

XVFI: eXtreme Video Frame Interpolation

Hyeonjun Sim*

Jihyong Oh*

Munchurl Kim†

Korea Advanced Institute of Science and Technology

{f1hy5836, jhoh94, mkimee}@kaist.ac.kr

(a) 96.6 (b) 71.0 (c) 34.9 (d) 40.6 (e) 196.5 (f) 152.2

Figure 1. Some examples of our X4K1000FPS dataset, which contain diverse motions in 4K-resolution of 1000-fps. The numbers below the examples are the magnitude means of optical flows between two input frames in 30 fps. This is a video figure that can be best viewed with motion using Adobe™ Reader. It should be noted that they are rendered in down-scales at 15 fps for visualization convenience.

Abstract

In this paper, we firstly present a dataset (X4K1000FPS) of 4K videos of 1000 fps with the extreme motion to the research community for video frame interpolation (VFI), and propose an extreme VFI network, called XVFI-Net, that first handles the VFI for 4K videos with large motion. The XVFI-Net is based on a recursive multi-scale shared structure that consists of two cascaded modules for bidirectional optical flow learning between two input frames (BiOF-I) and for bidirectional optical flow learning from target to input frames (BiOF-T). The optical flows are stably approximated by a complementary flow reversal (CFR) proposed in BiOF-T module. During inference, the BiOF-I module can start at any scale of input while the BiOF-T module only operates at the original input scale so that the inference can be accelerated while maintaining highly accurate VFI performance. Extensive experimental results show that our XVFI-Net can successfully capture the essential information of objects with extremely large motions and complex textures while the state-of-the-art methods exhibit poor performance. Furthermore, our XVFI-Net framework also performs comparably on the previous lower resolution benchmark dataset, which shows a robustness of our algorithm as well. All source codes, pre-trained models, and proposed X4K1000FPS datasets are publicly available at <https://github.com/JihyongOh/XVFI>.

1. Introduction

Video frame interpolation (VFI) converts low frame rate (LFR) contents to high frame rate (HFR) videos by synthesizing one or more intermediate frames between given two consecutive frames, and then the videos of fast motion can be smoothly rendered in an increased frame rate, thus yielding reduced motion judder [28, 24, 23, 10]. Therefore, it is widely used for various practical applications, such as adaptive streaming [46], novel view interpolation synthesis [11], frame rate up conversion [29, 5, 50], slow motion generation [18, 4, 30, 32, 27, 34] and video restoration [21, 43, 14, 42]. However, VFI is significantly challenging, which is attributed to diverse factors such as occlusions, large motions and change of light. Recent deep-learning-based VFI has been actively studied, showing remarkable performances [48, 4, 7, 37, 25, 13, 31, 51, 6, 33]. However, they are often optimized for existing LFR benchmark datasets of low resolution (LR), which may lead to poor VFI performance, especially for videos of 4K resolution (4096×2160) or higher with very large motion [1, 21]. Such 4K videos often contain frames of fast motion with extremely large pixel displacements for which conventional convolutional neural networks (CNNs) do not effectively work with receptive fields of limited sizes.

To solve the above issues for deep learning-based VFI methods, we directly photographed 4K videos to construct a high-quality HFR dataset of high resolution, called X4K1000FPS. Fig. 1 shows some examples of our X4K1000FPS dataset. As shown, our videos of 4K resolution have a wide variety of textures, extremely large mo-

*Both authors contributed equally to this work.

†Corresponding author.

tions, zoomings and occlusions, which have never been seen in the previous VFI benchmark datasets [3, 39, 35, 40, 49].

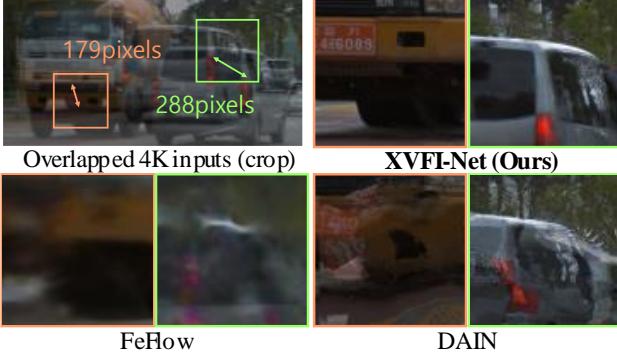


Figure 2. VFI results for extreme motions. Our XVFI-Net can generate a more stable intermediate frame with very large motions than two recent SOTA methods, FeFlow [13] and DAIN [4], which are newly trained on our dataset for fair comparisons.

We also first propose an extreme VFI model, called XVFI-Net, that is effectively designed to handle such a challenging dataset of 4K@1000fps. Instead of directly capturing extreme motions through consecutive feature spaces with deformable convolution as recent trends in video restoration [13, 47, 43, 42, 20], or using very large-sized pretrained networks with extra information such as contexts, depths, flows and edges [4, 51, 30, 13], our XVFI-Net is simple but effective, which is based on a recursive multi-scale shared structure. The XVFI-Net has two cascaded modules: one for the bidirectional optical flow learning between two input frames (BiOF-I) and the other for the bidirectional optical flow estimating from target to the inputs (BiOF-T). The BiOF-I and BiOF-T modules are trained in combination with multi-scale losses. However, once trained, the BiOF-I module can start from any down-scaled input upward while the BiOF-T module only operates at the original input scale at inference, which is computationally efficient and helps to generate an intermediate frame at any target time instance. Structurally, the XVFI-Net is adjustable in terms of the number of scales for inference according to the input resolutions or the motion magnitudes, even if training is once over. We also propose a novel optical flow estimation from time t to those of the inputs, called a complementary flow reversal (CFR) that effectively fills the holes by taking complementary flows. Extensive experiments are conducted for fair comparison and our XVFI-Net that has a relatively smaller complexity outperforms previous VFI SOTA methods on our X4K1000FPS, especially for extreme motions as shown in Fig. 2. A further experiment on the previous LR-LFR benchmark dataset also demonstrates the robustness of our XVFI-Net. Our contributions are summarized as:

- We first propose a high-quality of HFR video dataset

of 4K resolution, called X4K1000FPS (4K@1000fps) which contains a wide variety of textures, extremely large motions, zoomings and occlusions.

- We propose the CFR that can generate stable optical flow estimation from time t to 0 or 1, boosting both qualitative and quantitative performances.
- Our proposed XVFI-Net can start from any down-scaled input upward, which is adjustable in terms of the number of scales for inference according to the input resolutions or the motion magnitudes.
- Our XVFI-Net achieves state-of-the-art performance on the testset of X4K1000FPS with a significant margin compared to the previous VFI SOTA methods while having computational efficiency with a small number of filter parameters. All source codes and proposed X4K1000FPS dataset are publicly available at <https://github.com/JihyongOh/XVFI>.

2. Related Work

2.1. Video Frame Interpolation

Most VFI methods can be categorized into optical flow- or kernel-based [27, 32, 18, 30, 34, 48, 4, 1, 2, 25, 33, 31] and pixel hallucination-based [13, 47, 7, 37, 21] methods.

Flow-based VFI. Super-SloMo [18] first linearly combines predicted optical flows between two input frames to approximate flows from the target intermediate frame to the input frames. Quadratic video frame interpolation [48] utilizes four input frames to cope with nonlinear motion modeling by quadratic approximation, which limits the VFI generalization when two input frames are given. It also proposes flow reversal (projection) for more accurate image warping. On the other hand, DAIN [4] gives different weights of overlapped flow vectors depending on the object depth of the scene via a flow projection layer. However, DAIN employs and fine-tunes both PWC-Net [41] and MegaDepth [26], which is computationally burdened for inferring intermediate HR frames. AdaCoF proposes a generalized warping module to deal with complex motion [25]. However, it is not adaptive to handle the frames of higher resolutions due to a fixed dilation degree, after once trained.

Pixel Hallucination-based VFI. FeFlow [13] has benefited from deformable convolution [9] to the center frame generator by replacing optical flows with offset vectors. Zooming Slow-Mo [47] also interpolates middle frames with the help of deformable convolution in the feature domain. However, since these methods directly hallucinate pixels unlike the flow-based VFI methods, the predicted frames tend to be blurry when fast-moving objects are present.

Most importantly, the aforementioned VFI methods are difficult to operate on the entire HR frames *at once*, due to

their heavy computational complexity. On the other hand, our XVFI-Net is designed to efficiently operate on the entire 4K frame input at once with a smaller number of parameters and is capable of effectively capturing large motions.

2.2. Networks for Large Pixel Displacements

PWC-Net [41] is a state-of-the-art optical flow estimator that has been adopted in several VFI methods for pretrained flow estimators [48, 4, 31]. Since PWC-Net has a 6-level feature pyramid structure to have larger sizes of receptive fields, it enables to effectively predict large motions. IM-Net [34] also adopts a multi-scale structure to cover large displacements of objects in adjacent frames while the coverage is limited in the size of the adaptive filters. Despite of the multi-scale pyramid structures, the above methods are lack of adaptivity because the coarsest level of each network is fixed after once trained, i.e. each scale level consists of its own (*not shared*) parameters. The RRPN [51] shares weight parameters across different scale levels in a flexible recurrent pyramid structure. However, it only infers the centered frame, not at arbitrary time instances. So it can only synthesize recursively the intermediate frames of time at a power of 2. As a result, the prediction errors are accumulated as intermediate frames are generated iteratively between the two input frames. As a result, RRPN has limited temporal flexibility for VFI at arbitrary target time instance t .

Distinguished from the above methods, our proposed XVFI-Net has a scalable structure with shared parameters for various input resolutions. Different from RRPN [51], the XVFI-Net is structurally divided into the BiOF-I and BiOF-T modules, which allows predicting an intermediate frame at arbitrary time t with the help of the complementary flow reversal in an efficient way. That is, the BiOF-T module can be skipped at the down-scaled levels in inference so that our model can infer the intermediate frame of 4K at once, without any patch-wise iteration unlike all other previous methods, which can be applied in real-world applications.

3. Proposed X4K1000FPS Dataset

Although numerous methods for VFI have been both trained and evaluated over the diverse benchmark datasets, such as Adobe240fps [40], DAVIS [35], UCF101 [39], Middlebury [3] and Vimeo90K [49], none of the datasets contains rich amounts of 4K videos with HFR. These limits the study of elaborate VFI methods required for VFI applications for targeting very high resolution videos.

To tackle the challenging extreme VFI task, we provide a rich set of 4K@1000fps video that we photographed using a Phantom Flex4K™ camera by setting the 4K spatial resolution of 4096×2160 at 1,000 fps, producing 175 video clips of different scenes, each with 5,000 frames by shooting for 5 seconds. As shown in Fig. 1, our dataset provides

Dataset	Occlusion [16]			Flow magnitude [16]		
	25_{th}	50_{th}	75_{th}	25_{th}	50_{th}	75_{th}
Vimeo90K [49]	6.8	11.9	18.1	3.1	4.9	7.1
Adobe240fps [40]	0.8	1.7	3.2	3.8	8.9	16.3
X-TEST (ours)	2.1	5.6	17.7	23.9	81.9	138.5
X-TRAIN (ours)	6.9	10.1	15.7	5.5	18.0	59.5

25_{th} , 50_{th} and 75_{th} represent percentiles of each dataset.

Table 1. The occlusion and optical flow magnitude statistics of VFI datasets: 3,782 test triplets of Vimeo90K [49], randomly selected 200 clips of Adobe240fps [40], 15 clips of X-TEST and 4,408 clips of X-TRAIN.

a wide range of object motions and various camera motion types at different speeds in diverse locations.

In order to select valuable data samples for VFI, we estimated bidirectional occlusion maps and optical flows of every 32 frames of the scenes using IRR-PWC [16]. The occlusion map indicates part of the objects to be occluded in the next frames. The occlusion makes optical flow estimation and frame interpolation challenging [44, 4, 16]. Thus, we manually selected 15 scenes as our testset, called X-TEST, by considering the degrees of occlusion, optical flow magnitudes and scene diversity. Each scene for X-TEST simply contains one test sample that consists of two input frames in a temporal distance of 32 frames and approximately corresponds to 30 fps. The test evaluation is set to interpolate 7 intermediate frames, which results in the consecutive frames of 240 fps. For the training dataset, called X-TRAIN, we first divide each of the 1000-fps 4K-resolution scenes into about 386K cropped clips of the sizes of 768×768 and the lengths of 65 consecutive frames with overlap allowed. Then, the top 10% of non-overlapping samples (clips) in occlusion amounts are selected for our X-TRAIN, resulting in 4,408 clips.

Table 1 compares the statistics of datasets: Vimeo90K [49], Adobe240fps [40], our X-TEST and X-TRAIN. We estimated the occlusion range in $[0, 255]$ and optical flow magnitudes [16] between input pairs and calculated their percentiles for each dataset. As shown in Table 1, our datasets contain comparable occlusion but significantly larger motion, compared to the previous VFI datasets.

4. Proposed Method : XVFI-Net Framework

4.1. Design Considerations

Our XVFI-Net aims at interpolating an intermediate frame I_t at an arbitrary time t between two consecutive input frames, I_0 and I_1 , of HR with extreme motion.

Scale Adaptivity. An architecture with a fixed number of scale levels like PWC-Net [41] is difficult to adapt to various spatial resolutions of the input video, because the structure in each scale level is *not shared* across different scale levels, so the new architecture with an increased scale depth

needs to be retrained. In order to have a scale adaptivity to various spatial resolutions of input frames, our XVFI-Net is designed to have optical flow estimation starting at any desired coarse scale level, adapting to the degree of motion magnitudes in the input frames. To do so, our XVFI-Net shares its parameters across different scale levels.

Capturing Large Motion. In order to effectively capture a large motion between two input frames, the Feature Extraction Block of XVFI-Net first reduces the spatial resolution of two input frames by a module scale factor M via a strided convolution, thus yielding the spatially reduced feature that is then converted to two contextual feature maps C_0^0 and C_1^0 . The Feature Extraction Block in Fig. 3 is composed of the strided convolution and two residual blocks [15]. Then, XVFI-Net at each scale level estimates optical flows from target frame I_t to two input frames in the reduced spatial size by M . The predicted flows are up-scaled ($\times M$) to warp the input frames at each scale level to time t .

4.2. XVFI-Net Architecture

BiOF-I module. Fig. 4 shows our XVFI-Net architecture in scale s , where I^s denotes bicubically down-scaled by $1/2^s$. First, contextual pyramid $\mathbf{C} = \{C^s\}$ is recurrently extracted from C_0^0 and C_1^0 via a stride 2 convolution, and then utilized as an input for XVFI-Net at each scale level s ($s = 0, 1, 2, \dots$), where $s = 0$ denotes the scale of the original input frames. Let $F_{t_a t_b}^s$ denotes optical flow from time t_a to t_b at scale s . F_{01}^s and F_{10}^s are the bidirectional flows between input frames at scale s . F_{t0}^s and F_{t1}^s are the bidirectional flows from I_t^s to I_0^s and I_1^s , respectively.

The estimated flows $F_{01}^{s+1}, F_{10}^{s+1}$ from the previous scale ($s + 1$) are $\times 2$ bilinearly up-scaled to be set as the initial flows for the current scale s , i.e., $F_{01}^s = F_{01}^{s+1} \uparrow_2, F_{10}^s = F_{10}^{s+1} \uparrow_2$. To update the initial flows in the current scale, C_0^s and C_1^s are first warped by the initial flows, that is, $\tilde{C}_{01}^s = W(\tilde{F}_{01}^s, C_1^s)$ and $\tilde{C}_{10}^s = W(\tilde{F}_{10}^s, C_0^s)$, respectively, where W is a backward warping operation [17]. Then $\tilde{C}_{01}^s, \tilde{C}_{10}^s, C_0^s, C_1^s$ together with $\tilde{F}_{01}^s, \tilde{F}_{10}^s$ are fed to an auto-encoder-based BiFlownet as in Fig. 4 to output residual flows over the initial flows and a trainable importance mask z [31]. Then F_{01}^s, F_{10}^s are obtained. They are then fed as input to the BiOF-T module and are also used as the initial flows to the next scale $s - 1$.

BiOF-T module. Hereafter, we omit superscript s for the notion of feature tensors at each scale, unless mentioned. Although the linear approximation with optical flows F_{01}, F_{10} [18] or the flow reversal of F_{0t}, F_{1t} [48] allows to estimate the flows F_{t0}, F_{t1} at arbitrary time t , there are few shortcomings. The linear approximation is inaccurate to predict F_{t0} and F_{t1} for fast-moving objects because the anchor points of F_{01} and F_{10} are severely misaligned. On the other hand, the flow reversal can align the anchor points but *holes* may appear in estimated F_{t0} and F_{t1} . To

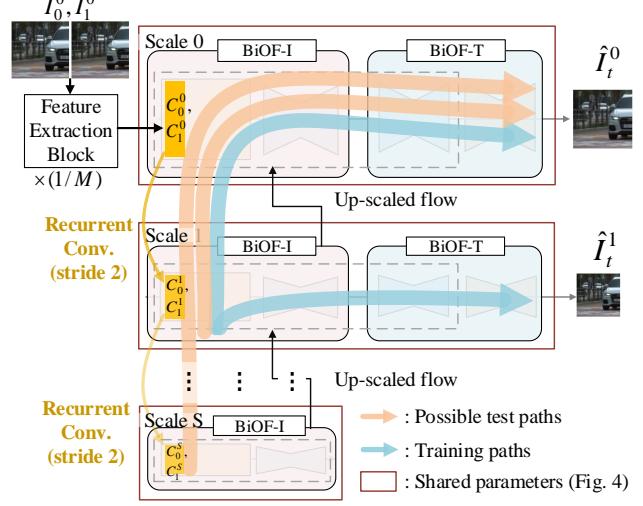


Figure 3. Adjustable and efficient scalability of our XVFI-Net framework. Even if the lowest scale depth S_{trn} during training is set to 1 in this example, inference can start from any scale level.

stabilize the performance of the flow reversal, we take complementary advantages of both the linear approximation and flow reversal. So, a stable optical flow estimate from time t to 0 or 1 can be computed by a normalized linear combination of a negative anchor flow and a complementary flow, which we call a complementary flow reversal (CFR). The resulting complementary reversed optical flow maps, \tilde{F}_{t0} and \tilde{F}_{t1} , from time t to 0 and 1 are given by,

$$\tilde{F}_{t0}^{\mathbf{x}} = \frac{(1-t) \sum_{N_0} w_0 \cdot (-F_{0t}^{\mathbf{y}}) + t \sum_{N_1} w_1 \cdot F_{1 \cdot (1-t)}^{\mathbf{y}}}{(1-t) \sum_{N_0} w_0 + t \sum_{N_1} w_1} \quad (1)$$

$$\tilde{F}_{t1}^{\mathbf{x}} = \frac{(1-t) \sum_{N_0} w_0 \cdot F_{0 \cdot (1-t)}^{\mathbf{y}} + t \sum_{N_1} w_1 \cdot (-F_{1t}^{\mathbf{y}})}{(1-t) \sum_{N_0} w_0 + t \sum_{N_1} w_1} \quad (2)$$

where \mathbf{x} denotes a pixel location at time t and \mathbf{y} is at time 0 or 1. $w_i = z_i^{\mathbf{y}} \cdot G(|\mathbf{x} - (\mathbf{y} + F_{it}^{\mathbf{y}})|)$ is a Gaussian weight depending on the distance between \mathbf{x} at time t and $\mathbf{y} + F_{it}^{\mathbf{y}}$ at time i ($= 0$ or 1) while also considering the learnable importance mask of each flow by $z_i^{\mathbf{y}}$ [31]. Also, $-F_{0t}^{\mathbf{y}}$ (or $-F_{1t}^{\mathbf{y}}$) and $F_{1 \cdot (1-t)}^{\mathbf{y}}$ (or $F_{0 \cdot (1-t)}^{\mathbf{y}}$) in Eq. 1 (or Eq. 2) are defined as a negative anchor flow and a complementary flow, respectively. Furthermore, the anchor flows are normalized flows that can be calculated as $F_{0t} = tF_{01}$ and $F_{1t} = (1-t)F_{10}$ to intermediate time t . It should be noted in Eq. 1 and Eq. 2 that the complementary flows are also *normalized* as $F_{1 \cdot (1-t)} = tF_{10}$ and $F_{0 \cdot (1-t)} = (1-t)F_{01}$ which are complementally fill the holes occurred in the reversed flows. By doing so, we can fully exploit the temporal-densely captured X4K1000FPS dataset to train our XVFI-Net for VFI at arbitrary time t . The neighborhoods of \mathbf{x} are defined as,

$$\mathcal{N}_0 = \{\mathbf{y} \mid \text{round}(\mathbf{y} + F_{0t}^{\mathbf{y}}) = \mathbf{x}\} \quad (3)$$

$$\mathcal{N}_1 = \{\mathbf{y} \mid \text{round}(\mathbf{y} + F_{1t}^{\mathbf{y}}) = \mathbf{x}\}. \quad (4)$$

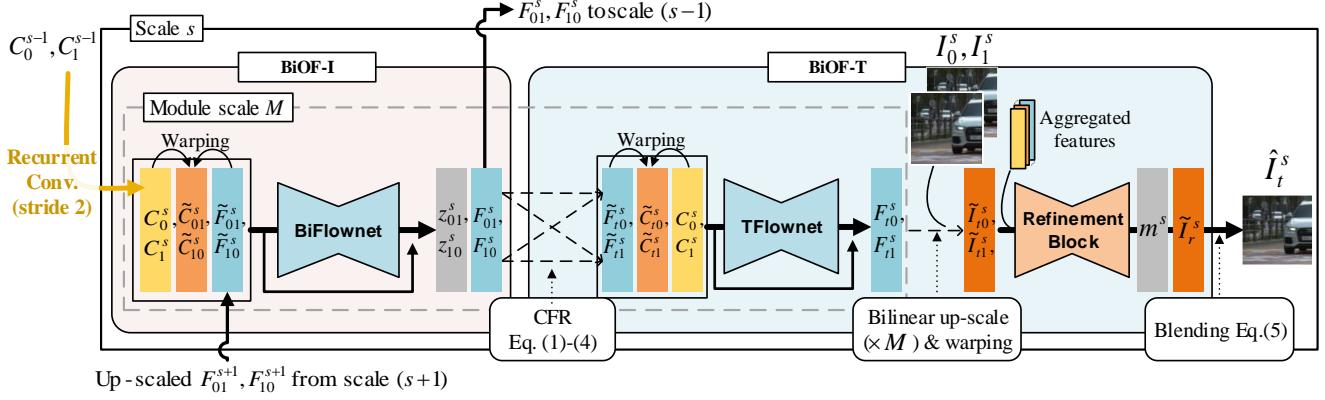


Figure 4. The architecture of our proposed XVFI-Net in scale s .

To refine the bidirectional flow approximates $\tilde{F}_{t0}, \tilde{F}_{t1}$, we re warp the feature maps (C_0, C_1) to \tilde{C}_{t0} and \tilde{C}_{t1} by \tilde{F}_{t0} and \tilde{F}_{t1} , respectively. We concatenate and feed $C_0, C_1, \tilde{C}_{t0}, \tilde{C}_{t1}$, and $\tilde{F}_{t0}, \tilde{F}_{t1}$ to the auto-encoder-based TFlownet as in Fig. 4 (similarly to refine $\tilde{F}_{01}, \tilde{F}_{10}$). The outputs of TFlownet are used to compose refined flows F_{t0}, F_{t1} which are then bilinearly up-scaled ($\times M$) back to the size of I_t^s . The flow estimation in the spatially reduced size by M has three advantages: (i) enlarged receptive fields, (ii) lowered computational costs and (iii) smooth optical flows. This strategy maximizes the benefit of flow-based VFI that can fully utilize the texture information of the original input frames by warping them with the estimated flows, compared to the hallucination-based methods that suffer from a lack of sharpness in restoration from down-scaled feature maps. The above up-scaled flows are used to warp the input frames I_0^s and I_1^s to be \tilde{I}_{t0}^s and \tilde{I}_{t1}^s , respectively. The $C_0^s, C_1^s, \tilde{C}_{t0}^s, \tilde{C}_{t1}^s, F_{t0}^s, F_{t1}^s, I_0^s, I_1^s, \tilde{I}_{t0}^s$ and \tilde{I}_{t1}^s are all aggregated to be fed into the U-Net [36]-based Refinement Block. Then, both the generated occlusion mask m^s and residual image \tilde{I}_r^s are finally used to blend the warped frames \tilde{I}_{t0}^s and \tilde{I}_{t1}^s , which is given by,

$$\hat{I}_t^s = \frac{(1-t) \cdot m^s \cdot \tilde{I}_{t0}^s + t \cdot (1-m^s) \cdot \tilde{I}_{t1}^s}{(1-t) \cdot m^s + t \cdot (1-m^s)} + \tilde{I}_r^s \quad (5)$$

where \hat{I}_t^s is the final result of each scale level s .

4.3. Adjustable and Efficient Scalability

Adjustable Scalability. Fig. 3 shows a VFI framework of our XVFI-Net that can begin from any scale level by $\times 1/2^s$ recurrent down-scaling the contextual feature map C_0^0 and C_1^0 , and predicts the coarsest optical flow to capture extreme motion effectively. Then the estimated flows F_{01}^s, F_{10}^s are transmitted to the next scale $s - 1$, and the flow is updated gradually to the original scale $s = 0$. We aim that the number of scales can be decided for inference, adaptive to the spatial resolution and degree of motion magnitudes for

the input frames, even after once trained. To generalize the XVFI-Net learning for the input of any scale level, a multi-scale reconstruction loss in Eq. 7 is applied for every output \hat{I}_t^s for the selected scale depth S_{trn} during training.

Efficient Scalability. As shown in Fig. 3, the computation through the BiOF-T module is always taken place at the original scale ($s = 0$) during inference no matter which scale level the BiOF-I starts from, which are denoted as the arrows in the light orange color. Since F_{01}^s and F_{10}^s are the only information that passes across different scale levels through the BiOF-I module (from the previous scale to the next scale level) as shown in Fig. 3, we only pass the two optical flows recursively until reaching the original scale level. Then, the BiOF-T module processes $F_{10}^{s=0}$ and $F_{01}^{s=0}$ to estimate $F_{t1}^{s=0}$ and $F_{t0}^{s=0}$ only at the original scale level. This is architecturally very beneficial because (i) the BiOF-I module is responsible to stably capture extreme motion by recursively learning the bidirectional flows between input time instances 0 and 1 across multiple scale levels, and (ii) the BiOF-T module finely predicts the bidirectional flows in the original scale only from any target time t to times 0 and 1 based on the stably estimated flows $F_{10}^{s=0}$ and $F_{01}^{s=0}$, unlike the RRPN [51].

Loss Functions. We adopt a multi-scale reconstruction loss to train the shared parameters of our XVFI-Net. To further encourage the smoothness of the obtained optical flow, the first-order edge-aware smoothness loss is used for F_{t0}^0 and F_{t1}^0 at the original scale [19]. The total loss function is a weighted sum of the two loss functions as follows:

$$\mathcal{L}_{total} = \mathcal{L}_r + \lambda_s \cdot \mathcal{L}_s \quad (6)$$

$$\mathcal{L}_r = \sum_{s=0}^{S_{trn}} \|\hat{I}_t^s - I_t^s\|_1 \quad (7)$$

$$\mathcal{L}_s = \sum_{i=0,1} \exp(-e^2 \sum_c |\nabla_x I_{t_c}^0|^\tau \cdot |\nabla_x F_{t_i}^0|) \quad (8)$$

where c, e^2 and x denote color channel index, an edge weighting factor and a spatial coordinate, respectively.

5. Experiment Results

The proposed X-TRAIN dataset contains 4,408 clips of the sizes of 768×768 and the lengths of 65 consecutive frames. Each training sample is randomly fetched on the fly from each clip. A training sample is defined as a triplet with two input frames (I_0, I_1) and one target frame ($I_t, 0 < t < 1$). The temporal distance between I_0 and I_1 is randomly selected in the range [2, 32] where I_t is also randomly determined between the selected I_0 and I_1 . By doing so, our training samples are stochastically well obtained by fully exploiting our X-TRAIN dataset of temporally dense video clips to learn various t accordingly.

The weights of the XVFI-Net are initialized with Xavier initialization [12] and the mini-batch size is set to 8. XVFI-Net is trained via total of 110,200 iterations (200 epochs) by using the Adam optimizer [22] with the initial learning rate of 10^{-4} , reduced by a factor of 4 at [100, 150, 180]-*th* epoch. The hyperparameter M , λ_s and e are set to 4, 0.5 and 150, respectively. We also randomly crop 384×384 -sized patches from the original size of X-TRAIN and randomly flip both spatial and temporal directions for data augmentation. Training takes about a half-day with an NVIDIA TITAN RTX™ GPU with PyTorch.

5.1. Comparison to the Previous Methods

We compare our XVFI-Net with three previous VFI methods, DAIN [4], FeFlow [13] and AdaCoF [25], whose training codes are *publicly available*. DAIN can generate the interpolated frame at arbitrary time t at once and the latter two can only synthesize the intermediate frame at the power of 2 in an iterative manner during the inference.

For a fair comparison, we *retrain* the three previous methods on X-TRAIN under their original hyperparameter settings except the patch size of 384×384 and the total iterations of 110,200. For further comparison, we also use the original pretrained models of the three methods, which are denoted as the subscript o to distinguish from their retrained models with the subscript f on X-TRAIN. The lowest scale depths for our XVFI-Net are set to 3 for training (S_{trn}) and 3 or 5 for testing (S_{tst}). We evaluate their performances for 7 interpolated frames per scene (multi-frame interpolation $\times 8$) on X-TEST in terms of three evaluation metrics: PSNR, SSIM [45] and tOF [8] that measures the *temporal* consistency for the pixel-wise difference of motions (the lower, the better). We also evaluate each method for 7 interpolated frames per clip on the Adobe240fps dataset [40], where 200 nonuplets clips are randomly extracted with 1280×720 (HD) at 240fps.

Quantitative Comparison. Table 2 shows the quantitative comparisons of the VFI methods on both X-TEST and Adobe240fps. Please note that all runtimes (R_t) in Table 2 are measured for 1024×1024 -sized frames because DAIN and FeFlow are too heavy to run for each of 4K input

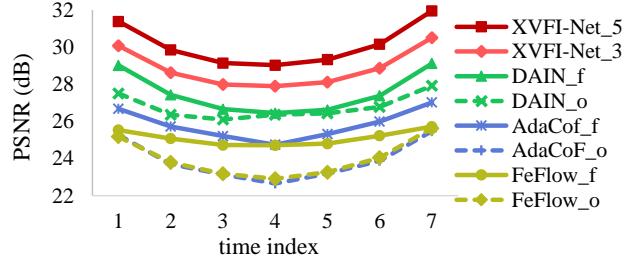


Figure 5. PSNR profiles for multi-frame interpolation results ($\times 8$) on X-TEST.

Methods ($\times N$)	X-TEST (PSNR/SSIM/tOF)	Adobe240fps (PSNR/SSIM)	#P ↓	R _t ↓
AdaCoF _o ($\times 5.8$)	23.90/0.727/6.89	25.26/0.785	21.8	0.005
AdaCoF _f [25]	25.81/0.772/6.42	25.21/0.791	21.8	0.005
FeFlow _o ($\times 5.3$)	24.00/0.756/6.59	25.18/0.785	102.5	1.681
FeFlow _f [13]	25.16/0.783/6.54	24.17/0.780	102.5	1.681
DAIN _o ($\times 9.3$)	26.78/0.807/3.83	29.89/ 0.911	24	1.375
DAIN _f [4]	27.52/0.821/3.47	29.99/0.910	24	1.375
Ours ($S_{tst}=3$)	28.86/0.858/2.67	30.29/0.912	5.5	0.074
Ours ($S_{tst}=5$)	30.12/0.870/2.15	30.18/0.911	5.5	0.075

$\times N$: The ratio of number of iterations of the original version to that of - retrained version in the fair condition. #P: The number of parameters (M).

R_t: The runtime on 1024×1024 -sized frames in sec.

RED: Best performance, BLUE: Second best performance.

Table 2. Quantitative comparisons on both X-TEST (4K) and Adobe240fps (HD) [40] for multi-frame interpolation ($\times 8$).

frames (4096×2160) *at once*. As shown in Table 2, our proposed XVFI-Net with S_{tst} set to both 3 and 5 clearly outperforms all the previous methods with large margins on both X-TEST and Adobe240fps, even though the number of model parameters (#P) of our model is significantly smaller than those of the others. It is also worthwhile to note that our model can infer the intermediate frames of 4K *at once*, without any patch-wise iteration. In particular, XVFI-Net ($S_{tst}=5$) outperforms DAIN_f by 2.6dB, 0.049 and 1.32 in terms of PSNR, SSIM and tOF, respectively, for X-TEST, by utilizing only 22.9% of DAIN's parameters.

Especially for the X-TEST that contains significantly extreme motions in 4K frames, our XVFI-Net can effectively capture large motion in earlier stages and then precisely interpolate the 4K input frames better than the previous methods. It is noted that FeFlow is inappropriate for large motion alignment in the feature domain, which results in blurry output and is computationally heavy for 4K input frames. In addition, the center-frame interpolation methods such as AdaCoF, FeFlow and others [51, 13, 25, 34, 32] tend to synthesize intermediate frames generally worse than those of arbitrary time VFI methods such as DAIN and XVFI-Net as shown in Fig. 5. The errors of the center-frame interpolation methods tend to be accumulated iteratively due to inaccurate predictions. On the other hand, our model can accurately generate intermediate frames at arbitrary time t .

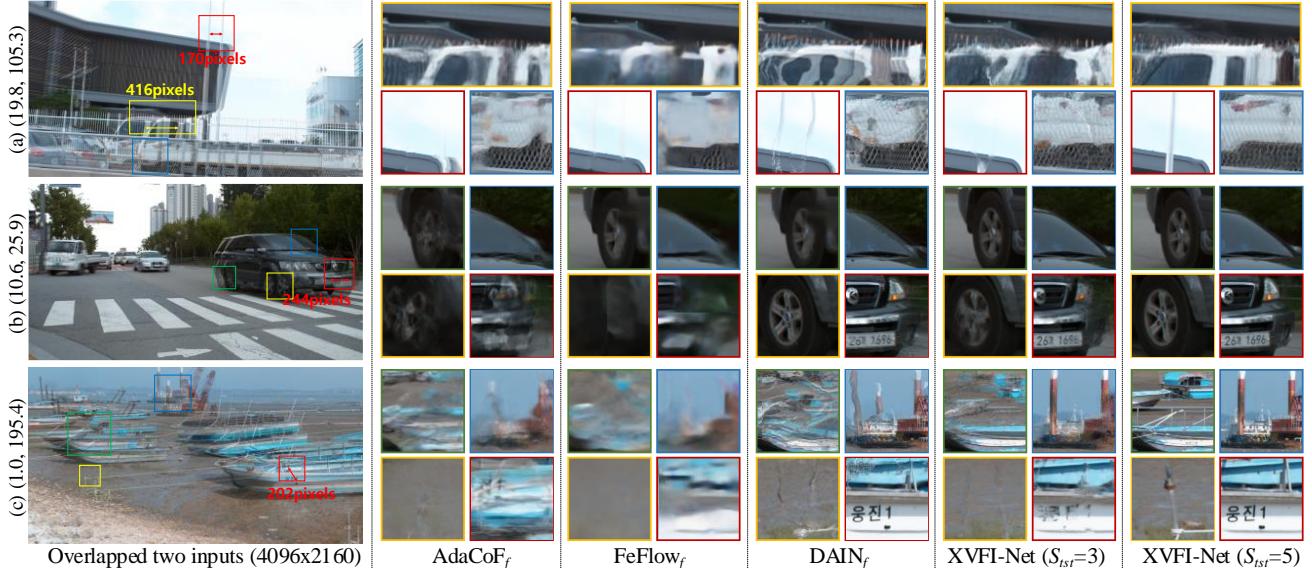


Figure 6. Visual comparisons for VFI results ($t = 0.5$) on X-TEST for our and retrained SOTA methods with X-TRAIN. (*, *): occlusions and optical flow magnitudes between the two input frames measured by [16], respectively. *Best viewed in zoom.*

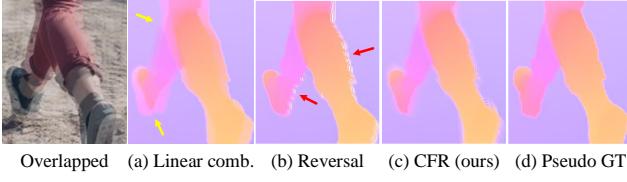


Figure 7. Approximated optical flows F_{t0} by three different flow approximation methods. (a) Linear combination, (b) flow reversal, (c) CFR (proposed). *Best viewed in zoom.*

Qualitative Comparison. Fig. 6 shows the visual comparison for VFI performances. The first column images in Fig. 6 show overlapped images of two 4K input frames. As shown, huge pixel displacements are observed between two input frames, which is very challenging for VFI. The interpolated results in Fig. 6 correspond to the center time ($t = 0.5$) of the two input frames which is the most challenging frame interpolation. As shown in Fig. 6, our XVFI-Net ($S_{tst} = 5$) surprisingly captures very complex structures of objects with extremely fast motions, which are failed by all the previous methods.

5.2. Ablation Studies

Flow Approximation. We compare the three flow approximations that enable to produce intermediate frames at arbitrary t : (a) the linear approximation [18] with F_{01}, F_{10} , (b) flow reversal [48] of F_{0t} and F_{1t} , and (c) our proposed complementary flow reversal (CFR). In this comparison, we approximated F_{t0} with the three methods using the estimated optical flows F_{01}, F_{10} which are obtained by IRR-

PWC [16] between the input I_0 and I_1 . The importance mask z 's in Eq. 1 and 2 are excluded in this comparison. Fig. 7 visualizes an example of the approximated optical flows by the three methods and the pseudo ground truth which is estimated between I_t and I_0 by IRR-PWC [16]. To evaluate the flow approximations quantitatively, the averaged endpoint errors (EPEs) for the three methods are calculated between the approximated flows and the pseudo ground truth on the testset of Vimeo90K [49], which are shown in Table 3. The linear approximation reveals misalignment due to the different anchor frames, which is indicated by yellow arrows in Fig. 7. The flow reversal resolves the misalignment problem, but is inferior to the linear approximation because it causes holes that are not projected from any flow vector, as shown in the second optical flow map (red arrows). Also, the EPE value of the flow reversal is the worst among the three methods. On the other hand, our proposed CFR can appropriately fill in the holes since the bidirectional flows complement each other, as shown in Fig. 7, which is consistent with the lowest EPE value of CFR in Table 3.

In order to investigate the efficacy of the proposed CFR for VFI, we trained three VFI models from scratch by adopting each of the three flow approximations in their BiOF-T modules, *without* any help of pretrained networks. The lowest scale depths for both training S_{trn} and test S_{tst} are set to 3. The VFI performances on our X-TEST (PSNR/SSIM/tOF) for the three models are listed in Table 3, showing a superiority of our proposed CFR.

Adjustable Scalability. The lowest scale depth S_{tst} for the inference can be adaptive to the degree of motion magnitudes and spatial resolution of the input frames, even after

Metrics \ Methods	EPE↓	PSNR↑	SSIM↑	tOF↓
(a) Linear comb.	0.0752	28.73	0.8518	2.89
(b) Flow reversal	0.0892	28.30	0.8425	2.98
(c) CFR (ours)	0.0721	28.86	0.8582	2.67

Table 3. The endpoint error (EPE) between the approximated F_{t0} and the pseudo ground truth is obtained by IRR-PWC [16] on Vimeo90K [49] testset. Note that the VFI performances are measured on X-TEST in terms of PSNR, SSIM and tOF for three models that adopt each approximation method.

S_{trn}	(PSNR(dB)↑ / SSIM[45]↑ / tOF[8]↓)		
	1	3	5
1	26.85/0.806/4.90	28.40/0.852/3.46	27.14/0.842/3.69
3	23.61/0.729/6.56	29.22/0.863/2.68	30.35/0.879/1.98
5	22.37/0.699/6.71	23.70/0.724/6.39	29.48/0.864/2.08

RED: Best performance of each row

Table 4. Ablation study on adjustable scalability depending on the lowest scale depth S_{trn} and S_{tst} measured on X-TEST.

once trained, as shown in Fig. 3. We show the adjustable scalability of our framework with $S_{trn} = 1, 3, 5$ for $S_{tst} = 1, 3, 5$. For this, we train XVFI-Net variants by fully utilizing 512×512 -sized patches because the spatial resolution of the training inputs should be multiple of 512 for $S_{trn} = 5$ where the number 512 is determined as $2^{S_{trn}=5} \times M (= 4) \times 4$ (via the bottlenecks of the autoencoders). Table 4 compares the performances of the XVFI-Net variants. As shown in Table 4, the performances are generally boosted by increasing the value of S_{tst} with the help of effectively enlarging receptive field sizes and elaborately refining the resulting flows, especially in capturing extremely large motions and detailed structures. This trend is also observed in Table 2 for the XVFI-Net trained with 384×384 -sized patches of $S_{trn} = 3$. Furthermore, as shown in the right-most four columns of Fig. 6, the details of the objects, letters and textures are more precisely synthesized for $S_{tst} = 5$ than 3 qualitatively. Both quantitative and qualitative results clearly show the effectiveness of the XVFI-Net’s adjustable scalability. On the other hand, the occlusions and flow magnitudes of the Adobe240fps dataset [40] are much smaller than those of X-TEST as shown in Table 1. It is noted in Table 2 that our XVFI-Net with $S_{tst} = 3$ shows better performance than that with $S_{tst} = 5$ on the Adobe240fps dataset with smaller resolutions than X-TEST, which also clearly supports the efficacy of our adjustable scalability.

Robustness of our XVFI-Net Framework. To show the robustness of our XVFI-Net framework for LR-LFR benchmark dataset, we construct a variant of XVFI-Net, called XVFI-Net_v, with $M = 2$ for the dataset with lower resolution frames. The XVFI-Net_v is then trained on a standard VFI dataset that is the Vimeo90K [49] training set with 51,313 triplets ($t = 0.5$) of 448×256 size. The training

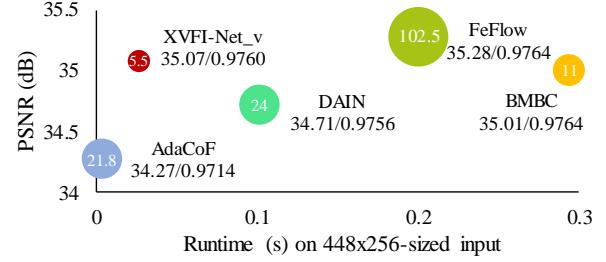


Figure 8. PSNR/SSIM vs runtime (s) on Vimeo90K [49] with model size (M) indicated in each circle.

went through 200 epochs with randomly cropped 256×256 -sized patches and a mini-batch size of 16, where both S_{trn} and S_{tst} are set to 2. We compare our XVFI-Net_v with four SOTA methods: DAIN [4], FeFlow [13], AdaCoF [25] and BMBC [33], where their pretrained models and testing code are publicly available. Fig. 8 shows the PSNR/SSIM and runtime (s) performances of our and SOTA methods with their model sizes (M) evaluated on Vimeo90K test-set. As shown, our XVFI-Net_v outperforms BMBC, DAIN and AdaCoF with a significantly smaller model size (5.4% of the number of the FeFlow’s parameters), by taking advantage of the recursive multi-scale and shared structure. However, the XVFI-Net_v shows lower performance than that of FeFlow but has a much smaller model size only with 5.4% of the number of the FeFlow’s parameters, thus leading to about $\times 7$ faster runtime. As a result, our XVFI-Net framework designed for high-resolution VFI with extremely large motion shows its robustness to the LR-LFR benchmark dataset by simply adjusting module scale factor M , S_{trn} and S_{tst} .

6. Conclusion

In this paper, we first proposed a high-quality HFR dataset in HR, called X4K1000FPS with a wide range of motions. We also proposed a VFI network, called XVFI-Net, that can handle large pixel displacements due to extremely fast motion and larger resolutions. The XVFI-Net has an adjustable scalability for inference to cope with the input resolutions or the motion magnitudes, even if training is once over. Throughout the extensive experiments with X4K1000FPS, the XVFI-Net achieved state-of-the-art performance compared to the previous VFI methods. We also showed that XVFI-Net framework, designed for high-resolution VFI, is robust to the LR-LFR benchmark dataset.

Although our proposed X4K1000FPS dataset was obtained by using one single camera, such an extreme HFR 4K dataset is very valuable for the research community of VFI because such kinds of cameras are few. Besides, we delicately select clips as X-TRAIN/X-TEST to be publicly available by considering both occlusions and flow magnitudes for a new challenging VFI task, called the eXtreme Video Frame Interpolation (XVFI). We hope that this re-

search would be a valuable milestone to extend the current VFI for more recent real-world applications with HR video.

Acknowledgement. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00419, Intelligent High Realistic Visual Processing for Smart Broadcasting Media).

References

- [1] Ha-Eun Ahn, Jinwoo Jeong, and Je Woo Kim. A fast 4k video frame interpolation using a hybrid task-based convolutional neural network. *Symmetry*, 11(5):619, 2019. [1](#), [2](#)
- [2] Ha-Eun Ahn, Jinwoo Jeong, Je Woo Kim, Soonchul Kwon, and Jisang Yoo. A fast 4k video frame interpolation using a multi-scale optical flow reconstruction network. *Symmetry*, 11(10):1251, 2019. [2](#)
- [3] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. [2](#), [3](#)
- [4] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, pages 3703–3712, 2019. [1](#), [2](#), [3](#), [6](#), [8](#), [12](#), [16](#)
- [5] Roberto Castagno, Petri Haavisto, and Giovanni Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on circuits and Systems for Video Technology*, 6(5):436–446, 1996. [1](#)
- [6] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *ECCV*, 2020. [1](#)
- [7] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, pages 10663–10671, 2020. [1](#), [2](#)
- [8] Mengyu Chu, Xie You, Mayer Jonas, Leal-Taixé Laura, and Thuerey Nils. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics*, 39(4):75–1, 2020. [6](#), [8](#)
- [9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *CVPR*, pages 764–773, 2017. [2](#)
- [10] Scott Daly. Engineering observations from spatiavelocity and spatiotemporal visual models. In *Vision Models and Applications to Image and Video Processing*, pages 179–200. Springer, 2001. [1](#)
- [11] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *CVPR*, pages 5515–5524, 2016. [1](#)
- [12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010. [6](#)
- [13] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *CVPR*, pages 14004–14013, 2020. [1](#), [2](#), [6](#), [8](#), [12](#), [16](#)
- [14] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *CVPR*, pages 2859–2868, 2020. [1](#)
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [4](#)
- [16] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, pages 5754–5763, 2019. [3](#), [7](#), [8](#), [11](#), [14](#)
- [17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015. [4](#)
- [18] Huaiyu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018. [1](#), [2](#), [4](#), [7](#)
- [19] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *ECCV*, 2020. [5](#)
- [20] Jaeyeon Kang, Younghyun Jo, Seoung Wug Oh, Peter Vajda, and Seon Joo Kim. Deep space-time video upsampling networks. In *ECCV*, 2020. [2](#)
- [21] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *AAAI*, pages 11278–11286, 2020. [1](#), [2](#)
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [6](#)
- [23] Yoshihiko Kuroki, Tomohiro Nishi, Seiji Kobayashi, Hideki Oyaizu, and Shinichi Yoshimura. A psychophysical study of improvements in motion-image quality by using high frame rates. *Journal of the Society for Information Display*, 15(1):61–68, 2007. [1](#)
- [24] Yoshihiko Kuroki, Haruo Takahashi, Masahiro Kusakabe, and Ken-ichi Yamakoshi. Effects of motion image stimuli with normal and high frame rates on eeg power spectra: comparison with continuous motion image stimuli. *Journal of the Society for Information Display*, 22(4):191–198, 2014. [1](#)
- [25] Hyeyoungmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, pages 5316–5325, 2020. [1](#), [2](#), [6](#), [8](#), [12](#), [16](#)
- [26] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. [2](#)
- [27] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017. [1](#), [2](#)
- [28] Alex Mackin, Fan Zhang, and David R. Bull. A study of subjective video quality at various frame rates. In *ICIP*, pages 3407–3411, 2015. [1](#)
- [29] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, pages 1410–1418, 2015. [1](#)
- [30] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, pages 1701–1710, 2018. [1](#), [2](#)

- [31] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, pages 5437–5446, 2020. [1](#), [2](#), [3](#), [4](#)
- [32] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *CVPR*, pages 261–270, 2017. [1](#), [2](#), [6](#)
- [33] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmhc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *ECCV*, 2020. [1](#), [2](#), [8](#), [16](#)
- [34] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *CVPR*, pages 2398–2407, 2019. [1](#), [2](#), [3](#), [6](#)
- [35] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. [2](#), [3](#)
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. [5](#), [12](#), [14](#)
- [37] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *CVPR*, pages 5114–5123, 2020. [1](#), [2](#)
- [38] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016. [14](#)
- [39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. [2](#), [3](#)
- [40] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. [2](#), [3](#), [6](#), [8](#), [11](#), [12](#)
- [41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. [2](#), [3](#)
- [42] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, pages 3360–3369, 2020. [1](#), [2](#)
- [43] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, pages 0–0, 2019. [1](#), [2](#)
- [44] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pages 4884–4893, 2018. [3](#)
- [45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. [6](#), [8](#)
- [46] Jiyan Wu, Chau Yuen, Ngai-Man Cheung, Junliang Chen, and Chang Wen Chen. Modeling and optimization of high frame rate video transmission over wireless networks. *IEEE Transactions on Wireless Communications*, 15(4):2713–2726, 2015. [1](#)
- [47] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *CVPR*, pages 3370–3379, 2020. [2](#)
- [48] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, pages 1647–1656, 2019. [1](#), [2](#), [3](#), [4](#), [7](#)
- [49] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. [2](#), [3](#), [7](#), [8](#), [11](#), [12](#), [16](#)
- [50] Songhyun Yu and Jechang Jeong. Hierarchical motion estimation algorithm using multiple candidates for frame rate up-conversion. In *IWAIT*, volume 11049, page 1104938. International Society for Optics and Photonics, 2019. [1](#)
- [51] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. In *ECCV*, 2020. [1](#), [2](#), [3](#), [5](#), [6](#)

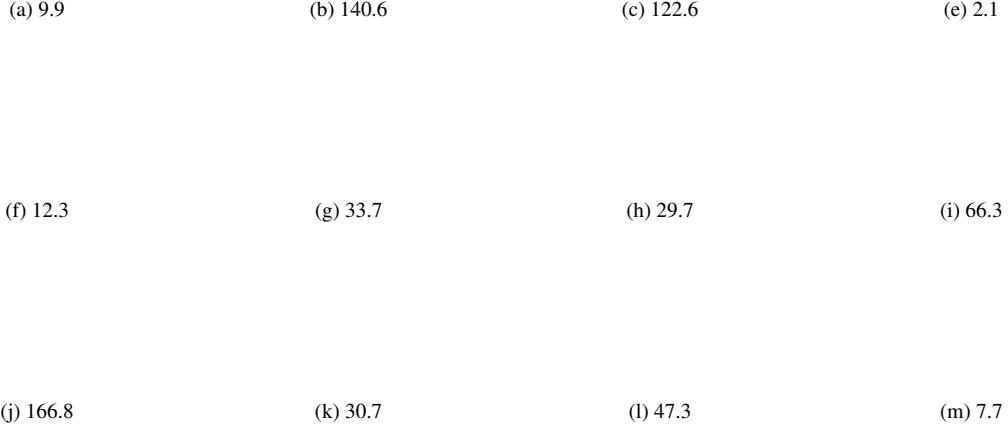


Figure 9. More examples of our X4K1000FPS dataset, which contain diverse motions in 4K-resolution of 1000 fps. The numbers below the examples are the magnitude means of optical flows between two input frames in 30 fps. This is a video figure that can be best viewed with motion using AdobeTM Reader. It should be noted that they are rendered in down-scales at 15 fps for visualization convenience.

Appendices

A. Details of Proposed X4K1000FPS Dataset

A.1. Photographing Videos

In order to provide a wide range of object motions and various camera motion types at different speeds in diverse locations, the shooting rules were guided as follows: (i) shooting various objects independently moving while the camera is stationary, (ii) shooting videos from a moving car (fast translated videos), (iii) shooting while walking (moving at normal speed), (iv) shooting with the camera in irregular motion trajectories at non-uniform speeds, (v) shooting with zooming out or in and panning at the same time. Besides, the contents of the videos also include various objects (crowds, cars, trains, plants, animals, boats, traffic signs, signboards, waterfalls, buildings, etc.) in various types of places such as stadiums, stations, beaches, markets, parks, rivers, playgrounds, etc. Fig. 9 shows some representative thumbnails of spatiotemporally down-sampled 4K video at 15 fps for a visualization purpose. As shown in Fig. 1 and Fig. 9 in this Supplementary Material, very extreme scenes including various camera motions, zooming, translations, speeds, occlusions and objects are contained in the X4K1000FPS dataset.

A.2. Test Dataset: X-TEST

We manually select the consecutive 32 frames for each test scene by considering the degrees of occlusion, optical flow magnitudes and diversity among 5,000 frames. We compose nonuplets by sampling every 4 frames from 32 frames of each test scene. Since the frame rates of videos are often given in multiples of 30 in VFI benchmark datasets [40, 49] or real-world industries, we approximate that our test videos are set to 960 fps ($= 32 \times 30$ fps) instead of 1,000 fps. Therefore, two input frames that are 32 frames apart are regarded as part of a 30 fps ($= 960 / 32$) video and converted to 240 fps by $\times 8$ multi-frame interpolation in the evaluation phase.

A.3. Train Dataset: X-TRAIN

To compose a valuable training dataset from our enormous videos for video frame interpolation, we select training samples based on the value of the occlusion map estimated by IRR-PWC [16]. The occlusion maps are approximated on the spatially down-sampled ($\times 1/4$) and temporally sub-sampled ($\times 1/32$) frames of the original 4K-resolution 1000 fps videos. Each target frame is fed into IRR-PWC with the previous and the next frames of the target frame, respectively. The resulting two occlusion maps are averaged to get the bidirectional occlusion map.

Then, we divide the 4K-resolution frame into overlapping patches of 768×768 , which forms an 81×31 grid, except for the boundary of the 4K-resolution frame. This is because the boundary patches have undesirably large occlusion values when there are translation motions. Similarly,

about 5,000 frames are divided into overlapping 154 clips of 65 consecutive frames except for the boundary period in the temporal dimension of the scenes. Thus, about 386K ($= 81 \times 31 \times 154$) candidate training samples per scene of the patch size of 768×768 and the lengths of 65 frames are extracted from a 4K-resolution 1000 fps video. After that, the candidate training samples whose bidirectional occlusion value is the top 10% of those of all candidate training samples remained, and the others are discarded. Finally, total 4,408 training samples are sparsely selected as training data to prevent similar samples from being selected to maintain the diversity of the training samples.

B. Details of Architecture of XVFI-Net

In addition to Fig. 3 and 4, we present the detailed architectures of sub-networks of XVFI-Net in the case of the module scale factor $M = 4$ in Table 5 to Table 9. The series of rows represents the consecutive operations. The first column represents each layer’s operation, and H, W and C indicate the spatial ratio with respect to bicubically downsampled ($\times 1/2^s$) input frames I_i^s for each scale s and the number of channels of the output tensors, respectively. The last column denotes the names of some output tensors, which are worth mentioning. We omit the names of the output tensors if they are just intermediate tensors in the sub-networks. When the multiple tensors are input to each layer, they are concatenated channel-wise. ‘resblock’ represents a residual block which consists of *conv2d - relu - conv2d - identity addition*. The stride of the convolutional layer is set to 1, if not mentioned. The convolution filter sizes are 3×3 and 4×4 for the strides of 1 and 2, respectively.

As shown in Table 5, the Feature Extraction Block is a simple residual block-based sub-network. On the other hand, the flow estimation sub-networks, the BiFlownet and TFlownet, have a simple auto-encoder architecture to enlarge the receptive field as shown in Table 6, 7 and 8. The Refinement Block has a U-Net [36]-based architecture as in Table 9. The parameters of each sub-network are shared for all scale levels except for the BiFlownet at the lowest scale depth S , which is isolated in Table 6. The bidirectional flows are estimated directly from two input features C_0^S, C_1^S at the lowest scale level S , because there does not exist any provided initial flow.

Efficiency of XVFI-Net During Inference. The BiOF-T module can start from any down-scaled level, while the BiOF-T module can be skipped in the down-scaled levels ($s = 1, \dots, S_{tst}$) as described in Fig. 3. By doing so, our XVFI-Net framework can accelerate run time about 22% faster compared to the full-recursion framework where both BiOF-I and -T modules are processed together in all scale levels for 4K video, when $S_{tst} = 5$. Besides, the additional runtimes induced by the smaller down-scaled levels ($s > 0$) are negligible since the runtimes of the BiOF-I module at

down-scaled levels are much smaller than those of BiOF-I and BiOF-T modules at the original scale ($s = 0$).

C. Visual Comparison with Videos

We provide a 4K comparison video for AdaCoF_f [25], FeFlow_f [13], DAIN_f [4] and XVFI-Net with $S_{tst} = 5$ as in a video uploaded on our project page on four scenes of X-TEST. The input frames have 4K resolution of 30 fps and are converted to 240 fps by each VFI method. All results are encoded at 30 fps to be played as *slow motion* for the interpolated results on multi-frame interpolation $\times 8$. It should be noted that four 4K video results of the methods are simultaneously shown at a single screen of 4K, each of which is played at 2K resolution (down-scaled for easier comparison).

As shown in the first scene with even extreme back and forth motions of a propeller with zoom-out, our XVFI-Net can surprisingly capture such a complex motion for VFI, but the SOTAs fail to interpolate the sophisticated tips of the propeller pointed by the yellow arrows. For the second scene, even while riding a fast-moving car, XVFI-Net better captures far tiny structures such as electric wires seen at the left part and a closer pole with a large pixel displacement pointed by the yellow arrows. For the third scene, the right-most front car moves very fast, so all the previous methods fail to capture it, denoted by the yellow arrow, yielding severe artifacts (structural distortions). On the other hand, XVFI-Net precisely captures the especially right edges of the rightmost car. Finally, in the last scene even with the *extremely* hand-shaken frames, the XVFI-Net can also synthesize repeating similar stairs but all SOTAs tend to generate baggy artifacts. As a result, XVFI-Net significantly better handles large pixel displacements due to extreme motion and huge spatial resolutions.

D. Additional Qualitative Results

Visual Comparisons for VFI methods. We provide additional qualitative results on X-TEST (4K) in Fig. 10, Adobe240fps [40] (HD) in Fig. 11 and Vimeo90K [49] in Fig. 12 by the each setting.

Visualization of Components of XVFI-Net. Fig. 13 shows the visualization of optical flows and occlusion masks of XVFI-Net_v. As expected, estimated flows at the upper level seem finer than those at the lower level (F_{t0}^1) as shown in Fig. 13. The coarse-to-fine structure gradually helps the whole XVFI framework boost the final VFI performance at original scale $s = 0$ based on the occlusion masks and the iteratively updated flows that are all *learned from scratch*.

Operation	H,W	C	Remarks
input I_i^0	1	3	I_i^0
conv2d - relu	1	64	-
conv2d - relu	1/2	64	-
conv2d	1/4	64	Feat _i
resblock ($\times 2$) - add to Feat _i	1/4	64	C_i^0
conv2d ($\times s$)	$1/4 \times 1/2^s$	64	C_i^s

Table 5. The detailed architecture of the Feature Extraction Block of XVFI-Net. C_i^s is obtained by applying the last convolutional layer to C_i^0 s times recurrently. The parameters are temporally shared for the two input frames ($i = 0, 1$).

Operation	H,W ($\times 1/2^S$)	C	Remarks
input $[C_0^S, C_1^S]$	1/4	64×2	-
conv2d (stride 2) - relu	1/8	128	-
conv2d (stride 2) - relu	1/16	256	-
NN upscale - conv2d - relu	1/8	128	-
NN upscale - conv2d - relu	1/4	64	-
conv2d	1/4	2+2+1+1	$F_{01}^S, F_{10}^S, z_{01}^S, z_{10}^S$

Table 6. The detailed architecture of the auto-encoder-based BiFlownet of XVFI-Net at the lowest scale depth.

Operation	H,W ($\times 1/2^s$)	C	Remarks
input $[C_0^s, \tilde{C}_{01}^s]$	1/4	64×2	-
conv2d	1/4	64	\hat{C}_{01}^s
input $[C_1^s, \tilde{C}_{10}^s]$	1/4	64×2	-
conv2d	1/4	64	\hat{C}_{10}^s
input $[\hat{C}_{01}^s, \hat{C}_{10}^s, \tilde{F}_{01}^s, \tilde{F}_{10}^s]$	1/4	$64 \times 2 + 2 \times 2$	-
conv2d (stride 2) - relu	1/8	128	-
conv2d (stride 2) - relu	1/16	256	-
NN upscale - conv2d - relu	1/8	128	-
NN upscale - conv2d - relu	1/4	64	-
conv2d - add to $[\tilde{F}_{01}^s, \tilde{F}_{10}^s]$	1/4	2+2+1+1	$F_{01}^s, F_{10}^s, z_{01}^s, z_{10}^s$

Table 7. The detailed architecture of the auto-encoder-based BiFlownet of XVFI-Net except for the lowest scale depth.

Operation	H,W ($\times 1/2^s$)	C	Remarks
input $[C_0^s, C_1^s, \tilde{C}_{t0}^s, \tilde{C}_{t1}^s, \tilde{F}_{t0}^s, \tilde{F}_{t1}^s]$	1/4	$64 \times 4 + 2 \times 2$	-
conv2d (filter 1×1) - relu	1/4	64	-
conv2d (stride 2) - relu	1/8	128	-
conv2d (stride 2) - relu	1/16	256	-
NN upscale - conv2d - relu	1/8	128	-
NN upscale - conv2d - relu	1/4	64	-
conv2d - add to $[\tilde{F}_{t0}^s, \tilde{F}_{t1}^s]$	1/4	2+2+1	F_{t0}^s, F_{t1}^s, m^s

Table 8. The detailed architecture of the auto-encoder-based TFlownet of XVFI-Net.

Operation	H,W ($\times 1/2^s$)	C	Remarks
input ($[C_0^s, C_1^s, \tilde{C}_{t0}^s, \tilde{C}_{t1}^s]$) pixel-shuffle [38] ($\uparrow 4$)	1/4	256	-
	1	16	PS
input [PS, $F_{t0}^s \uparrow 2, F_{t1}^s \uparrow 2, I_0^s, I_1^s, \tilde{I}_{t0}^s, \tilde{I}_{t1}^s]$ conv2d (stride 2) - relu conv2d (stride 2) - relu conv2d (stride 2) - relu conv2d - relu NN upscale - concat to enc ₂ conv2d - relu NN upscale - concat to enc ₁ conv2d - relu NN upscale - conv2d	1	16+2×2+3×4	-
	1/2	64	enc ₁
	1/4	128	enc ₂
	1/8	256	-
	1/8	256	-
	1/4	384	-
	1/4	128	-
	1/2	192	-
	1/2	64	-
	1	1+3	m^s, \tilde{I}_r^s

Table 9. The detailed architecture of the U-Net [36]-based Refinement Block of XVFI-Net.

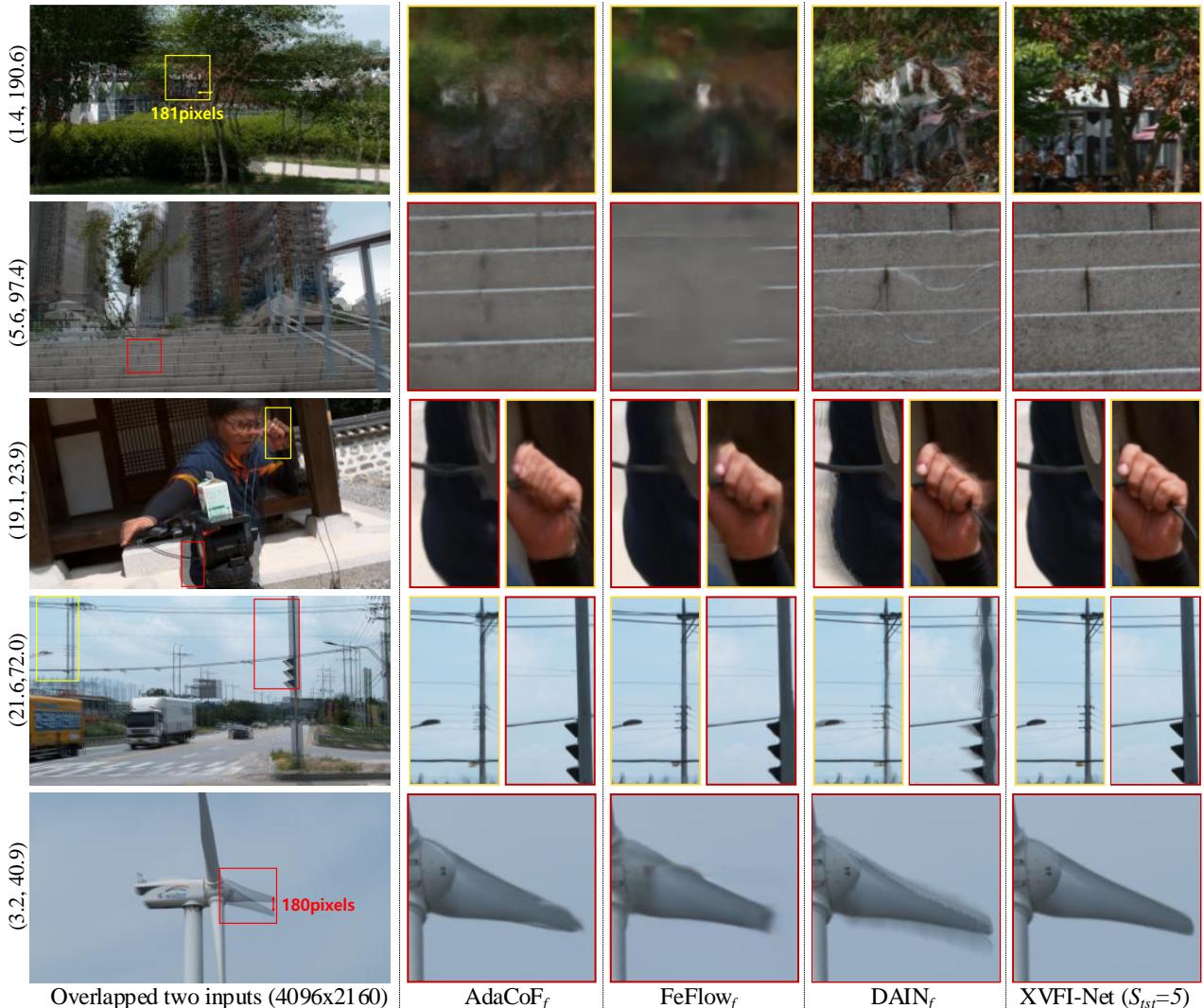


Figure 10. Visual comparisons for VFI results ($t = 0.5$) on X-TEST for our and *retrained* SOTA methods with X-TRAIN. (*, *): occlusions and optical flow magnitudes between the two input frames measured by IRR-PWC [16], respectively. *Best viewed in zoom*.



Figure 11. Visual comparisons for VFI results ($t = 0.5$) on Adobe240fps for our and *retrained* SOTA methods with X-TRAIN. *Best viewed in zoom.*

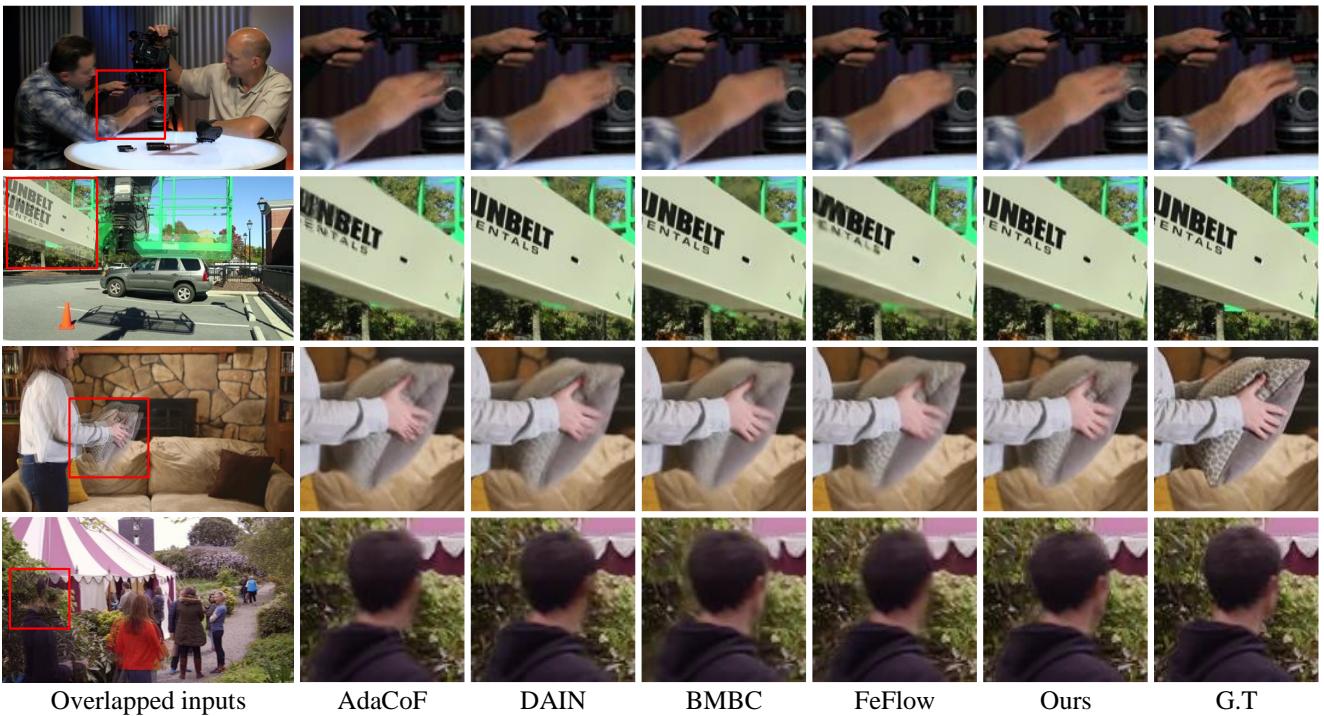


Figure 12. Visual comparisons of AdaCoF [25], DAIN [4], BMBC [33], FeFlow [13], our XVFI-Net_v, and the corresponding ground truth on the testset of Vimeo90K [49] triplet. *Best viewed in zoom.*

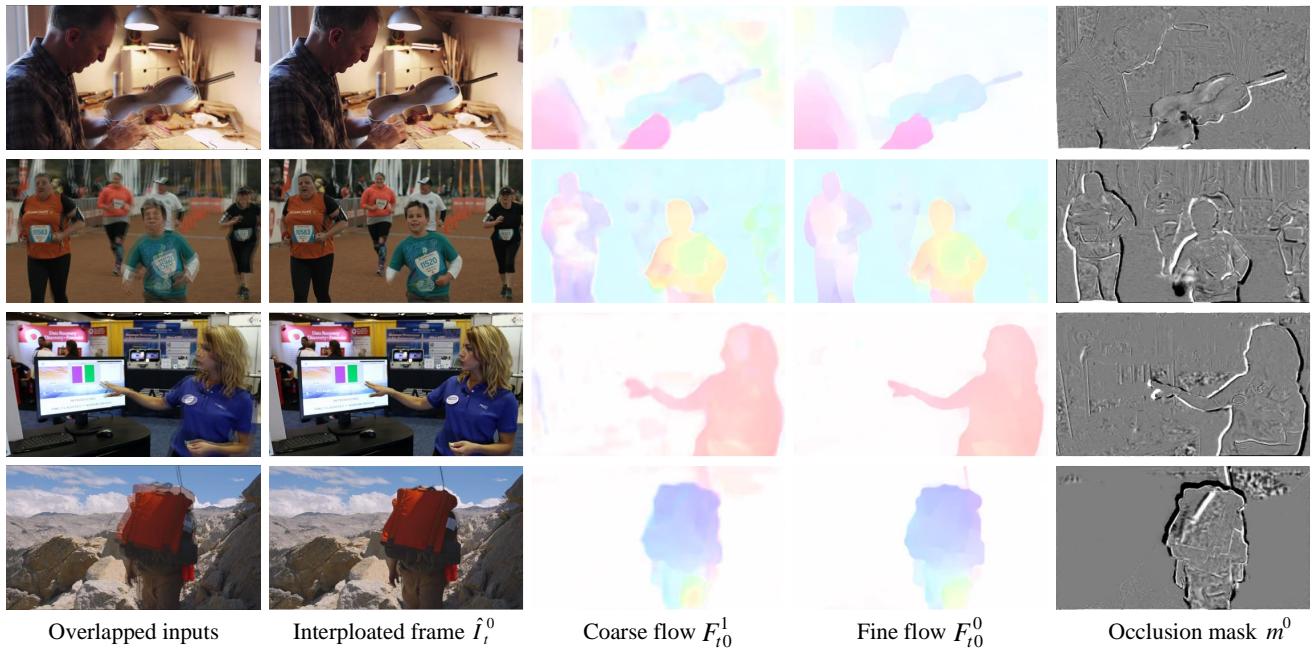


Figure 13. Visualization of optical flows and occlusion masks of XVFI-Net_v. The coarse and fine flows are extracted at scale 1 and 0, respectively.