SHINOBI: *Sh*ape and *I*llumination using *N*eural *O*bject Decomposition via *B*RDF Optimization *I*n-the-wild

Supplementary Material

Contents

Overview

In the supplement to SHINOBI, a method for 3D joint reconstruction of shape, illumination and materials from in-the-wild image sequences, we first present additional details on the method's architecture (Sec.A.2) and the optimization (Sec. A.5). In Sec. B we introduce additional qualitative results from object reconstructions of the NAVI dataset [11] and add visual examples to our ablation study. Finally, applications of our reconstructed data are shown in Sec. B.5. Please also visit our **project page** for an overview of this work and further visual results video.

A. Additional Method Details

A.1. NeRF Raymarching

As introduced in Sec. 3 the neural networks in NeRF [18] output a vector for view-dependent output color $\boldsymbol{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$ given a 3D location $\boldsymbol{x} \in \mathbb{R}^3$ and view direction $\boldsymbol{d} \in \mathbb{R}^3$. A camera ray $r(t) = \boldsymbol{o}+t\boldsymbol{d}$ is cast into the volume, with ray origin $\boldsymbol{o} \in \mathbb{R}^3$ and view direction $\boldsymbol{d}$. The final color is then approximated via numerical quadrature of the integral: $\hat{\boldsymbol{c}}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\boldsymbol{c}(t)\,dt$ with $T(t) = \exp(-\int_{t_n}^{t} \sigma(t)\,dt)$, using the near and far bounds of the ray t_n and t_f respectively [18]. Originally, the first MLP learns a coarse representation by sampling the volume in a fixed uniform sampling pattern along each ray. The second MLP is then evaluated sampled according to the coarse density distribution, placing more samples in high-density areas. In SHINOBI we only use one sampling stage with uniformly samples along rays. Using a raymarching scheme that skips empty space based iteratively updated occupancy data could bring additional performance gain during optimization.

A.2. Architecture

Hybrid hash encoding configuration. The hybrid encoding features two branches. For the base encoding we use 10 random offset annealed Fourier frequencies for the positional encoding followed by a small MLP featuring a single hidden layer with 64 dimensions and silu activation [8]. The output equals the input dimension (3), again as it is done by Zhu *et al*. [33]. We apply BARF's [16] Fourier annealing and add random frequencies as offsets to the logarithmically spaced frequencies [4, 28] to prevent artifacts from axis-aligned frequencies. The multiresolution hash grid is configured with 16 levels with a base resolution of 8 and a maximum target resolution of 2048. The embedding dimensions are 2 or 4. The experiments reported in Sec. 4 of our paper are generated using 2 dimensions. A slightly better decomposition quality can be achieved by increasing the dimensionality at the cost of increased memory consumption and runtime. Hence, the final feature dimension after encoding and concatenation is 35 or 67. See Sec. A.2 for an explanation of the annealing strategy applied to the hash grids.

Networks. The main network taking in the encoded features consists of 3 ReLU [9] activated layers with 64 channels. An additional linear layer generates the output for the σ density parameter from the 64 channel activation. Softplus $\mathrm{softplus}(x) = \ln(1 + \mathrm{e}^x)$ [7] is applied to the raw σ. The directions are encoded using 4 non-annealed regular Fourier components as in Mildenhall *et al*. [18] and then, concatenated with the main network output, fed to a secondary MLP to predict the view direction-dependent radiance $\tilde{\boldsymbol{c}}$ used in the beginning of the optimization. The secondary conditional network has a hidden dimension of 32 in our case. For the BRDF prediction a single linear layer compresses the main network output to 16 channels. From there the BRDF decoder is applied which consists of another two layers with 64 channels and ReLU activation each. Each BRDF output; basecolor, metallic and roughness has its own output layer followed by a sigmoid activation [3]. An additional diffuse embedding is added as conditioning to the basecolor branch before output. The illumination network decoding the per view latent vector is conditioned by the same configuration of mapping layers as outlined in Neural-PIL [3].

Multiresolution hash grid level annealing. Inspired by BARF [16] and Nerfies [21] we apply a coarse-to-fine annealing to the hash grid encoding by weighting the different

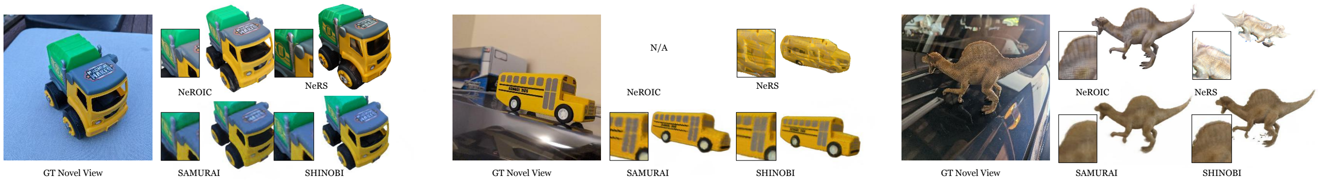


Figure 1. **Novel view synthesis compared to existing methods.** Additional example objects from NAVI [11] in-the-wild image collections. SHINOBI robustly reconstructs even when initialized with exteremely coarse poses while e.g. NeROIC [13] does not succeed on some scenes.

grid levels. Starting with only the features from the low resolution dense grid and all other features set to zero we increase the weights of the higher resolution levels gradually over time (cf. [15, 17]). Similar to the implementation by Lin *et al*. we formulate it as a truncated Hann window:

$$\Gamma_k(\boldsymbol{x};\alpha) = w_k(\alpha)\left[\sin(2^k\boldsymbol{x}), \cos(2^k\boldsymbol{x})\right] \tag{1}$$

$$w_k(\alpha) = \frac{1-\cos\left(\pi\,\mathrm{clamp}(\alpha-k,0,1)\right)}{2} \tag{2}$$

where $\alpha \in [0, L]$ with L being the number of resolution levels of the hash grid encoding.

We also tested the idea of BAA-NGP [17] replicating embeddings from low-resolution levels but observed reduced performance in our optimization setting. Similarly, we had no success with adding a straight-through operator to the interpolation on the hash grid as proposed in [10].

A.3. Camera Parameterization.

We label initial poses based on 3 simple binary questions: Left *vs*. Right, Above *vs*. Below, and Front *vs*. Back. This only takes about 4-5 minutes for a typical 80 image collection. Alternatively, our framework allows to extend the initialization to a camera multiplex spanning more than one quadrant. This can enable fully random initialization for front-facing scenes and image sets featuring rotating cameras with a fixed object distance as shown by Levy *et al*. [14]. As these constrained settings are uncommon for in-the-wild collections we discard it here. We use a perspective pinhole camera model and an initial field of view of 53.13 degrees. We optimize offsets to the original camera parameters of our 'lookat + direction' parameterization as outlined in the main paper. Here, we encode the trainable lookat parameter $\Delta\boldsymbol{d}$ directly as two direction components, ϕ, θ, which are used to offset the viewing direction $\boldsymbol{d}$ to obtain the updated $\hat{\boldsymbol{d}}$ as follows:

$$\boldsymbol{d} = (\boldsymbol{p}_{\text{eye}} + \Delta\boldsymbol{p}_{\text{eye}}) - \boldsymbol{p}_{\text{center}} \tag{3}$$

$$\theta = \arcsin(\boldsymbol{d}_y) + \Delta\boldsymbol{d}_\theta \tag{4}$$

$$\phi = \arctan2(\boldsymbol{d}_x, \boldsymbol{d}_z) + \Delta\boldsymbol{d}_\phi \tag{5}$$

$$\hat{\boldsymbol{d}} = \langle\cos\phi\sin\theta, \sin\phi, \cos\phi\cos\theta\rangle \tag{6}$$

We limit $\Delta\boldsymbol{d}$ to the range $[-0.5\pi, 0.5\pi]$.

We also tried other camera parameterizations like the popular 6D rotation representation by Zhou *et al*. [32] or FocalPose [23] that has recently been applied to NeRF with camera fine-tuning [22]. Interestingly, our lookat + direction parameterization performs the best in our setting as it seems to work well with the regularizations on camera poses.

A.4. Regularization and Losses.

Multiresolution hash grid regularization. To regularize the hash grid encoding we use the following normalized weight decay as proposed by Barron *et al*. [1]: $\mathcal{L}_{\text{Grid}} = \sum_l \text{mean}(V_l)$ with V_l referring to the grid embeddings at resolution level l. Computing the sum of the mean per-level puts a higher penalty on coarser grid levels compared to naive weight decay over all parameters at once. We find a weighting of 0.02 to 0.05 work well in our setting and settle for 0.02 as the final value. We apply gradient scaling to the gradients for the network by the norm of 0.1. Furthermore, gradient norm clipping with a clip value of 2.5 is applied to the camera gradients before the parameter update.

Surface normals regularization. We use the normal direction loss $\mathcal{L}_{\text{ndir}}$ from [29] to constrain the normals to face the camera until the ray reaches the surface. This helps in providing sharper surfaces without floater artifacts. Additionally, we observe that the explicit rendering step helps to constrain the surface normals as noise is reduced compared to optimization using only the predicted radiance.

Camera regularization. The camera regularization losses from SAMURAI are kept, particularly one to force the lookat-direction to point towards the origin ($\mathcal{L}_{\text{Lookat}}$) and one to prevent the cameras from moving too far away from the bounding volume ($\mathcal{L}_{\text{Bounds}}$) [4]. An additional term on the magnitude of the camera offset parameters helps to keep cameras from moving too far too fast with respect to the initial position due to strong updates in the beginning of the optimization.

BRDF losses. Joint estimation of BRDF and illumination is a delicate endeavor. For example, the illumination can easily fall into a local minimum. The object is then tinted in a bluish color, and the illumination is an orange color to express a more neutral color tone, for example. As our image collections have multiple illuminations, we can force the base color

$\boldsymbol{b}_c$ to replicate the pixel color from the input images. This way, a mean color over the dataset is learned and it becomes less likely to be trapped in local minima. We evaluate the Mean Squared Error (MSE) for this: $\mathcal{L}_{\text{Init}} = \mathcal{L}_{\text{MSE}}(\boldsymbol{C}^{\boldsymbol{s}}, \boldsymbol{b}_c)$. Additionally, we add a smoothness loss $\mathcal{L}_{\text{Smooth}}$ for the normal, roughness, and metallic parameters similar to the one used in UNISURF [20] to further regularize BRDF estimation [4].

Image reconstruction loss is a Charbonnier loss: $\mathcal{L}_{\text{Image}}(g, p) = \sqrt{(g-p)^2 + 0.001^2}$ between the input color from C for pixel s and the corresponding predicted color of the networks $\tilde{\boldsymbol{c}}$. We also calculate the loss with the rendered color $\hat{c}$ which becomes the main loss over time. This loss is computed over multiple resolution levels as outlined in Sec. 3 of the main paper whenever patches are rendered.

Mask losses. In total we use three mask loss terms. The $\mathcal{L}_{\text{silhouette}}$ as described in Sec 3.2 as well as the binary cross-entropy loss $\mathcal{L}_{\text{BCE}}$ between the volume-rendered mask and estimated foreground object mask and the background loss $\mathcal{L}_{\text{Background}}$ from NeRD [2]. The latter enforces all rays cast to the background to return 0. Consequently, the total mask loss is defined as: $\mathcal{L}_{\text{Mask}} = \lambda_{\text{xor}} \mathcal{L}_{\text{silhouette}} + \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Background}}$ where λ_{xor} is set to 50 and $\mathcal{L}_{\text{silhouette}}$ is normalized by the number of elements in the reference mask.

Final loss ensemble. Overall we compute two loss terms $\mathcal{L}_{\text{Network}}$ and $\mathcal{L}_{\text{Camera}}$ which consist of differently weighted versions of the photometric rendering loss and alignment losses plus the respective regularizations. The loss to optimize the decomposition network can be written as $\mathcal{L}_{\text{Network}} = \lambda_b \mathcal{L}_{\text{Image}}(\boldsymbol{C}^{\boldsymbol{s}}, \tilde{\boldsymbol{c}}) + (1 - \lambda_b)\mathcal{L}_{\text{Image}}(\boldsymbol{C}^{\boldsymbol{s}}, \hat{\boldsymbol{c}}) + \mathcal{L}_{\text{Mask}} + \lambda_a \mathcal{L}_{\text{Init}} + \lambda_{\text{ndir}} \mathcal{L}_{\text{ndir}} + \lambda_{\text{Smooth}} \mathcal{L}_{\text{Smooth}} + \lambda_{\text{DecSmooth}} \mathcal{L}_{\text{DecSmooth}} + \lambda_{\text{DecSparsity}} \mathcal{L}_{\text{DecSparsity}}$. Here, λ_b and λ_a are the optimization scheduling weights described below in more detail. As long as the camera multiplex has size $m > 1$ the camera multiplex consistency loss is added as follows: $\mathcal{L}_{\text{Network}} = \mathcal{L}_{\text{Network}} + 0.1(\mathcal{L}_{\text{multiplex}})$. To these losses the camera posterior scaling is applied as in SAMURAI [4]. The camera loss is weighted according to our view importance scaling instead. Badly initialized camera poses can still recover over the training duration as they get potentially large updates while cameras that perform well in terms of the losses are gradually faded out of the optimization. Additionally, the regularizations from above, $\mathcal{L}_{\text{Bounds}}$ and $\mathcal{L}_{\text{Lookat}}$ are added.

A.5. Optimization

Optimization scheduling. We use three fading λ variables to steer the optimization schedule smoothly as visualized in Fig. A2. Render resolution is continuously increased over the first half of the optimization while the number of active multiplex cameras is reduced. This is controlled by λ_c. Input image resolution is increased from 100 pixels to a resolution of 400 pixels on the longer image side over the first half of the training. For higher final output resolutions an even larger downsample factor (> 4) might be needed. This strategy allows the image patches to include even larger structures of the objects and improves camera alignment. The direct color optimization is faded to the BRDF optimization and the encoding annealing is performed over the first third of the optimization. λ_b is used for the BRDF transition and an independent α value is kept for the annealing. Finally, λ_a is used to scale some losses in a non-linear way. Focal length updates are delayed until a quarter of the optimization time. We start with the view importance weighting at the half-way point of the annealing schedule. SHINOBI renders image patches for most of the training time which adds more context to each update step, allowing us to add new losses tailored to camera alignment. The first 1000 steps are trained using regular random ray sampling, though, to help initialize a global shape quickly while both the render resolution as well as the hash grid resolution are low.

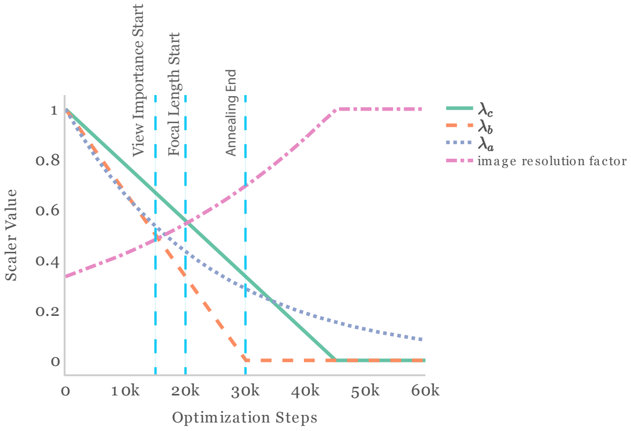


Figure A2. **Optimization schedule.** We use three λ parameter to scale losses to enable a smooth flow of the optimization parameters. Additionally, we indicate at which points in time the view importance weighting is introduced, the focal length parameters start to get updated and the encoding annealing ends.

Optimizer settings. The ADAM [12] optimizer updates the network weights based on $\mathcal{L}_{\text{Network}}$ with a learning rate of 1e-3 that is exponentially decayed by an order of magnitude over the training time. The same decay rate is applied to the optimizer concerned with the hash grid embeddings. The gradient are computed based on $\mathcal{L}_{\text{Network}}$ with the hash grid specific regularization $\mathcal{L}_{\text{Grid}}$ added. The learning rate of the camera optimizer is exponentially decayed by an order of magnitude every 40k steps. As mentioned before the $\beta 1$ parameter is set to 0.2 for the camera optimizer to stabilize the training in the presence of noisy gradients. It uses the gradients computed based on $\mathcal{L}_{\text{Camera}}$. The framework is trained using float16 mixed precision. The coordinate input to the encoding is 32 bit as is the rendering and illumination evaluation. The other MLPs and specifically the interpolation on the hash grids run at 16 bit precision, though.

B. Additional Experiments

B.1. Details on Compared Methods.

In addition to SAMURAI which has been introduced in Sec. 3 of the main paper we compare against two more recent methods for in-the-wild object reconstruction.

NeRS stands for Neural Reflectance Surfaces [31] that constrain reconstructions using a mesh-based representation. Starting from manually annotated rough initial poses and a template mesh the objects are decomposed into a surface mesh, illumination and surface reflectivity parameterized as albedo and shininess. We define the dimensions of an initial cuboid that approximates the object's bounding box for each scene in line with [11, 31].

NeROIC presents a multi-stage approach to reconstruct geometry and material properties of objects from online image collections. Camera poses are initialized with a COLMAP-based pipeline and fine-tuned during the first reconstruction stage. Following high-quality surface normals are estimated during the second stage. Finally, material properties and illumination are optimized to enable relighting in addition to novel view synthesis.

B.2. Additional Visual Results

Fig. 1 shows additional qualitative results on objects from the NAVI dataset compared to the baseline methods. Note, that the methods work at different image resolutions and that we show the original output. NeROIC is able to reconstruct high-frequency detail for scenes that have good initial poses but shows artifacts or fails on others. NeRS suffers from its low resolution mesh representation and often inaccurate camera alignment. SAMURAI and SHINOBI both reproduce appearance that is closer to the original illumination setting due to superior decomposition capabilities while SHINOBI recovers more high-frequency details. Consequently, on the reference free sharpness metric CPBD [19] our method clearly improves upon SAMURAI and NeRS, with CPBD scores of **0.82** (Ours) vs. 0.77 for SAMURAI vs. 0.65 for NeRS on the most challenging subset of the NAVI scenes where COLMAP reconstruction fails.

B.3. Qualitative Results of Ablations

Fig. B3 shows qualitative results corresponding to the numerical results from the ablation study reported in the main paper. It can be observed that a robust reconstruction is only possible using the full configuration of our method. While the multiplex consistency loss has only minimal impact on this example the result still shows some visible artifacts and overall increased noise level. It is apparent that a plain integration of multi-resolution hash grid encoding does not perform well on the task of joint camera pose and shape reconstruction Fig. B4 visualizes how the encoding schemes complement

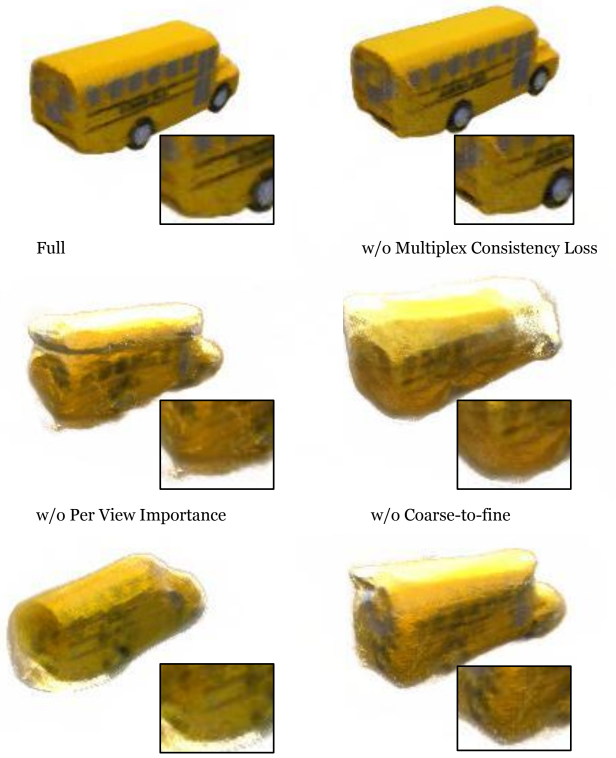


Figure B3. **Qualitative ablation study.** We show view synthesis results from novel view synthesis on the 'School Bus' scene from NAVI where we ablate components of our method. The visual results underline the importance of each part.

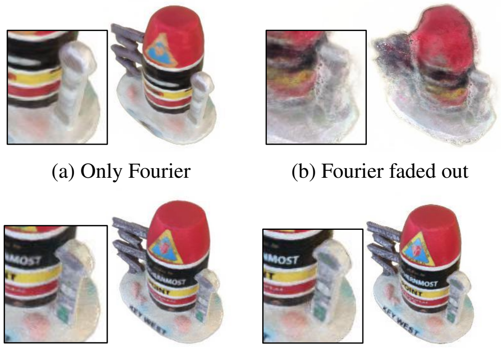


Figure B4. **Hybrid Encoding Ablation.** The Fourier encoding on its own is band limited, e.g. text is not reconstructed. Fading out the Fourier encoding during training destabilizes the optimization showing the importance of both encoding schemes. Evaluation using only the hash grid encoding results in noisy density but sharp texture. Our hybrid encoding yields sharp results with a consistent density at the object's surface.

each other. It has been shown that Fourier encoding together with a large MLP and a coarse-to-fine scheme does perform reasonable well for camera optimization [4, 16]. Hence, we merge the two encoders while keeping the MLP small and therefore band-limited. The main advantage is the continuity of the gradients for the coarse geometry that help to pull the optimization targets closer to the optimum in the beginning of the optimization. Updates based on higher-frequency details later in the optimization can propagate through the hash grids as they are constrained by our additional losses.

Method	Translation↓		Rotation °↓	
	S_C	$\sim S_C$	S_C	$\sim S_C$
PoseDiffusion [30]	0.51 ± 0.09	0.43 ± 0.11	41.33 ± 15.15	43.50 ± 13.67
HLoc [24, 25]	0.07 ± 0.13	0.06 ± 0.10	9.10 ± 18.75	9.72 ± 20.08
SHINOBI	0.250 ± 0.085	0.28 ± 0.09	22.84 ± 16.19	33.00 ± 19.97

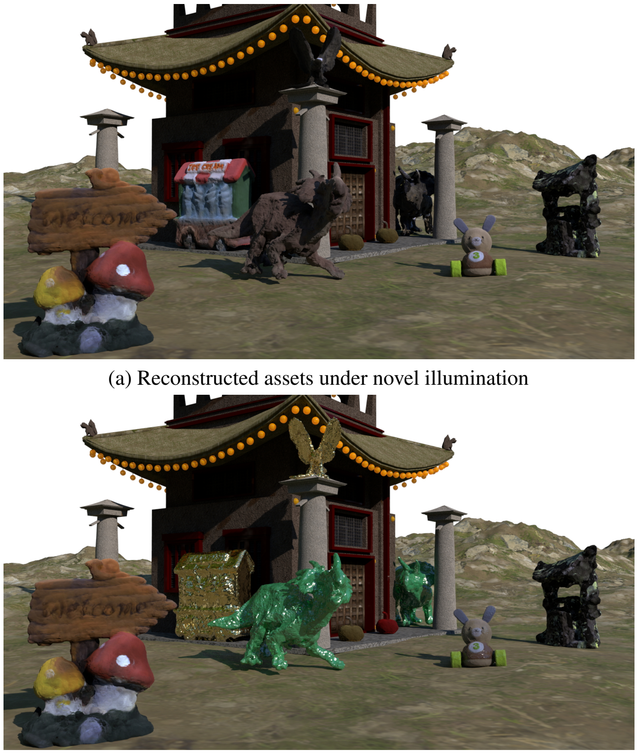
Table B1. **Pose estimation on in-the-wild data..** Evaluation of absolute rotation and translation errors after alignment on the NAVI [11] in-the-wild scenes. We compare SHINOBI against specialized camera pose estimation solutions. Note, that HLoc [24] fails completely on 5 scenes and is only able to recover 55% of views on average.

B.4. Comparison to other Camera Pose Estimation Methods

Tab. B1 compares methods for camera pose estimation on the NAVI in-the-wild scenes [11]. Traditional SfM methods like COLMAP [26, 27] paired with a neural feature detection and matching can recover poses with great accuracy but only succeed on a subset of scenes and images. PoseDiffusion [30] and ID-Pose [5], both fully neural models trained on large datasets, struggle on these out-of-distribution examples. We only report a full evaluation on PoseDiffusion as an example here. We observe that these models take important pose cues also from the background of object-centric image sets. This leads to poor results on in-the-wild image collections. A simple fine-tuning on masked images did not improve performance. In our experiments, camera pose estimation usually regresses to a front-facing camera layout for in-the-wild examples featuring different illumination and object scales. Consequently, our approach appears to be a good trade-off in-terms of camera pose quality.

B.5. Downstream Applications

The object decomposition into BRDF, illumination and shape enables us to edit illumination and material independently of the shape representation to re-light the object, for example. Furthermore, we can convert our neural representation into a parametric model like a mesh and physically based material suitable for easy integration into standard graphics pipelines. **Mesh extraction and asset generation.** We use a modified version of the mesh extraction component from SAMURAI [4] to extract triangle meshes from the learnt volume and the corresponding material parameters. Marching cubes is used to create an initial mesh. We post-process the mesh and perform automatic UV unwrapping using Blender [6]. Finally, textures are extracted by querying our pipeline for the BRDF around the baked surface locations. The extraction of a mesh takes around 3 minutes.


(a) Reconstructed assets under novel illumination

(b) Edited materials

Figure B5. **Integration and editing.** Although objects are initially captured under diverse illumination settings we can integrate multiple objects consistently into a scene in the end. BRDF parameters can be modified independently from the illumination.

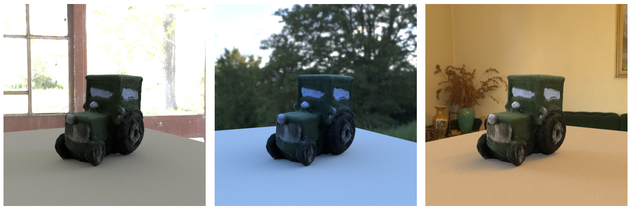


Figure B6. **Relighting application.** View synthesis under three different illumination settings using the estimated decomposition for a sample view from the “Tractor” scene.

Relighting and material editing. Our reconstructed assets can then be easily integrated into existing graphics pipelines. In Fig. B5 we show a SHINOBI themed scene featuring objects from the NAVI dataset in a new consistent illumination environment as it would be required for AR and VR applications. We can also modify the BRDF parameters independently of the lighting. Fig. B6 compares renderings of the same camera view but lit with different environment lights. Please also consider watching the supplementary video including more examples for the given applications.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *ICCV*, 2023. 2

[2] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. NeRD: Neural reflectance decomposition from image collections. *ICCV*, 2021. 3

[3] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS*, 2021. 1

[4] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T. Barron, Hendrik P.A. Lensch, and Varun Jampani. SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections. *NeurIPS*, 2022. 1, 2, 3, 5

[5] Weihao Cheng, Yan-Pei Cao, and Ying Shan. Id-pose: Sparse-view camera pose estimation by inverting diffusion models. *arXiv preprint arXiv:2306.17140*, 2023. 5

[6] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[7] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *NeurIPS*. MIT Press, 2000. 1

[8] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks : the official journal of the International Neural Network Society*, 107:3–11, 2017. 1

[9] Richard Hahnloser, Rahul Sarpeshkar, Misha Mahowald, Rodney Douglas, and H. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–51, 2000. 1

[10] Hwan Heo, Taekyung Kim, Jiyoung Lee, Jaewon Lee, Soohyun Kim, Hyunwoo J. Kim, and Jin-Hwa Kim. Robust camera pose refinement for multi-resolution hash encoding. *ICML*, 2023. 2

[11] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, Andre Araujo, Ricardo Martin-Brualla, Kaushal Patel, Daniel Vlasic, Vittorio Ferrari, Ameesh Makadia, Ce Liu, Yuanzhen Li, and Howard Zhou. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. *NeurIPS*, 2023. 1, 2, 4, 5

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 3

[13] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. NeROIC: Neural object capture and rendering from online image collections. *arXiv*, 2022. 2

[14] Axel Levy, Mark Matthews, Matan Sela, Gordon Wetzstein, and Dmitry Lagun. MELON: NeRF with Unposed Images Using Equivalence Class Estimation. 2

[15] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. *CVPR*, 2023. 2

[16] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. *ICCV*, 2021. 1, 5

[17] Sainan Liu, Shan Lin, Jingpei Lu, Shreya Saha, Alexey Supikov, and Michael Yip. BAA-NGP: Bundle-Adjusting Accelerated Neural Graphics Primitives. *arXiv*, 2023. 2

[18] Ben Mildenhall, Pratul Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 1

[19] Niranjan D. Narvekar and Lina J. Karam. A no-reference image blur metric based on the cumulative probability of blur detection (cpbd). *TIP*, 20(9):2678–2683, 2011. 4

[20] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *ICCV*, 2021. 3

[21] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *ICCV*, 2021. 1

[22] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T. Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. *ACM Trans. Graph.*, 2023. 2

[23] G. Ponimatkin, Y. Labbe, B. Russell, M. Aubry, and J. Sivic. Focal length and object pose estimation via render and compare. In *CVPR*, 2022. 2

[24] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 5

[25] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 5

[26] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *CVPR*, 2016. 5

[27] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. *ECCV*, 2016. 5

[28] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 1

[29] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-neRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 2

[30] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving pose estimation via diffusion-aided bundle adjustment. *ICCV*, 2023. 5

[31] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *NeurIPS*, 2021. 4

[32] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *CVPR*, 2019. 2

[33] Hao Zhu, Fengyi Liu, Qi Zhang, Xun Cao, and Zhan Ma. Rhino: Regularizing the hash-based implicit neural representation. *arXiv*, 2023. 1