Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L Zhang
(Guest Editors)

Volume 39 (2020), Number 7

A Bayesian Inference Framework for Procedural Material Parameter Estimation

Y. Guo[1], M. Hašan[2], L. Yan[3] and S. Zhao[1]

[1]University of California, Irvine, USA [2]Adobe Research, USA [3]University of California, Santa Barbara, USA

Figure 1: *A scene rendered with material parameters estimated using our method: bumpy dielectrics, leather, plaster, wood, brushed metal, and metallic paint. The insets show a few examples of the input (target) images, and renderings produced using our procedural models with parameters found by Bayesian posterior sampling.*

Abstract

Procedural material models have been gaining traction in many applications thanks to their flexibility, compactness, and easy editability. We explore the inverse rendering problem of procedural material parameter estimation from photographs, presenting a unified view of the problem in a Bayesian framework. In addition to computing point estimates of the parameters by optimization, our framework uses a Markov Chain Monte Carlo approach to sample the space of plausible material parameters, providing a collection of plausible matches that a user can choose from, and efficiently handling both discrete and continuous model parameters. To demonstrate the effectiveness of our framework, we fit procedural models of a range of materials—wall plaster, leather, wood, anisotropic brushed metals and layered metallic paints—to both synthetic and real target images.

CCS Concepts

• ***Computing methodologies*** → ***Rendering;***

1. Introduction

Physically accurate simulation of material appearance is an important yet challenging problem, with applications in areas from entertainment to product design and architecture visualization. A key ingredient to photorealistic rendering is high-quality material appearance data. Acquiring such data from physical measurements

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

such as photographs has been an active research topic in computer vision and graphics. Recently, *procedural* material models have been gaining significant traction in the industry (e.g., Substance [Ado19]). In contrast to traditional texture-based spatially varying BRDFs that represent the variation of surface albedo, roughness, and normal vectors as 2D images, procedural models generate such information using a smaller number of user-facing parameters, providing high compactness, easy editability, and automatic seamless tiling.

The estimation of procedural model parameters faces several challenges. First, the procedural generation and physics-based rendering of materials is a complex process with a diverse set of operations, making the relationship between procedural model parameters and properties of the final renderings non-linear and non-trivial. Additionally, designing a suitable loss function (metric) to compare a synthesized image to a target image is not obvious. Finally, given the soft nature of the image matching problem, a single point estimate of the "best" match may be less informative than a collection of plausible matches that a user can choose from.

In this paper, we introduce a new computational framework to estimate the parameters of procedural material models that focuses on these issues. Our framework enjoys high generality by not requiring the procedural model to take any specific form, and supporting any differentiable BRDF models, including anisotropy and layering.

To design the loss function, we consider neural summary functions (embeddings) based on Gram matrices of VGG feature maps [GEB15, GEB16], as well as hand-crafted summary functions (§4). The VGG feature map approach is becoming standard practice in computer vision, and was first introduced to material capture by Aittala et al. [AAL16]; we extend this approach to procedural material estimation.

We make two main contributions. The first contribution is a unified view of the procedural parameter estimation problem in a *Bayesian framework* (§5), precisely defining the posterior distribution of the parameters given the captured data and priors, allowing for both maximization and sampling of the posterior. Four components (priors, procedural material model, rendering operator, summary function) together define our posterior distribution (outlined in Figure 2).

Our second contribution is to introduce a *Bayesian inference* approach capable of drawing samples from the space of plausible material parameters. This provides additional information beyond single point estimates of material parameters (for example, though not limited to, discovering similarity structures in the parameter space). Further, due to an ability to combine multiple Markov-Chain Monte Carlo (MCMC) sampling techniques such as Metropolis-Hasting (MH), Hamiltonian Monte Carlo (HMC), and Metropolis-adjusted Langevin algorithm (MALA), our technique is capable of efficiently handling both discrete and continuous model parameters. Posterior sampling is a well-studied area within statistics and has been used in computer vision and inverse rendering [KKTM15], but to our knowledge, it has not yet been applied to material appearance acquisition.

To demonstrate the effectiveness of our framework, we fit procedural models for a diverse set of materials from standard opaque dielectrics (e.g. plastics, leather, wall paint) to dielectrics with 3D structure (wood) to anisotropic brushed metals and layered metallic paints (see Figure 1, §6, and the supplemental materials).

2. Related Work

We review previous work on material parameter estimation in computer graphics and vision, as well as on Markov-Chain Monte Carlo (MCMC) methods in Bayesian inference.

SVBRDF capture. A large amount of previous work focuses on acquisition of material data from physical measurements. The methods generally observe the material sample with a fixed camera position, and solve for the parameters of a spatially-varying BRDF model such as diffuse albedo, roughness (glossiness) and surface normal. They differ in the number of light patterns required and their type; the patterns used include moving linear light [GTHD03], Gray code patterns [FCMB09] and spherical harmonic illumination [GCP*09]. In these approaches, the model and its optimization are specific to the light patterns and the optical setup of the method, as general non-linear optimization was historically deemed inefficient and not robust enough.

More recently, Aittala et al. [AWL13] captured per-pixel SVBRDF data using Fourier patterns projected using an LCD screen; their algorithm used a fairly general, differentiable forward evaluation model, which was inverted in a maximum a-posteriori (MAP) framework. Later work by Aittala et al. [AWL15, AAL16] found per-pixel parameters of stationary spatially-varying SVBRDFs from two-shot and one-shot flash-lit photographs, respectively. In the latter case, the approach used a neural Gram-matrix texture descriptor based on the texture synthesis and feature transfer work of Gatys [GEB15, GEB16] to compare renderings with similar texture patterns but without pixel alignment. We demonstrate that this descriptor makes an excellent summary function within our framework; in fact, the approach works well in our case, as the procedural nature of the model serves as an additional implicit prior, compared to per-pixel approaches. On the other hand, our forward evaluation process is more complex than Aittala et al., since it also includes the procedural material generation itself.

Recent methods by Deschaintre et al. [DAD*18], Li et al. [LSC18] have been able to capture non-stationary SVBRDFs from a single flash photograph by training an end-to-end deep convolutional network. Gao et al. [GLD*19] introduced an auto-encoder approach, optimizing the appearance match in the latent space. All of these approaches estimate per-pixel parameters of the microfacet model (diffuse albedo, roughness, normal), and are not obviously applicable to estimation of procedural model parameters, nor to more advanced optical models (significant anisotropy, layering or scattering).

Procedural material parameter estimation. Focus on estimating the parameters of procedural models has been relatively rare. The dual-scale glossy parameter estimation work of Wang et al. [WSM11] finds, under step-edge lighting, the parameters of a bumpy surface model consisting of a heightfield constructed from

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

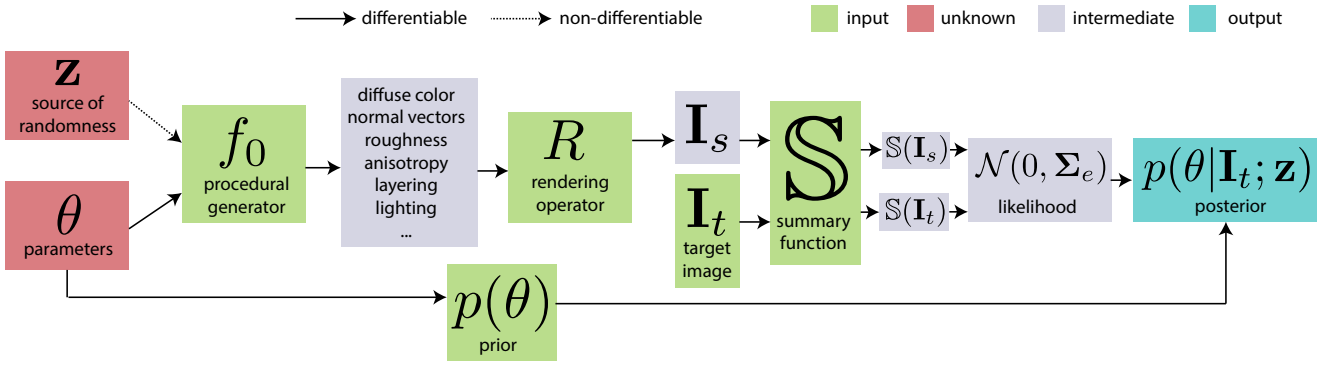


Figure 2: *Our posterior computation combines priors, a procedural material model, a rendering operator, a summary function, and a target image. This posterior distribution can then be sampled to provide plausible values of the parameter vector. The value of the posterior is computed up to a normalization term, which does not effect MCMC sampling. The entire posterior definition is differentiable in the material parameters (excluding optional discrete model parameters).*

a Gaussian noise power spectrum and global microfacet material parameters. Their results provide impressive accuracy, but the solution is highly specialized for this material model and illumination.

Recently, Hu et al. [HDR19] introduced a method for inverse procedural material modeling that treats the material as a black box, and trains a neural network mapping images to parameter vector predictions. The training data comes from evaluating the black box model for random parameters. In our experiments, this approach was less accurate; our fully differentiable models can achieve higher accuracy fits and can be used to explore posterior distributions through sampling. In a sense, this neural prediction method could be seen as orthogonal to ours, as we could use it for initialization of our parameter vector, continuing with our MCMC sampling.

Optical parameters of fiber-based models. Several approaches for rendering of fabrics model the material at the microscopic fiber level [ZJMB11, ZLB16, LWS*18]. However, the optical properties of the fibers (e.g. roughness, scattering albedo) have to be chosen separately to match real examples. Zhao et al. [ZJMB11] use a simple but effective trick of matching the mean and standard deviation (in RGB) of the pixels in a well-chosen area of the target and simulated image. Khungurn et al. [KSZ*15] have extended this approach with a differentiable volumetric renderer, combined with a stochastic gradient descent; however, their method is still specific to fiber-level modeling of cloth.

Bayesian inference and MCMC. A variety of methods used across the sciences are Bayesian in nature; in this paper, we specifically explore Bayesian inference for parameter estimation through Markov-Chain Monte Carlo (MCMC) sampling of the posterior distribution. Provided a nonnegative function f, MCMC techniques can draw samples from the probability density proportional to the given function f without knowing the normalization factor. Metropolis-Hastings [Has70] is one of the most widely used MCMC sampling methods. If f is differentiable, the presence of gradient information leads to more efficient sampling methods such as Hamiltonian Monte Carlo (HMC) [Nea12, Bet17] and Metropolis-adjusted Langevin algorithm (MALA) [RT96]. Our inference framework is not limited to any specific MCMC sampling technique. In practice, our implementation handles discrete model parameters using MH and continuous ones using MALA (with preconditioning [CCG*15]). We opt MALA for its simpler hyperparameter tweaking (compared to HMC).

MCMC applications in graphics and vision. Markov chain Monte Carlo techniques have been heavily studied in rendering, though not for Bayesian inference, but rather for sampling light transport paths with probability proportional to their importance; notably Metropolis light transport [VG97] and its primary sample space variant [KSKAC02]. Much further work has built on these techniques, including more recent work that uses a variant of Hamiltonian Monte Carlo [LLR*15]. However, all of these approaches focus on better sampling for traditional rendering, rather than parameter estimation in inverse rendering tasks.

In computer vision, Bayesian inference with MCMC has been used for the inverse problems of scene understanding. A notable previous work is Picture [KKTM15], a probabilistic system and programming language for scene understanding tasks, for example (though not limited to) human face and body pose estimation. The programming language is essentially used to specify a forward model (e.g., render a face in a given pose), and the system then handles the MCMC sampling of the posterior distribution through a combination of sampling (proposal) techniques. This is closely related to the overall design of our system. However, the Picture system does not appear to be publicly available, and our application is fairly distant from its original goals.

3. Preliminaries

Procedural model generation. We focus on *procedural material models* which utilize specialized procedures (pieces of code) to generate spatially varying surface reflectance information. Specifically, let $\boldsymbol{\theta}$ be the parameters taken by some procedural material generation process f_0. Then, $f_0(\boldsymbol{\theta})$ generates the material proper-

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

ties (e.g., albedo, roughness, surface normals, anisotropy, scattering, etc.), in addition to any other parameters required by rendering (e.g. light parameters), which can in turn be converted into a synthetic image $\boldsymbol{I}_s$ via a rendering operator R. This *forward evaluation* process can be summarized as

$$\boldsymbol{I}_s = R(f_0(\boldsymbol{\theta})) = f(\boldsymbol{\theta}), \tag{1}$$

where f is the composition of R and f_0.

When modeling real-world materials, it is desirable to capture naturally arising irregularities. In procedural modeling, this is usually achieved by making the model generation process f_0 to take extra random input $\boldsymbol{z}$ (e.g., random seeds, pre-generated noise textures, etc.) that is then used to create the irregularities. This also causes the full forward evaluation to become $f(\boldsymbol{\theta};\boldsymbol{z}) := R(f_0(\boldsymbol{\theta};\boldsymbol{z}))$.

Continuous and discrete parameters. While most procedural material parameters tend to be continuous, discrete parameters can be useful for switching certain components on and off, or for choosing between several discrete noise types. We model this by splitting the parameter vector into continuous and discrete components, $\boldsymbol{\theta} = (\boldsymbol{\theta}_c, \boldsymbol{\theta}_d)$. We assume the forward evaluator f to be differentiable with respect to $\boldsymbol{\theta}_c$ (but not $\boldsymbol{\theta}_d$ or the random input $\boldsymbol{z}$).

Inverse problem specification. We consider the problem of inferring procedural model parameters $\boldsymbol{\theta}$ given a target image $\boldsymbol{I}_t$ (which is typically a photograph of a material sample under known illumination). This, essentially, requires inverting f in Eq. (1): $\boldsymbol{\theta} = f^{-1}(\boldsymbol{I}_t)$. Direct inversion of $f = R \circ f_0$ is intractable for any but the simplest material and rendering models. Instead, we aim to identify a collection of plausible values $\boldsymbol{\theta}$ such that $\boldsymbol{I}_s$ has similar appearance to $\boldsymbol{I}_t$:

$$\text{find examples of } \boldsymbol{\theta} \text{ s.t. } \boldsymbol{I}_t \approx f(\boldsymbol{\theta};\boldsymbol{z}), \tag{2}$$

for some (any) $\boldsymbol{z}$, where $\approx$ is an *appearance-match* relation that will be discussed in the next section.

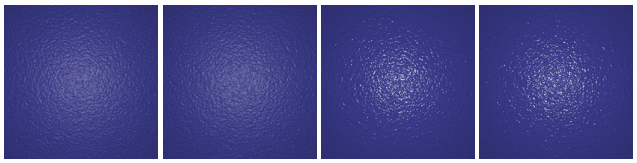
4. Summary Functions

To solve the parameter estimation problem using Eq. (2), a key ingredient is the appearance-match relation. Unfortunately, we cannot use simplistic image difference metrics such as the L2 or L1 norms. This is because the features (bumps, scratches, flakes, yarns, etc.) in the images of real-world materials are generally misaligned, even when the two images represent the same material. In procedural modeling, as shown in Figure 3, with irregularities created differently using $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, the same procedural model parameters $\boldsymbol{\theta}$ can yield slightly different results $f(\boldsymbol{\theta};\boldsymbol{z}_1)$ and $f(\boldsymbol{\theta},\boldsymbol{z}_2)$.

To overcome this challenge, we use the concept of a *summary function*, which abstracts away the unimportant differences in the placement of the features, and summarizes the predicted and target images into smaller vectors whose similarity can be judged with simple metrics like L2 distance.

We define an image summary function $\mathbb{S}$ to be a continuous function that maps an image of a material ($\boldsymbol{I}_t$ or $\boldsymbol{I}_s$) into a vector in $\mathbb{R}^k$. An idealized summary function would have the property that

$$\mathbb{S}(f(\boldsymbol{\theta}_1,\boldsymbol{z}_1)) = \mathbb{S}(f(\boldsymbol{\theta}_2,\boldsymbol{z}_2)) \;\Leftrightarrow\; \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{3}$$



(a1) (a2) (b1) (b2)

Figure 3: *Each pair of images among (a, b) are generated using identical model parameters $\boldsymbol{\theta}$ but different irregularities $\boldsymbol{z}$. The pixel-wise L2 norm of the difference between these image pairs is large and not useful for estimating model parameters.*

That is, applying the summary function would abstract away from the randomness $\boldsymbol{z}$ and the difference between the two summary vectors would be entirely due to different material properties $\boldsymbol{\theta}$. Practical summary functions will satisfy the above only approximately. However, a good practical summary function will embed images of the same material close to each other, and images of different materials further away from each other. Below we discuss several techniques for constructing summary functions.

Neural summary function. Gatys et al. [GEB15, GEB16] introduced the idea of using the features of an image classification neural network (usually VGG [SZ15]) as a descriptor T_G of image texture (or style). Optimizing images to minimize the difference in T_G (combined with other constraints) allowed Gatys et al. to produce impressive, state-of-the art results for parametric texture synthesis and style transfer between images. While further work has introduced improvements [RWB17], we find that the original version from Gatys et al. works already well in our case.

Aittala et al. [AAL16] introduced this technique to capturing material parameter textures (albedo, roughness, normal and specular maps) of stationary materials. They optimized for a 256×256 stationary patch that matches the target image in various crops, using a combination of T_G and a number of special Fourier-domain priors. In our case (for procedural materials), we find that the neural summary function T_G works even more effectively; we can simply apply it to the entire target or simulated images (not requiring crops nor Fourier-domain priors). Specifically, define the Gram matrix G of a set of feature maps $F_1, \cdots, F_n$ such that it has elements

$$G_{ij} = \text{mean}(F_i \cdot F_j), \tag{4}$$

where the product $F_i \cdot F_j$ is element-wise. T_G is defined as the concatenation of the flattened Gram matrices computed for the feature maps before each pooling operation in VGG19. Note that the size of the Gram matrices depends on the number of feature maps (channels), not their size; thus T_G is independent of input image size.

Statistics and Fourier transforms of image bins. While the neural summary function performs quite well, we find that in some cases we can improve upon it. A simple idea for a summary function is to use the (RGB) mean of the entire image; an improvement is to subdivide the image into k bins (regions) and compute the mean of each region. We found concentric bins perform well for isotropic materials, and vertical bins are appropriate for anisotropic highlights (e.g. brushed metal). Furthermore, we can additionally

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

use a fast Fourier transform of the entire image or within bins. Note that automatic computation of derivatives is possible with the FFT, and supported by the PyTorch framework. In our current results, we use a summary function that combines the means and 1D FFTs of 64 vertical bins for the brushed metal example; all other examples use the neural summary function combined with simple image mean.

5. Bayesian Inference of Material Parameters

In what follows, we first describe a Bayesian formulation of the estimation problem in terms of a posterior distribution. Next, we discuss how to use the posterior for point estimation in a maximum a-posteriori (MAP) framework, and how the Markov-Chain Monte Carlo (MCMC) methods for Bayesian inference extend the point estimate approach by sampling from the posterior.

5.1. Bayesian formulation

We treat the procedural model parameters $\boldsymbol{\theta}$ as random variables with corresponding probability distributions.

We first introduce a *prior* probability distribution $p(\boldsymbol{\theta})$ of the parameters, reflecting our pre-existing beliefs about the likelihood values of the unknown parameters. For example, for most material categories, we know what range the albedo color and roughness coefficients of the material should typically be in.

Further, we model the $\approx$ operator from Eq. (2) as an error distribution. Specifically, we postulate that the difference between the simulated image summary $\mathbb{S}(f(\boldsymbol{\theta},\boldsymbol{z}))$ and the target image summary $\mathbb{S}(\boldsymbol{I}_t)$ follows a known probability distribution. In practice, we use a (multi-variate) normal distribution with zero mean and the covariance $\boldsymbol{\Sigma}_e$:

$$\mathbb{S}(f(\boldsymbol{\theta},\boldsymbol{z})) - \mathbb{S}(\boldsymbol{I}_t) \sim \mathcal{N}(0,\boldsymbol{\Sigma}_e). \tag{5}$$

Our experiments indicate that this simple error distribution works well in practice, and we regard $\boldsymbol{\Sigma}_e$ as a hyper-parameter and set it manually.

We also have multiple options in handling the random vector $\boldsymbol{z}$. While it is certainly theoretically possible to estimate it, we are not really interested in its values; we find it simpler and more efficient to simply choose $\boldsymbol{z}$ randomly, fix it, and assume it known during the process of estimating the "interesting" parameters $\boldsymbol{\theta}$.

Under these assumptions, according to the Bayes theorem, we can write down the posterior probability of parameters $\boldsymbol{\theta}$, conditional on the known values of $\boldsymbol{I}_t$ and $\boldsymbol{z}$, as:

$$p(\boldsymbol{\theta}|\boldsymbol{I}_t,\boldsymbol{z}) \propto \mathcal{N}\left[\mathbb{S}(f(\boldsymbol{\theta},\boldsymbol{z})) - \mathbb{S}(\boldsymbol{I}_t); 0, \boldsymbol{\Sigma}_e\right] p(\boldsymbol{\theta}), \tag{6}$$

where the right side does not need to be normalized; the constant factor has no effect on parameter estimates. For numerical stability, we compute the negative log posterior, viz. $-\log p(\boldsymbol{\theta}|\boldsymbol{I}_t,\boldsymbol{z})$, in practice. Equation (6) also holds when $\boldsymbol{\theta}$ involves discrete parameters, as long as the prior is properly defined as a product of a continuous pdf $p(\boldsymbol{\theta}_c)$ and a discrete probability $p(\boldsymbol{\theta}_d)$.

ALGORITHM 1: MCMC sampling of material parameters $(\boldsymbol{\theta}_c, \boldsymbol{\theta}_d)$

```
samplePosterior(N, α, θ_c^(1), θ_d^(1))
  Input: Sample count N, probability α for sampling continuous
         parameters, initial continuous parameters θ_c^(1) and discrete
         ones θ_d^(1)
  Output: N material parameter estimates {(θ_c^(t), θ_d^(t)) : 1 ≤ t ≤ N}
begin
    for t = 1 to (N − 1) do
        Draw ξ ~ U[0, 1)
        if ξ < α then                          // Mutate continuous parameters
            θ'_c ← ContinuousSample(θ_c^(t))
            θ'_d ← θ_d^(t)
        else                                   // Mutate discrete parameters
            θ'_c ← θ_c^(t)
            θ'_d ← DiscreteSample(θ_d^(t))
        end
        (θ_c^(t+1), θ_d^(t+1)) ← MetropolisHasting((θ'_c, θ'_d), (θ_c^(t), θ_d^(t)))
    end
    return {(θ_c^(1), θ_d^(1)), ..., (θ_c^(N), θ_d^(N))}
end
```

5.2. Point estimate of parameter values

A point estimate of the parameter vector can be modeled in the maximum a-posteriori (MAP) framework. We simply estimate the desired parameter values $\boldsymbol{\theta}$ as the maximum of the posterior pdf $p(\boldsymbol{\theta}|\boldsymbol{I}_t,\boldsymbol{z})$ given by Eq. (6). For continuous $\boldsymbol{\theta}$, this problem can be solved using standard non-linear optimization algorithms. In the presence of discrete parameters, there is no single accepted solution. While various heuristics could be used, our sampling approach described below provides a cleaner solution to discrete parameter estimation.

5.3. Markov-Chain Monte Carlo Sampling of the Posterior

Although the point estimate approach gives satisfactory results in many cases, it is not without problems. For example, since a perfect match between a procedural material and a photograph is generally impossible, it can be desirable to have a set of imperfect matches for the user to choose from. Further, there could be an entire subset of the parameter space giving solutions of approximately equivalent fit under the target view and lighting; however, these may look quite different from each other in other configurations, and a user may want to explore those differences. Lastly, with the presence of discrete parameters, it is not obvious how to solve the maximization in Eq. (6) efficiently.

In this paper, we use the well-known technique of full Bayesian inference, sampling the posterior pdf defined in Eq. (6) using Markov-Chain Monte Carlo (MCMC) techniques, specifically Metropolis-Hasting (MH) [Has70], Hamiltonian Monte Carlo (HMC) [Bet17], and Metropolis-adjusted Langevin algorithm (MALA) [RT96]. While well explored in statistics and various scientific fields, to our knowledge, this technique has not been used for the inference of material parameters.

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

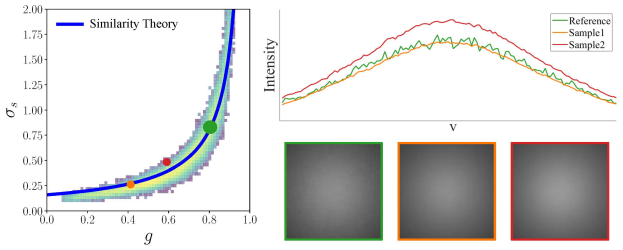


Figure 4: *A motivating example of a scattering material with two estimated parameters (scattering coefficient and phase function parameter). The posterior distribution sampled with our method for three synthetic input images is able to detect the full structure of the parameter space, matching the predictions from similarity theory.*

The goal of the sampling is to explore the posterior with many (typically thousands or more) samples, each of which represents a material parameter vector consistent with the target image. Plotting these samples projected into two dimensions (for a given pair of parameters) gives valuable insight into similarity structures. Furthermore, interactively clicking on samples and observing the predicted result can help a user to quickly zoom in on a preferred solution, which an automatic optimization algorithm is fundamentally incapable of.

Algorithm 1 summarizes our MCMC sampling process. At each iteration, we mutate either the continuous parameters (with probability α) or the discrete ones (with probability $1-\alpha$). For the former case, we utilize the gradient of the log pdf with respect to $\boldsymbol{\theta}_c$ to efficiently obtain a new proposal $\boldsymbol{\theta}_c'$ (Line 6). Our implementation uses MALA for this process, although HMC could also work. For the latter case, we obtain a new proposal $\boldsymbol{\theta}_d'$ of the discrete parameters, currently by uniformly sampling their joint probability mass function (Line 10). Upon obtaining a full proposal, we use the standard Metropolis-Hasting rule (Line 12) to stochastically select the new sample $(\boldsymbol{\theta}_c^{(t+1)}, \boldsymbol{\theta}_d^{(t+1)})$ by either accepting the newly proposed $(\boldsymbol{\theta}_c', \boldsymbol{\theta}_d')$ or (rejecting the proposal and) keeping the previous sample $(\boldsymbol{\theta}_c^{(t)}, \boldsymbol{\theta}_d^{(t)})$.

6. Material Models and Results

We now demonstrate the effectiveness of our technique by fitting several procedural material models to a mix of synthetic and real target images.

Our forward evaluation process uses collocated camera and light. This configuration closely matches a mobile phone camera with flash (which is used for most of the real target images) and simplifies some BRDF formulations (because the incoming, outgoing, and half-way vectors are all identical). Further, we assume that the distance between camera and sample is known as it is generally easy to measure or estimate. The knowledge of the camera field of view allows us to compute the physical scale of the resulting pixels. Lastly, we treat light intensity and camera vignetting (expressed as an image-space Gaussian function) as (unknown) parameters of the forward evaluation process so that they do not need to be calibrated.

Table 1: Performance statistics for our MCMC-based posterior sampling. The numbers are collected using a workstation equipped with an Intel i7-6800K six-core CPU and an Nvidia GTX 1080 GPU.

Material	bump	leather	plaster	flakes	metal	wood
# params.	8	12	11	13	10	23
MCMC (1k iter.)	180s	194s	190s	187s	182s	290s

Our parameter inference framework presented in §4 and §5 is not limited to this specific setup.

All the procedural material models we used, which will be detailed in §6.2, are implemented using PyTorch which automatically provides GPU acceleration and computes derivatives through back-propagation. For all material parameter inference tasks, our forward evaluation generates 512×512 images. Notice that the recovered parameters can then be used to generate results with much higher resolution because the procedural models are generally resolution-independent.

6.1. Similarity Relations in Translucency

As a motivating example, we first illustrate the behavior of the MCMC material parameter estimation process on the case of a homogeneous translucent material with two varying parameters. In this example, the shape of the posterior can be analytically derived (using the similarity theory) and easily plotted. This serves as a demonstration and validation of our approach.

Specifically, the material parameter space of translucent materials under the radiative transfer framework [Cha60] is known to be approximately over-complete [ZRB14]. Specifically, two sets of parameters (σ_s, σ_a, g) and $(\sigma_s^*, \sigma_a^*, g^*)$ satisfying the following *similarity relation* usually yield similar final appearances:

$$\sigma_a = \sigma_a^*, \quad (1-g)\,\sigma_s = (1-g^*)\,\sigma_s^*, \tag{7}$$

where σ_a and σ_s are, respectively, the absorption and scattering coefficients, and g is the first Legendre moment of the phase function. We show in Figure 4 that applying our Bayesian inference method to σ_s and g (with fixed σ_a) computes a posterior distribution that agrees well with the predicted similarity relation (7).

6.2. Procedural Material Models

We show results generated using synthetic images in Figures 5 and 6 as well as real photographs (taken with different cameras) in Figure 7. Please see the supplemental material for more results, including animations illustrating point estimations and sampling. Below we describe the six procedural models tested. Please refer to the supplement for additional detail and a PyTorch implementation. For each parameter, we define a reasonable truncated Gaussian distribution as its prior (also see supplement). In most cases, the MCMC sampling starts from the peak of the prior. In some examples (e.g wood), we first run posterior maximization and then switch to sampling from the optimized point. We drop some number (typically 200 to 1000) of initial MCMC samples due to burn-in.

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

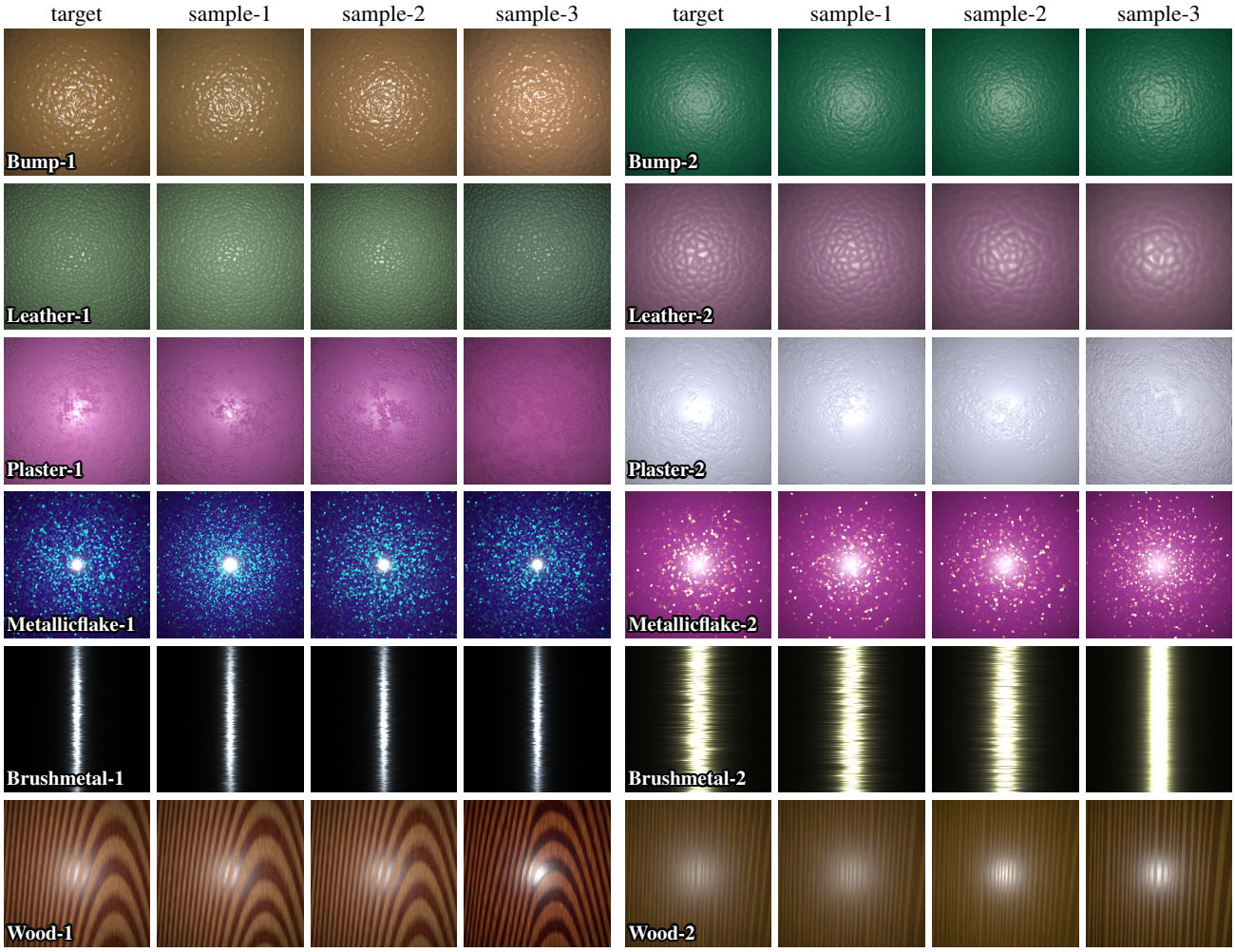


Figure 5: ***Results*** *of our MCMC sampling on* ***synthetic*** *inputs. Each row corresponds to two examples of a different material model. For each example, the first column is the synthetic target image. We show MCMC samples in the other columns, where sample-1 and sample-2 are chosen closer to the peak of the posterior distribution, and sample-3 is further away. More results please refer to supplemental materials.*

Bumpy microfacet surface. This model depicts an opaque dielectric surface with an isotropic noise heightfield. We use a standard microfacet BRDF with the GGX normal distribution [WMLT07] combined with a normal map computed from an explicitly constructed heightfield. We assume that the Fresnel reflectance at normal incidence can be computed from a known index of refraction (a value of 1.5 is a good estimate for plastics). We assume an unknown roughness r (GGX parameter $\alpha = r^2$) and a Lambertian diffuse term with unknown albedo ρ. This model is identical to Wang et al. [WSM11], except using the GGX instead of Beckmann microfacet distribution. The main practical difference from the capture setup in that paper is that we use a point light, instead of step-edge illumination.

The bumpy heightfield is constructed using an inverse Fourier process including: (i) choosing a power spectrum in the continuous Fourier domain; (ii) discretizing it onto a grid of complex numbers; (iii) randomly choosing the phase of each texel on the grid (while keeping the chosen amplitude); and (iv) applying an inverse fast Fourier transform whose real component becomes the resulting heightfield. At render time, we use the normal map derived from this heightfield.

Leather and plaster. These materials can be modeled similarly as the aforementioned bumpy surfaces except for the computation of the heightfield and roughness. For plaster, a fractal noise texture is scaled (in space and intensity) and thresholded (controlled by additional parameters) to produce both the heightfield and a roughness variation texture. For leather, on the contrary, a Voronoi cell map is used to get the effect of leather-like cells (with parameters for scaling and thresholding), and additional small-scale fractal noise is added. Further, we use multiple (pre-generated) noise textures and Voronoi cell maps to diversify the micro-scale appearances that our models can produce. The choice of these textures and maps is captured using a discrete parameter. In Figure 6, we show a few

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

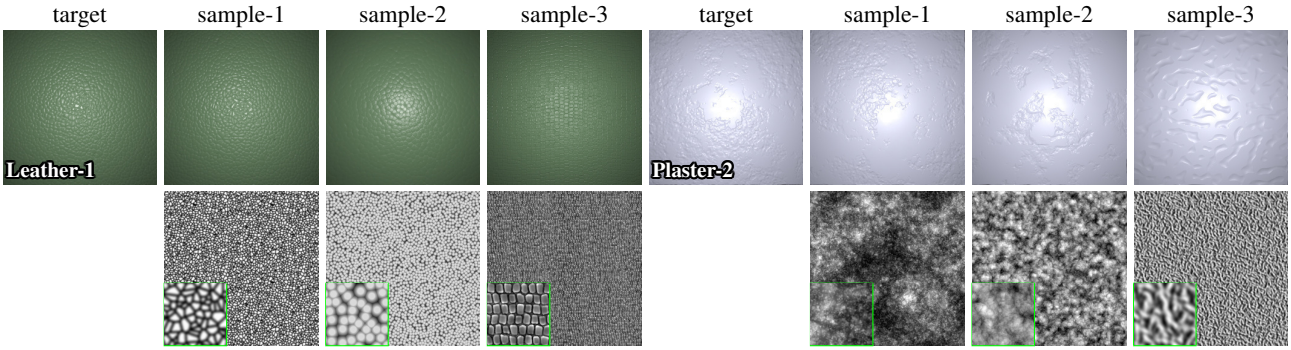


Figure 6: ***MCMC sampling with discrete parameters.*** *In these examples, we illustrate the ability of our sampling to handle discrete parameters. In both examples, one noise inputs used in the procedural model can be switched between several different types of noise. Out of the thousands of sampled solutions, we pick three that have different settings of the discrete parameter where the (log) pdf values decrease from sample-1 to sample-3.*

example samples drawn from the posterior distributions using Algorithm 1.

Brushed metal. The brushed metal material extends the above bumpy surface, by introducing anisotropy to both the GGX normal distribution and the noise heightfield used to compute the normal map, while dropping the diffuse term. We make both the BRDF and the Fourier-domain Gaussian power spectrum anisotropic. The parameters of the model thus include two roughnesses, as well as two Fourier-domain standard deviations. We make the anisotropic highlight vertical and centered in the target image.

Metallic flakes. Metallic paint with flakes is a stochastic material with multiple BRDF lobes (caused by light reflecting off the flakes). Our model involves three components, each being an isotropic microfacet lobe, to describe top coating, flakes and glow, respectively. The top coating is usually highly specular, and we make its roughness a model parameter. We assume an index of refraction of 1.5, implying a Fresnel (Schlick) reflectivity at normal incidence of 0.04. The flakes are chosen as Voronoi cells of a random blue-noise point distribution; they have a roughness parameter and varying normals chosen from the Beckmann distribution with an unknown roughness, and with unknown Fresnel reflectivity. The scale of the cell map is itself a (differentiable) parameter. Lastly, the glow is a component approximating the internal scattering between the top interface and the flakes, and has its own roughness, Fresnel reflectivity and a flat normal. An extra weight parameter linearly combines the flakes and the glow.

Wood. We also created a partial PyTorch implementation of the comprehensive 3D wood model of Liu et al. [LDHM16]. This material is a 3D model of the growth rings of a tree, with a number of parameters controlling colors and widths of growth rings, as well as global distortions and small-scale noise features. The 3D wood is finally projected by a cutting plane to image space, defining diffuse albedo, roughness and height.

Mismatched models. Lastly, we demonstrate in Figure 8 the impact of forward procedural models. Since these model-generating procedures are essentially material-specific priors, using mismatched models generally leads to results that match overall image statistics but with incorrect patterns.

6.3. Additional Comparisons

Comparison to Aittala et al. We first compare our technique to with one introduced by Aittala et al. [AAL16] in Figure 9 using input photos published as supplemental materials from their work. Their work uses the same VGG-based loss (summary function), but optimizes directly in texture space. Both methods manage to reproduce the overall pattern and reflectance of the input photos. Thanks to the underlying procedural models, our method is able to synthesize larger results without visually obvious periodic patterns, and with more plausible global variation.

Comparison to neural methods. We also compare our method with the forward neural prediction method of Hu et al. [HDR19]. Their method uses an AlexNet network structure [KSH12], mapping an image of a material sample to the parameters of an appropriate procedural model. We apply their network structure with our BRDFs and lighting conditions, as their original implementation assumes Lambertian materials and outdoor sun/sky lighting. We show the results in Figure 10. In general, we find the method gives moderately accurate results, which moreover tends to become worse for more complex BRDF models and with more parameters. The photo (top) is better matched by our MCMC sampling results (middle) than their prediction (bottom),

To some extent, the method of [HDR19] is orthogonal to ours, as it can be used as an efficient initialization for our sampling. In Figure 11, we compare our MCMC sampling results with a random starting point to one using the result of Hu et al. for initialization. This reduces the burn-in period required by the MCMC method.

Finally, we also compare to the single input SVBRDF estimation method of Deschaintre et al. [DAD*18] (See Figure 13). This method takes a 256×256 target image, and produces material maps at the same resolution, pixel-wise aligned to the input. This pixel-wise alignment is not achievable with our method (or any proce-

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

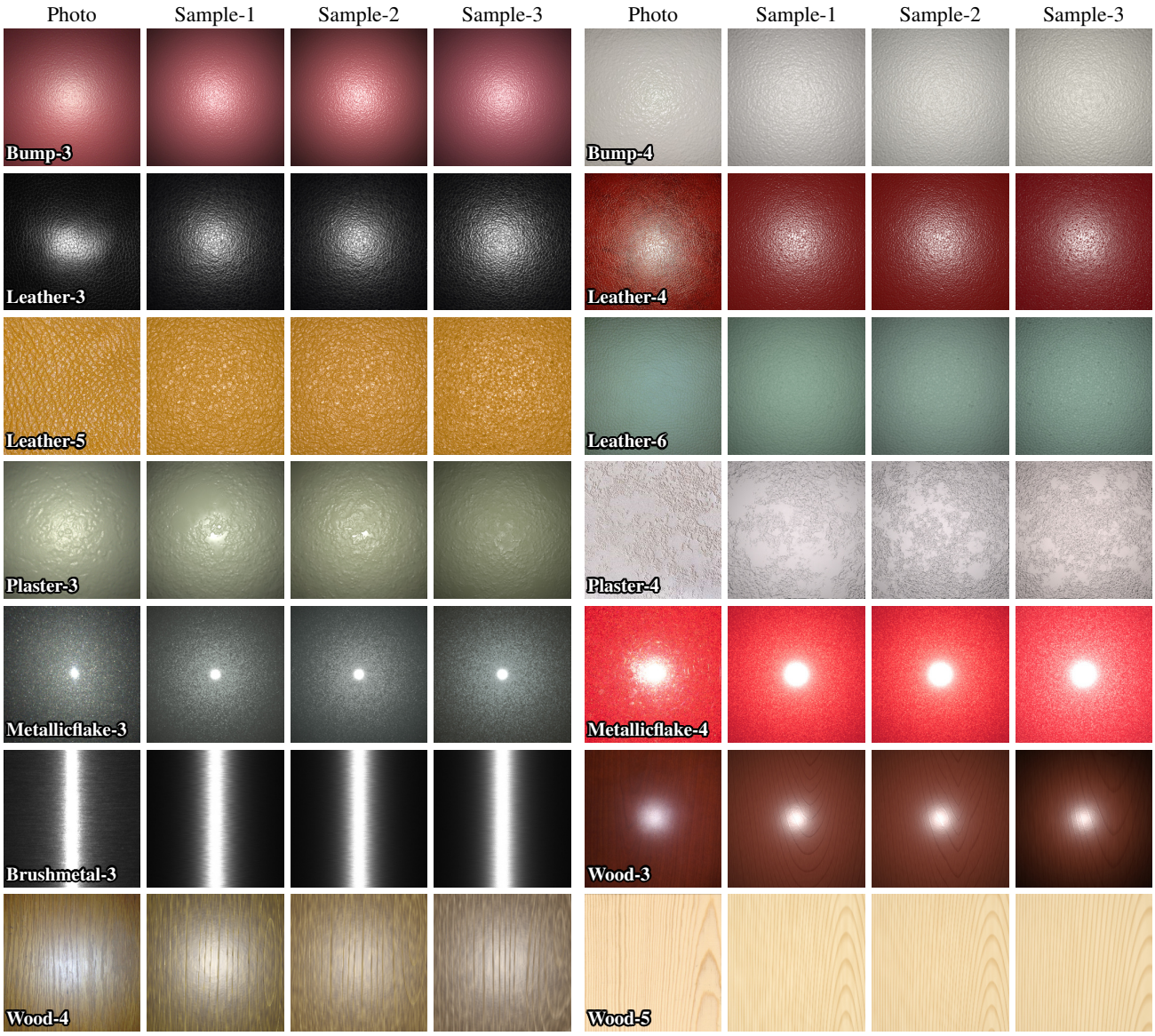


Figure 7: ***Results*** *of our MCMC sampling on* ***real*** *inputs. For each example, the first column is the real target image (photo). We show MCMC samples in the other columns, where sample-1 and sample-2 are chosen closer to the peak of the posterior distribution, and sample-3 is further away. Note that the target images for Plaster-4 and Wood-5 are captured under natural illumination, while the corresponding synthetic images still assume collocated flash illumination; despite this mismatch, the estimated material parameters are still reasonable. Note, target images for Leather-4, Leather-6 and Wood-4 are from the publicly released dataset of [AAL16]. For more results please refer to supplemental materials.*

dural material estimation method). However, the overall perceptual appearance match is usually worse than our method. In some cases, the method produces specular burn-in, as the strong highlight cannot be fully removed and causes holes in the resulting maps (Plaster-4, Metallicflake-4). Advanced BRDF models like brushed metal and metallic flakes are not explicitly handled by their method and usually fail. Finally, their result is fixed at the 256×256 resolution, and does not support higher resolutions, seamless tiling, nor editability; these benefits come from the use of a procedural model.

Figure 12 shows a quantitative comparison of the Learned Perceptual Image Patch Similarity (LPIPS) metric [ZIE*18] between the captured photos and the re-renderings using different methods.

7. Conclusion

Procedural material models have become increasingly more popular in the industry, thanks to their flexibility, compactness, as well as easy editability. In this paper, we introduced a new computational

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

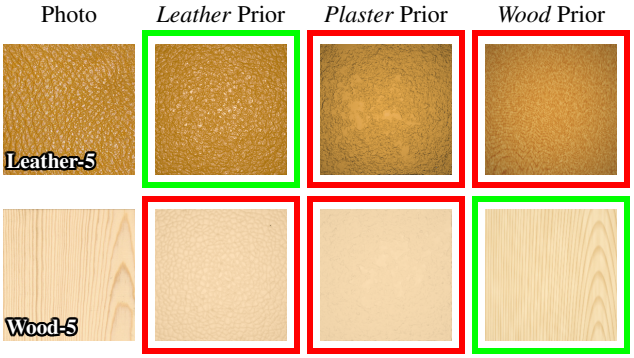


Figure 8: ***Comparison*** *with mismatched forward models. With an inappropriate model as the prior, it would only match the global color but missing all the details.*

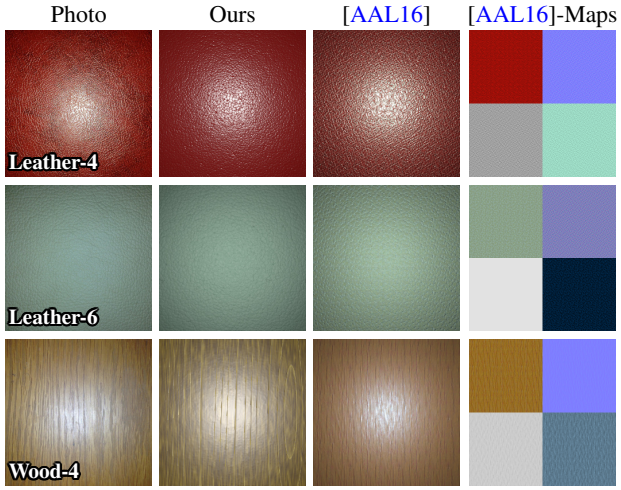


Figure 9: ***Comparison*** *with the Aittala et al. [AAL16]. Results in the third column are rendered using tiled versions of texture maps shown in the fourth column.*

framework to solve the inverse problem: the inference of procedural model parameters based on a single input image.

The first major ingredient to our solution is a *Bayesian framework*, precisely defining the posterior distribution of the parameters, combining four components (priors, procedural material model, rendering operator, summary function). The second ingredient is an *Bayesian inference approach* that leverages MCMC sampling to sample posterior distributions of procedural material parameters. This technique enjoys the generality to handle both continuous and discrete model parameters and provides users additional information beyond single point estimates and allows a cleaner extension to handle discrete parameters.

In the future, we would like to increase the complexity of the models supported even further, to handle materials like woven fabrics, transmissive BTDFs, and more. Finally, extensions to our approach could be used to estimate parameters of procedural models beyond materials, including geometry and lighting, as long as the parameters could be differentiated.

References

[AAL16] AITTALA M., AILA T., LEHTINEN J.: Reflectance modeling by neural texture synthesis. *ACM Trans. Graph. 35*, 4 (2016), 65:1–65:13. 2, 4, 8, 9, 10

[Ado19] ADOBE: Substance, 2019. www.substance3d.com. 2

[AWL13] AITTALA M., WEYRICH T., LEHTINEN J.: Practical svbrdf capture in the frequency domain. *ACM Trans. Graph. 32*, 4 (2013), 110:1–110:12. 2

[AWL15] AITTALA M., WEYRICH T., LEHTINEN J.: Two-shot svbrdf capture for stationary materials. *ACM Trans. Graph. 34*, 4 (2015), 110:1–110:13. 2

[Bet17] BETANCOURT M.: A conceptual introduction to hamiltonian monte carlo, 2017. arxiv:1701.02434. URL: http://arxiv.org/abs/1701.02434. 3, 5

[CCG*15] CHEN C., CARLSON D., GAN Z., LI C., CARIN L.: Bridging the gap between stochastic gradient mcmc and stochastic optimization, 2015. arXiv:1512.07962. 3

[Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Courier Corporation, 1960. 6

[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 (July 2018), 128:1–128:15. 2, 8, 11, 12

[FCMB09] FRANCKEN Y., CUYPERS T., MERTENS T., BEKAERT P.: Gloss and normal map acquisition of mesostructures using gray codes. In *Advances in Visual Computing* (2009), vol. 5876 of *Lecture Notes in Computer Science*, pp. 788–798. 2

[GCP*09] GHOSH A., CHEN T., PEERS P., WILSON C. A., DEBEVEC P.: Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Proceedings of the Twentieth Eurographics Conference on Rendering* (2009), EGSR'09, pp. 1161–1170. 2

[GEB15] GATYS L. A., ECKER A. S., BETHGE M.: A neural algorithm of artistic style, 2015. arXiv:1508.06576. 2, 4

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 2414–2423. 2, 4

[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), 134:1–134:15. 2

[GTHD03] GARDNER A., TCHOU C., HAWKINS T., DEBEVEC P.: Linear light source reflectometry. *ACM Trans. Graph. 22*, 3 (July 2003), 749–758. 2

[Has70] HASTINGS W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109. 3, 5

[HDR19] HU Y., DORSEY J., RUSHMEIER H.: A novel framework for inverse procedural texture modeling. *ACM Trans. Graph. 38*, 6 (2019), 186:1–186:14. 3, 8, 11

[KKTM15] KULKARNI T. D., KOHLI P., TENENBAUM J. B., MANSINGHKA V.: Picture: A probabilistic programming language for scene perception. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4390–4399. 2, 3

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. 8

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

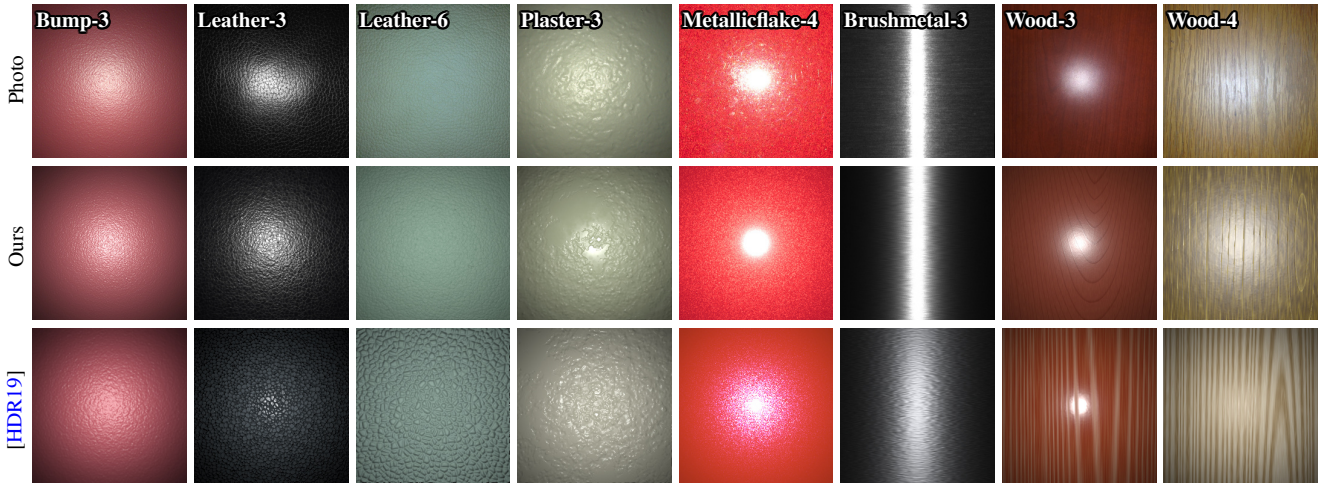


Figure 10: ***Comparison*** *to the forward neural prediction method of Hu et al. [HDR19], where we apply their network structure with our BRDFs and lighting conditions. The photo (top) is better matched by our MCMC sampling results (middle) than their prediction (bottom), which moreover tends to become worse for more complex BRDF models and with more parameters. On the other hand, [HDR19] can be used as an efficient initialization of our sampling, as shown in Figure 11.*

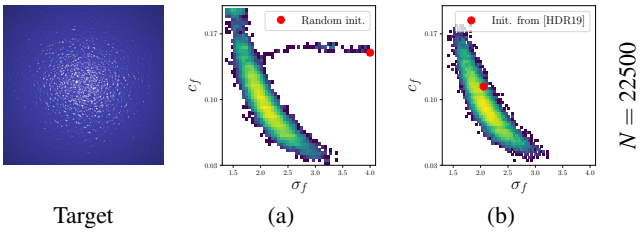


Figure 11: ***Initialization*** *of our sampling with the method of Hu et al. [HDR19] on a synthetic bumpy surface example. The figure shows joint posterior distributions over two parameters using different initializations: a random initialization drawn from the prior (a) and the prediction of [HDR19] (b). As we can see, starting from the result of [HDR19] can shorten the burn-in phase of the MCMC sampling process.*

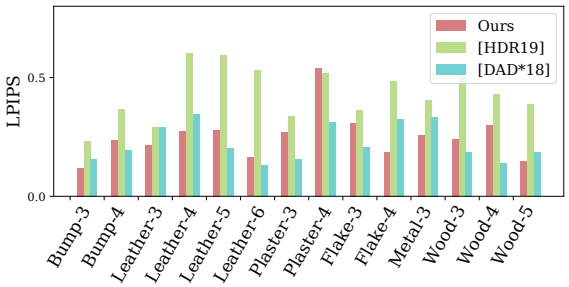


Figure 12: ***Quantitative evaluation.*** *The LPIPS of our results are consistently better than [HDR19]. Some LPIPS values from [DAD*18] are better than ours, since (as per-pixel methods) they can better match the noise patterns in the textures.*

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531–540. 3

[KSZ*15] KHUNGURN P., SCHROEDER D., ZHAO S., BALA K., MARSCHNER S.: Matching real fabrics with micro-appearance models. *ACM Trans. Graph. 35*, 1 (2015), 1:1–1:26. 3

[LDHM16] LIU A. J., DONG Z., HAŠAN M., MARSCHNER S.: Simulating the structure and texture of solid wood. *ACM Trans. Graph. 35*, 6 (2016), 170:1–170:11. 8

[LLR*15] LI T.-M., LEHTINEN J., RAMAMOORTHI R., JAKOB W., DURAND F.: Anisotropic gaussian mutations for metropolis light transport through hessian-hamiltonian dynamics. *ACM Trans. Graph. 34*, 6 (2015), 209:1–209:13. 3

[LSC18] LI Z., SUNKAVALLI K., CHANDRAKER M.: Materials for masses: SVBRDF acquisition with a single mobile phone image. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III* (2018), vol. 11207 of *Lecture Notes in Computer Science*, pp. 74–90. 2

[LWS*18] LEAF J., WU R., SCHWEICKART E., JAMES D. L., MARSCHNER S.: Interactive design of periodic yarn-level cloth patterns. *ACM Trans. Graph. 37*, 6 (2018), 202:1–202:15. 3

[Nea12] NEAL R. M.: MCMC using Hamiltonian dynamics. *arXiv e-prints* (Jun 2012), arXiv:1206.1901. 3

[RT96] ROBERTS G. O., TWEEDIE R. L.: Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli 2*, 4 (12 1996), 341–363. 3, 5

[RWB17] RISSER E., WILMOT P., BARNES C.: Stable and controllable neural texture synthesis and style transfer using histogram losses, 2017. arXiv:1701.08893. 4

[SZ15] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015). 4

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, pp. 65–76. 3

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.:

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.

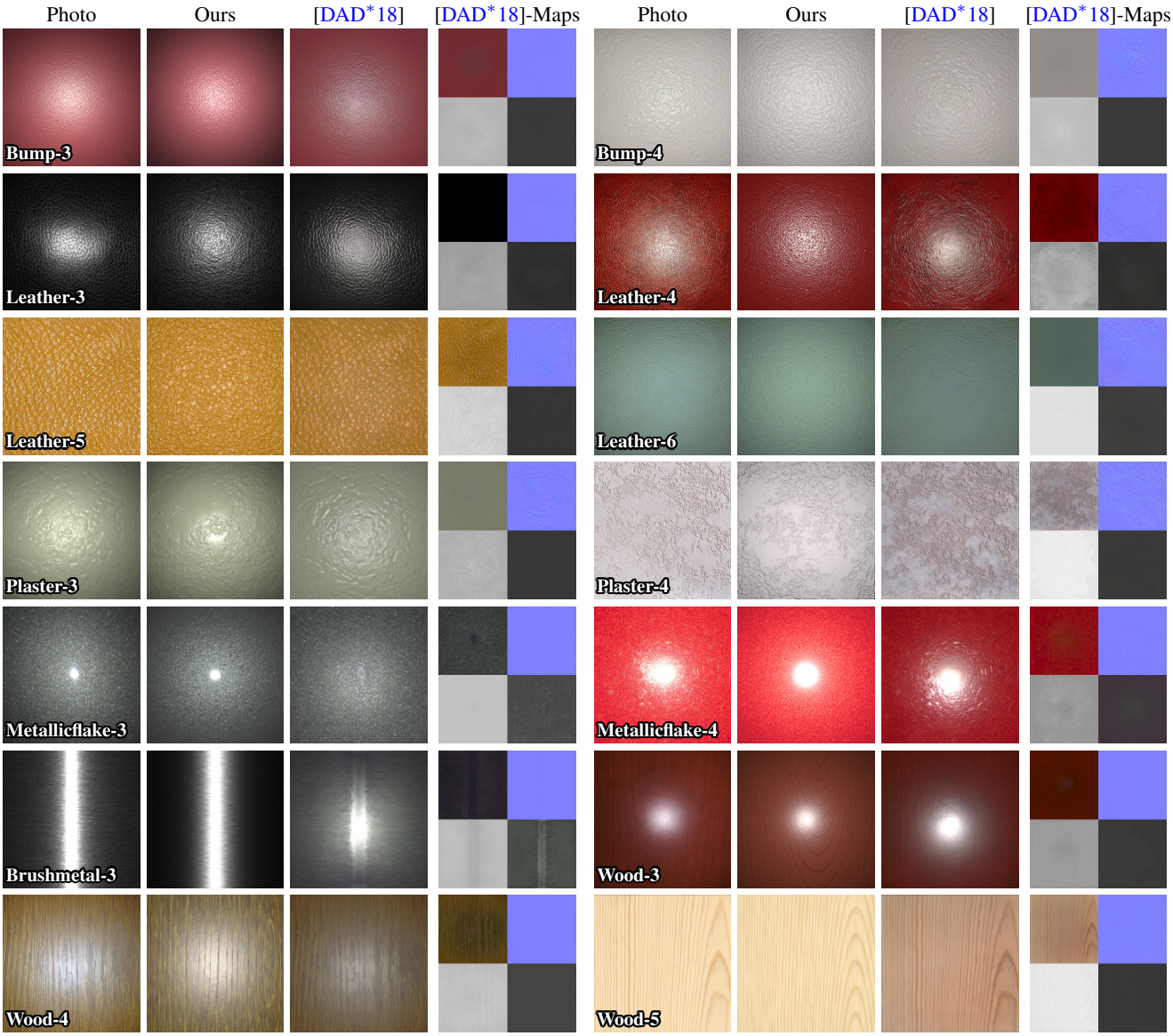


Figure 13: ***Comparison*** *to the single input SVBRDF estimation method of Deschaintre et al. [DAD*18]. Due to the nature of the method, their texture patterns are closely aligned with the input image; however, the overall perceptual appearance match is usually worse than our method. In some cases, the method produces specular burn-in, as the strong highlight cannot be fully removed and causes holes in the resulting maps (Plaster-4, Metallicflake-4). Advanced BRDF models like brushed metal and metallic flakes are not explicitly handled by their method and usually fail.*

Microfacet models for refraction through rough surfaces. *EGSR 07* (2007), 195–206. 7

[WSM11] WANG C.-P., SNAVELY N., MARSCHNER S.: Estimating dual-scale properties of glossy surfaces from step-edge lighting. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011), SA '11, pp. 172:1–172:12. 2, 7

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595. 9

[ZJMB11] ZHAO S., JAKOB W., MARSCHNER S., BALA K.: Building volumetric appearance models of fabric using micro ct imaging. *ACM Trans. Graph. 30*, 4 (July 2011), 44:1–44:10. 3

[ZLB16] ZHAO S., LUAN F., BALA K.: Fitting procedural yarn models for realistic cloth rendering. *ACM Trans. Graph. 35*, 4 (2016), 51:1–51:11. 3

[ZRB14] ZHAO S., RAMAMOORTHI R., BALA K.: High-order similarity relations in radiative transfer. *ACM Trans. Graph. 33*, 4 (2014), 104:1–104:12. 6

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.