Supplementary Document for Two-shot Spatially-varying BRDF and Shape Estimation

Mark Boss[1], Varun Jampani[2], Kihwan Kim[2], Hendrik P.A. Lensch[1], Jan Kautz[2]

[1]University of Tübingen, [2]NVIDIA

1. Overview

In this document, we provide additional details, discussions, and experiments to support the original submission.

2. Scale-Shift Invariant Metric

To allow for a fair comparisons we employ a scale and shift invariant loss for methods which either produce relative depth or depth in a different scale. A specialized scale invariant loss formulation for a comparison with intrinsic imaging based papers, as the predicted diffuse color is not subject to an absolute scale as diffuse albedo parameter is. To achieve the scale and shift in-variance we define it as

$$\mathcal{L}(x, x^{\text{gt}}) = \arg\min_{\alpha,\beta} \frac{1}{2D} \sum_{i=1}^{D} (\alpha x_i + \beta - x_i^{\text{gt}})^2 \quad (1)$$

where α accounts for the scale and β for the shift, D for the image dimension, x for the predicted result and x^{gt} for the corresponding ground truth. For the scale invariant loss only the α is optimized.

3. Detailed Network Architectures

The proposed cascaded network architecture uses four distinct network architectures. In the following we will denote a regular 2D convolution with a kernel size of 4, a stride of 2, InstanceNorm, ReLU activation and `k` filters as `c-k`. A transposed convolution is called `ct-k` with the same kernel size, stride, and activations.

Shape Estimation with Merge Convolutions: The input of the shape estimation network is the two-shot input images and the segmentation mask. We use MergeConv blocks in an encoder-decoder architecture. Refer to the paper for details about a MergeConv block. We use four MergeConv blocks for encoding and also for decoding in U-net inspired shape [6]. The initial input each of the pathways is one of the two-shot input images channel stacked with the segmentation mask.

To denote the network architecture, we use the following naming scheme. A MergeConv block with a kernel size of 4, a stride of 2, InstanceNorm, and ReLU activation is denoted as `mo-k`. Here, `k` defines the number of output filters for the merged and input pathways. Upsampling or downsampling is denoted in `o`, where `d` is used for the downsampling operation and `u` for the upsampling. A regular convolution with a kernel size of 5 and a stride of 1 is denoted as `c-k`. The `k` parameter also defines the number of output features, and a sigmoid activation function is used. The network architecture is described as:

```
md-32, md-64, md-128, md-256, mu-256,
mu-128, mu-64, mu-32, c-4
```

Shape Guided Illumination Estimation: The input for this network architecture now consists of the two-shot input images, the segmentation mask, and the previous shape estimation (Normals and Depth). Here, we do not employ the merge convolutions, and all inputs are channel stacked. As the network output is 24 RGB values, we only employ an encoder, followed by fully connected blocks. An additional convolution operation is denoted as `cn-k`, with a kernel size of 3, a stride of 2, and a ReLU activation. Lastly, a fully connected layer is referred to as `f-k`. The architecture is then denoted as:

```
c-16, c-32, c-64, c-128, c-256,
c-256, cn-256, cn-512, f-256, ReLU,
f-72, Sigmoid
```

The last fully-connected consists of 72 outputs, which corresponds to 24 RGB values for the spherical Gaussian amplitudes.

Guided SVBRDF Estimation: For BRDF prediction, we stack the channels of the previous predictions and the two-shot input images. The illumination prediction is here appended to each pixel of the input images. An additional output convolution is referred to as `co-k` with a kernel size of 5, a stride of 1, and sigmoid activation. The network architecture is defined as:

```
c-32, c-64, c-96, c-128, c-160,
c-192, ct-192, ct-160, ct-128, ct-96,
ct-64, ct-32, co-7
```

Joint Shape and SVBRDF Refinement: Similar to the

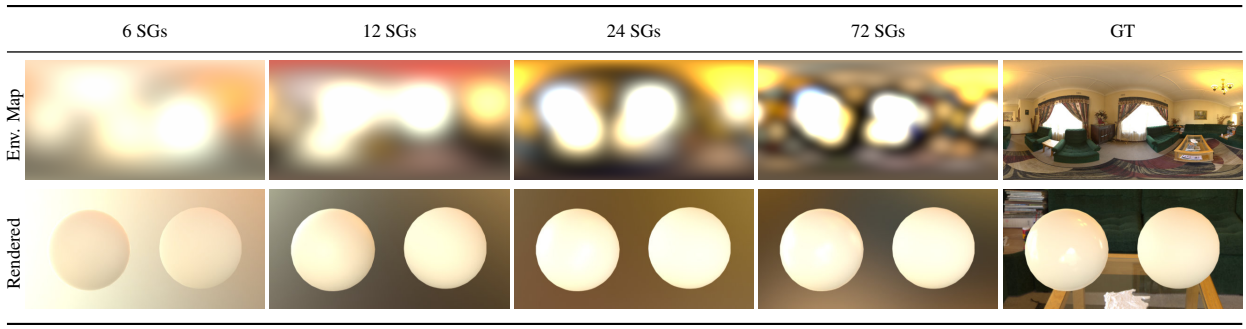

Figure 1: **Spherical Gaussian Environment Illumination.** Visualization of an environment with varying number of spherical Gaussians. The top row shows the evaluated spherical Gaussians, and the bottom rows show two spheres rendered with the spherical Gaussian approximation. Here, the left sphere is a glossy and the right a rough material.

BRDF estimation, we stack each of the previous predictions in the channel dimensions. We also add the residual loss image between the input flash image and the re-rendered initial predictions. A resnet block here consists of two pre-activated 2D convolutions with a kernel size of 3, a stride of 1, InstanceNorm, and ReLU activation. The shortcut connection is added from the input to the final block output. The block is denoted as `r-k`. A final output convolution is denoted as `c0-k` with a kernel size of 5, a stride of 1, and a sigmoid activation. The overall network is described as:

```
c-64, c-128, c-256, r-256, r-256,
r-256, r-256, ct-256, ct-128, ct-64,
co-11
```

The final output consists of 11 channels corresponding to diffuse (3), specular (3), roughness (1), depth (1), and normal (3).

4. Mobile Application

The mobile android application is written in kotlin and handles the capturing of objects, the segmentation, and prediction. The capturing automatically takes the two-shot input pair. The segmentation is done using OpenCV's GrabCut implementation on the device. For the prediction, we converted the trained models to TensorFlow lite. Here, we do not use quantization as the results degraded too harshly. An aware quantization training could remedy this effect. A quantized output and model increases the prediction speed even further, but it is already reasonably fast. On recent mobile phones (Pixel 4, Pixel 2, OnePlus 6t), the full inference takes about 6 seconds.

5. Rendering Setup

Rendering the domain randomized shape in a realistic setup for real-world usage is a crucial aspect of a successful domain transfer. To achieve this, our rendering setup closely follows real-world scenarios. The camera is positioned randomly on a sphere with a radius of 70cm from the origin. The objects are constructed at the origin, and due to the random translation, rotation and scaling can grow up to 17cm distance from the camera. This is a reasonable distance between an object and a mobile phone for real-world capture. To always have the object in focus, the camera view is rotated towards the origin.

The flash-light is approximated as a point light source and positioned in a 2cm radius around the camera with a flash strength of 45 Lumen, which are typical settings of smartphone cameras and flashes.

For the flash image, we separately rendered two HDR images with only flash and only environment illumination and linearly combine these two HDR renderings to obtain the final flash image. This strategy allows us to randomly vary the flash strength by using randomly sampled weights for a linear combination of the ambient and flash rendering. As the network receives LDR input images, we perform a Saturation Based Sensitivity auto exposure calculation [2].

In Fig. 1, the effect of a varying number of spherical Gaussians (SG) is shown. As seen, the detail and sharpness of the environment and object illumination increases with a growing number of SGs. As a compromise of estimating too many parameters, our method predicts 24 SGs.

Our SGs are parametrized by the direction μ, the sharpness λ, and the amplitude α of the lobe. The lobe is then defined as:

$$G(\mathbf{v}; \mu, \lambda, \alpha) = \alpha e^{\lambda \cdot \mathbf{v} - 1} \tag{2}$$

where $\mathbf{v}$ now represents the evaluation direction. Further details about the evaluation and the BRDF SG fitting are based on [7].

6. Results

We analyze the prediction quality improvements in synthetic examples. Further real-world and synthetic results are available as a website in the supplementary.

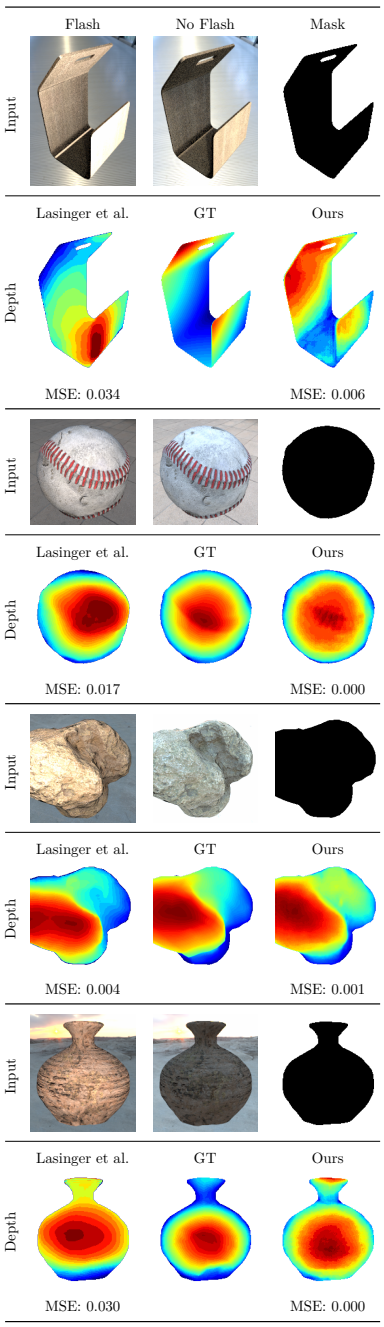

Figure 2: **Comparison with Lasinger *et al*.** Even on challenging shapes as the first example, our method provides a more accurate prediction.

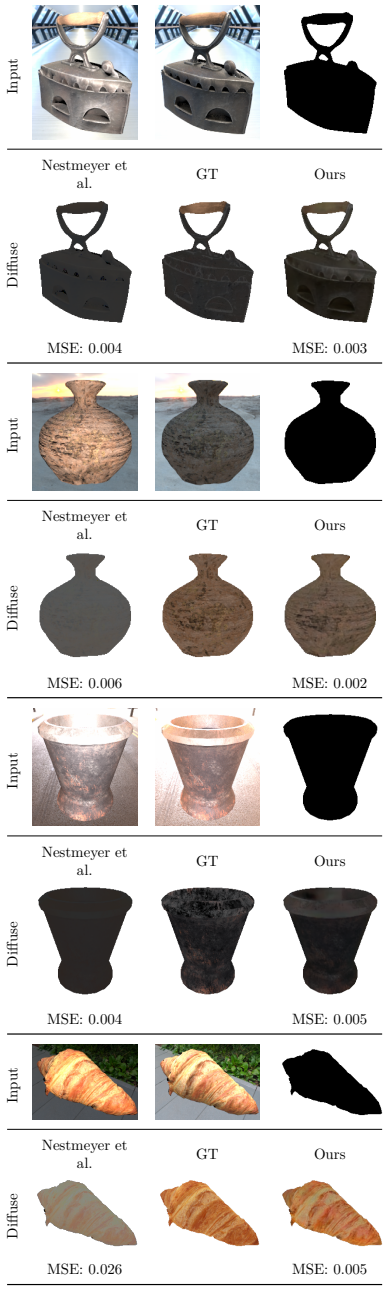

Figure 3: **Comparison with Nestmeyer *et al*.** In many cases our predictions is on par or surpases Nestmeyer *et al*. with a more complex BRDF model to disentangle.

6.1. Visual Comparison with Lasinger et al.

Larsinger *et al*. [3] predicts the relative depth from a single input image. We use the scale and shift-invariant metric to compare our work with theirs. In Fig. 2 several prediction examples are shown.

The first example is a sheet of wood formed into a complex object. Lasinger *et al*. and our method struggle with this object. However, overall, our method follows the shape of the object much closer and plausible than Lasinger *et al*. In the second example, our method predicts the shape of the baseball also closer. Here, the closest point is correctly predicted in the center of the ball. In the last example, the top of the pot is also predicted slightly better. Lasinger *et al*. nearly predicts the top part to be nearly flat while ours

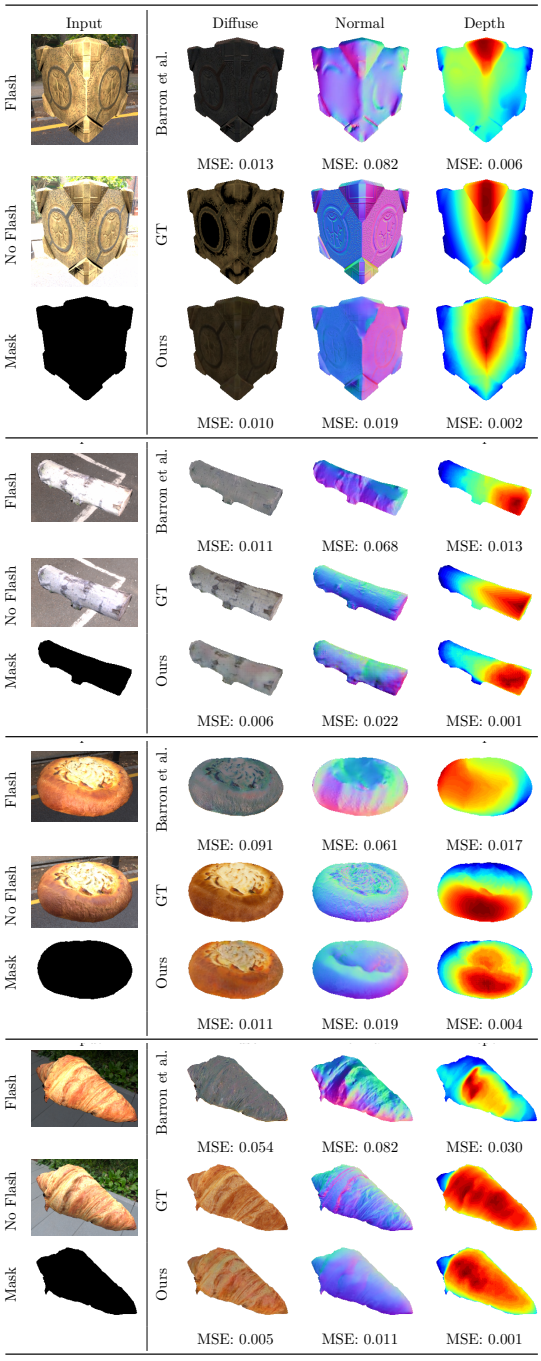

Figure 4: **Comparison with Barron *et al*.** Notice the overall improved shape and diffuse accuracy in the prediction.

follows the curved, extending top.

6.2. Visual Comparison with Nestmeyer et al.

Additional comparisons with Nestmeyer *et al*. [5] are shown in Fig. 3. Here, we want to highlight that our method tackles a complex BRDF model, which is more difficult to disentangle than the intrinsic imaging model of Nestmeyer *et al*. Due to this, we only compare, similar to Barron *et al*., with the scale-invariant diffuse color. As seen, our method can reconstruct the diffuse color either with equal quality or surpass the prediction quality of Nestmeyer *et al*. Especially texture details are preserved better in our method.

6.3. Additional Comparison with Barron et al.

As Barron *et al*. [1] predict a relative depth and the reflectance, which is a non-absolute diffuse color, we employ a scale-shift invariant loss for the depth map and a scale-invariant for the diffuse color. In Fig. 4 we compare our predictions with theirs.

Our method, in general, provides advances in the shape prediction with improved depth and normal parameters. In general, the features are captured more accurate and plausible. The material color is also separated better from the shading. This is especially visible in the last two examples where Barron *et al*. predicted most of the color in the shading of the objects.

6.4. Additional Comparison with Li et al.

We also provide more comparisons with Li *et al*. [4]. The results are shown in Fig. 5. In the depth map, the improved performance of our method is apparent. In the first example, Li *et al*. predict the barrel to be concave instead of convex. The bottom is also predicted as the closest point to the camera. Our method captures the general shape of the object quite well but also struggles with the top of the barrel slightly. Here, the visible top rim in the back is not predicted accurately as well as the top plate of the barrel. In the second example, Li *et al*. predict the baseball to be concave again. Our method predicts the spherical nature of the ball accurately. In the last example, the method of Li *et al*. again fails to predict the shape of the box correctly. Here, the corner closest to the user is predicted far away, and the sides in the bottom are predicted closer. In general, our method also produces smoother and more accurate normal maps, but are not as detailed as the normals from Li *et al*. This can be attributed to our cascaded network design, which still can not leverage correlations between maps as well as the joint prediction model. Our method also reduces the visible shading in the diffuse parameters, and the highlights from the flash input and environment illumination are less visible.

References

[1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015. 4

[2] Photography Digital still cameras Determination of exposure index, ISO speed ratings, standard output sensitivity, and recommended exposure index. Standard, 2019. 2

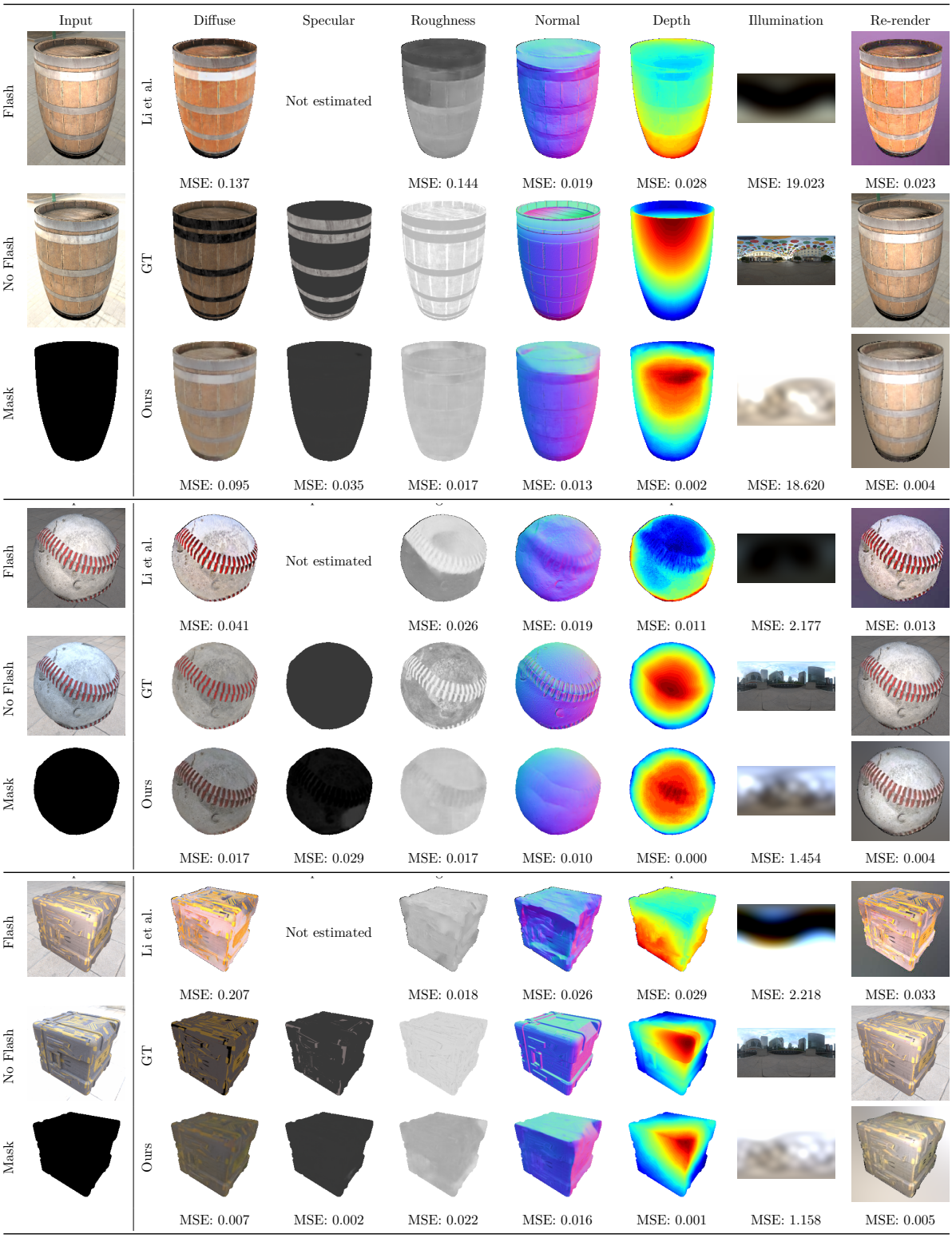

Figure 5: **Comparison with Li *et al*.** Notice the overall improved accuracy in the prediction.

[3] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *ArXiv e-prints*, 2019. 3

[4] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chan-

draker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018. 4

[5] T. Nestmeyer and P. V. Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[6] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015. 1

[7] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2009. 2