IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 6, 2021, accepted May 19, 2021, date of publication May 31, 2021, date of current version June 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3084968

Robust Single Image Deblurring Using Gyroscope Sensor

SEOWON JI, JUN-PYO HONG, JEONGMIN LEE, SEUNG-JIN BAEK, AND SUNG-JEA KO, (Fellow, IEEE)
School of Electrical Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Seung-Jin Baek (sjinbaek@korea.ac.kr)

ABSTRACT Motion blur in an image is caused by the movement of the camera during exposure time; thus, awareness of the camera motion is a key factor in image deblurring algorithms. Among the various sensors that can be utilized while taking a picture in handheld devices, a gyroscope sensor, which measures the angular velocity, can help in estimating the camera motion. To achieve accurate and efficient single-image deblurring with a gyroscope sensor, we present a novel deep network with a flexible receptive field that is appropriate for training features related to the nature of the blur. Two specialized modules are sequentially placed in the proposed network to adaptively convert the shapes of the convolutional kernels. The first module directly transforms the kernel shape into the direction of the camera motion indicated by the gyroscope measurements. In the middle of the network, where the feature abstraction is sufficiently proceeded, the second module integrates features from the blurry image along with the information from the gyroscope to convert the kernel shape effectively, even when the gyroscope sensor is unreliable. Using a new gyro-image paired dataset, extensive experiments were conducted to show the effects of the reliability of the gyroscope measurements on the deblurring performance and to prove the effectiveness of our strategy.

INDEX TERMS Convolutional neural network, gyroscope sensor, homography, single-image deblurring.

I. INTRODUCTION

Despite the rapid development of handheld cameras, motion blur in an image is still visible when the device moves during the image exposure time. Motion blur not only affects the visual quality of the image but also degrades the performance of various applications such as object detection, image segmentation, and visual odometry. Therefore, further improvements in single-image deblurring, in which a latent sharp image is recovered from a blurry one, are now being actively researched.

From a classical perspective, the research on single-image deblurring can be divided into two sub-categories: non-blind deblurring and blind deblurring. The non-blind methods [1], [2] employ a given blur kernel followed by deconvolution of the blurry image. In contrast, blind deblurring methods use only the blurry image to recover the sharp image. However, in many cases, blind methods [3]–[7] attempt to perform deblurring by first estimating the precise blur kernel from a blurry image and then adopting the estimated kernel in a non-blind deblurring approach to recover the sharp image. Therefore, the deblurring performance of both methods depends heavily on the quality of the blur kernel.

Owing to the rapid development of deep learning, various blind deblurring methods that adopt convolutional neural networks (CNNs) have been proposed. Early CNN-based methods were trained with blurry images synthesized using a uniform blur kernel [8]–[10]. In [11], the network was trained to classify the blur kernel used in traditional deblurring. Later, Nah *et al.* presented a new realistic blurry image dataset that includes the ground truth [12]. This was followed by the development of various methods [13]–[19] in which sophisticated relationships between realistic blurry images and their corresponding sharp images are learnt through CNN.

In addition to these works that utilize only the image, approaches for using external information to provide better blur estimates have also been proposed. In these previous studies [20]–[22], inertial sensors such as accelerometers and gyroscopes have proven to be helpful in improving the blur estimation. In particular, gyroscopes are substantially useful in blur estimation because the angular velocities measured by a gyroscope indicate the rotational motion of the camera,

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

which affects the formation of blur artifacts. Moreover, the gyroscope data can be easily obtained because most mobile devices are equipped with inertial measurement units (IMUs), which gather a collection of sensory information, including gyroscope measurements.

Mustaniemi *et al.* were the first researchers to apply gyroscope data to CNN-based non-blind deblurring in a network referred to as DeepGyro [23]. They designed a wide and deep network for image deblurring and exploited the gyroscope information along with the input blurry image. Based on the data from a gyroscope sensor, a 2-channel blur field indicating the camera motion during exposure time is first computed. This blur field is then concatenated to a blurry image as a guide and used as an input to the network, similar to general CNN-based parameterized image operators [24]. However, as pointed out in [25], this approach results in limited performance improvement owing to the network components, such as the receptive fields and weights, being fixed regardless of the variation in the guidance parameters. In real-case scenarios, the gyroscope measurements may become unreliable because of two dominant causes, namely, 1) the inherent noise in the sensor and 2) the miscalibration in which the gyroscope and image sensors are not accurately synchronized, thereby giving rise to erroneous deblurring results. Furthermore, the wide and deep network design results in a large receptive field, which leads to an increased number of network parameters and high computational complexity.

To improve the sensor robustness and image deblurring network efficiency performance, an effective gyroscope-guided network (EggNet) with a flexible receptive field is presented in this paper. We adaptively transform the receptive field of the network so that the network is trained with image features related to the direction and magnitude of the blur. In the front of the network, the kernel shape of the convolutional layer is converted in the direction indicated by the gyroscope measurements to reflect the device motion. Because the gyroscope measurements are occasionally degraded by noise or miscalibration, the feature information extracted from the gyroscope and the blurry image is integrated in the middle of the network to correctly convert the kernel shape by exploiting the appropriate features from both the image and the gyroscope information. Hence, the proposed network performs deblurring robustly, even if unreliable gyroscope information is provided.

We established a new massive gyroscope-image paired database to train the proposed network. To validate the effectiveness of our proposed network, two types of experiments were conducted: the controlled case where the noise and miscalibration were perfectly controlled, and the raw case where both problems existed. In addition, in view of industrial considerations, we comprehensively analyzed the deblurring performance in terms of the noise level and the calibration accuracy of the gyroscope measurements.

The remainder of this paper is organized as follows. Related works are presented in Section II. In Section III, we describe the proposed EggNet in detail. The experimental setup and results are presented in Section IV to demonstrate the superiority of the proposed method. In Section V, we present our conclusions.

II. RELATED WORKS

A. HOMOGRAPHY FROM GYROSCOPE MEASUREMENTS

To utilize the rotational angular velocity obtained from the gyroscope sensor for image deblurring, the measurements must first be transformed into an appropriate form. Based on the multi-view geometry [26] and camera motion model [27], we transform the gyroscope measurements into a rotational matrix that indicates the angular motion of the camera. The rotational matrix for a specific time duration Δt can be calculated as follows:

$$R(\boldsymbol{\theta}^{\Delta t}) = R_x(\theta_x^{\Delta t})R_y(\theta_y^{\Delta t})R_z(\theta_z^{\Delta t}), \tag{1}$$

$$R_x(\theta_x^{\Delta t}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(-\theta_x^{\Delta t}\right) & -\sin\left(-\theta_x^{\Delta t}\right) \\ 0 & \sin\left(-\theta_x^{\Delta t}\right) & \cos\left(-\theta_x^{\Delta t}\right) \end{bmatrix}, \tag{2}$$

$$R_y(\theta_y^{\Delta t}) = \begin{bmatrix} \cos\left(-\theta_y^{\Delta t}\right) & 0 & \sin\left(-\theta_y^{\Delta t}\right) \\ 0 & 1 & 0 \\ -\sin\left(-\theta_y^{\Delta t}\right) & 0 & \cos\left(-\theta_y^{\Delta t}\right) \end{bmatrix}, \tag{3}$$

$$R_z(\theta_z^{\Delta t}) = \begin{bmatrix} \cos\theta_z^{\Delta t} & \sin\theta_z^{\Delta t} & 0 \\ -\sin\theta_z^{\Delta t} & \cos\theta_z^{\Delta t} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\theta}^{\Delta t} = \left\{\theta_x^{\Delta t}, \theta_y^{\Delta t}, \theta_z^{\Delta t}\right\}$ represents the changes in rotational angles during Δt and $\left\{R_x, R_y, R_z\right\}$ represent the rotational matrices for each axis. The rotational matrix applies only to the angular motion of the camera coordinates. Therefore, the intrinsic camera matrix K should be considered to convert the motion into an image coordinate system. A homography matrix in Δt can be derived as

$$H(\boldsymbol{\theta}^{\Delta t}) = K(R(\boldsymbol{\theta}^{\Delta t}) + \mathcal{D})K^{-1}. \tag{5}$$

When the scene is far away or the motion is only caused by rotation, the term $\mathcal{D}$ can be omitted from (5). Then, the equation can be rewritten as

$$H(\boldsymbol{\theta}^{\Delta t}) = KR(\boldsymbol{\theta}^{\Delta t})K^{-1}. \tag{6}$$

Using the homography transformation matrix, the camera motion during Δt can be expressed as a 2D vector by calculating the coordinate differences between an initial point (x^{t_i}, y^{t_i}) and an end point $(x^{t_i+\Delta t}, y^{t_i+\Delta t})$. The end point is computed as

$$w\begin{bmatrix} x^{t_i+\Delta t} \\ y^{t_i+\Delta t} \\ 1 \end{bmatrix} = H(\boldsymbol{\theta}^{\Delta t})\begin{bmatrix} x^{t_i} \\ y^{t_i} \\ 1 \end{bmatrix}, \tag{7}$$

where w is a scale factor introduced to convert the resulting matrix into homogeneous coordinates.

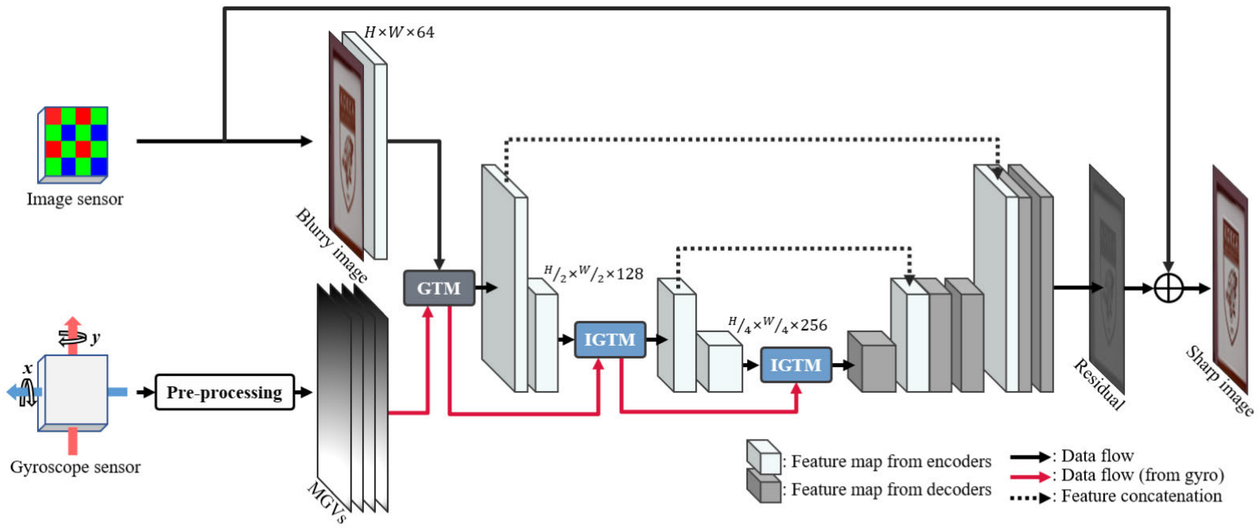

FIGURE 1. **Overall framework of the proposed image deblurring network.**

B. TRANSFORMATION OF RECEPTIVE FIELD

To enhance the geometric transformation capability of CNN models, we introduce a deformable convolution [28] that transforms the receptive field. The deformable convolution adds 2D offsets learned from the feature map to the regular grid location of the basic convolution operation, thereby enabling the receptive field to be transformed flexibly. For conciseness of explanation, let us assume that the number of channels in the input and output feature maps is equal to one. Then, the output of the deformable convolution at pixel $\mathbf{p}_0$ on the input feature map $X \in \mathbb{R}^{h\times w}$ is defined as

$$Y(\mathbf{p}_0) = \sum_{k=1}^{K} W(\mathbf{p}_k)X(\mathbf{p}_0 + \mathbf{p}_k + \Delta\mathbf{p}_k), \tag{8}$$

where Y is the output feature map, W is the learnable weight kernel of size K, and $\mathbf{p}_k \in (-(\sqrt{K} - 1)/2, -(\sqrt{K} - 1)/2), \ldots, ((\sqrt{K} - 1)/2, (\sqrt{K} - 1)/2)$ indicates the location of the convolution kernel with a dilation of 1. When $\Delta\mathbf{p}_k$ has a fractional value, bilinear interpolation is applied to the input feature map to sample the value at that position. For $X \in \mathbb{R}^{c\times h\times w}$, the 2D offsets are applied equally to all the channels. The flexible 2D offsets at $\mathbf{p}_0$ are learned from the input feature map as follows:

$$\Delta\mathbf{p}_k = (\Delta\mathrm{p}_k^x, \Delta\mathrm{p}_k^y), \tag{9}$$

$$\Delta\mathrm{p}_k^x = \sum_{l=1}^{L} W_O^{x,k}(\mathbf{p}_l)X(\mathbf{p}_0 + \mathbf{p}_l), \tag{10}$$

$$\Delta\mathrm{p}_k^y = \sum_{l=1}^{L} W_O^{y,k}(\mathbf{p}_l)X(\mathbf{p}_0 + \mathbf{p}_l), \tag{11}$$

where L is the spatial size of the learnable weight kernel for the offset $W_O = \{W_O^{x,1}, W_O^{y,1}, \ldots, W_O^{x,K}, W_O^{y,K}\}$. In short, the deformable convolution f_D can be expressed as

$$Y = f_D(X; W, W_O). \tag{12}$$

III. PROPOSED METHOD

We present EggNet, which robustly transforms the receptive field by exploiting information from both the input image and the gyroscope for efficient image deblurring. As shown in Fig. 1, the proposed method employs multi-modal input data, i.e., image and gyroscope data, for the U-Net-based [29] architecture, which incorporates two specialized modules, namely, the gyro-aided transformation module (GTM) and the image-gyro-aided transformation module (IGTM). In the pre-processing step, the gyroscope measurements are converted into motion guidance vectors (MGVs) based on the camera motion. These MGVs are then exploited throughout the network. Each step is described in detail in the following subsections.

A. IMAGE-GYRO PAIRED DATASET

To train the proposed network, following data are required:

1) The input blurry images and their corresponding target sharp images
2) Gyroscope measurements recorded during the exposure time of the input blurry images

We used the camera module in a smartphone device to obtain the sharp image sequences and the gyroscope measurements, as shown in Fig. 2. The images were captured in Bayer format, and the 2D gyroscope sensor for lens-shift optical image stabilization in the camera module was used to obtain the two rotational velocities (θ_x, θ_y). To generate the input and target images, we followed the methodology in [12], [30]. A blurry image is generated by accumulating the sharp images captured over the collective exposure time, which is the time

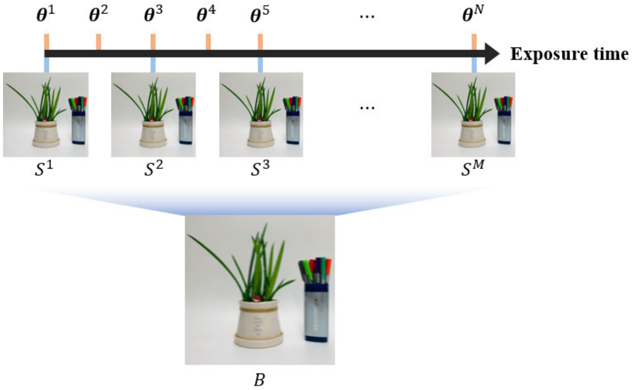

FIGURE 2. Image and gyroscope data acquisition process. (*M* and *N* are the numbers of sampled images and gyroscope measurements during the exposure time, respectively. For conciseness of explanation, we assume that *M* and *N* are odd numbers.)

duration of the image sensor receiving light, as follows:

$$B \simeq \frac{1}{M}\sum_{i=1}^{M} S^i, \tag{13}$$

where M is the number of sampled images, which determines the exposure time of the blurry image, and S^i is the i^{th} sharp image. Since the desired original texture can be assumed that it is located in the center of the blur artifact [31], the corresponding target image for the blurry image is set to the central image $S^{(M+1)/2}$ of the sampled sharp images. In addition, the gyroscope measurements $\{\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^N\}$, which are recorded simultaneously over identical exposure times, are saved in pairs together with the blurry and target sharp images. The target sharp image and the input blurry image are hence generated, as shown in Figs. 3(a) and (b).

B. MOTION GUIDANCE VECTOR (MGV)

Among the two types of data described above, the sharp and blurry images are used without pre-processing for training the proposed network. In contrast, the angular velocities measured by the gyroscope are converted into MGVs for utilization in the network.

The MGVs for a single blurry image are divided into two vectors: the pro-MGV indicating the camera motion from the center to the end of the exposure time, and the pre-MGV indicating the movement from the center to the start of the exposure time. The pro-MGV is computed as

$$\mathrm{MGV}_{\mathrm{Pro}}(x, y)_{x,y\in w,h} = \begin{bmatrix} x^{t_E} - x^{t_C} \\ y^{t_E} - y^{t_C} \end{bmatrix}, \tag{14}$$

where h and w represent the height and width of the blurry input image, respectively. (x^{t_E}, y^{t_E}) and (x^{t_C}, y^{t_C}) denote the 2D coordinates at the end and the center of the exposure time, respectively. The 2D coordinates (x^{t_E}, y^{t_E}) can be computed from (x^{t_C}, y^{t_C}) using the gyroscope measurements recorded during the corresponding exposure time as

$$w\begin{bmatrix} x^{t_E} \\ y^{t_E} \\ 1 \end{bmatrix} = H_{\mathrm{Pro}}\begin{bmatrix} x^{t_C} \\ y^{t_C} \\ 1 \end{bmatrix} \tag{15}$$

where H_{Pro} is calculated using (1) and (5),

$$H_{\mathrm{Pro}} = K \prod_{n=(N+1)/2}^{N-1} R(\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^{n})K^{-1}. \tag{16}$$

The pre-MGV can be obtained in the same manner. As shown in Fig. 3 (c), the MGVs indicate the motion during the exposure time. The set of MGVs is utilized in the GTM and the IGTM.

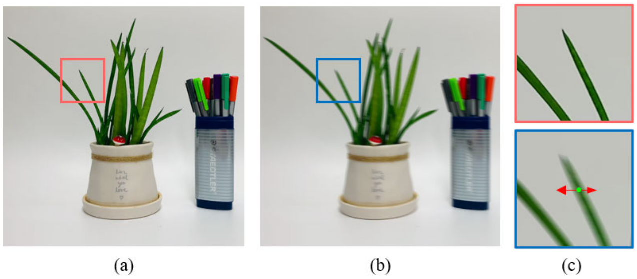

FIGURE 3. Paired data for training the proposed network. (a) Target sharp image; (b) input blurry image; (c) magnified patches of (a) and (b). The MGVs are shown in the magnified part of (b).

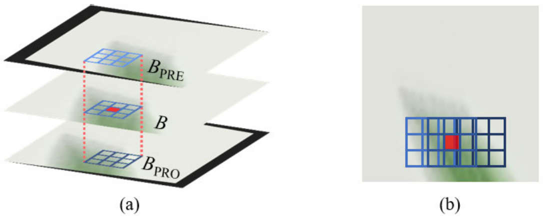

FIGURE 4. Conceptual visualization of the red pixel receptive field in blurry image *B*. (a) Visualized receptive fields on the stack of warped images. B_{Pre} and B_{Pro} are the warped images obtained using H_{Pre} and H_{Pro}, respectively, and (b) visualization of the receptive fields projected on *B*.

C. TRANSFORMATION OF RECEPTIVE FIELD

As proved in [32], not all the pixels in the receptive field contribute equally to the response of an output unit. Thus, if the receptive field can be adaptively transformed to include the pixels that contribute significantly to the resultant image, an efficient CNN model can be designed.

In our previous work [33], we presented an approach that exploits a channel-wise stacking of the warped images in which blur artifacts caused by camera motion are aligned in the input data for the CNN model. As shown in Fig. 4, when the receptive fields in the stacked images are projected onto the original blurry image, the union of the projected receptive fields is equivalent to a receptive field transformed in the blur direction. Therefore, the network to be trained directly uses the pixels related to the direction and magnitude

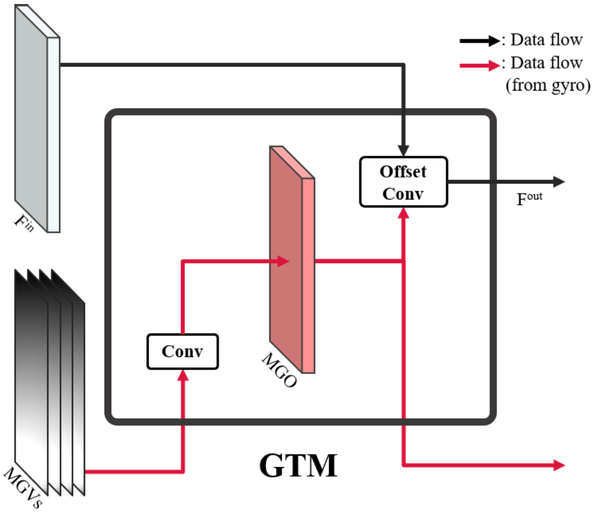

FIGURE 5. Detailed structure of the gyro-aided transformation module.

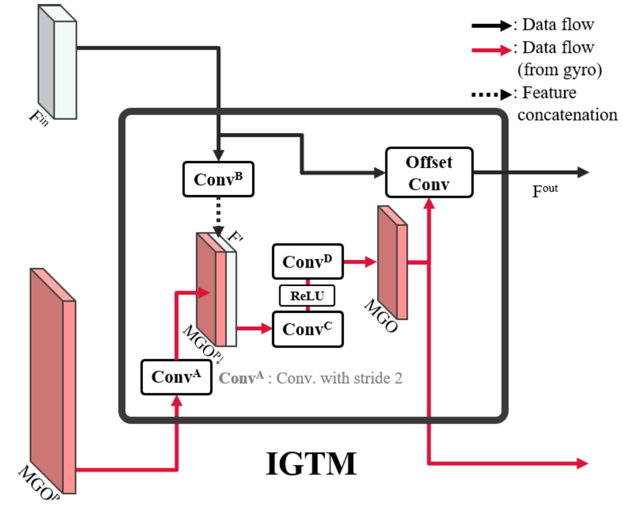

FIGURE 6. Detailed structure of image and gyro-aided transformation module.

of the blur. This results in a promising performance improvement compared to using a single blurry image as the input data without increasing the number of network parameters. Therefore, it can be argued that such a transformation of the receptive field enables an effective network design that uses the gyroscope data for guidance.

Several methods that flexibly transform the rigid receptive fields of regular CNN models [28], [34]–[36] have been proposed. Among these methods, deformable convolution [28] can effectively transform the receptive field with low computational complexity. Based on [28], we present two transformation modules that adaptively convert the shape of the receptive field using the blur information obtained from the gyroscope and the blurry image.

1) GTM

The first module, the GTM, is placed in the front of the network, as shown in Fig. 1. It transforms the receptive field in the blur direction using the gyroscope measurements. As shown in Fig. 5, the GTM is composed of a regular convolution followed by an offset convolution. The GTM can be expressed as

$$\{F^{\text{out}}, \text{MGO}\} = f_{\text{GTM}}(F^{\text{in}}, \text{MGVs}; W_R, W_{\text{OC}}), \qquad (17)$$

where f_{GTM} represents the GTM parameterized by W_R and W_{OC}, which are respectively the learnable weight kernels for the regular and offset convolution for performing convolution with the converted kernel. F^{out} and F^{in} are the output and input feature maps, respectively. Using the regular convolution layer, the motion guidance offset (MGO), which indicates the 2D offsets of the weight kernel in the offset convolution, is first learned from the MGVs. Using this MGO, the offset convolution then transforms the W_{OC} to be fitted in the blur direction, and the transformed kernel is applied to F^{in} to compute F^{out} using (8). Finally, the resultant F^{out} is fed into the main network, and the MGO is transferred to the next transformation module, i.e., the IGTM.

2) IGTM

The second module, the IGTM, is used twice in the network, as shown in Fig. 1. To robustly transform the receptive field in the blur direction regardless of the reliability of the gyroscope measurements, the IGTM exploits both the gyroscope and the image information. As shown in Fig. 6, the IGTM is composed of a set of regular convolutions (Conv^{A}, Conv^{B}, Conv^{C}, and Conv^{D}), and one offset convolution. The IGTM can be expressed as

$$\{F^{\text{out}}, \text{MGO}\} = f_{\text{IGTM}}(F^{\text{in}}, \text{MGO}^{\text{P}}; W_{Rs}, W_{\text{OC}}), \qquad (18)$$

where f_{IGTM} represents the IGTM parameterized by W_{Rs} and W_{OC}, which are learnable weight kernels for the set of regular convolutions and the offset convolution, respectively, and MGO^{P} is the MGO computed in the previous GTM (or IGTM). To fuse the information from the gyroscope and the blurry image, i.e., MGO^{P}, and F^{in}, respectively, we apply a convolutional layer Conv^{A} with a stride of 2 to MGO^{P}, which results in $\text{MGO}^{\text{P}\downarrow}$. Then, Conv^{B} is applied to F^{in} to match the number of channels to that of $\text{MGO}^{\text{P}\downarrow}$, resulting in F'. The MGO is computed by applying the serial operation Conv^{C}-ReLu-Conv^{D} to the concatenation of the two features $\text{MGO}^{\text{P}\downarrow}$ and F'. Finally, similar to the GTM, the offset convolution uses the MGO to transform W_{OC} in the direction of blur and applied to F^{in} to calculate F^{out}. The F^{out} again flows into the main network, and the MGO is transferred to the next IGTM until it reaches the last IGTM.

IV. EXPERIMENTAL RESULTS

A. IMPLEMENTATION DETAILS AND DATASET

1) IMPLEMENTATION DETAILS

The detailed structure of the proposed EggNet is presented in Table 1. The input Bayer image is reshaped into four channel (Gr, R, Gb, and B) images and fed into the proposed network. If the input image is normal RGB or gray, the input

TABLE 1. Detailed architecture of EggNet.

Group	Layer Type	Act	Weight Dimension	Stride	Remarks
Encoder LV1	Conv	ReLU	$3 \times 3 \times 4 \times 64$	1	
	GTM	-	-	-	
	Res	-	-	-	$c = 64$
Encoder LV2	Conv	ReLU	$3 \times 3 \times 64 \times 128$	2	
	IGTM	-	-	-	$c = 128$
	Res	-	-	1	$c = 128$
Encoder LV3	Conv	ReLU	$3 \times 3 \times 128 \times 256$	2	
	IGTM	-	-	-	$c = 256$
	Res	-	-	1	$c = 256$
Decoder LV3	Deform Conv	-	$3 \times 3 \times 256 \times 256$	1	$c = 256$
	Res	-	-	1	$c = 256$
	Upsample	-	-	-	
	Conv	ReLU	$3 \times 3 \times 256 \times 128$	1	
Decoder LV2	Conv	ReLU	$1 \times 1 \times 256 \times 128$	1	
	Deform Conv	-	-	1	$c = 128$
	Res	-	-	1	$c = 128$
	Upsample	-	-	-	
	Conv	ReLU	$3 \times 3 \times 128 \times 64$	1	
Decoder LV1	Conv	ReLU	$1 \times 1 \times 128 \times 64$	1	
	Deform Conv	-	-	1	$c = 64$
	Res	-	-	1	$c = 64$
	Conv	-	$3 \times 3 \times 64 \times 4$	1	

Module	Layer Type	Act	Weight Dimension	Stride	Remarks
Res	Conv	ReLU	$3 \times 3 \times c \times c$	1	
	Conv	-	$3 \times 3 \times c \times c$	1	
Deform Conv	Conv	-	$3 \times 3 \times c \times 18$	1	
	Custom Conv	-	$3 \times 3 \times c \times c$	1	
GTM	Conv	-	$3 \times 3 \times 6 \times 18$	1	
	Offset Conv	-	$3 \times 3 \times 64 \times 64$	1	
IGTM	Conv^A	-	$3 \times 3 \times 18 \times 18$	2	
	Conv^B	-	$3 \times 3 \times c \times 18$	1	
	Conv^C	ReLU	$3 \times 3 \times 36 \times 36$	1	
	Conv^D	-	$3 \times 3 \times 36 \times 18$	1	
	Offset Conv	-	$3 \times 3 \times c \times c$	1	

image can be directly fed into the network without reshaping. The network reconstructs the latent sharp image $\hat{S}$ by producing a residual image R and adding R to the input image B. The L1 criterion is applied as the loss for training the network. It can be formulated as

$$Loss = \left\| \hat{S} - S \right\|_1 , \tag{19}$$

where S is the target sharp image.

We used the Adam [37] optimizer with a mini-batch size of two for training. After 1×10^5 iterations, the learning rate was decreased to 1/2 of the initial learning rate. The entire training required 2×10^5 iterations to converge. We used the PyTorch [38] library. All the experiments were conducted on a PC with an Intel i3-8100 CPU and an NVIDIA Titan Xp GPU. The proposed EggNet took approximately 30 hours and 120 hours to train on the controlled case and the raw case, respectively.

2) DATASET

Because there is no available image-gyro paired dataset, we made variations in our dataset and conducted extensive experiments. The deblurring performance of EggNet was evaluated for two cases: the controlled case and the raw case. Both cases consisted of 2,119 sets (1,923 sets for training and 196 sets for testing) of gyroscope measurements, blurry images, and sharp images. The resolutions of the images in the controlled case and the raw case were 512×512 and 1024×1024, respectively. To simulate the gyroscope sensor noise and miscalibration in the controlled case, we synthesized the blurry image by integrating the homographies over the exposure time, as explained in [20]. Because the gyroscope records discrete measurements, the blurry image can be expressed as

$$B \simeq \frac{1}{M} \sum_{n=1}^{N} H(\boldsymbol{\theta}^i - \boldsymbol{\theta}^{\frac{N+1}{2}})S. \tag{20}$$

In addition, to validate the effectiveness of the proposed method in the presence of noise and miscalibration, two variations of the controlled case dataset were generated: 1) random noise vectors were added to the MGVs to imitate noise, and 2) the MGVs were generated with delayed gyroscope measurements to mimic miscalibration.

In the raw case, we used the training and test data, as explained in III-A. The numbers of sampled images M and that of the gyroscope measurements N were 7 and 13, respectively. The exposure time for each image was dependent on the lighting conditions. The average exposure time was 20 ms. The deblurring performance was evaluated using the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) in Bayer format, and the resultant images were displayed as RGB images using the de-mosaic algorithm in [39].

B. SELF-COMPARISON

To analyze the effectiveness of the proposed method in detail, we conducted self-comparison experiments. For these evaluations, we designed modified forms of EggNet, as shown in Fig. 7. The first network $\text{Base}^{\text{Deform}}$ substitutes the GTM and the IGTM with deformable convolutions, and it takes a blurry image as input. The second network $\text{Base}^{\text{I\&G}}$ exploits the concatenation of the blurry images and MGVs and adopts deformable convolutions instead of the proposed modules. Therefore, the offsets for the first deformable convolution in $\text{Base}^{\text{I\&G}}$ are computed from the integrated features of the image and gyroscope in a similar manner to the IGTM. The third network, Base^{GTM}, includes the GTM, but substitutes the IGTMs with deformable convolutions. The decoder part of these networks are the same as that of EggNet. Furthermore, in order to validate the effectiveness of L1 loss over L2 loss in image deblurring task, we additionally trained

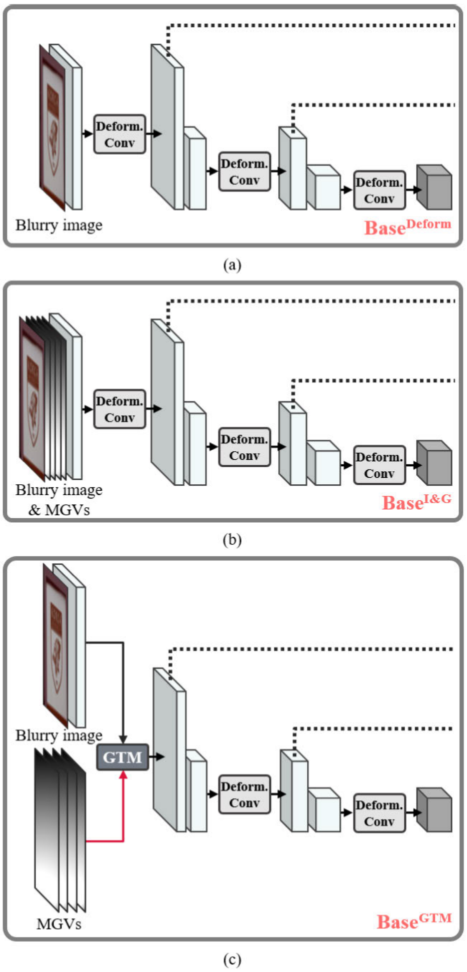

FIGURE 7. Architectures for self-comparison: (a) Base network with only deformable convolutions on the blurry image (Base$^{\text{Deform}}$); (b) base network with deformable convolution and concatenation of blurry image and MGVs as input (Base$^{\text{I\&G}}$); (c) base network with the proposed GTM, and deformable convolutions instead of IGTM (Base$^{\text{GTM}}$).

TABLE 2. Average PSNR and SSIM on the controlled case and the raw case.

Method	Params	Controlled case		Raw case	
	(MB)	PSNR (dB)	SSIM	PSNR(dB)	SSIM
Base$^{\text{Deform}}$	21.27	35.78	0.9622	35.69	0.9688
Base$^{\text{I\&G}}$	21.30	37.10	0.9844	<u>35.83</u>	<u>0.9698</u>
Base$^{\text{GTM}}$	21.24	<u>37.70</u>	<u>0.9857</u>	35.79	0.9692
EggNet$^{\text{L2}}$	21.39	37.21	0.9844	35.65	0.9686
EggNet	21.39	**37.72**	**0.9864**	**36.02**	**0.9702**
DeepGyro	118.38	34.07	0.9580	34.11	0.9598

The best and second-best results are boldfaced and underlined, respectively.

EggNet with L2 loss (EggNet$^{\text{L2}}$). These networks were trained and tested on both the controlled and raw cases, and the results are listed in Table 2. In terms of loss function, EggNet trained with L1 loss outperforms EggNet$^{\text{L2}}$ trained with L2 loss in both the cases.

1) EFFECTIVENESS OF FLEXIBLE RECEPTIVE FIELD

First, we compared Base$^{\text{Deform}}$ with DeepGyro to evaluate the effectiveness of the flexible receptive field for image deblurring. As listed in Table 2, the Base$^{\text{Deform}}$ outperforms DeepGyro in every case, even though both methods adopt the U-shaped network architecture. Moreover, DeepGyro exploits the additional gyroscope information and has five times more parameters than Base$^{\text{Deform}}$. These results validate the application of the flexible receptive field to achieve efficient network design.

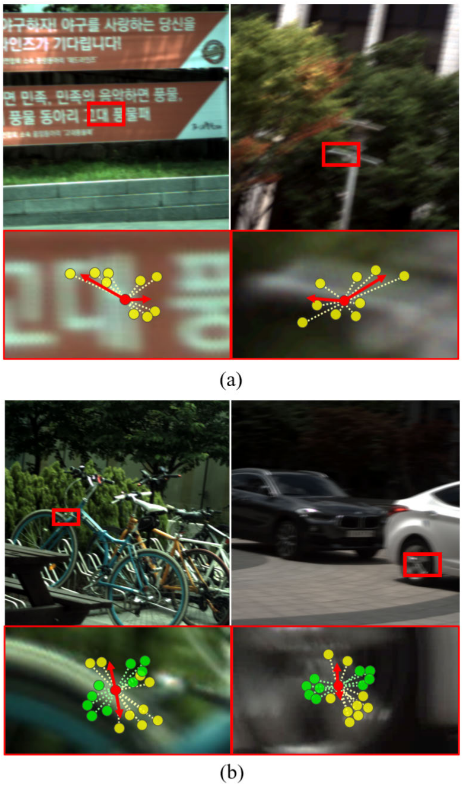

FIGURE 8. Visualized MGVs (red arrows), the MGO used in the GTM (yellow colored dashed line with circle), and the MGO used in the first IGTM (green dashed line with circle) on the sample images. (a) Sample images in which the GTM spreads the kernel in the blur direction indicated by the MGVs; (b) sample images in which the weight kernels of the GTM are spread in the wrong direction because of the degraded MGVs, and the weight kernels of the IGTM, which are correctly converted in the blur direction.

2) EFFECTIVENESS OF GYROSCOPE-BASED FLEXIBLE RECEPTIVE FIELD IN CONTROLLED CASE

The GTM aims to transform the receptive field in the blur direction using confident gyroscope measurements. The visualized MGO of the GTM shown in Fig. 8 (a) proves that the GTM performs as intended and transforms the kernel in the blur direction when confident gyroscope measurements are provided. To demonstrate the effectiveness of the GTM, we compared the deblurring performance in the controlled case. As shown in Table 2, Base$^{\text{GTM}}$ outperforms Base$^{\text{Deform}}$ in terms of PSNR by 1.92 (from 35.78 to 37.70). Therefore, it can be concluded that expanding the receptive field in the blur direction using the gyroscope measurements is more effective than deformable convolution in improving the deblurring performance. Compared with Base$^{\text{I\&G}}$, Base$^{\text{GTM}}$ shows a PSNR improvement of 0.60 (from 37.10 to 37.70).

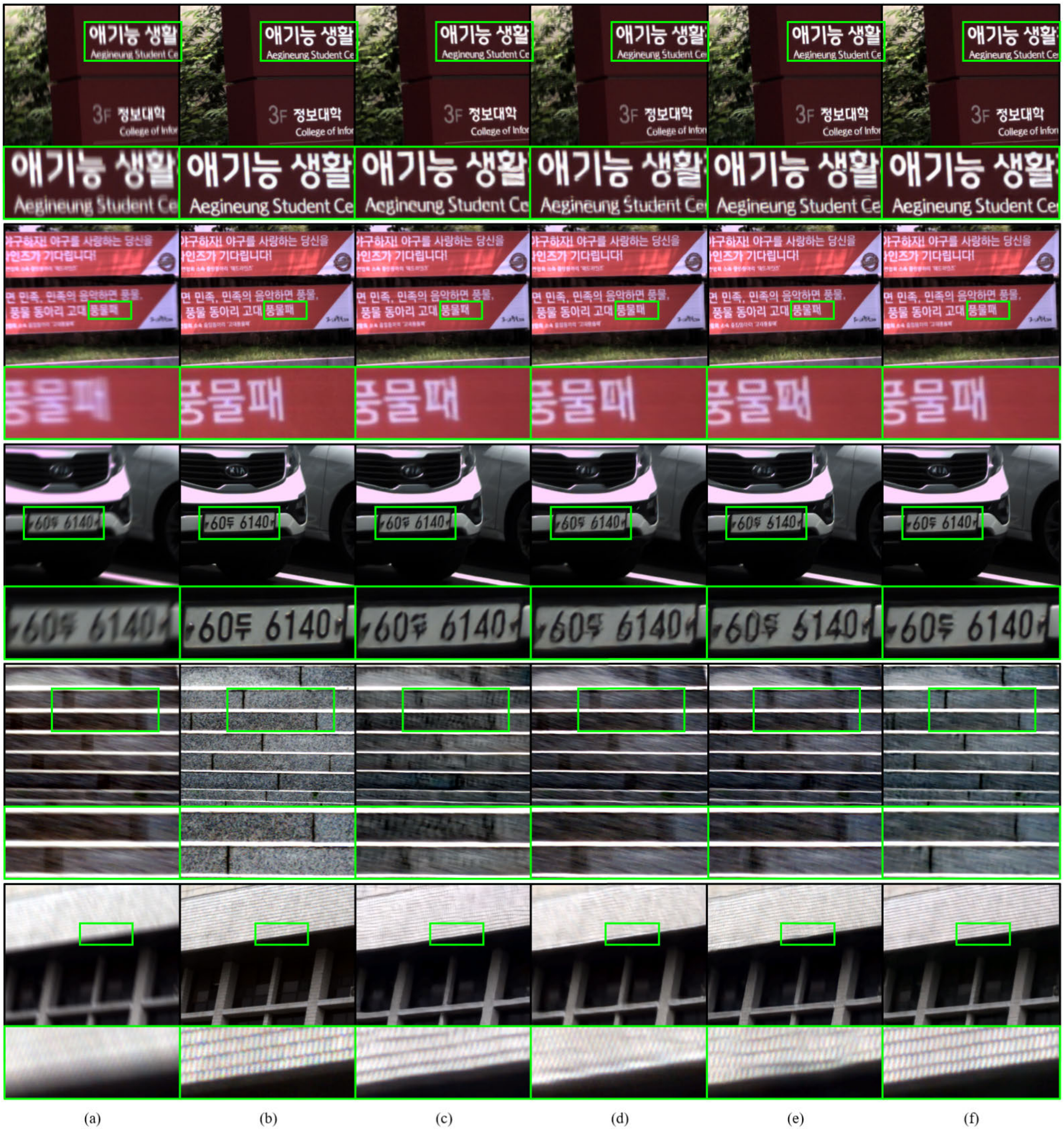

FIGURE 9. Experimental data and resulting images for various blur images in the controlled cases. From lightly blurred image (top) to heavily blurred image (bottom). The magnified parts of the images are shown at the bottom of each image. (a) Input blurry images; (b) target sharp images; (c) resultant images from DeepGyro; (d) resultant images from PSS-NSC; (e) resultant images from DMPHN; (f) resultant images from EggNet.

Consequently, it can be seen that converting the shape of the weight kernel in the front of the network using the gyroscope measurements is more effective than using both the image and gyroscope measurements if reliable gyroscope measurements are provided. Based on these results, we can assume that the features from the blurry image in the front of the network might not be sufficient for computing the offsets for the blur direction and magnitudes.

3) EFFECTIVENESS OF GYROSCOPE & IMAGE-BASED FLEXIBLE RECEPTIVE FIELD IN RAW CASE

When the gyroscope sensor is affected by noise or miscalibration, the gyroscope measurements may become unreliable. To prevent adverse effects from the degraded gyroscope measurements while exploiting the useful information from them, the IGTM aims to transform the receptive field in the blur direction using both the gyroscope measurements and the

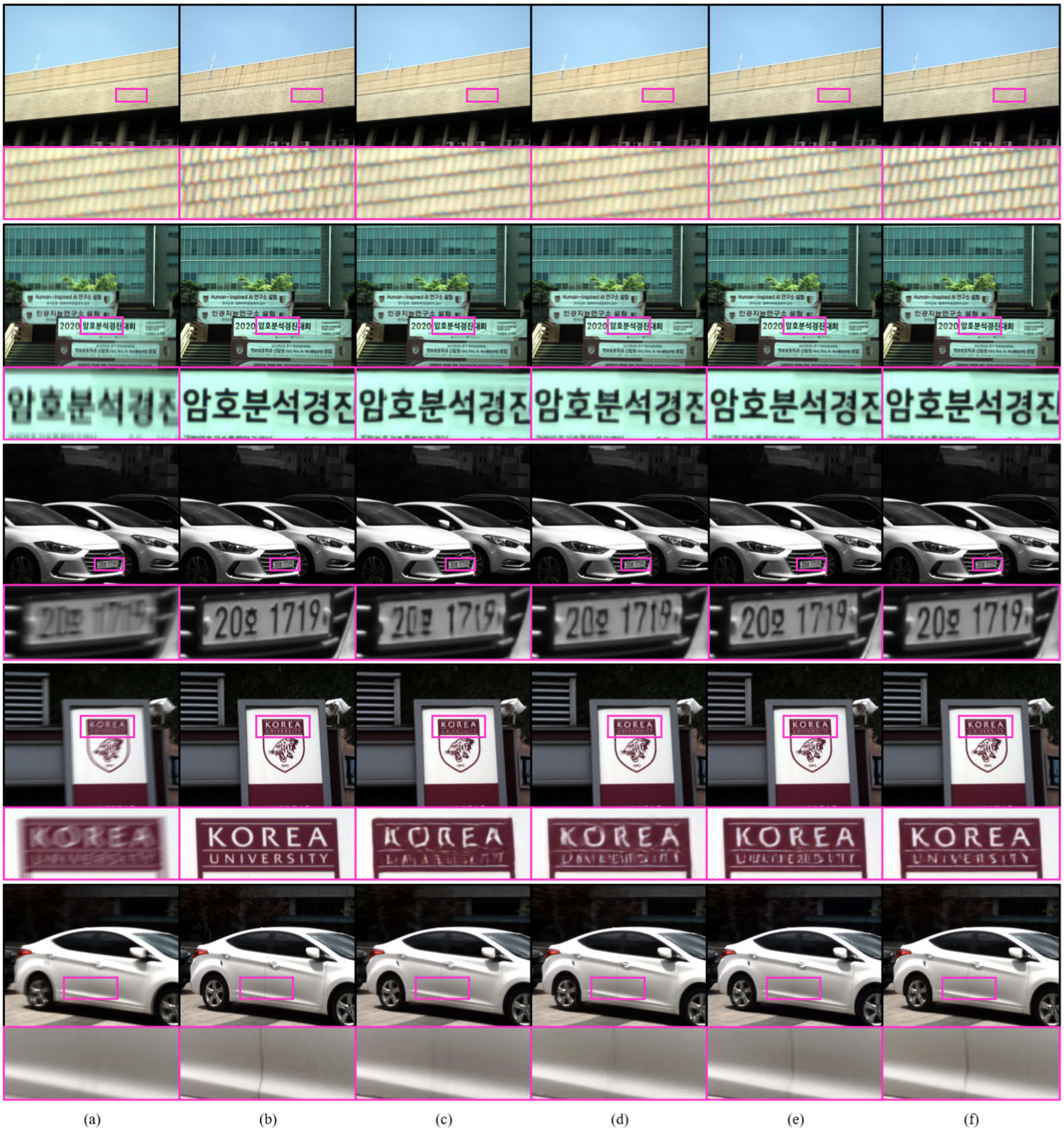

FIGURE 10. Experimental data and resulting images for various blur images in the raw cases. From lightly blurred image (top) to heavily blurred image (bottom). The magnified parts of the images are shown at the bottom of each image. (a) Input blurry images; (b) target sharp images; (c) resultant images from DeepGyro; (d) resultant images from PSS-NSC; (e) resultant images from DMPHN; (f) resultant images from EggNet.

feature maps from the input image. As shown in Fig. 8 (b), the IGTM spreads the kernel in the blur direction even when the GTM spreads the kernel in the wrong direction because of unreliable gyroscope measurements. We also compared the results of EggNet and Base^{GTM} in the raw case. As shown in Table 2, EggNet improves the deblurring performance in terms of PSNR by 0.23 (from 35.79 to 36.02). Base$^{I\&G}$ also exhibits a performance improvement of 0.04 dB (from 35.79 to 35.83) compared with Base^{GTM}. The visualized kernel and the comparison result demonstrate that converting the kernel shape using the integrated features from the gyroscope and image robustly transforms the kernel in the blur direction even if the gyroscope is adversely affected by noise or miscalibration.

C. DEBLURRING UNDER DIVERSE CONDITIONS: PERFORMANCE vs. NOISE & MISCALIBRATION

As mentioned in the previous sections, the gyroscope sensor can be affected and degraded by noise and miscalibration. Continuing from Section IV-B3, we conducted more

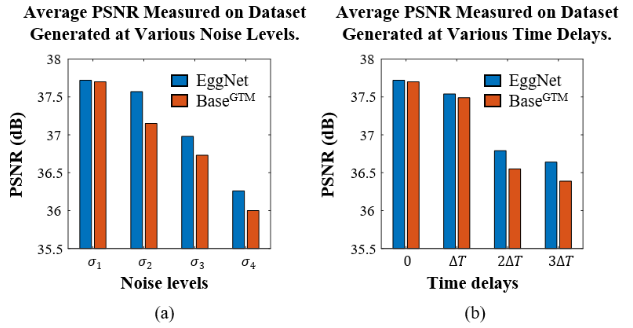

FIGURE 11. Average PSNR of EggNet and Base$^{\mathrm{GTM}}$ on the variant dataset. (a) Average PSNR measured on the dataset with noise; (b) average PSNR measured on the dataset with miscalibration.

experiments under diverse conditions to validate the effectiveness of EggNet in detail.

First, we studied the condition in which the gyroscope measurements are corrupted by noise. Because the gyroscope measurements cannot be directly expressed in the image domain, we added noise to the MGVs which indicate pixel movements in the image domain. Random noise offsets were added to each channel of the MGVs because the values of the MGVs change monotonically. Based on the fact that the values in the MGVs were usually distributed in range of $[-10, 10]$, we considered Gaussian noise with sigma of 0 (σ_1), 0.125 (σ_2), 0.25 (σ_3), and 1 (σ_4). The blurry and sharp image pairs used in the controlled case and the noise-added MGVs were used for network training. The bar chart in Fig. 11 (a) presents the average PSNR of EggNet and Base$^{\mathrm{GTM}}$ on the diverse noise-added MGVs. To improve the reliability of the results considering the randomness of the noise, the reported PSNR was averaged by performing 10 repeats of the experiments on the test data. Under the noise-free condition, σ_1, the performance difference between EggNet and Base$^{\mathrm{GTM}}$ is imperceptible. With σ_2 noise, EggNet still showed comparable results, but Base$^{\mathrm{GTM}}$ failed to prevent adverse effects from the noise. Under harsher noise conditions, although the PSNR of both methods decreased, the performance of EggNet deteriorated less compared with that of Base$^{\mathrm{GTM}}$.

We next considered the presence of miscalibration between the gyroscope sensor and image sensor. Similar to the noise case, the blurry and sharp images from the controlled case and the MGVs computed from the delayed gyroscope measurements were exploited to model the miscalibration. The bar chart in Fig. 11 (b) presents the average PSNR of EggNet and Base$^{\mathrm{GTM}}$ when miscalibration were present. In Fig. 11 (b), ΔT is one-tenth of the image exposure time. EggNet robustly performed image deblurring compared with Base$^{\mathrm{GTM}}$ even in the presence of miscalibration.

From these results, we validated the ability of EggNet to effectively exploit the gyroscope measurements and robustly perform image deblurring even if these measurements are degraded. From an industrial perspective, noise and miscalibration should be addressed thoroughly so that the image and gyroscope sensors can be simultaneously exploited for image deblurring. This would increase the hardware or software complexity. We found that using our proposed EggNet, the degradation under σ_2 noise and ΔT miscalibration is still acceptable, and therefore the increment of the hardware or software complexity can be mitigated.

TABLE 3. Average PSNR, SSIM, and execution time on the controlled case and the raw case.

Method		DeepGyro	PSS-NSC	DMPHN -Stack(4)	EggNet
Additional Gyro Params (MB)		✓ 118.38	✗ 24.51	✗ 82.67	✓ **21.39**
Control. case	PSNR (dB)	34.07	35.17	36.14	**37.72**
	SSIM	0.9580	0.9629	0.9669	**0.9864**
	Time (ms)	**15.27**	59.80	115.92	21.64
Raw case	PSNR (dB)	34.11	34.90	35.13	**36.02**
	SSIM	0.9598	0.9639	0.9642	**0.9702**
	Time (ms)	**41.62**	114.56	265.99	79.74

The best and second-best results are boldfaced and underlined, respectively.

D. COMPARISON WITH OTHER NETWORKS

We carried out quantitative and qualitative comparisons of EggNet with other deblurring networks. Because there are no other gyro-aided deblurring methods except for DeepGyro [23], we compared the proposed EggNet with non-gyro-aided deblurring networks. Among the various state-of-the-art non-gyro-aided deblurring networks [16]–[19], DMPHN [16] and PSS-NSC [17], which have been publicly released by their authors, were trained on our gyro-image paired dataset. Except for DMPHN, which was trained with a mini-batch size of one owing to memory limitations, a mini-batch size of two was used for training in the raw case, and the learning schedules of each network were followed in accordance to the strategy of the network. As demonstrated in Table 3, the proposed EggNet outperforms the competing networks in terms of PSNR and SSIM. In addition to the superior PSNR and SSIM, EggNet used the smallest number of parameters. Compared with the conventional gyro-aided deblurring network DeepGyro, the proposed EggNet achieved significant improvement in both cases while using only one-fifth the number of parameters used in DeepGyro. Although EggNet outperformed both DMPHN and PSS-NSC in terms of accuracy and efficiency, both of these methods were not designed for gyro-aided deblurring. In consideration of this, our approach can be integrated with the approaches used in DMPHN and PSS-NSC. In these two networks, a sharp image is obtained in a progressive manner and the parameters are shared through the network to further improve the deblurring performance and reduce the model complexity. Overall, the proposed network showed the best performance in terms of PSNR and SSIM, exploited the fewest number of parameters, and recorded the second-fastest processing time. The resultant images for the controlled case and the raw case from our image-gyro paired dataset are shown in Fig. 9 and Fig. 10, respectively. For clarity, the magnified parts of each image are displayed at the bottom. As can be seen, the proposed EggNet successfully restores the latent sharp images and produces sharpest detailed textures in all cases.

V. CONCLUSION

We presented an effective gyroscope-guided network (EggNet) that exploits a flexible receptive field to achieve effective image deblurring using an additional gyroscope sensor. Two specialized modules, namely, the GTM and the IGTM, are sequentially placed in EggNet. They adaptively transform the weight kernel in the blur direction to train the network with features related to the nature of the blur. The extensive experiments conducted on variants of our image-gyro dataset clearly demonstrate the effectiveness of our approach. The proposed EggNet robustly performs gyroscope-aided image deblurring compared to conventional methods.

ACKNOWLEDGMENT

(Seowon Ji, Jun-Pyo Hong, and Jeongmin Lee contributed equally to this work.) The authors thank Cheol-Hwan Yoo of Korea University for capturing image-gyro paired dataset.

REFERENCES

[1] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 495–502.

[2] S. Tang, W. Gong, W. Li, and W. Wang, "Non-blind image deblurring method by local and nonlocal total variation models," *Signal Process.*, vol. 94, pp. 339–349, Jan. 2014.

[3] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. CVPR*, Jun. 2011, pp. 233–240.

[4] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Proc. IEEE Int. Conf. Comput. Photogr. (ICCP)*, Apr. 2013, pp. 1–8.

[5] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1107–1114.

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

[7] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Blind image deblurring using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1628–1636.

[8] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1790–1798.

[9] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, Jul. 2016.

[10] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 221–235.

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

[12] 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. (CVPR)*, Jul. 2017, pp. 3883–3891.

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

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

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

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

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

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

[19] K. Purohit and A. Rajagopalan, "Region-adaptive dense network for efficient motion deblurring," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 11882–11889.

[20] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–9, Jul. 2010.

[21] O. Sindelar and F. Sroubek, "Image deblurring in smartphone devices using built-in inertial measurement sensors," *J. Electron. Imag.*, vol. 22, no. 1, Feb. 2013, Art. no. 011003.

[22] O. Sindelar, F. Sroubek, and P. Milanfar, "Space-variant image deblurring on smartphones using inertial sensors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 191–192.

[23] J. Mustaniemi, J. Kannala, S. Sarkka, J. Matas, and J. Heikkila, "Gyroscope-aided motion deblurring with deep networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1914–1922.

[24] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2497–2506.

[25] Q. Fan, D. Chen, L. Yuan, G. Hua, N. Yu, and B. Chen, "Decouple learning for parameterized image operators," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 442–458.

[26] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, vol. 30, pp. 1333–1341, Dec. 2001.

[27] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," *Continuously Stirred Tank Reactor*, vol. 1, no. 2, p. 13, 2011.

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

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput.-Assisted Intervent.* New York, NY, USA: Springer, 2015, pp. 234–241.

[30] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf, "Fast removal of non-uniform camera shake," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 463–470.

[31] A. Z. Averbuch, A. Schclar, and D. L. Donoho, "Deblocking of block-transform compressed images using weighted sums of symmetrically aligned pixels," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 200–212, Feb. 2005.

[32] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Red Hook, NY, USA: Curran Associates, 2016, pp. 4898–4906.

[33] J. Lee, S.-W. Ji, S.-J. Cho, J.-P. Hong, and S.-J. Ko, "Deep learning-based deblur using gyroscope data," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE-Asia)*, Nov. 2020, pp. 1–4.

[34] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2017–2025.

[35] Y. Jeon and J. Kim, "Active convolution: Learning the shape of convolution for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4201–4209.

[36] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in Pytorch," in *Proc. NIPS Autodiff Workshop*, Long Beach, CA, USA, 2017.

[39] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, 2000.

SEOWON JI received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include image processing, computer vision, and deep-learning.

JUN-PYO HONG received the B.S. degree in electronics engineering from Tinghua University, in 2013. He joined the Computer Vision and Image Processing Laboratory, Department of Electronic Engineering, Korea University, in 2019. His interests include image processing, computer vision, and deep-learning.

JEONGMIN LEE received the B.S. degree in electrical engineering from Korea University, in 2020. He joined the Computer Vision and Image Processing Laboratory, Department of Electrical Engineering, Korea University, in 2020. His interests include image processing, computer vision, and deep-learning.

SEUNG-JIN BAEK received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2007 and 2013, respectively. He joined the Digital Media and Communications Research and Development Center, Samsung Electronics Company Ltd., Suwon, South Korea, in 2013, where he was as a Senior Engineer, from 2014 to 2015. He was a Staff Engineer with the Visual Display Business Division, Samsung Electronics Company Ltd., from 2015 to 2020. He is currently a Research Professor with the Research Institute of Information and Communication Technology, Korea University. His current research interests include deep learning, image processing applications, and computer vision.

SUNG-JEA KO (Fellow, IEEE) received the B.S. degree in electronic engineering from Korea University, in 1980, and the M.S. and Ph.D. degrees in electrical and computer engineering from the State University of New York at Buffalo, in 1986 and 1988, respectively.

From 1988 to 1992, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn. In 1992, he joined the Department of Electronic Engineering, Korea University, where he is currently a Professor. He has published over 210 international journals articles. He also holds over 60 registered patents in fields, such as video signal processing and computer vision.

Prof. Ko is currently a member of the National Academy of Engineering of Korea. He was a recipient of the 1999 LG Research Award. He received the Hae-Dong Best Paper Award from the Institute of Electronics and Information Engineers (IEIE), in 1997, the Best Paper Award from the IEEE Asia Pacific Conference on Circuits and Systems, in 1996, the Research Excellence Award from Korea University, in 2004, the Technical Achievement Award from the IEEE Consumer Electronics (CE) Society, in 2012, the 15-Year Service Award from the TPC of ICCE, in 2014, and the Chester Sall Award (First Place Transaction Paper Award) from the IEEE CE Society, in 2017. He was honored with the Science and Technology Achievement Medal from the Korean Government, in 2020. He has served as the General Chairman for ITC-CSCC 2012 and the General Chairman for IEICE 2013. He was the President of the IEIE, in 2013, the Vice President of the IEEE CE Society, from 2013 to 2016, and a Distinguished Lecturer of the IEEE, from 2015 to 2017. He is also an Editorial Board Member of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.