Gyro-based Deep Video Deblurring

Jaesung Rim[1] Woohyeok Kim[1] Haeyun Lee[2] Heemin Yang[1] Ke Wang[3] Sunghyun Cho[1]
POSTECH[1] KOREATECH[2] Pika Labs[3]
http://cg.postech.ac.kr/research/GyroDVD

(a) Real-world Video Frame (b) Video Deblurring (ShiftNet+ [25]) (c) Gyro-based Image Deblurring (GyroDeblur [60]) **(d) Gyro-based Video Deblurring (GyroDVD - Ours)**

Abstract

Modern cameras, such as smartphone cameras and DSLRs, are equipped with gyro sensors that measure motion of the camera. While the motion information is valuable for deblurring, gyro-based deblurring has not been widely studied, particularly for video. A few gyro-based video deblurring methods have been proposed, but they exhibit inherent limitations. First, gyro sensors capture only rotational motion, leading these methods to ignore translational motion. Second, their dependence on simplified blur models and deconvolution-based solutions restricts overall performance. To address these limitations, we introduce GyroDVD, the first learning-based framework for gyro-based video deblurring. We propose a novel blur kernel construction scheme that jointly accounts for rotational and translational motion. A video deblurring network then restores sharp videos by exploiting the constructed kernels together with the video frames. For training and evaluation, we introduce the GyroVD dataset, a large-scale and realistic dataset specifically designed for gyro-based deblurring. Extensive experiments demonstrate that our method significantly outperforms prior gyro-based image and video deblurring methods.

1. Introduction

Hand-held video capturing in low-light environments is particularly challenging. Camera motion during exposure produces motion blur that severely degrades perceptual quality. To remove blur from videos, traditional video deblurring methods [1, 4, 8, 27, 49, 70] employ multi-image deconvolution to jointly optimize blur kernels and latent sharp frames. Recently, learning-based video deblurring methods [25, 28, 29, 38–40, 48, 50, 59, 62–64, 68] have achieved superior performance compared to traditional approaches. Nevertheless, their performance remains limited, especially in cases of severe blur.

To improve performance, several deblurring methods leverage auxiliary data captured by additional hardware (*e.g.*, dual cameras [3, 23, 26, 43, 45, 52, 53, 57] and event cameras [7, 13, 18, 21, 51, 58, 67]). However, such methods require additional cameras, and the acquisition and storage costs of the auxiliary data are substantially high, especially for videos. Instead, we focus on utilizing gyro sensors for deblurring, which are already integrated into modern cameras, such as smartphone cameras and DSLRs. The gyro sensors measure rotational camera motion at high frame rates (*e.g.*, 400 FPS) with minimal costs, providing valuable motion information for deblurring.

To exploit gyro sensors, several classical methods [34, 46, 47] and learning-based methods [17, 24, 35, 41, 55, 60] have been proposed for single-image deblurring. These approaches estimate a blur kernel from gyro data and restore a sharp image using the blur kernel, achieving improved performance. However, these methods have inherent limitations, as they rely on only rotational motion from gyro data and ignore translational motion. In addition, single-image deblurring remains highly ill-posed even with motion information. As a result, these methods still struggle to recover sharp details and often introduce noticeable artifacts.

Only a few methods [2, 14] address gyro-based video deblurring, employing multi-image deconvolution using blur kernels estimated from gyro data. These approaches rely on restrictive blur models, precise alignment, and deconvolution algorithms, which limit their performance. Further-

more, these methods also ignore translational motion. In particular, when capturing videos, the effect of translational motion becomes significant because the photographer often records videos while walking or moving.

In this paper, we propose GyroDVD (**Gyro**-based **D**eep **V**ideo **D**eblurring), the first learning-based approach for gyro-based video deblurring. We introduce a novel motion model that decomposes the effect of camera motion on each pixel into rotational and translational components, which are estimated from gyro data and video frames. Based on this model, we construct pixel-wise blur kernels that jointly capture both types of motion. Leveraging these kernels for both deblurring and feature propagation across frames, our deblurring network produces high-quality deblurred results.

Training gyro-based video deblurring requires a large-scale dataset, which unfortunately remains unavailable. Existing datasets are primarily designed for image deblurring and often lack realism. Previous methods [24, 41] synthesize blur by averaging frames from the Visual-Inertial dataset [44], but its low frame rate (20 FPS) and lack of moving objects limit realism. Other approaches [17, 35, 60, 65, 66] generate blur kernels from gyro data and convolve sharp images with them, resulting in blur that contains only rotational motion. Some methods simulate translation or object motion using Gaussian acceleration [65, 66] or synthetic moving objects [60], but realism remains limited. These limitations make existing datasets not only difficult to use for training but also unsuitable for video deblurring.

To address the limitations, we propose the GyroVD (**Gyro**-based **V**ideo **D**eblurring) dataset. To construct a realistic dataset, we developed a smartphone app that simultaneously captures high-speed (240 FPS) videos and gyro data. Using this app, we recorded various scenes containing both rotational and translational camera motion, as well as moving objects. Blurred videos were then generated by averaging consecutive frames of the high-speed videos, and the gyro data were sampled according to the videos. In total, GyroVD provides 63,200 blurred frames from 632 videos with the corresponding gyro data. GyroVD is the largest publicly available dataset for video deblurring (Tab. 1) and offers the most comprehensive coverage of realistic motion for gyro-based methods. To further support evaluation, we additionally collected 100 real-world videos with gyro data for no-reference metric analysis.

Our experimental results show that GyroDVD significantly outperforms existing gyro-based image deblurring and video deblurring methods, on both synthetic and real datasets. Our major contributions can be summarized as:

- We introduce a decomposed camera motion model and propose a novel blur kernel construction scheme that reflects both rotational and translational camera motion.
- We propose GyroDVD, the first learning-based framework for gyro-based video deblurring, which employs a network that exploits blur kernels for video deblurring.
- We construct GyroVD, the first gyro-based video deblurring dataset, which is the largest and most realistic dataset for gyro-based deblurring.

2. Related work

Most traditional video deblurring methods [1, 4, 8, 27, 49, 70] formulate the task as a multi-image deconvolution problem, where a single latent frame is deconvolved from multiple blurred frames. On the other hand, several classical methods [9–11, 31] align and directly fuse consecutive video frames without performing deconvolution.

Recently, learning-based video deblurring approaches have been proposed, which learn the temporal fusion process from large-scale training data. Various network architectures have been proposed, including 2D CNNs that process concatenated frames [38, 50] or fuse features across frames [5, 25, 39, 40, 56], 3D CNNs [63], recurrent networks [28, 48, 68, 69], and transformer-based networks [29, 62, 64]. However, under severe motion, the neighboring frames are also heavily degraded, making temporal fusion ineffective. As a result, these methods still struggle to restore sharp details in cases of severe blur.

To improve performance, several deblurring methods have been proposed to leverage gyro sensors. Most gyro-based deblurring methods have focused on image deblurring. Classical gyro-based image deblurring approaches [34, 46, 47] utilize gyro data to estimate blur kernels and apply the deconvolution method using them. Learning-based image deblurring methods [17, 24, 35, 41, 55, 60] also estimate blur kernels from gyro data and use them as additional inputs to deep networks. Various strategies have been explored to exploit the blur kernels effectively, including simple concatenation [24, 35], deformable convolution [17, 60], and attention-based mechanisms [41, 55]. However, all these methods rely solely on rotational motion computed from gyro data and ignore translational motion.

To consider translational motion, two classical methods [16, 19] additionally utilize accelerometer sensors for computing blur kernels. However, constructing blur kernels from acceleration is challenging, as it requires information on both the scene depth and the gravity direction. Moreover, both methods assume that the initial camera velocity is zero (*i.e.*, the camera is stationary at the beginning of the exposure), which does not hold in real-world scenarios.

Only a few methods [2, 14] have been proposed for gyro-based video deblurring. Park *et al.* [14] propose a multi-image deconvolution method using blur kernels estimated from gyro data. Arslan *et al.* [2] adopt grid-based kernel estimation and deconvolution for video deblurring. However, both methods also ignore translational motion and rely on

restrictive blur models, precise alignment, and simple deconvolution, which severely limit their performance.

3. Camera Motion Model

This section introduces our motion model, which enables accurate blur kernel construction. Our method constructs pixel-wise blur trajectories by modeling the underlying camera motion, which consists of rotational and translational motion. Rotational motion can be directly measured using a gyro sensor. However, the sensor does not capture translational motion, such as forward or lateral movement, which also contributes significantly to motion blur.

To estimate translation, we leverage inter-frame optical flow. While optical flow captures the overall motion between frames, it inherently entangles rotation and translation. To address this, we introduce a novel motion model that decomposes the effect of camera motion in the pixel space into rotational and translational components. Based on this model, we isolate the translational component from the optical flow. This enables us to recover translation for each pixel, which is critical for modeling accurate blur kernels. To formalize this process, we begin by modeling the warping of 2D pixel positions under rigid camera motion.

Rigid Camera Motion The camera motion can be modeled as a rigid-body transformation characterized by rotation and translation. Suppose the camera undergoes a motion parameterized by a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. A 2D pixel position $\mathbf{p} = [x, y]^\top$ is warped to a new position $\mathbf{p}'$ according to the standard projection model:

$$\mathbf{p}' = \pi\left(\mathbf{C}\left(\mathbf{R}\,\mathbf{C}^{-1}\mathbf{p}_H + \frac{1}{d}\,\mathbf{t}\right)\right) \tag{1}$$

where $\mathbf{C}$ is the camera intrinsic matrix, $\mathbf{p}_H = [\mathbf{p}^\top, 1]^\top$ is the homogeneous coordinate of $\mathbf{p}$, d is its depth, and $\pi((x, y, w)^\top) = (x/w,\, y/w)^\top$ denotes the projection from a 3D homogeneous coordinate to a 2D Cartesian coordinate.

Decomposed Motion Model If gyro data are available without error and no translational motion exists, pixel-wise blur kernels can be derived by computing the trajectories of warped pixels over the exposure interval using Eq. (1). In video capture, however, translational motion $\mathbf{t}$ plays a significant role, since the camera often moves rather than remaining stable as in still photography. Gyro sensors provide only rotational information and do not measure translation, and Eq. (1) additionally requires the depth d, which is difficult to obtain in practice. Consequently, it is impractical to directly model camera motion blur using Eq. (1).

To address this limitation, we introduce an approximation to Eq. (1). Consider a video frame with an exposure interval (t_s, t_e), and let $\{t_i\}_{i=0}^N$ denote $N{+}1$ uniformly spaced samples such that $t_0 = t_s$ and $t_N = t_e$. Let $\Delta t = t_{i+1} - t_i$ be the uniform temporal gap between consecutive sampling points. Suppose that the camera's rotation matrix $\mathbf{R}_i$ and translation vector $\mathbf{t}_i$ at time t_i are given. By substituting $\mathbf{R}_i$ and $\mathbf{t}_i$ into Eq. (1) and applying a first-order Taylor expansion of the projection function π, the warped pixel position $\mathbf{p}'_i$ at time t_i can be approximated as:

$$\begin{aligned}\mathbf{p}'_i &\approx \pi\big(\mathbf{C}\mathbf{R}_i\mathbf{C}^{-1}\mathbf{p}_H\big) + \tfrac{1}{d}\,\mathbf{J}\mathbf{C}\mathbf{t}_i \\ &= \pi\big(\mathbf{C}\mathbf{R}_i\mathbf{C}^{-1}\mathbf{p}_H\big) + \boldsymbol{\tau}_i\end{aligned} \tag{2}$$

where $\mathbf{J}$ is the Jacobian matrix of π, and $\boldsymbol{\tau}_i$ is a translational motion vector defined as $\boldsymbol{\tau}_i = \frac{1}{d}\,\mathbf{J}\mathbf{C}\mathbf{t}_i$.

In practice, while $\mathbf{R}_i$ can be derived from gyro data, $\mathbf{t}_i$ and d, or equivalently $\boldsymbol{\tau}_i$, are not available. Instead, we approximate the translational motion as having constant velocity and depth over the short time interval. Under this assumption, we obtain our decomposed motion model:

$$\mathbf{p}'_i \approx \pi\big(\mathbf{C}\mathbf{R}_i\mathbf{C}^{-1}\mathbf{p}_H\big) + \frac{t_i - t_s}{t_e - t_s}\,\boldsymbol{\tau} \tag{3}$$

where $\boldsymbol{\tau}$ denotes the cumulative translational motion vector over the exposure interval (t_s, t_e). Here, the translational motion vector at t_i is approximated as $\boldsymbol{\tau}_i \approx \frac{t_i - t_s}{t_e - t_s}\,\boldsymbol{\tau}$. To construct blur kernels that reflect both translational and rotational motion, we estimate $\boldsymbol{\tau}$ using optical flows between consecutive video frames. In the next section, we describe how to estimate $\boldsymbol{\tau}$ and construct blur kernels accordingly.

4. Blur Kernel Construction

Our method constructs pixel-wise camera motion trajectories from gyro data and optical flow, which are used as blur kernels to deblur video frames. Based on the model in Eq. (3), we separately estimate the rotational and translational components of the blur kernels and combine them.

Rotational Component Given a blurred video frame I captured over the exposure interval (t_s, t_e), we compute the warped pixel positions over $N{+}1$ uniformly spaced time samples $\{t_i\}_{i=0}^N$. Using the gyro sensor, we obtain angular velocity measurements $\{\boldsymbol{\omega}_k\}_{k=0}^M$ and corresponding timestamps $\{\tilde{t}_k\}_{k=0}^M$ over the exposure interval (t_s, t_e), where $M+1$ denotes the number of measurements within the interval. The cumulative rotation from t_s to t_i is computed as:

$$\mathbf{R}_i = \prod_{k:\,\tilde{t}_k \le t_i} \exp\big([\boldsymbol{\omega}_k]_\times \Delta\tilde{t}_k\big) \tag{4}$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix associated with an angular velocity vector. $\Delta\tilde{t}_k$ is the temporal gap between velocity measurements. For each pixel $\mathbf{p}$ in I, we substitute Eq. (4) into Eq. (3) while setting $\boldsymbol{\tau} = \mathbf{0}$, and compute the warped position $\mathbf{p}'_i$ at time t_i. This yields the pixel-wise trajectories induced by rotational camera motion:

$$\mathbf{k}^{\text{rot}} = \big\{\mathbf{d}_i^{\text{rot}}\big\}_{i=0}^N, \quad \mathbf{d}_i^{\text{rot}} = \mathbf{p}'_i - \mathbf{p} \tag{5}$$

(a) $\mathbf{k}^{\text{rot}}$ (b) $\mathbf{k}^{\text{tran}}$ (c) $\mathbf{k}^{\text{rot}} + \mathbf{k}^{\text{tran}}$

Figure 1. Visualization of rotational and translational components ($\mathbf{k}^{\text{rot}}$, $\mathbf{k}^{\text{tran}}$) of blur kernels, and their combined blur kernel.

where $\mathbf{d}_i^{\text{rot}}$ denotes the pixel displacement at the i-th sampling point, and $\mathbf{k}^{\text{rot}}$ represents the corresponding trajectory formed by N sampled displacements.

Translational Component We estimate the translational motion vector $\boldsymbol{\tau}$ using optical flows. For an input frame I, we compute an optical flow map from I to its next frame I'. For each pixel $\mathbf{p}$ in I, the optical flow vector $\mathbf{f}$ points to its corresponding pixel in I'.

However, this optical flow encodes displacement caused by both rotational and translational motions. To isolate the translational component, we first compute the rotation matrix $\mathbf{R}$ between I and I' by accumulating the rotational matrices, similar to Eq. (4). We then subtract the rotational motion from the optical flow vector, yielding an estimate of the translational motion vector:

$$\boldsymbol{\tau} = \frac{t_e - t_s}{\delta} \left\{ \mathbf{p}'_f - \pi\big(\mathbf{C}\mathbf{R}\mathbf{C}^{-1}\mathbf{p}_H\big) \right\} \tag{6}$$

where δ is the temporal interval between the temporal centers of I and I' (*e.g.*, $1/30$ sec. for 30 FPS videos) and $\mathbf{p}'_f = \mathbf{p} + \mathbf{f}$ is the warped pixel position of $\mathbf{p}$ in I' found by the flow vector $\mathbf{f}$. The estimated vector is then distributed across the sampling points $\{t_i\}_{i=0}^{N}$, respectively, yielding

$$\mathbf{k}^{\text{tran}} = \left\{ \mathbf{d}_i^{\text{tran}} \right\}_{i=0}^{N}, \quad \mathbf{d}_i^{\text{tran}} = \frac{t_i - t_s}{t_e - t_s} \boldsymbol{\tau} \tag{7}$$

where $\mathbf{d}_i^{\text{tran}}$ and $\mathbf{k}^{\text{tran}}$ denote a pixel displacement and a trajectory computed from τ, respectively. Finally, the per-pixel blur kernel can be obtained by combining both components:

$$\mathbf{k} = \mathbf{k}^{\text{rot}} + \mathbf{k}^{\text{tran}} \tag{8}$$

which captures the complete camera motion trajectory during the exposure interval.

For brevity, we assume that the pixel-wise trajectories start at the beginning of the exposure, t_s. However, most deblurring methods consider the GT sharp frame to correspond to the temporal center of the exposure. To account for this, we compute both the rotational and translational components of the pixel-wise trajectories using the temporal center $t_c = (t_s + t_e)/2$ as the starting point instead of t_s. We also interpret the optical flow vector $\mathbf{f}$ as the displacement between the temporal centers of frames. More details are provided in the supplementary material.

In our implementation, we estimate optical flows from blurred videos, since sharp frames are not available prior to deblurring. The estimated optical flows may contain errors, which degrade the accuracy of per-pixel blur kernels. To enhance robustness, we compute consistency masks from the optical flows and mask unreliable regions where inconsistencies are detected. For the masked regions, we discard $\mathbf{k}^{\text{tran}}$ and instead use those from nearest neighbors.

Fig. 1 presents the estimated rotational and translational components, along with the resulting blur kernels obtained by their combination. As shown, either component alone cannot accurately represent the motion in the blurred frame, whereas their combination produces blur kernels that closely match the actual motion blur, demonstrating the effectiveness of our decomposed motion model.

5. GyroDVD Network

We introduce a video deblurring network for exploiting blur kernels constructed from rotational and translational motion. Previous gyro-based methods [17, 60] have shown that deformable convolution [71] is effective in exploiting blur kernels. However, its high computational cost makes it impractical to apply across all stages of video processing. To balance accuracy and efficiency, we adopt a hybrid design: deformable convolution is used only in the image encoder, while the video decoder avoids it to reduce computation.

Our overall architecture follows the widely adopted encoder–decoder paradigm for video restoration, where a per-frame image encoder is followed by a temporal video decoder. In particular, our decoder builds upon ShiftNet [25], which uses shift operations to propagate features across frames. We extend this by integrating blur kernel guidance into both the encoder and decoder, enabling motion-aware deblurring with small additional computational overhead.

Fig. 2 presents an overview of our video deblurring network, which consists of a blur kernel encoding module, an image encoder, and a video decoder. Given input video frames $\{I_j\}$ and gyro data, where j is a frame index, our approach first computes the rotational and translational components $\{(\mathbf{k}_j^{\text{rot}}, \mathbf{k}_j^{\text{tran}})\}$ of blur kernels, as described in Sec. 4. Then, for each I_j, the blur kernel encoding module fuses $\mathbf{k}_j^{\text{rot}}$ and $\mathbf{k}_j^{\text{tran}}$ in the feature space and produces blur kernel features $\mathbf{K}_j$. The image encoder takes I_j and $\mathbf{K}_j$ and produces encoded features $\mathbf{F}_j$ using deformable convolution.

Finally, the video decoder aggregates temporal information from image features $\{\mathbf{F}_j\}$, guided by blur kernel features $\{\mathbf{K}_j\}$, and produces deblurred video frames. To exploit blur kernels in the decoder, we introduce a learnable scheme that adaptively predicts spatial shift patterns from $\mathbf{K}_j$, enhancing temporal propagation. In the following, we describe each component of our network. Additional architectural details are provided in the supplementary material.

Blur Kernel Encoding The rotational and translational components of blur kernels, $\mathbf{k}_j^{\text{rot}}$ and $\mathbf{k}_j^{\text{tran}}$, are estimated from gyro data and optical flow, respectively. Because these

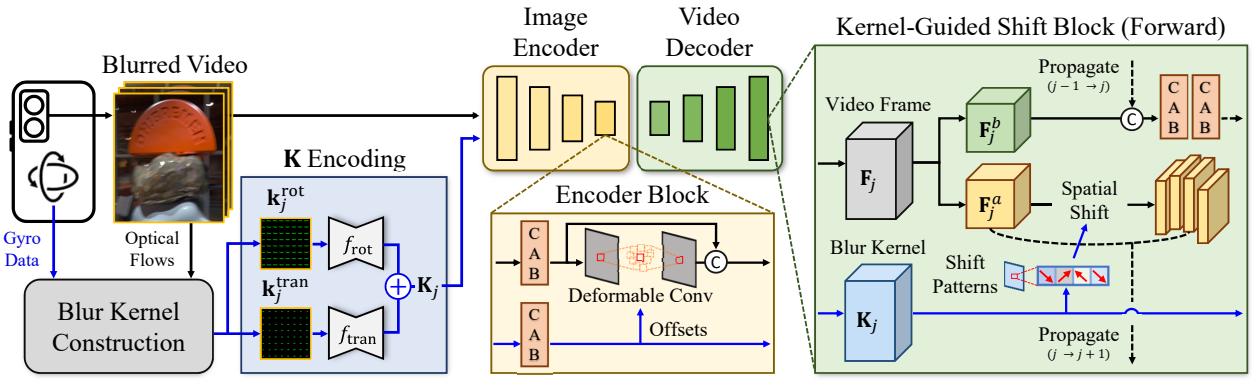


Figure 2. Overview of GyroDVD. GyroDVD constructs pixel-wise rotational and translational motion components ($\mathbf{k}_j^{\text{rot}}, \mathbf{k}_j^{\text{tran}}$) and encodes them into blur kernel features $\mathbf{K}_j$. The image encoder processes each frame using deformable convolution, whose offsets are estimated from $\mathbf{K}_j$, while the video decoder propagates features across neighboring frames using shift patterns predicted from $\mathbf{K}_j$.

two have different error characteristics, naïvely fusing them in the pixel domain via Eq. (8) can degrade kernel quality.

To address this, we encode $\mathbf{k}_j^{\text{rot}}$ and $\mathbf{k}_j^{\text{tran}}$ using separate encoding layers and integrate them in the feature space, allowing the network to learn modality-specific representations while preserving complementary motion cues. Formally, we define the fused blur kernel features as:

$$\mathbf{K}_j = f_{\text{rot}}(\mathbf{k}_j^{\text{rot}}) + f_{\text{tran}}(\mathbf{k}_j^{\text{tran}}) \tag{9}$$

where $f_{\text{rot}}(\cdot)$ and $f_{\text{tran}}(\cdot)$ are encoding layers for $\mathbf{k}^{\text{rot}}$ and $\mathbf{k}^{\text{tran}}$, respectively.

Image Encoder The image encoder first encodes an input video frame I_j to extract initial image features. It then removes blur from the image features using a modulated deformable convolution layer [71], following previous work [17, 60]. The offsets and masks of deformable convolution layers are computed from the blur kernel features $\mathbf{K}_j$. Finally, the encoder produces deblurred image features $\mathbf{F}_j$.

Video Decoder Following ShiftNet [25], our video decoder is composed of ShiftBlocks, which use temporal and spatial shift operations to efficiently propagate features across frames. Each block divides the feature map of a frame into two groups: one half is temporally propagated to neighboring frames, while the other half remains local. The propagated features are then spatially shifted using a fixed pattern to enhance temporal aggregation. Although effective for video deblurring, ShiftBlock is not designed to exploit blur kernel information.

To address this, we propose the KGS-Block (**K**ernel-**G**uided **S**hift **Block**), which replaces the fixed spatial shift pattern with a learnable, kernel-adaptive strategy. Specifically, given the deblurred features $\mathbf{F}_j$ of frame j, we split them equally along the channel dimension into two parts: $\mathbf{F}_j^a$ and $\mathbf{F}_j^b$. We further divide $\mathbf{F}_j^a$ into L channel groups $\{\mathbf{F}_{j,l}^a\}_{l=1}^L$, and spatially shift each group as:

$$\mathbf{F}_{j,l}^{\text{shift}} = \text{Warp}\left(\mathbf{F}_{j,l}^a, [\mathbf{s}_{j,l}^x, \mathbf{s}_{j,l}^y]\right) \tag{10}$$

where the shift patterns $[\mathbf{s}_{j,l}^x, \mathbf{s}_{j,l}^y]$ are predicted from the blur kernel features $\mathbf{K}_j$ via two convolutional layers. The shifted groups are concatenated to form $\mathbf{F}_j^{\text{shift}}$, and propagated to the next frame. Finally, the output is computed by fusing the propagated features from the previous frame, $\mathbf{F}_{j-1}^{\text{shift}}$, and the features from the current frame, $\mathbf{F}_j^{\text{b}}$:

$$\mathbf{F}_j^{\text{dec}} = \text{CABs}(\text{Cat}(\mathbf{F}_j^b, \mathbf{F}_{j-1}^a, \mathbf{F}_{j-1}^{\text{shift}})) \tag{11}$$

where CABs denotes two channel-attention blocks [25].

Following ShiftNet, we adopt a bi-directional decoding strategy by stacking forward and backward KGS-Blocks alternately. In forward blocks, features are propagated from frame j to $j+1$, while in backward blocks, they are propagated from j to $j-1$. The final decoded features are passed to a reconstruction module $f_{\text{rec}}(\cdot)$ to generate the deblurred output. Note that KGS-Block introduces only a small number of additional parameters for predicting shift patterns, resulting in low computational overhead while effectively leveraging the blur kernel information.

6. GyroVD Dataset

We present the GyroVD dataset for training and evaluating GyroDVD. The dataset consists of two datasets: GyroVD-Syn and GyroVD-Real. GyroVD-Syn contains synthetic blurred videos generated from high-speed videos, while GyroVD-Real provides real blurred videos.

To generate GyroVD-Syn, we developed an Android app that simultaneously records high-speed videos and gyro data. The app captures high-speed videos (240 FPS) at a resolution of 1080×1920, while the gyro data are captured at 400 FPS. In addition, the start and end exposure timestamps of each frame are recorded, allowing temporal alignment between gyro data and video frames. Blurred videos are then synthesized by averaging consecutive high-speed frames, and the temporally centered frame is used as the GT sharp image. The number of frames used for averaging is randomly selected from $\{7, 9, 11, 13, 15\}$. Following

Table 1. Statistics of the video and gyro-based deblurring datasets. The triangle (△) indicates that GyroBlur-Syn [60] includes only constantly moving objects, which remain unrealistic.

	# Videos	# Images	Gyro	Dynamic Scenes
GoPro [36]	33	3,214		✓
DVD [50]	71	6,708		✓
REDS [37]	300	30,000		✓
BSD [68, 69]	300	33,000		✓
EggNet [17]	×	4,238	✓	
IMU-Image [41]	×	6,624	✓	
GyroBlur-Syn [60]	×	15,240	✓	△
GyroVD-Syn	632	63,200	✓	✓
GyroVD-Real	100	10,000	✓	✓

Rim *et al*. [42], we further adopt a realistic blur synthesis pipeline to enhance the robustness of the synthetic dataset for real-world images. In particular, we apply frame interpolation [61] prior to averaging, and employ realistic saturation and noise synthesis [42].

We collected 632 videos across diverse scenes using a Google Pixel 9 Pro XL as our camera system and synthesized blurred videos following the pipeline described above. Each synthetic blurred video contains 100 frames with corresponding gyro data. For training and evaluation, we split GyroVD-Syn into training, validation, and test sets, consisting of 505, 50, and 77 videos, respectively. Furthermore, we divide the test set of GyroVD-Syn into three subsets (*i.e*., small, medium, and large blur) based on the average blur magnitude. The magnitude is computed by estimating trajectories of optical flows between sharp frames before averaging and measuring their maximum displacement.

Tab. 1 shows the statistics of GyroVD-Syn and other deblurring datasets. Notably, our dataset is the largest and most realistic gyro-based dataset, providing various camera motions in dynamic scenes with moving objects.

The GyroVD-Real consists of real-world blurred videos captured in various night and indoor scenes. To construct the dataset, we collected 100 real-world videos and corresponding gyro data using a Google Pixel 9 Pro XL and a Samsung Galaxy S23. As GT images are not available in this dataset, GyroVD-Real is used for evaluation using no-reference metrics [6, 32, 33].

7. Experiments

Implementation Details We use the AdamW optimizer [22] and ℓ_1 loss for training. The initial learning rate is set to $4\mathrm{e}{-4}$ and decayed to $1\mathrm{e}{-7}$ following the cosine annealing schedule [30]. GyroDVD is trained for 600K iterations with a batch size of 4, where each batch contains 13 consecutive frames. The training patch size is 256×256, and standard horizontal and vertical flips are applied for data augmentation. The number of time samples for constructing blur kernels is set to $N = 8$, and the number of channel groups in the video decoder to $L = 8$. We employ RAFT-small [54] to estimate inter-frame optical flows.

7.1. Results on the GyroVD Dataset

We compare variants of GyroDVD with different channel widths (*e.g*., GyroDVD-64) against gyro-based image deblurring methods [17, 35, 60] and video deblurring methods [5, 25, 28, 29, 39, 40, 56, 64]. Since GyroDVD is the first learning-based method for gyro-based video deblurring, there are no directly comparable methods under the same setting. To provide a reasonable comparison, we additionally include a modified version of ShiftNet [25], denoted as 'ShiftNet with $\mathbf{k}^{\mathrm{rot}}$'. This variant extracts features from video frames and blur kernels using only the rotational motion component $\mathbf{k}^{\mathrm{rot}}$. The extracted features are concatenated and fed into the subsequent ShiftNet for video deblurring. All methods are trained on GyroVD-Syn for a fair comparison.

Tab. 2 presents a quantitative comparison of GyroDVD and other methods on GyroVD-Syn and GyroVD-Real. Interestingly, gyro-based image deblurring methods achieve inferior performance compared to video deblurring methods. This is because single-image deblurring remains a highly ill-posed problem, even with motion information. Video deblurring methods exploit adjacent frames, which makes the problem more tractable, and results in superior performance over gyro-based image deblurring. Among video deblurring methods, the large version of ShiftNet (*i.e*., ShiftNet+) achieves the best performance, showing the effectiveness of feature propagation with spatial shift patterns.

An extension of ShiftNet for gyro-based video deblurring (*i.e*., ShiftNet with $\mathbf{k}^{\mathrm{rot}}$) achieves better performance than ShiftNet, benefiting from the rotational motion information of gyro data. However, GyroDVD achieves much higher performance. This result shows that the concatenation-based extension cannot fully exploit the blur kernels, and that rotational motion alone is insufficient. In contrast, GyroDVD considers both rotational and translational motions to construct blur kernels and adopts a sophisticated architecture to exploit them, achieving the best performance. Furthermore, Tab. 2 shows that the performance of the compared methods degrades significantly on the large-blur subset, whereas GyroDVD remains robust and achieves superior performance. Notably, even the smaller version, GyroDVD-48, still outperforms all compared methods on the large-blur subset.

Fig. 3 shows qualitative results on GyroVD-Syn and GyroVD-Real, demonstrating the effectiveness of GyroDVD. In particular, GyroDVD successfully restores sharp texts under severe blur (first and third rows) and in the saturated region (fourth row). This demonstrates the advantage of exploiting blur kernels in such challenging cases where blur characteristics are difficult to estimate from

Table 2. Quantitative comparison on GyroVD-Syn and GyroVD-Real. For GyroVD-Real, we use no-reference metrics (*i.e.*, NIQE [33], BRISQUE [32], and TopIQ [6] trained on KonIQ-10K [15]) for evaluation. The inference times are measured on 3×48×512×512 videos and averaged per frame. The inference time of GyroDVD includes the time required for blur kernel construction (*i.e.*, 0.013 sec.).

	GyroVD-Syn (PSNR ↑ / SSIM ↑)				**GyroVD-Real**	Param / Time
	Small	Medium	Large	Avg	(NIQE ↓ / BRISQUE ↓ / TopIQ ↑)	(M) / (Sec.)
DeepGyro [35]	32.30 / 0.8588	30.12 / 0.8006	27.98 / 0.7414	30.13 / 0.8003	4.44 / 39.85 / 0.3050	31.03 / 0.008
EggNet [17]	32.52 / 0.8632	30.17 / 0.8043	28.26 / 0.7523	30.32 / 0.8066	4.53 / 40.06 / 0.3046	6.34 / 0.022
GyroDeblur [60]	34.22 / 0.8902	32.31 / 0.8468	30.48 / 0.8003	32.34 / 0.8458	3.72 / 33.59 / 0.3866	16.31 / 0.024
BasicVSR++ [5]	35.70 / 0.9161	33.71 / 0.8804	30.25 / 0.8103	33.22 / 0.8689	3.51 / 31.98 / 0.3966	9.76 / 0.018
EDVR [56]	35.17 / 0.9065	33.44 / 0.8712	31.31 / 0.8230	33.31 / 0.8669	3.59 / 30.81 / 0.4022	23.60 / 0.054
STCT [64]	35.57 / 0.9160	33.54 / 0.8799	30.99 / 0.8227	33.37 / 0.8729	3.94 / 37.00 / 0.3724	8.02 / 0.099
DSTNet [39]	35.77 / 0.9188	34.07 / 0.8878	31.75 / 0.8363	33.86 / 0.8810	3.54 / 31.59 / 0.3945	7.45 / 0.017
ShiftNet [25]	36.16 / 0.9209	34.50 / 0.8907	32.46 / 0.8479	34.37 / 0.8865	3.85 / 39.64 / 0.4342	4.70 / 0.074
VRT [29]	36.62 / 0.9278	35.00 / 0.9011	32.70 / 0.8555	34.77 / 0.8948	3.68 / 37.49 / 0.4432	18.32 / 1.064
RVRT [28]	36.52 / 0.9264	35.10 / 0.9021	32.85 / 0.8585	34.82 / 0.8957	3.47 / 29.42 / 0.4554	13.57 / 0.087
DSTNet+L [40]	36.64 / 0.9295	35.13 / 0.9050	32.95 / 0.8626	34.90 / 0.8990	3.47 / 34.04 / 0.4461	14.08 / 0.034
ShiftNet+ [25]	36.97 / 0.9306	35.47 / 0.9065	33.51 / 0.8698	35.31 / 0.9023	3.55 / 37.37 / 0.4741	12.99 / 0.164
ShiftNet [25] with $\mathbf{k}^{\text{rot}}$	36.21 / 0.9217	34.66 / 0.8932	32.77 / 0.8544	34.55 / 0.8898	3.61 / 36.79 / 0.4507	4.72 / 0.076
GyroDVD-48	36.60 / 0.9271	35.24 / 0.9036	33.54 / 0.8712	35.12 / 0.9006	3.72 / 39.18 / 0.4564	3.22 / 0.088
GyroDVD-64	36.81 / 0.9296	35.51 / 0.9076	33.84 / 0.8769	35.39 / 0.9047	3.74 / 39.98 / 0.4609	5.04 / 0.106
GyroDVD-96	37.17 / 0.9332	35.87 / 0.9124	34.23 / 0.8841	35.76 / 0.9099	3.26 / 30.89 / 0.4886	10.13 / 0.151
GyroDVD-128	37.35 / 0.9343	36.06 / 0.9142	34.38 / 0.8855	35.93 / 0.9113	3.31 / 29.48 / 0.4889	17.15 / 0.202

Figure 3. Qualitative results on GyroVD-Syn (first and second rows) and GyroVD-Real (third and fourth rows).

video frames alone. In addition, the figure shows that GyroDVD handles dynamic scenes (second row), even though our blur kernel construction is derived from the camera motion model. We observe that the translational component $\mathbf{k}^{\text{tran}}$, as it is computed from optical flows, still provides useful cues for object motion in practice. More discussions and analyses on object motion are provided in the supplementary material. For further qualitative results, please refer to the supplementary video and material.

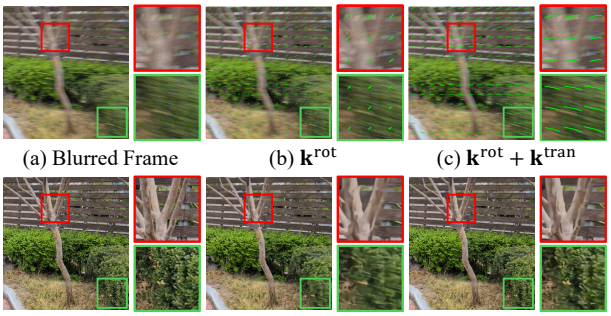
7.2. Ablation Studies

We evaluate the impact of various components of GyroDVD. For ablation studies, variants of GyroDVD-64 are trained for 150K iterations with a batch size of 2.

Impact of Blur Kernels To validate the effectiveness of the proposed blur kernels, we replace them with alternative

Table 3. Comparison of different sources for estimating the offsets of the deformable convolution and the learnable shift patterns.

Methods	PSNR ↑ / SSIM ↑	Param / Time
Baseline	33.51 / 0.8736	3.89 / 0.070
w/ video frame	33.51 / 0.8732	4.89 / 0.091
w/ optical flows	34.11 / 0.8846	4.89 / 0.102
w/ $\mathbf{k}^{\text{rot}}$ only	34.30 / 0.8869	4.89 / 0.092
w/ $\mathbf{k}^{\text{rot}}$, $\mathbf{k}^{\text{tran}}$	34.65 / 0.8928	5.04 / 0.106
w/ $\mathbf{k}^{\text{rot}}$, $\mathbf{k}^{\text{tran}}$ from GT	34.78 / 0.8964	5.04 / 0.106



(a) Blurred Frame (b) $\mathbf{k}^{\text{rot}}$ (c) $\mathbf{k}^{\text{rot}} + \mathbf{k}^{\text{tran}}$

(d) Ground Truth (e) Deblurred Frame w/ $\mathbf{k}^{\text{rot}}$ only (f) Deblurred Frame w/ $\mathbf{k}^{\text{rot}}$, $\mathbf{k}^{\text{tran}}$

Figure 4. Visualization of blur kernels and deblurred results.

inputs: the video frame itself, bi-directional optical flows, and rotation-only blur kernels using $\mathbf{k}^{\text{rot}}$. For these variants, the offsets of deformable convolutions and shift patterns are estimated from the respective inputs. As shown in Tab. 3, the rotation-only variant performs better than the other variants. This result again shows that rotational motion from gyro data provides valuable information compared to the others. However, our final model integrating both rotational and translational motion achieves even higher performance. Fig. 4 visualizes the blur kernels and deblurred results of the rotation-only model and our final model. The figure shows that rotational motion alone is often misaligned with the actual motion, leading to loss of sharp details. This emphasizes that translational motion has a significant impact and must be considered in deblurring.

We also analyze the impact of computing $\mathbf{k}^{\text{tran}}$ using optical flows estimated from blurred videos. Tab. 3 shows that GyroDVD achieves comparable results to the variant using $\mathbf{k}^{\text{tran}}$ computed from GT sharp frames, demonstrating the robustness of the proposed network.

Network Architecture We evaluate several strategies for exploiting blur kernels in both the image encoder and the video decoder, including variants without blur kernels, with simple concatenation, kernel-guided convolution (KGC) [20], and the kernel attention module (KAM) [12]. For the image encoder, each strategy replaces deformable convolution, while for the video decoder, the features are processed by each strategy and fed into subsequent ShiftBlocks. Tab. 4 shows that the proposed architecture, which

Table 4. Ablation study on the effects of blur kernels $\mathbf{K}$ in the image encoder and the video decoder.

Encoder	Decoder	PSNR ↑ / SSIM ↑	Param / Time
w/o $\mathbf{K}$	Shift [25]	33.51 / 0.8736	3.89 / 0.070
Cat $\mathbf{K}$	Shift [25]	33.83 / 0.8788	4.55 / 0.092
KGC [20]	Shift [25]	33.86 / 0.8796	4.44 / 0.092
KAM [12]	Shift [25]	33.87 / 0.8795	4.79 / 0.093
Def-Conv.	Shift [25]	33.95 / 0.8800	4.60 / 0.096
Def-Conv.	Shift [25] + Cat $\mathbf{K}$	33.42 / 0.8733	5.07 / 0.103
Def-Conv.	Shift [25] + KGC [20]	33.62 / 0.8763	5.37 / 0.110
Def-Conv.	Shift [25] + KAM [12]	34.15 / 0.8840	5.57 / 0.112
Def-Conv.	KGS-Block	34.65 / 0.8928	5.04 / 0.106

employs deformable convolutions in the image encoder and KGS-Blocks in the video decoder, achieves the best performance among all compared variants. Interestingly, naïvely exploiting blur kernels in the video decoder ("Shift + Cat $\mathbf{K}$" and "Shift + KGC") can even degrade performance. This highlights the need for a carefully designed kernel-based video network and the necessity of KGS-Block.

Additionally, we analyze the effect of the proposed blur kernel encoding in the feature domain. We evaluate a variant that fuses the blur kernels in the pixel domain using Eq. (8). This variant achieves 34.55 dB in terms of PSNR, which is 0.1 dB lower than the proposed method. This result indicates that separately encoding the blur kernels and fusing them in the feature domain enables the network to preserve complementary information and better handle distinct errors in $\mathbf{k}^{\text{rot}}$ and $\mathbf{k}^{\text{tran}}$.

8. Conclusion

In this paper, we present GyroDVD, the first learning-based method for gyro-based video deblurring. We introduce a novel motion model that decomposes pixel-wise motion into rotational and translational components. Based on this model, we propose a blur kernel construction scheme considering both rotational and translational motion. A video deblurring network exploits the constructed blur kernels together with video frames for deblurring. Furthermore, we introduce a large-scale and realistic video dataset for training and evaluation of gyro-based deblurring methods. Experimental results demonstrate that the proposed method significantly outperforms previous approaches and validate the effectiveness of gyro-based video deblurring.

Limitations GyroDVD computes the translational component using optical flows estimated from blurred frames, which may contain errors. We mitigate this using separate kernel encoding layers and by masking unreliable regions, but the translational component can still degrade when optical flow estimation fails severely. Another limitation is the inference time, which remains slow for on-device applications. Addressing these limitations would be an interesting direction for future work.

Acknowledgments This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH)) and by the IITP–ITRC (Information Technology Research Center) grant funded by the Korea government (Ministry of Science and ICT) (No. IITP-2026-RS-2024-00437866). This research was also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2022R1A6A1A03052954).

References

[1] Amit Agrawal, Yi Xu, and Ramesh Raskar. Invertible motion blur in video. ACM SIGGRAPH, 2009. 1, 2

[2] Ahmet Arslan, Gokhan Koray Gultekin, and Afsar Saranli. Imu-aided adaptive mesh-grid based video motion deblurring. *PeerJ Computer Science*, 10:e2540, 2024. 1, 2

[3] M. Ben-Ezra and S.K. Nayar. Motion deblurring using hybrid imaging. In *Proc. of CVPR*, 2003. 1

[4] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring using multiple images. *Journal of computational physics*, 228(14):5057–5071, 2009. 1, 2

[5] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *Proc. of CVPR*, 2022. 2, 6, 7

[6] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE Trans. Image Process.*, 2024. 6, 7

[7] Hoonhee Cho, Yuhwan Jeong, Taewoo Kim, and Kuk-Jin Yoon. Non-coaxial event-guided motion deblurring with spatial alignment. In *Proc. of ICCV*, 2023. 1

[8] Sunghyun Cho, Yasuyuki Matsushita, and Seungyong Lee. Removing non-uniform motion blur from images. In *Proc. of ICCV*, 2007. 1, 2

[9] Sunghyun Cho, Jue Wang, and Seungyong Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph.*, 31(4):1–9, 2012. 2

[10] Mauricio Delbracio and Guillermo Sapiro. Burst deblurring: Removing camera shake through fourier burst accumulation. In *Proc. of CVPR*, 2015.

[11] Mauricio Delbracio and Guillermo Sapiro. Hand-held video deblurring via efficient fourier aggregation. *IEEE Trans. Comput. Imaging*, 1(4):270–283, 2015. 2

[12] Zhenxuan Fang, Fangfang Wu, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *Proc. of CVPR*, 2023. 8

[13] Chen Haoyu, Teng Minggui, Shi Boxin, Wang YIzhou, and Huang Tiejun. Learning to deblur and generate high frame rate video with an event camera. *arXiv preprint arXiv:2003.00847*, 2020. 1

[14] Sung Hee Park and Marc Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *Proc. of CVPR*, 2014. 1, 2

[15] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Trans. Image Process.*, 2020. 7

[16] Zhe Hu, Lu Yuan, Stephen Lin, and Ming-Hsuan Yang. Image deblurring using smartphone inertial sensors. In *Proc. of CVPR*, 2016. 2

[17] Seowon Ji, Jun-Pyo Hong, Jeongmin Lee, Seung-Jin Baek, and Sung-Jea Ko. Robust single image deblurring using gyroscope sensor. *IEEE Access*, 2021. 1, 2, 4, 5, 6, 7

[18] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proc. of CVPR*, 2020. 1

[19] Neel Joshi, Sing Bing Kang, C Lawrence Zitnick, and Richard Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 2010. 2

[20] Adam Kaufman and Raanan Fattal. Deblurring using analysis-synthesis networks pair. In *Proc. of CVPR*, 2020. 8

[21] Taewoo Kim, Jeongmin Lee, Lin Wang, and Kuk-Jin Yoon. Event-guided deblurring of unknown exposure time videos. In *Proc. of ECCV*, 2022. 1

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Wei-Sheng Lai, YiChang Shih, Lun-Cheng Chu, Xiaotong Wu, Sung-Fang Tsai, Michael Krainin, Deqing Sun, and Chia-Kai Liang. Face deblurring using dual camera fusion on mobile phones. *ACM Trans. Graph.*, 2022. 1

[24] Jeongmin Lee, Seo-Won Ji, Sung-Jin Cho, Jun-Pyo Hong, and Sung-Jea Ko. Deep learning-based deblur using gyroscope data. In *Proc. ICCE-Asia*, 2020. 1, 2

[25] Dasong Li, Xiaoyu Shi, Yi Zhang, Ka Chun Cheung, Simon See, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. A simple baseline for video restoration with grouped spatial-temporal shift. In *Proc. of CVPR*, 2023. 1, 2, 4, 5, 6, 7, 8

[26] Feng Li, Jingyi Yu, and Jinxiang Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *Proc. of CVPR*, 2008. 1

[27] Yunpeng Li, Sing Bing Kang, Neel Joshi, Steve M Seitz, and Daniel P Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *Proc. of CVPR*, 2010. 1, 2

[28] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. In *Proc. of NeurIPS*, 2022. 1, 2, 6, 7

[29] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *IEEE Trans. Image Process.*, 33:2171–2182, 2024. 1, 2, 6, 7

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[31] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1150–1163, 2006. 2

[32] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.*, 2012. 6, 7

[33] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.*, 2012. 6, 7

[34] Janne Mustaniemi, Juho Kannala, Simo Särkkä, Jiri Matas, and Janne Heikkilä. Fast motion deblurring for feature detection and matching using inertial measurements. In *Proc. ICPR*, 2018. 1, 2

[35] Janne Mustaniemi, Juho Kannala, Simo Särkkä, Jiri Matas, and Janne Heikkila. Gyroscope-aided motion deblurring with deep networks. In *Proc. WACV*, 2019. 1, 2, 6, 7

[36] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proc. of CVPR*, 2017. 6

[37] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proc. of CVPRW*, 2019. 6

[38] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *Proc. of CVPR*, 2020. 1, 2

[39] Jinshan Pan, Boming Xu, Jiangxin Dong, Jianjun Ge, and Jinhui Tang. Deep discriminative spatial and temporal network for efficient video deblurring. In *Proc. of CVPR*, 2023. 2, 6, 7

[40] Jinshan Pan, Long Sun, Xu Boming, Jiangxin Dong, and Jinhui Tang. Learning efficient deep discriminative spatial and temporal networks for video deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025. 1, 2, 6, 7

[41] Wenqi Ren, Linrui Wu, Yanyang Yan, Shengyao Xu, Feng Huang, and Xiaochun Cao. Informer: Inertial-based fusion transformer for camera shake deblurring. *IEEE Trans. Image Process.*, 2024. 1, 2, 6

[42] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *Proc. of ECCV*, 2022. 6

[43] Jaesung Rim, Junyong Lee, Heemin Yang, and Sunghyun Cho. Deep hybrid camera deblurring for smartphone cameras. ACM SIGGRAPH Conference Papers, 2024. 1

[44] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *Proc. IROS*, 2018. 2

[45] Shayan Shekarforoush, Amanpreet Walia, Marcus A Brubaker, Konstantinos G Derpanis, and Alex Levinshtein. Dual-camera joint deblurring-denoising. *arXiv preprint arXiv:2309.08826*, 2023. 1

[46] Ondrej Sindelar and Filip Sroubek. Image deblurring in smartphone devices using built-in inertial measurement sensors. *Journal of Electronic Imaging*, 2013. 1, 2

[47] Ondrej Sindelar, Filip Sroubek, and Peyman Milanfar. Space-variant image deblurring on smartphones using inertial sensors. In *Proc. of CVPRW*, 2014. 1, 2

[48] Hyeongseok Son, Junyong Lee, Jonghyeop Lee, Sunghyun Cho, and Seungyong Lee. Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. *ACM Trans. Graph.*, 40(5):1–18, 2021. 1, 2

[49] Filip Sroubek and Peyman Milanfar. Robust multichannel blind deconvolution via fast alternating minimization. *IEEE Trans. Image Process.*, 21(4):1687–1700, 2011. 1, 2

[50] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proc. of CVPR*, 2017. 1, 2, 6

[51] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *Proc. of ECCV*, 2022. 1

[52] Yu-Wing Tai, Hao Du, Michael S. Brown, and Stephen Lin. Image/video deblurring using a hybrid camera. In *Proc. of CVPR*, 2008. 1

[53] Yu-Wing Tai, Hao Du, Michael S. Brown, and Stephen Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010. 1

[54] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. of ECCV*, 2020. 6

[55] Nisha Varghese, AN Rajagopalan, and Zahir Ahmed Ansari. Real-time large-motion deblurring for gimbal-based imaging systems. *IEEE JSTSP*, 2024. 1, 2

[56] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proc. of CVPRW*, 2019. 2, 6, 7

[57] Zeyu Xiao and Xinchao Wang. Asymmetric dual-lens video deblurring. In *Proc. of NeurIPS*, 2025. 1

[58] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. Motion deblurring with real events. In *Proc. of ICCV*, 2021. 1

[59] Honglei Xu, Zhilu Zhang, Junjie Fan, Xiaohe Wu, and Wangmeng Zuo. Selfhvd: Self-supervised handheld video deblurring. In *Proc. of CVPR*, 2026. 1

[60] Heemin Yang, Jaesung Rim, Seungyong Lee, Seung-Hwan Baek, and Sunghyun Cho. Gyro-based neural single image deblurring. In *Proc. of CVPR*, 2025. 1, 2, 4, 5, 6, 7

[61] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proc. of CVPR*, 2023. 6

[62] Huicong Zhang, Haozhe Xie, and Hongxun Yao. Blur-aware spatio-temporal sparse transformer for video deblurring. In *Proc. of CVPR*, 2024. 1, 2

[63] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *IEEE Trans. Image Process.*, 28(1):291–301, 2018. 2

[64] Liyan Zhang, Boming Xu, Zhongbao Yang, and Jinshan Pan. Deblurring videos using spatial-temporal contextual transformer with feature propagation. *IEEE Trans. Image Process.*, 2024. 1, 2, 6, 7

[65] Shuang Zhang, Ada Zhen, and Robert L Stevenson. A dataset for deep image deblurring aided by inertial sensor data. In *Proc. IS&T Symposium on Electronic Imaging*, 2020. 2

[66] Shuang Zhang, Ada Zhen, and Robert L Stevenson. Deblur-expandnet: image motion deblurring network aided by inertial sensor data. *Signal, Image and Video Processing*, 16(5): 1169–1176, 2022. 2

[67] Xiang Zhang, Lei Yu, Wen Yang, Jianzhuang Liu, and Gui-Song Xia. Generalizing event-based motion deblurring in real-world scenarios. In *Proc. of ICCV*, 2023. 1

[68] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *Proc. of ECCV*, 2020. 1, 2, 6

[69] Zhihang Zhong, Ye Gao, Yinqiang Zheng, Bo Zheng, and Imari Sato. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *Int. J. Comput. Vis.*, 131(1):284–301, 2023. 2, 6

[70] Xiang Zhu, Filip Šroubek, and Peyman Milanfar. Deconvolving psfs for a better motion deblurring using multiple images. In *Proc. of ECCV*, 2012. 1, 2

[71] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proc. of CVPR*, 2019. 4, 5