DyBluRF: Dynamic Neural Radiance Fields from Blurry Monocular Video

Supplementary Material

1. Appendix

We provide the following content in this supplementary:

- Detailed description of disocclusion weights;
- Principles and technical details of EVC;
- Detailed description of scene flow constrains $\mathcal{L}_{\text{sf}}$;
- Final loss;
- More information on the dataset;
- Detailed quantitative and qualitative results;
- Results on sharp inputs;
- Limitations.

A. Disocclusion weights

Recall in Eq. 7 of our main paper, the dynamic MLP additionally outputs a disocclusion weight $\mathcal{W}_t = (w_{\text{fw}}, w_{\text{bw}})$. The weights w_{fw} and $w_{\text{bw}} \in [0, 1]$ represent the probability of occlusion for a spatial point. They determine the confidence regarding occlusion occurring from a specific timestamp in the current frame to the corresponding timestamp in adjacent frames. Certainly, a value closer to 0 signifies minimal chances of the point being occluded during that interval. Conversely, a value closer to 1 indicates a higher likelihood of the point experiencing occlusion or disocclusion. Specifically, for a timestamp t_l^i from frame i, where $l \in \{1, \cdots, n\}$ represents the specific timestamp in an exposure time, the disocclusion weights are denoted as $\mathcal{W}_{t_l^i} = (w_l^{i\to i+1}, w_l^{i\to i-1})$. To obtain the disocclusion weight map in 2D plane, the weight along the ray $\mathbf{r}_l^i$ is used for volume rendering with opacity from adjacent frames:

$$\mathbf{W}_l^{j\to i}(\mathbf{r}_l^i) = \int_{s_n}^{s_f} T_{t_l^j}(s)\, \sigma_{t_l^j}(\mathbf{r}_l^{i\to j}(s))\, (1 - w_l^{i\to j}(\mathbf{r}_l^i(s)) ds\,, \tag{1}$$

where $j \in \mathcal{N}(i) = \{i+1, i-1\}$ denotes the adjacent frames of frame i, $\mathbf{r}_l^{i\to j}$ represents the warped ray mentioned by Eq. 14 in the main paper.

For each timestamp within the exposure time, the disocclusion weight map $\mathbf{W}_l^{j\to i}(\mathbf{r})$ is computed using Eq. 1. Subsequently, we average these maps to yield the motion disocclusion weight $\mathbf{W}^{j\to i}(\mathbf{r})$ for cross-time rendering:

$$\mathbf{W}^{j\to i}(\mathbf{r}) = \frac{1}{n} \sum_{l=1}^{n} \mathbf{W}_l^{j\to i}(\mathbf{r})\,. \tag{2}$$

B. Extreme Value Constraints (EVC)

In this section, we will introduce the operational principles and technical details of EVC. In Fig. 4 of our main paper, we highlight an issue encountered while predicting depth and optical flow from blurry inputs. The depth prediction network and optical flow prediction network often mistakenly interpret the blurry edges in the RGB image as foreground, resulting in inaccuracies in data priors. To enable model learning accurate scene geometry by using these inaccurate data priors, we propose the EVC data prior constraint method.

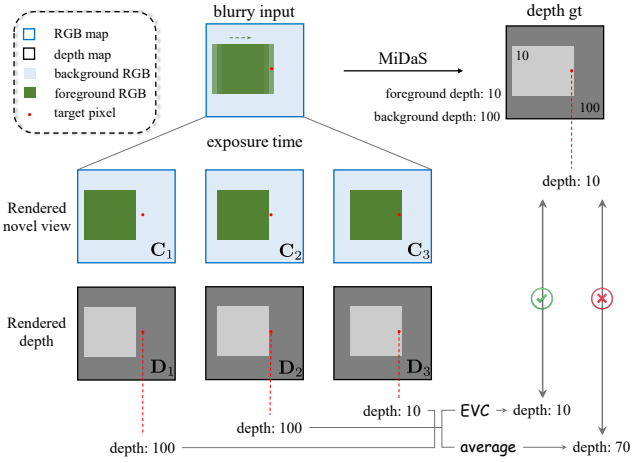

Figure S1. **Principle of EVC.**

Fig. S1 provides an illustration of a simulation experiment on the working principle of EVC. We suppose that each exposure time is represented by three sharp images (*i.e.*, $n = 3$), with only one green square foreground moving from left to right in the scene, while the rest is a blue background. Given an input blurry frame, the model produces 3 RGB images and depth maps within the exposure time. We consider a pixel position in blurry edges of the input frame, where the pixel changes the affiliation from the background (in $\mathbf{C}_1$ and $\mathbf{C}_2$) to the foreground (in $\mathbf{C}_3$). To ease the discussion, we set the depth of the foreground to be 10 while that of the background to be 100 respectively. An intuitive practice to simulate the depth of input blurry frames may be averaging the three depth values to a value of 70. However, due to the mistaken interpretation of blurry edges, MiDaS [8] identifies the red point as the foreground, leading to its predicted depth value of 10. Therefore the intuitive constraint cannot work for our method, manifesting as 'foreground-fatter' depth map predictions as shown in the first row of Fig. 4(b) in the main paper. In order to accommodate the depth prediction results, we should recognize a position as on the foreground (the green square), once it is covered by the foreground in a certain timestamp during the exposure time. This can be expressed as choosing the minimum depth value at the red-pointed position among the three sharp images (as EVC does). This ap-

proach enables learning accurate scene geometry and rendering a sharp depth map, as shown in the second row of Fig. 4(b) in the main paper.

The process of data prior constraint for optical flow closely resembles that of depth. The model generates an optical flow map for each timestamp. This map represents the scene motion from the timestamp of current exposure time to the corresponding timestamp of adjacent exposure time. However, in contrast to depth where foreground values are typically smaller than background ones, optical flow for the foreground often exceeds that of the background. According to the principle and process of EVC, we use the maximum optical flow as the simulated blurry optical flow and compare it with the ground truth predicted by RAFT [10]. In practice, we obtain the optical flow by projecting the predicted 3D scene flow onto a 2D plane.

C. Scene flow modeling details

Recall in Sec. 3.4 , $\mathcal{L}_{\text{sf}}$ is used as a regularization loss for the scene flow calculated by Eq. 13 in the main paper. $\mathcal{L}_{\text{sf}}$ consists of three components: scene flow cycle consistency, spatial-temporal smoothness, and minimal scene flow.

Scene flow cycle consistency enforces the forward scene flow $f_{\mathbf{x}}^{i,l}(j)$ of sampled 3D points at timestamps t_l^i consistent with the backward scene flow $f_{\mathbf{x}}^{j,l}(i)$ at the corresponding location at timestamps t_l^j:

$$\mathcal{L}_{\text{cyc}} = \sum_{j\in\{i\pm1\}} \sum_{l=1}^{n} (1 - w_l^{i\to j}) \| f_{\mathbf{x}}^{i,l}(j) + f_{\mathbf{x}}^{j,l}(i) \|_1 \,, \quad (3)$$

where $w_l^{i\to j}$ denotes the disocclusion weights used to reduce consideration of occlusion points.

The spatial-temporal smoothness is designed to maintain continuity in scene flow both spatially and temporally. To achieve spatial smoothness, we encourage consistency between the scene flows of adjacent spatial points. Specifically, we compute L1 loss between scene flows sampled at two neighboring spatial points along the ray $\mathbf{r}_l^i$. For temporal smoothness, we minimize the sum of forward and backward scene flow for each sampled spatial point to ensure the smoothness of predicted DCT trajectories. The spatial-temporal smoothness can be expressed as:

$$\begin{aligned} \mathcal{L}_{\text{smooth}} = & \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \sum_{j\in\{i\pm1\}} \sum_{l=1}^{n} \| f_{\mathbf{x}}^{i,l}(j) - f_{\mathbf{y}}^{i,l}(j) \|_1 \\ & + \frac{1}{2} \sum_{l=1}^{n} \| f_{\mathbf{x}}^{i,l}(i+1) + f_{\mathbf{x}}^{i,l}(i-1) \|_2^2 \,, \end{aligned} \quad (4)$$

where the first term denotes spatial smoothness and the second term denotes temporal smoothness. $\mathcal{N}(\mathbf{x})$ represents the neighboring spatial points of point $\mathbf{x}$ along the ray $\mathbf{r}_l^i$.

Finally, due to minor scene changes between adjacent frames, we additionally apply a minimal scene flow constraint to minimize the predicted 3D scene flow:

$$\mathcal{L}_{\text{min}} = \sum_{j\in\{i\pm1\}} \sum_{l=1}^{n} \| f_{\mathbf{x}}^{i,l}(j) \|_1 \,. \quad (5)$$

D. Final loss

Considering the RGB image rendering, temporal consistency of dynamic scenes, data-driven constraints, and scene flow modeling, the final training loss of our method is:

$$\mathcal{L} = \mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{cross}} + \lambda_{\text{data}}\mathcal{L}_{\text{data}} + \lambda_{\text{sf}}\mathcal{L}_{\text{sf}} \,, \quad (6)$$

where

$$\begin{aligned} \mathcal{L}_{\text{RGB}} &= \mathcal{L}_{\text{RGB}}^{\text{cb}} + \mathcal{L}_{\text{RGB}}^{\text{dy}} + \lambda_{\text{st}}\mathcal{L}_{\text{RGB}}^{\text{st}} \,, \\ \mathcal{L}_{\text{sf}} &= \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{min}} \,, \end{aligned} \quad (7)$$

λ_{data}, λ_{sf}, λ_{st}, λ_{cyc} denote the weights for $\mathcal{L}_{\text{data}}$, $\mathcal{L}_{\text{sf}}$, $\mathcal{L}_{\text{RGB}}^{\text{st}}$ and $\mathcal{L}_{\text{cyc}}$, respectively. During training, λ_{data} multiplies 0.1 every $50k$ iterations to prevent the model from overfitting to data priors.

E. Dataset

We conduct experiments using dynamic scenes from the Stereo Blur Dataset [15]. Similar to most datasets in image deblurring [5, 9], the Stereo Blur Dataset generates motion blur by averaging a sharp high frame rate sequence. In practice, this dataset is captured using a ZED stereo camera capable of acquiring high frame rate (60fps) image sequences of dynamic scenes at a resolution of 1280×720. However, because the frame rate is still not high enough, direct averaging may lead to some unrealistic artifacts. Therefore, the dataset employs a video frame interpolation method [6] to increase the frame rate to 480fps. Subsequently, averaging is performed on different numbers $(17, 33, 49)$ of consecutive frames to generate motion blur, with the sharp center frame among the consecutive frames serving as the ground truth for the blurry image.

However, not all scenes in the Stereo Blur Dataset can be used in our work. That is because of two issues: 1) Many scenes within the dataset are static and lack moving objects. 2) Several scenes have minimal camera motion, resulting in a lack of motion blur caused by camera movement and insufficient motion parallax to obtain camera parameters. Therefore, we select 6 dynamic scenes from the Stereo Blur Dataset that are tailored for NeRF-based methods. These scenes encompass both camera and object motion blur, showcasing varied size object movements like playing seesaw, walking, and skating. We employ COLMAP to acquire camera parameters from the image sequences and downsample the image resolution to 512×288 for experiments. Similar to NSFF, we process the input blurry image sequences through the MiDaS to obtain depth maps,

Methods	Sailor			Seesaw			Street			Children			Skating			Basketball		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
BAD-NeRF [11]	16.96	0.631	0.333	20.58	0.795	0.220	20.27	0.670	0.190	18.10	0.650	0.386	19.08	0.691	0.345	17.99	0.731	0.271
HyperNeRF [7]	18.56	0.743	0.275	20.25	0.779	0.182	19.99	0.662	0.137	21.36	0.762	0.279	19.52	0.702	0.319	21.21	0.818	0.136
DVS [1]	22.32	0.810	0.247	18.14	0.775	0.158	19.06	0.669	0.176	24.00	0.823	0.307	26.18	0.907	0.124	24.10	0.880	0.143
NSFF [3]	19.06	0.726	0.290	19.92	0.807	0.178	23.42	0.800	0.121	24.55	0.846	0.259	**27.96**	**0.923**	0.116	25.06	0.903	0.122
RoDynRF [4]	12.69	0.584	0.317	23.71	**0.894**	0.116	19.80	0.719	0.161	15.02	0.609	0.382	21.66	0.831	0.160	22.82	0.850	0.166
DyBluRF (ours)	**23.50**	**0.860**	**0.115**	**24.56**	0.882	**0.075**	**26.88**	**0.906**	**0.068**	**25.57**	**0.884**	**0.092**	27.94	0.917	**0.072**	**25.28**	**0.920**	**0.050**

Table S1. **Quantitative comparisons for every scene against all dynamic NeRF baselines.** The best performance is **boldfaced**, and the second is underlined.

Methods	Sailor			Seesaw			Street			Children			Skating			Basketball		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
[13] + [3]	18.96	0.728	0.252	20.19	0.827	0.149	23.39	0.790	0.125	24.88	0.864	0.182	26.91	**0.922**	**0.067**	24.70	0.903	0.084
[14] + [3]	22.60	0.805	0.150	21.31	0.856	0.135	23.06	0.791	0.114	24.27	0.853	0.195	26.88	0.917	0.078	24.64	0.903	0.076
[13] + [4]	17.16	0.703	0.227	23.63	0.892	0.090	19.75	0.727	0.149	17.41	0.665	0.282	23.22	0.869	0.110	23.38	0.868	0.114
[14] + [4]	17.28	0.772	0.225	23.85	**0.893**	0.080	19.69	0.700	0.153	18.39	0.683	0.226	22.54	0.849	0.126	23.51	0.867	0.098
DyBluRF (ours)	**23.50**	**0.860**	**0.115**	**24.56**	0.882	**0.075**	**26.88**	**0.906**	**0.068**	**25.57**	**0.884**	**0.092**	**27.94**	0.917	0.072	**25.28**	**0.920**	**0.050**

Table S2. **Quantitative comparisons for every scene against dynamic NeRF methods with blurry image preprocess.** The best performance is **boldfaced**, and the second is underlined.

use RAFT to generate optical flow maps, and employ an instance segmentation network (Mask r-cnn [2]) to derive motion masks for moving objects.

F. Detailed results

In this section, we present detailed quantitative and qualitative results of the comparative experiments in our main paper. In Tab. 1 of the main text, we have shown the average quantitative results of all baselines across the 6 scenes. Here, we will provide individual quantitative results for each scene, as depicted in Tab. S1 for blurry inputs and Tab. S2 for deblurring preprocess inputs. We also conduct comparative experiments with BAD-NeRF, which is a deblurring NeRF method designed for static scenes, in Tab. S1. Due to its inability to represent dynamic scenes, the performance of BAD-NeRF is much lower than ours. We also include detailed qualitative comparison results, as illustrated in Fig. S2 and Fig. S3. One can see that our method performs the best quantitative results in most scenes. Although our approach slightly trails NSFF in the Skating scene, the qualitative results of our method in the Skating scene are better than NSFF, regardless of whether blurry inputs or pre-processed sharp inputs, as shown in the second row of Fig. S2 and Fig. S3. We speculate that slightly lower metrics in the Skating scene of our method could be attributed to the background in the Skating scene having an extensive low-texture area with less motion blur, which cannot fully showcase the superiority of our method in handling motion blur input through metrics. However, the qualitative results demonstrate that our method outperforms all baselines, especially in handling motion blur in dynamic scenes.

From Tab. S1 and Tab. S2, we can also observe that preprocessing blurry images with 2D deblurring methods may yield inferior results compared to directly using blurry inputs. This phenomenon arises from the 2D deblurring methods disrupting the temporal consistency of scene information, causing inaccurate scene representation in NeRF. This further reflects that our approach effectively ensures spatial-temporal consistency in dynamic scenes.

G. Results on sharp inputs.

Although our method is specifically tailored for dynamic scenes with motion blur, it remains capable of achieving comparable results to existing methods when dealing with sharp inputs. We evaluate our method on the Nvidia Dynamic Scene Dataset [12] without blur, configured with $n = 1$ for sharp image input. Here, we compare with NSFF, which performs second best in Tab. 1 in the main paper. As shown in Tab. S3, our method is marginally better than NSFF even in sharp inputs, which underscores our sustained efficacy even in sharp inputs. Meanwhile, our method significantly outperforms NSFF under blurry inputs in Tab. 1 . That demonstrates the effectiveness of our method in handling both motion blur and representing dynamic scenes.

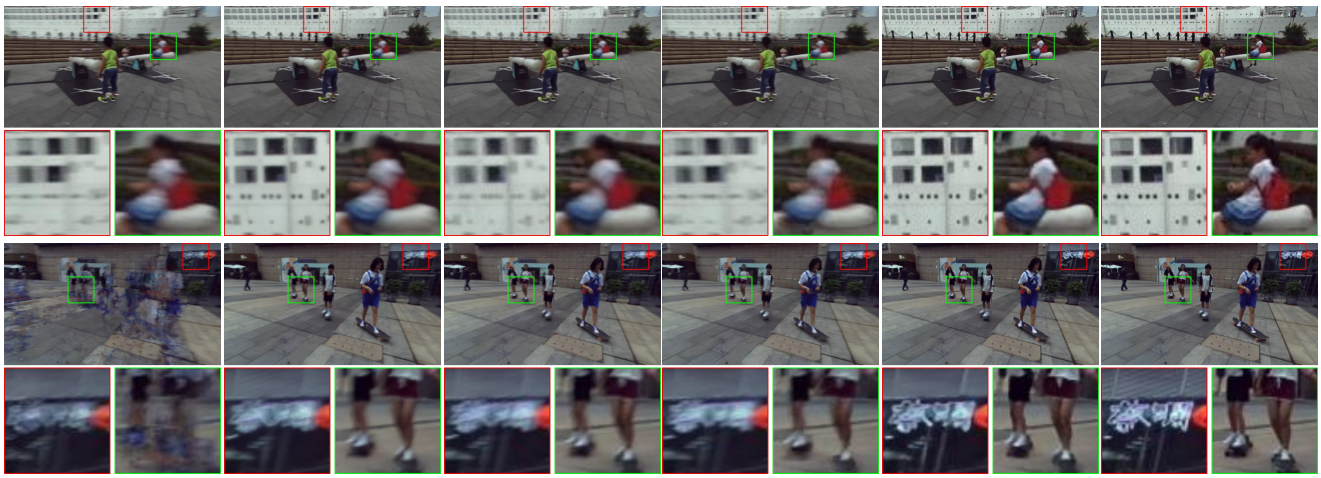

Figure S2. **Qualitative comparisons against all baselines.** Compared to existing dynamic NeRF methods, our method generates novel view images that are more faithful to the ground truth images, with less blur in both static and dynamic regions.

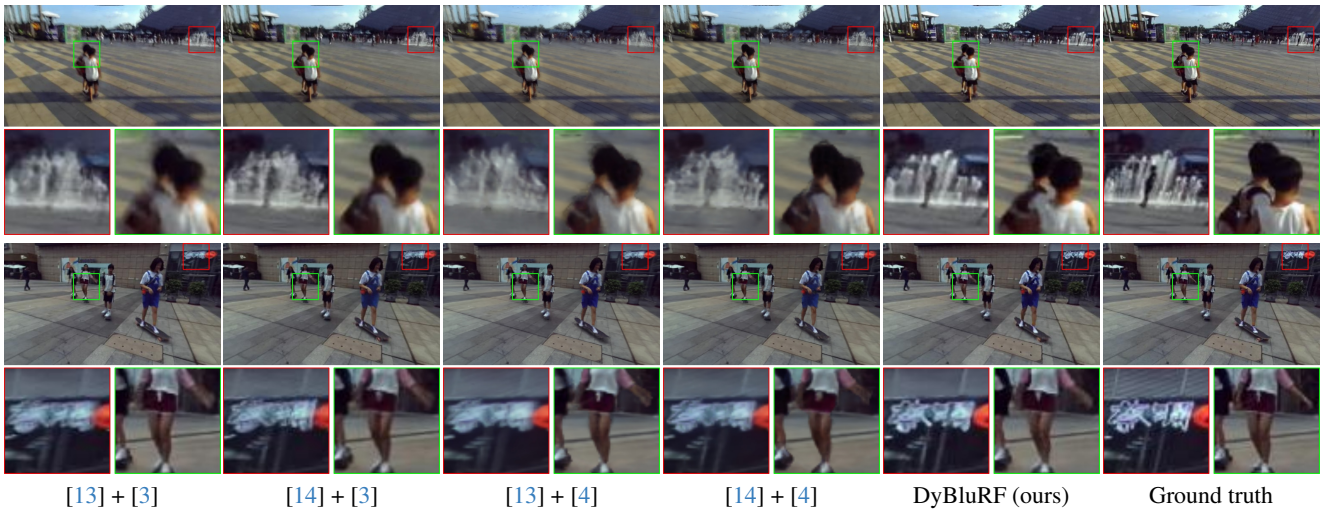

Figure S3. **Qualitative comparisons against dynamic NeRF baselines incorporated with 2D deblur method.** Even if we use preprocessed input blurry images by 2D deblur approaches to train existing dynamic NeRF methods, our method also generates more reliable novel views, with less blur in both static and dynamic regions.

Figure S4. **Limitation.** It is challenging to restore a sharp image when input blur is caused by extreme object motion (*e.g.*, the leg).

H. Limitations

Although our method can handle most of the motion blur in the input images, it might not be able to synthesize high-quality novel views when the motion blur is caused by highly complex and fast motion. As shown in Fig. S4, the rapid movement of the human leg causes extreme blurriness in the input image. This challenges our method in two aspects: firstly, excessive information loss of foreground may lead to the depth and flow prediction networks being unable to identify foreground objects, thereby affecting the representation of dynamic scenes. Secondly, the model tends to incorrectly learn the scene geometry supervised by such extremely blurry images, and gets stuck in the incorrect local minimum. In the future, we aim to explore combining explicit representations to enhance the temporal coherence of dynamic objects to solve this problem.

Methods	Balloon1			Balloon2			Jumping			Playground		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSFF [3]	**21.96**	**0.791**	0.215	24.27	0.825	0.222	24.65	0.872	0.151	21.22	0.780	0.212
DyBluRF (ours)	21.90	0.781	**0.181**	**24.92**	**0.880**	**0.117**	**26.21**	**0.901**	**0.091**	**23.64**	**0.861**	**0.113**

Methods	Skating			Truck			Umbrella			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSFF [3]	**29.29**	**0.936**	0.129	25.96	0.863	0.167	22.97	0.769	0.295	24.33	0.834	0.199
DyBluRF (ours)	28.36	0.913	**0.087**	**30.01**	**0.946**	**0.043**	**24.19**	**0.831**	**0.163**	**25.60**	**0.873**	**0.114**

Table S3. **Quantitative results on the non-blur dataset with NSFF.** The better performance is **boldface**.

References

[1] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5712–5721, 2021. 3

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 3

[3] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6508, 2021. 3, 4, 5

[4] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13–23, 2023. 3, 4

[5] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3883–3891, 2017. 2

[6] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 261–270, 2017. 2

[7] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40 (6):1–12, 2021. 3

[8] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 44(3):1623–1637, 2020. 1

[9] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1279–1288, 2017. 2

[10] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 2

[11] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. Bad-nerf: Bundle adjusted deblur neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4170–4179, 2023. 3

[12] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5336–5345, 2020. 3

[13] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5728–5739, 2022. 3, 4

[14] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9219–9228, 2021. 3, 4

[15] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy S Ren. Davanet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10996–11005, 2019. 2