Every Moment Matters: Detail-Aware Networks to Bring a Blurry Image Alive

Kaihao Zhang
Australian National University
super.khzhang@gmail.com

Wenhan Luo
Tencent AI Lab
whluo.china@gmail.com

Björn Stenger
Rakuten Institute of Technology
bjorn@cantab.net

Wenqi Ren
Institute of Information Engineering, Chinese Academy of Sciences
rwq.renwenqi@gmail.com

Lin Ma
Meituan-Dianping Group
forest.linma@gmail.com

Hongdong Li
Australian National University
hongdong.li@anu.edu.au

ABSTRACT

Motion-blurred images are the result of light accumulation over the period of camera exposure time, during which the camera and objects in the scene are in relative motion to each other. The inverse process of extracting an image sequence from a single motion-blurred image is an ill-posed vision problem. One key challenge is that the motions across frames are subtle, which makes the generating networks difficult to capture them and thus the recovery sequences lack motion details. In order to alleviate this problem, we propose a detail-aware network with three consecutive stages to improve the reconstruction quality by addressing specific aspects in the recovery process. The detail-aware network firstly models the dynamics using a cycle flow loss, resolving the temporal ambiguity of the reconstruction in the first stage. Then, a GramNet is proposed in the second stage to refine subtle motion between continuous frames using Gram matrices as motion representation. Finally, we introduce a HeptaGAN in the third stage to bridge the continuous and discrete nature of exposure time and recovered frames, respectively, in order to maintain rich detail. Experiments show that the proposed detail-aware networks produce sharp image sequences with rich details and subtle motion, outperforming the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**.

KEYWORDS

Deep blind image deblurring, motion blur, extract a sharp sequence.

ACM Reference Format:
Kaihao Zhang, Wenhan Luo, Björn Stenger, Wenqi Ren, Lin Ma, and Hongdong Li. 2020. Every Moment Matters: Detail-Aware Networks to Bring a Blurry Image Alive. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413929

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413929

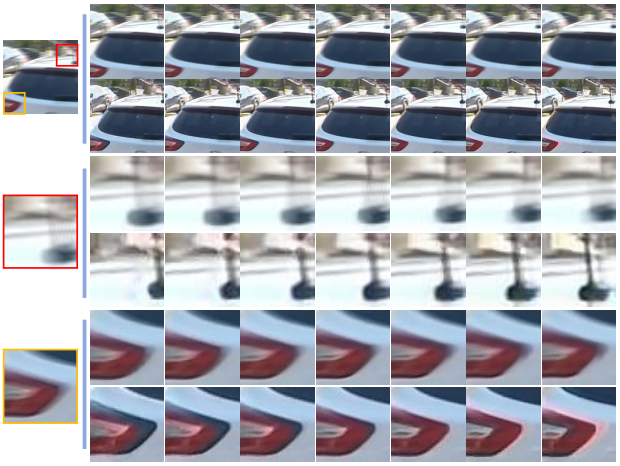

Figure 1: Video generation example. *The left column shows a blurry input image (top), and two zoomed-in regions. Rows to the right show frames extracted by the model released by Jin et al. [1] (top) and the proposed method (bottom), respectively. Our method recovers sharper detail (car antenna) and better preserves small motion (rear light).*

1 INTRODUCTION

Motion blur is a common artifact when taking photos and is caused by either camera shake [2–4] or object motion [5–8] during the exposure period within which light from the scene is accumulated [9, 10]. Observing a motion-blurred image, humans seem to be able to infer a plausible explanation of both the scene appearance and the underlying motion.

This paper aims to recover a temporal sequence of clean and sharp image frames from a single motion-blurred image, to mimic the above human ability. This task involves solving a severely under-constrained inverse problem, *i.e.*, to recovering multiple images from a single image which is the integration of the former. To some extent, the task is related to single image deblurring [11–13]. However, our task contains additional complexity, as we also want to get a set of temporally ordered sharp images that gave rise the single blurred version. This is particular challenging, since the image

integration operator is temporal-order invariant therefore multiple valid solutions exist. Moreover, besides multiple sharp frames, we also aim to recover the underlying motion across neighboring frames, yet often the motion is small between time-consecutive frames. For example, without modeling the subtle motion across frames, as shown in the fifth row of Fig. 1, the frames recovered by Jin *et al.* [1] look identical to each other. Finally, a motion-blurred image is generated during a *continuous* exposure period, yet one has to approximate this process by discretizing the time axis, leading to loss of information in image details.

To address the above challenges, we propose a generative model trained in three stages for video sequence extraction from a motion-blurred image. The first stage, called **BaseGAN**, learns to recover sharp video frames with a cycle flow loss to constrain that the motions across frames before and after the recovery are identical, thus resolving ambiguity in the recovery process. The second stage, **GramGAN**, is designed to recover subtle motions, employing a Gram matrix for motion feature representations. By minimizing the difference of Gram matrix description between the recovered frames and sharp frames, subtle motions are recovered. In the third stage, **HeptaGAN** training is carried out, taking multiple images, in our case seven, as input and generating the same number of output images. Specifically, we synthesize a motion-blurred image I_{blurry} from the seven input images $\{I_{in}\}$ and learn to recover a sequence of sharp images $\{I_{out}\}$ from the blurry image. The recovered frames $\{I_{out}\}$ are again used to produce a blurry image I'_{blurry}. The **HeptaGAN** model is optimized by not only forcing the recovered frames $\{I_{out}\}$ to be identical to the input sharp frames $\{I_{in}\}$, but also by minimizing the distance of the blurry images before and after the recovery procedure, I_{blurry} and I'_{blurry}, respectively. With this bi-cycle consistency, we minimize the disparity during the continuous-to-discrete transform.

Trained with this three-stage architecture, our generative model produces visually pleasing video frames given a motion-blurred image, as shown in Fig. 1. Different from existing approaches, which estimate frame sequences with multiple models, *e.g.*, [1], our method is able to extract multiple frames with a single model, which is more efficient and is better capable of exploiting spatio-temporal information. It is notable that the generators in different stages share weights.

Our main contributions are as follows:

- We develop a detail-aware networks to train a generative model, with each stage specifically addressing a challenge in the recovery of sharp video frames from a motion-blurred image.
- In order to model subtle movements across neighboring frames, GramGAN is proposed to produce frames preserving those subtle movements via Gram matrix.
- We propose a HeptaGAN module with bi-cycle consistency to minimize the lost information in the continuous-to-discrete transform, which helps not only to extract a sharp video sequence but also restore rich detail.
- Experiments show that our method achieves the state-of-the-art performance for producing a video sequence from a motion-blurred image, demonstrating its superiority over existing methods.

2 RELATED WORK

Our work in this paper is closely related to image deblurring, video deblurring and video generation, which are briefly introduced as follows, respectively.

2.1 Image Deblurring

CNN-based approaches have been successfully applied to various tasks of computer vision [14–18], including image deblurring [19, 20]. For non-blind deblurring, Xu *et al.* [21] introduced a CNN with two submodules for deconvolution, establishing a connection between traditional optimization schemes and CNN models. Sun *et al.* [22] employed a CNN to predict probabilistic blur kernels at patch level. An MRF was introduced to encourage motion smoothness, and blur was removed with a patch-level image prior. Schuler *et al.* [23] built a model for blind deblurring based on a traditional CNN, incorporating image deconvolution. More recently, Nah *et al.* [24] proposed a deep multi-scale CNN to remove complex motion blur based on content and adversarial loss, generating sharp images. Zhang *et al.* [8] introduced a framework to recover sharp images via learning how to make blurry images. Shen *et al.* [25, 26] develop two semantic face deblurring networks to remove blur from blurry facial images.

2.2 Video Deblurring

Video deblurring exploits the temporal dynamics implied from continuous image sequences [27–31], with applications of 3D reconstruction [32], SLAM [33], and object tracking [34]. Deep learning methods have been successfully applied to this problem. For example, Wieschollek *et al.* [35] built a recurrent network architecture to handle arbitrary spatial and temporal input sizes. Kim *et al.* [36] proposed a spatio-temporal recurrent method, which contains a dynamic temporal blending network considering the temporal consistency and shares features at the testing stage. Su *et al.* [37] proposed a DBN model to recover the central frame from neighboring frames. To model spatio-temporal characteristics and restore sharp images, Zhang *et al.* [4] introduced 3D convolutions and adversarial learning.

2.3 Video Generation

Generating videos from texts, images or videos poses challenges to existing generative models [38–40]. For motion prediction, recent methods focus on training transform networks to compress the current information and generate a sequence of future frames [41–44]. Using a GAN, Mathieu *et al.* [41] predicted future frames based on adversarial loss and image gradient difference loss. Villegas *et al.* [42] built a model based on an Encoder-Decoder CNN and a Conv-LSTM to capture the spatial-temporal dynamics. Their model effectively handles complex variations in pixel space. Zhao *et al.* [44] proposed a two-stage framework to generate frames and then refine by temporal signals.

The closest work to ours for producing a video sequence from a blurry image is the pioneering work in [1]. It first estimates the middle frame of the temporal sequence and then sequentially reconstructs pairs of frames, one forward and one backward in time, in each step. Following this work, Pan et al. [45] proposed an EDI model to reconstruct a sharp video from a single blurry frame based

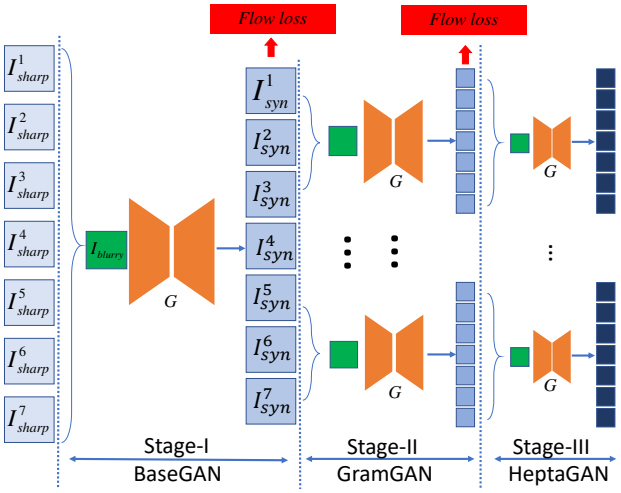

Figure 2: Cascaded structure for generator training. *Consecutive input frames are averaged and input to BaseGAN to recover sharp frames. At Stage-2, frames are averaged to be new blurry images and sent into GramGAN to recover images with more appreciable subtle movements. HeptaGAN at Stage-3 guides to recovers disparity information.*

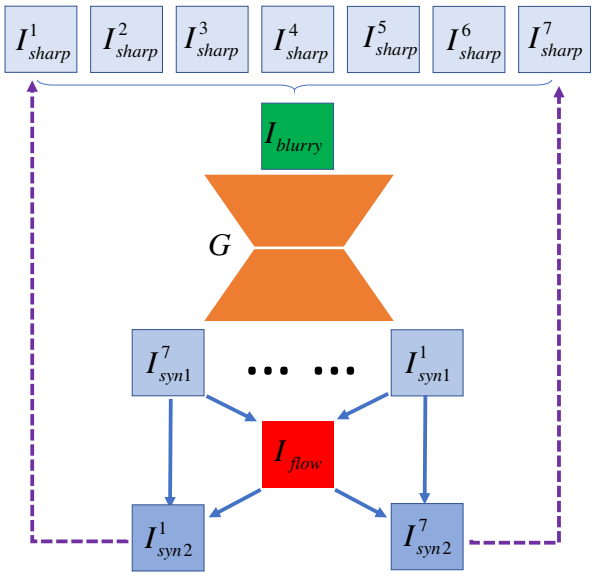

Figure 3: BaseGAN architecture with optical flow. *Seven continuous sharp frames are averaged into a blurry image as input to a generator to recover seven sharp frames* $\{I^i_{syn1}, i = 1, ..., 7\}$. *Flow images are calculated based on the 1st and 7th synthesized images. The flow is applied to warp the synthesized image* I^7_{syn1} *and results in* I^1_{syn2}. *Likewise,* I^1_{syn1} *is warped with the flow to produce* I^7_{syn2}. *These two images,* I^1_{syn2} *and* I^7_{syn2}, *are constrained to be close to their sharp counterparts* I^1_{sharp} *and* I^7_{sharp}, *to make sure the recovered motion across frames is identical to that before recovery.*

on event camera, while Purohit et al. [46] try to learn a motion encoder for blurred images based on a pre-trained convolutional recurrent video autoencoder network. Our approach differs mainly in that we employ a single model, trained in three stages, compared to using different models to reconstruct a multi-frame sequence.

3 APPROACH

To approach the task of extracting multiple frames from a motion-blurred image, we propose to train a generator G in a cascaded structure with three stages (Fig. 2): (1) In the first stage, a BaseGAN module with a flow loss function generates a sharp and realistic video without ambiguity. Seven continuous frames are averaged to simulate a motion-blurred image which is input to the BaseGAN, and the output is seven sharp frames. (2) Subtle movements are addressed by a GramGAN module in the second stage, which takes the output of the first stage as input and outputs seven sharp frames. (3) The third stage employs HeptaGAN training to recover the information of the discrete predicted frames regarding the continuous exposure process. G is an encoder-decoder model with 30 convolutional layers and seven consecutive frames output.The generator structure remains unchanged and weights are shared in the three stages. During inference, G predicts seven output frames from a motion-blurred image with a single forward pass.

3.1 Ambiguity Resolving with Flow: BaseGAN

The BaseGAN module in the first stage is a generative adversarial network. The generator produces seven frames recovering as much information as possible and the discriminator aims to discriminate the predicted frames against real frames to ensure the predicted frames are realistic. We adopt the illumination as input since it is the most salient channel. The pixel-wise MSE loss is widely used for generating deblurred images, which may have high PSNR values but are unsatisfying due to over-smoothed textures. Thus the content loss G for the central frame in this paper includes both MSE and perceptual loss [47] as

$$\mathcal{L}^{central}_{content} = \left\| I_{sharp} - G(I_{blurry}) \right\| + \left\| \Phi(I_{sharp}) - \Phi(G(I_{blurry})) \right\|, \quad (1)$$

where $G(I_{blurry})$ is the deblurred image, and I_{sharp} corresponds to the sharp frame. Φ denotes the features obtained from the last convolution layer of VGG19 [48], which is employed to measure the perceptual loss.

The procedure of recovering the other six neighboring frames is unstable if employing the same content loss defined above, because different orders among frames produce the same motion-blurred image. Thus, the content loss for these frames can be represented as [1]:

$$\mathcal{L}^{pair}_{content} = \sum_{i=1}^{3} \left([I^i_{sharp}, I^{8-i}_{sharp}]_+ - [G(I^i_{blurry}), G(I^{8-i}_{blurry})]_+ \right) + \left([I^i_{sharp}, I^{8-i}_{sharp}]_- - [G(I^i_{blurry}), G(I^{8-i}_{blurry})]_- \right), \quad (2)$$

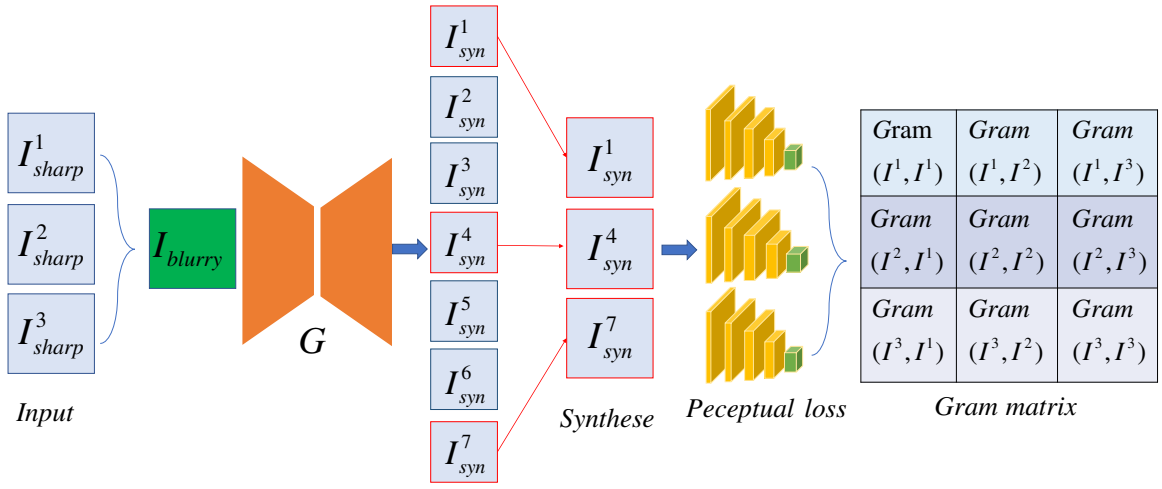

Figure 4: Gram matrix components for three sequential frames. *Blocks on the diagonal are the individual frames themselves, while the off-diagonal blocks are correlations between the sequential frames.*

where $[x,y]_+ = |sum(x,y)|^2$ and $[x,y]_- = |sub(x,y)|^2$ denote the summation and subtraction operation on corresponding positions of two input images, respectively.

To generate realistic sharp frames, an adversarial loss function is introduced with the goal to fool the discriminator D:

$$\mathcal{L}_{adv} = \log\left(1 - D(G(I_{blurry}))\right) \tag{3}$$

where $D(G(I_{blurry}))$ classifies a recovered frame to determine whether or not the reconstructed frame is a real image.

Since the reconstruction is invariant to the temporal order of frames, we introduce a loss function based on optical flow, shown in Fig. 3. Seven sharp frames are averaged to create a blurry image, which is input into a generator to produce seven synthesized sharp frames. The first and seventh synthesized frames, I^1_{syn1} and I^7_{syn1}, are then fed into a PWC-Net [49] which computes pair-wise optical flow. This is applied to the first and seventh synthesized frames to obtain new seventh and first frames, respectively. The loss function is calculated based on the input (*sharp*) and output (*syn*2) frames as

$$\begin{aligned}\mathcal{L}_{flow} = &||I^1_{sharp} - W(I^7_{syn1}, I^{7\to1}_{flow})||_2^2 \\ &+ ||I^7_{sharp} - W(I^1_{syn1}, I^{1\to7}_{flow})||_2^2,\end{aligned} \tag{4}$$

where I_{sharp} are real sharp frames, $I^{i\to j}_{flow}$ is the optical flow image from the ith to the jth frame. $W(I^7_{syn1}, I^{7\to1}_{flow})$ means that we generate the new first frame using the seventh synthesized frame and flow images via spatial transformer networks (STN) [50]. By constraining the generation process with the flow loss, the unique order among sequential frames is maintained in training and thus recovered in inference. During training in the first stage, the loss functions are combined as

$$\mathcal{L} = \mathcal{L}^{central}_{content} + \mathcal{L}^{pair}_{content} + \alpha\mathcal{L}_{adversarial} + \beta\mathcal{L}_{flow}. \tag{5}$$

In order to balance the content, adversarial and flow losses, we use hyper-parameters α and β to yield the final loss $\mathcal{L}$.

3.2 Learning Subtle Movements: GramGAN

The first stage guides the generator to produce sharp realistic frames, while the content loss is weak in learning motion across frames. The loss is very small in the case of subtle movements, resulting in small pixel variations across neighboring frames. This makes it difficult to learn the motion dynamics in training, reconstructing nearly identical sequence frames. Thus in the second stage, we focus on learning subtle movements, to improve the robustness of the model in the extreme case.

To this end, we introduce the Gram matrix at this stage to process high-level semantic features and incorporate temporal information. Note that the Gram matrix has been employed in recent work to represent motion for dynamic texture synthesis and generation of time-lapse videos [43, 51]. However, to the best of our knowledge, this is the first time to introduce it to the task of reconstruction from blurry images. Further, in contrast to prior work, which applies the Gram matrix on the features of a GAN discriminator, our model uses the Gram matrix in the generator.

The second training stage, GramGAN, is illustrated in Fig. 4. Three of the seven output frames of the first stage are averaged to create a blurry image, and a Gram matrix is computed by combining the feature maps of three sequential frames (I, I', I''). The feature map of a synthesized frame is a 3-dimensional tensor, whose axes are width, height, and channel, respectively. Firstly, we concatenate feature maps along an additional axis to produce a 4-d tensor, whose first axis corresponds to the three sequential frames. Then we reshape the 4-d tensor into a 2-d one, **F**, whose first axis is combined from the first two axes and second axis is from the last two axes. Finally, the product of the new tensor **F** and its transpose describes motion by spatio-temporal statistics. Thus the Gram matrix entry for three frames can be be formulated as a perceptual term as

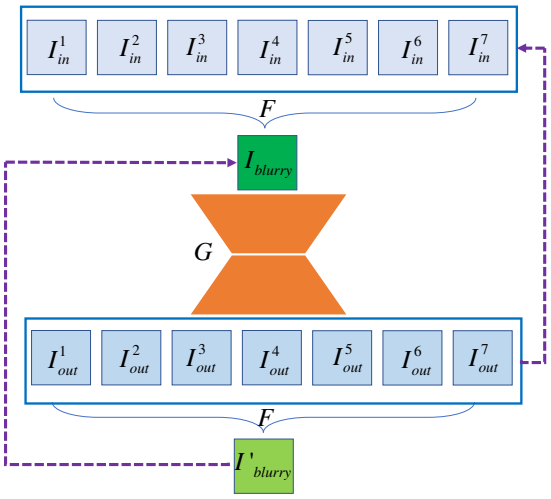

Figure 5: HeptaGAN schematic. *Given seven continuous frames, our system simultaneously creates a corresponding blurry image based on function F and learns a video recovery function G. G outputs seven frames, serving as input to F to produce a new blurry image. The learning constrain can be written as:* $G(F(\{I_{in}\})) = \{I_{out}\} \approx \{I_{in}\}$ *and* $F(G(I_{blurry})) = I'_{blurry} \approx I_{blurry}$.

$$Gram(I, I', I'') = \frac{1}{M}\mathbf{F}^{\mathrm{T}}\mathbf{F}, \tag{6}$$

where $M = CHW$ denotes the product of channel, height and width of feature maps.

Given seven output frames by the first stage, there are nine combinations of three frames with equal distance along the time axis (*i.e.*, $I_1I_2I_3$, $I_2I_3I_4$, ... , $I_1I_4I_7$). The additional loss function with regard to the Gram matrix is

$$\mathcal{L}_G = \sum_{i=1}^{9} \left\| Gram_i(G(I_{blurry})) - Gram_i(\{I_{sharp}\}) \right\|, \tag{7}$$

where I_{blurry} is the blurry image produced by averaging the three frames $\{I_{sharp}\}$ taken from the seven frames output by the BaseGAN. As Fig. 4 shows, I^1_{syn}, I^4_{syn}, and I^7_{syn} from the generated seven frames $G(I_{blurry})$ are taken as input to calculate the Gram matrix. We constrain these three frames by referring to the corresponding ground truth $\{I_{sharp}\}$. $Gram_i(\cdot)$ corresponds to the i-th way of taking three images.

Note that in Fig. 2 we take three frames of the seven frames from the first stage, rather than all of them to simulate a blurry image. The motivation is that, by doing so, we can interpolate the motion across the three frames into the fine-grained motion dynamics across the output seven frames. That also explains why we use the first synthesized frame (I^1_{syn}), the midterm frame (I^4_{syn}), and the last frame (I^7_{syn}), as the start, intermediate and end state of the motion dynamics, and use the corresponding input three frames to constrain them.

There are several advantages of GramGAN training in this stage. First, we can learn the motion dynamics more efficiently with the Gram matrix as motion representation, avoiding generation of multiple identical frames. Second, with less input and more output images, the model is able to unravel fixed time period into more discrete time steps with fine-grained motion. We verify this in Sec. 4.3. Thirdly, "continuous" and "discrete" correspond to $\left(\frac{1}{T}\int_{t=0}^{T} I_{S(t)}\mathrm{d}t\right)$ and $\left(\frac{1}{M}\sum_{t=0}^{M-1} I_{S[t]}\right)$, respectively. The real motion-blurred images are generated based on the former, while our goal is to recover M sharp images I_S like the latter. Therefore, we use HeptaGAN to calculate the differences between two blurry images, and two groups of sharp images to push our model keep original information.

3.3 Disparity Recovery: HeptaGAN

Output frames from the trained first and second stages are already realistic and exhibit more appreciable subtle motions across neighboring frames. The exposure process in the real world producing the motion-blurred image is *continuous*. However, our task of recovering multiple frames from a single motion-blurred image is a reverse process and it is actually *discrete*. To address this, we propose a HeptaGAN stage, using a blur function F and a blur-removal function G to encourage the preservation of original information contained in the single motion-blurred image.

In particular, given frames generated from the GramGAN stage as input, the idea is to produce a blurry image and recover the sharp frames by using the produced blurry image, and the recovered sharp frames are averaged to produce a blurry image again, forming a bi-cycle process. We train the model in an unsupervised manner in this stage. As illustrated in Fig. 5, we build two function approximators F and G. F produces blurry images from consecutive sharp frames and G recovers the video sequence from the synthesized motion-blurred image. Because of the assumption that motion-blurred images can be produced by averaging multiple frames, the model F is the average function and we only train G. Given seven sharp frames $\{I_{in}\}$, the motion-blurred images can be produced as $I_{blurry} = F(\{I_{in}\})$. We expect that G can generate continuous sharp frames $\{I_{out}\} = G(I_{blurry})$, whose corresponding averaging motion-blurred frame $I'_{blurry} = F(\{I_{out}\})$ is the same as I_{blurry}. This imposes the bi-cycle consistency. Note that, different from the traditional CycleGAN [39] which simultaneously trains G and F on paired or unpaired images, the input to our HeptaGAN are seven consecutive frames. We only train G, but with two cycle-like losses which are discussed as follows and shown in Fig. 5.

The L1 loss is used to constrain this learning process as

$$\begin{aligned}\mathcal{L}_C = &\frac{1}{N}\sum^{N} \|F(\{I_{in}\}) - F(G(F(\{I_{in}\})))\|_1 \\ &+ \frac{1}{7N}\sum^{K}\sum_{j=1}^{7}\left\|I^j_{in} - (G(F(I^j_{in})))\right\|_1,\end{aligned} \tag{8}$$

where N is the number of seven-frame groups.

Figure 6: Qualitative comparison. *Two input images and zoomed in regions are shown in the first row. The 2nd/6th rows show results of the method by Jin et al.[1]. The 3rd/7th to 5th/9th rows show the performance of our model trained after one (**B**), two (**BG**) and three stages (**BGH**). Note the movement across frames (referring to the fixed-position red boxes) is recovered successfully by the model **BG**, and the sharper detail inside the yellow boxes recovered by the fully trained **BGH** model.*

4 EXPERIMENTS

We test our approach on the widely used public GOPRO dataset [24], which is first introduced along with evaluation metrics. Then implementation details are given and ablation study is conducted, and a comparison with the state of the art is reported. Finally, we test the generalization of our method to blur caused by bokeh.

4.1 Dataset & Metrics

In our experiments we use the GOPRO_Large_all frames of the GOPRO dataset, including 22 training and 11 test videos, respectively. We average consecutive frames to produce blurry images. To compute the fidelity of the extracted frames, we use the PSNR as a metric. Additionally we check how accurately the motion across frames is preserved by computing the end-point error (EPE) of flow across the generated frames with respect to the flow from the ground truth frames.

Table 1: *Performance comparison with [24], [1], [45] and [46] on the* GOPRO_Large_all *dataset and ablation study of model G after different stages of training.*

Method	PSNR	SSIM	EPE
Nah et al.	28.98	0.911	-
Jin et al.	26.98	0.881	17.93
Pan et al.	28.49	0.920	-
Purohit et al.	30.58	0.941	-
B	28.14	0.905	13.62
BG	29.65	0.921	11.25
BGH	**30.64**	**0.942**	**10.03**

4.2 Implementation Details

During training, model weights are initialized from a normal distribution with zero mean and a standard deviation of 0.01. We update all weights with a mini-batch of size 4 in each iteration. To augment the dataset, 128×128 patches are cropped at random locations and horizontally mirrored at random. The model is trained with an annealing learning rate scheme, starting with 10^{-4} and decreasing to 10^{-5} after convergence. The hyper-parameters α and β in Eq. (5) are empirically set as 0.0005 and 0.001.

The training procedure is as follows. The generator G is trained using BaseGAN at first, and then we incrementally train G with GramGAN and HeptaGAN to fine-tune the model. The generator G in the BaseGAN, GramGAN and HeptaGAN shares weights, thus a video recovery model G is obtained which is robust to resolve ambiguity (BaseGAN), preserve subtle movements (GramGAN) and recover disparity information (HeptaGAN). The ablation study compares the model performance after different stages of training. During inference, given a motion-blurred image, we generate seven frames in one forward pass of G.

4.3 Ablation Study

In this section, we conduct experiments to investigate the effect of the different training stages. We show both qualitative results and quantitative results in the form of PSNR and EPE values. We compare the following models:

- **B** is the network trained as BaseGAN. The input to this model is a motion-blurred image, which is created from seven real consecutive frames.
- (2) **BG** is the generator trained with BaseGAN (Stage-1) and GramGAN (Stage-2) stages. The input to the GramGAN is the output of the BaseGAN.
- (3) **BGH** is the model trained after all three stages, adding the HeptaGAN third stage.

Table 1 shows the PSNR and EPE values. Performance increases after each training stage, with the fully trained model, **BGH**, achieving the best performance.

Fig. 6 shows qualitative results of the different models. Compared to model **B**, the results of **BG** shows more evident subtle movements across neighboring frames, suggesting the effectiveness of learning motion dynamics using GramGAN. The **BGH** model recovers more details and creates sharper images due to the disparity recovery.

Figure 7: Example of interpolation of subtle motions. *42 frames (from left to right, top to bottom) are extracted by the proposed method based on the input image shown in Fig. 1. Please check the movement of the rear light comparing the first frame with last one. Note that there are no 42 original frames as the input blurry frame is produced by averaging only 7 frames. By iteratively applying the model we are therefore able to create slow-motion videos from blurry images.*

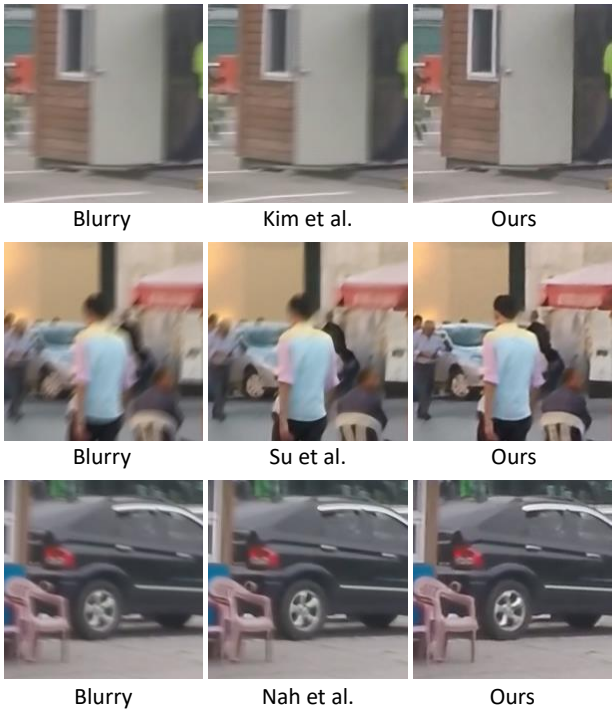

Figure 8: Comparison with deblurring methods. *Methods provided by Kim* et al. *[27], Su* et al. *[37] and Nah.* et al. *[24] are specialized for recovering a sharp frame from a blurry image.*

Please check the area marked with the yellow bounding boxes. The contrast of digit "3" by **BGH** is higher than **BG**. The ear of the man is also recovered with more details by **BGH**.

Fine-grained Motion Interpolation. We are able to recover more than seven frames by iteratively applying model G to output frames. Seven output frames form six groups (I_1I_2, I_2I_3, ... , I_6I_7), and each can be averaged to produce another blurry image, which can be fed in our generator to again produce seven frames. By doing so, we recover $6 \times 7 = 42$ frames with extremely subtle motions from one blurry image, as shown in Fig. 7. We can even recover arbitrarily many frames by repeating this procedure. This demonstrates our model can be employed to disassemble a single motion-blurred image into multiple frames with interpolated fine-grained motion dynamics across frames.

4.4 Comparison with Existing Methods

We compare our method with different methods, including [1], [45], [46], [24], [27] and [37]. [1], [45] and [46] are the state-of-the-art methods for extracting image sequences from a motion-blurred image. [24], [27] and [37] are popular image deblurring methods. Table 1 shows quantitative results. Our method achieves higher PSNR values than [1], [45] and [46]. The smaller EPE value suggests that our method is better able to learn subtle motion across frames. We suspect the improvement is attributed to our specific handling of the challenges faced by extracting video from a single motion-blurred image. Figs. 1 and 6 show qualitative comparisons, highlighting the improved ability of our method to recover subtle motion and image details.

We also compare our method with image deblurring methods [24]. Since deblurring methods typical output only a single image, we select the central frame of our reconstruction for comparison. As shown in Table 1, our method outperforms the one in [24]. This may be explained by the fact that we use consecutive sharp frames to produce motion-blurred images during training, while [24] only trains with one sharp image per motion-blurred image. Qualitative results comparing with [24], [27] and [37] are shown in Fig. 8. The proposed method produces sharper and more realistic frames.

4.5 Generalization to Other Types of Blur

Our model is trained on the GOPRO dataset, within which the blur artifacts are mainly caused by camera shake. In this section we apply our method to images containing a different type of blur. The KITTI dataset [52] includes images captured by a camera mounted on a moving vehicle, thus the dominant blur is caused by bokeh rather than camera shake. We test our model on this dataset and show example results in Fig. 9. The results demonstrate that the proposed method is able to recover sharper frames with evident subtle motion across neighboring frames and rich details for various kinds of blur artifacts.

Figure 9: Results on the KITTI dataset. *The first column shows details of the two blurry input images in the top row, and the following seven columns show images generated by the proposed model. The subtle motion outlined by boxes with different colors shows that the model generalizes well to blur caused by bokeh.*

5 CONCLUSION

This paper presents a detail-aware network, which is a cascaded generator to extract an image sequence from a blurry image. To handle the problems of ambiguity, subtle motion, and loss of details, we train a model using a BaseGAN constrained with optical flow, a GramGAN, using a Gram matrix as motion representation, and a HeptaGAN with a bi-cyclic constraint. Experimental results demonstrate that our generator not only produces compelling results but also outperforms state-of-the-art methods.

ACKNOWLEDGMENT

This work is funded in part by the ARC Centre of Excellence for Robotics Vision (CE140100016), ARC-Discovery (DP 190102261) and ARC-LIEF (190100080) grants. The authors gratefully acknowledge NVIDIA for GPU gift. This work is also supported by Tencent Rhino Bird Elite Graduate Program.

REFERENCES

[1] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *CVPR*, 2018.
[2] Yuval Bahat, Netalee Efrat, and Michal Irani. Non-uniform blind deblurring by reblurring. In *CVPR*, 2017.
[3] Haichao Zhang and David Wipf. Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In *NIPS*, 2013.
[4] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *TIP*, 2018.
[5] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014.
[6] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015.
[7] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016.
[8] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020.
[9] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *ECCV*, 2010.
[10] Stefan Harmeling, Hirsch Michael, and Bernhard Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *NIPS*, 2010.
[11] Taeg Sang Cho, Sylvain Paris, Berthold KP Horn, and William T Freeman. Blur kernel estimation using the radon transform. In *CVPR*, 2011.
[12] Michael Hirsch, Christian J Schuler, Stefan Harmeling, and Bernhard Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, 2011.
[13] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *IJCV*, 2012.
[14] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *WACV*, 2020.
[15] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *CVPR*, 2020.
[16] Kaihao Zhang, Wenhan Luo, Lin Ma, and Hongdong Li. Cousin network guided sketch recognition via latent attribute warehouse. In *AAAI*, 2019.
[17] Kaihao Zhang, Wenhan Luo, Lin Ma, Wei Liu, and Hongdong Li. Learning joint gait representation via quintuplet loss minimization. In *CVPR*, 2019.
[18] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. In *BMVC*, 2019.
[19] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *ECCV*, 2016.
[20] Jiangxin Dong, Jinshan Pan, Zhixun Su, and Ming-Hsuan Yang. Blind image deblurring with outlier handling. In *ICCV*, 2017.
[21] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014.
[22] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015.
[23] Christian Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *TPAMI*, 2016.
[24] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.
[25] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *CVPR*, 2018.
[26] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Exploiting semantics for face image deblurring. *IJCV*, 2020.
[27] Tae Hyun Kim and Kyoung Mu Lee. Generalized video deblurring for dynamic scenes. In *CVPR*, 2015.
[28] Haichao Zhang and Jianchao Yang. Intra-frame deblurring by leveraging inter-frame camera motion. In *CVPR*, 2015.
[29] Anita Sellent, Carsten Rother, and Stefan Roth. Stereo video deblurring. In *ECCV*, 2016.
[30] Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *ICCV*, 2017.
[31] Haesol Park and Kyoung Mu Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *CVPR*, 2017.
[32] Hee Seok Lee and Kuoung Mu Lee. Dense 3d reconstruction from severely blurred images using a single moving camera. In *CVPR*, 2013.
[33] Hee Seok Lee, Junghyun Kwon, and Kyoung Mu Lee. Simultaneous localization, mapping and deblurring. In *ICCV*, 2011.
[34] Hailin Jin, Paolo Favaro, and Roberto Cipolla. Visual tracking in the presence of motion blur. In *CVPR*, 2005.
[35] Patrick Wieschollek, Michael Hirsch, Bernhard Schölkopf, and Hendrik PA Lensch. Learning blind motion deblurring. In *ICCV*, 2017.
[36] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Schölkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *ICCV*, 2017.
[37] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017.
[38] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
[39] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.
[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NIPS*, 2018.
[41] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.
[42] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.
[43] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*, 2018.
[44] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, 2018.
[45] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *CVPR*, 2019.
[46] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *CVPR*, 2019.
[47] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
[49] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.
[50] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015.
[51] Matthew Tesfaldet, Marcus A Brubaker, and Konstantinos G Derpanis. Two-stream convolutional networks for dynamic texture synthesis. In *CVPR*, 2018.
[52] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013.