Deep Reflectance Scanning: Recovering Spatially-varying Material Appearance from a Flash-lit Video Sequence

Wenjie Ye[1,2] Yue Dong[2] Pieter Peers[3] Baining Guo[2]

[1]Tsinghua University
[2]Microsoft Research Asia
[3]College of William & Mary

Abstract

In this paper we present a novel method for recovering high-resolution spatially-varying isotropic surface reflectance of a planar exemplar from a flash-lit close-up video sequence captured with a regular hand-held mobile phone. We dot not require careful calibration of the camera and lighting parameters, but instead compute a per-pixel flow map using a deep neural network to align the input video frames. For each video frame, we also extract the reflectance parameters, and warp the neural reflectance features directly using the per-pixel flow, and subsequently pool the warped features. Our method facilitates convenient hand-held acquisition of spatially-varying surface reflectance with commodity hardware by non-expert users. Furthermore, our method enables aggregation of reflectance features from surface points visible in only a subset of the captured video frames, enabling the creation of high-resolution reflectance maps that exceed the native camera resolution. We demonstrate and validate our method on a variety of synthetic and real-world spatially-varying materials.

Keywords: SVBRDF, hand-held capture, automatic alignment

1. Introduction

Reproducing the appearance of real-world materials for use in virtual worlds is a challenging problem that has seen tremendous progress. Leveraging recent advances in deep learning, a number of techniques have been introduced that enable non-expert users to digitize real-world material appearance with commodity hardware from a single photograph [LDPT17, YLD*18, DAD*18, LSC18, LXR*18]. More recently, deep-learning based methods that operate on a variable number of input photographs have been introduced to overcome the inherent accuracy limitation of single photograph methods [GLD*19, DAD*19]. While producing more accurate results, such multiple-image methods limit the resolution of the material maps to that of a single photograph, and require an arduous and error-prone calibration and alignment of the photographs to compensate for the differences in view, limiting the practical applicability of these methods for non-expert users.

In this paper we introduce a novel easy-to-use deep-learning based method for digitizing the isotropic material appearance of a planar material exemplar by *"scanning"* a mobile phone over the surface of the sample lit by the camera's flash. Our method is self-calibrating; it does not require manual alignment or additional markers to be placed in the scene. Besides greatly simplifying acquisition and improving alignment accuracy, this automatic alignment also allows us to scan SVBRDFs at resolutions exceeding the camera resolution by aligning the video frames to a (relatively low resolution) macro view of the surface. Our method consists of two main components: a motion and warping alignment network, and an image-to-reflectance translation network. A key challenge is that the different observations might only partially overlap and consequently a variable number of observations are available for each surface point, precluding prior deep-learning based strategies that treat each observed pixel value as a valid measurement with an equal number of observations. To address this challenge, we train our image-to-reflectance network such that the encoded intermediate neural reflectance features can be warped by the results from the alignment network; allowing us to combine the extracted deep reflectance features directly and decode the combined reflectance property maps.

Our solution builds on two existing deep network architectures: PWC-net [RGS*19] for motion and warp estimation, and a multi-image reflectance estimation network [DAD*19]. We employ two instances of PWC-net; one trained to estimate the flow between subsequent frames, and one trained to estimate flow between distant frames to combat drift over longer sequences. The reflectance estimation network of Deschaintre et al. [DAD*19] is expanded by adding a "warping" layer before the max-pooling layer that aggregates the feature vectors from each input frame. The advantage of warping feature vectors instead of the input photographs, is that invalid pixels/features (i.e., pixels in the target SVBRDF without a corresponding pixel in the source video frame or vice versa) can be set to a low value such that they are effectively ignored during aggregation by max-pooling; this would not be possible if the input images were warped before inputting them into the network. We train our networks with a synthetic *"scanning"* dataset. This synthetic training set is based on the Abode Stock 3D Material dataset [Ado18] and the INRIA SVBRDF dataset [DAD*18] augmented with a Perlin noise [Per02] based specular enhancement. Furthermore, we enhance the reflectance-estimation network by replacing the instance normalization with a convolution weight normalization [KLA*20] to address small high value artifacts in the reconstructed reflectance property maps.

We demonstrate and thoroughly analyze our deep reflectance scanning technique on a wide variety of synthetic and real-world materials. In summary our contributions are:

1. A method for automatically aligning the photographs in a multi-image reflectance capture process;
2. that performs the alignment directly on the neural reflectance features to improve robustness against the adverse effects of unseen surface points (in one or more photographs);
3. and that enables aggregation of reflectance information from multiple viewpoints to produce a higher resolution SVBRDF.
4. Finally, we present various improvements on prior neural reflectance estimation networks that yield more accurate specular and diffuse reflectance properties.

2. Related Work

Measurement-based appearance modeling has been an active research topic over the past few decades. In this section, we focus our discussion on work related to *hand-held reflectance modeling* techniques. We refer to Weinmann et al. [WdBKK15] and Guarnera et al. [GGG*16] for a broader overview of reflectance and appearance modeling, and to the survey by Dong [Don19] on deep learning-based appearance modeling. We further categorize related work in single versus multi-image approaches:

Single Image Reflectance Estimation Methods These methods have the advantage that no registration between different photographs is required, thereby greatly easing processing. However, a single observation only provides limited information. To aid reconstruction, assumptions are typically made on either shape, material properties, or lighting. Romeiro et al. [RVZ08, RZ10] assume a spherical exemplar to recover the homogeneous reflectance properties under uncontrolled natural lighting. Li et al. [LDPT17], and the extension by Ye et al. [YLD*18], recover spatially-varying surface reflectance from a planar sample under uncontrolled natural lighting by assuming a homogeneous specular component and by leveraging prior knowledge embedded in a deep neural network.

While convenient, uncontrolled natural lighting only provides weak cues for estimating surface reflectance. To elicit stronger cues, a number of solutions employ active co-located flash lighting. Aittala et al. [AAL16] exploit self-similarity and neural texture synthesis to recover spatially-varying reflectance properties of stationary materials. Deschaintre et al. [DAD*18] and Li et al. [LSC18] use a deep translation network to recover spatially-varying material properties from a single flash photograph of a flat exemplar. Finally, Li et al. recover shape and surface reflectance from a photograph of an arbitrary shaped exemplar [LXR*18] or from a photograph of a scene under spatially varying lighting [LSR*20].

Despite the additional assumptions on either shape, surface reflectance, and/or lighting, these methods are fundamentally limited in what can be recovered of the 6D reflectance function from a 2D observation. In our method, we aggregate reflectance cues from multiple observations to improve the accuracy and robustness.

Multiple Image Reflectance Estimation Methods These methods aggregate reflectance cues from multiple observations to reconstruct the surface reflectance. Similar as with single image methods, a number of techniques infer surface reflectance from cues under natural lighting for recovering homogeneous surface reflectance [LN16, ON16, LPG19], or spatially-varying surface reflectance [DCP*14, XDPT16]. While multiple observations help to strengthen the cues, the overall reconstruction quality is still conditioned on the quality of the uncontrolled lighting.

To better regularize the reflectance reconstruction, active illumination methods control the lighting on the scene, either using a controlled light source (e.g., a linear light source [RWS*11]), structured illumination from an LED cube [KCW*18, KXH*19], photometric images [BXS*20a], or using the co-located camera flash light [AWL15,RPG16,HSL*17,GLD*19,DAD*19,GSH*20]. However, these methods all require that the full sample remains in view for all photographs, limiting the size of material samples that can be acquired, and they require careful geometrical alignment of the input photographs. A notable exception is the work by Deschaintre et al. [DDB20], who overfit a neural network to transfer material parameters from a small set of exemplars. While Deschaintre et al. do not require any alignment between the exemplars and the guidance photograph, they require fully labeled exemplars. Nam et al. [NLGK18] recover both shape and reflectance from backscatter observations and thus can take occlusions in account. However, their method relies on complex and time-consuming nonlinear optimizations. Bi et al. [BXS*20b] avoid non-linear optimization by using a deep learning framework to estimate shape and reflectance from multiple photographs, at the cost of requiring a complex and carefully calibrated acquisition setup. Using a similar acquisition procedure as us, Albert et al. [ACGO18] recover spatially varying surface reflectance by scanning a mobile camera over the planar exemplar. However, their method requires the addition of markers to the scene and relies on rigid homography-based alignment. Furthermore, explicit (recursive) clustering is employed to (non-linearly) fit BRDFs to the observations, possibly losing unique reflectance details. Our method avoids the use of fragile non-linear BRDF fitting, and instead builds on deep neural networks to aggregate the reflectance information similarly to Deschaintre et al. [DAD*19] but without requiring an explicit alignment of the input photographs.

3. Overview

Our method is designed from the top down starting from a user-friendly hand-held acquisition procedure. We will therefore first detail the acquisition process before detailing the technical machinery that enables reflectance reconstruction.

The acquisition process is designed to be accessible to non-

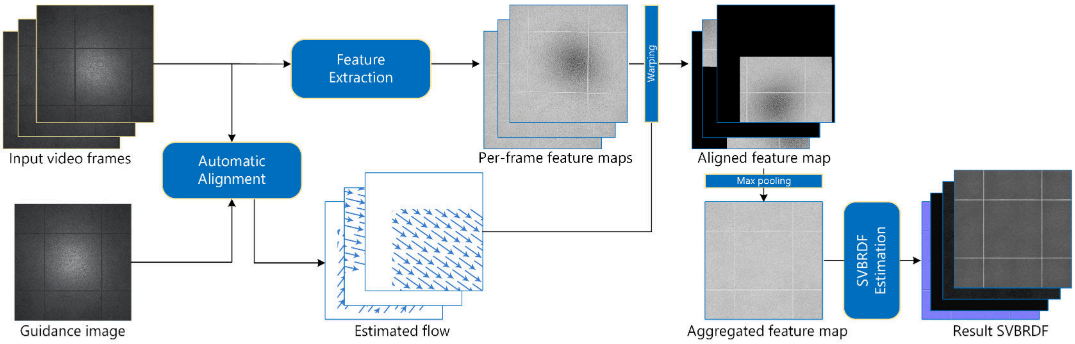


Figure 1: *Overview of our method. Given a guidance image and an input video, we first compute a warp function between each input video frame and the guidance image by estimating per-frame optical flow. Next, we extract SVBRDF feature vectors for each input frame and align (warp) the feature maps. Finally, the SVBRDF is reconstructed based on aggregated (max-pooled) feature maps.*

expert users, and therefore only relies on commodity hardware in the form of a standard mobile phone operating in video mode with the flash enabled. We opt for using a video sequence instead of a discrete set of photographs for two reasons. First, capturing a discreet set of photographs is only practical for a small number of photographs; users quickly loose track for which view positions they already captured a photograph and are consequently uncertain where to capture the next photograph. Second, automatic alignment is easier on video frames when subsequent frames will have less visual differences that can adversely affect alignment (e.g., moving highlights). Our capture procedure is straightforward: the user holds the mobile phone at a relatively fixed distance above the material sample pointing straight down at the sample, and "scans" the mobile phone over the exemplar. The user needs to take care that the target surface is fully scanned, and that the captured frames are free of motion blur; a specially designed app that provides feedback to the user regarding these constraints is a direction for future development. In addition, we require the user to acquire a guidance image of the area of interest for which the SVBRDF is recovered. This guidance image can be in the form of a selected frame from the video sequence or a separately captured photograph (possibly captured from a macro view). For ease of exposition and clarity, we will first explain the core components of our method by assuming the guidance image is a frame from the video sequence, and thus the resolution of the video frames, the guidance image, and the reconstructed SVBRDF are the same. In Section 7, we will generalize the guidance image to photographs under different lighting and at a possibly different scale and resolution to enable the reconstruction of high-resolution SVBRDFs.

Given the guidance image, our method fully automatically processes the video sequence and outputs spatially varying reflectance property maps corresponding to the guidance image in three stages. First, we estimate a per-frame flow between subsequent frames. Naively accumulating these per-frame flow maps, will result in drift over long sequences. Therefore, we regularize the accumulation by computing global transformation parameters for each frame based on the flow maps. While warping the frames purely based on the global transformation parameters matches the large scale motion well, it fails to capture small scale motions (e.g., due to occlusions, geometric details, deviations for the planar sample assumption, optical inaccuracies, etc.). Therefore, we compute a detail-flow map between the transformed frames and the guidance frame.

Directly warping the video frames to the guidance image will result in an uneven number of warped pixels for each guidance pixel which complicates SVBRDF estimation. Ideally we would like to leverage all the available information from all of the video frames for estimating the SVBRDF, therefore, instead of aggregating the directly warped pixels, we first extract per-pixel feature vectors for each frame, and then warp the per-pixel features based on the refined flow, before accumulating the features using a max-pooling layer from which the reflectance properties are estimated. The resulting reflectance properties maps contain per-pixel estimates of the diffuse albedo, specular albedo, specular roughness, and normal map that drive a (per-pixel) GGX microfacet BRDF model [WMLT07]. Figure 1 summarizes our method.

We will first describe the synthetic training dataset used for training both components of our framework (Section 4), before detailing the automatic alignment network (Section 5) and the SVBRDF estimation network (Section 6). Finally, in Section 7 we generalize our method to estimate high-resolution SVBRDFs from a guidance image which is not part of the video sequence, possibly captured from a macro-view of the exemplar under different lighting.

4. Training Data

To train both components of our solution (i.e., the automatic alignment network and the SVBRDF estimation network), we generate a synthetic dataset of video sequences with a simulated camera and a co-located light source scanning synthetic SVBRDFs.

SVBRDF datasets At its core, our synthetic dataset is synthesized from the INRIA SVBRDF dataset [DAD*18] and the Adobe Stock 3D Material dataset [Ado18]. This yields 1,195 SVBRDFs for the Adobe Stock 3D Material dataset, of which 1,000 are used for training and 195 are used for testing. We use the Adobe SVBRDFs at their native 4,096 × 4,096 resolution. The INRIA SVBRDF dataset includes 1,590 SVBRDF exemplars, and 38 test exemplars. However, the SVBRDFs in the INRIA dataset are only at 512 × 512

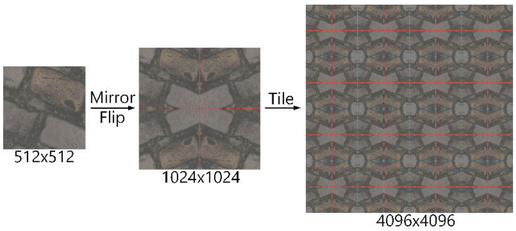


Figure 2: *We enlarge a* 512×512 *resolution SVBRDF into a* $4{,}096 \times 4{,}096$ *resolution SVBRDF by first creating a* $1{,}024 \times 1{,}024$ *resolution map by mirroring and flipping, and subsequently tiling the* $1{,}024 \times 1{,}024$ *SVBRDF to a full* $4{,}096 \times 4{,}096$ *resolution SVBRDF.*

resolution. Because this is too small for a simulated scanning acquisition, we augment their resolution to $4{,}096 \times 4{,}096$ as follows. First, we increase their resolution to $1{,}024 \times 1{,}024$ by tiling a mirrored and/or flipped copy to the other quadrants. Next, we simply tile this $1{,}024 \times 1{,}024$ SVBRDF to a full $4{,}096 \times 4{,}096$ resolution (Figure 2). To avoid biases when inferring the flow maps due to the tiling, we only use the augmented INRIA SVBRDF dataset for training the SVBRDF estimation network; we use the Adobe Stock 3D Material dataset for training both the alignment as well as the estimation component.

SVBRDF Augmentation Both the INRIA SVBRDF dataset and the Adobe Stock 3D Material dataset contain a relatively small number of specular materials, especially, materials with a structured specular component. While this bias is not critical for training the alignment network, it is important for training the SVBRDF estimation network. Hence, for the latter, we augment the SVBRDFs with a structured specular component. First, we precompute 100 Perlin noise maps at $4{,}096 \times 4{,}096$ resolution, using 1 to 3 octaves with frequencies randomly selected uniformly between 16 and 48, and threshold the resulting map with a threshold randomly sampled from $[-0.3, 0.3]$ to obtain a structured binary map. When we encounter a diffuse material during training (i.e., all specular values fall below 0.1), we decide with a 50-50 probability to blend (for a randomly selected thresholded binary Perlin noise map) a homogeneous material with the diffuse albedo, specular albedo, and specular roughness maps with randomly sampled specular albedo color, a specular roughness sampled from $[0.1, 0.4]$, and a diffuse albedo color proportional to the specular albedo multiplied with a random scale in $[0, 0.1]$ (Figure 3).

In addition, to further diversify the SVBRDF exemplars used for training the SVBRDF estimation network, we apply the following additional augmentations on the fly during training:

1. We flip horizontally, vertically, and rotate by 90 degrees with a 50% probability.
2. We randomly scale the diffuse albedo, the specular albedo, and specular roughness with a scale sampled from $[0.8, \min(1.25, 1/(v_{max} + 0.0001))]$, where v_{max} is the maximum value in the corresponding map. The resulting roughness map is furthermore remapped (i.e., scale + offset) to $[0.1, 1.0]$ to avoid very small roughness values.

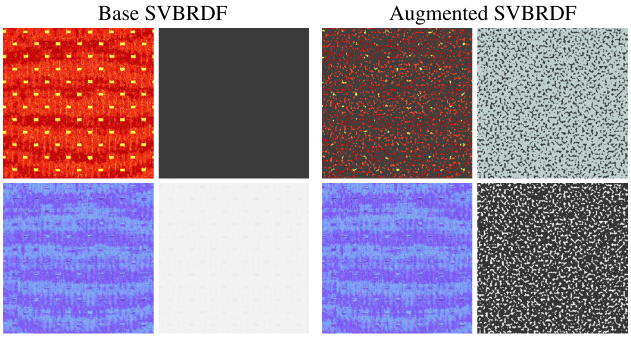


Figure 3: *Illustration of Perlin noise based specular augmentation. Given a SVBRDF (left; first row: diffuse and specular albedo, second row: normal and specular roughness), a homogeneous specular component is added with its spatial distribution determined by a thresholded Perlin noise map; the diffuse component is similarly modulated (right).*

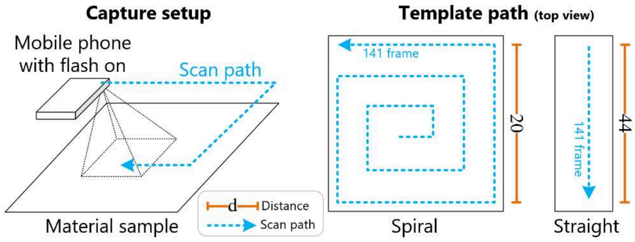


Figure 4: *Illustration of the capture setup, and the two template paths used for generating the synthetic video sequences.*

Video Sequence Synthesis Each synthetic video sequence contains 141 frames and, similarly to prior works [LDPT17, DAD*18, DAD*19], each video sequence is rendered on-the-fly during training. An augmented SVBRDF is put on the $z = 0$ plane and mapped such that it covers $[-1, 1] \times [-1, 1]$ in the x and y coordinates. The field of view is uniformly selected between 25° and 35°. Each video sequence starts at a random starting point, with x and y sampled uniformly from $[-0.5, 0.5]$, and the height offset is set to $\frac{\xi}{\tan(\frac{1}{2} fov)}$, where ξ is uniformly sampled from $[0.05, 0.1]$. Furthermore, we assume that the rotation around the view direction is independent of the camera motion, and we initially set it such that the 'up' axis of the view corresponds to $(0, 1, 0)$.

The motion of each video is based on 2 templates: a spiral template, and a straight (up-down) line (see Figure 4). The former is used for training both the alignment and the SVBRDF networks. The latter is only used for training the SVBRDF network (selected with $1/3$ probability). The templates are further augmented to better reflect the diversity of real-world acquisition conditions as follows:

1. The template path is flipped, mirrored, and rotated 90 degrees, each with a 50% probability.
2. The template paths are scaled with a factor uniformly sampled from $[0.075, 0.1]$ and then centered at the starting point. The

Table 1: *Magnitudes of λ_f (the strength of the perturbation on the acceleration), λ_f^v (the influence of the previous frame's velocity), and λ_f^p (the influence of the previous frame's position) for the different components: $\{c_x, c_y, c_z, l_x, l_y, t\}$.*

f	λ_f	λ_f^v	λ_f^p
c_x, c_y	0.001	0.1	0.2
c_z	0.001	0.01	0.1
l_x, l_y	0.001	0.01	0.1
t	0.001	0.01	0.1

camera (initially) looks straight down on the SVBRDF exemplar.

3. To simulate the variability of free-form hand-held acquisition, we perturb the acceleration of the camera; directly perturbing the velocity of the camera results in an unnatural discontinuous motion. We perturb the acceleration for a component 'f' for the 'i'-th frame by:

$$a_f[i] = \lambda_f \mathcal{N}(0,1) - \lambda_f^v v_f[i-1] - \lambda_f^p p_f[i-1], \tag{1}$$

where λ_f is the strength of the perturbation, λ_f^v measures the influence of the previous frame's velocity v_f, and λ_f^p is the influence of the previous frame's position p_f. The exact magnitude of the weight parameters depends on the component f; Table 1 summarizes the weights. The components f can be: the x and y-coordinate of the position (c_x and c_y respectively), the camera height (c_z), the look-at point (l_x, l_y) (defined as the projection of (c_x, c_y) on the SVBRDF plane), and the cotangent of up direction (in the SVBRDF plane) t, i.e., the 'up' axis corresponds to $(t, 1, 0)$. Given the acceleration a_f for the component $f \in \{c_x, c_y, c_z l_x, l_y, t\}$, the perturbed component $\bar{f}$ is computed as:

$$\begin{aligned} v_f[i] &= v_f[i-1] + a_f[i], & (2)\\ p_f[i] &= p_f[i-1] + v_f[i], & (3)\\ \bar{f} &= f + p_f[i]. & (4) \end{aligned}$$

Finally, given the 141 frame camera path, we render each frame at 256 × 256 resolution with the light source co-located with the camera. The light source intensity is randomly sampled for training PWC-net, following the strategy by Deschaintre et al. [DAD*19], and proportional to the (initial) square camera distance/height for training the SVBRDF estimation net. All rendering is implemented directly in TensorFlow and computed at training time.

5. Automatic Alignment

The first step in our method is to register the target frame to the guidance image. Registration is an active research area and a wide variety of solutions exist [Nag17, Sze06]. Robustly registering the input frames for reflectance scanning poses a number of challenges. First, the top-down view with semi-constant viewing distance precludes methods based on estimating camera extrinsic and intrinsic parameters which become unstable in such cases. Second, the co-located light source also induces strong view-dependent appearance changes that can confuse algorithms based on explicit tracking of sparse feature points. Finally, lens distortion and deviations from planarity of the material sample necessitates a per-surface point alignment and precludes simple global alignment methods. Therefore, we decided to opt for a dense per-surface point alignment strategy based on optical flow with a global correction step to minimize drift.

We will assume an input video sequence $\mathbf{V} = \{\mathbf{F}_0, ..., \mathbf{F}_{n-1}\}$ of n frames $\mathbf{F}_i$, and that (without loss of generality) $\mathbf{F}_0$ also serves as the guidance image. To estimate the surface reflectance sufficiently different appearance observations are needed, while for alignment as little as possible appearance difference is preferred. Hence, we will only use a regularly sampled (with step size k) subset of the captured frames $K = \{\mathbf{F}_0, \mathbf{F}_k, \mathbf{F}_{2k}, ...\}$ with sufficient appearance difference for estimating the reflectance (we denote these selected frames K as *key frames*), but we use all frames for estimating the alignment. Our goal is now to compute a set of warp function (i.e., flow) $\{W_0, W_k, W_{2k}, ...\}$ to warp a frame to the guidance image: $\mathbf{F}_0 \approx W_{tk}(\mathbf{F}_{tk})$.

Alignment Overview To avoid error accumulation over long sequences, we employ a three step algorithm to compute regularized warp maps between the guidance image and the key frames. In a first step, we will compute a warp between subsequent frames $w_{i\to i+1}$ for all frames in the sequence $\mathbf{V}$. Next, we compute a rough estimate of the warp function $\tilde{W}_{tk}$ from the guidance image to the t-th key frame $\mathbf{F}_{tk}$ based on the regularized warp function $W^r_{(t-1)k}$ of the previous key frame $\mathbf{F}_{(t-1)k}$, and the frame-per-frame flows of the intermediate frames: $\tilde{W}_{tk} = w_{tk-1\to tk} \circ w_{tk-2\to tk-1} \circ \ldots \circ w_{(t-1)k\to(t-1)k+1} \circ W^r_{(t-1)k}$. The regularized warp $W^r_{(t-1)k}$ is computed as the best per-image affine transformation approximation of $W_{(t-1)k}$. We then compute the final per-pixel warp W_{tk} as the superposition of the (regularized) rough warp $\tilde{W}^r_{tk}$ and a fine-scale alignment W^f_{tk} such that $\mathbf{F}_0 \approx W_{tk}(\mathbf{F}_{tk})$, with $W_{tk} = W^f_{tk} \circ \tilde{W}^r_{tk}$.

Warp Computation To compute the frame-to-frame warps $w_{i\to i+1}$ and the fine-scale alignment warps W^f_{tk} we use PWC-net [RGS*19]. We employ two networks, one for computing the frame-to-frame warps and one for the fine-scale alignment, that are both a refinement of a common pre-trained PWC-net model (trained for 50 epochs on the "Flying Chairs" dataset). For the frame-to-frame network, we use all adjacent frames from a synthesized video sequence for training; the reference flow for the sequence is computed directly from the camera parameters. For the refinement network, we select two frames from the first 20 and last 120 frames of the sequence respectively. We compute a rough flow $\tilde{W}^r_k$ as the average x and y displacement between both frames' camera positions perturbed by a random offset in x and y direction between 0 and 10 pixels. To compute the detail warp W^f_{tk}, we transform the target image with the coarse flow $\tilde{W}^r_{tk}$, and crop the source and target image to minimize invalid pixels (i.e., that do not feature a corresponding pixel in the other image). If the overlapping width/height is less than 30 pixels, we repeat the process with newly selected random frames. As a consequence each training pair can have a different size and therefore we train the refinement network for variable image sizes with a batch size of 1.

Regularization A final ingredient in our alignment algorithm is the regularization of the accumulated flow $\tilde{W}_{tk}$ to the rough regularized warp $\tilde{W}^r_{tk}$. The key reason for this regularization is that the

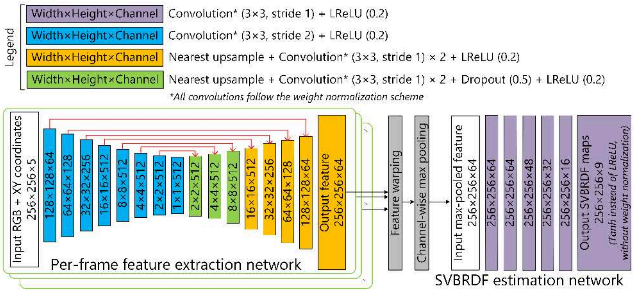

Legend
Width×Height×Channel Convolution* (3×3, stride 1) + LReLU (0.2)
Width×Height×Channel Convolution* (3×3, stride 2) + LReLU (0.2)
Width×Height×Channel Nearest upsample + Convolution* (3×3, stride 1) × 2 + LReLU (0.2)
Width×Height×Channel Nearest upsample + Convolution* (3×3, stride 1) × 2 + Dropout (0.5) + LReLU (0.2)
*All convolutions follow the weight normalization scheme
Input RGB + XY coordinates 256×256×5
128×128×64
64×64×128
32×32×256
16×16×512
8×8×512
4×4×512
2×2×512
1×1×512
2×2×512
4×4×512
8×8×512
16×16×512
32×32×256
64×64×128
128×128×64
Output feature 256×256×64
Per-frame feature extraction network
Feature warping
Channel-wise max pooling
Input max-pooled feature 256×256×64
256×256×64
256×256×64
256×256×48
256×256×32
256×256×16
Output SVBRDF maps 256×256×9 (Tanh instead of LReLU, without weight normalization)
SVBRDF estimation network

Figure 5: *SVBRDF Estimation Network.*

concatenation of long streams of frame-to-frame flows can result in drift (i.e., accumulation of subpixel errors and view-dependent appearance effects).

To regularize the flow, we assume that the field of view of the camera is 30°. Because the regularization step is refined in the next step, any errors introduced due to this assumption can still be corrected. Next, we collect all valid points between both frames (i.e., points that have corresponding points in the other frame), and solve for the rotation and translation transformations that best match the pairs using OpenCV's solvePnP function. Finally, we convert these transformation back to a dense flow map using OpenCV's projectPoints function.

6. SVBRDF Estimation

Given a set of key frames K and corresponding warps W_{tk} to the guidance image, our goal is to estimate the SVBRDF property maps $\mathbf{S} = \{\rho_d, \rho_s, m_s, n\}$, where ρ_d is the diffuse albedo map, ρ_s the specular albedo map, m_s the specular roughness map, and n the normal map.

SVBRDF Estimation Network Our SVBRDF estimation network is closely based on the SVBRDF estimation network of Deschaintre et al. [DAD*19]. Our solution differs on three important points. First, we employ a more richer training set, in particular, the Perlin Noise specular augmentation is a critical component for improving the estimation on materials with a structured specular component and for improving the accuracy of the specular roughness estimation. Second, we add a warping layer before the max-pooling aggregation layer on the $256 \times 256 \times 64$ feature maps. During warping we employ bilinear interpolation between the feature vectors, and invalid pixels are set to $-1e38$ such that they do not influence the max-pooling. Consequently, invalid pixels/features are effectively ignored during aggregation; this eases reflectance estimation as the network does not need to actively ignore invalid pixels as would be the case if the input images were warped before inputting them into the network. Third, we observe that the SVBRDF estimation network of Deschaintre et al. suffers from small high value artifacts. To solve this issue, we follow Karras et al. [KLA*20] who observed a similar phenomenon in the context of StyleGan, and replace the instance normalization by a convolution weight normalization. We normalize the convolution weights w_{ijk} as:

$$w'_{ijk} = \frac{w_{ijk}}{\sqrt{\sum_{i,k} w_{ijk}^2 + \varepsilon}}, \tag{5}$$

where i is the input channel dimension, j the output channel dimension, k the kernel spatial dimension, and $\varepsilon = 1e{-}8$. Finally, we simplify the network architecture by only using 3×3 convolutions, instead of a mixture of 3×3 and 4×4 in the network of Deschaintre et al. [DAD*19]. Figure 5 summarizes the SVBRDF estimation network.

Training For each 141 frame synthetic video sequence, we extract 20 different training exemplars for training the SVBRDF net. For each training exemplar, we select one of the first 20 frames as the guidance image, and we extract K key frames, where K is uniformly sampled from $[4, 16]$ (in other words, each training exemplar contains between 4 and 16 frames plus a guidance image). The key frames are selected at regular intervals starting at the guidance frame, with the interval length sampled from $[\lfloor 60/K \rfloor, ..., \lfloor 120/K \rfloor]$. The corresponding warp maps W_k are computed directly from the extrinsic camera parameters. The target SVBRDF property maps are resampled from the SVBRDF used to generate the synthetic video sequence based on the camera parameters of the guidance frame.

Our training loss consists of three components:

$$L = \delta_s L_s + \delta_r L_r + \delta_t L_t, \tag{6}$$

where L_s is the L_1 loss on the SVBRDF property maps with $\delta_s = 0.1$, L_r is the rendering loss as defined by Deschaintre et al. [DAD*19] which contains 3 renderings from diffuse directions, and 6 from specular directions, with $\delta_r = 1.0$. L_t is a novel top-view rendering loss with $\delta_t = 1.0$ for 6 top-view renderings from randomly selected camera positions, with the x and y coordinate uniformly sampled on the target SVBRDF, and the camera offset set to $\frac{1}{\tan(0.5\xi)}$, with ξ uniformly sampled from $[25, 35]$, making it similar to the video camera offsets.

7. High-resolution SVBRDFs

In the previous sections we assumed the guidance image is a frame from the video sequence. We will now generalize our method to a guidance image of the material taken from a macro vantage point (and thus seeing a larger area) possibly under different lighting. We aim to reconstruct the SVBRDF at the full combined coverage and resolution of the video frames, possibly exceeding the resolution of the guidance image and the video frames.

First, we bilinearly upsample the guidance image to match the pixels-per-area ratio of the video frames. We also manually set the position of the first frame with respect to the upsampled guidance image. Currently, this process is manual; automating this step would be an interesting avenue for future improvements.

Next, we run the alignment algorithm 2 times. After the first iteration, we update the guidance image based on the warped video frames. We update each pixel in the guidance image with the average of all corresponding pixels in the video key frames. If no video key frames 'overlap' with the guidance image pixel, we keep the original pixel value. As a result, the upsampled guidance image becomes more detailed, and consequently, the second alignment pass will be more accurate too.

Finally, we use the SVBRDF estimation network to translate all video key frames to feature vectors (at the resolution of the updated and upsampled guidance image), and max-pool the features. For large output resolutions, not all of the key frames and corresponding features fit the memory capacity of modern GPUs. Therefore, in such cases, we run the max-pooling and subsequent steps on the CPU.

8. Results and Discussion

8.1. Results

Figure 6 and Figure 7 show a selection of SVBRDFs from the Adobe Stock 3D Material dataset [Ado18] and INRIA SVBRDF dataset [DAD*18] (not part of the training set). For each material, we show the guidance frame (in this case a selected frame from the video sequence) and a few captured frames (after alignment) as well as the recovered and reference SVBRDF property maps and rerenderings from the top view of the guidance image, and from two random view and light directions. Unless noted otherwise, all recovered SVBRDFs and input frames are at 1,024 × 1,024 resolution and reconstructed from 29 key frames. From this we can see that our method works well for such SVBRDFs.

Figure 8 shows a similar visualization of materials captured with an iPhone XR at 1920 × 1080, 30fps in a dark environment with the cell phone's flash light on, with fixed exposure, fixed white balance, and fixed aperture. We crop the center (square) area and resample to a 1024 × 1024 resolution. The mobile phone is moved by hand, without additional aids such as markers, in a spiral pattern. The first frame of the video sequence is used as the guidance image. For comparison, we also capture 8 frames similar to the synthetic validation frames; these are not used for training. While the differences are more visible than in the synthetic case (due to uncertainty in matching the virtual flash light and virtual camera position to the capture parameters), they are still a good match.

A key advantage of our method is that the guidance image does not need to be part of the video sequence, allowing us to capture a larger exemplar at high resolution. Figure 9 showcases such recovered high-resolution SVBRDFs from physical material exemplars. The hand-held camera (iPhone XR) is moved in a zig-zag pattern (to cover a larger portion of the material), and the guidance image is captured from a higher vantage point under uncontrolled environment lighting. Note that none of prior work on multi-image SVBRDF recovery using deep learning is able to capture such a large material at the same level of detail without extrapolation.

The above results show that our method is capable of producing plausible SVBRDFs. To better illustrate the accuracy of our method, we also perform numerical comparisons on the synthetic test set of 195 SVBRDFs from the Adobe Stock 3D Material set and 38 from the INRIA SVBRDF dataset. We validate the accuracy of: the recovered SVBRDF property maps, rerenderings of the materials, and the warping functions. The accuracy of the property maps is computed by taking the average L_1 error over the corresponding property maps. The rerendering accuracy is computed as the L_1 and DSSIM error on: a top view, 8 views from the each of the sides with a view angle at 22.5° and 45° and co-located lighting, and 9 random view/light directions. The accuracy of the alignment is computed by A) the percentage of warped valid pixels (i.e., pixels visible in both guidance and target image) versus the reference warp, and B) the error on the warp (in pixel distance) over the valid predicted pixels. The former indicates how many pixels are valid, while the latter indicates of those valid pixels, how accurate they are. Table 2 (first row) shows the average error over the full test set. From this we can see that our method is able to accurately recover the material properties from reflectance scanning video sequences.

8.2. Comparison

While instructive, the error numbers presented in the previous subsection do not indicate how our method compares to prior work. A direct comparison is difficult due to differences in the acquisition. Therefore, we compare our method to the method of Deschaintre et al. [DAD*19] and Gao et al. [GLD*19] for an equal number (1, 5, 11 and 29) of input photographs optimized for each respective method. We use the provided rendering and inference code and trained network from Deschaintre et al. and Gao et al. for the comparison. Figure 10 and Table 3 show a visual and numerical comparison with the inference based method of Deschaintre et al. [DAD*19] and the optimization based method of Gao et al. [GLD*19]. We can see that for an equal number of input frames, our method on average yields a more accurate specular albedo and roughness compared to both Deschaintre et al. and Gao et al., and a more accurate diffuse albedo compared to Deschaintre et al. On average, our method yields slightly higher surface normal errors due to the reflectance scanning that trades angular coverage for spatial coverage; our input provides less lighting/view directions (i.e., angular coverage) because we only have top-down views, but un-

Table 2: *Ablation study with synthetic video sequences. All SVBRDFs are at* 1,024 × 1,024 *resolution and reconstructed with the automatic alignment.*

SVBRDF maps L1				Render	
Diffuse	Normal	Specular	Roughness	L1	DSSIM
Our method with default parameters					
0.0274	0.0363	0.0255	0.0840	0.0282	0.1676
Without Perlin Noise based specular augmentation					
0.0281	0.0400	0.0281	0.0996	0.0326	0.1833
Without top view rerendering loss L_t					
0.0302	0.0423	0.0272	0.0884	0.0342	0.1962
Instance normalization instead of convolution weight normalization					
0.0445	0.0437	0.0401	0.1317	0.0454	0.2581
Without alignment regularization					
0.0312	0.0461	0.0263	0.1065	0.0329	0.2098
Warping input frames instead of feature maps					
0.0425	0.0544	0.0302	0.1712	0.0437	0.2673

	Alignment	
	Valid rate	Pixel error
With alignment regularization	0.9174	5.623
Without alignment regularization	0.2770	10.683

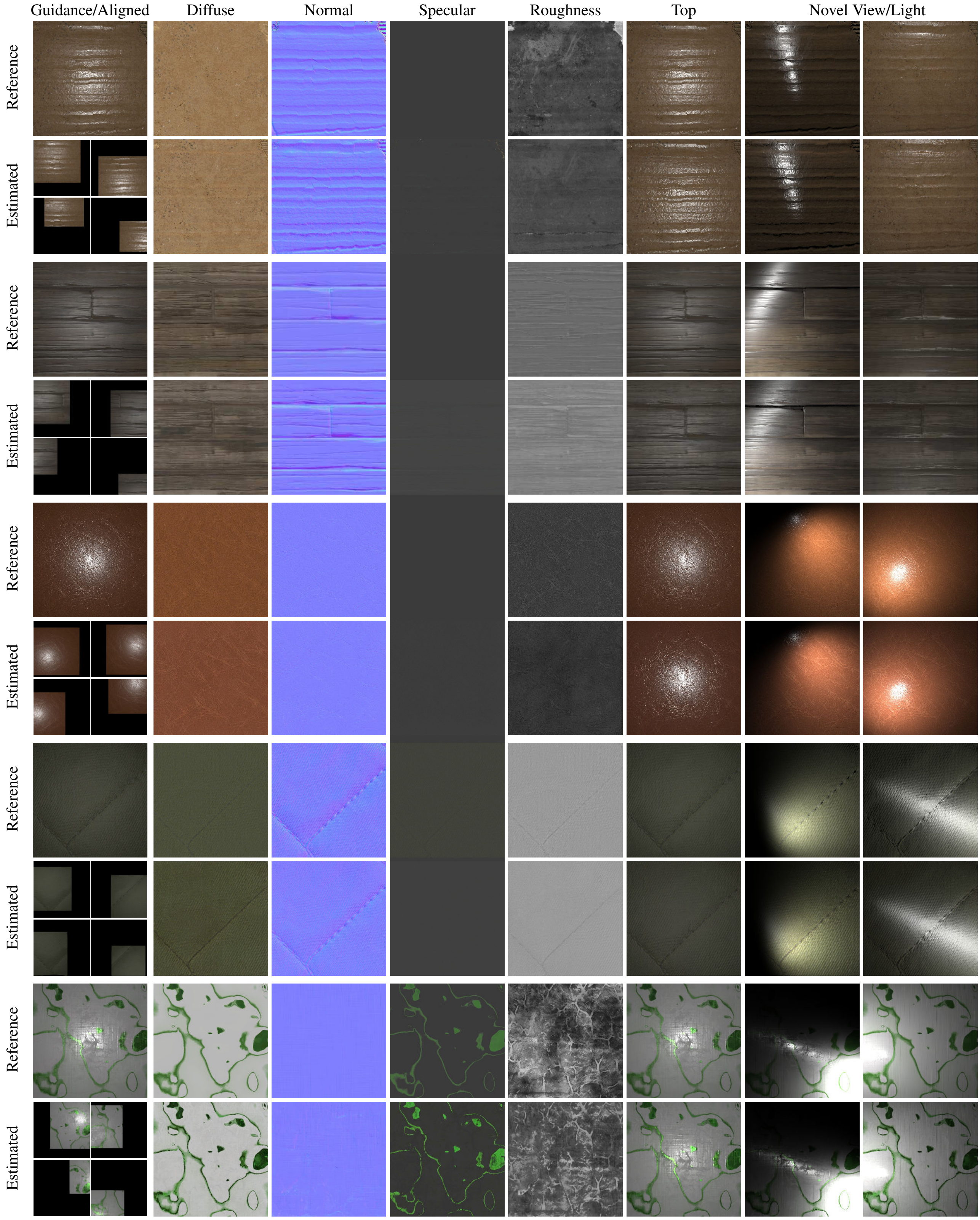


Figure 6: *SVBRDF estimation results from synthetic video sequences. For each SVBRDF we show the guidance image and* 4 *selected aligned frames, the reference and recovered SVBRDF property maps, and a comparison of a top view and two novel view/light direction renderings.*

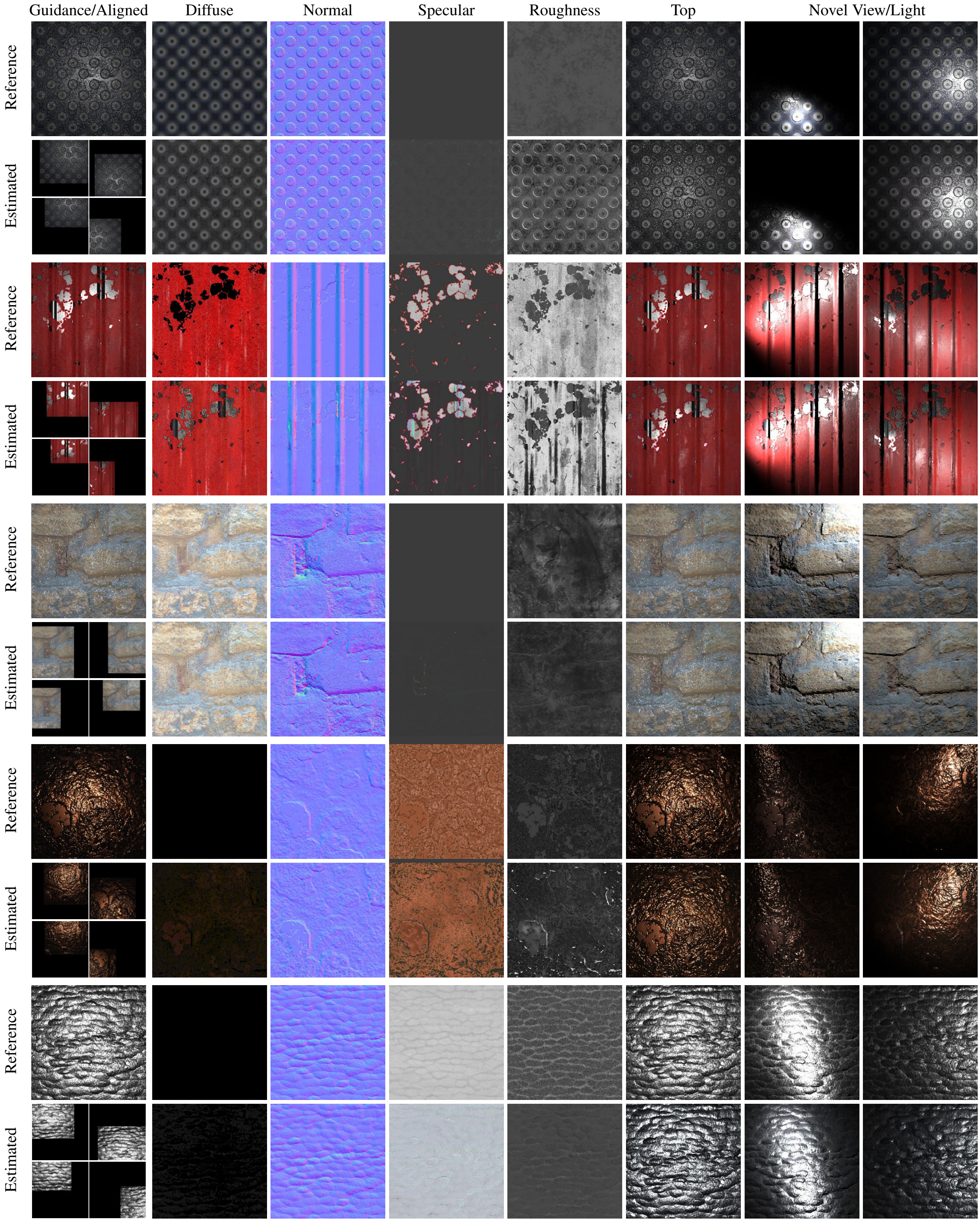


Figure 7: *Additional SVBRDF estimation results from synthetic video sequences. For each SVBRDF we show the guidance image and* 4 *selected aligned frames, the reference and recovered SVBRDF property maps, and a comparison of a top view and two novel view/light direction renderings.*

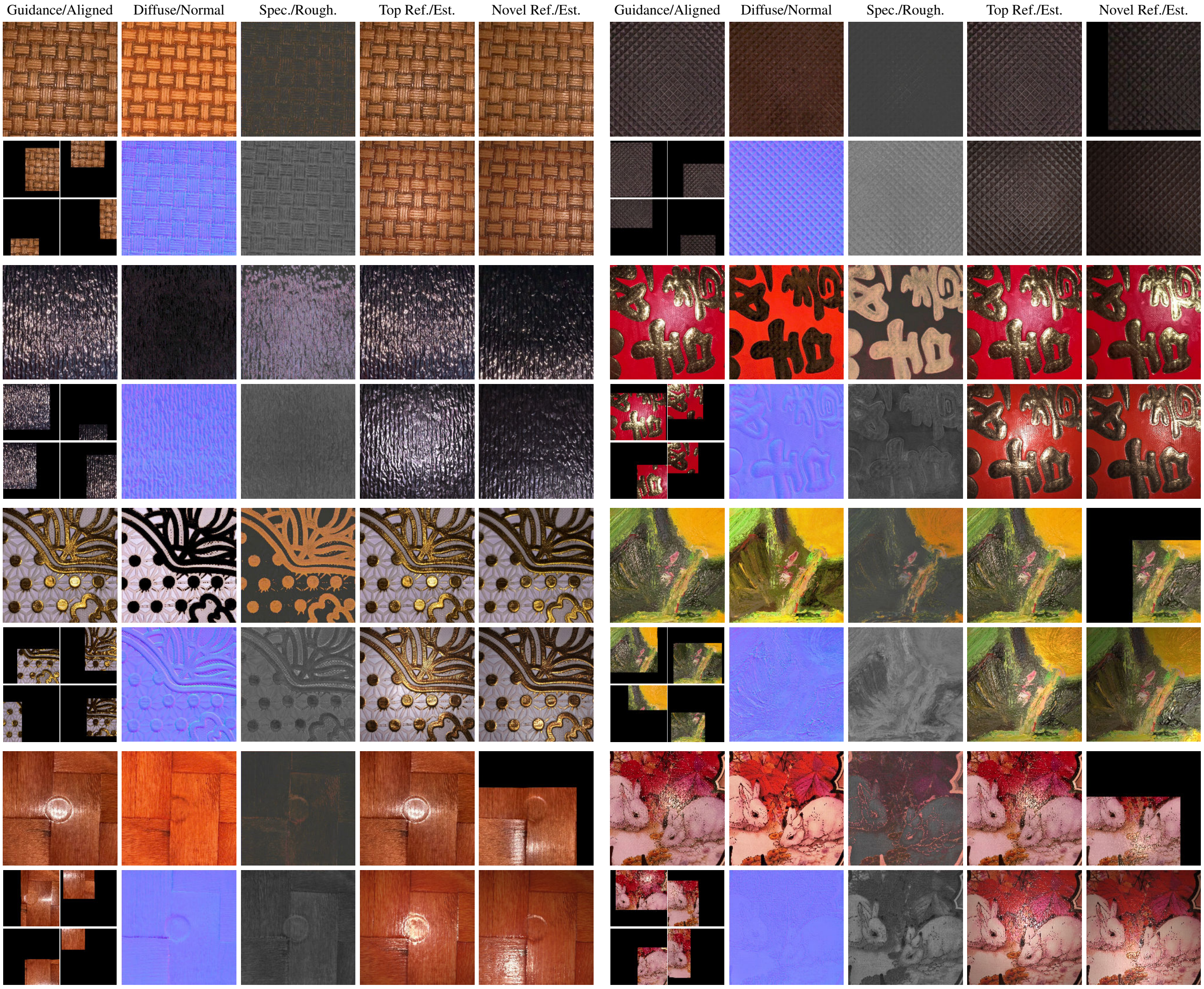


Figure 8: *Recovered SVBRDFs from a hand-held captured video sequence where the guidance image is the first frame in the sequence. For each example, we show the guidance image and 4 selected aligned frames, the material property maps (without a reference), and a comparison between a captured and estimated top view and novel view/light direction. Note that for some novel view/light directions, the reference photograph does not cover the full image after alignment to the guidance image view.*

like Deschaintre et al. and Gao et al., our method does not require the full sample to be visible in each input photograph (i.e., spatial coverage). Our method also produces more accurate revisualizations than Deschaintre et al. under novel view and lighting directions; Gao et al. explicitely optimize visual error and produce even lower revisualization errors at the cost of significantly higher computational and alignment costs. A key difference is that both prior methods vary the viewpoint while keeping the look-at-point fixed, whereas our method varies the viewpoint and look-at-point in tandem. Consequently, our method suffers less from depth-of-field issues compared tilted-camera captures, and whereas the output resolution of both prior methods is limited by the camera resolution, our method aggregates pixels over all scanned surface points, and can therefore recover sharper SVBRDFs with a resolution that exceeds the native camera resolution. To avoid biases due to the enlarged training data set (i.e., the network of Deschaintre et al. is trained without the Adobe Stock 3D Material dataset), we also include a comparison using only the INRIA test data in Table 3. Furthermore, to avoid additional bias due to the differences in the BRDF models between Deschaintre et al. and Gao et al. / our method, we generated the input image and reference with both BRDF models, and compare the reconstruction quality (with the methods' native BRDF model) on both; the difference in revisualization error between both BRDF models is minimal for all methods (Table 3). The conclusions of the numerical results are echoed in the visual comparison in Figure 10.

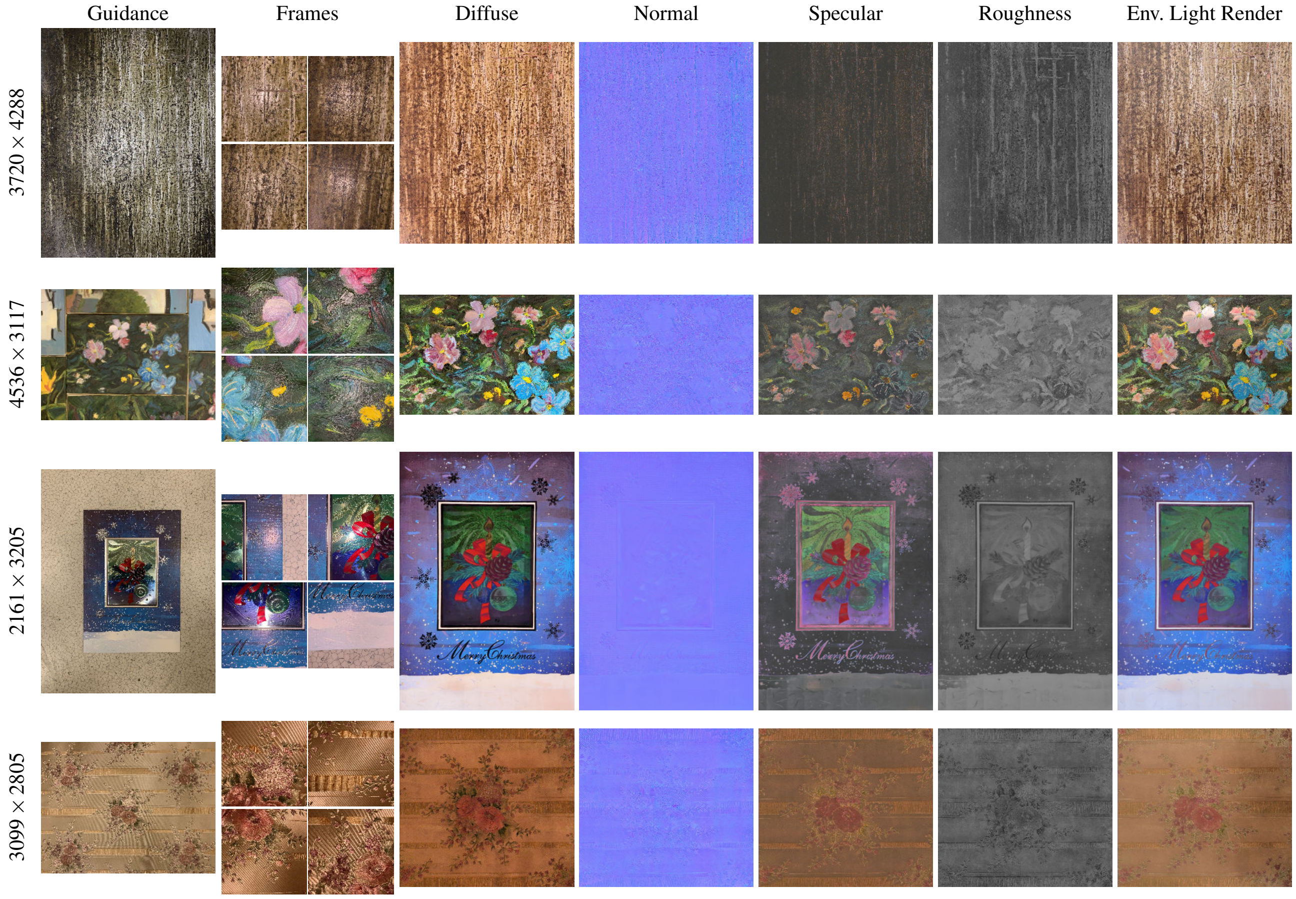


Figure 9: *Hand-held mobile reflectance scanning results for a guidance image captured from a macro-view position under environment lighting recovered at high resolution (first column). The visualization of the SVBRDF is shown under randomly selected environment lighting. Note, the guidance image is slightly larger than the area of interest to aid the alignment at the boundaries.*

Figure 11 and Table 4 show a visual and numerical comparison with the guided upsampling method of Deschaintre et al. [DDB20] which can generate high resolution results from a single high resolution guidance image and SVBRDF exemplars of smaller patches. The numerical comparison is performed on the dataset (including the SVBRDF exemplar patches) provided by Deschaintre et al. [DDB20] on their project web page. We use randomly selected environment lighting to generate the guidance image for our method, and a point light for the method of Deschaintre et al. which is very sensitive to the guidance image lighting; we found that the method of Deschaintre et al. performed not as well when using environment lighting. Compared to the results generated by the method of Deschaintre et al., our method produces more accurate results. Even when using the ground truth SVBRDF patches in the guided upsample method of Deschaintre et al., our method produces more accurate results, with exception for the specular albedo. The visual comparison in Figure 11 confirms the conclusions from the numerical comparison. For example, in the bottom example, the guided upsampling method [DDB20] yields a less correct visualization due to the larger error in the normal map and roughness. Even in the case of ground truth exemplar patches (top example), there are still important details missing such as the orange specular dots.

8.3. Ablation Study

We perform extensive ablation studies to investigate the improvement in the SVBRDF estimation component (the impact of Perlin noise based augmentation on the specular properties and the impact of the additional top view rerendering loss), and the improvement in the alignment component (alignment regularization, and the impact of feature vs frame warping).

Impact of Perlin Noise Augmentation We employ a Perlin noise based augmentation which introduces more structured specular components in the training data to improve the accuracy of the roughness and specular albedo. Figure 12 compares SVBRDF inference networks trained with and without this augmentation. Table 2 (2nd row) lists the per-map property map errors, and the rerendering errors for the SVBRDF estimation network trained with and without the Perlin noise based specular augmentation. The error comparison in this table indicates modest improvements in diffuse albedo and normal maps, but a more significant improvement in the specular albedo and roughness.

Impact of Top view Rerendering Loss L_t In addition to the Perlin noise based specular component augmentation, we also include an additional term in the loss function that measures the accuracy of top-view renderings (from different positions). Leaving out this

Table 3: *Quantitative comparison with synthetic test data at* 256 × 256 *resolution between our deep reflectance scanning, the deep multi-image SVBRDF method of Deschaintre et al. [DAD*19], and the deep inverse rendering method of Gao et al. [GLD*19]. For all methods, the input images follow the methods' prescribed capture setups, and utilize the reference alignments. Furthermore, we also include a comparison using only the test set from Deschaintre et al.*

# Input Image	Method	SVBRDF maps L1				Render with Inria BRDF model		Render with our BRDF model	
		Diffuse	Normal	Specular	Rough.	L1	DSSIM	L1	DSSIM
		Full test set							
1	Refl. Scan	0.0374	0.0545	**0.0246**	**0.1311**	0.0360	0.2432	0.0364	0.2443
	[DAD*19]	0.0957	**0.0525**	0.0314	0.1439	0.0599	0.2591	0.0611	0.2612
	[GLD*19]	**0.0279**	0.0578	0.0490	0.2566	**0.0323**	**0.2073**	**0.0323**	**0.2062**
5	Refl. Scan	0.0302	0.0439	**0.0227**	**0.0840**	0.0297	0.1837	0.0300	0.1847
	[DAD*19]	0.0928	0.0328	0.0317	0.1416	0.0550	0.1696	0.0562	0.1712
	[GLD*19]	**0.0175**	**0.0325**	0.0456	0.1894	**0.0153**	**0.0870**	**0.0147**	**0.0845**
11	Refl. Scan	0.0276	0.0374	**0.0221**	**0.0732**	0.0268	0.1572	0.0272	0.1584
	[DAD*19]	0.0938	0.0289	0.0319	0.1384	0.0547	0.1564	0.0559	0.1578
	[GLD*19]	**0.0136**	**0.0232**	0.0417	0.1694	**0.0106**	**0.0566**	**0.0098**	**0.0540**
29	Refl. Scan	0.0272	0.0352	**0.0216**	**0.0697**	0.0261	0.1476	0.0264	0.1487
	[DAD*19]	0.0959	0.0272	0.0319	0.1345	0.0557	0.1543	0.0569	0.1555
	[GLD*19]	**0.0132**	**0.0223**	0.0385	0.1582	**0.0097**	**0.0514**	**0.0089**	**0.0489**
		Test set from [DAD*19]							
1	Refl. Scan	0.0206	0.0336	**0.0234**	**0.1039**	0.0313	0.1792	0.0319	0.1804
	[DAD*19]	0.0490	**0.0302**	0.0242	0.1041	0.0424	0.1697	0.0433	0.1718
	[GLD*19]	**0.0169**	0.0389	0.0433	0.1932	**0.0289**	**0.1457**	**0.0284**	**0.1450**
5	Refl. Scan	0.0155	0.0252	**0.0205**	**0.0573**	0.0231	0.1129	0.0236	0.1139
	[DAD*19]	0.0457	**0.0189**	0.0270	0.0976	0.0373	0.1122	0.0383	0.1141
	[GLD*19]	**0.0076**	0.0190	0.0348	0.1430	**0.0097**	**0.0422**	**0.0090**	**0.0405**
11	Refl. Scan	0.0144	0.0200	**0.0183**	**0.0459**	0.0205	0.0883	0.0212	0.0901
	[DAD*19]	0.0455	0.0166	0.0308	0.0964	0.0359	0.1013	0.0369	0.1030
	[GLD*19]	**0.0053**	**0.0102**	0.0292	0.1245	**0.0059**	**0.0207**	**0.0051**	**0.0191**
29	Refl. Scan	0.0145	0.0179	**0.0177**	**0.0400**	0.0199	0.0798	0.0206	0.0819
	[DAD*19]	0.0458	0.0153	0.0288	0.0938	0.0354	0.0999	0.0364	0.1013
	[GLD*19]	**0.0048**	**0.0093**	0.0248	0.1086	**0.0048**	**0.0170**	**0.0042**	**0.0155**

Table 4: *Quantitative comparison with synthetic test data at* 2048 × 2048 *resolution between deep reflectance scanning and the guided fine-tuning method of Deschaintre et al. [DDB20]. In both cases, we follow the assumed acquisition procedures. We use the test set from [DDB20] which contains* 24 *SVBRDFs. For our method, we by default use auto alignment, except for the* 5 *materials that exhibit little texture; in this case we use the reference alignment. For [DDB20] we include results with reference SVBRDF patches as input, and also with the SVBRDF patches predicted using the method of Dechaintre et al. [DAD*18].*

Method	SVBRDF maps L1				Render	
	Diffuse	Normal	Specular	Rough.	L1	DSSIM
Refl. Scan	**0.0246**	**0.0504**	0.0664	**0.0969**	**0.0289**	**0.1926**
[DDB20], ref. patch	0.0290	0.0758	**0.0388**	0.1238	0.0420	0.2641
[DDB20], pred. patch	0.0431	0.0831	0.0771	0.1829	0.0517	0.2998

additional term from the loss function (Table 2, 3rd row) affects the diffuse and normal map errors as well as the rerendering errors significantly. The specular albedo and roughness are modestly affected. This indicates that both the Perlin noise based augmentation as well as the additional loss term work in unison to improve different parts of the SVBRDF.

Instance Normalization We employ instance normalization instead of convolution weight normalization as in Deschaintre et al. [DAD*19] to avoid small high value artifacts in the reconstructions. Table 2 (4th row) compares the errors on the property maps and the rerenderings of the test data. From this we can see that instance normalization significantly improves all property maps and rendering errors.

Alignment Regularization The alignment regularization step greatly improves the accuracy of the warp estimation. Figure 13 shows an example of an SVBRDF recovered from a video sequence with and without alignment regularization. Without regularization, long-range drift can introduce misalignments that result in visual errors; most visible in the aligned frames and the normal map. Table 2 (5th row) shows the average L_1 errors on each of the recovered SVBRDF property maps for the test set. We can see a clear increase in error on the diffuse albedo and normals. The error on the specular component is more limited. In terms of error on the alignment (Table 2 bottom), we can observe a significantly lower number of valid pixels, and a larger error on the valid pixels. Because we throw out invalid pixels before max-pooling, the estimated SVBRDF values are estimated from fewer observations, and thus less reliable.

Feature Warping vs. Frame Warping One of our key contributions is that we directly warp features instead of the input frames. To illustrate the importance of this step, we compare our method to a direct modification of the method of Deschaintre et al. where the

Table 5: *Quantitative robustness analysis on synthetic video sequences. All SVBRDFs are at* $1{,}024 \times 1{,}024$ *resolution and reconstructed with the automatic alignment.*

	Alignment		SVBRDF maps L1				Render	
Parameter	Valid rate	Pixel error	Diffuse	Normal	Specular	Roughness	L1	DSSIM
Impact of environment lighting (Parameter indicates the overall environment map brightness)								
0.01	0.9171	5.510	0.0268	0.0363	0.0254	0.0842	0.0286	0.1707
0.02	0.9174	5.356	0.0269	0.0365	0.0257	0.0848	0.0295	0.1747
0.05	0.9173	5.330	0.0296	0.0369	0.0270	0.0856	0.0328	0.1844
0.1	0.9170	5.141	0.0398	0.0374	0.0285	0.0870	0.0398	0.1955
0.2	0.9173	5.110	0.0675	0.0389	0.0306	0.0911	0.0550	0.2207
Impact of frame rate (Parameter indicates frame rate factor)								
4	0.9179	3.442	0.0274	0.0362	0.0253	0.0840	0.0282	0.1667
2	0.9181	3.792	0.0274	0.0363	0.0254	0.0845	0.0282	0.1671
0.5	0.8994	23.216	0.0320	0.0385	0.0255	0.0917	0.0315	0.1918
Camera Zig-zag motion								
	0.9363	5.265	0.0275	0.0366	0.0254	0.0848	0.0282	0.1664
Deviations from template path (Parameter indicates scale factor on parameters in Table 1)								
0	0.9770	4.319	0.0275	0.0361	0.0255	0.0840	0.0282	0.1662
0.25	0.9330	5.971	0.0275	0.0363	0.0253	0.0843	0.0283	0.1678
0.5	0.9250	5.562	0.0275	0.0364	0.0253	0.0841	0.0283	0.1682
2	0.9054	11.039	0.0284	0.0372	0.0255	0.0870	0.0290	0.1753

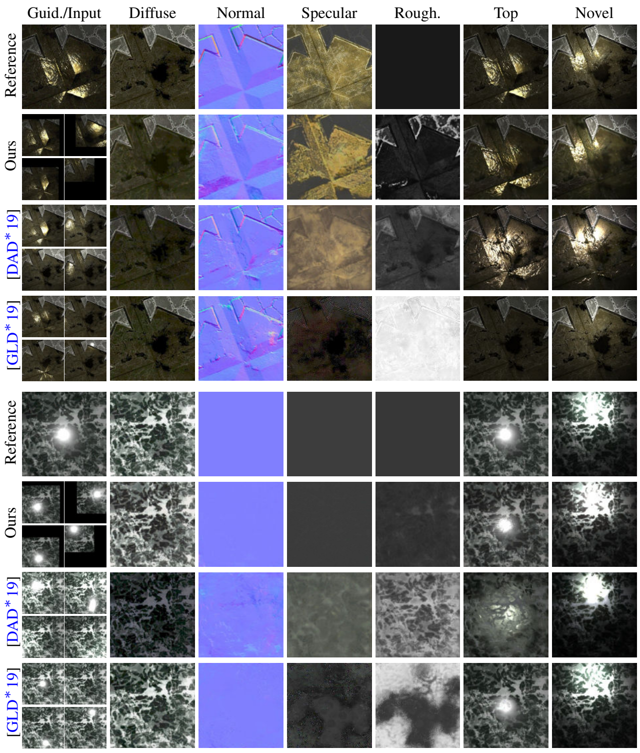
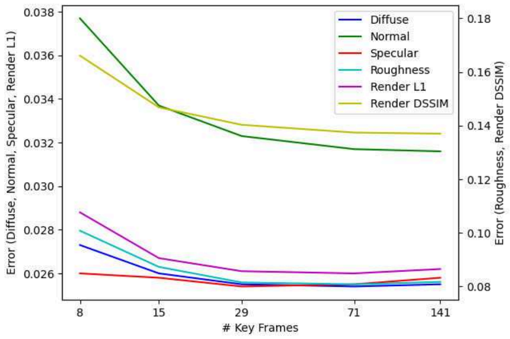


Figure 10: *Comparison with Deschaintre et al. [DAD* 19] and Gao et al. [GLD* 19] on* 256×256 *resolution SVBRDFs with 29 input images following the respective methods' prescribed acquisition procedure.*

Table 6: *Impact of the number of key frames on the SVBRDF reconstruction accuracy at* $1{,}024 \times 1{,}024$ *resolution. To avoid potential bias due to errors in the alignment, we perform this experiment using the reference alignment warps.*

# Key	SVBRDF maps L1				Render	
Frames	Diffuse	Normal	Specular	Roughness	L1	DSSIM
8	0.0273	0.0377	0.0260	0.1008	0.0288	0.1661
15	0.0260	0.0337	0.0258	0.0873	0.0267	0.1468
29	0.0255	0.0323	0.0254	0.0815	0.0261	0.1403
71	0.0254	0.0317	0.0255	0.0808	0.0260	0.1374
141	0.0255	0.0316	0.0258	0.0817	0.0262	0.1370



frames are aligned in the image domain (as opposed to warping the feature vectors). Invalid pixels are set to a -1 value, and we also concatenate a mask for valid pixels to the input. Figure 14 shows for a selected material the recovered SVBRDF as well as the reference SVBRDF, and Table 2 (6th row) lists the respective errors. From this we can see that a direct modification does not work well due to the invalid pixel values.

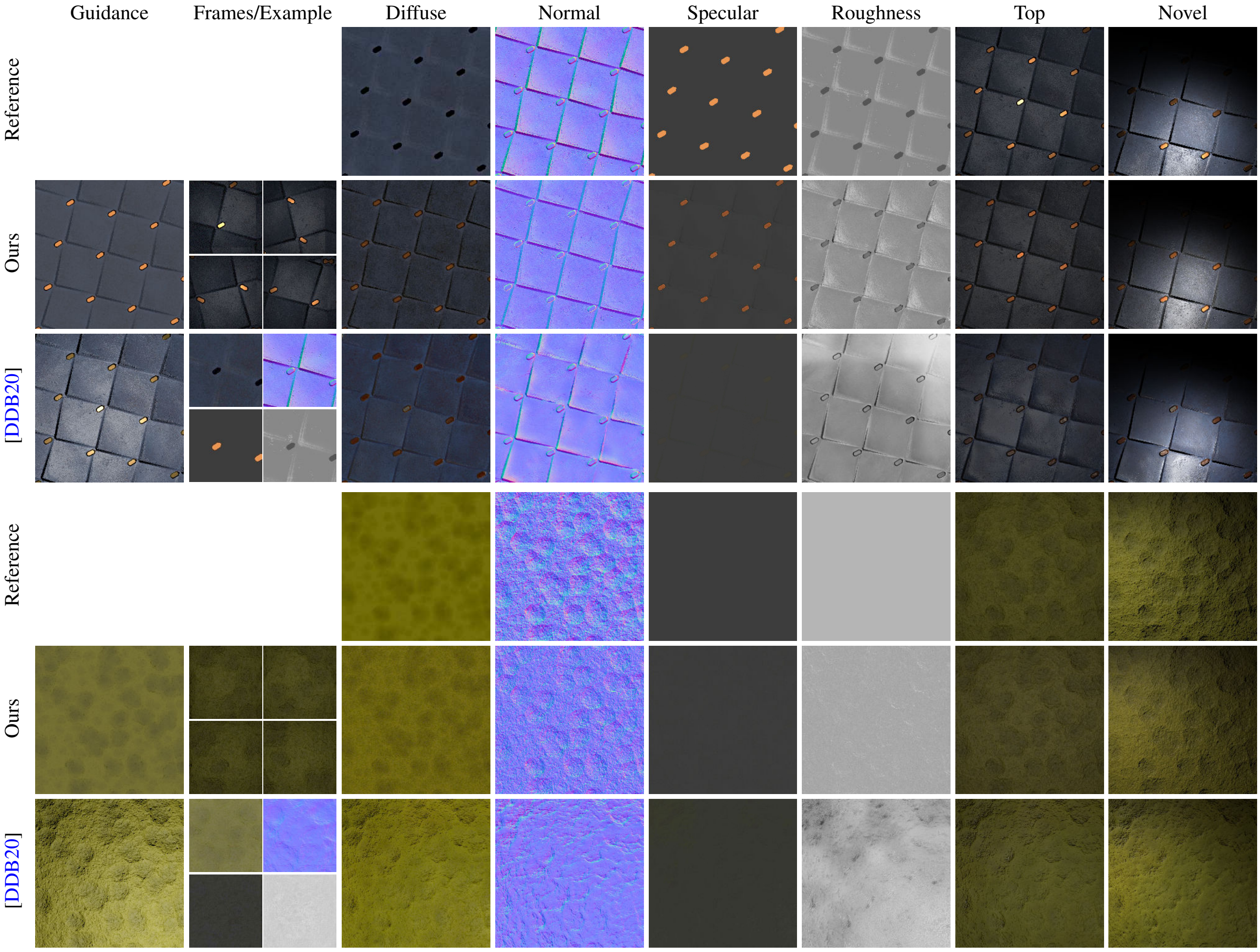


Figure 11: *Comparison with Deschaintre et al. [DDB20] on* 2048 × 2048 *resolution SVBRDFs following the respective methods' assumed acquisition procedure. For robust flow calculation our method uses a slightly larger guidance image, and we crop the corresponding region for comparison. For the guided upsample method of Deschaintre et al. [DDB20], we used the ground-truth SVBRDF patch as input for the top example (best case scenario), and a predicted patch for the bottom example (real-world scenario).*

8.4. Validation

To better understand the many parameters that can potentially impact the quality of the results, we also validate the impact of variations in lighting, number of key frames, and camera motion.

Environment Light In our experiments we assumed that the only source of lighting is the co-located flash lighting. However, in practice, pre-existing environment lighting "pollutes" the measurements. To validate the adverse effect of environment lighting, we include a randomly selected environment light from the dataset of Li et al. [LDPT17] during rendering, and scale it to control the overall contribution. Table 5 plots the alignment, SVBRDF estimation, and rerendering errors for the environment lighting at 1%, 2%, 5%, 10%, and 20% of its total brightness. From this we can see that the alignment is robust to additional environment lighting, as is the estimation of the normal map, and specular properties to some degree. Unsurprisingly, the diffuse albedo is greatly affected as it integrates the incident lighting over the full hemisphere.

Number of Key Frames Increasing the number of key frames effectively provides more cues to the SVBRDF estimation algorithm for recovering the material property maps. Table 6 shows a steady improvement for an increasing number of key frames upto 29, after which the accuracy improves less quickly, and ultimately converges.

Frame Rate In the above experiment, we kept the video frame rate fixed, and just selected more key frames from the same video sequence. However, we can also change the frame rate (or conversely, move the camera more rapidly), while keeping the number of key frames fixed (at 29). Table 5 shows that the alignment accuracy decreases for a decrease in frame rate (i.e., larger difference between subsequent frames). This is not unexpected; optical flow fails for large motions.

Camera Motion We trained our network with two template paths. This raises the question whether our method is robust to other forms of motion.

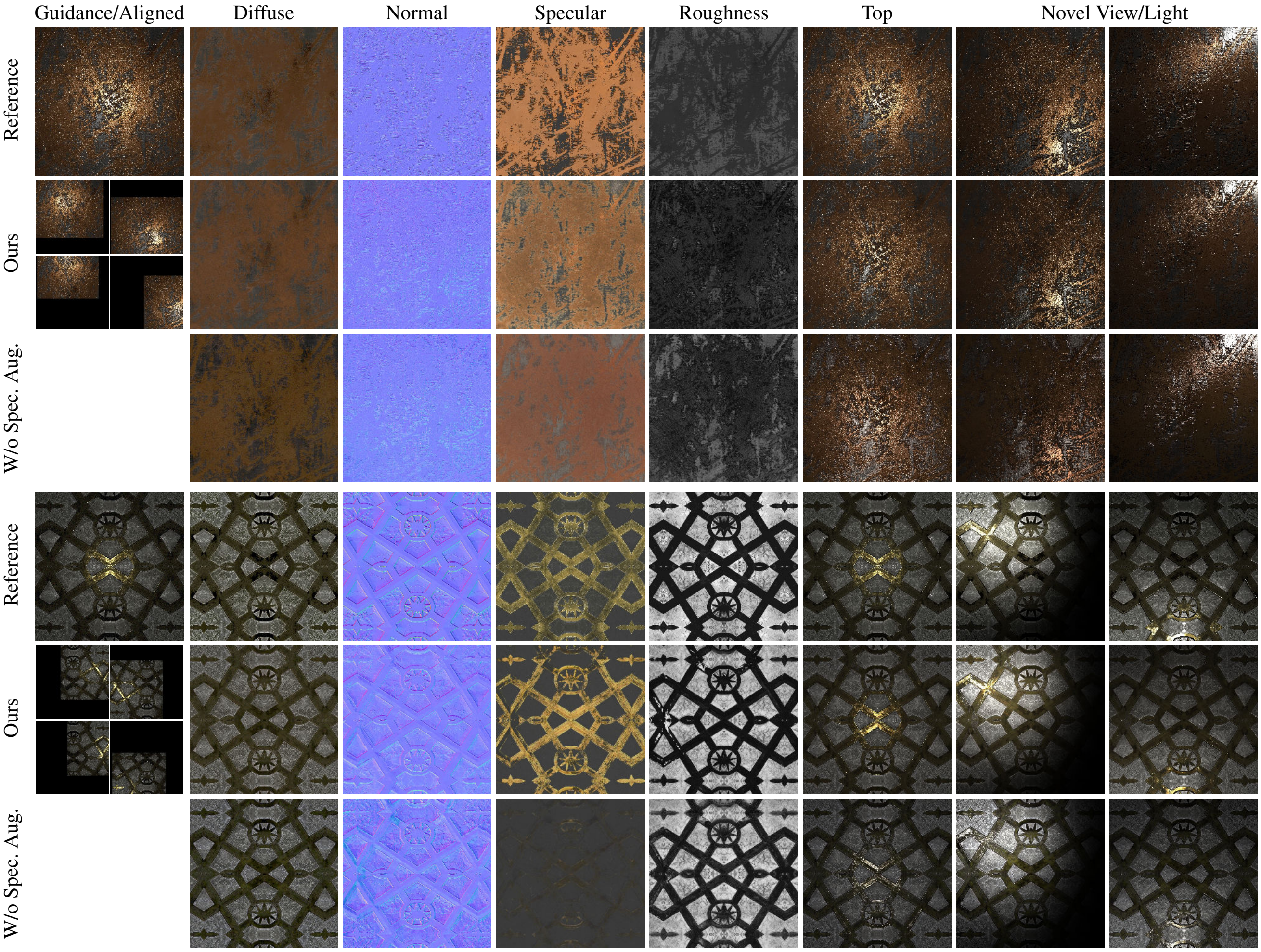


Figure 12: *Ablation examples illustrating the impact of the Perlin noise based augmentation of the training data.*

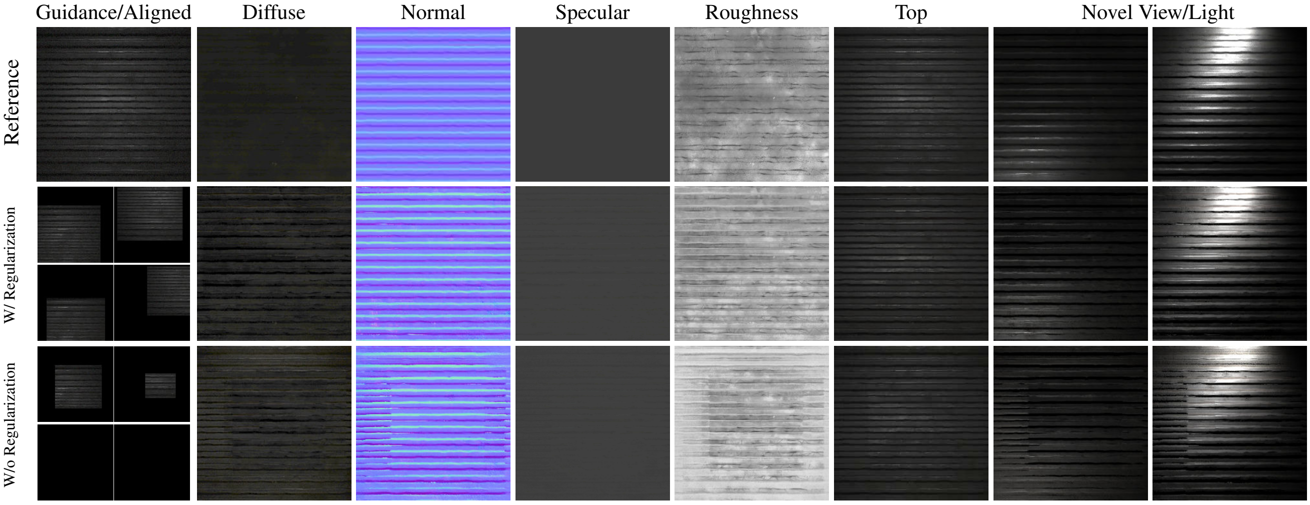


Figure 13: *Ablation examples of the impact of alignment regularization. Without alignment regularization we observe a steady alignment drift over long sequences, and consequently, errors in the SVBRDF reconstruction.*

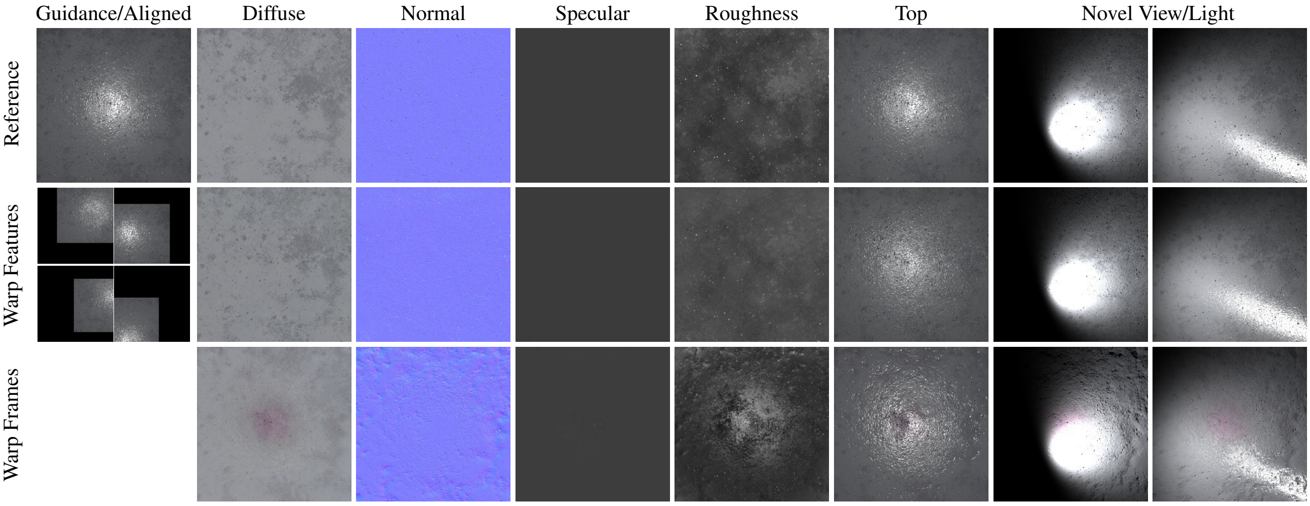


Figure 14: *Ablation example of warping features versus directly warping video frames. Invalid pixels are set to "-1", and we also concatenate a mask for valid pixels to the input. However, this example shows that warping feature maps better handles invalid pixels.*

Table 7: *Robustness analysis of the guidance image's relative scale with respect to the video frames for high resolution SVBRDF reconstruction performed on synthetic video sequences on a high resolution SVBRDF from the dataset of Deschaintre et al. [DDB20]. We reconstruct* 2,048 × 2,048 *SVBRDF maps using our robust alignment estimation for a variety of guidance image scales. The first column reports the relative size of a guidance image pixel compared to a pixel in the video sequence; a larger size means a more blurred guidance image.*

	Alignment		SVBRDF maps L1				Render	
Scale	Valid rate	Pixel error	Diffuse	Normal	Specular	Roughness	L1	DSSIM
1.0	0.9558	8.221	0.0209	0.0413	0.0212	0.0864	0.0251	0.1350
2.0	0.9558	8.219	0.0209	0.0414	0.0212	0.0864	0.0252	0.1356
5.0	0.9557	8.260	0.0211	0.0425	0.0212	0.0864	0.0257	0.1467
10.0	0.9549	8.629	0.0238	0.0485	0.0213	0.0875	0.0292	0.2193

As a first test, we scan the material in a zig-zag motion (since this also covers the full material sample). Despite the significant difference in scanning pattern, we observe that the errors are similar to those from a spiral pattern camera path (Table 5).

As a second test, we evaluate smaller deviations from the template paths, by generating paths with a larger range of variability. Practically, we scale the parameters in Table 1 when generating the test video sequences. Table 5 shows that while increased variability slightly increases the errors on the alignment, the overall error is still too small to significantly impact the SVBRDF estimation.

Both experiments indicate that our alignment algorithm is robust to typical camera motion variations encountered in real-world settings.

Guidance Image Scale A key benefit of our method is that the guidance image does not necessarily need to be at the same resolution as the video frames. To validate this claim, we reconstruct the high resolution SVBRDF from the database of Deschaintre et al. [DDB20] with a varying guidance-pixel to video-pixel ratio. Table 7 shows that our method is robust for a wide range of scales. At a scale ratio of 10 video pixels per guidance pixel, we observe a slight reduction in quality.

8.5. Limitations

Our method is not without limitations. First, our method is sensitive to environment lighting. Including a constant environment light during training as in Deschaintre et al. [DAD*19] could potentially alleviate this. Second, we observe that the benefit of additional frames decreases with increased number of key frames, and eventually levels off. Improving the performance for a large number of input key frames is an interesting avenue for future work. Third, materials with a sharp specular component are difficult to capture manually as it requires a dense sampling of the surface (to elicit a specular response at each surface point). Furthermore, the large dynamic range required to accurately capture the bright specular highlights and the diffuse surface reflectance simultaneously might exceed the dynamic range capabilities of current mobile phones. Fourth, our method is limited to isotropic surface reflectance only. Extending our method to model anisotropic surface reflectance would be an interesting avenue for future research. Finally, our method is somewhat robust to rotations of the camera around the view axis, but for strong rotations our method fails. A key reason for this is that currently each feature vector encodes the local reflectance behavior in a rotation-variant manner. Empirically, we found that training rotation-invariant features poses difficulties for surface normal estimation.

9. Conclusion

In this paper we presented a novel method for recovering spatially-varying surface reflectance of a planar exemplar from a guidance image of the exemplar and a video sequence captured by "scanning" a mobile phone manually over the surface while illuminating the material exemplar by the co-located flash light. Our method requires no calibration or manual alignment of the captured frames, making our method suited for use by non-expert users. The robust automatic alignment allows us to scan SVBRDFs at resolutions exceeding the camera resolution and produce an SVBRDF at the combined resolution of all video frames. Our method builds on prior work in recovering SVBRDFs using deep convolutional neural networks. Key to our method is that the alignment is performed on feature vectors instead of the input frames directly. Our method has the benefit that it can naturally handle uneven coverage of surface points in the target SVBRDF by the input video frames.

For future work we would like to investigate other acquisition protocols such as changing the view angle instead of the camera position as in our work. Additionally, we would like to improve the robustness of the method for a large number of key frames as well as for rotational invariance.

Acknowledgments Pieter Peers was supported in part by NSF grant IIS-1909028.

References

[AAL16] AITTALA M., AILA T., LEHTINEN J.: Reflectance modeling by neural texture synthesis. *ACM Trans. Graph. 35*, 4 (2016). 2

[ACGO18] ALBERT R. A., CHAN D. Y., GOLDMAN D. B., O'BRIEN J. F.: Approximate svbrdf estimation from mobile phone video. In *29th Eurographics Symposium on Rendering, Experimental Ideas & Implementations, EGSR 2018, EI&I Track* (July 2018), pp. 11–22. 2

[Ado18] ADOBE: Adobe stock 3d material dataset. https://stock.adobe.com/3d-assets, 2018. 2, 3, 7

[AWL15] AITTALA M., WEYRICH T., LEHTINEN J.: Two-shot SVBRDF capture for stationary materials. *ACM Trans. Graph. 34*, 4 (2015). 2

[BXS*20a] BI S., XU Z., SUNKAVALLI K., HASAN M., HOLD-GEOFFROY Y., KRIEGMAN D. J., RAMAMOORTHI R.: Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *ECCV* (2020), pp. 294–311. 2

[BXS*20b] BI S., XU Z., SUNKAVALLI K., KRIEGMAN D. J., RAMAMOORTHI R.: Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *CVPR* (2020), pp. 5959–5968. 2

[DAD*18] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph. 37*, 4 (2018). 1, 2, 3, 4, 7, 12

[DAD*19] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Flexible SVBRDF capture with a multi-image deep network. *Comput. Graph. Forum 38*, 4 (2019). 1, 2, 4, 5, 6, 7, 12, 13, 16

[DCP*14] DONG Y., CHEN G., PEERS P., ZHANG J., TONG X.: Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting. *ACM Trans. Graph. 33*, 6 (2014). 2

[DDB20] DESCHAINTRE V., DRETTAKIS G., BOUSSEAU A.: Guided fine-tuning for large-scale material transfer. *Comput. Graph. Forum 39*, 4 (2020), 91–105. 2, 11, 12, 14, 16

[Don19] DONG Y.: Deep appearance modeling: A survey. *Visual Informatics 3*, 2 (2019). 2

[GGG*16] GUARNERA D., GUARNERA G. C., GHOSH A., DENK C., GLENCROSS M.: BRDF representation and acquisition. *Comput. Graph. Forum 35*, 2 (2016), 625–650. 2

[GLD*19] GAO D., LI X., DONG Y., PEERS P., XU K., TONG X.: Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Trans. Graph. 38*, 4 (2019). 1, 2, 7, 12, 13

[GSH*20] GUO Y., SMITH C., HAŠAN M., SUNKAVALLI K., ZHAO S.: Materialgan: Reflectance capture using a generative svbrdf model. *ACM Trans. Graph. 39*, 6 (2020). 2

[HSL*17] HUI Z., SUNKAVALLI K., LEE J., HADAP S., WANG J., SANKARANARAYANAN A. C.: Reflectance capture using univariate sampling of brdfs. In *ICCV* (2017), pp. 5372–5380. 2

[KCW*18] KANG K., CHEN Z., WANG J., ZHOU K., WU H.: Efficient reflectance capture using an autoencoder. *ACM Trans. Graph. 37*, 4 (July 2018). 2

[KLA*20] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and improving the image quality of StyleGAN. In *CVPR* (2020). 2, 6

[KXH*19] KANG K., XIE C., HE C., YI M., GU M., CHEN Z., ZHOU K., WU H.: Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Trans. Graph. 38*, 6 (Nov. 2019). 2

[LDPT17] 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 (2017). 1, 2, 4, 14

[LN16] LOMBARDI S., NISHINO K.: Reflectance and illumination recovery in the wild. *IEEE PAMI 38*, 1 (2016), 129–141. 2

[LPG19] LIN Y., PEERS P., GHOSH A.: On-site example-based material appearance acquisition. *Comput. Graph. Forum 38*, 4 (2019), 15–25. 2

[LSC18] LI Z., SUNKAVALLI K., CHANDRAKER M.: Materials for masses: SVBRDF acquisition with a single mobile phone image. In *ECCV* (2018), pp. 74–90. 1, 2

[LSR*20] LI Z., SHAFIEI M., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR* (2020), pp. 2475–2484. 2

[LXR*18] LI Z., XU Z., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph. 37*, 6 (2018). 1, 2

[Nag17] NAG S.: Image registration techniques: A survey. *CoRR abs/1712.07540* (11 2017). 5

[NLGK18] NAM G., LEE J. H., GUTIERREZ D., KIM M. H.: Practical SVBRDF acquisition of 3d objects with unstructured flash photography. *ACM Trans. Graph. 37*, 6 (2018). 2

[ON16] OXHOLM G., NISHINO K.: Shape and reflectance estimation in the wild. *IEEE PAMI 38*, 2 (2016), 376–389. 2

[Per02] PERLIN K.: Improving noise. *ACM Trans. Graph. 21*, 3 (July 2002), 681–682. 2

[RGS*19] REN Z., GALLO O., SUN D., YANG M.-H., SUDDERTH E. B., KAUTZ J.: A fusion approach for multi-frame optical flow estimation. In *WACV* (2019). 1, 5

[RPG16] RIVIERE J., PEERS P., GHOSH A.: Mobile surface reflectometry. *Comput. Graph. Forum 35*, 1 (2016), 191–202. 2

[RVZ08] ROMEIRO F., VASILYEV Y., ZICKLER T. E.: Passive reflectometry. In *ECCV* (2008), vol. 5305, pp. 859–872. 2

[RWS*11] REN P., WANG J., SNYDER J., TONG X., GUO B.: Pocket reflectometry. *ACM Trans. Graph. 30*, 4 (2011). 2

[RZ10] ROMEIRO F., ZICKLER T. E.: Blind reflectometry. In *ECCV* (2010), pp. 45–58. 2

[Sze06] SZELISKI R.: Image alignment and stitching: A tutorial. *Found. Trends. Comput. Graph. Vis. 2*, 1 (Jan. 2006), 1–104. 5

[WdBKK15] WEINMANN M., DEN BROK D., KRUMPEN S., KLEIN R.: Appearance capture and modeling. In *SIGGRAPH Asia 2015 Courses* (2015). 2

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Rendering Techniques* (2007), p. 195–206. 3

[XDPT16] XIA R., DONG Y., PEERS P., TONG X.: Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Trans. Graph. 35*, 6 (2016). 2

[YLD*18] YE W., LI X., DONG Y., PEERS P., TONG X.: Single image surface appearance modeling with self-augmented cnns and inexact supervision. *Comput. Graph. Forum 37*, 7 (2018), 201–211. 1, 2