Supplementary

A. Overview

In this document we provide additional experimental results and extended technical details to supplement the main submission. We first discuss the effects on the output of the system made by changes in the loss functions (Sec. B), scene surface characteristics (surface roughness) (Sec. C), and number of material bases (Sec. D). We then showcase our system's ability to model the Fresnel effect (Sec. E), and compare our method against a recent BRDF estimation approach (Sec. F). In Sections G,H, we explain the data capture process and provide additional implementation details. Finally, we describe our supplementary video (Sec. I), show additional novel-view synthesis results along with their intermediate rendering components (Sec. J).

B. Effects of Loss Functions

In this section, we study how the choice of loss functions affects the quality of environment estimation and novel view synthesis. Specifically, we consider three loss functions between prediction and reference images as introduced in the main paper: (i) pixel-wise $L1$ loss, (ii) neural-network based perceptual loss, and (iii) adversarial loss. We run each of our algorithms (environment estimation and novel-view synthesis) for the three following cases: using (i) only, (i+ii) only, and all loss functions combined (i+ii+iii). For both algorithms we provide visual comparisons for each set of loss functions in Figures 1,2.

B.1. Environment Estimation

We run our joint optimization of SRMs and material weights to recover a visualization of the environment using the set of loss functions described above. As shown in Fig. 2, the pixel-wise L1 loss was unable to effectively penalize the view prediction error because it is very sensitive to misalignments due to noisy geometry and camera pose. While the addition of perceptual loss produces better results, one can observe muted specular highlights in the very bright regions. The adversarial loss, in addition to the two other losses, effectively deals with the input errors while simultaneously correctly capturing the light sources.

B.2. Novel-View Synthesis

We similarly train the novel-view neural rendering network in Sec. 6 using the aforementioned loss functions. Results in Fig. 1 shows that while L1 loss fails to capture specularity when significant image misalignments exist, the addition of perceptual loss somewhat addresses the issue. As expected, using adversarial loss, along with all other losses, allows the neural network to fully capture the intensity of specular highlights.

(a) GT (b) L1 Loss (c) L1+Percept (d) All Losses

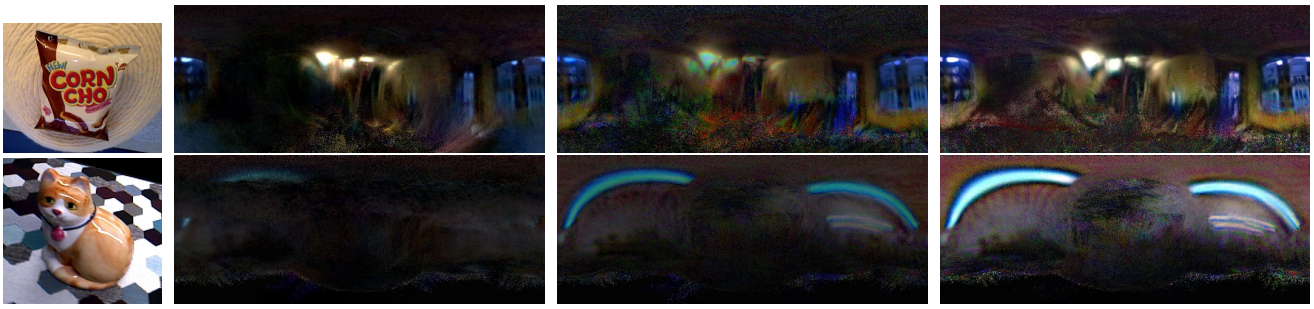
Figure 1: Effects of loss functions on neural-rendering. The specular highlights on the forehead of the Labcat is expressed weaker than it actually is when using L1 or perceptual loss, likely due to geometric and calibration errors. The highlight is best expressed when the neural rendering pipeline of Sec. 6 is trained with the combination of L1, perceptual, and adversarial loss.

C. Effects of Surface Roughness

As descrbied in the main paper, our recovered specular reflectance map is environment lighting convolved with the surface's specular BRDF. Thus, the quality of the estimated SRM should depend on the roughness of the surface, e.g. a near Lambertian surface would not provide significant information about its surroundings. To test this claim, we run the SRM estimation algorithm on a synthetic object with varying levels of specular roughness. Specifically, we vary the roughness parameter of the GGX shading model [11] from 0.01 to 1.0, where smaller values correspond to more mirror-like surfaces. We render images of the synthetic object, and provide those rendered images, as well as the geometry (with added noise in both scale and vertex displacements, to simulate a real scanning scenario), to our algorithm. The results show that the accuracy of environment estimation decreases as the object surface gets more rough, as expected (Fig. 6). Note that although increasing amounts of surface roughness does cause the amount of detail in our estimated environments to decrease, this is expected, as the recovered SRM still faithfully reproduces the convolved lighting (Fig. 5).

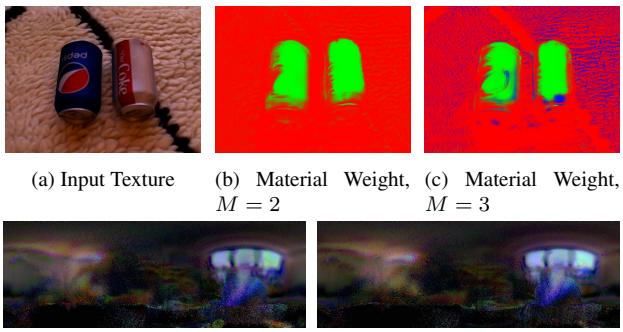
D. Effects of Number of Material Bases

The joint SRM and segmentation optimization of the main paper requires a user to set the number of material bases. In this section, we study how the algorithm is affected by the user specified number. Specifically, for a scene containing two cans, we run our algorithm twice, with number of material bases set to be two and three, respectively. The results of the experiment in Figure 3 suggest that the number of material bases does not have a significant effect on the output of our system.



(a) Scene (b) L1 Loss (c) L1+Perceptual Loss (d) L1+Perceptual+GAN Loss

Figure 2: Environment estimation using different loss functions. From input video sequences (a), we run our SRM estimation algorithm, varying the final loss function between the view predictions and input images. Because L1 loss (b) is very sensitive to misalignments caused by geometric and calibration errors, it averages out the observed specular highlights, resulting in missing detail for large portions of the environment. While the addition of perceptual loss (c) mitigates this problem, the resulting SRMs often lose the brightness or details of the specular highlights. The adoption of GAN loss produces improved results (d).



(a) Input Texture (b) Material Weight, $M = 2$ (c) Material Weight, $M = 3$

(d) Recovered SRM, $M = 2$ (e) Recovered SRM, $M = 3$

Figure 3: Sensitivity to the number of material bases M. We run our SRM estimation and material segmentation pipeline twice on a same scene but with different number of material bases M, showing that our system is robust to the choice of M. We show the predicted combination weights of the network trained with two (b) and three (c) material bases. For both cases (b,c), SRMs that correspond to the red and blue channel are mostly black, i.e. diffuse BRDF. Note that our algorithm consistently assigns the specular material (green channel) to the same regions of the image (cans), and that the recovered SRMs corresponding to the green channel (d,e) are almost identical.

E. Fresnel Effect Example

The Fresnel effect is a phenomenon where specular highlights tend to be stronger at near-glancing view angles, and is an important visual effect in the graphics community. We show in Fig. 4 that our neural rendering system correctly models the Fresnel effect. In the supplementary video, we show the Fresnel effect in motion, along with comparisons to the ground truth sequences.

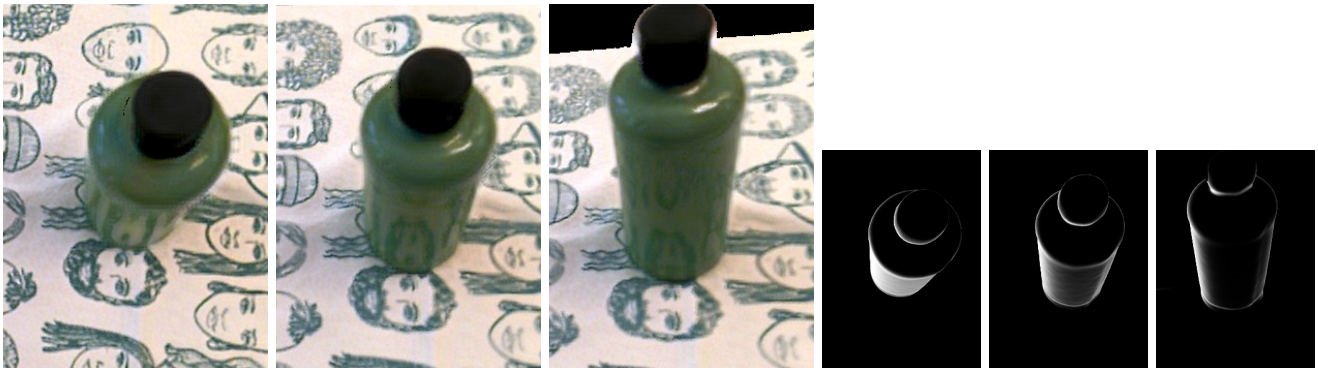
F. Comparison to BRDF Fitting

Recovering a parametric analytical BRDF is a popular strategy to model view-dependent effects. We thus compare our neural network-based novel-view synthesis approach against a recent BRDF fitting method of [8] that uses an IR laser and camera to optimize for the surface specular BRDF parameters. As shown in Fig. 7, sharp specular BRDF fitting methods are prone to failure when there are calibration errors or misalignments in geometry.

G. Data Capture Details

As described in Sec. 7 of the main paper, we capture ten videos of objects with varying materials, lighting and compositions. We used a Primesense Carmine RGBD structured light camera. We perform intrinsic and radiometric calibrations, and correct the images for vignetting. During capture, the color and depth streams were hardware-synchronized, and registered to the color camera frame-of-reference. The resolution of both streams are VGA (640x480) and the frame rate was set to 30fps. Camera exposure was manually set and fixed within a scene.

We obtained camera extrinsics by running ORB-SLAM [6] (ICP [7] was alternatively used for feature-poor scenes). Using the estimated pose, we ran volumetric fusion [7] to obtain the geometry reconstruction. Once geometry and rough camera poses are estimated, we ran frame-to-model dense photometric alignment following [8] for more accurate camera positions, which are subsequently used to fuse in the diffuse texture to the geometry. Following [8], we use iteratively reweighted least squares to compute a robust minimum of intensity for each surface point across viewpoints, which provides a good approximation to the diffuse texture.



(a) View 1 (b) View 2 (c) View 3 (d) View 1 (e) View 2 (f) View 3

Figure 4: Demonstration of the Fresnel effect. The intensity of specular highlights tends to be amplified at slant viewing angles. We show three different views (a,b,c) for a glossy bottle, each of them generated by our neural rendering pipeline and presenting different viewing angles with respect to the bottle. Notice that the neural rendering correctly amplifies the specular highlights as the viewing angle gets closer to perpendicular with the surface normal. Images (d,e,f) show the computed Fresnel coefficient (FCI) (see Sec. 6.1) for the corresponding views. These images are given as input to the neural-renderer that subsequently use them to simulate the Fresnel effect. Best viewed digitally.

H. Implementation Details

Our pipeline is built using PyTorch [9]. For all of our experiments we used ADAM optimizer with learning rate 2e-4 for the neural networks and 1e-3 for the SRM pixels. For the SRM optimization described in Sec. 5 of the main text the training was run for 40 epochs (i.e. each training frame is processed 40 times), while the neural renderer training was run for 75 epochs.

We find that data augmentation plays a significant role to the view generalization of our algorithm. For training in Sec. 5, we used random rotation (up to 180°), translation (up to 100 pixels), and horizontal and vertical flips. For neural renderer training in Sec. 6, we additionally scale the input images by a random factor between 0.8 and 1.25.

We use Blender [1] for computing the reflection direction image R_P and the first bounce interreflection (FBI) image described in the main text.

H.1. Network Architectures

Let `C(k,ch_in,ch_out,s)` be a convolution layer with kernel size `k`, input channel size `ch_in`, output channel size `ch_out`, and stride `s`. When the stride `s` is smaller than 1, we first conduct nearest-pixel upsampling on the input feature and then process it with a regular convolution layer. We denote `CNR` and `CR` to be the Convolution-InstanceNorm-ReLU layer and Convolution-ReLU layer, respectively. A residual block `R(ch)` of channel size `ch` contains convolutional layers of `CNR(3,ch,ch,1)-CN(3,ch,ch,1)`, where the final output is the sum of the outputs of the first and the second layer.

Encoder-Decoder Network Architecture The architecture of the texture refinement network and the neural rendering network in Sec.5 and Sec.6 closely follow the architecture of an encoder-decoder network of Johnson *et al.* [5]: `CNR(9,ch_in,32,1)-CNR(3,32,64,2)-CNR(3,64,128,2)-R(128)-R(128)-R(128)-R(128)-R(128)-CNR(3,128,64,1/2)-CNR(3,64,32,1/2)-C(3,32,3,1)`, where `c_in` represents a variable input channel size, which is 3 and 13 for the texture refinement network and neural rendering generator, respectively.

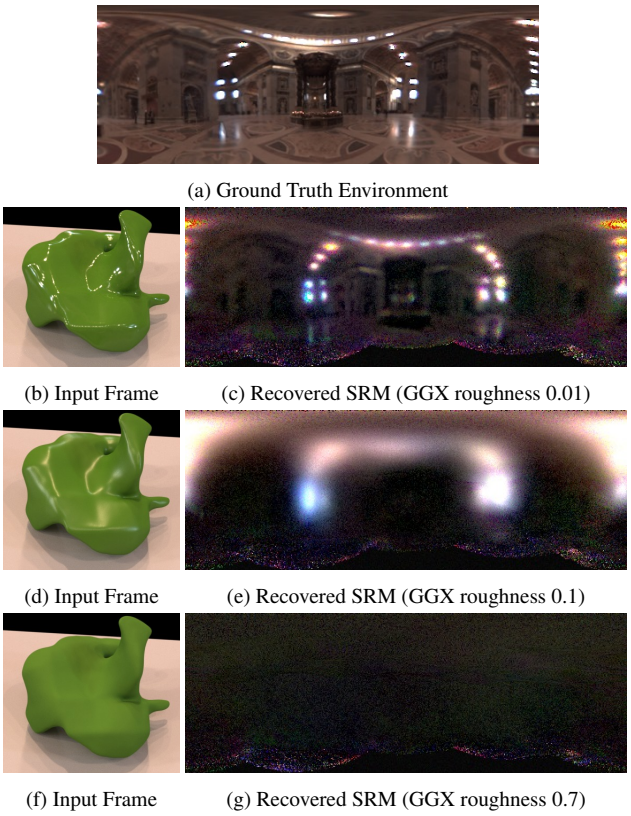
Material Weight Network The architecture of the material weight estimation network in Sec. 5 is as follows: `CNR(5,3,64,2)-CNR(3,64,64,2)-R(64)-R(64)-CNR(3,64,32,1/2)-C(3,32,3,1/2)`.

Discriminator Architecture The discriminator network used for the adversarial loss in Eq.7 and Eq.8 of the main paper both use the same architecture as follows: `CR(4,3,64,2)-CNR(4,64,128,2)-CNR(4,128,256,2)-CNR(4,256,512,2)-C(1,512,1,1)`. For this network, we use a LeakyReLU activation (slope 0.2) instead of the regular ReLU, so CNR used here is a Convolution-InstanceNorn-LeakyReLU layer. Note that the spatial dimension of the discriminator output is larger than 1x1 for our image dimensions (640x480), i.e., the discriminator scores realism of patches rather than the whole image (as in PatchGAN [3]).

I. Supplementary Video

We strongly encourage readers to watch the supplementary video†, as many of our results we present are best seen as videos. Our supplementary video contains visualizations

†Video URL: https://youtu.be/9t_Rx6n1HGA



(a) Ground Truth Environment

(b) Input Frame (c) Recovered SRM (GGX roughness 0.01)

(d) Input Frame (e) Recovered SRM (GGX roughness 0.1)

(f) Input Frame (g) Recovered SRM (GGX roughness 0.7)

Figure 5: Recovering SRM for different surface roughness. We test the quality of estimated SRMs (c,e,g) for various surface materials (shown in (b,d,f)). The results closely match our expectation that environment estimation through specularity is challenging for glossy (d) and diffuse (f) surfaces, compared to the mirror-like surfaces (c). Note that the input to our system are rendering images and noisy geometry, from which our system reliably estimates the environment.

of input videos, environment estimations, our neural novel-view synthesis (NVS) renderings, and side-by-side comparisons against the state-of-the-art NVS methods. We note that the ground truth videos of the NVS section are cropped such that regions with missing geometry are displayed as black. The purpose of the crop is to provide equal visual comparisons between the ground truth and the rendering, so that viewers are able to focus on the realism of reconstructed scene instead of the background. Since the reconstructed geometry is not always perfectly aligned with the input videos, some boundaries of the ground truth stream may contain noticeable artifacts, such as edge-fattening. An example of this can be seen in the 'acryl' sequence, near the top of the object.

J. Additional Results

Table 1 shows numerical comparisons on novel-view synthesis against state-of-the-art methods [2, 10] for the

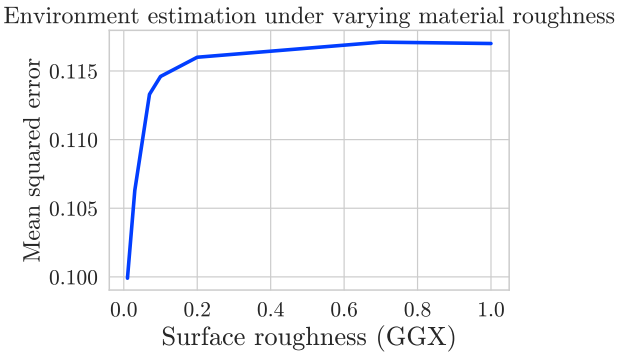


Figure 6: Accuracy of environment estimation under dif ferent amounts of surface roughness. We see that increas ing the material roughness does indeed decrease the over all quality of the reconstructed environment image measured in pixel-wise L2 distance. Note that the roughness parameter is from the GGX [11] shading model which we use to render the synthetic models.

	Cans-L1	Labcat-L1	Cans-perc	Labcat-perc
[2]	9.82e-3	6.87e-3	0.186	0.137
[10]	9.88e-3	8.04e-3	0.163	0.178
Ours	**4.51e-3**	**5.71e-3**	**0.103**	**0.098**

Table 1: Average pixel-wise L1 error and perceptual error values (lower is better) across the different view synthesis methods on the two datasets (Cans, Labcat). The L1 metric is computed as mean L1 distance across pixels and channels between novel-view prediction and ground-truth images. The perceptual error numbers correspond to the mean values of the measurements shown in Figure 7 of the main paper. As described in the main paper, we mask out the background (e.g. carpet) and focus only on the specular object surfaces.

two scenes presented in the main text (Fig. 7). We adopt two commonly used metrics, i.e. pixel-wise L1 and deep perceptual loss [5], to measure the distance between a predicted novel-view image and its corresponding ground-truth test image held-out during training. As described in the main text we focus on the systems' ability to extrapolate specular highlight, thus we only measure the errors on the object surfaces, i.e. we remove diffuse backgrounds.

Fig. 8 shows that the naïve addition of diffuse and specular components obtained from the optimization in Sec. 5 does not results in photorealistic novel view synthesis, thus motivating a separate neural rendering step that takes as input the intermediate physically-based rendering components.

Fig. 9 shows novel-view neural rendering results, together with the estimated components (diffuse and specular images D_P, S_P) provided as input to the renderer. Our approach can synthesize photorealistic novel views of a scene with wide range of materials, object compositions, and lighting condition. Note that the featured scenes con-

(a) Reference (b) Our Reconstruction (c) Reconstruction by [8]

Figure 7: Comparison with Surface Light Field Fusion [8]. Note that the sharp specular highlight on the bottom-left of the Corncho bag is poorly reconstructed in the rendering of [8] (c). As shown in Sec. B and Fig. 9, these high frequency appearance details are only captured when using neural rendering and robust loss functions (b).

(a) Ground Truth (b) Rendering with SRM

Figure 8: Motivation for neural rendering. While the SRM and segmentation obtained from the optimization of Sec. 5 of the main text provides high quality environment reconstruction, the simple addition of the diffuse and specular component does not yield photorealistic rendering (b) compared to the ground truth (a). This motivates the neural rendering network that takes input as the intermediate rendering components and generate photorealistic images (e.g. shown in Fig. 9).

tain challenging properties such as bumpy surfaces (Fruits), rough reflecting surfaces (Macbook), and concave surfaces (Bowls). Overall, we demonstrate the robustness of our approach for various materials including fabric, metals, plastic, ceramic, fruit, wood, glass, etc.

On a separate note, reconstructing SRMs of planar surfaces could require more views to fully cover the environment hemisphere, because the surface normal variation of each view is very limited for a planar surface. We refer readers to Janick *et al*. [4] that studies capturing planar surface light field, which reports that it takes about a minute using their real-time, guided capture system.

References

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

[2] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. In *SIGGRAPH Asia 2018 Technical Papers*, page 257. ACM, 2018. 4

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 3

[4] Jan Jachnik, Richard A Newcombe, and Andrew J Davison. Real-time surface light-field capture for augmentation of planar specular surfaces. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 91–97. IEEE, 2012. 5

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 3, 4

[6] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 2

[7] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, volume 11, pages 127–136, 2011. 2

[8] Jeong Joon Park, Richard Newcombe, and Steve Seitz. Surface light field fusion. In *2018 International Conference on 3D Vision (3DV)*, pages 12–21. IEEE, 2018. 2, 5

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 3

[10] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *arXiv preprint arXiv:1904.12356*, 2019. 4

[11] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206. Eurographics Association, 2007. 1, 4

(a) Ground Truth G_P (b) Our Rendering $g(C_P)$ (c) Specular Component S_P (d) Diffuse Component D_P

Figure 9: Novel view renderings and intermediate rendering components for various scenes. From left to right: (a) reference photograph, (b) our rendering, (c) specular reflectance map image, and (d) diffuse texture image. Note that some of the ground truth reference images have black "background" pixels inserted near the top and left borders where reconstructed geometry is missing, to provide equal visual comparisons to rendered images.