Eurographics Symposium on Rendering 2019
T. Boubekeur and P. Sen
(Guest Editors)

Volume 38 (2019), Number 4

Learned Fitting of Spatially Varying BRDFs

S. Merzbach[1], M. Hermann[2], M. Rump[1,2] and R. Klein[1]

[1]University of Bonn, Institute of Computer Science II, Germany
[2]X-Rite Inc.

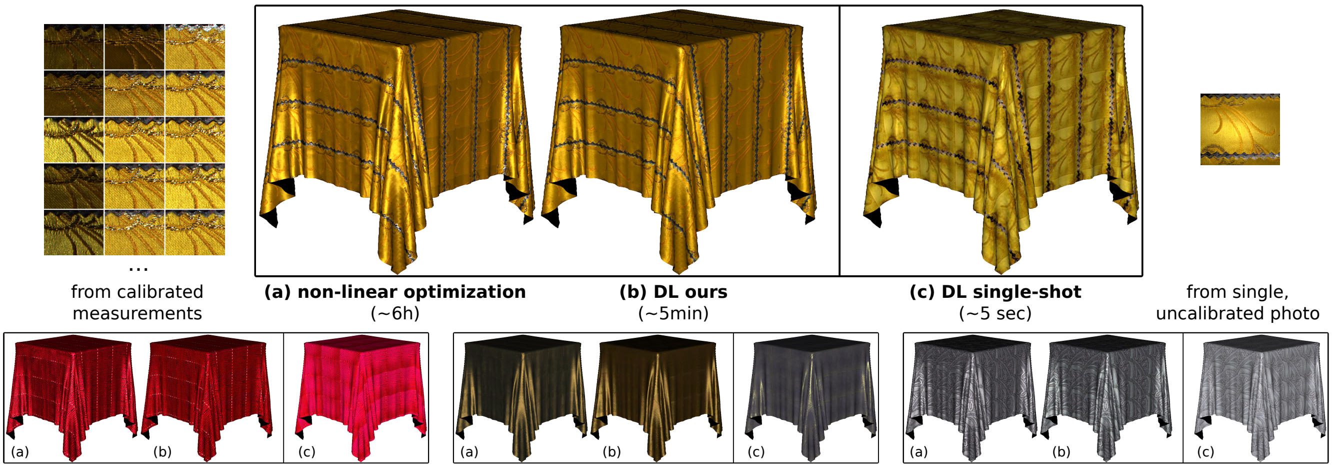
...
from calibrated measurements
(a) non-linear optimization
(~6h)
(b) DL ours
(~5min)
(c) DL single-shot
(~5 sec)
from single, uncalibrated photo
(a)
(b)
(c)
(a)
(b)
(c)
(a)
(b)
(c)

Figure 1: *High quality SVBRDF fitting via **(a)** non-linear optimization vs. **(b)** our deep learning approach on calibrated measurements vs. **(c)** a state-of-the-art single-shot SVBRDF estimation based on a single, uncalibrated photograph [DAD*18]. Though the single-shot method's result obtained from a single photograph are impressive, the limitations of such approaches are obvious. Conventional non-linear optimization provides high quality results, but at the price of extreme running times of several hours per material. Our method is trained on a large scale fabric database and produces results of comparable high quality at much shorter running times.*

Abstract

The use of spatially varying reflectance models (SVBRDF) is the state of the art in physically based rendering and the ultimate goal is to acquire them from real world samples. Recently several promising deep learning approaches have emerged that create such models from a few uncalibrated photos, after being trained on synthetic SVBRDF datasets. While the achieved results are already very impressive, the reconstruction accuracy that is achieved by these approaches is still far from that of specialized devices. On the other hand, fitting SVBRDF parameter maps to the gibabytes of calibrated HDR images per material acquired by state of the art high quality material scanners takes on the order of several hours for realistic spatial resolutions. In this paper, we present a first deep learning approach that is capable of producing SVBRDF parameter maps more than two orders of magnitude faster than state of the art approaches, while still providing results of equal quality and generalizing to new materials unseen during the training. This is made possible by training our network on a large-scale database of material scans that we have gathered with a commercially available SVBRDF scanner. In particular, we train a convolutional neural network to map calibrated input images to the 13 parameter maps of an anisotropic Ward BRDF, modified to account for Fresnel reflections, and evaluate the results by comparing the measured images against re-renderings from our SVBRDF predictions. The novel approach is extensively validated on real world data taken from our material database, which we make publicly available under `https://cg.cs.uni-bonn.de/svbrdfs/`.

CCS Concepts

• ***Computing methodologies*** → ***Reflectance modeling;*** *Neural networks;*

1. Introduction

Realistic simulation of surface appearance requires reflectance models rich in detail. Unlike simple textures, which nowadays can easily be captured with handheld smartphone cameras, reflectance models encode the dependence of observed colors on illumination and viewing directions. The conventional approach of relying on artists for creating such models is costly and often leads to results that look obviously synthetic. Furthermore, the produced reflectance models are not easily reusable because they do not generalize to arbitrary illumination conditions, since artists often tweak the material appearance for a certain scene by adding physically unplausible lighting conditions. An alternative that has emerged over the course of the last two decades is to obtain appearance models from measurements of real-world samples. This is done by capturing the surface reflectance under varying viewing and illumination directions. As opposed to artistic hand-crafting, this can be done in an automated and cost efficient manner, and the resulting reflectance models *do* generalize to arbitrary scenes and lighting conditions.

For easy integration into rendering frameworks, the reflectance measurements are usually encoded in simple, parameterized models, the most widespread one being the Bidirectional Reflectance Distribution Function (BRDF). Their number of parameters is not much larger than the RGB triplets in conventional textures. Examples for parameters are diffuse and specular albedos, as well as parameters describing surface roughness and therefore glossiness. BRDF parameters can thus be stored in the pixels of textures, making them spatially varying (SVBRDFs).

SVBRDF parameters are estimated from measured data in a process called “fitting”, which is usually performed via non-linear optimization. This is a challenging problem, as it involves many hyperparameters, e.g. the choice of metric, parameter initializations and optimization method. For details, see e.g. Ngan et al.’s systematic survey about fitting different BRDF models to measured data. The BRDF parameters ultimately have to be estimated per pixel. For SVBRDFs this leads to millions of parameters for realistic spatial resolutions, which makes the fitting process computationally very demanding and time consuming. Processing times of several hours are inevitable, even in commercial and highly optimized solutions.

In this paper, we try to alleviate the fitting efforts by pre-training a neural network. This allows us to reduce the fitting times from up to 10 hours to a few minutes, without any significant losses regarding the quality of the obtained models.

Learning based methods have been previously applied in the field of appearance capture. In the last years a series of works that utilize deep learning models to extract reflectance from varying kinds of input images has been published [AAL16; LDPT17; YLD*18; KCW*18; LSC18; LXR*18; DAD*18; VCGL19]. The trend is to feed very sparsely measured inputs into a network, usually one or a few photographs, to obtain an estimate of the observed surface reflectance. This is obviously a notoriously ill-posed problem, considering conventional fitting approaches require an as dense as possible coverage of light and view directions, necessary to reliably capture reflectance lobe shapes and other effects like surface anisotropy or Fresnel reflections. Though the results obtained by few-shot methods are impressive, one cannot expect to achieve the high quality necessary for photorealistic reproductions. A good indication of this are the limited models utilized in all of these works. We provide a more detailed comparison in the related work section.

In our work, we choose to tackle a different problem. Instead of limiting our observations to a few images, we train a neural network on photographs that are captured in a calibrated setup with a dense angular sampling. By doing this, we aim to produce high-quality reflectance models without limitations with respect to surface geometry, texture, or the complexity of the reflectance characteristics like anisotropy or the Fresnel effect. For this, we do a thorough evaluation of the quality of our results and of the generalization capabilities of our trained model to new, unseen inputs.

Our method: We train a convolutional neural network (CNN) that receives input patches from a stack of radiometrically calibrated images and learns to map them to the parameters of an anisotropic BRDF model. We capture photographs with a commercially available reflectance acquisition device, the TAC7 manufactured by X-Rite [XRi18]. In several postprocessing steps, those photographs are converted to radiometrically calibrated high dynamic range (HDR) images. The same device projects structured light patterns onto the material sample to obtain a coarse surface geometry, which is represented as a heightmap. The pixels in the HDR images contain reflectance information in the form of apparent BRDFs (ABRDFs), a generalization of BRDFs containing global effects like interreflections, shadowing and subsurface scattering. The ABRDFs are projected onto the pixels of the heightmap geometry and rectified into a common reference frame. In the remainder of the device’s postprocessing pipeline, these projected images are passed to a non-linear optimization, which estimates the model parameters in each pixel of the reference frame. This last step is exactly what we try to learn with our network.

Dataset: We capture a dataset of 378 fabric samples, which we all pass through the described postprocessing pipeline. We thus obtain 378 pairs of calibrated input images and their corresponding SVBRDF fits. The BRDF model fit by the software is described in detail in Section 3. We chose to limit our acquisition to the class of fabrics as surface materials, as we consider this class to provide the widest range of variability, reaching from very fuzzy and diffuse materials of all colors to almost metal-like reflections for special effect fabrics. Though our model is only trained on fabrics, our method is by no means limited to this category of surface reflectance. Given a large enough training set it can be applied to arbitrary classes of materials.

Evaluation: For our evaluation we can directly use the measured photographs, as they capture the surface appearance under various light and viewing conditions. We use the calibration information of the TAC7 device to re-render our regressed SVBRDF model under exactly the same illumination and viewing directions. Furthermore, we compare against renderings of the fittings obtained by the commercial Pantora software [XRi19]. To the best of our knowledge, this represents the state of the art for high-quality material fits. Furthermore, we provide some comparisons against a state of the art few-shot method [DAD*18].

Contributions of this work can be summarized as follows:

- first application of deep learning models to calibrated, dense measurements for high quality SVBRDF fitting
- evaluation against state of the art, commercial SVBRDF fitting method
- public release of our large scale SVBRDF dataset of 378 fabrics

2. Related Work

In this section we give an overview of existing SVBRDF fitting methods. We coarsely group related work into “traditional” approaches that rely on dense inputs and use non-linear optimization or similar methods to perform the fitting. Such methods resemble the pipeline we use to acquire our material database, consisting of acquisitions with the commercial TAC7 appearance scanner [XRi18; KRFS18] and SVBRDF fits by the accompanying Pantora software [XRi19]. Next, we present methods that work with sparser measurements. The more recent ones of those make use of deep learning models to solve the fitting problem, making them closely related to our own approach.

Conventional methods: The first systematic survey of fitting different BRDF models to measured data of homogenous surfaces was performed by Ngan et al. [NDM05]. Based on the MERL BRDF dataset measured by Matusik et al. [Mat03], augmented with their own measurement of anisotropic BRDFs, they perform constrained non-linear optimization via Sequential Quadratic Programming and examine which BRDF models are suitable for what kind of material types. Their employed non-linear optimization is prone to local optima and therefore requires good initial parameter guesses or repeated random initializations. It is possible to avoid the potentially only locally optimal results obtained by conventional non-linear optimization methods. Yu et al. employ a branch and bound scheme via interval arithmetics to find solutions for the Cook-Torrance BRDF model that are guaranteed optimal under the $\mathcal{L}_2$ metric [YSL10] or the $\mathcal{L}_1$ metric [YL16]. However, these works require involved derivations for new BRDF models, and more importantly, are only practically applicable to homogenous reflectance due to drastically longer running times compared to non-linear optimization.

One of the earliest works that tackles fitting of spatially varying reflectance is the method by Lensch et al. [LKG*03]. Though they only capture 15-25 HDR photographs, they group the observed samples into clusters of similar appearance to augment the angular sampling, which makes a non-linear optimization more stable. They then re-assign all pixels to new clusters, depending on the reconstruction errors of average BRDF fit in the previous cluster and repeat the fitting. Finally, they estimate a normal map. The rather small number of input images requires the surface to be sufficiently homogenous so that the clusters contain enough angular samples. Similarly, Wang et al. [WZT*08] cluster discrete $32 \times 32 \times 3$ normal distribution functions and can even use a single view capture. Dong et al. [DWT*10] capture representative BRDFs via a hand-held device and perform a non-linear optimization similar to Ngan et al. [NDM05]. Additional key measurements captured for a fixed view under varying illumination spatially resolve the reflectance. Their final SVBRDF is composed as linear combination of their representative BRDFs in each pixel. Wu et al. [WDR11] represent Bidirectional Texture Functions with a much more compact sparse mixture of multiple SVBRDFs to allow for easy editing and faster rendering. They fit their models with a stagewise Lasso algorithm, which avoids the limitations to a fixed set of BRDF models and instabilities for high parameter numbers of nonlinear optimization. These approaches are comparable to the iterated fitting procedure in the commercial Pantora software [XRi19], which we use for obtaining our SVBRDF fits and describe in detail in Sec. 3.

As it is crucial part of reflectance measurements in the TAC7 scanner, we provide a brief list of approaches that use strip-like linear light sources for reflectance measurements. Gardner et al. [GTHD03] are the first to make use of a linear light source to get more reliable estimates of diffuse and specular albedos and lobe parameters. Their work was later extended by Ren et al. [RWS*11] to a hand-held linear light source, and by Chen et al. [CDP*14] for the acquisition of anisotropic reflectance.

In a quite different line of works Aittala et al. [AWL13] and Fichet et al. [FSH16] exploit the frequency domain to obtain reliably reflectance estimates from few samples. Similar to the approach of Aittala et al. which uses frequency domain illumination patterns, Kang et al. [KCW*18] use an autoencoder to learn arbitrary illumination patterns, allowing them to reduce acquisition times to a few seconds. Ren et al. [RWS*11], Riviere et al. [RPG16] and Albert et al. [ACGO18] avoid the complexity and acquisition effort of calibrated setups and instead estimate SVBRDFs from mobile phone video. While Ren et al. and Riviere et al. rely on calibrated BRDF charts or color checkers to account for semi- or uncontrolled illumination, Albert et al. only require fiducial markers and a constant flash illumination. Aittala et al. [AWL*15] extend such approaches even further to the reconstruction of stationary, texture-like reflectance from a single pair of flash and no-flash smartphone images.

Learning-based few-shot methods: Inspired by Vidaurre et al. [VCGL19] we list closely related, deep learning based (SV)BRDF estimation works and their relevant characteristics in Table 1. All these works are described in more detail in the following paragraphs. A work by Aittala et al. [AAL16] sparked a branch of new methods that solve the problem of material parameter estimation by relying on neural networks. In their work, Aittala et al. make use of neural style [GEB15] based differences between patches rendered with their current parameter estimates and the input image. They can backpropagate through this style loss to update their model parameters. This formulation allows relaxing the requirement of pixel-wise correspondences and enables comparison to arbitrary patches from the input image. It requires the input images to be of stationary texture-like character, though. Due to the rich nature of our training samples, our network has no such limitations. Georgoulis et al. [GRR*17] estimate reflectance maps from objects covered in homogeneous materials under natural environment illumination. They feed their reflectance map estimates into two other networks which decompose them to the parameters of the Phong [Pho75] BRDF model and an environment map. Kim et al. [KGT*17] estimate parameters for the isotropic Ward model [War*92] from sparse RGBD observations under uncontrolled illumination and real-time constraints. For this purpose they introduce two alternative neural network architectures, HemiCNN and Grouplet, which both estimate the 7 BRDF parameters (diffuse & specular albedo & roughness) homogenosly on the surface of an object. This problem is rather orthogonal to the one solved in this work, as the authors make some necessary simplifications for solving this very difficult task under real-time requirements, whereas our work focuses on accelerating the complex reflectance fitting pipeline

reference	BRDF model	$\mathbf{n}_s$	$\mathbf{a}_d$	$\mathbf{a}_s$	σ	F_0	illum	rloss	comment
Aittala, ToG 2016, [AAL16]	Blinn [Bli77]	SV	SV	SV	SV	⊘	N	Y	stationary textures
Georgoulis, PAMI 2017, [GRR*17]	Phong [Pho75]	SV	HO	HO	HO	⊘	Y	N	homogeneous
Yu, CVPR 2017, [YS17]	Nielsen [NJR15]	(⊘)	(⊘)	(⊘)	(⊘)	(⊘)	N	Y	homog., statistical BRDF
Liu, ICCV 2017, [LCY*17]	DSBRDF [Nis09]	SV	(⊘)	(⊘)	(⊘)	(⊘)	Y	Y	homog., statistical BRDF
Kim, ICCV 2017, [KGT*17]	isot. Ward [War*92]	(⊘)	HO	(HO)	HO	⊘	N	N†	homog., white $\mathbf{a}_s$ [PFG00]
Meka, CVPR 2018, [MMZ*18]	Blinn [Bli77]	SV	HO	(HO)	(HO)	⊘	(Y)	Y	homog., RGB-D inp., white $\mathbf{a}_s$
Li/Ye, ToG17/CGF18, [LDPT17; YLD*18]	isot. Ward [War*92]	SV	SV	HO	HO	⊘	N	Y	self-augmentation
Li, ECCV 2018, [LSC18]	Unreal [KG13]	SV	SV	⊘	SV	⊘††	N	Y	DCRFs
Li, SIGG. ASIA 2018, [LXR*18]	Unreal [KG13]	SV	SV	⊘	SV	⊘††	Y	Y	depth, SH env., GI-layer
Deschaintre, ToG 2018, [DAD*18]	Cook-Torrance [CT82]	SV	SV	SV	SV	⊘	N	Y	global features track
Vidaurre, WACV 2019, [VCGL19]	Ashikhmin [AS00]	⊘	(HO)	HO	HO (aniso.)	(HO)	N	N	homogeneous
ours	anisot. Ward [GD10]	SV	SV	SV	SV (aniso.)	(SV)†††	N	N	calibrated inputs

Table 1: ***Overview of related work:*** *we list important characteristics of the literature that is closest related to our method, i.e. deep learning based (SV)BRDF estimation. Columns indicate if parameters are estimated in a spatially varying (SV) or a spatially homogeneous (HO) manner, or if they are absent from the model (⊘). The* ***rloss*** *column indicates if the network is trained using a rendering loss,* ***illum*** *whether the illumination is estimated as well.* ***Parameters:*** $\mathbf{n}_s$*: shading normal,* $\mathbf{a}_d$*: diffuse albedo,* $\mathbf{a}_s$*: specular albedo,* σ*: roughness,* F_0*: Fresnel reflectance at* 0° *inclination.* ***Notes:*** † *Kim et al. do not use a conventional rendering loss, but their* L_c *term ensures color constancy w.r.t. scene illumination;* †† *Li et al. do not estimate the Fresnel* F_0 *parameter, but assign a fixed value depending on the material type (metal / dielectric);* ††† *though our training samples only feature a spatially homogeneous* F_0 *parameter, our network still allows it to vary spatially.*

without significant quality penalties. Liu et al. [LCY*17] and Yu et al. [YS17] both estimate parameters of spatially homogeneous statistical BRDF models, trained on the MERL BRDFs [Mat03]. Liu et al. fit 108 coefficients of Nishino's Directional Statistics BRDF [Nis09], along with spatially varying surface normals and a low resolution HDR environment map to allow for material editing. The network of Yu et al. is used to estimate 5 coefficients per color channel for the statistical BRDF introduced by Nielsen et al. [NJR15]. Li et al. [LDPT17] train a CNN for single-photo SVBRDF estimation. For training, they face the problem of a limited set of samples labeled with model parameters. They therefore bootstrap the model on a small training set and later rely on self-augmentation via intermediate network predictions, which they can use to generate new labeled training samples due to the generative nature of SVBRDFs. Similar to our model, they train the network to estimate parameters for the Ward BRDF, however, only in its isotropic variant and without a Fresnel term. Most importantly, their SVBRDF specular component is completely homogenous over the surface, whereas we predict *all* parameters per pixel. In their estimates, only diffuse albedo and normal map are allowed to vary accross the surface. Their network model works under uncontrolled illumination, which they achieve by rendering synthetic training samples from a database of SVBRDFs and light probes. Ye et al. [YLD*18] extend this work by completely removing the need for labeled training samples. They further investigate the self-augmentation process and conclude that the intermediate network only serves to resolve ambiguities in the ill-posed single image reflectance estimation problem. They therefore train their initial network on randomly generated synthetic training samples that do not even need to resemble the target distribution of a specific material class. Another work [LSC18] presents a network that combines a single encoder, a classifier and multiple decoders for different SVBRDF parameter maps with a rendering loss layer. To ensure a high visual quality and physical plausibility of the estimated parameters, they furthermore add densely-connected conditional random fields (DCRFs) before their loss layer, which they train end-to-end. The DCRFs refine the decoder outputs by enforcing smoothness priors. Their training is based on 588 high-resolution measured or artist-created SVBRDFs from the commercial large-scale Adobe stock material database [Ado19]. Similar to previous works, their microfacet-based SVBRDF model [KG13] only allows the diffuse albedo, the shading normal and the roughness to vary spatially. The same BRDF model is used in a work by Li et al. [LXR*18]. While previous works for spatially varying reflectance estimation were limited to near planar samples, Li et al. develop cascaded multi-branch CNNs to iteratively refine material, environment, global illumination effects and depth from images of arbitrary shapes. Though they achieve impressive results for separating all these coupled effects, their Cook-Torrance-like SVBRDF lacks a specular albedo and the authors assign a fixed value for the Fresnel F_0 parameter. Deschaintre et al. [DAD*18] tackle the problem of limited datasets by creating new materials from permutations and combinations of procedural SVBRDF models obtained from Allegorithmic Substance share [all19]. They use their 200,000 generated SVBRDFs to train an encoder-decoder network, with an additional global feature track to bypass and fuse features that are removed by instance normalizations in their main network. This is necessary to maintain global color information until the output layer of the decoder. To the best of our knowledge, in the line of deep-learning-based related works for single-shot spatially varying material estimation, after Aittala et al. [AAL16] (limted to stationary materials), Deschaintre et al. are the only ones allowing for spatially varying colored specular albedos in their Cook-Torrance SVBRDF model [CT82] with isotropic GGX distribution [WMLT07], making it the work most closely related to our own, which uses an anisotropic Ward BRDF with colored specular albedo and Fresnel term. In our evaluation (Sec. 6) we therefore compare against results obtained from their model to give insights about the qualitative differences between few shot approaches and our network based on calibrated dense inputs. Vidaurre et al. [VCGL19] derive spatially homogeneous parameters for a variant of the Ashikhmin-Shirly BRDF [AS00] from two input images, which show a material sample under two view inclination angles and next to a color checker. They train a two-branch CNN on a dataset synthetically generated in Maxwell Render [Nex19]. They pose the parameter estimation as a multi-task classification problem and train two dedicated CNNs to extract feature vectors from their rectified input images, which are white-balanced for one of the two network branches. The BRDF parameters are ultimately estimated by several nested fully connected layers which model interdependencies between the different model parameters. Though the degrees of freedom (anisotropy and Fresnel reflections) of their chosen BRDF model are on par with ours, in contrast to our work, they are limited to homogeneous materials, i.e. one set of ten parameters per pair of input images.

Learning-based BTF encoding: Recently, another deep learning based representation for spatially varying reflectance was presented [RJGW19]. Rainer et al. train a neural network per material to pixel-wise encode measured ABRDFs (i.e. the pixels of Bidirectional Texture Functions, BTFs), represented as N-dimensional vectors (N = 1508 or 22801), to a k-dimensional latent space (k = 8). At the same time, they train a decoder network that maps the latent representation, together with a pair of light and view direction, to an RGB triplet, which corresponds to the ABRDF observations for these directions. Their latent representation is similarly compact as an SVBRDF ($k = 8$ latent coefficients, mean & standard deviation per pixel) and even encodes non-local effects like shadowing and subsurface scattering. However, due to its fully connected layers the decoder network is much more costly to evaluate than our BRDF model. Furthermore, the latent coefficients lack the intuitive meaning of our SVBRDF parameters, which rules out easy material editing. The main difference between their and our approach is that our network is trained to generalize over an entire category of materials, whereas Rainer et al. have to train one network per material. Their proposed encoder network has similarities to our CNN. We detail the similarities and differences in Section 5.

As one can see in Table 1, our work is the first to estimate fully spatially varying parameters of a BRDF model that is complex enough to capture all important effects observed in measured materials. Though conventional optimization approaches already provide these capabilities (at the cost of extreme running times), our proposed solution combines this flexibility with the performance gains obtained from deep learning. With our method we thus bridge the gap between conventional optimization-based fitting and few-shot reflectance estimation methods.

3. Data Acquisition and Processing

In this section we describe the acquisition and properties of the fabrics we use for training our model, as described in Sec. 5.

Acquisition: The Total Appearance Capture (TAC7) [XRi18; KRFS18] device is a commercially produced appearance scanner. It is controlled by the Pantora software [XRi19], which is also used for postprocessing and optional editing of the resulting SVBRDFs. The TAC7 consists of a hemisphere covered with 29 white LEDs and four monochrome cameras arranged on an arc above a turntable. Additionally, five of the LEDs are equipped with color filter wheels with 10 filters covering the visible spectrum. The cameras are sensitive to the entire visible spectrum and thus capture panchromatic images for white illumination. Color information is captured by tuning band-pass filters in front of the five LEDs with filter wheels. To capture an arbitrarily fine sampling of reflectance lobes, a strip-type light source (called linear light source) can be rotated through the space above the sample holder. A low-resolution surface geometry is captured with a structured light projector. Transparent materials can additionally be illuminated from below with a single LED, shining through a diffusor plate in the sample holder. Both the turntable rotations and the linear light source movement are crucial for capturing textiles. The fibers impose a strong anisotropic reflectance behavior in many fabrics, and can cause myriads of tiny highlights that only light up for very specific constellations of light and view directions. With a coarse hemispherical sampling such as the 29 LEDs, one would never be able to reliably capture all those effects.

For each measured material we obtain 100 color images, 348 monochrome point-lit images and, depending on the glossiness, 280, 560 or 3300 linear light source images. The dynamic range of the material reflectance is another important factor, as it influences the number of exposures necessary to capture the full dynamic range. Typical acquisition times of the TAC7 for anisotropic fabrics range from 30 minutes for low gloss to 60 minutes for high gloss materials. Please see our supplementary material for more details about the acquisition and postprocessing.

SVBRDF Fitting: After the raw measurement images have been postprocessed by the Pantora software (see the supplementary material for details), the inputs to the last fitting step are HDR-combined images projected on the surface geometry as observed from a reference camera (top view), along with the heightmap. Pantora's fitting procedure is based on clustering the surface into areas of similar appearance to increase the number of angular samples for a non-linear optimization per cluster. The optimization is divided into monochrome and color fitting steps. After obtaining a first estimate of the monochrome SVBRDF model parameters, the fitting procedure then updates the 3D geometry by photometrically estimating normals and refining the height values for each pixel. This process is iterated a few times, with pixel-wise refinements of the SVBRDF parameters, given the new geometry information, until convergence is reached. Finally, the so far monochrome albedos are "colorized" by first estimating the specular albedo, which is then subtracted from the observations to estimate the diffuse albedo. For further details about the individual fitting steps, please see Kohlbrenner et al. [KRFS18]. In our supplementary material we give a detailed description about the SVBRDF parameters obtained from the fitting.

SVBRDF model: When selecting the textile preset, the commercial Pantora software fits the parameters of the Geisler-Moroder [GD10] variant of the Ward BRDF [War*92], modified by Schlick's Fresnel approximation term [Sch94]. The model parameters are the shading normal $\mathbf{n}_s \in \mathbb{R}^3$, the diffuse albedo $\mathbf{a}_d \in \mathbb{R}^3$, the specular albedo $\mathbf{a}_s \in \mathbb{R}^3$, the lobe roughness parameters $\sigma_x, \sigma_y \in \mathbb{R}$, the anisotropy angle $\alpha \in \mathbb{R}$ and the 0° inclination reflection coefficient $F_0 \in \mathbb{R}$. We give a detailed description of the model and its parameters in our supplementary material to alleviate usage of our SVBRDF database. Fig. 2 shows some exemplary parameter maps for one of our materials.

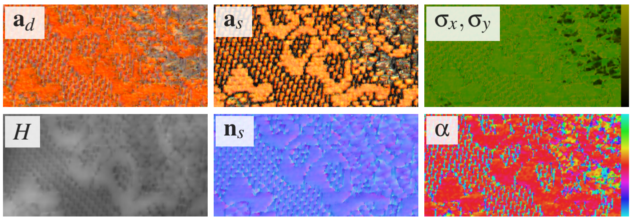

Figure 2: *Parameter maps for an example anisotropic Ward SVBRDF from our dataset. Diffuse ($\mathbf{a}_d$) and specular ($\mathbf{a}_s$) albedos are represented in sRGB, the roughness parameters σ_x, σ_y are color coded in the red and green channel of their map, height and normals are mapped to $[0,1]$, the anisotropy angle α is coded with a hue colormap and ranges from $-\pi/2$ to $\pi/2$.*

4. Material Database

We collected 378 different fabric samples, covering many types of textiles (cotton, brocade, duchesse, Lycra, polyester, satin, silk, velvet and others). There are both homogenously colored materials, as well as samples with patterns. Most of the measured samples have rather small dimensions, e.g. around 3cm × 10cm. The average physical sample area is around 20cm^2, with a corresponding average side length of 630 pixels. We acquired as many materials in parallel as we could fit in the sample holder, which has a diameter of about 14cm. Not only did this reduce the effective acquisition times per material, it also prevented unnecessary storage overhead caused by the unprocessed raw images. For materials with patterns on a larger scale we tried to use the full sample holder area when possible. For parallel measurements we tried to group materials by their glossiness to ensure that the optimal measurement strategy is applied to each sample. There were in total 141 low, 173 medium and 64 high gloss materials. The postprocessing was performed on a per-material basis. We chose to fit a uniform Fresnel F_0 for all materials, as it provides more robust results and saves some processing time.

The most time-consuming part of the postprocessing, the SVBRDF fitting, took on average about 30ms per pixel on an i7 5820K CPU with 32GB RAM and an NVIDIA GTX 1080 GPU. Given the in total more than 160 million pixels in the 378 fabrics in our dataset, this amounts to a total fitting time of 56 days! We preprocess the different input modalities (monochrome & color images from Pantora) and store them as losslessly compressed OpenEXR files. The labels of our dataset, the Pantora SVBRDF fit, are stored in the proprietary appearance exchange format (AxF) [ML15]. The total storage space is 337 GB for the input images and 6.5 GB for the material AxFs. Our dataset is available under `https://cg.cs.uni-bonn.de/svbrdfs/`.

5. Model

In the following paragraphs we explain in detail how we construct and train our convolutional neural network (CNN).

Inputs: We train our model on all types of materials at once, i.e. on multiple different glossiness levels. The number of linear light source images is the only one that depends on the selected material glossiness preset. As our neural network requires each sample to have the same number of channels, before passing them into the network, we downsample the linear light source images to the lowest angular sampling, i.e. 280 for low gloss materials. Though in principle one could estimate each pixel's parameters independently, we provide some spatial neighborhood for each pixel. Together with the heightmap the network can thus easier learn to recognize shadows and ignore them when estimating the parameters from the remaining inputs. Our inputs are patches with 15×15 pixels, and in total 934 channels ($3+3+3 \times 100+348+280$ for per-pixel x-, y- & z-coordinates, geometric normals, color, monochrome point lit and linear light source images). The corresponding labels are $1 \times 1 \times 14$ dimensional, as we train the network to map the input patches to a single pixel. To facilitate the learning, we map the lobe parameters similar to Deschaintre et al. [DAD*18] to

$$\sigma'_{\{x,y\}} = \frac{\log(\sigma_{\{x,y\}} + 0.01) - \log(0.01)}{\log(1.01) - \log(0.01)}.$$

The reason is the strong non-linear effect they have on the BRDF model. The sharpest and brightest highlights occur for very small $\sigma_{\{x,y\}}$. We therefore spread this range by applying the logarithm and mapping back to $[0,1]$. Similarly, we avoid the wraparound of the anisotropy angle by converting it to a 2D cartesian vector, spreading it over two channels: $\alpha \mapsto \{\cos(2\alpha), \sin(2\alpha)\}$. Our labels $\{\mathbf{a}_d, \mathbf{a}_s, \mathbf{n}_s, \sigma'_x, \sigma'_y, \cos(2\alpha), \sin(2\alpha), F_0\}$ are thus 14-dimensional vectors. We randomly split our dataset of 378 fabrics into 311 training and 67 test materials.

Architecture: Our first architecture was a pixel-wise multi-layer perceptron. However, we observed artifacts in the form of underestimated albedos and tilted normals in shadowed input-pixels, which we show in our supplementary material. A network can only learn to deal with these areas if it can see a local neighborhood (i.e. patches) along with the heightmap. If one considered the TAC7 manufacturer's requirements for surface flatness (at most ± 3mm height variation), and the lowest light inclination angle (around 70 degrees), one can derive a theoretical lower bound of the necessary patch size to ensure that all potential occluder sources for the patch's center pixel are fully contained inside the patch and that the network can learn to recognize shadows. One can thus derive a necessary patch side length of at least 8mm ($\approx$240 pixels). However, in our selection of fabrics the fewest pixels stick out further than 0.5mm, which would only require a side length > 40 pixels. Besides that, it is in fact not necessary to enclose the entire cast shadow in the patch for the network to be able to recognize it. The patch size is a tradeoff between a loss of performance speed (especially training effort) and inclusion of a large enough neighborhood. We were able to remove the artifacts by changing our inputs to patches and our architecture to a convolutional neural network. Our experiments show that patches of size 15×15 pixels provide good results. The main challenge in designing our network is the large dimensionality of the inputs to our network. Normally, CNNs reduce spatial resolution by strided convolutions or pooling layers, and store the extracted features in the channel dimension of subsequent layers. If we applied this standard design, we would quickly run into memory problems, as our input is already very high-dimensional. We could of course apply dimensionality reduction to our 934-channel input patches. Den Brok et al. [dBWK15] showed that ABRDFs can be well represented in linear bases com-

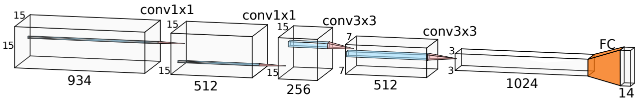

Figure 3: *Overview of our network architecture. The first two layers perform a dimensionality reduction, 3×3 convolutions with stride 2 start in the third layer. non-linearities throughout the network are leaky ReLUs. We do not use Batch Normalization and instead rely on dropout (rate 0.1) to prevent overfitting.*

puted over several materials of the same class. In some experiments we observe similar trends on our inputs. Recent works have shown that neural networks are well suited for reducing high-dimensional inputs to a compact representation [RJGW19] and outperform conventional linear bases for reflectance data by a significant factor. This is why we follow a similar scheme for reducing the high dimensionality of our inputs in the first two layers of our network. This strategy also saves us from the additional preprocessing step of computing a common linear basis on our materials and projecting all inputs into this basis. Our inputs are patches, so to limit network complexity, we apply 1×1 convolutions in the first two layers to reduce the input dimensionality without losing spatial resolution. The 1×1 convolutions consecutively map the 934 channels to 512 and 256 feature dimensions. Rainer et al. have to deal with much higher input dimensions of 22801 for the Bonn BTF dataset [WGK14], so they apply 1D convolutions on their input vectors. The subsequent layers in our model all perform 3×3 convolutions with a stride of 2, until the spatial dimensions reach 3×3 pixels and a feature dimension of 1024. We show an overview of our network's layer dimensions and convolution parameters in Fig. 3. The output activations of the convolution layers are passed through rectified linear units (leaky ReLUs), with a leak factor of 0.01. Similar to the last layer in Rainer's Neural BTF encoder, which maps to a latent space, our final layer is fully connected and maps all elements of the last $3 \times 3 \times 1024$ tensor to the 14-dimensional BRDF parameter space. We feed ground truth BRDF parameter labels and our network's predictions to a loss layer that computes the robust Huber loss [Hub64] with parameter $\delta = 1$. We weight the differences between labels and predictions by multiplicative instance weights, which are all 1 except for $w_{\mathbf{a}_d} = 20, w_{\mathbf{a}_s} = 10$.

Optimization: We use the Adam optimizer [KB14] with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-8}$) and a learning rate of 10^{-5}, which we progressively lower down to 10^{-6} over the iterations. Our inputs are grouped in mini-batches of 64, and we train for 768,000 iterations. Though we could theoretically sample as many input patches as we have pixels in our training set, i.e. about 130 million, we still have to prevent overfitting, as those patches come from only 311 different material samples in our training split. Though it has become a de facto standard for regularization, we found it difficult to train our network with Batch Normalization [IS15]. The inputs' mean intensity is lost due to the normalization, which is problematic for reliably predicting radiometrically meaningful parameters. Deschaintre et al. made similar observations for their U-Net like network. To solve this problem they introduced a global feature track which bypasses the batch statistics in a fully connected branch and fuses them into the encoder-decoder network. We chose a simpler solution and only rely on weight decay (5×10^{-4}) and dropout (rate 0.02) for regularization.

6. Evaluation

We test our model performance in three ways. First, we can directly compare our model's estimated parameters to the ground truth labels obtained from the Pantora fits. Second, we can evaluate the models obtained from Pantora and our network to compute image-based metrics between the resulting renderings. Third, we have the calibrated images of the measured material to evaluate the quality of our predictions and compare them to those of Pantora. The upper graph in Fig. 4 shows normalized RMS errors for the individual model parameters, averaged over all materials in our test split. The lower plots in Fig. 4 provide $\mathcal{L}_1$-errors computed on all color images between re-renderings of the Pantora fit and our estimated model, as well as the respective deviations between Pantora or our re-renderings and the measured photographs, for each of the 67 materials in our test set. In Fig. 5 we show a visual comparison of parameter maps by Pantora and our method on representative materials from our test set. Fig. 6 shows re-renderings of the same materials based on fits by Pantora and our method in comparison to the measured ground truth images. Additional error metrics and visualizations can be found in our supplementary material.

Our observations are as follows: first of all, the absolute $\mathcal{L}_1$ errors between re-renderings of our network and the Pantora fit are very low for the largest part of our test set, with only a few outliers at the right hand side of the lower plot of Fig. 4. This shows that our network is able to learn a mapping from the inputs to the ground truth parameter maps produced by Pantora. Second, the errors between re-renderings of Pantora to the ground truth measurements are with very few exceptions on par with ours, showing that our network produces meaningful combinations of parameters. One should bear in mind that our network – contrary to the Pantora fits – is able to estimate the Fresnel F_0 parameter in a spatially varying manner. Each of our training materials only feature a spatially homogeneous F_0 parameter. One cannot expect the network to magically learn a sophisticated prediction of this parameter. However, between the different materials there are variations of F_0, and thus there are different training samples which the network – to some extent – can learn from. There are still obvious ambiguities, causing the network to shift energy between $\mathbf{a}_s$ and F_0, as can be seen in the results in Fig. 5. Our explanation is that there are still too many contradicting training samples, where the enforced homogeneous F_0 "confuses" the network during training. At the same time, the quality of our re-renderings in Fig. 6 generally matches that of the Pantora fits, while the absence of significant outliers speaks for the robustness of our approach. Since Pantora heavily relies on heuristics for many steps in the fitting procedure, the obtained results may exhibit imperfections as well. We conclude that the results in Fig. 6 nicely illustrate that our model has learned to generalize from the "good" samples in our training set to compensate for the shortcomings of the heuristic-based non-linear optimization in Pantora.

6.1. Comparison against Single-Shot Method

Finally, we conducted a comparison against the method of Deschaintre et al. [DAD*18]. We want to stress that this comparison

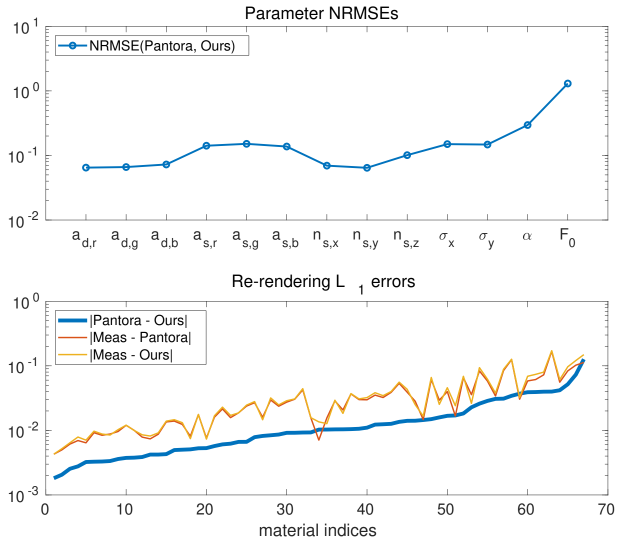

Figure 4: *Numeric errors of our method computed over the full set of color images for all 67 materials in our test set:* ***top:*** *averaged normalized RMS errors for each of the model parameters.* ***below:*** *$\mathcal{L}_1$ errors computed between measured images and re-renderings of our / Pantora's fits. The graph is sorted by the error between our re-renderings and those of Pantora.*

is between two methods with completely different inputs and that we are well aware that our method gains a lot more information than that of Deschaintre et al., which only takes a single *uncalibrated* photograph captured with a smartphone close to the surface illuminated by the camera flash. Furthermore, Deschaintre et al. state in their work that their network is not trained for estimating anisotropic, complex materials like fabrics. Given these facts, the presented results of their network are quite impressive. Our intention for adding this comparison is not to devalue the authors' work, but to re-emphasize the limitations of single-shot approaches in general. For the evaluation we use the Pantora SVBRDF to simulate images comparable to Deschaintre's inputs by taking a small crop of each material in our test set and placing the virtual smartphone camera head-on at a distance of a few centimeters. Deschaintre's network is trained on patches with 256×256 resolution, and is limited by its encoder-decoder structure to this resolution, so we crop our materials to this size without downsampling. This allows a direct comparison between Pantora's fit, our and Deschaintre's estimates, as they have pixel-wise correspondences. The alternative would be to take overlapping 256×256 patches of our bigger samples and blend the reconstructions, but we consider our simple experiment expressive enough already. We tonemap our renderings with a constant scaling factor and a simple gamma correction and then clamp them to $[0, 1]$ to simulate the smartphone photograph. In Fig. 7 we show renderings with the same viewpoint but a changed illumination direction. To keep the comparison fair, we employ the same unrefined input heightmap for our and Deschaintre's renderings, whereas the Pantora materials are rendered with their refined heightmap. We provide further rendering comparisons on a bigger scale with Deschaintre et al. in our teaser, see Fig. 1.

Our test materials show various degrees of glossiness and anisotropy, as well as patterns of various scales. As the authors also mention themselves [DAD*18], it is obvious that their Cook-Torrance BRDF with isotropic GGX normal distribution cannot represent anisotropy or Fresnel effects, and trying to estimate those from a single photograph is inherently ill-posed. We further observe that the albedo maps seem to be systematically overestimated compared to the Pantora fit, where our results match much closer. Besides that, the central highlight from our single input photograph manifests itself as a dark spot in the shinier materials.

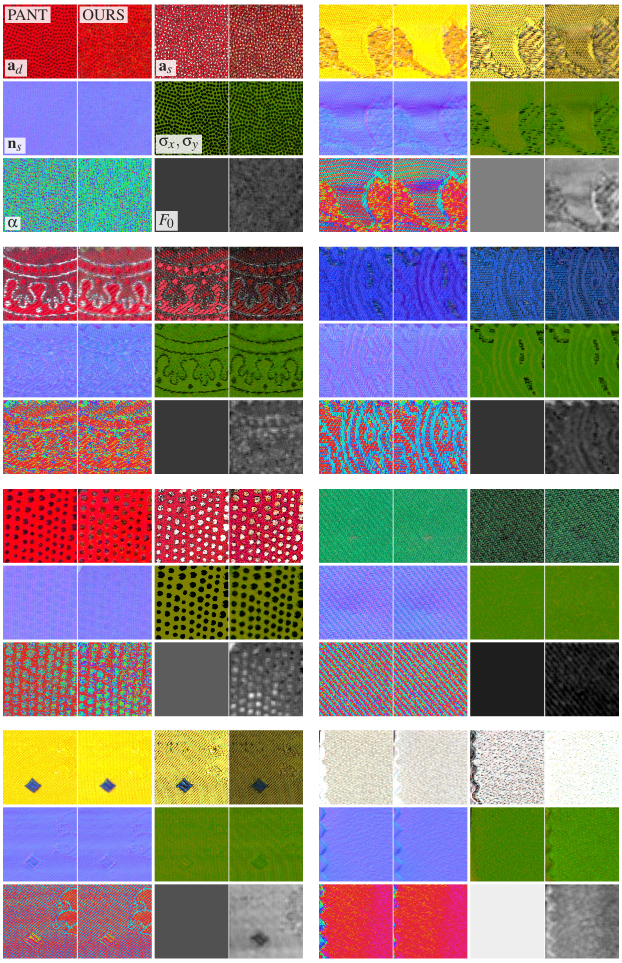

Figure 5: *Comparison between the Pantora SVBRDF (left,* ***PANT****) and our estimate (right,* ***OURS****). All eight materials are from our test set (67 in total). Parameter maps are diffuse albedo ($\mathbf{a}_d$), specular albedo ($\mathbf{a}_s$), shading normal ($\mathbf{n}_s$), anisotropic roughness parameters (σ_x, σ_y), anisotropy angle α and Fresnel reflectance at $0°$ inclination (F_0). Notice the overall good results; mismatching specular albedos are compensated by accordingly adjusted Fresnel F_0.*

6.2. Evaluation timings

After loading all input images into main memory, which occupies around 1.5GB for an average sample size, we evaluate our network by extracting 15×15 patches in a sliding window manner. We fit as many patches at once into the GPU memory as possible, and write

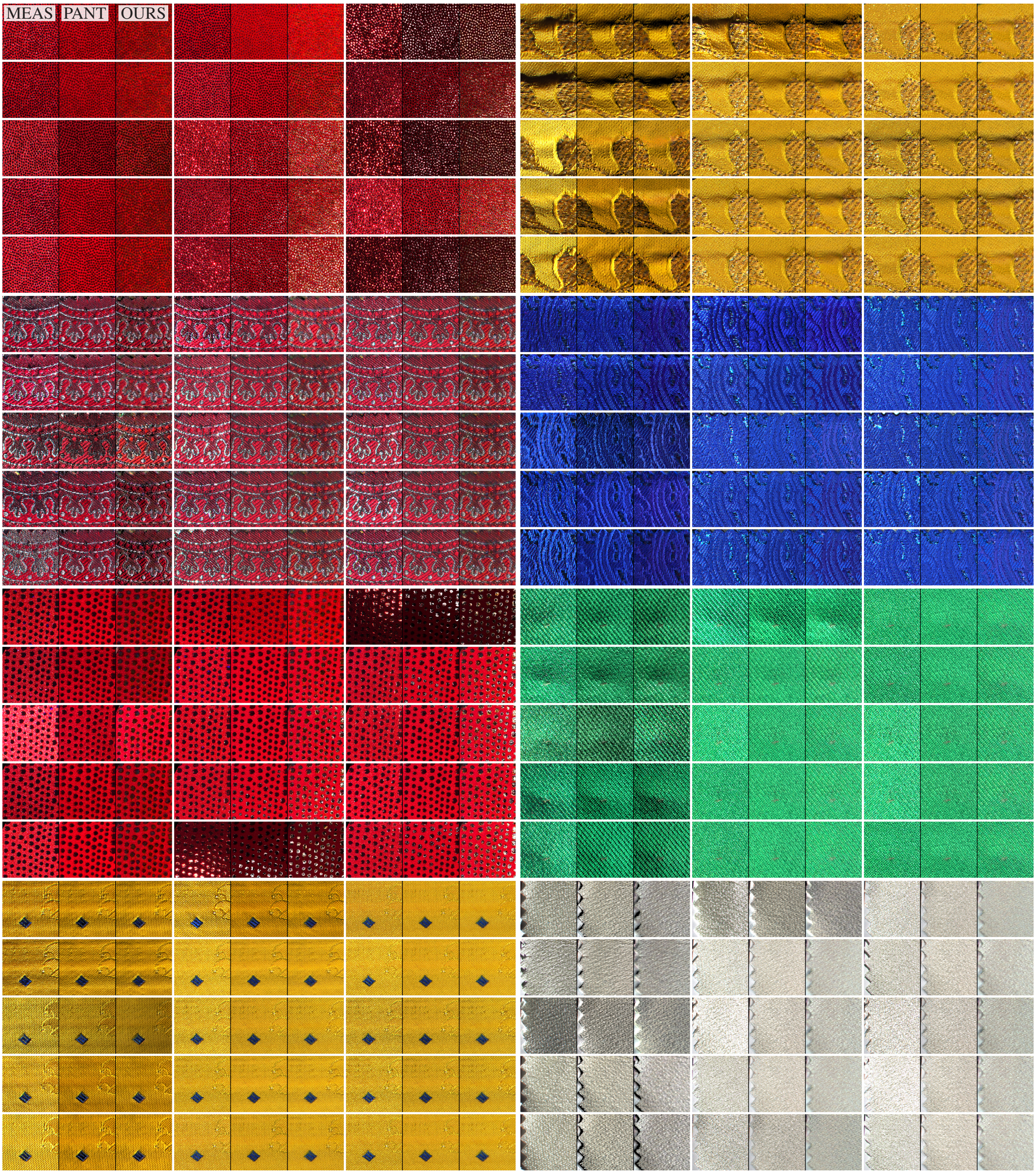

Figure 6: *Comparison of some materials from our test set. For each group of three images, the leftmost (**MEAS**) is a measured image (for some random light and view directions), the middle one is a rerendering of the Pantora SVBRDF (**PANT**), and the right one is a rerendering of our SVBRDF estimate (**OURS**).*

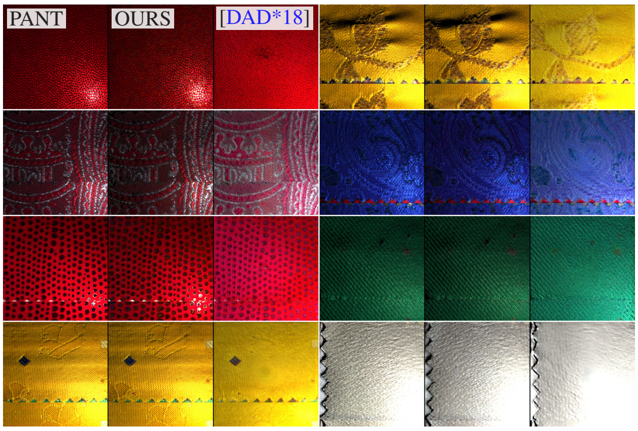

Figure 7: *Comparison between Pantora SVBRDF fit (**PANTORA**), our estimate (**OURS**) and that of Deschaintre et al. ([DAD*18]).*

the reconstructed 13-dimensional vectors into the parameter maps of the output SVBRDF. For an average-sized material, we obtain timings of about 3 minutes on an NVIDIA TITAN X Pascal GPU with 12 GB VRAM and about 15 minutes on an NVIDIA GTX 780 with 3 GB VRAM.

7. Limitations

Our model is not perfect and there are still some limitations in the form of some materials that are not estimated very well. In Fig. 8 we show materials where some of the parameter maps are mismatching the Pantora fit in a too strong way, so that it is directly visible when comparing renderings. This poor performance can partially be explained by the materials themselves and the already imperfect Pantora fit, as we observe a much stronger mismatch between the ground truth measurements and the re-renderings created from the Pantora SVBRDF; compare the differences in the respective first two columns in Figs. 6 and 9. However, our network model is rather simple and there are many ways to improve it.

We still believe that our results clearly show that our approach is following exactly the right direction for three reasons: First, few- and in particular single-shot approaches are inherently limited regarding the quality of the results that can be obtained. For high-quality reflectance estimation, it is necessary to provide rich calibrated inputs to the reconstruction method. Second, even industrial-grade state-of-the-art optimization-based fitting methods are limited because they heavily rely on heuristics and user selections, which do not generalize reliably over arbitrary surfaces, even within one class of materials. Third, conventional optimization has enormous running times that can reach up to the order of days. Our approach is the first step in a direction that solves all three of these problems. In the next section, we will lay out a list of improvements for potential future works.

8. Future Work

We want to outline some first steps to improve upon our initial approach in order to achieve better fitting quality. First of all, as Deschaintre et al. and several other related works [LSC18; DAD*18; LSC18; LXR*18] have shown, rendering-based losses provide superior results for reflectance data. The choice of metric in conventional optimization-based fitting has always been a problem, and various weighting schemes have been proposed over the years. A rendering loss elegantly solves this by both applying a suitable weighting on the loss computation, as well as directly backpropagating into the SVBRDF model parameter when it is implemented as a differentiable network layer. Second, during our training we so far do not use any kind of augmentation. Though we have more than enough input samples for our patch-based network, those patches are still drawn from a pool of only a few hundred materials. The conventional augmentation techniques like channel flipping or scaling are ruled out for our calibrated inputs. Though it would be applicable to our color image inputs, we lack the necessary spectral information to apply the same mapping for the bulk of our inputs that are only captured as monochrome images and therefore would break calibration, probably causing more problems than augmentation would solve during training. In a future work, we want to exploit the generative nature of SVBRDF models to augment our dataset, similar to the self-augmentation used in related works [LDPT17; YLD*18]. Another obvious extension is to train an encoder-decoder network like Deschaintre et al., though this will certainly require more data and sophisticated augmentation methods, as the number of larger non-overlapping patch-samples required for training such a network is certainly too limited on our current dataset.

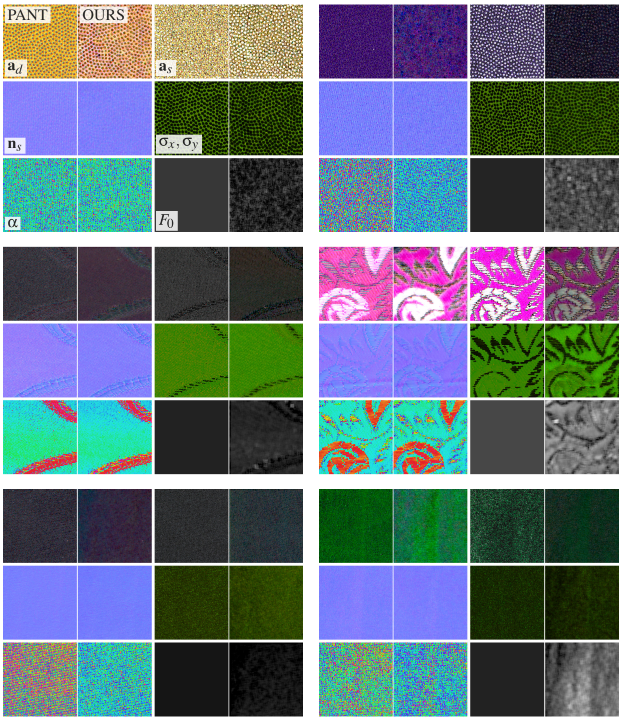

Figure 8: *Some failure cases, comparison between the Pantora SVBRDF (left, **PANT**) and our estimate (right, **OURS**). All six materials are from our test set (67 in total). The Parameters are described in Fig. 5. Most problems occur in the albedo maps. Mismatching specular albedos are only partially compensated by F_0.*

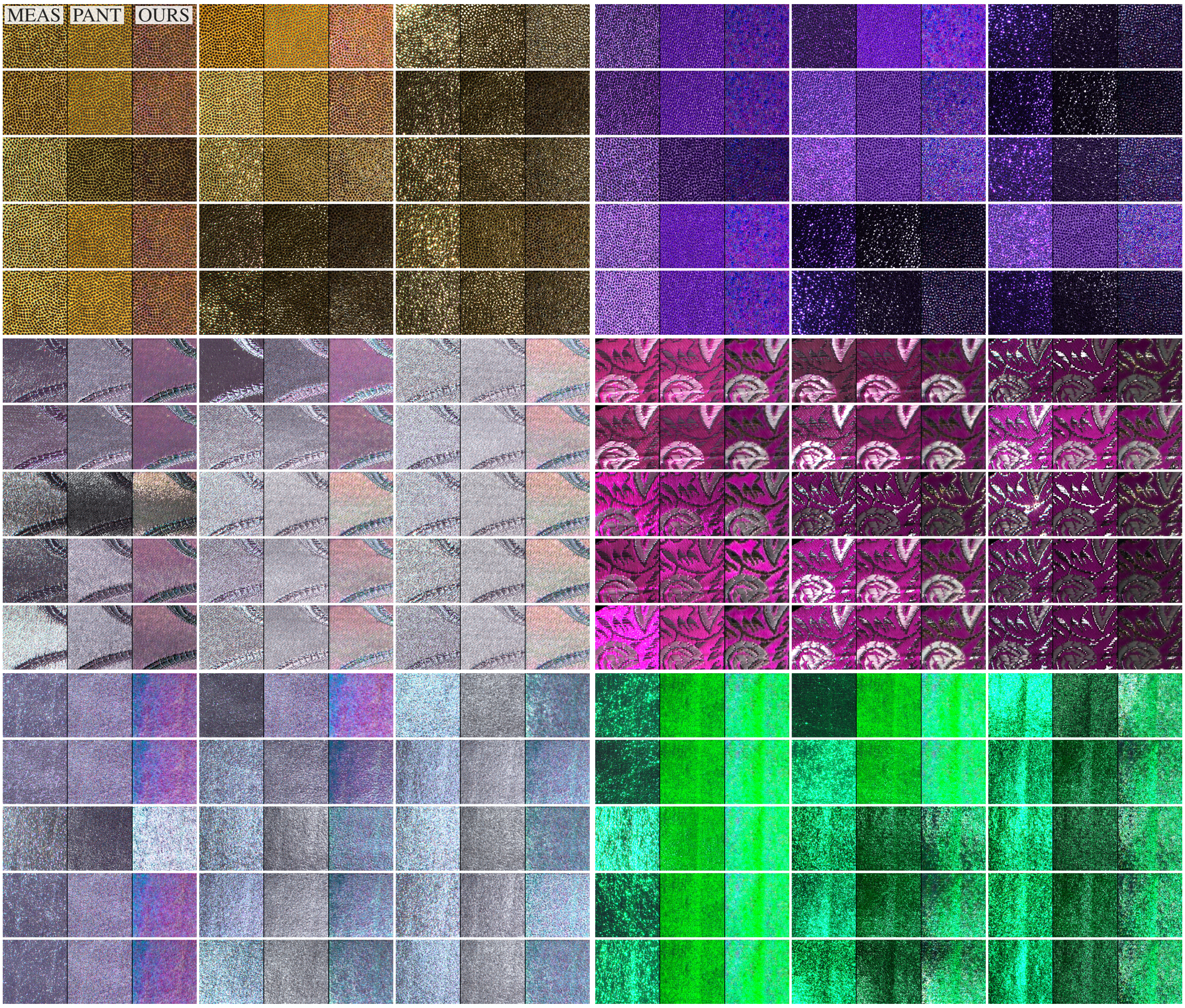

Figure 9: *Some failure cases, comparison of some materials from our test set. For each group of three images, the leftmost (**MEAS**) is a measured image (for some random light and view directions), the middle one is a rerendering of the Pantora SVBRDF (**PANT**), and the right one is a rerendering of our SVBRDF estimate (**OURS**).*

We believe, that future deep learning approaches similar to the presented one have the potential to outperform conventional approaches not only in terms of efficiency, but also in terms of fitting quality, while at the same time requiring only a fraction of the calibrated data used here.

9. Conclusion

In this paper we present a deep learning based method for estimating reflectance model parameters at a quality that is comparable to that of a commercial state of the art fitting method. We achieve this by training our convolutional neural network on our newly collected large-scale fabric database of calibrated input images and corresponding SVBRDF labels obtained with the commercial optimization-based fitting software. Contrary to the several hours of processing time of the optimization-based method, our method provides high quality results in a few minutes, i.e. two orders of magnitude faster. This allows reducing the total fitting times of *two months* using the commercial software down to a few hours for the entire database. Furthermore, we show that our results are of much higher quality than what could possibly be achieved by few-shot reflectance estimation methods, showing their general limitations.

10. Acknowledgements

This work was supported by the DFG project KL 1142/11-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior).

References

[AAL16] AITTALA, MIIKA, AILA, TIMO, and LEHTINEN, JAAKKO. "Reflectance modeling by neural texture synthesis". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 65. URL: https://mediatech.aalto.fi/publications/graphics/NeuralSVBRDF/ 2–4.

[ACGO18] ALBERT, RACHEL A, CHAN, DORIAN YAO, GOLDMAN, DAN B, and O'BRIEN, JAMES F. "Approximate svBRDF estimation from mobile phone video". *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*. Eurographics Association. 2018, 11–22 3.

[Ado19] ADOBE. *3D assets*. June 2019. URL: https://web.archive.org/web/20190610001634/https://stock.adobe.com/3d-assets 4.

[all19] ALLEGORITHMIC. *Substance share*. May 2019. URL: https://web.archive.org/web/20190529075802/https://share.substance3d.com/ 4.

[AS00] ASHIKHMIN, MICHAEL and SHIRLEY, PETER. "An anisotropic phong BRDF model". *Journal of graphics tools* 5.2 (2000), 25–32 4.

[AWL*15] AITTALA, MIIKA, WEYRICH, TIM, LEHTINEN, JAAKKO, et al. "Two-shot SVBRDF capture for stationary materials." *ACM Trans. Graph.* 34.4 (2015), 110–1 3.

[AWL13] AITTALA, MIIKA, WEYRICH, TIM, and LEHTINEN, JAAKKO. "Practical SVBRDF capture in the frequency domain." *ACM Trans. Graph.* 32.4 (2013), 110–1 3.

[Bli77] BLINN, JAMES F. "Models of light reflection for computer synthesized pictures". *ACM SIGGRAPH computer graphics*. Vol. 11. 2. ACM. 1977, 192–198 4.

[CDP*14] CHEN, GUOJUN, DONG, YUE, PEERS, PIETER, et al. "Reflectance scanning: estimating shading frame and BRDF with generalized linear light sources". *ACM Transactions on Graphics (TOG)* 33.4 (2014), 117 3.

[CT82] COOK, ROBERT L and TORRANCE, KENNETH E. "A reflectance model for computer graphics". *ACM Transactions on Graphics (TOG)* 1.1 (1982), 7–24 4.

[DAD*18] DESCHAINTRE, VALENTIN, AITTALA, MIIKA, DURAND, FREDO, et al. "Single-image svbrdf capture with a rendering-aware deep network". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 128. URL: https://team.inria.fr/graphdeco/projects/deep-materials/ 1, 2, 4, 6–8, 10.

[dBWK15] Den BROK, DENNIS, WEINMANN, MICHAEL, and KLEIN, REINHARD. "Linear Models for Material BTFs". *Eurographics Workshop on Material Appearance Modeling*. The Eurographics Association, 2015, 15–20. ISBN: 978-3-905674-83-5. DOI: 10.2312/mam.20151198 6.

[DWT*10] DONG, YUE, WANG, JIAPING, TONG, XIN, et al. "Manifold bootstrapping for SVBRDF capture". *ACM Transactions on Graphics (TOG)* 29.4 (2010), 98 3.

[FSH16] FICHET, ALBAN, SATO, IMARI, and HOLZSCHUCH, NICOLAS. "Capturing spatially varying anisotropic reflectance parameters using fourier analysis". *Graphics Interface Conference 2016*. CHCCS/SCDHM. 2016, 65–73 3.

[GD10] GEISLER-MORODER, DAVID and DÜR, ARNE. "A new ward brdf model with bounded albedo". *Computer Graphics Forum*. Vol. 29. 4. Wiley Online Library. 2010, 1391–1398 4, 5.

[GEB15] GATYS, LEON A, ECKER, ALEXANDER S, and BETHGE, MATTHIAS. "A neural algorithm of artistic style". *arXiv preprint arXiv:1508.06576* (2015) 3.

[GRR*17] GEORGOULIS, STAMATIOS, REMATAS, KONSTANTINOS, RITSCHEL, TOBIAS, et al. "Reflectance and natural illumination from single-material specular objects using deep learning". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.8 (2017), 1932–1947. URL: http://www.vision.ee.ethz.ch/~georgous/DRM_DeLight/index.html 3, 4.

[GTHD03] GARDNER, ANDREW, TCHOU, CHRIS, HAWKINS, TIM, and DEBEVEC, PAUL. "Linear light source reflectometry". *ACM Transactions on Graphics (TOG)* 22.3 (2003), 749–758 3.

[Hub64] HUBER, PJ. "Robust estimation of a location parameter". *Annals of Mathematical Statistics* (1964) 7.

[IS15] IOFFE, SERGEY and SZEGEDY, CHRISTIAN. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *arXiv preprint arXiv:1502.03167* (2015) 7.

[KB14] KINGMA, DIEDERIK P and BA, JIMMY. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014) 7.

[KCW*18] KANG, KAIZHANG, CHEN, ZIMIN, WANG, JIAPING, et al. "Efficient reflectance capture using an autoencoder". *ACM Trans. Graph* 37.4 (2018) 2, 3.

[KG13] KARIS, BRIAN and GAMES, EPIC. "Real shading in unreal engine 4". *Proc. Physically Based Shading Theory Practice* 4 (2013) 4.

[KGT*17] KIM, KIHWAN, GU, JINWEI, TYREE, STEPHEN, et al. "A lightweight approach for on-the-fly reflectance estimation". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 20–28. URL: https://research.nvidia.com/publication/reflectance-estimation-fly 3, 4.

[KRFS18] KOHLBRENNER, ADRIAN, RUMP, MARTIN, FRICK, BEAT, and SCHWARTZ, CHRISTOPHER. *Method and apparatus for digitizing the appearance of a real material*. US Patent App. 10/026,215. July 2018 3, 5.

[LCY*17] LIU, GUILIN, CEYLAN, DUYGU, YUMER, ERSIN, et al. "Material editing using a physically based rendering network". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 2261–2269. URL: http://masc.cs.gmu.edu/wiki/material 4.

[LDPT17] LI, XIAO, DONG, YUE, PEERS, PIETER, and TONG, XIN. "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 45. URL: http://msraig.info/~sanet/sanet.htm 2, 4, 10.

[LKG*03] LENSCH, HENDRIK, KAUTZ, JAN, GOESELE, MICHAEL, et al. "Image-based reconstruction of spatial appearance and geometric detail". *ACM Transactions on Graphics (TOG)* 22.2 (2003), 234–257 3.

[LSC18] LI, ZHENGQIN, SUNKAVALLI, KALYAN, and CHANDRAKER, MANMOHAN. "Materials for masses: SVBRDF acquisition with a single mobile phone image". *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 72–87 2, 4, 10.

[LXR*18] LI, ZHENGQIN, XU, ZEXIANG, RAMAMOORTHI, RAVI, et al. "Learning to reconstruct shape and spatially-varying reflectance from a single image". *SIGGRAPH Asia 2018 Technical Papers*. ACM. 2018, 269 2, 4, 10.

[Mat03] MATUSIK, WOJCIECH. "A data-driven reflectance model". PhD thesis. Massachusetts Institute of Technology, 2003 3, 4.

[ML15] MÜLLER, GERO and LAMY, FRANCIS. *AxF-appearance exchange format*. Tech. rep. Technical report, X-Rite Inc., 4300 44th St. SE, Grand Rapids, MI 49505, 2015 6.

[MMZ*18] MEKA, ABHIMITRA, MAXIMOV, MAXIM, ZOLLHOEFER, MICHAEL, et al. "Lime: Live intrinsic material estimation". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 6315–6324. URL: http://gvv.mpi-inf.mpg.de/projects/LIME/ 4.

[NDM05] NGAN, ADDY, DURAND, FRÉDO, and MATUSIK, WOJCIECH. "Experimental analysis of brdf models." *Rendering Techniques* 2005.16th (2005), 2 3.

[Nex19] NEXTLIMIT. *Maxwell Render*. May 2019. URL: https://web.archive.org/web/20190505055737/http://www.nextlimit.com/maxwell/ 5.

[Nis09] NISHINO, KO. "Directional statistics BRDF model". *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, 476–483 4.

[NJR15] NIELSEN, JANNIK BOLL, JENSEN, HENRIK WANN, and RAMAMOORTHI, RAVI. "On optimal, minimal BRDF sampling for reflectance acquisition". *ACM Transactions on Graphics (TOG)* 34.6 (2015), 186 4.

[PFG00] PELLACINI, FABIO, FERWERDA, JAMES A, and GREENBERG, DONALD P. "Toward a psychophysically-based light reflection model for image synthesis". *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 2000, 55–64 4.

[Pho75] PHONG, BUI TUONG. "Illumination for computer generated pictures". *Communications of the ACM* 18.6 (1975), 311–317 3, 4.

[RJGW19] RAINER, GILLES, JAKOB, WENZEL, GHOSH, ABHIJEET, and WEYRICH, TIM. "Neural BTF Compression and Interpolation". *Computer Graphics Forum (Proceedings of Eurographics)* 38.2 (Mar. 2019) 5, 7.

[RPG16] RIVIERE, JÉRÉMY, PEERS, PIETER, and GHOSH, ABHIJEET. "Mobile surface reflectometry". *Computer Graphics Forum*. Vol. 35. 1. Wiley Online Library. 2016, 191–202 3.

[RWS*11] REN, PEIRAN, WANG, JIAPING, SNYDER, JOHN, et al. "Pocket reflectometry". *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM. 2011, 45 3.

[Sch94] SCHLICK, CHRISTOPHE. "An inexpensive BRDF model for physically-based rendering". *Computer graphics forum*. Vol. 13. 3. Wiley Online Library. 1994, 233–246 5.

[VCGL19] VIDAURRE, RAQUEL, CASAS, DAN, GARCES, ELENA, and LOPEZ-MORENO, JORGE. "BRDF Estimation of Complex Materials with Nested Learning". *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, 1347–1356 2–4.

[War*92] WARD, GREGORY J et al. "Measuring and modeling anisotropic reflection". *Computer Graphics* 26.2 (1992), 265–272 3–5.

[WDR11] WU, HONGZHI, DORSEY, JULIE, and RUSHMEIER, HOLLY. "A sparse parametric mixture model for BTF compression, editing and rendering". *Computer Graphics Forum*. Vol. 30. 2. Wiley Online Library. 2011, 465–473 3.

[WGK14] WEINMANN, MICHAEL, GALL, JUERGEN, and KLEIN, REINHARD. "Material classification based on training data synthesized using a BTF database". *European Conference on Computer Vision*. Springer. 2014, 156–171 7.

[WMLT07] WALTER, BRUCE, MARSCHNER, STEPHEN R, LI, HONGSONG, and TORRANCE, KENNETH E. "Microfacet models for refraction through rough surfaces". *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association. 2007, 195–206 4.

[WZT*08] WANG, JIAPING, ZHAO, SHUANG, TONG, XIN, et al. "Modeling anisotropic surface reflectance with example-based microfacet synthesis". *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM. 2008, 41 3.

[XRi18] X-RITE. *TAC7-Scanner*. June 2018. URL: http://web.archive.org/web/20180615015942/https://www.xrite.com/categories/appearance/tac7 2, 3, 5.

[XRi19] X-RITE. *Pantora Material Hub*. Apr. 2019. URL: https://web.archive.org/web/20190424232441/https://www.xrite.com/categories/appearance/pantora-software 2, 3, 5.

[YL16] YU, CHANKI and LEE, SANG WOOK. "Branch and bound algorithm for accurate estimation of analytical isotropic bidirectional reflectance distribution function models". *Applied optics* 55.15 (2016), 4193–4200 3.

[YLD*18] YE, WENJIE, LI, XIAO, DONG, YUE, et al. "Single Image Surface Appearance Modeling with Self-augmented CNNs and Inexact Supervision". *Computer Graphics Forum*. Vol. 37. 7. Wiley Online Library. 2018, 201–211. URL: http://msraig.info/~InexactSA/inexactsa.htm 2, 4, 10.

[YS17] YU, YE and SMITH, WILLIAM AP. "Pvnn: A neural network library for photometric vision". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 526–535 4.

[YSL10] YU, CHANKI, SEO, YONGDUEK, and LEE, SANG WOOK. "Global optimization for estimating a BRDF with multiple specular lobes". *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, 319–326 3.