



Deep Recurrent Network for Fast and Full-Resolution Light Field Deblurring

Jonathan Samuel Lumentut, *Student Member, IEEE*, Tae Hyun Kim, *Member, IEEE*,
Ravi Ramamoorthi , *Fellow, IEEE*, and In Kyu Park , *Senior Member, IEEE*

Abstract—The popularity of parallax-based image processing is increasing while in contrast early works on recovering sharp light field from its blurry input (deblurring) remain stagnant. State-of-the-art blind light field deblurring methods suffer from several problems such as slow processing, reduced spatial size, and simplified motion blur model. In this paper, we solve these challenging problems by proposing a novel light field recurrent deblurring network that is trained under 6 degree-of-freedom camera motion-blur model. By combining the real light field captured using Lytro Illum and synthetic light field rendering of 3D scenes from UnrealCV, we provide a large-scale blurry light field dataset to train the network. The proposed method outperforms the state-of-the-art methods in terms of deblurring quality, the capability of handling full-resolution, and a fast runtime.

Index Terms—Recurrent network, light field image, blind deblurring, dataset, 6-DOF motion.

I. INTRODUCTION

IN RECENT years, 2D image deblurring problem became a popular topic in computer vision with the specific objectives such as fast and robust processing [5], [6], [10], [13], [17]. Unlike the 2D image, a 4D light field (LF) image contains both spatial and angular information to represent a pixel value. Thus, a variety of tasks, such as depth estimation [19], refocus [11], and view synthesis [4], have been carried out by exploiting the 4D information from an LF image. However, there exist few works on LF deblurring. The latest one by Srinivasan *et al.* [15] estimates 3-degree of freedom (3-DOF) LF camera motion to perform deblurring. Unfortunately, significant drawbacks are shown from this state-of-the-art work. First, it suffers from high computational complexity (~ 2 hours for single LF deblurring on a GPU). Second, their approach is limited by reduced spatial

resolution capability. These problems occur because they optimize the whole 4D LF directly, which takes significantly large space in GPU memory.

Recently, a couple of deblurring algorithms are introduced to tackle those problems. The limited 3-DOF translation camera motion is improved in the work of Mohan *et al.* [9] by utilizing 2D translation and 1D rotation to estimate the correct camera motion for deblurring. Moreover, they provide the capability of deblurring full spatio-angular resolution LF. On the other hand, Lee *et al.* [8] show the capability of solving depth estimation and LF deblurring jointly. It utilizes the intrinsic and extrinsic parameters of LF to predict the 6-DOF camera motion deblurring. However, these approaches [8], [9] take around 30 minutes for an LF, which indicates time consuming problem. That is, currently available LF deblurring methods are not suitable for post-capture image enhancement.

To overcome these problems, we introduce a fast LF deblurring network that is trained using a newly achieved large-scale blurry LF dataset. The dataset is generated by providing 6-DOF camera motion that synthetically blurs the sharp LF input. Subsequently, in the proposed deep neural network, we combine the convolution and deconvolution layers with recurrent scheme to achieve robust and consistent LF deblurring result. Moreover, we also apply an effective angular sampling scheme to train with the large dataset effectively. The network is trained in an end-to-end fashion and demonstrates superior performance compared to the state-of-the-art methods through quantitative and qualitative results.

II. 4D LF IMAGE DEBLURRING

In general, the 4D LF can be represented as 2D spatial (x, y) and 2D angular (u, v) parameterizations. A specific angular location represents a 2D image from a unique viewpoint. Thus, each image varies in the angular domain, and it is called a sub-aperture image (SAI) of an LF. By obtaining the SAIs, one may deblur them independently using any 2D deblurring algorithm. However, 2D deblurring algorithms are not customized to utilize the consistency between SAIs. Therefore, the deblurring performance is far from satisfactory. In this context, we design an effective deblurring network to preserve the SAIs' consistency. Also, the proposed network can fulfill the fast processing and enable full-resolution LF deblurring.

A. LF Recurrent Deblurring Network

We introduce a learning based LF deblurring network, dubbed Light Field Recurrent DeBlurring Network (LFRDBN), which

Manuscript received July 27, 2019; revised October 6, 2019; accepted October 10, 2019. Date of publication October 14, 2019; date of current version November 15, 2019. This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1702-06 and SRFC-IT1901-06. This work was supported under the framework of international cooperation program managed by the College of Engineering, Inha University. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sheng Li. (*Corresponding author: In Kyu Park.*)

J. S. Lumentut and I. K. Park are with the Department of Information and Communication Engineering, Inha University, Incheon 22212, South Korea (e-mail: jlumentut@gmail.com; pik@inha.ac.kr).

T. H. Kim is with the Department of Computer Science, Hanyang University, Seoul 04763, South Korea (e-mail: taehyunkim@hanyang.ac.kr).

R. Ramamoorthi is with the Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: ravir@cs.ucsd.edu).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier 10.1109/LSP.2019.2947379

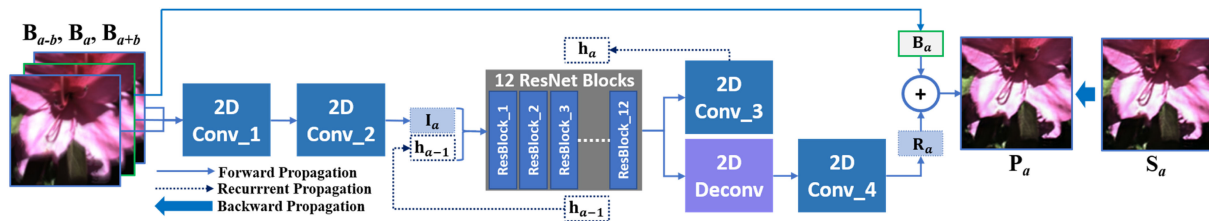


Fig. 1. The proposed end-to-end LF deblurring network constructed with the combination of convolution-deconvolution and residual [3] layers. The network takes multiple images as input, renders deblurred SAI, and works in a recurrent manner.

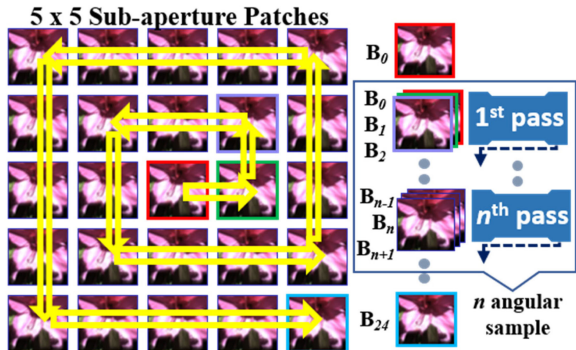


Fig. 2. **Left:** The 4D to 3D stacking in spiral direction in the proposed approach. The $u \times v$ SAIs from single LF are stacked from 4D (x, y, u, v) into 3D (x, y, a) . **Right:** After the SAIs are stacked in 3D, n angular sampling is performed to take several consecutive SAIs.

runs in a recurrent manner as depicted in Fig. 1. We employ the combination of convolution-deconvolutional and residual styles on the network, which are proven to produce the promising result on image/video deblurring [6], [10]. The proposed network consists of 2 layers of convolution, 12 ResNet blocks [3], 1 convolutional layer for recurrent propagation, and 2 layers of deconvolution-convolution for predicting an output. Fully convolutional operation is applied to the network so that arbitrary input size is allowed.

Another unique aspect in the proposed network is that it takes a number (i.e., j) of consecutive SAIs as input that benefits the network of preserving its consistency between horizontal or vertical or horizontal-vertical directions. To achieve full consistency along all SAIs, SAIs are reshaped from 4D (x, y, u, v) to 3D (x, y, a) and stacked as illustrated in Fig. 2 where the angular domain varies along a . The proposed network is trained under spiral direction, as shown in Fig. 2, to capture all possible direction changes between consecutive SAIs (horizontal, vertical, and horizontal-vertical).

In our implementation, j is set to 3 so that the network takes at least 2 neighboring SAIs. The network takes input of 3 consecutive SAIs to produce a single predicted deblurred SAI at angular position a . In practice, we perform angular sampling, as shown in Fig. 2 by taking n SAIs to be processed in the network during n time steps ($n = 10$ in our experiments). This approach is ideal for the proposed network, which is incorporated with the recurrent connection. Thus, the network may learn to ensure the consistency among all SAIs by passing the hidden feature h to the next consecutive SAI process. Table I shows the configuration details of the LFRDBN subblocks.

TABLE I
LFRDBN'S ARCHITECTURE DETAILS

Layer	Filter + Normalization + Activation + Stride
2D Conv_1	7×7 + Instance Normalization + ReLU + 1
2D Conv_2	3×3 + Instance Normalization + ReLU + 2
ResBlock_1-12	3×3 + Instance Normalization + ReLU + 1
2D Conv_3	3×3 + Instance Normalization + ReLU + 1
2D Deconv	3×3 + Instance Normalization + ReLU + 2
2D Conv_4	3×3 + Tanh + 1

TABLE II
LFRDBN'S INPUT AND OUTPUT RESOLUTION

Layer	Input	Output
2D Conv_1	$H \times W \times 9$ $(\mathbf{B}_{a-1}, \mathbf{B}_a, \mathbf{B}_{a+1})$	$H \times W \times 64$
2D Conv_2	$H \times W \times 64$	$H/2 \times W/2 \times 64 (\mathbf{I}_a)$
ResBlock_1	$H/2 \times W/2 \times 128$ $(\mathbf{I}_a, \mathbf{h}_{a-1})$	$H/2 \times W/2 \times 128$
ResBlock_2-12	$H/2 \times W/2 \times 128$	$H/2 \times W/2 \times 128$
2D Conv_3	$H/2 \times W/2 \times 128$	$H/2 \times W/2 \times 64 (\mathbf{h}_a)$
2D Deconv	$H/2 \times W/2 \times 128$	$H \times W \times 128$
2D Conv_4	$H \times W \times 128$	$H \times W \times 3 (\mathbf{R}_a)$

B. Network Details

The details of the input and output size of LFRDBN's subblock are shown in Table II. The proposed network is fed from **2D Conv_1** layer with a concatenation of 3 consecutive blurry SAIs, $j = 3$ ($\mathbf{B}_{a-b}, \mathbf{B}_a, \mathbf{B}_{a+b}$; where $b = 1$) and is assigned to predict a residual SAI (\mathbf{R}_a). In the final stage, \mathbf{R}_a needs to be added with the input (\mathbf{B}_a) to produce the final output of deblurred SAI (\mathbf{P}_a) as listed in Table II. Unlike the original ResNet model [3], the proposed network is constructed with instance normalization [16] and ReLU activation. Note that, the layer of **ResBlock_1** in Table II is separated from the others (**ResBlock_2-12**) to show the concatenated input of previously hidden feature (\mathbf{h}_{a-1}) and the feature (\mathbf{I}_a) from **2D Conv_2** layer. The next hidden feature (\mathbf{h}_a) is passed through the **2D Conv_3** layer. The proposed network is fully convolutional, and thus, its spatial size (H, W) can be arbitrary. The final layer of **2D Conv_4** is activated with *tanh* to predict the residual SAI (\mathbf{R}_a).

C. 6-DOF LF Camera Motion

In this work, we derive 6-DOF (3-DOF translation and 3-DOF rotation) LF camera motion to generate motion-blurred LF synthetically. The earlier work [15] approximated only 3-DOF LF translational blur model by 2D in-plane motion along the

x - and y -axis and 1D out-of-plane motion along the z -axis. According to Bok's [1] model, an LF image is considered as a collection of pinhole camera models which are positioned at angular coordinates (u, v) . The blurry LF is produced by integrating all warped copies of the sharp LF (S) at each time t . Thus, a blurry pixel B at a location (x, u) on LF is defined as

$$B(x, u) = \int_t S(x, u + p_x(t) - xp_z(t)) dt, \quad (1)$$

where $p_x(t)$ and $p_z(t)$ are the in-plane and out-of-plane translational motion from the 3-DOF model [15]. As suggested in the work of Whyte *et al.* [18], we consider out-of-plane rotation as the shifting of spatial coordinates with the rotation angles as shown in the following

$$p'_x(t) = p_x(t) + f\theta_t; p'_y(t) = p_y(t) + f\phi_t, \quad (2)$$

where θ_t , ϕ_t , and f are the rotation angle around y -axis, x -axis and the focal length, respectively. On the other hand, in-plane rotation is simply applied by a 2D rotation, as defined in

$$x' = (x - \delta_u) \cos(\psi_t) - (y - \delta_v) \sin(\psi_t) + \delta_u, \quad (3)$$

$$y' = (x - \delta_u) \sin(\psi_t) + (y - \delta_v) \cos(\psi_t) + \delta_v, \quad (4)$$

where (δ_u, δ_v) are the baselines between SAI while ψ_t is the z -axis rotation angle at time t . Combining Eq. (2) to (4) to Eq. (1) generates a new 6-DOF motion blur equation for LF and it can be simply written as

$$B(x, u) = \int_t S(x', u + p'_x(t) - x'p_z(t)) dt. \quad (5)$$

The representation of 4D LF parameters in Eq. (5) is reduced from (x, y, u, v) to (x, u) to preserve the parameter consistency of Eq. (1). However, the extension to 4D LF is straightforward.

III. NETWORK TRAINING

To train the proposed LFRDBN, we generate a large-scale LF blur dataset using our real Lytro Illum images and synthetic 3D scenes acquired from UnrealCV [14]. The synthetic 3D scenes are included in our dataset to provide images with high contrast that are not present in the Lytro Illum data as discussed in [2]. To acquire the SAIs from the 3D rendering of UnrealCV, we simulate virtual camera movement and captured scenes from multiple views using the publicly available source code [14]. Note that the distance between the offset view (δ_u, δ_v) in UnrealCV is set to 1 pixel. This distance is also applied to the real dataset since the baseline distance of Lytro Illum is approximately 0.9 pixel which is estimated using Bok's toolbox [1]. It is rounded up to 1 pixel so that the integration in Eq. (5) is performed on the integer location of the shifted pixels.

Our LF dataset has the combination of real LF captured using our own Lytro Illum and 3D rendered scenes from UnrealCV captured using [14]. We divide a training and test set into 360 and 40 LF images, respectively. The population ratio between Lytro Illum and UnrealCV datasets is 50:50. With these sharp 360 training data, we generate synthetic non-uniform blur dataset using randomly sampled 860 6-DOF camera motions based on Eq. (5). To synthesize the motion blur, we simulate the shutter-time effect on light field camera by taking the average of differently warped images from the 6-DOF camera motion

TABLE III
ABLATION STUDY ON VARIATED ANGULAR SAMPLE AND NEIGHBOR

j	PSNR/SSIM ($n=10$)	n	PSNR/SSIM ($j=3$)
1	24.68 / .819	5	24.69 / .824
3	24.72 / .824	10	24.72 / .824
5	24.68 / .823	15	24.62 / .822

during shutter-time T . Our camera motions are randomly generated; thus, both large and small motion blurs are included in the dataset. For a single LF, we utilize a total of 5×5 SAIs with the spatial size of 320×512 on each LF data that is extracted using [1]. Note that we trim the original spatio-angular size from $13 \times 13 \times 383 \times 552$ to $5 \times 5 \times 320 \times 512$ to avoid ghosting and vignetting effects and the boundary noise from the synthetic blur. The sharp frame at $T/2$ is selected as the ground truth sharp LF image. The 3-DOF translation and 3-DOF rotation are interpolated using Bézier and spherical linear method, respectively.

Using the acquired dataset, LFRDBN is trained with a mini-batch size 1 and the RGB color channel switching is applied to augment the dataset. SAIs are cropped to 256×256 spatially, and the network is optimized using ADAM optimizer with the learning rate of 0.0001 for 400 K iterations with NVIDIA GTX 1080 GPU. The mean-squared error between the sharp ground truth (S_a) and output SAI (P_a) is utilized for finding the optimal weights during backpropagation. In the training phase, angular sampling with $n = 10$ is applied to provide the recurrent procedure in the network using the 10 consecutive SAIs sampled in spiral direction as illustrated in Fig. 2. Note that the recurrent fashion supports arbitrary angular resolution.

IV. EXPERIMENTAL RESULTS

We provide ablation study results by changing the numbers of neighboring SAIs (j) and angular sampling (n). We first conduct experiment with different j ($= 1, 3, 5$) with fixed $n = 3$. Next, we measure the performance by changing n ($= 5, 10, 15$) with fixed $j = 3$. As shown in Table III, best performance is achieved with $n = 10$ and $j = 3$. Taking 1 previous and next neighboring frames ($j = 3$), achieves higher result compared to no-neighbor case ($j = 1$). The quantitative result of $j = 5$ is still lower than $j = 3$ since it needs more iterations for convergence. On the other hand, when n is varied, best PSNR is achieved with $n = 10$. However, same SSIM results are shown when $n = 5$ and $n = 10$. Increasing n without creating larger channel on the recurrent propagation degrades the performance. In this study, channel number ($= 64$) in the recurrent propagation only fits when $n \leq 10$. The network in the ablation experiment is trained around 200 K iterations while the full model for test procedure is trained for 400 K iterations. This causes PSNR/SSIM scores of $n = 10$, $j = 3$ (24.72/.824) in Table III to be lower than the scores in Full-Resolution 6-DOF case in Table IV (25.73/.840).

In the testing stage, we compare our result with the state-of-the-art LF deblurring algorithm [15]. For more objective evaluation, we also perform the comparison with 2D image deblurring algorithms by Krishnan *et al.* [5] and Pan *et al.* [12] as suggested by Lai *et al.* [7]. Note that all SAIs (*i.e.*, $n = 5 \times 5 = 25$) are restored during test time.

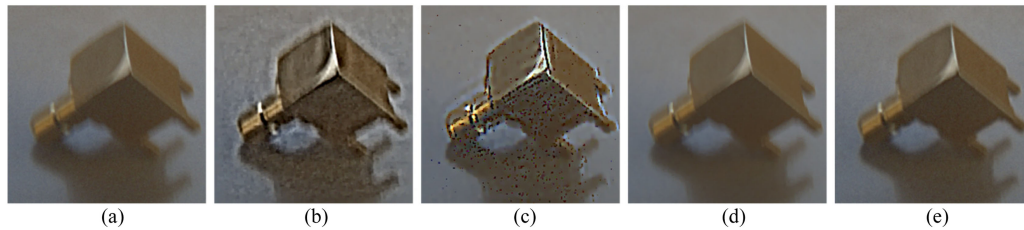


Fig. 3. Deblurring result from the real motion blurred LF originally obtained from [15]. (a) Real blurred LF, deblurring result of (b) Krishnan *et al.* [5], (c) Pan *et al.* [12], (d) Srinivasan *et al.* [15], and (e) ours, respectively. The images shown are the central SAI of all deblurred SAIs.

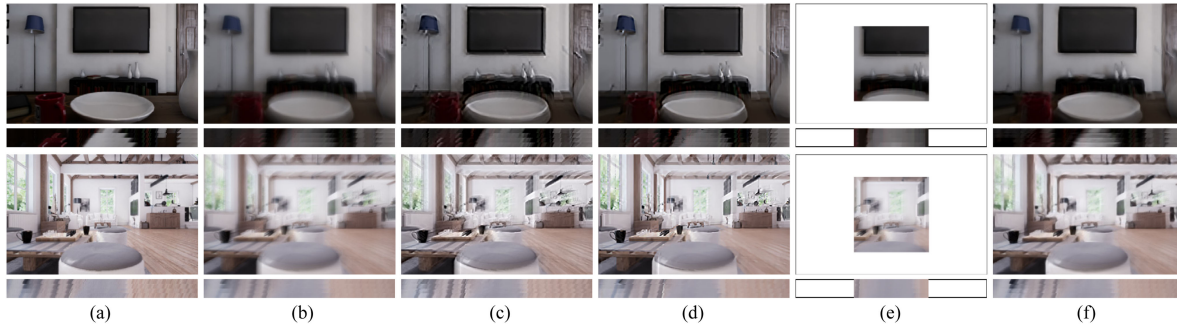


Fig. 4. Deblurring results from the 6-DOF (**first row**) and 3-DOF [15] (**second row**) motion blurred LF. (a) Ground truth, (b) Blurred LF, deblurring result of (c) Krishnan *et al.* [5], (d) Pan *et al.* [12], (e) Srinivasan *et al.* [15], and (f) ours, respectively. The images shown are the central SAI of all deblurred SAIs. Below each SAI is the corresponding Epipolar Plane Image (EPI) which shows that the parallax is reconstructed between SAI.

TABLE IV
PSNR/SSIM/RMSE RESULT BETWEEN ALGORITHMS ON 6-DOF MOTION

Method	Cropped Resolution ($40 \times 5 \times 5 \times 200 \times 200 \times 3$)	Full Resolution ($40 \times 5 \times 5 \times 320 \times 512 \times 3$)
[5]	23.41 / .778 / 0.0732	23.41 / .774 / 0.0703
[12]	21.70 / .732 / 0.0875	21.58 / .730 / 0.0857
[15]	23.61 / .765 / 0.0703	N/A (Memory Overflow)
Proposed	26.57 / .846 / 0.0490	25.73 / .840 / 0.0531

TABLE V
PSNR/SSIM/RMSE RESULT BETWEEN ALGORITHMS ON 3-DOF MOTION

Method	Cropped Resolution ($6 \times 5 \times 5 \times 200 \times 200 \times 3$)	Full Resolution ($6 \times 5 \times 5 \times 320 \times 512 \times 3$)
[5]	24.50 / .787 / 0.0717	25.08 / .809 / 0.0622
[12]	21.98 / .731 / 0.0866	22.86 / .767 / 0.0769
[15]	24.75 / .781 / 0.0673	N/A (Memory Overflow)
Proposed	27.57 / .855 / 0.0453	27.21 / .871 / 0.0459

A smaller set of 3-DOF translation data in [15] is used to show the advantage of our 6-DOF model over 3-DOF. The PSNR, SSIM, and RMSE of the 6-DOF and 3-DOF test sets are shown in Table IV and Table V, respectively. In terms of quantitative performance, the proposed method achieves the best PSNR, SSIM, and RMSE scores as shown in Table IV and Table V. Notably, the proposed LFRDBN, which is trained to handle 6-DOF motion, can also remove blur caused by the 3-DOF motion as shown in Table V.

LFRDBN can deblur a full 4D LF within 1.8 seconds, which is 16,000 times faster than the state-of-the-art method [15] on the same GPU, as shown in Table VI. As most of the LF dataset has objects closer to the lens to capture the parallax effect,

TABLE VI
COMPARISON OF EXECUTION TIME

Method	Cropped Resolution [sec.] ($5 \times 5 \times 200 \times 200 \times 3$)	Full Resolution [sec.] ($5 \times 5 \times 320 \times 512 \times 3$)
[5]	~100 (CPU)	~445 (CPU)
[12]	~240 (CPU)	~1,000 (CPU)
[15]	~8,000 (GPU)	N/A (Memory Overflow)
Proposed	~0.5 (GPU)	~1.7 (GPU)

motion blurs within those objects are generally large and difficult to remove. The proposed method can handle these blurs and produces qualitatively much better results as shown in Fig. 4. The first and second rows in Fig. 4 are the deblurring result on the test set with 6-DOF (our model) and 3-DOF ([15]) motion blur, respectively. In addition, deblurring result with real blurred LF case is shown in Fig. 3. Note that the result of [15] are shown in cropped resolution as their algorithm is limited by the hungry-memory issue. Finally, to show a better viewing quality and more result, readers may refer to our project webpage (<http://image.inha.ac.kr/project/LFRDBN>).

V. CONCLUSION

We presented the LF deblurring method that is based on the recurrent network. The proposed network could secure the large receptive field to remove large motion blur and enforce consistency among adjacent SAIs. To do it, we introduced an effective angular sampling for training the network with the recurrent scheme. Moreover, the proposed network was designed to render full-resolution 4D LF and outperformed the state-of-the-art LF deblurring works in terms of deblurring quality, larger spatio-angular resolution, and faster runtime.

REFERENCES

- [1] Y. Bok, H.-G. Jeon, and I. S. Kweon, "Geometric calibration of micro-lens-based light field cameras using line features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 287–300, Feb. 2017.
- [2] D. Cho, M. Lee, S. Kim and Y.-W. Tai, "Modelling the calibration pipeline of the Lytro camera for high quality light field image reconstruction," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 3280–3287.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
- [4] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Grap.*, vol. 35, no. 6, 2016, Art. no. 193.
- [5] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 233–240.
- [6] T. H. Kim, K. M. Lee, B. Scholkopf, and M. Hirsch, "Online video deblurring via dynamic temporal blending network," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 4058–4067.
- [7] W. S. Lai, J. B. Huang, Z. Hu, N. Ahuja, and M. H. Yang, "A comparative study for single image blind deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1701–1709.
- [8] D. Lee, H. Park, I. K. Park, and K. M. Lee, "Joint blind motion deblurring and depth estimation of light field," in *Proc. Eur. Conf. Comput. Vision*, 2018.
- [9] M. Mahesh Mohan and A. N. Rajagopalan, "Divide and conquer for full-resolution light field deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 6421–6429.
- [10] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 257–265.
- [11] R. Ng, "Fourier slice photography," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 735–744, 2005.
- [12] J. Pan, Z. Hu, Z. Su, and M. H. Yang, "Deblurring text images via L0-regularized intensity and gradient prior," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 2901–2908.
- [13] J. Pan, D. Sun, H. Pfister, and M. H. Yang, "Blind image deblurring using dark channel prior," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1628–1636.
- [14] W. Qiu, F. Zhong, Y. Zhang, S. Qiao, Z. Xiao, T. S. Kim, and Y. Wang, "UnrealCV: Virtual worlds for computer vision," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 1221–1224.
- [15] P. P. Srinivasan, R. Ng, and R. Ramamoorthi, "Light field blind motion deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2262–2270.
- [16] D. Ulyanov, A. Veldadi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv preprint arXiv:1607.08022*.
- [17] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 157–170.
- [18] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 168–186.
- [19] Williem, I. K. Park and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2484–2497, Oct. 2018.