SNeS: Learning Probably Symmetric Neural Surfaces from Incomplete Data

—Supplementary Material—

A Characterisation of Symmetries

While in the main paper we represent symmetry as a general coordinate transformation, or rather as a symmetry transformation in a canonical coordinate frame, that is, $T_c^{-1}ST_c$, here we provide a taxonomy of some specific symmetry types and how they can be implemented. Some terminology and formulae are adapted from Rosen [8] and Thrun & Wegbreit [10]. Note that here we only consider point transformations; direction vectors are invariant to translation and so only undergo the rotation or scaling parts of the transformation.

1. **Reflection symmetry:** whereby a point is mapped to another point (a bijection) on the opposite side of a plane, line or point.
   (a) Planar reflection: for a plane defined by a normal vector $\boldsymbol{n}$ and scalar distance from the origin Δ, a point $\boldsymbol{x}$ is mapped to
   $$\boldsymbol{x}' = \boldsymbol{x} - 2\boldsymbol{n}(\boldsymbol{n}^\mathsf{T}\boldsymbol{x} - \Delta) \tag{1}$$
   $$= (I - 2\boldsymbol{n}\boldsymbol{n}^\mathsf{T})\boldsymbol{x} + 2\Delta\boldsymbol{n}. \tag{2}$$
   (b) Line reflection: for a line defined by a point $\boldsymbol{p}$ and a normal $\boldsymbol{n}$, a point $\boldsymbol{x}$ is mapped to
   $$\boldsymbol{x}' = R^\mathsf{T} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} R(\boldsymbol{x} - \boldsymbol{p}) + \boldsymbol{p}, \tag{3}$$
   where the rotation is defined by the implicit equation $\boldsymbol{n} = R[0, 0, 1]^\mathsf{T}$. A minimal parametrisation can be constructed by defining the line by a direction vector $\boldsymbol{n}$ parallel to the line and a 2D offset $\Delta \in \mathbb{R}^2$ where the line intersects the plane, in the plane's coordinate system (4 DoF).
   (c) Point reflection: for a point $\boldsymbol{p}$, a point $\boldsymbol{x}$ is mapped to
   $$\boldsymbol{x}' = -(\boldsymbol{x} - \boldsymbol{p}) + \boldsymbol{p} = -\boldsymbol{x} + 2\boldsymbol{p}. \tag{4}$$
2. **Rotation symmetry:** whereby a point is mapped to n points (not including the source point) about an arbitrary axis with angle $\theta = 2\pi/(n+1)$. Thrun & Wegbreit [10] use the terminology *axial symmetry* to describe the limiting case as $n \to \infty$, such as a cylinder. For an axis defined by a point $\boldsymbol{p}$ and a normal $\boldsymbol{n}$, a point $\boldsymbol{x}$ is mapped to
   $$\boldsymbol{x}' = R^\mathsf{T} \begin{pmatrix} \cos\theta_k & \sin\theta_k & 0 \\ -\sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{pmatrix} R(\boldsymbol{x} - \boldsymbol{p}) + \boldsymbol{p}, \tag{5}$$

where the rotation is defined by the implicit equation $\boldsymbol{n} = R[0,0,1]^\mathsf{T}$, and the angle $\theta_k = k\theta$, for $k \in \{1,\dots,n\}$.

3. **Spherical symmetry:** whereby a point is mapped to a sphere. For a spherical symmetry with centre $\boldsymbol{p}$, a point $\boldsymbol{x}$ is mapped to

$$\boldsymbol{x}' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} (\boldsymbol{x} - \boldsymbol{p}) + \boldsymbol{p}, \tag{6}$$

for arbitrary angles α, β.

4. **Translation symmetry:** whereby a point is mapped to another point by applying a fixed translation t. For a translation t, a point $\boldsymbol{x}$ is mapped to $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$.
5. **Scale symmetry:** whereby a point is mapped to another point by a scaling operation. For scaling factors s_x and s_y, a point $\boldsymbol{x}$ is mapped to

$$\boldsymbol{x}' = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}. \tag{7}$$

Composite types can be formed from combinations of these basic types. We implement rotational and spherical symmetries in the same way as the bijective symmetries, except that we sample a single point at random from the resulting set, since our approach is point-based. More points could be sampled, at the expense of greater memory and computational requirements.

Multiple symmetries. As mentioned in Section 3, our framework has the flexibility to handle multiple symmetries, including different types (e.g., reflections and rotations). To do so, whenever we apply a symmetry transformation, we choose it at random from the set of pre-defined symmetries. For example, points sampled in a scene containing a table with two planar reflection symmetries are transformed according to one of those symmetries at random each iteration. This has the effect of encouraging all of the symmetries in the set without increasing the computational burden. We can further define relations between the symmetries, such as orthogonality constraints, which limit the degrees of freedom and so facilitate optimisation.

B Further Implementation Details

B.1 Architecture and Parameters

Following prior art [12, 11], we implement the SDF network ϕ_{SDF} as an 8-layer MLP with hidden dimension 256, position-encoded inputs (6 frequencies) [5], a skip connection at layer 4, and geometric initialisation for the network weights [1]. The latter provides a spherical prior for the SDF, which we squash into an ellipsoid using the bounding box dimensions. The material, diffuse, and specular networks are also implemented as MLPs with 4/2/4 hidden layers respectively, with a 4-frequency positional encoding on the normal (diffuse and

specular networks) and view (specular network) directions. The lighting networks have two sets of weights $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ to allow them to model source lighting and symmetry transformed lighting without any symmetry constraint. NeRF++ [13] is used as the background model, with default parameters.

We follow the hierarchical sampling strategy of NeuS [11] with 64 coarse, 64 fine, and 32 background samples per ray, with 1024 rays sampled per batch. The fine samples are obtained using importance sampling, given the density w estimated from the SDF values obtained under the coarse samples (Eq. 3 from the main paper). The peakiness of this distribution is controlled by the learned scalar τ (initialised to 20), being proportional to the inverse standard deviation of the weight function. The default hyperparameters are $[\lambda^{\text{K}}, \lambda^{\text{d}}, \lambda^{\text{l}}, \lambda^{\text{e}}] = [0.1, 0.01, 0.001, 0.1]$.

B.2 Optimisation

We optimise the network with Adam [3] and an initial learning rate of 5e-4, and train for 300k iterations on a single GPU, with an un-optimised implementation taking 31h on a NVIDIA V100. Recent work, such as the multi-resolution hash encoding [6], is likely to speed up training by orders of magnitude. We apply cosine annealing to reduce the learning rate gradually over training, reaching $0.05 \times$ LR at the end of training. We also ramp the learning rate of the SDF network, specular lighting network, symmetry parameters, and the inverse variance τ from 0 at the start of training, for the first 2500 iterations. On the other hand, the learning rate for the background, the diffuse lighting and the material networks are kept constant at the initial level. This warm-up period gives the background model a head-start, to avoid modelling the background with the foreground model (e.g., a sky-blue foreground blob above the car can entirely explain the training data in some scenes), and suppresses the specular network, which can otherwise explain away the geometry with arbitrary reflections. The latter only becomes a problem in NeRF-like models when the view density is much lower for some parts of the object. An alternative approach, which is sometimes able to reconstruct more crisp geometry, is to treat all parameters equally. However, this occasionally fails to correctly model highly-reflective surfaces that are only viewed obliquely, instead deforming the geometry to model the reflections in a process akin to an optical illusion or trompe-l'œil.

B.3 Initialisation

We initialise our coordinate system in the following way. Given a noisy, sparse, and outlier-ridden structure-from-motion (SfM) point cloud, we first filter the point cloud by removing point clusters or singletons further than a threshold (0.2m) from the biggest cluster, and then remove points with fewer than 16 neighbours within the threshold. Next, we fit a 3D bounding box by estimating the up direction from the camera. Then we flatten the point cloud along that dimension, extract the line with greatest support (using a threshold of 0.05) from the 2D point cloud using RANSAC [2], and rotate the point cloud so that

Table 1. Results for N scenes of the structured test split of the CO3D dataset [24] for *non-car categories*. We report the peak signal-to-noise ratio (PSNR) and LPIPS distance between the estimated and ground-truth masked images, the mean squared error (MSE) between the estimated and ground-truth masked depth maps, and the intersection-over-union (IoU) of the estimated and ground-truth masks.

		PSNR RGB ↑		LPIPS RGB ↓		MSE Depth ↓		IoU Mask ↑	
Category	N	NeuS	Ours	NeuS	Ours	NeuS	Ours	NeuS	Ours
toyplane	10	13.6	**15.1**	0.56	**0.51**	1.18	**0.61**	0.45	**0.56**
bench	8	**16.0**	14.8	**0.50**	**0.50**	0.46	**0.41**	0.43	**0.55**
chair	1	**18.6**	16.7	0.48	**0.47**	1.01	**0.44**	0.85	**0.88**
toytrain	1	9.60	**14.6**	0.58	**0.43**	0.14	**0.09**	**0.63**	0.62
toaster	1	17.8	**18.8**	0.50	**0.48**	3.70	**0.05**	0.82	**0.83**
motorcycle	1	11.0	**13.0**	0.63	**0.62**	**2.30**	3.88	0.70	**0.71**
skateboard	1	11.4	**13.0**	**0.68**	0.69	0.13	**0.11**	**0.20**	**0.20**
couch	1	11.5	**12.6**	**0.62**	0.63	1.25	**0.32**	**0.37**	0.35
parkingmeter	1	11.4	**15.3**	0.50	**0.42**	0.42	**0.36**	0.82	**0.83**
suitcase	1	13.5	**13.7**	**0.50**	0.51	0.23	**0.08**	**0.80**	**0.80**

the y axis is parallel to the extracted line. The rationale for this approach was that the SfM point clouds often only model one side of the object, and have highly-variable point densities. As a result, extracting the principal direction of the point cloud using a non-robust technique such as eigendecomposition results in poorly aligned bounding boxes. Composing these transformations, we obtain a new coordinate system and an axis-aligned bounding box. This is used to initialise the symmetry parameters (e.g., the reflection plane is taken as one of the principal planes), the ground plane (initialised at the height of the bottom of the bounding box), and the SDF ellipsoid prior (the spherical SDF prior is squashed according to the bounding box dimensions).

In our experiments, the 3D point clouds were provided by the benchmark dataset (CO3D [7]), which were originally obtained using structure-from-motion (COLMAP [9]), followed by filtering with PointRend [4] instance segmentation masks to identify the 3D points on the object. See the CO3D paper [7] for details.

C Results on Other CO3D Categories

In this section, we present results on non-car categories from the CO3D dataset [7]. We evaluate 10 non-car categories under the challenging setting where we take the minor sector (130°) of the structured split as training data and test on the unseen major sector. We test 1 scene per category, except for toyplane and bench, where we evaluated 10 and 8 scenes respectively. The results in Tab. 1 show that our method generalises to other symmetric categories. While SNeS always reconstructs the unseen side better than the baseline (NeuS [11]), neither method can predict the unseen lighting. Since our model generates plausible symmetric lighting in the absence of other evidence, this can result in shadow

Fig. 1. Reconstructions and renders of the unseen sides of 10 non-car categories of the CO3D dataset [7]. (Rows 1 & 5) Our reconstruction; (rows 2 & 2) ground-truth novel view; (rows 3 & 7) NeuS [11] render; (rows 4 & 8) our render.

being predicted when the unseen side is under direct sunlight. As a result, our model's PSNR can sometimes be worse than the implausible predictions of the baseline methods on the unseen side. See, for example, the suitcase in Fig. 1.

Table 2. Ablation study on a random subset of our structured test split of the CO3D cars dataset [7]. 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. The indices jk of the colour losses are 0 for source and 1 for symmetry-transformed geometry/material (j) and lighting (k).

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**
− ground pred.	**14.3**	**0.0372**	**0.558**	**0.0593**	0.762
$\lambda = 1$	13.7	0.0429	0.571	0.0627	0.873
$-\{\mathcal{L}^{\text{col}}_{01}, \mathcal{L}^{\text{col}}_{11}\}$	13.9	0.0408	0.587	0.0717	0.864
$-\{\mathcal{L}^{\text{col}}_{01}, \mathcal{L}^{\text{col}}_{10}\}$	13.6	0.0437	0.561	0.0732	0.913
$-\{\mathcal{L}^{\text{col}}_{01}, \mathcal{L}^{\text{col}}_{10}, \mathcal{L}^{\text{col}}_{11}\}$	13.7	0.0422	0.576	0.0782	0.906

D Ablation Study

In this section, we present an expanded ablation study. To investigate the effect of the different components of our model, we ablate its performance on a subset of 4 randomly selected scenes from our structured test split of the CO3D cars dataset, as shown in Tab. 2 and Fig. 5. 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 (visually) 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, and exhibit better geometry. We verify that removing the diffuse colour loss harms the geometry, since it helps decouple the symmetric and asymmetric properties facilitating symmetry learning. The ablation of the symmetry losses $\mathcal{L}^{\text{col}}_{jk}$ indicates that each term, including the mixed losses, are important for visual and geometric quality. In particular, removing all symmetry terms has a significantly detrimental effect on performance, as does over-symmetrising the scene ($\lambda = 1$). An accompanying visual ablation study is given in Appendix E.4.

E Additional Qualitative Results

In this section, we present additional high-resolution qualitative results on the CO3D dataset [7]. We also present a visual decomposition of the material and lighting properties that our model learns, as well as a visual ablation of the

Fig. 2. Additional qualitative results on the car category of the CO3D dataset [7]. The last column shows the side of the car unseen during training.

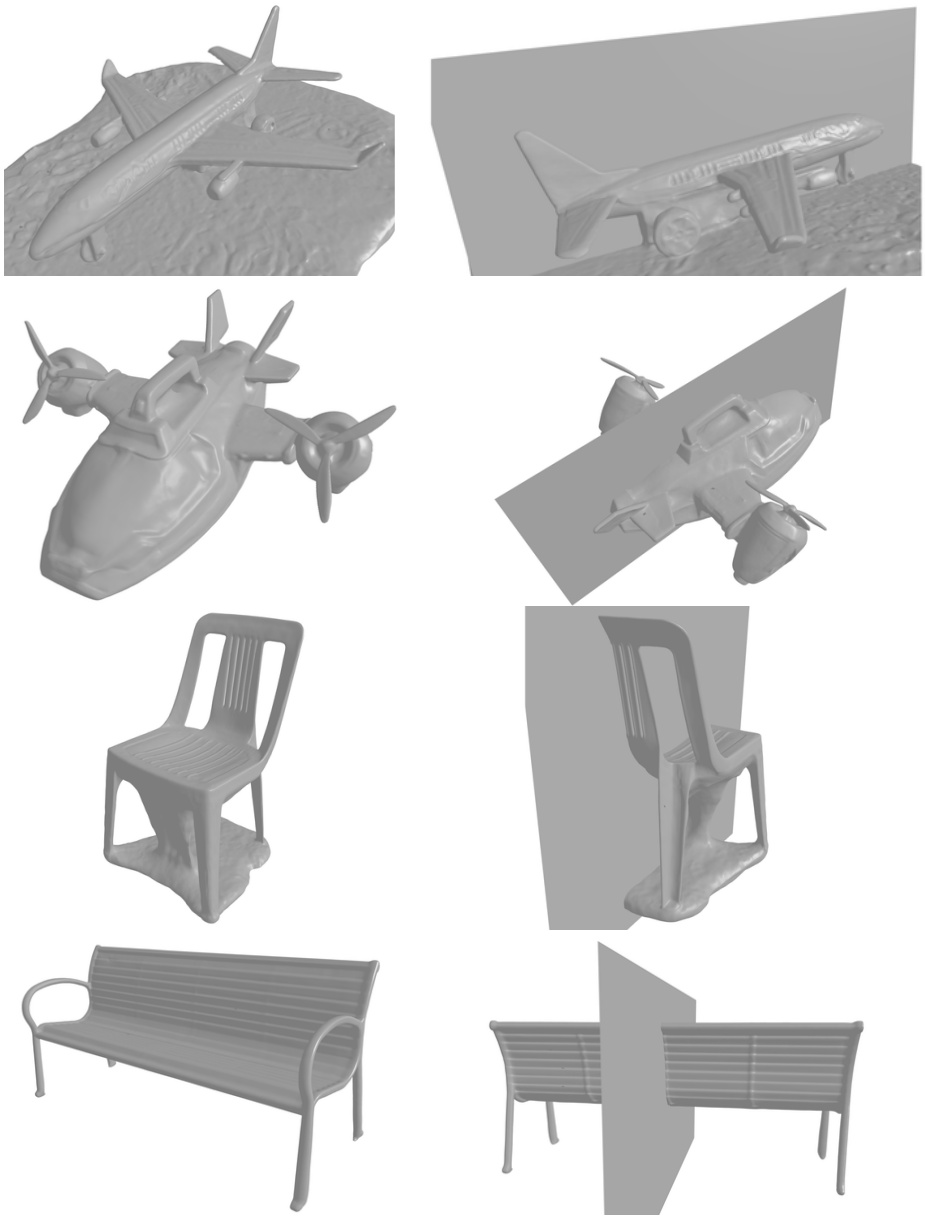


Fig. 3. Reconstructions on additional categories of the CO3D dataset exhibiting reflection symmetry: toyplane (rows 1-2), chair (row 3) and bench (row 4). The second column shows the plane of symmetry.

components of our model. We also include high-resolution videos of our novel view synthesis results alongside this document, for the *structured* (i.e. partial-

view) test split. They clearly show a consistent reconstruction with smoothly varying view-dependent effects and high-frequency specular highlights appearing where appropriate, despite one side of the car being only obliquely seen in the training data.

E.1 Additional Results on the CO3D Car Dataset

In this section, we present additional high-resolution qualitative results on the challenging car category of the CO3D dataset [7], shown in Fig. 2. We are able to recover high-fidelity details across the category, despite using real-world data of highly-reflective and texture-poor objects. The bottom two rows show failure cases where the model was not able to adequately deal with the extreme reflections on the car body.

E.2 Additional Results on Other CO3D Categories

In this section, we demonstrate our model's effectiveness on other near-symmetric categories of the CO3D dataset [7], see Fig. 3. Our method produces high quality reconstructions while accurately estimating the symmetry plane. However, the symmetry parameters do require a reasonable initialisation, as previously discussed, for the local optimisation to converge. Our heuristic (Appendix B.3) for estimating the initial alignment exploits the geometric property of cars that their longest dimension is parallel to the symmetry plane. This heuristic could be extended to exhaustively search for the alignment closest to the assumed symmetry plane. Instead, we manually select, for each category, the coordinate axis that is normal to the symmetry plane. For example, the heuristic-extracted axis of the airplane category is the wing axis, which is perpendicular to the symmetry plane. In future work, we intend to predict the symmetry plane parameters directly from the posed images or point cloud, and use this to initialise our model.

E.3 Appearance Factorisation

In this section, we render images from the factorised components of our colour model, to demonstrate the effectiveness of the decomposition. We show the albedo colour, diffuse shading, reflectivity, and specular colour separately, as well as the diffuse colour (albedo colour $\times$ diffuse shading) and the specular lighting (reflectivity $\times$ specular colour), in Fig. 4. We observe that the colour model is able to disentangle the different components very effectively. In particular, despite significant differences in illumination on the left and right sides of the car, our model is able to recover a plausible and symmetric albedo.

E.4 Qualitative Ablation Study

In this section, we visually ablate the model, showing the unseen side of one of the ablation sequences. This is a particularly difficult sequence, with strongly

Fig. 4. Visualisation of colour decomposition. Note that, despite significant differences in illumination on the left and right sides of the car, our model recovers a plausible and symmetric albedo.

asymmetric lighting and moving shadows cast by the cameraperson. We show the results in Fig. 5. We observe that adding the symmetric lighting loss (third row) improves the visual quality of the renders, at the cost of over-symmetrising the lighting. In comparison, the base model without symmetric lighting (second row) estimates the diffuse lighting reasonably well, but is unable to recover the specular lighting, since these viewing directions are unobserved. A symmetric prior on the specular lighting would be helpful in this case, or a null prior.

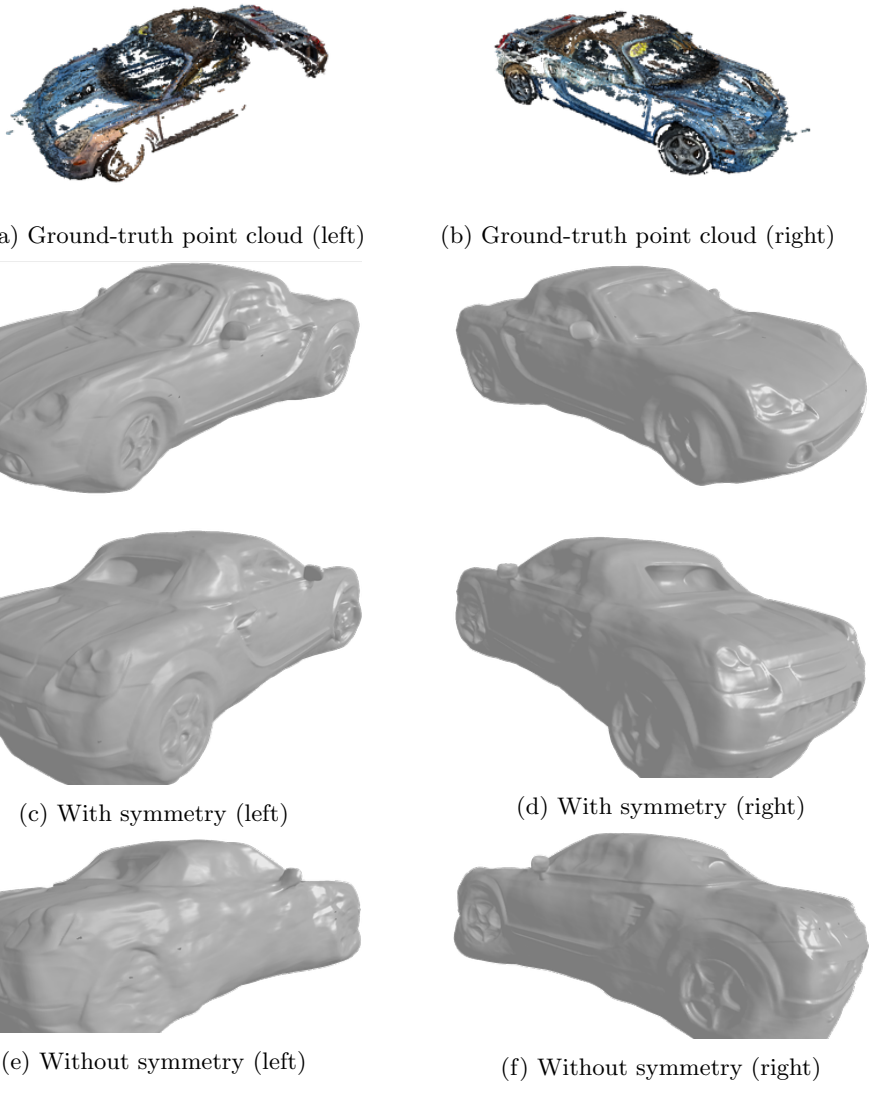
Removing the diffuse lighting loss makes it more difficult for the model to disentangle the material, diffuse lighting and specular lighting components. This is particularly visible at the wheels, where the lighting is (incorrectly) modelled as specular reflections. A higher symmetry factor $\lambda_{\leftrightarrow} = 1$ is quite helpful for this

particular scene, since the geometry and (most of) the appearance is symmetric. With a higher weight on the symmetry terms, more detail is transferred from the visible side. It is also notable that, in the absence of contrary information, the default parameters ($\lambda_\kappa = 0.1$) symmetrise the wheels, which are rotationally offset from the ground truth.

The ablation of the symmetric colour losses are shown in the final three rows. The indices jk of the colour losses are 0 for source and 1 for symmetry-transformed geometry/material (j) and lighting (k). We see that removing any of these components significantly damages the view synthesis quality, especially removing all symmetry terms (last row).

We also show the ground-truth point cloud for this scene in Fig. 6. The ground-truth is very sparse and noisy, with incorrect geometry in numerous places. In contrast, our reconstruction is dense and reconstructs the geometry very well, despite seeing only half of the car. This can also be seen in Fig. 7, where the ground-truth depth map is plotted alongside our predicted depth map. When the symmetry components are removed from our model, the reconstruction is unable to recover any detail on the unseen side of the car. Instead, the smoothness prior of the SDF imposes a closure of the seen part of the car.

Fig. 5. Qualitative ablation study. Novel view renderings of the unseen side of the car.



(a) Ground-truth point cloud (left)

(b) Ground-truth point cloud (right)

(c) With symmetry (left)

(d) With symmetry (right)

(e) Without symmetry (left)

(f) Without symmetry (right)

Fig. 6. Ground-truth point cloud (top), our reconstruction (middle), and reconstruction without symmetry (bottom) of this ablated sequence from the CO3D dataset [7]. The ground-truth is very sparse and noisy, with incorrect geometry in numerous places. In contrast, our reconstruction is dense and detailed, able to reconstruct texture-poor regions despite their high reflectivity. When the symmetry terms are not used, the model fails to reconstruct any detail on the unseen side.

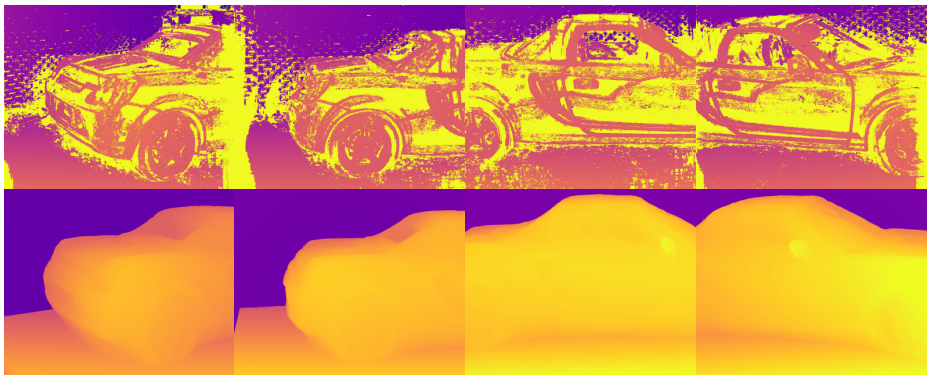


Fig. 7. Comparing ground-truth and predicted depth images from the unseen side of the car. Top: ground-truth sparse depth image. Bottom: our predicted depth image.

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. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: image segmentation as rendering. In: Proc. CVPR. pp. 9799–9808 (2020)
5. 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)
6. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. arXiv:2201.05989 (Jan 2022)
7. 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)
8. Rosen, J.: Symmetry discovered: Concepts and applications in nature and science. Cambridge University Press (1975)
9. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proc. CVPR (2016)
10. Thrun, S., Wegbreit, B.: Shape from symmetry. In: Proc. ICCV. vol. 2, pp. 1824–1831. IEEE (2005)
11. 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)
12. 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)
13. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: NeRF++: analyzing and improving neural radiance fields. arXiv:2010.07492 (2020)