Ying Shen, and Hongyu Li. 2014. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing* 23, 10 (2014), 4270–4281.

[153] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 586–595.

[154] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 286–301.

[155] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. 2020. Video frame interpolation without temporal priors. *Advances in Neural Information Processing Systems* 33 (2020), 13308–13318.

[156] Bin Zhao and Xuelong Li. 2021. EA-Net: Edge-aware network for flow-based video frame interpolation. *arXiv preprint arXiv:2105.07673* (2021).

[157] Lei Zhao, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. 2018. Enhanced CTU-level inter-prediction with deep frame rate up-conversion for high efficiency video coding. In *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 206–210.

[158] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. 2020. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315* (2020).

[159] Chengcheng Zhou, Zongqing Lu, Linge Li, Qiangyu Yan, and Jing-Hao Xue. 2021. *How Video Super-Resolution and Frame Interpolation Mutually Benefit*. Association for Computing Machinery, New York, NY, USA, 5445–5453.

[160] Jingyue Zhou, Yihuai Wang, Kaoru Ota, and Mianxiong Dong. 2019. AAIoT: Accelerating artificial intelligence in IoT systems. *IEEE Wireless Communications Letters* 8, 3 (2019), 825–828.

[161] Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: Exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878* (2017).

[162] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9308–9316.

[163] Zezhi Zhu, Lili Zhao, Xuhu Lin, Xuezhou Guo, and Jianwen Chen. 2021. Deep inter prediction via reference frame interpolation for blurry video coding. In *Proceedings of the 2021 International Conference on Visual Communications and Image Processing (VCIP)*. 1–5.

Received 24 February 2022; revised 13 June 2022; accepted 30 July 2022