IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Human and Scene Motion Deblurring using Pseudo-blur Synthesizer

JONATHAN SAMUEL LUMENTUT[1], (Student Member, IEEE), IN KYU PARK.[1], (Senior Member, IEEE)

[1]Department of Information and Communication Engineering, Inha University, Incheon 22212, Korea (e-mail: jlumentut@gmail.com, pik@inha.ac.kr)

Corresponding author: In Kyu Park (e-mail: pik@inha.ac.kr)

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFCIT1901-06. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University)). This work was supported by Inha University Research Grant.

ABSTRACT Present-day deep learning-based motion deblurring methods utilize the pair of synthetic blur and sharp data to regress any particular framework. This task is designed for directly translating a blurry image input into its restored version as output. The aforementioned approach relies heavily on the quality of the synthetic blurry data, which are only available before the training stage. Handling this issue by providing a large amount of data is expensive for common usage. We answer this challenge by providing an on-the-fly blurry data augmenter that can be run during training and test stages. To fully utilize it, we incorporate an unorthodox scheme of deblurring framework that employs the sequence of blur-deblur-reblur-deblur steps. The reblur step is assisted by a reblurring module (synthesizer) that provides the reblurred version (pseudo-blur) of its sharp or deblurred counterpart. The proposed module is also equipped with hand-crafted prior extracted using the state-of-the-art human body statistical model. This prior is employed to map human and non-human regions during adversarial learning to fully perceive the characteristics of human-articulated and scene motion blurs. By engaging this approach, our deblurring module becomes adaptive and achieves superior outcomes compared to recent state-of-the-art deblurring algorithms.

INDEX TERMS Motion deblur, pseudo-blur, augmentation, synthesize, generative adversarial network, human motion, deep neural network

I. INTRODUCTION

The idea of recovering blurry images into their sharp version has been presented since a decade ago and remains an active research area in computer vision. The spread of pixels usually causes a blurry image owing to the motion effect during capture time. This motion is modeled by a specific point spread function (PSF) and can be represented as a blur kernel. Early motion-blurred image is modeled by the blur kernel that is directly convolved to a sharp image with few additive noises. The task of restoring PSF-based degradation is known as motion deblurring. Based on the previous assumption, traditional methods solve motion deblurring by deconvolving back the blurry image with a predicted blurry kernel. This idea was implemented in famous state-of-the-art works [1]–[7] where various regularization priors are also advocated to help the deblurring procedures. With the rise of deep learning, many kernel-free deblurring works are introduced. The work of Nah *et al.* [8] and Kupyn *et al.* [9], [10] utilize generative-adversarial-network (GAN) [11] to solve this issue. The recent deep-learning based deblurrings are improved by adopting feature-level modification [12]–[14] and region-based prior utilization [14]–[16].

Following the vast growth of image restoration works, we observe that the current highlight involves an unorthodox approach for improving performance. Early work by Chen *et al.* [17] solves deblurring by reblurring the deblur output. The reblur output is being supervised with the blurred input during training. To simplify, Chen *et al.* [17] apply the sequence of *blur-deblur-reblur* ($B \rightarrow D \rightarrow R$) in the training scheme with $R \approx B$, while maintaining the ($B \rightarrow D$) in the test scheme. Recently, Zhang *et al.* [18] provide a unique approach by supplying a reblur network at the top of the deblurring module for unpaired strategy. Their approach [18] employs noise-based re-blurred version of any sharp image $R_{\mathcal{N}}$ and utilize them in the training process ($R_{\mathcal{N}} \rightarrow D$). Take note that $R_{\mathcal{N}}$ will be inconsistent as noise is generated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

randomly. Thus, they [18] also still perform ($B \rightarrow D$) sequence in the test scheme without utilizing $R_{\mathcal{N}}$. These approaches indicate that re-corrupting initial information is useful for augmenting the data.

From this motivation, we introduce the order of *blur-deblur-reblur-deblur* ($B \rightarrow D \rightarrow R \rightarrow D$) as a modish approach to solve human and scene motion deblurring. Unlike the previous unorthodox methods [17], [18], our sequence is performed in **training** and **test** stages. This approach allows our deblurring method to learn the augmented blurry data R that are **different** from the blurry input B (with $R \not\approx B$) in **both** stages. To obtain a consistent R, we provide a reblurring module (pseudo-blur synthesizer) that only receives a single RGB image, which is trained with the localized regions of the human body and scene. This strategy is applied as human-articulated and scene motion display different blur characteristics. Our motivation is that, up to recent time, only a few pioneer works [14], [16] that particularly handle the deblurring on human body case. The idea of human deblurring is presented by Shen *et al.* [14] that utilizes separated foreground and background maps to distinguish human and non-human regions. However, this approach is un-precise as they produce pre-generated rectangular maps to cover the human body. Moreover, in their case, the blur is unlikely to represent the motion blur caused by human articulated body joint movement. This approach is tackled by Lumentut *et al.* [16] by producing a localized map that covers both *human body* and *its nearby regions* that are affected by the *body-joint articulation motion*. In this work, the idea of the localized prior map is utilized in the *reblurring* procedure as part of the novel sequence. To achieve it, we propose an adversarial-based framework that learns both scene and human motion blur characteristics for supplying the reblurring module.

Once the reblurring module is settled, R is treated as augmentation data. However, the ultimate goal of adopting the proposed sequence is to perform self-adaptation in the test stage. This is important as many restoration models are trained with a limited dataset but required to solve various issues. To achieve it, the proposed sequence is plugged in a model-agnostic meta-learning algorithm [19], that shows significant performance improvement in previous non-deblurring studies [20]–[22]. By implementing this strategy, we show that our approach is superior in deblurring the real-world scenario, where no related training data is available. This benefit is obtained due to the presence of our pseudo-blur synthesizer that supports the proposed sequence. To summarize, we describe our contributions as 3 manifolds:

- We present a unique sequence of deblurring in training and testing procedures, which allows a self-adapting capability that yields superior results compared to recent state-of-the-art works.
- We provide, to our best knowledge, a novel way to synthesize a blurry image from only a single RGB image input, achieved by employing localized human and non-human regions of an image.
- We show that the hand-crafted human-prior in the reblurring module is learnable via adversarial strategy, subsequently improving the deblurring performance.

II. RELATED WORKS

a: Motion deblurring

Early deblurring algorithms utilize the classical way of restoration by firstly estimating the blur kernel. The estimated kernel is used to deconvolve the blurry input to obtain sharp input. Various regularization priors are utilized for improving this approach [1]–[7], [23]. These works further target multi-view imaging, as shown in these studies [24]–[26]. A recent trend on deep learning moves toward GAN-based [11] architecture for its capability to directly translate an image to a certain domain [27]. This approach is then followed by these notable deblurring works [8]–[10]. Recent works in deblurring include hand-crafted priors to prioritize certain regions for learning. These works are shown by Shen *et al.* [15] and Ren *et al.* [28] for face deblurring as well as Shen *et al.* [14] and Lumentut *et al.* [16] for human deblurring. As explained in the previous section, the recent works of non-classic restoration (non $B \rightarrow D$ in training sequence) [17], [18] capture our attention. These approaches show that the addition of a reblurring step improves the deblurring performance. Both closely related works of Shen *et al.* [14] and Lumentut *et al.* [16] inspire our works to solve human deblurring. As described in the previous section, our method takes advantage of the prior map in generating realistic augmented blurry data via reblurring module. We utilize this module to help the deblurring module in achieving its self-adaptive capability.

b: Synthetic blur generation

The early work of generating a blurry image is initiated by [29]. Their work produces camera motion with a robot system that moves the camera in an accurate position. This approach is bulky and hard to be applied in daily use. A more complex blurry dataset is introduced by [30] in which sets of motion blurs that are recorded using the inertial sensor of a consumer cellphone are collected. These motion blurs are convolved directly to the sharp image to produce a synthetic blurry image. Recent approaches show that the averaging multiple-frames is faithful enough to generate realistic scene blur as expressed by [8]. Another non-typical approach is introduced by Brooks and Barron [31] wherein a blurry image is produced from the two successive sharp frames. Their idea is based on frame interpolation work as it produced several intermediate frames within the two inputs to generate a smooth blurry result. The recent work of [14] provides blurry human images as their dataset to solve a particular issue, notably human deblurring. However, their blur result is affected by non-human articulated motion. Unlike [14], we consider both constraints on generating the synthetic blur achieved by pursuing the local human body region. The closest work to ours is by Zhang *et al.* [32] that provides a reblurring network to blur the sharp image during training ($R \rightarrow D$). This method, however, relies fully on

the additional noise at the input. On the contrary, our blur synthesizer network only requires a single red-green-blue (RGB) input, which guarantees a consistent reblur output.

c: Meta-learning application

The meta-learning approach paves a unique way for recent restoration works [20]–[22]. Its objective is to provide an updated version of a network that is adaptive during test time. In general, meta-learning is categorized into three groups. The first group belongs to the metric-based method [33], [34]. This approach has the objective of seeking metric space that provides efficient learning under a few samples. The second group belongs to the memory network-based approach [35]–[37] where its objective is to train a network that learns across various tasks to be robust to the unseen task. The last group belongs to the optimization-based approach, wherein gradient-based learning is employed. The main idea is to find an initial transferable point that helps the network adapt within a few gradient updates [19], [38], [39]. The recent model-agnostic meta-learning [19] method that utilizes gradient descent learning shows a significant impact on super-resolution studies [20], [22]. Just recently, a test-time adaptive version of the motion deblurring method is proposed by Chi *et al.* [40]. This work is closely related to ours, however they only utilize ($B \rightarrow D \rightarrow R$) sequence similar to [17] which enforces R to be equal to B. As opposed to this setup, we employ our reblurrer to synthesize R differently to B as its augmented version. In our experiments, we show that our proposed strategy succeeds in enhancing the deblurring network via meta-learning.

III. METHOD

We introduce a set of procedures for training (Figure 2) and testing (Figure 5) stages that fully utilize the proposed sequence. Initially, the deblurring network is trained until it converges (θ_T). The deblurring network is then frozen for training the reblurring network (Ω_T). These outcomes are utilized in our meta-transfer-learning procedure to obtain a ready-to-be-adapted deblurring weight (θ_M). Finally, the meta-testing procedure is employed to run the self-adaptation strategy (θ_k). In these subsections, we describe briefly on each particular scheme: *initial deblurring training*, *pseudo-blur synthesizer training*, *meta-transfer learning*, and *meta-testing* procedures.

A. INITIAL DEBLURRING TRAINING

The whole process is started by initially train the deblurring network. This network is trained with the involvement of GoPro [8] and HIDE [14] dataset. As displayed in Figure 1, the deblurring network receives an input of blurry RGB image B and produces deblurred output D. In this stage, training the deblurring is collaborated with the *global deblur* discriminator module. The term *global* represents the utilization of full image region. As shown in Figure 1, the deblurring module is represented by the *deblurring generator* annotation. *Global deblur discriminator* module is also utilized to influence the generator. These networks, highlighted with blue color, are trained using GoPro [8] and HIDE [14] dataset. The configuration of *deblurring generator*[1] and *global deblur discriminator* networks are provided in Table 1 and Table 2, respectively. 9 Residual Blocks [41] configuration is stacked to convey intermediate features.

TABLE 1. Detailed settings of our deblurring generator module. The spatial output size is downsampled and upsampled according to the stride number.

Layer	Detail	Output size	Stride
Input (B or R)	-	$(H \times W \times 3)$	-
Conv 7×7	*IN+ReLU*	$(H \times W \times 64)$	1
Conv 3×3	*IN+ReLU*	$(H/2 \times W/2 \times 64)$	2
Conv 3×3	*IN+ReLU*	$(H/4 \times W/4 \times 128)$	2
Res_Blocks_1-9 3×3	*IN+ReLU*	$(H/4 \times W/4 \times 256)$	1
ConvTrans 3×3	*IN+ReLU*	$(H/2 \times W/2 \times 128)$	2
ConvTrans 3×3	*IN+ReLU*	$(H \times W \times 64)$	2
Conv 7×7	*Tanh*	$(H \times W \times 3)$	1

TABLE 2. Architecture details of global deblurring discriminator in our proposed framework.

Layer	Detail	Output size	Stride
Input (S or D)	-	$(H \times W \times 3)$	-
Conv 4×4	*IN+LeakyReLU*	$(H/2 \times W/2 \times 64)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/4 \times W/4 \times 128)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/8 \times W/8 \times 256)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/16 \times W/16 \times 512)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/16 \times W/16 \times 512)$	1
Conv 4×4	*IN+Sigmoid*	$(H/16 \times W/16 \times 1)$	1

Optimization of deblurring module To optimize the deblurring network in the *initial deblurring training* stage, we utilize a simple absolute error calculation between deblurred output D and sharp ground truth image S within a mini-batch b, represented as $(L_1^{deb} = \frac{1}{b}\sum_c^b \|S_c - D_c\|)$, where c represents data on each batch. This loss is countered with a single *global deblur discriminator* as shown in Figure 1. The discriminator receives the input of both sharp S for the real case and deblurred D for the fake case following the recent least-square GAN (LSGAN) introduced in [42]. In detail, the adversarial real and fake losses of the deblurring discriminator are represented as:

$$L_{\mathbf{R}eal}^{deb} = \frac{1}{b}\sum_c^b \Pi_{\mathbf{G}lo}(S_c); L_{\mathbf{F}ake}^{deb} = \frac{1}{b}\sum_c^b \Pi_{\mathbf{G}lo}(D_c), \quad (1)$$

where $\Pi(\cdot)$ represents the discriminative function. The two functions are combined in the generator and discriminator losses of the deblurring, which are written as:

$$L_{\mathbf{D}isc}^{deb} = 0.5 \times (\|L_{\mathbf{R}eal}^{deb} - 1\|^2 + \|L_{\mathbf{F}ake}^{deb}\|^2); \quad (2)$$

$$L_{\mathbf{G}en}^{deb} = L_1^{deb} + 0.5 \times (\|L_{\mathbf{F}ake}^{deb} - 1\|^2). \quad (3)$$

Note that our discriminator requires an input image that is fully divided by 16; thus, we utilize a patch size of 128×128 in the training procedures.

[1]The reblurred image R is mentioned in Table 1 as our naive-finetuned version (Ours-**F**) utilizes the deblurring module to deblur R in the sequence of $B \rightarrow D \rightarrow R \rightarrow D$ during training stage (see the Ablation Study in the subsection IV-B)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

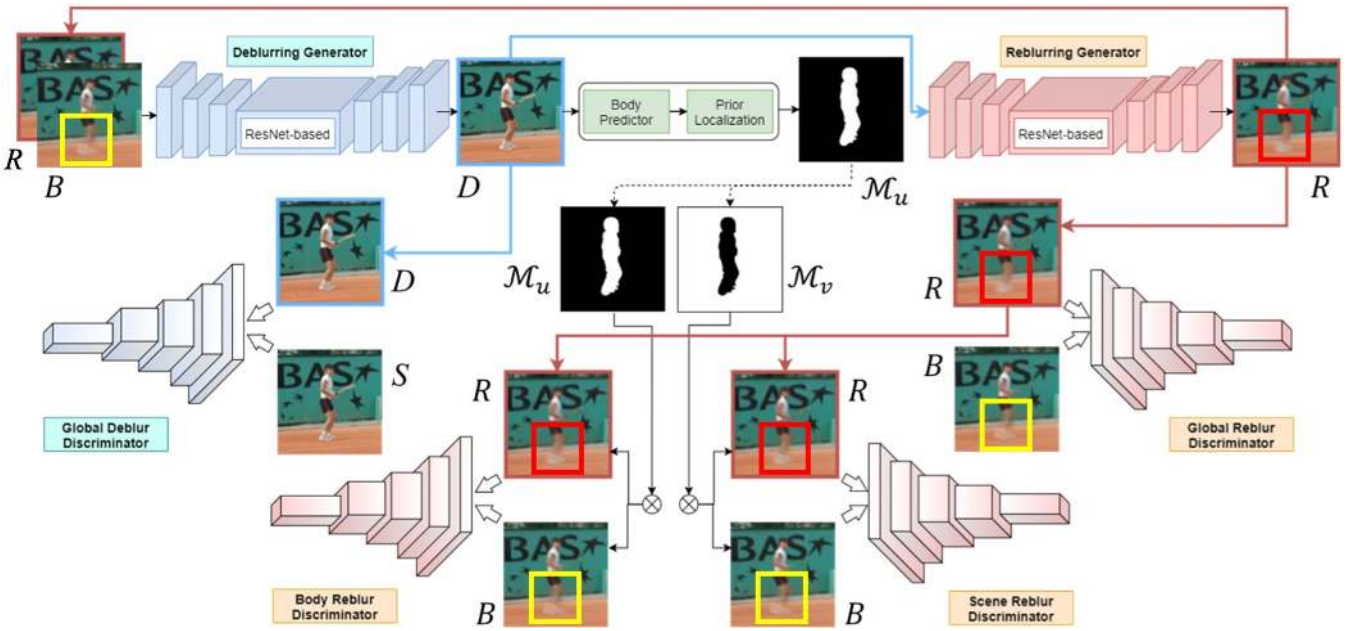

FIGURE 1. The figure above represents the framework of our deblurring-reblurring tasks. Blue modules are utilized for training the deblurring procedure, while red modules are used to train the reblurring procedure. As emphasized in the smaller rectangular boxes (red and yellow boxes), our synthesized blur result, R, is **different** with blurry input B ($R \not\approx B$). Moreover, R is deblurred again in our algorithm to induce the self-adaptive capability. Viewing it on an *electronic screen* is advised.

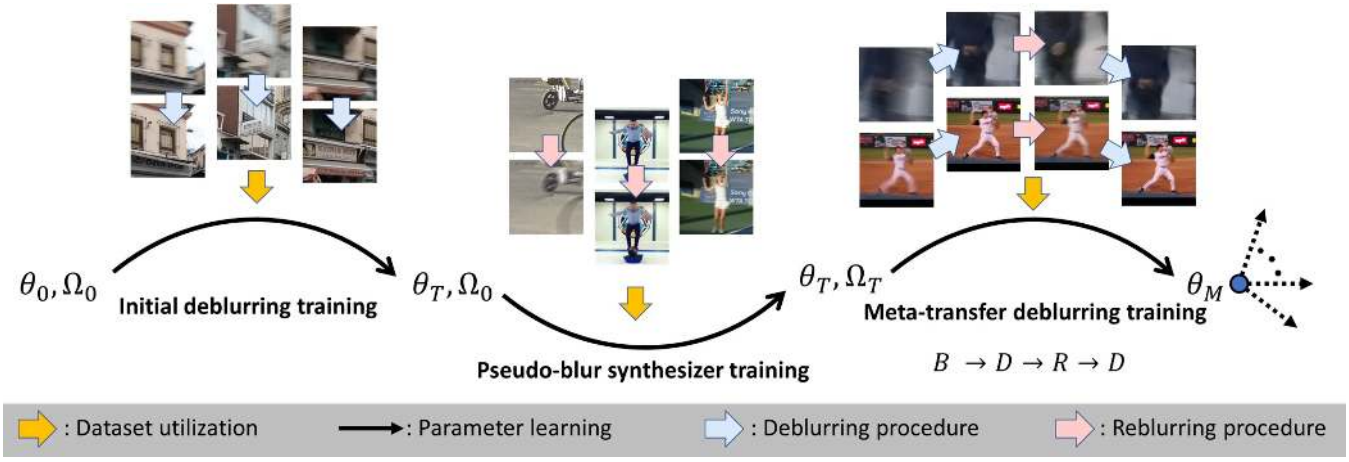

FIGURE 2. The main scheme of our **training** phase. The training procedures are run in a progressive stages, namely: *initial deblurring training*, *pseudo-blur synthesizer training*, and *meta-transfer deblurring training*. In the initial stage, only deblurring modules (blue objects) are trained. Then, in the pseudo-blur synthesizer training stage, only the reblurring modules (red objects) are trained. In the meta-transfer deblurring training stage, the deblurring modules are furtherly optimized with the utilization of frozen reblurrer module. Note that, in the final stage, no discriminators are further optimized.

B. PSEUDO-BLUR SYNTHESIZER TRAINING

The next important step is the learning process of the pseudo-blur synthesizer Ω. The related modules are reflected in Figure 1 as red-colored objects. The reblur module is fed with the deblurred image D or sharp image S to produce the reblurred version R. At the first 50 epochs, we train the reblurrer generator and *global reblur* discriminator using S only to fully learns correct features from the sharp image while treating B as ground truth. The next 100 epochs are performed with the input data of D, which is produced by θ_T using the dataset of HIDE [14]. Finally, in the last 100 epochs, we utilize an additional dataset that placed the human image in the middle region. In specific, we utilize the pairs of ground truth clean S and blurry B human images from LSP [43] dataset. For simplicity, we denote this modified dataset as LSPBlur. Its extraction procedure is discussed in the following discussion.

To produce the desired blurs in LSPBlur, we incorporate a region-separation method that splits both human (foreground) and scene (background) regions. This operation produces 2 prior binary maps, namely $\mathcal{M}_u$ and $\mathcal{M}_v$. The task of $\mathcal{M}_u$ is to cover blurry regions inside and the nearby human body. This is done as our LSPBlur is defined by human-articulated motion blur (foreground) and scene-motion blur

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

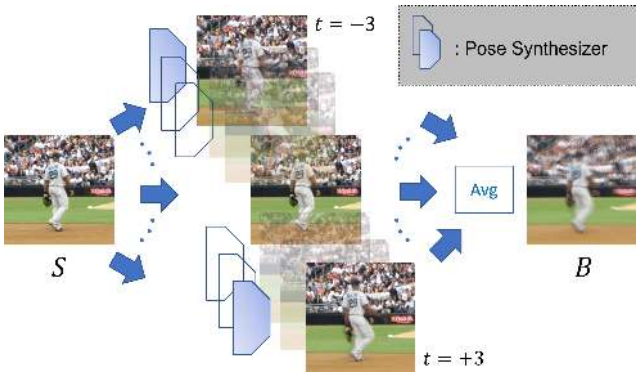

FIGURE 3. Single sharp image S is utilized to produce multiple images with changed human-pose and translated scene background. These images are averaged together to produce the blurry image B.

(background). Simply utilizing any human segmentation algorithm is ineffective as it excludes the blurry region nearby the human body. The complete procedure of this map extraction is described in the following discussion Finally, to obtain the scene-blur region map, the reversed version of $\mathcal{M}_v = 1 - \mathcal{M}_u$ is utilized.

This module is trained using LSGAN [42]; specifically optimizing the reblurring generator along with the *global*, *human*, and *scene reblur* discriminators. Both *human* and *scene reblur* discriminators utilize the prior maps $\mathcal{M}_u$ and $\mathcal{M}_v$ to explicitly penalize human and scene regions. The first 50 and 100 epochs (using S and D from HIDE) only include *global* discriminator. Full discriminators are employed in the last 100 epochs (using LSPBlur dataset). Take note that, in this stage, the optimized deblurring network is frozen so that the framework is focused on training the reblurring module. The details of the pseudo-blur synthesizer training are elaborated in the following passages.

1) Training Preliminary

Training this module is divided into 3 categories: (i) first 50 epochs using S of HIDE dataset, (ii) next 100 epochs using the deblurred result D of HIDE dataset, and (iii) last 100 epochs using the sharp image obtained from our LSPBlur dataset. The HIDE dataset is chosen as it provides the characteristic of a single sharp image as ground truth and multiple blurry images as output (one-to-many effect). Although HIDE dataset [14] is known for providing human presence, their motion blurs are affected only by the scene blur. The LSPBlur is utilized to tackle this issue.

Acquiring the LSPBlur We realize this drawback and generate a new blurry human dataset that contains human-articulated and scene motions. This dataset is collected from the Leeds Sport Dataset (LSP) [43] that contains a human in the middle region of each image. Using this image as input, we synthesize a new image with a new human pose and a newly translated scene-background. The scene translation is generated randomly while the human pose changes are obtained from various Youtube sources.

TABLE 3. Detailed settings of the pseudo-blur synthesizer generator module. The spatial output size is also downsampled and upsampled according to the stride number.

Layer	Detail	Output size	Stride
Input (S or D)	-	$(H \times W \times 3)$	-
Conv 7×7	*IN+ReLU*	$(H \times W \times 64)$	1
Conv 3×3	*IN+ReLU*	$(H/2 \times W/2 \times 64)$	2
Conv 3×3	*IN+ReLU*	$(H/4 \times W/4 \times 128)$	2
Res_Blocks_1-9 3×3	*IN+ReLU*	$(H/4 \times W/4 \times 256)$	1
ConvTrans 3×3	*IN+ReLU*	$(H/2 \times W/2 \times 128)$	2
ConvTrans 3×3	*IN+ReLU*	$(H \times W \times 64)$	2
Conv 7×7	*Tanh*	$(H \times W \times 3)$	1

TABLE 4. Architecture details of the global, scene, and body reblurring discriminators in our framework.

Layer	Detail	Output size	Stride
Input (B or R)	-	$(H \times W \times 3)$	-
Conv 4×4	*IN+LeakyReLU*	$(H/2 \times W/2 \times 64)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/4 \times W/4 \times 128)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/8 \times W/8 \times 256)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/16 \times W/16 \times 512)$	2
Conv 4×4	*IN+LeakyReLU*	$(H/16 \times W/16 \times 512)$	1
Conv 4×4	*IN+Sigmoid*	$(H/16 \times W/16 \times 1)$	1

In detail, from the example of Figure 3, the sharp S image of a man playing baseball is obtained from the LSP dataset. We then collect the poses of people playing baseball from YouTube video using AlphaPose [44], and the pose difference between each video frame is taken as the *change* parameters (Δ). Δ values are then utilized to transform the original human pose of the input image S into its new pose printed in the new image at a specific time-stamp t. This procedure is done by employing the *pose-synthesizer* method [45]. Take note that this work is done for the human part (foreground) while the scene background is translated randomly. In our experiment, we empirically produce 7 consecutive frames with slightly different body poses and translated backgrounds, as shown in Figure 3. These images are then averaged together to generate a single blurry output. This modified dataset (LSPBlur) contains 2,000 pairs of sharp and blurry images.

2) Reblurring Module Configuration

Detailed configurations of our reblurrer module are shown in Table 3 while the reblurrer discriminators are displayed in Table 4. Note that the body and scene discriminators receive the input that is masked with the map $\mathcal{M}$ that is extracted using the human prior. The prior extraction is explained in the next section. Both deblurring (Tables 1-2) and reblurring (Tables 3-4) modules are processed through the Instance Normalization (*IN*). In both generators, we utilize 9 Res_Blocks [41] layers. Each block is constructed by the pattern of Conv→*IN*→*ReLU*→Conv→*IN* added with initial input.

3) Optimization of Reblurring Module

Content loss The reblurring module is trained to translate S or D into its reblurred version R. The real blurry image B from the dataset is treated as the label. In our experiment, the one-to-many effect of HIDE dataset may produce a slight color change between B and R. Therefore, we utilize the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

Y channel only in the reblurring loss (L_1^{reb}) after these parameters are converted from RGB to YUV spaces. The full representation of L_1^{reb} is written as follows:

$$L_1^{reb} = \frac{1}{b}\sum_{c}^{b} \|y(B_c) - y(R_c)\|, \tag{4}$$

where $y(\cdot)$ represents the Y channel extraction function.

Human and scene prior extraction To fully utilize the adversarial losses, we firstly elaborate the detail of extracting the human and scene prior maps that penalize the input of *body reblur* and *scene reblur* discriminators (refer to Figure 1). The prior is defined as a binary map ($\mathcal{M}$), and its ultimate goal is to find human-motion blur *inside* and *nearby* human body region in an image of our LSPBlur dataset. In detail, this map is firstly obtained by finding human body keypoints from deblurred image D detected using the sophisticated body predictor module of Kanazawa *et al.* [46]. This module extracts the human *body-joint* and *shape* parameters from the input image using the statistical body model [47]. Instead of *shape*, we opt to utilize *body-joint* parameter that extracts 14 body keypoints. These keypoints are then connected with lines to cover the region *inside* human body. We denote this map as *body-joint* map.

To obtain the blurry region *nearby* human body, we firstly find the edge difference between D and R using the Sobel filter. This *difference* map is max-pool-ed to fill the holes and then cropped using the most-top, -right, -bottom, and -left coordinates of the extracted keypoints. We then combine the *difference* and *body-joint* maps to produce single binary map $\mathcal{M}_u$ that fully covers the region *inside* and *nearby* human body. Its reversed version $\mathcal{M}_v$ is utilized to cover the remaining scene region.

Adversarial losses Finally, we determine the discriminator losses to distinguish the real and fake reblurred data. The discriminator losses for the real case of the body, scene, and non-masked (global) images are represented as:

$$L_{\mathbf{R}eal}^{reb} = \frac{1}{3b}\sum_{c}^{b} \Pi_{\mathbf{G}lo}(B_c) + \Pi_u(\mathcal{M}_u \odot B_c) + \Pi_v(\mathcal{M}_v \odot B_c). \tag{5}$$

Similarly, we modeled the losses for fake case in the reblurring as follows:

$$L_{\mathbf{F}ake}^{reb} = \frac{1}{3b}\sum_{c}^{b} \Pi_{\mathbf{G}lo}(R_c) + \Pi_u(\mathcal{M}_u \odot R_c) + \Pi_v(\mathcal{M}_v \odot R_c). \tag{6}$$

Finally, we construct them together as generator and discriminator losses through LSGAN [42] approach as:

$$L_{\mathbf{D}isc}^{reb} = 0.5 \times (\|L_{\mathbf{R}eal}^{reb} - 1\|^2 + \|L_{\mathbf{F}ake}^{reb}\|^2); \tag{7}$$

$$L_{\mathbf{G}en}^{reb} = L_1^{reb} + 0.5 \times (\|L_{\mathbf{F}ake}^{reb} - 1\|^2). \tag{8}$$

Note that the LSGAN implementation is only applied up to this stage. The next meta-learning-based training stages exclude these discriminator modules as both the reblurring and deblurring generator modules are learned.

FIGURE 4. Example of our synthesized blurry results (R) or *pseudo-blurred* on the third row using Ω_T that are **different** with the blurry input (B) in the second row. The first row indicates sharp image input (S).

Pseudo-blur data examples The clear visual representation of our synthesized blur results are demonstrated in Figure 4. Our reblurred results R in the second rows expose different blur patterns than the original blurry input B in the third rows. The scene blur example is shown in the first two columns from the left. The articulated motion blur that receives the full human body from the image input is shown in the last two columns from the right. During training, if a batch is contained with our dataset (LSPBlur), then the global, scene, and human reblur discriminators are utilized. If the HIDE [14] dataset is selected, only the global reblur discriminator is employed. The dataset selection is done randomly at each training iteration.

C. META-TRANSFER-LEARNING FOR DEBLURRING

Until the previous step, our method achieves initial deblurring θ_T and reblurring weights Ω_T, as shown in Figure 2. The next objective is to find an optimized deblurring parameter that is suitable for the $B \rightarrow D \rightarrow R \rightarrow D$ procedures. To achieve it, we apply a meta-transfer-learning operation that seeks to find the initial stable weight θ_M to be transferred during meta-testing. Algorithm 1 explains our approach, where lines 6–17 illustrate the inner loop implementation. The weight is gradually updated via Gradient Descent optimizer with $\alpha = 0.01$ within each task $\mathcal{T}_i$ using $L^{tr}(\theta)$. The meta-learner updates the final weight in line 18 using the average of task-test loss ($\Sigma_i L^{te}(\theta_i)$) that is optimized using ADAM with $\beta = 0.0001$. The large learning rate of α is empirically utilized to obtain fast updates on each task. β is determined with a smaller value to carefully backpropagate through the average of the task-test loss (line 18). The $2\times$ downsamplings are applied in lines 8 and 13 aims to simulate a clear blur difference during augmentation. The $B \rightarrow D \rightarrow R \rightarrow D$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

Algorithm 1 Meta-Transfer Training

Input: Pairs of blurry B and sharp images S from data distribution $\mathcal{D}$, reblurring model Ω_T, and learning rates α, β
Output: Deblurring model θ_M

1: Initialize θ with θ_T
2: Initialize Ω with Ω_T
3: Generate task distribution $p(\mathcal{T})$ from $\mathcal{D}$
4: **while** not done **do**
5: Sample task batch $\mathcal{T}_i^{tr}$ and $\mathcal{T}_i^{te}$ from $p(\mathcal{T})$
6: **for** i **do**
7: **if** $\mathcal{T}_i^{tr}$ **then**
8: Data augmentation : $(B^{tr}, S^{tr}) \rightarrow (B_*^{tr}, S_*^{tr})$
9: Apply proposed sequence : $B_*^{tr} \xrightarrow{\theta} D_{in}^{tr} \xrightarrow{\Omega} R^{tr} \xrightarrow{\theta} D_{out}^{tr}$
10: Evaluate task-training loss : $L^{tr}(\theta) = ||S_*^{tr} - D_{in}^{tr}|| + ||y(S_*^{tr}) - y(D_{out}^{tr})||$
11: Adapt : $\theta_i \leftarrow \theta - \alpha \nabla_\theta L^{tr}(\theta)$
12: **else if** $\mathcal{T}_i^{te}$ **then**
13: Data augmentation : $(B^{te}, S^{te}) \rightarrow (B_*^{te}, S_*^{te})$
14: Apply proposed sequence : $B_*^{te} \xrightarrow{\theta_i} D_{in}^{te} \xrightarrow{\Omega} R^{te} \xrightarrow{\theta_i} D_{out}^{te}$
15: Evaluate task-testing loss : $L^{te}(\theta_i) = ||S_*^{te} - D_{in}^{te}|| + ||y(S_*^{te}) - y(D_{out}^{te})||$
16: **end if**
17: **end for**
18: Update θ_M with respect to average test loss : $\theta_M \leftarrow \theta - \beta \nabla_\theta \Sigma_i L^{te}(\theta_i)$
19: **end while**
20: **return** meta-transferred $\theta : \theta_M$

Algorithm 2 Meta Testing

Input: Blurry image B, meta-transfer trained model θ_M, number of gradient updates n, and learning rate α
Output: Deblurred image D

1: Initialize Ω with Ω_T
2: Initialize θ with θ_M
3: Initial deblurring : $B \xrightarrow{\theta} D$
4: Augmentation : $(B, D) \rightarrow B_*, D_*$
5: Apply sequence : $B_* \xrightarrow{\theta} D_{in} \xrightarrow{\Omega} R \xrightarrow{\theta} D_{out}$
6: **for** n **do**
7: Evaluate test loss : $L(\theta) = ||D_* - D_{in}|| + ||y(D_*) - y(D_{out})||$
8: Update : $\theta_k \leftarrow \theta - \alpha \nabla_\theta L(\theta)$
9: **end for**
10: Final deblur using adapted weight : $B \xrightarrow{\theta_k} D$

is identified at lines 9 and 14 for training ($\mathcal{T}_i^{tr}$) and testing ($\mathcal{T}_i^{te}$) tasks, respectively. The loss functions are compared between the deblurring results ($D_{in}^{tr}, D_{out}^{tr}, D_{in}^{te}, D_{out}^{te}$) with the sharp versions (S_*^{tr}, S_*^{te},) from both training and testing tasks, respectively (Lines 10 and 15 of Algorithm 1).

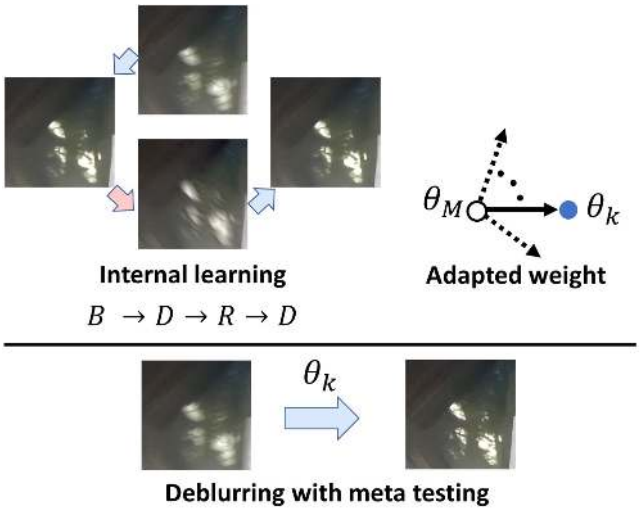

FIGURE 5. The main scheme of our **testing** phase. Test-time adaptations ($\theta_M \rightarrow \theta_k$) are applied to perform the internal learning using the proposed sequence. Final adapted θ_k is used for producing final deblurred output.

TABLE 5. Ablation study results of our approach. The scores are shown to display the pseudo-blur synthesizer's effect in the fine-tuned (-**F**) and meta-learned (-**M**) versions of our deblurring.

Methods	Ours-**0**	Ours-**F**	Ours-**M**(1)	Ours-**M**(5)	Ours-**M**(10)
SSIM	0.7588	0.8235	0.8470	0.8470	0.8472
PSNR	32.2870	33.1490	33.4614	33.4750	33.4776
Time (s)	0.1446	0.1446	5.5567	27.205	54.266
Complex	$O(m)$	$O(m)$	$O(mn+m)$	$O(mn+m)$	$O(mn+m)$

D. META-TESTING FOR DEBLURRING

The objective of this stage is to perform the testing procedure using the self-adapted weight. As illustrated in Figure 5, the meta-testing procedure of Algorithm 2 transfers the weight of a meta-learned position (θ_M) into its new adapted position (θ_k). This procedure is done individually in each input data to induce the self-adaptation capability. Similar to meta-training, our algorithm applies the proposed sequence (Line 5) after the initial deblurring and augmentation. However, in this test stage, where no sharp label is available, we opt to utilize the initial deblurred result D_* as the supervisor. As shown in Line 7 (Algorithm 2), the test-loss $L(\theta)$ is evaluated using the self-extracted data (D_*, D_{in}, D_{out}). The self-adaptation process is performed by gradually updating θ in the iteration (Line 6-9) using $\alpha = 10^{-2}$ via Gradient Descent optimizer similar to the inner-loop scope of Algorithm 1. After a certain number of iterations (n), the adapted weight θ_k is utilized for final deblurring (Line 10).

IV. EXPERIMENT

A. IMPLEMENTATION DETAILS

Our implementations are written using TensorFlow and run on a Titan RTX GPU. The meta-training procedure is fed with 128×128 patches and processed with a mini-batch of 8 ($\mathcal{T}^{tr} = 4$ and $\mathcal{T}^{te} = 4$). The loss functions of $\|*\|$ and $\|*\|^2$ in all equations and algorithms denote standard

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

TABLE 6. Quantitative comparisons using PSNR and SSIM metrics betweeen deblurring algorithms in HIDE [14] test set. The blue- and red-colored texts represent the first and second best results.

Methods	DebGAN-V1 [9]	DebGAN-V2 [10]	DHMP [12]	HIDE [14]	SAPH [13]	Ours-**F**	Ours-**M**(1)
SSIM	0.7424	0.8485	0.8376	0.9310	0.9300	0.8096	0.8319
PSNR	29.17	33.29	33.12	28.89	29.98	32.96	33.24
Complex	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(mn+m)$

TABLE 7. Quantitative comparisons using PSNR and SSIM metrics betweeen deblurring algorithms in GoPro [8] test set. The blue- and red-colored texts represent the first and second best results.

Methods	DebGAN-V1 [9]	DebGAN-V2 [10]	DHMP [12]	HIDE [14]	SAPH [13]	RDeb [18]	Ours-**F**	Ours-**M**(1)
SSIM	0.9580	0.9340	0.9453	0.9400	0.9530	0.9424	0.8073	0.8253
PSNR	28.70	29.55	31.20	30.26	32.02	31.10	32.62	32.85
Complex	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$	$O(mn+m)$

FIGURE 6. Example of deblurring results from our ablation study. Top-left-to-right: blurry input and Ours-**F**. Bottom-left-to-right: Ours-**M**(1) and Ours-**M**(10).

absolute and *mean-squared* errors, respectively. Our whole training scheme involves 3 particular datasets: (i) GoPro [8] and HIDE [14] for deblurring and (ii) LSPBlur for the reblurring module, respectively. Total time required to train the networks from θ_0, Ω_0 to θ_M, Ω_T (from *initial deblurring training* stage up to *meta-transfer deblurring training* stage refer to Figure 2) is 3 days.

For clarity, we re-explain the parameter details in the following. In *initial deblurring* training $(\theta_0, \Omega_0 \rightarrow \theta_T, \Omega_0)$, ADAM with the learning rate of 10^{-4} that is utilized. Same setting with the learning rate of 10^{-4} is also applied for the *reblurring* training $(\theta_T, \Omega_0 \rightarrow \theta_T, \Omega_T)$. In the *meta-transfer* training scheme for deblurring $(\theta_T, \Omega_T \rightarrow \theta_M, \Omega_T)$, the learning rate of $\alpha = 10^{-2}$ with Gradient Descent optimizer is utilized in the inner-loop scope, while $\beta = 10^{-4}$ and ADAM optimizer are determined in the outer-loop, respectively. Finally, during *meta-testing*, the self-adaptation of Algorithm 2 utilized $\alpha = 10^{-2}$ learned via gradient-descent optimizer.

B. ABLATION STUDY

In this work, the key factor that determines the meta-learning performance is the reblurring module. As the main contribution of this paper is the addition of the reblurring task via pseudo-blur synthesizer module, we provide ablation studies regarding its effect in various conditions, namely:

- Ours-**0** ▷ No-reblurrer influence in training and testing stages (training/testing: $B \rightarrow D$).
- Ours-**F** ▷ Reblurrer utilization only in training stage (training: $B \rightarrow D \rightarrow R \rightarrow D$; testing: $B \rightarrow D$).
- Ours-**M** ▷ Full utilization of reblurrer in training and testing stages (training/testing: $B \rightarrow D \rightarrow R \rightarrow D$).

Ours-**F** utilizes a naive-learning strategy, where trained reblurrer Ω_T is utilized to further fine-tunes θ_T (along with *global deblur* discriminator) by providing the additional augmented blurred data R during training. We demonstrate this ablation study on a recent HIDE dataset [14] as it focuses on human and scene motion deblurring cases. We utilize the *long-shot* test-cases of [14] where the scores of each condition is shown in Table 5. Number of adaptation n (scripted in Line 6 of Algorithm 2) is placed next to each Ours-**M** method. In our results, increasing the number of adaptations (n) improves the performance and indicates that Ours-**M** setting succeeds in performing self-adaptation using the meta-transferred weight. The proposed work that utilizes self-adaptation, Ours-**M**, performs superior compared to the naive fine-tuned (Ours-**F**) and the classic versions (Ours-**0**) where the qualitative evidences are shown in Figure 6.

Moreover, to clearly display the performance, we also included the time and complexity scores as shown in Table 5. The complexity formula is defined by big $O(.)$ notation with the parameters of m, which denotes the number of tested data, and n, the number of adaptations. Without any self-adaptation (Ours-**0** and Ours-**F**), the execution is only affected by number of data (m). Each test data $(m = 1)$ requires the computational footprint of 0.1446 seconds in a TITAN RTX GPU for processing a 1280×720 image. On the case of Ours-**M**, the computational time required in performing single-adaptation (Ours-**M**(1) or $n = 1$) is 5.4121 seconds. Thus, the total time required for performing the *adaptations* and *final meta-testing* $(O(mn + m))$ of Algorithm 2 is 5.5567 seconds. As the quantitative scores of Ours-**M**(1), (5), and (10) are marginal (shown in Table 5), we utilize $n = 1$ in the further experiment to avoid large time-consumption. In the next discussions, we elaborate our experiment using the common benchmark dataset (train-available dataset) and recent real-world motion-blurred dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

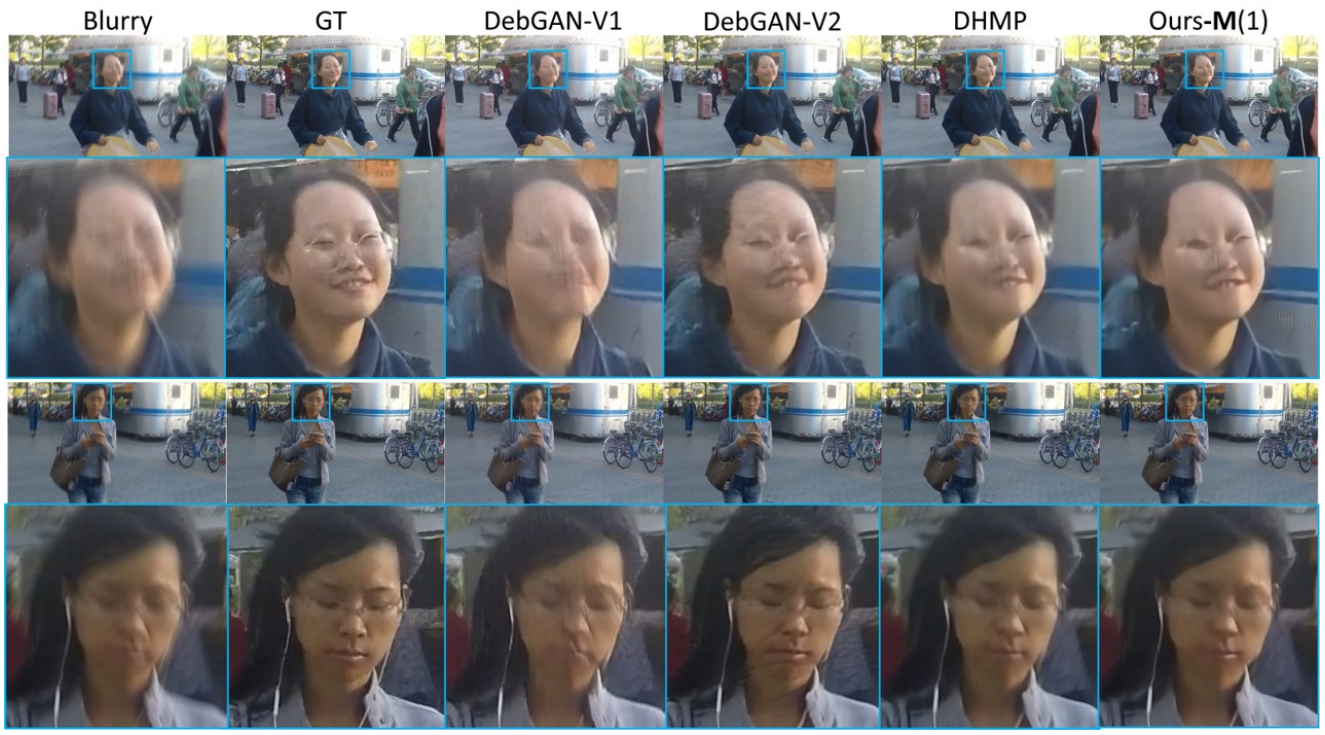

FIGURE 7. Qualitative results of recent deblurring algorithms compared to our approaches using HIDE [14] test set.

FIGURE 8. Qualitative results of recent deblurring algorithms compared to our approaches using GoPro [8] test set.

C. TEST USING TRAIN-AVAILABLE DATASET

In this stage, we demonstrate the performance of our approach along with other state-of-the-art deblurring methods of DebGAN-V1 [9], DebGAN-V2 [10], DHMP [12], HIDE [14], SAPH [13], and RDeb [18]. We utilize the test set of recent blurry human (HIDE) [14], and general (GoPro) [8] dataset for measuring the quantitative performance. The quantitative scores of this experiment are reflected in Table 6 for the HIDE case and Table 7 for the GoPro case. Blue and red colors annotate the top 1 and 2 achievers. The qualitative results are visualized in Figures 7 and 8 for HIDE and GoPro cases, respectively.

In the quantitative measurement, we include: Ours-**F** and Ours-**M**(1), as both approaches have the assistance of the blurry synthesizer. In the HIDE test case, as seen in Table 6, Ours-**M** achieves a high PSNR result compared with the recent algorithms. The score of Ours-**M** is faithful enough as it achieves similar PSNR compared with the recent deep architecture method [12]. Our high quantitative scores on the HIDE case are supported by the results in Figure 7. From the observation on the electronic screen, our qualitative results in Figure 7 are close to DebGAN-V2 [10] in a positive manner. DebGAN-V2 [10] provides clear restored edge output; however, it suffers from an artifact that is seen in the homogeneous region (*e.g.*, face in Figure 7). Since DebGAN-V2 [10] is optimized via dual discriminators, we believe this artifact is a product of the synthesizing procedure rather than the restoration. Our method that is also coupled with multiple discriminators also produces a similar issue; yet, it still preserves realistic output in these regions. This benefit is obtained as our method initially learns the internal features of the input image during the test stage (Line 8 of Algorithm 2).

In the GoPro [8] case, we show that Ours-**M** achieves the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3123059, IEEE Access

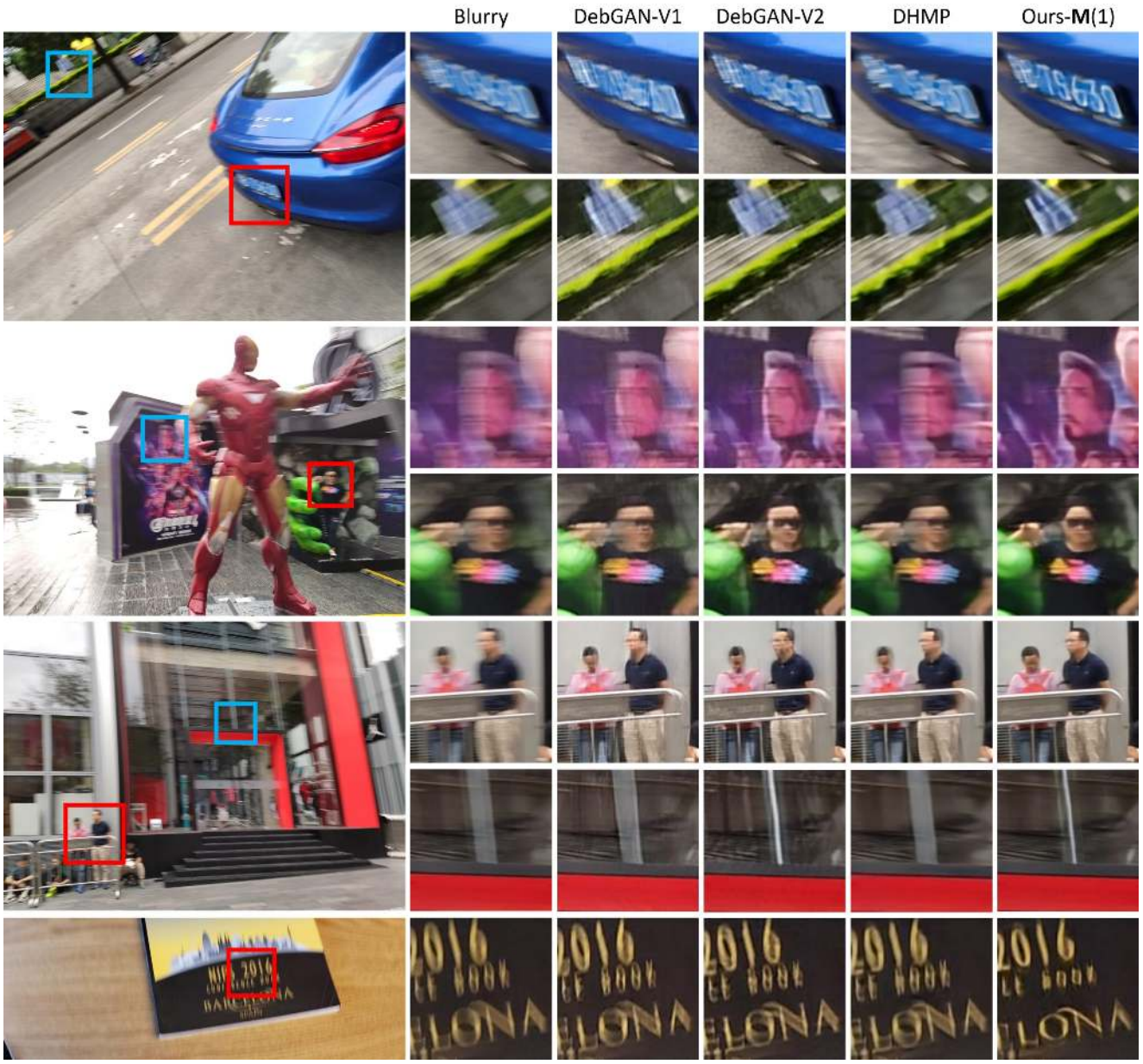

FIGURE 9. Qualitative results of recent deblurring algorithms compared to our approach using in-the-wild blurry data from RWBI [18].

best performance in terms of PSNR score (Table 7). Although both datasets contain humans, the GoPro [8] dataset is closely related to the scene and human-articulated motion blur scenario. This characteristic is obtained because the blur in the GoPro dataset is extracted from the sequential video frames that automatically capture the natural scene and human motion blurs. As shown in Figure 8, all deblurring results in the GoPro case are visually similar. While the proposed work is robust in restoring the blurry input, our deblurring approach came with two limitations, namely: realistic but unreliable structures and longer execution times.

In the first case, our deblurring module is prone to extract structures that are unmatched to the ground truth sharp image, which cost the SSIM scores to be low in all quantitative measurements (Tables 5-7). We believe this anomaly is caused by the GAN method that basically synthesizes the deblurring output rather than restoring it. Our GAN-based approach is crafted with multiple, precisely 4, discriminators that contribute fully to each initial deblurring and reblurring training. Nonetheless, our deblurring approach still produces realistic results with consistent color formation, as expressed by the competitive yet high PSNR scores. As argued in the previous discussion, this achievement is obtained through the internal learning procedure, which includes the pseudo-blur synthesizer role through the reblurring.

In the second case, our method requires longer runtime as it is designed to perform test-time improvements through several n adaptations (our complexity is $O(mn + m)$ as de-

scribed in the ablation study subsection). The other methods have their own unique architectures that are designed to solve dynamic scenes. Our deblurring generator module, on the other hand, is designed equally to DebGAN-V1 [9] which guarantees that our major contributions are located on the reblurrer and the algorithms themselves. Other methods that only rely on single-forward pass surely surpass our runtime since their execution is only affected by the number of test-data m. By utilizing single batch in test-time, the iterations of other methods only reach the complexity of $O(m)$ in Tables 6-7. However, we believe our case with $n = 1$ iteration (Ours-**M**(1)) is still tolerable as our ultimate goal is to invoke the self-adaptive capability of the deblurring module in solving *unknown* data.

As seen in Tables 6-7, the recent deblurring methods are already adjusted to this scenario as HIDE [14] and GoPro [8] test sets are extracted in a similar way to their training sets. Certainly, the compared state-of-the-art works are able to solve them with marginal scores. The recent deblurring methods crafted with sophisticated deep-learning functions are already robust to these benchmarks. To fully witness the self-adaptive capability of our approach, other unknown test cases not relevant to the training set (GoPro [8] and HIDE [14]) are demanded. In the next experiment, we perform an in-the-wild deblurring test using Ours-**M**(1) to tackle this issue.

D. TEST USING IN-THE-WILD DATASET

Our method's superiority is reflected in the scenario of real-world (in-the-wild) blurry data restoration. To demonstrate it, we utilize the recent Real-World Blurry Images (**RWBI**) dataset released by Zhang *et al.* [18], where no training nor sharp ground truth data are available. This dataset is captured using various devices, namely: iPhone XS, Samsung S9 Plus, Huawei P30 Pro, and GoPro Hero 5 Black. The previous deep-learning-based deblurring methods mostly rely on the provided training set. This practice seems limited as the real-world blurry case can differ in terms of motion blur patterns or scales or color distribution.

As shown in Figure 9, other deblurring methods suffer from several issues. To be precise, the method of DebGAN-V1 [9] fails to preserve consistent color information where the deblurred version tends to be brighter than its blurry input. Large-scale blur pattern cases are also not solved using this method. Its improved version, DebGAN-V2 [10], solves the consistency but still lacks to restore large motion blur pattern. Another recent work by DHMP [12] seems to suffer in restoring regions with strong and weak edges. This issue is shown by large and small blur patterns that are failed to be restored by the recent methods.

This phenomenon shows that Ours-**M** is more reliable and adaptive than other methods [9], [10], [12] that still utilize traditional way ($B \rightarrow D$). As shown in Figure 9, our approach is able to restore various cases, including text, human, and other object's motion, with consistent results. By this exploration, we believe that the proposed sequence can be utilized in any deblurring scenario, although the training data is limited. More convincing results are provided in our supplementary video. Readers are encouraged to check all results on an electronic screen.

V. CONCLUSION

In this work, we present an unorthodox approach of deblurring under $B \rightarrow D \rightarrow R \rightarrow D$ strategy, which includes in-the-fly reblurring operation during training and testing stages. To achieve it, we supply our network with a pseudo-blur synthesizer module. The synthesizer acts as a blurry data augmenter, which helps improve the deblurring network's performance. To form a reliable blurry synthesizer, we opt to utilize hand-crafted prior extracted from the human statistical model. Its objective is to let the network learns to produce human-articulated and scene motion blurs simultaneously. This is achieved in the image spatial domain where blurry regions of scene and human are distinguished by the prior.

Finally, we show that by employing the blur synthesizer, the deblurring module learns new blur information, which subsequently improves the performance. This benefit is clearly shown in our experiments, especially in restoring the real-world blurry data. We believe this finding is gainful in deblurring studies where only limited training data is available. Furthermore, our method emboldens future works to shift from traditional way ($B \rightarrow D$) to the more-adaptive proposed sequence ($B \rightarrow D \rightarrow R \rightarrow D$) for deblurring. We leave the study of the best network architecture that fits our sequence as part of future contributions.

REFERENCES

[1] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. on Graphics*, vol. 28, no. 5, pp. 145:1–145:8, Dec. 2009.

[2] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 491–498.

[3] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *European Conference on Computer Vision*, vol. 6311, Sep. 2010, pp. 157–170.

[4] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2011, pp. 233–240.

[5] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, Jun. 2012.

[6] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via L_0-regularized intensity and gradient prior," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 2901–2908.

[7] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Blind image deblurring using dark channel prior," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 1628–1636.

[8] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 257–265.

[9] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8183–8192.

[10] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. of the IEEE International Conference on Computer Vision*, Oct. 2019, pp. 8877–8886.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Dec. 2014, pp. 2672–2680.

[12] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 5978–5986.

[13] M. Suin, K. Purohit, and A. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 3606–3615.

[14] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-aware motion deblurring," in *Proc. of the IEEE International Conference on Computer Vision*, Nov. 2019, pp. 5571–5580.

[15] Z. Shen, W.-S. Lai, T. Xu, J. Kautz, and M.-H. Yang, "Deep semantic face deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8260–8269.

[16] J. S. Lumentut, J. Santoso, and I. K. Park, "Human motion deblurring using localized body prior," in *Asian Conference on Computer Vision*, Nov. 2020.

[17] H. Chen, J. Gu, O. Gallo, M.-Y. Liu, A. Veeraraghavan, and J. Kautz, "Reblur2deblur: Deblurring videos via self-supervised learning," in *IEEE International Conference on Computational Photography*, May 2018, pp. 1–9.

[18] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by realistic blurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 2737–2746.

[19] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. of the 34th International Conference on Machine Learning*, Aug. 2017, pp. 1126–1135.

[20] S. Park, J. Yoo, D. Cho, J. Kim, and T. H. Kim, "Fast adaptation to super-resolution networks via meta-learning," in *Proc. of the European Conference on Computer Vision*, Aug. 2020, pp. 754–769.

[21] M. Choi, J. Choi, S. Baik, T. H. Kim, and K. M. Lee, "Scene-adaptive video frame interpolation via meta-learning," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 9444–9453.

[22] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 3516–3525.

[23] R. W. Liu, W. Yin, S. Xiong, and S. Peng, "Lo-regularized hybrid gradient sparsity priors for robust single-image blind deblurring," in *IEEE International Conf. on Acoustics, Speech and Signal Processing*, Apr. 2018, pp. 1348–1352.

[24] A. Sellent, C. Rother, and S. Roth, "Stereo video deblurring," in *Proc. of the European Conference on Computer Vision*, vol. 9906, Oct. 2016, pp. 558–575.

[25] P. P. Srinivasan, R. Ng, and R. Ramamoorthi, "Light field blind motion deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 2354–2362.

[26] J. S. Lumentut, T. H. Kim, R. Ramamoorthi, and I. K. Park, "Deep recurrent network for fast and full-resolution light field deblurring," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1788–1792, Dec. 2019.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 5967–5976.

[28] W. Ren, J. Yang, S. Deng, D. Wipf, X. Cao, and X. Tong, "Face video deblurring using 3D facial priors," in *Proc. of the IEEE International Conference on Computer Vision*, Oct. 2019, pp. 9387–9396.

[29] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *European Conference on Computer Vision*, vol. 7578, Oct. 2012, pp. 27–40.

[30] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang, "A comparative study for single image blind deblurring," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 1701–1709.

[31] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 6840–6848.

[32] S. Zhang, Y. Lin, and H. Sheng, "Residual networks for light field image super-resolution," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 11 046–11 055.

[33] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, Dec. 2016, pp. 3630–3638.

[34] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, Dec. 2017, pp. 4077–4087.

[35] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, Jun. 2016, pp. 1842–1850.

[36] B. Oreshkin, P. Rodríguez López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems*, vol. 31, Dec. 2018, pp. 721–731.

[37] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *International Conference on Learning Representation*, Apr. 2018.

[38] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," in *International Conference on Learning Representation*, Apr. 2018.

[39] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," in *International Conference on Learning Representation*, Apr. 2018.

[40] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp. 9137–9146.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[42] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. of the IEEE International Conference on Computer Vision*, Oct. 2017, pp. 2813–2821.

[43] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proc. of the British Machine Vision Conference*, Aug. 2010, pp. 1–11.

[44] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proc. of the IEEE International Conference on Computer Vision*, Oct. 2017, pp. 2353–2362.

[45] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8340–8348.

[46] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7122–7131.

[47] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: a skinned multi-person linear model," *ACM Trans. on Graphics*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.