

Deblur-NeRF: Neural Radiance Fields from Blurry Images

Supplementary Material

Li Ma¹

Xiaoyu Li²

Jing Liao³

Qi Zhang²

Xuan Wang²

Jue Wang²

Pedro V. Sander¹

¹The Hong Kong University of Science and Technology

²Tencent AI Lab

³City University of Hong Kong

Object Motion Blur



(a) Samples of Blurry Source Views

(b) Novel Views from NeRF

(c) Novel Views from Deblur-NeRF

Figure 1. One application of our method is to handle object motion blur. The flowers in the input images (a) are swaying with the wind, resulting in object motion blur. The original NeRF reconstructs blurry novel views (b) due to the view inconsistency. Our method successfully suppresses the motion blur caused by object motion and synthesizes sharp novel views (c).

CAMERA MOTION BLUR			DEFOCUS BLUR				
	PSNR↑	SSIM↑	LPIPS↓		PSNR↑	SSIM↑	LPIPS↓
<i>naive NeRF</i>	22.95	.6333	.3742	<i>naive NeRF</i>	22.53	.6627	.2480
<i>MPR + NeRF</i>	23.38	.6655	.3140	<i>KPAC + NeRF</i>	23.04	.6917	.1847
<i>PVD + NeRF</i>	23.10	.6389	.3425	Ours	23.47	.7244	.1220
Ours	25.65	.7586	.1818				

Table 1. Quantitative comparison on real scenes. We show the averages on 10 scenes of each blur type. The results are provided for reference only.

1. More Comparison Results

We first describe the details about capturing the real-world dataset for quantitative comparison. We captured 20 scenes with 10 scenes for each blur type using a Canon EOS RP camera. For all scenes, we set the exposure mode to manual to prevent exposure change during capturing. To capture the camera motion blur images, we set the shutter speed to values between 1/3 to 1/8 depending on the brightness of the scene. We manually shake the camera during one shot to capture the camera motion blur images. To capture the reference, we keep the exposure settings the same and let the camera stay still to capture sharp images. As for the defocus blur, we set the aperture to the value $f/4$ to get

a narrow depth of field. In each view, we randomly select an object as the focus. When capturing the reference images, we first set the aperture to its minimum value ($f/22$), and then adjust the ISO and shutter speed so that the exposure level indicator shows the same exposure. We finally captured $23 \sim 45$ blurry input images and $4 \sim 8$ sharp reference images for each scene.

We demonstrate more visual results of our method and several baselines in Fig. 6 and 7. Our method shows obvious improvement compared to other baselines. We also show quantitative comparison in Tab. 1. Even though we carefully capture the reference images, they may still suffer from misalignment and exposure differences against the ground truth, so the numeric result is provided for reference only.

2. Handling Object Motion Blur

Our pipeline does not make any assumption on the pattern of the kernel, but instead deblurs by utilizing the inconsistency between views. Therefore, theoretically our method can also handle other types of degradation that can be explained using convolutional model and cause the inconsistency between views. One application is to remove the object motion blur in dynamic scenes, i.e., blur caused

DSK	1.966ms	NeRF rendering	168.8ms	Ray blending	0.1926ms
-----	---------	----------------	---------	--------------	----------

Table 2. Running time on V100 GPU when ray batch size = 1024.



Figure 2. Deblurring results with gamma correction and learned CRF. Both methods achieve similar performance.

by the motion of the scene objects. One example is shown in Fig. 1, where the blurriness is caused by the flowers swaying (note that in the source views, the background is clear and the flower is blurry). Our method successfully reconstructs sharp flowers. In object motion blur, where the scene is dynamic, the observed scene varies when the view changes, causing the inconsistency between views. Our method reconstructs a static “canonical NeRF” similar to the Nerfies [2]. Instead of deforming the “canonical NeRF” in 3D to model the dynamic scene, our proposed DSK module warps and blends the canonical observations in image space to synthesize the dynamic blurry images.

3. Run-time Analysis

In our pipeline, the DSK module is parameterized as a coordinate-based MLP, and in order to get the N sparse kernel points, we need to forward the MLP N times, which potentially presents a large computation overhead. Luckily, the forward process can be done in parallel, and in the experiments we find that using a small MLP is adequate to model the spatially-varying kernels. Moreover, unlike NeRF, where the MLP is queried hundreds of times to render one ray, the DSK module only requires forward N times for each ray. As a result, the DSK only takes a small percentage of the processing time. During testing, since we have already restored a sharp NeRF, the DSK is no longer needed. We list the average running time of each component *during training* in Tab. 2.

4. Modeling the CRF

We conduct an experiment on actually modeling the CRF instead of simply using the gamma correction described in Sec. 4.2 in the paper. Specifically, we model the CRF $g(\cdot)$ using a small MLP (3 hidden layers with each layer having 16 neurons). We initialize the MLP so that the CRF is monotonically increasing. We show qualitative results in Fig. 2. The results demonstrate that optimizing a CRF does not achieve a substantial performance boost. This is possibly due to the difficulty in accurately fitting CRFs in the real world while deblurring.



Figure 3. Deblurring results with view consistent/inconsistent blur. Our method cannot successfully remove view consistent blur.

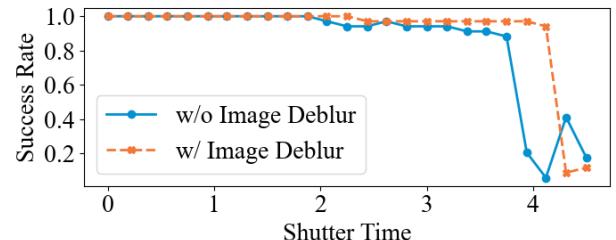


Figure 4. Experiments on the success rate of COLMAP under different degrees of blur with and without image deblur. The unit of shutter time is the same as in Blender. Bigger Shutter Time indicates larger degree of blur. The result shows that image deblur could improve the success rate of COLMAP, but the improvement is minor.

5. More discussions about Limitations

5.1. Consistent and Inconsistent Blur

As illustrated in the paper, our method works by exploiting the inconsistency of the blur pattern between different views. As a result, our method may fail when handling view consistent blur. Fig. 3 shows an example of this limitation. We synthesize the view consistent camera motion blur by forcing all the views to be perturbed in the same way (translation along the z-axis for a constant distance). The inconsistent blur is synthesized by randomly perturbing the camera as in our synthetic dataset. Our method fails to produce sharp NeRF with consistent blur, while successfully removing the blur when the motion is randomly generated.

5.2. Severe Blur

As discussed in the paper, when encounters input images that are severely blurred, the COLMAP [3, 4] may fail to reconstruct the camera poses. One may wonder whether the image-space baselines would be a better option. Here we investigate how the COLMAP is robust to the degree of blur and whether it will benefit from the image-space baselines. Then we experiment on the performance of our method and the image-space baseline to see whether the latter achieves better quality under severe blur.

Since it is difficult to control the degree of blur in real world, we conduct the experiment on a synthetic scene. We render the scene in Blender [1], where the camera motion is fixed and we change the shutter time to model the different degrees of blur. For each shutter time, we render multiple blurry input images and feed these images into COLMAP

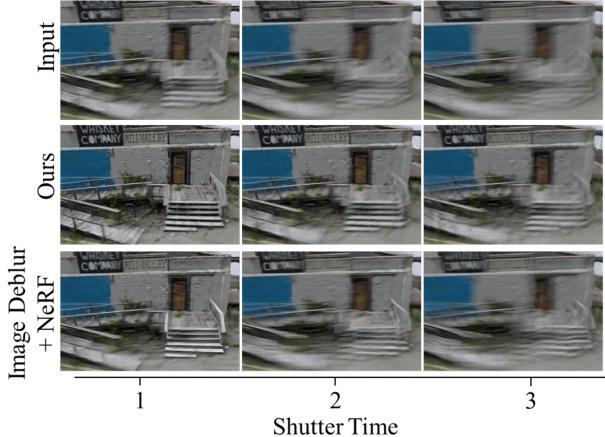


Figure 5. Qualitative results of our method and the image-space baseline (Image Deblur + NeRF) under different degrees of blur. The unit of shutter time is the same as in Blender. The results show that the image-space baseline does not achieve better visual results in case of severe blur.

(*w/o Image Deblur*). We may optionally run the image deblur algorithm [5] on the rendered blurry images, and then use the COLMAP to calibrate the cameras from the deblurred images (*w/ Image Deblur*). We define the *Success Rate* as the proportion of the input images being successfully registered by the COLMAP among all input images. We plot the *Success Rate* over the shutter time in Fig. 4. We can see that deblurring the input images before running the COLMAP helps to improve the registration robustness to the blurriness, but the improvement is minor. And the image that is successfully registered by the COLMAP already contains a considerable amount of blur. This indicates that our method works for most cases. When the input images get really blurry, we may use other more powerful calibration methods.

We also evaluate whether the image-space baseline will actually improve the deblurring under severe blur. We show qualitative results of our method and the image-space baseline under different degrees of blur in Fig. 5. We can see that in all degrees of blur where the COLMAP successfully registers the input, our method outperforms the image-space baseline. This proves that although doing image deblur before COLMAP could improve the registration, it cannot beat our method in terms of deblurring.

6. Supplementary Video

We provide a supplementary video with more visual comparison results, where novel views along a camera path are synthesized for visualization in each example. We highly recommend readers to view our supplementary video where our method achieves better view consistency and produces fewer artifacts than other baselines. Note that all

datasets we use contain only blurry input images. Some input images with mild blur may look like sharp frames in the video.

References

- [1] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [2](#)
- [2] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. [2](#)
- [3] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [2](#)
- [4] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. [2](#)
- [5] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. [3](#)

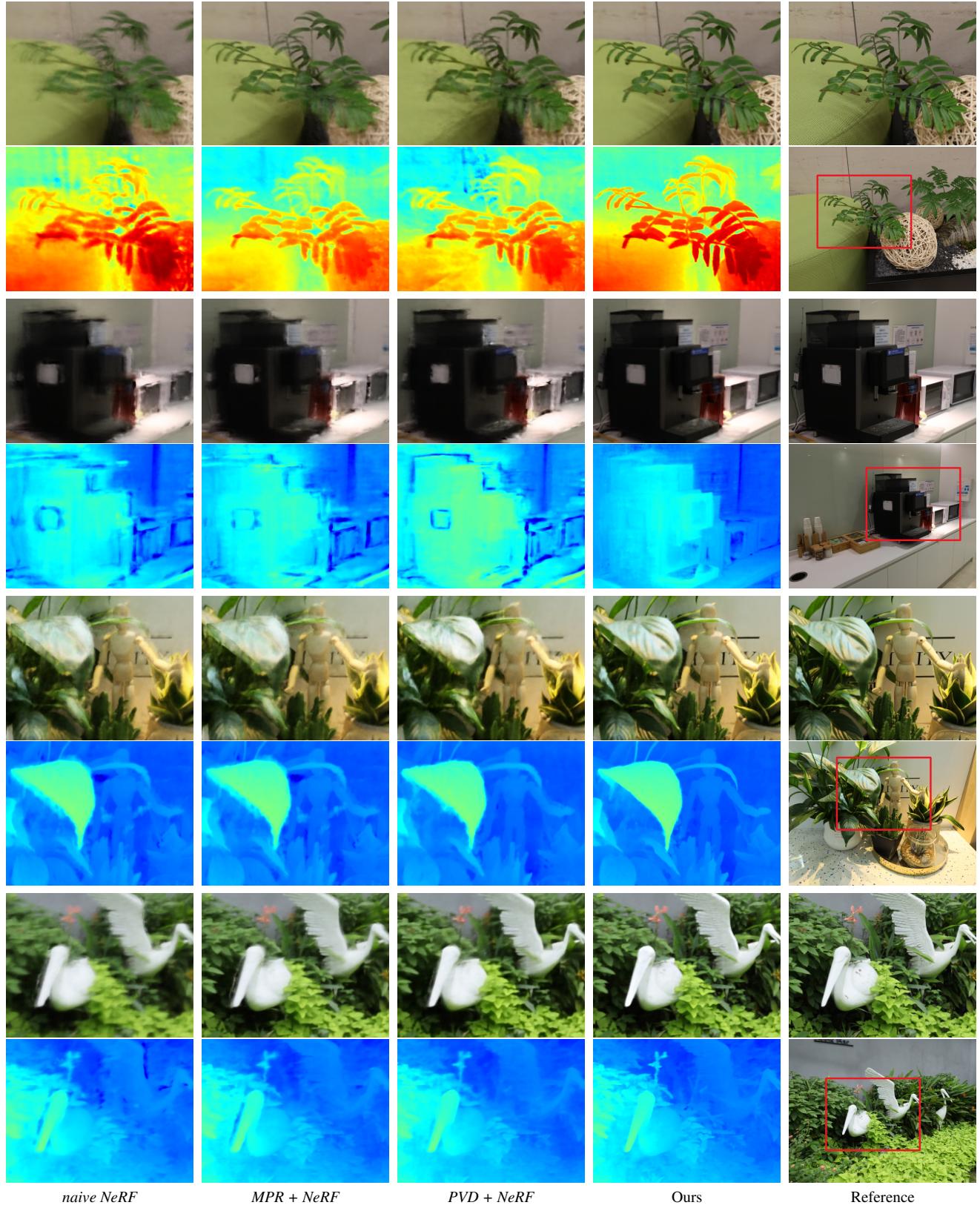


Figure 6. Qualitative comparison on real world camera motion blur. The last column is captured for reference only and may be misaligned with the ground truth.

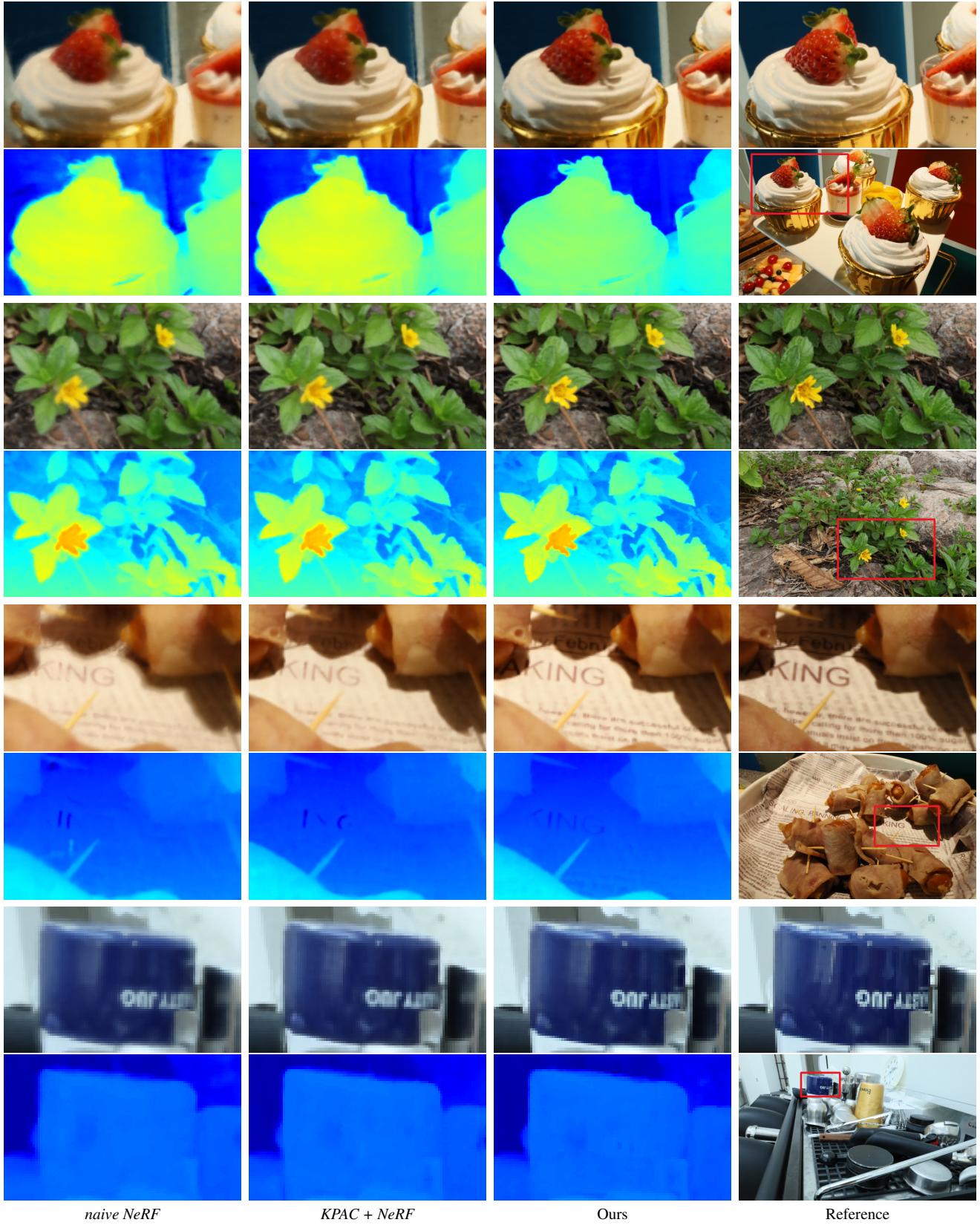


Figure 7. Qualitative comparison on real world defocus blur. The last column is captured for reference only and may be misaligned or have different exposures than ground truth.