Binary and Rotational Coded-Aperture Imaging for Dynamic Light Fields

Kohei SAKAI†(a), Member, Keita TAKAHASHI†, Senior Member, and Toshiaki FUJII†, Fellow

SUMMARY  Coded-aperture imaging has been utilized for compressive light field acquisition; several images are captured using different aperture patterns, and from those images, an entire light field is computationally reconstructed. This method has been extended to dynamic light fields (moving scenes). However, this method assumed that the patterns were gray-valued and of arbitrary shapes. Implementation of such patterns required a special device such as a liquid crystal on silicon (LCoS) display, which made the imaging system costly and prone to noise. To address this problem, we propose the use of a binary aperture pattern rotating along time, which can be implemented with a rotating plate with a hole. We demonstrate that although using such a pattern limits the design space, our method can still achieve a high reconstruction quality comparable to the original method.

key words: light field, CNN, coded aperture camera

1. Introduction

A light field, which is represented as a set of dense multi-view images, has been used in various applications, such as view synthesis, depth estimation, and 3-D displays. Acquiring a light field is challenging due to the large amount of data. A promising approach for efficiently acquiring a light field is to use coded-aperture imaging [1]–[5]. This approach has a potential advantage in efficiency because with it, an entire light field can be reconstructed from only a few images captured with different aperture patterns. Further, the acquired light field has the full spatial resolution of the image sensor, in contrast with the commercialized light field cameras [6], [7] that have an inherent trade-off between the angular (viewpoint) and spatial (pixel) resolutions.

Inagaki et al. [3] modeled the entire process of coded-aperture imaging using a deep convolutional neural network (CNN), and jointly optimized the aperture patterns and reconstruction algorithm. Sakai et al. [4] extended this model to dynamic light fields (moving scenes) and proposed “V-shape observation,” in which two aperture patterns were alternately repeated along time. In this method, the aperture patterns were gray-valued and of arbitrary shapes. Implementation of such patterns required a special device such as a liquid crystal on silicon (LCoS) display, which made the imaging system costly and prone to noise.

In this paper, we propose the use of a binary aperture pattern rotating along time for acquiring dynamic light fields. Such a pattern can be implemented using a rotating plate with a hole, which would make the imaging system less expensive and less prone to noise. Using such a pattern limits the design space compared to that of the conventional gray-valued arbitrary patterns. However, our method can still achieve a high reconstruction quality comparable to that of the original method [4], which we demonstrate through a simulation experiment.

Binary representations are utilized for some light field applications, e.g. feature description for recognition tasks [8] and data compression [9]. Meanwhile, we apply binarization to an imaging mechanism through which a light field is captured.

2. Background

2.1 Light Field and Coded Aperture Camera

We follow the notation and problem formulation described in Sakai et al. [4]. A schematic diagram of a coded aperture camera is shown in Fig. 1 (left). All light rays incoming to the camera are parameterized with four variables \((u, v, x, y)\), where \((u, v)\) and \((x, y)\) denote the intersections with the aperture and imaging planes, respectively. The light field is defined over 4-D space \((u, v, x, y)\), with which the light intensity is described as \(I(u, v, x, y)\). When we consider scene motions over time \(t\), the light intensity is described as \(I(u, v, x, y, t)\) on 5-D space.

We consider a coded aperture design where the transmittance of the aperture can be controlled at any position and time. The transmittance at position \((u, v)\) and time \(t\) is defined as \(a(u, v, t)\). Here, we assume that the light field and aperture pattern are constant during each exposure time and the aperture plane is discretized into square blocks indexed by a pair of integers \((u, v)\), i.e., \(I(u, v, x, y, \tau) = I_{u,v}(x, y)\)
and \(a(u, v, \tau) = a(u, v)\) for \(\tau \in E_t\), where \(E_t\) is the exposure time around \(t\). The image-formation process through a coded aperture camera is described as
Reconstruction of dynamic light field with “V-shape observation”

\[ i_t(x, y) = \sum_{u, v} a_{t}(u, v) I_{u,v}(x, y), \]

(1)

where \( i_t(x, y) \) is the observed image at \( t \). Figure 1 (right) illustrates a case in which the aperture plane was discretized in \( 5 \times 5 \) regions; thus, a light field at each \( t \) is represented as \( 5 \times 5 \) multi-view images. The observed image given by Eq. (1) is a weighted sum of those multi-view images.

Given the model of Eq. (1), our goal is to reconstruct the original light field at each \( t \), \( I_{u,v}(x, y) \), from several observed images around \( t \): \( I_t(x, y) \) for \( t \in \{ ..., t-1, t, t+1, ... \} \). The aperture patterns and reconstruction algorithm were jointly optimized under the framework of deep learning.

### 2.2 Reconstruction of Dynamic Light Field

If the target scene is stationary, the same scene can be observed several times by using different aperture patterns over time, from which we can easily deduce disparity information (related to the 3-D structure) and thus reconstruct the light field of the target scene. If the target scene is dynamic, though, both the aperture patterns and the light field itself can vary along time. To handle this situation, Sakai et al. [4] proposed “V-shape observation,” illustrated in Fig. 2. In this figure, a light field at time \( t \) is denoted as \( L_t \), and an image observed through aperture pattern \( A \) at \( t \) is denoted as \( I^A_t \). Two aperture patterns, \( A \) and \( A^R \), are alternately repeated over time. Sakai et al. [4] proposed the use of three consecutive images, \( I^A_{t-1}, I^A_t, \) and \( I^A_{t+1} \), to reconstruct \( L_t \), in the hope that using these three images would help the CNN successfully distinguish the 3-D structure from motion information, which is explained as follows. Images \( I^A_{t-1} \) and \( I^A_{t+1} \) are captured with the same aperture pattern, i.e., \( A \), so that the difference between these images is exclusively attributed to scene motions. Meanwhile, image \( I^R_t \) contains both disparity (related to the 3-D structure) and motion information with respect to the other two images. Sakai et al. [4] verified that this approach worked well for reconstructing a dynamic light field, showing accurate results for real scenes obtained using an LCoS-based imaging system.

### 3. Proposed Method

Our method is constructed on Sakai et al. [4]’s method with constraints on the aperture pattern. Instead of using gray-valued arbitrarily-shaped patterns, we use a single binary-valued pattern that is rotated along time. Our design eliminates the need for a costly and noise-prone imaging system that uses an LCoS display. For simplicity, we assume that the target light field consists of \( 5 \times 5 \) viewpoints, but our discussion here can easily be extended to different configurations.

#### 3.1 Binary and Rotational Coded-Aperture

Whereas two distinct aperture patterns \( A \) and \( B \) were used in the previous work [4], we use only a single aperture pattern \( A \) but rotate it by 180 degrees along time. We have, in effect, two aperture patterns \( A \) and \( A^R \) (rotated \( A \) by 180 degrees), but they are related to each other. In addition, we assume that each of the \( 5 \times 5 \) aperture regions can take a value of either 0 (closed) or 1 (opened), to make the aperture pattern physically implementable by punching a hole in a plate. Note that these constraints limit the design space of the original method [4]. We use three consecutive images \( I^A_{t-1}, I^A_t, \) and \( I^A_{t+1} \) to reconstruct \( L_t \), following the idea of “V-shape observation” [4].

#### 3.2 Network Architecture and Training

Our network architecture, illustrated in Fig. 3, is essentially the same as that of the original method [4], except for the constraints imposed on the aperture pattern.

To summarize, the entire process is regarded as an auto-encoder in which image acquisition and light field reconstruction correspond to the encoder and decoder (denoted as \( N_A \) and \( N_R \)), respectively. The network consists of 2-D convolutional layers, where the image size (height and width) is kept constant throughout the network but only the number of channels is changed as the data proceed in the network. An instance of a light field at \( t \) \( (L_t) \) is treated as a 2-D image having multiple channels, i.e., a light field with \( 5 \times 5 \) viewpoints is translated into an image with 25 channels\(^1\). At each time \( t \), the light field \( L_t \) is compressed into a single acquired image \( I_t \) through the encoder network. Then, from the three acquired images \( I^A_{t-1}, I^A_t, \) and \( I^A_{t+1} \), a light field \( L_t \) is reconstructed through the decoder network.

The encoder \( N_A \) is implemented using a single 2-D convolutional layer that has a \( 1 \times 1 \) convolution kernel and reduces the number of channels from 25 to 1. The weights of this layer correspond to the transmittance of the aperture pattern \( A \). The rotated pattern \( A^R \) is implemented by rearranging the weights of \( A \) or, equivalently, by rearranging the order of channels of the input light field. In the initial stage of the training, we enable the values for the aperture pattern to be continuous (grayscale) and train \( N_A \) and \( N_R \) simultaneously. Here, the values for the aperture pattern are restricted

\(^1\)We assume that a light field has only one color channel; RGB color channels of a color light field are processed as three individual light fields.
to be within the range [0, 1] because the values should represent transmittance. Then, we binarize the aperture pattern using 0.5 as the threshold. Finally, we fix the weights of the encoder $N_A$ (equivalently, the aperture pattern) and further train the decoder $N_R$ to make it suited for the binarized aperture pattern.

### 4. Experiment

#### 4.1 Experimental Condition

Our method imposes two constraints to the aperture pattern: it should be binary and rotational. To evaluate the effect of these constraints, we considered four scenarios as an ablation study. (a) Standard: the aperture patterns ($A$ and $B$) can be gray-valued (but should be within the range of $[0, 1]$). (b) Binary: the aperture patterns should be binary-valued. (c) Rotational: the aperture patterns can be gray-valued, but $B$ should be the rotation of $A$ by 180 degrees ($B = A^*$). (d) Proposed: the aperture patterns should be binary-valued, and $B$ should be $A^*$. As summarized in Table 1, these four scenarios were implemented through the two-stages training procedure. We trained the networks for 20 epochs in the initial stage, and 10 epochs for the final stage, imposing appropriate restrictions in each stage.

We used the same training dataset as Sakai et al. [4].

|                          | Initial stage | Final stage |
|--------------------------|---------------|-------------|
| Sakai et al. [4]         | (i)           | –           |
| Standard                 | (i)           | (i)         |
| Binary                   | (i)           | (ii)        |
| Rotational               | (i) + (iii)   | (i) + (iii) |
| Proposed                 | (i) + (iii)   | (i) + (iii) |

Light field patches with $64 \times 64$ pixels and $5 \times 5$ views were collected from several datasets [10]–[13]. Then, by giving various pseudo-motions to the collected patches, a dynamic light field dataset was constructed. The batch size for training was set to 15. We used a built-in Adam optimizer. To simulate practical imaging conditions, we added zero-means Gaussian noise to the observed image $I_t$, whose standard deviation was $\sigma = 0.005$ with respect to the image-intensity range $[0, 1]$ of $I_t$.

For quantitative evaluation, we used the Rotation-Planets dataset provided by Sakai et al. [14], which is a computer-generated dynamic-light-field sequence with $840 \times 593$ pixels and $5 \times 5$ views over 200 temporal frames. We applied the same amount of noise to the observed images as was applied in the training stage. For the purpose of comparison, we included the original results from Sakai et al. [4], which had the same restriction as (a) but differ-
ent training conditions. We also included two other methods as the baselines. Guo et al. [5]’s method is the latest compressive acquisition method for static light fields, and it can reconstruct a light field with $5 \times 5$ views from a single coded image. “Lytro-like camera” is a simulation of a lens-array based camera [6], [7] that obtains $5 \times 5 \times 1$ coded image. “Lytro-like camera” is a simulation of a lens-array based camera that obtains $5 \times 5 \times 1$ coded image. “Lytro-like camera” is a simulation of a lens-array based camera that obtains $5 \times 5 \times 1$ coded image. “Lytro-like camera” is a simulation of a lens-array based camera that obtains $5 \times 5 \times 1$ coded image. “Lytro-like camera” is a simulation of a lens-array based camera that obtains $5 \times 5 \times 1$ coded image.

Figure 4 shows quantitative scores (peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) averaged over 25 views) for the reconstructed light fields along time. Figure 5 shows the reconstructed top-left views at the 81-st frame, for each of which an epipolar plane image (EPI) and the difference from the ground truth (magnified by 3 for visualization) were also presented.

We observe that the standard case was clearly better than the Sakai et al. [4]’s original due to the increased number of epochs for training. Interestingly, the binary constraint had little negative effect on the reconstruction quality, while the rotational constraint obviously resulted in lower reconstruction quality than the standard case. Our method, which came with both binary and rotation constraints, shows further degradation than in the rotational case. Note again that using a binary and rotational pattern limits the design space compared to that of the conventional gray-valued arbitrary patterns. However, our method still maintained reasonable reconstruction quality that was even comparable to the original results of Sakai et al. [4]. Further, our method outperformed Guo et al. [5] and Lytro-like camera with significant margins.

5. Conclusion

We proposed the use of a binary aperture pattern rotating along time for compressively acquiring a dynamic light field. Such a pattern can be implemented using a rotating plate with a hole, which would eliminate the need for costly and noise-prone imaging system that use a special device, such as an LCoS display. Despite the constraints on the aperture pattern, our proposed method still achieved high reconstruction quality comparable to the original method with fewer constraints [4]; in terms of the quantitative scores (PSNR and SSIM), the difference between our method and the original method [4] was limited in a small margin. Our future work will include the development of hardware for an imaging system with a binary and rotational aperture pattern.

References

[1] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H.H. Chen, “Programmable aperture photography: multiplexed light field acquisition,” ACM Trans. Graphics, vol.27, no.3, p.55, 2008.
[2] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S.K. Nayar, “Programmable aperture camera using LCoS,” European Conf. on Computer Vision, vol.6316, pp.337–350, 2010.
[3] Y. Inagaki, Y. Kobayashi, K. Takahashi, T. Fujii, and H. Nagahara, “Learning to capture light fields through a coded aperture camera,” European Conf. on Computer Vision, vol.11211, pp.418–434, 2018.
[4] K. Sakai, K. Takahashi, T. Fujii, and H. Nagahara, “Acquiring dynamic light fields through coded aperture camera,” European Conf. on Computer Vision, vol.12364, pp.368–385, 2020.
[5] M. Guo, J. Hou, J. Jin, J. Chen, and L.-P. Chau, “Deep spatial-angular regularization for compressive light field reconstruction over coded apertures,” European Conf. on Computer Vision, vol.12347, pp.278–294, 2020.
[6] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Computer Science Tech Report CSTR, vol.2, no.11, pp.1–11, 2005.
[7] R. Ng. “Digital light field photography,” Ph.D. thesis, Stanford University, 2006.
[8] A. Sepas-Moghaddam, P.L. Correia, and F. Pereira, “Light field local binary patterns description for face recognition,” IEEE International Conf. on Image Processing, pp.3815–3819, 2013.
[9] K. Koji, I. Kohei, T. Keita, and F. Toshiaki, “Light field coding using weighted binary images,” IEICE Trans. Inf. & Syst., vol.E102.D, no.11, pp.2110–2119, 2013.
[10] MIT Media Lab’s Camera Culture Group, “Compressive light field camera,” http://cameraculture.media.mit.edu/projects/compressive-light-field-camera/.
[11] Computer Graphics Laboratory, Stanford University, “The (new) stanford light field archive,” 2018. http://lightfield.stanford.edu.
[12] Heidelberg Collaboratory for Image Processing, “4D light field dataset,” 2018. http://hci-lightfield.iwr.uni-heidelberg.de/.
[13] Heidelberg Collaboratory for Image Processing, “Datasets and benchmarks for densely sampled 4D light fields;” 2016. http://lightfieldgroup.iwr.uni-heidelberg.de/?page_id=713.
[14] Fujii Laboratory, “Computational camera project,” 2020. https://www.fujii.nuee.nagoya-u.ac.jp/Research/CompCam/.