Supplementary Materials to Learning Dynamic Interpolation for Extremely Sparse Light Fields with Wide Baselines

In this document, we provide the supplementary information for the ICCV 2021 paper titled with "Learning Dynamic Interpolation for Extremely Sparse Light Fields with Wide Baselines". The rest of the document is organized as follows:

- Section 1 visually presents the dynamic weights learned by our method.
- Section 2 gives the quantitative comparisons of different methods over narrow-baseline light field (LF) datasets.
- Section 3 gives more visual results on the MPI dataset [1].
- Section 4 gives the detailed network architectures of convolutional neural networks and multilayer perceptrons (MLPs) of our framework.

1. Learned Dynamic Weights

We presented the learned weights for synthesizing pixels of the novel views, taking the foreground and background pixels around the occlusion boundary as examples. From Fig. 1 and Fig. 2, we can observe that (1) our method assigns large weight values to the correspondences of the synthesized pixels, except for the occluded ones, which validates the learned weights implicitly incorporate the geometry information of the novel view; and (2) our method well handles the occlusion boundary by assigning large weights only on the foreground or background neighbors, which validates our method can learn content-adaptive interpolation weights. Additionally, Fig. 2 shows that although the correspondence of the synthesized pixel in the right input view is occluded by the foreground object, our method can still adaptively assign large weight to a background pixel which has the same intensity as the synthesized one, demonstrating the potential of our method to handle the challenging problem of occlusions.

2. Quantitative Comparisons on Narrow-baseline Datasets

We quantitatively compared different methods over two narrow-baseline LF datasets, including 30 real-world LF images from the Kalantari Lytro dataset [4], denoted as 30 scenes, and 24 synthetic LF images from the HCI dataset [2]. All methods directly used the models trained on the wide-baseline dataset, i.e., Inria Sparse LF dataset [5]. The input disparity ranges and quantitative results of different methods are shown in Table 1, where it can be observed that Ours (RAFT) outperforms other methods on both two narrow-baseline LF datasets.

Table 1. Quantitative comparisons (PSNR/SSIM) of different methods over the 30 scenes [4] and HCI [2] datasets.

Dataset	Disparity range	Baseline	Kalantari *et al.* [4]	Wu *et al.* [6]	Jin *et al.* [3]	Ours (PWCNet)	Ours (RAFT)
30 scenes	[-4.0, 4.0]	38.47/0.973	22.27/0.718	41.31/0.983	41.63/0.985	41.34/0.983	**42.28/0.986**
HCI	[-16.0, 16.0]	37.13/0.959	27.64/0.768	39.31/0.966	39.28/0.966	39.88/0.971	**41.25/0.977**

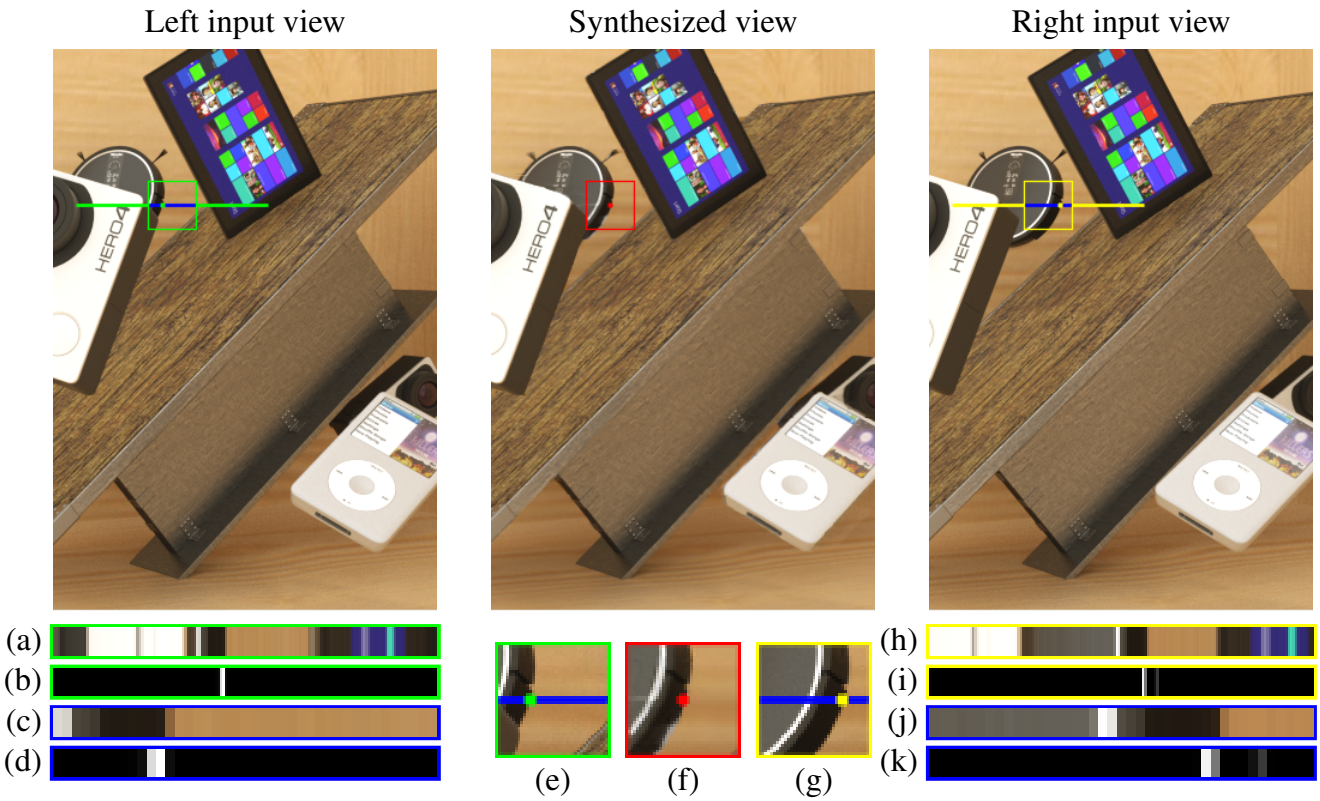

Figure 1. Neighborhoods and learned dynamic weights for synthesizing the **foreground** pixel around the occlusion boundary. The synthesized pixel is highlighted with a red dot in the synthesized view. The pixels corresponding to the maximum weight values in the neighborhoods are highlighted with green and yellow dots in left and right input views, respectively. Below input views, from top to bottom: (a) and (h) are the zoom-in of the neighbors highlighted with green and yellow straight lines, respectively; (b) and (i) are the learned weights corresponding to (a) and (h), respectively; (c) and (j) are the zoom-in of the neighbors highlighted with blue straight lines; and (d) and (k) are learned dynamic weights corresponding to (c) and (j), respectively. The regions with red, green, and yellow frames around the synthesized pixel and maximum-weight pixels are also zoomed in, i.e., (e), (f), and (g), for better visualization.

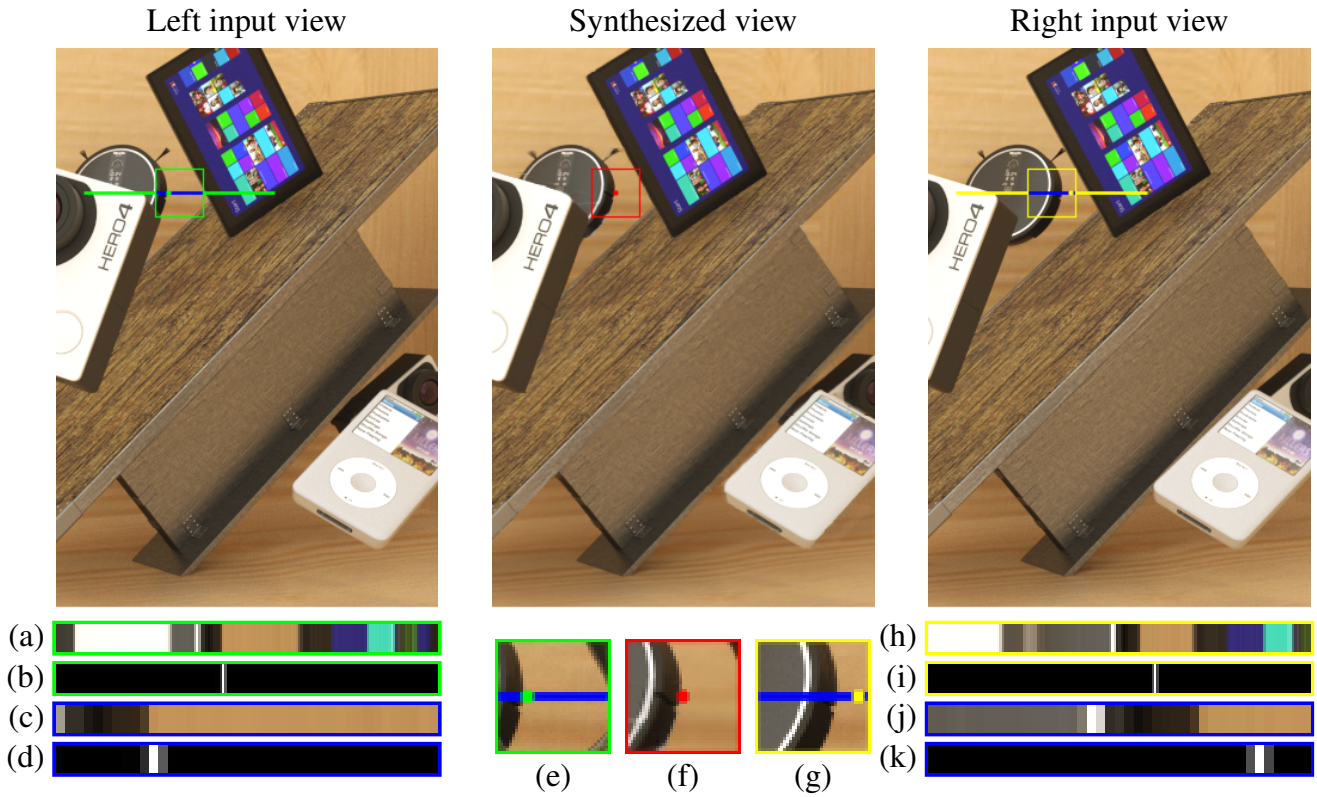

Figure 2. Neighborhoods and learned dynamic weights for synthesizing the **background** pixel around the occlusion boundary. The synthesized pixel is highlighted with a red dot in the synthesized view. The pixels corresponding to the maximum weight values in the neighborhoods are highlighted with green and yellow dots in left and right input views, respectively. Below input views, from top to bottom: (a) and (h) are the zoom-in of the neighbors highlighted with green and yellow straight lines, respectively; (b) and (i) are the learned weights corresponding to (a) and (h), respectively; (c) and (j) are the zoom-in of the neighbors highlighted with blue straight lines; and (d) and (k) are learned dynamic weights corresponding to (c) and (j), respectively. The regions with red, green, and yellow frames around the synthesized pixel and maximum-weight pixels are also zoomed in, i.e., (e), (f), and (g), for better visualization.

3. Visual Results

We provide more visual comparisons of reconstructed LFs from different methods over the MPI dataset [1]. As shown in Fig. 3, it can be observed that our method can reconstruct LFs with higher visual quality than other methods, which further demonstrates the advantages of our method.

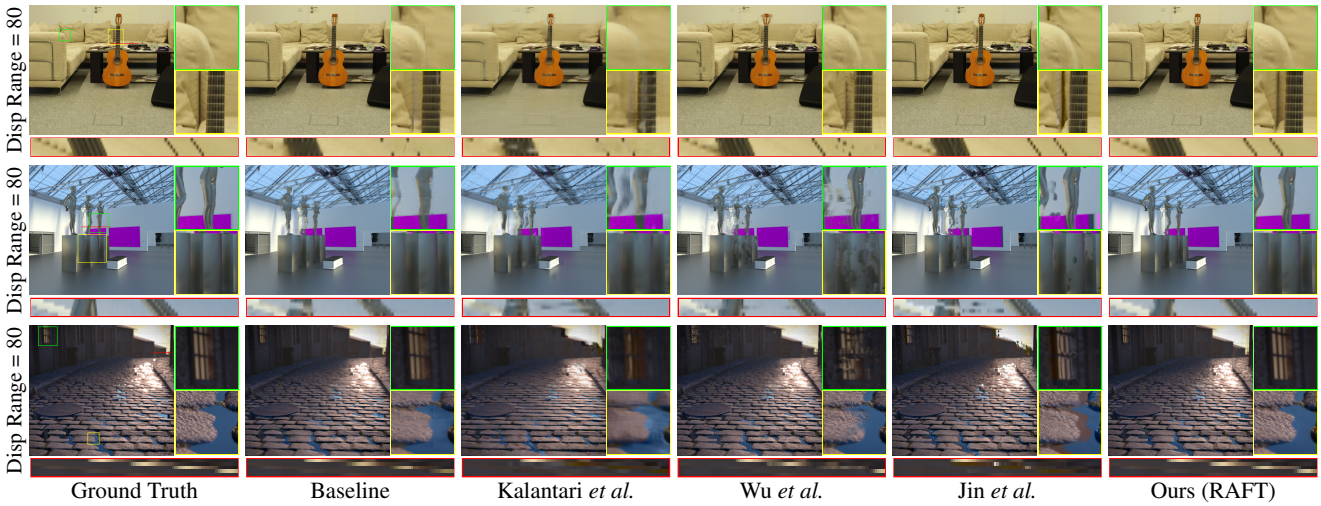

Figure 3. Visual comparisons of reconstructed LFs from different methods over the MPI dataset [1]. The input disparity range of each LF is shown on the left.

4. Detailed Network Architectures

We provide the detailed network architectures of the content embedding network $f_c(\cdot)$, geometry-based spatial refinement network $f_r(\cdot)$, dynamic weights MLP $f_w(\cdot)$, and confidence MLP $f_b(\cdot)$ in Tables 2, 3, 4, and 5, respectively.

Table 2. The detailed architecture of content embedding network $f_c(\cdot)$.

input size	layer	kernel size	output size
32×192	conv_0	3×3, 64 relu	32×192
32×192	res_block 1	3×3, 64 relu 3×3, 64	32×192
32×192	res_block 2	3×3, 64 relu 3×3, 64	32×192
32×192	res_block 3	3×3, 64 relu 3×3, 64	32×192
32×192	res_block 4	3×3, 64 relu 3×3, 64	32×192
32×192	conv_last	3×3, 64	32×192

Table 3. The detailed architecture of geometry-based spatial refinement network $f_r(\cdot)$.

input size	layer	kernel size	output size
32×32	conv_0	3×3, 64 relu	32×32
32×32	res_block 1	3×3, 64 relu 3×3, 64	32×32
32×32	res_block 2	3×3, 64 relu 3×3, 64	32×32
32×32	res_block 3	3×3, 64 relu 3×3, 64	32×32
32×32	res_block 4	3×3, 64 relu 3×3, 64	32×32
32×32	conv_last	3×3, 1	32×32

Table 4. The detailed architecture of dynamic weights MLP $f_w(\cdot)$.

input size	layer	output size
67	linear relu	67
67	linear relu	67
67	linear relu	67
67	linear relu	67
67	linear relu	67
67	linear	1

Table 5. The detailed architecture of confidence MLP $f_b(\cdot)$.

input size	layer	output size
67	linear relu	67
67	linear relu	67
67	linear relu	67
67	linear relu	67
67	linear relu	67
	reshape	
161*67	linear	1

References

[1] Vamsi Kiran Adhikarla, Marek Vinkler, Denis Sumin, Rafał Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Towards a quality metric for dense light fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision (ACCV)*, pages 19–34. Springer, 2016.

[3] Jing Jin, Junhui Hou, Jie Chen, Huanqiang Zeng, Sam Kwong, and Jingyi Yu. Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[4] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics*, 35(6):1–10, 2016.

[5] Jinglei Shi, Xiaoran Jiang, and Christine Guillemot. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Transactions on Image Processing*, 28(12):5867–5880, 2019.

[6] Gaochang Wu, Yebin Liu, Qionghai Dai, and Tianyou Chai. Learning sheared epi structure for light field reconstruction. *IEEE Transactions on Image Processing*, 28(7):3261–3273, 2019.