RESTORATION OF VIDEO FRAMES FROM A SINGLE BLURRED IMAGE WITH MOTION UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a novel framework to generate clean video frames from a single motion-blurred image. While a broad range of literature focuses on recovering a single image from a blurred image, in this work, we tackle a more challenging task *i.e.*video restoration from a blurred image. We formulate video restoration from a single blurred image as an inverse problem by setting clean image sequence and their respective motion as latent factors, and the blurred image as an observation. Our framework is based on an encoder-decoder structure with spatial transformer network modules to restore a video sequence and its underlying motion in an end-to-end manner. We design a loss function and regularizers with complementary properties to stabilize the training and analyze variant models of the proposed network. The effectiveness and transferability of our network are highlighted through a large set of experiments on two different types of datasets: camera rotation blurs generated from panorama scenes and dynamic motion blurs in high speed videos. Our code and models will be publicly available.

1 INTRODUCTION

Capturing an image is not an instant process; to capture enough photons, the photosensitive elements of a camera have to be exposed to light for a certain interval of time, called exposure time. Therefore, during this interval if an object is moving in the observed scene or the camera is undergoing an arbitrary motion, the resulting image will contain a blurring artifact known as *motion blur*. In general, motion blur is an unwanted behaviour in vision applications *e.g.*image editing (Gunturk & Li, 2012), visual SLAM (Lee et al., 2011) and 3D reconstruction (Seok Lee & Mu Lee, 2013), as it degrades the visual quality of images. To cope with this type of artifact, image deblurring aims to restore a sharp image from a blurred image. This problem is known to be ill-posed since the blur kernel used for deconvolution is generally assumed to be unknown.

Earlier studies assume a uniform-blur over the image to simplify the estimation of the single deconvolution blur kernel used to remove the blur (Fergus et al., 2006; Cho & Lee, 2009; Levin et al., 2009). Even though the methods deploy deblurring tasks with uniform-blur assumption, the assumption is often violated in practice. For instance, when the blur is caused by out-of-plane camera rotation, the blur pattern becomes spatially variant. Moreover, the problem is more complex when objects in a scene are moving *i.e.*dynamic blur. While previous literature focuses on recovering a sharp image from a blurred image, we tackle a more challenging task *i.e.*video restoration from a blurred image. Restoring the underlying image sequence of a blurred image requires both contents and motion prediction. We formulate video restoration from a blurred image as an inverse problem where a clean sequence of images and their motion as latent factors, and a blurred image as an observation. Some of previous deblurring approaches (Hyun Kim & Mu Lee, 2014; Zhang & Yang, 2015; Sellent et al., 2016; Ren et al., 2017; Park & Mu Lee, 2017) also estimate the underlying motion in a blurred image, however, their goal remains in single frame restoration. Recently Jin et al. (2018) proposed to extract video frames from a single motion-blurred image. Their approach is close to image translation model without inferring underlying motions between the latent frames. Purohit et al. (2019) addressed this issue by estimating pixel level motion from a given blurred input. However, their model is still prone to sequential error propagation as frames are predicted in a sequential manner using a deblurred middle frame.

In this paper, we propose a novel framework to generate a clean sequence of images from a single motion-blurred image. Our framework is based on a single encoder-decoder structure with Spatial Transformer Network modules (STN) and Local Warping layers (LW) to restore an image sequence and its underlying motion. Specifically, a single encoder is used to extract intermediate features which are passed to multiple decoders with predicted motion from STN and LW modules to generate a sequence of deblurred images.

We evaluate our model on two types of motion blur. For rotation blur, which is caused by abrupt camera motion, we generated a synthetic dataset from panoramic images (J. Xiao & Torralba., 2012). For dynamic blur caused by fast moving objects in a scene, we used a high speed video dataset (Nah et al., 2017). The proposed model is evaluated on the panorama and the high speed video datasets under various motion patterns. Both the quantitative metrics and qualitative results highlight that our method is more robust and performs favorably against the competing approach (Jin et al., 2018) [1]. For further investigation, we demonstrate the transferability of our model by cross-dataset evaluation. We also propose a simpler and lighter variation of our model guiding that our approach is flexible and can be easily extended to arbitrary number of frame prediction model with negligible performance trade-off.

In short, our contributions are as follows. 1) We propose a novel unified architecture to restore clean video frames from a single motion-blurred image in an end-to-end manner. 2) Loss terms are designed to stably train the proposed network. 3) We perform thorough experiments to analyze the transferability and flexibility of the proposed architecture. 4) The performance of our model quantitatively and qualitatively performs favorably against the competing approach. Moreover due to flexibility of our model, we show that our approach is robust to heavy blurs where the previous approach fails.

2 Related Works

2.1 Image deblurring

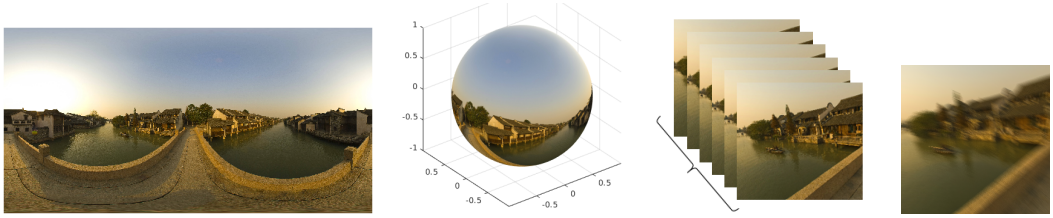
Image deblurring in general refers to the restoration of an image affected by blur. In this paper, we focus exclusively on the motion blur. Image deblurring is an ill-posed inverse problem when a blur kernel is unknown *i.e.*blind deconvolution problem, as different latent images can be transformed to a blurred image depending on its blur kernel. Early stage of deblurring studies (Cho & Lee, 2009; Fergus et al., 2006; Pan et al., 2014; Michaeli & Irani, 2014; Pan et al., 2016; Chakrabarti, 2016; Dong et al., 2017; Yan et al., 2017) assume a single blur kernel that is applied to an image globally. The restoration of blur images is often modeled as a maximization problem of probabilistic models (Cho & Lee, 2009; Fergus et al., 2006). To narrow down the ambiguity of the blur kernel estimation, natural image priors (Michaeli & Irani, 2014; Pan et al., 2014; 2016; Yan et al., 2017) are exploited. For instance, Michaeli & Irani (2014) formulate the blur kernel estimation as a process to recover internal recurrence of image patches. $\ell 0$ regularization (Pan et al., 2014), dark channel prior (Pan et al., 2016) and extreme channel prior (Yan et al., 2017) are also used to improve image deblurring.

While single blur kernel estimation approaches are effective when blur kernels are shift-invariant, they fail when the blur is not spatially uniform. A non-uniform blur can be caused by camera rotation, depth variation or moving objects in a scene. To restore images affected by motion blur from pure rotations, Dong et al. (2017) use the geometric information of the camera motion as a prior to recover the non-uniform blur model. Recently, deep network based methods (Nah et al., 2017; Zhang et al., 2018) are proposed to handle general blur patterns without the uniform blur assumption. Nah et al. (2017) propose multi-scale deep networks with multi-scale loss that mimics coarse-to-fine approaches to restore sharp images under non-uniform blurred images. Zhang et al. (2018) proposed a spatially variant neural networks to learn spatially variant kernels.

2.2 Sequence restoration from a blurred image

Recently, Jin et al. (2018) proposed to extract a video sequence from a single motion-blurred image using multiple deep networks. They showed that deep networks can successfully generate an image sequence from a blurred image, however there remains a few limitations. Their proposed framework

[1] We could not compare with Purohit et al. (2019) since they did not open source their code.



(a) (b) (c) (d)

Figure 1: Rotational blur dataset generation. (a) input panorama image, (b) panorama projection on a unit sphere , (c) intermediate frames between the initial and final images (d) blurred image obtained by averaging the captured frames.

consists of multiple networks of which each network is specialized to predict a specific frame in a sequence. Each network is trained separately and sequentially starting from the middle frame and then adjacent frames taking previously predicted frames as inputs. As a result, the non-middle frame prediction heavily relies on previously predicted frames including the middle frame itself, therefore when the middle frame is erroneous the error propagates across frames.

Purohit et al. (2019) proposed a two-step strategy to generate a video from a motion-blurred image using three complementary networks. They used video autoencoder to learn motion and frame generation from clean frames as a pretraining phase. Latter, they introduced a motion disentangle network to extract motion from blurred image. They also used independent deblurring network as their approach requires a clean middle frame generated from a blurred image in advance. Although their approach takes motion information into account, the approach generates frames sequentially starting from the middle frame to adjacent frames which results in error propagation just as in Jin et al. (2018). Unlike the previous works, our approach runs in an end-to-end manner within a single training stage without error propagation across frames.

3 Dataset Generation

Collecting a large number of natural motion-blurred images is a daunting task. Hence, a common practice in computer vision research is to generate blurry images by combining a sequence of sharp images using various approaches ranging from simple averaging (Nah et al., 2017; Jin et al., 2018) to learnable methods (Brooks & Barron, 2019). The source of motion blur in an image can be generalized into two main categories: rapid camera motion (camera shake) and dynamic motion of objects in the scene. In this section, we briefly explain how we generate a blurry image dataset by considering each case individually.

3.1 Rotational blur (synthetic)

In order to generate a rotation blurred image dataset, we use the SUN360 panorama dataset (J. Xiao & Torralba., 2012). This dataset provides various panoramas with 360° field of view. Hence, a virtual camera can be modeled to point at different orientations to represent the camera rotation in $SO(3)$. Given a panorama P of size $H \times W$, we developed a simple yet effective framework to generate blurred images. First, the panorama is projected onto a unit sphere by linearly mapping each pixel coordinate $(x, y) \in P$ into spherical coordinates (θ, ϕ) with $\theta \in (0, 2\pi)$ and $\phi \in (-\pi/2, \pi/2)$. Then, a synthetic image can be captured via a virtual camera by re-projecting the 3D points on the sphere into an image plane as briefly discussed in Mei & Rives (2007) and Oleksandr et al. (2018).

Using this procedure we first capture an image by positioning the virtual camera at an arbitrary orientation. We call the image generated at this orientation *initial image*. Then, we rotate the camera by a random rotation matrix (with $\beta = (\beta_x, \beta_y, \beta_z)$ its Euler angle representation) and capture a second image at the new camera position called *final image*. We finally use a quaternion spherical linear interpolation technique (Slerp) (Dam et al., 1998) to capture intermediate frames between the initial and final images. All the resulting images (initial, final and intermediate frames) are averaged to generate a blurry image. The dataset generation process is summarized in Fig. 1.

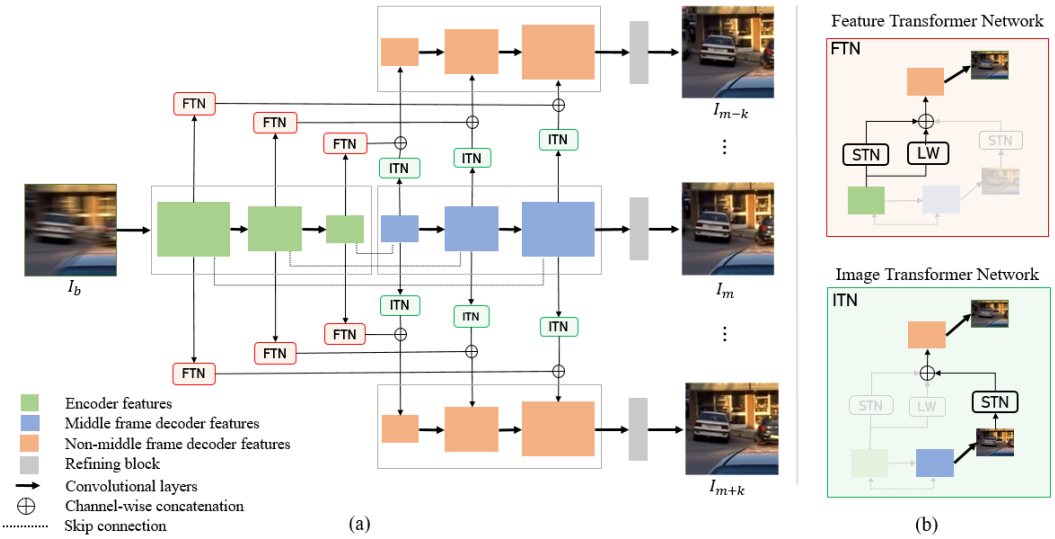


Figure 2: Overview of our network. (a) The middle frame is predicted using an encoder-decoder structure. The non-middle frames are reconstructed by transforming the multi-layer features of the middle frame. (b) Feature transformer network (FTN) transforms features locally via local warping (LW) and globally via spatial transformer network (STN). Image transformer network (ITN) transforms predicted middle frame via STN. Finally, the predicted frames are passed through a refining network.

The camera rotation angle is uniformly sampled from $[-10°, 10°]$. In order to generate a realistic blurred image, the number of intermediate images have to be adjusted automatically depending upon the rotation magnitude between the initial and final frames. Therefore, we use a simple linear relationship between the number of frames to be generated (n) and the rotation magnitude as follows: $n = c + \frac{1}{3}\|\beta\|$, where c is a constant and $\|\beta\|$ is the magnitude of β. In this manner, we use 1000 panoramic images from which we generate $26,000$ training and $3,200$ test images of size 128×128px.

3.2 Dynamic motion (real)

In order to generate more realistic and generic (arbitrary camera motions and dynamic scene) blurred images, we take advantage of a GoPro high speed video dataset (Nah et al., 2017). This dataset provides 22 training and 11 test scenes, each scene containing frames of size 1280×720px. A blurry image is generated by averaging n consecutive frames (Nah et al., 2017; Jin et al., 2018). In our experiments, we fixed $n = 7$ and generated $20,000$ training and 2000 test images by randomly cropping images of size 256×256px.

4 Method

In this section, we describe our network structure and loss functions. To explicitly take the camera motion into consideration, our network is designed with spatial transformer networks (STNs) (Jaderberg et al., 2015) in an encoder-decoder structure (Fig. 2). Given a blurry image I_b as an input, our model outputs $\{I_1, ..., I_{m-1}, I_m, I_{m+1}, ..., I_n\}$, where I_m is the deblurred middle frame and $\{I_j\}_{j=1}^{n}$ where $j \neq m$ are the recovered non-middle frames. To ensure sharp image generation and stable training, the network is trained using three loss terms: multi-scale photometric loss, transformation consistency loss and penalty term.

4.1 Middle frame

The middle frame I_m is reconstructed using a U-net (Ronneberger et al., 2015) like network. The *encoder* contains five convolutional blocks, each block containing two layers of convolutions with

spatial kernel size of 3×3 and stride size of 2 and 1, respectively. The feature maps are downsampled to half size and the number of channels is doubled after each convolutional block. The *decoder* network also contains five convolutional blocks to upsample features and to predict images at different scales. In each block, a feature is first upscaled using a transposed convolution (deconvolution) layer of kernel size 4×4 and a stride size of 2. The image predicted in the previous block is also upscaled in the same manner. The upsampled feature and its respective image are then concatenated channel-wise with the corresponding feature from the encoder (skip connection as shown in the Fig. 2a), then passed through five layers of convolutions with dense connections to output a feature, which will be used to predict an image at current block. In this manner, features and images are successively upsampled to predict a full scale middle frame. Along with the last feature map from the decoder, the predicted image is finally passed through a *refining* convolutional block. It contains seven dilated convolutional layers (Yu & Koltun, 2015) each with kernel size of 3×3 and different dilation constants. The purpose of this network is to further refine the predicted frame with contextual information by effectively enlarging the receptive field size of the network.

4.2 Non-middle frame

The non-middle frames are reconstructed based on the encoded features of the middle frame via learned transformation by STN modules and *local warping* (LW) networks. First, the encoded middle frame feature $U_e \in \mathbb{R}^{H \times W \times C}$ with width W, height H and C channels from the encoder is transformed into a non-middle frame feature using a *feature transformer network* (FTN). Second, the decoded middle frame image $I_d \in \mathbb{R}^{H \times W \times 3}$ predicted from the corresponding middle frame decoder is transformed using an *image transformer network* (ITN). Third, the transformed feature and image are concatenated channel-wise and are passed through a decoder to predict a non-middle frame (Fig. 2b). We also input the middle frame feature into the non-middle frame decoder in order to guide the decoder to learn the spatial relation between the middle and the non-middle frame. The decoder network here is similar to the one used for predicting a middle frame in the previous section. These mechanisms are summarized in the following equations,

$$U_t^l = \text{STN}^l(U_e^l) \oplus \text{LW}^l(U_e^l), \qquad I_t^l = \text{STN}(I_d^l), \tag{1}$$

$$I_p^l = \mathcal{D}^l(U_t^l \oplus I_t^l \oplus U_e^l) \tag{2}$$

where $l = \{1, \ \dots, k\}$ is an index for k feature levels (scales) and t is a subscript for transformed feature/image. I_p is the predicted frame from a non-middle frame decoder $\mathcal{D}$.

Each non-middle frame is reconstructed by applying multi-scale transformer networks to the middle frame encoder. Given ground truth non-middle frames during training, our model learns the transformation parameters to be applied to the middle frame at different scales in order to output the desired non-middle frames. The fact that unique transformer networks are applied at each feature and image scale gives the model a capacity to learn various types of transformations, hence, making it robust to different blur patterns including large blurs.

4.3 Transformer networks

STNs in our model learn non-local transformations in a given motion-blurred image. In order to compensate for locally variant transformations, we designed a *local warping* network. This network is conditioned on the input feature like STN, however, instead of predicting global transformation parameters, it predicts pixel-wise displacement *i.e.motion flow*. Given an input feature $U \in \mathbb{R}^{H \times W \times C}$, the local warping network outputs a motion flow of size $H \times W \times 2$. We used two convolutional layers of kernel size 3×3 for the network. By warping the input feature with the predicted motion flow, we obtain a locally transformed feature which is used as an input to the decoder (Fig. 2b).

4.4 Loss function

4.4.1 Multi-scale photometric loss

Given a blurry input, in order to generate video frames that are sharp both locally and globally, we trained our network with a multi-scale photometric loss between the images predicted by the decoder

network and the ground truth image. A bilinear downsampling is used to resize the ground truth image to the corresponding predicted frame size at different scales. Let l denotes a scale level, k denotes total number of scales, $\{\hat{y}\}_{l=1}^{k}$ denotes a set of predicted images from the smallest size ($\hat{y}_1$) to the full scale ($\hat{y}_k$), and $\{y\}_{l=1}^{k}$ represent a set of downsampled ground truth images where y_k is a full scale ground truth image. For training a model predicting a sequence with n frames from a single blurry image, we compute multi-scale photometric loss as the term $\mathcal{L}_{mp}$ in Eq. (3),

4.4.2 TRANSFORMATION CONSISTENCY LOSS

We use individual transformer networks for each feature level when predicting non-middle frames. This augments our model with the capacity to learn transformations at different levels making it robust to various blur patterns. However, we expect the transformations at different scales to be aligned for successfully reconstructing temporally consistent non-middle frames. Especially at the initial stages of the training where the transformer parameters are random, it is beneficial that our model understands the relationship between the transformations across different frame levels. In order to impose this notion into our model and facilitate a smooth training, we propose the *transformation consistency loss*. Let $\{\boldsymbol{\theta}\}_{l=1}^{k}$ be the set of predicted transformation parameters at different scales. The *transformation consistency loss* for predicting $n-1$ non-middle frames can be defined as the term $\mathcal{L}_{tc}$ in Eq. (3), where $|.|_2$ is an $\ell 2$ loss between the transformation parameters.

4.4.3 PENALTY TERM

Predicting multiple frames from a single blurry image can be problematic at times when the model fails to learn any type of transformation and simply replicates the middle frame predicition as non-middle frames. In order to remedy this issue, we design a penalty term to enforce diversity among generated images. This is accomplished by explicitly maximizing the sum of absolute difference (SAD) *i.e.* minimizing the negative SAD between a predicted frame and its time-symmetric (about the middle frame) ground truth frame. For example, when predicting seven frames $\{I_1, ..., I_4, ..., I_7\}$, we enforce the predicted image I_1 to be different content-wise from the ground truth image I_7 and vice versa. The penalty is imposed in a symmetric manner such that the model learns to be sensitive to smaller transformations close to the middle frame as well as larger transformations at the end frames. Given a predicted frame $\hat{y}_i$ and the corresponding time-symmetric ground truth y_{n+1-i}, the penalty term is computed as the term $\mathcal{L}_p$ in Eq. (3), where m is the middle frame index, n is the total number of frames.

$$\mathcal{L}_{mp} = \sum_{j=1}^{n}\sum_{l=1}^{k} \mathbf{w}_l \cdot \left|\mathbf{y}_{j,l} - \hat{\mathbf{y}}_{j,l}\right|_1, \quad \mathcal{L}_{tc} = \sum_{j=1}^{n-1}\sum_{l=2}^{k} \left|\boldsymbol{\theta}_{j,l} - \boldsymbol{\theta}_{j,l-1}\right|_2, \quad \mathcal{L}_{p} = -\sum_{j=1, j\neq m}^{n} \left|y_{n+1-j} - \hat{y}_j\right|_1 \tag{3}$$

The final training loss function is defined as follows,

$$\mathcal{L} = \mathcal{L}_{mp} + \lambda_{tc}\mathcal{L}_{tc} + \lambda_p\mathcal{L}_p, \tag{4}$$

where λ_{tc} and λ_p are weight coefficients for *transformation consistency loss* and *penalty term*, respectively. We used $\lambda_{tc} = 1e+3$ and $\lambda_p = 0.5$.

4.5 TEMPORAL AMBIGUITY AND NETWORK TRAINING

The task at hand has two main ambiguities. i. *temporal shuffling* and ii. *reverse ordering*. As explained in section 3, motion-blur is the result of an averaging process and, restoring temporally consistent (no shuffling) sharp frame sequence from a given motion-blurred input is a non-trivial task as the averaging destroys the temporal order. Jin et al. (2018) mentions that photometric loss is not a sufficient constraint to make their network converge. Hence, they propose a pair-wise order invariant loss to train their network. Purohit et al. (2019) also uses the same loss function to fine-tune the recurrent video decoder in their network.

We find experimentally that a multi-scale photometric loss is a sufficient constraint to train our network. We further impose more constraints using other loss terms to improve performance (see ablation studies in Sec. 5.4.2). By design nature, our model allows motions to be learned in a symmetric manner (about the middle frame) with transformer networks close to the middle frame decoding smaller motions and those further from the middle frame decoding larger motions. This

Table 1: Quantitative results on initial (F_i), middle (F_m) and final (F_f) frames.

		High speed video			Panorama blur		
		F_i	F_m	F_f	F_i	F_m	F_f
PSNR (dB)	Jin et al. (2018)	23.713	29.473	23.681	-	-	-
	Ours	**27.357**	**31.989**	**27.414**	23.693	23.874	24.049
SSIM	Jin et al. (2018)	0.660	0.846	0.659	-	-	-
	Ours	**0.794**	**0.885**	**0.793**	0.699	0.704	0.716

notion is enforced by transformation consistency loss and symmetric constraint term during training. The fact that our model is optimized in a joint manner allows frames to be reconstructed in a motion-guided sequence.

Other than *temporal shuffling*, another issue is *reverse ordering*. Given a single motion-blurred input, recovering ground truth order is a highly ill-posed problem which is intractable since reversely ordered frames result in the same motion-blurred image. Neither our work nor previous works ((Jin et al., 2018), Purohit et al. (2019)) are capable of predicting the right order. Hence, we evaluate frame reconstructions using both ground truth order and its reverse order, then report the higher metric in the experiment section.

5 Experiment

5.1 Implementation details

Our model is implemented and trained using pyTorch (Paszke et al., 2017). We chose Adam (Kingma & Ba, 2015) as an optimizer with β_1 and β_2 fixed to 0.9 and 0.999, respectively. On our synthetic blur dataset, we train the model using images of size 128×128px and a mini-batch size of 8 to predict initial, middle and final frames. A mini-batch size of 4 and input size of 256×256px is used to predict sequences of frames when training on the high speed video dataset. In all experiments, we train our model for 80 epochs. We set the learning rate $\lambda = 1e-4$ at the start of the training and decay it by half at epochs 40 and 60. All the training and test images are cropped from the original resolution images without resizing.

5.2 Results

In this section, we analyze the performance of our model qualitatively and quantitatively on both camera shake blurs generated from panoramic scenes and dynamic blurs obtained from averaging frames in high speed videos.

5.2.1 Quantitative evaluation

We report test results using *Peak Signal-to-Noise Ratio* (PSNR) and *Structural Similarity* (SSIM) metrics. To purely evaluate the quality of generated images without ordering estimation issue due to *reverse ordering*, we report the higher PSNR/SSIM metric of either ground truth order or reverse order of frames *i.e.*$\max\{\text{PSNR/SSIM}(F_i \rightarrow F_f), \text{PSNR/SSIM}(F_f \rightarrow F_i)\}$. For dynamic blur, we compare our results with Jin et al. (2018)[2] as they also use the same high speed video dataset (Nah et al., 2017). The averaged metrics obtained on the testing set are summarized in Table 1. The results show that our model performs favorably against their model on both PSNR and SSIM on the high speed video dataset. While the middle frame prediction shows moderate performance increase, for the initial and final frame predictions, our model outperforms Jin et al. (2018) by a large margin. The performance gap between the middle frame and non-middle frames is relatively larger in Jin et al. (2018) than our method. This is due to sequential prediction in Jin et al. (2018) which makes non-middle frame prediction heavily dependent on the generated middle frame, resulting in error propagation. As stated in Jin et al. (2018), this limitation is particularly problematic when a heavy blur affects the input image since the middle frame prediction becomes less reliable. Our approach is

[2]Jin et al. (2018) used additional 20 scenes other than the high speed video dataset for training.

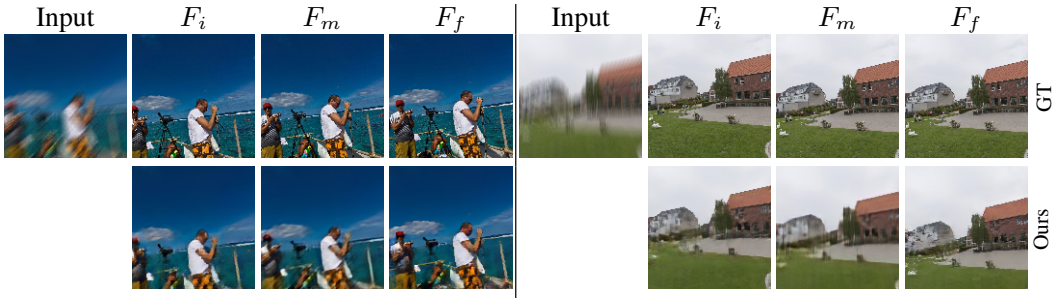


Figure 3: Rotation blurred images generated from panorama scenes. The top row is ground truth frames and the bottom row is restored frames from the blurs.

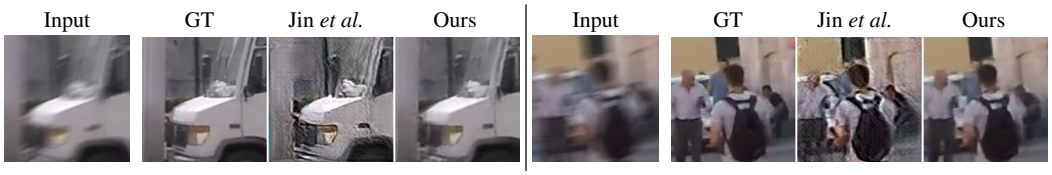


Figure 4: Heavily blurred (dynamic) inputs from the high speed videos and the restored video frames. Click on the images in *Adobe Reader* to play the videos.

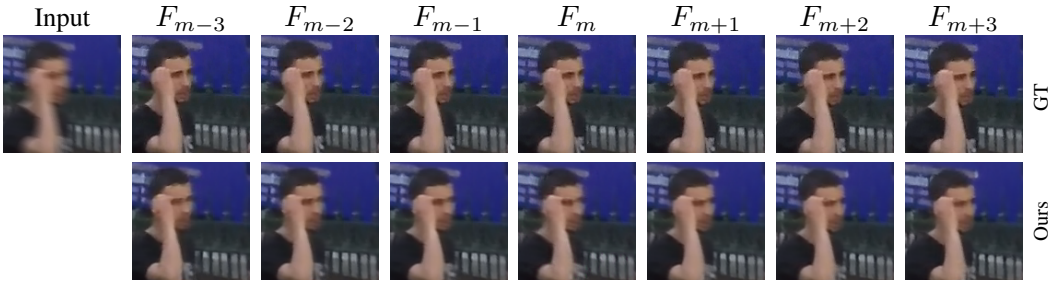


Figure 5: Partially blurred dynamic motion example. The man is blurred with dynamic motion while the background is close to static.

relatively robust to heavy blur as the proposed model generates frames independently from multiple decoders, therefore the error is not propagated (Fig. 7).

We also evaluate our model on camera rotational blurs (Table 1 Panorama blur). In terms of blurriness, the panorama scenario is more challenging as we simulate broader range of blurs than the blurs in the high speed video dataset *i.e.*from sharp images to heavy blurred images. We observed visually better results for the panorama scenario but lower quantitative number. One main reason is that panorama ground truths are relatively sharper while the high speed video contains blurry ground truth images due to dynamic motion and short exposure time. Therefore, the visual results of panorama scenes from our model are sharp even though the quantitative performance is relatively lower. Please refer to Sec. 7.1 in the appendix for the quantitative results on seven frame predictions.

5.2.2 Qualitative evaluation

The qualitative results for panoramic scenes and high speed videos show that our model can successfully restore multiple frames from a blurred input under various blur patterns. The restored frames from a blurred image in the panorama scenario (Fig. 3) are relatively sharper than dynamic blur case. As mentioned earlier, one of reasons is due to high quality ground truth image. In the high speed video scenario (Fig. 4), ground truth images have locally blurred contents due to fast dynamic motions. We compare our approach and previous method (Jin et al., 2018) on relatively heavily blurred images from the high speed video dataset. As can be seen from Fig. 4, our method reconstructs contents consistently across frames and restores visually sharper videos compared to Jin et al. (2018). In case of dynamic blurs where the motion of multiple objects and static objects are

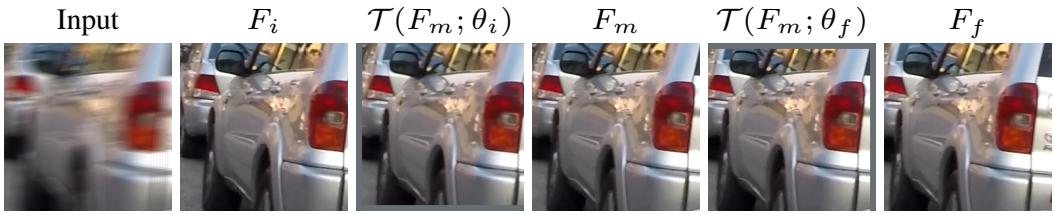


Figure 6: STN transformation visualization. The ground truth middle frame is transformed to non-middle frame using pure STN transformation.

mixed as shown in Fig. 5, our model can generate video frames unravelling the underlying spatially varying motion. Failure cases happen for undersampled and heavily blurred inputs as can be seen from Fig. 7 and Fig. 9d. Please refer to Sec. 8 in the appendix for more qualitative results.

5.3 Analysis

5.3.1 Visualization of STN transformation

In order to visually analyze if the STN module in our model reasonably infers the blur motion, we apply the STN transformation to the middle frame and compare it with the ground truth non-middle frames. Let i be the index of the initial frame, m the middle frame, and f the final frame. The input image is obtained by averaging a set of frames $\{F_j\}_{j=i}^{f}$. We can obtain the STN transformation parameters $\{\theta\}_{j=1,j\neq m}^{f}$ from the *feature transformer network* which transforms middle frame features to non-middle frame features. We apply the STN transformation $\mathcal{T}$ with the parameter θ_j to middle frame F_m to visualize whether the transformation implies valid motion information. Fig. 6 shows that the transformation spatially aligns contents of the middle frame to that of the non-middle frames.

5.3.2 Cross-dataset evaluation

We report a cross-dataset *panorama→high speed video* evaluation to assess the generalization capability of our model. A model trained on the panoramic scenes is evaluated on high speed video test set (Table 2). Despite a performance degradation, our model trained on the panorama dataset performs on par with the competing approach (Jin et al., 2018) trained on the high speed video dataset. The absence of dynamic motion on the panorama dataset, which is apparent in high speed videos, can be one contributing factor explaining the performance loss in addition to the domain gap *e.g.*image contents, sharpness, blurriness.

5.4 Ablation studies

5.4.1 Lighter model

Model size can be a bottleneck when the number of frames in a video to be predicted increases since our model uses multiple decoders and transformer networks. Hence, we experiment with a lighter model by replacing decoders and STN modules to weight shared layers. As opposed to using individual decoders for each non-middle frame, we use a single decoder *i.e.*weight shared decoders. For STN, we apply inverse transformations by assuming symmetric motion about the middle frame, therefore, reducing the number of transformer networks by half. We tested this light architecture on a model that predicts three (initial, middle and final) frames and it reduces the number of model parameters by 48%, yet yielding only a 0.4dB performance decrease on average (Table 2).

5.4.2 Network Components and Loss terms

The *feature transformer network* (FTN) at different levels of abstraction is the core part of our model for network convergence. *Local warping* (LW) layer also significantly improves the performance of our model. The best model performance is, yet, achieved with the three network components (FTN, LW, ITN) combined (Table 3). As mentioned earlier, the multi-scale *photometric loss* (PML) is a sufficient constraint to make our network converge during training. We also experimentally find that a model trained with *transformation consistency loss* (TCL) not only converges faster with smoother

behavior but also gives a better performance during testing. The *penalty term* (PT) gives a marginal performance improvement when predicting fewer frames as photometric loss is already a sufficient constraint. In 3 frame prediction model, the penalty term improved performance marginally around 0.25dB while in 7 frame prediction model, it improved approximately 0.6dB. Penalty term enforces the model to consider subtle differences especially when the motion is small.

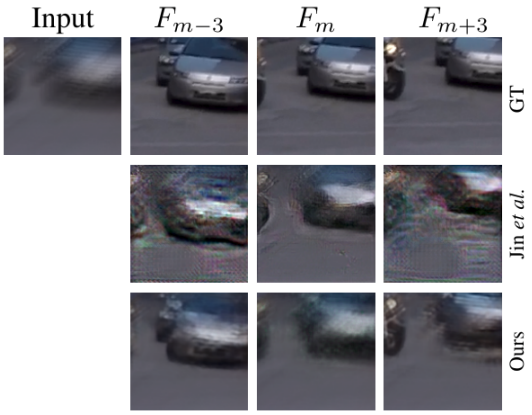


Figure 7: *Sequential error propagation.* Unlike Jin et al. (2018), our model can successfully recover non-middle frames even when the middle-frame prediction fails.

Table 2: Quantitative results for cross-dataset evaluation and single decoder model.

		F_i	F_m	F_f
Cross-dataset	PSNR (dB)	23.383	30.300	23.380
	SSIM	0.649	0.832	0.651
Lighter model	PSNR (dB)	27.120	31.322	27.151
	SSIM	0.762	0.863	0.761

Table 3: Ablation studies for network components and loss terms on PSNR metric.

		F_i	F_m	F_f
Network components	FTN	25.865	31.202	25.788
	FTN + LW	26.678	32.023	26.585
	FTN + ITN	26.065	31.785	26.058
	FTN + ITN + LW	27.357	31.989	27.414
Loss terms	PML	25.983	30.771	25.975
	PML + TCL	27.085	31.789	27.124
	PML + TCL + PT	27.357	31.989	27.414

6 Conclusion

We present a novel unified architecture that restores video frames from a single blurred image in an end-to-end manner. During training, *feature* and *image transformer networks* indirectly learn blur motions without motion supervision. The designed a loss function with regularizers enforce the model to consider subtle differences with fast loss convergence. We evaluate our model on the two datasets with rotation blurs and dynamic blurs and demonstrate qualitatively and quantitatively favorable performance against the competing approach. The cross-dataset evaluation demonstrates that our model can generalize even when the training and test set have significantly different blur patterns and domain gap. We additionally propose a lighter version of the model by weight-sharing of the decoders and STN modules under symmetric frame motion assumption. This modification enables the model to have negligible parameter size increment even when the number of predicted frames are high. Unlike the previous approaches, our model predicts frames in a single step without middle frame dependency. It is advantageous not only because it is simple to use but also robust to heavy blurs where middle frame prediction often fails. Overall, the simplicity and flexibility of our method makes it a promising approach for future applications such as deblurring, temporal super resolution and flow estimation from a motion-blurred image.

References

Tim Brooks and Jonathan T. Barron. Learning to synthesize motion blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, 2016.

Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 28(5):145, 2009.

Erik B. Dam, Martin Koch, and Martin Lillholm. Quaternions, interpolation and animation. In *Technical Report DIKU-TR-98/5*, pp. 42–48, 1998.

Jiangxin Dong, Jinshan Pan, Zhixun Su, and Ming-Hsuan Yang. Blind image deblurring with outlier handling. In *IEEE International Conference on Computer Vision*, 2017.

Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 25(3):787–794, 2006.

Bahadir Kursat Gunturk and Xin Li. *Image restoration: fundamentals and advances*. CRC Press, 2012.

Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

A. Oliva J. Xiao, K. A. Ehinger and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.

Jiaya Jia Jianping Shi, Li Xu. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Hee Seok Lee, Junghyun Kwon, and Kyoung Mu Lee. Simultaneous localization, mapping and deblurring. In *IEEE International Conference on Computer Vision*, 2011.

Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, 2014.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Bogdan Oleksandr, Eckstein Viktor, Rameau Francois, and Bazin Jean-Charles. Single view point omnidirectional camera calibration from planar grids. In *ACM SIGGRAPH European Conference on Visual Media Production*, 2018.

Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Haesol Park and Kyoung Mu Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *IEEE International Conference on Computer Vision*, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NIPS-W*, 2017.

Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6830–6839, 2019.

Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *IEEE International Conference on Computer Vision*, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

Anita Sellent, Carsten Rother, and Stefan Roth. Stereo video deblurring. In *European Conference on Computer Vision*, 2016.

Hee Seok Lee and Kuoung Mu Lee. Dense 3d reconstruction from severely blurred images using a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Haichao Zhang and Jianchao Yang. Intra-frame deblurring by leveraging inter-frame camera motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson W.H. Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

7 APPENDIX

Here, we present additional details pertaining to the experiments that could not be included in the main text due to space constraints.

7.1 QUANTITATIVE RESULTS FOR SEVEN FRAME PREDICTIONS

As can be inferred from Table 4, our method consistently performs favorably against the competing method Jin et al. (2018). The prediction performs best for middle frames and consistently decreases for non-middle frames on both our method and the competing method. The overall performance reported here is consistent with the results in the main paper.

Table 4: Seven frame prediction on the high speed video dataset (Nah et al., 2017) (128×128px.)

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
PSNR (dB)	Jin et al. (2018)	23.084	24.051	25.582	28.698	25.028	24.235	23.257
	Ours	**27.516**	**28.991**	**30.911**	**32.163**	**30.884**	**28.936**	**27.485**
SSIM	Jin et al. (2018)	0.588	0.643	0.733	0.820	0.715	0.652	0.617
	Ours	**0.767**	**0.804**	**0.843**	**0.861**	**0.842**	**0.802**	**0.765**

8 QUALITATIVE RESULTS FOR THE HIGH SPEED VIDEO DATASET

Here, we show qualitative results (frame-by-frame comparison) from the high speed video dataset (Nah et al., 2017) that could not be included in the main text due to space constraints. We also tested our model on motion-blurred examples from Jianping Shi (2014) and the restored videos are shown in Fig. 9.

8.1 QUANTITATIVE RESULTS FOR THE SUN360 PANORAMA DATASET

Here, we show qualitative results from the SUN360 panorama dataset (J. Xiao & Torralba., 2012) that could not be included in the main text due to space constraints. The samples contain various blur patterns *e.g.*static, partial blur, and heavy blur.

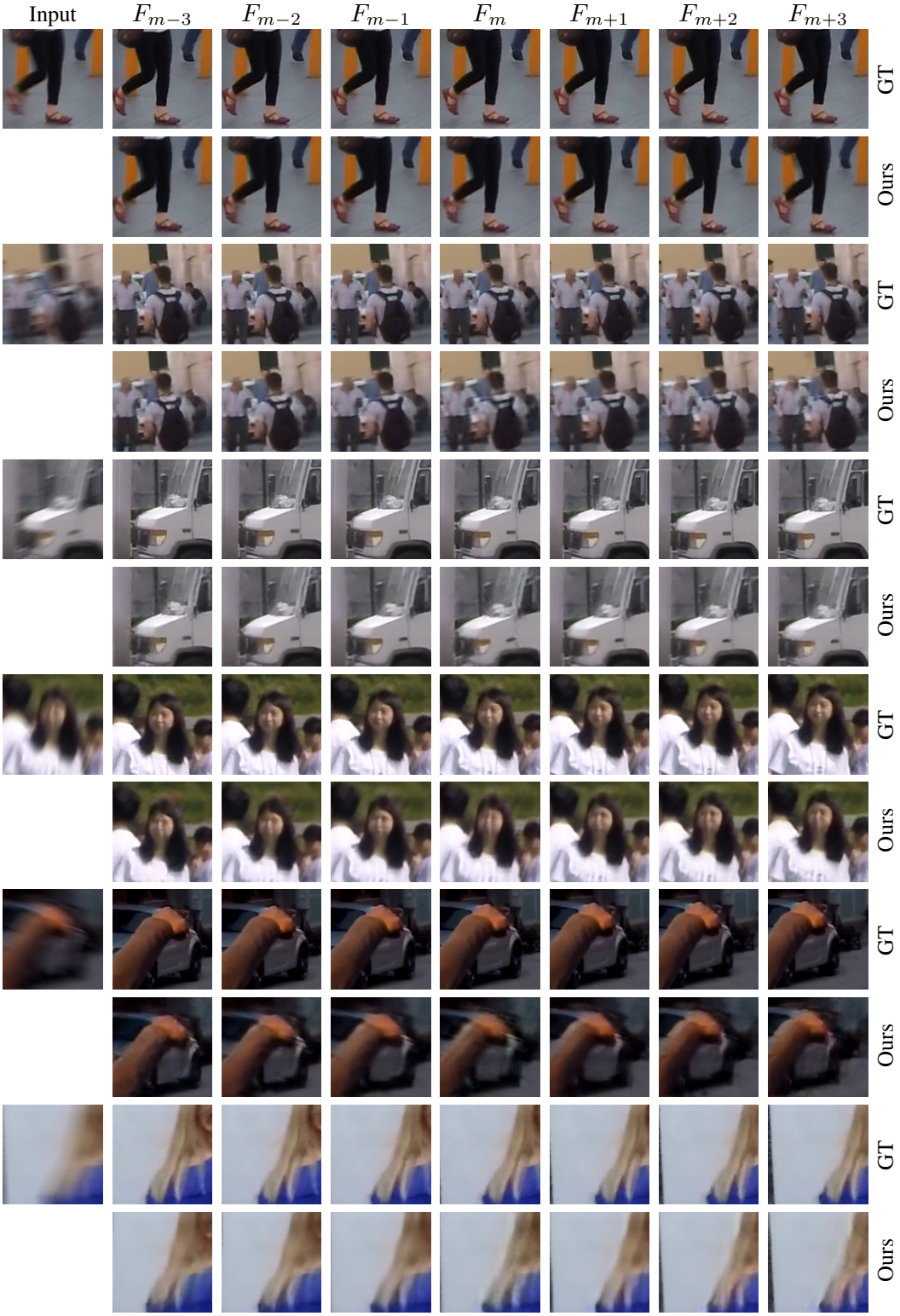


Figure 8: Dynamic blurred images from the high speed videos.

(a) (b) (c) (d)

Figure 9: Restored video frames for motion-blurred examples from Jianping Shi (2014). Click on the images in *Adobe Reader* to play the videos.

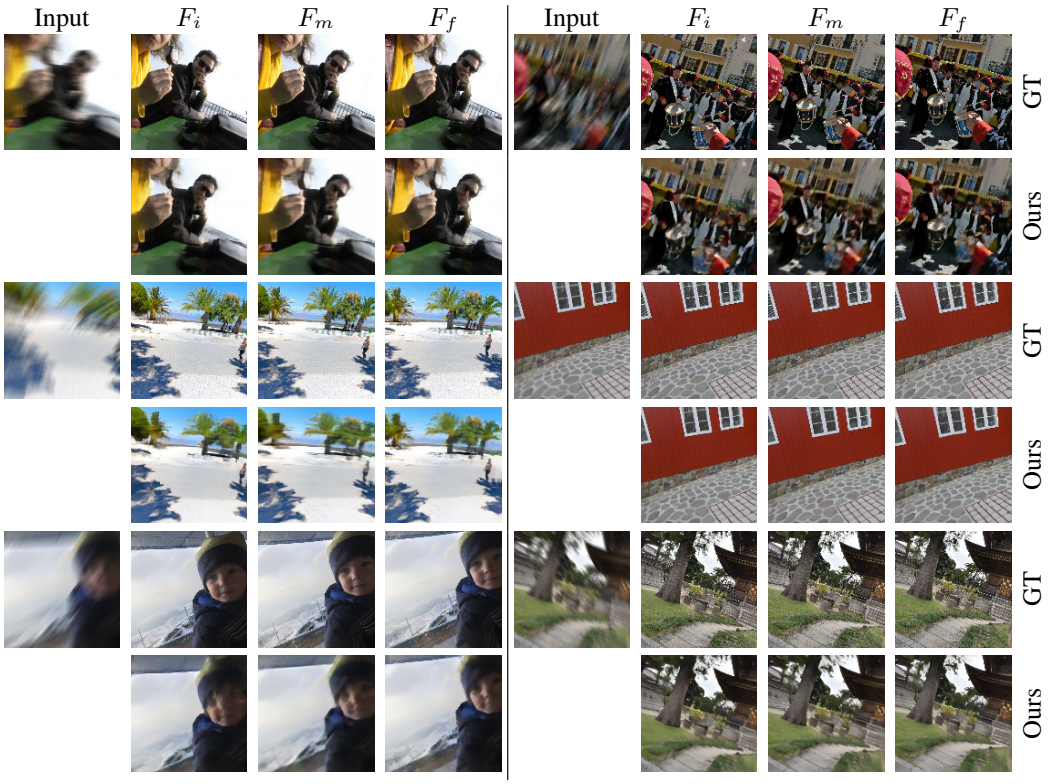


Figure 10: Rotation blurred images from the SUN360 panorama dataset.