VIDEO FRAME INTERPOLATION FOR HIGH DYNAMIC RANGE SEQUENCES CAPTURED WITH DUAL-EXPOSURE SENSORS

Ugur Cogalan, Mojtaba Bemana, Hans-Peter Seidel, Karol Myszkowski
Max Planck Institute for Informatics
Saarbrücken
{ucogalan, mbemana, hpseidel, karol}@mpi-inf.mpg.de

ABSTRACT

Video frame interpolation (VFI) enables many important applications that might involve the temporal domain, such as slow motion playback, or the spatial domain, such as stop motion sequences. We are focusing on the former task, where one of the key challenges is handling high dynamic range (HDR) scenes in the presence of complex motion. To this end, we explore 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 the test time. Supplementary videos can be found at: deephdr.mpi-inf.mpg.de/.

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 [Jiang et al.(2018), Bao et al.(2019), Sim et al.(2021)] 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 [Reda et al.(2022), Sim et al.(2021)] 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 [Xu et al.(2019), Park et al.(2021)]; 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 [Shen et al.(2020), Zhang et al.(2020)], 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 [Sony(1999), Carey et al.(2013)] greatly expand the dynamic range of contrast in captured scenes [Hajsharif et al.(2014), Go et al.(2019), Choi et al.(2017), Heide et al.(2014), Cogalan et al.(2022), Nguyen et al.(2022), Cho et al.(2014)]. 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 [CVM(accessed on Sept. 17, 2021)]. Importantly, while the exposure duration differs, the exposure completion is temporally aligned, which enables recovering two

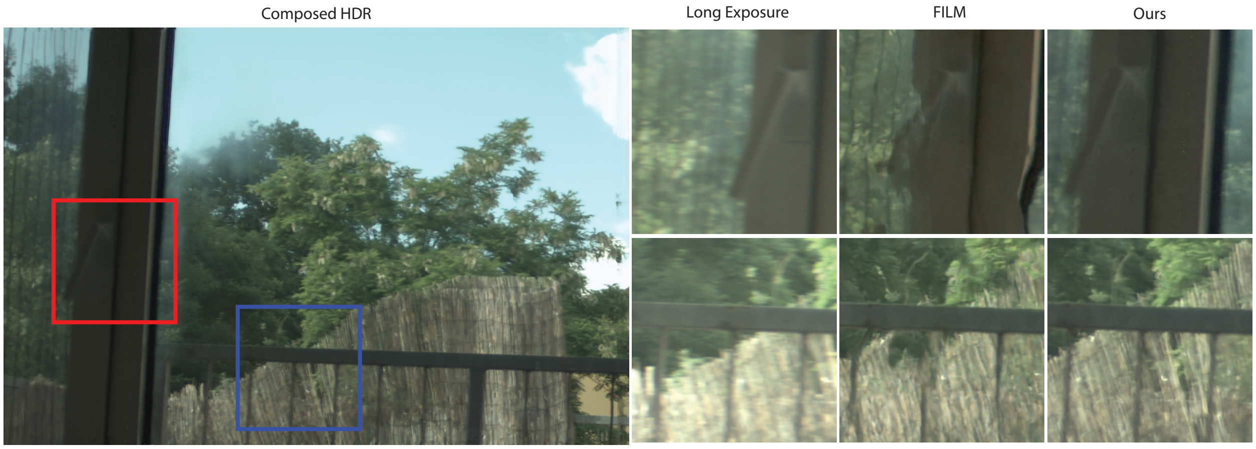

Figure 1: Video frame interpolation (VFI) in the framerate extension task using the state-of-the-art FILM [Reda et al.(2022)] and our methods. The first column shows our in-between frame reconstruction, while the insets focus on the dark scene details that require long exposure durations and feature significant motion blur due to camera motion (second column). Such blur adversely affects VFI for the FILM method (third column). Our method employs temporally continuous information on the scene motion that is encoded in motion blur to improve the VFI quality (fourth column).

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 high dynamic range 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.

The key contributions of our work are:

- We propose a machine learning solution for VFI that can handle high-dynamic-range 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 exploit motion blur information inherent in the long exposure to further improve motion estimation quality.
- We develop a metric of motion complexity that provides interesting insights into existing datasets used in the training of VFI methods.

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 a meaningful comparison of existing VFI methods, while Sec. 5 provides implementation details of our network. Sec. 6 confronts 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 either sharp (Sec. 2.1) or blurry (Sec. 2.2) input video. We focus on the problems of handling non-uniform motion and high-dynamic-range content (Sec. 2.3) that are central to this work. We refer the reader to a recent survey where a more complete treatment of VFI is presented [Parihar et al.(2021)].

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 [Niklaus et al.(2017)] 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 [Jiang et al.(2018)] uses bi-directional flows and an occlusion map to synthesize intermediate frames at any arbitrate time. DAIN [Bao et al.(2019)] 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 [Liu et al.(2019)] or without any supervision [Reda et al.(2019)]. BMBC [Park et al.(2020)] warps the input frames with a proposed bilateral motion model and combines them using learned dynamic blending filters. CAIN [Choi et al.(2020)] uses a channel attention module to interpolate video frames without the need for estimation of motion. SoftSplat [Niklaus and Liu(2020)] proposes differentiable forward warping via softmax splatting and shows its benefits for VFI. AdaCoF [Lee et al.(2020)] 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 [Sim et al.(2021)] presents a high-speed (1000fps) video dataset and proposes a multi-scale recursive approach to handle large motion in the scene. Recently, FILM [Reda et al.(2022)] 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 [Xu et al.(2019)] is one of the first video interpolation methods to model curvilinear motion with the quadratic equation using four temporal frames. Chi et al. [Chi et al.(2020)] extend QVI by introducing an additional cubic term that accounts for the change in the acceleration. ABME [Park et al.(2021)] handles the non-uniform motion in the scene by extending the BMBC [Park et al.(2020)] 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 for the longer exposure as an additional cue to the flow estimation. Moreover, by explicitly accounting for motion non-uniformity statistics in our training data, we can improve our network's ability to handle a wide variation of such motions.

2.2 Joint Video Deblurring and Interpolation

It is inevitable to have motion blur during capturing caused by low shutter speed, long exposure time, or rapid movement in the scene. Recent works demonstrate that a joint deblurring and frame interpolation greatly improves the resulting VFI quality over an independent treatment of these tasks. TNTT [Jin et al.(2019)] adopts a joint optimization scheme to extract sharp keyframes from the blurry input frames and then smoothly interpolate between the extracted keyframes. BIN [Shen et al.(2020)] and its extended version PRF [Shen et al.(2020)] simultaneously remove the motion blur and interpolate the in-between frames by employing a recurrent pyramid framework to efficiently aggregate the temporal information. ALANET [Gupta et al.(2020)] relaxes the strong assumption that all the input frames in a captured video are blurry and adapts attention mechanisms to decide on deblurring each frame based on the information from the neighbor frames. These methods mainly perform central frame interpolation, meaning they need to be recursively applied in order to interpolate an arbitrary frame in time. Moreover, in these methods, the blurry input frames are simply created by averaging neighbor frames of existing high framerate videos with mutually overlapping temporal windows, and without accounting for the camera readout time. Those two factors can significantly hinder the reconstruction of physically correct blur. UTI-VFI [Zhang et al.(2020)] periodically skips selected frames in the input high framerate video to emulate the lost information during the sensor readout time, and it can also generate the intermediate frames at an arbitrary time. It first extracts two sharp keyframes corresponding to the start and the end of a given blurry frame sequence, and then it utilizes an off-the-shelf optical flow network to interpolate in-between frames. Moreover, it adapts quadratic motion formulation in order to properly handle non-uniform motion. Rengarajan et al. [Rengarajan et al.(2020)] take a triplet of short-long-short exposures captured by a programmable machine vision camera and recover a sharp high framerate sequence. In their capturing setup, the short exposures are sharp but noisy and can act as a pivot for deblurring the long exposure. While these methods attempt to remove the motion blur in the video frames, the inherent motion blur can potentially reveal information about the magnitude and direction of the motion, especially in the case of large non-uniform motion. In this work, we explicitly extract the motion flow from the blurry input frames. Moreover, large motions could make the joint deblurring and VFI tasks very challenging. In our approach, we take advantage of the less blurry short exposure in each frame to make the flow estimation more reliable.

2.3 High dynamic range video

HDR video reconstruction is 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 [Kalantari et al.(2013), Kalantari et al.(2017), Kalantari and Ramamoorthi(2019), Yan et al.(2020), Chen

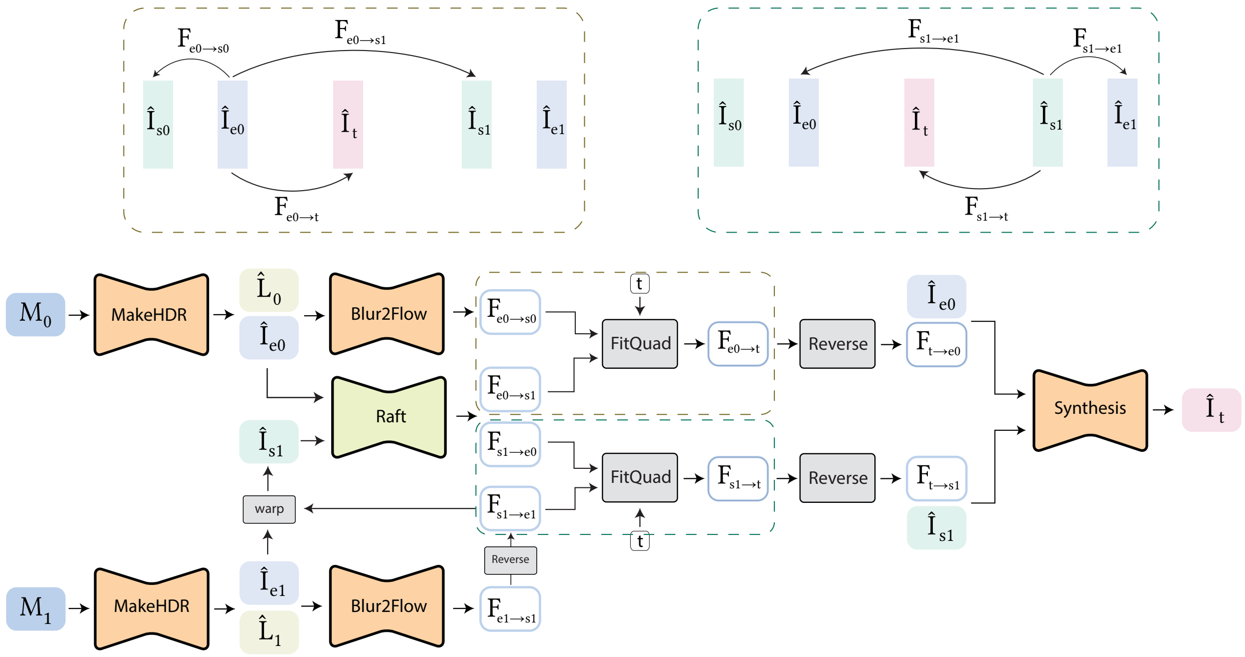

Figure 2: Our HDR VFI pipeline takes as the input two subsequent frames $\mathtt{M}_0$ and $\mathtt{M}_1$ as captured using our dual-exposure sensor. The input frames are independently processed by a learned **MakeHDR** network, so that the corresponding sharp HDR images $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ are aligned with the short exposures, as well as blurry long exposure images $\hat{\mathtt{L}}_0$ and $\hat{\mathtt{L}}_1$, whose ends are aligned with $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ are obtained. Next, the motion flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}$ and $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}$ are extracted from the blur inherent to the long exposures $\hat{\mathtt{L}}_0$ and $\hat{\mathtt{L}}_1$ using a learned **Blur2Flow** network. Also, the motion flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}$ estimated between the sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ in both directions using the state-of-the-art flow estimation method Raft [Teed and Deng(2020)]. Then, the motion flow pairs $\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}$ and $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$, as well as $\mathtt{F}_{\mathtt{s1}\to\mathtt{e1}}$ (reverted from $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}$) and $\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}$ are independently fed to a non-learnable **FitQuad** module to calculate the forward flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{t}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{t}}$ that are parametrized using a quadratic motion model for a position t. Then, we compute the backward flows $\mathtt{F}_{\mathtt{t}\to\mathtt{e0}}$ and $\mathtt{F}_{\mathtt{t}\to\mathtt{s1}}$ using a differentiable flow **Reverse** module [Xu et al.(2019)] in order to warp the frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ to the position t. Finally, the warped images are combined using a trained **Synthesis** network to produce the output frame $\hat{\mathtt{I}}_{\mathtt{t}}$.

et al.(2021)]. To alleviate the need for such alignment, single-shot HDR techniques are developed that rely on specialized dual-ISO sensors [Hajsharif et al.(2014), Go et al.(2019), Choi et al.(2017), Cogalan and Akyuz(2020)]. Dual-exposure sensors [CVM(accessed on Sept. 17, 2021)] offer two motion samples per frame, which, as we show in this work, is greatly beneficial for VFI, albeit requires long-exposure deblurring in the HDR video reconstruction [Heide et al.(2014), Cogalan et al.(2022), Nguyen et al.(2022), Cho et al.(2014)]. The scope of 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. [Rebecq et al.(2019)], 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 method for video frame interpolation that reconstructs high-dynamic range frames in the continuous time domain. Fig. 2 summarizes our processing pipeline and the following paragraphs provide more detailed description of its key components. Our method takes as an 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 a suffix i we denote any input frame, contains a pair of spatially interleaved short and long exposures, and it is processed by the **MakeHDR** network to produce a sharp HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$ that is aligned with the short exposure, and a blurry long exposure frame $\hat{\mathtt{L}}_{\mathtt{i}}$, whose end is aligned with $\hat{\mathtt{I}}_{\mathtt{ei}}$. Both frames are fed to the **Blur2Flow** network to predict the flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}$ that extracts the motion from the blur within the $\hat{\mathtt{L}}_{\mathtt{i}}$ frame. Next, we compute the flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}$ between the sharp HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ in both directions using an off-the-shelf flow estimation method such as Raft [Teed and Deng(2020)]. Note that the frame $\hat{\mathtt{I}}_{\mathtt{s1}}$

is reconstructed by warping the $\hat{\mathtt{I}}_{\mathtt{e1}}$ using the backward flow $\mathtt{F}_{\mathtt{s1}\to\mathtt{e1}}$, where a **`Reverse`** module for differentiable flow reversal [Xu et al.(2019)] is applied to the forward flow $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}$, as originally derived by **`Blur2Flow`** network. Given two estimated flows per each frame ($\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}$, $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$ for $\mathtt{M}_0$, and $\mathtt{F}_{\mathtt{s1}\to\mathtt{e1}}$, $\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}$ for $\mathtt{M}_1$), we fit a quadratic motion model [Xu et al.(2019)] using a non-learnable **`FitQuad`** module, and compute the forward flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{t}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{t}}$. Then, we compute the backward flows $\mathtt{F}_{\mathtt{t}\to\mathtt{e0}}$ and $\mathtt{F}_{\mathtt{t}\to\mathtt{s1}}$ using the **`Reverse`** module. Finally, the frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ are warped with the backward flows and combined using the **`Synthesis`** module to interpolate the frame $\hat{\mathtt{I}}_{\mathtt{t}}$.

HDR reconstruction: `MakeHDR` We acquire our input video using a dual-exposure sensor [CVM(accessed on Sept. 17, 2021)] 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 [Cogalan et al.(2022)] to jointly deblur, denoise, and upsample our input frames $\mathtt{M}_{\mathtt{i}}$ to produce sharp, clean, and full-resolution short and long exposures. Both exposures are then combined using a non-learnable technique, similar to [Debevec and Malik(2008)], 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` Existing motion blur in the long exposure can potentially reveal information about the motion in the scene [Shen et al.(2020), Rengarajan et al.(2020)]. Using our **`Blur2Flow`** network, we aim to recover the motion flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}$ that is responsible for the blur pattern registered in the long exposure $\hat{\mathtt{L}}_{\mathtt{i}}$. The sensor design imposes 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. Since we use $\hat{\mathtt{I}}_{\mathtt{ei}}$ as a reference for the predicted flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}$ alignment and given that its time span is equivalent to the long exposure duration, an additional per-pixel temporal motion sample at the beginning of long exposure is readily available by warping $\hat{\mathtt{I}}_{\mathtt{ei}}$ to $\hat{\mathtt{I}}_{\mathtt{si}}$.

Flow reversal: `Reverse` As the forward warping of the HDR frame $\hat{\mathtt{I}}_{\mathtt{ei}}$ to $\hat{\mathtt{I}}_{\mathtt{si}}$ using the forward flow $\mathtt{F}_{\mathtt{ei}\to\mathtt{si}}$ is not differentiable, we compute the backward flow $\mathtt{F}_{\mathtt{si}\to\mathtt{ei}}$ using a flow reversal module introduced in QVI [Xu et al.(2019)].

Quadratic motion model: `FitQuad` In this step, we first **`Reverse`** the forward flow $\mathtt{F}_{\mathtt{e1}\to\mathtt{s1}}$, as derived by **`Blur2Flow`**, to produce the backward flow $\mathtt{F}_{\mathtt{s1}\to\mathtt{e1}}$ that we use to warp $\hat{\mathtt{I}}_{\mathtt{e1}}$ into $\hat{\mathtt{I}}_{\mathtt{s1}}$. Then, we compute the bidirectional flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}$ between the HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ using a state-of-the-art flow estimation method as proposed in [Teed and Deng(2020)]. We found that in particular, for complex motion, using $\hat{\mathtt{I}}_{\mathtt{s1}}$, instead of $\hat{\mathtt{I}}_{\mathtt{e1}}$, eases the bidirectional flow computation and improves its quality, as the time span with respect to $\hat{\mathtt{I}}_{\mathtt{e0}}$ is reduced. Now, to warp $\hat{\mathtt{I}}_{\mathtt{e0}}$ to a given time point t, we employ the flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}$ and $\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}}$ to derive a quadratic motion model as:

$$\mathtt{F}_{\mathtt{e0}\to\mathtt{t}} = \frac{1}{2}\mathtt{a}_0 \times \mathtt{t}^2 + \mathtt{v}_0 \times \mathtt{t} \tag{1}$$

where the acceleration $a_0 = \dfrac{2(\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}} + \lambda \mathtt{F}_{\mathtt{e0}\to\mathtt{s0}})}{\lambda^2 + \lambda}$ and the velocity $v_0 = \dfrac{\mathtt{F}_{\mathtt{e0}\to\mathtt{s1}} - \lambda^2 \mathtt{F}_{\mathtt{e0}\to\mathtt{s0}}}{\lambda^2 + \lambda}$ directly depend on the ratio λ between the camera readout time and its complete duty cycle that is required for capturing each frame. Similarly, to warp $\hat{\mathtt{I}}_{\mathtt{s1}}$ we compute the flow $\mathtt{F}_{\mathtt{s1}\to\mathtt{t}}$:

$$\mathtt{F}_{\mathtt{s1}\to\mathtt{t}} = \frac{1}{2}\mathtt{a}_1 \times \mathtt{t}^2 + \mathtt{v}_1 \times \mathtt{t} \tag{2}$$

where $a_1 = \dfrac{2(\mathtt{F}_{\mathtt{s1}\to\mathtt{e0}} + \lambda \mathtt{F}_{\mathtt{s1}\to\mathtt{e1}})}{\lambda^2 + \lambda}$ and $v_1 = \dfrac{\lambda^2 \mathtt{F}_{\mathtt{s1}\to\mathtt{e1}} - \mathtt{F}_{\mathtt{s1}\to\mathtt{e0}}}{\lambda^2 + \lambda}$.

Intermediate frame synthesis: `Synthesis` As the forward warping of HDR frames $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}}$ to a novel position t using the flows $\mathtt{F}_{\mathtt{e0}\to\mathtt{t}}$ and $\mathtt{F}_{\mathtt{s1}\to\mathtt{t}}$ is not differentiable, we use the **`Reverse`** module to compute the backward flows $\mathtt{F}_{\mathtt{t}\to\mathtt{e0}}$ and $\mathtt{F}_{\mathtt{t}\to\mathtt{s1}}$. Then, we employ a differentiable warping operator `warp` [Jaderberg et al.(2015)] as:

$$\hat{\mathtt{I}}_{\mathtt{e0}\to\mathtt{t}} = \mathtt{warp}(\hat{\mathtt{I}}_{\mathtt{e0}}, \mathtt{F}_{\mathtt{t}\to\mathtt{e0}}) \text{ and } \hat{\mathtt{I}}_{\mathtt{s1}\to\mathtt{t}} = \mathtt{warp}(\hat{\mathtt{I}}_{\mathtt{s1}}, \mathtt{F}_{\mathtt{t}\to\mathtt{s1}}). \tag{3}$$

Afterwards, the warped frames along with the backward flows are fed to the **`Synthesis`** network:

$$\alpha = \mathbf{Synthesis}(\hat{\mathtt{I}}_{\mathtt{e0}\to\mathtt{t}}, \hat{\mathtt{I}}_{\mathtt{s1}\to\mathtt{t}}, \mathtt{F}_{\mathtt{t}\to\mathtt{e0}}, \mathtt{F}_{\mathtt{t}\to\mathtt{s1}}) \tag{4}$$

which predicts the soft occlusion weight α that controls the contribution of input warped images $\hat{\mathtt{I}}_{\mathtt{e0}\to\mathtt{t}}$ and $\hat{\mathtt{I}}_{\mathtt{s1}\to\mathtt{t}}$ for each pixel. Finally, we synthesize the interpolated frame $\hat{\mathtt{I}}_{\mathtt{t}}$ as:

$$\hat{\mathtt{I}}_{\mathtt{t}} = \alpha \odot \hat{\mathtt{I}}_{\mathtt{e0}\to\mathtt{t}} + (1 - \alpha) \odot \hat{\mathtt{I}}_{\mathtt{s1}\to\mathtt{t}} \tag{5}$$

Table 1: The statistics of scenes with uniform and non-uniform moving content for the Adobe240 [Su et al.(2017)], GoPro [Nah et al.(2017)], X4K1000FPS [Sim et al.(2021)], and SlowFlow [Janai et al.(2017)] datasets.

	Adobe 240	**GoPro**	**X4K1000FPS**	**SlowFlow**
Uniform	26 (20%)	2 (6%)	108 (98%)	14 (35%)
Non-uniform	107 (80%)	31 (94%)	2 (2%)	25 (65%)

where $\odot$ stands for per pixel multiplication.

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

$$\mathtt{L}_{\text{hdr}} = \sum_{i=0,1} \|\mathtt{I_{ei}} - \hat{\mathtt{I}}_{\mathtt{ei}}\|_1 + \|\mathtt{L_i} - \hat{\mathtt{L}}_{\mathtt{i}}\|_1 \tag{6}$$

As the ground truth flow is not available, we supervise the **`Blur2Flow`** network using the re-projection loss:

$$\mathtt{L}_{\text{flow}} = \sum_{i=0,1} \|\mathtt{I_{ei}} - \mathtt{warp}(\mathtt{I_{si}}, \mathtt{F_{ei\to si}})\|_1 \tag{7}$$

where the ground truth frame $\mathtt{I_{si}}$ is warped using the flow $\mathtt{F_{ei\to si}}$ to align it with the ground truth frame $\mathtt{I_{ei}}$. Finally, we supervise the **`Synthesis`** network using the reconstruction loss:

$$\mathtt{L}_{\text{synth}} = \|\mathtt{I_t} - \hat{\mathtt{I}}_{\mathtt{t}}\|_1 \tag{8}$$

where the interpolated frame $\hat{\mathtt{I}}_{\mathtt{t}}$ and the corresponding ground truth $\mathtt{I_t}$ are compared. The final loss $\mathtt{L}_{\text{total}}$ is then computed as:

$$\mathtt{L}_{\text{total}} = \mathtt{L}_{\text{hdr}} + \mathtt{L}_{\text{flow}} + \mathtt{L}_{\text{synth}} \tag{9}$$

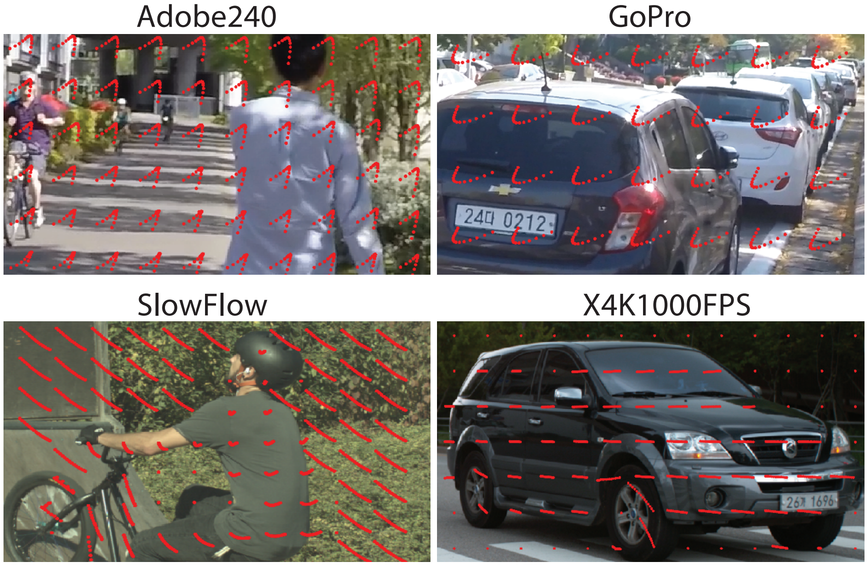

Figure 3: Trajectories of pixels (red dots) for 16 consecutive frames in a sample scene from different dataset. The scenes in Adobe240 and GoPro dataset mostly have globally non-uniform motion due to non-uniform camera motion while in dataset such as X4K1000FPS and SlowFlow, the scenes mostly contain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

4 Motion non-uniformity analysis

In order to properly validate our proposed method, we need to assure that our dataset contains diverse examples of scene motion non-uniformity. To this end, we perform an analysis of motion non-uniformity in some of popular

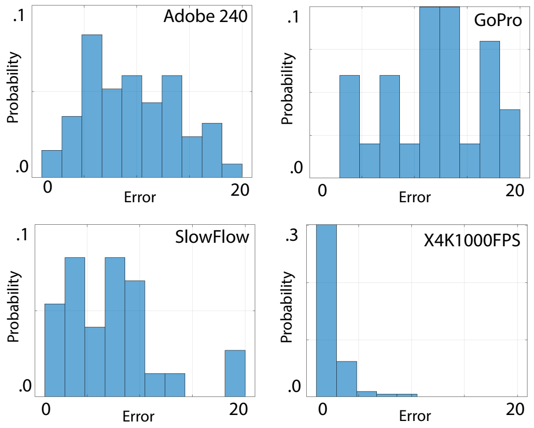

Figure 4: The histogram of measured non-uniform motions for different datasets, where the horizontal axis shows the actual motion error ($\times 10^{-2}$) with respect to the linear motion fit, and the vertical axis denotes the probability of the observed frames given an error value.

high-framerate video datasets including Adobe240 [Su et al.(2017)], GoPro [Nah et al.(2017)], X4K1000FPS [Sim et al.(2021)], and SlowFlow [Janai et al.(2017)]. Our procedure is as follows: For each pixel in a given frame we use Raft [Teed and Deng(2020)] to track the corresponding pixels for $N = 15$ consecutive frames. 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 [Jonschkowski et al.(2020)] 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. In the next step, we find a linear model that in the least square sense fits to 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. Fig. 3 shows the trajectories of pixels for a sample scene containing regions with non-uniform motion in each dataset. 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 90th 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 ($\times 10^{-2}$), and the vertical axis is the probability of observing scene for a given error value. In our experiments, we identify a scene as containing the non-uniform motion, when the average error is over 6.0×10^{-2} (the threshold value is chosen empirically based on our visual inspection of the pixel trajectories as illustrated in Fig. 3). Based on this threshold, we provide statistics of scenes with uniform and non-uniform motions for each dataset in Tbl. 1. 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 [Cogalan et al.(2022)]. The network output is given in the Bayer domain, and we apply demosaicing using OpenCV [Bradski(2000)], followed by a gamma correction to create the final short and long exposures in the sRGB format. The **Blur2Flow** network employs a network architecture similar to the PWCNet [Sun et al.(2017)]. Our **Synthesis** 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 component 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.

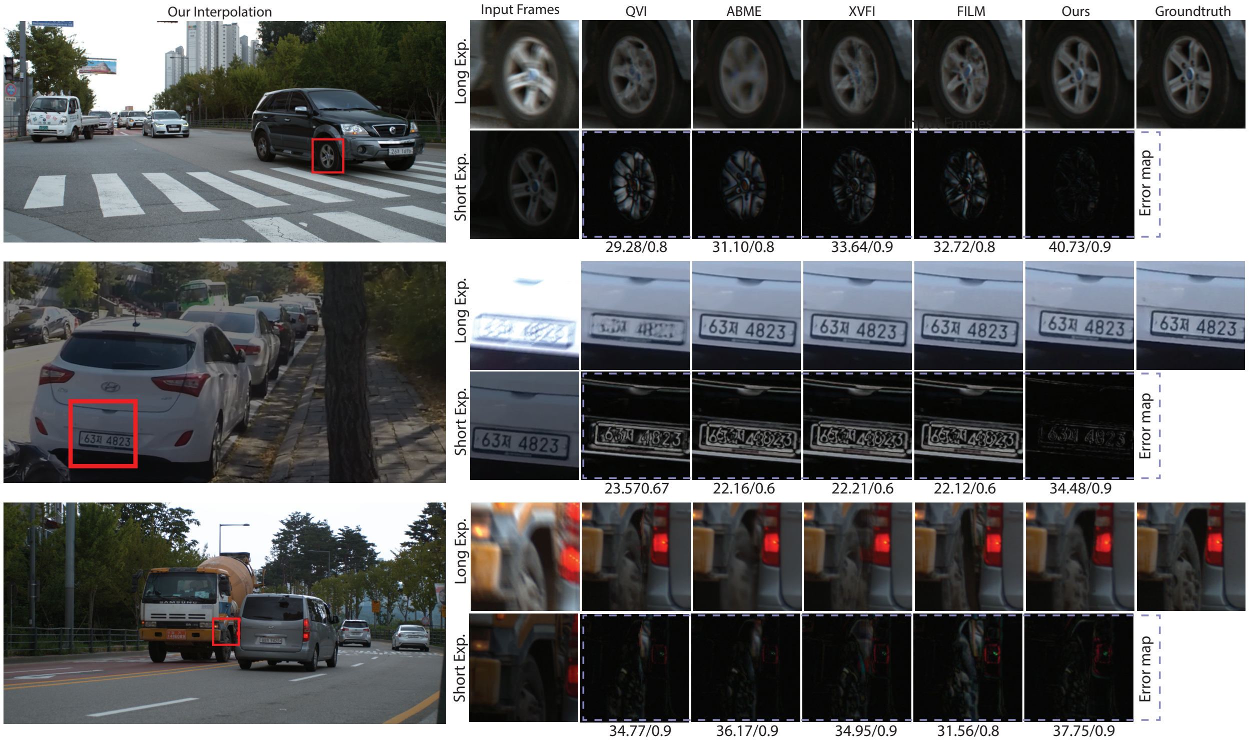

Figure 5: Visual comparisons of our method with the state-of-the-art VFI methods using the synthetic dataset described in Sec. 6.1. For each of three scenes, the first row of insets shows the performance of respective VFI methods, while the second row presents the corresponding per-pixel error maps between 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.

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. In our experiments, we take the scenes from X4K1000FPS [Sim et al.(2021)] and SlowFlow [Janai et al.(2017)] as our training dataset, and we consider Adobe240 [Su et al.(2017)] and GoPro [Nah et al.(2017)] as our evaluation dataset. 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 frame 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. For our test set, we are interested in evaluating our method for a different dataset containing the scenes with solely uniform and non-uniform motion; hence we split all the scenes in the Adobe240 [Su et al.(2017)] and GoPro [Nah et al.(2017)] datasets into uniform and non-uniform sets based on our metric discussed in Sec. 4.

6.2 Quantitative comparison

We compare our proposed method with state-of-the-art sharp VFI methods (refer to Sec. 2.1): FILM [Reda et al.(2022)] and XVFI [Sim et al.(2021)] that rely on a uniform motion assumption, and QVI [Xu et al.(2019)] and ABME [Park

Table 2: Quantitative comparison of our method with state-of-the-art video frame interpolation 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.

	Uniform		**Non-uniform**	
Methods	PSNR	SSIM	PSNR	SSIM
ABME [Park et al.(2021)]	33.16	0.88	31.04	0.83
QVI [Xu et al.(2019)]	32.10	0.89	31.32	0.85
XVFI [Sim et al.(2021)]	32.56	0.87	30.88	0.82
FILM [Reda et al.(2022)]	32.89	0.87	30.90	0.82
Ours	**35.97**	**0.94**	**35.57**	**0.93**

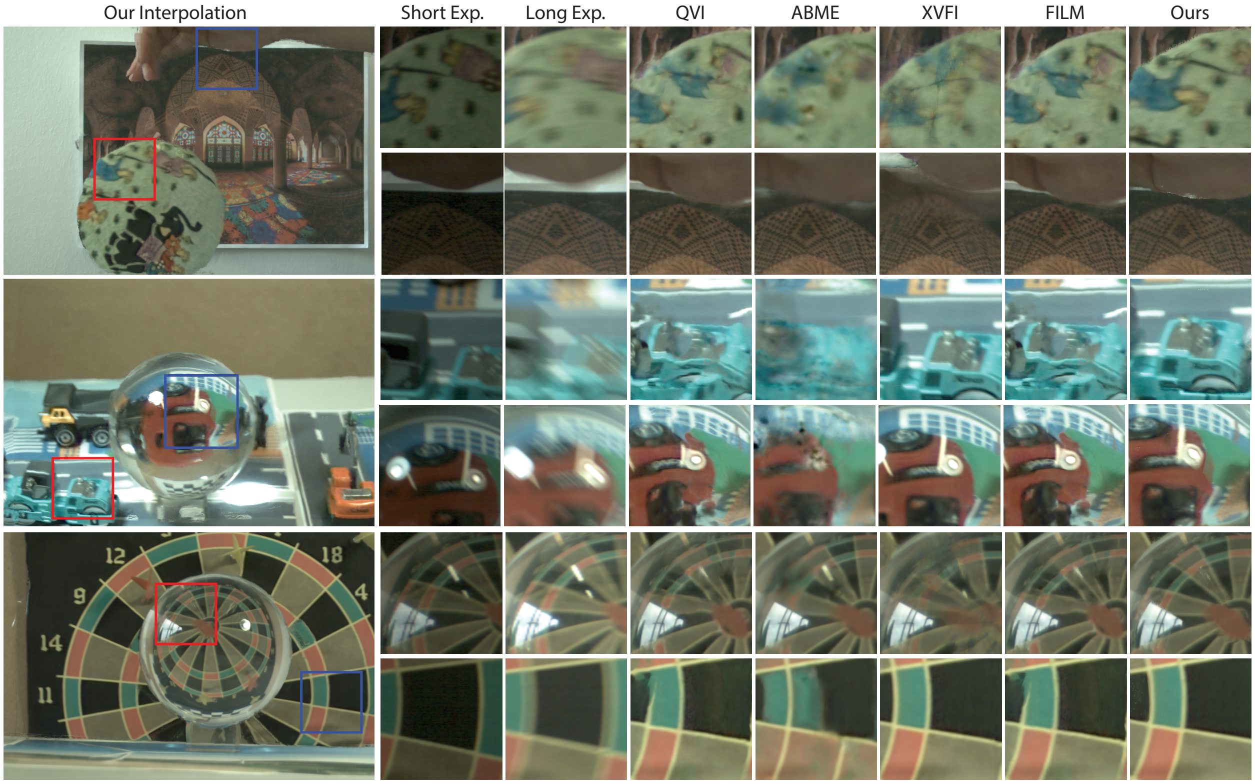

Figure 6: The visual comparisons of our interpolation results for three scenes captured using our camera with a dual-exposure sensor. Our method is able to correctly interpolate the frames in the scenes with the challenging cases of a rolling round box (the upper scene), a rotary camera motion (the middle scene) and a moving object behind a refractive object (the bottom scene).

et al.(2021)] that explicitly support non-uniform motion. FILM, XVFI, and AMBE require just two neighboring frames as an input, while the method QVI needs to process four consecutive frames. Since these VFI methods expect post-processed sRGB images as an input, and moreover, we are interested in evaluating our interpolation part, we feed the sharp HDR output to the sharp frame interpolation methods. Note that we are unable to compare with the blurry VFI methods (refer to Sec. 2.2), as they require well-exposed blurry input frames (effectively, blurry HDR frames) while our long exposure typically contains a considerable amount of saturation that is poorly handled by these methods. Table 1 summarizes our comparisons with the VFI methods for the aggregated Adobe240 and GoPro datasets that we split into two categories with respect to the motion uniformity (Sec. 6.1). For the dataset with uniform motion, the competing VFI methods perform similarly, while for the non-uniform motion dataset, the methods with a non-uniform motion assumption (AMBE and QVI) perform, as expected, better compared to ones with a uniform motion assumption (FILM and XVFI). In any case, our method outperforms the existing VFI methods by a large margin.

6.3 Qualitative comparison

We first provide visual comparisons with the state-of-the-art VFI methods for three scenes synthesized data with ground truth in Fig. 5. Moreover, in Fig. 1 and Fig. 6 we visualize the results for four captured scenes using our Axiom-beta camera with a CMOSIS CMV12000 sensor. In both setups, we use the exposure ratio of 4 between the short and long exposures. Since the captured frames using our camera cannot be fed directly to the other VFI methods, we provide them with the reconstructed sharp HDR images $\hat{\mathtt{I}}_{\mathtt{e0}}$ and $\hat{\mathtt{I}}_{\mathtt{e1}}$ using our **MakeHDR** network. In Fig. 1, we have an example a scene with high dynamic range content where there is a moving camera with acceleration during capturing; the magnitude of the motion blur in the long exposure can show the amount of camera motion. In this case, we can see a method such as FILM drastically fails, while our method employs the encoded blur information to improve the interpolation quality. The upper scene in the Fig. 6 shows an example of a rolling box in which the existing VFI methods even the ones designed to deal with non-uniform motion such as AMBE and QVI fail to properly interpolate an intermediate frame due to non-uniform motion caused by rotatory motion of the round box. 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 a 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.

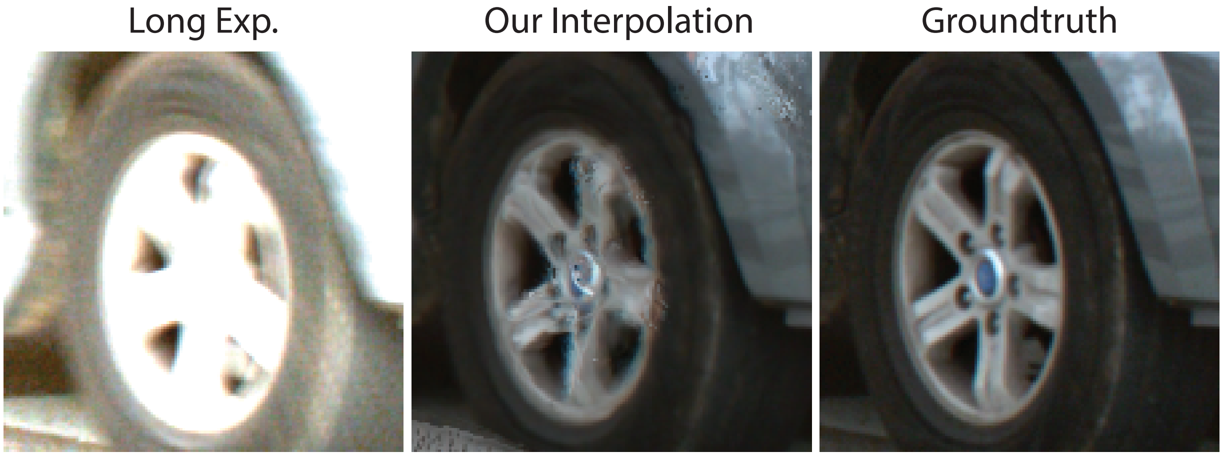

Figure 7: Our interpolation failure example in case the moving content is largely saturated in the long exposure.

Table 3: The effect of including the scenes with non-uniform motion in our training dataset.

	Uniform		Non-uniform	
Methods	PSNR	SSIM	PSNR	SSIM
Ours-uniform	34.86	0.93	34.42	0.91
Ours-non-uniform	34.42	0.93	34.10	0.91

6.4 Ablation study

We perform an experiment to show the effect of including the scenes with non-uniform motion in our training dataset. We summarize the results in table Tbl. 3. We first train our method using only the scenes with uniform motion (first row in the Tbl. 3) and train using scenes with non-uniform motion (in the second row). Our results show no significant difference in the performance of our method when trained using the scenes with non-uniform motion. This is due to the fact that the quadratic motion fitting part in our method does not have any learnable parameters that could benefit from non-uniform motion samples in the training set.

Moreover, 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 [Teed and Deng(2020)]. In this case, we linearly split each flow to reach to any position t between the frames. The results are shown in the Tbl. 4. The higher number for our full method shows the effectiveness of including the **Blur2Flow** network in our pipeline.

Table 4: The effect of **Blur2Flow** network in our pipeline.

	Uniform		**Non-uniform**	
Methods	PSNR	SSIM	PSNR	SSIM
Ours-w/o blur2flow	31.51	0.86	30.10	0.81
Ours-full	**35.97**	**0.94**	**35.57**	**0.93**

6.5 Limitation

Saturation is inevitable in the long exposure for bright scene regions. If the moving content is present in such regions, our flow prediction using the **Blur2Flow** network becomes less accurate. Fig. 7 shows an example of this case where we synthetically increase the saturation in the long exposure for the wheel example shown in Fig. 5 and our method fails to correctly reconstruct the intermediate frame.

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 [Cogalan et al.(2022)], which could adversely affect the accuracy of HDR video interpolation.

7 Conclusion

In this work, we presented a method for high-dynamic-range video frame interpolation using the 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

[Bao et al.(2019)] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 1, 3

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

[Carey et al.(2013)] Stephen J Carey, Alexey Lopich, David RW Barr, Bin Wang, and Piotr Dudek. A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array. In *2013 Symposium on VLSI Circuits*, pages C182–C183. IEEE, 2013. 1

[Chen et al.(2021)] Guanying Chen, Chaofeng Chen, Shi Guo, Zhetong Liang, Kwan-Yee K Wong, and Lei Zhang. Hdr video reconstruction: A coarse-to-fine network and a real-world benchmark dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2502–2511, 2021. 3

[Chi et al.(2020)] 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 *European Conference on Computer Vision*, pages 107–123. Springer, 2020. 3

[Cho et al.(2014)] Hojin Cho, Seon Joo Kim, and Seungyong Lee. Single-shot high dynamic range imaging using coded electronic shutter. *Comp. Graph. Forum*, 33(7):329–338, 2014. 1, 4

[Choi et al.(2017)] Inchang Choi, Seung-Hwan Baek, and Min H Kim. Reconstructing interlaced high-dynamic-range video using joint learning. *IEEE Transactions on Image Processing*, 26(11):5353–5366, 2017. 1, 4

[Choi et al.(2020)] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020. 3

[Cogalan and Akyuz(2020)] U. Cogalan and A. O. Akyuz. Deep joint deinterlacing and denoising for single shot dual-ISO HDR reconstruction. *IEEE Trans. Image Processing*, 29:7511–7524, 2020. 4

[Cogalan et al.(2022)] Ugur Cogalan, Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. Learning hdr video reconstruction for dual-exposure sensors with temporally-alternating exposures. *Computers & Graphics*, 2022. 1, 4, 5, 7, 11

[CVM(accessed on Sept. 17, 2021)] CMOSIS CVM. https://ams.com/cmv12000, accessed on Sept. 17, 2021. 1, 4, 5

[Debevec and Malik(2008)] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, pages 1–10. 2008. 5

[Go et al.(2019)] Chihiro Go, Yuma Kinoshita, Sayaka Shiota, and Hitoshi Kiya. 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):1856–1864, 2019. 1, 4

[Gupta et al.(2020)] Akash Gupta, Abhishek Aich, and Amit K Roy-Chowdhury. Alanet: Adaptive latent attention network forjoint video deblurring and interpolation. *arXiv preprint arXiv:2009.01005*, 2020. 3

[Hajsharif et al.(2014)] Saghi Hajsharif, Joel Kronander, and Jonas Unger. Hdr reconstruction for alternating gain (iso) sensor readout. In *Eurographics, Strasbourg, France, April 7-11, 2014*, 2014. 1, 4

[Heide et al.(2014)] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pająk, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (ToG)*, 33(6):1–13, 2014. 1, 4

[Jaderberg et al.(2015)] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 5

[Janai et al.(2017)] Joel Janai, Fatma Guney, Jonas Wulff, Michael J Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3597–3607, 2017. 6, 7, 8

[Jiang et al.(2018)] Huaizu 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018. 1, 3

[Jin et al.(2019)] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8112–8121, 2019. 3

[Jonschkowski et al.(2020)] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *European Conference on Computer Vision*, pages 557–572. Springer, 2020. 7

[Kalantari and Ramamoorthi(2019)] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep hdr video from sequences with alternating exposures. In *Computer graphics forum*, volume 38, pages 193–205. Wiley Online Library, 2019. 3

[Kalantari et al.(2017)] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.*, 36(4):144–1, 2017. 3

[Kalantari et al.(2013)] Nima Khademi Kalantari, Eli Shechtman, Connelly Barnes, Soheil Darabi, Dan B Goldman, and Pradeep Sen. Patch-based high dynamic range video. *ACM Trans. Graph.*, 32(6):202–1, 2013. 3

[Lee et al.(2020)] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 3

[Liu et al.(2019)] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8794–8802, 2019. 3

[Nah et al.(2017)] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 6, 7, 8

[Nguyen et al.(2022)] Cindy M Nguyen, Julien NP Martel, and Gordon Wetzstein. Learning spatially varying pixel exposures for motion deblurring. *arXiv preprint arXiv:2204.07267*, 2022. 1, 4

[Niklaus and Liu(2020)] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 3

[Niklaus et al.(2017)] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 3

[Parihar et al.(2021)] Anil Singh Parihar, Disha Varshney, Kshitija Pandya, and Ashray Aggarwal. A comprehensive survey on video frame interpolation techniques. *The Visual Computer*, pages 1–25, 2021. 2

[Park et al.(2020)] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision*, pages 109–125. Springer, 2020. 3

[Park et al.(2021)] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14539–14548, 2021. 1, 3, 8, 9

[Rebecq et al.(2019)] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019. 4

[Reda et al.(2022)] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. *arXiv preprint arXiv:2202.04901*, 2022. 1, 2, 3, 8, 9

[Reda et al.(2019)] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 892–900, 2019. 3

[Rengarajan et al.(2020)] Vijay Rengarajan, Shuo Zhao, Ruiwen Zhen, John Glotzbach, Hamid Sheikh, and Aswin C Sankaranarayanan. Photosequencing of motion blur using short and long exposures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 510–511, 2020. 3, 5

[Shen et al.(2020)] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5114–5123, 2020. 1, 3, 5

[Shen et al.(2020)] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Transactions on Image Processing*, 30:277–292, 2020. 3

[Sim et al.(2021)] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14489–14498, 2021. 1, 3, 6, 7, 8, 9

[Sony(1999)] Sony. Quad bayer sensors, 1999. 1

[Su et al.(2017)] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 6, 7, 8

[Sun et al.(2017)] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *CoRR*, abs/1709.02371, 2017. 7

[Teed and Deng(2020)] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 4, 5, 7, 10

[Xu et al.(2019)] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3, 4, 5, 8, 9

[Yan et al.(2020)] Qingsen Yan, Lei Zhang, Yu Liu, Yu Zhu, Jinqiu Sun, Qinfeng Shi, and Yanning Zhang. Deep hdr imaging via a non-local network. *IEEE Transactions on Image Processing*, 29:4308–4322, 2020. 3

[Zhang et al.(2020)] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. Video frame interpolation without temporal priors. *Advances in Neural Information Processing Systems*, 33:13308–13318, 2020. 1, 3