SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inverse rendering of an object under entirely unknown capture conditions is a fundamental challenge in computer vision and graphics. Neural approaches such as NeRF have achieved photorealistic results on novel view synthesis, but they require known camera poses. Solving this problem with unknown camera poses is highly challenging as it requires joint optimization over shape, radiance, and pose. This problem is exacerbated when the input images are captured in the wild with varying backgrounds and illuminations. In such image collections in the wild, standard pose estimation techniques fail due to very few estimated correspondences across images. Furthermore, NeRF cannot relight a scene under any illumination, as it operates on radiance (the product of reflectance and illumination). We propose a joint optimization framework to estimate the shape, BRDF, and per-image camera pose and illumination. Our method works on in-the-wild online image collections of an object and produces relightable 3D assets for several use-cases such as AR/VR. To our knowledge, our method is the first to tackle this severely unconstrained task with minimal user interaction.

1 Introduction

Capturing high-quality 3D shape and materials of real-world objects is essential for many graphics applications in AR, VR, games, movies, etc. Using active multi-view object capture setups can provide high-quality 3D assets [6, 44] but cannot scale to a large-scale set of objects that are present in the world. By contrast, image collections provided by image search results or product review images exist for nearly every object. In this work, we propose a category-agnostic technique to estimate the 3D shape and material properties of objects from such Internet image collections. Estimating 3D shapes and materials from Internet object image collections poses several challenges as the images are highly unconstrained with varying backgrounds, illuminations, and camera intrinsics. Fig. 1 (left) shows a sample image collection of an object which forms the input to our technique.

Concretely, we estimate 3D shape and BRDF material properties [17] while also estimating per-image illumination, camera poses and intrinsics. Several contemporary works on shape and material estimation [6, 11, 12, 53, 63] assume constant camera intrinsics, near-perfect segmentation masks as well as almost-correct camera poses given by COLMAP [50, 51]. However, it is tedious to manually annotate object masks in the input images. We also observe that COLMAP often fails due to insufficient correspondences in real-world image collections with highly varying illuminations and backgrounds even when we constrain the correspondences to lie within object masks. Instead, we use a rough quadrant-based pose initialization, *e.g.* (Front, Above, Right), (Front, Below, Left) etc., as in NeRS [61], which usually takes only a few minutes of annotation time per image collection.

We base our technique on the recent Neural-PIL [12] method that proposes to learn illumination priors along with a novel pre-integration illumination network for estimating a neural volume with 3D shape, BRDF, and per-image illumination. Neural-PIL [12] assumes perfect camera poses and the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

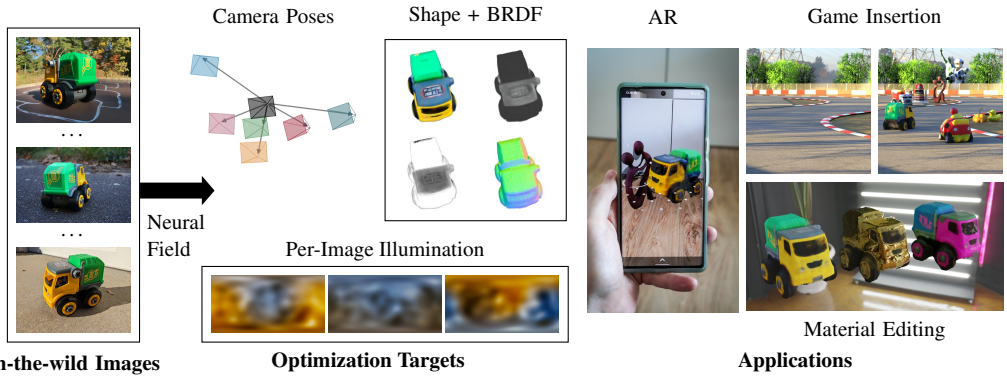

Figure 1: **Sample SAMURAI outputs and applications.** Sample input collection and the outputs on a challenging real-world unconstrained image collection. We extract meshes with material properties from the learned volumes enabling several applications in AR/VR, material editing etc.

same camera intrinsics across images. Given that traditional camera pose estimations like COLMAP may fail on in-the-wild images, we propose SAMURAI to jointly optimize camera poses as well as intrinsics with a carefully designed optimization protocol (Fig. 1). Furthermore, Neural-PIL requires perfect object masks, whereas we leverage automatically estimated object masks and deal with noisy masks using a posterior scaling loss. Some key distinguishing features of SAMURAI include:

- *Flexible camera parametrization for varying distances.* Standard techniques such as NeRF [41] assume fixed near/far clipping planes with equidistant cameras to the object. In contrast, we define the neural volume in global coordinates and propose to learn clipping planes per image.
- *Camera multiplex optimization.* Optimizing a single camera per image is prone to getting stuck in local minima. We propose using a multiplex camera where we optimize several camera poses per image and then phase out the incorrect poses throughout the optimization. Although camera multiplexes are previously used in mesh optimization [22], optimizing camera multiplex with neural volumes is challenging due to inefficient ray-based neural volume rendering.
- *Posterior scaling of input images.* Different input images have different noise characteristics (e.g., noisy masks), some images would be more useful for the optimization. We propose to use posterior scaling of input images which weighs the influence of different images on the optimization.
- *Mesh extraction.* We extract explicit meshes with BRDF texture from the learned neural volume making the resulting 3D models readily usable in existing graphics engines.

We observe that existing related datasets such as NeRD-dataset [11] do not capture the variations present in in-the-wild image collections. For instance, NeRD-dataset images have non-varying background making it easier for COLMAP to work. In addition, the illumination variations are more drastic in internet images captured by different people/cameras and at different times. To evaluate on the practical in-the-wild setting, we collected image collections with 8 objects in which each image is captured under unique background and illumination conditions. In addition, we also vary the cameras used for capturing the images. Experiments on our new dataset as well as existing datasets demonstrate better view synthesis and relighting results with SAMURAI compared to existing works. In addition, explicit mesh extraction allows for seamless use of learned 3D assets in graphics applications such as AR. Fig. 1 shows a sample AR scene with the learned 3D assets. We will release the code and datasets upon publication.

2 Related works

Neural fields encode spatial information in the MLP network weights and we can retrieve the information by simply querying the coordinates [15, 40, 46, 54]. With these MLPs, we can store alpha or density values and then explicitly render the volume using ray marching [37]. Recent works such as NeRF [41] leverage this neural volume rendering to achieve photo-realistic view synthesis results with view-dependent appearance variations. Rapid research in neural fields followed, which alternated the surface representations [45, 56], provided general improvements to the method [4, 55], reduced the long training times [14, 36, 42, 49, 57] and inference times [11, 23, 27, 36, 42, 60], enabled extraction of 3D geometry and materials [11, 43, 61], added generalization capabilities [13, 57, 65] or enabled relighting of scenes [5, 11, 12, 28, 38, 53, 63, 64]. However, most methods rely on COLMAP poses, which can fail in complex settings, such as varying illumination and locations.

Joint camera and shape estimation is a highly ambiguous task. An accurate shape reconstruction is only possible with accurate poses and vice versa. Often techniques rely on correspondences across images to estimate camera poses [50, 51]. Recently, several methods combined camera calibration with a joint neural volume training. Jeong *et al*. [25] (SCNeRF) rely on correspondences and BARF [35] proposes a coarse to fine optimization using a varying number of Fourier frequencies and requires rough camera poses and NeRF-- [58] requires training the neural volume twice while keeping the previous camera parameter optimization. GNeRF [39] proposes to use a discriminator on randomly sampled views to jointly learn a pose estimation network on synthesized views. Over time the pose estimation network can be used to estimate the real camera poses, which can then be used for the full neural volume training. NeRS [61] deforms a sphere to the specific shape using coordinate-based MLPs and converts the deformation field to a mesh; while also optimizing camera poses and single illumination. Compared to previous work, our method does not rely on correspondences (vs. SCNeRF), which might be hard to obtain in varying illuminations, can have extremely coarse poses due to the camera multiplexing (vs. BARF), works in multiple Illumination (vs. all prior art), does not require training twice (vs. NeRF--) or a GAN-style training (vs. GNeRF).

BRDF and illumination estimation is a challenging ambigious research problem. One needs controlled laboratory capture setups for high-quality BRDF capture [3, 9, 29–31]. Casual estimation exists which enable on-site material acquisition with simple cameras and a co-located camera flash. These technique often constrain the problem to planar surfaces with either a single shot [1, 8, 18, 24, 33, 48], few-shot [1] or multi-shot [2, 9, 19–21] captures. This casual capture setup can also be extended to a joint BRDF and shape reconstruction [5–7, 10, 26, 44, 48, 62] or entire scenes [34, 52]. Most of these methods require a known active illumination like a co-located flash. Recovering a BRDF under unknown passive illumination is significantly more challenging and ambiguous as it requires disentangling the BRDF from the illumination. Often the specular parameter is constrained to be non-spatially varying or omitted [32, 59, 63]. Recently, neural field-based decomposition achieved decomposition of scenes under varying illumination [11, 12] or fixed illumination [63], but require known, near-perfect camera poses. This can fail on challenging datasets and our method enables decomposition of these in-the-wild datasets.

3 Method

Problem setup. The input is a collection of q object images $C_j \in \mathbb{R}^{s_j \times 3}; j \in \{1, \dots, q\}$ captured with different backgrounds, cameras and illuminations; and can also have varying resolutions. We denote the value of specific pixel as $\boldsymbol{C}^s$. In addition, we roughly annotate camera pose quadrants with 3 simple binary questions: Left *vs*. Right, Above *vs*. Below, and Front *vs*. Back. We automatically estimate foreground segmentation masks $M_j \in \{0, 1\}^{s \times 1}$ using U^2-Net [47], which can be imperfect. Given these, we jointly optimize a 3D neural volume with shape and BRDF material information along with per-image illumination, camera poses, and intrinsics. This practical capture setup allows the conversion of most 2D image collections into a 3D representation with little manual work. The rough pose quadrant annotation takes about a few (3-5) minutes for a typical 80 image collection. At each point $\boldsymbol{x} \in \mathbb{R}^3$ in the 3D neural volume $\mathcal{V}$, we estimate the BRDF parameters for the Cook-Torrance model [17] $\boldsymbol{b} \in \mathbb{R}^5$ (basecolor $\boldsymbol{b}_c \in \mathbb{R}^3$, metallic $\boldsymbol{b}_m \in \mathbb{R}^1$, roughness $b_r \in \mathbb{R}$), unit-length surface normal $\boldsymbol{n} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$. We also estimate the latent per-image illumination vectors $\boldsymbol{z}_j^l \in \mathbb{R}^{128}; j \in \{1, \dots, q\}$ used in Neural-PIL [12]. We also estimate per-image camera poses and intrinsics, which we represent using a 'look-at' parameterization that we explain later. Next, we provide a brief overview of prerequisites: NeRF [41] and Neural-PIL [12].

Brief overview of NeRF [41]. NeRF [41] models a neural volume for novel view synthesis with two Multi-Layer-Perceptrons (MLP). The MLPs take 3D location $\boldsymbol{x} \in \mathbb{R}^3$ and view direction $\boldsymbol{d} \in \mathbb{R}^3$ as input and outputs a view-dependent output color $\boldsymbol{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$. These output colors for a target view are computed by casting a camera ray $r(t) = \boldsymbol{o} + t\boldsymbol{d}$ 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 [41]. The first MLP is trained to learn a coarse representation by sampling the volume in a fixed sampling pattern along each ray. Using the coarse density distribution in the coarse MLP, a finer sampling pattern is created, placing more samples on high-density areas. The second MLP is then trained on the fine and coarse sampling combined.

Brief overview of Neural-PIL [12]. Similar to NeRF [41], Neural-PIL [12] also uses two MLPs to learn a neural volume. The first MLP is similar to NeRF with an additional GLO (generative

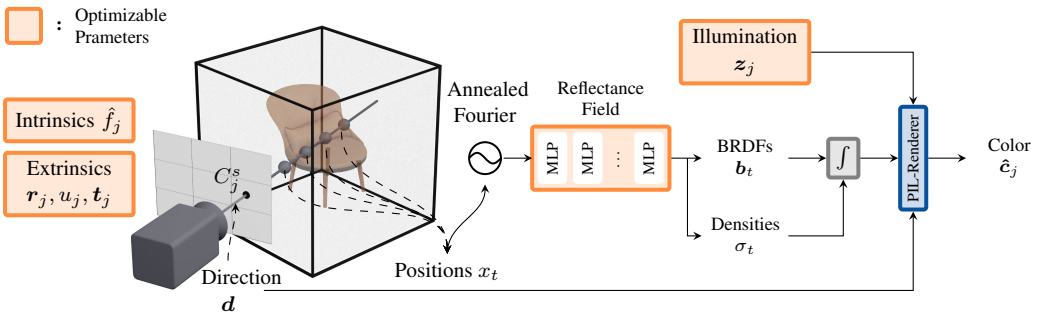

Figure 2: **Overview.** We jointly optimize the intrinsic ($\hat{f}_j$) and extrinsic camera ($\boldsymbol{r}_j, u_j, \boldsymbol{t}_j$) parameters alongside the shape (σ) and BRDF ($\boldsymbol{b}$) in a *Reflectance Field* and per-image illumination ($\boldsymbol{z}_j$). The shape is encoded in the density σ and also used to integrate all BRDFs along a ray in direction $\boldsymbol{d}$. The composed BRDF is then rendered using Neural-PIL [12] in a deferred rendering-style.

latent optimization) embedding that models the changes in appearances (due to different illuminations) across images. In the second MLP, breaking with NeRF, Neural-PIL predicts not only a view-dependent output color but also BRDF parameters at each 3D location. The second MLP takes 3D location as input and outputs volume density and BRDF parameters (diffuse, specular, roughness, and normals). Unlike NeRF, this second MLP does not take the view direction as input but can model view-dependent and relighting effects in the rendering due to explicit BRDF decomposition. A key distinguishing factor of Neural-PIL is the use of latent illumination embeddings and a specialized illumination pre-integration (PIL) network for fast rendering, which we refer to as 'PIL rendering'. More concretely, Neural-PIL optimizes per-image illumination embedding $\boldsymbol{z}_j^l$ to model image-specific illumination. The rendered output color $\hat{\boldsymbol{c}}$ is equivalent to NeRF's output $\boldsymbol{c}$, but due to the explicit BRDF decomposition and illumination modeling, it enables relighting.

3.1 SAMURAI - Joint optimization of shape, BRDF, cameras, and illuminations

The main limitations of Neural-PIL include the assumption of near-perfect camera poses and the availability of perfect object segmentation masks. We observe that COLMAP either produces incorrect poses or completely fails due to an insufficient number of correspondences across images when the backgrounds and illuminations are highly varying across the image collection. In addition, camera intrinsics could vary across image collections, and the automatically estimated object masks could also be noisy. We propose a technique (we refer to as 'SAMURAI') for joint optimization of 3D shape, BRDF, per-image camera parameters, and illuminations for a given in-the-wild image collection. This is a highly under-constrained and challenging optimization problem when only image collections and rough camera pose quadrants are given as input. We address this highly challenging problem with a carefully designed camera parameterization and optimization schemes.

Architecture overview. A high-level overview of SAMURAI architecture is shown in Fig. 2, which mostly follows the architecture of Neural-PIL [12] and NeRD [11]. However, we do not use a coarse network for efficiency reasons and only use the fine or decomposition network with 8 MLPs of 128 features. Overall we sample 128 points along the ray in fixed steps. The base network is followed by a hidden layer with 1 feature output for the density, and an MLP with a hidden layer for the view and appearance conditioned radiance. We also leverage a BRDF decoder similar to NeRD [11], which first compresses the feature output of the main network to 16 features and expands them again to the BRDF (base color, metallic, roughness). We encourage sparsity in the embedding space using: $\mathcal{L}_{\text{Dec Sparsity}} = {}^1\!/_N \sum_i^N |\boldsymbol{e}_i|$, an $\mathcal{L}_1$-Loss on the BRDF embedding $\boldsymbol{e}$ and a smoothness loss $\mathcal{L}_{\text{Dec Smooth}} = {}^1\!/_N \sum_i^N |f(\theta; \boldsymbol{e}_i) - f(\theta; \boldsymbol{e}_i + \boldsymbol{\varepsilon})|$, where N denotes the number of random rays, $f(\theta)$ the BRDF decoder with the weights θ and ε is normal distributed Gaussian noise with a standard deviation of 0.01. Similar to NeRD, we also predict a regular direction-dependent radiance $\tilde{\boldsymbol{c}}$ in the early stages of the training. This is mainly used for stabilization in the early stages. As this direct color prediction is only used in the early stages, we omitted it for clarity in Fig. 2. Inspired by Tancik *et al*. [54] we add a Gaussian distributed noise to the Fourier embedding. However, as we also leverage BARF's [35] Fourier annealing, we add these random frequencies as offsets to the logarithmically spaced frequencies. Without these random offsets, artifacts from the axis-aligned frequencies like stripes can occur. Further details are available in the supplementary material.

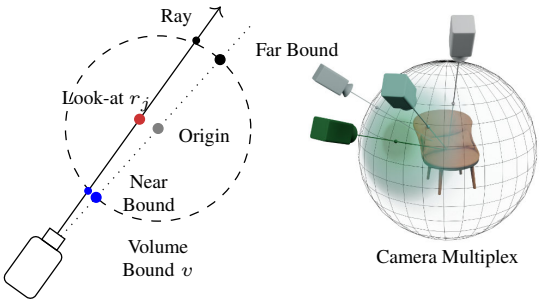

Figure 3: **Ray parametrization and Camera Multiplex.** (Left) Our ray bounds are defined by a world space sphere. The distance from the origin to the near and far points of the sphere define the sampling range. By defining a globally consistent sampling range our cameras can be placed in arbitrary distances. (Right) When optimizing multiple camera hypotheses, only the best camera should optimize the shape and appearance, here visualized in a deep green. Cameras which are not aligned that well are visualized in off-white and cannot influence the reflectance field. When the non-aligned camera poses improve during the training, they may become applicable for the network optimization.

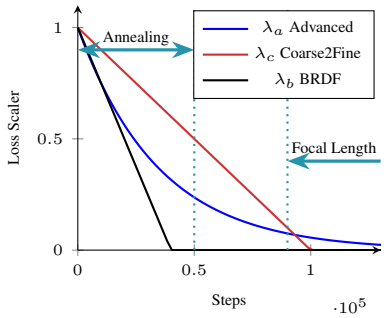

Figure 4: **Optimization scheme.** Our method performs a smooth flow of optimization parameters using three λ variables for loss scaling. Additionally, we perform a Fourier frequency annealing in the first phase of the training and delay the training of the focal length for later stages in the training. The BRDF estimation is mainly regulated by the λ_b parameter.

Rough camera pose quadrant initialization. We observe that camera pose optimization is a highly non-convex problem and tends to quickly get stuck in local minima. To combat this, we propose to annotate camera pose quadrants with 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. Note that our pose quadrant initialization is much more noisy compared to adding some noise around GT camera poses as in some related works such as NeRF-- [58]. This rough pose initialization is in line with recent works such as NeRS [61] that also use rough manual pose initialization.

Flexible object-centric camera parameterization for varying camera distances. We define the trainable per-image camera parameters using a 'look-at' parameterization with a 3D look-at vector $\boldsymbol{r}_j \in \mathbb{R}^3; j \in \{1, \ldots, q\}$, a scalar up rotation $u_j \in \mathbb{R}[-\pi, \pi]$ and a 3D camera position $\boldsymbol{t}_j \in \mathbb{R}^3$ as well as a focal length $f_j \in \mathbb{R}$. Furthermore, these are stored as offsets to the initial parameters, enabling easier regularization. We additionally store the offset vertical focal lengths f_j in a compressed manner in a similar manner to NeRF--: $\hat{f}_j = \sqrt{f_j/h}$ [58], where h is the image height in pixels. The cameras are initialized based on the given pose quadrants and with an initial field of view of 53.13 degrees. We optimize a perspective pinhole camera with a fixed principal point but per-image focal lengths. The cameras are not always equidistant to the object for the in-the-wild image collections. To account for variable camera-object distances, we do not set fixed near and far bounds for each ray which is a standard practice in neural volumetric optimizations such as NeRF [41]. Instead, we define a sampling range based on the camera distance to origin, *e.g*. the near bound is $|\boldsymbol{o}| - v$ and the far bound is $|\boldsymbol{o}| + v$, where v is defined as our sampling radius with a diameter of 1. We illustrate this sphere with near and far bounds in Fig. 3. This explicit computation of near and far bounds for each ray enables placing the cameras at arbitrary distances from the object. This is not possible with the existing neural volume optimization techniques that use fixed near and far bounds for each camera ray. The cameras are then placed based on the quadrants and placed in a distance so that the sampling radius v is visible given the field of view. This look-at parameterization is more flexible for optimizing object-centric neural volumes than more commonly used 3D rotation matrices.

Camera multiplexes. We observe that camera pose optimization gets stuck in local minima even with rough quadrant pose initialization. To combat this, inspired by mesh optimization works (*e.g*. [22]), we propose to optimize a camera multiplex with 4 randomly jittered poses around the quadrant center direction for each image. Optimizing multiple cameras per image would reduce the number of rays we can cast in a single optimization step due to memory and computational limitations. This makes camera multiplex optimization noisy and challenging in learning neural volumes. We propose techniques to make camera multiplex learning more robust by dynamically re-weighing the loss functions associated with different cameras in a multiplex during the optimization. This

process is visualized in Fig. 3. Specifically, we compute the mask reconstruction loss ${\mathcal{L}_{\text{Mask}}}_j^i \in \mathbb{R}$ associated with each camera i and image j. We then re-weigh each camera loss in a multiplex with $S_j = softmax(-\lambda_s {\mathcal{L}_{\text{Mask}}}_j^i)$, where $S_j \in \mathbb{R}^4$ and λ_s is a scalar that is gradually increased during the optimization. That is, we re-weigh the loss with ${\mathcal{L}_{\text{Network}}}_j = \sum_i S_j^i {\mathcal{L}_{\text{Network}}}_j^i$. This dynamic re-weighing reduces the influence of bad camera poses while learning the shape and materials. Since we can only render a random set of rays within each batch, we update the camera multiplex weights S_j with a memory bank and momentum across the batches. See the supplements for more details.

Posterior scaling of input images. Some images are noisier than others (*e.g.*, due to camera shake) or noisy object masks M_j. To be robust against such noisy data, we propose to re-weigh images in the given collection. We keep a circular buffer of around 1000 elements with the recent mask losses and rendered image losses with multiplex scaling applied. We use this buffer to calculate the mean μ_l and standard deviation σ_l of these losses. Given the recent loss statistics we limit the backpropagation to the network by multiplying with $\mathcal{L}_{\text{Network}}$, if a particular image is worse than the current statistics using: $s_{\text{p}_j} = \max(\tanh\left(\frac{\mu_l - ({\mathcal{L}_{\text{Mask}}}_j + {\mathcal{L}_{\text{Image}}}_j)}{\sigma_l}\right) + 1, 1)$.

Mesh extraction. Similar to NeRD [11] we perform a mesh extraction from the learned reflectance neural volume. However, we differ from their method. In the first step, we perform a marching cubes extraction step similar to the one proposed in NeRF [41]. However, as the naive marching cube algorithm can have block artifacts, we sample 2 million points on the mesh surface and cast rays towards the surface. The resulting point cloud is converted to a refined mesh using Poisson reconstruction. This refined mesh provides more details and smoother surfaces. We then UV unwrap the resulting mesh in Blender's [16] automatic UV unwrapping tool, and bake the the world space positions in to the texture map. We can then query all surface locations in our fine network and compute the BRDF texture maps. For easy deployment, we then save the textured mesh in the GLB format. The extraction of a mesh takes around 3-5 minutes.

3.2 Losses and Optimization

Image reconstruction loss is a Chabonnier 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 additionally calculate the loss with the rendered color $\hat{c}$.

Mask losses consist of two terms. One is the binary cross-entropy loss $\mathcal{L}_{\text{BCE}}$ between the volume-rendered mask and estimated foreground object mask. The second one is the background loss $\mathcal{L}_{\text{Background}}$ from NeRD [11], which forces all samples for rays cast towards the background to be 0. We combine these losses as the mask loss: $\mathcal{L}_{\text{Mask}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Background}}$

Regularization losses We compute the gradient of the density to estimate the surface normals. We use the normal direction loss $\mathcal{L}_{\text{ndir}}$ from [55] to constrain the normals to face the camera until the ray reaches the surface. This helps in providing sharper surfaces without cloud-like artifacts.

BRDF losses. The task of joint estimation of BRDF and illumination is quite challenging. For example, we observe that the illumination can 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. As our image collections have multiple illuminations, we can force the base color $\boldsymbol{b}_c$ to replicate the pixel color from the images. This way, a mean color over the dataset is learned and prevents falling into the local minima. We leverage the Mean Squared Error (MSE) for this: $\mathcal{L}_{\text{Init}} = \mathcal{L}_{\text{MSE}}(\boldsymbol{C^s}, \boldsymbol{b}_c)$. Additionally, we find that a smoothness loss $\mathcal{L}_{\text{Smooth}}$ for the normal, roughness, and metallic parameters similar to the one used in UNISURF [45] further regularizes the solution.

Overall network and camera losses. The final loss to optimize the decomposition network is then defined as $\mathcal{L}_{\text{Network}} = \lambda_b \mathcal{L}_{\text{Image}}(\boldsymbol{C^s}, \tilde{\boldsymbol{c}}) + (1-\lambda_b)\mathcal{L}_{\text{Image}}(\boldsymbol{C^s}, \hat{\boldsymbol{c}}) + \mathcal{L}_{\text{Mask}} + \lambda_a \mathcal{L}_{\text{Init}} + \lambda_{\text{ndir}}\mathcal{L}_{\text{ndir}} + \lambda_{\text{Smooth}}\mathcal{L}_{\text{Smooth}} + \lambda_{\text{Dec Smooth}}\mathcal{L}_{\text{Dec Smooth}} + \lambda_{\text{Dec Sparsity}}\mathcal{L}_{\text{Dec Sparsity}}$. Here, λ_b and λ_a are the optimization scheduling variables. Furthermore, the camera posterior scaling is applied to these losses. For the camera optimization, we leverage the same losses as described in $\mathcal{L}_{\text{Network}}$. However, as the camera should always be optimized, we do not apply posterior scaling to the losses when optimizing cameras. This enables cameras that are not aligned well to be optimized and allows each camera to leave the local minima. Here, we only calculate the mean loss, and therefore badly initialized camera poses can still recover over the training duration. Additionally, we define an $\mathcal{L}_1$ loss on the look-at vector $\boldsymbol{r}_j$ to constrain the camera pose to look at the object. The loss is defined as $\mathcal{L}_{\text{lookat}}$. We

also use a volume padding loss, which prevents cameras from going too far into our volume bound v: $\mathcal{L}_{\text{Bounds}} = \max((v - |\boldsymbol{t}_j|)^2, 0)$.

Optimization scheduling. Fig. 4 shows the optimization schedule of different loss weights. We use three fading λ variables, which smoothly transition the optimization schedule. The λ_c is mainly used for increasing the image resolution and reducing the number of active multiplex cameras. The direct color $\tilde{\boldsymbol{c}}$ optimization is faded to the BRDF optimization using λ_b and some losses are scaled by λ_a as defined earlier. Furthermore, we perform the BARF [35] frequency annealing in the early stages of the training and delay the focal length optimization to the later stages of the training. We use two different optimizers. The networks are optimized by an Adam optimizer with a learning rate of 1e-4 and exponentially decayed by an order of magnitude every 300k steps. The camera optimization is performed with a learning rate of 3e-3 and exponentially decayed by an order of magnitude every 70k steps. Further details of the optimization schedule are available in the supplements.

	Poses Not Known		Poses Available	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
BARF-A	16.9	0.79	19.7	0.73
SAMURAI	**23.46**	**0.90**	**22.84**	**0.89**
NeRD	—	—	26.88	0.95
Neural-PIL	—	—	27.73	0.96

Table 1: **Novel View Synthesis on varying illumination datasets.** We split our datasets into those where we have poses, and highly challenging ones where the poses were not recoverable with classical methods. SAMURAI achieves considerably better performance compared to BARF-A. For reference, we also show the metrics from NeRD and Neural-PIL which require GT poses and do not work on images with unknown poses.

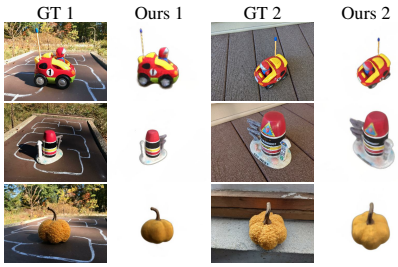

Figure 5: **SAMURAI-Dataset Novel Views.** Here, we show novel views and illuminations from the SAMURAI datasets alongside our reconstruction. The illumination conditions are accurately reproduced.

Method	PSNR↑	SSIM↑
w/o Camera Multiplex	23.01	0.87
w/o Posterior Scaling	23.51	0.88
w/o Coarse 2 Fine	22.73	0.83
w/o Random Fourier Offsets	24.01	0.91
w/o Regularization	21.77	0.86
Full	**24.31**	**0.92**

Table 2: **Ablation study.** view synthesis and relighting results on two scenes (Garbage Truck and NeRD car) show that ablating any of the proposed aspects of SAMURAI can results in worse results demonstrating their importance.

Method	Pose Init	PSNR↑	SSIM↑
BARF	Directions	14.96	0.47
GNeRF	Random	20.3	0.61
SAMURAI	Directions	**21.08**	**0.76**
NeRD	GT	23.86	0.88
Neural-PIL	GT	23.95	0.90

Table 3: **Novel View Synthesis on single illumination datasets.** For scenes under single illumination the poses are easily recoverable. Furthermore, we can now compare with GNeRF which does not require pose initialization. As seen our method achieves a good performance. For reference the view synthesis metrics with NeRD and Neural-PIL that use GT known poses are also shown.

4 Experiments

Datasets For evaluations, we created new datasets of 8 objects captured under unique illuminations and locations and a few different cameras. We refer to this dataset as the SAMURAI dataset. The images reflect the practical and challenging input scenario we are targeting in this work. Common methods such as COLMAP fail to estimate correspondences and camera poses for this dataset. Therefore, we cannot run methods that require poses on this dataset. Additionally, we evaluate on 2 CC-licensed image collections from online sources of the statue of liberty and a chair. We also use the 3 synthetic and 2 real-world datasets of NeRD [11] under varying illumination, where poses are available. Lastly, to showcase the performance with other methods, we use the 2 real-world datasets from NeRD, which are taken under fixed illumination. In total, we evaluate SAMURAI on 17 scenes. Please refer to the supplementary material for an overview of the SAMURAI datasets.

Baselines. Currently, there exist no prior art that can tackle varying illumination input images while jointly estimating camera poses. So, we compare with a modified BARF [35] technique, which

Figure 6: **Comparison with BARF-A.** When comparing novel view synthesis and relighting results of SAMURAI (top) with BARF-A (bottom), SAMURAI produces more accurate camera poses and captures the object better. BARF-A is sometimes unable to recover the shape and poses.

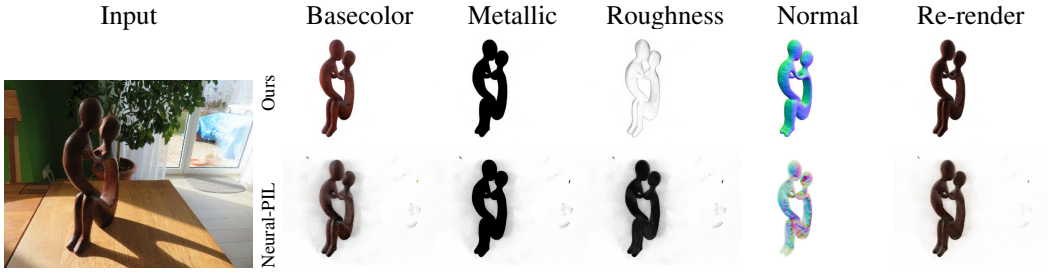

Figure 7: **Comparison with Neural-PIL decomposition.** Notice our accurate pose alignment, plausible geometry, reduced floaters and accurate BRDF decomposition, compared to Neural-PIL, even without relying on near perfect poses.

can store per-image appearances in a latent vector. We call this baseline BARF-A. Additionally, on scenes with fixed illumination, we can compare with GNeRF [39] and also the regular BARF. On the datases where poses are easily recovered or given, we can also compare with NeRD [11] and Neural-PIL [12], which require known, near-perfect camera poses. We supply BARF and BARF-A with the same pose initialization as SAMURAI. Technically, one could compare with NeRS [61] as well. However, due to the implementation, NeRS is optimized to run on a low number of images. The method runs out of memory on datasets with many images.

Evaluation. We perform novel view synthesis using learned volumes and use standard PSNR and SSIM metrics w.r.t. ground-truth views. We held-out every 16th image for testing. For evaluation purposes, we optimize the cameras and illuminations on the test images but did not allow the test images to affect the main decomposition network or camera training.

Ablation Study. We perform an ablation study where we ablate different aspects of the SAMURAI model to analyze their importance. Table 2 shows the novel view synthesis average metrics on the Garbage Truck collection from the SAMURAI dataset and the synthetic car dataset from NeRD [11]. Metrics show that regularization and the coarse to fine optimization are the most significant contributing factors to the final reconstruction quality. The multiplex cameras and the posterior scaling also improve the reconstruction quality, stabilizing the training and preventing cameras to get stuck in local minima. Visual comparisons of the specific ablations are available in the supplements.

Results on varying illumination datasets. We divide the varying illumination datasets into the ones without GT poses and ones with accurate camera poses (either via GT or COLMAP). Since these datasets have varying illumination, we need to perform both view synthesis and relighting to obtain target views. Table 1 shows the metrics computed w.r.t. test views. Results show that we considerably outperform BARF-A in both PSNR and SSIM metrics while at the same time solving a more challenging BRDF decomposition task. Visual results in Fig. 6 also clearly demonstrate better view synthesis and relighting results compared to BARF-A. BARF-A failed to align the camera poses, whereas SAMURAI achieves more accurate camera poses and drastically improved reconstruction quality. Only slightly perturbed poses are leveraged as starting positions in the original BARF method. Our coarse pose initialization is too noisy for the method to work accurately. SAMURAI overcomes this issue with the camera multiplex and other optimization strategies.

Fig. 7 shows the visual comparison of BRDF decompositions on MotherChild dataset from NeRD [11] along with the corresponding results from Neural-PIL [12]. In general, our method

can decompose the scene even from unknown camera poses. SAMURAI also produces fewer floating artifacts and creates a more coherent surface. The roughness parameter is also more plausible in our result, as the object is rough, whereas Neural-PIL estimated a near mirror-like surface.

Further results from the SAMURAI dataset are shown in Fig. 5. We show novel views and relighting results w.r.t the target test views. Visual clearly show that SAMURAI can recover the pose and provide a consistent illumination w.r.t the ground-truth target views.

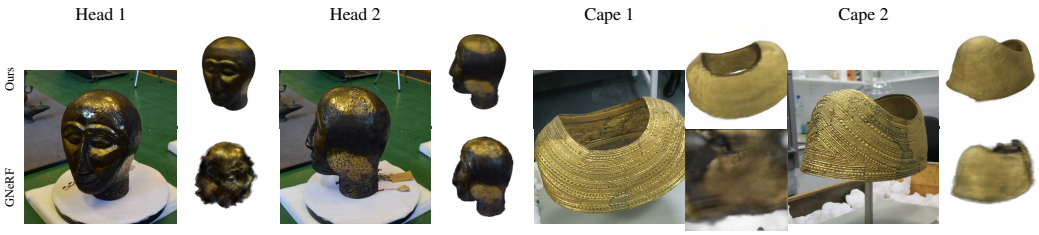

Figure 8: **Comparison with GNeRF.** SAMURAI view synthesis results (top) clearly outperforms those of GNeRF (bottom) in terms of pose estimation. Due the discriminator training GNeRF is sometime capable of producing higher texture detail (Head 2). However, it can also fail and the pose can get stuck in a local minima (Head 1).

Results on fixed illumination datasets. For image collections captured under fixed illumination, we can compare with more techniques. We compare with GNeRF and the default BARF. We additionally can compare with Neural-PIL [12] and NeRD [11] on the near-perfect camera poses recovered from COLMAP. Results in Table 3 show that SAMURAI outperforms the baselines BARF and GNeRF and is also close to Neural-PIL and NeRD that uses GT camera poses. Fig. 8 shows sample view synthesis results of SAMURAI and GNeRF on sample single illumination datasets. GNeRF does not require a rough pose initialization. Visuals indicate better results with SAMURAI compared to GNeRF. Due to the discriminator-based training, GNeRF produces a higher detailed texture. The cape example is especially taxing for both methods, as the fine detail requires perfect alignment for successful reconstruction.

Applications. One of the contributions of this work is the extraction of explicit mesh with material properties from the learned neural reflectance volume. The resulting mesh can be realistically placed in an Augmented Reality (AR) scene or in a 3D game. In addition, one could edit the BRDF materials on the recovered mesh. See Fig. 1 for sample results of these applications, where our recovered 3D assets blend well in a given 3D scene.

Limitations. SAMURAI achieves large strides in the decomposition of in-the-wild image collections compared to prior art. However, we still rely on rough pose initialization. GNeRF proposes a reconstruction technique without any pose initialization but it fails on the challenging in-the-wild datasets. Furthermore, SAMURAI produces slightly blurry textures. This is especially noticeable in the cape scene in Fig. 8. Here, the cape has a repeating, high-frequent texture. Reconstruction of this high-frequency texture requires near-perfect camera poses. Since this dataset is in a single location and illumination, COLMAP-based pose estimation will outperform SAMURAI based pose alignment. However, SAMURAI enables the reconstruction of highly challenging datasets of online image collections where COLMAP completely fails. Our BRDF and illumination decomposition is also not capable of modelling shadowing and inter-reflections. As we mainly tackle object decomposition the shadows and inter-reflections are not crucial. Removing the need for pose initialization along with modeling shadows and inter-reflections form an important future work.

5 Conclusion

SAMURAI is a carefully designed optimization framework for joint camera, shape, BRDF and illumination estimation. It can work on in-the-wild image collections captured in varying background and illuminations and with multiple cameras. Results on existing and our new challenging dataset demonstrate good view synthesis and relighting results, where several existing techniques fail. In addition, our mesh extraction allows the resulting 3D assets to be readily used in several graphics applications such as AR/VR, gaming etc.

References

[1] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. In *ACM Transactions on Graphics (ToG)*, 2018.

[2] Rachel Albert, Dorian Yao Chan, Dan B. Goldman, and James F. O'Brian. Approximate svBRDF estimation from mobile phone video. In *Eurographics Symposium on Rendering*, 2018.

[3] Louis-Philippe Asselin, Denis Laurendeau, and Jean-François Lalonde. Deep SVBRDF estimation on real materials. In *International Conference on 3D Vision (3DV)*, 2020.

[4] 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. *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[5] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *ArXiv e-prints*, 2020.

[6] 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 (ECCV)*, 2020.

[7] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] Mark Boss and Hendrik P.A. Lensch. Single image brdf parameter estimation with a conditional adversarial network. In *ArXiv e-prints*, 2019.

[9] Mark Boss, Fabian Groh, Sebastian Herholz, and Hendrik P. A. Lensch. Deep Dual Loss BRDF Parameter Estimation. In *Workshop on Material Appearance Modeling*, 2018.

[10] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot spatially-varying BRDF and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[11] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. NeRD: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[12] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[13] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ArXiv e-prints*, 2022.

[15] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL `http://www.blender.org`.

[17] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1982.

[18] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. In *ACM Transactions on Graphics (ToG)*, 2018.

[19] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. In *Eurographics Symposium on Rendering*, 2019.

[20] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. In *Eurographics Symposium on Rendering*, 2020.

[21] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2019.

[22] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without key-points. In *European Conference on Computer Vision (ECCV)*, 2020.

[23] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[24] Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. Generative modelling of BRDF textures from flash images. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2021.

[25] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[26] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncalibrated neural inverse rendering for photometric stereo of general surfaces. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[27] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[28] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. NeROIC: Neural object capture and rendering from online image collections. *ArXiv e-prints*, 2022.

[29] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 2004.

[30] Hendrik P. A. Lensch, Jan Kautz, Michael Gosele, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Conference on Rendering*, 2001.

[31] Hendrik P.A. Lensch, Jochen Lang, M. Sa Asla, and Hans-Peter Seidel. Planned sampling of spatially varying BRDFs. In *Computer Graphics Forum*, 2003.

[32] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. In *ACM Transactions on Graphics (ToG)*, 2017.

[33] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *European Conference on Computer Vision (ECCV)*, 2018.

[34] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and SVBRDF from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[35] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[36] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[37] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (ToG)*, 2019.

[38] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[39] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. GNeRF: GAN-based Neural Radiance Field without Posed Camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[40] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[41] Ben Mildenhall, Pratul Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[42] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 2022.

[43] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Mueller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[44] Giljoo Nam, Diego Gutierrez, and Min H. Kim. Practical SVBRDF acquisition of 3d objects with unstructured flash photography. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018.

[45] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. In *Pattern Recognition*, volume 106, 2020.

[48] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and SVBRDF estimation. In *European Conference on Computer Vision (ECCV)*, 2020.

[49] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[51] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixel-wise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[52] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[53] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[55] 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. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[56] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[57] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[58] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *ArXiv e-prints*, 2021.

[59] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Single image surface appearance modeling with self-augmented cnns and inexact supervision. *Computer Graphics Forum*, 2018.

[60] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[61] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[62] Jianzhao Zhang, Guojun Chen, Yue Dong, Jian Shi, Bob Zhang, and Enhua Wu. Deep inverse rendering for practical object appearance scan with uncalibrated illumination. In *Advances in Computer Graphics*, 2020.

[63] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical Gaussians for physics-based material editing and relighting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[64] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2021.

[65] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image GANs meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *International Conference on Learning Representations (ICLR)*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] No directly negative societal impacts are easily apparent from our method.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]