

DynaMoDe-NeRF: Motion-aware Deblurring Neural Radiance Field for Dynamic Scenes (Supplementary Material)

Ashish Kumar Rajagopalan A. N.
 Indian Institute of Technology Madras, India
 ee20d006@smail.iitm.ac.in, raju@iitm.ac.in

All figures and tables in supplementary are labeled with the prefix S. The supplementary material is structured as follows:

1. BlurKernelNet architecture and non-dependence on kernel size.
2. Effect of different regularizers/loss functions
3. Comparisons with baselines
4. Effect of kernel size
5. Ray sampling strategy
6. Dataset

S1. BlurKernelNet

S1.1. Architecture

The BlurKernelNet (Fig.S1) design is motivated by the dependence of blur on the motion and depth of a particular foreground pixel. BlurKernelNet takes a 4D input $(\mathbf{v}(t) \in \mathbb{R}^3 \text{ concatenated with depth } d_r^t \in \mathbb{R}) \in \mathbb{R}^4$ and outputs weights $\mathbf{w} \in \mathbb{R}^{2m+1}$, assuming $(2m+1)$ kernel size. The output \mathbf{h}_1 , \mathbf{h}_2 of hidden layers with Exponential Linear Unit (ELU)[3] as activation are 128-dimensional vectors. The output layer activation is softmax to ensure $\sum_j w_j = 1$ and $w_j \in [0, 1]$

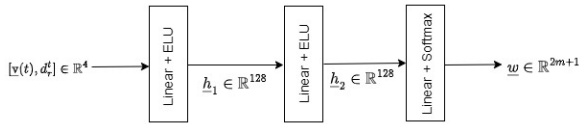


Figure S1. Block diagram of BlurKernelNet.

S1.2. Non-dependence of weights on kernel size

The weight contribution of each kernel location depends on motion and depth, rather than the kernel size which is seldom known apriori. In Fig. S2, we demonstrate that the weights estimated by BlurKernelNet are consistently distributed as the weight values are about the same at any kernel location for any given kernel size. This behavior suggests coherence within the estimated 3D radiance field and

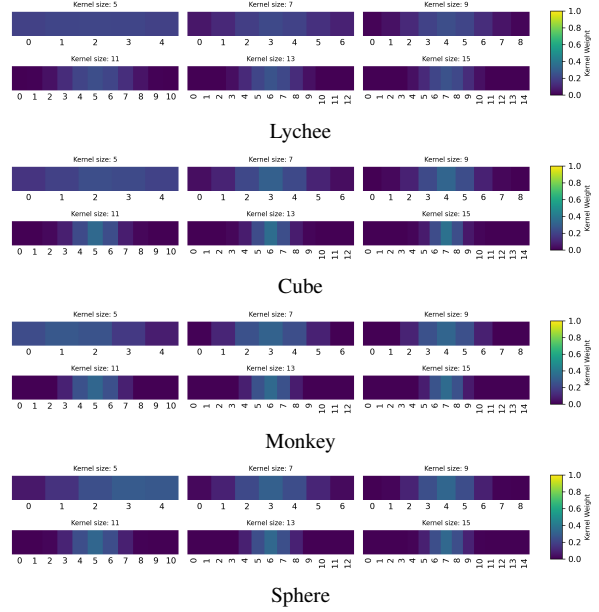


Figure S2. Weights vs. kernel size: The weight distribution is independent of kernel size.

motion. The central position in the weight distribution corresponds to the observed blurry foreground pixel, while the weights on either side correspond to the respective kernel locations away from the foreground pixel.

S2. Effect of Regularizers/Loss functions

We study each loss combination (Sec. 5 on ablations in the main paper) by analyzing their effect on the rendered novel view quality (photometry) and on the 3D motion profile.

S2.1. Effect on Rendered Novel Views

In the main paper in Table 2, we presented quantitative results, where it was observed that there is gain in dynamic region using the final loss \mathcal{L} (Eqn. 12 in main paper) as compared to $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$. In Fig. S3, we illustrate the qualitative effect of each loss combination.

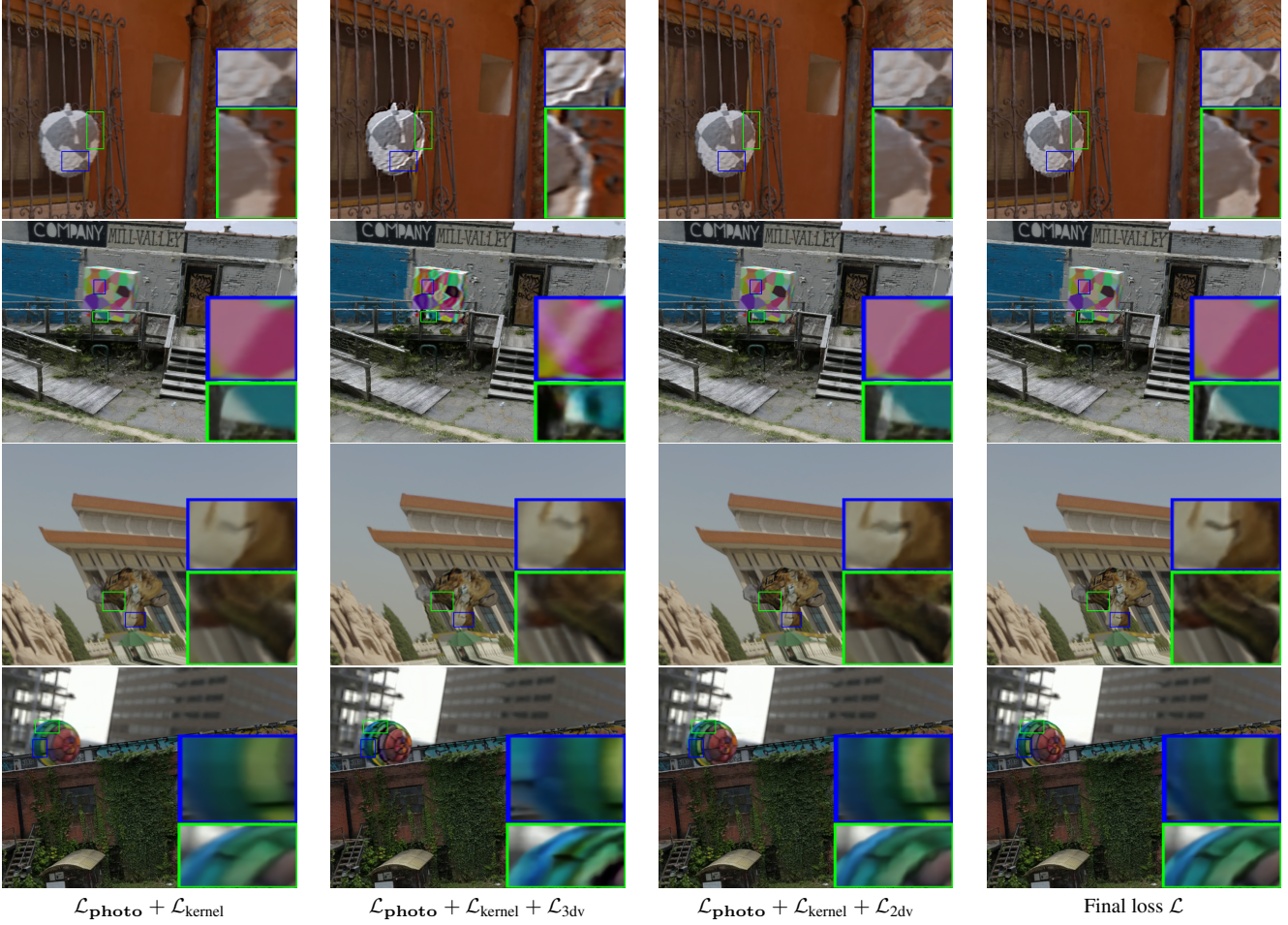


Figure S3. Qualitative analysis of different loss combinations on a test view at a single time instant for Lychee (row 1), Cube (row 2), Monkey (row 3) and Sphere (row 4). $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}}$ produces blurry image, $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{3\text{dv}}$ produces oversharpened and distorted image due to high 3D motion. $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$ and the final loss \mathcal{L} synthesize almost indistinguishable images. For further refinement, we incorporate additional motion analysis to guide the selection of optimal loss combinations, ensuring improved alignment and consistency to make DynaMoDe-NeRF motion-aware.

While we observe an almost indistinguishable difference in the rendered novel view visual quality, in addition we found that introducing to $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$ in the final loss \mathcal{L} brings consistency in the 3D motion profile.

S2.2. Effect on 3D Motion Profile

Though the loss combination $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$ synthesizes acceptable sharp novel views, we aim to maintain consistency in the 3D motion. We analyze the effect of different losses on 3D motion and report 3D motion profile similarity with ground truth (GT) motion in Table S1. We also analyze the qualitative similarity of the estimate motion with the GT motion for different loss combinations in Fig. S5. Note that the final loss \mathcal{L} consistently outperforms all other loss combinations in terms of speed profile similarity. Additionally, it achieves comparable cosine similarity (CS)

Loss	Speed CS				3D Trajectory CS			
	Lychee	Cube	Monkey	Sphere	Lychee	Cube	Monkey	Sphere
$\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}}$	0.913	0.910	0.930	0.932	0.9790	0.9656	0.9844	0.4394
$\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$	0.999	0.997	0.992	0.993	0.9980	0.9989	0.9938	0.6673
$\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{3\text{dv}}$	0.945	0.959	0.938	0.896	0.9956	0.9969	0.7752	0.6095
\mathcal{L}	1.000	0.999	0.995	0.999	0.9963	0.9998	0.9898	0.8158

Table S1. Effect of loss on motion profile similarity with GT motion: The cosine similarity of the estimated motion is consistently better across scenes using \mathcal{L} . ($\mathcal{L}_{3\text{dv}} = \mathcal{L}_{3\text{dd}} + \mathcal{L}_{3\text{dm}}$)

values for the 3D trajectory profile while surpassing the performance of $\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{kernel}} + \mathcal{L}_{2\text{dv}}$ on complex trajectories.

S3. Comparisons

For the synthetic dataset, we designated one view as the test view and synthesized its blurry version to evaluate rendered quality against both sharp and blurred GT images. In

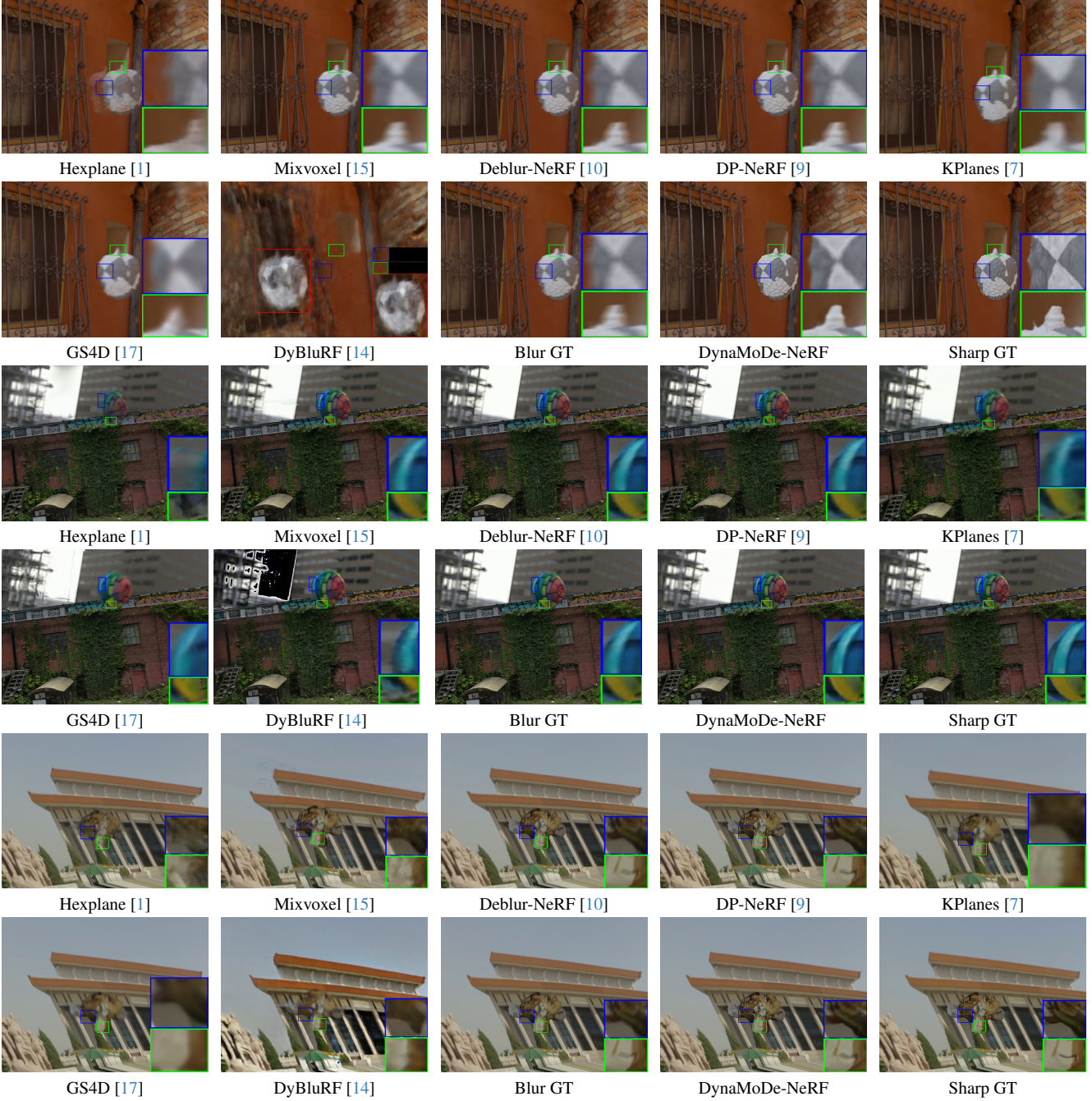


Figure S4. Comparison of baselines for novel view synthesis for the same test view and time instant. Our method outperforms all prior works to illustrate state of the art performance. Note that DyBluRF [14] produces a slightly misaligned novel view similar to the misalignment observed with the Cube dataset (see Fig. 4 in the main paper). Video results are provided in the accompanying .zip file.

Fig. S4, we show novel view synthesis from an unseen view at a particular time instant. We consistently outperformed HexPlane [1] and MixVoxel [15] across all scenes. DynaMode-NeRF consistently outperforms all other approaches. DyBluRF [14] rendered distorted and misaligned views. Deblur-NeRF [10] and DP-NeRF [9], which are

trained for a specific time instant, fail to handle significant blur as observed in the Lychee scene. These methods produce comparable results for the Monkey scene but generate outputs similar to the ground-truth (GT) blurred image for the Sphere scene. Video results for 4 synthetic and 1 real scene are provided in the accompanying .zip file.

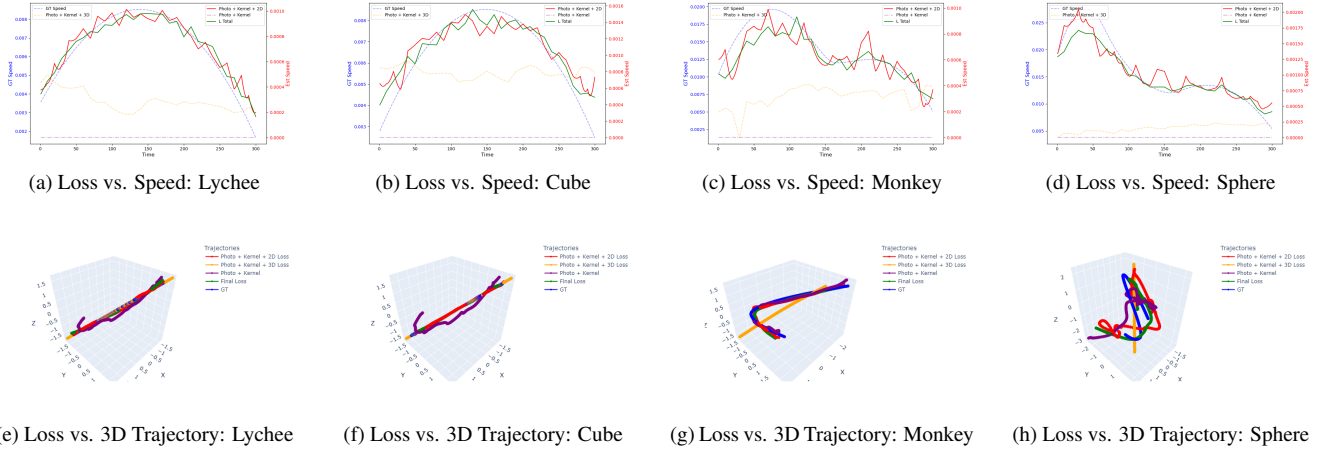


Figure S5. Effect of different losses on 3D motion profiles: The GT trajectory was defined in Blender [4] coordinate system, while the estimated trajectory was obtained in COLMAP coordinate system during pose estimation for each view. To ensure fair comparison, we aligned the ground truth (GT) trajectory with the estimated trajectory. The cosine similarity was calculated to assess their alignment.

S4. Effect of kernel size

We tabulate the average photometric metrics and the 3D motion cosine similarity (CS) values for different kernel sizes in Table S2. Kernel size 7 consistently achieves the highest PSNR across all kernels while delivering comparable SSIM and LPIPS values relative to other kernel sizes. Therefore, we have chosen 7 as our final kernel size.

S5. Ray Sampling Strategy

In each iteration, we fix the batch size $B = 2048$. Of this, 25% of the rays are randomly selected from the foreground (moving rigid object) and the remaining 75% are from the background. In experiments with real data, when the foreground occupies less than 25% of the frame due to its smaller size, we select more than 75% of the rays from the background. If the real data also contains humans, then we consider them as part of the background and select 25% of the background rays from the human-occupied regions. This ray sampling strategy is driven by the intuition that, in each iteration, the network optimizes all parameters and unknowns, ensuring that smaller foreground regions are not overlooked.

S6. Dataset

In image deblurring literature, it is common to synthesize realistic blurry images by averaging consecutive frames of a high-frame-rate video to simulate real-world, depth-dependent blurring [11–13, 16]. State-of-the-art approaches, such as [2, 5, 6, 18], use these frame-averaged synthetic datasets instead of blur-kernel synthesized datasets which often fail to capture real-world blur accurately. Following this practice, we also average 5 to 7

consecutive frames to generate object motion blur that takes into account the 3D nature of the object, its motion and the camera view. The number of averaging frames is selected based on the authenticity of the visual inspection of the averaged frame. The Lychee, Cube, Monkey, and Sphere datasets consist 20, 15, 12, and 8 videos, respectively. For each dataset, one video is reserved for testing/inference, while the remaining videos are used for training. Note that for the test view, we also synthesize a blurry video to compare our deblurred result against both the blurry video and the corresponding sharp video.

Training: We have trained all the models with Nvidia RTX3090 GPU for 300000 iterations for all the scenes.

Kernel Size	Background			Foreground			Overall			3D Motion Profile (CS) Similarity	
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	Speed	Trajectory
5	35.958	0.947	0.024	26.620	0.748	0.01	34.075	0.937	0.026	0.998	0.950
7	36.030	0.947	0.026	28.081	0.810	0.01	34.701	0.938	0.027	0.998	0.950
9	35.953	0.947	0.023	27.697	0.803	0.01	35.125	0.939	0.025	0.997	0.961
11	35.847	0.947	0.023	27.872	0.809	0.01	34.486	0.939	0.025	0.997	0.887
13	35.521	0.949	0.025	27.948	0.796	0.01	34.271	0.938	0.027	0.988	0.918
15	35.662	0.945	0.026	27.924	0.807	0.01	34.351	0.937	0.027	0.981	0.915

Table S2. Average performance metrics for different kernel sizes in terms of PSNR, SSIM, and LPIPS[8] values for Background (static region), Foreground (rigid moving object), and Overall (entire image consisting of both foreground and background).

References

- [1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–141, 2023. 3
- [2] Zheng Chen, Yulun Zhang, Liu Ding, Xia Bin, Jinjin Gu, Linghe Kong, and Xin Yuan. Hierarchical integration diffusion model for realistic image deblurring. In *NeurIPS*, 2023. 4
- [3] Djork-Arné Clevert. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 1
- [4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4
- [5] Yuning Cui, Wenqi Ren, Sining Yang, Xiaochun Cao, and Alois Knoll. Irnext: rethinking convolutional network design for image restoration. In *Proceedings of the 40th International Conference on Machine Learning. JMLR.org*, 2023. 4
- [6] Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Image restoration via frequency selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2): 1093–1108, 2024. 4
- [7] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12479–12488, 2023. 3
- [8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5
- [9] Dogyoon Lee, Minhyeok Lee, Chajin Shin, and Sangyoun Lee. Dp-nerf: Deblurred neural radiance field with physical scene priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12386–12396, 2023. 3
- [10] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12861–12870, 2022. 3
- [11] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4
- [12] Ziyi Shen, Wenguan Wang, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision*, 2019.
- [13] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4
- [14] Huiqiang Sun, Xingyi Li, Liao Shen, Xinyi Ye, Ke Xian, and Zhiguo Cao. Dyblurf: Dynamic neural radiance fields from blurry monocular video. *arXiv preprint arXiv:2403.10103*, 2024. 3
- [15] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19706–19716, 2023. 3
- [16] Patrick Wieschollek, Michael Hirsch, Bernhard Scholkopf, and Hendrik P. A. Lensch. Learning blind motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 4
- [17] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 3
- [18] Anas Zafar, Danyal Aftab, Rizwan Qureshi, Xinqi Fan, Pingjun Chen, Jia Wu, Hazrat Ali, Shah Nawaz, Sheheryar Khan, and Mubarak Shah. Single stage adaptive multi-attention network for image restoration. *IEEE Transactions on Image Processing*, 33:2924–2935, 2024. 4