EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

Video frame interpolation for high dynamic range sequences captured with dual-exposure sensors

U. Çoğalan M. Bemana HP. Seidel and K. Myszkowski

Max-Planck-Institut für Informatik, Germany

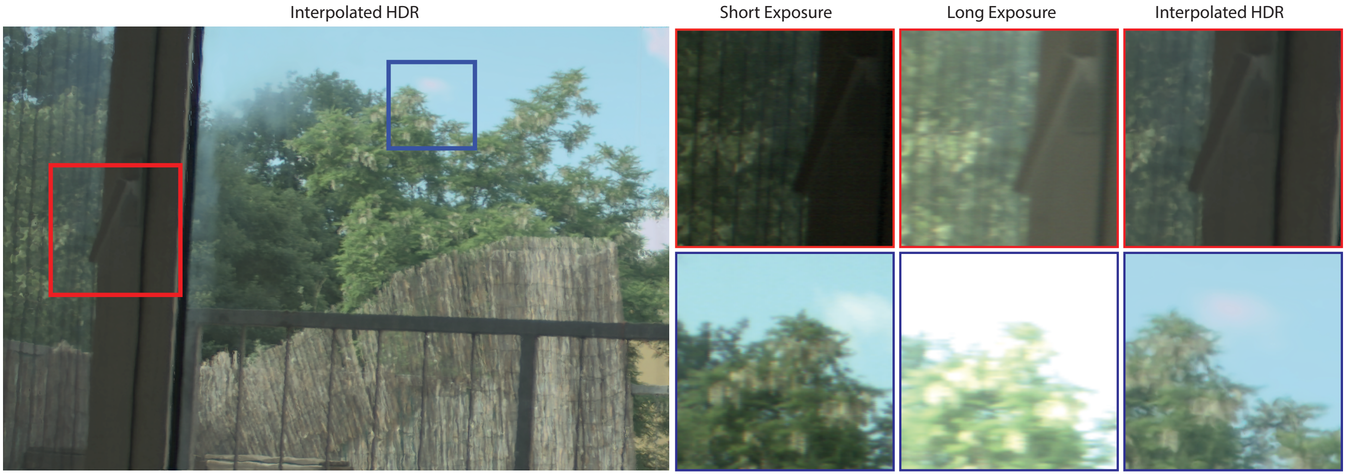

Figure 1: *We propose a method for high dynamic range (HDR) video frame interpolation (VFI) for dual-exposure sensors that gain on popularity due to their use in recent smartphones. The first column shows interpolated HDR frame, while the insets focus on the dark and bright scene details. Note that the short and long exposures, as captured by the sensor (middle columns), are shifted with respect to the interpolated HDR frame (right column). The dark region (the upper row) requires a long exposure duration and features significant motion blur due to camera motion. Our method employs temporally continuous information on the scene motion that is encoded in motion blur to improve the VFI quality. At the same time, the short exposure avoids pixel saturation in the sky region (the bottom row) and enables its reconstruction in the interpolated HDR frame.*

Abstract

Video frame interpolation (VFI) enables many important applications such as slow motion playback and frame rate conversion. However, one major challenge in using VFI is accurately handling high dynamic range (HDR) scenes with complex motion. To this end, we explore the possible advantages of dual-exposure sensors that readily provide sharp short and blurry long exposures that are spatially registered and whose ends are temporally aligned. This way, motion blur registers temporally continuous information on the scene motion that, combined with the sharp reference, enables more precise motion sampling within a single camera shot. We demonstrate that this facilitates a more complex motion reconstruction in the VFI task, as well as HDR frame reconstruction that so far has been considered only for the originally captured frames, not in-between interpolated frames. We design a neural network trained in these tasks that clearly outperforms existing solutions. We also propose a metric for scene motion complexity that provides important insights into the performance of VFI methods at test time.

CCS Concepts

• ***Computing methodologies*** → ***Computational photography;*** *Image processing;*

1. Introduction

Video frame interpolation (VFI) enables many interesting applications ranging from video compression and framerate up-conversion in TV broadcasting to artistic video effects such as speed ramp in professional cinematography. The performance of VFI methods is largely affected by various factors such as scene lighting conditions, the magnitude and complexity of motion in the scene, the spatial extension of resulting motion blur, the presence of complex occlusions, or thin structures in the scene. Popular VFI methods [JSJ*18, BLM*19, SOK21] mostly rely on well-exposed frames in the captured video. Nevertheless, in the case of high dynamic range (HDR) scenes captured using traditional single-exposure sensors, undesired under- and over-exposure effects might appear. The resultant noise and intensity clamping can adversely affect the quality of VFI as finding the pixel correspondence between the frames becomes more ambiguous. Another major challenge is the large and non-uniform motion in the scene. Although recent methods [RKT*22, SOK21] have shown progress in handling large motion, they typically heavily rely on the motion linearity assumption that might not hold in practice. Explicit handling of non-linear motion becomes possible by processing more than two subsequent frames [XSS*19, PLK21]; however, temporal sampling might still be too low for reliable motion reconstruction. Motion blur due to low shutter speed and long exposure times further leads to spatial and temporal loss of image details. For this reason, handling blurry frames is typically treated as a challenge in the VFI task [SBZ*20a, ZWT20], while potentially, motion blur encodes continuous temporal information on the magnitude and direction of motion, particularly for large motion.

Programmable sensors with spatially varying exposures greatly expand the dynamic range of contrast in captured video [HKU14, GKSK19, CBK17, HST*14, CBM*22, CKL14], and become an attractive choice for modern smartphones [GSM22], e.g. Sony's Quad Bayer [Son22], and Samsung's Tetracell/Nonacell [Sam22] technologies. In this work, we explore such sensor capabilities toward improving the motion estimation accuracy in VFI. In particular, we consider a dual-exposure sensor that captures short and long exposures for spatially interleaved pixel columns in a single shot [CMV21]. Importantly, while the exposure duration differs, the exposure completion is temporally aligned, which enables recovering two temporal samples of the scene motion that are perfectly spatially registered at the sensor. We show that such an increased temporal sampling rate substantially improves the accuracy of complex motion interpolation, as motion non-linearity can readily be reconstructed for two subsequent frames. Furthermore, the short exposure typically leads to a sharp image, while the long exposure results in substantial motion blur that provides additional insights into the motion direction and magnitude (Fig. 1). This is of particular importance in dark scene regions, where the short exposure might be strongly underexposed and noisy, and the long exposure becomes the only reliable measurement of scene motion. As in other works, we employ a multi-exposure technique to reconstruct HDR video frames, but for the first time, we simultaneously perform VFI that can handle complex, non-linear motion in the scene. We train an end-to-end convolutional network to achieve those goals. We also propose a metric of motion non-linearity that allows us to analyze the existing high-speed videos and measure the performance of VFI methods as a function of motion complexity.

The key contributions of our work are:

- We propose a compact machine learning solution for VFI that can handle HDR content and complex non-uniform motion, enabled by deriving two temporal samples of the scene motion for each frame by joint processing of short and long exposures as captured using a dual-exposure sensor.
- We adopt a PWC-Net architecture to estimate the motion flow from motion blur in the long exposure that, in our setup, is uniquely supported by sharp image content in the short exposure. Spatial registration of both exposures and temporal alignment of their ends greatly improves the motion flow accuracy.
- We develop a metric of motion complexity that provides interesting insights into existing datasets used in the training of VFI methods and enables us to evaluate the performance of those methods for different levels of motion non-linearity.

In the following section, we discuss previous work, and in Sec. 3, we present our VFI method for HDR sequences. In Sec. 4 we introduce our metric of scene motion uniformity that enables meaningful comparison of existing VFI methods while Sec. 5 provides implementation details of our network. Sec. 6 contrasts our technique with existing works in a performance comparison and reports an outcome of ablation studies. Finally, we conclude this work in Sec. 7.

2. Previous work

In this section, we discuss existing VFI methods dealing with sharp input video (Sec. 2.1), considering either a uniform or non-uniform motion assumption. We focus on the problems of recovering motion from the blur (Sec. 2.2), joint deblurring and VFI (Sec. 2.3), and HDR video reconstruction (Sec. 2.4) that are central to this work. We refer the reader to recent surveys where more complete treatments of deep VFI [PVPA21] and HDR video [WY21] solutions are presented.

2.1. Sharp video frame interpolation

A vast majority of existing VFI techniques assume that the motion in the input video is uniform, but there are also methods explicitly designed without this assumption.

Uniform motion SepConv [NML17] merges flow estimation and frame warping into a single convolution step. They predict spatially-varying 1D kernels and convolve with them input frames to interpolate new frames. SuperSlowMo [JSJ*18] uses bi-directional flows and an occlusion map to synthesize intermediate frames at arbitrary times. DAIN [BLM*19] utilizes additional interpolation kernels and depth maps for blending the input frames. A cycle consistency loss is introduced to learn frame interpolation with fewer training pairs [LLLC19], or without any supervision, [RSD*19]. BMBC [PKLK20] warps the input frames with a proposed bilateral motion model and combines them using learned dynamic blending filters. CAIN [CKH*20] uses a channel attention module to interpolate video frames without the need for estimation of motion. SoftSplat [NL20] proposes differentiable forward warping via softmax splatting and shows its benefits for VFI. AdaCoF [LKC*20] proposes a warping module in which a target pixel can refer to not only one but many pixels at any location in the reference. XVFI [SOK21] presents a high-speed (1000fps) video dataset and proposes a multi-scale recursive approach to handle large motion in the scene. Recently, FILM [RKT*22] has introduced a unified framework that achieves superior results for large and complex motions by balancing the motion range distribution in the training dataset. For all methods discussed here, a combination of large and strongly non-uniform motion might lead to highly objectionable artifacts.

Non-uniform motion QVI [XSS*19] is one of the first video interpolation methods to model curvilinear motion with the quadratic equation using four temporal frames. Chi et al. [CMNL*20] extend QVI by introducing an additional cubic term that accounts for the change in acceleration. ABME [PLK21] handles the non-uniform motion in the scene by extending the BMBC [PKLK20] for asymmetric bilateral motion between input frames. In all those methods, more than two consecutive frames are required to capture the motion non-uniformity that, for large and complex motions might be challenging, both because of temporal sampling deficits as well as overall reduced flow estimation accuracy. In our approach, we capture two exposures in a single frame that increase the sampling rate twice, and we employ motion blur inherent to the longer exposure as an additional cue to the flow estimation.

2.2. Motion flow reconstruction from motion blur

A combination of longer exposure times and rapid motion in the scene or camera might lead to visible motion blur that typically is considered degradation and eliminated using the dedicated image and video deblurring solutions. We refer the reader to extensive surveys on this topic [KLY21, ZRL*22], and we focus our discussion on deblurring solutions that explicitly recover intra-frame optical flow from motion blur that we employ in this work. Earlier works [Rek95, SSR09] assume global motion models that lead to spatially-invariant deblurring kernels. More advanced solutions support spatially-varying kernels that are approximated by linear motion [HKML14, DW08]. Gong et al. [GYL*17] propose a deep-learning approach to handle heterogeneous blur; however, they simulate motion flows with a set of constrained flow magnitudes and directions to generate the training pairs. Argaw et al. [AKR*21] alleviate this issue by deploying available synthetic and real scene blur datasets without any restrictive motion assumptions and estimating a dense optical flow directly from motion blur in the image. However, their estimation may be subject to ambiguity in predicting the correct direction of flow, which is crucial in our case. Beyond restoring latent sharp images, a joint estimate of the 3D shape and motion are feasible, but highly motion-blurred images are required [QWMT19, ROFP22]. While these methods aim to recover the motion flow from blur, we can not apply them right away, as they assume that the input blurry image is mostly well-exposed, while we have a considerable amount of saturated pixels in the blurry long exposure. We deal with this problem using the sharp short exposure that also enables bypassing the task of image deblurring.

2.3. Joint video deblurring and interpolation

Recent works demonstrate that joint deblurring and frame interpolation greatly improves the resulting VFI quality over an independent treatment of these tasks. Jin et al. [JHF19] adopt a joint optimization scheme to extract sharp keyframes within a frame by processing four consecutive blurry frames and then smoothly interpolating the in-between frame using the extracted keyframes. Shen et al. [SBZ*20a, SBZ*20b] simultaneously remove the motion blur and interpolate the in-between frames by employing a recurrent pyramid framework to efficiently aggregate the temporal information. Gupta et al. [GARC20] relax the strong assumption that all the input frames in a captured video are blurry and adapt attention mechanisms to decide on deblurring each frame based on the information from the neighbor frames. While these methods mainly attempt to remove the motion blur in the VFI task, the inherent motion blur, as we discuss in Sec. 2.2, can potentially reveal information about the magnitude and direction of the motion, especially in the case of large non-uniform motion. Along these lines Zhang et al. [ZWT20] propose a VFI solution that is the closest to our work. They first extract two sharp keyframes corresponding to the start and the end of a blurry frame, and then by taking two consecutive frames, they compute the optical flow between the resulting four keyframes. By employing a quadratic motion formulation, they can handle non-uniform motion. However, in this approach, the inaccuracy in predicting the keyframes affects the quality of the flow estimation, which in turn is prone to error, especially for large motion, whereas we benefit from the less blurred short exposure in each frame to make the flow estimation more reliable. This allows us to consider more intra- and inter-frame flows that are independently estimated, and we carry our processing across subsequent stages of our multi-network pipeline using a multiresolution approach. Also, we uniquely support HDR VFI, so we need to deal with extensive saturation regions in the blurry long exposure.

2.4. HDR video reconstruction

HDR video reconstruction is typically performed using multi-exposure techniques, where subsequent frames with temporally interleaved different exposures are combined, and their dynamic content is aligned, typically using optical flow methods [KSB*13, KR*17, KR19, YZL*20, CCG*21]. Disparity information can be used for such alignment in stereo cameras or dual-lens systems that are widely used in modern smartphones [LC09, CYC*19, CJY*20, DHL*21]. To alleviate the need for such alignment, single-shot HDR techniques are developed that rely on specialized dual-ISO/dual-gain sensors that require larger photosites to reduce the photon noise as typically short exposures are captured to avoid highlight clipping [HKU14, GKSK19, CBK17, CA20]. Dedicated hardware solution such as coded sensors using spatially-varying optical mask [SHG*16, AFTH19] can also enable HDR imaging with only a single-shot. Multi-exposure sensors [CMV21, Son22, Sam22] that, as we show in this work, are greatly beneficial for HDR VFI, require motion deblurring in longer exposures to reconstruct sharp HDR video [HST*14, CKL14, JCJG21]. This task is relatively easy for machine learning solutions [CBM*22], where recently proposed neural sensors [MMC*20, NMW22] can learn spatially varying pixel exposures for efficient motion deblurring. The scope of all these methods is mainly limited to HDR video reconstruction, and they do not aim for the VFI task. An exception here is the work of Rebecq et al. [RRKS19], where high framerate HDR video is reconstructed using a highly specialized event camera that, in a frameless manner, asynchronously responds to per-pixel brightness changes.

3. Method

In this section, we propose a VFI method that reconstructs HDR frames in the continuous-time domain. Fig. 2 summarizes our pro-

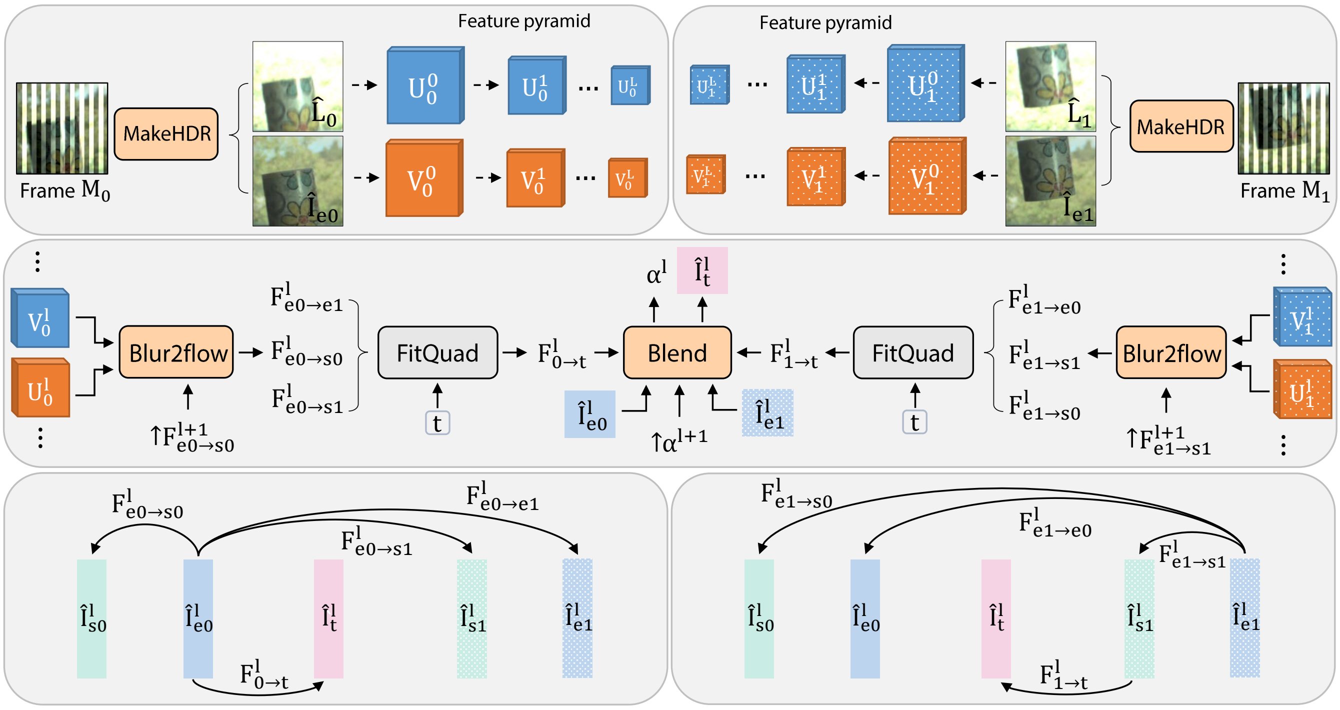

Figure 2: *Overview of our HDR VFI pipeline.* Upper row: *Two subsequent frames $\mathtt{M}_0$ and $\mathtt{M}_1$, as captured using our dual-exposure sensor, are independently processed by a learned* **MakeHDR** *network, so that the output sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{ei}}$ aligned with the end of the long exposure (the suffix* **e** *stands for the end) and blurry long exposure frames $\hat{\mathtt{L}}_{\mathtt{i}}$ are obtained. Next, each frame is fed separately to a feature extractor to build feature pyramids.* Middle row: *At each pyramid scale l, given the features $\mathtt{V}_{\mathtt{i}}^{\mathtt{l}}$ and $\mathtt{U}_{\mathtt{i}}^{\mathtt{l}}$ along with $\uparrow \mathtt{F}_{\mathtt{ei}\to\mathtt{si}}^{\mathtt{l+1}}$ upsampled from the previous scale, the intra-frame flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}^{\mathtt{l}}$, the flow between the start (denoted with the suffix* **s***) and end of the long exposure in each frame is recovered using a learned* **Blur2Flow** *network. We then find bidirectional flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{e1}}^{\mathtt{l}}$ and $\mathtt{F}_{\mathtt{e1}\to\mathtt{e0}}^{\mathtt{l}}$ estimated between $\hat{\mathtt{I}}_{\mathtt{e0}}^{\mathtt{l}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}^{\mathtt{l}}$ (which are the sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ down-sampled by 2^l) using the state-of-the-art flow estimation method Raft [TD20]. Next, given two estimated flows for each frame, we also derive additional flows of $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}^{\mathtt{l}}$ and $\mathtt{F}_{\mathtt{e1}\to\mathtt{s0}}^{\mathtt{l}}$. The motion flow triplets ($\mathtt{F}_{\mathtt{e0}\to\mathtt{e1}}^{\mathtt{l}}$, $\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}^{\mathtt{l}}$, $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}^{\mathtt{l}}$) as well as ($\mathtt{F}_{\mathtt{e1}\to\mathtt{e0}}^{\mathtt{l}}$, $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}^{\mathtt{l}}$, $\mathtt{F}_{\mathtt{e1}\to\mathtt{s0}}^{\mathtt{l}}$) are independently fed to a non-learnable* **FitQuad** *module to calculate the forward flows $\mathtt{F}_{\mathtt{0}\to\mathtt{t}}^{\mathtt{l}}$ and $\mathtt{F}_{\mathtt{1}\to\mathtt{t}}^{\mathtt{l}}$ that are parametrized using a quadratic motion model for a position t (refer to the two bottom insets). Finally, using the module* **Blend***, we fuse $\hat{\mathtt{I}}_{\mathtt{e0}}^{\mathtt{l}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}^{\mathtt{l}}$ with the forward flows $\mathtt{F}_{\mathtt{0}\to\mathtt{t}}^{\mathtt{l}}$ and $\mathtt{F}_{\mathtt{1}\to\mathtt{t}}^{\mathtt{l}}$ and a soft occlusion map $\uparrow \alpha^{\mathtt{l+1}}$ upsampled from the previous scale to reconstruct the intermediate frame $\hat{\mathtt{I}}_{\mathtt{t}}^{\mathtt{l}}$ at scale l. We repeat this procedure until we reach to the scale of the original input frames.* Bottom row: *A schematic presentation of all involved flows and their relation to the input and interpolated frames.*

cessing pipeline, and the following paragraphs provide a more detailed description of its key components. Our method takes as input two subsequent video frames $\mathtt{M}_0$ and $\mathtt{M}_1$ that are captured using our dual-exposure sensor and produces a sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{t}}$ for any position t between $\mathtt{M}_0$ and $\mathtt{M}_1$. Each captured frame $\mathtt{M}_{\mathtt{i}}$, where with the suffix i we denote any input frame, contains a pair of spatially interleaved short and long exposures and is processed by the **MakeHDR** network to produce a sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$ that is aligned with the end of the long exposure (the suffix **e** stands for the end), and a blurry long exposure frame $\hat{\mathtt{L}}_{\mathtt{i}}$. Both frames are decomposed into their respective multi-resolution feature pyramids, and from this stage, the whole processing is performed at different scales, where as shown in the middle row in Fig. 2, information reconstructed at a lower-resolution scale $l+1$ contributes to the higher-resolution scale l. Here, for brevity, we omit the scale index l. The feature pyramids are fed to the **Blur2Flow** network to predict the flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}$ that extracts the flow between the start (denoted with the suffix **s**) and end of the long exposure.. Next, we compute the flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{e1}}$ and $\mathtt{F}_{\mathtt{e1}\to\mathtt{e0}}$ between the sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ in both directions using an off-the-shelf flow estimation method such as Raft [TD20]. This way, we obtain the flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}$ and $\mathtt{F}_{\mathtt{e0}\to\mathtt{e1}}$ that are aligned with $\hat{\mathtt{I}}_{\mathtt{e0}}$, then we additionally derive the flow $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$, and employ all three flows to fit a quadratic motion model using a non-learnable **FitQuad** module. We repeat this process for the flows $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}$, $\mathtt{F}_{\mathtt{e1}\to\mathtt{e0}}$, and $\mathtt{F}_{\mathtt{e1}\to\mathtt{s0}}$ that are aligned with $\hat{\mathtt{I}}_{\mathtt{e1}}$. Refer to the bottom row in Fig. 2 for the depiction of the discussed flows. Next, to warp the keyframes $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ to a novel temporal position t, we first find the forward flows $\mathtt{F}_{\mathtt{0}\to\mathtt{t}}$ and $\mathtt{F}_{\mathtt{1}\to\mathtt{t}}$ and then compute the backward flows $\mathtt{F}_{\mathtt{t}\to\mathtt{0}}$ and $\mathtt{F}_{\mathtt{t}\to\mathtt{1}}$ using differentiable flow reversal as introduced in [XSS*19]. Finally, using a multi-scale blending scheme **Blend**, we combine the warped images with a soft occlusion weight at different scales to synthesize the frame $\hat{\mathtt{I}}_{\mathtt{t}}$. We now provide more details on all the processing steps discussed here.

HDR reconstruction: MakeHDR We acquire our input video using a dual-exposure sensor [CMV21] that simultaneously captures a short and long exposure for each frame. In our setup, the exposure time for the long exposure is four times higher than the short exposure. Each exposure is stored at odd and even columns in the sensor. As a result, both exposures are provided as half-resolution images, and they need to be up-sampled in the horizontal direction. Moreover, the short exposure exhibits strong noise in dark scene regions and requires denoising. On the other hand, the long exposure is less noisy, while it might contain considerable motion blur and requires deblurring. To do so, we employ the network design and the training strategy introduced in [CBM*22] to jointly deblur, denoise, and upsample our input frames $\mathtt{M_i}$ to produce sharp, clean, and full-resolution short and long exposures. Both exposures are combined using a non-learnable technique, similar to [DM08], to produce a sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$. We also extend the network output to produce an additional full-resolution blurry long exposure $\hat{\mathtt{L}}_{\mathtt{i}}$.

Motion from blur: `Blur2Flow` As we discuss in Sec. 2.2, motion blur can potentially reveal information about the motion in the scene. We pursue this idea and propose the **`Blur2Flow`** network that derives the motion flow $\mathtt{F}_{\mathtt{ei}\rightarrow\mathtt{si}}$ that is associated with the blur pattern in the long exposure $\hat{\mathtt{L}}_{\mathtt{i}}$. The sensor design ensures that the short and long exposures are completed precisely at the same time point, and in our HDR reconstruction, the sharp frame $\hat{\mathtt{I}}_{\mathtt{ei}}$ is aligned with the short exposure. Given $\hat{\mathtt{L}}_{\mathtt{i}}$ and $\hat{\mathtt{I}}_{\mathtt{ei}}$ provided in each frame, one can employ a standard motion estimation method to estimate the intra-frame flow. However, in our case, the two inputs are overlapping in time, and finding the correct correspondence of $\hat{\mathtt{I}}_{\mathtt{ei}}$ in the long exposure $\hat{\mathtt{L}}_{\mathtt{i}}$ is ambiguous. Therefore, an existing method such as PWC-Net [SYLK18] cannot be adopted as is, so we apply the following modification to the PWC-Net architecture tailoring it to our inputs. In the original PWC-Net, the two nearby frames are fed to the same feature extractor to build the feature pyramids. Then, at each pyramid scale l, the feature of the second frame is warped to the position of the first frame using the upsampled flow, and a cost volume is created to compare the features of the first frame with the warped features from the second one. In our case, as the sharp HDR frame and long exposures are different in type, we process them with two independent feature extractors and create multi-scale features $\mathtt{V}^{\mathtt{l}}_{\mathtt{i}}$ and $\mathtt{U}^{\mathtt{l}}_{\mathtt{i}}$ that correspond to the sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$ and long exposure $\hat{\mathtt{L}}_{\mathtt{i}}$, respectively. Then, at each scale l, the intra-frame flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{ei}\rightarrow\mathtt{si}}$ is estimated as follows:

$$\mathtt{F}^{\mathtt{l}}_{\mathtt{ei}\rightarrow\mathtt{si}} = \mathbf{\mathtt{Blur2Flow}}(\mathtt{V}^{\mathtt{l}}_{\mathtt{i}}, \mathtt{U}^{\mathtt{l}}_{\mathtt{i}}, \uparrow \mathtt{F}^{\mathtt{l+1}}_{\mathtt{ei}\rightarrow\mathtt{si}}) \tag{1}$$

where **`Blur2Flow`** is a multi-layer CNN with DenseNet connections [SYLK18, HLVDMW17] and $\uparrow \mathtt{F}^{\mathtt{l+1}}_{\mathtt{ei}\rightarrow\mathtt{si}}$ is the upsampled flow from the previous layer. Note at each scale, we do not need to warp the features of the sharp HDR frame, hence no cost volume must be computed. This process is repeated until a desired scale l_0 is reached.

Quadratic motion model: `FitQuad` We continue such multi-scale processing in our non-learnable quadratic motion modeling. Given the intra-frame flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s0}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s1}}$ that are recovered by **`Blur2Flow`** separately for each frame, we also find the inter-frame flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{e0}}$ between the HDR frames $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$ (downscaled to a given scale l) using a state-of-the-art flow estimation method as proposed in [TD20]. While in practice, a quadratic motion model that is aligned with $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ can be derived with only two flows ($\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s0}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}}$), we establish another possible flow, namely $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s1}}$, which corresponds to the flow between $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{s1}}$. It is computed as follows:

$$\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s1}} = \mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}} + \mathtt{warp}(\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s1}}) \tag{2}$$

where `warp` is a differentiable warping operator using bilinear sampling [JSZ*15]. Here, the flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s1}}$ is aligned with the frame $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$; therefore, we need to warp $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s1}}$ using the flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}}$ to become aligned with $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ (refer to the bottom row in Fig. 2). Since the two flows are opposite in their directions, we sum up the flows instead of subtracting them. Similarly, for the frame $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$, we compute the additional flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s0}}$ as:

$$\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s0}} = \mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{e0}} + \mathtt{warp}(\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{e0}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s0}}) \tag{3}$$

Now, for warping $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ to a novel time t, we derive a quadratic motion flow as:

$$\mathtt{F}^{\mathtt{l}}_{\mathtt{0}\rightarrow\mathtt{t}} = \frac{1}{2}\mathtt{a}_0 \times \mathtt{t}^2 + \mathtt{v}_0 \times \mathtt{t} \tag{4}$$

where a_0 and v_0 express the acceleration and velocity of a non-uniform motion, and they are derived from $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s0}}$, $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{e1}}$, and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s1}}$ using the least square fit. Note that the derived model explains the non-uniform motion for the entire range of $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{s0}}$ to $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$. For a curvilinear motion, e.g. a rotatory motion, these parameters can be considered as the first two terms in the Taylor approximation of the curvilinear motion. Similarly, we can compute the flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{1}\rightarrow\mathtt{t}}$:

$$\mathtt{F}^{\mathtt{l}}_{\mathtt{1}\rightarrow\mathtt{t}} = \frac{1}{2}\mathtt{a}_1 \times \mathtt{t}^2 + \mathtt{v}_1 \times \mathtt{t} \tag{5}$$

where the parameters a_1 and v_1 are calculated from the triplet of flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s1}}$, $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{e0}}$, and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s0}}$ using a least square fit. Existing VFI methods with non-uniform motion assumptions usually require more than two frames as the input. However, this enforces that the parameters of non-uniformity (acceleration and velocity) are fixed along multiple frames, which might not hold in practice. In contrast, our method only relies on two immediate frames, and as a result, we impose such constraints in a closer temporal range that allows us to model more complex non-uniform motion. Moreover, providing the additional flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s1}}$ not only allows us to approximate a higher order motion, e.g., a cubic motion model but also incorporates the motion flow information from the other frame to increase flow consistency between $\mathtt{F}^{\mathtt{l}}_{\mathtt{1}\rightarrow\mathtt{t}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{0}\rightarrow\mathtt{t}}$. In Sec. 6.4, we ablate the effect of including $\mathtt{F}^{\mathtt{l}}_{\mathtt{e0}\rightarrow\mathtt{s1}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{e1}\rightarrow\mathtt{s0}}$ in our motion model. Since the time interval between $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{s1}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$ is shared when computing the motion model for the frame pairs $\mathtt{M_0}$ and $\mathtt{M_1}$, and then $\mathtt{M_1}$ and $\mathtt{M_2}$, the temporal consistency is also preserved.

Multiscale blending: `Blend` In the last step, we introduce a multi-scale blending scheme to reconstruct the final interpolated image $\hat{\mathtt{I}}_{\mathtt{t}}$. Specifically, at each scale l, given the forward flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{0}\rightarrow\mathtt{t}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{1}\rightarrow\mathtt{t}}$, we compute the backward flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{t}\rightarrow\mathtt{0}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{t}\rightarrow\mathtt{1}}$ using the flow reversal introduced in QVI [XSS*19]. We then warp the sharp HDR frames $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$ to the novel position t using the backward flows as:

$$\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{0}\rightarrow\mathtt{t}} = \mathtt{warp}(\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{t}\rightarrow\mathtt{0}}) \text{ and } \hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{1}\rightarrow\mathtt{t}} = \mathtt{warp}(\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{t}\rightarrow\mathtt{1}}) \tag{6}$$

where $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{e1}}$ are the input frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s0}}$ downsampled by 2^l. Afterward, we predict the soft occlusion weight $\alpha^{\mathtt{l}}$ that controls the contribution of input warped images $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{0}\to\mathtt{t}}$ and $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{1}\to\mathtt{t}}$:

$$\alpha^{\mathtt{l}} = \mathbf{Blend}(\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{0}\to\mathtt{t}}, \hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{1}\to\mathtt{t}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{t}\to\mathtt{0}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{t}\to\mathtt{1}}, \uparrow \alpha^{\mathtt{l}+\mathtt{1}}) \tag{7}$$

where **Blend** is a multilayer CNN and $\uparrow \alpha^{\mathtt{l}+\mathtt{1}}$ is the upsampled weight from the previous scale. Note the input flows $\mathtt{F}^{\mathtt{l}}_{\mathtt{t}\to\mathtt{0}}$ and $\mathtt{F}^{\mathtt{l}}_{\mathtt{t}\to\mathtt{1}}$ aid the network in reasoning about the occlusion regions. Given the occlusion weight, the warped images are combined as follows:

$$\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{t}} = \frac{(1-t)\alpha^{\mathtt{l}} \odot \hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{0}\to\mathtt{t}} + t(1-\alpha^{\mathtt{l}}) \odot \hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{1}\to\mathtt{t}}}{(1-t)\alpha^{\mathtt{l}} + t(1-\alpha^{\mathtt{l}})} \tag{8}$$

where $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{t}}$ is the synthesized intermediate frame at scale l, as required in the loss computation (Eq. 11). The operator $\odot$ stands for per-pixel multiplication. Finally, at the finest scale l_0, the interpolated frame $\hat{\mathtt{I}}_{\mathtt{t}}$ is derived.

Loss function Our loss function is composed of three components that are targeted to train the **MakeHDR**, **Blur2Flow**, and **Blend** networks. First, the output of the **MakeHDR** network is supervised with the ground truth $\mathtt{I}_{\mathtt{ei}}$ and $\mathtt{L}_{\mathtt{i}}$ (refer to Sec. 6.1 on details of how we acquire the ground truth frames from high-framerate video datasets) using the reconstruction loss:

$$\mathrm{L}_{\mathrm{hdr}} = \sum_{i=0,1} \|\mathtt{I}_{\mathtt{ei}} - \hat{\mathtt{I}}_{\mathtt{ei}}\|_1 + \|\mathtt{L}_{\mathtt{i}} - \hat{\mathtt{L}}_{\mathtt{i}}\|_1 \tag{9}$$

As the ground truth flow is not available, we employ a multiscale image loss to supervise the **Blur2Flow** network:

$$\mathrm{L}_{\mathrm{flow}} = \sum_{i=0,1} \sum_{l=l_0}^{L} \|\mathtt{I}^{\mathtt{l}}_{\mathtt{ei}} - \mathtt{warp}(\mathtt{I}^{\mathtt{l}}_{\mathtt{si}}, \mathtt{F}^{\mathtt{l}}_{\mathtt{ei}\to\mathtt{si}})\|_1 \tag{10}$$

where $\mathtt{I}^{\mathtt{l}}_{\mathtt{ei}}$ and $\mathtt{I}^{\mathtt{l}}_{\mathtt{si}}$ are the ground truth frames $\mathtt{I}_{\mathtt{ei}}$ and $\mathtt{I}_{\mathtt{si}}$ downsampled by 2^l. At each scale l, we warp $\mathtt{I}^{\mathtt{l}}_{\mathtt{si}}$ using the predicted flow $\mathtt{F}^{\mathtt{l}}_{\mathtt{ei}\to\mathtt{si}}$ and compare with $\mathtt{I}^{\mathtt{l}}_{\mathtt{ei}}$. Note that this loss component will try to align the warped image and the input frames for all regions in an image, including the occluded part. However, we argue this is not a significant issue because the intra-frame motion captured in the long exposure is relatively small compared to the inter-frame motion. Hence, we deal with small disoccluded areas within a frame, and the only degradation that can occur is over-smoothed flow at occlusion boundaries which can be resolved with a more sophisticated occlusion treatment. Lastly, we supervise the output of the **Blend** network using the reconstruction loss at each scale:

$$\mathrm{L}_{\mathrm{synth}} = \sum_{l=l_0}^{L} \|\mathtt{I}^{\mathtt{l}}_{\mathtt{t}} - \hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{t}}\|_1 \tag{11}$$

where $\mathtt{I}^{\mathtt{l}}_{\mathtt{t}}$ is the corresponding ground truth for interpolated frame $\hat{\mathtt{I}}^{\mathtt{l}}_{\mathtt{t}}$ at each scale l. The final loss $\mathrm{L}_{\mathrm{total}}$ is then computed as:

$$\mathrm{L}_{\mathrm{total}} = \mathrm{L}_{\mathrm{hdr}} + \mathrm{L}_{\mathrm{flow}} + \mathrm{L}_{\mathrm{synth}} \tag{12}$$

It is worth mentioning that based on our observation, optimizing the network based solely on the final loss would create ambiguity as to whether the network should improve **Blur2Flow** or **Blend** network to decrease the loss; therefore, intermediate supervision (Eq. 10 and Eq. 11) is essential to train each component properly.

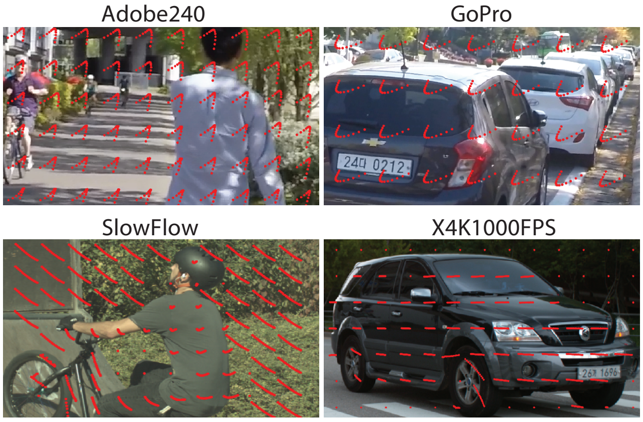

Figure 3: *Trajectories of pixels (red dots) for 16 consecutive frames in a sample scene from four different datasets. The scenes in Adobe240 and GoPro datasets mostly have globally non-uniform motion due to non-uniform camera motion, while in datasets such as X4K1000FPS and SlowFlow, the scenes mostly contain locally non-uniform motion.*

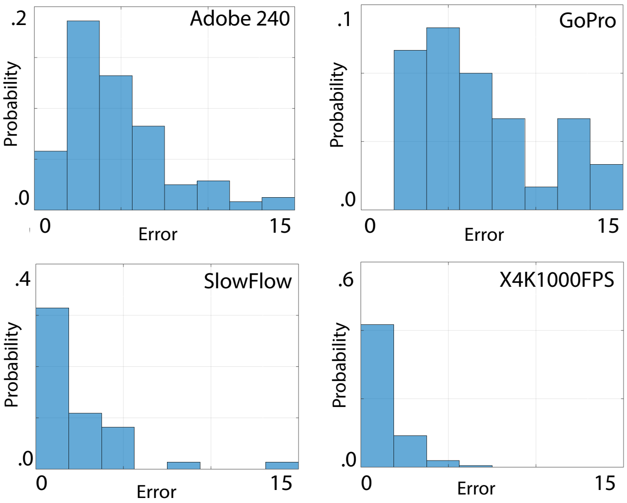

Figure 4: *The histogram of measured non-uniform motions for different datasets, where the horizontal axis shows the normalized motion error ($\times 10^{-2}$) with respect to the linear motion fit, and the vertical axis denotes the probability of the observed frames given an error value.*

4. Motion non-uniformity analysis

In order to properly validate our proposed method, we must ensure that our dataset contains diverse examples of scene motion non-uniformity. To this end, we analyze motion non-uniformity in some popular high-framerate video datasets, including Adobe240 [SDW*17], GoPro [NHKML17], X4K1000FPS [SOK21], and SlowFlow [JGW*17]. Our procedure is as follows: For each pixel in a given frame, we use Raft [TD20] to track the corresponding pixels for N consecutive frames. We choose $N = 8$ for the Adobe240, GoPro, and SlowFlow datasets as they are captured with 240FPS, and eight frames represent the time gap between two consecutive frames in a 30FPS video, and we choose $N = 33$ for X4K1000FPS containing 1000FPS videos. Note that in some cases, such tracking might fail due to occlusions and textureless regions. We find the occlusion regions by applying a forward-backward flow consistency check [JSB*20] between the first and last frames, and exclude them in our measurements. Likewise, as the estimated flow in the textureless regions is usually erroneous, we clip the flow to zero if its value is less than one pixel. Fig. 3 shows the trajectories of pixels for four sample scenes that contain regions with non-uniform motion. In the next step, we find a linear model that, in the least square sense, fits the motion trajectory for each pixel. We then consider the mean square error with respect to such a linear fit, where higher errors indicate more motion non-uniformity. Note that for each pixel, the error value is normalized by the aggregated pixel displacement across the consecutive frames. Since the error is calculated for individual pixels, we measure the amount of motion non-uniformity in a frame by taking the 50th percentile of the calculated error over all pixels. We then repeat this procedure for non-overlapping sets of N consecutive frames in each scene in each dataset. Fig. 4 shows the histogram of measured non-uniform motions for each dataset, where the horizontal axis denotes the error of the linear fit ($\times 10^{-2}$) divided into eight discrete bins, and the vertical axis is the probability of observing the scene for a given error value. The Adobe240 and GoPro datasets feature significant percentages of non-uniform motion as they are captured with a handheld camera. Although large motions are present in the X4K1000FPS dataset, the camera moves along mostly linear trajectories.

5. Implementation

Our **`MakeHDR`** network architecture follows [CBM*22]. The network output is given in the Bayer domain, and we apply demosaicing using OpenCV [Bra00], followed by a gamma correction to create the final short and long exposures in the sRGB format. The **`Blur2Flow`** network employs an architecture similar to the PWCNet [SYLK18], and also outputs the motion flow at a quarter resolution and employs the context network for refining the flow. We then apply bilinear interpolation to obtain the half- and full-resolution flows. Our **`Blend`** network is implemented as a 12-layer conventional neural network with dilated convolutions and skip connections. During training, we use the patch size of 768×768; nevertheless, at the inference time, our convolutional network, as well as all non-learnable components, scale with resolution.

6. Results

In this section, we first introduce the training and evaluation datasets. Then we show quantitative and qualitative comparisons of our method with existing VFI methods. Finally, we provide ablation to justify our training set and different components of our method.

6.1. Dataset

As it is impossible to capture ground truth high-framerate HDR videos using our dual exposure sensor, and third-party high-framerate HDR videos are unavailable, we synthesize our training and evaluation datasets using existing LDR high-framerate videos.

Table 1: *Quantitative comparison of our method with state-of-the-art VFI methods. The ABME and QVI methods are designed to handle non-uniform motions, while the XVFI and FILM methods rely on a linear motion assumption but can handle large motions. Methods are indicated with * when they are trained from scratch with our training set.*

	Adobe240		**GoPro**	
Methods	PSNR	SSIM	PSNR	SSIM
ABME [PLK21]	31.28	0.83	30.98	0.82
QVI [XSS*19]	31.30	0.86	30.80	0.84
QVI* [XSS*19]	31.16	0.86	30.70	0.84
XVFI [SOK21]	31.07	0.83	30.75	0.82
XVFI* [SOK21]	30.66	0.83	30.41	0.82
FILM [RKT*22]	31.11	0.83	30.75	0.82
FILM* [RKT*22]	31.04	0.83	30.74	0.82
Ours	**34.82**	**0.93**	**35.01**	**0.92**

In our experiments, we take the scenes from X4K1000FPS [SOK21] and SlowFlow [JGW*17] as our training datasets, and we consider Adobe240 [SDW*17] and GoPro [NHKML17] as our evaluation datasets. Our training and testing video sequences are defined as follows: We take 16 consecutive frames in a high-framerate video, where the 1st and 4th frames are our sharp beginning and ending frames ($\hat{\mathtt{I}}_{\mathtt{s0}}$ and $\hat{\mathtt{I}}_{\mathtt{e0}}$). We sum up the four neighboring frames starting from 1 to 4 to simulate the long exposure $\hat{\mathtt{L}}_{\mathtt{0}}$. We then skip 9 frames to simulate the camera readout gap. Similarly, we take the 13th and 16th frames as the $\hat{\mathtt{I}}_{\mathtt{s1}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ and sum the frames from 13 to 16 to create the long exposure $\hat{\mathtt{L}}_{\mathtt{1}}$. We consider frames 7 and 10 as the target frames for the reconstructions. Note that in our simulation of long exposures, we clip the aggregated pixel intensity if it exceeds the value of 255. In our simulation, we ignore each patch if more than 20% of its content is already saturated in the original high-framerate video. In order to make our method robust to high blur and saturation, we perform data augmentation by creating different amounts of blur and different amounts of saturation. For our test set, we are interested in evaluating our method against the other methods for different ranges of non-uniformity; hence we split all scenes in the Adobe240 [SDW*17] and GoPro [NHKML17] datasets into four different categories of Easy, Medium, Difficult, and Extreme based on the error magnitude of the linear fit derived in Sec. 4. Specifically, we divide the entire histogram range (15×10^{-2} here) into four equal segments (expressing our four motion non-uniformity categories), and we draw 125 sample frames both for the Adobe240 and GoPro datasets per each category.

6.2. Quantitative comparison

We compare our proposed method with state-of-the-art sharp VFI methods (refer to Sec. 2.1): FILM [RKT*22] and XVFI [SOK21] which rely on a uniform motion assumption, and QVI [XSS*19], and ABME [PLK21] which explicitly support the non-uniform motion. QVI employs four consecutive frames as the input, and FILM and XVFI require just two frames. While ABME also uses only two

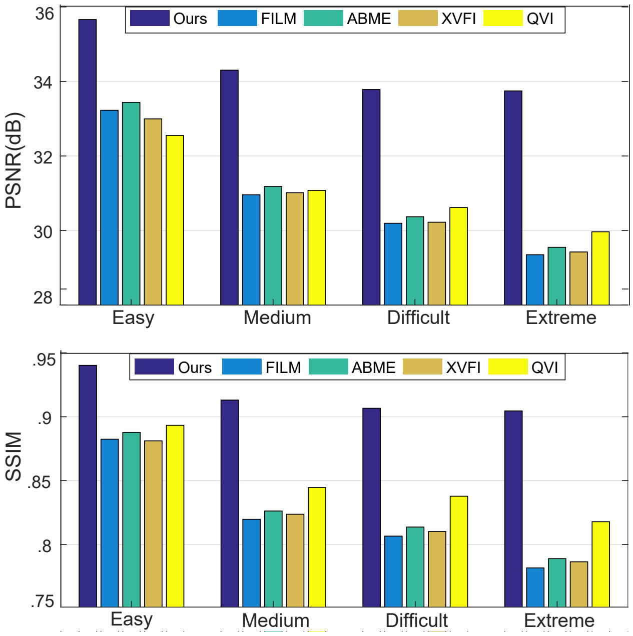

Figure 5: *Quantitative comparison of our method with state-of-the-art VFI methods for four different motion non-uniformity categories (refer to Sec. 6.1). Each bin reports the average reconstruction error for a given method over 250 sample frames per category.*

frames as input, it relaxes the uniform motion constraint by first estimating symmetric bilateral motion fields and then refining them to become asymmetric. As the LDR (sRGB) images in the high-framerate dataset are used to synthesize our training and evaluation set, we can directly feed them as input to the VFI methods. For our method, though, we feed them along with the simulated long exposure as described in Sec. 6.1. Note that we are unable to compare with the blurry VFI methods (refer to Sec. 2.3), as they require well-exposed blurry input frames (effectively, blurry HDR frames) while our long exposure typically contains a considerable amount of saturation that poorly handled by these methods. Tbl. 1 summarizes our comparisons with the VFI methods (used with their pre-trained weights) for each of our test datasets (Adobe240 and GoPro) separately as specified in Sec. 6.1. Note that XVFI uses almost the same training set as ours while applying extra data augmentation, and a method such as FILM carefully prepared their dataset to include all the possible motion ranges, with a much larger training data size than we consider. Nevertheless, for a fair comparison, we have re-trained XVFI, FILM, and QVI using our training set (indicated with * in Tbl. 1) and observed a lower performance. Unfortunately, the training code for ABME is not publicly available. Moreover, Fig. 5 provides a deeper insight into each method performance when we aggregate those datasets and split them into four different categories with respect to motion complexity (Sec. 4). Overall for more uniform motion, the competing VFI methods perform similarly, while clear advantages of the QVI method can be seen for more complex motion. In all cases, our method outperforms the existing VFI methods by a large margin. It is also more stable in the interpolation quality for higher motion non-uniformity. We hypothesize that this stability could be attributed to our quadratic motion fitting part, which has no learnable parameters and only relies on the accuracy of flows, which might drop off slightly at higher non-uniform motion. Other VFI solutions that mostly learn how to handle non-uniform motion might impose higher requirements on the training set.

6.3. Qualitative comparison

We first visualize the examples of HDR scenes captured in daylight and dark conditions in Fig. 1 and Fig. 6. The flow map reconstructed by our **`Blur2Flow`** module in Fig. 6, as well as the motion blur magnitude in the long exposures indicate the complexity of motion. In the accompanying videos, we demonstrate that competing VFI methods struggle with the scene in Fig. 1, while our method benefits from additional information that is encoded in the motion blur pattern to improve the interpolation quality. We then provide visual comparisons with the state-of-the-art VFI methods for three synthesized scenes with ground truth in Fig. 7. Moreover, we compare to other methods in Fig. 8 using the captured sequences. All the capturing processes were done with our Axiom-beta camera with a CMOSIS CMV12000 sensor [CMV21]. In both setups, we use the exposure ratio of 4 between the short and long exposures. Since the frames captured using our camera cannot be fed directly to the other VFI methods, we first reconstruct the sharp HDR images $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ using our **`MakeHDR`** network. They are then tonemapped using Reinhard-Global 2002 [RSSF02] and gamma-corrected, and are fed to the LDR VFI methods. The upper scene in Fig. 8 shows an example of a rolling disc in which the existing VFI methods, even the ones designed to deal with non-uniform motion such as ABME and QVI, fail to properly interpolate an intermediate frame due to non-uniform motion caused by the rotatory motion of the disc. In the next examples, we captured a crystal ball while the camera is rapidly rotating (the middle scene) or an object is moving behind the crystal ball (the bottom scene). We can observe that in these challenging examples where even a uniform motion in the scene might appear non-uniform in the refracted image, other methods struggle to correctly reconstruct an in-between frame. In all cases, we can see our method faithfully reconstruct the in-between frames even in difficult conditions where there are reflections on the crystal ball (the middle and bottom scenes). Please refer to our supplementary video for the temporal consistency of our method.

6.4. Ablation study

We perform a series of ablations to show the contributions of each key component in our proposed method and to analyze the alternative solutions. We summarize the obtained results in Fig. 10 and Tbl. 2, where each ablation component we denote with a unique label that is also included in the related paragraph title.

Impact of `Blur2Flow` network: NoBlur2Flow We analyze the contribution of the **`Blur2Flow`** network where we attempt to reconstruct the intermediate frames using only the backward and forward flows between the sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ using Raft [TD20]. This experiment suggests the version of our method that makes the linear assumption, in which we linearly split the flow at any position t between the frames; however, this leads to large positional errors in the interpolated content, as seen in Fig. 10. Our results clearly

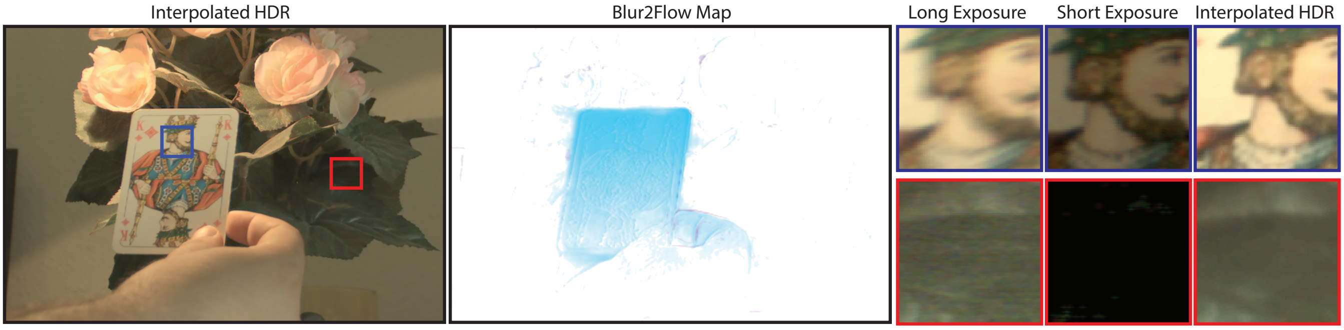

Figure 6: *Visualization of flow maps reconstructed by our* **`Blur2Flow`** *network. Otherwise, the figure layout follows the one in Fig. 1.*

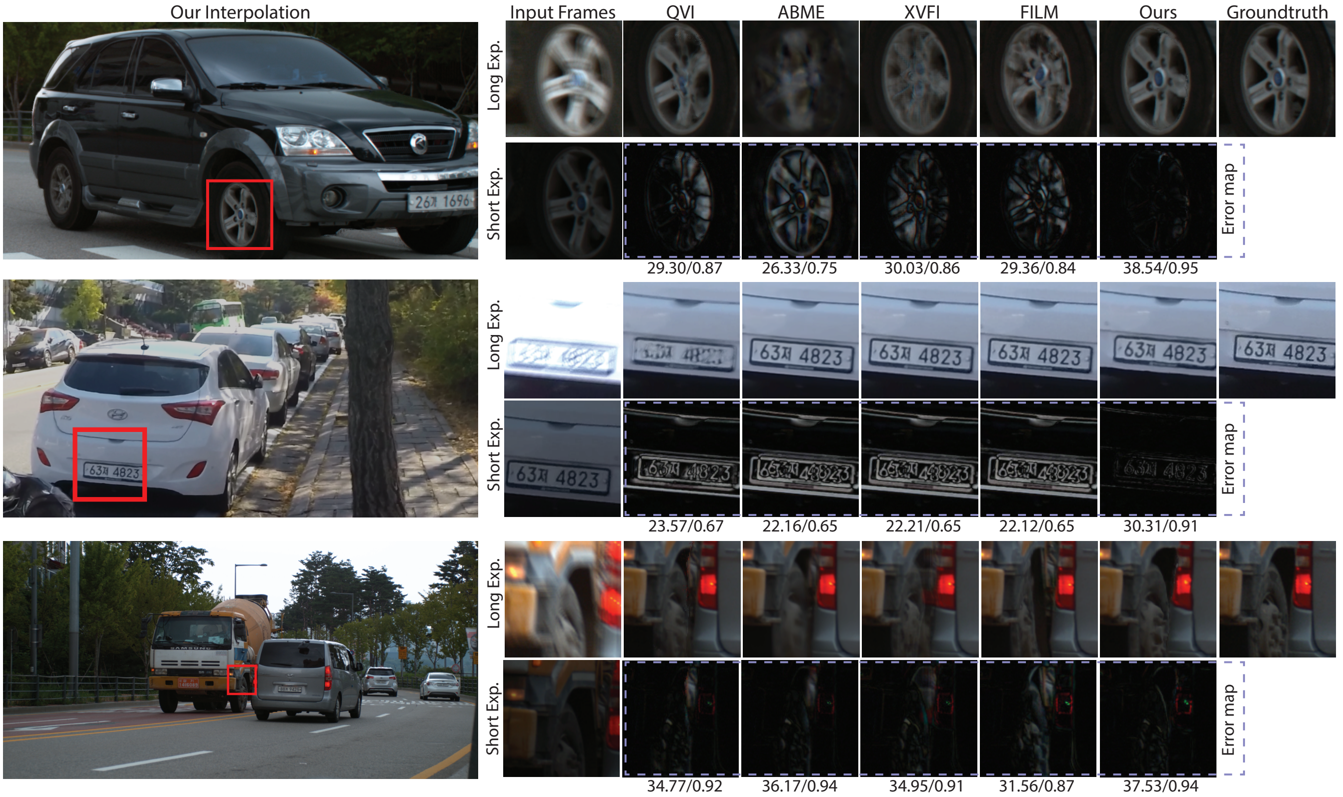

Figure 7: *Visual comparisons of our method with the state-of-the-art VFI methods using the synthetic dataset described in Sec. 6.1. For each of the three scenes, the first row of insets shows the performance of respective VFI methods, while the second row presents the corresponding per-pixel error maps between the interpolated results and the ground truth. The PSNR/SSIM values written below each error map are computed for each inset rather than the entire image. In the upper scene taken from the X4K1000FPS test set, the wheel moves in a non-linear trajectory, and the existing VFI methods struggle to position the wheel correctly for the interpolated frames, while our method leads to a good alignment with the ground truth. In the middle scene taken from the GoPro dataset, the camera is moving with an extremely non-uniform motion as shown in Fig. 3. While the existing VFI methods produce visually plausible results, they are not correctly aligned with the ground truth as the error map reveals. The bottom scene, taken again from the X4K1000FPS test set, contains a combination of camera and object movements. In this case, the existing VFI methods fail to properly handle the occlusion boundaries.*

indicate the effectiveness of including the **`Blur2Flow`** network in our pipeline (Tbl. 2).

Impact of sharp HDR frame: NoSharp We investigate the effect of including the sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$, along with the long blurry exposure $\hat{\mathtt{L}}_{\mathtt{i}}$, on the accuracy of motion from blur derivation. To do so, we consider $\hat{\mathtt{L}}_{\mathtt{i}}$ as the only input to the **`Blur2Flow`** network and exclude $\hat{\mathtt{I}}_{\mathtt{ei}}$ (note that $\hat{\mathtt{I}}_{\mathtt{ei}}$ is still available for other components in our pipeline). As it can be seen in Fig. 10 the availability of $\hat{\mathtt{I}}_{\mathtt{ei}}$ reduces geometric image distortions and $\hat{\mathtt{I}}_{\mathtt{ei}}$ compensates for the

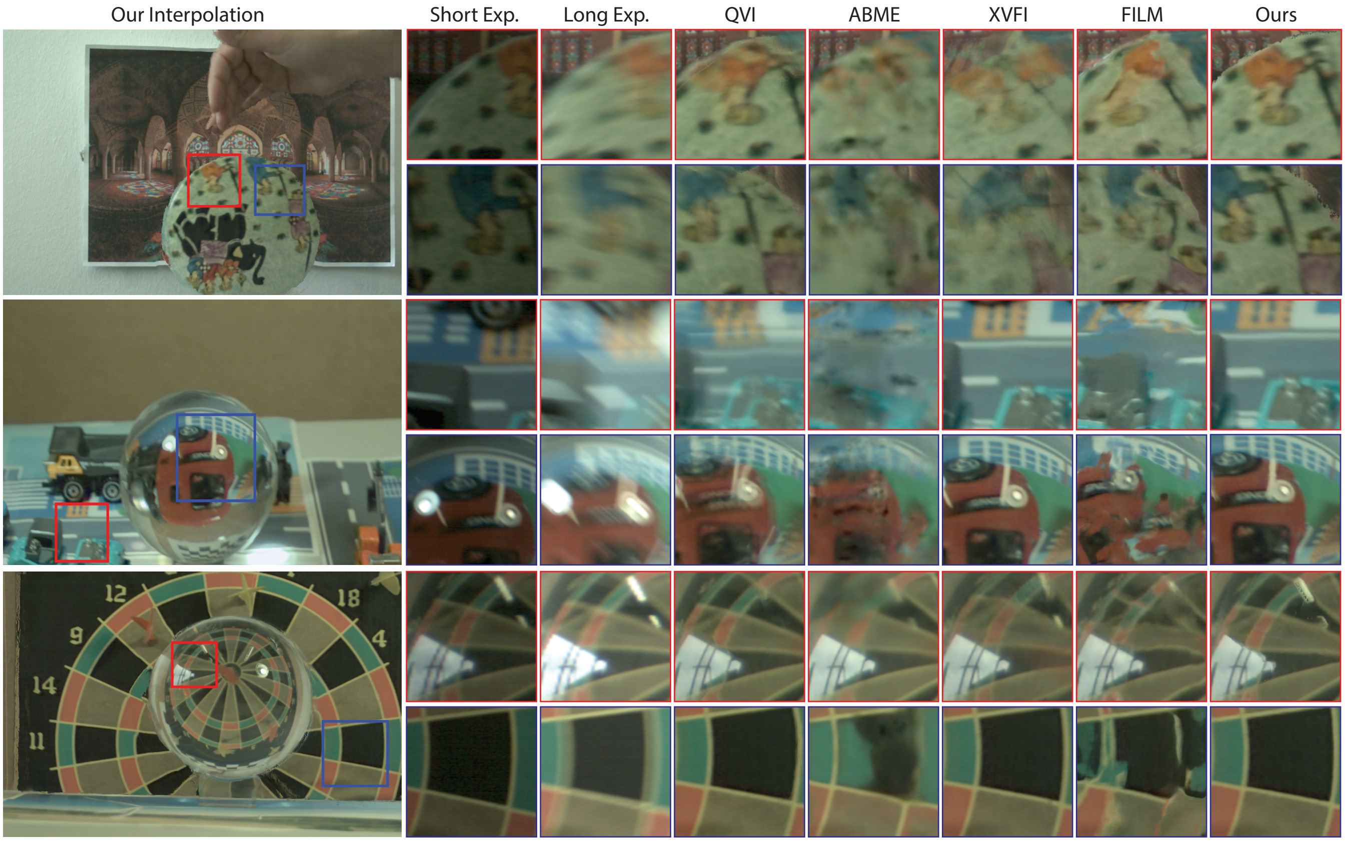

Figure 8: *The visual comparisons of our interpolation results for three scenes captured using our camera with a dual-exposure sensor. Our method is able to correctly interpolate the frames in the scenes with the challenging cases of a rolling disc (the upper scene), a rotary camera motion (the middle scene), and a moving object behind a refractive object (the bottom scene).*

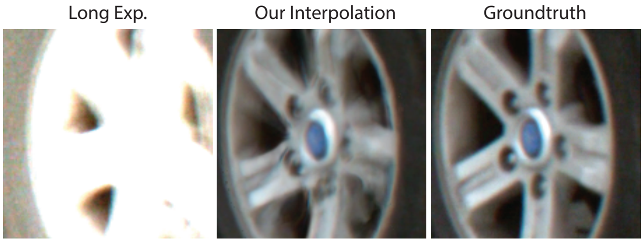

Figure 9: *Our interpolation failure example in a case where the moving content is highly saturated in the long exposure.*

lack of information for saturated pixels that are inherent for $\hat{\mathtt{L}}_{\mathtt{i}}$ in our setup with a dual-exposure sensor. Following this observation, we expect that replacing our **`Blur2Flow`** network with a solution, where the intra-frame flow is extracted solely based on $\hat{\mathtt{L}}_{\mathtt{i}}$ [ZWT20] should lead to a similar outcome as this ablation.

Quadratic model with temporal flows: TemporalFlows Considering more than two consecutive frames involves a larger time span; as a result, fine-grained motion cannot be properly handled. We have made such observations when comparing our method with a method like QVI, which uses four frames to compute the quadratic model. Nonetheless, to highlight the advantage of the intra-flow $\mathtt{F}_{\mathtt{ei}\rightarrow\mathtt{si}}$ estimated from the **`Blur2Flow`** module, we conduct an ablation where we fit the quadratic motion using the temporal flows extracted from four consecutive HDR frames (similar to QVI); however, we observed a lower performance than ours with two frames, while it still has a better performance compared to QVI.

Alternative approach to `Blur2Flow` network: SharpStart Instead of directly recovering the motion flow from the blur, we employ a 12-layer conventional neural network with dilated convolutions to predict the sharp frame $\hat{\mathtt{I}}_{\mathtt{si}}$ aligned with the beginning of the frame, then use the Raft [TD20] to estimate the intra-frame flow $\mathtt{F}_{\mathtt{ei}\rightarrow\mathtt{si}}$ between the $\hat{\mathtt{I}}_{\mathtt{ei}}$ and predicted $\hat{\mathtt{I}}_{\mathtt{si}}$. This ablation demonstrates that the particular method of deriving the intra-frame flow from motion is less important, under the condition that sharp, saturation-free reference $\hat{\mathtt{I}}_{\mathtt{ei}}$ is available. Still, our proposed method leads to slight quality improvement.

Quadratic vs. cubic motion model: CubicModel Since our method provides three estimated flows in each frame, we are able to approximate a higher-order motion, e.g., cubic. Hence, we perform an ablation where we replace the quadratic motion model derived in Sec. 3 with a cubic model. Overall the obtained results are comparable in terms of the SSIM prediction, but the quadratic model

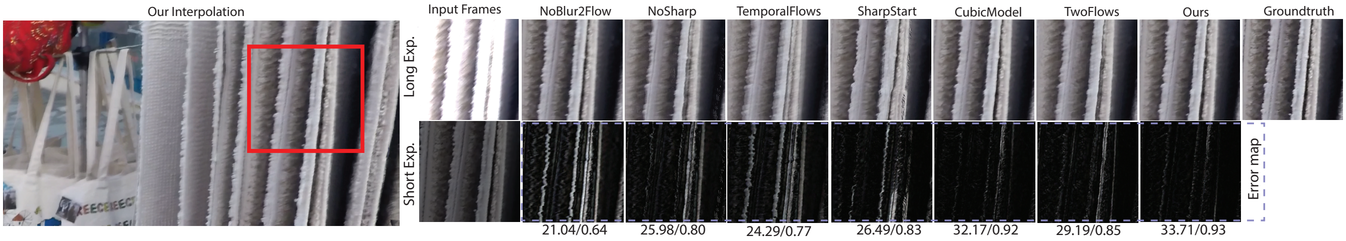

Figure 10: *Ablation results. The figure layout is similar to Fig. 7. Refer to Sec. 6.4 for more details on each ablation scenario.*

Table 2: *The ablation results indicate the performance of alternative solutions for major design choices in our proposed method. Refer to Sec. 6.4 where we provide more details on each ablation.*

	PSNR	SSIM
NoBlur2Flow	30.97	0.82
NoSharp	30.28	0.82
TemporalFlows	31.93	0.85
SharpStart	34.35	0.91
CubicModel	33.84	0.92
TwoFlows	34.00	0.90
Ours	**34.92**	**0.93**

is slightly better in terms of PSNR and visual results (Fig. 10). A key difference is that while the cubic model involves a closed-form solution, we derive the quadratic model in a least-squares fashion that allows for the correction of slight errors in the derived flows.

Two vs. three flows: TwoFlows To see the effect of including the additional flows $\mathtt{F}_{\mathtt{e1}\rightarrow\mathtt{s0}}$ and $\mathtt{F}_{\mathtt{e0}\rightarrow\mathtt{s1}}$ in the derivation of our quadratic motion model, we exclude them from the input to the **`FitQuad`** module. The obtained results (Tbl. 2) indicate that including an independent estimate of the third flow contributes toward correcting for potential inconsistencies in the other two flows. For example, in Fig. 10, ghosting artifacts along higher contrast edges are clearly visible when only two flows are employed.

6.5. Limitations and future work

Saturation is inevitable in long exposure for bright scene regions. In case of a local motion blur that is fully covered with saturation, our flow prediction using the **`Blur2Flow`** network becomes less accurate. Fig. 9 shows an example of this case where we synthetically increase the saturation in the long exposure for the wheel example shown in Fig. 7, and our method fails to correctly reconstruct the intermediate frame. However, in case of a local motion blur with partial saturation or a global camera motion, even with fully saturated regions, as shown in Fig. 7 and Fig. 10, our method can recover the flow by propagating the flow information from the unsaturated regions.

The dynamic range that we can reconstruct is limited by the exposure ratio of four that we assume in this work. For larger ratios, the accuracy of HDR frame reconstruction by the **`MakeHDR`** network might be reduced [CBM*22], which could adversely affect the accuracy of HDR video interpolation. Moreover, when capturing an HDR scene, we adjust the lowest exposure time in such a way that the long exposure is not very saturated so that there is enough valuable blurry information. This procedure is currently done manually; an automatic selection of the optimal exposure time is an interesting future work direction that could lead to further performance improvements. We also relegate as future work porting our technique to other multi-exposure sensors that are used in modern smartphones [GSM22], such as Sony's Quad Bayer [Son22] and Samsung's Tetracell/Nonacell [Sam22] sensors. Such sensors should enable further improvements in the VFI quality via a more uniform layout of pixels with varying exposures. It would also be interesting to experiment with more than two exposures, as supported by such sensors.

Lastly, investigating optical blur and finding ways to remove it along with motion blur could be an interesting, but challenging, future direction. The current state-of-the-art image restoration methods [ZAK*22,WCB*22] still treat them as two separate tasks due to the difficulties in removing the coexisting blur. However, we believe our employed sensor design can significantly facilitate disentangling the motion blur that changes with exposure from the optical blur that remains constant between exposures.

7. Conclusion

In this work, we presented a method for high-dynamic-range video frame interpolation using dual-exposure sensors. Our method outperforms the existing VFI methods both in terms of quantitative metrics as well as visual results for the challenging scenes containing non-uniform motions. In particular, we achieve high-precision alignment of scene motion with the ground truth, where other methods clearly fail, although they may produce visually plausible results. Our method can handle complex motion with consistently high performance as it depends little on explicitly training this reconstruction aspect. Instead, we capitalize on the increased temporal sampling rate due to motion reconstruction from blur information. Also, our method is less dependent on scene lighting conditions, whereas other methods designed for single-exposure sensors may suffer from image saturation in bright regions or excessive noise in dark conditions.

References

[AFTH19] Alghamdi M. M., Fu Q., Thabet A. K., Heidrich W.: Reconfigurable snapshot hdr imaging using coded masks and inception network. 3

[AKR*21] ARGAW D. M., KIM J., RAMEAU F., CHO J. W., KWEON I. S.: Optical flow estimation from a single motion-blurred image. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 891–900. 3

[BLM*19] BAO W., LAI W.-S., MA C., ZHANG X., GAO Z., YANG M.-H.: Depth-aware video frame interpolation. In *Proc. CVPR* (2019), pp. 3703–3712. 2

[Bra00] BRADSKI G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000). 7

[CA20] COGALAN U., AKYUZ A. O.: Deep joint deinterlacing and denoising for single shot dual-ISO HDR reconstruction. *IEEE Trans. Image Proc. 29* (2020), 7511–7524. 3

[CBK17] CHOI I., BAEK S.-H., KIM M. H.: Reconstructing interlaced high-dynamic-range video using joint learning. *IEEE Trans Image Processing 26*, 11 (2017), 5353–5366. 2, 3

[CBM*22] COGALAN U., BEMANA M., MYSZKOWSKI K., SEIDEL H.-P., RITSCHEL T.: Learning HDR video reconstruction for dual-exposure sensors with temporally-alternating exposures. *Computers & Graphics* (2022). 2, 3, 5, 7, 11

[CCG*21] CHEN G., CHEN C., GUO S., LIANG Z., WONG K.-Y. K., ZHANG L.: HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset. In *Proc. CVPR* (2021), pp. 2502–2511. 3

[CJY*20] CHEN Y., JIANG G., YU M., YANG Y., HO Y.-S.: Learning stereo high dynamic range imaging from a pair of cameras with different exposure parameters. *IEEE Trans Comp Imaging 6* (2020), 1044–1058. 3

[CKH*20] CHOI M., KIM H., HAN B., XU N., LEE K. M.: Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 10663–10671. 2

[CKL14] CHO H., KIM S. J., LEE S.: Single-shot high dynamic range imaging using coded electronic shutter. *Comp Graph Forum 33*, 7 (2014), 329–338. 2, 3

[CMNL*20] CHI Z., MOHAMMADI NASIRI R., LIU Z., LU J., TANG J., PLATANIOTIS K. N.: All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Proc. ECCV* (2020), pp. 107–123. 3

[CMV21] CMV12000: High speed machine vision global shutter CMOS image sensor, 2021. 2, 3, 5, 8

[CYC*19] CHEN Y., YU M., CHEN K., JIANG G., SONG Y., PENG Z., CHEN F.: New stereo high dynamic range imaging method using generative adversarial networks. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), pp. 3502–3506. 3

[DHL*21] DONG X., HU X., LI W., WANG X., WANG Y.: MIEHDR CNN: Main image enhancement based ghost-free high dynamic range imaging using dual-lens systems. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 1264–1272. 3

[DM08] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*. 2008, pp. 1–10. 5

[DW08] DAI S., WU Y.: Motion from blur. In *Proc. CVPR* (2008), pp. 1–8. 3

[GARC20] GUPTA A., AICH A., ROY-CHOWDHURY A. K.: ALANET: Adaptive latent attention network for joint video deblurring and interpolation. *arXiv preprint arXiv:2009.01005* (2020). 3

[GKSK19] GO C., KINOSHITA Y., SHIOTA S., KIYA H.: An image fusion scheme for single-shot high dynamic range imaging with spatially varying exposures. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences 102*, 12 (2019), 1856–1864. 2, 3

[GSM22] GSMARENA: Quad Bayer sensors: what they are and what they are not, 2022. 2, 11

[GYL*17] GONG D., YANG J., LIU L., ZHANG Y., REID I., SHEN C., VAN DEN HENGEL A., SHI Q.: From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *Proc. CVPR* (2017), pp. 2319–2328. 3

[HKML14] HYUN KIM T., MU LEE K.: Segmentation-free dynamic scene deblurring. In *Proc. CVPR* (2014), pp. 2766–2773. 3

[HKU14] HAJSHARIF S., KRONANDER J., UNGER J.: HDR reconstruction for alternating gain (ISO) sensor readout. In *Comp Graph Forum (Proc. Eurographics)* (2014). 2, 3

[HLVDMW17] HUANG G., LIU Z., VAN DER MAATEN L., WEINBERGER K. Q.: Densely connected convolutional networks. In *Proc. CVPR* (2017), pp. 4700–4708. 5

[HST*14] HEIDE F., STEINBERGER M., TSAI Y.-T., ROUF M., PAJĄK D., REDDY D., GALLO O., LIU J., HEIDRICH W., EGIAZARIAN K., ET AL.: FlexISP: A flexible camera image processing framework. *ACM Trans. Graph. 33*, 6 (2014), 1–13. 2, 3

[JCJG21] JIANG Y., CHOI I., JIANG J., GU J.: HDR video reconstruction with tri-exposure quad-bayer sensors. *arXiv preprint arXiv:2103.10982* (2021). 3

[JGW*17] JANAI J., GUNEY F., WULFF J., BLACK M. J., GEIGER A.: Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proc. CVPR* (2017), pp. 3597–3607. 6, 7

[JHF19] JIN M., HU Z., FAVARO P.: Learning to extract flawless slow motion from blurry videos. In *Proc. CVPR* (2019), pp. 8112–8121. 3

[JSB*20] JONSCHKOWSKI R., STONE A., BARRON J. T., GORDON A., KONOLIGE K., ANGELOVA A.: What matters in unsupervised optical flow. In *Proc. ECCV* (2020), pp. 557–572. 7

[JSJ*18] JIANG H., SUN D., JAMPANI V., YANG M.-H., LEARNED-MILLER E., KAUTZ J.: Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. CVPR* (2018), pp. 9000–9008. 2

[JSZ*15] JADERBERG M., SIMONYAN K., ZISSERMAN A., ET AL.: Spatial transformer networks. *Advances in Neural Information Processing Systems 28* (2015). 5

[KLY21] KOH J., LEE J., YOON S.: Single-image deblurring with neural networks: A comparative survey. *Computer Vision and Image Understanding 203* (2021), 103134. 3

[KR*17] KALANTARI N. K., RAMAMOORTHI R., ET AL.: Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph. 36*, 4 (2017), 144–1. 3

[KR19] KALANTARI N. K., RAMAMOORTHI R.: Deep HDR video from sequences with alternating exposures. In *Comp Graph Forum* (2019), vol. 38, pp. 193–205. 3

[KSB*13] KALANTARI N. K., SHECHTMAN E., BARNES C., DARABI S., GOLDMAN D. B., SEN P.: Patch-based high dynamic range video. *ACM Trans. Graph. 32*, 6 (2013), 202–1. 3

[LC09] LIN H.-Y., CHANG W.-Z.: High dynamic range imaging for stereoscopic scene representation. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (2009), pp. 4305–4308. 3

[LKC*20] LEE H., KIM T., CHUNG T.-Y., PAK D., BAN Y., LEE S.: AdaCoF: Adaptive collaboration of flows for video frame interpolation. In *Proc. CVPR* (2020), pp. 5316–5325. 2

[LLLC19] LIU Y.-L., LIAO Y.-T., LIN Y.-Y., CHUANG Y.-Y.: Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 8794–8802. 2

[MMC*20] MARTEL J. N., MUELLER L. K., CAREY S. J., DUDEK P., WETZSTEIN G.: Neural sensors: Learning pixel exposures for HDR imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*, 7 (2020), 1642–1653. 3

[NHKML17] NAH S., HYUN KIM T., MU LEE K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proc. CVPR* (2017), pp. 3883–3891. 6, 7

[NL20] Niklaus S., Liu F.: Softmax splatting for video frame interpolation. In *Proc. CVPR* (2020), pp. 5437–5446. 2

[NML17] Niklaus S., Mai L., Liu F.: Video frame interpolation via adaptive separable convolution. In *Proc. ICCV* (2017), pp. 261–270. 2

[NMW22] Nguyen C. M., Martel J. N., Wetzstein G.: Learning spatially varying pixel exposures for motion deblurring. *arXiv preprint arXiv:2204.07267* (2022). 3

[PKLK20] Park J., Ko K., Lee C., Kim C.-S.: BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Proc. ECCV* (2020), pp. 109–125. 2, 3

[PLK21] Park J., Lee C., Kim C.-S.: Asymmetric bilateral motion estimation for video frame interpolation. In *Proc. ICCV* (2021), pp. 14539–14548. 2, 3, 7

[PVPA21] Parihar A. S., Varshney D., Pandya K., Aggarwal A.: A comprehensive survey on video frame interpolation techniques. *The Visual Computer* (2021), 1–25. 2

[QWMT19] Qiu J., Wang X., Maybank S. J., Tao D.: World from blur. In *Proc. CVPR* (2019), pp. 8493–8504. 3

[Rek95] Rekleitis I.: Visual motion estimation based on motion blur interpretation. 3

[RKT*22] Reda F., Kontkanen J., Tabellion E., Sun D., Pantofaru C., Curless B.: FILM: Frame interpolation for large motion. *arXiv preprint arXiv:2202.04901* (2022). 2, 7

[ROFP22] Rozumnyi D., Oswald M. R., Ferrari V., Pollefeys M.: Motion-from-blur: 3d shape and motion estimation of motion-blurred objects in videos. In *Proc. CVPR* (2022), pp. 15990–15999. 3

[RRKS19] Rebecq H., Ranftl R., Koltun V., Scaramuzza D.: High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence 43*, 6 (2019), 1964–1980. 3

[RSD*19] Reda F. A., Sun D., Dundar A., Shoeybi M., Liu G., Shih K. J., Tao A., Kautz J., Catanzaro B.: Unsupervised video interpolation using cycle consistency. In *Proc. ICCV* (2019), pp. 892–900. 2

[RSSF02] Reinhard E., Stark M., Shirley P., Ferwerda J.: Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 267–276. 8

[Sam22] Samsung: ISOCELL GN1 sensors, 2022. 2, 3, 11

[SBZ*20a] Shen W., Bao W., Zhai G., Chen L., Min X., Gao Z.: Blurry video frame interpolation. In *Proc. CVPR* (2020), pp. 5114–5123. 2, 3

[SBZ*20b] Shen W., Bao W., Zhai G., Chen L., Min X., Gao Z.: Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Trans Image Proc 30* (2020), 277–292. 3

[SDW*17] Su S., Delbracio M., Wang J., Sapiro G., Heidrich W., Wang O.: Deep video deblurring for hand-held cameras. In *Proc. CVPR* (2017), pp. 1279–1288. 6, 7

[SHG*16] Serrano A., Heide F., Gutierrez D., Wetzstein G., Masia B.: Convolutional sparse coding for high dynamic range imaging. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 153–163. 3

[SOK21] Sim H., Oh J., Kim M.: XVFI: Extreme video frame interpolation. In *Proc. ICCV* (2021), pp. 14489–14498. 2, 6, 7

[Son22] Sony: Quad Bayer coding, 2022. 2, 3, 11

[SSR09] Schoueri Y., Scaccia M., Rekleitis I.: Optical flow from motion blurred color images. In *2009 Canadian Conference on Computer and Robot Vision* (2009), pp. 1–7. 3

[SYLK18] Sun D., Yang X., Liu M.-Y., Kautz J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR* (2018), pp. 8934–8943. 5, 7

[TD20] Teed Z., Deng J.: RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV* (2020), pp. 402–419. 4, 5, 6, 8, 10

[WCB*22] Wang Z., Cun X., Bao J., Zhou W., Liu J., Li H.: Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 17683–17693. 11

[WY21] Wang L., Yoon K.-J.: Deep learning for HDR imaging: State-of-the-art and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 2

[XSS*19] Xu X., Siyao L., Sun W., Yin Q., Yang M.-H.: Quadratic video interpolation. *Advances in Neural Information Processing Systems 32* (2019). 2, 3, 4, 5, 7

[YZL*20] Yan Q., Zhang L., Liu Y., Zhu Y., Sun J., Shi Q., Zhang Y.: Deep HDR imaging via a non-local network. *IEEE Trans. Image Proc. 29* (2020), 4308–4322. 3

[ZAK*22] Zamir S. W., Arora A., Khan S., Hayat M., Khan F. S., Yang M.-H.: Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5728–5739. 11

[ZRL*22] Zhang K., Ren W., Luo W., Lai W.-S., Stenger B., Yang M.-H., Li H.: Deep image deblurring: A survey. *International Journal of Computer Vision 130*, 9 (2022), 2103–2130. 3

[ZWT20] Zhang Y., Wang C., Tao D.: Video frame interpolation without temporal priors. *Advances in Neural Information Processing Systems 33* (2020), 13308–13318. 2, 3, 10