 Copyright © 2021 by ITE Transactions on Media Technology and Applications (MTA)

Paper

CFA Handling and Quality Analysis for Compressive Light Field Camera

Kohei Sakai †, Yasutaka Inagaki †, Keita Takahashi (member)†,
Toshiaki Fujii (member)†, Hajime Nagahara (member)††

Abstract A light field can carry rich visual information of a real 3-D scene, leading to many attractive applications. However, the acquisition of a light field is challenging due to the large amount of data. In our previous work, we proposed an efficient method for this task using a coded-aperture camera with a convolutional neural network (CNN) which can computationally reconstruct a light field from several images acquired with different aperture patterns. In this work, we report two follow-up contributions to the previous work. First, we integrated a color filter array, which is common in RGB cameras, and the related color processing into the algorithm pipeline. This integration led to better reconstruction quality for color light fields. We then analyzed how the reconstruction quality obtained with our method was affected by the complexity of light fields. We also showed the possibility of using this analysis to predict the reconstruction quality from the acquired images.

Key words: Light field, Computational photography, Convolutional neural network

1. Introduction

A light field can carry rich visual information of a real 3-D scene by describing the light rays traveling in 3-D free space [1–3]. The light field representation has been utilized in various applications, such as view synthesis [4–6], depth estimation [7–9], synthetic refocusing [10,11], super resolution [7,12], 3D displays [13–16], and object recognition [17, 18]. A light field is usually represented as a set of multi-view images that are aligned densely with tiny viewpoint intervals.

Acquiring a light field is challenging due to the large amount of data required because a light field typically consists of dozens of images. Many works have researched this task, including ones to capture all individual images separately [19–21] or on a single image sensor collectively [22–24]. Other works have applied the idea of compressive sensing in which an entire light field is computationally reconstructed from a smaller number of coded observations [25–30]. In our previous work [31], we focused on compressive acquisition using a coded aperture camera, in which a programmable coded-aperture pattern was inserted into the optical path [32] (see Fig. 1). We demonstrated that using this camera with a well-trained neural network enables the entire light field (consisting of 5×5 or 8×8 images) to be reconstructed from only a few acquired images, which is a state-of-the-art performance at the time of this writing.

In this paper, we report two follow-up contributions to the previous work [31]. First, we integrated a color filter array (CFA), which is common in RGB cameras, and the related color processing into the algorithm pipeline. The entire pipeline is implemented as a fully convolutional neural network and trained end-to-end. This integration led to better reconstruction quality for color light fields, with an improvement of up to 3 dB compared to the case with a naive Bayer demosaicing method. We then analyzed how the reconstruction quality obtained with our method was affected by the complexity of light fields. Extending this analysis, we also showed the possibility that the reconstruction quality of a light field can be predicted from only the images acquired by the coded-aperture camera. This prediction will be helpful when our method is applied to a real scene; we do not know the ground truth of the light field, but we can still estimate the quality of the reconstructed light field from the acquired images that are to be used for reconstruction.

2. Background

2.1 Light field and coded aperture camera

A light field is defined over a 4D space (s, t, u, v)

Received May 22, 2020; Revised August 14, 2020; Accepted September 7, 2020

† Graduate School of Engineering, Nagoya University (Furo-cho, Chikusa-ku, Nagoya 464-8603 Japan)
†† Institute for Datability Science, Osaka University (Yamadaoka, Suita, Osaka 565-0871 Japan)

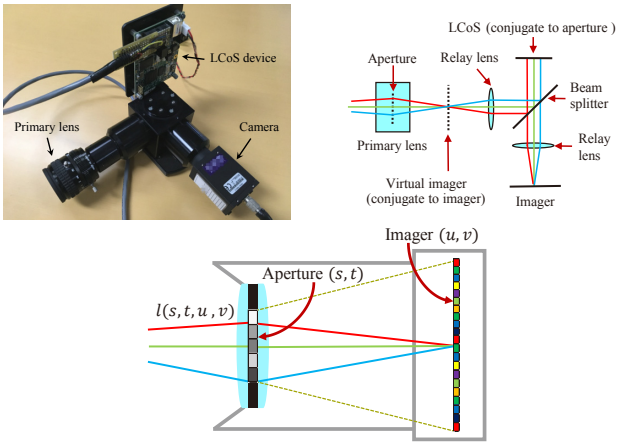


Fig. 1: Coded aperture camera. (Top) Appearance and optical configuration. (Bottom) Schematic diagram.

in which the intensity of a light ray is described as $l(s, t, u, v)$. When applied to a camera, (s, t) and (u, v) are defined as the intersections of a light ray with the aperture and imager planes, respectively, as shown in Fig. 1.

We assume that the 4-D space is discretized and all the parameters take integer values as $s \in [1, \ldots, S]$, $t \in [1, \ldots, T]$, $u \in [1, \ldots, U]$, and $v \in [1, \ldots, V]$. With this assumption, the light field is equivalently described as a set of rectified multi-view images called "sub-aperture images", i.e., $x_{s,t}(u, v) = l(s, t, u, v)$. Here, (s, t) corresponds to the viewpoint defined on the aperture plane, and the total number of images is given as $M = ST$.

In the case of a coded aperture camera, a semi-transparent mask is located at the aperture plane to modulate the incoming light rays. The mask pattern can be changed for each acquisition, and the pattern for the n-the acquisition is denoted as $a_n(s, t)$ $(n \in [1, \ldots, N])$. The n-th observed image, $y_n(u, v)$, is given as

$$y_n(u, v) = \sum_{(s,t)} a_n(s, t) x_{s,t}(u, v), \qquad (1)$$

which is interpreted as a weighted sum of all the images with the weight $a_n(s, t)$ for the viewpoint (s, t).

Reconstructing a light field is equivalent to obtaining all the sub-aperture images $\hat{x}_{s,t}(u, v)$ from given observations $y_n(u, v)$ $(n \in [1, \ldots, N])$, where $\hat{x}_{s,t}(u, v)$ is an estimation of $x_{s,t}(u, v)$. In particular, we are interested in the case of $N \ll M$, where the entire light field can be reconstructed from only a few observed images.

2.2 CNN-based Compressive Acquisition

In our previous work [31], the entire pipeline from observation to reconstruction was regarded as an auto-encoder, where the original M images were first reduced (encoded) to N observed images and then expanded (decoded) to M images again. The encoder part (acquisition of images) and decoder part (reconstruction of a light field) of the network are dubbed as $\mathcal{N}_A$ and $\mathcal{N}_R$, respectively.

This auto-encoder was implemented as a fully convolutional neural network, where 2-D features stacked along the channel dimensions are sequentially processed by 2-D convolution layers. More specifically, the input to (and output from) the network was a set of M images stacked along the channel dimension. The intermediate representation (between the encoder and decoder) was also a set of 2-D features with N channels. The spatial size (height and width) of the features was kept constant throughout the network, and only the number of channels was changed as the data proceeded in the network.

The network was trained end-to-end on a collection of light field datasets. In the training stage, all the computations were conducted on the computer. However, when applied to a physical camera, the encoder part ($\mathcal{N}_A$) was conducted by the physical imaging process of the camera, the aperture of which was set to the patterns learned as the parameters of $\mathcal{N}_A$, and only the decoder part ($\mathcal{N}_R$) was conducted on the computer to reconstruct the target light field from the observed images.

As reported in [31], quantitative evaluations showed that this method can reconstruct an entire light field (consisting of 5×5 or 8×8 images) with high quality from only a few (2–4) acquired images. This method was also applied to a physical coded-aperture camera to validate its effectiveness for a real scene.

2.3 Issues

Despite the excellent performance of our previous work [31], we feel it still has two major drawbacks.

First, the method was constructed in a gray-scale world. When it is applied to a color light field, three RGB color channels have to be processed individually. However, individual handling of color channels is obviously incompatible with the imaging model using a color filter array (CFA), where RGB color filters are interleaved on the imager and each pixel can capture only one of the color components. In [31], we simply assumed that the camera could capture three color channels with each pixel, but in practice we applied a naive demosaicing method for the raw observations from the camera to

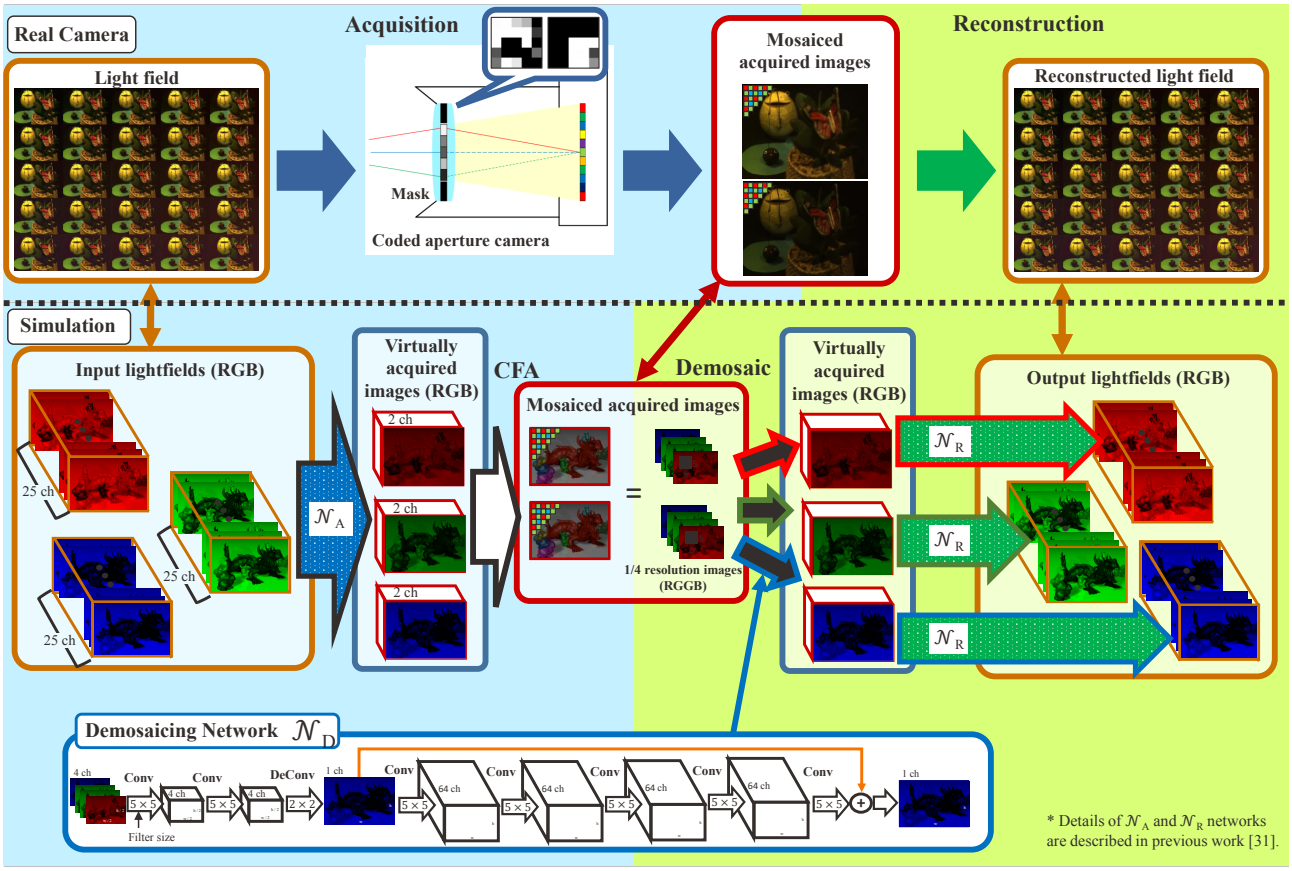


Fig. 2: Our CNN model extended to handle CFA.

obtain the images $y_n(u, v)$ with full color. As a remedy for this drawback , we integrate these mosaicing (color filtering) and demosaicing processes, which obviously have negative effects on the reconstruction quality, into the algorithm pipeline in Section 3. The demosaicing process is also implemented as a convolutional neural network and is optimized jointly with the image acquisition and light field reconstruction processes.

Second, the reconstruction quality obtained with this method depended largely on the content of the target scene, but no analysis was presented on this point in [31]. It would be helpful to determine some measure of complexity for the target light field to indicate the difficulty of reconstruction. More importantly, it is desirable to predict the reconstruction quality even in cases where the ground truth light field is not provided. These issues will be addressed in Section 4.

3. Handling Color Filter Array

In the remainder of this paper, we use $M = 25$ (5×5) and $N = 2$, meaning that 25 images should be reconstructed from only two observed images, which was a successful configuration in our previous work [31].

3.1 Method

We have extended the network of our previous work [31] to include color processing with a color filter array (CFA), as shown in Fig. 2. The newly introduced part is inserted between the image acquisition network ($\mathcal{N}_A$) and the light field reconstruction network ($\mathcal{N}_R$). The image acquisition network ($\mathcal{N}_A$) is shared among all the colors because our camera supports only gray-scale transmittance. However, for the reconstruction network ($\mathcal{N}_R$), we used three distinct networks (with the same architecture) for each of the three color components, as the sampling patterns are different color to color in accordance with the Bayer arrangement.

We now describe the details of the newly introduced part. The input for this part is two full-color images with $U \times V$ pixels generated by $\mathcal{N}_A$. These images are called "virtually" acquired images, as they cannot be acquired in practice with a CFA imager. They undergo a color filtering operation in accordance with the Bayer arrangement, resulting in two mosaiced feature maps with $U \times V$ pixels, which correspond to the raw observations from a CFA camera. Each of the mosaiced feature maps is reorganized into four small feature maps with $U/2 \times V/2$ pixels, which correspond to the R, G,

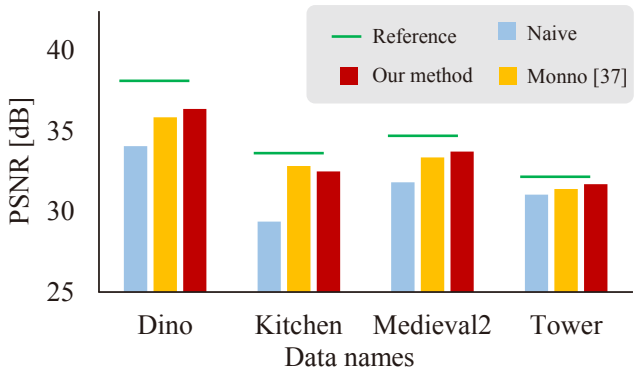


Fig. 3: Performance comparison on CFA handling.

G, and B color components, respectively. From these reduced feature maps, we reconstruct the virtually acquired images that have all three color values in the original resolution ($U \times V$ pixels).

Reconstruction of the virtually acquired images corresponds to the demosaicing operation in the ordinary imaging pipeline. We implemented this process as a convolutional neural network dubbed as a demosaicing network, or $\mathcal{N}_D$, the details of which are illustrated in the bottom of the figure. The input to $\mathcal{N}_D$ is a set of reduced feature maps (RGGB) taken from each acquisition. In the network, the input was first processed by two convolutional layers (kernel size: 5×5, stride: 1, padding: 2), followed by a deconvolution layer (kernel size: 2×2, stride: 2, padding: 0) to obtain a feature map in the original resolution. This feature map was further refined by the following four convolution layers (kernel size: $5\times$ 5, stride: 2, padding: 2) with a residual connection. The output from $\mathcal{N}_D$ was supervised by one of the color channels of the virtually acquired images. We use three distinct networks with the same architecture for each of the three colors. Finally, the reconstructed images, grouped by color, were passed to the reconstruction networks $\mathcal{N}_R$.

3.2 Implementation

Our method was implemented using Chainer version 3.2.0, a Python-based framework for neural networks. The batch size for training was set to 15. We used a built-in Adam optimizer. The training of the network was conducted step-by-step. First, we individually pre-trained two networks, $\mathcal{N}_A + \mathcal{N}_R$ and $\mathcal{N}_D$, each with 20 epochs. Then, we connected all the network components and fine-tuned the entire network in an end-to-end manner with 20 epochs.

As the training dataset, we collected light field samples from [33–36] following our previous work [31].

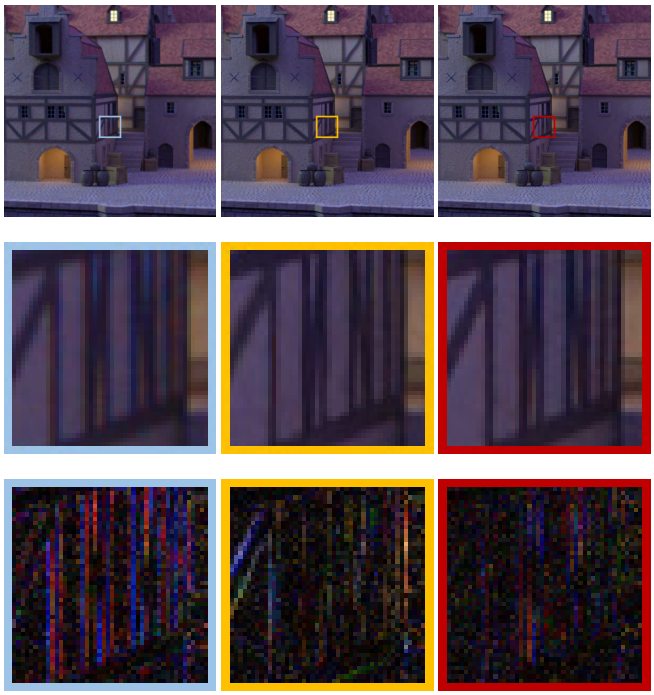
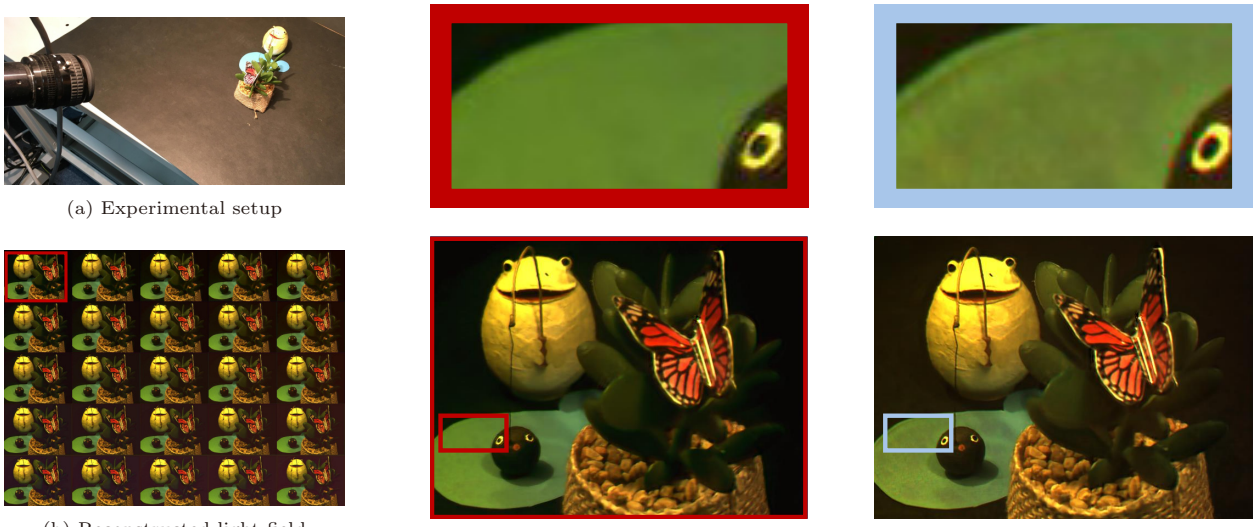


Fig. 4: Top: reconstructed images (at top-left viewpoint in light field) using Naive method (left), Monno et al.'s method [37] (center), and our method (right). Middle: close-ups of squared regions. Bottom: differences from ground truth for close-ups (intensities are magnified by 5 for visualization), where darker is better.

Each light field sample is a set of 2-D image blocks with 64×64 pixels, where 25 image blocks (corresponding to 5×5 viewpoints) are stacked along the channel dimension. With data augmentation in the intensity level, we gathered 98,400 samples, each of which has three color components. The three color components were treated as individual samples for the pre-training stage for $\mathcal{N}_A$+$\mathcal{N}_R$. Only the central views were used for the pre-training of $\mathcal{N}_D$.

3.3 Results

We evaluated our method using the Dino, Kitchen, Medieval2, and Tower datasets, which were not included in the training samples. We compared our method against three cases as follows. "Reference" corresponds to the ideal case where no color filtering was applied. In this case, three color components were individually processed by the pipeline $\mathcal{N}_A + \mathcal{N}_R$. "Naive" and "Monno" correspond to the cases where demosaicing operations using linear interpolation and Monno et al.'s method [37] were applied to the raw observation from a CFA camera, respectively. In those cases, the output from $\mathcal{N}_A$ underwent mosaicing (color filtering) and demosaicing operations before $\mathcal{N}_R$. We used the pre-trained $\mathcal{N}_R$ and the three color components were



(a) Experimental setup (b) Reconstructed light field (c) Proposed method (d) Naive method

Fig. 5: Experiment using physical coded-aperture camera.

also individually processed. Note that Monno et al. [37] is one of the state-of-the-art methods for demosaicing, which yields much better quality than the naive linear interpolation.

The quantitative and visual results are summarized in Figs. 3 and 4. The reconstruction quality was measured by PSNR against the ground truth, which were obtained from the squared errors averaged over all the viewpoints and color channels. The results indicate that our method achieved clearly better quality than the "Naive"; the improvement over "Naive" was up to 3 dB. Our method also achieved slightly better quality than "Monno" in average, but these two methods are competing in the reconstruction quality. However, our method was 2000 times faster than "Monno"; our method took about 0.01 secs for demosaicing, while "Monno" about 20 secs using the Matlab code provided by the authors. Note that the fast computation of our method is partly attributed to the sophisticated GPU-based framework of deep neural networks. The availability of such a framework is also an advantage that a method implemented as a neural network can inherently enjoy.

We also conducted an experiment using a physical coded-aperture camera. The experimental setup and reconstructed light fields are presented in Fig. 5. It is clearly seen that our method had fewer artifacts than the "Naive". Please refer to the supplementary video for more details.

4. Analyzing Reconstruction Quality

4.1 Measuring Light Field Complexity

The reconstruction quality obtained with a compressive light field camera depends largely on the content of the target scene. Obviously, the resulting quality is related to the complexity of the target scene. However, it is difficult to give a specific definition for this difficulty because several factors—depth variation, texture, occlusions, and surface reflectance—are involved. Taking these into consideration, we first tried a simple method that uses the difference between the two images in the original light field as a measure for the complexity. We refer to this measure as "difference in LF". If the target scene is locally approximated by a single surface with a constant disparity d, the difference between the viewpoints (s,t) and $(s+\Delta_s, u+\Delta_u)$ can be written as

$$\begin{aligned} &x_{s,t}(u,v) - x_{s+\Delta_s,t+\Delta_t}(u,v) \\ &= x_{s,t}(u,v) - x_{s,t}(u-d\Delta_s, v-d\Delta_t). \end{aligned} \tag{2}$$

This is approximated by applying Taylor series expansion for the second term, as

$$\begin{aligned} &x_{s,t}(u,v) - x_{s,t}(u-d\Delta_s, v-d\Delta_t) \\ &\simeq \left\{ \Delta_s \frac{\delta x_{s,t}(u,v)}{\delta u} + \Delta_t \frac{\delta x_{s,t}(u,v)}{\delta v} \right\} d \end{aligned} \tag{3}$$

which involves both the texture complexity (horizontal and vertical edges in the spatial dimension) and the disparity. Specifically, we took the absolute difference between the top-left and bottom-right viewpoint images and averaged the differences over all pixels and color channels to obtain a measure of complexity for a target

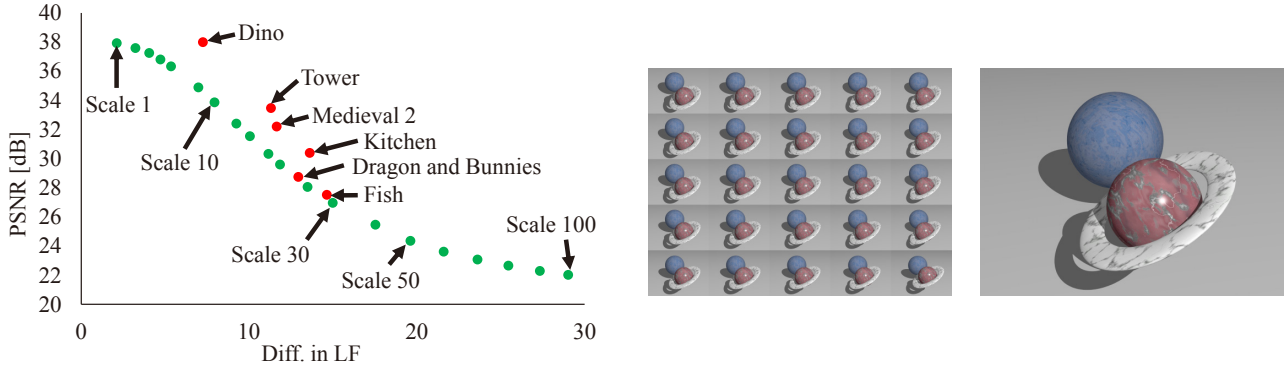


Fig. 6: Reconstruction quality against difference in LF (left) and computer generated light field scene for green plots (right). Difference in LF is represented in scale of 8-bits pixel intensity values (0 – 255).

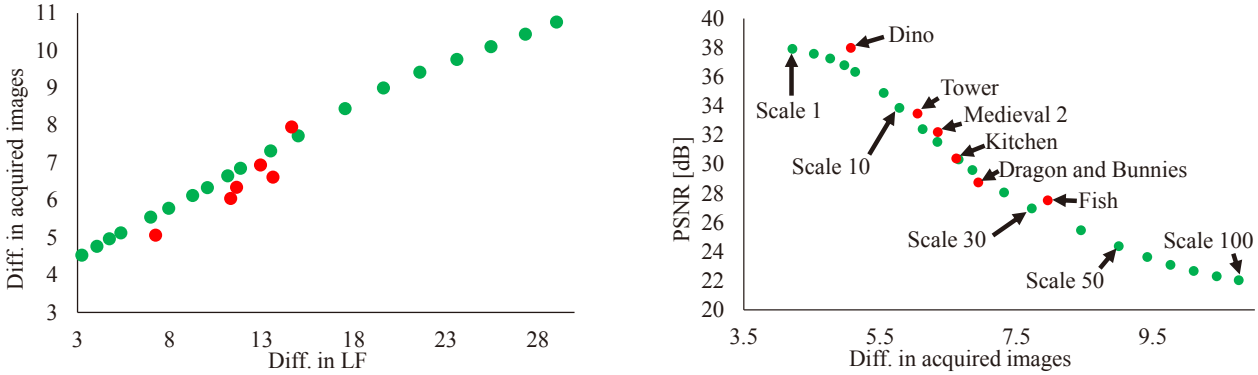


Fig. 7: Relation between the difference in acquired images and difference in LF (left), which shows close correlations between the two measures. Differences in LF and acquired images are represented in scale of 8-bits pixel intensity values (0 – 255). The reconstruction quality is plotted against the difference in acquired images (right), from which we can regress the reconstruction quality for a given difference in acquired images.

light field.

In Fig. 6 (left), we plot the reconstruction quality (PSNR) against the difference in LF. The plots in red correspond to the datasets that were not included in the training samples. Additionally, we synthesized a light field from a computer generated (CG) scene as shown in Fig. 6 (right), whose viewpoint interval (controlled by the scale factor) can be changed freely; a larger interval leads to a more difficult situation for light field reconstruction. The green plots in the graph were generated from this CG scene. The graph on the left shows that the difference in LF is a good indicator to predict the reconstruction quality of light fields: namely, a larger value for the difference in LF tends to indicate a lower reconstruction quality.

4.2 Toward Practical Quality Indicator for Real Scene

The complexity measurement discussed above comes with the ground truth light field, which is not available for a real scene. Considering this limitation, we next tried to use the difference between the two images* acquired by a coded aperture camera as an indicator for the reconstruction quality. This new indicator is dubbed as the "difference in acquired images". Once the network has been trained, two mask patterns for the aperture are fixed. As a result of optimization, these mask patterns are complementary to each other, so the 3-D information can be best embedded into the two acquired images. Therefore, these images generally have some difference from which the reconstruction network can deduce the underlying 3-D scene information. We felt that this difference could also be used as an indicator for the reconstruction quality. Before subtracting two images, the intensity ranges should be normalized; we did this by dividing the intensity values by the corresponding aperture ratio ($\sum_{s,t} a_n(s,t) / \sum_{s,t} 1^{**}$). The absolute difference between the two normalized images is written as

* Our discussion here is limited to $N = 2$, but applicable to any M. Extension to arbitrary N is left for future work.

** $\sum_{s,t} 1 = ST = M$

$$\left| \frac{\sum_{s,t} 1}{\sum_{s,t} a_0(s,t)} y_0(u,v) - \frac{\sum_{s,t} 1}{\sum_{s,t} a_1(s,t)} y_1(u,v) \right|, \quad (4)$$

where $y_0(u,v)$ and $y_1(u,v)$ are the first and second acquired images in the raw format (mosaiced acquired images), respectively. We averaged Eq. (4) over all the pixels to obtain the indicator value.

We evaluated this revised indicator for predicting the reconstruction quality of light fields using the same datasets as Fig. 6. The new indicator is closely correlated to the old one, as shown in Fig. 7 (left). Moreover, this indicator can be directly used to predict the reconstruction quality, as shown in Fig. 7 (right), which exhibits even better compaction around a single curve than Fig. 6 (left).

Although we cannot prove the accuracy of this quality estimation, we are confident that our findings here will pave the way to make compressive light field acquisition more convenient. We showed the possibility that the reconstruction quality of a light field could be predicted from only the two images acquired by a coded aperture camera. If we want to achieve a better reconstruction quality, we can make a small adjustment to the shooting condition. For example, slightly changing the zoom or focus parameters, or stepping a little further away from the target scene, will affect the indicator value, from which we can predict the resulting reconstruction quality.

5. Conclusion

In this paper, we reported two follow-up contributions to our previous work [31] on compressive light field acquisition using a coded-aperture camera. We first integrated a color filter array (CFA), which is common in RGB cameras, and the related color processing into the algorithm pipeline. This integration led to better reconstruction quality for color light fields, with an improvement of up to 3 dB compared to the case with a naive Bayer demosaicing method. Our method was also applied to a physical coded aperture camera. We then analyzed the relation between the reconstruction quality obtained with our method and the complexity of light fields. Extending this analysis, we showed the possibility that the reconstruction quality of a light field could be predicted from only the images acquired by the coded-aperture camera.

Our future work will take several directions. First, we only considered a fixed color filter (Bayer arrangement) in this paper. We intend to jointly optimize the color filtering operation with the mask pattern, demosaicing, and light field reconstruction. Moreover, the quality analysis presented in this paper is only the first step for the prediction of reconstruction quality. Deeper analyses on larger datasets will facilitate the prediction accuracy, which will lead to greater convenience for compressive light field acquisition. Finally, extending the framework of compressive light-field acquisition to dynamic scenes (scenes with motions) would be another interesting and fruitful topic.

6. Acknowledgment

This study was supported by JSPS Kakenhi under Grant 18K19810.

References

1) E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, 1991, pp. 3–20.
2) M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 31–42.
3) S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 43–54.
4) M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 67–76, 2011.
5) L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous fourier domain," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 12, 2014.
6) N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.
7) S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 606–619, 2014.
8) T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Depth estimation with occlusion modeling using light-field cameras," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2170–2181, 2016.
9) Williem, I. K. Park, and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.
10) A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 297–306.
11) R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.
12) T. E. Bishop, S. Zanetti, and P. Favaro, "Light field superresolution," in *Computational Photography (ICCP), 2009 IEEE International Conference on*. IEEE, 2009, pp. 1–9.
13) G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar, "Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 1–11, 2012.
14) F.-C. Huang, K. Chen, and G. Wetzstein, "The light field stereoscope: immersive computer graphics via factored near-eye light field displays with focus cues," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 60, 2015.
15) S. Lee, C. Jang, S. Moon, J. Cho, and B. Lee, "Additive light field

displays: realization of augmented reality with holographic optical elements," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 60, 2016.
16) T. Saito, Y. Kobayashi, K. Takahashi, and T. Fujii, "Displaying real-world light fields with stacked multiplicative layers: Requirement and data conversion for input multiview images," *Journal of Display Technology*, vol. 12, no. 11, pp. 1290–1300, 2016.
17) K. Maeno, H. Nagahara, A. Shimada, and R. I. Taniguchi, "Light field distortion feature for transparent object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2786–2793.
18) T.-C. W. andJun Yan Zhu, E. Hiroaki, M. Chandraker, A. Efros, and R. Ramamoorthi, "A 4D light-field dataset and CNN architectures for material recognition," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 121–138.
19) B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.
20) T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound-100-camera and microphone system," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 437–440.
21) Y. Taguchi, T. Koike, K. Takahashi, and T. Naemura, "TransCAIP: A live 3D TV system using a camera array and an integral photography display with interactive control of viewing parameters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 841–852, Sept 2009.
22) E. H. Adelson and J. Y. Wang, "Single lens stereo with a plenoptic camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 99–106, 1992.
23) J. Arai, F. Okano, H. Hoshino, and I. Yuyama, "Gradient-index lens-array method based on real-time integral photography for three-dimensional images," *Applied optics*, vol. 37, no. 11, pp. 2034–2045, 1998.
24) R. Ng, "Digital light field photography," Ph.D. dissertation, stanford university, 2006.
25) A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 69, 2007.
26) C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen, "Programmable aperture photography: multiplexed light field acquisition," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 55, 2008.
27) S. D. Babacan, R. Ansorge, M. Luessi, P. R. Mataran, R. Molina, and A. K. Katsaggelos, "Compressive light field sensing," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4746–4757, 2012.
28) K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 46, 2013.
29) S. Tambe, A. Veeraraghavan, and A. Agrawal, "Towards motion aware light field video for dynamic scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1009–1016.
30) Y. Yagi, K. Takahashi, T. Fujii, T. Sonoda, and H. Nagahara, "PCA-coded aperture for light field photography," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3031–3035.
31) Y. Inagaki, Y. Kobayashi, K. Takahashi, T. Fujii, and H. Nagahara, "Learning to capture light fields through a coded aperture camera," in *The European Conference on Computer Vision (ECCV)*, September 2018, pp. 431–448.
32) H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S. K. Nayar, "Programmable aperture camera using LCoS," in *European Conference on Computer Vision*. Springer, 2010, pp. 337–350.
33) MIT Media Lab's Camera Culture Group, "Compressive light field camera," http://cameraculture.media.mit.edu/projects/compressive-light-field-camera/.
34) Computer Graphics Laboratory, Stanford University, "The (new) stanford light field archive," 2018, http://lightfield.stanford.edu.
35) Heidelberg Collaboratory for Image Processing, "4D light field dataset," 2018, http://hci-lightfield.iwr.uni-heidelberg.de/.
36) ——, "Datasets and benchmarks for densely sampled 4D light fields," http://lightfieldgroup.iwr.uni-heidelberg.de/?page_id=713, 2016.
37) Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color and multispectral image demosaicking," *Sensors*, vol. 17, no. 12, p. 2787, 2017.



Kohei Sakai received a B.E. degree in electrical engineering from Nagoya University, Japan, in 2019. He is currently a student at the Graduate School of Engineering, Nagoya University, Japan. His research topics are light-field acquisition and its applications.



Yasutaka Inagaki received a B.E. degree in electrical engineering from Nagoya University, Japan, in 2018, and his M.S. degree Graduate School of Engineering, Nagoya University, Japan, in 2020. His research topics are light-field acquisition and its applications.



Keita Takahashi received B.E., M.S., and Ph.D. degrees in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2001, 2003, and 2006. He was a project assistant professor at the University of Tokyo from 2006 to 2011 and was an assistant professor at the University of Electro-Communications from 2011 to 2013. He is currently an associate professor at the Graduate School of Engineering, Nagoya University, Nagoya, Japan. His research interests include computational photography, image-based rendering, and 3D displays.



Toshiaki Fujii received B.E., M.E., and Dr.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1990, 1992, and 1995. In 1995, he joined the Graduate School of Engineering, Nagoya University, where he is currently a professor. From 2008 to 2010, he was with the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multi-dimensional signal processing, multi-camera systems, multi-view video coding and transmission, free-viewpoint television, and their applications.



Hajime Nagahara has been a professor at the Institute for Datability Science, Osaka University, since 2017. He received a Ph.D. in system engineering from Osaka University in 2001. He was a research associate of the Japan Society for the Promotion of Science from 2001 to 2003. He was an assistant professor at the Graduate School of Engineering Science, Osaka University, Japan from 2003 to 2010. He was an associate professor in the faculty of information science and electrical engineering at Kyushu University from 2010 to 2017. He was a visiting associate professor at CREA University of Picardie Jules Verns, France, in 2005. He was a visiting researcher at Columbia University in 2007–2008 and 2016–2017. His research interests are computational photography and computer vision. He received an ACM VRST2003 Honorable Mention Award in 2003, IPSJ Nagao Special Researcher Award in 2012, ICCP2016 Best Paper Runner-up Award, and SSII Takagi Award in 2016.