Universal Framework for Joint Image Restoration and 3D Body Reconstruction

JONATHAN SAMUEL LUMENTUT[1], **(Member, IEEE), MATTHEW MARCHELLUS**[1],
JOSHUA SANTOSO[1], **TAE HYUN KIM**[2], **(Member, IEEE), JU YONG CHANG**[3],
AND IN KYU PARK[1], **(Senior Member, IEEE)**

[1]Department of Information and Communication Engineering, Inha University, Incheon 22212, South Korea
[2]Department of Computer Science, Hanyang University, Seoul 04763, South Korea
[3]Department of Electronics and Communication Engineering, Kwangwoon University, Seoul 01897, South Korea

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

ABSTRACT Recent works have demonstrated excellent state-of-the-art achievements in image restoration and 3D body reconstruction from an input image. The 3D body reconstruction task, however, relies heavily on the input image's quality. A straightforward way to solve this issue is by generating vast degraded datasets and using them in a re-finetuned or newly-crafted body reconstruction network. However, in future usage, these datasets may become obsolete, leaving the newly-crafted network outdated. Unlike this approach, we design a universal framework that is able to utilize prior state-of-the-art restoration works and then self-boosts their performances during test-time while jointly carrying out the 3D body reconstruction. The self-boosting mechanism is adopted via test-time parameter adaptation capable of handling various types of degradation. To accommodate, we also propose a strategy that generates pseudo-data on the fly during test-time, allowing both restoration and reconstruction modules to be learned in a self-supervised manner. With this advantage, the universal framework intelligently enhances the performance without any new dataset or new neural network model involvement. Our experimental results show that using the proposed framework and pseudo-data strategies significantly improves the performances of both scenarios.

INDEX TERMS Restoration, deblur, super-resolution, denoising, 3D body reconstruction, meta-learning, self-adaptive, pseudo-data.

I. INTRODUCTION

The task of 3D body reconstruction is gaining popularity in recent times. The 3D version of the human body can be parsed directly using only a single image as input. Various works [1]–[8] have been explored to achieve this objective. Unfortunately, these methods rely heavily on the input data, namely image, that should satisfy the clean scenario. Degraded image case affects negatively unto the body reconstruction output, and most of the state-of-the-art works exclude this constraint.

One may solve this issue straightforwardly by synthesizing a large corrupted image dataset and utilizing it in

The associate editor coordinating the review of this manuscript and approving it for publication was Bidyadhar Subudhi.

a fine-tuned or newly re-created body network. Another solution is to generate the corrupted images using physical tools [9] and annotate the 3D body data via external devices [10]. However, this pre-generated dataset approach comes with the cost of high labor work while still unable to support the network in solving the constantly-change real-world data. This situation subsequently leaves the newly crafted network to be outdated. To tackle this issue, the recent trend is shifted to the life-long learning direction, which can be applied via test-time adaptations instead of running a single-forward pass algorithm that only depends on the pre-generated dataset. Recent 3D body reconstruction work [11] followed this pathway by self-adapting their network using its test data via meta-learning [12]. However, again, it is still constrained to the clean image scenario. Thus, solving

restoration and reconstruction jointly and adaptively is still an open matter.

With the challenges exposed in the previous discussion, we propose a joint restoration and reconstruction framework that solely relies on the input data processed through a test-time self-improvement fashion. Instead of crafting a genuine neural network architecture, we construct a framework in the algorithm forms capable of plugging any deep-learning-based image restoration modules while jointly tasked with the 3D body reconstruction module. For that very reason, we define our framework as a universal one. To achieve the test-time improvement, we present a self-adaptation algorithm that fits both modules and optimizes them using only the input data. This strategy is made possible by borrowing the concept of model agnostic meta-learning (MAML) [12] and adjust it within our algorithm.

Without any new dataset acquired, one might question on how the self-adaptive algorithm works seamlessly. In this work, we introduce the readers to a term coined as *pseudo-data* which is generated on-the-fly during the test-time and aimed to assist our algorithm. Our straightforward yet unique approach for acquiring the pseudo-data is meant to substitute the new dataset approach itself. The pseudo-data, generated directly from the test input image, is split into pseudo-clean and pseudo-corrupted information. Their elaborations, along with our algorithms, are furthermore discussed in the *Method* section. The scope of our study in the restoration case involves 3 major degradation-solver, namely: *denoising*, *deblurring*, and *super-resolution* (SR). Each of these scenarios is worked jointly with the 3D body reconstruction under the universal framework.

In the experimental procedures, we show that the test-time self-adaptive capability is preserved even though the joint scenario is applied, yielding better results in both quantitative and qualitative aspects. Furthermore, we analyze the characteristic of each restoration scenario along with the 3D body reconstruction results. With this finding, future joint framework applications may utilize this information for further improvements. In summary, we define our contributions as follows:

- We introduce a modular-based plug-and-play universal framework (PPUF) encapsulated within a self-adaptive algorithm capable of receiving any deep-learning-based image restoration methods while jointly carrying the 3D body reconstruction task.
- We introduce the utilization of pseudo-data in test-time as an alternative yet cheap way than generating a new vast dataset to support our algorithm.
- We show that using the proposed algorithm and pseudo-data alone within the joint framework, both restoration and 3D body reconstruction modules work simultaneously while producing significant quantitative scores and visual quality improvements.

II. RELATED WORKS
A. IMAGE RESTORATION AND 3D BODY RECONSTRUCTION

The work of image restoration has been explored since decades ago. The arts of restoration are mainly categorized into deblurring, denoising, and SR scenarios. In the deblurring case, most works focused on motion deblurring tasks. These studies have been evolved from traditional blur kernel prediction based [13]–[15] up to the recent kernel-prediction-free method via deep learning [16]–[18], particularly GAN strategy [19]. The trends followed by equipping them with real-world priors such as human faces [20], [21] and bodies [22]–[24]. Denoising and SR works are also evolving these days. Reference [25] translated the cleaned version of an image directly from its noisy input. This approach is made possible by providing the pairs of clean and noisy images as a training dataset. Many works [26]–[29] then followed the same strategy with many modifications on the neural network level. In the SR case, the trends were moving from the hand-crafted function-based, such as: random forest [30], [31] and decision tree [32] approaches up to the common deep-learning utilization [33]–[35]. Although this work is mainly implemented for the deblurring, SR, and denoising scenarios, we believe that other restoration, such as dehazing [36], [37] and de-raining [38], [39] are also applicable for future implementation.

3D body reconstruction, on the other hand, is a computer vision task that reconstructs a 3D human body model from an input image. Most works are evolved by using the statistical human body prior of [40] which is directly regressed via hand-crafted functions [5], or even deep-learning [2], [6]–[8], [41] approaches. The earlier hand-crafted approach by Bogo *et al.* [5] utilizes the initial output of the estimated 2D body joints to regress the statistical body prior final output. The following work by Kanazawa *et al.* [6] embraced the deep learning-based implementation via generative adversarial network [19]. Their work [6] utilized specific discriminators that constraint the 3D body reconstruction results within the boundary of realistic human body poses. Its improved version [41] is aimed to hallucinate temporal-based 3D body outputs with plausible movements. Kolotouros *et al.* [8] introduced a straightforward refinement from the work of [6] via body-fitting mechanism during training. The work of Moon and Lee [2] surpassed their performances by utilizing the line-based pixel (lixel) information to better regresses the 3D body prior output. In recent times, some additional works [42]–[44] are tasked to refine temporal-based 3D body reconstruction works via (i) gated recurrent units that connect relatable body features in sequential frames [42], (ii) additional module that penalizes outlier body poses in adjacent frames [43], and (iii) temporal re-weighting strategy that balances the sequential frames' features [44]. Unlike these human prior-based approaches, Saito *et al.* [4] avoided the prior utilization by directly employing the pixel-aligned
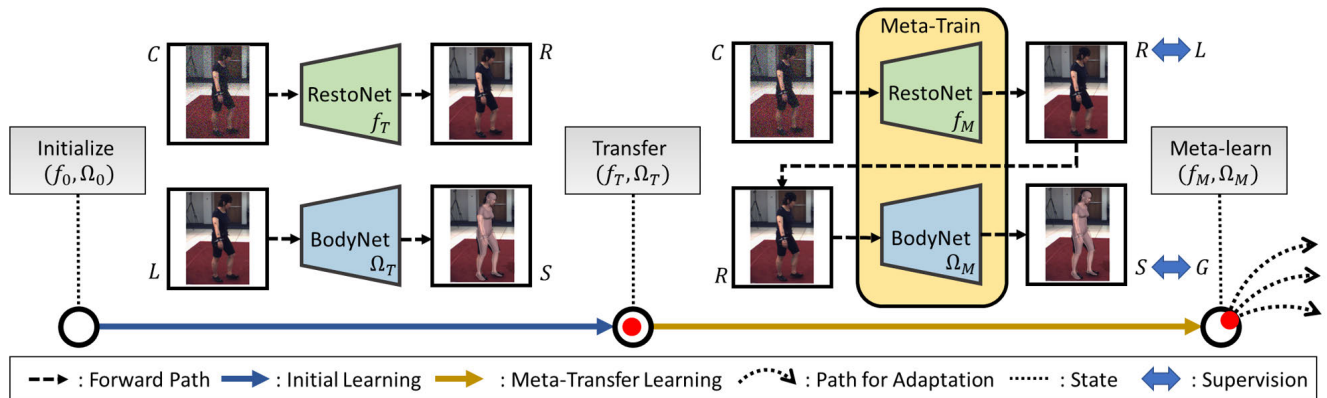
FIGURE 1. The figure above depicts the overall **training** scheme for our framework. The procedure is started by the **initial learning** denoted by the blue arrow. The transferred weight is then trained under the meta-transfer learning (**universal meta training**) scheme denoted by the brown arrow.

implicit function to predict the 3D cloth-textured human body output.

B. LEARNING STRATEGIES FOR COMPUTER VISION TASKS

Unlike the discussions above, some arts mainly focus on strategies that boost the performance. The early work by [45] utilized on-demand learning to study multiple restoration cases such as in-painting, pixel interpolation, deblurring, and denoising being solved separately. Reference [46] then introduced a reinforcement learning-based restoration that is able to solve combined cases automatically. Their cases include deblurring, de-raining, denoising, and de-JPEG-ing. The work by [47] constructs an architecture that also solves a similar scenario intending to improve the recognition task. Recent trends move toward the adaptive learning strategy to achieve the life-long demand. This demand emphasizes the capability of any computer vision function to adapt to the *unknown* test data instead of relying on the pre-generated dataset that might become obsolete in future usage. To achieve it, recent works exploit the model agnostic meta-learning [12] algorithm that is known to induce the self-adaptive capability. These particular works in SR [48]–[50], motion-deblurring [51], video interpolation [52] and body reconstruction [11] utilize and adjust the meta-learning algorithm to boost their performances in the test phase.

Inspired by these studies, our work is focused on boosting the joint work of restoration and 3D body reconstruction without any modification in the network level and additional dataset. We describe our strategies in the following section.

III. METHOD

Our PPUF is built by a restoration module (RestoNet (f)) followed by a 3D body reconstruction module (BodyNet (Ω)), which outputs both restored images and reconstructed bodies, respectively. It is trained into 2 progressive stages, namely: (i) *initial learning*, denoted as blue arrow and (ii) *meta-transfer learning*, denoted as brown arrow in Figure 1. Once trained, the PPUF is run through our *meta-testing* algorithm to
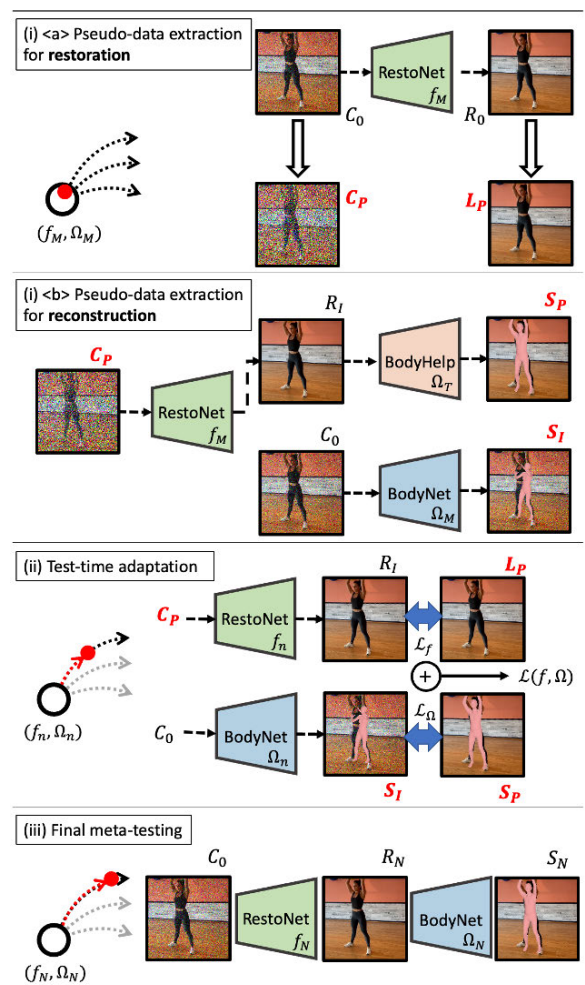

FIGURE 2. Our universal meta testing scheme, which executes *pseudo-data extraction*, *test-time adaptation*, and *final testing* procedure using the adapted weights. Blue arrows indicate the supervision function for loss calculation.

perform test-time self-improvement, as reflected in Figure 2. We elaborate the details from training to testing procedures on the following subsections:

A. INITIAL LEARNING

This stage is regarded as the initial procedure of our scheme. The aim of this training is to transfer the untrained weights to their initial trained versions $((f_0, \Omega_0) \rightarrow (f_T, \Omega_T))$ that are robust in solving restoration as well as 3D body reconstruction. The training involves only the dataset of Human3.6M [10] as it provides the human keypoint ground truth while we carefully degraded the images according to the degradation scenario: noisy, blurry, and low-resolution. In the noisy case, we determine the range of sigma noise randomly between 10 to 20 during training. In the deblurring case, fixed Gaussian blur with the window size of 5×5 and randomized standard deviation (σ) between 0.05 to 1.2 are utilized. Finally, for the case of SR, we determine the case of $2\times$ downsampling in our study.

B. UNIVERSAL META-TRAINING

The next stage is to perform the meta-transfer training that utilizes the output weights f_T, Ω_T of *initial learning*. This training is aimed to provide initial stable weights of both modules that are ready to be adapted during meta-testing. Our strategy in performing this training is visualized in Figure 1 under the chunk of the brown directional arrow $(f_T, \Omega_T) \rightarrow (f_M, \Omega_M)$. This figure is translated directly to our implementation in Algorithm 1.

Algorithm 1 followed the meta-learning [12] concept as it is constructed by 2 important snippets: inner-loop (Line 8-12) and outer-loop (Line 14-18) operations. The idea of this training is to explore the best weights position for test-time adaptation. To achieve this goal, our the training algorithm utilizes 2 data samples, namely task-training $(\mathcal{T}^{tr})$ and task-testing $(\mathcal{T}^{te})$ batches sampled from a data collection $p(\mathcal{T})$. $\mathcal{T}^{tr}$ and $\mathcal{T}^{te}$ are utilized within the inner- and outer-loop scopes of Algorithm 1, respectively. Each time the networks are optimized within the inner-loop scope (Line 11-12), their behaviors are supervised by the outer-loop optimizer (Line 17-18).

From this point forward, we provide the detailed elaboration of Algorithm 1. Algorithm 1 fed the input data of corrupted image C, clean image L, and the body ground truth (3D and 2D keypoints) G. These data are re-sampled for the $\mathcal{T}^{tr}$ and $\mathcal{T}^{te}$ batches (Line 4-6). The process is then continued directly to the inner-loop scope with a pre-determined number of iterations i (Line 8-12). Our algorithm restores the image inside the inner-loop to obtain R and reconstruct the 3D body parameters S from the restored image R. The next step is to perform a loss calculation (Line 10) that combines restoration loss $\mathcal{L}_f$ and body reconstruction loss $\mathcal{L}_\Omega$. In details, $\mathcal{L}_f$ is defined by:

$$\mathcal{L}_f(L, R) = \|L - R\|^2. \tag{1}$$

Similarly, $\mathcal{L}_\Omega$ is represented by:

$$\mathcal{L}_\Omega(S, G) = \|P_3(S) - P_3(G)\|^2 + \|P_2(S) - P_2(G)\|^2, \tag{2}$$

Algorithm 1 Universal Meta Training

Input: Collection of corrupted C, clean images L, body ground truth G from data distribution $\mathcal{D}$, restoration model f_T, and 3D body reconstruction model Ω_T
Parameter: Learning rates α, β
Output: Meta-transferred weights of the restoration and reconstruction models (f_M, Ω_M)

1: Initialize f, Ω with f_T, Ω_T.
2: Generate task distribution $p(\mathcal{T})$ from $\mathcal{D}$.
3: **while** not done **do**
4:     Sample task batch $\mathcal{T}^{tr}$ and $\mathcal{T}^{te}$ from $p(\mathcal{T})$.
5:     Obtain data C_a, L_a, G_a from $\mathcal{T}^{tr}$.
6:     Obtain data C_b, L_b, G_b from $\mathcal{T}^{te}$.
7:     **for** i **do**
8:         Restore : $f(C_a) = R_a$.
9:         Reconstruct : $\Omega(R_a) = S_a$.
10:        Eval : $\mathcal{L}^{tr}(f, \Omega) = \mathcal{L}_f(L_a, R_a) + \lambda\mathcal{L}_\Omega(S_a, G_a)$.
11:        Update : $f_i \leftarrow f - \alpha\nabla_f\mathcal{L}^{tr}(f, \Omega)$.
12:        Update : $\Omega_i \leftarrow \Omega - \beta\nabla_\Omega\mathcal{L}^{tr}(f, \Omega)$.
13:    **end for**
14:    Restore : $f(C_b) = R_b$.
15:    Reconstruct : $\Omega(R_b) = S_b$.
16:    Eval : $\mathcal{L}^{te}(f, \Omega) = \mathcal{L}_f(L_b, R_b) + \lambda\mathcal{L}_\Omega(S_b, G_b)$.
17:    Update : $f_M \leftarrow f - \alpha\nabla_f\mathcal{L}^{te}(f, \Omega)$.
18:    Update : $\Omega_M \leftarrow \Omega - \beta\nabla_\Omega\mathcal{L}^{te}(f, \Omega)$.
19: **end while**
20: **return** meta-transferred : f_M, Ω_M

---

where $P_3(.)$ and $P_2(.)$ denote the 3D keypoints and 2D keypoints extraction functions, obtained from the body parameters of Skinned-Multi Person Linear model (SMPL) [40] using the 3D body reconstruction network Ω [6]. Variable a and b represent the data batch index inside the inner/outer-loop scope, respectively. Instead of using gradient descent in the inner loop, we utilize the ADAM method with certain learning rates α, β for the restoration and reconstruction cases, respectively. Once the inner-loop operation is finished, the process is progressed through the outer-loop algorithm with similar procedures (Line 14-18).

C. UNIVERSAL META-TESTING

The meta-testing is the main actor in executing the self-improvement strategy. Our meta-testing in the joint framework is reflected in Algorithm 2. Its scheme is shown in Figure 2 where 3 sub-procedures are run progressively: (i) *pseudo-data extraction* followed by (ii) *test-time adaptation*, and the (iii) *final meta-testing*. These procedures correspond directly to the Algorithm 2 in Lines 3-5, 6-10, and 11-12, respectively.

The *pseudo-data extraction* stage is important as it provides a cue to perform self-adaptation of the restoration and reconstruction modules. In our approach, the pseudo-data is constructed by 2 particular information: *pseudo-clean*

Algorithm 2 Universal Meta Testing

Input: corrupted image C, meta-trained model (f_M, Ω_M)
Parameter: Learning rates α, β, maximum iterations N
Output: Self-improved restored image R and reconstructed 3D body S

1: Initialize f, Ω with f_M, Ω_M.
2: Initial restoration: $f(C_0) = R_0$.
3: Extract pseudo-images : $C_0, R_0 \rightarrow C_P, L_P$.
4: Intermediate restoration: $f(C_P) = R_I$.
5: Extract pseudo-bodies : $\Omega_T(C_0) = S_I, \Omega_T(R_I) = S_P$.
6: **for** $n \leq N$ **do**
7:     Eval : $\mathcal{L}(f, \Omega) = \mathcal{L}_f(R_I, L_P) + \lambda \mathcal{L}_\Omega(S_I, S_P)$.
8:     Update : $f_n \leftarrow f - \alpha \nabla_f \mathcal{L}(f, \Omega)$.
9:     Update : $\Omega_n \leftarrow \Omega - \beta \nabla_\Omega \mathcal{L}(f, \Omega)$.
10: **end for**
11: Final restoration : $f_N(C_0) = R_N$.
12: Final reconstruction : $\Omega_N(R_N) = S_N$

FIGURE 3. Re-corrupting the test input data (C_0) in the super-resolution case. This chunk can be substituted directly to the *pseudo-data extraction for restoration* step in Figure 2.

FIGURE 4. Re-corrupting the test input (C_0) data in the deblurring case. This chunk can be substituted directly to the *pseudo-data extraction for restoration* step in Figure 2.

and *pseudo-corrupted* data. The restoration is supported by the **pseudo-corrupted image** C_P and **pseudo-clean image** L_P while the reconstruction module is supported by the **pseudo-corrupted body** S_I and **pseudo-clean body** S_P. These pseudo-corrupted data are treated as input data, while the pseudo-clean data is regarded as the label information during self-supervised learning. In the following discussion, we describe our strategies to extract this information for various restoration cases. For simplicity, we provide the example of the denoising case in Figure 2.

- **Denoising case** For the denoising case, we extract C_P by adding a consistent severe noise to the original corrupted input image (C_0). To be precise, we add noise to the original noisy input (C_0) image with the σ level of 20 so that it becomes severely corrupted ($C_0 \rightarrow C_P$). This image is then treated as the pseudo-corrupted image (C_P), while the intermediate denoised image from the input image (R_0) is treated as the pseudo-clean image (L_P) (reflected in Line 2-3 in Algorithm 2). The pseudo-clean and pseudo-corrupted bodies are extracted from the restored version of R_1 (restored from C_p) and its initial corrupted input C_0, respectively. The visualization of these approaches are shown in Figure 2 within the chunks of *pseudo-data extraction for* **restoration** and *pseudo-data extraction for* **reconstruction**.

- **Super-resolution case** In the super-resolution case, the main objective is to super-resolved low-resolution image input to a desired $2\times$ spatial dimension. Its pseudo-corrupted image (C_P) is obtained by downscaling the low-resolution input by $2\times$. The input is then treated as pseudo-clean image L_P. With this approach, the pseudo-corrupted image C_P has the size of $h/2 \times w/2$, which is $0.5\times$ smaller than the original low-resolution ($h \times w$) input image (treated as L_P), and $0.25\times$ than the original desired output ($2h \times 2w$). Note that, in the SR approach, the pseudo-clean S_P and pseudo-corrupted S_I
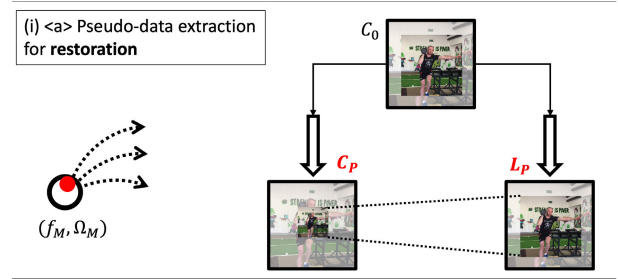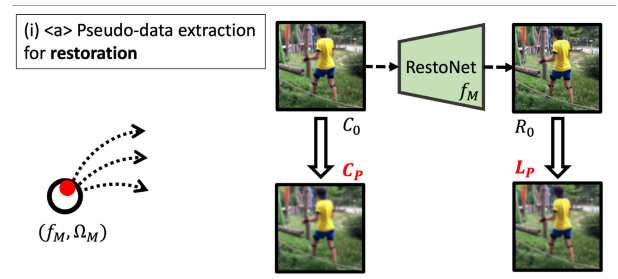
bodies are extracted by the re-upsampled version of L_P ($2\times$) and C_P ($4\times$), respectively. The re-upsampling process is applied via classical bilinear interpolation. This approach is visualized in Figure 3 and can be utilized for replacing the *pseudo-data extraction for restoration* chunk of Figure 2.

- **Deblurring case** In the deblurring case, we firstly obtain the initial deblurring result of the blurry input image ($C_0 \rightarrow R_0$). Both the blurry input and the deblurring result images are then manually blurred with Gaussian degradation. Specifically, we set the kernel window of 5 with $\sigma = 0.8$. The degraded version of blurry input ($C_0 \rightarrow C_P$) and the intermediate deblur (R_0) result are treated as pseudo-corrupted (C_P) and pseudo-clean images (L_P), respectively. Similar to the denoising case, the pseudo-clean and pseudo-corrupted bodies are extracted from the restored version of R_1 (restored from C_p) and its initial corrupted input C_0, respectively. This approach is visualized in Figure 4 and can also be substituted with the *pseudo-data extraction for restoration* chunk of Figure 2.

Take note that in the case of generating the pseudo-body, we make use of another identical 3D body reconstruction network (BodyHelp in Figure 2) to produce the pseudo-clean body S_P from the intermediate restored version R_I (Line 4 and second term of Line 5 of Algorithm 2). The pseudo-corrupted body is extracted by using the initial corrupted input C_0 without being restored (first term of Line 5 of Algorithm 2). The BodyHelp module itself is excluded for optimization during
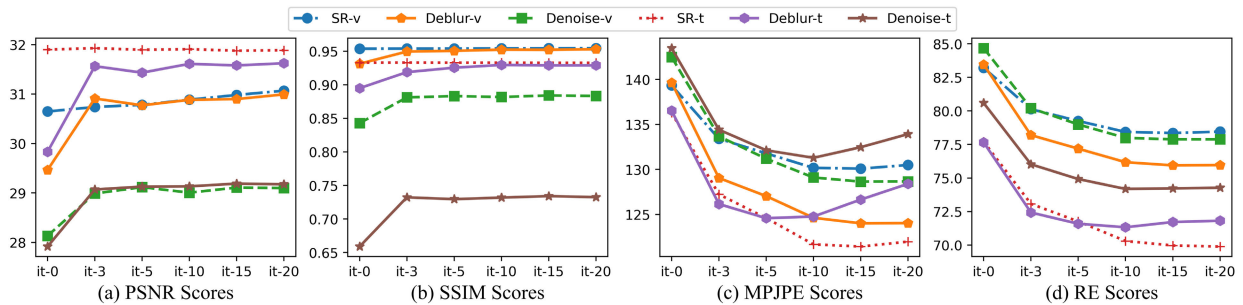
FIGURE 5. The plots above are the summary of the quantitative results in performing test-time adaptation for the tasks of image restoration ((a)-(b)) and 3D body reconstruction ((c)-(d)) in 3DPW benchmark cases. The scores are represented in vertical axes. Our method clearly boosts the performance without modifying the original architectures of the respective modules.

the test-time backpropagation. Once the pseudo-data is ready, the algorithm performs the adaptation procedure that is run within a number of iteration (N), as highlighted in Line 6-10 in Algorithm 2. The self-adaptation test loss (Line 7) is performed by utilizing the same formula of Eqs. 1 and 2. The adaptation is illustrated in the *test-time adaptation* chunk of Figure 2 where the losses calculation is jointly summed, and only the RestoNet and BodyNet are being backpropagated. In our implementation, we simply utilize an ADAM optimizer with the learning rates of each respected algorithm (described in *Experiment* section). After the self-adaptation procedure, both end-to-end networks (f, Ω) secure their adapted position. As illustrated in Figure 2, the red-dot object (f_N, Ω_N) is already transferred to its new location that is adapted according to the pseudo-data. The final restoration and reconstruction are performed in Line 11-12 to produce R_N, S_N.

IV. EXPERIMENT

A. IMPLEMENTATION SETTINGS

In our experiments, several backbone methods are used for the restoration procedures, namely: URIE [47] (deblurring and denoising cases) and EDSR [35] (SR case), which are available in PyTorch. For the backbone of the 3D body reconstruction function, we opt to utilize the PyTorch-based method of SPIN [8] without its additional Simplify [5] optimization, which essentially equals to HMR [6]. These PyTorch-based methods were elected as our algorithms are scripted using the Pytorch library. Human3.6M [10] is the *sole* dataset utilized in training the whole framework. The human image of Figures 1 and 2 are obtained through HumanEVA [53] and InstaVariety [6] dataset for clearer illustration sake. During training, both networks fed the patch of 224×224 following the strict requirement of HMR [6] that is plugged into our BodyNet module. For deblurring and denoising cases, α (learning rate for restoration) is set to 1e-03 using URIE [47] method while the SR case utilizes α=1e-07 as presented in EDSR [35] method. For the 3D body reconstruction, β (learning rate for reconstruction) and weighted constant λ are set to 1e-05 and 5.0 respectively, following the original work [6]. Batch of data sampled, $p(\mathcal{T})$ is set to 16 with 50:50 ratio divided to $\mathcal{T}^{tr}$ (batch size of 8) and $\mathcal{T}^{te}$ (batch size of 8). $\mathcal{T}^{tr}$ and $\mathcal{T}^{te}$ are utilized within

TABLE 1. PSNR scores of PPUF in various restoration cases.

Cases	SR-v	Deblur-v	Denoise-v	SR-t	Deblur-t	Denoise-t
Initial	30.6478	29.4648	28.1327	31.8992	29.8329	27.9155
Iter-3	30.7388	30.9102	28.9888	**31.9286**	31.5639	29.0698
Iter-5	30.7827	30.7722	**29.1169**	31.8953	31.4328	29.1280
Iter-10	30.8868	30.8829	29.0050	31.9074	31.6116	29.1336
Iter-15	30.9824	30.8987	29.1085	31.8777	31.5794	**29.1886**
Iter-20	**31.0680**	**30.9939**	29.0982	31.8866	**31.6236**	29.1745

inner- and outer-loop operations, respectively. The data batch in $\mathcal{T}^{tr}$ are different from the one in $\mathcal{T}^{te}$. The total required time for the training scheme is around ~4 days using a Titan RTX GPU. In testing, the maximum iteration N is set to 20.

B. PERFORMANCE ANALYSIS

1) PRELIMINARY

We provide a vast benchmarking experiment using the 3DPW [54] dataset. We make use of the *validation* (**-v**) and *test* (**-t**) sets of 3DPW as they provide the human images and equipped with the body ground truth such as 3D key points and SMPL [40] parameters. The mentioned annotations are utilized for calculating the *mean per joint position error* (MPJPE) and *reconstruction error* (RE) commonly used in SMPL-based benchmarking. Lower error scores of these metrics indicate better results. Note that the training set of 3DPW [54] is even excluded during the learning process of PPUF. The images on the benchmarked set are pre-degraded randomly before the testing procedure is run. For the noisy scenario, the noises are varied using the sigma values from 10 to 20, and they are added directly to the image. For the blur scenario, we determine the Gaussian sigma randomly for 0.05 to 1.2 with the kernel size of 5. The blur is convolved directly in each benchmarked image. Finally, on the low-resolution scenario, we determine the case of 2× downsampling following the recent studies [48], [49] that are directly applied to the input image. The total benchmark data of the 3DPW-validation (3DPW-v) and 3DPW-test (3DPW-t) cases are 10,412 and 35,515 images. The restored images are then compared with the clean ground truth by the metrics of *peak signal-to-noise ratio* (PSNR) and *structural similarity index* (SSIM). Higher scores indicate better results.

2) JOINT DENOISING AND 3D BODY RECONSTRUCTION

We begin our analysis by interpreting the results of restoration as it is run before the reconstruction task. In the case of

TABLE 2. SSIM scores of PPUF in various restoration cases.

Cases	SR-v	Deblur-v	Denoise-v	SR-t	Deblur-t	Denoise-t
Initial	0.9536	0.93129	0.84285	0.93288	0.89488	0.65891
Iter-3	0.95381	0.94945	0.88113	**0.93302**	0.91880	0.73227
Iter-5	0.95391	0.95053	0.88311	0.93287	0.92537	0.72953
Iter-10	0.95414	0.95225	0.88176	0.93289	**0.92951**	0.73190
Iter-15	0.95433	0.95204	**0.88409**	0.93273	0.92881	**0.73409**
Iter-20	**0.95451**	**0.95291**	0.88334	0.93274	0.92881	0.73255

TABLE 3. MPJPE scores of PPUF in various restoration cases.

Cases	SR-v	Deblur-v	Denoise-v	SR-t	Deblur-t	Denoise-t
Initial	139.337	139.653	142.485	136.294	136.529	143.481
Iter-3	133.410	129.045	133.681	127.230	126.154	134.415
Iter-5	131.786	127.033	131.192	124.591	**124.576**	132.132
Iter-10	130.174	124.625	129.104	121.668	124.763	**131.289**
Iter-15	**130.086**	**124.007**	128.642	**121.425**	126.648	132.471
Iter-20	130.491	124.034	128.661	121.974	128.398	133.928

TABLE 4. RE scores of PPUF in various restoration cases.

Cases	SR-v	Deblur-v	Denoise-v	SR-t	Deblur-t	Denoise-t
Initial	83.215	83.445	84.674	77.708	77.650	80.589
Iter-3	80.129	78.193	80.199	73.073	72.426	76.020
Iter-5	79.247	77.184	78.990	71.763	71.590	74.920
Iter-10	78.432	76.176	77.987	70.303	**71.325**	**74.183**
Iter-15	**78.349**	**75.942**	77.879	69.968	71.718	74.223
Iter-20	78.451	75.961	**77.876**	**69.889**	71.814	74.270

denoising, the restoration is consistently improved during test time. This argument is clearly visible in our quantitative scores that are presented in Tables 1 and 2 which denote the PSNR and SSIM scores, respectively. Here, the denoising cases are shown in the columns of **Denoise-v** and **Denoise-t** where their data is obtained from the *validation* and *test* sets of the images in 3DPW [54]. The degraded images are added with the noise as previously discussed in the preliminary section. Thanks to the pseudo-clean and pseudo-corrupted data generated in test-time, our PPUF is able to improve the performance by maximum +**0.9842 dB** (*Denoise-v*) and +**1.2731 dB** (*Denoise-t*) from the initial denoising outputs. Similarly, the SSIM gains about +**0.04124** (*Denoise-v*) and +**0.07518** (*Denoise-t*). The interesting finding here is that the performance is relatively stable, although the iterations are increased. Using the PPUF, no PSNR nor SSIM bounce back to the initial scores. For clarity, these PSNR and SSIM scores are re-plot in Figure 5 (a)-(b). The graph of Figure 5 (a)-(b) clearly demonstrate that the scores of *Denoise-v* and *Denoise-t* are vastly improved between the initial 3 iterations, followed by a slight yet consistent improvement when the iterations are increased.

The latter analysis is then aimed at the 3D body reconstruction result that is coupled with the denoising task. As shown in Table 3, the MPJPE scores are suppressed with the margin of −**13.843 mm** (*Denoise-v*) and −**12.192 mm** (*Denoise-t*) from the initial 3D body reconstruction outputs. Using the PPUF, the MPJPE also hold the same characteristic with the restoration module as the scores jump around −**8.804 mm** (*Denoise-v*) and −**9.066 mm** (*Denoise-t*) with only 3 iterations. Similar and consistent results are also reflected in the RE performance as seen in Table 4. Maximum errors to be suppressed are −**6.798 mm** and −**6.406 mm** for *Denoise-v* and *Denoise-t* cases. Within only 3 iterations, the RE scores are suppressed by −**4.475 mm** (*Denoise-v*) and −**4.569 mm**
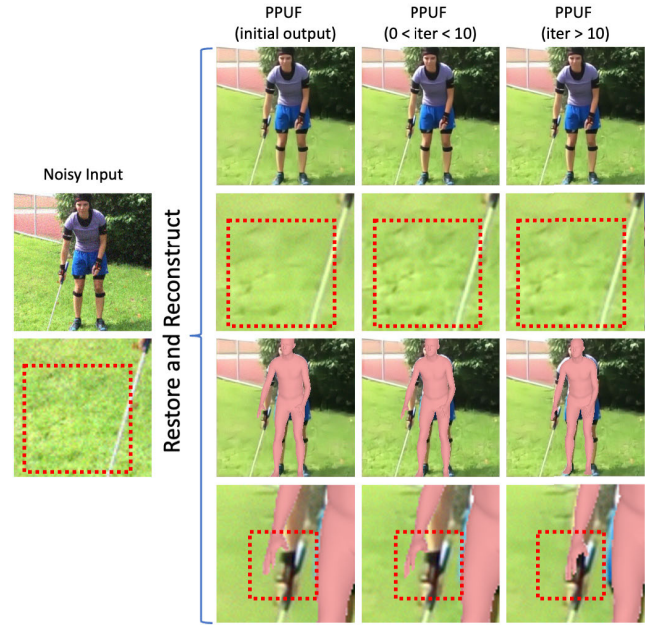

FIGURE 6. Qualitative results of PPUF in noisy case. *Electronic screen* is advised for proper viewing.
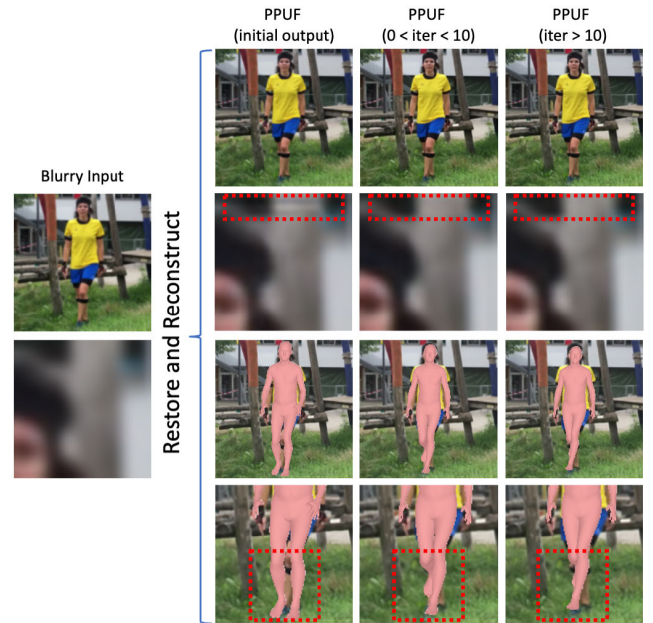

FIGURE 7. Qualitative results of PPUF in blurry case. *Electronic screen* is advised for proper viewing.

(*Denoise-t*). From the analysis above, it is clear that our PPUF helps the denoising and 3D body reconstruction module to self-improve jointly in test time with the capability of fast adaptation.

The visual performance of the joint denoising and 3D body reconstruction is shown in Figure 6. From these examples, detailed structures that were not present in the initial output are recovered after several iterations. The reconstructed bodies are also improved when the iteration of the PPUF is increased. The improvements are emphasized in the cropped regions. From our perception, the reconstructed body tends
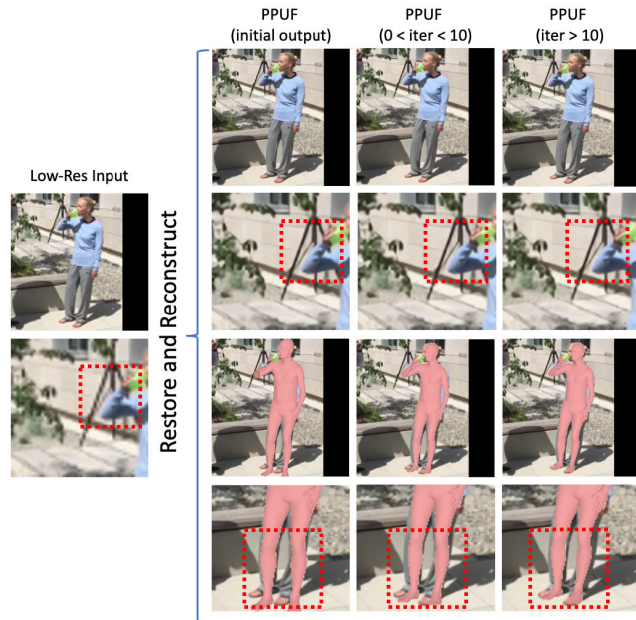
FIGURE 8. Qualitative results of PPUF in low-resolution case. *Electronic screen* is advised for proper viewing.

to have a correct initial position while the minor incorrect position is self-improved within several iterations. Additional qualitative results in the denoising and 3D body reconstruction case are provided in the supplementary material.

3) JOINT DEBLURRING AND 3D BODY RECONSTRUCTION

The deblurring restoration within our PPUF framework also demonstrates its self-improvement capability. In this case, the deblurring also utilized the pseudo-clean and pseudo-corrupted data that are extracted during test-time. From Table 1, The PSNR scores are significantly improved by +**1.529 dB** (*Deblur-v*) and +**1.791 dB** (*Deblur-t*). Similar to the denoising case, large gains are already scored within 3 iterations as they reach +**1.445 dB** (*Deblur-v*) and +**1.731 dB** (*Deblur-t*) improvements. These results are also improved in the SSIM level by the maximum gain of +**0.02162** (*Deblur-v*) and +**0.03463** (*Deblur-t*) as written in Table 2.

In the 3D body reconstruction case, consistent performances are also demonstrated in Tables 3 and 4. The predicted body joints are re-improved by the reduction of the error scores by −**15.646 mm** (*Deblur-v*) and −**11.953 mm** (*Deblur-t*) as shown in Table 3. Similar to the denoising case, significant cuts are shown within 3 iterations only, indicating that the task of deblurring and 3D body reconstruction succeed in performing fast adaptation jointly.

Qualitative results are shown in Figure 7 where weak artifacts are restored after several iterations. The case of rendered 3D body reconstruction results shows a clear difference. Partial body regions are clearly improved after iterations occur. Figure 7 shows that the foot positioning are even re-corrected according to the depth position. To our visual perception, most updates are developed in the scale as well as the body-joint rotations levels. Additional qualitative results

in the deblurring and 3D body reconstruction case are also provided in the supplementary material.

4) JOINT SUPER-RESOLUTION AND 3D BODY RECONSTRUCTION

In terms of super-resolution, our restoration results still maintain high scores output. Unlike the denoising and deblurring cases, the super-resolution is improved slightly by +**0.421 dB** (*SR-v*) and +**0.029 dB** (*SR-t*) as shown in Table 1. Nonetheless, the achieved gains present in each iteration case denote that the PPUF still maintain the SR module's self-improvement capability with no tendency to get significantly lower scores when iterations are increased. This phenomenon indicates that the SR module in PPUF avoids the *catastrophic forgetting* [55] effect, which is a performance reduction anomaly due to the inability to preserve initial knowledge.

Interestingly, the 3D body reconstruction results in the SR case achieve the best performance compared to other degradation cases. This phenomenon indicates that the SR case is relatively easier to be solved by the reconstruction module. As shown in Tables 3 and 4, the MPJPE and RE scores of (*SR-t*) obtain the values of **121.425 mm** and **69.889 mm**, respectively. These scores are improved through the margin of −**14.869 mm** and −**7.819 mm** compared to each respective initial output. These results are directly reflected in the qualitative outputs in Figure 8 where restoration performance is still preserved with high quality, while the body reconstruction is significantly improved over iterations.

With the vast experiments above, our work proves that any restoration and 3D body reconstruction arts can be coupled to achieve test-time adaptation. The self-improvement capability can be achieved thanks to the assistance of the pseudo-training data introduced in the test stage. Moreover, we demonstrate the nature of fast-adaptation capability using the proposed PPUF as most of the modules obtain vast gains within few iterations. Using only the prior works within our modules, we observe that adaptive learning is feasible as long as they are adjusted with a correct algorithm and reliable information, such as pseudo-clean and pseudo-corrupted data. More qualitative results in the SR and 3D body reconstruction case are also provided in the supplementary material. We suggest the readers check all the visual results on an electronic screen. The supplementary material is provided in video format to accommodate the reader to easily view the test-time improvement of the restoration and 3D body reconstructions results.

While we demonstrate the successes above, our method, like other self-adaptive works, requires extra time to perform test-stage iterations (Line 6-10 in Algorithm 2). However, our algorithm only spends around ∼**3.5 seconds** to run *both* restoration (SR case using EDSR [35]) and reconstruction (using HMR [6]) for **20 iterations** using the input of $224 \times 224 \times 3$ image in a Titan RTX machine. The case of denoising or deblurring (using URIE [47]) joint with the reconstruction task takes about ∼**2.8 seconds**. Thus, future

work that solves automatic-iteration-cutting can benefit test-time self-improvement schemes in joint-task scenario.

V. CONCLUSION

We presented a universal framework in the algorithm forms capable of utilizing various prior state-of-the-art restoration modules, particularly in solving denoising, deblurring, and SR, while jointly tasked with 3D body reconstruction module. Our work is motivated by the need for life-long learning that focuses on adaptive capability while avoiding the classic approach that requires a vast amount of datasets and re-crafting specific novel networks that suit the datasets. With the support of the pseudo-training data during test-time, our framework can perform temporary self-supervised training that yields to the performance gain. By this approach, both restoration and reconstruction modules are endowed with self-adaptive capability. Our experimental results show that coupling these tasks with a correct framework and reliable pseudo-data obtain significant improvement even within a few iterations without any performance drop anomaly. We believe our exploration can be further extended to the more challenging restoration tasks such as dehazing and de-raining as their synthesized data are feasible.

REFERENCES

[1] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4501–4510.

[2] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 752–768.

[3] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "Bodynet: Volumetric inference of 3D human body shapes," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2018, pp. 20–36.

[4] S. Saito, Z. Huang, R. Natsume, S. Morishima, H. Li, and A. Kanazawa, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2304–2314.

[5] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 561–578.

[6] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7122–7131.

[7] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *Proc. Int. Conf. 3D Vis.*, Sep. 2018, pp. 484–494.

[8] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2252–2261.

[9] J. Rim, H. Lee, J. Won, and S. Cho, "Real-world blur dataset for learning and benchmarking deblurring algorithms," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 184–201.

[10] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[11] S. Guan, J. Xu, Y. Wang, B. Ni, and X. Yang, "Bilevel online adaptation for out-of-domain human mesh reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 10472–10481.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[13] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, vol. 6311. Springer, 2010, pp. 157–170.

[14] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 233–240.

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

[16] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 257–265.

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

[18] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8877–8886.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[20] Z. Shen, W.-S. Lai, T. Xu, J. Kautz, and M.-H. Yang, "Deep semantic face deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8260–8269.

[21] W. Ren, J. Yang, S. Deng, D. Wipf, X. Cao, and X. Tong, "Face video deblurring using 3D facial priors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9387–9396.

[22] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-aware motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5571–5580.

[23] J. S. Lumentut, J. Santoso, and I. K. Park, "Human motion deblurring using localized body prior," in *Proc. Asian Conf. Comput. Vis.*, Nov. 2020, pp. 1–16.

[24] J. S. Lumentut and I. K. Park, "Human and scene motion deblurring using pseudo-blur synthesizer," *IEEE Access*, vol. 9, pp. 146366–146377, 2021.

[25] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 341–349.

[26] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[27] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3929–3938.

[28] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[29] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2480–2495, Jul. 2020.

[30] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3791–3799.

[31] J. Salvador and E. Perez-Pellitero, "Naive Bayes super-resolution forest," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 325–333.

[32] J. J. Huang and W. C. Siu, "Learning hierarchical decision trees for single-image super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 937–950, May 2015.

[33] J. Kim, J. K. Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1646–1654.

[34] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4681–4690.

[35] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 136–144.

[36] Y. Li, Y. Liu, Q. Yan, and K. Zhang, "Deep dehazing network with latent ensembling architecture and adversarial learning," *IEEE Trans. Image Process.*, vol. 30, pp. 1354–1368, 2021.

[37] K. Zhang and Y. Li, "Single image dehazing via semi-supervised domain translation and architecture search," *IEEE Signal Process. Lett.*, vol. 28, pp. 2127–2131, 2021.

[38] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 8346–8355.

[39] C. Chen and H. Li, "Robust representation learning with feedback for single image deraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 7742–7751.

[40] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015, doi: 10.1145/2816795.2818013.

[41] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3D human dynamics from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5614–5623.

[42] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5253–5263.

[43] Z. Luo, S. A. Golestaneh, and K. M. Kitani, "3D human motion estimation via motion compression and refinement," in *Proc. Asian Conf. Comput. Vis.*, Nov. 2020, pp. 1–17.

[44] H. Choi, G. Moon, J. Y. Chang, and K. M. Lee, "Beyond static features for temporally consistent 3D human pose and shape from a video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 1964–1973.

[45] R. Gao and K. Grauman, "On-demand learning for deep image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1086–1095.

[46] K. Yu, C. Dong, L. Lin, and C. C. Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2443–2452.

[47] T. Son, J. Kang, N. Kim, S. Cho, and S. Kwak, "Urie: Universal image enhancement for visual recognition in the wild," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 749–765.

[48] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3516–3525.

[49] S. Park, J. Yoo, D. Cho, J. Kim, and T. H. Kim, "Fast adaptation to super-resolution networks via meta-learning," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 754–769.

[50] S. Lee, M. Choi, and K. M. Lee, "DynaVSR: Dynamic adaptive blind video super-resolution," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2093–2102.

[51] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9137–9146.

[52] M. Choi, J. Choi, S. Baik, T. H. Kim, and K. M. Lee, "Scene-adaptive video frame interpolation via meta-learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9444–9453.

[53] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, pp. 4–27, Mar. 2010.

[54] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3D human pose in the wild using imus and a moving camera," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2018, pp. 601–617.

[55] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9229–9248.

JONATHAN SAMUEL LUMENTUT (Member, IEEE) received the B.Comp.Sc. and M.S. degrees in computer science from Bina Nusantara University, Indonesia, in 2013 and 2014, respectively, and the Ph.D. degree in electrical and computer engineering from Inha University, South Korea, in 2021. He was a Visiting Student with the Center of Visual Computing, University of California San Diego (UCSD). His research interests include computer vision, computational photography, human body reconstruction, and image processing.

MATTHEW MARCHELLUS received the B.Comp.Sc. degree in computer science from Bina Nusantara University, Indonesia, in 2020. His research interests include computer vision, human video prediction, and human motion prediction.

JOSHUA SANTOSO received the B.Comp.Sc. degree in computer science from Bina Nusantara University, Indonesia, in 2019, and the M.S. degree from Inha University, South Korea, in 2021. His research interests include computer vision, computational photography, human body reconstruction, and image processing.

TAE HYUN KIM (Member, IEEE) received the B.S. and M.S. degrees from the Department of Electrical Engineering, KAIST, Daejeon, South Korea, in 2008 and 2010, respectively, and the Ph.D. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2016. He worked as a Postdoctoral Researcher with the Max Planck Institute for Intelligent Systems. He is currently an Assistant Professor with the Department of Computer Science, Hanyang University, Seoul. His research interests include low level computer vision and computational imaging.

JU YONG CHANG received the B.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2001 and 2008, respectively. From 2008 to 2009, he was a Postdoctoral Researcher with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. He is currently an Associate Professor with Kwangwoon University, Seoul. His research interests include computer vision and machine learning, with a focus on human body pose estimation and human activity recognition.
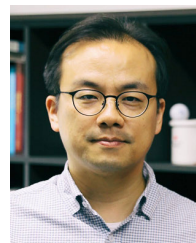
IN KYU PARK (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Seoul National University, in 1995, 1997, and 2001, respectively, all in electrical engineering and computer science. From September 2001 to March 2004, he was a member of Technical Staff with the Samsung Advanced Institute of Technology. Since March 2004, he has been with the School of Information and Communication Engineering, Inha University, where he is currently a Full Professor. From January 2007 to February 2008, he was an Exchange Scholar with Mitsubishi Electric Research Laboratories. From September 2014 to August 2015, he was a Visiting Associate Professor with the MIT Media Lab. From July 2018 to June 2019, he was a Visiting Scholar with the Center for Visual Computing, University of California at San Diego. His research interests include the joint area of computer vision and graphics, including 3D shape reconstruction from multiple views, image-based rendering, computational photography, deep learning, and GPGPU for image processing and computer vision. He is a member of ACM.

• • •