Deblur4DGS: 4D Gaussian Splatting from Blurry Monocular Videos

Renlong Wu, Zhilu Zhang*, Mingyang Chen, Zifei Yan, Wangmeng Zuo

Harbin Institute of Technology
hirenlongwu@gmail.com,cszlzhang@outlook.com, youngmchan269@gmail.com,
yanzifei@hit.edu.cn,wmzuo@hit.edu.cn

Abstract

Recent 4D reconstruction methods have yielded impressive results but rely on sharp videos as supervision. However, motion blur often occurs in videos due to camera shake and object movement, while existing methods render blurry results when using such videos for reconstructing 4D models. Although a few approaches attempted to address the problem, they struggled to produce high-quality results, due to the inaccuracy in estimating continuous dynamic representations within the exposure time. Encouraged by recent works in 3D motion trajectory modeling using 3D Gaussian Splatting (3DGS), we take 3DGS as the scene representation manner, and propose Deblur4DGS to obtain a high-quality 4D model from blurry monocular video. Specifically, we transform continuous dynamic representations estimation within an exposure time into the exposure time estimation. Moreover, we introduce the exposure regularization term, multi-frame, and multi-resolution consistency regularization term to avoid trivial solutions. Furthermore, to better represent objects with large motion, we suggest blur-aware variable canonical Gaussians. Beyond novel-view synthesis, Deblur4DGS can be applied to improve blurry video from multiple perspectives, including deblurring, frame interpolation, and video stabilization. Extensive experiments in both synthetic and real-world data on the above four tasks show that Deblur4DGS outperforms state-of-the-art 4D reconstruction methods.

Introduction

Substantial efforts have been made for 4D reconstruction, which has extensive applications in augmented reality and virtual reality. To model static scenes, Neural Radiance Field (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl, Kopanas, and et al. 2023) propose implicit neural representation manner and explicit Gaussian ellipsoids one, respectively. To model dynamic objects, implicit neural fields (Zhu, Liang, and et al. 2024; Yang et al. 2024b; Wu et al. 2024; Yan, Li, and Lee 2023) and explicit deformation (Duan et al. 2024b; Chu, Ke, and Fragkiadaki 2024; Katsumata, Vo, and Nakayama 2024; Lin, Dai, and et al. 2024; Li, Chen, and et al. 2024; Wang, Ye, and et al. 2024) are suggested for motion representation. While achieving great progress, most methods rely on synchronized multi-view videos. They yield unsatisfactory results when applied to monocular video, where dynamic objects are only observed once at each timestamp. To alleviate the under-constrained nature of the problem, recent studies have introduced data-driven priors, such as depth maps (Lee et al. 2023; Yang et al. 2024a), optical flows (Gao, Xu, and et al. 2024; Zhu, Liang, and et al. 2024), tracks (Seidenschwarz and et al. 2024; Lei et al. 2024), and generative models (Wu et al. 2025; Chu, Ke, and Fragkiadaki 2024; Zeng, Jiang, and et al. 2025) for better 4D reconstruction.

Unfortunately, motion blur often arises due to camera shake and object movement. When reconstructing the 4D scene from the blurry video, the above methods usually render blurry results. The first step to solving this problem is to deal with camera motion blur, which is relatively simple. Some NeRF-based (Lee, Lee, and et al. 2023; Wang et al. 2023; Lee, Oh, and et al. 2023) and 3DGS-based (Zhao, Wang, and Liu 2024; Chen and Liu 2024; Oh et al. 2024) methods have suggested jointly optimizing 3D representation and camera poses within the exposure time by calculating the reconstruction loss between the synthetic blurry images and the input blurry frames. In contrast, the object motion blur is more challenging to address, as the solution has to estimate continuous and sharp dynamic representations within the exposure time to simulate blurry frames.

In this work, we take 3DGS (Kerbl, Kopanas, and et al. 2023) as the scene representation manner to explore the problem, driven by two main motivations. First, its successful application in 4D reconstruction make this method highly promising. Second, the explicit 3D motion modeling presents an opportunity to simplify the complex continuous dynamic representations estimation within the exposure time into exposure time estimation, avoiding complex extra motion modeling in DyBluRF (Sun, Li, and et al. 2024; Bui and et al. 2023). Once the exposure time is estimated, continuous dynamic representations can be obtained by directly interpolating between representations at the nearest integer timestamps. We note that the concurrent work BARD-GS (Lu et al. 2025) adopts a similar strategy, but they perform unsatisfactorily due to the under-constrained optimization, especially for large object motion.

Specifically, we propose Deblur4DGS, a Gaussian Splatting framework for 4D reconstruction from blurry monoc-

*Corresponding author.

ular video. For the static scene, we jointly optimize the camera poses at exposure start and end with static Gaussians. For the dynamic objects, we optimize learnable exposure time parameters and dynamic Gaussians of the integer timestamps, simultaneously. Then continuous camera poses and dynamic Gaussians within exposure time can be obtained by interpolation, and they are used to render continuous sharp frames to calculate the reconstruction loss. Moreover, to avoid trivial solutions, we introduce the exposure regularization term, as well as the multi-frame and multi-resolution consistency regularization terms. Furthermore, existing 4D reconstruction methods generally select Gaussians at a single timestamp as canonical Gaussians. However, it may produce results with missing details in scenes with large motion, especially when processing blurry videos with a low frame rate. To alleviate this issue, we suggest variable canonical Gaussians as time progresses based on the image blur level. Gaussians corresponding to the sharper frame are selected as the canonical ones for better blur removal, and each canonical Gaussian is only used for nearby timestamps to reduce difficulty of modeling large motion.

Blurry videos suffer from not only motion blur, but also low frame rates and scene shake generally. Beyond novel-view synthesis, the optimized Deblur4DGS can be applied to address these problems, achieving deblurring, frame interpolation, and video stabilization. We evaluate Deblur4DGS from all four perspectives. Extensive experiments on both synthetic and real-world data demonstrate that Deblur4DGS outperforms state-of-the-art 4D reconstruction methods quantitatively and qualitatively while maintaining real-time rendering speed. Furthermore, Deblur4DGS has competitive capabilities in comparison with task-specific video processing models trained in a supervised manner.

The main contributions can be summarized as follows:

- We propose Deblur4DGS, a 4D Gaussian Splatting framework specially designed to reconstruct a high-quality 4D model from blurry monocular video.
- We propose transforming dynamic representation estimation into exposure time estimation, where a series of regularizations are suggested to tackle under-constrained optimization and blur-aware variable canonical Gaussians is present to better represent dynamic objects.
- Extensive experiments in synthetic and real-world data show that Deblur4DGS outperforms state-of-the-art 4D reconstruction methods on novel-view synthesis, deblurring, frame interpolation, and video stabilization tasks.

Related Work

Image and Video Deblurring

Deep learning-based image (Ren et al. 2023; Li and et al. 2023; Wang et al. 2022; Zhang, Xu, and et al. 2022; Zhang et al. 2024) and video (Zhong et al. 2023; Pan et al. 2023; Zhong, Cao, and et al. 2023; Zhong, Gao, and et al. 2020; Chan et al. 2022) deblurring methods have been widely explored. Compared to image deblurring methods, video ones leverage temporal clues between consecutive frames for more effective restoration. DSTNet (Pan et al. 2023) develops a deep discriminative spatial and temporal network. BasicVSR++ (Chan et al. 2022) improves feature fusion with second-order feature propagation and flow-guided alignment. BSSTNet (Zhang and et al. 2024) introduces a blur map to sufficiently utilize the entire video, achieving recent state-of-the-art. When reconstructing from a blurry video, pre-processing it with the 2D deblurring method is a straightforward manner. However, 2D deblurring methods cannot perceive 3D structures and maintain scene geometric consistency, leading to unsatisfactory scene reconstruction.

3D and 4D Reconstruction

To reconstruct 3D models, NeRF (Mildenhall et al. 2021) and 3DGS (Kerbl, Kopanas, and et al. 2023) introduce implicit neural representation manner and explicit Gaussian ellipsoids one respectively, where the latter generally achieves better results. To reconstruct 4D models, most works (Somraj et al. 2024; Duan et al. 2024a; Lu et al. 2024; Lin, Dai, and et al. 2024; Li, Chen, and et al. 2024; Sun, Jiao, and et al. 2024; Wu et al. 2024; Yang et al. 2024b; Mihajlovic et al. 2024; Wang and et al. 2025) incorporate implicit neural fields and explicit deformation for motion representation. Moreover, to better reconstruct from monocular video, some studies enhance 4D reconstruction with data-driven priors, such as depth maps (Lee et al. 2023; Yang et al. 2024a), optical flows (Gao, Xu, and et al. 2024; Wang and et al. 2025), tracks (Wang, Ye, and et al. 2024; Seidenschwarz and et al. 2024), and generative models (Wu et al. 2025; Chu, Ke, and Fragkiadaki 2024). For example, GFlow (Wang and et al. 2025) utilizes only 2D priors to lift a video to a 4D scene. GaussianMarbles (Stearns, Harley, and et al. 2024) reduces the degrees of freedom of each Gaussian.

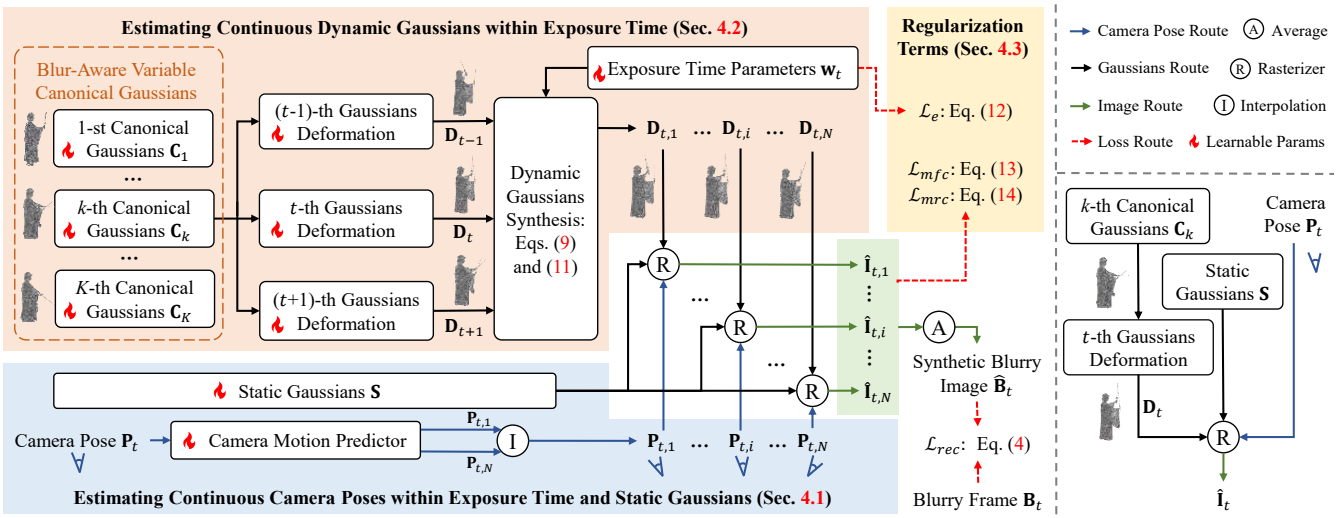
Note that these methods heavily rely on high-quality sharp videos for supervision and perform poorly when facing blurry inputs. To process camera motion in static areas, recent works (Lee, Lee, and et al. 2023; Ma, Li, and et al. 2022; Wang et al. 2023; Lee, Oh, and et al. 2023; Lee et al. 2024a; Zhao, Wang, and Liu 2024; Peng et al. 2024; Lee et al. 2024b; Chen and Liu 2024; Oh et al. 2024) suggest jointly optimizing the scene representation and recovering the camera poses within the exposure time. To process object motion blur in dynamic scenes, DyBluRF (Sun, Li, and et al. 2024; Bui and et al. 2023) incorporates object motion blur formation into dynamic model optimization but faces challenges in producing high-quality images and achieving real-time rendering. In this work, with 3DGS as the scene representation manner, we develop Deblur4DGS to reconstruct a high-quality 4D model from a blurry video.

Preliminary

4D Gaussian Splatting

A 3D Gaussian (Kerbl, Kopanas, and et al. 2023) is parameterized by $\{\mathbf{x}, \mathbf{r}, \mathbf{s}, \mathbf{o}, \mathbf{c}\}$, where $\mathbf{x}$ characterizes the center position in the world space, rotation matrix $\mathbf{r}$ and scale matrix $\mathbf{s}$ define the shape, α is opacity, and spherical harmonics (SH) coefficients $\mathbf{c}$ represent the view-dependent color.

4D Gaussian Splatting (4DGS) usually process static and dynamic regions separately. Static regions can be repre-



(a) Training of Deblur4DGS (b) Rendering of Deblur4DGS

Figure 1: (a) Training of Deblur4DGS. When processing t-th frame, we first discretize its exposure time into N timestamps. Then, we estimate continuous camera poses $\{\mathbf{P}_{t,i}\}_{i=1}^{N}$ and dynamic Gaussians $\{\mathbf{D}_{t,i}\}_{i=1}^{N}$ within exposure time. Next, we render each latent sharp image $\hat{\mathbf{I}}_{t,i}$ with the camera pose $\mathbf{P}_{t,i}$, dynamic Gaussians $\mathbf{D}_{t,i}$ and static Gaussians $\mathbf{S}$. Finally, $\{\hat{\mathbf{I}}_{t,i}\}_{i=1}^{N}$ are averaged to obtain the synthetic blurry image $\hat{\mathbf{B}}_t$, which is used to calculate the reconstruction loss $\mathcal{L}_{rec}$ with the given blurry frame $\mathbf{B}_t$. To regularize the under-constrained optimization, we introduce exposure regularization $\mathcal{L}_e$, multi-frame consistency regularization $\mathcal{L}_{mfc}$ and multi-resolution consistency regularization $\mathcal{L}_{mrc}$. (b) Rendering of Deblur4DGS. Deblur4DGS produces the sharp image with user-provided timestamp t and camera pose $\mathbf{P}_t$.

sented by a set of 3D Gaussians, named $\mathbf{S}$. For the dynamic areas, 4DGS generally selects a timestamp (*e.g.*, the first timestamp) and represents the objects by canonical dynamic Gaussians, *i.e.*, $\mathbf{C}$. Then, $\mathbf{C}$ is deformed to other timestamps for motion representation. Denote by $\mathbf{D}_t$ the dynamic Gaussians at t-th timestamp, it can be written as,

$$\mathbf{D}_t = \mathcal{F}(\mathbf{C}, t; \Theta_{\mathcal{F}}). \tag{1}$$

$\mathcal{F}$ is the deformation operation with parameters $\Theta_{\mathcal{F}}$. The Gaussians for t-th timestamp is the union of $\mathbf{S}$ and $\mathbf{D}_t$.

Collectively, 4DGS models a scene with static Gaussians $\mathbf{S}$, canonical dynamic Gaussians $\mathbf{C}$, and a deformation operation $\mathcal{F}$. With the provided camera pose, the Gaussians at t-th timestamp $\mathbf{D}_t$ can be projected into 2D spaces and rasterized to obtain the corresponding image.

Motion Blur Formation

Motion blur occurs due to camera shake and object movement, which can be regarded as the integration of latent sharp images (Nah and et al. 2017), *i.e.*,

$$\mathbf{B}(u,v) = \phi \int_0^{\tau} \mathbf{I}_t(u,v)dt. \tag{2}$$

$\mathbf{B} \in \mathbb{R}^{H\times W\times 3}$ is the blurry image and $\mathbf{I}_t$ is the latent sharp one at t-th timestamp. (u, v) is pixel location, τ is the camera exposure time, and ϕ is a normalization factor. To approximate the integral operation, recent works (Zhao, Wang, and Liu 2024; Sun, Li, and et al. 2024; Wang et al. 2023) divide the exposure time into N timestamps and regard the blurry image as the average of N sharp images, *i.e.*,

$$\mathbf{B}(u,v) \approx \frac{1}{N}\sum_{i=0}^{N-1} \mathbf{I}_i(u,v). \tag{3}$$

In this work, we reconstruct 4D model from a blurry video by integrating the blur formation into model optimization.

Proposed Method

Let $\{\mathbf{B}_t\}_{t=1}^{T}$ and $\{\mathbf{M}_t\}_{t=1}^{T}$ denote a blurry video with T timestamps and the corresponding masks indicating dynamic areas (extracted by SAM2 (Ravi et al. 2024)), respectively. As shown in fig. 1(a), when processing t-th frame, we first evenly divide its camera exposure time into N timestamps. Then, we estimate continuous camera poses $\{\mathbf{P}_{t,i}\}_{i=1}^{N}$ and dynamic Gaussians $\{\mathbf{D}_{t,i}\}_{i=1}^{N}$ to simulate camera shake and object movement. Next, we render each sharp image $\hat{\mathbf{I}}_{t,i}$ with the corresponding camera pose $\mathbf{P}_{t,i}$, dynamic Gaussians $\mathbf{D}_{t,i}$ and static Gaussians $\mathbf{S}$. After that, we average $\{\hat{\mathbf{I}}_{t,i}\}_{i=1}^{N}$ to obtain the synthetic blurry image $\hat{\mathbf{B}}_t$, which is used to calculate the reconstruction loss $\mathcal{L}_{rec}$ with the given blurry frame $\mathbf{B}_t$, *i.e.*,

$$\mathcal{L}_{rec} = (1-\beta)\mathcal{L}_1(\hat{\mathbf{B}}_t, \mathbf{B}_t) + \beta\mathcal{L}_{ssim}(\hat{\mathbf{B}}_t, \mathbf{B}_t). \tag{4}$$

$\mathcal{L}_1$ and $\mathcal{L}_{ssim}$ are ℓ_1 loss and SSIM (Wang et al. 2004) loss, respectively. β is set to 0.2. The setting all follows 3DGS (Kerbl, Kopanas, and et al. 2023).

Continuous Camera Poses Estimation

To estimate continuous camera poses, recent methods (Zhao, Wang, and Liu 2024; Peng et al. 2024; Chen and Liu 2024; Oh et al. 2024) directly optimize exposure start and end poses (*i.e.*, $\mathbf{P}_{t,1}$ and $\mathbf{P}_{t,N}$). Then, the linear interpolation is performed between $\mathbf{P}_{t,1}$ and $\mathbf{P}_{t,N}$ to obtain the camera pose at i-th intermediate timestamp (*i.e.*, $\mathbf{P}_{t,i}$),*i.e.*,

$$\mathbf{P}_{t,i} = \mathbf{P}_{t,1} \odot \texttt{exp}(\frac{i-1}{N-1} \odot \texttt{log}(\frac{\mathbf{P}_{t,N}}{\mathbf{P}_{t,1}})). \tag{5}$$

`exp` and `log` are exponential and logarithmic functions, respectively. $\odot$ is a pixel-wise multiply operation.

We follow the manner but deploy a tiny MLP as the camera motion predictor (see details in the *Suppl*) for more stable optimization. We pre-train it and static Gaussians $\mathbf{S}$ with static reconstruction loss $\mathcal{L}^s_{rec}$, *i.e.*,

$$\mathcal{L}^s_{rec} = (1-\beta)\mathcal{L}_1(\hat{\mathbf{B}}^s_t, \mathbf{B}^s_t) + \beta\mathcal{L}_{ssim}(\hat{\mathbf{B}}^s_t, \mathbf{B}^s_t). \tag{6}$$

$\hat{\mathbf{B}}^s_t = (1-\mathbf{M}_t) \odot \hat{\mathbf{B}}_t$ and $\mathbf{B}^s_t = (1-\mathbf{M}_t) \odot \mathbf{B}_t$ are the static areas of $\hat{\mathbf{B}}_t$ and $\mathbf{B}_t$, respectively.

Continuous Dynamic Gaussians Estimation

We first introduce blur-aware variable canonical Gaussians for better dynamic representation at integer timestamps. Then, we describe Gaussian deformation manner. Finally, we detail how to take learnable exposure time parameters to obtain continuous dynamic Gaussians within exposure time.

Blur-Aware Variable Canonical Gaussians. Existing 4D reconstruction methods generally select a single canonical Gaussians $\mathbf{C}$ across the entire video, which may produce results with missing details in scenes with large motion. To alleviate the issue, we suggest varying the canonical Gaussians as time progresses. In such case, the k-th canonical Gaussians $\mathbf{C}_k$ is only used for some nearby timestamps, thus reducing the difficulty of motion modeling. One way to achieve this is to uniformly divide the video into K segments and select $\mathbf{C}_k$ for k-th segment. Although it improves performance, selecting the one corresponding to the sharper frame is better for blur removal. In particular, we first uniformly divide the video into K segments and calculate the blur level b_t of dynamic areas for t-th frame following (Bansal, Raj, and Choudhury 2016; Ren, Qian, and Chen 2020), *i.e.*,

$$b_t = \sum_{(u,v)\in\mathbf{M}_t} (\Delta\mathbf{B}_t(u,v) - \overline{\Delta\mathbf{B}_t})^2. \tag{7}$$

$\mathbf{M}_t$ indicates dynamic areas. $\Delta\mathbf{B}_t$ is the image Laplacian and $\overline{\Delta\mathbf{B}_t}$ is its mean value. The larger b_t is, the sharper the frame is. To make the start and end frame of the segment as sharp as possible, we look for the sharp frame among their surrounding H frames and redefine them as the start and end of current segment. Finally, we select the Gaussians for the sharpest frame in each segment as its canonical ones.

Gaussian Deformation. We deform dynamic Gaussians with a set of rigid transformation matrices, following Shape-of-Motion (Wang, Ye, and et al. 2024). Let $\{\mathbf{x}_c, \mathbf{r}_c, \mathbf{s}, \mathbf{o}, \mathbf{c}\}$, $\{\mathbf{x}_t, \mathbf{r}_t, \mathbf{s}, \mathbf{o}, \mathbf{c}\}$, and $\{\mathbf{A}_t, \mathbf{E}_t\}$ denote a Gaussian in $\mathbf{C}_k$, the ones in $\mathbf{D}_t$, and the corresponding transformation matrix, respectively. It can be written as,

$$\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_c + \mathbf{E}_t, \qquad \mathbf{r}_t = \mathbf{A}_t\mathbf{r}_c. \tag{8}$$

Interpolation with Exposure Time Parameters. To get continuous dynamic Gaussians $\{\mathbf{D}_{t,i}\}^N_{i=1}$, one straightforward way is to deploy a series of learnable Gaussian or deformation parameters, but it is unstable to optimize. With the explicit object motion representation in eq. (8), $\mathbf{D}_{t,i}$ can be calculated by interpolating between the ones at the nearest integer timestamps, *i.e.*,

$$\begin{aligned} \mathbf{D}_{t,i} &= \mathbf{w}_{t,i} \odot \mathbf{D}_{t-1} + (1-\mathbf{w}_{t,i}) \odot \mathbf{D}_t,\ i \in [1, N/2], \\ \mathbf{D}_{t,i} &= (1-\mathbf{w}_{t,i}) \odot \mathbf{D}_t + \mathbf{w}_{t,i} \odot \mathbf{D}_{t+1},\ i \in [N/2, N]. \end{aligned} \tag{9}$$

$\mathbf{w}_{t,i}$ is the normalized time interval between $\mathbf{D}_{t,i}$ and $\mathbf{D}_{t,N/2}$. Thus, the problem is transformed to estimate $\mathbf{w}_{t,i}$. In the implementation, we can estimate the one at exposure start and end (*i.e.*, $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,N}$) and then interpolate between them to get the i-th intermediate one $\mathbf{w}_{t,i}$, *i.e.*,

$$\mathbf{w}_{t,i} = (1 - \frac{i-1}{N-1}) \odot \mathbf{w}_{t,1} + \frac{i-1}{N-1} \odot \mathbf{w}_{t,N}. \tag{10}$$

As the object motion within the exposure can be regarded as uniform, the absolute value of $\mathbf{w}_{t,1}$ and $\mathbf{w}_{t,N}$ are equal, which is half the exposure time $\mathbf{w}_t$. Thus, eq. (10) can be re-written as,

$$\mathbf{w}_{t,i} = (1 - \frac{i-1}{N-1}) \odot \frac{\mathbf{w}_t}{2} + \frac{i-1}{N-1} \odot (-\frac{\mathbf{w}_t}{2}). \tag{11}$$

Finally, we set learnable parameters $\mathbf{w}_t$ for continuous dynamic Gaussians estimation within the exposure time. The canonical Gaussians, Gaussian deformation modules, and $\mathbf{w}_t$ are jointly optimized. The reconstruction loss for dynamic areas is similar to eq. (6).

Regularization Terms

After optimization with eq. (4), static areas of $\hat{\mathbf{I}}_{t,i}$ are sharp while dynamic areas can with notable artifacts. The reasons are below. (1) Note that multiple solutions exist for the model to fulfill eq. (4). The most ideal one is that every $\hat{\mathbf{I}}_{t,i}$ is sharp, and the most trivial one is that every $\hat{\mathbf{I}}_{t,i}$ is as blurry as $\mathbf{B}_t$. (2) As static areas are consistent across the entire video, the model tends to learn the underlying sharp representation for inter-frame consistency. In other words, the inter-frame consistency implicitly regularizes model optimization. To further validate this, we conduct an experiment that removes the inter-frame consistency by reducing the number of frames to one. In such a case, the static areas are blurry after optimization with eq. (4), which supports our confirmation. (3) Compared to static areas, the inter-frame consistency in dynamic ones is weaker due to object motion. It may provide insufficient regularization to guide sharp representation learning, thus leading to artifacts. To avoid this, we introduce regularization terms $\mathcal{L}_{reg}$, including exposure regularization $\mathcal{L}_e$, multi-frame consistency term $\mathcal{L}_{mfc}$, and multi-resolution consistency term $\mathcal{L}_{mrc}$.

First, the continuous dynamic Gaussians $\{\mathbf{D}_{t,i}\}^N_{i=1}$ should not be the same. In other words, the value of exposure time parameters $\mathbf{w}_t$ should not be too small. If $\mathbf{w}_t$ is

too small, $\mathbf{D}_{t,i}$ is nearly the same as $\mathbf{D}_t$, leading to trivial solutions. We constrain $\mathbf{w}_t$ by $\mathcal{L}_e$, as,

$$\mathcal{L}_e = \texttt{max}(0, \epsilon - \mathbf{w}_t). \tag{12}$$

max is the maximum function and ϵ is a threshold.

Second, despite different motions, the content of multiple frames within exposure time should be similar. We utilize $\mathcal{L}_{mfc}$ to constrain consistency between neighbor frames, and that between each frame and the first frame, *i.e.*,

$$\begin{aligned}\mathcal{L}_{mfc} = \frac{1}{N-1}\sum_{i=2}^{N}(&\left\|\mathbf{M}_{t,i} \odot (\hat{\mathbf{I}}_{t,i-1\to i} - \hat{\mathbf{I}}_{t,i})\right\|_1 \\ &+ \left\|\mathbf{M}_{t,1} \odot (\hat{\mathbf{I}}_{t,i\to 1} - \hat{\mathbf{I}}_{t,1})\right\|_1).\end{aligned} \tag{13}$$

$\hat{\mathbf{I}}_{t,i-1\to i}$ and $\hat{\mathbf{I}}_{t,i\to 1}$ are obtained by aligning $\hat{\mathbf{I}}_{t,i-1}$ to $\hat{\mathbf{I}}_{t,i}$ and aligning $\hat{\mathbf{I}}_{t,i}$ to $\hat{\mathbf{I}}_{t,1}$ with a pre-trained optical flow network (Sun et al. 2018), respectively. $\mathbf{M}_{t,i}$ and $\mathbf{M}_{t,1}$ are dynamic masks for $\hat{\mathbf{I}}_{t,i}$ and $\hat{\mathbf{I}}_{t,1}$, respectively.

Third, the blur in the lower resolution is lower level and is easier to remove (Kim, Lee, and Cho 2022; Tao, Gao, and et al. 2018), thus the artifacts are less in models trained with down-sampled blurry video. Taking this advantage, we impose $\mathcal{L}_{mrc}$ to assist the optimization of high-resolution models with results from low-resolution models,*i.e.*,

$$\mathcal{L}_{mrc} = ||(\mathbf{M}_{t,i})_\downarrow \odot ((\hat{\mathbf{I}}_{t,i})_\downarrow - \texttt{sg}(\hat{\mathbf{I}}^l_{t,i}))||_1. \tag{14}$$

$\hat{\mathbf{I}}^l_{t,i}$ is the rendered sharp image from the low-resolution model, which is pre-trained by taking the down-sampled video as supervision. $(\cdot)_\downarrow$ is an image down-sampling operation. sg is the stop-gradient operation.

Overall, regularization terms $\mathcal{L}_{reg}$ can be denoted as,

$$\mathcal{L}_{reg} = \lambda_e\mathcal{L}_e + \lambda_{mfc}\mathcal{L}_{mfc} + \lambda_{mrc}\mathcal{L}_{mrc}. \tag{15}$$

λ_e, λ_{mfc}, and λ_{mrc} are set to 0.1, 2, 1, respectively. Besides, following Shape-of-Motion (Wang, Ye, and et al. 2024), we also use some other regularization terms $\mathcal{L}_{oth}$ to help reconstruct 3D motion better, and the details are in the *Suppl*.

Application to Multiple Tasks

The blurry videos suffer from not only motion blur, but also low frame rates and scene shake generally. Beyond novel-view synthesis, Deblur4DGS can adjust the camera poses and timestamps to address these problems, achieving video deblurring, frame interpolation, and video stabilization. First, when inputting camera poses of the blurry video, Deblur4DGS can render corresponding deblurring results. Second, when feeding the interpolated camera poses and timestamps, Deblur4DGS can produce frame-interpolated results. Third, Deblur4DGS can render a more stable video with the smoothed camera poses as inputs.

Experiments

Experimental Settings

Training Details. For stable optimization, we pre-train the camera motion predictor and static Gaussians $\mathbf{S}$ for 400 epochs. After that, we jointly optimize the camera motion predictor, $\mathbf{S}$, canonical dynamic Gaussians $\{\mathbf{C}_k\}_{k=1}^K$, deformable operation $\mathcal{F}$ and exposure time parameters $\{\mathbf{w}_t\}_{t=1}^T$ for 200 epochs. The learning rate for camera motion predictor is set to 5×10^{-4} and decayed to 1×10^{-5}. The learning rate for $\{\mathbf{w}_t\}_{t=1}^T$ is set to 1×10^{-1} and decayed to 1×10^{-5}. The learning rate for $\mathbf{S}$, $\{\mathbf{C}_k\}_{k=1}^K$ and $\mathcal{F}$ follows Shape-of-Motion (Wang, Ye, and et al. 2024). N is set to 11. K and H are set to 5 and 3 respectively. ϵ is set to 1.0. Experiments are conducted with PyTorch (Paszke, Gross, and et al. 2019) on one Nvidia GeForce RTX A6000 GPU.

Methods	PSNR↑/SSIM↑/LPIPS↓ 288×512	PSNR↑/SSIM↑/LPIPS↓ 720×1080
DeformableGS	15.73 / 0.623 / 0.382	15.55 / 0.667 / 0.421
4DGaussians	21.98 / 0.801 / 0.197	21.69 / 0.831 / 0.264
E-D3DGS	23.09 / 0.830 / 0.175	22.46 / 0.844 / 0.258
Shape-of-Motion	26.06 / 0.910 / 0.144	25.81 / 0.897 / 0.246
SplineGS	26.05 / 0.901 / 0.158	24.92 / 0.883 / 0.252
DyBluRF	26.04 / 0.916 / 0.090	25.71 / 0.908 / 0.159
BARD-GS	<u>26.91</u> / <u>0.923</u> / <u>0.077</u>	<u>26.34</u> / <u>0.909</u> / <u>0.139</u>
Deblur4DGS (Ours)	**27.66 / 0.935 / 0.060**	**27.16 / 0.927 / 0.123**

Table 1: novel view synthesis results on synthetic videos.

Methods	CLIPIQA↑/MUSIQ↑ Redmi	PSNR↑/SSIM↑/LPIPS↓ BARD-GS
DeformableGS	0.238 / 25.903	15.63 / 0.781 / 0.361
4DGaussians	0.236 / 26.514	21.32 / 0.863 / 0.221
E-D3DGS	0.257 / 24.967	22.69 / <u>0.883</u> / 0.217
Shape-of-Motion	0.277 / 24.538	20.53 / 0.854 / 0.289
SplineGS	0.252 / 32.022	**23.93 / 0.899** / 0.197
DyBluRF	0.263 / 34.006	22.71 / 0.878 / 0.192
BARD-GS	<u>0.288</u> / <u>35.505</u>	22.69 / 0.874 / <u>0.177</u>
Deblur4DGS (Ours)	**0.356 / 36.756**	<u>23.10</u> / 0.879 / **0.161**

Table 2: novel view synthesis results on real-world videos.

Evaluation Configurations. The synthetic data contains 9 scenes with significant motion blur from Stereo Blur Dataset (Zhou, Zhang, and et al. 2019), where each scene contains blurry stereo videos and the corresponding sharp ones. Note that DyBluRF (Sun, Li, and et al. 2024) conducts experiments on $\times 2.5$ down-sampled data. We evaluate on both $\times 2.5$ down-sampled ones (*i.e.*, 288×512) and the original ones (*i.e.*, 720×1080). For novel-view synthesis and deblurring, the rendering results may be spatially misaligned with ground truth due to the calibration error of camera parameters. Thus, we first freeze the pre-trained 4D model and optimize camera poses by minimizing the photometric error between rendering results and ground truth, and then calculate metrics (*i.e.*, PSNR, SSIM and LPIPS (Zhang et al. 2018)), following COLMAP-Free 3DGS (Fu, Liu, and et al. 2024). As there is no ground truth for frame interpolation and video stabilization, we employ recent no-reference metrics, *i.e.*, CLIPIQA (Wang, Chan, and Loy 2023) and MUSIQ (Ke and et al. 2021). Besides, we evaluate on 6 real-world blurry videos (*i.e.*, Redmi data) captured by a Redmi K50 Ultra smartphone and 12 real-world ones (*i.e.*,

Figure 2: Visual comparisons of novel-view synthesis on real-world videos. Our method produces more photo-realistic details and less visual artifacts in both static and dynamic areas, as marked with yellow and red boxes respectively.

BARD-GS data) from BARD-GS (Lu et al. 2025), where each one contains 24 frames. For novel-view synthesis, we employ no-reference metrics (*i.e.*, CLIPIQA and MUSIQ) and full-reference metrics (*i.e.*, PSNR, SSIM and LPIPS) for the two data, respectively. For the other three tasks, we use no-reference metrics due to no ground truth.

Comparison with State-of-the-Art Methods

We compare with 7 state-of-the-art methods (*i.e.*, DeformableGS (Yang et al. 2024b), 4DGaussians (Wu et al. 2024), E-D3DGS (Bae et al. 2024), Shape-of-Motion (Wang, Ye, and et al. 2024), SplineGS (Park and et al. 2025), DyBluRF (Sun, Li, and et al. 2024) and BARD-GS (Lu et al. 2025)), where DyBluRF and BARD-GS are designed to reconstruct 4D models from blurry monocular videos based on NeRF and 3DGS respectively.

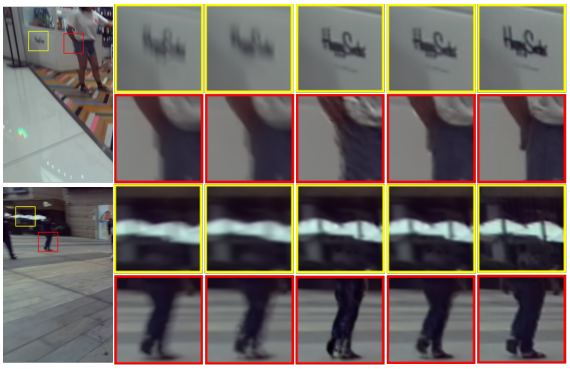
Novel-view synthesis. table 1 and table 2 summarize the results. First, methods (*i.e.*, DyBluRF and BARD-GS) that perform 4D reconstruction and motion blur modeling jointly yield overall better performance, especially in LPIPS score. Although SplineGS gets better PSNR and SSIM scores in BARD-GS data, it produces blurry outputs. It is consistent with the finding in BARD-GS (Wang, Ye, and et al. 2024) that PSNR can sometimes yield higher values even when images appear blurrier. Second, benefiting the explicit 3D representation manner, BARD-GS outperforms DyBluRF, being the most competitive method. Third, compared with BARD-GS, our Deblur4DGS performs better due to the introduction of a series of regularization terms to avoid trivial solutions and blur-aware variable canonical Gaussians to better represent dynamic objects. Visual results in fig. 2 shows that Deblur4DGS removes blur more clearly and produces less visual artifacts in both static and dynamic areas. Per-scene results and more visual results are in the *Suppl*.

In addition, to further demonstrate the effectiveness of Deblur4DGS, we first pre-process the blurry videos with state-of-the-art image (*i.e.*, Restormer (Zamir et al. 2022)) or video (DSTNet (Pan et al. 2023) and BSSTNet (Zhang and et al. 2024)) deblurring methods and then perform 4D reconstruction. The results are summarized in the *Suppl*. Compared with reconstruction from blurry videos, the incorporation of deblurring models improves performance. This is because the deblurring models remove some blur, facilitating sharp scene reconstruction. However, as the deblurring methods cannot perceive 3D structure and maintain scene geometric consistency, the reconstruction results are still unsatisfactory. In contrast, Deblur4DGS jointly reconstructs scene geometry and processes motion blur in 3D space, achieving better scene reconstruction results.

Deblurring. Apart from 4D reconstruction-based methods, we compare with some state-of-the-art image and video deblurring ones. The results are in the *Suppl*. Deblur4DGS obtains better results than 4D reconstruction-based methods and comparable ones to deblurring-specific ones. Compared with the former, Deblur4DGS better reconstructs the scene, thus performing better. Note that the latter ones are trained on large paired data in supervised manner while Deblur4DGS is optimized with the given blurry video in a self-supervised manner. Although the data prior makes them perform better, Deblur4DGS is more convenient to use.

Frame Interpolation. We interpolate camera poses and timestamps to generate $\times 16$ frame interpolation results. We compare with 4D reconstruction-based methods and some video frame interpolation ones (*i.e.*, RIFE (Huang, Zhang, and et al. 2022), EMAVFI (Zhang, Zhu, and et al. 2023), and VIDUE (Shang et al. 2023)). The results are in the *Suppl*. VIDUE is trained with large paired data for joint deblurring and fame interpolation, thus achieving better results.

Video Stabilization. We employ a Gaussian filter to smooth camera poses for video stabilization, following (Peng, Ye, and et al. 2024). The results are in the *Suppl*. Deblur4DGS achieves pleasant scores compared with 2D video stabilization methods (*i.e.*, MeshFlow (Liu et al. 2016) and NNDVS (Zhang et al. 2023)) and 4D reconstruction-based ones, which benefits from better geometry reconstruction.



(a) Blurry (b) w/o ECP, w/o EDG (c) w/ ECP, w/o EDG (d) w/ ECP, w/ EDG (e) Sharp

Figure 3: Effect of continuous camera pose (ECP) and dynamic Gaussian (EDG) estimation.

Ablation Study

We conduct experiments to validate the effectiveness of each strategy. As our strategies in Deblur4DGS are mainly designed to process dynamic areas, we exclude the pixels of static areas to calculate the metrics in dynamic ones.

Effect of ECP and EDG

ECP and EDG are introduced to process camera motion blur and object motion blur, respectively. Quantitative results and visual comparisons are shown in table 3 and fig. 3, respectively. First, without ECP and EDG, the results are almost as blurry as the input frame, as shown in fig. 3(b). Second, only with ECP, the static areas are sharp but may lead to visual artifacts in dynamic areas, as shown in fig. 3(c). It is because ECP cannot simulate the object movement. Third, we further introduce EDG to simulate that, producing visually pleasant results in both areas, as shown in fig. 3(d).

Effect of Regularization Terms

The effect of exposure regularization $\mathcal{L}_e$, multi-frame consistency term $\mathcal{L}_{mfc}$ and multi-resolution consistency term $\mathcal{L}_{mrc}$ are in table 4. Visual results are in the Sec.D of *Suppl*. Without these regularization terms, noticeable artifacts appear in dynamic regions, resulting in degraded performance. By regularizing the object motion within the exposure time distinguished, $\mathcal{L}_e$ improves performance. Besides, $\mathcal{L}_{mfc}$ and $\mathcal{L}_{mrc}$ additionally regularize multi-frame and multi-resolution consistency respectively, helping to alleviate artifacts. Their combinations perform best.

Effect of BAV Canonical Gaussians.

The effect of blur-aware variable (BAV) Canonical Gaussians are in table 5. First, selecting a single canonical Gaussians across the entire video (*i.e.*, None) leads to poor performance, due to the challenge of modeling large object motion. Second, selecting variable canonical Gaussians uniformly (*i.e.*, w/o Blur-Aware) alleviates this, leading to performance gain. We also experiment with an optical flow-based strategy (Shaw, Nazarczuk, and et al. 2024) to select canonical Gaussians, performs similar to the uniform selection. This may be due to the inaccurate estimation of optical flow from blurry images. Third, our blur-aware selection is better, as the canonical Gaussians from the sharper frame help blur removal. Visual results are in the *Suppl*.

ECP	EDG	PSNR↑/SSIM↑/LPIPS↓ 288 × 512	PSNR↑/SSIM↑/LPIPS↓ 720 × 1080
×	×	22.10 / 0.988 / 0.018	22.34 / 0.988 / 0.016
✓	×	22.30 / 0.989 / 0.016	22.39 / 0.988 / 0.015
✓	✓	22.36 / 0.989 / 0.015	22.63 / 0.990 / 0.014

Table 3: Effect about the estimation of continuous camera poses (ECP) and dynamic Gaussians (EDG).

$\mathcal{L}_e$	$\mathcal{L}_{mfc}$	$\mathcal{L}_{mrc}$	PSNR↑/SSIM↑/LPIPS↓ 288 × 512	PSNR↑/SSIM↑/LPIPS↓ 720 × 1080
×	×	×	21.87 / 0.987 / 0.017	22.30 / 0.988 / 0.016
×	✓	✓	22.30 / 0.989 / 0.015	22.56 / 0.989 / 0.015
✓	×	×	22.01 / 0.988 / 0.015	22.40 / 0.989 / 0.015
✓	✓	×	22.16 / 0.989 / 0.016	22.49 / 0.989 / 0.015
✓	×	✓	22.22 / 0.989 / 0.015	22.54 / 0.989 / 0.014
✓	✓	✓	22.36 / 0.989 / 0.015	22.63 / 0.990 / 0.014

Table 4: Effetct of regularization terms (see eq. (15)).

Methods	PSNR↑/SSIM↑/LPIPS↓ 288 × 512	PSNR↑/SSIM↑/LPIPS↓ 720 × 1080
None	22.13 / 0.988 / 0.017	22.30 / 0.988 / 0.016
w/o Blur-Aware	22.29 / 0.989 / 0.016	22.57 / 0.989 / 0.015
Ours	22.36 / 0.989 / 0.015	22.63 / 0.990 / 0.014

Table 5: Effect of blur-aware variable (BAV) canonical Gaussians. 'None' denotes selecting a single one.

Conclusions

In this work, we propose Deblur4DGS, a 4D Gaussian Splatting framework to reconstruct a high-quality 4D model from blurry monocular video. In particular, with the explicit motion trajectory modeling, we propose to transform the challenging continuous dynamic representation estimation within an exposure time into the exposure time estimation, where a series of regularizations are suggested to tackle the under-constrained optimization. Besides, a blur-aware variable canonical Gaussians is present to represent objects with large motion better. Beyond novel-view synthesis, Deblur4DGS can improve blurry video quality from multiple perspectives, including deblurring, frame interpolation, and video stabilization. Extensive results show Deblur4DGS outperforms state-of-the-art 4D reconstruction methods.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62371164 and the National Key RD Program of China under Grant No. 2022YFA1004100.

References

Bae, J.; Kim, S.; Yun, Y.; Lee, H.; Bang, G.; and Uh, Y. 2024. Per-Gaussian Embedding-Based Deformation for Deformable 3D Gaussian Splatting. *arXiv*.

Bansal, R.; Raj, G.; and Choudhury, T. 2016. Blur image detection using Laplacian operator and Open-CV. In *SMART*.

Bui, M.-Q. V.; and et al. 2023. Dyblurf: Dynamic deblurring neural radiance fields for blurry monocular video. *arXiv*.

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*.

Chen, W.; and Liu, L. 2024. Deblur-GS: 3D Gaussian Splatting from Camera Motion Blurred Images. *ACM*.

Chu, W.-H.; Ke, L.; and Fragkiadaki, K. 2024. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *arXiv*.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024a. 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. *arXiv*.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024b. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *SIGGRAPH*.

Fu, Y.; Liu, S.; and et al. 2024. COLMAP-Free 3D Gaussian Splatting. In *CVPR*.

Gao, Q.; Xu, Q.; and et al. 2024. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv*.

Huang, Z.; Zhang, T.; and et al. 2022. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*.

Katsumata, K.; Vo, D. M.; and Nakayama, H. 2024. A Compact Dynamic 3D Gaussian Representation for Real-Time Dynamic View Synthesis. In *ECCV*.

Ke, J.; and et al. 2021. Musiq: Multi-scale image quality transformer. In *ICCV*.

Kerbl, B.; Kopanas, G.; and et al. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM*.

Kim, K.; Lee, S.; and Cho, S. 2022. Mssnet: Multi-scale-stage network for single image deblurring. In *ECCV*.

Lee, B.; Lee, H.; Sun, X.; Ali, U.; and Park, E. 2024a. Deblurring 3d gaussian splatting. *arXiv*.

Lee, D.; Lee, M.; and et al. 2023. Dp-nerf: Deblurred neural radiance field with physical scene priors. In *CVPR*.

Lee, D.; Oh, J.; and et al. 2023. Exblurf: Efficient radiance fields for extreme motion blurred images. In *ICCV*.

Lee, J.; Kim, D.; Lee, D.; Cho, S.; and Lee, S. 2024b. CRiM-GS: Continuous Rigid Motion-Aware Gaussian Splatting from Motion Blur Images. *arXiv*.

Lee, Y.-C.; Zhang, Z.; Blackburn-Matzen, K.; Niklaus, S.; Zhang, J.; Huang, J.-B.; and Liu, F. 2023. Fast view synthesis of casual videos. *arXiv*.

Lei, J.; Weng, Y.; Harley, A.; Guibas, L.; and Daniilidis, K. 2024. MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds. *arXiv*.

Li, J.; and et al. 2023. Self-supervised blind motion deblurring with deep expectation maximization. In *CVPR*.

Li, Z.; Chen, Z.; and et al. 2024. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*.

Lin, Y.; Dai, Z.; and et al. 2024. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *CVPR*.

Liu, S.; Tan, P.; Yuan, L.; Sun, J.; and Zeng, B. 2016. Meshflow: Minimum latency online video stabilization. In *ECCV*.

Lu, Y.; Zhou, Y.; Liu, D.; Liang, T.; and Yin, Y. 2025. Bardgs: Blur-aware reconstruction of dynamic scenes via gaussian splatting. In *CVPR*, 16532–16542.

Lu, Z.; Guo, X.; Hui, L.; Chen, T.; Yang, M.; Tang, X.; Zhu, F.; and Dai, Y. 2024. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *CVPR*.

Ma, L.; Li, X.; and et al. 2022. Deblur-nerf: Neural radiance fields from blurry images. In *CVPR*.

Mihajlovic, M.; Prokudin, S.; Tang, S.; Maier, R.; Bogo, F.; Tung, T.; and Boyer, E. 2024. SplatFields: Neural Gaussian Splats for Sparse 3D and 4D Reconstruction. *arXiv*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *ACM*.

Nah, S.; and et al. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*.

Oh, J.; Chung, J.; Lee, D.; and Lee, K. M. 2024. DeblurGS: Gaussian Splatting for Camera Motion Blur. *arXiv*.

Pan, J.; Xu, B.; Dong, J.; Ge, J.; and Tang, J. 2023. Deep Discriminative Spatial and Temporal Network for Efficient Video Deblurring. In *CVPR*.

Park, J.; and et al. 2025. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *CVPR*.

Paszke, A.; Gross, S.; and et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*.

Peng, C.; Tang, Y.; Zhou, Y.; Wang, N.; Liu, X.; Li, D.; and Chellappa, R. 2024. BAGS: Blur Agnostic Gaussian Splatting through Multi-Scale Kernel Modeling. *arXiv*.

Peng, Z.; Ye, X.; and et al. 2024. 3D Multi-frame Fusion for Video Stabilization. In *CVPR*.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; and et al. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv*.

Ren, M.; Delbracio, M.; Talebi, H.; Gerig, G.; and Milanfar, P. 2023. Multiscale structure guided diffusion for image deblurring. In *ICCV*.

Ren, X.; Qian, Z.; and Chen, Q. 2020. Video deblurring by fitting to test data. *arXiv*.

Seidenschwarz, J.; and et al. 2024. DynOMo: Online Point Tracking by Dynamic Online Monocular Gaussian Reconstruction. *arXiv*.

Shang, W.; Ren, D.; Yang, Y.; Zhang, H.; Ma, K.; and Zuo, W. 2023. Joint Video Multi-Frame Interpolation and Deblurring under Unknown Exposure Time. In *CVPR*.

Shaw, R.; Nazarczuk, M.; and et al. 2024. SWinGS: Sliding Windows for Dynamic 3D Gaussian Splatting.

Somraj, N.; Choudhary, K.; Mupparaju, S. H.; and Soundararajan, R. 2024. Factorized Motion Fields for Fast Sparse Input Dynamic View Synthesis. In *SIGGRAPH*.

Stearns, C.; Harley, A.; and et al. 2024. Dynamic Gaussian Marbles for Novel View Synthesis of Casual Monocular Videos. *arXiv*.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*.

Sun, H.; Li, X.; and et al. 2024. DyBluRF: Dynamic Neural Radiance Fields from Blurry Monocular Video. In *CVPR*.

Sun, J.; Jiao, H.; and et al. 2024. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *CVPR*.

Tao, X.; Gao, H.; and et al. 2018. Scale-recurrent network for deep image deblurring. In *CVPR*.

Wang, J.; Chan, K. C.; and Loy, C. C. 2023. Exploring clip for assessing the look and feel of images. In *AAAI*.

Wang, P.; Zhao, L.; Ma, R.; and Liu, P. 2023. Bad-nerf: Bundle adjusted deblur neural radiance fields. In *CVPR*.

Wang, Q.; Ye, V.; and et al. 2024. Shape of motion: 4d reconstruction from a single video. *arXiv*.

Wang, S.; and et al. 2025. Gflow: Recovering 4d world from monocular video. In *AAAI*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *TIP*.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022. Uformer: A general u-shaped transformer for image restoration. In *CVPR*.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*.

Wu, Z.; Yu, C.; Jiang, Y.; Cao, C.; Wang, F.; and Bai, X. 2025. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. In *ECCV*.

Yan, Z.; Li, C.; and Lee, G. H. 2023. Nerf-ds: Neural radiance fields for dynamic specular objects. In *CVPR*.

Yang, X.; Xie, W.; Fu, Y.; Fan, W.; and Dong, X. 2024a. 4d Gaussian Splatting for High-Fidelity Dynamic Reconstruction of Single-View Scenes. *SSRN*.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024b. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*.

Zeng, Y.; Jiang, Y.; and et al. 2025. Stag4d: Spatial-temporal anchored generative 4d gaussians. In *ECCV*.

Zhang, G.; Zhu, Y.; and et al. 2023. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*.

Zhang, H.; and et al. 2024. Blur-aware Spatio-temporal Sparse Transformer for Video Deblurring. In *CVPR*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhang, Z.; Liu, Z.; Tan, P.; Zeng, B.; and Liu, S. 2023. Minimum latency deep online video stabilization. In *ICCV*.

Zhang, Z.; Xu, R.; and et al. 2022. Self-supervised image restoration with blurry and noisy pairs. *NeurIPS*.

Zhang, Z.; Zhang, S.; Wu, R.; Yan, Z.; and Zuo, W. 2024. Bracketing is all you need: Unifying image restoration and enhancement tasks with multi-exposure images. *ICLR*.

Zhao, L.; Wang, P.; and Liu, P. 2024. Bad-gaussians: Bundle adjusted deblur gaussian splatting. *arXiv*.

Zhong, Z.; Cao, M.; and et al. 2023. Blur interpolation transformer for real-world motion from blur. In *CVPR*.

Zhong, Z.; Gao, Y.; and et al. 2020. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*.

Zhong, Z.; Gao, Y.; Zheng, Y.; Zheng, B.; and Sato, I. 2023. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *IJCV*.

Zhou, S.; Zhang, J.; and et al. 2019. Davanet: Stereo deblurring with view aggregation. In *CVPR*.

Zhu, R.; Liang, Y.; and et al. 2024. MotionGS: Exploring Explicit Motion Guidance for Deformable 3D Gaussian Splatting. *arXiv*.