SNeS: Learning Probably Symmetric Neural Surfaces from Incomplete Data

Eldar Insafutdinov*, Dylan Campbell*, João F. Henriques, and Andrea Vedaldi

University of Oxford, Oxford OX1 3PJ, United Kingdom
{eldar,dylan,joao,vedaldi}@robots.ox.ac.uk

Abstract. We present a method for the accurate 3D reconstruction of partly-symmetric objects. We build on the strengths of recent advances in neural reconstruction and rendering such as Neural Radiance Fields (NeRF). A major shortcoming of such approaches is that they fail to reconstruct any part of the object which is not clearly visible in the training image, which is often the case for in-the-wild images and videos. When evidence is lacking, structural priors such as symmetry can be used to complete the missing information. However, exploiting such priors in neural rendering is highly non-trivial: while geometry and non-reflective materials may be symmetric, shadows and reflections from the ambient scene are not symmetric in general. To address this, we apply a soft symmetry constraint to the 3D geometry and material properties, having factored appearance into lighting, albedo colour and reflectivity. We evaluate our method on the recently introduced CO3D dataset, focusing on the car category due to the challenge of reconstructing highly-reflective materials. We show that it can reconstruct unobserved regions with high fidelity and render high-quality novel view images.

Keywords: 3D reconstruction · Novel view synthesis · Neural rendering

1 Introduction

Photogrammetry has made substantial progress with recent advances in neural rendering [29]. Given a collection of posed images of an object, we can now use techniques such as COLMAP [25] and NeRF [18] to learn photo-realistic models of the object from which novel views can be generated. Extensions such as NeuS [32] and VolSDF [38] can also accurately recover the 3D shape of the object. Many of these advances arise from using neural networks to represent the complex functions that describe the geometry and reflectance of the object.

Despite such successes, significant practical limitations remain. While networks often have excellent generalisation capabilities, in methods such as NeRF and NeuS they are overfitted to individual scenes, such as a single 3D object. As a result, such networks generalise poorly and are unable to predict the parts of the object that are not visible in the training images; instead, they require a

* Both authors contributed equally to this research.

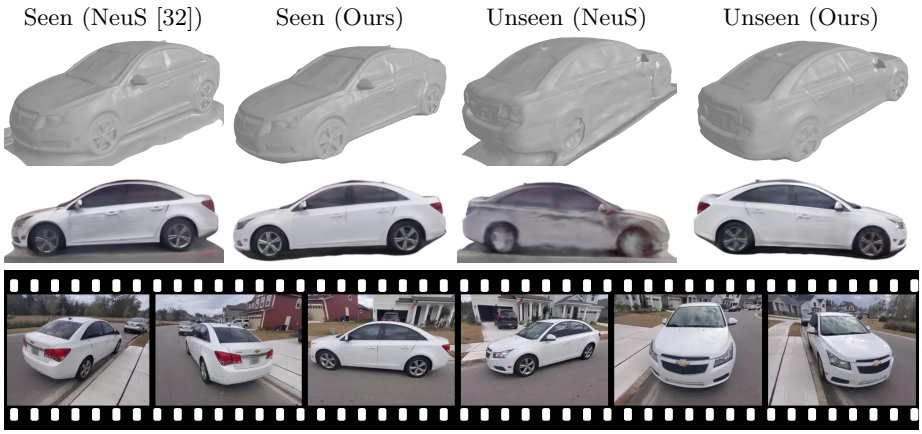


Fig. 1. From a sequence of frames that view a car in passing, our Symmetric Neural Surfaces (SNeS) model simultaneously learns the parameters of a symmetry transformation from the data and applies the symmetry as a soft constraint to reconstruct the model, despite the significantly different view densities between the seen and unseen sides. The learned symmetry allows SNeS to share information across the model, resulting in more accurate reconstructions and higher-fidelity novel synthesised views.

large number of views capturing uniformly on all sides of the object. This prevents applications in many realistic scenarios where only a limited and biased set of views is available, such as egocentric video or self-driving vehicles.

Bilateral symmetry is a strong geometric prior that applies approximately to many man-made and natural objects, and can be used to extrapolate beyond the field of view. Unfortunately, symmetry is not directly applicable to current neural renderers, because they entangle potentially symmetric parts of the model (geometry, material) with ambient illumination and view-dependent effects (shadows, specularity, and reflections), which are not symmetric. Our proposed approach, named *Symmetric Neural Surfaces* (SNeS), decomposes a neural renderer's colour model into several components: material albedo (absorption), reflectivity, diffuse lighting, and reflected lighting. These components are combined linearly, inspired by Phong shading [23], and are modelled by neural networks with different input constraints to ensure that they factorise correctly. For example, albedo only depends on the position and not on the viewpoint. During training, we encourage symmetry for only a subset of these components, albedo and reflectivity, which are material-dependent. We also apply symmetry to the geometry model, which is a neural surface model based on a signed distance function (SDF) [39]. Given the emphasis on bilateral symmetry and highly-reflective materials, our experiments are focused on vehicle reconstruction, which presents these unique challenges. Our contributions are:

- an algorithm for reconstructing objects with arbitrary learned symmetries of a pre-defined type from incomplete observations;
- a technique for disentangling symmetric and asymmetric appearance; and

– a prior for handling violations of geometry and material symmetry.

We demonstrate high fidelity of reconstruction, both in visual appearance and in the accuracy of surface geometry, for parts of the objects that are unseen during training. Our method achieves state-of-the-art results on the CO3D dataset [24], improving the geometry estimates considerably compared to the baselines, especially on sequences where the view density between sides is unbalanced.

2 Related Work

The field of neural volume rendering has expanded rapidly in the last two years, with increasing photo-realism and reconstruction quality. We focus on the closest works, and refer readers to recent review papers for a complete account [28,29].

Neural volume rendering and reconstruction. Neural Radiance Fields (NeRF) [18] and related approaches [15,41,3,16,40,33] generate images via a physically-based rendering process, where a ray is traced into the volume and neural network estimates of colour and density at sample points are integrated to render the pixel colour. With careful network design or regularisation, such a model will be able to accurately reconstruct the scene's geometry as well as modelling view-dependent effects. NeRF also introduced positional encoding, allowing MLPs to represent high frequency signals without increasing network capacity. Our rendering pipeline is similar, but extended to model symmetries.

Many works investigate more sophisticated lighting models that reason about the transport and scattering of light through the volume, allowing relighting and material editing [4,27,5,6,43,31]. For example, NeRFactor [43] converts a pre-trained NeRF model into a surface model, and optimises MLPs to represent light source visibility, surface normals, albedo, and the BRDF at any point on the surface, in addition to environment lighting, factoring appearance into material and lighting. Ref-NeRF [31], in contrast, trains a NeRF-like model from scratch, but replaces its parametrisation of outgoing radiance with one of reflected radiance to better model light transport, and estimates surface roughness to interpolate between blurry and sharp reflections. Our model also decomposes appearance into material properties and lighting, using a Phong colour model [23] and a loss that encourages the diffusely-lit albedo of a surface point to match the ground truth on average, integrating over viewing directions. Unlike existing work, this is motivated by symmetry learning, rather than editing applications, since lighting is typically asymmetric and impedes symmetry learning if ignored.

Many volume rendering approaches [18,41,3] attempt to concentrate their samples near surfaces, e.g., by using stratified sampling. Hybrid surface–volume representations [39,20,2,32,38] take this further by modelling surfaces directly, albeit implicitly, using occupancy [17] or signed distance function (SDF) [21] networks, combined with volume rendering for modelling view-dependent appearance. This was motivated by the observation that NeRF, while able to handle sudden depth changes, is unable to learn high-fidelity surfaces from its implicit representation. IDR [39] represents the geometry as an SDF and uses a NeRF-like view-dependent head to estimate colour, which also receives the surface normal

to better disentangle geometry and appearance. However, the appearance network only receives one point per ray, at the first surface, which can cause the model to get stuck in local optima. UNISURF [20] relaxes this by using hierarchical sampling with root-finding in an occupancy field, allowing it to spread the gradient over multiple points, which nonetheless concentrate at the surface as training progresses. A similar approach is taken by NeuS [32] and VolSDF [38], which represent surfaces as the zero-level set of an SDF and explore approaches for mapping signed distances to opacities. Our work is a hybrid surface–volume approach of this type, since our aim is to reconstruct high-quality symmetric surfaces. However, unlike previous work, we exploit additional structure in the data by learning symmetries and use them to share information between views.

Symmetry in 3D reconstruction. Symmetry cues have been used extensively in reconstruction, with shape-from-symmetry enabling single-view reconstruction by using the reflected image as another view [11,19,8,13,10,30,26,22,7]. Symmetry detection has also been investigated [9,26]. Of particular relevance is the approach of Wu et al. [36,35,37], who use reflective and rotational symmetries to recover shape, material properties and lighting from single images. They enforce mirror symmetry by flipping internal representations of depth and albedo in image space, and estimate a confidence mask to allow asymmetries. Our work is inspired by this use of symmetry for reconstruction, and by the observation that asymmetric lighting must be removed to reason about appearance symmetries. However, we target the task of multi-view reconstruction, apply a soft symmetry constraint in 3D directly rather than in 2D, and learn the symmetry parameters to obviate the need for fronto-parallel images.

3 Disentangled Neural Rendering

In this section, we outline our disentangled neural rendering model that takes a collection of posed images and produces a signed distance function (SDF) of the geometry and an appearance model that can be queried from novel viewpoints. In the subsequent section, we show how this baseline model can be used to learn symmetric neural surfaces. A flowchart of our full model is shown in Fig. 2.

3.1 Disentangling Geometry and Appearance

Since the release of the NeRF model [18], there has been considerable research into improving the noisy surface reconstructions it obtains [39,32,20]. These have focused on replacing NeRF's density estimation network with a regularised SDF [39,32] or occupancy [20] network. We use NeuS [32] as our baseline, since it effectively disentangles geometry and appearance, and is able to model fine structures. For completeness, we recap the NeuS model now.

Given a set of images $\{\boldsymbol{I}_\ell\}$ with associated camera poses and intrinsic matrices, the task is to reconstruct the geometry and view-dependent appearance of the object or scene. The geometry is represented implicitly as a signed distance function with zero-level set $\{\boldsymbol{x}_i \in \mathbb{R}^3 \mid \phi_{\text{SDF}}(\boldsymbol{x}_i) = 0\}$ that coincides with

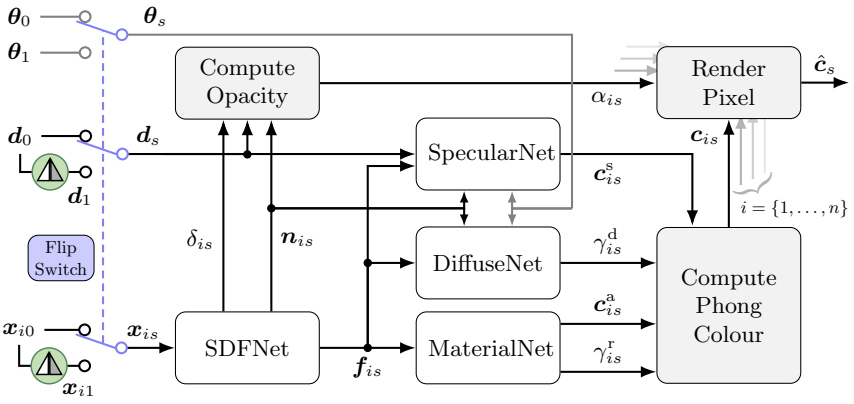


Fig. 2. The Symmetric Neural Surfaces (SNeS) model. For an input 3D point $\boldsymbol{x}_i$ and direction vector $\boldsymbol{d}$, the model estimates the geometry with an SDF network that generates a signed distance δ_i, a normal vector $\boldsymbol{n}_i$, and a feature vector $\boldsymbol{f}_i$. The first two are used to compute the opacity α_i according to Eq. (3), which assigns high opacity to points near surfaces. The feature vector is passed to the appearance networks to compute the material properties of albedo colour $\boldsymbol{c}^{\mathrm{a}}_i$ and reflectivity γ^{r}_i, and the lighting properties of diffuse shading γ^{d}_i and specular colour $\boldsymbol{c}^{\mathrm{s}}_i$. Lastly, the Phong model is used to compute the colour of the 3D point, and each sample along the ray is combined to render the pixel with colour $\hat{\boldsymbol{c}}$. The subscript s indicates whether the geometry, material and lighting components were computed with inputs that had undergone a symmetry transformation (1) or not (0), denoted by the triangular symbol. In each case, the lighting networks take different parameters θ, since lighting is typically asymmetric.

opaque surfaces in the scene. The map $\phi_{\mathrm{SDF}} : \mathbb{R}^3 \to \mathbb{R}$, which converts a 3D point $\boldsymbol{x}_i \in \mathbb{R}^3$ to a signed distance δ_i, is estimated with a fully-connected neural network. The view-dependent appearance is also estimated by fully-connected neural network layers, parametrising the function $\phi_{\mathrm{colour}} : \mathbb{R}^3 \times \mathbb{S}^3 \to \mathbb{R}^3$, which maps a 3D point and view direction $\boldsymbol{d}$ to a colour $\boldsymbol{c}_i \in [0, 1]^3$. Unlike NeuS, in this work the colour is estimated by a composition of functions to disentangle material and lighting properties, as shall be detailed in Sec. 3.2.

To learn these functions from images, physically-based rendering accumulates colours along a pixel ray. The ray is parametrised as $\{\boldsymbol{x}(t) = \boldsymbol{o} + t\boldsymbol{d} \mid t > 0\}$ for a ray with camera centre $\boldsymbol{o}$ and view direction $\boldsymbol{d}$. Rendering is performed by

$$\hat{\boldsymbol{c}}(\boldsymbol{o}, \boldsymbol{d}) = \int_0^\infty w(t)\, \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{d})\, \mathrm{d}t, \tag{1}$$

where w is a weight function that satisfies $w(t) \geqslant 0$ and $\int_0^\infty w(t)\mathrm{d}t = 1$, and should be high near opaque surfaces. In particular, w should attain a local maximum at the zero-level set of the SDF, and should decay with distance from the

camera. NeuS derives an appropriate weight function with these properties,

$$w(t) = \exp\left(-\int_0^t \rho(u)\,\mathrm{d}u\right)\rho(t), \text{ with } \rho(t) = \max\left\{0, \frac{-\frac{\mathrm{d}\sigma_\tau}{\mathrm{d}t}(\delta(t))}{\sigma_\tau(\delta(t))}\right\}, \qquad (2)$$

where $\rho(t)$ is the opaque density function and $\sigma_\tau(x) = (1+\exp(-\tau x))^{-1}$ is the sigmoid function parametrised by a learned scalar $\tau > 0$. As can be seen, NeuS does not predict the volume density directly like NeRF, but rather computes the density using the predicted signed distances in closed form. The learned scalar τ is proportional to the inverse standard deviation of the weight function (approximately a logistic density distribution), and controls the spread of the density about the zero-level crossing. It adapts to the data during training, resulting in a more concentrated distribution over time. This has two effects: colours of points near surfaces are assigned an increasingly high weight, and points are sampled increasingly close to surfaces, via an importance-sampling strategy. We refer the reader to Wang et al. [32] for a detailed derivation.

A discrete approximation of the weight function follows from the quadrature technique used in NeRF [18]. For n sampled points $\{\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{d} \mid i = 1,\ldots,n;\ t_i < t_{i+1}\}$ along the ray, their weights are given by

$$w_i = \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \text{ with } \alpha_i = \max\left\{0, \frac{\sigma_\tau(\delta(t_i)) - \sigma_\tau(\delta(t_{i+1}))}{\sigma_\tau(\delta(t_i))}\right\}, \qquad (3)$$

where the product term is the accumulated transmittance, and α_i is the discrete opacity. Note that to obtain the signed distance $\delta(t_i)$, the model uses the gradient (normal) vector to adjust the value of the nearest sampled signed distance. The final colour is then rendered as $\hat{\boldsymbol{c}} = \sum_{i=1}^{n} w_i \boldsymbol{c}_i$.

3.2 Disentangling Material and Lighting Properties

It is well-known that the NeRF colour formation model under-constrains the geometry, exhibiting a shape–radiance ambiguity where the training images can be perfectly explained by arbitrary geometry [41]. To impose a more realistic inductive bias on colour formation, without losing the flexibility and representation power of the unconstrained model, we disentangle the material and lighting properties using a Phong model [23]. As we shall show, this is also a necessary requirement for learning symmetric geometries from the data.

We separate the apparent colour into material and lighting properties. Specifically, albedo colour and reflectivity (or inverse roughness) represent material, and diffuse shading (assuming a white diffuse illuminant) and specular colour represent lighting. Here, we define the albedo as the average colour of a 3D point across viewpoints, under the scene lighting. Our colour formation model is

$$\boldsymbol{c}_i = f_{\text{Phong}}\left(\gamma_i^{\text{d}}, \boldsymbol{c}_i^{\text{a}}, \gamma_i^{\text{r}}, \boldsymbol{c}_i^{\text{s}}\right) = \gamma_i^{\text{d}}(\boldsymbol{x}_i, \boldsymbol{n}_i)\,\boldsymbol{c}_i^{\text{a}}(\boldsymbol{x}_i) + \gamma_i^{\text{r}}(\boldsymbol{x}_i)\,\boldsymbol{c}_i^{\text{s}}(\boldsymbol{x}_i, \boldsymbol{n}_i, \boldsymbol{d}_i), \qquad (4)$$

where $\boldsymbol{c}_i \in [0,1]^3$ is the estimated colour of the 3D point $\boldsymbol{x}_i$, $\gamma_i^{\text{d}} \in [0,2]$ is the diffuse lighting coefficient, $\boldsymbol{c}_i^{\text{a}} \in [0,1]^3$ is the lighting-invariant albedo colour of

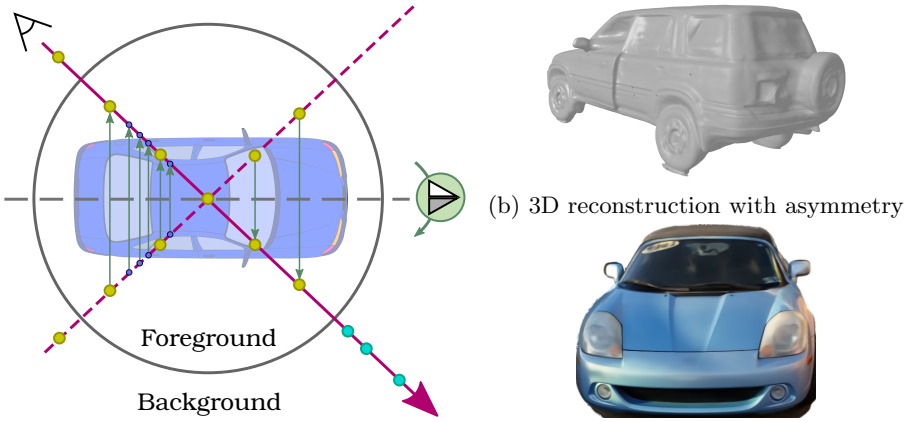
the material, $\gamma_i^{\mathrm{r}} \in [0,1]$ is the material reflectivity, and $\boldsymbol{c}_i^{\mathrm{s}} \in [0,1]^3$ is the specular colour of the reflected light. We see that the material properties depend on the geometry only, while the lighting depends additionally on the normal vector (diffuse lighting with self-shadows) and the viewing direction (specular colour). A more constrained parametrisation would learn the specular colour from the viewing ray reflected about the surface normal [31]. However, we found that this significantly over-smoothed the SDF model. While this colour model has the capacity to disentangle material and lighting properties, it needs to be regularised in order to do so. However, some objects, such as cars, are highly specular, making it undesirable to regularise the reflectivity. We instead encourage the diffusely-lit colour $\gamma_i^{\mathrm{d}} \boldsymbol{c}_i^{\mathrm{a}}$, rendered along the ray, to match the ground-truth colour, as we shall detail in the next section. This acts to average the colour of a surface location across all viewing directions. In practice, the appearance networks also depend on a feature vector from the SDF network, encoding the geometric context of the 3D point [39].

4 Symmetric Neural Surfaces

The model described thus far is unable to take advantage of known or suspected symmetries. We define a symmetry as an arbitrary coordinate transformation, especially an affine transformation such as a reflection, rotation, translation, or scaling, that confers an invariance. To share information across symmetries, we explicitly model and optimise the transformation parameters and use the map induced by the symmetry to aggregate information in 3D. However, not all information is symmetric. Cars, for example, tend to have a bilateral symmetry in their geometry and material properties, but the lighting of the scene is rarely symmetric. There are also exceptions that break the geometric and material symmetry, such as asymmetrically-positioned spare tyres (geometric) and stickers (material colour), as shown in Fig. 3 (c). Due to these real-world partial symmetries, it is best implemented as a soft constraint mediated by loss functions, rather than a hard constraint. Our framework has the flexibility to handle multiple arbitrary and localised symmetries. Spatially-restricted symmetries can be useful for modelling parts of an object or scene that are locally-symmetric, like the wheels of a car. In this work we focus on the common case of reflection (bilateral) symmetry, and localise the predicted symmetry to the unit sphere, to avoid symmetrising the background. A diagram of our approach is shown in Fig. 3. In the following, we denote the original ray, and everything computed with respect to it, as "source", and the symmetry-transformed ray as "transformed".

4.1 Parametrising Symmetry

We parametrise a symmetry as a coordinate transformation with form $T_{\mathrm{c}}^{-1} S T_{\mathrm{c}}$, where $T_{\mathrm{c}} = \left(\begin{smallmatrix} R_{\mathrm{c}} & \boldsymbol{t}_{\mathrm{c}} \\ \mathbf{0} & 1 \end{smallmatrix}\right)$ is the learned rigid transformation matrix from world coordinates to the canonical coordinates of the symmetry, defining its plane or axis, and S is the symmetry transformation in canonical coordinates. In this



(a) Symmetry transformation

(b) 3D reconstruction with asymmetry

(c) Novel view with asymmetry

Fig. 3. (a) Applying a symmetry transformation for physically-based rendering. The SNeS algorithm scales the object of interest to fit inside the unit sphere, where it is modelled by an SDF network with appearance heads, while the region outside the sphere is represented by a NeRF++ background model [41]. Here, yellow dots denote points sampled coarsely along the ray, small blue dots denote points importance-sampled near dominant surfaces, green dots denote points inversely-sampled in the background, and the horizontal dashed line denotes the plane of (reflection) symmetry. The symmetry induces a transformation on the point samples inside the sphere, and the transformed points are used to compute the geometry and material properties. These components are combined with the diffuse and specular lighting estimates from the source ray to form a colour estimate. If the symmetry holds, and is accurately estimated, the resulting colour should match the source colour. (b) SNeS reconstruction showing that geometry asymmetries (spare tyre, slightly-open door) are conserved. (c) SNeS novel view showing that appearance asymmetries (windshield sticker, lighting) are conserved.

work, we consider a single bilateral reflection symmetry about the XZ plane in canonical coordinates, and so the symmetry transformation matrix is given by $S = I - 2e_2e_2^\mathsf{T}$, where e_i is the i^{th} 4D unit basis vector. We apply the transformation to the source points $\boldsymbol{x}_0^\text{h}$ in homogeneous coordinates. The direction vectors $\boldsymbol{d}_0^\text{h}$ also undergo a symmetry transformation, although they are translation-invariant. This is implemented by the homogeneous coordinates, since directions are points at infinity with final coordinate equal to 0. Thus we obtain

$$\boldsymbol{x}_{i1}^\text{h} = T_\text{c}^{-1} S T_\text{c} \boldsymbol{x}_{i0}^\text{h} \tag{5}$$

$$\boldsymbol{d}_{i1}^\text{h} = T_\text{c}^{-1} S T_\text{c} \boldsymbol{d}_{i0}^\text{h}. \tag{6}$$

4.2 Learning Symmetric Geometry and Material

To encourage symmetric points to have the same geometry and material properties, we compute these quantities at both the source and transformed points, and compose them with predictions from the corresponding lighting model. Thus, for

each point, we obtain the source colour $\boldsymbol{c}_{i00} = \gamma^{\mathrm{d}}_{i0}\boldsymbol{c}^{\mathrm{a}}_{i0} + \gamma^{\mathrm{r}}_{i0}\boldsymbol{c}^{\mathrm{s}}_{i0}$ and the symmetry-transformed colour $\boldsymbol{c}_{i11} = \gamma^{\mathrm{d}}_{i1}\boldsymbol{c}^{\mathrm{a}}_{i1} + \gamma^{\mathrm{r}}_{i1}\boldsymbol{c}^{\mathrm{s}}_{i1}$. The lighting models for the source and symmetry-transformed paths do not share weights, since lighting is rarely symmetric. The resulting point colours are rendered along the ray, and compared to the ground-truth source pixel colour. If symmetry is valid at that pixel, and is accurately estimated, the error should be low. However, most objects and scenes are not perfectly symmetric, and so symmetry should not be enforced when better visual evidence is available. Therefore, we penalise the error of the symmetry-transformed colour at a discount compared to the source colour.

We also mix the source and the transformed components, generating hybrid colours. This acts to supervise the transformed lighting network to emulate the source lighting network, up to the symmetry transformation. Without these terms, the lighting networks may diverge, allowing the network to explain away deviations from symmetry as fake perturbations in lighting. Specifically, we form the hybrid point-wise colours $\boldsymbol{c}_{i01} = \gamma^{\mathrm{d}}_{i1}\boldsymbol{c}^{\mathrm{a}}_{i0} + \gamma^{\mathrm{r}}_{i0}\boldsymbol{c}^{\mathrm{s}}_{i1}$ and $\boldsymbol{c}_{i10} = \gamma^{\mathrm{d}}_{i0}\boldsymbol{c}^{\mathrm{a}}_{i1} + \gamma^{\mathrm{r}}_{i1}\boldsymbol{c}^{\mathrm{s}}_{i0}$, render these along the ray, and compute the colour error as before.

It is important to disentangle the material and lighting, since the former is usually asymmetric. This means that simply applying symmetry to the NeRF colour model would not work, since the colour is entangled with a systematic nuisance variable. Another strategy to help estimate the symmetry parameters is to learn the ground plane simultaneously and enforce orthogonality between the ground plane and the symmetry plane. To do so, we model the foreground as a joint SDF, which consists of the minimum of the object's SDF and a ground plane SDF (an infinite plane). This allows the SDF network to spend more capacity on the object, and enables ground removal without post-processing.

4.3 Loss functions

To fit our model, we minimise the error between the rendered and ground-truth pixels while regularising the SDF. No 3D supervision is used, beyond the known camera poses. We optimise the network parameters, symmetry transformation parameters, and the scalar τ that controls the variance of the density near surfaces. The per-pixel colour loss is given by

$$\mathcal{L}^{\mathrm{colour}}_{jk} = \tfrac{1}{3}\|\hat{\boldsymbol{c}}_{jk} - \boldsymbol{c}\|_1, \tag{7}$$

where $\boldsymbol{c}$ is the ground-truth colour and $\hat{\boldsymbol{c}}_{jk}$ is the predicted colour. The indices jk indicate whether the colour prediction uses the source or symmetry-transformed geometry and material properties (j), and lighting (k). This is the mechanism by which symmetry is encouraged in regions with visual evidence to the contrary.

We also use two additional losses with the same form as Eq. (7). The first is a diffuse colour loss $\mathcal{L}^{\mathrm{diffuse}}_{jk}$ where the predicted colour is rendered without the specular components, that is, the pixel colour is rendered from point colours $\boldsymbol{c}^{\mathrm{diffuse}}_i = \gamma^{\mathrm{d}}_i\boldsymbol{c}^{\mathrm{a}}_i$. This encourages the network to disentangle the diffuse and specular components, setting the diffuse colour of a given surface location to the average colour across all viewing directions. This is important for symmetry,

since the specular colour is usually not symmetric, so assisting the network to disentangle it can speed up convergence. The second is a symmetric lighting loss $\mathcal{L}_{jk}^{\text{lighting}}$ that applies a weak prior to the model to prefer symmetric lighting in the absence of contrary evidence. It applies the same colour loss as Eq. (7), but with the source lighting networks receiving symmetry-transformed inputs. This acts to apply symmetric lighting, which is generally incorrect, except at midday. Nonetheless, in the absence of image evidence, this prior provides a more naturalistic appearance. However, this loss should not be applied for quantitative analysis of unseen sides, because applying the symmetric lighting model is likely to be more detrimental than applying a baseline lighting model. For example, it may apply direct sunlight and specular reflections on the shadowed side of the object, which may look qualitatively convincing, but will be quantitatively poor.

Finally, we regularise the SDF network by applying an Eikonal loss [12] at the n sampled points along the ray, which encourages a unit gradient SDF:

$$\mathcal{L}_{j}^{\text{eikonal}} = \frac{1}{n}\sum_{i}^{n}(\|\nabla\phi_{\text{SDF}}(\boldsymbol{x}_{ij})\|_2 - 1)^2. \tag{8}$$

The total per-pixel loss is given by

$$\mathcal{L} = \sum_{j,k}(1 + (\lambda\!\!\!^{-} - 1)j)\left(\mathcal{L}_{jk}^{\text{colour}} + \lambda^{\text{d}}\mathcal{L}_{jk}^{\text{diffuse}} + \lambda^{\text{l}}\mathcal{L}_{jk}^{\text{lighting}} + \lambda^{\text{e}}\mathcal{L}_{j}^{\text{eikonal}}\right), \tag{9}$$

where $\lambda\!\!\!^{-} \in [0, 1]$ is the symmetricity factor that determines a prior on how symmetric an object or scene is expected to be, and the other λ factors denote the weights assigned to the remaining losses.

5 Results

5.1 Experimental Setup

Dataset. We evaluate our method on the cars subset of the recent Common Objects in 3D (CO3D) dataset [24], released under the BSD License. CO3D is a large-scale multi-view image dataset with ground-truth camera pose, intrinsics, depth map, object mask, and 3D point cloud annotations, collected in-the-wild by outdoor video capture. This real-world dataset is particularly challenging for reconstruction algorithms, having highly reflective (non-Lambertian) and low-textured surfaces, such as mirrors, dark windows, and metallic paint. Moreover, only 64% of the test sequences circumnavigate the object, with many seeing only one side of the car. This incomplete data motivates the use of symmetry for completing the reconstruction of partially-symmetric objects. Additional nuisance factors include significant motion blur from the handheld cameras, auto-exposure, and adverse weather, including fog and rain. One of the consequences of this challenging data is that the ground-truth point clouds and depth maps are sparse, very noisy, and contain many outliers, and 8% of test object masks

entirely miss the object. This makes evaluating the reconstructed geometry, especially fine details, quite difficult. For the task of single-scene 3D reconstruction and novel view synthesis, the 'car' category has 22 test scenes with 102 frames each. We present results on other partly-symmetric categories in the appendix.

Metrics. We report five metrics to measure visual and geometric quality: the peak signal-to-noise ratio (PSNR), the mean squared colour error (MSE), and the perceptual LPIPS distance [42] between the masked predicted and ground-truth novel-view images; the mean absolute error (MAE) between the masked predicted and ground-truth depth maps; and the intersection-over-union (IoU) between the predicted and ground-truth object masks.

Baselines. We compare with two state-of-the-art baselines for novel-view synthesis and 3D reconstruction in unbounded, real-world scenes: NeRF++ [41] and NeuS [32]. We do not compare with the state-of-the-art classical multi-view stereo algorithm COLMAP [25], because the dataset's ground-truth point clouds and depth maps were obtained using this algorithm and are extremely noisy and sparse for this reflective and low-texture category. We focus on two strong and well-regarded baselines to avoid the evaluation becoming prohibitively expensive (each baseline trains for at least 24h on a single GPU).

Implementation details. Following prior art [39,32], we implement the SDF network ϕ_{SDF} as an 8-layer MLP with hidden dimension 256, position-encoded inputs (6 frequencies) [18], and geometric initialisation for the network weights [1]. The material, diffuse, and specular networks are also implemented as MLPs with 4/2/4 hidden layers, with a 4-frequency positional encoding on the normal and view directions. NeRF++ [41] is used as the background model. We follow the hierarchical sampling strategy of NeuS [32] with 64 coarse, 64 fine, and 32 background samples per ray, with 1024 rays sampled per batch. We optimise the network with Adam [14] and an initial learning rate of 5e-4 and train for 300K iterations on a single GPU. Unless otherwise stated, we use the hyperparameters $[\lambda, \lambda^{\text{d}}, \lambda^{\text{l}}, \lambda^{\text{e}}] = [0.1, 0.01, 0.001, 0.1]$. Complete implementation details are reported in the appendix, and code is available at `github.com/eldar/snes`.

5.2 Random Test Split

For this experiment, we use the train–test split provided by the dataset for single scene experiments ("test_known" and "test_unseen") [24], assigned at random from the frames of the video. This evaluates the model's ability to interpolate between a dense set of views. This is the standard mode for evaluating novel view synthesis algorithms. Note that only 64% (14) of the video sequences entirely encircle the object of interest, with the remainder having coverage of as little as 135°. Our method is able to reconstruct the unseen sides, though we are unable to evaluate this as the requisite ground-truth is not present in the dataset. The results are shown in Tab. 1, and indicate that applying symmetry does not harm the performance of the baseline model, and indeed improves the geometry. This suggests that the model is able to learn the symmetry and integrate information from both sides of the object to improve the geometry estimate.

Table 1. Results on the random and structured test splits of the CO3D cars dataset [24]. We report the peak signal-to-noise ratio (PSNR), mean squared error (MSE), and LPIPS distance between the estimated and ground-truth masked images, the mean absolute error (MAE) between the estimated and ground-truth masked depth maps, and the intersection-over-union (IoU) of the estimated and ground-truth masks.

	Random Split (overlapping views)					Structured Split (biased views)				
	PSNR	MSE	LPIPS	MAE	IoU	PSNR	MSE	LPIPS	MAE	IoU
	RGB	RGB	RGB	Depth	Mask	RGB	RGB	RGB	Depth	Mask
Method	↑	↓	↓	↓	↑	↑	↓	↓	↓	↑
NeRF++[41]	21.4	0.007	0.407	0.222	–	13.9	0.041	0.581	0.177	–
NeuS [32]	**23.3**	**0.005**	0.355	0.108	0.523	13.4	0.046	0.556	0.105	0.566
SNeS (ours)	**23.3**	**0.005**	**0.348**	**0.086**	**0.787**	**14.1**	**0.039**	**0.503**	**0.077**	**0.906**

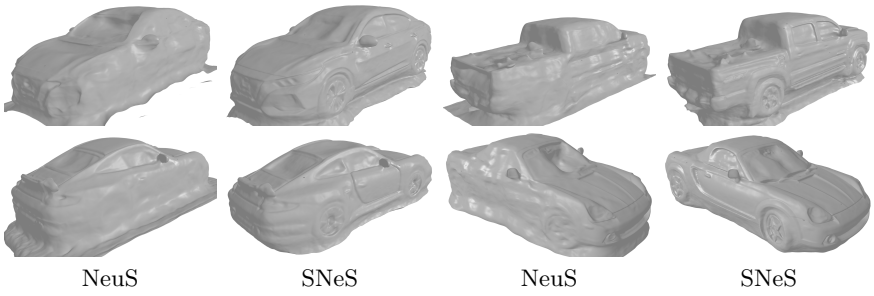


Fig. 4. Qualitative results on the structured test split of the CO3D cars dataset [24].

Fig. 5. Novel view renderings of the partly-observed (left) and fully-observed (right) sides. Top row: NeuS. Middle row: SNeS. Bottom row: SNeS albedo maps.

5.3 Structured Test Split

We propose a new train–test split that simulates the common situation where one side of an object is observed more thoroughly than the other. This tests the model's ability to handle variable view densities and incomplete information.

Table 2. Ablation study on a random subset of our structured test split of the CO3D cars dataset [24]. We report the peak signal-to-noise ratio (PSNR), mean squared error (MSE), and LPIPS distance between the estimated and ground-truth masked images, the mean absolute error (MAE) between the estimated and ground-truth masked depth maps, and the intersection-over-union (IoU) of the estimated and ground-truth masks.

Method	PSNR RGB ↑	MSE RGB ↓	LPIPS RGB ↓	MAE Depth ↓	IoU Mask ↑
SNeS (ours)	**14.3**	**0.0372**	**0.564**	0.0706	0.894
$+\mathcal{L}^{\text{lighting}}$	13.7	0.0425	0.585	**0.0685**	0.914
$-\mathcal{L}^{\text{diffuse}}$	**14.3**	**0.0372**	0.566	0.0722	**0.917**
$-\mathcal{L}^{\text{col}}$	13.7	0.0422	0.576	0.0782	0.906

Ground-truth Ours $+\mathcal{L}^{\text{lighting}}$ $-\mathcal{L}^{\text{diffuse}}$ $-\mathcal{L}^{\text{col}}$

Fig. 6. Qualitative ablation study. Novel view renderings of the unseen side.

To do so, we select the 14 test scenes where the camera circumnavigates the object, and define a test split that sets aside all camera poses within a 130° sector emanating from the object's centre, approximately perpendicular to the plane of bilateral symmetry. Thus, one side of the car is only seen obliquely. From the set aside poses, we systematically sample 8 test frames. This setting makes it possible for existing methods to reconstruct both sides of the object, but tests how well they are able to reconstruct the side that is viewed less fully. The results are shown in Tab. 1. Our method consistently outperforms the NeuS baseline on the novel view synthesis metrics and significantly improves the depth accuracy on the unseen side, validating the effectiveness of our approach. Qualitative comparisons are shown in Figs. 4 and 5, demonstrating high-fidelity reconstructions and synthesised views on the unseen side. We include additional high-resolution qualitative results in the supplementary material, including a comparison of the different appearance components (material and lighting).

5.4 Ablation Study

To investigate the effect of different components, we ablate our model's performance on 4 randomly selected scenes from our structured test split of the CO3D cars dataset, as shown in Tab. 2 and Fig. 6. We ablate with respect to the model without the lighting loss (ours), since this loss is designed to produce qualitatively convincing renders in the absence of image evidence, but is unlikely to be quantitatively accurate in those areas. We indeed see that the symmetric lighting loss has a detrimental effect on the image-based results, predominantly in situations where direct sunlight is applied to the shadowed side of the car and vice versa. However, the resulting renders are qualitatively preferable, as shown

at high resolution in the appendix. We verify that removing the diffuse colour loss harms the geometry, since it helps decouple the symmetric and asymmetric properties facilitating symmetry learning. Finally, we show that removing the symmetry loss $\mathcal{L}^{\text{col}}$ significantly reduces the visual and geometric quality.

6 Discussion and Limitations

One limitation of the approach is that it is only beneficial for objects or scenes with significant symmetries. However, this is not as restrictive as it might seem. While the natural world rarely has large-scale symmetries, they abound in the human environment, in architecture and object design. For example, out of the CO3D dataset, 90% of the categories have at least one major symmetry, such as ball, baseball bat, bench, bicycle, book, bottle, and bowl. More significant limitations of the approach, then, are that the type and number of symmetries must be specified in advance, that the symmetry has to be significant enough to be learnable from the data, and that the initialisation of the symmetry plane or axis must be good enough to avoid the network getting trapped in a local optimum. An alternative approach, such as multiple initialisations, may be necessary to prevent the latter in some cases. Another limitation of the approach is that it requires a significant number of views, even with the reductions facilitated by the symmetry. This is because it can be difficult to optimise the symmetry parameters, such as finding the reflection plane, without reasonable view coverage. This could be mitigated by learning about symmetries from a collection of scenes, such that a single view may be enough to partially constrain the symmetry plane parameters [44]. Our approach also relies on good camera estimates. While this requirement can be relaxed [34], additional unknown variables are likely to make the symmetry parameters more difficult to estimate. Finally, our approach does not explicitly handle symmetries that are present at different scales or resolutions. For example, a decorated cake or a pizza is symmetric at one scale, but may violate that symmetry when considering the finer details.

7 Conclusion

We have presented a 3D reconstruction and novel-view synthesis method for partly-symmetric objects, which learns symmetry parameters from a collection of posed images and uses the learned symmetry to share information across the model. This reduces the need for dense multi-view coverage of the object, making it suitable for use on in-the-wild data like the CO3D dataset. We demonstrated our algorithm on objects that exhibit bilateral symmetry at most locations—cars—and show that it can reconstruct unobserved regions with high fidelity.

Acknowledgements. We are grateful for support from Continental AG (E.I., D.C.), the European Research Council Starting Grant (IDIU 638009, E.I., D.C.), and the Royal Academy of Engineering (RF/201819/18/163, J.H.).

References

1. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)
2. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. arXiv preprint arXiv:2104.04532 (2021)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proc. ICCV. pp. 5855–5864 (2021)
4. Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hašan, M., Hold-Geoffroy, Y., Kriegman, D., Ramamoorthi, R.: Neural reflectance fields for appearance acquisition. arXiv preprint arXiv:2008.03824 (2020)
5. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: NeRD: neural reflectance decomposition from image collections. In: Proc. ICCV. pp. 12684–12694 (2021)
6. Chen, W., Litalien, J., Gao, J., Wang, Z., Fuji Tsang, C., Khamis, S., Litany, O., Fidler, S.: DIB-R++: learning to predict lighting and material with a hybrid differentiable renderer. In: NeurIPS. vol. 34 (2021)
7. Chen, X., Li, Y., Luo, X., Shao, T., Yu, J., Zhou, K., Zheng, Y.: Autosweep: Recovering 3d editable objects from a single photograph. IEEE Transactions on Visualization and Computer Graphics **26**(3), 1466–1475 (2018)
8. Fawcett, R., Zisserman, A., Brady, J.M.: Extracting structure from an affine view of a 3D point set with one or two bilateral symmetries. Image and Vision Computing **12**(9), 615–622 (1994)
9. Forsyth, D.A., Mundy, J.L., Zisserman, A., Rothwell, C.A.: Recognising rotationally symmetric surfaces from their outlines. In: Proc. ECCV. LNCS 588, Springer-Verlag (1992)
10. François, A.R., Medioni, G.G., Waupotitsch, R.: Mirror symmetry → 2-view stereo geometry. Image and Vision Computing **21**(2), 137–143 (2003)
11. Gordon, G.G.: Shape from symmetry. In: Intelligent Robots and Computer Vision VIII: Algorithms and Techniques. vol. 1192, pp. 297–308. SPIE (1990)
12. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: Proceedings of Machine Learning and Systems 2020, pp. 3569–3579 (2020)
13. Huynh, D.: Affine reconstruction from monocular vision in the presence of a symmetry plane. In: Proc. ICCV (Sep 1999)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)
16. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proc. CVPR. pp. 7210–7219 (2021)
17. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: Proc. CVPR. pp. 4460–4470 (2019)
18. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Proc. ECCV. pp. 405–421. Springer (2020)

19. Mukherjee, D.P., Zisserman, A., Brady, J.M.: Shape from symmetry – detecting and exploiting symmetry in affine images. Philosophical Transactions of the Royal Society of London **351**, 77–106 (1995)
20. Oechsle, M., Peng, S., Geiger, A.: UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: Proc. ICCV. pp. 5589–5599 (2021)
21. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: Proc. CVPR. pp. 165–174 (2019)
22. Phillips, C.J., Lecce, M., Daniilidis, K.: Seeing glassware: from edge detection to pose estimation and shape recovery. In: Robotics: Science and Systems. vol. 3, p. 3 (2016)
23. Phong, B.T.: Illumination for computer generated pictures. Communications of the ACM **18**(6), 311–317 (1975)
24. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: Proc. ICCV. pp. 10901–10911 (2021)
25. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proc. CVPR (2016)
26. Sinha, S.N., Ramnath, K., Szeliski, R.: Detecting and reconstructing 3d mirror symmetric objects. In: Proc. ECCV. pp. 586–600. Springer (2012)
27. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: NeRV: neural reflectance and visibility fields for relighting and view synthesis. In: Proc. CVPR. pp. 7495–7504 (2021)
28. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., et al.: State of the art on neural rendering. In: Computer Graphics Forum. vol. 39, pp. 701–727. Wiley Online Library (2020)
29. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. arXiv preprint arXiv:2111.05849 (2021)
30. Thrun, S., Wegbreit, B.: Shape from symmetry. In: Proc. ICCV. vol. 2, pp. 1824–1831. IEEE (2005)
31. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-NeRF: Structured view-dependent appearance for neural radiance fields. arXiv (2021)
32. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NeurIPS (2021)
33. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: IBRNet: learning multi-view image-based rendering. In: Proc. CVPR. pp. 4690–4699 (2021)
34. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF−−: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)
35. Wu, S., Makadia, A., Wu, J., Snavely, N., Tucker, R., Kanazawa, A.: De-rendering the world's revolutionary artefacts. In: Proc. CVPR (2021)
36. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In: Proc. CVPR (2020)
37. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. IEEE PAMI (2021). https://doi.org/10.1109/TPAMI.2021.3076536

38. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: NeurIPS (2021)
39. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS. vol. 33, pp. 2492–2502. https://nips.cc/Conferences/2020/ (2020)
40. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: PixelNeRF: neural radiance fields from one or few images. In: Proc. CVPR. pp. 4578–4587 (2021)
41. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: NeRF++: analyzing and improving neural radiance fields. arXiv:2010.07492 (2020)
42. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. CVPR. pp. 586–595 (2018)
43. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. ACM Transactions on Graphics (TOG) **40**(6), 1–18 (2021)
44. Zhou, Y., Liu, S., Ma, Y.: Nerd: Neural 3d reflection symmetry detector. In: Proc. CVPR. pp. 15940–15949 (2021)