Exposure Trajectory Recovery from Motion Blur

Youjian Zhang, Chaoyue Wang, Stephen J. Maybank, *Fellow, IEEE,* and Dacheng Tao, *Fellow, IEEE*

Abstract—Motion blur in dynamic scenes is an important yet challenging research topic. Recently, deep learning methods have achieved impressive performance for dynamic scene deblurring. However, the motion information contained in a blurry image has yet to be fully explored and accurately formulated because: (i) the ground truth of dynamic motion is difficult to obtain; (ii) the temporal ordering is destroyed during the exposure; and (iii) the motion estimation from a blurry image is highly ill-posed. By revisiting the principle of camera exposure, motion blur can be described by the relative motions of sharp content with respect to each exposed position. In this paper, we define exposure trajectories, which represent the motion information contained in a blurry image and explain the causes of motion blur. A novel motion offset estimation framework is proposed to model pixel-wise displacements of the latent sharp image at multiple timepoints. Under mild constraints, our method can recover dense, (non-)linear exposure trajectories, which significantly reduce temporal disorder and ill-posed problems. Finally, experiments demonstrate that the recovered exposure trajectories not only capture accurate and interpretable motion information from a blurry image, but also benefit motion-aware image deblurring and warping-based video extraction tasks. Codes are available on https://github.com/yjzhang96/Motion-ETR.

Index Terms—Motion blur, Exposure trajectory recovery, Motion-aware image deblurring, Video extraction from a single blurry image.

1 Introduction

Motion blur in the dynamic scene caused by camera shake, object motion, or depth variation is one of the commonest image degradations. Estimating motion information and restoring sharp content in dynamic blurry images would benefit many real-world applications including segmentation, detection, and recognition. Benefiting from the powerful fitting ability of deep convolutional neural networks (CNNs), deep learning-based deblurring methods [1], [2], [3], [4] have achieved impressive performance for dynamic motion blur removal. Nevertheless, exploring motion information in a blurry image remains an academic and commercial challenge.

Most conventional blur estimation/removal methods are based on blur kernel optimization [5], [6], [7], [8], [9], [10], [11], which assumes that a blurry area can be represented as a weighted sum of its latent sharp surrounding content. A blur kernel is a weighted matrix that performs convolution on a sharp image patch to synthesize a blurry pixel. Conversely, blur kernel estimation is cast as an energy minimization problem that aims to recover both the blur kernels and the latent sharp image from a blurry image. Such optimizations are highly ill-posed, so most conventional methods are restricted by assumptions of motion types and predefined image priors. For example, [7], [12], [13], [14] only handle blur caused by camera rotations, in-plane translations, or forward out-of-plane translations. For more complex dynamic motion blur, identifying a suitably informative and general prior is extremely difficult.

Accompanying the development of deep neural networks, learning-based methods [15], [16] have been proposed to estimate blur kernels directly from blurry images. Compared to optimization-based methods, learning-based methods utilize predefined kernels to synthesize blurry data and then train an estimation network in a supervised manner. A well-trained estimation network is usually more effective and efficient at modeling object motion blur. However, due to the inherent limitations of blurry data synthesis, existing predefined blur kernels only cover limited motion types such as 2D vectors (*i.e.,* linear motions), as in [15]. As a consequence, these methods may not be as effective for complex real-world dynamic scene blur.

Taking advantage of advanced photographic equipment, dynamic scene datasets [1], [17] containing high frame-rate videos have been compiled to further understand dynamic motion blur. A real-world blurry image can be regarded as an accumulation of multiple "instant" frames, where a sequence of instant frames implicitly records blur (motion) information during an exposure period. Some methods [18], [19] are trained to directly recover these high frame-rate sharp frames (*video extraction*) without explicitly depicting dynamic motions. Moreover, in some video deblurring studies [20], [21], optical flow is estimated between adjacent frames as another motion representation. However, since the optical flow between two frames is inherently linear and multiple frames may be misaligned, the estimated optical flow cannot perfectly match the dynamic motion contained in a single blurry image.

Although recent years the end-to-end training models can directly recovery the sharp content [2], [3], [22], video sequence [18], [19] or 3D scene [23] from a blurry image, we argue that the motion information carried by a blurry image is important and has not been fully explored due to the aforementioned limitations, *e.g.* synthetic ground truths, predefined priors, or temporal disorder. In this paper, our main target is to better estimate the motion information of a single dynamic blurry image without using any motion

- *Y. Zhang, C. Wang and D. Tao are with the School of Computer Science, Faculty of Engineering, University of Sydney, Darlington, NSW, Australia.*
  E-mail: yzha0535@uni.sydney.edu.au, chaoyue.wang@sydney.edu.au, dacheng.tao@sydney.edu.au
- *S. Maybank is with the Department of Computer Science and Information Systems, Birkbeck College, Malet Street WC1E 7HX, London, UK.*
  E-mail: sjmaybank@dcs.bbk.ac.uk

Manuscript received Month Day, Year; revised Month Day, Year.
(Corresponding authors: Chaoyue Wang and Dacheng Tao.)

ground truth.

According to camera exposure principles, dynamic motion blur is caused by the relative motions of sharp content with respect to each exposed position. Inspired by this principle, we define the trajectories of these relative motions as *exposure trajectories*. Compared to the convolutional blur kernel, the exposure trajectory can describe a specific physical movement in the temporal order. Through modeling pixel-wise displacements of the latent sharp image at continuous timepoints, exposure trajectories break the linear assumption in existing methods. In addition, we propose a novel motion offsets estimation framework to recover exposure trajectories of a single blurry image. Since the proposed differentiable motion offset module can easily be plug into CNNs for backpropagation, our motion/trajectory estimation framework is trained in a reblurring cycle that only need paired blurry/sharp image pairs. Moreover, to overcome the ill-posed nature of blur estimation and to model complex non-linear motions, we further apply a variety of constraints to ensure that the learned motion offsets form different types of trajectories, *e.g.,* linear, bidirectional linear, or quadratic curves.

Besides recovering accurate and interpretable motion information from a blurry image, we successfully apply the recovered exposure trajectories to two downstream tasks. For image deblurring, we devise a motion-aware deblurring module that takes pixel-wise trajectories to modulate the shape of convolution filters. Experiments show that the proposed motion-aware module enables a more effective deconvolution operation to handle large-scale dynamic motion blur with promising results. In addition, warping-based video extraction from a single blurry image can easily be achieved using the learned exposure trajectories. Compared to existing video extraction models, our solution generates the video with more accurate motions and is capable of interpolating an arbitrary number of middle frames, *i.e.,* deriving slow-motion videos.

In summary, the contributions of this work are four-fold:

- We propose a novel framework to model exposure trajectory, which represents the motion information contained in a dynamic blurry image. Compared with existing motion representations, *e.g.,* conventional blur kernels, high-speed video frames (or estimated optical flow), and linear 2D motion vectors, our recovered exposure trajectories are more accurate and easier to interpret. Specifically, the causes of dynamic image blur are the first time modeled as dense, non-linear and continuous trajectories.
- To recover exposure trajectories from a single blurry image, we proposed a motion offset estimation framework that contains a motion offsets estimation network and a blur creation module. To our best knowledge, it is the first differentiable image (re)blurring module that enables training an end-to-end motion estimation network without supervision on ground-truth motion information.
- To address the ill-posed nature of dynamic motion/trajectory recovery, we propose multiple constraints such that the learned exposure trajectories follow certain patterns. We implement linear, bidirectional linear, and quadratic constraints, and in doing so demonstrate that our motion offsets with non-linear quadratic constraints outperform existing methods in fitting realistic dynamic motion blur.
- We present extensive experiments and analysis to demonstrate (i) our method can recover accurate and interpretable motion information from a single blurry image (Sec 5.3); (ii) the recovered exposure trajectories further benefit motion blur related tasks. By introducing recovered trajectories and motion-aware convolution, we improved the image deblurring performance over the baseline model (Sec 5.4). For extracting videos from a blurry image (Sec 5.5), though our model is only trained on blurry/sharp image pairs, it can synthesize high-quality videos with arbitrary frame rates and further delivers an impressive optical-flow field of the dynamic scene.

The rest of the paper is organized as follows. After a brief summary of the related works in Section 2, we illustrate our exposure trajectory recovery framework in Section 3. Specifically, we first explain how to model the exposure trajectory, and then introduce the training scheme for exposure trajectory estimation. In Section 4, we attempt to apply the recovered exposure trajectories to image deblurring and video extraction tasks, which justifies the advantages of estimating exposure trajectories. Experiments in Section 5 validate the accuracy of our trajectory estimation, improvements of deblurring performance, and superiority in video extraction, respectively. Limitations and failure cases are discussed in Section 6. Finally, we conclude this paper with some future directions in Section 7.

2 Related Work

Single image blur estimation and removal have been extensively studied, with many methods proposed to solve different deblurring or blur estimation problems. Here, we focus our discussion on recent motion blur studies, reviewing optimization- and learning-based methods for blur estimation and removal, respectively.

2.1 Optimization-based Methods

A blur process is conventionally modeled as a convolution operation in which blur kernels are applied to a latent sharp image to generate a blurry output. Given a blurry image, optimization-based methods aim to iteratively recover its deblurred result and the blur kernels that model blur motions. However, this problem is ill-posed, so optimization-based methods adopt predefined image priors [5], [6], [24], [25], [26], [27], [28], [29], [30] or specific camera motion types [12], [13], [14], [31] to constrain the solution space of the blur kernels. For example, Tai *et al.* [7] proposed a general projective motion model for cameras undergoing ego motion. Gupta *et al.* [12] generalized camera motion to 2D translation and in-plane rotation and modeled them as motion density functions. Whyte *et al.* [13], [32] focused on solving the non-uniform blur caused by camera shake, aiming to recover the 3D rotation of the camera during an exposure process. Zheng *et al.* [14] attempted to handle another type of motion blur in which the camera moves primarily forwards or backwards by exploring homography associated with different

3D planes. Overall, under predefined priors/assumptions, the ill-posed optimization problem becomes solvable, and these methods have achieved reasonable performance on specific blurry data. However, most of these priors assume that the underlying scene is static and that the blur is caused by camera motion rather than the movement of objects in the captured image.

However, it is difficult to identify a suitably informative and general prior for object motion within a dynamic scene. Therefore, some authors [9], [33], [34] have segmented different types of motion blur to overcome this problem. For example, Hyun *et al.* [9] proposed a novel energy function designed from the weighted sum of multiple blur data models. To handle different types of motion, their method estimated different motion blurs and their associated pixel-wise weights. Then, [33] proposed soft-segmentation for object layer estimation. By jointly estimating object segmentation and camera motion, they achieved favorable object motion blur removal performance. Although motion segmentation seems an ideal extension of optimization-based methods, it is hard to estimate an accurate segmentation due to ambiguous pixels between regions. Furthermore, even within a segmented area, existing priors can only handle a limited number of motion types.

2.2 Learning-based Methods

In order to overcome the limitations of manually designed image priors or specific camera motions, learning-based methods aim to directly predict blur kernels (or deblurred results) from an input blurry image. Benefiting from the development of CNNs, learning-based models can be trained on a large amount of blurry data and can perform blur estimation (or removal) in an end-to-end manner.

Most learning-based methods were originally proposed to estimate blur causes/representations from blurry images [31], [35], [36], [37]. For example, [35], [36] attempted to identify the type of blur from a restricted set of parametrized blurs. Schuler *et al.* [37] proposed a CNN module for learning a gradient-like representation and estimated the blur kernels by dividing the learned representation in Fourier space. Similarly, [38] predicted the Fourier coefficients of a deconvolution kernel that modeled blind motions of an image patch. Sun *et al.* [16] proposed a CNN-based model to predict the probabilistic distribution of motion blur at the patch level. In their method, a well-trained model estimated the direction and length of non-uniform linear motions. Then, [15] developed a fully convolutional framework to achieve pixel-wise prediction of blur kernels. Compared to optimization-based methods, these learning-based methods were more flexible and more efficiently estimated motion blur. However, during training, most learning-based methods required the ground truths of blur representations for supervision. Since the ground truths of real-world blurry data are rarely available, these methods were trained on artificially-generated training examples, limiting the approach to some simple blur types (*e.g.*, linear motion). For more complex real-world dynamic motion, new blur representations and learning schemes are required to improve the estimations.

Accompanying the increased fitting capability of CNNs, many learning-based methods have been proposed to directly restore the latent sharp image from a blurry input [1], [2], [3], [22], [39], [40], [41], [42], [43]. Among these methods, [1] proposed a multi-scale network which performed deblurring in a "coarse-to-fine" pipeline. Then, [2], [22] further improved on this strategy by altering the parameter sharing and independent scheme. By combining three CNNs and a recurrent neural network (RNN), Zhang *et al.* [40] employed the learned variant RNN weights to model spatial-variant blurs. Inspired by [40], many methods [4], [44], [45] have adopted a spatial-variant convolutional module as a substitute for some of the original convolution layers to increase the size of the receptive field in a more compact way. In addition, Kupyn *et al.* [39] and Ramakrishnan *et al.* [46] combined deblurring with generative adversarial networks (GANs) to synthesize more realistic sharp images. Overall, the combination of recently established real-world blurry datasets [1] and the powerful learning capability of CNNs have allowed learning-based methods to achieve impressive performance for directly synthesizing deblurred images. Unfortunately, the causes of blur (motions) are generally ignored in these works, preventing the exploration of the rich dynamic information contained in blurry images and introducing training difficulties for related tasks due to a poor understanding of dynamic motion blur. For example, in the absence of motion information, some deblurring and video extraction approaches either require a large receptive field to model large-scale blur [40] or require a complex training scheme and iterative inferences [18], [19], [47]. In this work, we show that improving blur estimation can contribute to overcoming these problems and solving these tasks.

3 Exposure Trajectory Recovery

3.1 Motion Exposure Mechanism

When a camera takes a photograph, the exposure time cannot be instant due to technological constraints and physics (*i.e.*, exposure requirements). Therefore, a photograph records a target scene over a period of time. The exposure process can be formulated as:

$$B = \int_0^\tau H(L,t)\,dt, \tag{1}$$

where L represents the latent content/scene in the photograph, $H(L,t)$ denotes the instant frame at time t, and τ denotes the camera exposure time. Due to camera shake or the motion or deformation of objects in the scene, $H(L,t)$ may continuously vary with respect to time, leading to dynamic scene blurry image B.

In this work, we assume the middle instant sharp frame L_s records all visual information of latent content/scene L.[1] According to the principle of camera exposure, the function $H(L_s,t)$ can be defined as an image wrapping operation that performs a pixel-wise shift over different times, *i.e.*,

$$H(L_s,t) = L_s(\mathbf{P} + \Delta\mathbf{P}^t), \tag{2}$$

where $\mathbf{P}$ denotes all pixels in L_s, and $\Delta p^t = (\Delta x^t, \Delta y^t)$ is the shift of pixel (x,y) at time t. Assuming the brightness

[1] For most dynamic scene motion blur datasets, the middle instant frame is regarded as a sharp ground truth.

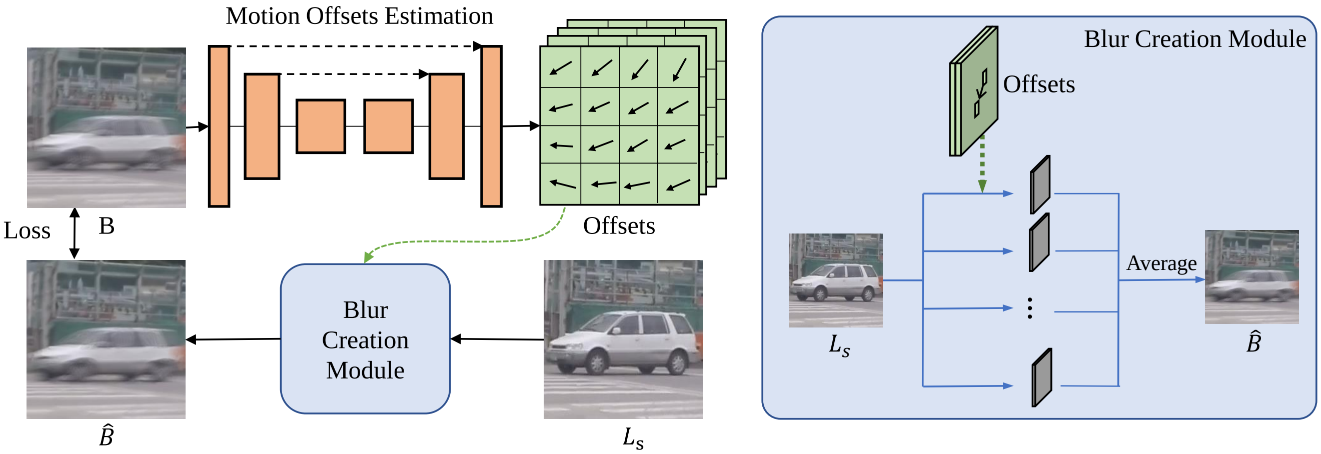


Fig. 1: Illustration of our proposed motion offset estimation method. The figure on the left is our motion offset generation network. It takes blurry images as input and outputs the corresponding motion offsets. Afterwards, the *blur creation module* (on the right) takes a sharp image and the extracted motion offsets to reconstruct the input blurry image.

remains constant during exposure, we consider Eq. (1) and (2) and discretize them over multiple time steps N to derive the formation of a blurry pixel p_0 as:

$$B(p_0) = \frac{1}{N}\sum_{n=0}^{N-1} L_s(p_0 + \Delta p_0^{t_n}), \tag{3}$$

which means a blurry pixel can be represented as the accumulation of pixels in the latent image moved by Δp^{t_n}.

In this work, instead of deriving the blur kernels of a blurry image, we directly focus on the spatial shift Δp^{t_n} of each pixel. Thus, we propose a new time-dependent blur representation, exposure trajectory which can be sampled by a set of motion offsets ($\{\Delta \mathbf{P}^{t_n}\}_{n=0}^{N-1}$). Similar to conventional blur kernels, the proposed exposure trajectory directly act on sharp images and then output blurry results. In contrast, our exposure trajectory models the blur formation as a spatial shift through time.

3.2 Blur Creation Module

Based on the proposed exposure trajectory and motion exposure mechanism, we devise a *blur creation module* which takes one sharp image L_s and a set of motion offsets as inputs to generate a dynamic blurry image. For each blurry pixel (*i.e.*, exposure location) p, the proposed *blur creation module* is asked to locate pixels $p^{t_n} = p + \Delta p^{t_n}$ in a latent sharp image L_s (Eq. (3)) and further average them to obtain a blurry pixel. Since a real-world dynamic motion is continuous, we employ the bilinear interpolation to calculate the pixel value of location p^{t_n},

$$L_s(p + \Delta p^{t_n}) = L_s(p^{t_n}) = \sum_q G(q, p^{t_n}) \cdot L_s(q), \tag{4}$$

where q enumerates the referenced neighborhood points of the sampling location p^{t_n}, and $G(\cdot,\cdot)$ is the bilinear interpolation kernel. As illustrated in Fig. 1, our motion offsets are of the same spatial resolution as the input image. Each offset has two channels corresponding to 2D axes. In practice, the *blur creation module* takes N motion offsets and a sharp image L_s as inputs and synthesizes an averaged blurry output.

Discussion. Compared to conventional blur kernels, the proposed exposure trajectory (or motion offset) aims to mimic the exposure process of a camera sensor. If we assume the latent content/scene is known, motion offsets encode motion/dynamic information during an exposure period and can further synthesize a blurry image. Mathematically, the proposed motion offsets can be expressed in the formulation of blur kernels. Specifically, in the general blur kernel model, a blurry image B is represented as $B = k * L + noise$, where k represents a blur kernel. In such a framework, our motion offsets can be regarded as an equivalent blur kernel $k(p_0, t_n)$ of the location p_0 over time $\{t_n\}_{n=0}^{N-1}$,

$$k(p_0, t_n) = \begin{cases} \frac{\delta(p - (p_0 + \Delta p_0^{t_n}))}{N}, & \text{if } p_0 + \Delta p_0^{t_n} \in L_s \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\delta(\cdot)$ denotes the Dirac delta function.

Analyzing the equivalent formulation, the following differences between conventional blur kernels and our proposed motion offsets may exist. First, a time variable t_n is introduced as a new and important element to model the exposure process. Under reasonable constraints (discussed in Section 3.4), our time-dependent motion offsets can act as a visualizable and interpretable representation of motion blur. Second, different from the weight matrix in a conventional blur kernel, our motion offsets calculate a uniform average of wrapped frames over a time sequence. We assume each time step is equally discretized; thus, the degree of motion (or blur) in each position is represented by the learned spatial displacements $\{\Delta p^{t_n}\}$. Since the values of Δp^{t_n} are continuous, bilinear interpolation (Eq. (4)) is performed to derive the value on each discrete pixel position. Note that the bilinear interpolation operation plays the same role as the weight matrix in blur kernels. Third, benefiting from the spatial shift operation (*i.e.*, Δp^{t_n}), our equivalent blur kernel (*i.e.*, motion offset) will not be limited by size, shape, pattern, or resolution, as traditional kernels are. For example, compared to the dense blur kernels estimated in [20], where each kernel size is 33×33, the proposed motion offsets only carry $N \times 2$ parameters[2]. Finally, since our motion offsets are compact and differentiable, the blur creation module can easily be integrated into deep neural networks and trained in an end-to-end manner.

3.3 Motion Offset Estimation

The biggest problem of previous learning-based blur kernel (motion) estimation is the universal absence of ground truth blur kernels for real-world data. Thus, [15], [16] must use synthetic data for training. Based on the motion exposure mechanism, the proposed motion offset could replace conventional blur kernels. Exploiting its compact and differentiable advantages, we devise a training scheme that performs motion offset estimation without any ground truths of the motion information. Specifically, the *blur creation module* is connected with the motion offset estimation network to form a cyclical pipeline.

Given a ground truth blurry image B and a sharp reference frame L_s, the motion offset estimation network takes B as an input and outputs N motion offsets; then, the blur creation module takes L_s and the motion offsets as input to reproduce the estimated blurry image $\hat{B}$. Fig. 1 illustrates this procedure. The motion offset estimation network is based on an encoder-decoder network with skip connections, and the detailed model structure is provided in Section 5.1.

The loss of this cyclic reconstruction can be written as:

$$\mathcal{L}_{circle} = \mathcal{L}_{l_2} + \lambda_{SSIM}\mathcal{L}_{SSIM}, \tag{6}$$

where $\mathcal{L}_{l_2}$ and $\mathcal{L}_{SSIM}$ denote the ℓ_2 loss and **SSIM** loss respectively. Both are applied to measure the difference between B and $\hat{B}$. We elaborate these two terms as follows:

$$\mathcal{L}_2 = ||B - \hat{B}||_2^2, \tag{7}$$

$$\mathcal{L}_{SSIM}(P) = 1 - \textbf{MS-SSIM}(\tilde{p}), \tag{8}$$

where $\tilde{p}$ is the center pixel of patch P, and **MS-SSIM** denotes the multi-scale SSIM. A more specific definition and implementation can be found in [48]. The reason that we use the SSIM loss is that the ℓ_2 loss only weakly penalizes our output because it tends to generate average results, and the blurry image is already averaged. In this case, the SSIM loss more accurately measures the distance between two blurry images.

We also introduce other losses to regularize motion offsets. First, we apply a regularization loss to encourage offsets that search for nearby pixels as solutions. This benefits the situation in which there is a large smooth region, *e.g.,* the sky or ground, where large displacements (offsets) should be suppressed. Moreover, due to dynamic motion blur usually being continuous along the space, we apply the total variation loss to encourage spatial smoothness within offset maps. These two losses can be formulated as:

$$\mathcal{L}_{reg} = \frac{1}{Nwh}\sum_{n=1}^{N}\sum_{i=1}^{w}\sum_{j=1}^{h} M_n(i,j)^2, \tag{9}$$

$$\begin{aligned}\mathcal{L}_{tv} = \frac{1}{N}\sum_{n=1}^{N}\Big(&\frac{1}{(w-1)h}\sum_{i=0}^{w-1}|M_n(i,j) - M_n(i+1,j)| + \\ &\frac{1}{w(h-1)}\sum_{j=0}^{h-1}|M_n(i,j) - M_n(i,j+1)|\Big),\end{aligned} \tag{10}$$

where $M_n(i,j)$ denotes the location (i,j) in the n^{th} offset map.

In summary, the final loss function is a weighted sum of the above losses:

$$\mathcal{L} = \mathcal{L}_{circle} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{tv}\mathcal{L}_{tv}. \tag{11}$$

3.4 Different Constraints to Motion Offsets

If we directly learn all the motion offsets using the framework described above, namely a zero constraint (ZC) model, the results will be as shown in Fig. 2 (a). Though achieving impressive reblurring accuracy, its ill-posed nature creates the following problems: (1) the learned motion offsets are one of several possible solutions of blur formation, and since it is difficult to form them into an explicit trajectory as in real-world blur formation, the learned motion offsets are usually sub-optimal for describing realistic motion; and (2) although there exists the temporal variable t_n in our learned offsets, these offsets are disordered due to a lack of spatial-temporal relationship modeling.

Therefore, we devise several constraints to reduce the ill-posed nature of motion estimation and to form the motion offsets into an explainable exposure trajectory:

(Bidirectional) linear trajectory constraint. The linear assumption is used to fit motion blur in many methods [10], [15], [16]. We also devise a linear trajectory constraint for motion offsets. Recall our assumption (Section 3.1) that the sharp image L_s represents the middle instant frame, *i.e.* $\Delta p^{t_{\text{mid}}} = (0,0)$. To represent linear motion, the motion offset estimation network only needs to predict another point on the exposure trajectory. Suppose the blurred pixel is caused by uniform linear motion and the predicted offset Δp is an endpoint of the exposure trajectory, the other offsets can be derived as:

$$\Delta p^{t_n} = (1 - \frac{2n}{N-1})\Delta p, n = 0, \ldots, N-1. \tag{12}$$

We attempt to predict the furthest point (endpoint) of the exposure trajectory based on the observation that the blurred edge is easier to capture and estimate.

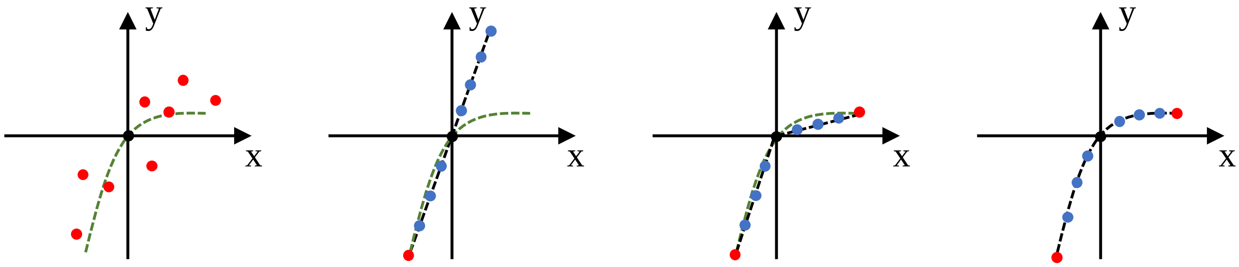
Taking a further step, we can apply a bidirectional linear (b-d linear) constraint to our motion offsets. As shown in Fig. 2 (c), we predict two offsets Δp_1, Δp_2 to represent the start and end points of each exposure trajectory. Then, the other offsets can be calculated as:

$$\Delta p^{t_n} = \begin{cases}(1 - \frac{2n}{N-1})\Delta p_1, & n = 0, \ldots, \frac{N-1}{2}, \\ (\frac{2n}{N-1} - 1)\Delta p_2. & n = \frac{N+1}{2}, \ldots, N-1.\end{cases} \tag{13}$$

As shown in Fig. 2, this trajectory better fits a curve than the linear one.

Quadratic trajectory constraint. Although the bi-directional linear constraint already introduces a certain non-linearity into trajectory learning, the quadratic function can better approximate real-world motion [49], [50]. A quadratic curve

[2] As shown our experiments, setting N as 15 already achieves extraordinary performance.



(a) No Constraint (b) Linear (c) B-d Linear (d) Quadratic

Fig. 2: Examples of motion offsets with different constraints. Suppose the green curve is the ground truth exposure trajectory. (a)-(d) simulate the fitting results of motion offsets with no constraint, linear constraint, b-d linear constraint, and quadratic constraint, respectively. Red points are the offsets that output by the estimation network and blue points are calculated by the different constraints.

can be derived when an object is moving with constant acceleration, a much stronger fitting than the (bi-)linear assumption. Thus, we devise a quadratic trajectory constraint to force a smooth quadratic trajectory on our motion offsets. Unlike previous works, which apply a quadratic trajectory between video frames, we extract this trajectory inside a single blurry frame. Specifically, we still predict two offsets Δp_1, Δp_2 as the start and end points of the exposure trajectory, with the other offsets written as:

$$\begin{aligned}\Delta p^{t_n} = &\frac{\Delta p_1 + \Delta p_2}{2}(\frac{2n}{N-1} - 1)^2 \\ &+ \frac{\Delta p_2 - \Delta p_1}{2}(\frac{2n}{N-1} - 1), n = 0, \dots, N-1.\end{aligned} \tag{14}$$

Thus, motion offsets will be formed into a quadratic trajectory (Fig. 2 (d)). Note that since our motion offsets are modeled in equidistant time, the learned motion offsets not only match a curvilinear exposure trajectory but also reflect the changing velocity. For example, a longer displacement between adjacent time steps corresponds to faster movement.

4 Applications Benefiting from Recovered Exposure Trajectories

In this section, we apply our recovered exposure trajectory to two motion blur-related downstream tasks, *i.e.* image deblurring and video extraction from a single blurry image, which further demonstrates the benefit of recovering accurate motions from a blurry image.

4.1 Motion-aware Image Deblurring

To handle the challenging problem of dynamic scene deblurring, existing works employ complex network architectures to enlarge the model capacity, such as a multi-scale structure [1], [2], [22]. Some other methods [4], [40], [44] claim that spatially invariant convolution filters, *i.e.* spatially uniform and limited receptive fields are sub-optimal for modeling dynamic scene blur. With the recovered exposure trajectories, we aim to design a spatial-variant deblurring network, which leads to a more compact and efficient model.

As derived in [4], [40], the blurry image deconvolution can be written as an Infinite Impulse Response (IIR) formula, from which they drew two main conclusions: 1) the deblurring process requires a very large receptive field; and 2) for a CNN-based deblurring model, the convolution filters should have a similar direction/shape with the blur kernel. For example, if a blur kernel is linear and horizontal, the latent pixels can be calculated using only horizontal blurry pixels, thus the deconvolution filters should also be pure horizontal. However, the conventional square-shaped convolutional filter cannot meet these requirements. In this work, we propose a motion-aware deblurring network with spatial-variant convolution filters that are shaped by the recovered exposure trajectories.

To build a spatial-variant convolution module, the deformable convolution unit [51] provides a general solution. In recent works [4], [44], spatial-variant deblurring modules based on deformable convolutions have achieved reasonable performance. However, since the ground truth of the kernel shape is absent, these methods attempt to derive deformation offsets from encoded features of an input blurry image. We propose the motion-aware convolution (MA Conv.), which directly employs the recovered motion offsets to model the aforementioned filter deformation. Our motion-aware convolution can be formulated as:

$$y(p_0) = \sum_{n=0}^{N} w(p_n) \cdot x(p_0 + \alpha \Delta p_0^n), \tag{15}$$

where x is an input feature map, y is an output feature map, and w is the weight of the convolution filter. The coordinate $p_0 + \alpha \Delta p_0^n$ denotes the sampling location calculated by the centering coordinate p_0 and an offset $\alpha \Delta p_0^n$, which controls the shape and size of the convolution, and $w(p_n)$ is the weight corresponding for the sampling point $p_0 + \alpha \Delta p_0^n$. For a square-shaped 3×3 convolutional filter with dilation 1, $\Delta p_0^{t_n} \in (-1,-1), (-1,0), \dots, (0,1), (1,1)$ and $\alpha = 1$. In our motion-aware convolution, Δp_0^n are decided by our recovered motion offsets. Specifically, giving the recovered exposure trajectory, we calculate 9 Δp_0^n centered by the middle offset $\Delta p_0^{\text{mid}} = (0,0)$ to modulate the original 3×3 kernel, as shown in Fig. 3. Moreover, we use hyper-parameter α to scale the size of the convolution, and α is set to 0.1 in our experiments. In this way, the proposed motion-aware convolution takes full advantage of the information contained in motion offsets, *i.e.* both direction

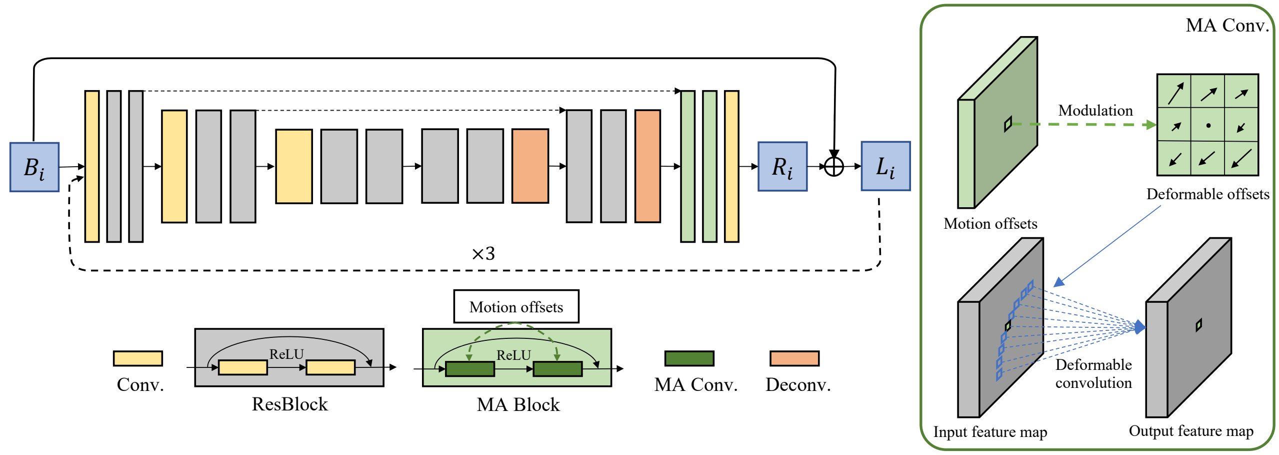


Fig. 3: The proposed motion-aware deblurring network. An encoder-decoder residual architecture for image deblurring is shown on the left, while the schematic of a motion-aware convolution in the motion-aware block is shown on the right.

and magnitude, resulting in a more mathematically accurate deconvolution.

Here, we adopts the DMPHN(1-2-4) [3] as the backbone architecture of our motion-aware deblurring network, since it is relatively compact among the state-of-the-art models. As shown in Fig. 3, similar with the most existing image deblurring methods, the encoder-decoder structure is employed. Compare to the vanilla DMPHN(1-2-4), our motion-aware deblurring network can be easily derived by replacing the selected convolutional layers with the proposed motion-aware convolution. According to our experiments, adding the motion-aware convolutions in the last stage of the decoder achieved the best performance. In addition, to build a compact deblurring network, we do not employ the stack-DMPHN as Zhang *et al* [3]. Adding the motion-aware module can already achieve comparable results, and our model largely reduces the memory cost.

4.2 Warping-based Video Extraction from a Single Blurry Image

Different from conventional blur kernels, our motion offsets contain temporal information that could help us to restore time series from a blurry input. As indicated in Eq. (2), frame L^{t_n} can be obtained through a transformation $H(\cdot,\cdot)$. Now, with the deblurring result $\hat{L}_s$ and the estimated motion offsets $\hat{\mathbf{P}}^{t_n}$, we can generate the estimated frame $\hat{L}^{t_n}$:

$$\hat{L}^{t_n} = \hat{L}_s(\mathbf{P} + \Delta\hat{\mathbf{P}}^{t_n}). \tag{16}$$

Unlike forward optical flow which is a many-to-one mapping, our motion offset is equivalent to backward warping flow. Specifically, the backward warping flow represents the pixel displacement from warped result to the warping input. Therefore, there will be no holes in our warped result. According to Section 3.4, since we have added different trajectory constraints, theoretically we can interpolate arbitrary N offsets into our start and end offsets, which further leads to smooth or even slow-motion video output.

To our best knowledge, only two existing works have been capable of restoring a video sequence from a single blurry image. [18] first attempted to generated a video sequence from a single blurry image by training different networks to generate frames at different time t_n, only producing limited frames. [19] proposed a recurrent network to address temporal ambiguity, inferring the recurrent state at each time step t_n. Unlike these methods, we only need to calculate our motion offsets once, which is more time efficient. Moreover, these previous methods are trained with a series of ground truth sharp frames for supervision, which limit the generated outputs to specific time intervals. Our motion offset estimation module is easy to train and requires fewer annotations. Moreover, during test, our model is more compact and efficient.

TABLE 1: Detailed architecture of the motion offset estimation network. + denotes that a skip connection concatenates this layer with the corresponding layer in the encoder.

Stage	Output	Layer Details
	$\frac{H}{2} \times \frac{W}{2}$	Space to Depth
Conv1	$\frac{H}{2} \times \frac{W}{2}$	$5 \times 5, 12, 16$, stride 1
ResBlock1	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 5 \times 5, 16 \\ 5 \times 5, 16 \end{bmatrix} \times 3$
Conv2	$\frac{H}{4} \times \frac{W}{4}$	$5 \times 5, 16, 32$, stride 2
ResBlock2	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 5 \times 5, 32 \\ 5 \times 5, 32 \end{bmatrix} \times 3$
Conv3	$\frac{H}{8} \times \frac{W}{8}$	$5 \times 5, 32, 64$, stride 2
ResBlock3	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 5 \times 5, 64 \\ 5 \times 5, 64 \end{bmatrix} \times 3$
Bottleneck1	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 1 \times 1, 64, 128 \\ 3 \times 3, 128, 64 \end{bmatrix}$
Dconv1	$\frac{H}{4} \times \frac{W}{4}$	$5 \times 5, 64, 32$, stride 2
Bottleneck2	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 1 \times 1, 32+32, 128 \\ 3 \times 3, 128, 64 \end{bmatrix}$
Dconv2	$\frac{H}{2} \times \frac{W}{2}$	$5 \times 5, 64, 16$, stride 2
Bottleneck3	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 1 \times 1, 16+16, 64 \\ 3 \times 3, 64, 32 \end{bmatrix}$
Dconv3	$H \times W$	$5 \times 5, 32, 32$, stride 2
Conv4	$H \times W$	$5 \times 5, 32, 4$, stride 1

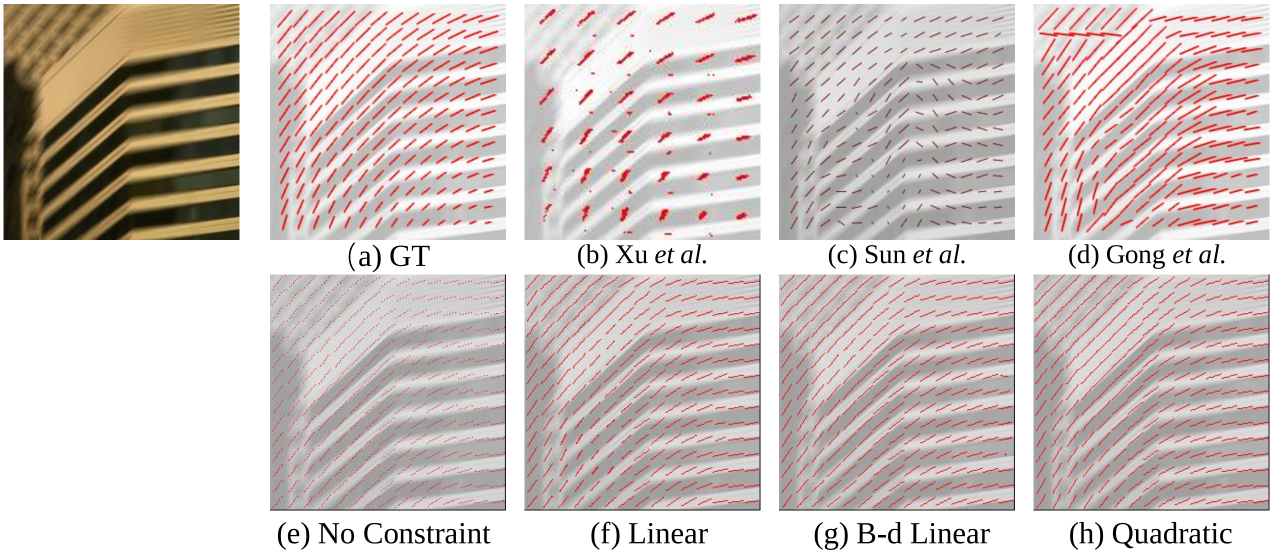


Fig. 4: Examples of motion estimation on the synthetic dataset. The top row shows the blurry input, ground truth motion, and results of previous methods. The bottom row shows our estimated motion offsets under different constraints.

5 Experiments

In this section, we first introduce our training configuration before carrying out quantitative and qualitative comparisons between our method and state-of-the-art methods for motion estimation, image deblurring, and video extraction.

5.1 Implementation Details

We provide layer-wise details of our motion offset estimation networks in Table 1. H and W represent the height and width of an input blurry image. For training both the *motion estimation network* and *deblurring network*, we use Adam [52] for optimization, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is set initially to 10^{-4} and it is linearly decayed to 0. For motion offset estimation, we set the offset number to $N = 15$, $\lambda_{SSIM} = 0.1$, $\lambda_{reg} = 0.00002$, $\lambda_{tv} = 0.0005$. All weights are initialized using Xavier [53], and bias is initialized to 0. We first train the motion estimation network using the blurry image and the ground-truth sharp image as the training pair. Then we train the deblurring network with the pre-trained motion estimation network. To train the deblurring network, besides paired training images, the estimated exposure trajectories are utilized as an auxiliary input.

5.2 Datasets

We employ two different datasets. The synthetic dataset provides ground truth blur kernels, while the GoPro dataset is synthesized from real-world frame with more challenging dynamic motion blur without ground truth blur kernels.

Synthetic Dataset. We follow the same approach as in [15] to generate blurry/sharp image pairs with pre-defined blur kernels. Specifically, blur kernels are represented by a motion flow map filled with pixel-wise non-uniform motion vectors. Each vector can form a linear blur kernel. Same as [15], we use images from BSD500 [54], which consists of 200 training images and 100 test images, as sharp ground truths. We then generate 50 motion flow maps for each training image and 3 motion flow maps for the test images. Finally, the sharp images are convolved with the corresponding flow maps to generate blurry images.

The **GoPro Dataset** [1] addresses the problem that synthetic data are different from real-world blurry images containing more complex dynamic motion. More realistic blurry images are generated by averaging consecutive short-exposure frames from a high frame rate video, *e.g.,* 240fps, taken from a GoPro camera. In this way, [1] collected 3214 blurry/sharp image pairs, and split them into a training set with 2103 pairs and a test set with 1111 pairs. In following experiments, unless stated, the quantitative results are based on the GoPro dataset.

5.3 Evaluation of Motion Offset Estimation

We compare the proposed exposure trajectory recovery with one conventional blur kernel estimation method (Xu *et al.* [55]) and two recent learning-based blur kernel estimation methods (Sun *et al.* [16] and Gong *et al.* [15]). Our comparisons are based on both the synthetic and GoPro datasets.

Evaluation Metrics. In order to evaluate the accuracy of motion estimation, we calculate the PSNR and SSIM metrics between the input blurry image and reblurred image via estimated blur kernel/motion offsets for both datasets. Specifically, the reblurred results of [16] and [15] can be obtained by convolving a sharp image with the estimated motion flow map. We also apply the MSE metric of motion to evaluate the synthetic data. This metric defines the mean squared error between the ground truth motion and estimated motion [15]. The MSE is easy to calculate in [16] and [15] since their estimated blur kernels share the same form as the ground truth, namely 2D vectors. However, our motion offsets are a set of points, so we calculate the vector of two endpoints as a simplification based on the assumption that the motion is linear. Note that we only provide the kernel visualization results of [55], since its blur kernel cannot be represented as a pixel-wise motion flow map like the others.

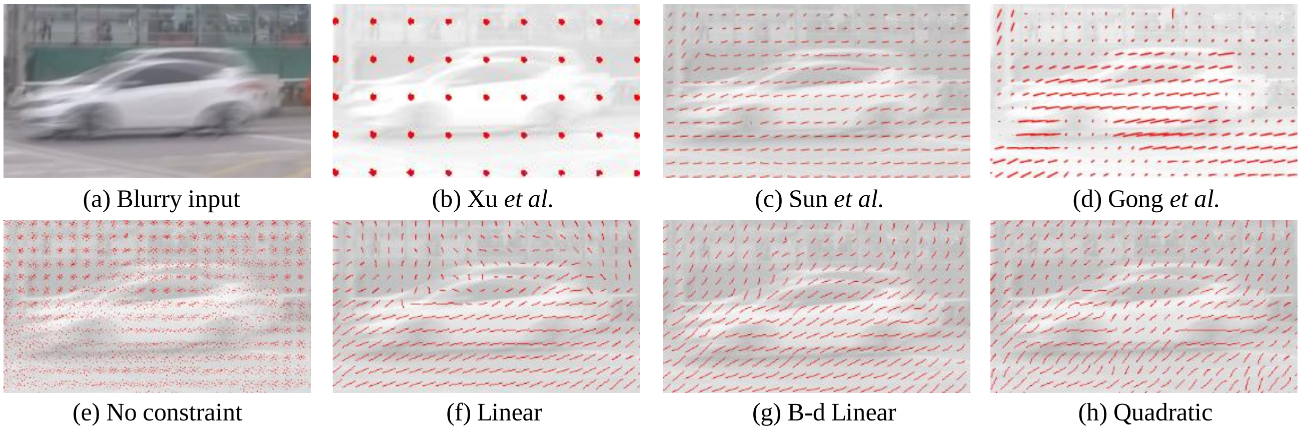


Fig. 5: Examples of motion estimation on the GoPro dataset. The top row shows the blurry input and results of previous methods. The bottom row shows our estimated motion offsets under different constraints.

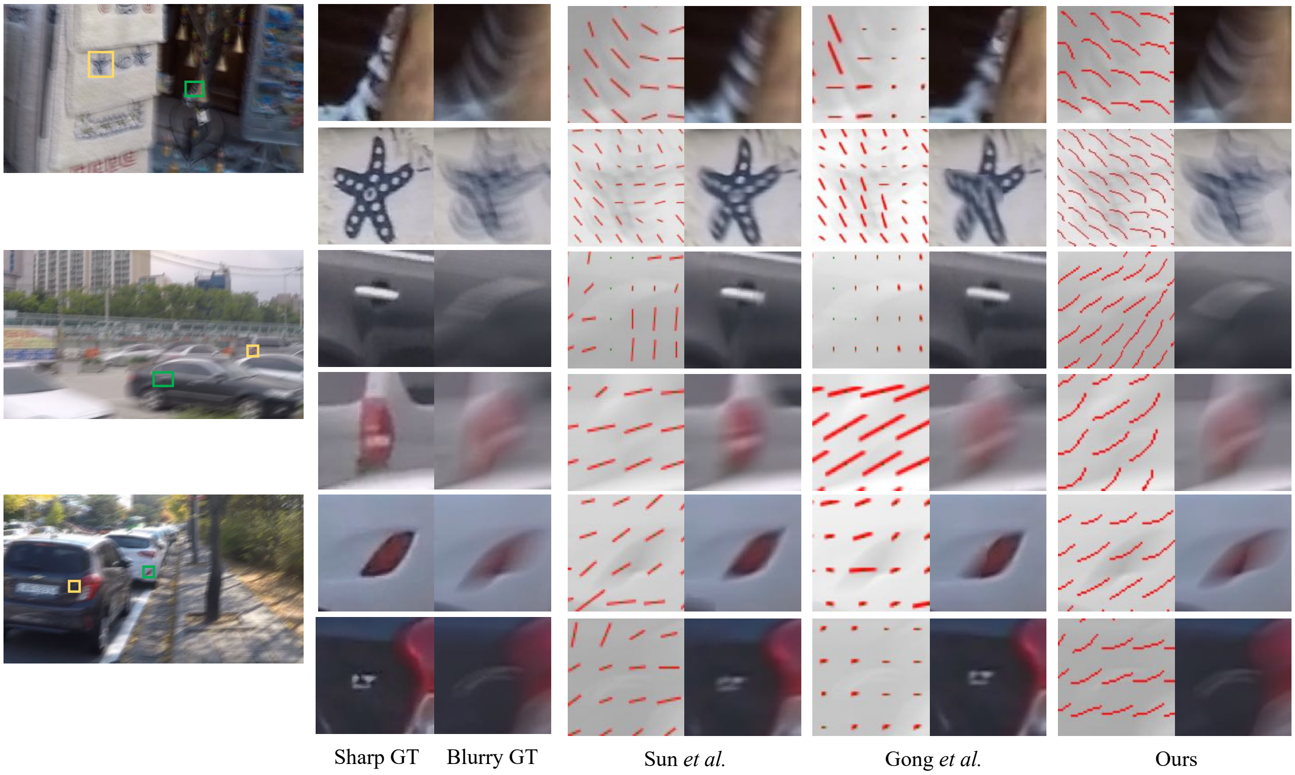
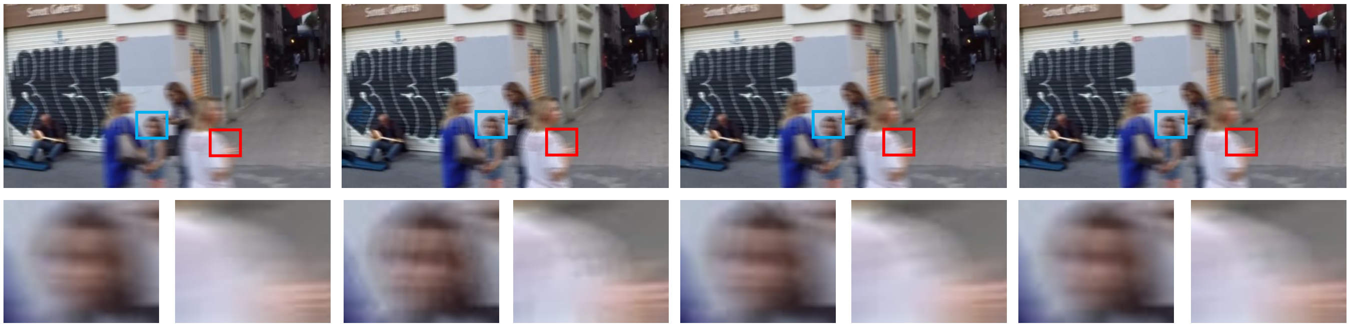


Fig. 6: Visual comparison of motion estimation on GoPro test set. From left to right shows the groud-truth image patches, the results of Sun *et al.* [16], Gong *et al.* [15] and our quadratic model. For each method, we visualize the estimated motion field and the reblurred result.

Motion Estimation on the Synthetic Dataset. Table 2 shows our quantitative comparisons on the synthetic dataset. Our models with different constraints achieve comparable or better performance to [15]. It is noteworthy that [15] is learned in a supervised manner, yet our training scheme need no ground-truth motion as supervision. Although the synthetic blur is linear, we can see the two non-linear constraint models (our b-d linear and quadratic) producing better reblurring PSNR results than the two linear constraint models ([15] and our linear), we infer that (i) comparing with linear model, the non-linear model has a higher degree of freedom brought by the estimation of two endpoints(only one for linear model), thus a greater fitting ability; (ii) the motion field of synthetic blur changes continuously and form into curves in the space (Fig. 4). Overall, although the blurry kernel on each pixel is linear, the quadratic model has better representation ability and delivers a more accurate approximation for continuous change modeled in the synthetic dataset.

We can also make some observations from Fig. 4. Xu *et al.* [55] generates non-trajectory kernels, for which we can only vaguely observe the flow after post-processing. Since Sun *et al.* [16] performs a patch-level prediction from a blurry input, it is usually misled by the smooth area. Gong *et al.* [15] shows more continuity across space, but there is also the possibility of when a region of predictions going wrong.



Blurry input 5 offsets 9 offsets 15 offsets

Fig. 7: The effect of offset number on blur creation. Left to right show the ground truth blurry image, the result of the model with 5 offsets, the result of the model with 9 offsets, and the result of the model with 15 offsets. It is clear that increasing the number of offsets creates a smoother and more realistic blurry output.

TABLE 2: Quantitative comparison of motion estimation on both synthetic and the GoPro [1] dataset.

Model		Sun *et al.* [16]	Gong *et al.* [15]	Zero constraint (ZC)	Linear	B-d Linear	Quadratic
Synthetic	PSNR	29.34	37.61	37.62	37.34	38.64	**38.9**
	SSIM	0.9001	0.9818	0.9763	0.9857	0.9872	**0.9882**
	MSE	50.12	10.05	-	7.42	7.16	**3.27**
GoPro	PSNR	29.68	30.61	**35.82**	33.45	33.79	34.68
	SSIM	0.9282	0.9363	**0.9800**	0.9669	0.9687	0.9740
Runtime(s)		45.2	8.4	0.011	0.011	0.011	0.011

TABLE 3: Comparison for the setting of offset numbers N.

# of motion offsets	5	9	15
PSNR	34.09	34.52	**34.68**
SSIM	0.9668	0.9727	**0.974**

TABLE 4: Ablation study for loss function.

	Proposed	w/o SSIM	w/o tv	w/o reg
PSNR	**34.68**	34.16	33.96	34.56
SSIM	**0.974**	0.97	0.9672	0.9727

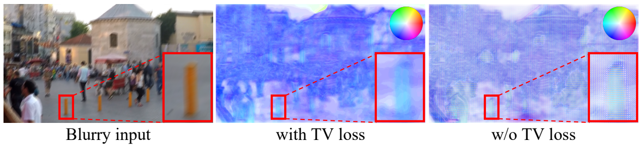


Fig. 8: Visualize the magnitude and direction of motion offsets in a color coded map. Different colors represent different directions. The model with TV loss generates a much more smooth color map than the model w/o TV loss (best view in high resolutions).

Conversely, ours are more accurate and can perfectly fit into linear motion regardless of the employed constraints.

Motion Estimation on the GoPro Dataset. Since there is no motion ground truth for the GoPro dataset, the methods in [15], [16] cannot train their networks. Here, we employ their models pre-trained on the synthetic dataset and then test them on the GoPro test set. It is unfair to directly compare these results with our own; however, to our best knowledge, no other method is trained without motion ground truths, so their results seem to be a legitimate reference.

Table 2 shows that the performance of [15], [16] decreases significantly with more complex dynamic scenes. This decrease in quality can also be observed in the example in Fig. 5 and Fig. 6. We first provide an example of visualized blur kernels estimated from different models in Fig. 5. As we can see, Xu *et al.* [55] fails to estimate dynamic motion blur, and Sun *et al.* [16] is obviously inaccurate and tends to generate spatially uniform kernels. The results using Gong *et al.* [15], although spatially variant, tend to produce many non-blurry regions. Our results, however, show a different flow direction in the background and foreground, *e.g.*, the moving car. To better demonstrate the accuracy and effectiveness of our proposed quadratic trajectory, we show more examples of the estimated motions and the corresponding reblurring regions in Fig. 6. Our models generate more accurate reblurring results compared existing works. Also, the quadratic model is more effective in recovering the quadratic motion trajectory. Note that since the warping through motion offsets is conducted backward rather than forward, the exposure trajectory is center-symmetrical to the object moving trajectory.

Ablation Studies. First, we discuss the setting of offset numbers N. As shown in Table 3, the model with $N = 15$ is notably better than the other models. The visual differences produced by altering the offset numbers are shown in Fig. 7. There is ghosting artifact with the model with 5 offsets, which becomes smoother as the offset number increases from 5 to 15. With 15 offsets, the result is very close to the ground truth image. Since increasing the number of offsets has little effect on performance, we set $N = 15$, considering the balance between performance and efficiency.

To demonstrate the effectiveness of the proposed loss function, we trained a model with all the proposed losses

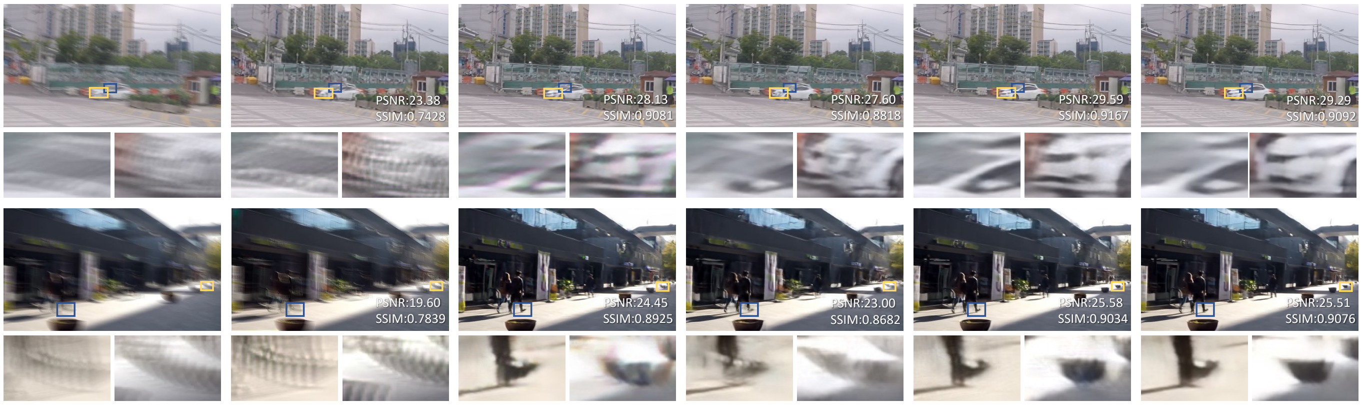


Fig. 9: Visual comparison with GoPro dataset. From left to right, we show input, deblurring result of DeblurGAN-V2 [56], Gao *et al.* [2], DMPHN [3], ours, and stack(4)-DMPHN [3] (best view in high resolutions).

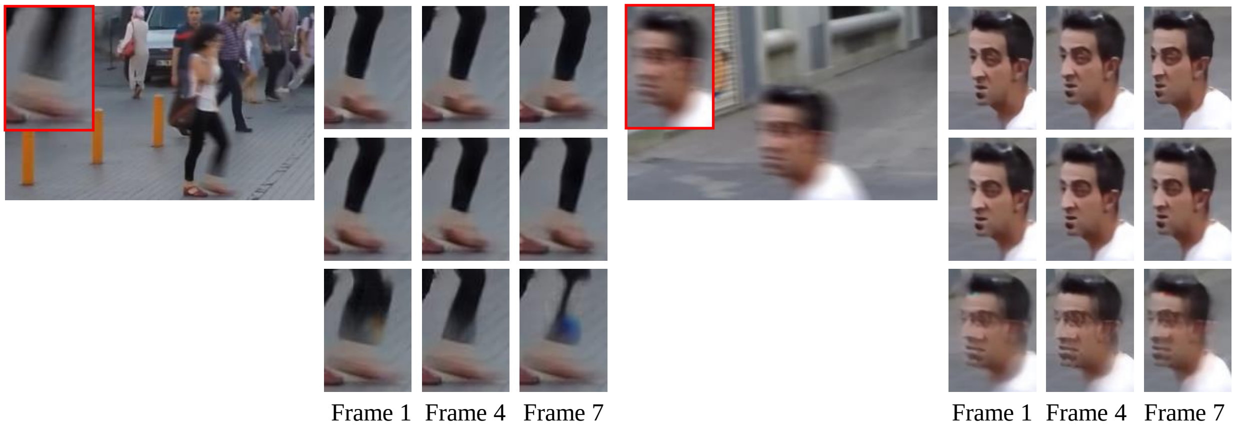


Fig. 10: Comparison of video extraction results. In the top-down order, we show ours, result of [19], result of [18].

TABLE 5: Quantitative deblurring results on GoPro dataset.

Model	Gong [15]	Nah [1]	Tao [22]	DeblurGAN [39]	DeblurGAN-V2 [56]	Gao [2]	DMPHN [3]	Ours	Stack(4)-DMPHN [3]
PSNR	26.89	29.08	30.26	28.7	29.55	30.92	30.21	31.05	**31.20**
SSIM	0.8639	0.9135	0.9342	0.9270	0.9340	0.9421	0.9345	**0.9485**	0.9453
Size(MB)	54.1	303.6	33.6	35.4	244.5	46.5	**21.7**	26.3	86.8

TABLE 6: Ablation study of motion-aware convolution on GoPro dataset.

Model	Baseline (w/o MA Conv.)	Zero constraint (ZC)	Linear	B-d Linear	Quadratic
PSNR	30.21	30.79	30.82	31.04	**31.05**
SSIM	0.9345	0.9459	0.9462	0.9483	**0.9485**

(Model **Proposed**), one without the SSIM loss (Model **w/o SSIM**), one without the total variation loss (Model **w/o tv**), and one without the regulation loss (Model **w/o reg**). The quantitative results are shown in Table 4. The proposed loss combination is better than those without certain losses. The total variation loss which encourage the local uniformity not only improves the motion estimation accuracy, but also solves the ambiguity of the motion direction. The ambiguity of direction always exists since a single blurry image barely contains the direction information, yet the predicted directions under total variance loss will be spatially continuous rather than random. as shown in Fig. 8, we visualize the magnitude and direction of optimized motion offsets by calculating the vector subtraction between the first motion offset and the last motion offset. The model with TV loss generates a smooth color map, while the model w/o TV loss generates the color map with noise dots, which means the directions of these pixels are discontinuous or even opposite to the adjacent pixels. Moreover, although the regulation loss has little influence on the metrics, it prevents the network from estimating large offsets in the smooth area.

5.4 Evaluation of Dynamic Scene Deblurring

We quantitatively and qualitatively compare our method with recent state-of-the-art dynamic scene deblurring methods: DeblurGAN(-v2) [39], [56] based on a conditional GAN

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3116135

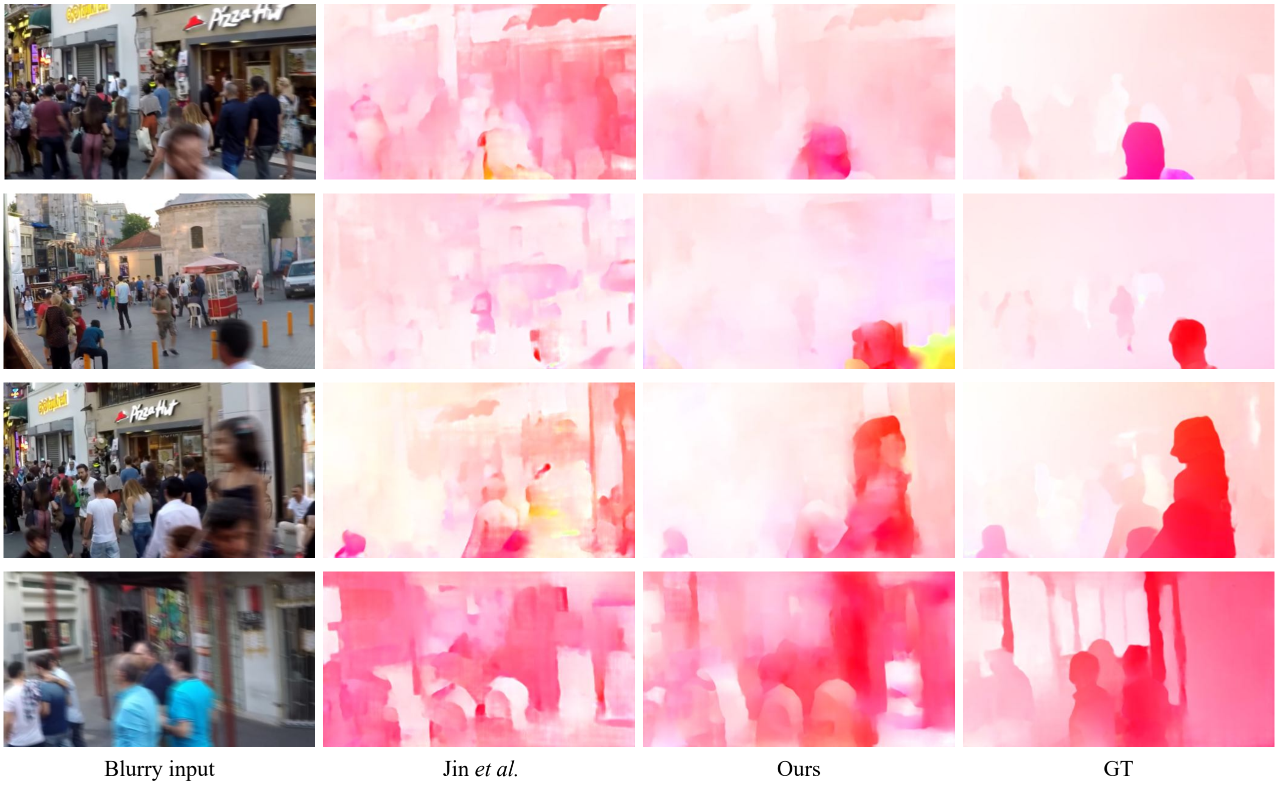


Fig. 11: Comparison of optical flow from the extracted videos. The optical flow is calculated from the first and last frame of the extracted videos (Jin *et al.* [18] and ours) and the ground truth high-frame-rate video (GT).

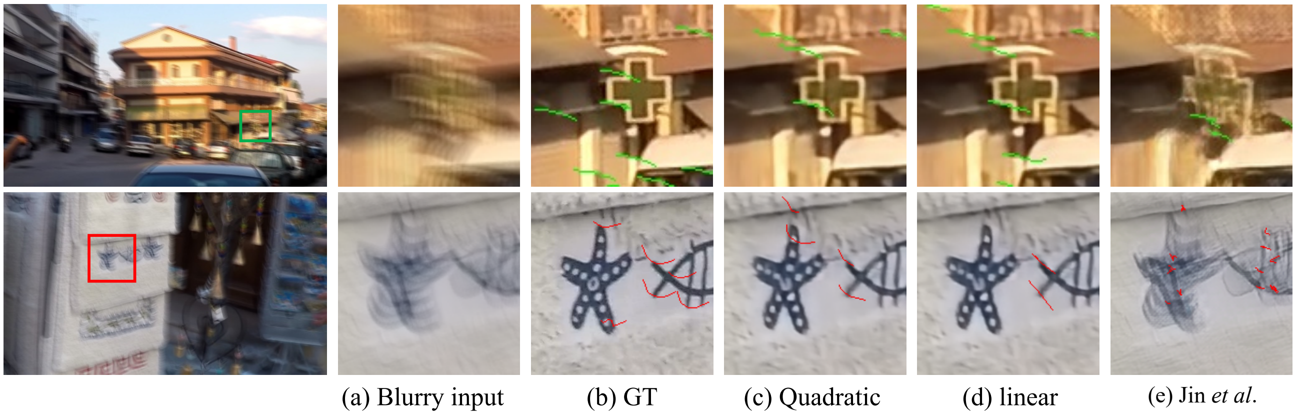

(a) Blurry input (b) GT (c) Quadratic (d) linear (e) Jin *et al.*

Fig. 12: Visualized trajectory result of extracted frames (best view in high resolutions).

to obtain a more realistic texture; Nah *et al.* [1], Tao *et al.* [22], and Gao *et al.* [2] built multi-scale networks but with different parameter sharing and parameter independence schemes; Zhang *et al.* [3] applies a Deep Multi-Patch Hierarchical Network (DMPHN), which is also the backbone network of our method. We also provide the deblurring results with [15] as representative of conventional MAP optimization. The quantitative results are presented in Table 5.

As illustrated in Table 5, our motion-aware deblurring network achieves comparable results to current state-of-the-art methods with respect to PSNR and achieved slightly better result with respect to SSIM. Note that, our model achieves such performance using a single-stack, which only costs about 30% of the model size compared to the model of the stack(4)-DMPHN. Considering the only difference between our model with DMPHN is the proposed motion-aware convolutional layer, it contributes 0.84 and 0.014 increasing in PSNR and SSIM, respectively. Also, as shown in Fig. 9, the visual results of ours are almost the same as stack(4)-DMPHN, while better than the other methods.

Besides verifying that the learned exposure trajectories could contribute to dynamic scene deblurring, we also conduct ablation studies to discuss the effects of different kinds of exposure trajectories. As Table 6 shows, compared to the baseline model, all kinds of exposure trajectories could improve the deblurring performance. The Model Linear and B-d linear perform slightly inferior to the Model Quadratic, owning to the less accurate of the motion estimation. Note that, though the exposure trajectory learned with zero-constraint (ZC) achieves the best score in above reblurring

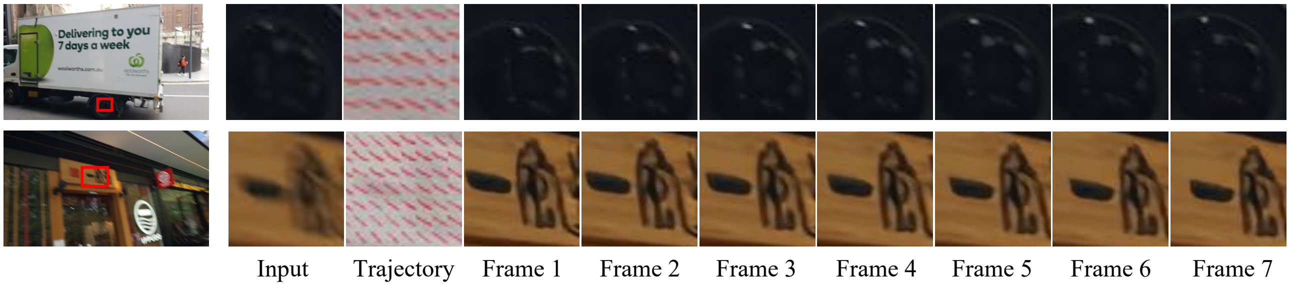


Fig. 13: Video extraction results with real images. More video results are provided in our supplementary video.

experiments (Table 2-GoPro), it demonstrates less effect in our deblurring module. A reasonable explanation is that the zero-constraint motion offsets are only one of the ill-posed solutions for reblurring reconstruction, yet it will not be the most accurate trajectory estimation.

5.5 Evaluation of Video Extraction

To evaluate the performance of our approach for video extraction, we compare our results with those of Jin *et al.* [18], Purohit *et al.* [19] and Zhang *et al.* [47]. Since the source codes of methods of Purohit *et al.* [19] and Zhang *et al.* [47] are not public yet, we can only provide the data recorded in their papers. We first directly compare the accuracy of the extracted frames. In Table 7, the PSNR and SSIM metrics are applied to measure the deblurring performance of the centered frame, and our method achieves the highest scores. Also, we can see from the Fig. 10 that the deblurring result (frame 4) and video extraction results (frames 1 and 7) of [18] both perform poorly when handling large blur. The results of [19] and our own show relatively sharp frames and clear object movements.

TABLE 7: Quantitative comparison of video extraction on GoPro dataset.

Method	PSNR	SSIM	EPE
Jin *et al.* [18]	26.98	0.881	9.32
Purohit *et al.* [19]	30.58	0.941	-
Zhang *et al.* [47]	30.64	0.942	10.03
Jin *et al.* + Ours	26.98	0.881	**6.07**
Ours	**31.05**	**0.949**	6.09

Then we further validate the accuracy of the motion in extracted videos through comparisons on the optical flow estimated from the synthesized videos. We follow the approach of calculating end-point error (EPE) in [47]. Specifically, we first estimate optical flow from the first generated frame to the last one with PWC-net [57] and vice versa. Then, EPE calculates the errors between estimated flows and the flow estimated from ground truth high-frame-rate video. The lower EPE value is chosen as the result, since the direction is uncertain. As shown in Table 7, our methods achieve the best EPE. For a fair comparison, we combine the centered frame generated from model of Jin *et al.* [18] with our motion offsets to warp the other frames, termed Jin *et al.* [18] + Ours, demonstrating that our improvement on EPE score comes mainly from the more accurate motion estimation rather than a superior deblurring result. Due to the limitation of using optical flow to evaluate video clarity, we obtain similar EPE scores from the generated video of Jin *et al.* [18] + ours and ours. To summary, the metrics in Table 7 demonstrate that the extracted video from our model is sharper and the encoded motion is more accurate. Qualitative comparison of the visualized optical flow in Fig. 11 also shows that the videos extracted using our method present a more accurate optical flow compared to [18].

Besides the validation of optical flow from first frame to last frame, we also validate the effectiveness of our quadratic trajectory. As shown in Fig. 12, we visualize the trajectory using feature point tracking [58]. Our quadratic motion offsets better fit the curve to the ground truth, especially in the second example.

Our exposure trajectory recovery framework employs blurry/sharp image pairs as training data. It delivers impressive optical flow estimation/trajectory results of dynamic scenes without accessing any motion supervision. Moreover, our motion offsets can be combined with any state-of-the-art deblurring method to generate even sharper video clips, while other methods need to train a whole new model. Finally, our model can generate arbitrary numbers of frames, while [18] can only achieve a fixed number. Although [19], [47] can also generate slow-motion videos, they need to conduct a iterative generation which increase the inference time. However, we only need to interpolate in our trajectory after a single forward prediction. As a result, our network is more compact and faster. The runtimes of [18], [19], and our model are 1.1 s, 0.39 s, and 0.22 s respectively. We also provide an example of video extraction from real images in Fig. 13, demonstrating a good generalization ability of our proposed method. More video results can be found in our supplementary video.

6 Limitations

There are two main limitations that existed in our proposed method. First, a more complicated motion may be caused by a large camera shake or a highly dynamic scene, which may need to be modeled by a higher order of exposure trajectory. In these situations, our quadratic constrained exposure trajectory can only act as an approximation of the ground-truth trajectory. However, since the motion captured in the dataset is relatively small, these situations rarely happen in the existing blurry image datasets. Second, similar to most learning-based deep models, our method may have generalization issues, it may fail to handle blur patterns that have a domain gap with the blur in training data. Considering our model still need to be trained on blurry/sharp image pairs, *i.e.*, the GoPro dataset, which is an approximation of

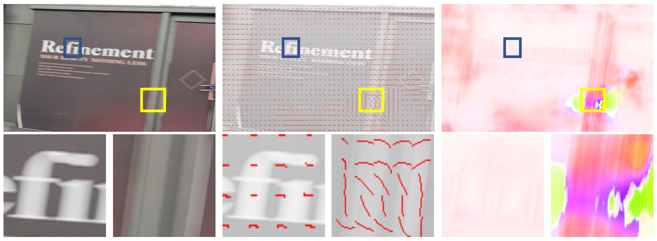


Fig. 14: A failure case on real-world blurry image. From left to right are the blurry input, the visualized exposure trajectory and color-coded motion offset map. The obvious error exists where the estimated motion is very small or has a wrong direction.

real-world blurry images, our well-trained network may fail in recovering the trajectory when encounter unseen real-world blur (Fig. 14). Solving this domain-gap problem is very challenging since an unsupervised training strategy is required for the unpaired real-world data. However, we believe our fully differentiable (re)blurring module can potentially contribute to unsupervised motion estimation/image deblurring, since it can provide a cycle-consistency loss. In the future, we will continue to devote to develop fully unsupervised deblurring and motion estimation methods for motion estimation.

7 Conclusion

Here we propose an exposure trajectory recovery scheme to generate motion offsets which are superior to conventional blur kernels in many respects. By imposing different constraints, these offsets can fit into different exposure trajectories. Moreover, we utilize the learned motion offsets for image deblurring and video extraction from a single blurry image. Experiments show that our motion offsets can produce useful information for solving these tasks. However, the learned exposure trajectories are still limited to motion of constant acceleration, and may not perfectly fit real situations. We will further devote to provide more accurate motion estimation and further improve the deblurring and video extraction tasks.

Acknowledgments

The authors would like to thank anonymous reviewers and the handlining editor for their constructive comments. This work was supported by Australian Research Council Projects FL-170100117, IH-180100002, IC-190100031.

References

[1] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3883–3891.

[2] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3848–3856.

[3] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5978–5986.

[4] K. Purohit and A. Rajagopalan, "Region-adaptive dense network for efficient motion deblurring." in *Assoc. Advanc. Artif. Intell.*, 2020, pp. 11 882–11 889.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 787–794.

[6] J. Jia, "Single image motion deblurring using transparency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2007, pp. 1–8.

[7] Y.-W. Tai, P. Tan, and M. S. Brown, "Richardson-lucy deblurring for scenes under a projective motion path," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 33, no. 8, pp. 1603–1618, 2010.

[8] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2011*. IEEE, 2011, pp. 2657–2664.

[9] T. Hyun Kim, B. Ahn, and K. Mu Lee, "Dynamic scene deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3160–3167.

[10] T. Hyun Kim and K. Mu Lee, "Segmentation-free dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2766–2773.

[11] L. Pan, R. Hartley, M. Liu, and Y. Dai, "Phase-only image based kernel estimation for single image blind deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6034–6043.

[12] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2010, pp. 171–184.

[13] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 98, no. 2, pp. 168–186, 2012.

[14] S. Zheng, L. Xu, and J. Jia, "Forward motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1465–1472.

[15] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2319–2328.

[16] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 769–777.

[17] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1279–1288.

[18] M. Jin, G. Meishvili, and P. Favaro, "Learning to extract a video sequence from a single motion-blurred image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6334–6342.

[19] K. Purohit, A. Shah, and A. Rajagopalan, "Bringing alive blurred moments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6830–6839.

[20] H. Chen, J. Gu, O. Gallo, M.-Y. Liu, A. Veeraraghavan, and J. Kautz, "Reblur2deblur: Deblurring videos via self-supervised learning," in *Proc. IEEE Int. Conf. Comput. Photography*. IEEE, 2018, pp. 1–9.

[21] P. Liu, J. Janai, M. Pollefeys, T. Sattler, and A. Geiger, "Self-supervised linear motion deblurring," *IEEE Trans. Robot. Autom.*, vol. 5, no. 2, pp. 2475–2482, 2020.

[22] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8174–8182.

[23] J. Qiu, X. Wang, S. J. Maybank, and D. Tao, "World from blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8493–8504.

[24] L. Chen, F. Fang, T. Wang, and G. Zhang, "Blind image deblurring with local maximum gradient prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1742–1750.

[25] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, 2008.

[26] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, p. 145, 2009.

[27] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2010, pp. 157–170.

[28] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "*l*_0-regularized intensity and gradient prior for deblurring text images and beyond," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 39, no. 2, pp. 342–355, 2016.

[29] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Deblurring images via dark channel prior," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 40, no. 10, pp. 2315–2328, 2017.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3116135

[30] M. Li, Y. Li, Y. Tian, L. Jiang, and Q. Xu, "Appealnet: An efficient and highly-accurate edge/cloud collaborative architecture for dnn inference," *arXiv preprint arXiv:2105.04104*, 2021.

[31] S. K. Nayar and M. Ben-Ezra, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 26, no. 6, pp. 689–698, 2004.

[32] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 110, no. 2, pp. 185–201, 2014.

[33] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, and M.-H. Yang, "Soft-segmentation guided object motion deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 459–468.

[34] J. Shi, L. Xu, and J. Jia, "Discriminative blur detection features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2965–2972.

[35] C. Khare and K. K. Nagwanshi, "Image restoration in neural network domain using back propagation network approach," *Image*, vol. 2, no. 5, 2011.

[36] I. Aizenberg, D. Paliy, C. Moraga, and J. Astola, "Blur identification using neural network for image restoration," in *Comput. Intell., Theory Appl.* Springer, 2006, pp. 441–455.

[37] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 38, no. 7, pp. 1439–1451, 2015.

[38] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 221–235.

[39] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8183–8192.

[40] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2521–2529.

[41] S. Nah, S. Son, and K. M. Lee, "Recurrent neural networks with intra-frame iterations for video deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8102–8111.

[42] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 0–0.

[43] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 921–934, 2019.

[44] M. Suin, K. Purohit, and A. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3606–3615.

[45] Y. Yuan, W. Su, and D. Ma, "Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3555–3564.

[46] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman, "Deep generative filter for motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2993–3000.

[47] K. Zhang, W. Luo, B. Stenger, W. Ren, L. Ma, and H. Li, "Every moment matters: Detail-aware networks to bring a blurry image alive," in *Proc. ACM Int. Conf. Multimed.*, 2020, pp. 384–392.

[48] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, 2016.

[49] W. Ren, J. Pan, X. Cao, and M.-H. Yang, "Video deblurring via semantic segmentation and pixel-wise non-linear kernel," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1077–1085.

[50] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 1645–1654.

[51] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[53] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[54] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach Intell.*, vol. 33, no. 5, pp. 898–916, 2010.

[55] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1107–1114.

[56] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8878–8887.

[57] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8934–8943.

[58] J. Shi *et al.*, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 1994, pp. 593–600.

Youjian Zhang received the BE degree in electronics science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently working toward the PhD degree in computer science from the University of Sydney, Camperdown, NSW, Australia. His research interests are computer vision with emphases on image deblurring, image/video quality enhancement and restoration. Recently, one of his works has been accepted by the Conference on Neural Information Processing Systems (NeurIPS).

Chaoyue Wang is a postdoctoral researcher in Machine Learning and Computer Vision at the School of Computer Science, The University of Sydney. He received a bachelor degree from Tianjin University (TJU), China, and a Ph.D. degree from the University of Technology Sydney (UTS), Australia. His research outcomes have been published in prestigious journals and prominent conferences, such as IEEE T-EVC, IEEE T-IP, NeurIPS, CVPR, IJCAI. He received the Distinguished Student Paper Award in the 2017 International Joint Conference on Artificial Intelligence (IJCAI-17).

Stephen J. Maybank (Fellow, IEEE) received the BA degree in mathematics from King's CollegeCambridge, in 1976 and the PhD degree in computer science from Birkbeck College, University of London, in 1988. He is currently a professor with the School of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, *etc.* He is a fellow of the IEEE and fellow of the Royal Statistical Society.

Dacheng Tao (Fellow, IEEE) is an advisor and chief scientist of the digital science institute in the University of Sydney. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences. He received the 2015 and 2020 Australian Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award. He is a Fellow of the Australian Academy of Science, AAAS, ACM and IEEE.