Unified Video Reconstruction for Rolling Shutter and Global Shutter Cameras

Bin Fan, Zhexiong Wan, Boxin Shi, *Senior Member, IEEE*, Chao Xu, and Yuchao Dai, *Member, IEEE*

***Abstract*—Currently, the general domain of video reconstruction (VR) is fragmented into different shutters spanning global shutter and rolling shutter cameras. Despite rapid progress in the state-of-the-art, existing methods overwhelmingly follow shutter-specific paradigms and cannot conceptually generalize to other shutter types, hindering the uniformity of VR models. In this paper, we propose UniVR, a versatile framework to handle various shutters through unified modeling and shared parameters. Specifically, UniVR encodes diverse shutter types into a unified space via a tractable shutter adapter, which is parameter-free and thus can be seamlessly delivered to current well-established VR architectures for cross-shutter transfer. To demonstrate its effectiveness, we conceptualize UniVR as three shutter-generic VR methods, namely Uni-SoftSplat, Uni-SuperSloMo, and Uni-RIFE. Extensive experimental results demonstrate that the pre-trained model without any fine-tuning can achieve reasonable performance even on novel shutters. After fine-tuning, new state-of-the-art performances are established that go beyond shutter-specific methods and enjoy strong generalization. The code is available at https://github.com/GitCVfb/UniVR.**

***Index Terms*—Unified model, video reconstruction, rolling shutter, global shutter, motion approximation, deep learning.**

I. Introduction

As a fundamental video processing task, the goal of video reconstruction (VR), is to generate the desired in-between frames given a pair of consecutive image frames [1], [2]. VR involves the understanding of pixel motion, image appearance, and even 3D structure, which contributes to many practical applications, such as slow-motion animation [3], [4], video compression [5], [6], novel view synthesis [7], [8], and other real-world systems [9], [10], [11], [12]. In recent years, a plethora of VR techniques has been actively studied around the common global shutter (GS) and rolling shutter (RS) cameras, *e.g.*, GS video frame interpolation [3] and RS temporal super-resolution [13], with increasingly impressive results powered by the rapid progress of deep neural networks.

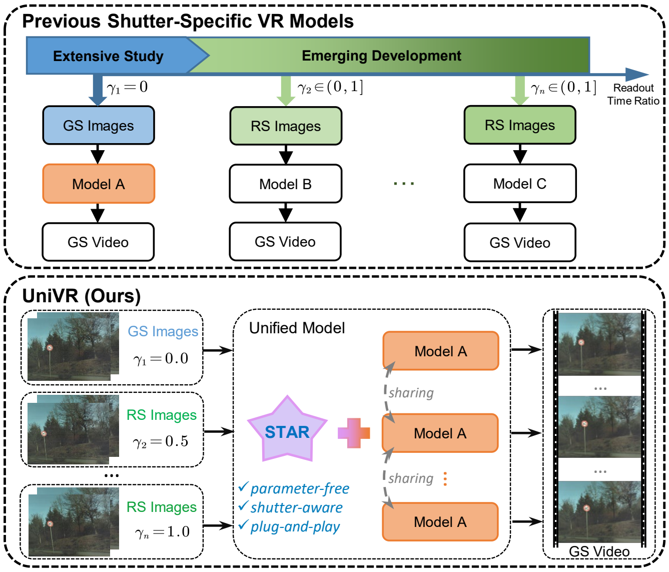


Fig. 1. Comparison between previous shutter-specific VR models and our proposed UniVR. The generality and scalability of the existing approaches are poor, *i.e.*, shutter-specific models struggle to interconvert across different shutter types, including GS images ($\gamma_1 = 0$) and various RS images ($\gamma_{2:n} \in (0,1]$). Note that γ denotes the readout time ratio [14], which can be employed to identify different camera types. In contrast, our unified model can seamlessly cope with consecutive image input of various shutter types through a tractable *shutter adapter (STAR)* and a well-established GS-based VR model with shared parameters.

Recently, the GS-based VR algorithms have received *significant attention* and exhibited remarkable performances. In general, the flow-based scheme is the most prevalent and can be roughly classified into three types:

- Estimating the bidirectional optical flow, then approximating the intermediate motion field for forward warping **(Type 1)** or backward warping **(Type 2)** of the input frame, and finally adding a frame synthesis module to complete the intermediate frame reconstruction. Among them, the most representative baseline efforts are SoftSplat [15] and SuperSloMo [3], which are based on forward and backward warping, respectively.
- Designing sub-networks to directly estimate the intermediate motion field **(Type 3)**, which is used to warp the input frame, and then similarly grafting a frame synthesis module. RIFE [16] is a seminal baseline on this line.

Manuscript received 20 March 2024; revised 22 September 2024; accepted 5 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62271410, 62136001, 62088102, 62276007, and 62401021; in part by China National Postdoctoral Program for Innovative Talents under Grant BX20230013; and in part by China Postdoctoral Science Foundation under Grant 2024M750101. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dong Liu. *(Corresponding author: Yuchao Dai.)*

Bin Fan is with the National Key Laboratory of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China, and also with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: binfan@pku.edu.cn).

Zhexiong Wan and Yuchao Dai are with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: wanzhexiong@mail.nwpu.edu.cn, daiyuchao@nwpu.edu.cn).

Boxin Shi is with the State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing 100871, China (e-mail: shiboxin@pku.edu.cn).

Chao Xu is with the National Key Laboratory of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China (e-mail: xuchao@cis.pku.edu.cn).

With the ubiquity of RS cameras in commercial and industrial applications, it has also gradually attracted researchers' interest in reviving and reliving the latent GS video from adjacent RS frames, driving the *emerging development* of RS-based VR. The row-by-row readout of RS cameras notably yields geometric image distortions (*e.g.*, skew, wobble) [17], [18], [19], [20], [21], [22], [23], [24], [25] when shooting dynamic objects or moving the RS camera. As such, unlike GS-based VR, RS-based VR paradigms must perform RS correction and frame interpolation simultaneously, which poses additional challenges for network design [26], [27]. To eliminate the unintended RS distortion, existing RS-based VR methods either implicitly encapsulate the underlying RS geometry in the overall network model to embed the dedicated RS correction capability [2], [13], [27], or explicitly engineer a two-stage combination of RS correction [28], [29], [30] and GS-based VR [3], [15], [16] methods.

Unfortunately, these crafted VR approaches currently suffer from several obvious drawbacks when popularized to general camera types, as shown in Fig. 1. **First**, the prevailing works are largely driven by designing shutter-specific models to solve each camera type independently, resulting in poor scalability. For example, a well-designed VR method for GS images can only be dedicated to GS cameras, but cannot conceptually generalize to RS-based VR, and vice versa. **Second**, the model trained on a customized RS dataset is only applicable to its corresponding RS camera, leading to difficulties in robust generalization across different types of RS cameras. This is because RS correction, as a favored component of the RS-based VR model, essentially encodes RS-specific underlying geometry. Additionally, two-stage RS-based VR methods struggle to eliminate error accumulation via joint optimization, and thus suffer from the disadvantages of larger and more time-consuming models [26], [31]. All of these issues increase the marginal cost of developing VR models for cameras with novel shutter types, and greatly limit the effective deployment of existing methods in practice, indicating shutter-specific paradigm is not suitable for generic VR modeling.

To deal with these fragmentations, our key insight is to *break shutter-specific designs by seamlessly extending generally-applicable GS-based VR approaches to the RS image domain*. We argue that RS-based VR itself can also be regarded as a pure frame interpolation task like GS-based VR, without compounding an additional RS correction procedure. During its network design, RS correction can be naturally included in the frame warping proposal of the GS-based VR network at hand, which lays the foundation for generic VR modeling. This design philosophy eliminates the gap between the formulations of GS-based and RS-based VR tasks, thereby encouraging information sharing and mutual collaboration of cross-shutter VR paradigms.

In this paper, we propose a unified video reconstruction architecture, termed UniVR, for both RS and GS cameras. It focuses on a flexible, parameter-free ShutTer AdapteR (STAR) to adaptively identify the RS-specific geometry, as illustrated in Fig. 1. Note that our shutter adapter is plug-and-play and does not introduce any shutter-specific learnable parameters, ensuring the scalability and generality of the resulting unified model. Specifically, for **Type 1** and **Type 2**, we present a shutter-adaptive frame warping model and complete the unification of forward and backward motion approximations. This makes it possible to extend the GS-based VR model to work with both RS and GS cameras. For **Type 3**, we unify the scanline times of the input RS and GS images under a shutter-aware imaging formulation. This ensures that intermediate motions specific to varying shutter types can be learned directly by simply changing the temporal interpolation instance outside the network. Subsequently, STAR can be seamlessly embedded into the well-established GS-based VR framework. In this way, the RS-based and GS-based VR models will be abstracted into a shutter-generic model.

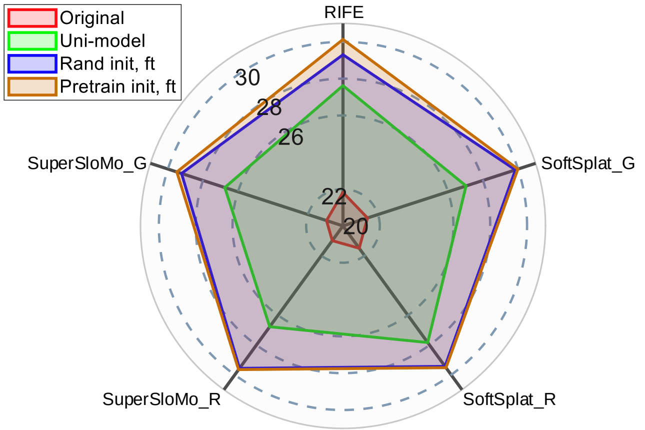


Fig. 2. The PSNR performance of different methods on the Fastec-RS dataset [28]. The original GS-based VR methods (*i.e.*, SoftSplat [15], SuperSloMo [3], and RIFE [16]) have extremely poor generalization (red). Our unified model can hot-swap from GS images to RS images without any shutter-specific retraining (green). Also, fine-tuning based on either random initialization (blue) or pre-trained models (brown) can further improve the performance. "G" and "R" indicate that GMFlow [32] and RAFT [33] are adopted as the optical flow estimation backbone, respectively.

We approach three widely used GS-based VR benchmarks, *i.e.*, the aforementioned SoftSplat [15], SuperSloMo [3], and RIFE [16], and adapt them to the RS principle to form our UniVR model. Extensive experiments on multiple RS datasets show that, by aligning the generic modeling of GS-based and RS-based VR tasks, the pre-trained knowledge on one shutter data can be well transferred to another shutter data without any fine-tuning, *e.g.*, from GS to RS cameras (Fig. 2), and between RS cameras with diverse γ (Sec. IV-E). Such cross-shutter transfer benefits from shared model parameters to efficiently handle various shutter types in the inference process. Moreover, the performance can be further improved after fine-tuning with additional shutter-specific data, outperforming state-of-the-art methods by a large margin, as evidenced by our experiments (*e.g.*, Fig. 2). Meanwhile, our unified model also supports end-to-end training from scratch based on sufficient new shutter data, achieving promising sub-optimal results. The main contributions of this paper are three-fold:

- To the best of our knowledge, we offer the first attempt to develop a unified video reconstruction pipeline, termed UniVR, which supports cross-shutter transfer at runtime without any shutter-specific retraining.

- We propose a tractable shutter adapter for shutter-adaptive frame warping that is seamlessly compatible with GS and RS cameras.
- Experiments demonstrate that our approach not only outperforms the best-specialized architectures but also enjoys good generalization for cross-shutter deployment.

II. Related Work

GS-based video reconstruction. The GS-based VR task, *a.k.a.*, video frame interpolation (VFI), has been a long-standing researched topic in computer vision. With recent advances in optical flow estimation [32], [33], [34], [35], flow-based VFI methods have been actively studied to exploit pixel-level motion information explicitly [36]. In general, VFI can be viewed as the process of approximating the intermediate motion field and then blending the warped image appearance to synthesize the target frame. In addition to carrying out *e.g.*, occlusion reasoning [37], [38], [39], context warping [40], [41], [42] for frame synthesis, researchers have also worked on developing linear [3], [9], [15], [37], [41], [43], [44], [45], quadratic [46], [47], quasi-quadratic [48], cubic [49], and hybrid [50] schemes for motion approximation. Furthermore, designing sub-networks to predict intermediate motions directly is a recent research hotspot, such as [16], [51], [52], [53], [54]. More generally, the linear motion can often be subdivided into backward warping (*e.g.*, [3], [41], [43], [44]) and forward warping (*e.g.*, [1], [9], [15], [45]).

The computer vision community has witnessed the tremendous success of VFI, however, they work with a common assumption that the camera employs a GS mechanism. Due to design defects, their scalability remains unsatisfactory, especially the inability to effectively migrate to the widely used RS camera. Although Naor *et al.* [55] recently tried to sample a GS proposal by taking the relevant scanline from temporally-interpolated RS frames that have the same number as image rows, the connection between GS-based and RS-based VR tasks has not been essentially established, limited by high computational costs and time-specific GS recovery. In contrast, we bridge this gap with a plug-and-play shutter adapter, which can seamlessly extend well-established VFI baselines pre-trained on a large variety of real-world GS videos to shutter-varying RS cameras.

RS-based video reconstruction. In the recent literature, the RS-based VR task, *a.k.a.*, RS temporal super-resolution [13], is in its infancy being developed for extracting high frame rate GS videos hidden in adjacent RS frames. Typically, constant velocity [2], [13] and constant acceleration [26], [27] motion assumptions are employed to model the underlying spatio-temporal coherence, so that the intermediate motion is approximated for forward warping. Alternatively, a cluster of work bypasses the high complexity of time-arbitrary GS recovery, aiming to generate just one time-specific GS image from consecutive RS images, *e.g.*, corresponding to the first [55], [56], [57] or central [28], [29], [30], [58] scanline time.

Despite the RS-based VR becoming emerging, their trained model inherently encapsulates the underlying RS geometry, which is determined by the intrinsic camera parameter, resulting in poor generalizability. Hence, this specific model design becomes a critical bottleneck when popularized to RS cameras with varying intrinsic parameters, let alone GS cameras. In this paper, we propose for the first time to unify GS-based and RS-based VR tasks, yielding a shutter-flexible, time-arbitrary, and performance-excellent video frame reconstruction framework. Importantly, our unified model triggers a new paradigm of designing unified architectures for consecutive image input of various shutter types.

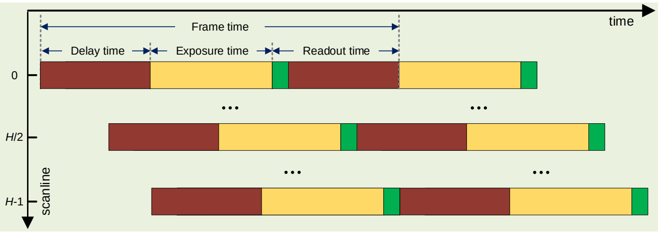


Fig. 3. Illustration of the RS imaging mechanism between two consecutive frames. The readout time ratio $\gamma \in (0, 1]$ [14] of the RS camera can be defined as the ratio between the total readout time and the total frame time. Note that $\gamma = 0$ can be utilized to represent GS cameras, while $\gamma \in (0, 1]$ is capable of identifying various γ-specific RS cameras.

III. Method

In this section, we first introduce the readout time ratio for shutter interpretation in Sec. III-A and define a unified VR task in Sec. III-B. Then, we derive a shutter-aware imaging formulation to develop an intermediate motion estimation model for our shutter adapter in Sec. III-C. Finally, we detail how to seamlessly extend the off-the-shelf well-established GS-based VR model to a variety of RS camera types for shutter-generic VR in Sec. III-D.

A. Shutter Interpretation with Readout Time Ratio

To conceptualize a unified representation of GS and RS cameras, we propose to employ the readout time ratio γ to interpret different camera types. Fig. 3 illustrates the RS imaging mechanism between two consecutive RS frames. The readout time ratio γ [14] is defined as the ratio between the total readout time and the total frame time (including inter-frame delay time). As a hardware-specified camera parameter, γ can be calibrated by [17], [22], and $\gamma \in (0, 1]$ is often employed to model the underlying RS geometry in [13], [14], [26], [57], [59], [60]. Note that, $\gamma \in [0, 1]$ can be leveraged to identify different camera types in the subsequent unified formulation, *i.e.*, $\gamma = 0$ for GS cameras and $\gamma \in (0, 1]$ for various RS cameras, which facilitates the unification of the video reconstruction model for RS and GS cameras.

B. Problem Definition of Unified VR

The unified VR task takes consecutive frames $\mathbf{I}_0$ and $\mathbf{I}_1$ as input and synthesizes an intermediate GS frame $\mathbf{I}_t^{\mathrm{GS}}$ at interpolation time $t \in [-\gamma/2, 1 - \gamma/2]$ as output, as displayed in Fig. 4. It is easy to verify that when $\gamma = 0$, it advocates generating the non-existent intermediate GS frame at interpolation time $t \in [0, 1]$, consistent with the video frame interpolation task [3], [10]. And when $\gamma = 1$, the latent GS frame corresponding to the interpolation time $t \in [-0.5, 0.5]$

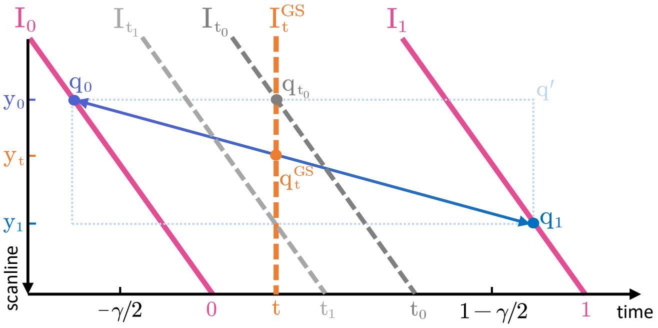


Fig. 4. Illustration of shutter-aware imaging formulation. We define three timelines. **1**) Interpolation time $t \in [-\gamma/2, 1-\gamma/2]$ (horizontal black line), *e.g.*, the subscript of $\mathbf{I}_t^{\mathrm{GS}}$. **2**) Scanline time (ideally perpendicular to the pink line and with the same origin as the interpolation time axis), *i.e.*, the time for each scanline in an image (*e.g.*, $\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_{t_0}$). **3**) Temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$, *i.e.*, the scanline time of target GS images, which is a function of t and scanline y (see Eq. 1) and thus varies with the scanline. Our motivation for unified modeling is to uniformly fix the scanline times of the two consecutive inputs $\mathbf{I}_0$ and $\mathbf{I}_1$ at 0 and 1, respectively. Given the interpolation time $t \in [-\gamma/2, 1-\gamma/2]$, the scanline-varying temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$ can be attached to the target GS image $\mathbf{I}_t^{\mathrm{GS}}$ based on the shutter type (*cf.*, γ).

is recovered, consistent with the RS temporal super-resolution task defined in [2], [31]. As such, we can perform a uniform formulation regardless of whether the inputs $\mathbf{I}_0$ and $\mathbf{I}_1$ belong to GS images or RS images.

C. Shutter Adapter

In addition to the unified representation based on γ, we notice that an important reason why the existing GS-based and RS-based VR methods struggle to unify is that their input time instances are unaligned. Namely, the GS-based VR method characterizes the scanline times of the input images $\mathbf{I}_0$ and $\mathbf{I}_1$ as 0 and 1 respectively, while the RS-based VR method makes different scanlines of $\mathbf{I}_0$ and $\mathbf{I}_1$ have different scanning times. Such a shutter-specific VR formulation, physically tailored on the input camera type, appears conceptually straightforward. However, it limits the unification of GS-based and RS-based VR methods, making general VR modeling intractable.

In order to make the proposed UniVR model compatible with both GS and RS cameras, we first encode the scanline times of $\mathbf{I}_0$ and $\mathbf{I}_1$ uniformly as 0 and 1, respectively, in the input dimension. For example, all scanlines of $\mathbf{I}_0$ share the same scantime 0. At the same time, we allow the desired GS image to have scanline-varying temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$ in the output dimension so that GS-based and RS-based VR tasks can be aligned under a unified representation framework. Furthermore, we point out that UniVR can be characterized as a pure frame interpolation task. During its network design, it only needs to model a reasonable frame warping, which can naturally integrate intermediate frame approximation and RS correction for RS-generic VR. In this way, the gap between the formulations of GS-based and RS-based VR tasks can be bridged for universal VR modeling.

To sum up, we introduce a tractable shutter adapter, which is parameter-free and thus can be seamlessly delivered to current well-established GS-based VR networks (*e.g.*, [3], [15], [16]), as illustrated in Fig. 1. Concretely, in the modeling process of STAR, we first derive a shutter-aware imaging formulation depicted in Fig. 4, and then approximate the temporal interpolation factor $\mathbf{\Phi}_t^{\mathrm{GS}}$ to efficiently estimate the intermediate motion fields (*i.e.*, $\mathbf{F}_{0\to t}, \mathbf{F}_{1\to t}$ or $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ as shown in Fig. 5) for frame warping. The details will be elaborated in the following.

1) Shutter-aware imaging formulation: After uniformly representing the scanline times of the input images $\mathbf{I}_0$ and $\mathbf{I}_1$ as 0 and 1, respectively, as illustrated in Fig. 4, we build a shutter-aware imaging formulation to characterize the latent GS image $\mathbf{I}_t^{\mathrm{GS}}$ corresponding to the interpolation time $t \in [-\gamma/2, 1-\gamma/2]$. In particular, the scanline-varying temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$ and the temporal interpolation factor $\mathbf{\Phi}_t^{\mathrm{GS}}$ can be explicitly computed such that the intermediate motion fields are estimated effectively.

Temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$. In Fig. 4, one can observe that the y_0-th scanline of the target GS image $\mathbf{I}_t^{\mathrm{GS}}$ is identical to the y_0-th scanline of the temporally-interpolated RS frame $\mathbf{I}_{t_0}$. Therefore, an RS proposal corresponding to the interpolation time t_0 and the y_0-th scanline can be sampled to approximate the pixel $\mathbf{q}_{t_0}$ of the y_0-th scanline of $\mathbf{I}_t^{\mathrm{GS}}$. Given an image with a total number of H scanlines, we can easily get $\mathbf{T}_t^{\mathrm{GS}}(\gamma, y_0) \triangleq t_0 = t - \frac{\gamma}{H} y_0 + \gamma$ by the similarity scaling of $H(t_0 - t) = \gamma(H - y_0)$. More generally, we have

$$\mathbf{T}_t^{\mathrm{GS}}(\gamma, y) = t - \frac{\gamma}{H} y + \gamma, \quad y = 0, 1, ..., H-1, \tag{1}$$

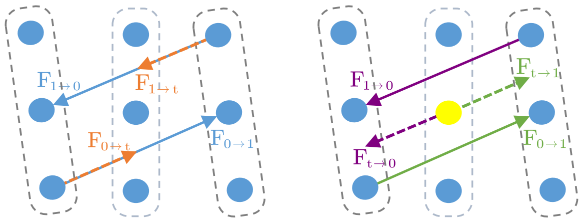
which models the scanline-wise interpolation time of the target GS image $\mathbf{I}_t^{\mathrm{GS}}$. Stacking all image scanlines in matrix form, the scanline-varying temporal interpolation instance $\mathbf{T}_t^{\mathrm{GS}}$ of the target image $\mathbf{I}_t^{\mathrm{GS}}$ can be obtained.

Temporal interpolation factor $\mathbf{\Phi}_t^{\mathrm{GS}}$. As shown in Fig. 4, it is assumed that the pixel $\mathbf{q}_0$ in $\mathbf{I}_0$ matches the pixel $\mathbf{q}_1$ in $\mathbf{I}_1$, with $\mathbf{q}_0$ and $\mathbf{q}_1$ located at the y_0-th and y_1-th scanlines, respectively. Assume also that this pixel passes through the pixel $\mathbf{q}_t^{\mathrm{GS}}$ in $\mathbf{I}_t^{\mathrm{GS}}$ at interpolation time $t \in [-\gamma/2, 1-\gamma/2]$. As a result, $\overrightarrow{\mathbf{q}_0\mathbf{q}_1}$ and $\overrightarrow{\mathbf{q}_1\mathbf{q}_0}$ can denote the forward and backward optical flows $\mathbf{F}_{0\to 1}(\mathbf{q}_0)$ and $\mathbf{F}_{1\to 0}(\mathbf{q}_1)$, respectively. According to the RS geometry modeling in [13], [14] based on the constant velocity motion assumption, we can easily obtain $||\overrightarrow{\mathbf{q}_0\mathbf{q}'}|| = 1 + \frac{\gamma}{H}(y_1 - y_0)$. Also, Eq. 1 is able to produce $||\overrightarrow{\mathbf{q}_0\mathbf{q}_{t_0}}|| \triangleq t_0 = \mathbf{T}_t^{\mathrm{GS}}(\gamma, y_0)$. Further, to approximate the intermediate motion field by scaling the optical flow vector, we define the corresponding temporal interpolation factor as $\mathbf{\Phi}_t^{\mathrm{GS}}(\gamma, y_0, y_1) \triangleq ||\overrightarrow{\mathbf{q}_0\mathbf{q}_t^{\mathrm{GS}}}|| / ||\overrightarrow{\mathbf{q}_0\mathbf{q}_1}||$.

Using the principle of similar triangles, the constraint satisfies $||\overrightarrow{\mathbf{q}_0\mathbf{q}_t^{\mathrm{GS}}}|| \cdot ||\overrightarrow{\mathbf{q}_0\mathbf{q}'}|| = ||\overrightarrow{\mathbf{q}_0\mathbf{q}_{t_0}}|| \cdot ||\overrightarrow{\mathbf{q}_0\mathbf{q}_1}||$. Substituting the above notations yields

$$\mathbf{\Phi}_t^{\mathrm{GS}}(\gamma, y_0, y_1) = \frac{t - \frac{\gamma}{H} y_0 + \gamma}{1 + \frac{\gamma}{H}(y_1 - y_0)}. \tag{2}$$

Note that for the forward optical flow $\mathbf{F}_{0\to 1}(\mathbf{q}_0)$ associated with the pixel $\mathbf{q}_0$ located at the y_0-th scanline of $\mathbf{I}_0$, its vertical component $\mathbf{F}^v_{0\to 1}(\mathbf{q}_0)$ follows: $y_1 - y_0 = \mathbf{F}^v_{0\to 1}(\mathbf{q}_0)$. Similarly, for each pixel $\mathbf{q}_1$ that lies on the y_1-th scanline of $\mathbf{I}_1$, we have $y_1 - y_0 = -\mathbf{F}^v_{1\to 0}(\mathbf{q}_1)$. Consequently, the forward and backward temporal interpolation factors can be obtained by Eq. 2 as $\mathbf{\Phi}_t^{\mathrm{GS;f}}(\mathbf{q}_0) \triangleq \mathbf{\Phi}_t^{\mathrm{GS}}(\gamma, y_0, y_1)$, $\mathbf{\Phi}_t^{\mathrm{GS;b}}(\mathbf{q}_1) \triangleq \mathbf{\Phi}_t^{\mathrm{GS}}(\gamma, y_1, y_0)$. When bidirectional optical flows $\mathbf{F}_{0\to 1}$ and $\mathbf{F}_{1\to 0}$ are known,



(a) Forward motion (b) Backward motion

Fig. 5. Illustration of intermediate motion approximation. Based on the shutter-aware imaging formulation, we can approximate the forward motion field in (a) by scaling the corresponding optical flow vector, and the backward motion field of the yellow pixel in (b) by borrowing the optical flow vector from pixels at the same location in the first and second input images.

stacking all pixels in matrix form can yield the temporal interpolation factor $\mathbf{\Phi}_t^{\text{GS}} = \{\mathbf{\Phi}_t^{\text{GS;f}}, \mathbf{\Phi}_t^{\text{GS;b}}\}$. Thus, Eq. 2 achieves the unification of forward and backward interpolations and facilitates the intermediate motion approximation.

2) Intermediate motion approximation: As shown in Fig. 5, with the pixel-wise temporal interpolation factor $\mathbf{\Phi}_t^{\text{GS}}$, the forward motion fields $\mathbf{F}_{0\to t}, \mathbf{F}_{1\to t}$ and the backward motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ can be approximated by scaling the bidirectional optical flow fields $\mathbf{F}_{0\to 1}, \mathbf{F}_{1\to 0}$ between the two input frames.

Forward motion fields $\mathbf{F}_{0\to t}, \mathbf{F}_{1\to t}$. Given the temporal interpolation factors $\mathbf{\Phi}_t^{\text{GS;f}}, \mathbf{\Phi}_t^{\text{GS;b}}$ derived from Eq. 2, we can obtain the forward motion fields shown in Fig. 5 (a) by a simply scaling operation as follows:

$$\begin{aligned} \mathbf{F}_{0\to t} &= \mathbf{\Phi}_t^{\text{GS;f}} \odot \mathbf{F}_{0\to 1}, \\ \mathbf{F}_{1\to t} &= (\mathbf{1} - \mathbf{\Phi}_t^{\text{GS;b}}) \odot \mathbf{F}_{1\to 0}, \end{aligned} \tag{3}$$

where $\odot$ is an element-wise multiplier. Note that the analytical proof of Eq. 3 based on the assumption of constant velocity motion can be found in [13], [27]. Using the forward motion fields, one can forward warp the input images to the destination of a virtual GS image by

$$\hat{\mathbf{I}}_{i\to t}^{\text{GS}} = \mathcal{W}_{\text{F}}\left(\mathbf{I}_i, \mathbf{F}_{i\to t}\right), \quad i = 0, 1, \tag{4}$$

where $\mathcal{W}_{\text{F}}$ indicates the forward warping operator. Notably, Softmax splatting [15] is often used to alleviate hole or occlusion artifacts in forward warping, caused by collisions when multiple pixels are mapped to the same location, by adaptively combining overlapping pixel information.

Backward motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$. Inspired by the assumption in [3], [43], [44], [61] that neighboring pixels have similar motion vectors, we can simply borrow optical flow vectors from $\mathbf{F}_{0\to 1}$ and $\mathbf{F}_{1\to 0}$ to approximate $\mathbf{F}_{t\to 0}$ and $\mathbf{F}_{t\to 1}$ shown in Fig. 5 (b) as follows:

$$\begin{aligned} \mathbf{F}_{t\to 0} &= \underbrace{-\mathbf{\Phi}_t^{\text{GS;f}} \odot \mathbf{F}_{0\to 1}}_{\mathbf{F}_{t\to 0}^{\text{f}}} \;\; \textit{or} \;\; \underbrace{\mathbf{\Phi}_t^{\text{GS;b}} \odot \mathbf{F}_{1\to 0}}_{\mathbf{F}_{t\to 0}^{\text{b}}}, \\ \mathbf{F}_{t\to 1} &= \underbrace{(\mathbf{1} - \mathbf{\Phi}_t^{\text{GS;f}}) \odot \mathbf{F}_{0\to 1}}_{\mathbf{F}_{t\to 1}^{\text{f}}} \;\; \textit{or} \;\; \underbrace{-(\mathbf{1} - \mathbf{\Phi}_t^{\text{GS;b}}) \odot \mathbf{F}_{1\to 0}}_{\mathbf{F}_{t\to 1}^{\text{b}}}, \end{aligned} \tag{5}$$

where superscripts "f" and "b" denote forward and backward candidates for $\mathbf{F}_{t\to 0}$ and $\mathbf{F}_{t\to 1}$, respectively.

Further, similar to [3], [10], [46], [50], we also take advantage of the temporal distance $\mathbf{D}_t^{\text{GS}}$ for intermediate motion approximations, such that the temporally-closer pixels can be assigned higher motion confidence. Specifically, for brevity, we harness the temporal interpolation instance $\mathbf{T}_t^{\text{GS}}$ and restrict its value between 0 and 1 to ensure temporally meaningful weighting, *i.e.*,

$$\mathbf{D}_t^{\text{GS}} = \min(\max(\mathbf{T}_t^{\text{GS}}, \mathbf{0}), \mathbf{1}). \tag{6}$$

Therefore, the backward motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ can be approximated holistically by combining the candidates in Eq. 5 based on Eq. 6, given by

$$\begin{aligned} \mathbf{F}_{t\to 0} &= (\mathbf{1} - \mathbf{D}_t^{\text{GS}}) \odot \mathbf{F}_{t\to 0}^{\text{f}} + \mathbf{D}_t^{\text{GS}} \odot \mathbf{F}_{t\to 0}^{\text{b}}, \\ \mathbf{F}_{t\to 1} &= (\mathbf{1} - \mathbf{D}_t^{\text{GS}}) \odot \mathbf{F}_{t\to 1}^{\text{f}} + \mathbf{D}_t^{\text{GS}} \odot \mathbf{F}_{t\to 1}^{\text{b}}, \end{aligned} \tag{7}$$

which can be adopted to backward warp the input images to a virtual GS canvas corresponding to interpolation time t as

$$\hat{\mathbf{I}}_{t\leftarrow i}^{\text{GS}} = \mathcal{W}_{\text{B}}\left(\mathbf{F}_{t\to i}, \mathbf{I}_i\right), \quad i = 0, 1, \tag{8}$$

where $\mathcal{W}_{\text{B}}$ indicates the backward warping operator.

In summary, the forward and backward motion fields are modeled in Eqs. 3 and 7, respectively, which can be used for forward and backward warping as shown in Eqs. 4 and 8. Note that our formulation is parameter-free and shutter-adaptive, with γ to determine the camera type, thus ensuring shutter-aware frame warping. And it is easy to verify that when using $\gamma = 0$ to model the GS camera, Eqs. 3 and 7 will degenerate into the warping models commonly used in classical GS-based VR methods (*e.g.*, [9], [15] and [3], [10]).

D. Unified Architecture for Shutter-generic VR

To address the problem that VR methods dealing with different camera types are often incoherent and crafted for camera shutters, *we seamlessly integrate the above simple yet flexible STAR into the motion estimation module of the well-established GS-based VR framework, forming our unified model.* The overall architecture of our UniVR model is shown in Fig. 6. Thanks to the fact that the model parameters are shared across shutters, our method allows hot-swapping from the generally-applicable GS-based VR model, which is pre-trained on a large variety of real-world GS videos, to shutter-varying RS-based VR. Specifically, based on three most representative baselines, *i.e.*, SoftSplat [15], SuperSloMo [3], and RIFE [16], we implement three tractable UniVR models as follows:

1) **Uni-SoftSplat:** We replace the forward motion fields $\mathbf{F}_{0\to t}, \mathbf{F}_{1\to t}$ in the original SoftSplat with Eq. 3 for forward warping without changing the other parts (see *Type 1* in Fig. 6).
2) **Uni-SuperSloMo:** We replace the backward motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ in the original SuperSloMo with Eq. 7 for backward warping without altering the rest (see *Type 2* in Fig. 6).
3) **Uni-RIFE:** Since the original RIFE can predict the time-arbitrary backward motion fields, we implant the scanline-varying temporal interpolation instance $\mathbf{T}_t^{\text{GS}}$ defined in Eq. 1 to directly generate $\mathbf{F}_{t\to 0}$ and $\mathbf{F}_{t\to 1}$ corresponding to the interpolation time t while leaving everything else unchanged (see *Type 3* in Fig. 6).

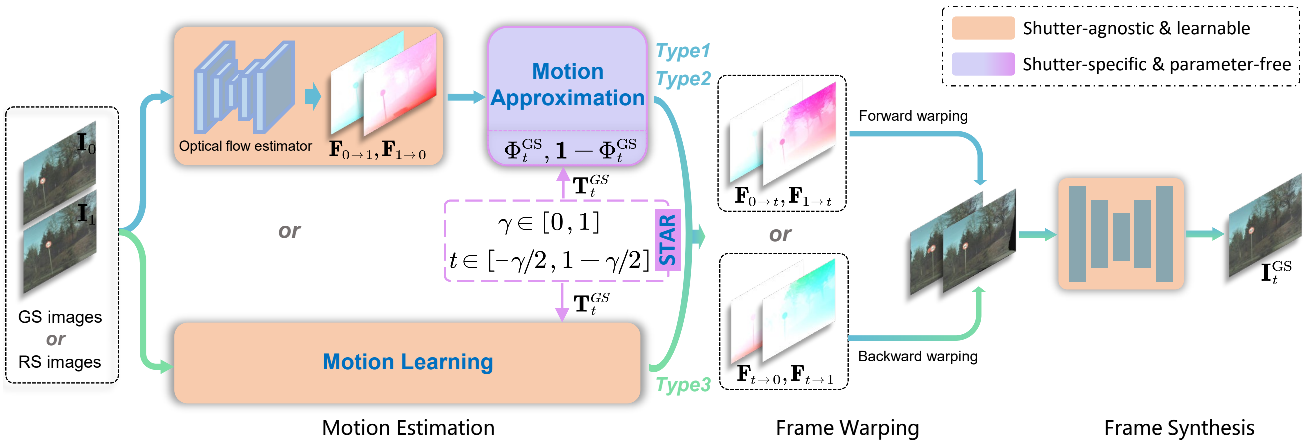


Fig. 6. Overall architecture of our UniVR. We derive our unified model by embedding the parameter-free shutter adapter (STAR) into the well-established GS-based VR framework (*e.g.*, SoftSplat [15], SuperSloMo [3], RIFE [16]). In our STAR modeling, the temporal interpolation factor $\mathbf{\Phi}_t^{\text{GS}}$ and the temporal interpolation instance $\mathbf{T}_t^{\text{GS}}$ can be explicitly calculated. According to motion estimation and frame warping, our UniVR can be divided into three types: ***Type 1*** **(Uni-SoftSplat)**: using $\mathbf{\Phi}_t^{\text{GS}}$ to estimate the intermediate motion fields $\mathbf{F}_{0\to t}, \mathbf{F}_{1\to t}$ for forward warping, without altering the original SoftSplat network. ***Type 2*** **(Uni-SuperSloMo)**: using $\mathbf{\Phi}_t^{\text{GS}}$ to estimate intermediate motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ for backward warping, without altering the original SuperSloMo network. ***Type 3*** **(Uni-RIFE)**: directly learning shutter-adaptive intermediate motion fields $\mathbf{F}_{t\to 0}, \mathbf{F}_{t\to 1}$ for backward warping, by simply varying $\mathbf{T}_t^{\text{GS}}$ outside the network of the original RIFE.

It is also easy to verify that when $\gamma = 0$, our UniVR model will degenerate into the original GS-based VR model. Moreover, unlike current shutter-specific VR designs, our UniVR establishes a shutter-generic frame interpolation framework, such that time-arbitrary GS video frames can be faithfully recovered by accessing γ, regardless of the input image type.

IV. Experiments

A. Experimental Setup

Datasets. We devote to adapting the well-established GS-based VR method to the RS domain. Since their effectiveness on large-scale GS datasets (*e.g.*, Vimeo90K [37], Adobe240-fps [62]) has been fully verified, we will not employ the GS dataset. On the contrary, we exploit the standard RS benchmark dataset, *i.e.*, Carla-RS and Fastec-RS [28], similar to [2], [13], [26], [27]. The Carla-RS dataset is generated from a virtual 3D environment using the Carla simulator, involving general six-degree-of-freedom camera motions. The Fastec-RS dataset contains real-scene RS images synthesized by row-by-row stitching of high-speed GS videos collected by a ground vehicle. They provide GS ground-truth (GT) corresponding to interpolation times $-\gamma/2$, $1-\gamma$, and $1-\gamma/2$, which can be used as supervisory signals to fine-tune our UniVR model. At the inference phase, our method can recover latent GS video frames corresponding to arbitrary interpolation time $t \in [-\gamma/2, 1-\gamma/2]$. Moreover, we utilize the real-world BS-RSC dataset [30], in which various camera and object motions (*e.g.*, vehicles and pedestrians) are recorded by a beam-splitter system in the dynamic urban environment. Since it contains only the GS GT at the specific interpolation time of $1-\gamma/2$, we use it for RS correction evaluation in Sec. IV-H.

Implementation details. Given the time $t \in [-\gamma/2, 1-\gamma/2]$, the original SoftSplat[1], SuperSloMo[2], and RIFE[3] can directly synthesise the corresponding in-between images. We integrate the proposed STAR to develop Uni-SoftSplat, Uni-SuperSloMo, and Uni-RIFE, which can hot-swap pre-trained models yielded on disjoint GS datasets to recover latent GS videos for RS cameras. Note that SoftSplat and RIFE are pre-trained on the Vimeo90K dataset, and SuperSloMo uses the Adobe240-fps dataset. GMFlow [32] is adopted as the optical flow estimator of Uni-SoftSplat and Uni-SuperSloMo, which is frozen during fine-tuning. The readout time ratio γ of Carla-RS, Fastec-RS, and BS-RSC datasets is specified to 1.0, 1.0, and 0.45, respectively. Besides, our Uni-SoftSplat, Uni-SuperSloMo, and Uni-RIFE are fine-tuned for 300 epochs based on random initialization or pre-trained models, with batch sizes of 2, 4, 6, respectively. The learning rate is fixed at 0.0001 in the Adam optimizer [63]. Meanwhile, data augmentation similar to [2] is utilized. All experiments are conducted using a single NVIDIA RTX 3090 GPU.

Evaluation metrics. We apply PSNR, SSIM, and LPIPS [64] metrics to compute the quantitative result. Methods with high PSNR/SSIM and low LPIPS scores are favored. Note that unless otherwise stated, *we refer to the GS images corresponding to* $1-\gamma$ *for consistent comparison.* Since the GT occlusion mask is available in the Carla-RS dataset, following [2], [27], [28], we report Carla-RS with occlusion mask (*CRM*), Carla-RS without occlusion mask (*CR*), and Fastec-RS (*FR*).

Comparison methods. We perform comparisons with four classes of methods: (i) Non-learning-based **DiffSfM** [14], **DiffHomo** [59], and learning-based **DeepUnrollNet** [28], VideoRS [55], **SUNet** [56], **JCD** [58], **AdaRSC** [30], **JAM-Net** [29], **DFRSC_2F** [65] are classical RS correction methods, which can only recover a time-specific GS image. (ii) **RSSR** [13], **CVR** [2] are RS-based VR methods, which can produce time-arbitrary GS images from two adjacent RS images. We do not compare with [27] because it requires five RS frames as input. (iii) **SoftSplat** [15], **SuperSloMo** [3], **RIFE** [16] are generally-applicable GS-based VR methods, which are tailored specifically for GS cameras. (iv) **Two-**

[1] https://github.com/JHLew/SoftSplat-Full
[2] https://github.com/avinashpaliwal/Super-SloMo
[3] https://github.com/megvii-research/ECCV2022-RIFE

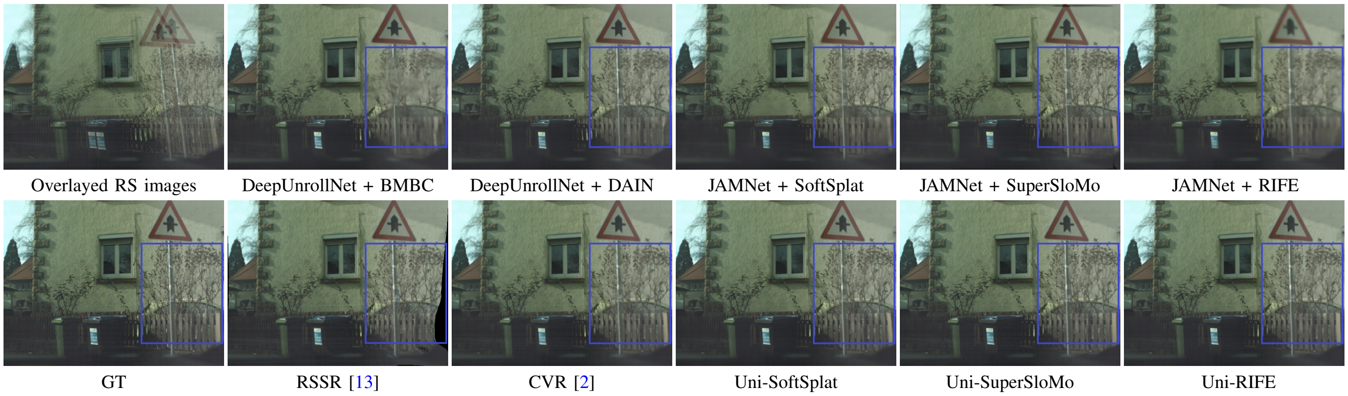


Fig. 7. Visual comparisons against existing RS-based VR methods on the Fastec-RS dataset [28]. Our method can successfully remove the RS artifact to generate crisp and pleasing GS images. It is worth mentioning that high-quality GS frames at any interpolation time can be recovered by our method.

TABLE I

Quantitative comparisons of recovering GS images at interpolation time $1-\gamma$ on Carla-RS and Fastec-RS datasets [28]. Gray backgrounds are used to mark the methods that can only produce one GS image. The average runtime for 640×480 images on a single 3090 GPU and the number of model parameters are reported. Based on the pre-trained model on the large-scale GS dataset, our method shows reasonable performance on the RS dataset without any fine-tuning. After fine-tuning, our method achieves the most satisfactory results in terms of speed and accuracy. Note that our Uni-RIFE embraces real-time performance.

Method	Runtime	#Params	PSNR↑ (dB)			SSIM↑		LPIPS↓	
	(ms)	(Million)	CRM	CR	FR	CR	FR	CR	FR
DiffHomo [59]	$4.2e^5$	-	19.60	18.94	18.68	0.606	0.609	0.1798	0.2229
DiffSfM [14]	$4.7e^5$	-	24.20	21.28	20.14	0.775	0.701	0.1322	0.1789
VideoRS [55]	$1.3e^6$	24.26	31.84	31.43	28.57	0.919	0.844	-	-
DeepUnrollNet [28]	131	3.91	26.90	26.46	26.52	0.807	0.792	0.0703	0.1222
SUNet [56]	92	12.0	29.28	29.18	28.34	0.850	0.837	0.0658	0.1205
DeepUnrollNet [28] + BMBC [43]	964	14.9	27.29	27.58	24.95	0.829	0.787	0.0980	0.2024
DeepUnrollNet [28] + DAIN [41]	297	27.9	27.48	27.88	26.19	0.874	0.807	0.0821	0.1453
JAMNet [29] + SoftSplat [15]	115	12.1	30.40	30.14	26.63	0.895	0.815	0.0629	0.1982
JAMNet [29] + SuperSloMo [3]	93	29.2	28.89	28.04	25.17	0.869	0.790	0.0924	0.1634
JAMNet [29] + RIFE [16]	66	15.4	29.96	29.74	26.81	0.877	0.813	0.1241	0.2315
RSSR [13]	58	26.0	30.17	24.78	21.23	0.867	0.776	0.0695	0.1659
CVR [2]	70	42.7	32.02	31.74	28.72	0.929	0.847	0.0368	0.1107
SoftSplat [15]	67	**7.44**	20.84	20.71	21.40	0.638	0.683	0.0859	0.2086
Uni-SoftSplat (w/o ft)	67	**7.44**	30.78	30.73	27.03	0.912	0.826	0.0372	0.1716
Uni-SoftSplat (ft, rand init)	67	**7.44**	32.92	32.80	29.85	0.938	0.869	0.0197	0.0814
Uni-SoftSplat (ft, pretrain init)	67	**7.44**	**33.12**	**33.02**	**29.99**	**0.942**	**0.872**	**0.0182**	**0.0767**
SuperSloMo [3]	44	24.5	20.90	20.69	20.94	0.643	0.657	0.1142	0.1642
Uni-SuperSloMo (w/o ft)	44	24.5	30.97	30.43	26.75	0.910	0.822	0.0463	0.0967
Uni-SuperSloMo (ft, rand init)	44	24.5	33.15	33.07	29.22	0.941	0.856	0.0249	0.0759
Uni-SuperSloMo (ft, pretrain init)	44	24.5	**33.20**	**33.11**	**29.49**	**0.942**	**0.859**	**0.0226**	**0.0710**
RIFE [16]	**19**	10.7	21.25	21.13	21.82	0.666	0.702	0.1408	0.2331
Uni-RIFE (w/o ft)	**19**	10.7	30.47	30.19	27.62	0.891	0.829	0.0851	0.1768
Uni-RIFE (ft, rand init)	**19**	10.7	31.87	31.75	29.30	0.918	0.866	0.0267	0.0719
Uni-RIFE (ft, pretrain init)	**19**	10.7	**32.97**	**32.82**	**30.13**	**0.937**	**0.875**	**0.0204**	**0.0711**

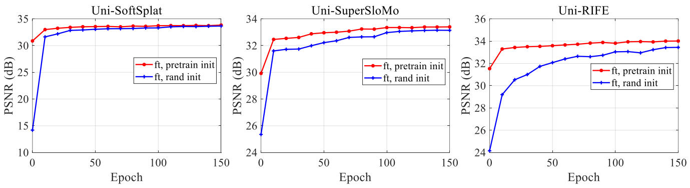


Fig. 8. Fine-tuning performance on the validation set of the Fastec-RS dataset. Using the pre-trained model yields better performance in both training speed and reconstruction accuracy than a random initialized model, demonstrating the effectiveness of cross-shutter transfer.

stage methods contain five cascades, namely "DeepUnrollNet + BMBC [43]", "DeepUnrollNet + DAIN [41]", "JAMNet + SoftSplat", "JAMNet + SuperSloMo", and "JAMNet + RIFE". In our implementation, we first utilize DeepUnrollNet or JAMNet to sequentially recover two GS images corresponding to the central scanline (*i.e.*, interpolation time $1-\gamma/2$) from three consecutive RS images, and then use various GS-based VR methods to reconstruct one GS image corresponding to the first scanline (*i.e.*, interpolation time $1-\gamma$).

B. Comparison and Analysis

The quantitative results are reported in Tables I and II. It can be seen that our approach consistently achieves excellent RS correction performance, outperforming state-of-the-art methods by a large margin. Using GS-based VR methods directly on RS cameras leads to disappointing results due to network flaws, alluding to the extremely poor scalability and generality of shutter-specific methods. After introducing our shutter adapter to export the shutter-generic model, the pre-trained knowledge on the large-scale GS dataset can be seam-

TABLE II

QUANTITATIVE COMPARISONS OF RECOVERING GS IMAGES AT INTERPOLATION TIME $1-\gamma/2$ ON CARLA-RS AND FASTEC-RS DATASETS [28]. GRAY BACKGROUNDS ARE USED TO MARK THE METHODS THAT CAN ONLY PRODUCE ONE GS IMAGE. IN ADDITION TO THE EXCELLENT RS CORRECTION PERFORMANCE OF OUR METHOD AT TIME $1-\gamma$ IN TABLE I, OUR METHOD ALSO CONSISTENTLY ACHIEVES THE HIGHEST GS RECONSTRUCTION ACCURACY AT TIME $1-\gamma/2$, WHICH REQUIRES ONLY A SIMPLE MODIFICATION OF EXISTING GS-BASED VR METHODS, INDICATING THAT OUR PROPOSED UNIFIED VR MODEL CAN SERVE AS A FRUITFUL BASIS FOR CONNECTING GS AND RS CAMERAS.

Method	PSNR↑ (dB)			SSIM↑		LPIPS↓	
	CRM	CR	FR	CR	FR	CR	FR
DiffSfM [14]	25.93	22.88	21.44	0.770	0.710	0.1201	0.2180
AdaRSC [30]	-	-	28.56	-	0.855	-	0.0796
JCD [58]	28.12	27.75	26.48	0.836	0.821	0.0595	0.0943
DeepUnrollNet [28]	27.86	27.54	26.73	0.829	0.819	0.0555	0.0995
SUNet [56]	28.44	28.17	27.06	0.838	0.825	0.0702	0.1030
JAMNet [29]	31.00	30.70	28.70	0.905	0.865	0.0371	0.0691
DFRSC_2F [65]	-	31.33	28.88	0.921	0.870	0.0228	0.0699
SUNet [56] + BMBC [43]	28.51	28.69	25.49	0.848	0.796	0.1033	0.2118
SUNet [56] + DAIN [41]	28.63	28.93	27.12	0.851	0.823	0.0919	0.1642
RSSR [13]	29.36	26.57	24.89	0.900	0.824	0.0553	0.1109
CVR [2]	29.41	29.19	26.67	0.915	0.838	0.0403	0.1011
Uni-SoftSplat (ft, pretrain init)	**32.10**	**31.93**	**28.96**	**0.933**	**0.866**	**0.0206**	**0.0760**
Uni-SuperSloMo (ft, pretrain init)	**32.10**	**31.93**	**28.41**	**0.935**	**0.853**	**0.0244**	**0.0683**
Uni-RIFE (ft, pretrain init)	**32.01**	**31.80**	**28.72**	**0.928**	**0.863**	**0.0229**	**0.0718**

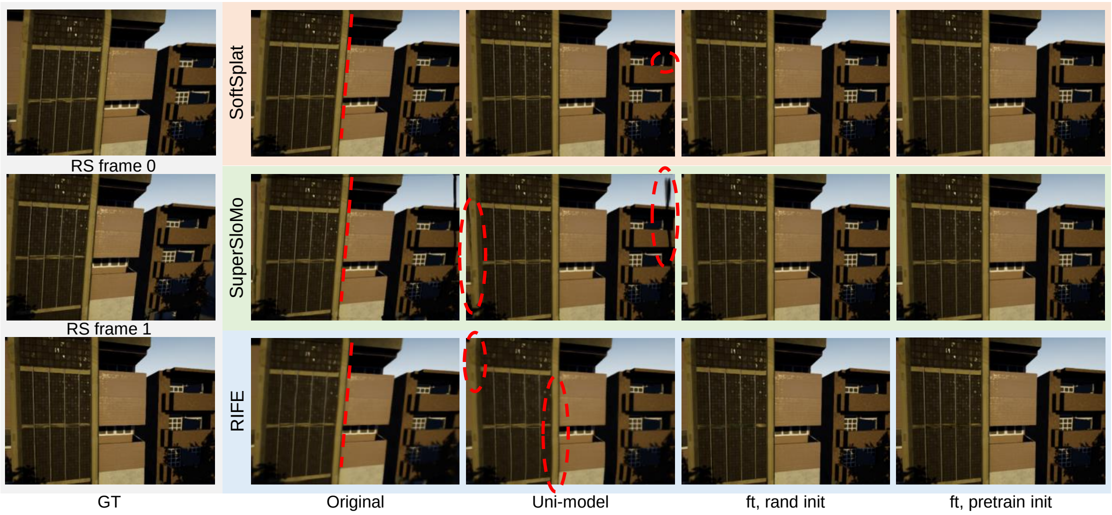


Fig. 9. Visual results of different models. Directly applying the original GS-based method to RS images cannot remove RS artifacts (see red lines), while our Uni-model can tackle this issue by hot-swapping in a shutter-adaptive manner. However, a small amount of artifacts may exist, as shown by red circles. Higher-quality GS images can be further recovered after fine-tuning. A dynamic presentation of these results can be seen in the *supplementary video*.

lessly hot-swapped to RS-based VR, with 47.7%, 48.2%, and 43.4% enhancements on the Carla-RS dataset, and with 26.3%, 27.7%, and 26.6% enhancements on the Fastec-RS dataset, respectively. Note that such direct cross-shutter transfer even surpasses the corresponding two-stage method.

As visualized in Fig. 7, high-fidelity GS video frames are restored with rich details and fewer artifacts. Remarkably, this also demonstrates the superior generalization ability from the GS camera ($\gamma = 0$) to the RS camera ($\gamma = 1$). Furthermore, the performance can be further boosted after fine-tuning with additional training data. Compared to the randomly initialized model, when fine-tuning with a pre-trained model as initialization, we not only achieve better performance (also see Fig. 2) but also faster training speed (see Fig. 8). In particular, we can observe that the two-stage method is prone to suffer from local errors and blurring artifacts due to error accumulation, and is also computationally inefficient. Overall, these experiments validate the superiority of our unified architecture for shutter-generic video reconstruction.

C. *Visualization of UniVR Models*

In addition to the quantitative analysis in Sec. IV-B, we further provide visual results of different UniVR models, *i.e.*, original model, hot-swapped Uni-model, fine-tuned Uni-model based on random initialization, and fine-tuned Uni-model based on pre-trained parameters, as displayed in Fig. 9. It can be seen that applying the original GS-based VR model directly to RS images cannot successfully remove RS artifacts, *e.g.*, the buildings are still curved. By contrast, our proposed Uni-model can hot-swap models pre-trained on large-scale GS datasets to various RS cameras, effectively removing most of the RS artifacts and interpolating intermediate frames. By fine-tuning on the new shutter data, higher-quality GS images can be reconstructed, which further demonstrates the effectiveness and advancement of our unified VR model. In particular, using a pre-trained model as initialization for fine-tuning can achieve better results than randomly initializing the model.

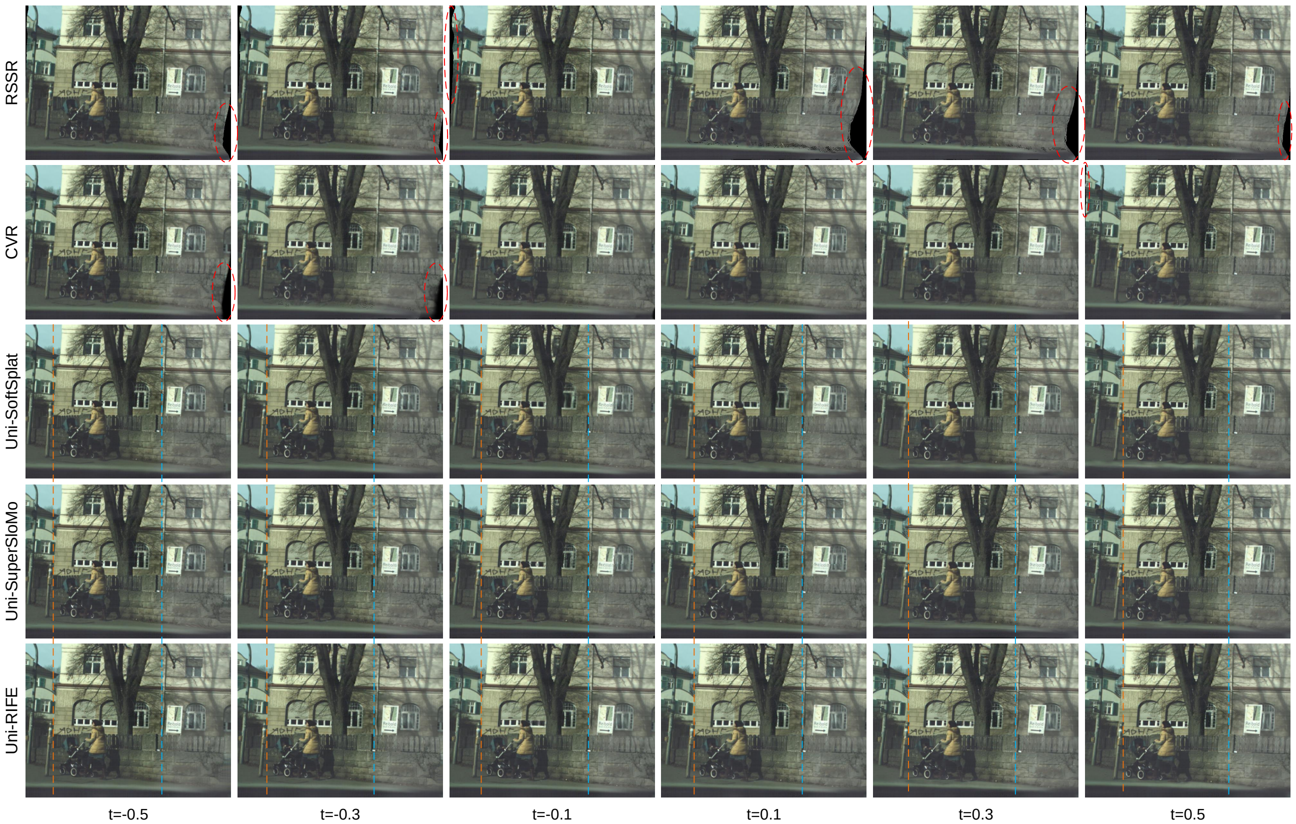


Fig. 10. Example results of 5× temporal upsampling on the Fastec-RS dataset [28]. These GS video images correspond to interpolation times $\{-0.5, -0.3, -0.1, 0.1, 0.3, 0.5\}$, respectively. Our method is able to produce temporally consistent and visually pleasant GS video sequences with arbitrary frame rates. Also, more image details can be restored by our method. Best viewed on screen.

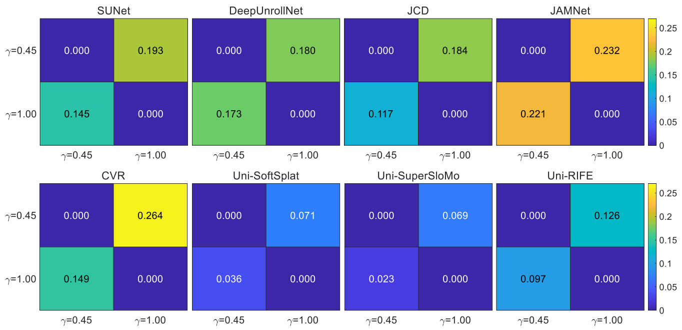


Fig. 11. Cross-shutter transfer evaluation on SSIM against state-of-the-art methods. Our model exhibits the most superior generalization ability, *e.g.*, between the Carla-RS dataset ($\gamma = 1$) and the BS-RSC dataset ($\gamma = 0.45$).

D. Video Reconstruction Results

We generate multiple in-between GS images at arbitrary interpolation time $t \in [-\gamma/2, 1-\gamma/2]$. Fig. 10 illustrates the visual result of 5× temporal upsampling, where smooth and continuous GS video sequences are reconstructed successfully by our approach. In principle, our approach supports arbitrary-framerate GS video reconstruction. Moreover, we attach a *supplementary video* to dynamically show the video reconstruction result, where $10\times$ temporally-upsampled slow-motion GS videos are effectively restored by our approach from two consecutive RS frames. In summary, our unified model can cope with general-purpose VR tasks, regardless of the input shutter type of the camera.

E. Cross-shutter Transfer between Varying RS Cameras

A unique benefit of our unified model is that it naturally enables cross-shutter transfer since all learnable parameters are identical. Sec. IV-B demonstrates the generalization from GS camera to RS camera, and here we verify the transfer capability between RS cameras with varying γ. We leverage the Carla-RS dataset ($\gamma = 1$) and the BS-RSC dataset ($\gamma = 0.45$) for cross-testing. Inspired by [27], we adopt the relative decay rate $1 - \text{SSIM}_{\gamma_j}^{\gamma_i}/\text{SSIM}_{\gamma_j}^{\gamma_j}$, where training is performed at the superscript (γ_i-dataset) and testing at the subscript (γ_j-dataset). The heatmap in Fig. 11 shows that state-of-the-art methods struggle to robustly perform cross-shutter transfers due to significant changes in the intrinsically encapsulated γ-specific geometry. Thanks to our unified model, strong generalization ability is achieved by our approach with an average relative decay rate of less than 0.1.

We further leverage real RS data, provided by [14] and [66] with γ of 0.96 and 0.92, respectively, in which handheld cameras move rapidly and irregularly in the outdoor scene to capture consecutive RS image frames. We qualitatively demonstrate the effectiveness of an RS-based VR model pre-trained on the Carla-RS dataset ($\gamma = 1.0$) when hot-swapped to these new cameras, *i.e.*, the cross-shutter transfer from γ of 1.0 to 0.96 and 0.92, as illustrated in Fig. 12. It can be seen that our method exhibits excellent cross-shutter transfer capabilities, which can be generalized to various types of RS cameras to remove diverse RS artifacts in realistic scenes, thereby achieving high-quality GS video reconstruction.

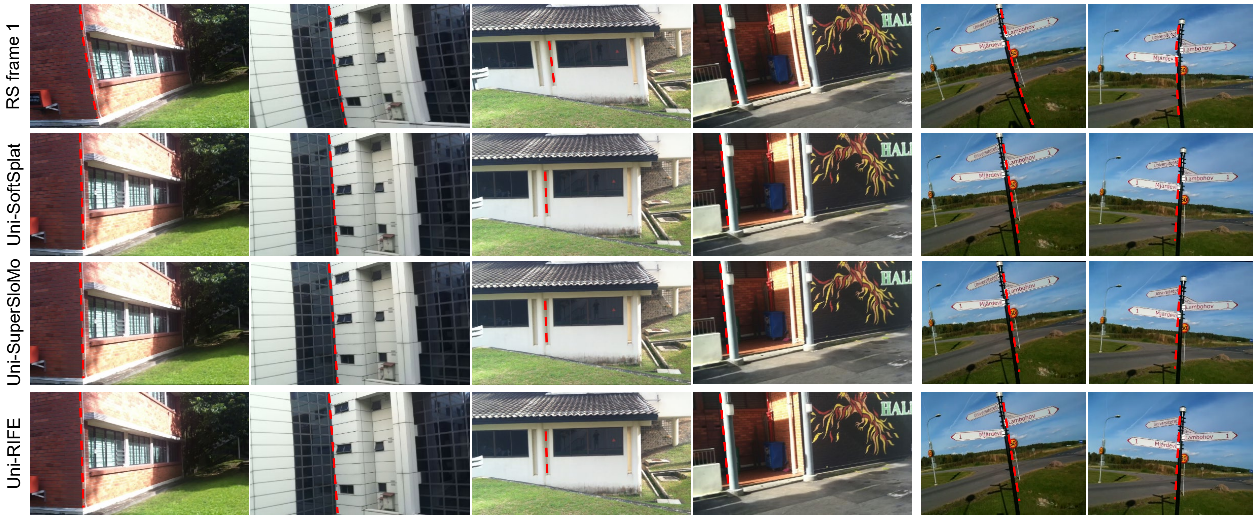


Fig. 12. Cross-shutter transfer results between real-world RS images with varying γ. The left four columns are from [14] ($\gamma = 0.96$) and the right two columns are from [66] ($\gamma = 0.92$). Note that they use the same model pre-trained on the Carla-RS dataset ($\gamma = 1.0$). Our approach can hot-swap to remove various noticeable RS distortions in real-world scenarios for high-quality GS image recovery.

TABLE III

ABLATIONS ON OPTICAL FLOW ESTIMATOR. RESULTS CORRESPONDING TO INTERPOLATION TIME $1 - \gamma$ ON THE CARLA-RS AND FASTEC-RS DATASETS [28] ARE REPORTED. COMPARED TO BUILDING ON PWCNET [34] AND RAFT [33], THE PIPELINE WITH GMFLOW [32] CONSISTENTLY ACHIEVES THE BEST PERFORMANCE, DEMONSTRATING THE EFFECTIVENESS OF THE PROPOSED UNIFIED ARCHITECTURE.

Method	Optical Flow Estimator	PSNR↑ (dB)			SSIM↑		LPIPS↓	
		CRM	CR	FR	CR	FR	CR	FR
Uni-SoftSplat (ft, rand init)	PWCNet	32.29	32.17	28.43	0.928	0.840	0.0237	0.1642
Uni-SoftSplat (ft, pretrain init)	PWCNet	32.44	32.34	28.75	0.933	0.847	0.0204	0.1590
Uni-SoftSplat (ft, rand init)	RAFT	32.07	32.00	29.35	0.931	0.864	0.0231	0.0784
Uni-SoftSplat (ft, pretrain init)	RAFT	32.24	32.15	29.52	0.935	0.866	0.0212	0.0781
Uni-SoftSplat (ft, rand init)	GMFlow	32.92	32.80	29.85	0.938	0.869	0.0197	0.0814
Uni-SoftSplat (ft, pretrain init)	GMFlow	**33.12**	**33.02**	**29.99**	**0.942**	**0.872**	**0.0182**	**0.0767**
Uni-SuperSloMo (ft, rand init)	PWCNet	32.77	32.65	28.53	0.936	0.839	0.0299	0.1235
Uni-SuperSloMo (ft, pretrain init)	PWCNet	32.88	32.75	28.57	0.938	0.842	0.0243	0.1230
Uni-SuperSloMo (ft, rand init)	RAFT	32.89	32.75	29.35	0.940	0.857	0.0242	0.0743
Uni-SuperSloMo (ft, pretrain init)	RAFT	32.94	32.81	29.37	0.941	0.858	0.0227	0.0738
Uni-SuperSloMo (ft, rand init)	GMFlow	33.15	33.07	29.22	0.941	0.856	0.0249	0.0759
Uni-SuperSloMo (ft, pretrain init)	GMFlow	**33.20**	**33.11**	**29.49**	**0.942**	**0.859**	**0.0226**	**0.0710**

TABLE IV

COMPARISONS UNDER DIFFERENT FINE-TUNING SETTINGS. THE DATA PERCENTAGE (DP) USED FOR FINE-TUNING IS NOTED. FINE-TUNING ON NEW SHUTTER DATA WITH FEWER TRAINING SAMPLES (~10%) CAN CONVERGE THE PERFORMANCE TO A SATISFACTORY LEVEL.

Method	Carla-RS		Fastec-RS		DP
	PSNR	SSIM	PSNR	SSIM	
Uni-SoftSplat	32.57 **33.02**	0.937 **0.942**	29.07 **29.99**	0.858 **0.872**	~10% 100%
Uni-SuperSloMo	32.74 **33.11**	0.939 **0.942**	28.57 **29.49**	0.844 **0.859**	~10% 100%
Uni-RIFE	32.46 **32.82**	0.932 **0.936**	29.23 **30.13**	0.857 **0.875**	~10% 100%

F. Ablations on Optical Flow Estimator

Previously, we employed GMFlow [32] as the optical flow estimation backbone of UniSoftSplat and Uni-SuperSloMo. To investigate the influence of optical flow estimation baselines on the proposed method, we replace GMFlow with other classical optical flow estimators, namely PWCNet [34] and RAFT [33]. Note that the recently proposed GMFlow surpasses PWCNet and RAFT in terms of optical flow estimation accuracy. As shown in Table III as well as Fig. 2, the overall performance decreases when using the PWCNet or RAFT as the optical flow estimator, but it is still significantly better than the state-of-the-art RS-based VR methods (*e.g.*, [2], [13]). This demonstrates the flexibility and scalability of our proposed architecture. In the future, our UniVR will also benefit from the advancement of optical flow estimation models.

G. Effectiveness on Limited Fine-tuning Data

During the construction process of the RS dataset in [28], the Carla-RS dataset has 210 video sequences for training, each containing 9 consecutive RS image pairs; the Fastec-RS dataset is divided into a training set of 56 sequences, each with 33 consecutive RS image pairs. We previously fine-tuned the proposed UniVR models using all the training data (which is abundant), and the effectiveness of our method has been validated. To simulate new shutter data with limited training samples and evaluate the performance of our method, we sample the training sets of Carla-RS and Fastec-RS datasets. Specifically, we take the first set of RS image pairs of each sequence in the Carla-RS training set, resulting in a sampled data percentage of 1/9. For the Fastec-RS dataset, we take the first three sets of RS image pairs of each sequence for fine-tuning, resulting in a sampled data percentage of 1/11. From Table IV, we can observe that, even when dealing with

TABLE V
QUANTITATIVE COMPARISONS OF RS CORRECTION RESULTS ON THE REAL-WORLD BS-RSC DATASET [30]. GRAY BACKGROUNDS ARE USED TO MARK THE METHODS THAT CAN ONLY PRODUCE ONE GS IMAGE. OUR METHOD HAS COMPARABLE PERFORMANCE TO STATE-OF-THE-ART RS CORRECTION METHODS. IT IS WORTH MENTIONING THAT OUR METHOD CAN RECOVER LATENT GS IMAGES AT ARBITRARY INTERPOLATION TIMES, REGARDLESS OF THE SHUTTER TYPE OF THE INPUT IMAGE.

Method	PSNR$\uparrow$ (dB)	SSIM$\uparrow$
DiffSfM [14]	19.80	0.698
JCD [58]	25.59	0.841
SUNet [56]	27.76	0.875
AdaRSC [30]	28.23	0.882
JAMNet [29]	32.93	0.941
DFRSC_2F [65]	**33.39**	**0.947**
FRSC [57]	24.97	0.872
DeepUnrollNet [28]	25.21	0.833
RSSR [13]	26.47	0.880
CVR [2]	28.14	0.895
Uni-SoftSplat (Ours)	**32.60**	**0.944**
Uni-SuperSloMo (Ours)	31.57	0.929
Uni-RIFE (Ours)	31.60	0.927

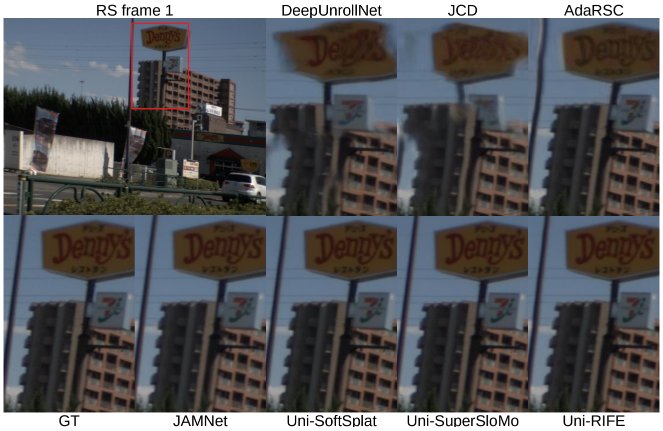


Fig. 13. Qualitative comparisons with state-of-the-art RS correction methods on the BS-RSC dataset [30]. Compared to DeepUnrollNet [28], JCD [58], AdaRSC [30], and JAMNet [29], our approach can successfully remove real-world RS artifacts and estimate high-fidelity GS images.

new shutter data with fewer training images, our pipeline still demonstrates competitive performance. This is attributed to our unified architecture that bridges the gap between GS-based and RS-based VR tasks, enabling cross-shutter information sharing and mutual collaboration.

H. RS Correction Results on Real-world BS-RSC Dataset

Since the real-world BS-RSC dataset [30] is released for time-specific RS correction task, we leverage it to train our method to generate the GS image corresponding to the central scanline of the second RS frame, *i.e.*, $t = 1 - \gamma/2$. As shown in Table V and Figs. 13 and 14, our unified model can perform well for real-world RS correction, where significant de-distortion capabilities are highlighted, even surpassing networks dedicated to the time-specific RS correction. Note that this does not require laborious network design, but only the migration of the GS-based VR method in conjunction with our tractable shutter adapter. Taken together, our pipeline owns great potential both for time-specific RS effect removal and for time-arbitrary GS interpolation.

V. Discussion and Limitation

A. Motivation for Selecting GS-based VR Baselines

We approach three well-established GS-based VR baselines (*i.e.*, SoftSplat [15], SuperSloMo [3], and RIFE [16]), and adapt them to three tractable unified VR models (namely, Uni-SoftSplat, Uni-SuperSloMo, and Uni-RIFE) by embedding the parameter-free shutter adapter. We selected these three GS-based VR baselines because they are the most representative flow-based frame interpolation frameworks. (i) Among the methods oriented to the intermediate motion approximation of forward warping (*Type 1*), the most prevalent GS-based VR baseline is SoftSplat [15]. We believe that some subsequent improvements of SoftSplat, such as [1], [45], [67], can also be easily transformed into an effective UniVR model in the same vein. (ii) Among the methods oriented to the intermediate motion approximation of backward warping (*Type 2*), the pioneering GS-based VR method is SuperSloMo [3]. We believe that some of its subsequent improvement work, such as [43], [44], [61], can also be similarly converted into a viable UniVR model. (iii) Among the methods of directly learning intermediate motions by sub-networks (*Type 3*), the seminal GS-based VR baseline is RIFE [16]. It will also be feasible to adapt its successors, *e.g.*, [52], [53], [54], by simply changing the temporal interpolation instance outside the network. In the future, we will port our ideas to these follow-up GS-based VR methods to more fully validate the generality and effectiveness of our proposed shutter adapter.

B. Discussion on the Characteristics of Three Motion Models

As far as we know, there is no common conclusion in the generally-applicable GS-based VR about which forward, backward, and learning motion models are better. We have the following observations: (i) The forward model is more conducive to temporal continuity interpolation, but it may lead to hole artifacts when multiple pixels are mapped to the same position [2], [13], [15], [26]. (ii) The backward model is more efficient in implementation, but recovering a smooth video often requires supervision from a larger number of GS GT frames [3], [43], [61], [68]. (iii) The learning model can adaptively estimate more general motion in a data-dependent manner without the need for motion assumptions, becoming a hot topic in the past two years [16], [31], [51], [52], [53], [54]. However, to achieve consistent multi-frame reconstruction, it also relies on more moments of GS GT for supervision.

C. Limitation and Future Work

Our approach can be adapted to GS and various RS inputs by customizing the readout time ratio $\gamma \in [0, 1]$. Note that the GS-based VR method is a special case of our unified VR model when $\gamma = 0$. Nevertheless, there are still several limitations, such as:

- **Model architecture aspect**. In this paper, we derive three UniVR variants, *i.e.*, Uni-SoftSplat, Uni-SuperSloMo, and Uni-RIFE, which can be used to reconstruct a coherent video from two consecutive images, regardless of the input shutter type. They involve intermediate motion

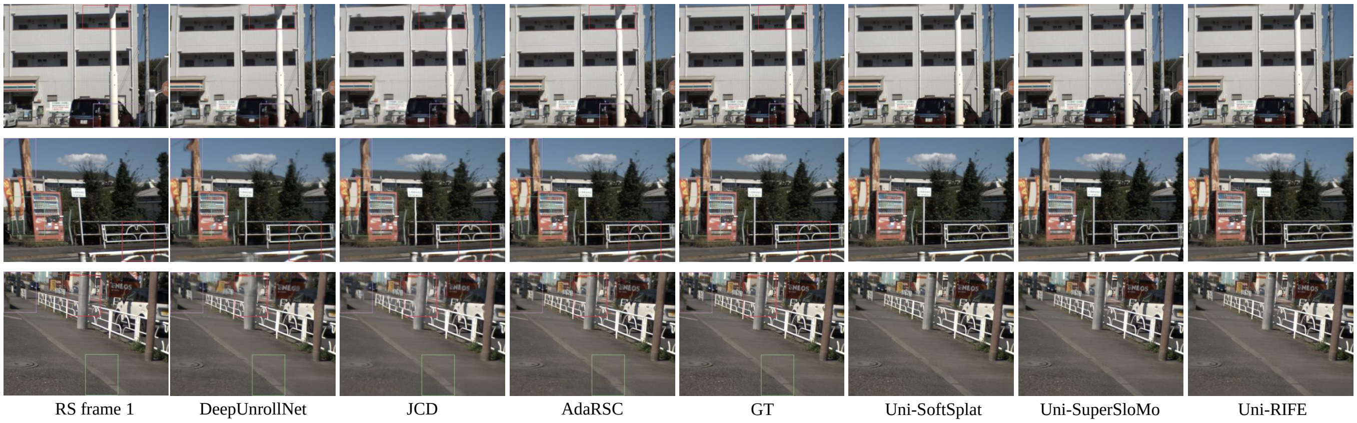


Fig. 14. Four sets of RS correction results on the real-world BS-RSC dataset [30]. Existing RS correction methods (*i.e.*, DeepUnrollNet [28], JCD [58], AdaRSC [30]) are difficult to completely remove RS distortions and even introduce other undesired artifacts (*e.g.*, ghosting, non-smoothing, missing details, local errors). In contrast, our approach obtains higher-quality GS image reconstruction results.

estimation based on forward warping [15], backward warping [3], and direct learning [16], respectively. As such, integrating their essence into a more unified and versatile VR model architecture deserves further research.

- **Training data aspect**. Our UniVR insight might shed some light on jointly training a single model based on diverse shutter datasets to adapt to cameras with multiple shutter types. We have not shown such an experiment in this paper. This would require addressing some additional challenges, *e.g.*, how to balance different shutter data during joint training. In this paper, we focus on and demonstrate the feasibility and effectiveness of extending the well-established GS-based VR approach to compatible GS and RS cameras, where abundant or limited new RS training data is employed (*cf.*, Sec. IV-G). In the future, it is promising that a single UniVR model is jointly trained on a collection of datasets spanning different shutters.
- **Multi-frame fusion aspect**. We have verified the applicability of our UniVR model to the 2-frame input setting. Recent studies have shown that exploiting temporal contextual information from neighboring multiple frames can effectively improve the image reconstruction quality, whether based on GS [46], [47], [69] or RS [27], [30], [58] cameras. Therefore, extending our UniVR pipeline to multi-frame inputs is an intriguing future direction.
- **Extreme scene aspect**. Currently, both GS-based and RS-based VR methods primarily focus on clear video frames. However, in real-world applications, extreme conditions such as low-light [70] and high-speed motion [71] often degrade the image quality. Developing a unified model to address these challenges would be a valuable research topic, which can open up new avenues for improving VR models in complex environments.

VI. Conclusion

In this paper, we proposed a universal video reconstruction framework that processes a variety of shutters using a single model with shared parameters. Compared with shutter-specific methods, our approach shows a superior ability for cross-shutter transfer, which opens up the possibility of bridging the shutter-generic video reconstruction task. Based on models pre-trained on large-scale GS datasets, our approach allows hot-swapping to various RS camera types during inference. With fine-tuning new shutter data, our approach significantly outperforms the best shutter-specific methods in terms of performance and generalization.

References

[1] G. Zhang, Y. Zhu, H. Wang, Y. Chen, G. Wu, and L. Wang, “Extracting motion and appearance via inter-frame attention for efficient video frame interpolation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 5682–5692.

[2] B. Fan, Y. Dai, Z. Zhang, Q. Liu, and M. He, “Context-aware video reconstruction for rolling shutter cameras,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 572–17 582.

[3] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. 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 (CVPR)*, 2018, pp. 9000–9008.

[4] L. Kong, B. Jiang, D. Luo, W. Chu, Y. Tai, C. Wang, and J. Yang, “Dynamic frame interpolation in wavelet domain,” *IEEE Transactions on Image Processing*, vol. 32, pp. 5296–5309, 2023.

[5] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, “Video compression through image interpolation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 416–431.

[6] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao, “High-order model and dynamic filtering for frame rate up-conversion,” *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3813–3826, 2018.

[7] W. Shangguan, Y. Sun, W. Gan, and U. S. Kamilov, “Learning cross-video neural representations for high-quality frame interpolation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 511–528.

[8] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, “Video frame interpolation and enhancement via pyramid recurrent framework,” *IEEE Transactions on Image Processing*, vol. 30, pp. 277–292, 2020.

[9] L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. C. Loy, and Z. Liu, “Deep animation video interpolation in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6587–6595.

[10] A. Paliwal and N. K. Kalantari, “Deep slow motion video reconstruction with hybrid imaging system,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1557–1569, 2020.

[11] W. Shang, D. Ren, C. Feng, X. Wang, L. Lei, and W. Zuo, “Self-supervised learning to bring dual reversed rolling shutter images alive,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 13 086–13 094.

[12] X. Ji, Z. Wang, Z. Zhong, and Y. Zheng, “Rethinking video frame interpolation from shutter mode induced degradation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 12 259–12 268.

[13] B. Fan and Y. Dai, "Inverting a rolling shutter camera: Bring rolling shutter images to high framerate global shutter video," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 4228–4237.

[14] B. Zhuang, L.-F. Cheong, and G. Hee Lee, "Rolling-shutter-aware differential SfM and image rectification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 948–956.

[15] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5437–5446.

[16] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 624–642.

[17] M. Meingast, C. Geyer, and S. Sastry, "Geometric models of rolling-shutter cameras," *arXiv preprint arXiv:cs/0503076*, 2005.

[18] H. Wu, L. Xiao, and Z. Wei, "Simultaneous video stabilization and rolling shutter removal," *IEEE Transactions on Image Processing*, vol. 30, pp. 4637–4652, 2021.

[19] B. Fan, Y. Dai, and M. He, "Rolling shutter camera: Modeling, optimization and learning," *Machine Intelligence Research*, vol. 20, no. 6, pp. 783–798, 2023.

[20] C. Albl, Z. Kukelova, V. Larsson, M. Polic, T. Pajdla, and K. Schindler, "From two rolling shutters to one global shutter," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2505–2513.

[21] B. Fan, Y. Guo, Y. Dai, C. Xu, and B. Shi, "Self-supervised learning for rolling shutter temporal super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[22] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1360–1367.

[23] W. Yan, R. T. Tan, B. Zeng, and S. Liu, "Deep homography mixture for single image rolling shutter correction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 9868–9877.

[24] F. Bai, A. Sengupta, and A. Bartoli, "Scanline homographies for rolling-shutter plane absolute pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8993–9002.

[25] Y. Lao and O. Ait-Aider, "Rolling shutter homography and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 8, pp. 2780–2793, 2021.

[26] B. Fan, Y. Dai, and H. Li, "Rolling shutter inversion: Bring rolling shutter images to high framerate global shutter video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6214–6230, 2022.

[27] D. Qu, Y. Lao, Z. Wang, D. Wang, B. Zhao, and X. Li, "Towards nonlinear-motion-aware and occlusion-robust rolling shutter correction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 10 680–10 688.

[28] P. Liu, Z. Cui, V. Larsson, and M. Pollefeys, "Deep shutter unrolling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5941–5949.

[29] B. Fan, Y. Mao, Y. Dai, Z. Wan, and Q. Liu, "Joint appearance and motion learning for efficient rolling shutter correction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 5671–5681.

[30] M. Cao, Z. Zhong, J. Wang, Y. Zheng, and Y. Yang, "Learning adaptive warping for real-world rolling shutter correction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 785–17 793.

[31] B. Fan, Y. Dai, and H. Li, "Learning bilateral cost volume for rolling shutter temporal super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3862–3879, 2024.

[32] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8121–8130.

[33] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 402–419.

[34] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.

[35] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4161–4170.

[36] C. Liu, H. Yang, J. Fu, and X. Qian, "TTVFI: Learning trajectory-aware transformer for video frame interpolation," *IEEE Transactions on Image Processing*, vol. 32, pp. 4728–4741, 2023.

[37] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[38] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 933–948, 2021.

[39] Y. Luo, J. Pan, and J. Tang, "Bi-directional pseudo-three-dimensional network for video frame interpolation," *IEEE Transactions on Image Processing*, vol. 31, pp. 6773–6788, 2022.

[40] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1701–1710.

[41] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3703–3712.

[42] G. L. Moing, J. Ponce, and C. Schmid, "CCVS: Context-aware controllable video synthesis," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 14 042–14 055.

[43] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 109–125.

[44] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 14 539–14 548.

[45] P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, "Many-to-many splatting for efficient video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3553–3562.

[46] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[47] M. Liu, C. Xu, C. Yao, C. Lin, and Y. Zhao, "JNMR: Joint non-linear motion regression for video frame interpolation," *IEEE Transactions on Image Processing*, vol. 32, pp. 5283–5295, 2023.

[48] Y. Liu, L. Xie, L. Siyao, W. Sun, Y. Qiao, and C. Dong, "Enhanced quadratic video interpolation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 41–56.

[49] Z. Chi, R. Mohammadi Nasiri, Z. Liu, J. Lu, J. Tang, and K. N. Plataniotis, "All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 107–123.

[50] H. Sim, J. Oh, and M. Kim, "XVFI: Extreme video frame interpolation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 14 489–14 498.

[51] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3532–3542.

[52] L. Kong, B. Jiang, D. Luo, W. Chu, X. Huang, Y. Tai, C. Wang, and J. Yang, "IFRNet: Intermediate feature refine network for efficient frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1969–1978.

[53] J. Park, J. Kim, and C.-S. Kim, "BiFormer: Learning bilateral motion estimation via bilateral transformer for 4K video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1568–1577.

[54] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless, "FILM: Frame interpolation for large motion," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 250–266.

[55] E. Naor, I. Antebi, S. Bagon, and M. Irani, "Combining internal and external constraints for unrolling shutter in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 119–134.

[56] B. Fan, Y. Dai, and M. He, "SUNet: Symmetric undistortion network for rolling shutter correction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 4541–4550.

[57] D. Qu, B. Liao, H. Zhang, O. Ait-Aider, and Y. Lao, “Fast rolling shutter correction in the wild,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11 778–11 795, 2023.

[58] Z. Zhong, Y. Zheng, and I. Sato, “Towards rolling shutter correction and deblurring in dynamic scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9219–9228.

[59] B. Zhuang and Q.-H. Tran, “Image stitching and rectification for hand-held cameras,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 243–260.

[60] B. Fan, Y. Dai, and K. Wang, “Rolling-shutter-stereo-aware motion estimation and image correction,” *Computer Vision and Image Understanding*, vol. 213, p. 103296, 2021.

[61] F. A. Reda, D. Sun, A. Dundar, M. Shoeybi, G. Liu, K. J. Shih, A. Tao, J. Kautz, and B. Catanzaro, “Unsupervised video interpolation using cycle consistency,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 892–900.

[62] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, “Deep video deblurring for hand-held cameras,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1279–1288.

[63] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[65] M. Cao, S. Yang, Y. Yang, and Y. Zheng, “Rolling shutter correction with intermediate distortion flow estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 25 338–25 347.

[66] P.-E. Forssén and E. Ringaby, “Rectifying rolling shutter video from hand-held devices,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 507–514.

[67] S. Niklaus, P. Hu, and J. Chen, “Splatting-based synthesis for video frame interpolation,” in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 713–723.

[68] Z. Li, Z.-L. Zhu, L.-H. Han, Q. Hou, C.-L. Guo, and M.-M. Cheng, “AMT: All-pairs multi-field transforms for efficient frame interpolation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9801–9810.

[69] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, “FLAVR: Flow-agnostic video representations for fast frame interpolation,” in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 2071–2082.

[70] C. Li, C. Guo, L. Han, J. Jiang, M.-M. Cheng, J. Gu, and C. C. Loy, “Low-light image and video enhancement using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9396–9416, 2021.

[71] M. Jin, G. Meishvili, and P. Favaro, “Learning to extract a video sequence from a single motion-blurred image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6334–6342.

Bin Fan received the B.S. degree, M.E. degree, and Ph.D. degree from Northwestern Polytechnical University, Xi’an, China, in 2016, 2019, and 2023, respectively. He was selected to the CVPR 2022 Doctoral Consortium (the only one among Chinese universities). He co-organized the CVPR 2023 and ACCV 2022 tutorials on the topic of rolling-shutter cameras. His research interests include computer vision, image processing, and deep learning, especially regarding the rolling shutter camera.

Zhexiong Wan received the B.E. degree from Northwestern Polytechnical University, Xi’an, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information. His research interests include event-based vision, optical flow estimation, scene understanding, and deep learning.

Boxin Shi received the B.E. degree from the Beijing University of Posts and Telecommunications, the M.E. degree from Peking University, and the Ph.D. degree from the University of Tokyo, in 2007, 2010, and 2013. He is currently a Boya Young Fellow Associate Professor (with tenure) and Research Professor at Peking University, where he leads the Camera Intelligence Lab. Before joining PKU, he did research with MIT Media Lab, Singapore University of Technology and Design, Nanyang Technological University, National Institute of Advanced Industrial Science and Technology, from 2013 to 2017. His papers were awarded as Best Paper, Runners-Up at CVPR 2024, ICCP 2015 and selected as Best Paper candidate at ICCV 2015. He is an associate editor of TPAMI/IJCV and an area chair of CVPR/ICCV/ECCV. He is a senior member of IEEE.

Chao Xu received the B.E. degree from Tsinghua University, Beijing, China, in 1988, the M.S. degree from the University of Science and Technology of China, Hefei, China, in 1991, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 1997. From 1991 to 1994, he was an Assistant Professor with the University of Science and Technology of China. Since 1997, he has been with the School of Electronics Engineering and Computer Science (EECS), Peking University, Beijing, where he is currently a professor. His research interests are in image and video coding, processing, and understanding. He has authored or co-authored more than publications and five patents in these fields.

Yuchao Dai received the B.E. degree, M.E degree and Ph.D. degree all in signal and information processing from Northwestern Polytechnical University (NPU), Xi’an, China, in 2005, 2008 and 2012, respectively. He is currently a Professor with School of Electronics and Information at NPU. He was an ARC DECRA Fellow with the Research School of Engineering at the Australian National University, Canberra, Australia. He won the Best Paper Award in IEEE CVPR 2012, the Best Paper Award Nominee at IEEE CVPR 2020, the DSTO Best Fundamental Contribution to Image Processing Paper Prize at DICTA 2014, the Best Algorithm Prize in NRSFM Challenge at CVPR 2017, the Best Student Paper Prize at DICTA 2017, the Best Deep/Machine Learning Paper Prize at APSIPA ASC 2017. He served/serves as Area Chair in CVPR, ICCV, NeurIPS, ACM MM etc. He serves as Publicity Chair in ACCV 2022 and Distinguished Lecture at APSIPA. His research interests include structure from motion, multi-view geometry, low-level computer vision, deep learning, compressive sensing and optimization.