Supplementary Material of SupeRVol: Super-Resolution Shape and Reflectance Estimation in Inverse Volume Rendering

Mohammed Brahimi[1,3] Bjoern Haefner[1,3] Tarun Yenamandra[1,3] Bastian Goldluecke[2] Daniel Cremers[1,3]
[1] Technical University of Munich, [2] University of Konstanz, [3] Munich Center for Machine Learning
{mohammed.brahimi, bjoern.haefner, tarun.yenamandra, cremers}@tum.de
bastian.goldluecke@uni-konstanz.de

Abstract

In this supplementary material, we show further insight into SupeRVol. Specifically we describe the networks architecture with all its parameters and training specifications. Then we elaborate on the capturing process to retrieve the synthetic and real world photometric images. After that we show novel renderings with changed illumination and reflectance, and finally we further analyse our approach with additional evaluations.

1. Network Details

1.1. Architecture

As mentioned in the main paper, we use three multilayer perceptrons (MLPs). One describes the geometry via an SDF, d_θ, one describes the BRDF's diffuse albedo, ρ_{γ_1}, and one is used for the specular parameters of the material, α_{γ_2}. The MLP of d_θ consists of 5 layers of width 512, with a skip connection at the 4-th layer. The MLPs of ρ_{γ_1} and α_{γ_2} consist of 4 layers of width 512, and 3 layers of width 256, respectively.

In order to compensate the spectral bias of MLPs [7], the input is encoded by positional encoding using 6 frequencies for both d_θ and α_{γ_2}, and 12 frequencies for ρ_{γ_1}.

1.2. Parameters and Cost Function

Similarly to [12, 13], we assume that the scene of interest lies within the unit sphere, which can be achieved by normalizing the camera positions appropriately. To approximate the Volume rendering integral (2) using (4), we use $m = 98$ samples which are also used to approximate (3), all with the sampling strategy of [11].

In the following, we distinguish between the ablation study noSR of the main paper and SupeRVol.

For SupeRVol, we set the objective's function trade-off parameters $\lambda_1 = \lambda_2 = 0.1$. Furthermore, in order to approximate the convolution with a Gaussian PSF (8), we use $N_s = 25$ in (9), and the terms of the objective function (10) and (11) consist of a batch size of 100 (inside the silhouette) and 1000, respectively. For the mask term (12) of the objective function, we use the same batch as (10), and add around 500 additional rays outside the silhouette whose rays still intersect with the unit sphere.

Concerning the noSR parameters, we set the objective's function trade-off parameters $\lambda_1 = 0.1$, $\lambda_2 = 0$, i.e. we *turn off* mask supervision, and the terms of the objective function (10) and (11) consist of a batch size of 2000 and 1000, respectively.

Note, that we always normalize each objective function's summand with its corresponding batch size.

1.3. Training

We train our networks using the Adam optimizer [4] with a learning rate initialized with $5\text{e}-4$ and decayed exponentially during training to $5\text{e}-5$, except for the MLP α_{γ_2} whose learning rate is constantly equal to $1\text{e}-5$. The remaining parameters are kept to Pytorch's default.

We train for 2000 epochs, which lasts about 2 days for noSR, and less than 3 days for SupeRVol using a single NVIDIA P6000 GPU with 24GB memory and 60 input images. We fix the geometry after the end of the training, and refine the BRDF's parameters using a larger batch size of 700 – all within the object's silhouette.

2. Data Acquisition

In this section we describe how we generated the datasets used in this paper

2.1. Synthetic Data

The synthetic datasets *dog1*, *dog2*, *girl1*, *girl2* were generated using Blender [3] and Matlab [6], where Blender [3] is used to render depth, normal and BRDF parameter maps

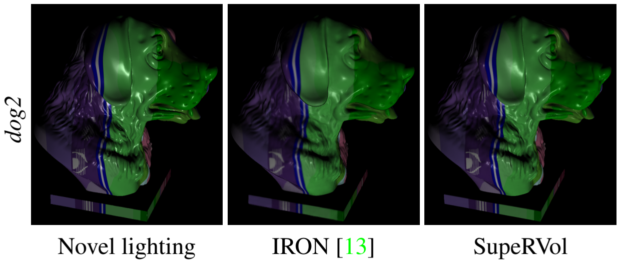

Figure 1. Generalization to novel non-colocated lighting. Compared to IRON [13], SupeRVol yields more accurate specularities. This demonstrates a better generalization for unseen views and illumination environments.

for each viewpoint, and Matlab [6] is used to render images using equation (6) and (7) of the main paper. The low-resolution images, of size 320 × 240, are obtained by blurring and downsampling high-resolution images, of size 1280 × 960, by a factor four (in each direction).

2.2. Real World Data

The real world data of *pony* and *dragon* were shared by the authors of [2], and the real world data of *bird* and *squirrel* were created by ourselves. We use a Samsung Galaxy Note 8 and the application ”CameraProfessional”[1] to generate RAW images as well as the smartphone’s images in parallel. We use the RAW images for our algorithm, and we pre-processed those using Matlab [6] by following [8]. Low-resolution images are obtained similarly to synthetic data, which are of size 270 × 480 for *pony* and *dragon*, and 504 × 378 for *bird* and *squirrel*.

3. Generalization to non-colocated relighting

We visualize in Fig. 1 how well our approach generalizes to non-colocated lighting setups. SupeRVol can create realistic specular behavior, while IRON [13] shows visible differences to the ground truth. Finally, Fig. 2 shows both relighting and material editing of *pony* estimated with SupeRVol. Although no ground truth is available for comparison, we can clearly see that relighting and editing are intuitively correct. Hence, it yields a coherent behaviour in terms of specularities and shadows. This highlights the validity of the estimated material parameters.

4. Novel Renderings

To further validate that our approach results in the scene’s parameters which can be used to alter the material and visualize it under novel illumination with standard software (Blender [3]), we show novel renderings in Fig. 3.

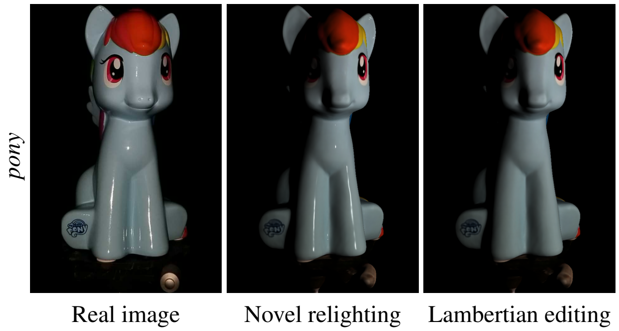

Figure 2. Novel non-colocated lighting and material editing of real world data with SupeRVol. From the given viewpoint (left), we first moved the light source to the right (middle). It can be seen that specularities and shadows moved appropriately. Finally, we perform material editing by removing the specular component (right). Both together demonstrate the quality of the underlying estimated material.

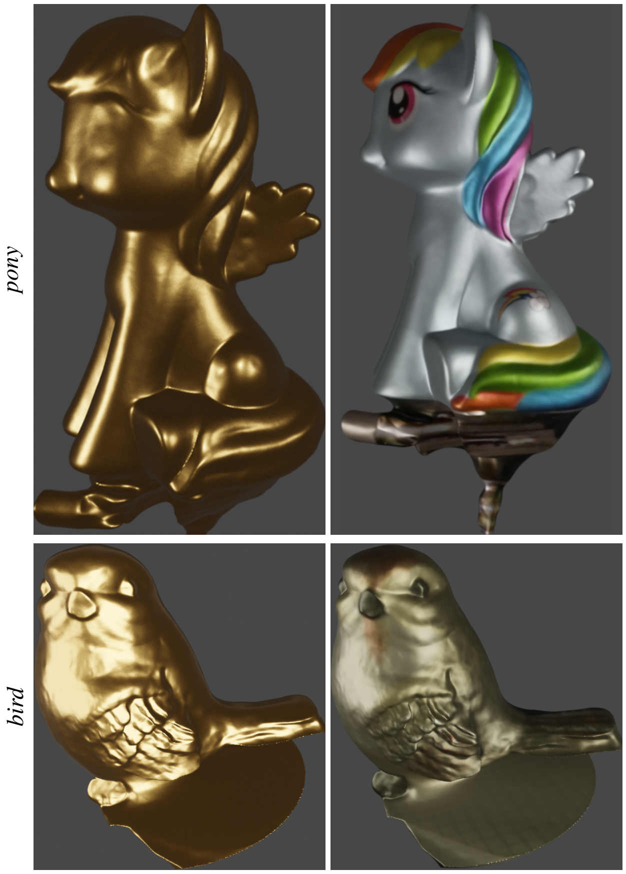

Figure 3. Novel rendering of *pony* and *bird* dataset. Both shapes where extracted from the learned sdf d using [5] and their BRDF was altered in Blender [3]. (left) shows a BRDF simulating gold, (right) uses the estimated diffuse albedo, with a more metallic, rougher and emissive material.

[1] https://play.google.com/store/apps/details?id=com.azheng.camera.professional, accessed 20-th June. 2023, 4.00PM

	↑PSNR			↑SSIM [10]		
	NeRF-SR	Mip-NeRF	SuperVol	NeRF-SR	Mip-NeRF	SuperVol
dog1	26.7970	27.4747	**32.0417**	0.8066	0.8243	**0.9236**
squirrel	21.6719	27.1066	**35.0512**	0.4548	0.7390	**0.9173**

Table 1. Average PSNR and SSIM [10] for *dog1* and *squirrel* datasets, using low-resolution input for training, for NeRF-SR [9], Mip-NeRF [1] and SupeRVol.

	↑PSNR		↑SSIM [10]		↓MAE	
	Vol-SR	SuperVol	Vol-SR	SupeRVol	Vol-SR	SupeRVol
synthetic	32.1979	**32.4413**	0.9123	**0.9170**	5.3063	**5.2795**
real world	31.6476	**31.7068**	0.9150	**0.9157**	×	×

Table 2. Average PSNR, SSIM [10] and MAE for the entire dataset using low-resolution input for training, for both Vol-SR and SupeRVol.

5. Additional evaluations

In this section, we explore the connection between our approach and the NeRF-SR [9] and Mip-NeRF [1] methods. Additionally, we exchange ideas and conduct a comparative analysis of our image formation model and sampling strategy with those introduced in NeRF-SR [9]. Finally, we assess the quality of both geometric and image synthesis as a function of the number of employed samples.

5.1. Evaluation of NeRF-SR [9] and Mip-NeRF [1]

To further evaluate the advantages of our modeling approach, we conducted a comparative analysis against NeRF-SR [9] and Mip-NeRF [1] in the context of novel view super-resolution synthesis. Fig. 4 visually demonstrates the substantial shortcomings of both NeRF-SR [9] and Mip-NeRF [1] when applied to our dataset, which is further confirmed quantitatively in Table 1. We attribute this failure to the inherent capturing process of our datasets, which consist of photometric images characterized by varying lighting conditions in each frame. This divergence from the static illumination assumption, a fundamental premise in NeRF-SR [9] and Mip-NeRF [1], likely contributes to the decline in their performance.
In light of these considerations, we conducted an ablation study to facilitate a more equitable comparison with NeRF-SR [9], which is explicitly tailored for novel view super-resolution synthesis and shares a super-sampling strategy akin to our approach, making it a relevant benchmark for our evaluation explained in the next section.

5.2. SupeRVol with NeRF-SR [9]

We conducted an evaluation of an ablated variant of our framework, referred to as Vol-SR. In Vol-SR, we maintain the identical architecture and hyper-parameters as in our main model but adopt the image formation model and super-sampling strategy employed in NeRF-SR [9]. Specifically, NeRF-SR's image formation model corresponds to a particular case of ours, where the PSF assumes the form of an average kernel. To elaborate, let d denote the downsampling factor; their PSF assigns a weight of $1/d^2$ if a point falls within the pixel footprint, and zero otherwise. Regarding their super-sampling strategy, which approximates the convolution with the PSF in Equation (8), NeRF-SR employs a fixed grid of size $d \times d$ within a given low-resolution pixel p. This grid is sampled at the centers of all high-resolution pixels located within pixel p, resulting in a total of $N_s = d^2$ samples. In our experiments, we set $d = 4$. The quantitative distinction observed in Table 2, are visually evident in Fig. 6, highlighting that SupeRVol consistently delivers sharper results compared to Vol-SR.

	↑PSNR	↑SSIM [10]	↓MAE	Time
5 samples	27.6830	0.8100	8.8945°	26h
10 samples	27.7873	0.8162	8.6499°	37h
15 samples	28.2177	0.8322	8.2731°	52h
20 samples	28.3308	0.8379	8.2418°	64h
25 samples	**28.3946**	**0.8411**	**8.1311°**	75h

Table 3. Evaluation of different number of samples per pixel, showing average PSNR, SSIM [10], and MAE along with the required training time for *girl1* using low resolution input for training.

5.3. Effect of the number of samples

As previously detailed, we employed a total of $N_s = 25$ random samples to approximate the convolution with the Gaussian PSF in Equation (8). In this section, we delve into the effects of this hyper-parameter on both training time and result quality. To investigate this, we specifically concentrate on the *girl1* dataset and train on the low-resolution images while varying the number of samples per pixel from 5 to 25. The findings, as presented in Tab. 3 and depicted in Fig. 5, indicate that increasing the number of samples has a favorable impact on the results, both in quantitative and qualitative terms. However, it comes at the cost of prolonged training times. A balance between quality and efficiency appears to be struck at $N_s = 15$, which offers a satisfactory compromise.

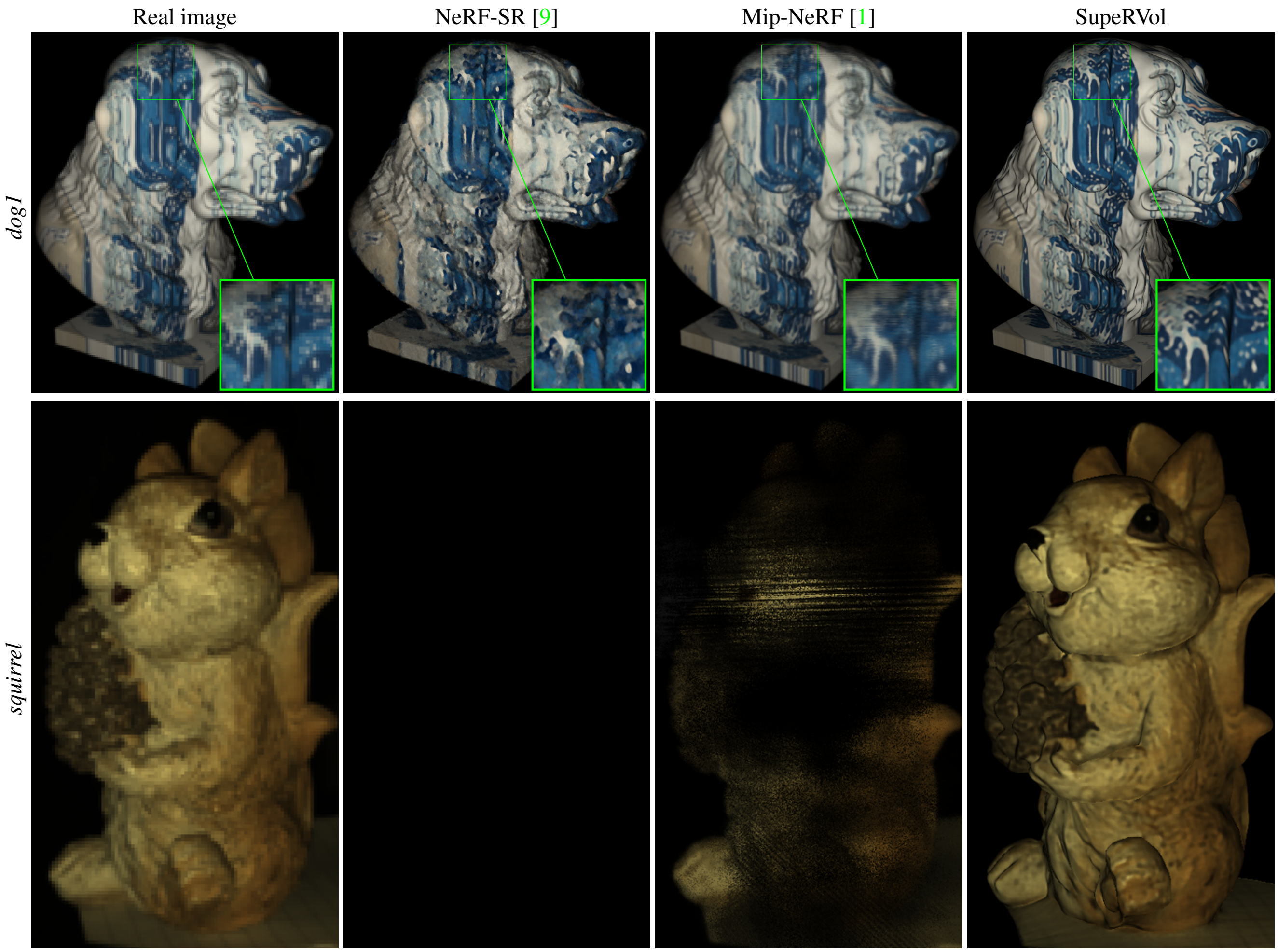

Figure 4. Image synthesis results of NeRF-SR [9] and Mip-NeRF [1]. NeRF-SR [9] leads to completely dark images for all the test viewpoints of *squirrel*.

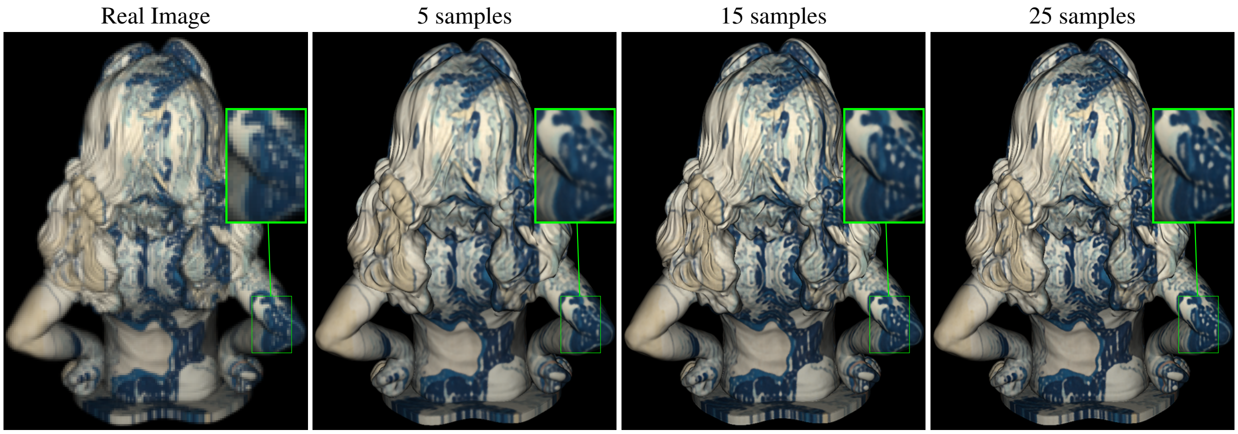

Figure 5. Image synthesis results obtained with 5, 15 and 25 samples per pixel for *girl1*.

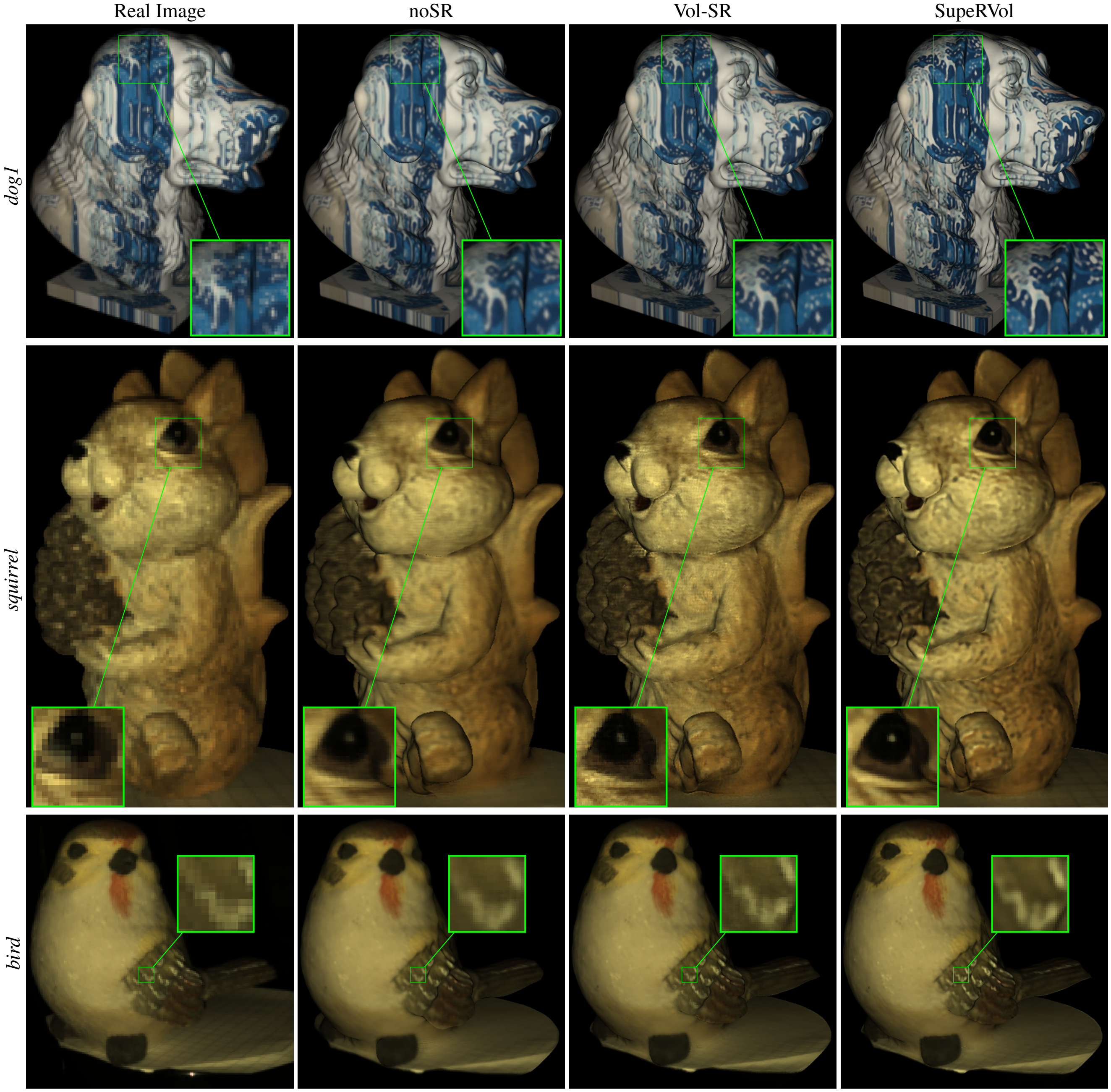

Figure 6. Image synthesis results of novel viewpoints with a colocated light source after low resolution training.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3, 4

[2] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *European Conference on Computer Vision*, pages 294–311. Springer, 2020. 2

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 1, 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, aug 1987. 2

[6] MATLAB. *version 9.8.0.1873465 (R2020a) Update 8*. The MathWorks Inc., Natick, Massachusetts, 2020. 1, 2

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[8] Rob Sumner. Processing raw images in matlab. *Department of Electrical Engineering, University of California Sata Cruz*, 2014. 2

[9] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6445–6454, 2022. 3, 4

[10] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 3

[11] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1

[12] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 1

[13] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022. 1, 2