# Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image

Zhengqin Li*        Kalyan Sunkavalli†        Manmohan Chandraker*

*University of California, San Diego        †Adobe Research, San Jose

**Abstract.** We propose a material acquisition approach to recover the spatially-varying BRDF and normal map of a near-planar surface from a single image captured by a handheld mobile phone camera. Our method images the surface under arbitrary environment lighting with the flash turned on, thereby avoiding shadows while simultaneously capturing high-frequency specular highlights. We train a CNN to regress an SVBRDF and surface normals from this image. Our network is trained using a large-scale SVBRDF dataset and designed to incorporate physical insights for material estimation, including an in-network rendering layer to model appearance and a material classifier to provide additional supervision during training. We refine the results from the network using a dense CRF module whose terms are designed specifically for our task. The framework is trained end-to-end and produces high quality results for a variety of materials. We provide extensive ablation studies to evaluate our network on both synthetic and real data, while demonstrating significant improvements in comparisons with prior works.

## 1   Introduction

The wide variety of images around us are the outcome of interactions between lighting, shapes and materials. In recent years, the advent of convolutional neural networks (CNNs) has led to significant advances in recovering shape using just a single image [1,2]. In contrast, material estimation has not seen as much progress, which might be attributed to multiple causes. First, material properties can be more complex. Even discounting more complex global illumination effects, materials are represented by a spatially-varying bidirectional reflectance distribution function (SVBRDF), which is an unknown high-dimensional function that depends on exitant and incident lighting directions [3]. Second, while large-scale synthetic and real datasets have been collected for shape estimation [4,5], there is a lack of similar data for material estimation. Third, pixel observations in a single image contain entangled information from factors such as shape and lighting, besides material, which makes estimation ill-posed.

In this work, we present a practical material capture method that can recover an SVBRDF from a *single* image of a near-planar surface, acquired using the camera of an off-the-shelf consumer mobile phone, under unconstrained environment illumination. This is in contrast to conventional BRDF capture setups that usually require significant equipment and expense [6,7]. We address this challenge
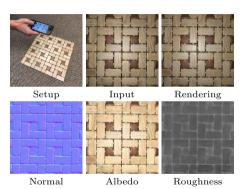
Setup      Input      Rendering

Normal     Albedo     Roughness

**Fig. 1.** We propose a deep learning-based light-weight SVBRDF acquisition system. From a single image of a near planar surface captured with a flash-enabled mobile phone camera under arbitrary lighting, our network recovers surface normals and spatially-varying BRDF parameters – diffuse albedo and specular roughness. Rendering the estimated parameters produces an image almost identical to the input image.

by proposing a novel CNN architecture that is specifically designed to account for the physical form of BRDFs and the interaction of light with materials, which leads to a better learning objective. We also propose to use a novel dataset of SVBRDFs that has been designed for perceptual accuracy of materials. This is in contrast to prior datasets that are limited to homogeneous materials, or juxtapose material properties with other concepts such as object categories.

We introduce a novel CNN architecture that encodes the input image into a latent representation, which is decoded into components corresponding to surface normals, diffuse texture, and specular roughness. We propose a differentiable rendering layer that recombines the estimated components with a novel lighting direction. This gives us additional supervision from images of the material rendered under arbitrary lighting directions during training; only a single image is used at test time. We also observe that coarse classification of BRDFs into material meta-categories is an easier task, so we additionally include a material classifier to constrain the latent representation. The inferred BRDF parameters from the CNN are quite accurate, but we achieve further improvement using densely-connected conditional random fields (DCRFs) with novel unary and smoothness terms that reflect the properties of the underlying microfacet BRDF model. We train the entire framework in an end-to-end manner.

Our approach — using our novel architecture and SVBRDF dataset — can outperform the state-of-art. We demonstrate that we can further improve these results by leveraging a form of acquisition control that is present on virtually every mobile phone — the camera flash. We turn on the flash of the mobile phone camera during acquisition; our images are thus captured under a combination of unknown environment illumination and the flash. The flash illumination helps further improve our reconstructions. First, it minimizes shadows caused by occlusions. Second, it allows better observation of high-frequency specular highlights, which allows better characterization of material type and more accurate estimation. Third, it provides a relatively simple setup for acquisition that eases the burden on estimation and allows the use of better post-processing techniques.

In contrast to recent works such as [8] and [9] that can reconstruct BRDFs with stochastic textures, we can handle a much larger class of materials. Also, our results, both with and without flash, are a significant improvement over the recent method of Li et al. [10] even though our trained model is more compact.

Our experiments demonstrate advantages over several baselines and prior works in quantitative comparisons, while also achieving superior qualitative results. In particular, the generalization ability of our network trained on the synthetic BRDF dataset is demonstrated by strong performance on real images, acquired in the wild, in both indoor and outdoor environments, using multiple different phone cameras. Given the estimated BRDF parameters, we also demonstrate applications such as material editing and relighting of novel shapes.

To summarize, we propose the following contributions:

– A novel lightweight SVBRDF acquisition method that produces state-of-the-art reconstruction quality.
– A CNN architecture that exploits domain knowledge for joint SVBRDF reconstruction and material classification.
– Novel DCRF-based post-processing that accounts for the microfacet BRDF model to refine network outputs.
– An SVBRDF dataset that is large-scale and specifically attuned to estimation of spatially-varying materials.

## 2    Related Work

**BRDF Acquisition:** The Bidirectional Reflection Distribution function (BRDF) is a 4-D function that characterizes how a surface reflects lighting from an incident direction toward an outgoing direction [3]. Alternatively, BRDFs are represented using low-dimensional parametric models [11,12,13,14]. In this work, we use a physically-based microfacet model [15] that our SVBRDF dataset uses.

Traditional methods for BRDF acquisition rely on densely sampling this 4-D space using expensive, calibrated acquisition systems [6,7,16]. Recent work has demonstrated that assuming BRDFs lie in a low-dimensional subspace allows for them to be reconstructed from a small set of measurements [17,18]. However, these measurements still to be taken under controlled settings. We assume a single image captured under largely uncontrolled settings.

Photometric stereo-based methods recover shape and/or BRDFs from images. Some of these methods recover a homogeneous BRDF given one or both of the shape and illumination [19,20,21]. Chandraker et al. [22,23,24] utilize motion cues to jointly recover the shape and BRDF of objects from images under known directional illumination. Hui et al. [25] recover SVBRDFs and shape from multiple images under known illuminations. All of these methods require some form of calibrated acquisition; in contrast, we wish to capture SVBRDFs and normal maps "in-the-wild".

Recent work has shown promising results for "in-the-wild" BRDF acquisition. Hui et al. [26] demonstrate that the collocated camera-light setup on mobile devices is sufficient to reconstuct SVBRDFs and normals. They need capture 30+ images and calibrate them to reconstruct SVBRDFs; we aim to do this from a single image. Aittala et al. [8] propose using a flash/no-flash image pair to reconstruct *stochastic* SVBRDFs and normals using a slow optimization-based

scheme. Our method can handle a larger class of materials and is orders of magnitude faster.

**Deep learning-based Material Estimation:** Inspired by the success of deep learning for a variety of vision and graphics tasks, recent work has looked at CNN-based material recognition and estimation. Bell et al. [27] train a material parsing network using crowd-sourced labeled data. However, their material recongition is driven more by object context, rather than appearance. Liu et al. [28] demonstrate image-based material editing using a network trained to recover homogenous BRDFs. Methods have been proposed to decompose images into their intrinsic image components which are an intermediate representation for material and shape [29,30,31]. Rematas et al. [32] train a CNN to reconstruct the reflectance map – a convolution of the BRDF with the illumination – from a single image of a shape from a known class. In subsequent work, they disentangle the reflectance map into the BRDF and illumination [33]. Neither of these methods handle SVBRDFs, nor do they recover fine surface normal details. Kim et al. [34] reconstruct a homogeneous BRDF by training a network to aggregate multi-view observations of an object of known shape .

Similar to us, Aittala et al. [9] and Li et al. [10] reconstruct SVBRDFs and surface normals from a single image of a near-planar surface. Aittala et al. use a neural style transfer-based optimization approach to iteratively estimate BRDF parameters, however, they can only handle stationary textures and there is no correspondence between the input image and the reconstructed BRDF [9]. Li et al. use supervised learning to train a CNN to predict SVBRDF and normals from a single image captured under environment illumination [10]. Their training set is small, which necessitates a self-augmentation method to generate training samples from unlabeled real data. Further, they train a different set of networks for each parameter (diffuse texture, normals, specular albedo and roughness) and each material type (wood, metal, plastic). We demonstrate that by using our novel CNN architecture, supervised training on a high-quality dataset and acquisition under flash illumination, we are able to (a) reconstruct all these parameters with a single network, (b) learn a latent representation that also enables material recognition and editing, (c) produce results that are significantly better qualitatively and quantitatively.

## 3   Acquisition Setup and SVBRDF Dataset

In this section, we describe the setup for single image SVBRDF acquisition and the dataset we use for learning.

*Setup* Our goal is to reconstruct the spatially-varying BRDF of a near planar surface from a single image captured by a mobile phone with the flash turned on for illumination. We assume that the $z$-axis of the camera is approximately perpendicular to the planar surface (we explicitly evaluate against this assumption in our experiments). For most mobile devices, the position of the flash light is usually very close to the position of the camera, which provides us a univariate sampling of a isotropic BRDF [26]. We argue that by imaging with a collocated

Fig. 2. Examples of our material types.

| Materials | Train | Test | Materials | Train | Test |
|-----------|-------|------|-----------|-------|------|
| fabric | 165 | 29 | polymer | 33 | 6 |
| ground | 23 | 4 | stone-diff | 177 | 30 |
| leather | 10 | 2 | stone-spec | 38 | 6 |
| metal | 82 | 13 | wood | 60 | 10 |

Table 1. Distribution of materials in our training and test sets.

camera and point light, we can have additional constraints that yield better BRDF reconstructions compared to acquisition under just environment illumination.

Our surface appearance is represented by a microfacet parametric BRDF model [15]. Let $\mathbf{d}_i$, $\mathbf{n}_i$, $r_i$ be the diffuse color, normal and roughness, respectively, at pixel $i$. Our BRDF model is defined as:

$$\rho(\mathbf{d}_i, \mathbf{n}_i, r_i) = \mathbf{d}_i + \frac{D(\mathbf{h}_i, r_i)F(\mathbf{v}_i, \mathbf{h}_i)G(\mathbf{l}_i, \mathbf{v}_i, \mathbf{h}_i, r_i)}{4(\mathbf{n}_i \cdot \mathbf{l}_i)(\mathbf{n}_i \cdot \mathbf{v}_i)} \tag{1}$$

where $\mathbf{v}_i$ and $\mathbf{l}_i$ are the view and light directions and $\mathbf{h}_i$ is the half angle vector. Given an observed image $I(\mathbf{d}_i, \mathbf{n}_i, r_i, \mathbf{L})$, captured under unknown illumination $\mathbf{L}$, we wish to recover the parameters $\mathbf{d}_i$, $\mathbf{n}_i$ and $r_i$ for each pixel $i$ in the image. Please refer to the supplementary material for more details on the BRDF model.

*Dataset* We train our network on the Adobe Stock 3D Material dataset[1], which contains 688 materials with high resolution (4096 × 4096) spatially-varying BRDFs. Part of the dataset is created by artists while others are captured using a scanner. We use 588 materials for training and 100 materials for testing. For data augmentation, we randomly crop 12, 8, 4, 2, 1 image patches of size 512, 1024, 2048, 3072, 4096. We resize the image patches to a size of 256 × 256 for processing by our network. We flip patches along $x$ and $y$ axes and rotate them in increments of 45 degrees. Thus, for each material type, we have 270 image patches.[2] We randomly scale the diffuse color, normal and roughness for each image patch to prevent the network from overfitting and memorizing the materials. We manually segment the dataset into 8 materials types. The distribution is shown in Table 1, with an example visualization of each material type in Figure 2. More details on rendering the dataset are in supplementary material.

## 4   Network Design for SVBRDF Estimation

In this section, we describe the components of our CNN designed for single-image SVBRDF estimation. The overall architecture is illustrated in Figure 3.

[1] https://stock.adobe.com/3d-assets
[2] The total number of image patches for each material can be computed as $(12 + 8 + 4 + 2 + 1) \times (1 + 2 + 7) = 270$.
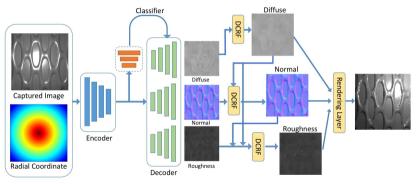
**Fig. 3.** Our network for SVBRDF estimation consists of an encoder, three decoder blocks with skip links to retrieve SVBRDF components, a rendering layer and a material classifier, followed by a DCRF for refinement (not visualized). See Section 4 for how our architectural choices are influenced by the problem structure of SVBRDF estimation and supplementary material for the hyperparameter details.

### 4.1   Considerations for Network Architecture

Single-image SVBRDF estimation is an ill-posed problem. Thus, we adopt a data-driven approach with a custom-designed CNN that reflects physical intuitions.

   Our basic network architecture consists of a single encoder and three decoders which reconstruct the three spatially-varying BRDF parameters: diffuse color $\mathbf{d}_i$, normals $\mathbf{n}_i$ and roughness $r_i$. The intuition behind using a single encoder is that different BRDF parameters are correlated, thus, representations learned for one should be useful to infer the others, which allows significant reduction in the size of the network. The input to the network is an RGB image, augmented with the pixel coordinates as a fourth channel. We add the pixel coordinates since the distribution of light intensities is closely related to the location of pixels, for instance, the center of the image will usually be much brighter. Since CNNs are spatially invariant, we need the extra signal to let the network learn to behave differently for pixels at different locations. Skip links are added to connect the encoder and decoders to preserve details of BRDF parameters.

   Another important consideration is that in order to model global effects over whole images like light intensity fall-off or large areas of specular highlights, it is necessary for the network to have a large receptive field. To this end, our encoder network has seven convolutional layers of stride 2, so that the receptive field of every output pixel covers the entire image.

### 4.2   Loss Functions for SVBRDF Estimation

For each BRDF parameter, we have an L2 loss for direct supervision. We now describe other losses for learning a good representation for SVBRDF estimation.

*Rendering layer* Since our eventual goal is to model the surface appearance, it is important to balance the contributions of different BRDF parameters. Therefore, we introduce a differentiable rendering layer that renders our BRDF model
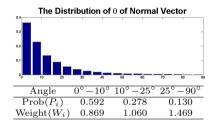
**The Distribution of θ of Normal Vector**

**Table 2.** The $\theta$ distribution of the normal vector in the dataset, where $\theta$ is the angle between normal vector and $z$ axis. To avoid the network from over-smoothing the normal map, we group normal vectors into three bins according to $\theta$. With probability $P_i$ for bin $i$, its weight is $W_i = 0.7 + 1/10P_i$.

| Angle | $0° - 10°$ | $10° - 25°$ | $25° - 90°$ |
|---|---|---|---|
| Prob($P_i$) | 0.592 | 0.278 | 0.130 |
| Weight($W_i$) | 0.869 | 1.060 | 1.469 |

(Eqn. 1) under the known input lighting. We add a reconstruction loss based on the difference between these renderings with the predicted parameters and renderings with ground-truth BRDF parameters. The gradient can be backpropagated through the rendering layer to train the network. In addition to rendering the image under the input lighting, we also render images under by *novel* lights. For each batch, we create novel lights by randomly sampling the the point light source on the upper hemisphere. This ensures that the network does not overfit to collocated illumination and is able to reproduce appearance under other light conditions. The final loss function for the encoder-decoder part of our network is:

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_r \mathcal{L}_r + \lambda_{rec} \mathcal{L}_{rec}, \qquad (2)$$

where $\mathcal{L}_d$, $\mathcal{L}_n$, $\mathcal{L}_r$ and $\mathcal{L}_{rec}$ are the L2 losses for diffuse, normal, roughness and rendered image predictions, respectively. Here, $\lambda$'s are positive coefficients to balance the contributions of various terms, which are set to 1 in our experiments.

Since we train on near planar surfaces, the majority of the normal directions are flat. Table 2 shows the normal distributions in our dataset. To prevent the network from over-smoothing the normals, we group the normal directions into different bins and for each bin we assign a different weight when computing the L2 error. This balance various normal directions in the loss function.

*Material Classification* The distribution of BRDF parameters is closely related to the surface material type. However, training separate networks for different material types similar to [10] is expensive. Also the size of the network grows linearly with the number of material types, which limits utility. Instead, we propose a split-merge network with very little computational overhead.

Given the highest level of features extracted by the encoder, we send the feature to a classifier to predict its material type. Then we evaluate the BRDF parameters for each material type and use the classification results as weights (the output of softmax layer). This averages the prediction from different material types to obtain the final BRDF reconstruction results. Suppose we have $N$ channels for BRDF parameters and $K$ material types. To output the BRDF reconstruction for each type of material, we only modify the last convolutional layer of the decoder so that the output channel will be $K \times N$ instead of $N$. In practice, we set $K$ to be 8, as shown in Table 1.

The classifier is trained together with the encoder and decoder from scratch, with the weights of each label set to be inversely proportional to the number of examples in Table 1 to balance different material types in the loss function. The

overall loss function of our network with the classifier is

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_r \mathcal{L}_r + \lambda_{rec} \mathcal{L}_{rec} + \lambda_{cls} \mathcal{L}_{cls}, \tag{3}$$

where $\mathcal{L}_{cls}$ is cross entropy loss and $\lambda_{cls} = 0.0005$ to limit the gradient magnitude.

### 4.3 Designing DCRFs for Refinement

The prediction of our base network is quite reasonable. However, accuracy may further be enhanced by post-processing through a DCRF (trained end-to-end). *Diffuse color refinement* For diffuse prediction, when capturing the image of specular materials, parts of the surface might be saturated by specular highlight. This can sometimes lead to artifacts in the diffuse color prediction since the network has to hallucinate the diffuse color from nearby pixels. To remove such artifacts, we incorporate a densely connected continuous conditional random field (DCRF) [35] to smooth the diffuse color prediction. Let $\hat{\mathbf{d}}_i$ be the diffuse color prediction of network at pixel $i$, $\mathbf{p}_i$ be its position and $\bar{\mathbf{I}}_i$ is the normalized diffuse RGB color of the input image. We use the normalized color of the input image to remove the influence of light intensity when measuring the similarity between two pixels. The energy function of the dense connected CRF that is minimized over $\{\mathbf{d}_i\}$ for diffuse prediction is defined as:

$$\sum_{i=1}^{N} \alpha_i^d (\mathbf{d}_i - \hat{\mathbf{d}}_i)^2 + \sum_{i,j}^{N} (\mathbf{d}_i - \mathbf{d}_j)^2 \left( \beta_1^d \kappa_1(\mathbf{p}_i; \mathbf{p}_j) + \beta_2^d \kappa_2(\mathbf{p}_i, \bar{\mathbf{I}}_i; \mathbf{p}_j, \bar{\mathbf{I}}_j) + \beta_3^d \kappa_3(\mathbf{p}_i, \hat{\mathbf{d}}_i; \mathbf{p}_j, \hat{\mathbf{d}}_j) \right). \tag{4}$$

Here $\kappa_i$ are Gaussian smoothing kernels, while $\alpha_i^d$ and $\{\beta_i^d\}$ are coefficients to balance the contribution of unary and smoothness terms. Notice that we have a spatially varying $\alpha_i^d$ to allow different unary weights for different pixels. The intuition is that artifacts usually occur near the center of images with specular highlights. For those pixels, we should have lower unary weights so that the CRF learns to predict their diffuse color from nearby pixels.
*Normal refinement* Once we have the refined diffuse color, we can use it to improve the prediction of other BRDF parameters. To reduce the noise in normal prediction, we use a DCRF with two smoothness kernels. One is based on the pixel position while the other is a bilateral kernel based on the position of the pixel and the gradient of the diffuse color. The intuition is that pixels with similar diffuse color gradients often have similar normal directions. Let $\hat{\mathbf{n}}_i$ be the normal predicted by the network. The energy function for normal prediction is defined as

$$\min_{\{\mathbf{n}_i\}} : \sum_{i=1}^{N} \alpha^n (\mathbf{n}_i - \hat{\mathbf{n}}_i)^2 + \sum_{i,j}^{N} (\mathbf{n}_i - \mathbf{n}_j)^2 \left( \beta_1^n \kappa_1(\mathbf{p}_i; \mathbf{p}_j) + \beta_2^n \kappa_2(\mathbf{p}_i, \Delta\mathbf{d}_i; \mathbf{p}_j, \Delta\mathbf{d}_j) \right) \tag{5}$$

*Roughness refinement* Since we use a collocated light source to illuminate the material, once we have the normal and diffuse color predictions, we can use them to estimate the roughness term by either grid search or using a gradient-based method. However, since the microfacet BRDF model is not convex nor monotonic with respect to the roughness term, there is no guarantee that we can find a global

minimum. Also, due to noise from the normal and diffuse predictions, as well as environment lighting, it is difficult to get an accurate roughness prediction using optimization alone, especially when the glossiness in the image is not apparent. Therefore, we propose to combine the output of the network and the optimization method to get a more accurate roughness prediction. We use a DCRF with two unary terms, $\hat{r}_i$ and $\tilde{r}_i$, given by the network prediction and the coarse-to-fine grid search method of [26], respectively:

$$\min_{\{r_i\}} : \sum_{i=1}^{N} \alpha_{i0}^r (r_i - \hat{r}_i)^2 + \alpha_{i1}^r (r_i - \tilde{r}_i)^2 + \sum_{i,j}^{N} (r_i - r_j)^2 \left( \beta_0 \kappa_0(\mathbf{p}_i; \mathbf{p}_j) + \beta_1 \kappa_1(\mathbf{p}_i, \mathbf{d}_i; \mathbf{p}_j, \mathbf{d}_j) \right) \quad (6)$$

All DCRF coefficients are learned in an end-to-end manner using [36]. Here, we have a different set of DCRF parameters for each material type to increase model capacity. During both training and testing, the classifier output is used to average the parameters from different material types, to determine the DCRF parameters. More implementation details are in supplementary material.

## 5    Experiments

In this section, we demonstrate our method and compare it to baselines on a wide range of synthetic and real data.

*Rendering synthetic training dataset* To create our synthetic data, we apply the SVBRDFs on planar surfaces and render them using Mitsuba [37] with the BRDF importance sampling suggested in [38]. We choose a camera field of view of $43.35°$ to mimic typical mobile phone cameras. To better model real-world lighting conditions, we render images under a combination of a dominant point light (flash) and an environment map. We use the 49 environment maps used in [10], with random rotations. We sample the light source position from a Gaussian distribution centered at the camera to make the inference robust to differences in real-world mobile phones. We render linear images, though clamped to $(0, 1)$ to mimic cameras with insufficient dynamic range. However, we still wish to reconstruct the full dynamic range of the SVBRDF parameters. To aid in this, we can render HDR images using in-our network rendering layer and compute reconstruction error w.r.t HDR ground truth images. In practice, this leads to unstable gradients in training; we mitigate this by applying a gamma of 2.2 and minor clamping to $(0, 1.5)$ when computing the image reconstruction loss. We find that this, in addition to our L2 losses on the SVBRDF parameters, allows us to hallucinate details from saturated images.

*Training details* We use Adam optimizer [39] to train our network. We set $\beta_1 = 0.5$ when training the encoder and decoders and $\beta_1 = 0.9$ when training the classifier. The initial learning rate is set to be $10^{-4}$ for the encoder, $2 \times 10^{-4}$ for the three decoders and $2 \times 10^{-5}$ for the classifier. We cut down the learning rate by half in every two epochs. Since we find that the diffuse color and normal direction contribute much more to the final appearance, we first train their encoder-decoders for 15 epochs, then we fix the encoder and train the roughness decoder separately for 8 epochs. Next, we fix the network and train the parameters for the DCRFs, using Adam optimizer to update their coefficients.
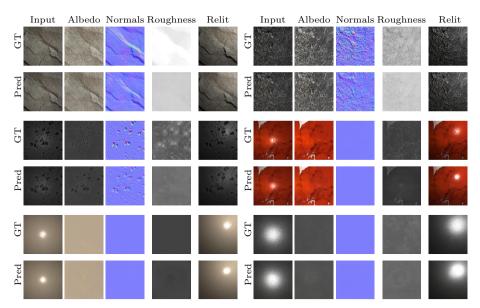
**Fig. 4.** BRDF reconstruction results from our full method (`clsCRF-pt` in Table 3) on the test set. We compare the ground truth parameters with our reconstructions as well as renderings of these parameters under novel lighting. The accuracy of our renderings indicates the accuracy of our method.
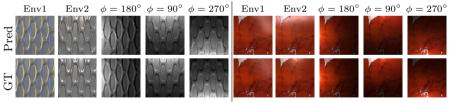


**Fig. 5.** Materials estimated with our method and rendered under two environment lights and three point lights (placed on a unit sphere at $\theta = 50°$ and various $\phi$ angles).

## 5.1   Results on Synthetic Data

*Qualitative results* Figure 5.1 shows results of our network on our synthetic test dataset. We can observe that spatially varying surface normals, diffuse albedo and roughness are recovered at high quality, which allows relighting under novel light source directions that are very different from the input. To further demonstrate our BRDF reconstruction quality, in Figure 5, we show relighting results under different environment maps and point lights at oblique angles. Note that our relighting results closely match the ground truth even under different lighting conditions; this indicates the accuracy of our reconstructions.

We next perform quantitative ablation studies to evaluate various components of our network design and study comparisons to prior work.

*Effects of material classifier and DCRF:* The ablation study summarized in Table 3 shows that adding the material classifier reduces the L2 error for SVBRDF and normal estimation, as well as rendering error. This validates the intuition

**Table 3.** Left to right: basic encoder-decoder, adding material classifier, adding DCRF and a pure material classifier. −`pt` indicates training and testing with dominant point and environment lighting.
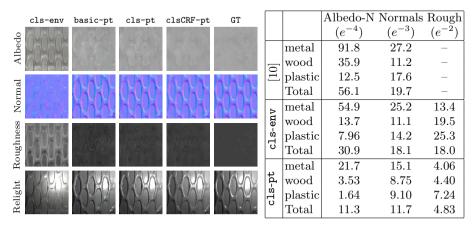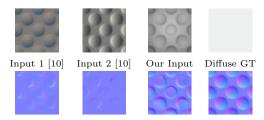
| Method | basic-pt | cls-pt | clsCRF-pt | clsOnly-pt |
|---|---|---|---|---|
| Albedo ($e^{-3}$) | 7.78 | 7.58 | **7.42** | |
| Normal ($e^{-2}$) | 1.55 | 1.52 | **1.50** | |
| Rough ($e^{-2}$) | 8.75 | 8.55 | **8.53** | |
| Classify (%) | | **73.65** | **73.65** | 54.96 |



**Fig. 6.** Qualitative comparison of BRDF reconstruction results of different variants of our network. The notation is the same as Table 3 and −`env` represents environment illumination.

| | | Albedo-N $(e^{-4})$ | Normals $(e^{-3})$ | Rough $(e^{-2})$ |
|---|---|---|---|---|
| [10] | metal | 91.8 | 27.2 | – |
| | wood | 35.9 | 11.2 | – |
| | plastic | 12.5 | 17.6 | – |
| | Total | 56.1 | 19.7 | – |
| cls-env | metal | 54.9 | 25.2 | 13.4 |
| | wood | 13.7 | 11.1 | 19.5 |
| | plastic | 7.96 | 14.2 | 25.3 |
| | Total | 30.9 | 18.1 | 18.0 |
| cls-pt | metal | 21.7 | 15.1 | 4.06 |
| | wood | 3.53 | 8.75 | 4.40 |
| | plastic | 1.64 | 9.10 | 7.24 |
| | Total | 11.3 | 11.7 | 4.83 |

**Table 4.** BRDF reconstruction accuracy for different material types in our test set. Albedo-N is normalized diffuse albedo as in [10], that is, the average norm of each pixel will be 0.5.



**Fig. 7.** The first two inputs rendered under different environment maps are very different. Thus, the normals recovered using [10] are inaccurate. Our method uses point illumination (third input) which alleviates the problem, and produces better normals.

that the network can exploit the correlation between BRDF parameters and material type to produce better estimates. We also observe that training the classifier together with the BRDF reconstruction network results in a material classification error of 73.65%, which significantly improves over just our pure material classification network that achieves 54.96%. This indicates that features trained for BRDF estimation are also useful for material recognition. In our experiments, incorporating the classifier without using its output to fuse BRDF reconstruction results does not improve BRDF estimation. Figure 6 shows the reconstruction result on a sample where the classifier and the DCRF qualitatively improve the BRDF estimation, especially for the diffuse albedo.
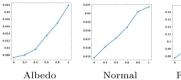
|   |   |   |   |
|---|---|---|---|
| Albedo | Normal | Roughness | Classification |

**Fig. 8.** SVBRDF estimation errors for relative intensities of environment against point light ranging from 0 to 0.8.

*Effect of acquisition under point illumination* Next we evaluate the effect of using point illumination during acquisition. For this, we train and test two variants of our full network – one on images rendered under only environment illumination (-`env`) and another on images illuminated by a point light besides environment illumination (-`pt`). Results are in Table 4 with qualitative visualizations in Figure 6. The model from [10] in Table 4, which is trained for environment lighting, performs slightly worse than our environment lighting network `cls-env`. But our network trained and evaluated on point and environment lighting, `cls-pt`, easily outperforms both. We argue this is because a collocated point light creates more consistent illumination across training and test images, while also capturing higher frequency information. Figure 7 illustrates this: the appearance of the same material under different environment lighting can significantly vary and the network has to be invariant to this, limiting reconstruction quality.

*Relative effects of flash and environment light intensities* In Figure 8, we train and test on a range of relative flash intensities, showing our network works well for each. Note that as relative flash intensity decreases, errors increase, which justifies our use of flash light. Using flash and no-flash pairs can help remove environment lighting, but needs alignment of two images, which limits applicability.

## 5.2    Results on Real Data

*Acquisition setup* To verify the generalizabity of our method to real data, we show results on real images captured with different mobile devices in both indoor and outdoor environments. We capture linear RAW images (with potentially clipped highlights) with the flash enabled, using the Adobe Lightroom Mobile app. The mobile phones were hand-held and the optical axis of the camera was only approximately perpendicular to the surfaces (see Figure 1).

*Qualitative results with different mobile phones* Figure 9 presents SVBRDF and normal estimation results for real images captured with three different mobile devices: Huawei P9, Google Tango and iPhone 6s. We observe that even with a single image, our network successfully predicts the SVBRDF and normals, with images rendered using the predicted parameters appear very similar to the input. Also, the exact same network generalizes well to different mobile devices, which shows that our data augmentation successfully helps the network factor out variations across devices. For some materials with specular highlights, the network can hallucinate information lost due to saturation. The network can also reconstruct reasonable normals even for complex instances.

*A failure case* In Figure 10, we show a failure case. Here, the material is misclassified as metal which causes the specular highlight in the center of image to be
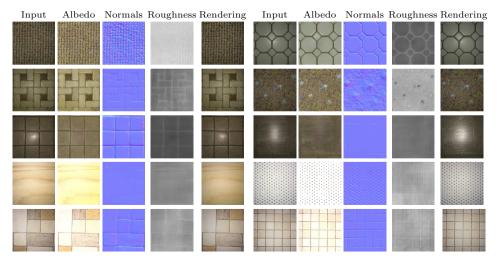
| Input | Albedo | Normals | Roughness | Rendering | Input | Albedo | Normals | Roughness | Rendering |
|---|---|---|---|---|---|---|---|---|---|



**Fig. 9.** BRDF reconstruction results on real data. We tried different mobile devices to capture raw images using Adobe LightRoom. The input images in were captured using a Huawei P9 (first three rows), Google Tango (fourth row) and iPhone 6s (fifth row), all with a handheld mobile phone where the z-axis of camera was only approximately perpendicular to the sample surface.

over-suppressed. In future work, we may address this with more robust material classification, potentially exploiting datasets like [27].

*Material editing* We can edit the reconstructed SVBRDFs by transferring material properties. Figure 11 shows an example where we transfer BRDF properties across different material types and render in a novel lighting condition.

### 5.3   Further Comparisons with Prior Works

*Comparison with two-shot BRDF method [8]* The two-shot method of [8] can only handle images with stationary texture while our method can reconstruct arbitrarily varying SVBRDFs. For a meaningful comparison, in Figure 13, we compare our method with [8] on a rendered stationary texture. We can see that even for this restrictive material type, the normal maps reconstructed by the two methods are quite similar, but the diffuse map reconstructed by our method is closer to ground truth. While [8] takes about 6 hours to reconstruct a patch of size $192 \times 192$, our method requires 2.4 seconds. The aligned flash and no-flash pair for [8] is not trivial to acquire (especially on mobile cameras with effects like rolling shutter), making our single image BRDF estimation more practical.

*Comparison normals with environment light and photometric stereo* In Figure 12, we compare our normal map and the output of a single image SVBRDF reconstruction method under environment lighting [10] with photometric stereo [25]. We observe that the normals reconstructed by our method are of higher quality than [10], with details comparable or sharper than photometric stereo.

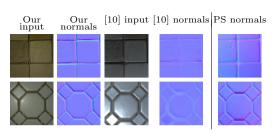**Appendix** The appendix provides further experiments and details, including:
– Details of data augmentation, continuous DCRF and visualization of weights

Input   Albedo   Normal   Roughness   Relighting

**Fig. 10.** A failure case, due to incorrect material classification into *metal*, which causes the specularity to be over-smoothed.



**Fig. 11.** A material editing example. Having reconstructed the SVBRDF and normals of the two samples, we swap the original geometry and material properties, then relight under novel illumination.



Our input   Our normals   [10] input   [10] normals   PS normals

**Fig. 12.** Comparison of normal maps using our method and [10], with photometric stereo as reference. Even with a lightweight acquisition system, our network predicts high quality normal maps.



Flash   [8]   Ours   GT

Guide   [8]   Ours   GT

**Fig. 13.** Comparison with [8], which requires two images, assumes stationary textures and takes over 6 hours (with GPU acceleration), yet our result is more accurate.

- Spherical renderings of estimated real spatially varying BRDFs
- Visualization of SVBRDF estimation with respect to prediction error
- Further qualitative results on synthetic and real data.

## 6    Discussion

We have proposed a framework for acquiring spatially-varying BRDF using a single mobile phone image. Our solution uses a convolutional neural network whose architecture is specifically designed to capture various physical insights into the problem of BRDF estimation. We also propose a dataset that is larger and better-suited to the problem of material estimation as compared to prior ones, as well as simple acquisition settings that are nevertheless effective for SVBRDF estimation. Our network generalizes very well to real data, obtaining high-quality results in unconstrained test environments. A key goal for our work is to take accurate material estimation from expensive and controlled lab setups, into the hands of non-expert users with consumer devices, thereby opening the doors to new applications. Our future work will take the next step of acquiring SVBRDF with unknown shapes, as well as study the role of other semantic signals such as object categories in material estimation.

# References

1. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658
2. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision, Springer (2016) 628–644
3. Nicodemus, F.E.: Directional reflectance and emissivity of an opaque surface. Applied optics **4**(7) (1965) 767–775
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. (2012)
6. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (2000) 145–156
7. Marschner, S.R., Westin, S.H., Lafortune, E.P., Torrance, K.E., Greenberg, D.P.: Image-based brdf measurement including human skin. In: Rendering Techniques 99. Springer (1999) 131–144
8. Aittala, M., Weyrich, T., Lehtinen, J., et al.: Two-shot svbrdf capture for stationary materials. ACM Trans. Graph. **34**(4) (2015) 110–1
9. Aittala, M., Aila, T., Lehtinen, J.: Reflectance modeling by neural texture synthesis. ACM Transactions on Graphics (TOG) **35**(4) (2016)  65
10. Li, X., Dong, Y., Peers, P., Tong, X.: Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. ACM Trans. Graph. **36**(4) (July 2017) 45:1–45:11
11. Blinn, J.F., Newell, M.E.: Texture and reflection in computer generated images. Communications of the ACM **19**(10) (1976) 542–547
12. Cook, R.L., Torrance, K.E.: A reflectance model for computer graphics. ACM Transactions on Graphics (TOG) **1**(1) (1982) 7–24
13. Ward, G.J.: Measuring and modeling anisotropic reflection. ACM Transactions on Graphics (TOG) **26**(2) (1992) 265–272
14. Oren, M., Nayar, S.K.: Generalization of the lambertian model and implications for machine vision. International Journal on Computer Vision (IJCV) **14**(3) (1995) 227–251
15. Burley, B.: Physically-based shading at disney. In: ACM SIGGRAPH 2012 Courses. (2012)
16. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. ACM Transactions on Graphics (TOG) **22**(3) (2003) 759–769
17. Nielsen, J.B., Jensen, H.W., Ramamoorthi, R.: On optimal, minimal brdf sampling for reflectance acquisition. ACM Transactions on Graphics (TOG) **34**(6) (2015) 186
18. Xu, Z., Nielsen, J.B., Yu, J., Jensen, H.W., Ramamoorthi, R.: Minimal brdf sampling for two-shot near-field reflectance acquisition. ACM Transactions on Graphics (TOG) **35**(6) (2016) 188
19. Romeiro, F., Vasilyev, Y., Zickler, T.: Passive reflectometry. In: European Conference on Computer Vision (ECCV). (2008)

20. Romeiro, F., Zickler, T.: Blind reflectometry. In: European Conference on Computer Vision (ECCV). (2010)
21. Oxholm, G., Nishino, K.: Shape and reflectance estimation in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **38**(2) (2016) 376–389
22. Chandraker, M.: On shape and material recovery from motion. In: European Conference on Computer Vision, Springer (2014) 202–217
23. Chandraker, M.: What camera motion reveals about shape with unknown brdf. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 2171–2178
24. Chandraker, M.: The information available to a moving observer on shape with unknown, isotropic brdfs. IEEE transactions on pattern analysis and machine intelligence **38**(7) (2016) 1283–1297
25. Hui, Z., Sankaranarayanan, A.C.: A dictionary-based approach for estimating shape and spatially-varying reflectance. In: International Conference on Computational Photography (ICCP). (2015)
26. Hui, Z., Sunkavalli, K., Lee, J.Y., Hadap, S., Wang, J., Sankaranarayanan, A.C.: Reflectance capture using univariate sampling of BRDFs. In: IEEE Intl. Conf. Computer Vision (ICCV). (2017)
27. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. Computer Vision and Pattern Recognition (CVPR) (2015)
28. Liu, G., Ceylan, D., Yumer, E., Yang, J., Lien, J.M.: Material editing using a physically based rendering network. ICCV (2017)
29. Narihira, T., Maire, M., Yu, S.X.: Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2992–2992
30. Shelhamer, E., Barron, J.T., Darrell, T.: Scene intrinsics and depth from a single image. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2015) 37–44
31. Shi, J., Dong, Y., Su, H., Yu, S.X.: Learning non-lambertian object intrinsics across shapenet categories. arXiv preprint arXiv:1612.08510 (2016)
32. Rematas, K., Ritschel, T., Fritz, M., Gavves, E., Tuytelaars, T.: Deep reflectance maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4508–4516
33. Georgoulis, S., Rematas, K., Ritschel, T., Fritz, M., Van Gool, L., Tuytelaars, T.: Delight-net: Decomposing reflectance maps into specular materials and natural illumination. arXiv preprint arXiv:1603.08240 (2016)
34. Kim, K., Gu, J., Tyree, S., Molchanov, P., Nießner, M., Kautz, J.: A lightweight approach for on-the-fly reflectance estimation. arXiv preprint arXiv:1705.07162 (2017)
35. Ristovski, K., Radosavljevic, V., Vucetic, S., Obradovic, Z.: Continuous conditional random fields for efficient regression in large fully connected graphs. In: AAAI. (2013)
36. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. arXiv preprint arXiv:1704.02157 (2017)
37. Jakob, W.: Mitsuba renderer (2010) http://www.mitsuba-renderer.org.
38. Karis, B., Games, E.: Real shading in unreal engine 4. Proc. Physically Based Shading Theory Practice (2013)

39. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
40. Ristovski, K., Radosavljevic, V., Vucetic, S., Obradovic, Z.: Continuous conditional random fields for efficient regression in large fully connected graphs. In: AAAI. (2013)

## A    Further Experimental Analysis

*Error distribution on test set* To provide better intuition into our quantitative results, we plot the distributions of prediction errors for diffuse albedo ($\mathcal{L}_d$), normals ($\mathcal{L}_n$), roughness ($\mathcal{L}_r$) and relighting ($\mathcal{L}_{rec}$) in Figure 14. Then, we sort the BRDF reconstruction results in the test set according to $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec}$ and illustrate the estimation and relighting quality for a random material picked from various percentiles of the above error distribution. The qualitative comparison is shown in Figure 15.

Even though our network is trained end-to-end, we observe physically meaningful trends in Figure 14. For instance, the materials that correspond to lower error percentiles tend to have flat normals, uniform diffuse color and wide specular lobes. On the other hand, materials with higher errors tend to have more complex normals, stronger local variations in diffuse color and roughness, or more prominent highlights. This demonstrates the benefits of our network design which considers the underlying problem structure. We also observe that normal and diffuse color estimates are quite accurate even at error percentiles higher than 50, which contributes to reasonable relighting results under *novel* lighting even at high error percentiles.
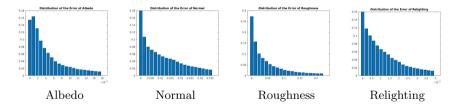


Albedo        Normal        Roughness        Relighting

**Fig. 14.** From the left to the right, error distributions of diffuse albedo, normal, roughness and relighting.

## B    Further Results on Real Data

*Comparison with photometric stereo as reference* In Figure 16, we compare the normals estimated by our method with that of [10], using the normal map from photometric stereo as reference. In the main paper, we use the photometric stereo method of [25]. Here, we instead use a simpler but more robust method. We acquire images of a material sample under 52 different directional point light sources. We abandon the 5 most brightest observations and 5 darkest observations and use the rest for a Lambertian photometric stereo. We find such a method to be quite robust to shadows, as well as the effects of complex BRDF such as glossiness or specularity. We observe that our method is able to capture very fine details in the normal map, in particular, better than the method of [10]. For instance, note the detail within the grooves of the material in the first and third
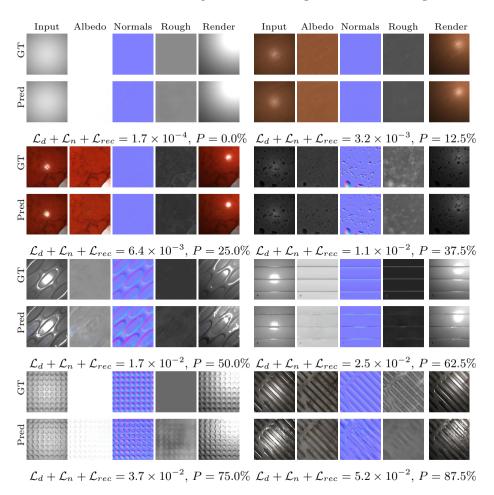
$\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 1.7 \times 10^{-4},\ P = 0.0\%$     $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 3.2 \times 10^{-3},\ P = 12.5\%$

$\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 6.4 \times 10^{-3},\ P = 25.0\%$     $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 1.1 \times 10^{-2},\ P = 37.5\%$

$\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 1.7 \times 10^{-2},\ P = 50.0\%$     $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 2.5 \times 10^{-2},\ P = 62.5\%$

$\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 3.7 \times 10^{-2},\ P = 75.0\%$     $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec} = 5.2 \times 10^{-2},\ P = 87.5\%$

**Fig. 15.** SVBRDF estimation results sorted according to the prediction error. The error here is defined as $\mathcal{L}_d + \mathcal{L}_n + \mathcal{L}_{rec}$. We do not consider $\mathcal{L}_r$ here roughness has relatively smaller influence towards the final appearance of the surface. Here, $P$ denotes the percentage of samples in the test set with error higher than the considered sample.

rows. This demonstrates the efficacy of the proposed method for normal and SVBRDF estimation.

*Real data results in unconstrained environments* In Figure 17, we show several more examples of surface normal and BRDF estimation with real data using the proposed method. The images are acquired in unconstrained settings with the camera flash enabled, for several different material types derived from wood flooring, tiles, carpets and so on. In all rows, the mobile phone is hand-held and only approximately parallel to the surface. In each case, we observe that the recovered normals, as well as the diffuse albedo and specular components of the
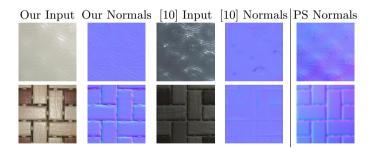
**Fig. 16.** Comparison of normal maps using our method and [10], with photometric stereo as reference. Even with a lightweight acquisition system, our network predicts high quality normal maps.
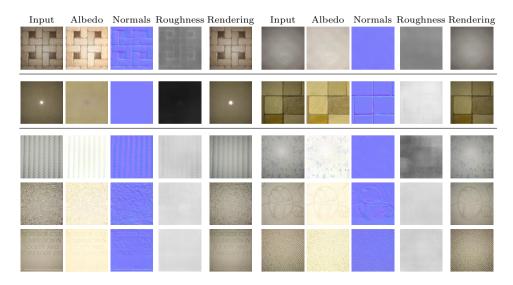


**Fig. 17.** SVBRDF estimation results on real data. All the input images are captured using a mobile phone. All the rows are imaged with a handheld mobile phone, where the z-axis of the camera is only approximately perpendicular to the sample surface. The inaccuracy in positional calibration of the camera is visible in the input image of the example in the second row of the first column, where the highlight is clearly not in the center of the image. However, our method still obtains reasonable normal and SVBRDF estimation results in all cases. The images in the first row are captured by iPhone 6s. The second row are captured by Huawei P9 while the last third rows are captured by Lenovo Phab 2. Our algorithm can handle new unknown devices very well.
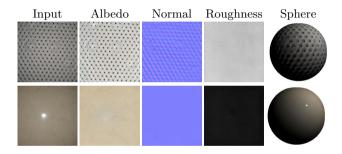
**Fig. 18.** Rendering of the estimated real spatially-varying BRDF on a sphere, under a very different oblique lighting direction.

spatially-varying BRDF appear qualitatively correct. In some cases, such as the second row of the first column, we observe that even very tight specular lobes are well-estimated, as evident from the lobe's compactness in the relighted image. The first row is captured by iPhone 6s, the second row by Huawei P9 and the last three rows by Lenovo Phab 2. Even though we never calibrate the mobile phone, our network generalizes very well to the new device.

*Another visualization for relighting* For another visualization of the normal and BRDF estimation on real data, we render the estimated material on a sphere illuminated under an oblique lighting direction that is very different from the input lighting. Recall that we only use an approximately planar patch of material as input. The BRDF estimation and relighted sphere are illustrated in Figure 18. We observe that the appearance of the sphere even under a novel lighting direction is quite reasonable.

## C   Microfacet BRDF Model

We use the microfacet BRDF model proposed in [38]. Let $\mathbf{d}_i$, $\mathbf{n}_i$, $r_i$ be the diffuse color, normal and roughness, respectively, at pixel $i$ and $I(\mathbf{d}_i, \mathbf{n}_i, r_i)$ be its intensity observed by the camera. Our BRDF model is defined as

$$I(\mathbf{d}_i, \mathbf{n}_i, r_i) = \mathbf{d}_i + \frac{D(\mathbf{h}_i, r_i)F(\mathbf{v}_i, \mathbf{h}_i)G(\mathbf{l}_i, \mathbf{v}_i, \mathbf{h}_i, r_i)}{4(\mathbf{n}_i \cdot \mathbf{l}_i)(\mathbf{n}_i \cdot \mathbf{v}_i)}, \tag{7}$$

where $\mathbf{v}_i$ and $\mathbf{l}_i$ are the view and light directions, while $\mathbf{h}_i$ is the half angle vector. Further, $D(\mathbf{h}_i, r_i)$, $F(\mathbf{v}_i, \mathbf{h}_i)$ and $G(\mathbf{l}_i, \mathbf{v}_i, \mathbf{h}_i, r_i)$ are the distribution, Fresnel and

geometric terms, respectively, which are defined as

$$D(\mathbf{h}_i, r_i) = \frac{\alpha_i^2}{\pi((\mathbf{n}_i \cdot \mathbf{h}_i)^2(\alpha_i^2 - 1) + 1)^2} \tag{8}$$

$$\alpha_i = r_i^2 \tag{9}$$

$$F(\mathbf{v}, \mathbf{h}) = (1 - F_0)2^{(-5.55473(\mathbf{v}\cdot\mathbf{h})-6.98316)\mathbf{v}\cdot\mathbf{h}} + F_0 \tag{10}$$

$$G(\mathbf{l}, \mathbf{v}, \mathbf{n}) = G_1(\mathbf{v}, \mathbf{n})G_1(\mathbf{l}, \mathbf{n}) \tag{11}$$

$$k_i = \frac{(r_i + 1)^2}{8} \tag{12}$$

$$G_1(\mathbf{v}, \mathbf{n}) = \frac{\mathbf{n} \cdot \mathbf{v}}{(\mathbf{n} \cdot \mathbf{v})(1 - k) + k} \tag{13}$$

$$G_1(\mathbf{l}, \mathbf{n}) = \frac{\mathbf{n} \cdot \mathbf{l}}{(\mathbf{n} \cdot \mathbf{v})(1 - k) + k}, \tag{14}$$

with $F_0$ the specular reflectance at normal incidence. For a dielectric material, the value of $F_0$ is determined by the index of refraction $\eta$:

$$F_0 = \frac{(1 - \eta)^2}{(1 + \eta)^2}. \tag{15}$$

For a conductor material, it is determined by the index of refraction $\eta$ and the absorption coefficient $\kappa$:

$$F_0 = \frac{(1 + \eta)^2 + \kappa^2}{(1 - \eta)^2 + \kappa^2}. \tag{16}$$

When rendering our dataset, we set $F_0 = 0.5$ for *metal* and $F_0 = 0.05$ for other kinds of materials. Figure 19 shows an example of smooth aluminum material rendered with $F_0 = 0.05$ and $F_0 = 0.5$. We observe that the material rendered with $F_0 = 0.5$ has a much larger area of specular highlight, which matches appearances of metals in practice.
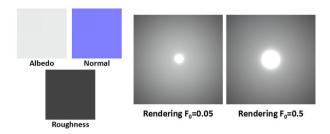


Albedo      Normal

Roughness

Rendering $F_0$=0.05      Rendering $F_0$=0.5

**Fig. 19.** An aluminum material rendered with different $F_0$. We obersve that when rendering with $F_0 = 0.5$ the area of specular highlight is much larger and better matches appearnaces of metals in the real world. For all other materials, we use $F_0 = 0.05$ as the most reasonable value.

# D    Details of Continuous DCRFs

We use continuous densely connected conditional random fields (DCRFs) for post-processing to remove artifacts caused by saturated highlights and noise in the prediction of the neural network [40,36]. We customize the DCRFs to better suit our problem of spatially-varying BRDF reconstruction. The distinguishing factor for our DCRF construction is the design of spatially varying weight maps that allow incorporating domain specific knowledge into the CRF inference. In the following, we will discuss the design and the intuition behind the usage of the weight map, as well as the details of training and inference for the DCRF.

*Weight Maps of DCRFs* We first discuss the DCRF for diffuse albedo prediction. Its energy function is defined as

$$\min_{\{\mathbf{d}_i\}} : \sum_{i=1}^{N} \alpha_i^d (\mathbf{d}_i - \hat{\mathbf{d}}_i)^2 + \sum_{i,j}^{N} (\mathbf{d}_i - \mathbf{d}_j)^2 \Big( \beta_1^d \kappa_1(\mathbf{p}_i; \mathbf{p}_j)$$
$$+ \beta_2^d \kappa_2(\mathbf{p}_i, \bar{\mathbf{I}}_i; \mathbf{p}_j, \bar{\mathbf{I}}_j) + \beta_3^d \kappa_3(\mathbf{p}_i, \hat{\mathbf{d}}_i; \mathbf{p}_j, \hat{\mathbf{d}}_j) \Big). \tag{17}$$

Here, the coefficient $\alpha_i^d$ is spatially varying. A larger $\alpha_i^d$ indicates greater confidence in the prediction from the neural network. Since we use a colocated point light source for illumination, an observation is that saturations caused by the specular highlight are usually in the middle of the image. Another observation is that since the flash illumination is white in color, the saturated pixels are usually white, which means the minimum of their RGB values will be large. Therefore, for regions near the center of the image or regions with specular highlights, we should have a smaller unary weight so that the DCRF may smooth out the artifacts. Based on these two observations, we define the weight map for the unary term $\alpha_i^d$ as

$$\alpha_i^d = \alpha_{i0}^d \max(1 - \exp(-\frac{\mathbf{p}_i^2}{\sigma_{d0}^2}), 1 - \exp(-\frac{(c_i^{min} - 1)^2}{\sigma_{d1}^2}))$$
$$+ \alpha_{i1}^d, \tag{18}$$

where $c_i^{min}$ is the minimum of the three color channels at pixel $i$:

$$c_i^{min} = \min(R_i, G_i, B_i). \tag{19}$$

Here, $\alpha_{i0}^d$ and $\alpha_{i1}^d$ are two learnable parameters. We set $\alpha_{i1}^d = 0$ and $\alpha_{i0}^d = 1$ at the beginning of the training process. We set $\sigma_{d0} = 0.5$ and $\sigma_{d1} = 0.08$ through the whole training process. Figure 20 shows examples of the weight map for diffuse albedo prediction.

For normal prediction, we do not observe such strong correlation between the prediction error and the position or intensity of the image. Therefore, we just set

a uniform weight for every pixel in the image. The energy function is defined as

$$\min_{\{\mathbf{n}_i\}} : \sum_{i=1}^{N} \alpha^n (\mathbf{n}_i - \hat{\mathbf{n}}_i)^2 + \sum_{i,j}^{N} (\mathbf{n}_i - \mathbf{n}_j)^2 \Big( \beta_1^n \kappa_1(\mathbf{p}_i; \mathbf{p}_j)$$
$$+ \beta_2^n \kappa_2(\mathbf{p}_i, \Delta \mathbf{d}_i; \mathbf{p}_j, \Delta \mathbf{d}_j) \Big), \tag{20}$$

where $\alpha^n$, $\beta_1^n$ and $\beta_2^n$ are learnable parameters that trade-off relative confidences in the unary, a pairwise smoothness prior and a prior on correlation between normals and albedo boundaries.

Finally, for roughness prediction, the energy function is defined as

$$\min_{\{r_i\}} : \sum_{i=1}^{N} \alpha_{i0}^r (r_i - \hat{r}_i)^2 + \alpha_{i1}^r (r_i - \tilde{r}_i)^2 + \sum_{i,j}^{N} (r_i - r_j)^2$$
$$\Big( \beta_1 \kappa_1(\mathbf{p}_i; \mathbf{p}_j) + \beta_2 \kappa_2(\mathbf{p}_i, \mathbf{d}_i; \mathbf{p}_j, \mathbf{d}_j) \Big), \tag{21}$$

where $\hat{r}_i$ is the prediction from the network and $\tilde{r}_i$ is the prediction from a grid search. We find that the prediction from grid search is usually only accurate near the glossy regions, which means these regions should have a larger $\alpha_{i1}^r$. Therefore, we define the weight map to be

$$\alpha_{i1}^r = \max(\exp(-\frac{\mathbf{p}_i^2}{\sigma_{r0}^2}), \exp(-\frac{c_m^i - 1}{\sigma_{r1}^2})), \tag{22}$$

where $\alpha_{i0}^r$ is constant across the whole image. Both $\alpha_{i0}^r$ and $\alpha_{i1}^r$ can be learned through back propagating the gradient. We set $\sigma_{r0} = 0.5$ and $\sigma_{r1} = 0.2$.
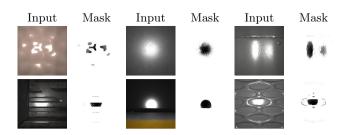


**Fig. 20.** The spatially varying weight $\alpha_i^d$ for the DCRF of diffuse albedo prediction.

*Hyperparameters for Training And Inference* In order to increase the capacity of the DCRF model, we learn different sets of BRDF parameters for each type of material. During both inference and training time, we average the DCRF coefficients according to the output of our material classifier. Let $\{\theta_i\} = \{\{\alpha_i\}, \{\beta_i\}\}$

be the DCRF coefficients for one material. To enhance the robustness of our method, we re-parameterize the coefficients as

$$\bar{\theta}_i = \frac{\theta_i}{\sum_i \theta_i}. \tag{23}$$

We clip the DCRF coefficients to always be positive. We use the Adam optimizer to optimize the coefficients. The learning rate is set to $2 \times 10^{-4}$ and we reduce it by half after every 2000 iterations. We adopt the method in [36] to train our DCRF model. The batch size is set to 32. We train the DCRF for diffuse albedo prediction over 4000 iterations and the DCRF for roughness and normal prediction over 3000 iterations. The standard deviations of Gaussian smooth kernels for the three DCRFs are shown in Table 5.

| Gaussian Kernels of DCRF for Diffuse Albedo | | | |
|---|---|---|---|
| | $\mathbf{p}_i$ | $\bar{\mathbf{I}}_i$ | $\mathbf{d}_i$ |
| $\kappa_1$ | 0.04 | - | - |
| $\kappa_2$ | 0.06 | 0.2 | - |
| $\kappa_3$ | 0.06 | - | 0.1 |

| Gaussian Kernels of DCRF for Normal Map | | |
|---|---|---|
| | $\mathbf{p}_i$ | $\Delta\mathbf{d}_i$ |
| $\kappa_1$ | 0.03 | - |
| $\kappa_2$ | 0.06 | 0.1 |

| Gaussian Kernels of DCRF for Roughness Map | | |
|---|---|---|
| | $\mathbf{p}_i$ | $\mathbf{d}_i$ |
| $\kappa_1$ | 0.04 | - |
| $\kappa_2$ | 0.06 | 0.2 |

**Table 5.** Standard deviations of the Gaussian smoothing kernels of the DCRFs for diffuse albedo, normal and roughness prediction.

# E    Details of Data Augmentation

In experiments, besides rotating and cropping the original high resolution spatially-varying materials, another important data augmentation is to scale the BRDF parameters for each patch before rendering them into images. For diffuse albedo, we uniformly sample scale coefficients in the range 0.8 to 1.4. For normal map, we sample the scale coefficients in the same way, apply the coefficients to the $x$

and $y$ components, then normalize the normal vector to be of unit length. For roughness, we sample the scale coefficients from a Gaussian distribution centered at 1, with standard deviation equal to 0.2. Empirically, we observe that such data augmentation can greatly improve the generalization ability of the network. For example, simply scaling the roughness parameter for each patch decreases the validation error for roughness prediction by 15%.