Supplementary Materials for "Divide-Conquer-and-Merge: Memory- and Time-Efficient Holographic Displays"

Zhenxing Dong* Jidong Jia* Yan Li Yuye Ling †

Department of Electronic Engineering, Shanghai Jiao Tong University

1 Additional Implementation Details

In this section, we give the detailed network structure in Sect. 1.1 and provide a comprehensive numerical analysis of networks in different methods in Sect. 1.2.

1.1 Network Structure

As mentioned in the main paper, we have made modifications to the phase generator and phase encoder of HoloNet [1] and CCNNs [5] to accommodate the division of the input image into r^2 sub-image.

For HoloNet with our method, we just modify the input and output channels of the first and last convolution layers in both the phase generator and phase encoder. The other parts remain consistent with HoloNet, and the specific modifications are listed in Table 1 and Table 2. Thanks to the pixel-unshuffle layer, the resolution of the input images to sub-networks in our frameworks is smaller than the resolution of the input images to sub-networks in the baseline HoloNet. Therefore, the hologram generation rate of our frameworks is faster than that of the naïve HoloNet, as shown in Table 1 in the main paper.

Similar modifications are also made in CCNNs. Specifically, we not only modify the input and output channels but also further adjust the number of convolutional layers to maintain maximum channel number consistent with the baseline CCNNs. The specific results are shown in Table 3 and Table 4.

Furthermore, we provide the detailed structure of the proposed lightweight super-resolution (SR) network in Table 5. Finally, to demonstrate that our method, like other divide- and-conquer algorithms, can be implemented in a recursive form, we design a pyramid framework to synthesize large-scale high-quality holograms to illustrate this point, as shown in Fig. 2.

1.2 Numerical analysis

To provide a more comprehensive comparison of the proposed frameworks and the baseline frameworks in terms of computational complexity, we further analyze the network parameters and floating-point operations per second (FLOPs) of various methods. The quantitative results are shown in Table 6 and Table 7. It can be clearly observed that, in contrast to previous methods [3, 5], our proposed divide-and-conquer strategy does not reduce, and in some cases even increases, the network parameters to ensure the generation of high-quality holograms. Furthermore, during the divide-and-conquer stage, our method also accelerates the generation of holograms.

2 Additional Simulation Results

In this section, we provide more visual results of the 8K holograms and reconstructed images in Fig. 2. To the best of our knowledge, it is the first demonstration of 8K hologram generation and reconstruction on single consumer-grade GPU.

Table 6: **The overall network parameters and Floating Point Operations (Flops) for HoloNet and HoloNet w/ ours.** "Params" represents the overall parameters of the network.

Methods	HoloNet			
	1080p		4K	
	Params (M)	Flops (G)	Params (M)	Flops (G)
w/o	2.87	333	Out-of-memory	
w/ ×2	3.01	158	2.91	425
w/ ×4	3.11	53	2.96	132
w/ pyramid	3.17	170	2.97	261

Table 7: **The overall network parameters and Floating Point Operations (Flops) for CCNNs and CCNNs w/ ours.** "Params" represents the overall parameters of the network.

Methods	CCNNs			
	4K		8K	
	Params (K)	Flops (G)	Params (K)	Flops (G)
w/o	42.3	29	42.3	117
w/ ×4	112.7	71	112.7	284

References

[1] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein. Neural holography with camera-in-the-loop training. *ACM Trans. Graph.*, 39(6), nov 2020.

[2] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

[3] J. Wu, K. Liu, X. Sui, and L. Cao. High-speed computer-generated holography using an autoencoder-based deep neural network. *Opt. Lett.*, 46(12):2908–2911, Jun 2021. doi: 10.1364/OL.425485

[4] K. Zhang, D. Li, W. Luo, W. Ren, B. Stenger, W. Liu, H. Li, and M.-H. Yang. Benchmarking ultra-high-definition image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14769–14778, 2021.

[5] C. Zhong, X. Sang, B. Yan, H. Li, D. Chen, X. Qin, S. Chen, and X. Ye. Real-time high-quality computer-generated hologram using complex-valued convolutional neural network. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

*Z. Dong and J. Jia are joint first authors and contribute equally.

†Y. Ling is the corresponding author.

Email:{d_zhenxing, jjd1123, yan.li, yuye.ling}@sjtu.edu.cn.

Table 1: **Detailed modification on the phase generator of HoloNet.** The underlined number and the **bolded number** represent the skip connection and the modification we made respectively. Here, the skip connections employ a concatenate operation. We take an input image with a definition of 4K as an example.

Index	HoloNet w/o			HoloNet w/ $\times 4$			HoloNet w/ $\times 2$		
	I/O	InpShape	OutShape	I/O	InpShape	OutShape	I/O	InpShape	OutShape
1	1/16	2176×3840	2176×3840	**16/16**	544×960	544×960	**4/16**	1088×1920	1088×1920
2	16/16	2176×3840	2176×3840	16/16	544×960	544×960	16/16	1088×1920	1088×1920
3	16/32	2176×3840	1088×1920	16/32	544×960	272×480	16/32	1088×1920	544×960
4	32/32	1088×1920	1088×1920	32/32	272×480	272×480	32/32	544×960	544×960
5	32/64	1088×1920	544×960	32/64	272×480	136×240	32/64	544×960	272×480
6	64/64	544×960	544×960	64/64	136×240	136×240	64/64	272×480	272×480
7	64/128	544×960	272×480	64/128	136×240	68×120	64/128	272×480	136×240
8	128/128	272×480	272×480	128/128	68×120	68×120	128/128	136×240	136×240
9	128/128	272×480	136×240	128/128	68×120	34×60	128/128	136×240	68×120
10	128/128	136×240	272×480	128/128	34×60	68×120	128/128	68×120	136×240
11	128/128	272×480	272×480	128/128	68×120	68×120	128/128	136×240	136×240
12	256/64	272×480	544×960	256/64	68×120	136×240	256/64	136×240	272×480
13	64/64	544×960	544×960	64/64	136×240	136×240	64/64	272×480	272×480
14	128/32	544×960	1088×1920	128/32	136×240	272×480	128/32	272×480	544×960
15	32/32	1088×1920	1088×1920	32/32	272×480	272×480	32/32	544×960	544×960
16	64/16	1088×1920	2176×3840	64/16	272×480	544×960	64/16	544×960	1088×1920
17	16/16	2176×3840	2176×3840	16/16	544×960	544×960	16/16	1088×1920	1088×1920
18	32/1	2176×3840	2176×3840	**32/16**	544×960	544×960	**32/4**	1088×1920	1088×1920

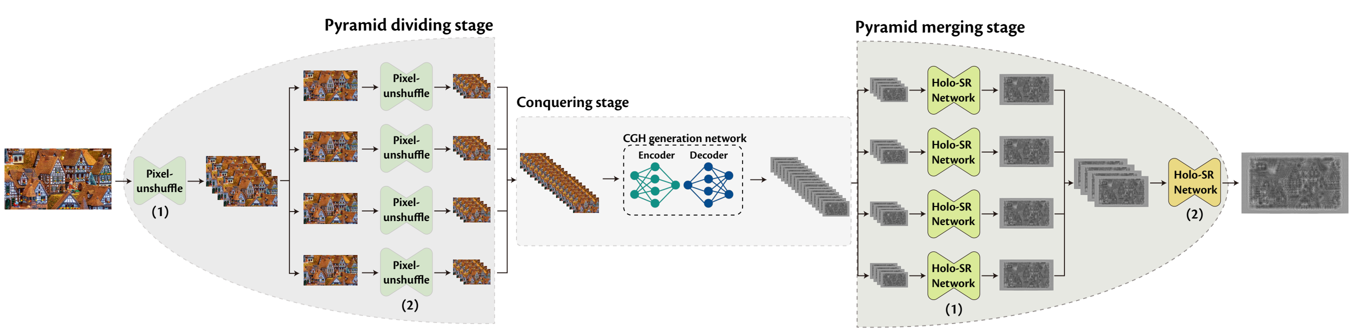

Figure 1: Overall architecture of the pyramid framework.

Table 2: **Detailed modification on the phase encoder of HoloNet.** The $\underline{\text{underlined number}}$ and the **bolded number** represent the skip connection and the modification we made respectively. Here, the skip connections employ a concatenate operation. We take an input image with a definition of 4K as an example.

Index	**HoloNet w/o**			**HoloNet w/ ×4**			**HoloNet w/ ×2**		
	I/O	InpShape	OutShape	I/O	InpShape	OutShape	I/O	InpShape	OutShape
1	1/16	2176×3840	2176×3840	**32/16**	544×960	544×960	**8/16**	1088×1920	1088×1920
2	16/16	2176×3840	2176×3840	16/16	544×960	544×960	16/16	1088×1920	1088×1920
3	16/32	2176×3840	1088×1920	16/32	544×960	272×480	16/32	1088×1920	544×960
4	32/32	1088×1920	1088×1920	32/32	272×480	272×480	32/32	544×960	544×960
5	32/64	1088×1920	544×960	32/64	272×480	136×240	32/64	544×960	272×480
6	64/64	544×960	544×960	64/64	136×240	136×240	64/64	272×480	272×480
7	64/128	544×960	272×480	64/128	136×240	68×120	64/128	272×480	136×240
8	128/128	272×480	272×480	128/128	68×120	68×120	128/128	136×240	136×240
9	128/128	272×480	136×240	128/128	68×120	34×60	128/128	136×240	68×120
10	128/128	136×240	272×480	128/128	34×60	68×120	128/128	68×120	136×240
11	128/128	272×480	272×480	128/128	68×120	68×120	128/128	136×240	136×240
12	$\underline{256}$/64	272×480	544×960	$\underline{256}$/64	68×120	136×240	$\underline{256}$/64	136×240	272×480
13	64/64	544×960	544×960	64/64	136×240	136×240	64/64	272×480	272×480
14	$\underline{128}$/32	544×960	1088×1920	$\underline{128}$/32	136×240	272×480	$\underline{128}$/32	272×480	544×960
15	32/32	1088×1920	1088×1920	32/32	272×480	272×480	32/32	544×960	544×960
16	$\underline{64}$/16	1088×1920	2176×3840	$\underline{64}$/16	272×480	544×960	$\underline{64}$/16	544×960	1088×1920
17	16/16	2176×3840	2176×3840	16/16	544×960	544×960	16/16	1088×1920	1088×1920
18	$\underline{32}$/1	2176×3840	2176×3840	**$\underline{32}$/16**	544×960	544×960	**$\underline{32}$/4**	1088×1920	1088×1920

Table 3: **Detailed modification on the phase generator of CCNNs.** The $\underline{\text{underlined number}}$ and the **bolded number** represent the skip connection and the modification we made respectively. Here, the skip connections employ an additive operation. We take an input image with a definition of 4K as an example.

Index	**CCNNs w/o**			**CCNNs w/ ×4**		
	I/O	InpShape	OutShape	I/O	InpShape	OutShape
1	1/4	2176×3840	1088×1920	**16/32**	544×960	272×480
2	4/8	1088×1920	544×960	**$\underline{32}$/16**	272×480	544×960
3	8/16	544×960	272×480	/	/	/
4	16/32	272×480	136×240	/	/	/
5	32/16	136×240	272×480	/	/	/
6	$\underline{16}$/8	272×480	544×960	/	/	/
7	$\underline{8}$/4	544×960	1088×1920	/	/	/
8	$\underline{4}$/1	1088×1920	2176×3840	/	/	/

Table 4: **Detailed modification on the phase encoder of CCNNs.** The underlined number and the **bolded number** represent the skip connection and the modification we made respectively. Here, the skip connections employ an additive operation. We take an input image with a definition of 4K as an example.

Index	CCNNs w/o			CCNNs w/ ×4		
	I/O	InpShape	OutShape	I/O	InpShape	OutShape
1	1/4	2176×3840	1088×1920	**16/32**	544×960	272×480
2	4/8	1088×1920	544×960	**32/16**	272×480	544×960
3	8/16	544×960	272×480	/	/	/
4	16/8	272×480	544×960	/	/	/
5	8/4	544×960	1088×1920	/	/	/
6	4/1	1088×1920	2176×3840	/	/	/

Table 5: **Detailed components of the proposed hologram SR network.** We take the hologram SR network utilized in CCNNs w/ ×4 with a definition of 4K as an example. Specifically, "GRN" represents the Global Response Normalization layer [2]. And "Conv2d" represents the convolution layer. For iteration manner, we use two SR networks with the same architecture.

Repetition Times	Index	Name	Kernel	Stride	I/O	InpShape	OutShape
1	1	Conv2d + GRN	3	1	16/24	544×960	544×960
2	2	Conv2d + LeakyReLU (0.1)	3	1	24/12	544×960	544×960
	3	Conv2d + LeakyReLU (0.1)	3	1	12/12	544×960	544×960
	4	Conv2d + Sigmoid	3	1	12/24	544×960	544×960
	5	Conv2d + LeakyReLU (0.1)	3	1	24/30	544×960	544×960
	6	Conv2d	3	1	30/24	544×960	544×960
1	12	Conv2d	3	1	24/16	544×960	544×960
1	13	PixelShuffle + Hardtanh	/	/	16/1	544×960	2176×3840

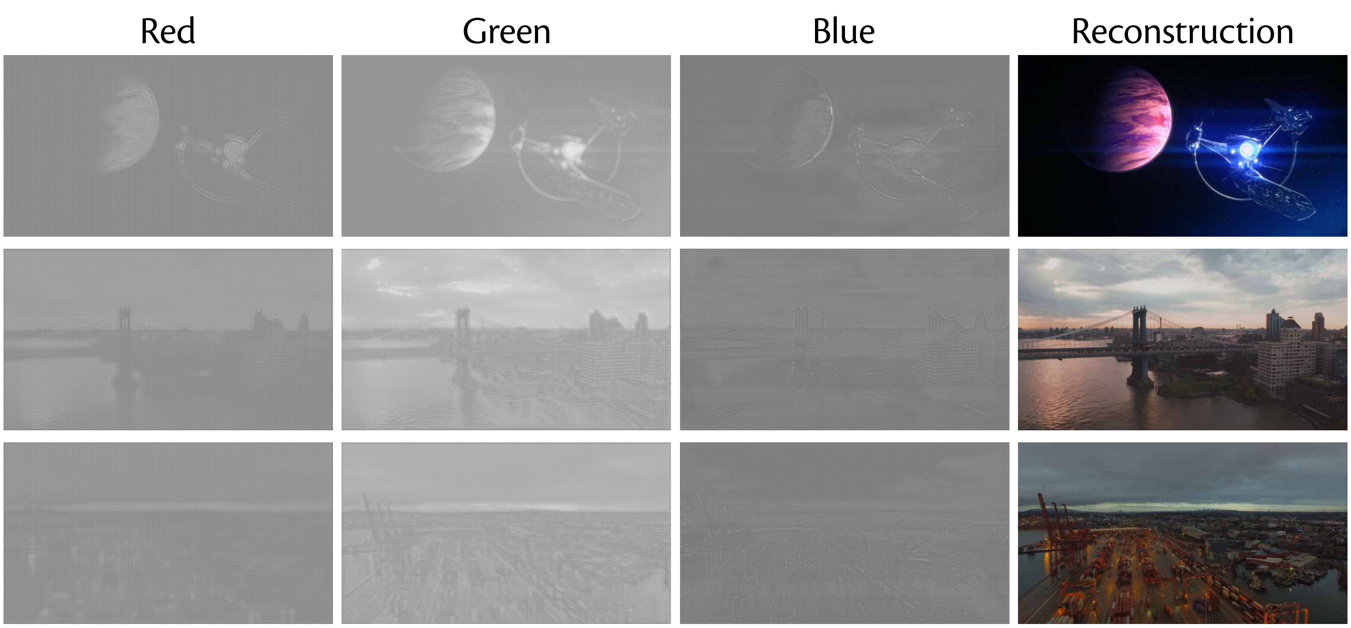

Figure 2: 8K generated hologram and reconstructed image from hologram generated by CCNNs w/ **ours**. Images come from wall.alphacoders.com and UHD8K [4] dataset.