Learning to Capture Light Fields through a Coded Aperture Camera

Yasutaka Inagaki[1][0000−0001−6199−3139], Yuto Kobayashi[1][0000−0003−1615−2183], Keita Takahashi[1][0000−0001−9429−5273], Toshiaki Fujii[1][0000−0002−3440−5132], and Hajime Nagahara[2][0000−0003−1579−8767]

[1] Graduate School of Engineering, Nagoya University, Japan
{inagaki,kobayashi,takahasi,fujii}@fujii.nuee.nagoya-u.ac.jp
[2] Institute for Datability Science, Osaka University, Japan
nagahara@ids.osaka-u.ac.jp

Abstract. We propose a learning-based framework for acquiring a light field through a coded aperture camera. Acquiring a light field is a challenging task due to the amount of data. To make the acquisition process efficient, coded aperture cameras were successfully adopted; using these cameras, a light field is computationally reconstructed from several images that are acquired with different aperture patterns. However, it is still difficult to reconstruct a high-quality light field from only a few acquired images. To tackle this limitation, we formulated the entire pipeline of light field acquisition from the perspective of an auto-encoder. This auto-encoder was implemented as a stack of fully convolutional layers and was trained end-to-end by using a collection of training samples. We experimentally show that our method can successfully learn good image-acquisition and reconstruction strategies. With our method, light fields consisting of 5×5 or 8×8 images can be successfully reconstructed only from a few acquired images. Moreover, our method achieved superior performance over several state-of-the-art methods. We also applied our method to a real prototype camera to show that it is capable of capturing a real 3-D scene.

Keywords: Light Field, CNN, Coded Aperture

1 Introduction

The notion of a light field, which describes all light-rays traveling in 3-D free space [1–3], has been utilized in various applications, such as view synthesis [4–6], depth estimation [7–9], synthetic refocusing [10, 11], super resolution [12, 7], 3D displays [13–16], and object recognition [17, 18]. Several researchers have published light field datasets [19–22] for accelerating research in this field. A light field dataset is usually represented as a set of multi-view images that are aligned densely with tiny viewpoint intervals.

Acquiring a light field is a challenging task due to the amount of data because a light field typically consists of dozens of images. Several researchers have employed direct approaches such as using a moving camera gantry [2] or multiple cameras [23–25] to capture a target from different viewpoints. These approaches are costly in terms of the hardware or the time required to capture the entire light field. Meanwhile, lens-array based cameras [26, 27, 11, 28] and coded aperture/mask cameras [29–35] have also been utilized to achieve more efficient acquisition. In the case of lens-array based cameras, an entire light field can be captured with a single acquired image, but the spatial resolution of each image is in a trade-off relationship with the number of viewpoints. In contrast, coded aperture/mask cameras can capture a light field with the same spatial resolution as that of an image sensor, but the quality of the light field is in a trade-off relationship with the number of acquired images [3]. Other researchers used view synthesis techniques [7, 6, 36, 37] to generate a dense light field from a few images that were acquired with sparser intervals or even from one image. However, view synthesis involves added difficulty in terms of accurately estimating depth/disparity from the given images and reproducing view-dependent phenomena such as non-Lambertian reflections and occlusions.

This paper is focused on the problem of efficiently acquiring a light field by using a coded aperture camera. Using this camera, a light field is computationally reconstructed from several images that are acquired with different aperture patterns. Earlier methods [30–32] could not drastically reduce the number of images required to reconstruct a light field. However, with recent methods [33, 35], the number of acquired images has been reduced to only a few for a light field consisting of 5×5 viewpoints. We want to further improve the trade-off relationship between the number of acquired images and the quality of the reconstructed light field. The key to achieving this goal is to find good aperture patterns and a corresponding good reconstruction algorithm.

It is important to understand the principles behind reducing the number of acquired images. In previous works, this problem has been tackled from the perspective of compressive sensing [38–40]. Compressive sensing is a framework for reconstructing a signal from a reduced number of samples, where inherent structures in a target signal are exploited to seek the optimal sampling and reconstruction strategies. Along this theme, light-field acquisition has been formalized from the perspective of sparse representation using a learned dictionary [33] and approximation using the most significant basis vectors [35]. These methods can successfully reconstruct a light field from only a few acquired images. However, they often fail to reconstruct the details of the light field due to the limited representation capability of the dictionary and basis. Moreover, sparse reconstruction requires significant computational time due to the inherent complexity of the iterative algorithms that are used.

In contrast, we view the problem of acquiring a light field from the perspective of an auto-encoder [41, 42]. An auto-encoder employs a simple structure, in which an encoder network is connected to a decoder network, and it is trained to best approximate the identity mapping between the input and output. In our method,

[3] Dappled photography [29] is categorized as a coded mask method, but there is a trade-off between the maximum spatial and angular frequencies.

an encoder and decoder correspond to the image acquisition and reconstruction processes, respectively, because the original light field is once reduced (encoded) to only a few images through the physical imaging process of the coded aperture camera, and these images are then combined to reconstruct (decode) a light field with the same size as the original one. We implemented this auto-encoder as a fully convolutional neural network (CNN) and trained it end-to-end by using a collection of training samples. The parameters of the trained network correspond to the aperture patterns and reconstruction algorithm that are jointly optimized over the training dataset. In short, our method can learn to capture and reconstruct a light field through a coded aperture camera by utilizing the powerful framework of a deep neural network (DNN).

We implemented our method by using Chainer [43] and trained it by using samples taken from 51 light-field datasets. We experimentally show that our method can successfully learn good image-acquisition and reconstruction strategies. By using our method, light fields consisting of 5×5 or 8×8 images can be reconstructed with high quality from only a few acquired images. Moreover, our method achieved superior performance over several state-of-the-art methods [33, 6, 35, 36] including both compressive-sensing based and view-synthesis based approaches. We also applied our method to a real prototype camera to show that it is capable of capturing a real 3-D scene.

We briefly review the related works on the usage of DNNs for similar applications. DNNs have been utilized to optimize the imaging pipelines including camera designs for color image acquisition [44], depth estimation [45], and high-speed video acquisition [46]. A learning based approach was also used for temporal interpolation of light field video [47]. As the most similar work to ours, compressively sampled light fields were successfully reconstructed using DNNs in [48]. However, in contrast to our method, the compression process (camera design) in [48] was not optimized in the training stage but determined beforehand. To the best of our knowledge, we are the first to use a DNN for designing the entire pipeline of a computational camera for light-field acquisition. Moreover, the difference between the network architectures resulted in significant difference in the reconstruction time: 6.7 minutes for [48] and only 1 second for ours.

2 Proposed Method

2.1 Light field capturing through coded aperture camera

A schematic diagram of a coded aperture camera is illustrated in Fig. 1. An architecture that is equivalent to this diagram can be implemented by using relay optics and an LCoS (liquid crystal on silicon) device [31, 33, 49]. Here, we present a mathematical formulation for acquiring a light field through a coded aperture camera.

All incoming light rays that will be recorded by this camera are parameterized with four variables (s, t, u, v), where (s, t) and (u, v) denote the intersections with the aperture and imaging planes, respectively. Therefore, the light field is defined

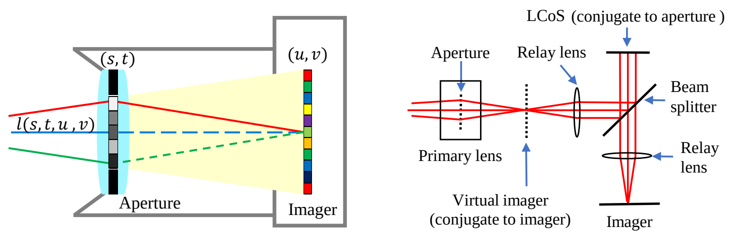


Fig. 1: Schematic diagram and implementation of coded aperture camera.

over 4-D space (s, t, u, v), with which the light intensity is described as $l(s, t, u, v)$. The light field is equivalently described as a set of rectified multi-view images called "sub-aperture images", $\{x_{s,t}(u, v) = l(s, t, u, v)\}$, where (s, t) corresponds to the viewpoint defined on the aperture plane.

We consider a coded aperture design where the transmittance of the aperture can be controlled for each position and for each acquisition. Let $a_n(s, t)$ be the transmittance at position (s, t) for the n-th acquisition $(n = 1, \ldots, N)$. The observed image $y_n(u, v)$ is formed as

$$y_n(u,v) = \iint a_n(s,t) l(s,t,u,v) dsdt = \iint a_n(s,t) x_{s,t}(u,v) dsdt. \tag{1}$$

When the aperture plane is quantized into a finite number of square blocks, they can be numbered by an integer m $(m = 1, \ldots, M)$. Equation (1) is rewritten as

$$y_n(u,v) = \sum_{m=1}^{M} a_{n,m} x_m(u,v), \tag{2}$$

where M is the total number of blocks, and $a_{n,m}$ [equivalent to $a_n(s, t)$] is the transmittance of the aperture at position m for the n-th acquisition.

Reconstructing the light field is equivalent to estimating M sub-aperture images $x_m(u, v)$ from the N given observations $y_n(u, v)$. In particular, we are interested in the case of $N \ll M$, where the entire light field could be reconstructed from only a few observed images. To this end, we need to find good transmittance patterns $a_{n,m}$ and a corresponding reconstruction method.

In the remainder of this paper, we assume that each image has only a single channel for simplicity, and this applies to both $x_m(u, v)$ and $y_n(u, v)$. When dealing with a color image, we simply divide it into three independent single-channel images and apply the same procedure to them.

2.2 Formulation as convolutional neural network

The observation (image acquisition) process of a coded aperture camera, which is given by Eq. (2), can be written in a form of a mapping as

$$f : \mathcal{X} \rightarrow \mathcal{Y} \tag{3}$$

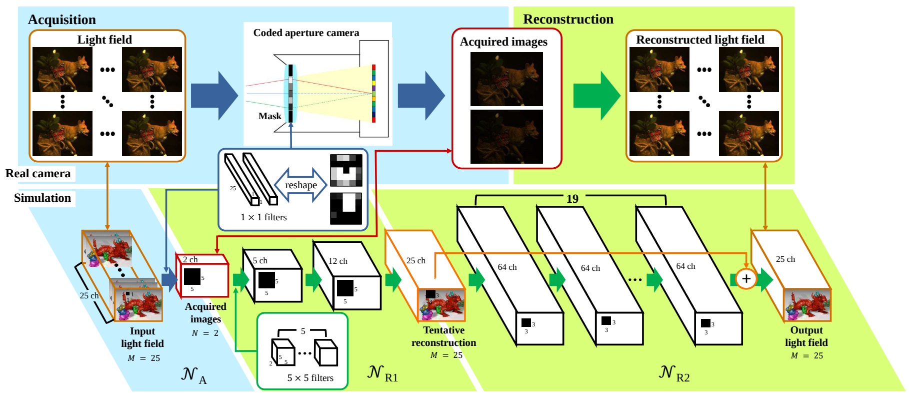


Fig. 2: Modeling light-field acquisition using CNN

where $\mathcal{X}$ represents a vector that contains all the pixels of $x_m(u,v)$ for all $m \in \{1, ..., M\}$. When each image has $U \times V$ pixels, the length of the vector is UVM. Similarly, $\mathcal{Y}$ represents a vector that contains all the pixels of $y_n(u,v)$ for all $n \in \{1, ..., N\}$. The reconstruction is also written as

$$g : \mathcal{Y} \rightarrow \hat{\mathcal{X}} \tag{4}$$

where $\hat{\mathcal{X}}$ corresponds to an estimate of $\mathcal{X}$. The composite mapping $h = g \circ f$ should be as close to the identity as possible, under the condition that $N \ll M$. This problem can be regarded as that of an auto-encoder, where the encoder and decoder correspond to f and g, respectively. The goal of optimization is formulated with the squared error loss as

$$\arg\min_{f,g} |\mathcal{X} - \hat{\mathcal{X}}|^2 = \arg\min_{f,g} \sum_m \sum_{u,v} |x_m(u,v) - \hat{x}_m(u,v)|^2 \tag{5}$$

where an estimate of $x_m(u,v)$ is written as $\hat{x}_m(u,v)$.

On top of this formulation, we implemented composite mapping $h = g \circ f$ as a stack of 2-D convolutional layers. The entire network can be trained end-to-end by using a collection of training samples. The learned parameters in f and g correspond to the aperture patterns $a_{n,m}$ and the reconstruction algorithm, respectively, both of which are jointly optimized over the training dataset. In short, our method can learn to capture a light field through a coded aperture camera by utilizing the powerful framework of a DNN. This network can easily be applied to a physical coded-aperture camera. The mapping f is conducted by the physical imaging process of the camera, the aperture patterns of which are configured in accordance to the learned parameters in f. Then, the images acquired by the cameras are inputted to the network corresponding to g, by which we can computationally reconstruct a target light field.

An example with $M = 25$ and $N = 2$ is illustrated in Fig. 2. The architecture of our network is summarized below. Note that this is only an example we have found through trial and error, and one may find other architectures that are essentially equivalent in spirit but can perform better than ours.

In a 2-D convolutional network, data are represented as 2-D images with multiple channels. For the l-th convolution layer, the relationship between the input $x_c^l(u,v)$ and output $x_{c'}^{l+1}(u,v)$, where c and c' denote channels, is represented as

$$x_{c'}^{l+1}(u,v) = \sum_c k_{c,c'}(u,v) * x_c^l(u,v) + b_{c'} \tag{6}$$

which is optionally followed by an activation function. The 2-D convolution kernels represented by $k_{c,c'}(u,v)$ and the bias terms $b_{c'}$ are optimized through the training stage. We used one-pixel stride and an appropriate zero padding to keep the image size unchanged before and after the convolution. Meanwhile, the number of channels (the range for c and c') can arbitrarily be changed between the input and output. Therefore, 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. To apply this data structure to our purpose, we treated the indices n and m as the channels. Therefore, the viewpoint m corresponds to the channels in the input and output data, and the index for the acquired images n corresponds to the channel of the intermediate data between f and g.

The mapping f has a few restrictions because it corresponds to the physical imaging process described by Eq. (2); the transmittance $a_{n,m}$ should be limited within $[0,1]$, each pixel (u,v) cannot be mixed with other neighbors, and the number of channels should be reduced from M to N. We implemented this using a single 2-D convolution layer with 1×1 convolution kernels, where the weights are limited within the range $[0,1]$, and $b_{c'} = 0$ for all c'. In this case, $k_{c,c'}$ is equivalent to $a_{c',c}$, and Eq. (6) corresponds to Eq. (2). To keep the range limitation during the training stage, we clipped the weights within $[0,1]$ each time when the network was updated by using the gradient information obtained from a mini-batch. To better simulate the physical imaging process, Gaussian noise with standard deviation σ was added to the output of this network. This network is called the "image acquisition network" and denoted as $\mathcal{N}_A$.

Meanwhile, the mapping g can take an arbitrary form because it is a truly computational process. We decomposed this mapping into two networks, which are denoted as $\mathcal{N}_{R1}$ and $\mathcal{N}_{R2}$, respectively, where R stands for reconstruction. In the former network $\mathcal{N}_{R1}$, the number of channels is gradually increased from M to N by using several convolution layers with 5×5 kernels and linear activations, resulting in a tentative reconstruction of the target light field. This tentative result is further refined by the latter network $\mathcal{N}_{R2}$ that employs the structure of very deep super resolution (VDSR) [50]. Here, 19 convolution layers with 3×3 kernels and ReLU activations are followed by the addition of the tentative light field itself, which forces the network to learn only the residual (difference between the tentative result and ground truth). The former and latter networks are also called the "tentative reconstruction and refinement networks," respectively.

Table 1: Network architecture for reconstruction

M	N	Tentative reconstruction	Refinement
25 (5×5)	1	$1 \rightarrow 2 \rightarrow 5 \rightarrow 12 \rightarrow 25$	$25 \rightarrow 64 \rightarrow 64, ..., 64 \rightarrow 25$
	2	$2 \rightarrow 5 \rightarrow 12 \rightarrow 25$	
	4	$4 \rightarrow 7 \rightarrow 13 \rightarrow 25$	
64 (8×8)	1	$1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64$	$64 \rightarrow 64 \rightarrow 64, ..., 64 \rightarrow 64$
	2	$2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64$	
	4	$4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64$	

Table 2: Datasets used for training and testing

M	Stage	Datasets
25 (5×5)	Training	Chess, Lego Bulldozer, Lego Truck, Eucalyptus Flowers, Amethyst, Bracelet, The Stanford Bunny, Jelly Beans, Lego Knights, Tarot Cards and Crystal Ball (small angular extent), Treasure Chest (Stanford [20]), Red Dragon, Happy Buddha, Messerschmitt, Dice, Green Dragon, Mini Cooper, Butterfly, Lucy (MIT [19]), Bedroom, Bicycle, Herbs, Origami, Boxes, Cotton, Sideboard, Antinous, Boardgames, Dishes, Greek, Museum, Pens, Pillows, Platonic, Rosemary, Table, Tomb, Town, Vinyl (New HCI [21]), Buddha, Buddha 2, StillLife, Papillon, MonaRoom, Medieval, Horse, Couple, Cube, Maria, Pyramide, Statue (Old HCI [22])
	Testing	Dragon and Bunnies, Fish (MIT [19]), Dino, Kitchen, Medieval2, Tower (New HCI [21])
64 (8×8)	Training	Chess, Lego Bulldozer, Lego Truck, Eucalyptus Flowers, Amethyst, Bracelet, The Stanford Bunny, Jelly Beans, Lego Knights, Tarot Cards and Crystal Ball (small angular extent), Treasure Chest Bedroom (Stanford [20]), Bicycle, Herbs, Origami, Boxes, Cotton, Sideboard, Antinous, Boardgames, Dishes, Greek, Museum, Pens, Pillows, Platonic, Rosemary, Table, Tomb, Town, Vinyl (New HCI [21])
	Testing	Dino, Kitchen, Medieval2, Tower (New HCI [21])

2.3 Implementation and training details

We considered several network configurations for different values of M and N, as summarized in Table 1. The number of viewpoints in the target light field (M) was set to 25 (5×5) or 64 (8×8). The number of acquired images (N) was set to 1, 2, or 4. We employed different network architectures for $\mathcal{N}_{R1}$ for different numbers of N. The numbers connected by arrows in the table indicate the transition in the number of channels. The design principle here is to increase the number of channels gradually from N to M. Meanwhile, $\mathcal{N}_{R2}$ was only subject to M, and the number of channels for the intermediate layers was set to 64 for both $M = 25$ and $M = 64$. All of these architectures were found through trial and error, and there might be room for improvement.

In the training stage, each training sample is a set of 2-D image blocks with 64×64 pixels that are extracted from the same positions in the multi-view images. As mentioned earlier, a viewpoint is considered to be a channel, and thus, each sample is represented as an M channel image that has 64×64 pixels. The training samples were collected from several light field datasets provided at [19–22], the details of which are summarized in Table 2. Three color channels of each dataset were used as three individual datasets. Moreover, we extracted many training samples from each dataset by changing the position of the 2-D blocks; we took blocks every 32 pixels both in the horizontal and vertical directions. We also augmented the data by changing the intensity levels of each sample uniformly; we multiplied 1.0, 0.9, 0.8, 0.7, 0.6 and 0.5 with the original samples. Samples that were too smooth, i.e., those in which the intensity was

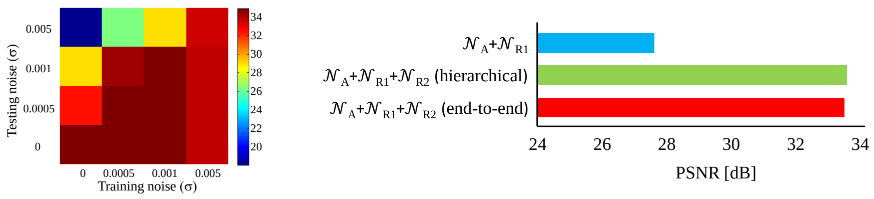


Fig. 3: Performance analysis of our method with different noise levels (left) and different network architectures and training schemes (right).

almost uniform across 64 × 64 pixels, were removed from the training samples. Finally, we collected 295,200 and 166,680 samples for $M = 25$ and $M = 64$, respectively. Similar to the case with VDSR [50], our network can be trained with a small training dataset because it consists of only convolutional layers with small kernels, and thus, the number of parameters does not increase too much. In the testing stage, the entire light field can be handled at once because fully convolutional networks can accept images with arbitrary size.

Our method was implemented by using Chainer [43] 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 [51]. The number of epochs was fixed to 20 for all experiments. The training with $M = 25$ and $N = 2$ took approximately 4 hours on a PC equipped with an NVIDIA Geforce GTX 1080 Ti. In the testing stage, it took about 0.5 sec to reconstruct the entire light field with 840 × 593 pixels and 25 viewpoints. Our software will be made public soon [52].

3 Experiments

3.1 Network design and performance

One of the key factors of our network is the noise that is added to the acquired images. Ideally, the noise level assumed in the training stage should be close to that in real situations. However, it is impractical to train different networks for all possible noise levels. It is more likely that a network trained with a certain noise level will be applied to different noise levels. To see the effect of the discrepancy in noise levels, we used different noise levels in the training and testing stages. Here, the noise level σ was defined with respect to the image intensity range $[0, 1]$ for acquired images $y_n(u, v)$. The results obtained with the Dragon and Bunnies dataset and $N = 2$ are summarized in Fig. 3 (left) [4]. When noise in the testing stage was larger than that in the training stage, the reconstruction quality was largely degraded. Meanwhile, in the opposite case, the degradation was moderate. Therefore, we concluded that assuming a high noise level in the

[4] The PSNR values were obtained from the mean-squared errors over all viewpoints, pixels, and color channels. These values correspond to a logarithmic representation of the loss function given by Eq. (5).

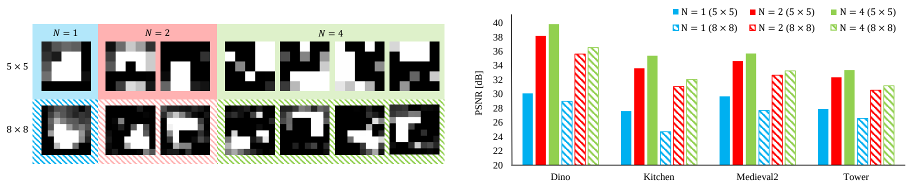


Fig. 4: Obtained aperture patterns (left) and PSNR (right).

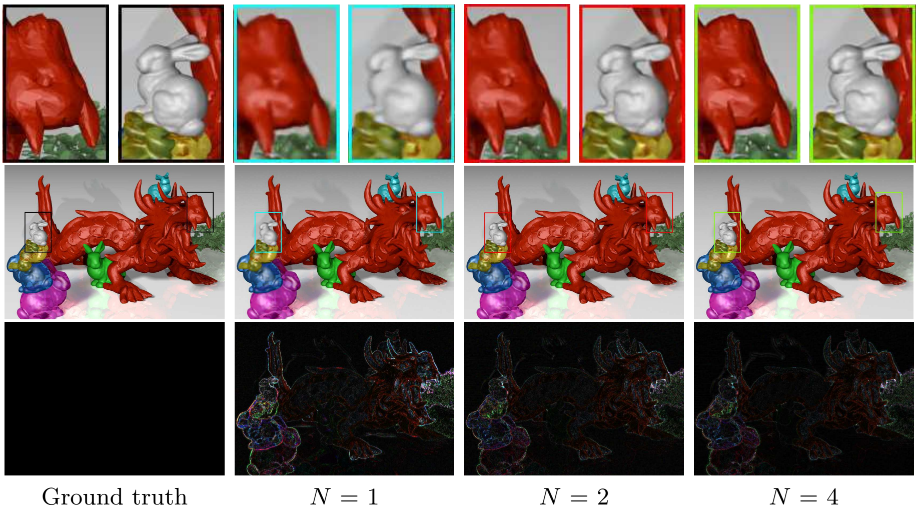
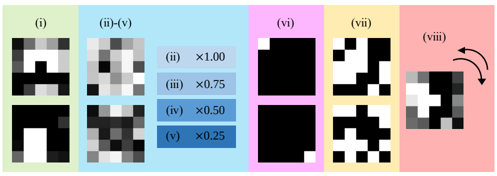


Fig. 5: Central views reconstructed by our method. Differences from ground truth (magnified by 5) are also presented.

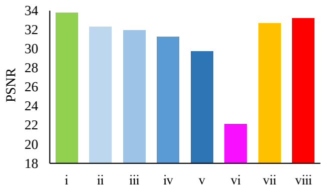
training stage is a safe strategy. We use σ=0.005 in the training stage in the remainder of this paper.

We next compared different network architectures and training schemes. We tested three cases: (a) $\mathcal{N}_A + \mathcal{N}_{R1}$, (b) $\mathcal{N}_A + \mathcal{N}_{R1} + \mathcal{N}_{R2}$, where $\mathcal{N}_A + \mathcal{N}_{R1}$ and $\mathcal{N}_{R2}$ were trained individually and then concatenated and fine-tuned, and (c) $\mathcal{N}_A + \mathcal{N}_{R1} + \mathcal{N}_{R2}$, where the entire network was trained end-to-end from scratch. As shown in Fig. 3 (right), the importance of the refinement network $\mathcal{N}_{R2}$ is clear. Meanwhile, the difference between (b) and (c) was negligible despite the fact that (b) required additional complexity for the training stage. Therefore, we adopted training strategy (c) for the remainder of this paper.

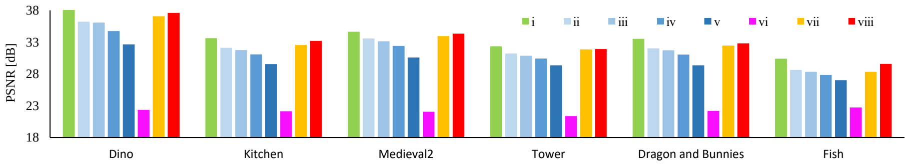
We then evaluated the performance of our method over several datasets. The number of viewpoints M was set to 25 (5×5) and 64 (8×8), and the number of acquired images N was set to 1, 2, and 4. The noise level for testing was set to 0.005. The obtained aperture patterns and quantitative scores in PSNR are summarized in Fig. 4. Several reconstructed images are presented in Fig. 5. From these results, we can see that the reconstruction quality improved as the number of acquired images increased. With N=1, our method could not



(a) Aperture patterns



(b) Average PSNR



(c) PSNR values for individual datasets

Fig. 6: Analysis on aperture patterns with $M = 25$ and $N = 2$.

correctly reproduce the disparities among the viewpoints. However, by changing N from 1 to 2, the reconstruction quality improved significantly; fine details (see the close-ups) were successfully produced, and the correct amount of disparities was observed among the viewpoints. Meanwhile, the quality improvement from $N = 2$ to $N = 4$ was moderate. We observed the same tendency both with $M = 25$ and 64, though $M = 64$ was more challenging than 25 due to the increased number of viewpoints and less number of training samples.

3.2 Analysis on aperture patterns

To analyze the effect of aperture patterns, we conducted another experiment. We used several aperture patterns with $M = 25$ and $N = 2$, as summarized in Fig. 6 (a). For (i), we used a set of aperture patterns that was trained normally, which is denoted as "ours (normal)" or simply "ours". For (ii)–(vii), we fixed the aperture patterns during the training stage; in this case, only the reconstruction network was optimized for the training dataset. We generated a set of random aperture patterns for (ii) and changed its brightness constantly for (iii)–(v). The patterns for (vi) corresponds to light field reconstruction from a pair of stereo images. The patterns in (vii) took randomized binary values, which serves as a simulation for binary apertures. Finally, the patterns in (viii) were trained over the dataset under the constraint that the second pattern was the first one rotated 90 degrees. This was easily implemented by using the same convolution kernels for the first and second image acquisition but tweaking the order of channels in the input data. This pattern set may have a practical value because it can be implemented by using a static aperture pattern (such as a printed one, instead of an expensive electronic device such as an LCoS) with a simple rotation mechanism. This is denoted as "ours (rotated)." We set the noise level for testing to 0.005.

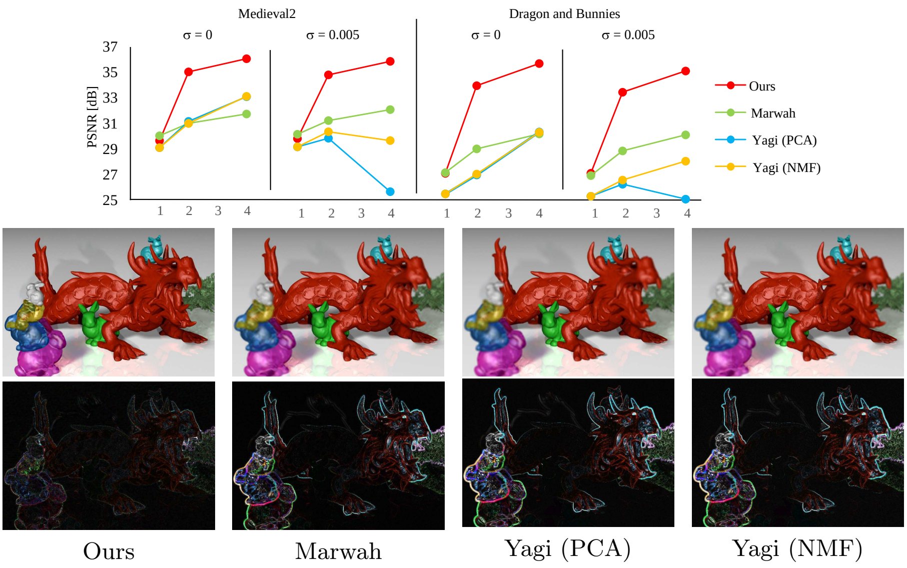


Fig. 7: Comparison with compressive-sensing based methods [33, 35]. (Top) quantitative reconstruction quality. (Bottom) reconstructed images and difference from the ground truth (magnified by 5).

3.3 Comparison with state-of-the-art methods

Several interesting indications are found from the results shown in Fig. 6(b)(c). First, a good reconstruction quality was attained even with non-optimized aperture patterns such as (ii) and (vii), probably because the reconstruction network is capable of adapting to different aperture patterns. Second, the brightness of the apertures (the total amount of light) is important for the quality. The quality degradation observed in (iii)–(vi) can be attributed to the bad signal-to-noise ratio of the acquired images. Finally, the rotated patterns in (viii) achieved quality close to that of normally trained patterns (i), suggesting the possibility of developing a camera with a rotatable static aperture in the future.

We compared our method against four state-of-the-art methods [33, 6, 35, 36], for which we used the software provided by the respective authors. To make the comparison fair, the configurations for the methods were kept unchanged from the original implementations as much as possible. To see the results more closely, please refer to the supplementary video.

We first present a comparison with two compressive-sensing based methods: Marwah's method [33] using a sparse dictionary and Yagi's method [35] using the most significant basis vectors selected by PCA or NMF. The number of viewpoints M was set to 25 (5×5) in accordance with the original implementations. The noise level was set to $\sigma = 0$ and $\sigma = 0.005$. The results are summarized in Fig. 7. As shown in the graphs of the figure, our method performed clearly better than Yagi's method. Marwah's method achieved slightly better scores than ours

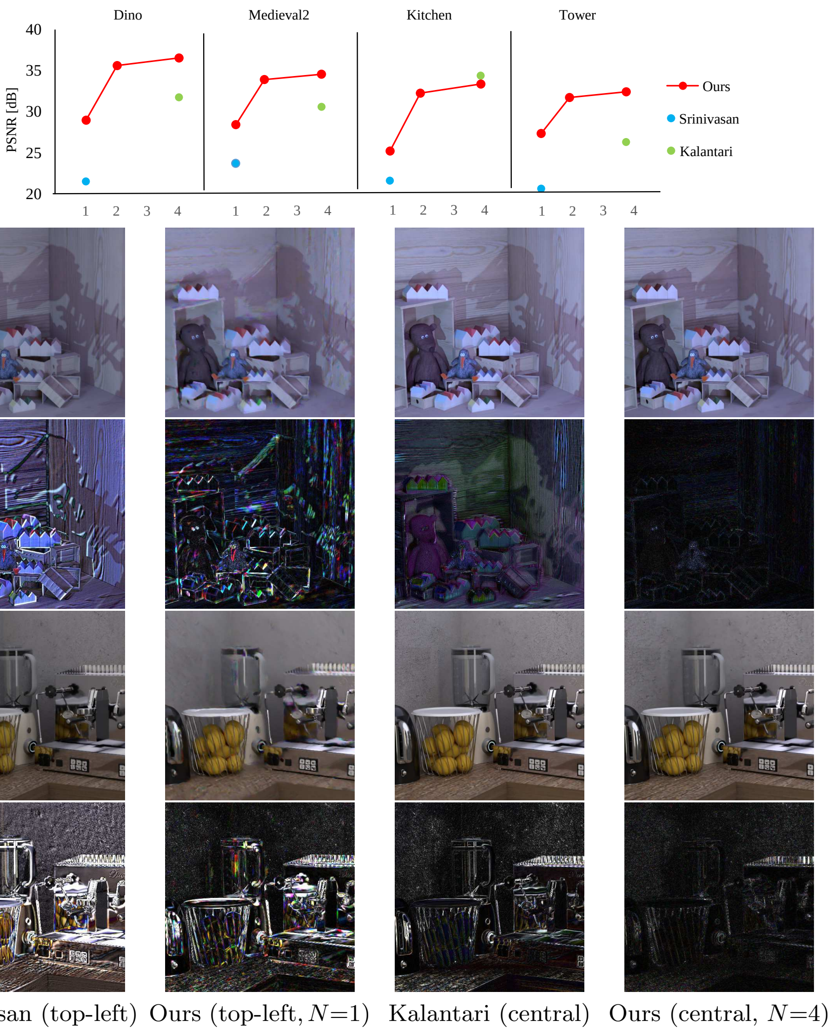


Fig. 8: Comparison with view-synthesis based methods [6, 36]. (Top) quantitative reconstruction quality. (Bottom) reconstructed images and difference from ground truth (magnified by 5) obtained with Dino and Kitchen.

with $N = 1$, but with $N = 2$ and $N = 4$, it was inferior to our method with significant margins. Moreover, the main drawback of Marwah's method is the computational complexity; it took $11 - 90$ hours to reconstruct a single light field on our desktop computer. Shown in the bottom are reconstructed central views and the differences from the ground truth obtained with Dragon and Bunnies, $N = 2$, and σ=0.005. We can see that a better visual quality was achieved by our method over the others.

We next show another comparison with two view-synthesis based methods. Kalantari's method [6] can reconstruct a light field with 8 $\times$ 8 viewpoints given the four corner images. It was reported that this method performed better than several state-of-the-art view synthesis methods. Meanwhile, Srinivasan's method [36] can reconstruct a light field with 8 $\times$ 8 viewpoints only from one image corresponding to the central viewpoint (actually, it is located at the bottom-right nearest neighbor to the center). No noise was applied for these two methods. As for our method, the number of viewpoints M was set to 64 (8 $\times$ 8) as well, but noise with $\sigma = 0.005$ was applied. The results are summarized in Fig. 8. Srinivasan's method produced natural-looking images with around 30 $-$ 60 seconds computation time, but it failed to reproduce correct disparities among the viewpoints. Kalantari's method achieved better quality than Srinivasan's method, but it required four images as the input and took about 30 minutes without GPU boost. Our method achieved overall better reconstruction quality than the other two with much less computation time (about 1 second).

3.4 Experiment using real camera

Finally, we conducted an experiment using a real coded-aperture camera. We adopted the same hardware design as the ones reported in [31, 33, 49]. The resolution of the camera (FLIR GRAS-14S5C-C) was 1384 $\times$ 1036 pixels, which corresponds to the spatial resolution of the light field acquired with it. The exposure time was set to 40 msec. We used a Nikon Rayfact lens (25 mm F1.4 SF2514MC). The aperture was implemented as an LCoS display (Forth Dimension Displays, SXGA-3DM) with 1280 $\times$ 1024 pixels. We divided the central area of the LCoS display into 5 $\times$ 5 regions, each with 150 $\times$ 150 pixels. Accordingly, the angular resolution of the light field was 5 $\times$ 5. We found that light rays could not completely be shut off by using the black aperture (all of the 25 regions were set to 0), which means that an image acquired with the black aperture was not completely black. Therefore, we subtracted the image acquired with the black aperture from each of the acquired images before using them for reconstruction.

The experimental setup and reconstructed light fields are presented in Fig. 9. We used four sets of aperture patterns: ours (normal), ours (rotated), NMF [35], and sequential pinholes.[5] For the former three methods, the number of acquired images was set to 2. The reconstruction was performed with the trained reconstruction networks for the first and second ones and the method presented in [35] for the third one. The last one, sequential pinholes, corresponds to the direct acquisition of 25 images, where only 1 region out of 25 on the aperture was opened in sequence. As shown in the bottom of Fig. 9, our method achieved a striking reconstruction quality with only two acquired images. The results with our methods were even better than those obtained with 25 sequentially acquired images because pinhole patterns suffer from noise due to a lack of light. We also

[5] We did not test Marwah's method [33] with our real camera because this method was designed for an optical configuration different from our camera, and the reconstruction with this method would require prohibitively expensive computation.

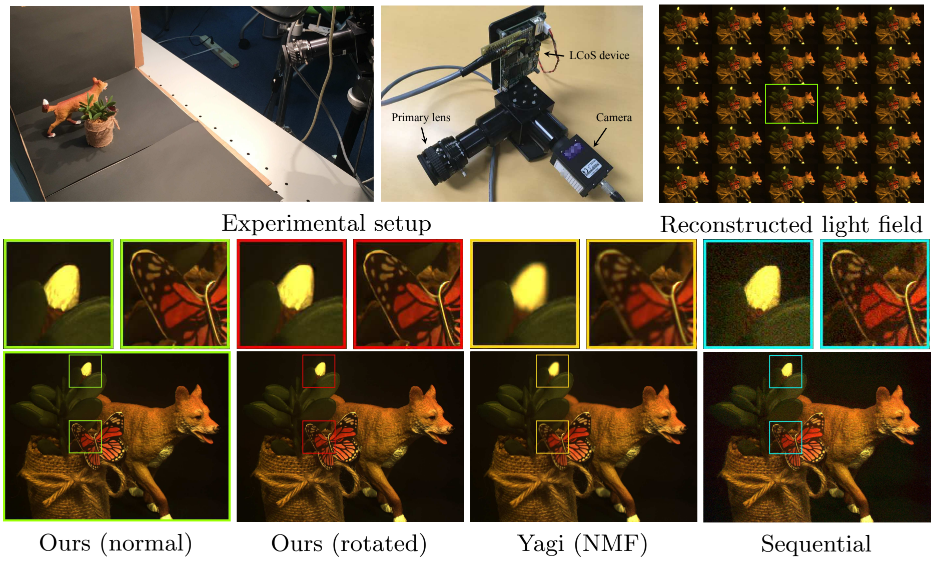


Fig. 9: Experiment using real coded-aperture camera

observed natural disparities among the viewpoints in the reconstructed results. Please refer to the supplementary video for more detail.

4 Conclusion

We proposed a learning-based framework for acquiring a light field through a coded aperture camera to reduce the required number of acquired images to only a few. This framework was formulated from the perspective of an auto-encoder. This auto-encoder was implemented as a stack of fully convolutional layers and was trained end-to-end by using a collection of training samples. We experimentally showed that our method can successfully learn good image-acquisition and reconstruction strategies. Using our method, light fields consisting of 5×5 or 8×8 images can be reconstructed with high quality from only a few acquired images. Moreover, our method achieved superior performance over several state-of-the-art methods. We also applied our method to a real prototype camera to show that it is capable of capturing a real 3-D scene.

Our future work includes several directions. The performance of our method could be improved by tweaking the architecture of the reconstruction network or increasing the number of training samples. The squared loss used in our method could be replaced with or supplemented by other loss functions that are closer to the perceptual image quality such as VGG loss and the framework of a generative adversarial network (GAN) [53]. The development of a camera with a rotatable static aperture pattern would also be an interesting direction.

References

1. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Computational Models of Visual Processing. (1991) 3–20
2. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM (1996) 31–42
3. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. (1996) 43–54
4. Tanimoto, M., Tehrani, M.P., Fujii, T., Yendo, T.: Free-viewpoint TV. IEEE Signal Processing Magazine **28**(1) (2011) 67–76
5. Shi, L., Hassanieh, H., Davis, A., Katabi, D., Durand, F.: Light field reconstruction using sparsity in the continuous fourier domain. ACM Transactions on Graphics (TOG) **34**(1) (2014) 12
6. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016) **35**(6) (2016)
7. Wanner, S., Goldluecke, B.: Variational light field analysis for disparity estimation and super-resolution. IEEE transactions on pattern analysis and machine intelligence **36**(3) (2014) 606–619
8. Wang, T.C., Efros, A.A., Ramamoorthi, R.: Depth estimation with occlusion modeling using light-field cameras. IEEE transactions on pattern analysis and machine intelligence **38**(11) (2016) 2170–2181
9. Williem, W., Park, I.K., Lee, K.M.: Robust light field depth estimation using occlusion-noise aware data costs. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99) (2017) 1–1
10. Isaksen, A., McMillan, L., Gortler, S.J.: Dynamically reparameterized light fields. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. (2000) 297–306
11. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Computer Science Technical Report CSTR **2**(11) (2005) 1–11
12. Bishop, T.E., Zanetti, S., Favaro, P.: Light field superresolution. In: Computational Photography (ICCP), 2009 IEEE International Conference on, IEEE (2009) 1–9
13. Wetzstein, G., Lanman, D., Hirsch, M., Raskar, R.: Tensor Displays: Compressive Light Field Synthesis using Multilayer Displays with Directional Backlighting. ACM Trans. Graph. (Proc. SIGGRAPH) **31**(4) (2012) 1–11
14. Huang, F.C., Chen, K., Wetzstein, G.: The light field stereoscope: immersive computer graphics via factored near-eye light field displays with focus cues. ACM Transactions on Graphics (TOG) **34**(4) (2015) 60
15. Lee, S., Jang, C., Moon, S., Cho, J., Lee, B.: Additive light field displays: realization of augmented reality with holographic optical elements. ACM Transactions on Graphics (TOG) **35**(4) (2016) 60
16. Saito, T., Kobayashi, Y., Takahashi, K., Fujii, T.: Displaying real-world light fields with stacked multiplicative layers: Requirement and data conversion for input multiview images. Journal of Display Technology **12**(11) (2016) 1290–1300
17. Maeno, K., Nagahara, H., Shimada, A., Taniguchi, R.I.: Light field distortion feature for transparent object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2786–2793

18. andJun Yan Zhu, T.C.W., Hiroaki, E., Chandraker, M., Efros, A., Ramamoorthi, R.: A 4d light-field dataset and cnn architectures for material recognition. In: European Conference on Computer Vision (ECCV). (2016) 121–138
19. MIT Media Lab's Camera Culture Group: Compressive light field camera http://cameraculture.media.mit.edu/projects/compressive-light-field-camera/.
20. Computer Graphics Laboratory, Stanford University: The (new) stanford light field archive (2018) http://lightfield.stanford.edu.
21. Heidelberg Collaboratory for Image Processing: 4D light field dataset (2018) http://hci-lightfield.iwr.uni-heidelberg.de/.
22. Heidelberg Collaboratory for Image Processing: Datasets and benchmarks for densely sampled 4D light fields. http://lightfieldgroup.iwr.uni-heidelberg.de/?page_id=713 (2016)
23. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. ACM Transactions on Graphics (TOG) **24**(3) (2005) 765–776
24. Fujii, T., Mori, K., Takeda, K., Mase, K., Tanimoto, M., Suenaga, Y.: Multipoint measuring system for video and sound-100-camera and microphone system. In: 2006 IEEE International Conference on Multimedia and Expo, IEEE (2006) 437–440
25. Taguchi, Y., Koike, T., Takahashi, K., Naemura, T.: 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 **15**(5) (Sept 2009) 841–852
26. Adelson, E.H., Wang, J.Y.: Single lens stereo with a plenoptic camera. IEEE transactions on pattern analysis and machine intelligence **14**(2) (1992) 99–106
27. Arai, J., Okano, F., Hoshino, H., Yuyama, I.: Gradient-index lens-array method based on real-time integral photography for three-dimensional images. Applied optics **37**(11) (1998) 2034–2045
28. Ng, R.: Digital light field photography. PhD thesis, stanford university (2006)
29. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Transactions on Graphics (TOG) **26**(3) (2007) 69
30. Liang, C.K., Lin, T.H., Wong, B.Y., Liu, C., Chen, H.H.: Programmable aperture photography: multiplexed light field acquisition. ACM Transactions on Graphics (TOG) **27**(3) (2008) 55
31. Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H., Nayar, S.K.: Programmable aperture camera using LCoS. In: European Conference on Computer Vision, Springer (2010) 337–350
32. Babacan, S.D., Ansorge, R., Luessi, M., Mataran, P.R., Molina, R., Katsaggelos, A.K.: Compressive light field sensing. IEEE Transactions on image processing **21**(12) (2012) 4746–4757
33. Marwah, K., Wetzstein, G., Bando, Y., Raskar, R.: Compressive light field photography using overcomplete dictionaries and optimized projections. ACM Transactions on Graphics (TOG) **32**(4) (2013) 46
34. Tambe, S., Veeraraghavan, A., Agrawal, A.: Towards motion aware light field video for dynamic scenes. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1009–1016
35. Yagi, Y., Takahashi, K., Fujii, T., Sonoda, T., Nagahara, H.: Pca-coded aperture for light field photography. In: IEEE International Conference on Image Processing (ICIP). (2017)

36. Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to synthesize a 4D RGBD light field from a single image. In: IEEE International Conference on Computer Vision. (2017) 2262–2270
37. Yoon, Y., Jeon, H.G., Yoo, D., Lee, J.Y., Kweon, I.S.: Learning a deep convolutional network for light-field image super-resolution. 2015 IEEE International Conference on Computer Vision Workshop (ICCVW) (2015) 57–65
38. Donoho, D.L.: Compressed sensing. IEEE Transactions on information theory **52**(4) (2006) 1289–1306
39. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE signal processing magazine **25**(2) (2008) 21–30
40. Candes, E.J., Eldar, Y.C., Needell, D., Randall, P.: Compressed sensing with coherent and redundant dictionaries. Applied and Computational Harmonic Analysis **31**(1) (2011) 59–73
41. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (July 2006) 504–507
42. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08 (2008) 1096–1103
43. Tokui, S., Oono, K., Hido, S., Clayton, J.: Chainer: a next-generation open source framework for deep learning. In: Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS). (2015)
44. Chakrabarti, A.: Learning sensor multiplexing design through back-propagation. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. (2016) 3089–3097
45. Shedligeri, P.A., Mohan, S., Mitra, K.: Data driven coded aperture design for depth recovery. In: 2017 IEEE International Conference on Image Processing (ICIP). (2017) 56–60
46. Iliadis, M., Spinoulas, L., Katsaggelos, A.K.: Deep fully-connected networks for video compressive sensing. arXiv, http://arxiv.org/abs/1603.04930 (2016)
47. Wang, T.C., Zhu, J.Y., Kalantari, N.K., Efros, A.A., Ramamoorthi, R.: Light field video capture using a learning-based hybrid imaging system. ACM Trans. Graph. **36**(4) (2017) 133:1–133:13
48. Gupta, M., Jauhari, A., Kulkarni, K., Jayasuriya, S., Molnar, A., Turaga, P.: Compressive light field reconstructions using deep learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2017) 1277–1286
49. Sonoda, T., Nagahara, H., Taniguchi, R.: Motion-invariant coding using a programmable aperture camera. IPSJ Transactions on Computer Vision and Applications **6** (6 2014) 25–33
50. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2016) 1646–1654
51. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: The International Conference on Learning Representations (ICLR). (2015)
52. Keita Takahashi: Computational camera project http://www.fujii.nuee.nagoya-u.ac.jp/%7Etakahasi/Research/CompCam/index.html.
53. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27. (2014) 2672–2680