Appendix

Details of Decoder Networks

We show the details of the decoder networks in the tables below. The former and latter parts of each network correspond to the tentative reconstruction and residual refinement, respectively. **k** is the kernel size. **chns** is the number of input/output channels. **activ** is the activation functions. **input** denotes the input to each layer. "Images" means the images captured by a coded-aperture camera. **output** corresponds to the reconstructed light field. Single and 2-S are the same configurations as the counterparts used in Inagaki et al. [17].

Table 1. 5×5 views

Layer	k	chns	activ	input
		Single		
conv1-0	5×5	1/2	-	Images
conv1-1	5×5	2/5	-	conv1-0
conv1-2	5×5	5/12	-	conv1-1
conv1-3	5×5	12/25	-	conv1-2
		2-S and 2-D		
conv1-1	5×5	2/5	-	Images
conv1-2	5×5	5/12	-	conv1-1
conv1-3	5×5	12/25	-	conv1-2
		3-D (V-shape)		
conv1-1	5×5	3/5	-	Images
conv1-2	5×5	5/12	-	conv1-1
conv1-3	5×5	12/25	-	conv1-2
Single, 2-S, 2-D, and 3-D (V-shape)				
conv2-1	3×3	25/64	ReLU	conv1-3
conv2-2	3×3	64/64	ReLU	conv2-1
conv2-3	3×3	64/64	ReLU	conv2-2
⋮	⋮	⋮	⋮	⋮
conv2-19	3×3	64/64	ReLU	conv2-18
conv2-20	3×3	64/25	-	conv2-19
output	conv1-3 + conv2-20			

Table 2. 8×8 views

Layer	k	chns	activ	input
		Single		
conv1-0	5×5	1/2	-	Images
conv1-1	5×5	2/4	-	conv1-0
conv1-2	5×5	4/8	-	conv1-1
conv1-3	5×5	8/16	-	conv1-2
conv1-4	5×5	16/32	-	conv1-3
conv1-5	5×5	32/64	-	conv1-4
		2-S		
conv1-1	5×5	2/4	-	Images
conv1-2	5×5	4/8	-	conv1-1
conv1-3	5×5	8/16	-	conv1-2
conv1-4	5×5	16/32	-	conv1-3
conv1-5	5×5	32/64	-	conv1-4
		3-D (V-shape)		
conv1-1	5×5	3/4	-	Images
conv1-2	5×5	4/8	-	conv1-1
conv1-3	5×5	8/16	-	conv1-2
conv1-4	5×5	16/32	-	conv1-3
conv1-5	5×5	32/64	-	conv1-4
Single, 2-S, and 3-D (V-shape)				
conv2-1	3×3	64/64	ReLU	conv1-5
conv2-2	3×3	64/64	ReLU	conv2-1
conv2-3	3×3	64/64	ReLU	conv2-2
⋮	⋮	⋮	⋮	⋮
conv2-19	3×3	64/64	ReLU	conv2-18
conv2-20	3×3	64/64	-	conv2-19
output	conv1-5 + conv2-20			

Evaluation against scene motions

The performance of our method against scene motions depends on the training dataset we used. Our training dataset, generated with pseudo motions, does not include large motions and temporal disocclusions.

To analyze this effect, we conducted a controlled experiment. We used the CG scene mentioned in the paper, and changed the global scale of the scene motions. For the 101-st frame, the reconstruction quality was 32.42 dB with the default speed (this is the condition reported in Fig. 6 left). When we doubled the speed, the quality decreased by 4.49 dB. Meanwhile, when we halved the speed, the quality increased by 1.53 dB. When we stopped the motion (the scene was static), the quality was 34.10 dB, which was slightly better than 33.77 dB of Inagaki's method [17] for the same static scene. We think this slightly better quality comes with an increased amount of training dataset (25 times larger than that in [17]) and three acquired images instead of two; three images lead to better robustness against noise even though the aperture patterns are only two.

To overcome the limitation, we need to enhance the training dataset and improve the network architecture. We also need to improve the camera hardware to increase the frame-rate, which will help to reduce the amount of motions between the frames.