Image-based Remapping of Spatially Varying Material Appearance

Alejandro Sztrajman[1] Jaroslav Křivánek[2] Alexander Wilkie[2] Tim Weyrich[1]

[1]University College London [2]Charles University

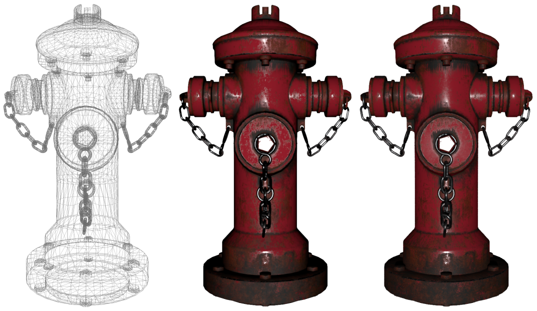


Figure 1. Cross-renderer remapping of spatially varying material. Mesh (left); Blender-Ward (center)→ Cycles-GGX (right).

Abstract

BRDF models are ubiquitous tools for the representation of material appearance. However, an astonishingly large number of different models are now in practical use. Both a lack of BRDF model standardization across implementations found in different renderers, as well as the often semantically different capabilities of various models, have become a major hindrance to the interchange of production assets between different rendering systems. Current attempts to solve this problem rely on manually finding visual similarities between models, or mathematical similarities between their functional shapes, which requires access to the shader implementation, usually unavailable in commercial renderers. We present a method for automatic translation of material appearance between different BRDF models that uses an image-based metric for appearance comparison and that delegates the interaction with the model to the renderer. We analyze the performance of the method, both with respect to robustness and also visual differences of the fits for multiple combinations of BRDF models. While it is effective for individual BRDFs, the computational cost does not scale well for spatially varying BRDFs. Therefore, we also present two regression schemes that approximate the shape of the transformation function and generate a reduced representation that evaluates instantly and without further interaction with the renderer. We present respective visual comparisons of the remapped SVBRDF models for commonly used renderers and shading models and show that our approach is able to extrapolate transformed BRDF parameters better than other complex regression schemes. Finally, we analyze the transformation between specular and metallic workflows, comparing our results with two analytic conversions.

1. Introduction

Computer-generated imagery workflows commonly involve a broad range of modelling and rendering tools, each targeting different goals and requirements [Schregle et al. 2013], and the exchange of data between these tools is hindered by incompatible representations. As a consequence, existing model assets frequently have to be redesigned to be used in other software, resulting in large modelling overheads. This is particularly true in the case of material models. So far, a great number of BRDF models has been developed for appearance representation, but a lack of a standardisation and renderer-specific implementation details lead to visual deviations even between identically named reflectance models.

The current abundance of BRDF models reflects the fact that no single model is able to realistically reproduce the full range of available measured materials [Brady et al. 2014; Guarnera et al. 2016]. However, for a given material represented using one model, it is often possible to find a new set of parameters that approximates its appearance with a different model. Many existing rendering systems support such *remapping* of material parameters to address the incompatibility between models and to remain backwards-compatible to older versions of their software [Pharr et al. 2016; Corona 2017]. That said, these remappings are often based on manually determined or heuristic relations between the functional shapes of the models; that requires access to the model implementations, or oversimplifications by assuming one-to-one correspondence between individual parameters of both models will occur. The problem is exacerbated by popular renderers and graphics engines using proprietary shading models [Unr 2017; Uni 2017].

An automatic solution to this problem needs to consider the constraints of the real-world scenario where a material is interchanged between different third-party renderers. In this situation, we do not have access to the implementation of the shaders, only to the model parameters and the resulting renderings. To address this problem, we present an image-based method for the remapping of BRDFs that works for closed-source renderers, assuming no knowledge of the model implementations. We analyze the robustness of the method applied to a set of BRDF models, and we discuss common issues and strategies to improve the stability of the method in different types of materials.

In addition, we present a regression scheme to generate a reduced representation of the transformation that evaluates instantly and without further interaction with the renderer, allowing for fast remapping of entire parameter texture maps. We show visual comparisons of the remapping of spatially varying BRDF models (SVBRDF) that illustrate the ability of our approach to provide a close match between different renderers, even when remapping between very different shading models.

2. Related Work

2.1. Reflectance Remapping

Traditionally, appearance modelling dealt with finding reflectance models that would agree well with real-world observations [Nicodemus et al. 1977]. Accordingly, a large body of work on fitting of reflectance models to measured data exists [Marschner 1998; Weyrich et al. 2008; Guarnera et al. 2016]. In contrast, little academic attention has been paid to the direct translation between BRDF models, apart from initial studies of the well-posedness [Sztrajman et al. 2017] and perceptual quality of the transformations [Guarnera et al. 2018].

Among commercial products, the arguably most prominent software to remap reflectance from one model to another is Allegorithmic's Substance Painter [Allegorithmic 2017], a dedicated tool to author appearance for a wide range of target platforms. In order to address the variability in renderer-specific BRDF models and implementations, the software contains various export functions that employ manually optimized heuristics to remap BRDF parameters for specific target rendering engines [Damez Oct. 2017]. Manual creation of such heuristics, however, can be costly and does not necessarily lead to optimal results.

Other examples of available products are renderers, such as PBRT [Pharr et al. 2016] or Corona [Corona 2017], that remap reflectance from an older versions' legacy representations. The latter switched from a variant of the Ashikhmin-Shirley BRDF [Ashikhmin and Shirley 2000] to a GGX microfacet BRDF [Walter et al. 2007; Burley 2012] and remaps BRDF specifications by analytically matching the width of the models' specular lobes [Krivanek Jul. 2017].

2.2. Appearance Comparison

Quantifying (dis)similarity between two BRDFs is a problem encountered in any BRDF fitting work, and our remapping is no exception. Ngan et al. [Ngan et al. 2005] follow Lafortune et al. [Lafortune et al. 1997] and employ a simple L_2 distance between cosine-weighted BRDF values. We find that such a metric puts disproportionate emphasis on matching BRDF peaks at the expense of tails, which can result in appearance deviations.

In their follow-up work, Ngan et al. [Ngan et al. 2006] argue for an image-based metric, where the dissimilarity between two BRDFs is modelled as the difference between rendered images with the respective BRDFs under natural illumination. Recently, Havran et al. [Havran et al. 2016] confirmed the validity of the image-based methodology through psychophysical experiments and, furthermore, designed specialized geometries that provide richer information on material properties than the simple sphere used by Ngan et al. We follow this image-based strategy for two reasons: first, it has been repeatedly shown to correlate well with the perceived material differences; second, our setup lends itself well to rendering images using any (unknown) BRDF, whereas obtaining individual BRDF values using an off-the-shelf renderer may be more difficult.

We focus our efforts on finding a remapping scheme that results in parameters that vary smoothly with respect to changes in the source-material parameters. For uniform BRDFs, this is an expected behavior of the transformation, and deviations are suggestive of problems in the optimization, such as finding local minima or output parameters that only look similar in a particular scene setting. For spatially varying (SV)BRDFs, the smoothness of the mapping is even more important, because the material parameters across the surface are computed by interpolation. If the parameters vary abruptly, this is likely to produce incorrect SVBRDF remappings, even if the appearance of each individual texel is correctly matched.

3. Remapping of Uniform Materials

The process of BRDF remapping is similar in structure to the fitting of BRDFs, where we start with a *target* BRDF model and an initial guess of the parameters, and we want to fit reflectance data, generally measured from a real-world material. This involves the minimization of the difference between the appearances of the BRDF and the data, performed through nonlinear optimization of the parameters of the BRDF model (Figure 2).

In the case of BRDF remapping (Figure 3), the scheme is analogous; however, instead of using measured data, we are now matching the appearance of the target

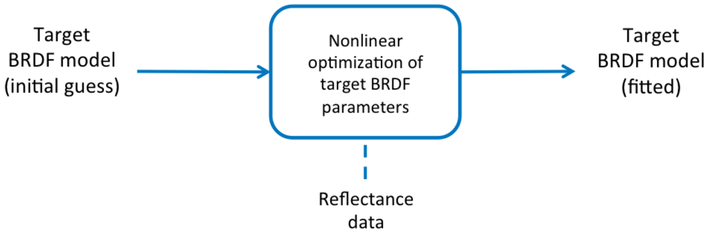


Figure 2. Broad scheme for BRDF fitting.

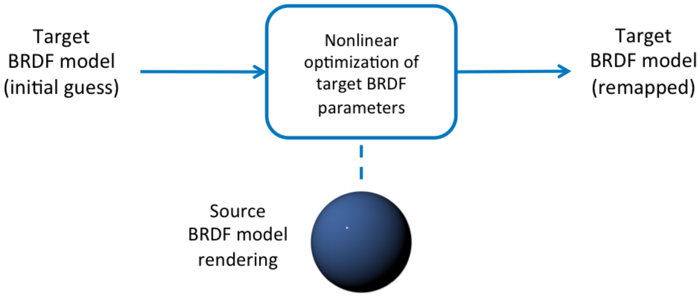


Figure 3. BRDF remapping scheme.

BRDF model with another *source* BRDF model. In this scheme, we assume no direct access to the implementation of the BRDF models. In particular, the target model is assumed to belong to an external renderer in a typical usage scenario of our technique. In order to perform an appearance comparison under these conditions, we measure the difference in image space, by comparing rendered images of a single scene using each of the two BRDF models.

Thus, in each step of the optimization, we only need to be able to generate new renders of this scene with the target BRDF. The image difference is then computed with an L_2 metric in color space, which is common practice in the context of BRDF fitting [Ngan et al. 2006] (other metrics are used as well, but no single distance metric has emerged as a superior choice for general BRDF fitting).

In the remainder of this paper, we will consider three optimization strategies to remap a (uniform) BRDF specification to parameters of a different model. Section 5 will then present our approach to extend the remapping to spatially varying appearance.

3.1. Optimization Strategies

A simple optimization scheme that attempts to fit all model parameters at once (as shown in Figure 3) often leads to local minima during the optimization, due to the coupling between the diffuse and specular terms in the model. In Section 4.1, we provide a systematic analysis of the stability of the remapping scheme. In order to improve the stability of the optimization, we test the following two variants of our remapping scheme.

Two-stage Remapping

In this scheme, the diffuse and specular terms are remapped independently (Figure 4). This is not unlike BRDF fitting to real-word data where diffuse and specular reflectance may be separated optically [Debevec et al. 2000] or statistically [Weyrich et al. 2006] before conducting separate fits. In our case, it requires source renderings of the diffuse-only, and purely specular components, respectively. In the end we ob-

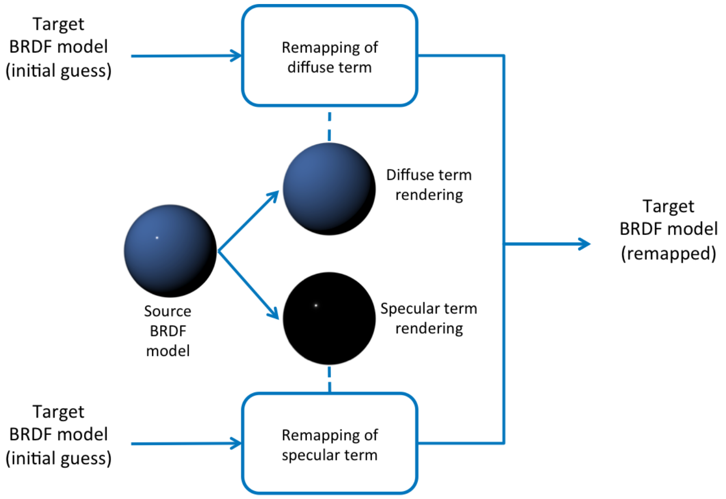


Figure 4. BRDF remapping scheme in two stages. Diffuse and specular components are remapped independently.

tain remapped versions of each term that are merged in the remapped target BRDF model.

Three-stage Remapping

The two-stage remapping assumes an independence of diffuse and specular terms that might not hold true for some layered materials. The three-stage scheme (Figure 5) recovers the coupling between both terms by using the results of the two-stage scheme as a good starting guess for a subsequent remapping that optimizes all parameters simultaneously, reducing the chance of falling into local minima.

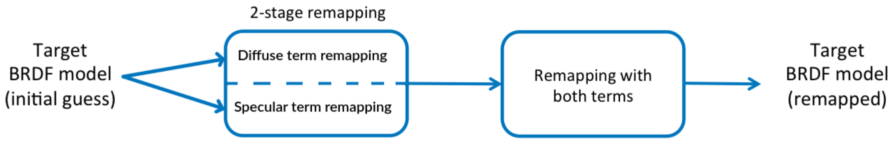


Figure 5. BRDF remapping scheme in three stages.

4. Analysis of Uniform Material Remapping

We tested our approach using three renderers: Mitsuba; Blender's internal preview renderer; and Cycles, a physically-based renderer that is currently the most commonly used off-line renderer in Blender. Due to differences in how light-source intensities are specified across renderers, we further had to match irradiance before remapping an external BRDF to a Mitsuba BRDF. This was done by a global scale determined from the ratio of diffuse-only renderings from the two renderers.

Our uniform BRDF remapping code calls Mitsuba in its optimization loop, and thus the remap can take source BRDFs from arbitrary renderers while the target BRDF has to be from within Mitsuba. In Section 6, we will show that this does not represent a limitation for the remapping, since the transformation to Mitsuba can be used as an intermediate step in a sequence of remappings:

$$\text{source BRDF} \rightarrow \text{Mitsuba BRDF} \rightarrow \text{target BRDF}.$$

4.1. Uniform Fitting Strategies

We begin by evaluating the three optimization strategies for remapping of uniform BRDFs that were introduced in Section 3.1. We performed a systematic study of these remapping schemes via an analysis of the robustness of the transformation that links the parameters of the models. We did this for multiple combinations of BRDF models that are available in Mitsuba [Jakob 2010] (Ashikhmin-Shirley, Beckmann, GGX, Phong, Ward). For the sake of brevity, we focus here only on a few of these combinations, to demonstrate a few common effects we encountered when dealing with remapping between different BRDFs within Mitsuba.

The nonlinear optimization of parameters was performed using the Least Squares TRF (Trust Region Reflective) method [Jones et al. 2001–; Branch et al. 1999], enforcing positive values on the target remapped parameters. For the image-based appearance comparison, we generated linear RGB renderings in HDR (512×512) of a simple scene: a sphere of unit radius located at the origin illuminated by a point lightsource ($r_l = 3$, $\theta_l = 45°$). This produces a sampling of only a two-dimensional slice of the BRDF space, which depends on the relative position of the illumination source. Although we find that the result of the optimization is not highly influenced by the choice of light position, in Section 9 we will discuss its effect on the visual match between BRDF models.

4.1.1. Conductors

Our analysis of the remapping of conductors comprises 60 materials from Mitsuba's database. In Figure 6, we show the results for a remapping from Ashikhmin-Shirley (source) to Ward (target). In these implementations the specular terms in both models are described by an RGB specular parameter and the roughness (single-channel), essentially characterizing the intensity and the spread of the lobes. We show the remapping of a specular parameter in Ashikhmin-Shirley to an analogous parameter in Ward (both single-channel), for multiple fixed values of roughness. The information provided by the IOR parameter in the standard BRDF interface in Mitsuba is here condensed into the Fresnel coefficient F_0, which can be expressed in its more general form as (with c the complex IOR)

$$F_0 = \frac{(c-1)(c^*-1)}{(c+1)(c^*+1)}. \tag{1}$$

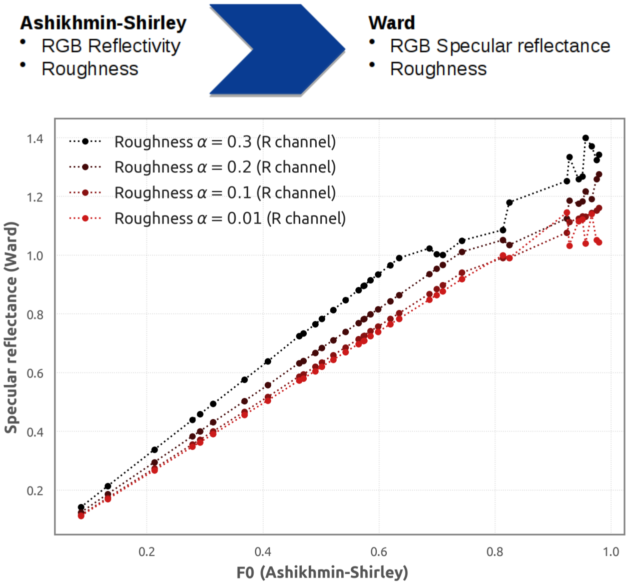


Figure 6. Remapping of conductors from Ashikhmin-Shirley to Ward. Detail of parameters in Mitsuba, and plot of specular reflectance (Ward) vs Fresnel coefficient (AS).

The expected output of the remapping is a smoothly varying correspondence between source and target parameters. In a typical usage case, the user would expect small changes in the source material to correspond to small changes in the exported appearance. We will show that a deviation from this behavior usually signals numerical instability due to a decreasing capacity of the target model to match the source, or the occurrence of local minima during the optimization. The stability of the transformation will prove crucial when we deal with the remapping of spatially-varying BRDFs (Section 5) that are reconstructed by interpolation of multiple uniform materials.

In Figure 6, the transformation shows a smooth behavior for most materials, but exhibits instabilities for parameters that are remapped to specular reflectance > 1. These can be traced back to this particular implementation of Ward, where the values of specular reflectance are trimmed to ensure energy preservation, and it is representative of the implementation-dependent behavior that we may find in renderers. Most of the instabilities found during our study shared this behavior of exhibiting well-localized regions in parameter space where the remapping becomes unreliable.

After filtering the unstable cases from Figure 6, Figure 7 shows the results for a round-trip remapping, where we transformed the parameters back to the initial Ashikhmin-Shirley model. The result is a straight line of unit slope, which shows that in this particular case, the parameters reliably go back to their original values

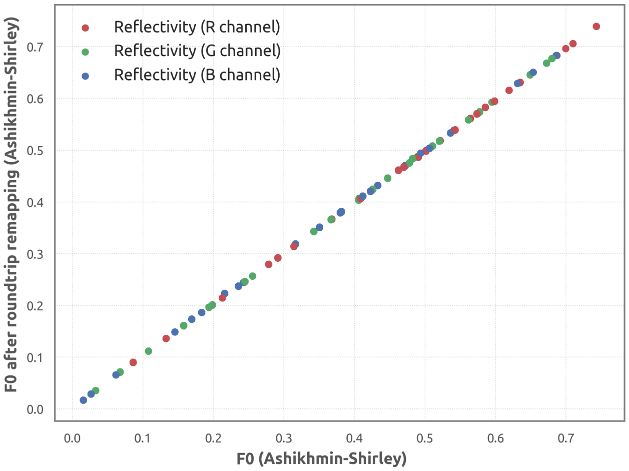


Figure 7. Round-trip remapping of conductors from Ashikhmin-Shirley to Ward and then back to Ashikhmin-Shirley. Remapped Fresnel coefficient F_0 vs original F_0.

after the two remappings. This speaks for the general robustness of the approach and indicates that we generally can recover the original appearance after a remapping takes place.

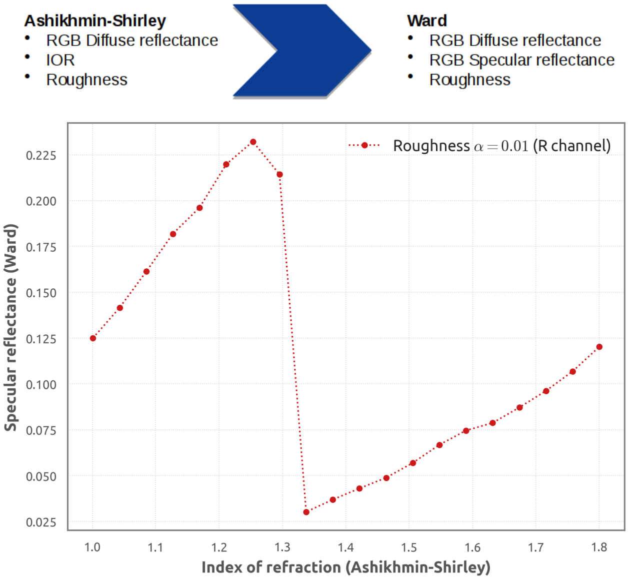


Figure 8. Single-stage remapping of dielectrics from Ashikhmin-Shirley to Ward. Detail of parameters in Mitsuba for both models, and plot of fitted specular reflectance (Ward) vs input IOR (AS). The optimisation becomes unstable around IOR= 1.3.

4.1.2. Dielectrics

The reflectance of dielectric materials includes an additional diffuse component. The intensity of the specular term is similar for all light wavelenghts and is usually approximated by a single-channel parameter (e.g., the index of refraction). In Figure 8, we show the results of remapping from Ashikhmin-Shirley to Ward, using a single-stage optimization with both diffuse and specular parameters. The plot corresponds to a parameter sweep of the IOR in Ashikhmin-Shirley for fixed diffuse and roughness parameters.

In this case, the instability signals a change of regime in the optimization. In Figure 9, we show renderings that correspond to the points at both sides of the jump in the curve of Figure 8. In one case the remapping is working correctly, and we obtain a similar appearance in both models. In the other case, we observe that the optimization arrives at a local minimum, and the remapping is unable to recover the characteristic highlight from the source.

Figures 10(a) and 10(b) show the results of the two- and three-stage approaches, developed to improve the stability of the remapping process and illustrated in the diagrams of Figures 4 and 5. The two-stage approach effectively recovers a smooth relationship between the parameters, by avoiding the coupling between the diffuse and specular terms. With the additional optimization step of the three-stage approach, in some cases we were able to slightly reduce the optimization error with respect to the two-stage approach, but unfortunately the coupling between diffuse and specular terms still causes several instabilities that make this second approach unreliable. In summary, in order to generate a robust remapping we need to avoid the coupling of the diffuse and specular components, by remapping each term independently (two-stage method). In the following sections we will proceed by basing all BRDF transformations on this method.

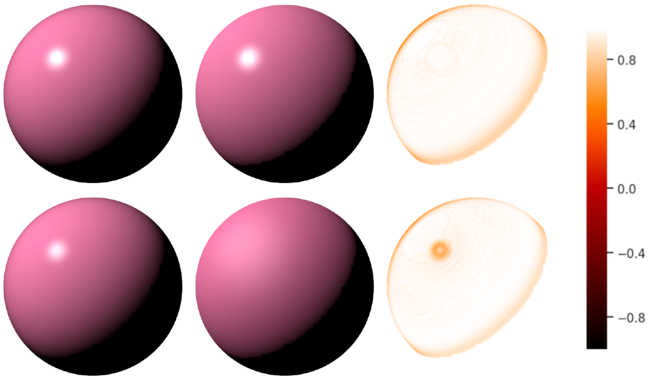
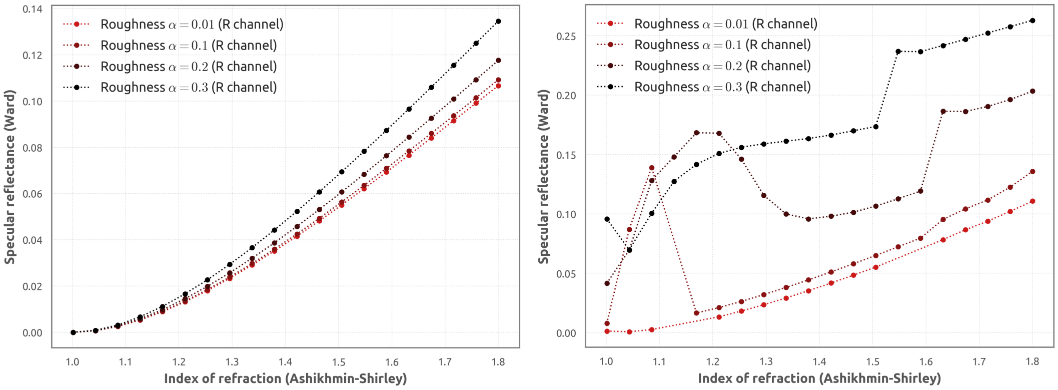


Figure 9. Source model (left), remapped target (center) and SSIM error (right) corresponding to IOR $= 1.3$ (top), and IOR $= 1.34$ (bottom) in the unstable single-stage remappings of Figure 8. In the bottom row the optimization fails to reproduce the highlight of the material.



(a) *Two-stage remapping.* (b) *Three-stage remapping.*

Figure 10. Remapping of conductors from Ashikhmin-Shirley to Ward illustrating specular reflectance vs Fresnel coefficient F_0 for multiple values of roughness. The three-stage remapping shows multiple instabilities.

5. Remapping of Spatially Varying Materials

Spatially varying materials are commonly defined using texture maps that provide the value of each model parameter across the surface. In Figure 11 we display the decomposition of an SVBRDF asset into texture maps, describing four different parameters involved in the shading process (diffuse reflectance, specular roughness, specular reflectance, and surface normals).

The remapping of the corresponding spatially varying material to a different BRDF model or renderer requires the remapping of each individual texel, which can be performed using one of the schemes for uniform BRDF remapping from the previous section. However, the optimization required by the remapping of a single material

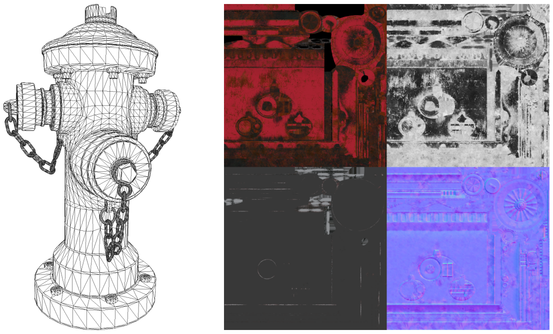


Figure 11. Decomposition of 3D asset into a low-resolution geometry and four texture maps describing the spatially varying parameters of the material (diffuse reflectance, roughness, specular reflectance and normals).

usually takes a few minutes, which is acceptable for uniform BRDFs but intractable for high-resolution texture maps with hundred-thousands of texels.

To solve this problem we employ a regression scheme that utilizes the data from the remapping of uniform materials to learn the relationship between parameters in both BRDF models involved. Through parameter sweep over the source model, we generate a database of uniform material parameters and their remapped counterparts in the target model that is used as input for a regression scheme. Thus, we are able to generate a reduced representation of the transformation between the two models, which can then be evaluated efficiently without the need for further optimization or interaction with the renderer. Note that, even though our uniform remapping implementation requires the target BRDF to be defined within Mitsuba, we can reverse the role of source and target in the regression process, thus implementing BRDF remappings in the opposite direction. Furthermore, in Section 6, we will show the implementation of chained remappings that allow the transformation between arbitrary renderers and BRDF models:

$$\mathbf{T}_1\mathbf{:}\ \mathrm{BRDF}_1 \rightarrow \text{Mitsuba BRDF} \qquad\qquad \mathbf{T}_2\mathbf{:}\ \mathrm{BRDF}_2 \rightarrow \text{Mitsuba BRDF}$$

$$\mathbf{T}_2^{-1}\mathbf{T}_1\mathbf{:}\ \mathrm{BRDF}_1 \rightarrow \text{Mitsuba BRDF} \rightarrow \mathrm{BRDF}_2$$

5.1. Regression Scheme

Figure 12 shows an example remapping between two implementations of Ward: from Mitsuba to Blender internal renderer. The left plot shows a one-to-one nonlinear relationship between the two "roughness" parameters, illustrating that the implementations of Ward differ non-trivially in the influence of the roughness parameter. The

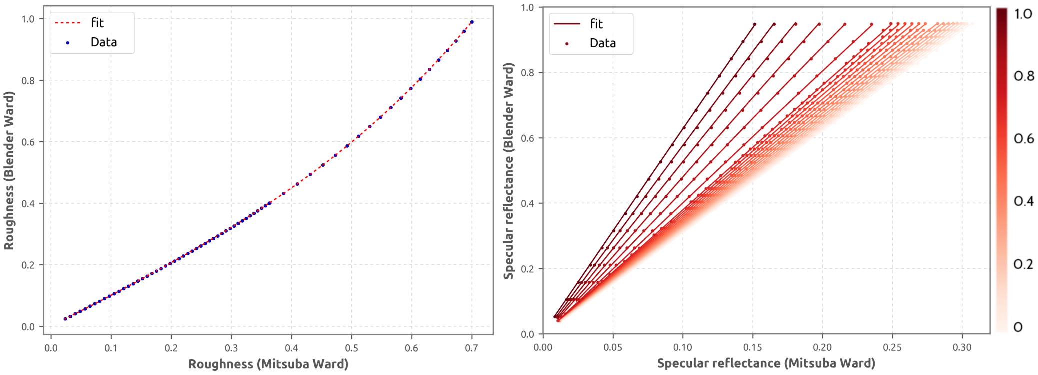


Figure 12. Remapping from Mitsuba-Ward to Blender internal-Ward, with data points and SVR regression. (Left): Remapping of roughness. The nonlinear mapping shows that the two variants of Ward are distinct (left); Remapping of specular reflectance with color indicating roughness. The transformation follows different mappings depending on the roughness parameter (right).

right plot offers a different set of cross-sections through the same remapping of parameter spaces: for a wide range of roughness values, we show varying specular reflectance values of the Mitsuba-Ward map to Blender-Ward model parameters. In contrast with the one-to-one correspondence between parameters found in the left plot, here the transformation traces a different mapping depending on the roughness parameter, signaling a complex relationship between multiple parameters in the models that would be hard to recover by a manual mapping of model parameters.

The mapping of BRDF parameters can be approximated by learning methods: we tested both support vector regression (SVR) [Chang and Lin 2011; Pedregosa et al. 2011] and fully-connected neural networks [Paszke et al. 2017] with similar results. In Figure 12, we show example regressions obtained using the ϵ-SVR method with radial basis functions (RBF), which requires the tuning of three hyperparameters: C (error penalty), ϵ (error margin), and γ (radial function dumping factor). For this purpose we implemented a simple gradient descent scheme that was able to converge to appropriate hyperparameter values in a few minutes on the CPU (a Python implementation of the SVR fitting can be found in the additional material). Once trained, the SVR is able to correctly model and interpolate parameter values inside the region sampled by the training dataset. However, in Section 7 we will show cases where this method fails to extrapolate outside of the sampled region of parameter space used for training, leading to unexpected changes in appearance, and we will present an improved parametric regression scheme that is able to extrapolate properly.

5.2. Sampling of Parameters

An accurate regression of the mapping between two BRDF models usually requires a dense sampling of the parameter space in the source model. In most cases we used a uniformly-spaced sampling of parameters, sweeping through 30 values for each parameter in the source model, taking advantage whenever possible of the fact that usually the parameter sweep can be done on a single channel. In Section 8, we will analyze a case with a large number of parameters to be sampled. Here we will perform a random sampling of the parameter space.

In general, the valid range of values for the parameter sweep varies across model-pairs and regions of the parameter space. In Figure 12 (right), we observe that the largest valid value for specular reflectance is 0.3 (higher values land in values above 1 in the target renderer, thus causing instabilities). For higher values of roughness, this limit is reached much sooner. In Section 7, we will analyze a case where the remapped parameters fall outside of the sampled domain, and the regression scheme is unable to properly extrapolate the shape of the transformation. We will show that, under certain circumstances, it is possible to produce an accurate regression with only a sparse sampling of the parameter domain.

6. Results of Cross-renderer SVBRDF Remapping

In Figure 12 we analysed the remapping between two implementations of Ward: from Mitsuba to Blender internal renderer. Once the compact representation of the transformation is generated, it can be used to perform the remapping of every texel in the parameter maps of an SVBRDF. In Figures 13 and 14, we display the roughness and specular maps for Mitsuba-Ward and their corresponding remapped versions in Blender internal-Ward.

As previously seen, the transformation of the roughness (Figure 12 (left)) depends only on the original roughness, and thus the remapping of a roughness texture map is essentially a tonemapping operation, as observed in Figure 13. In contrast the remapping of the specular parameter (Figure 12 (right)) depends on both the roughness and the specular parts, which produces a remapped specular map that inherits details of the roughness map (Figure 14).

Figure 15 displays the rendering of a 3D asset using the texture maps from Figures 13 and 14. The corresponding renderings show the efficacy of the remapping:

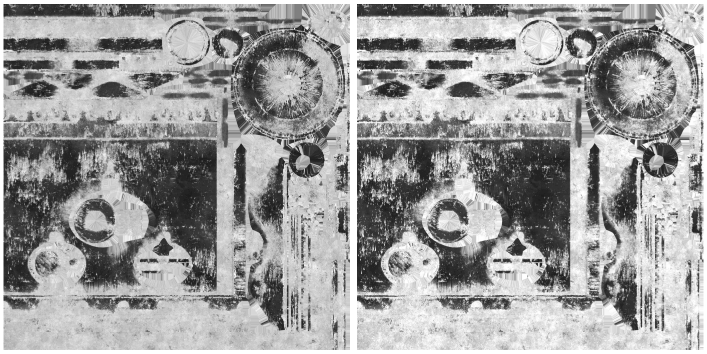


Figure 13. Roughness texture maps. Original Mitsuba-Ward (left); remapped Blender-Ward (right).

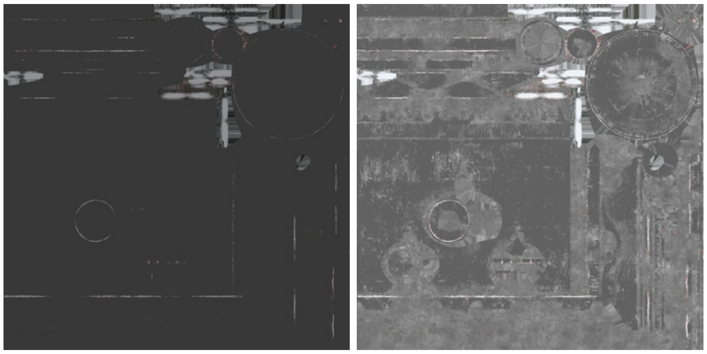


Figure 14. Specular texture maps. Original Mitsuba-Ward (left); remapped Blender-Ward (right).

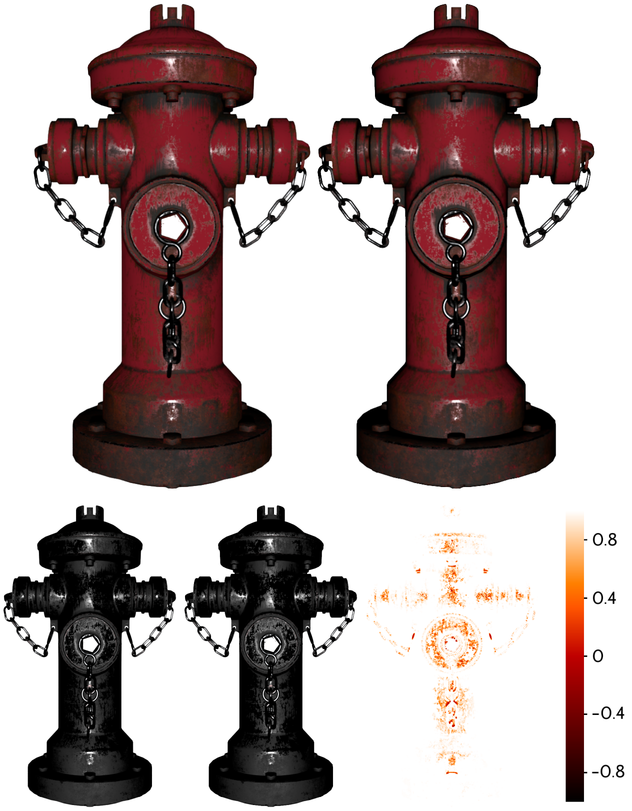


Figure 15. SVR remapping of SVBRDF from Mitsuba's default Ward implementation to Ward within Blender's internal renderer. Corresponding renderings using Mitsuba (top-left) and Blender (top-right); specular-only renderings with SSIM error (SSIM $= 1$ indicates absolute similarity) (bottom).

faint visual differences are limited to surface parts that face both camera and light source. The bottom row shows specular rendering in isolation, including an SSIM error image (SSIM $= 1$ indicates absolute similarity); removing the diffuse term, which is very similar across the renderers, highlights visual differences further.

Figure 16 displays a remapping from the Mitsuba Ashikhmin-Shirley shader to Cycles' GGX model. Once again, we show cross-sectional plots of the parameter remapping function, as well as specular-only renderings with difference image. This result is of particular interest, as the specular lobe of GGX significantly differs from traditional microfacet models, such as Ashikhmin-Shirley. With its heavy tails, the specular term tends to add persistent sheen to a surface, making it challenging to match the appearance of a model with more compact reflectance lobes. Considering that, we believe that our remapping preserves the overall appearance exceptionally well.

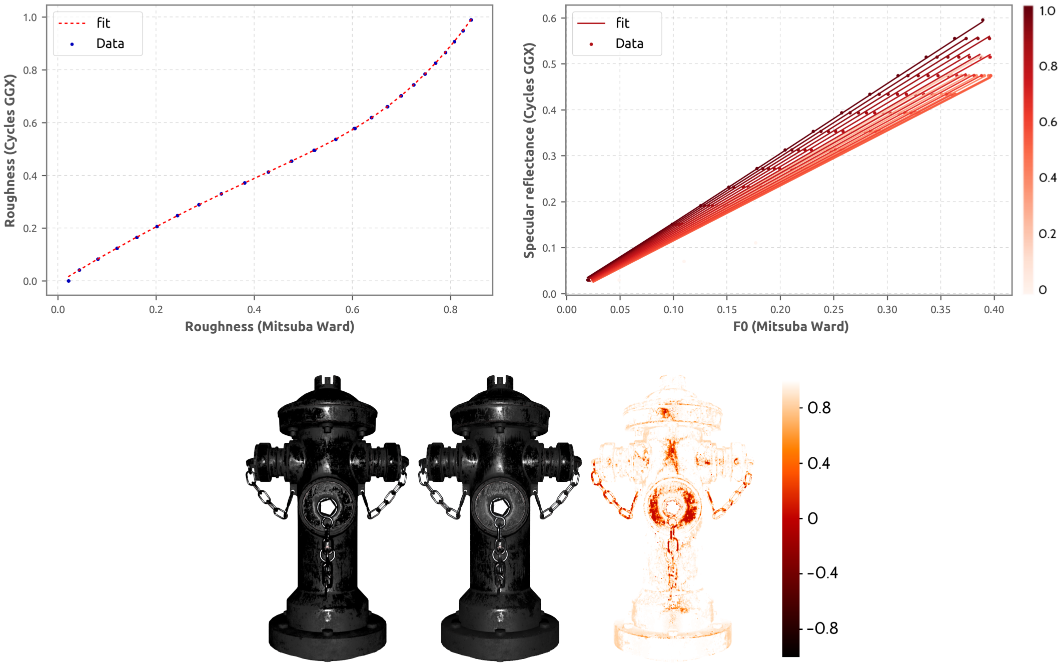


Figure 16. SVR remapping from Mitsuba Ashikhmin-Shirley to Cycles GGX. Remapping of roughness (top-left); remapping of Fresnel coefficient F_0 with color indicating roughness (top-right); specular-only renderings with difference image (bottom).

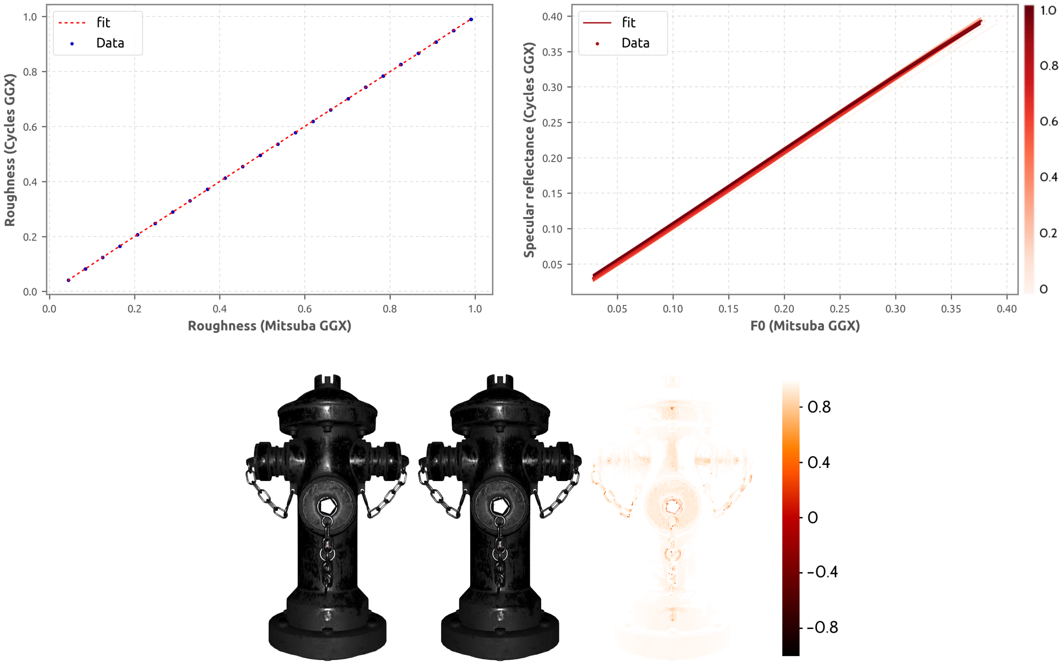


Figure 17. SVR remapping from Mitsuba GGX to Cycles GGX. Remapping of roughness (top-left); remapping of Fresnel coefficient F_0 with color indicating roughness (top-right); specular-only renderings with SSIM error (bottom).

Even within implementations of GGX, however, we observe slight differences between renderers, as can be seen in Figure 17 where we display a remapping from Mitsuba-GGX to Blender Cycles-GGX. Here the relationship between parameters is linear and only shows slight deviations from the identity transformation for the Fresnel coefficient F_0. The simplest explanation for this behavior is a small difference in the parameterization of F_0 in each model, although other contributions can not be discarded (e.g., small differences in the light sources in each renderer).

In general, we find the remapping scheme stable enough to enable chaining of transformations, such as the one shown in Figure 18: Blender internal-Ward → Mitsuba-Ward → Mitsuba-GGX → Blender Cycles-GGX. This makes it conceivable that even within a large collection of BRDF models, a compact spanning graph of pre-learned transformations is sufficient to map from any model to another one.

Apart from the inherently increased sheen due to GGX, the result remains remarkably close to the input. Note that in spite of working with three different renderers, the most noticeable differences occur when remapping from Ward to GGX within the same renderer (Mitsuba). This hints that the BRDF model's shape is the main factor determining the remappability of a material, despite other additional differences that may be in play between renderers (e.g., point light parameters and distance dependence, post-processing, etc.).

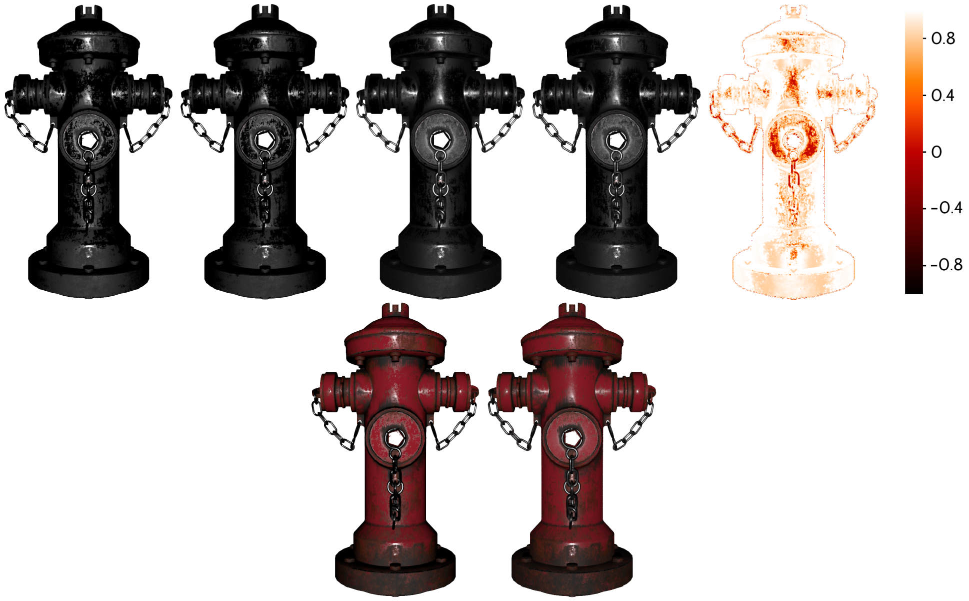


Figure 18. SVR chained remapping from Ward (Blender) to GGX (Cycles) via two intermediate BRDF models. Blender internal-Ward, Mitsuba-Ward, Mitsuba-GGX, Blender Cycles-GGX, SSIM error (left to right); Blender internal-Ward, Blender Cycles-GGX (bottom with diffuse term).

7. A Parametric Regression Scheme

So far we have shown examples of remappings where the material parameters (i.e., the texels of the asset texture maps) lie inside the region of parameter space that was pre-sampled and used for SVR training. However when applying the SVR remapping to a general SVBRDF material which does not meet this requirement, we found that the SVR failed to extrapolate the behavior of the transformation outside of the sample space, thus generating fringe changes in the resulting asset.

However, upon careful scrutiny of the functional shapes of the transformations from Section 6, which present common properties along different BRDF model pairs, we are able to formulate a parametric function which capitalizes on these properties and models the behavior of the transformations with only a few parameters. Below we analyze the properties of these transformations and formulate the parametric approach for regression (an implementation of the parametric fitting can be found in the additional material).

In the roughness plots of Section 6 (Figures 12, 16, 17) we empirically observe that the target roughness depends on the source roughness alone, with no further influence by other source parameters. This behavior is easy to model with a simple univariate polynomial fit of low degree (≤ 4). In contrast, the remapping of the specular parameter depends on both the roughness and the specular parts, requiring a more complex functional for the regression. However, the complexity of the regression can be drastically reduced by observing that for a fixed value of roughness, the relationship between specular parameters results in a very low-degree polynomial.

In particular, when the specular parameters are linearly related to the intensity of the BRDF lobe, the specular transformation results in straight lines (in the case of the IOR, as seen in Figure 10(a), the nonlinear change of variable of Equation (1) can be used to recover the specular reflectance). Thus, our parametric model for the specular transformation results in

$$\mathrm{s}_2(\mathrm{s}_1, \alpha_1) = k(\alpha_1) \cdot \mathrm{s}_1,$$

where s and α refer to specular and roughness parameters, and the subindices 1 and 2 indicate source, and target, respectively. With this parametric model, all that we have to do is compute the slope of the curves for each value of roughness, and then use this data to fit the nonlinear relationship $k(\alpha_1)$ between roughness and slope. This can be done with a univariate nonlinear fit with only a few coefficients c, such as

$$k(\alpha) = c_0 + c_1 e^{-c_2 \alpha} + c_3 e^{-c_4 \alpha^2}.$$

In Figure 19 we show the remapping of a spatially varying material by means of both parametric (bottom-center) and SVR (bottom-right) regressions. Due to the incorrect extrapolation of the SVR, the resulting remapping has suffered a very noticeable change in chromaticity. In contrast, the functional shape of our parametric

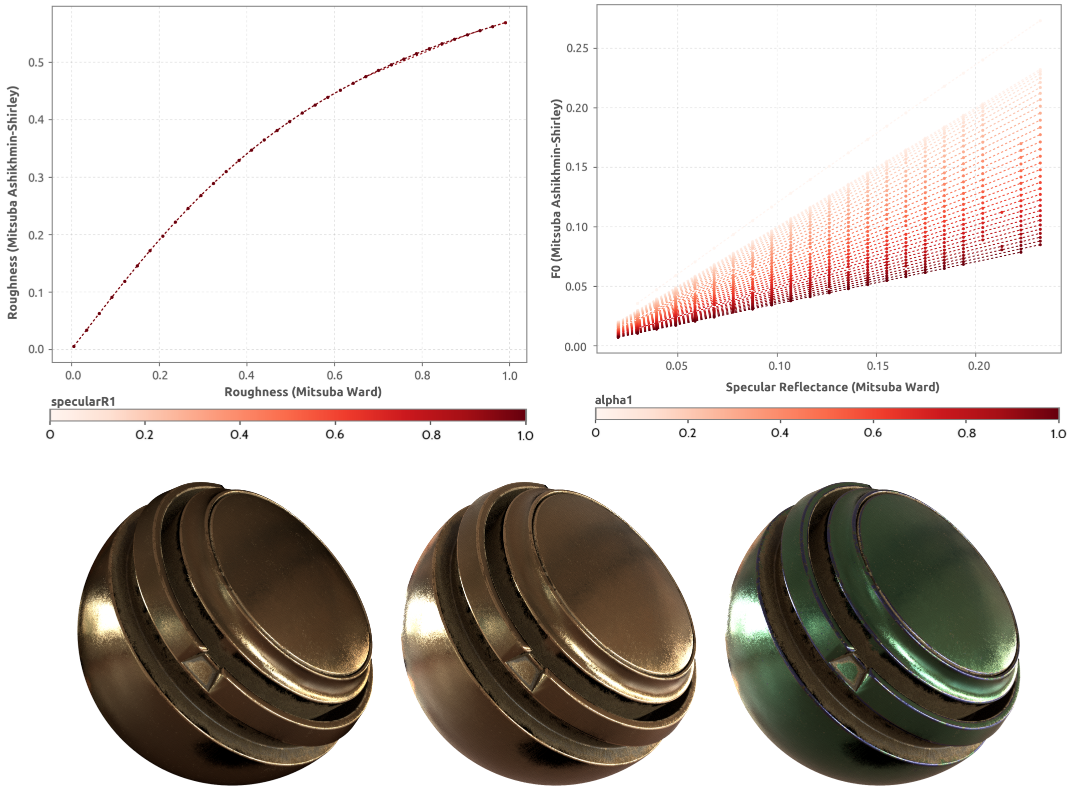


Figure 19. Plots showing the remapping from Mitsuba-Ward to Mitsuba-Ashikhmin-Shirley. The dataset covers a limited portion of the specular reflectance parameter range (top). Renderings of the remapping of a spatially varying material scratched gold) using two different methods for regression. Original Mitsuba-Ward (bottom-left); remapped Mitsuba Ashikhmin-Shirley with our parametric scheme (bottom-center); remapped Mitsuba Ashikhmin-Shirley with SVR (bottom-right). Illumination: *Tabac Plant* environment map [Vogl 2010].

function assumes a linear extrapolation of the specular values. This means that in the transformation of parameters, the three channels of each texel are multiplied by the same factor, thus leading to a conservation of the chromaticity. The numerical results of the parametric fit for all the previously analyzed cases can be found in the Appendix, for both roughness and specular regressions

8. Remapping Between Metallic and Specular Workflows

So far most of the examples we have shown correspond to a *specular workflow*, where the material is specified by at least seven parameters, usually corresponding to diffuse and specular reflectances (RGB), and a single-channel roughness or glossiness. In some renderers, materials are specified through a *metallic workflow*, which provides the user with a reduced number of intuitive parameters: an *albedo* (RGB), and two single-channel parameters for *metallicness* and roughness. The reduction of parameters is based on the observation that in most common materials there is no need for both diffuse and specular reflectances to have three channels, as explained in Sections 4.1.2 and 4.1.1. For metallicness $= 0$ we have a dielectric and the albedo plays the role of diffuse reflectance. For metallicness $= 1$ we obtain a conductor and the albedo determines the color of specular reflectance. Intermediate values of metallicness provide a blend between these two sets of materials. This is reflected in Equation (2) which expresses the transformation from the metallic to the specular parameters:

$$\begin{aligned} \text{diffuse} &= \text{albedo} \cdot (1 - \text{metallicness}), \\ \text{specular} &= \text{lerp}(0.03, \text{albedo}, \text{metallicness}) \\ &= 0.03 + \text{metallicness} \cdot (\text{albedo} - 0.03), \end{aligned} \tag{2}$$

where we have chosen a default specular value 0.03 for dielectrics. Although working with different parameters, the underlying BRDF model is the same for both cases and hence all materials described within the metallic workflow have an exact match in the specular workflow. However the opposite is not true, because of the higher number of free parameters in the specular workflow, and so in general the conversion from specular to metallic is ill-defined. Equation (3) displays the analytic inverse of Equation (2), where i takes the values of the three color channels:

$$\begin{aligned} \text{albedo}_i &= \frac{\text{specular}_i + \text{diffuse}_i}{2} \left(1 + \sqrt{1 - \frac{4 \cdot 0.03 \cdot \text{diffuse}_i}{(\text{specular}_i + \text{diffuse}_i)^2}} \right), \\ \text{metallicness}_i &= \frac{\text{specular}_i - 0.03}{\text{albedo}_i - 0.03}. \end{aligned} \tag{3}$$

In general this will generate different values of *metallicness* for each channel, and so we need to decide how to convert them into the single-channel parameter from the metallic workflow, for which there is no established best practice. We will show two different strategies to perform this conversion and compare them with an implementation of our remapping scheme.

Figure 20 shows the uniform remapping of parameters from the metallic to the specular workflow, comparing with the analytic conversion provided by Equation (2). The dataset of remapped parameters uniformly spans the full range of values in the metallic workflow (albedo and metallicness) in a single color channel. The result is a perfect match between remapping and analytic conversion, which reaffirms the validity of the method.

The inverse conversion from specular to metallic parameters presents multiple additional obstacles for remapping:

- The conversion cannot be done on a per-channel basis; the optimization needs to be done simultaneously with all channels.

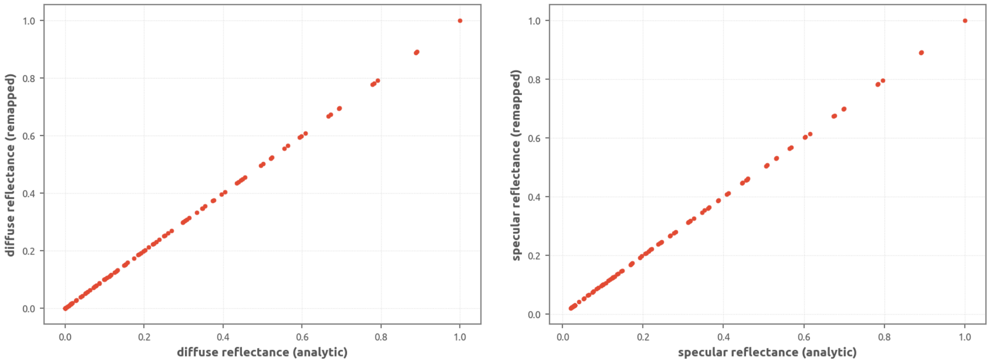


Figure 20. Single-channel uniform remapping from metallic to specular parameters using the GGX BRDF model (Mitsuba). Remapped parameters are plotted against the analytic remapping from Equation (2). Remapped diffuse reflectance vs analytic diffuse reflectance (left); remapped specular reflectance vs analytic specular reflectance (right).

- In the metallic workflow there is no separation between diffuse and specular terms, hence we need to optimize both terms at the same time.
- As explained, in the general case the conversion is ill-defined, and we do not have a reference analytic conversion with which to compare.

As analyzed in previous sections, this combination of factors makes it likely that the optimization will fall into local minima for many materials, and this is indeed what we find. However, we can avoid this by remapping a dataset of parameters that we know to have a well-defined inverse transformation. We generate this dataset of specular parameters by converting a uniformly random generated dataset of metallic parameters with Equation (2). The convergence of the optimization is further helped by the fact that the underlying BRDF model is the same in both workflows.

Once the dataset is generated and remapped, we can fit it with a regression scheme as explained in previous sections and use this learned function to convert materials in the entire parameter space. For this regression, the parametric scheme from Section 5.1 is not applicable for this transformation, and so we must rely on a general purpose regression, such as described in Section 7. In this case, because of the large number of source and target BRDF parameters, we decided to perform the regression with a neural network, which offered a faster convergence in previous tests. The architecture used was a simple fully-connected network with one hidden-layer ($6 \times 300 \times 4$) and ReLU activation.

In Figures 21 and 22 we show the remapping of SVBRDF assets from specular to metallic parameters. The results are compared with analytic remappings using Equation (3) and two different strategies for computing the single-channel metallicness: 1) a simple mean and 2) a mean weighted by the albedo values. In both figures

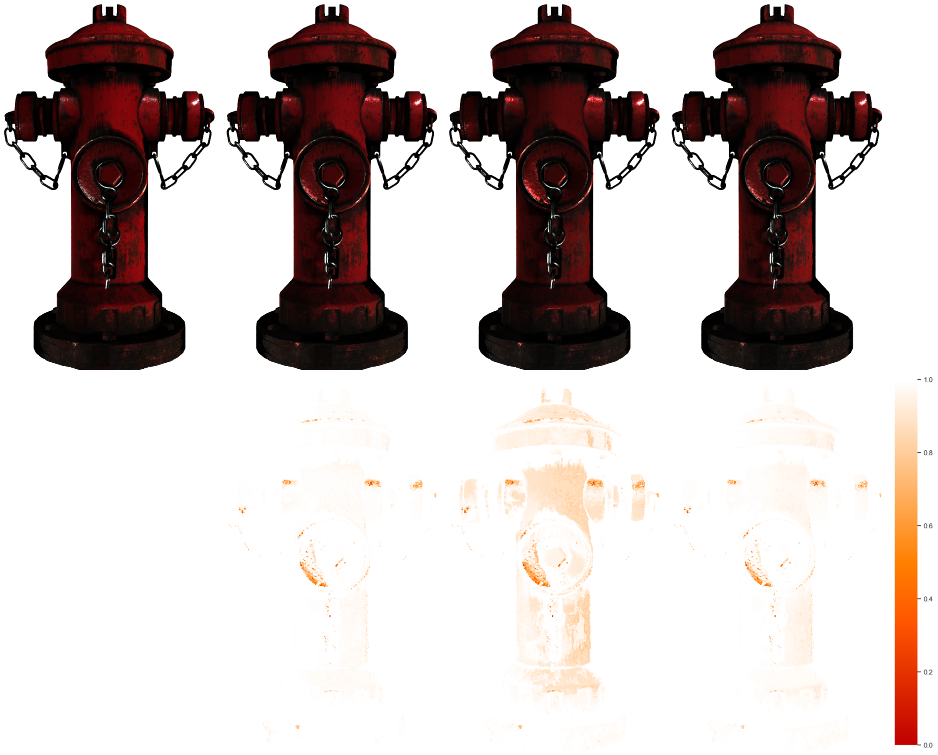


Figure 21. SVBRDF remapping of hydrant from specular to metallic parameters. From left to right: reference, remapping (SSIM: 0.98), analytic remapping with mean metallicness (SSIM: 0.91) and with wighted mean metallicness (SSIM: 0.97).

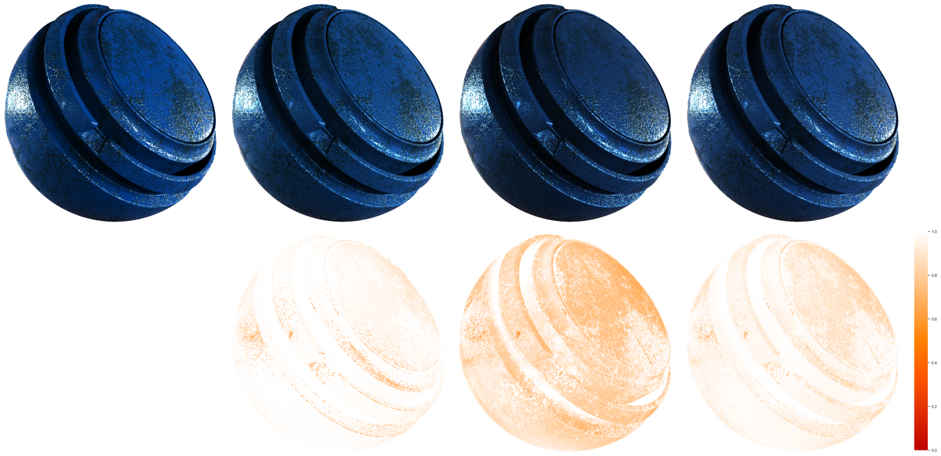


Figure 22. SVBRDF remapping of 'blue plastic' material from specular to metallic parameters. From left to right: reference, remapping (SSIM: 0.995), analytic remapping with mean metallicness (SSIM: 0.9802) and with wighted mean metallicness (SSIM: 0.993).

we observe that our approach produces the best visual match of materials, followed by the mean weighted analytic transformation.

9. Illuminant Position

As previously discussed in Section 4.1, our scheme for uniform BRDF remapping utilizes a rendered scene with point-light illumination to provide a partial sampling of the material's reflectance. The main requirement that we found for the light position during our experiments is that a large-enough proportion of the pixels in the resulting renderings should be illuminated, so that the optimization is able to converge and we obtain a stable transformation. This needs to happen for all considered combinations of parameters, including materials with low roughness where the highlight does not spread far from the direction of specular reflection. An obvious choice for this purpose is headlight illumination, which maximizes the size of the specular

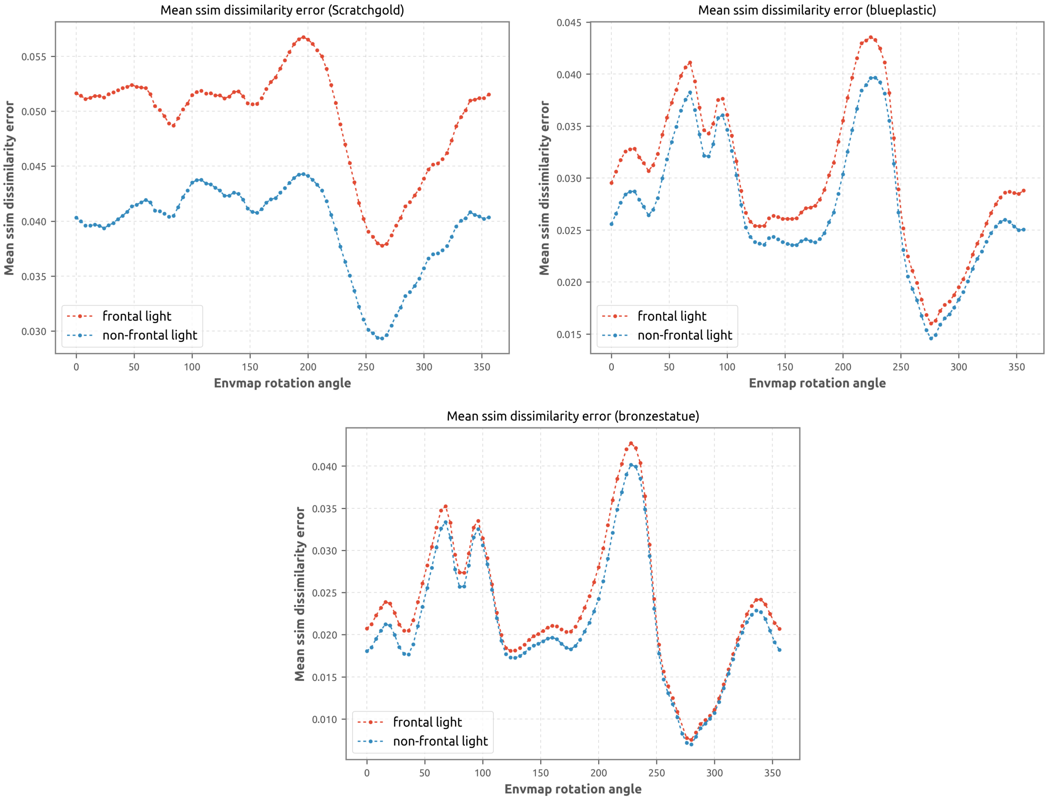


Figure 23. Mean dissimilarity error (1 - SSIM) for three SVBRDF materials as a function of the rotation angle of the illuminating environment map (Tabac Plant [Vogl 2010]). Remapping scene with frontal (red) and non-frontal point light (blue). Higher values indicate higher error. Scratched gold (top-left); blue plastic (top-right); bronze statue (bottom).

highlight in the rendering, thus improving the characterization of the lobe. However, the symmetry of the scene configuration poses multiple potential issues: (1) an over-representation of the retroreflective lobe in the sampling; (2) a repeated sampling of directions which in isotropic BRDFs are equivalent and do not provide new information; (3) for many BRDF models, insufficient coverage of the parameter space to constrain all parameters, for instance the Fresnel term.

In Figures 24–26 we display the remappings of spatially-varying materials with two different light settings: frontal light (top) and non-frontal light with $\theta_l = 45°$ (bottom). Visual differences between these two settings for remapping are hard to spot, but in some cases a slight decrease of the global dissimilarity is apparent when using non-frontal light. This is confirmed by the plots in Figure 23, where we can observe the corresponding mean dissimilarity errors of these materials as we rotate

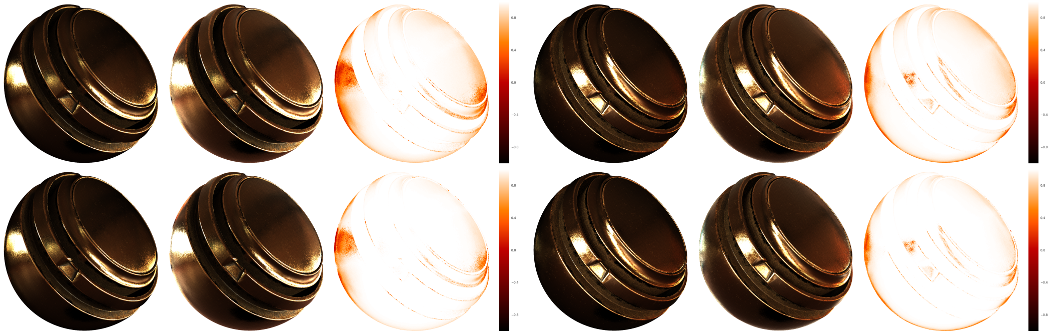


Figure 24. Parametric remappings of scratched gold with frontal (top) and non-frontal light (bottom). Illumination provided by the *Tabac Plant* environment map [Vogl 2010] at rotation angles 0° (left side) and 272° (right side). Within each side: original material in Mitsuba-Ward (left); remapped in Mitsuba-Ashikhmin-Shirley (center); SSIM difference (right).

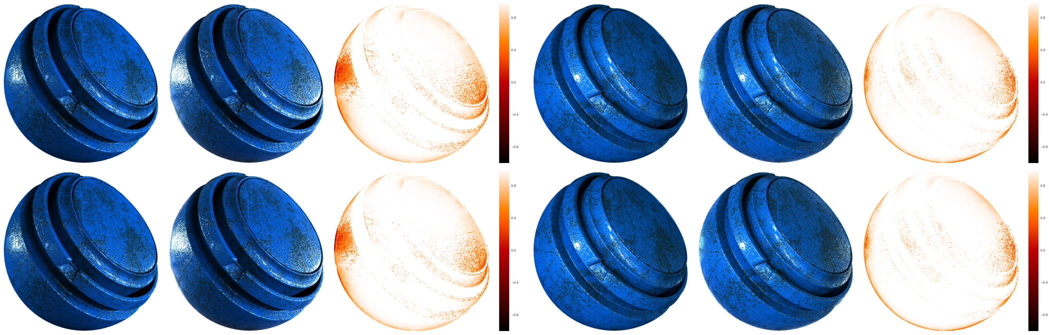


Figure 25. Parametric remappings of blue plastic with frontal (top) and non-frontal light (bottom). Illumination provided by the *Tabac Plant* environment map [Vogl 2010] at rotation angles 0° (left side) and 272° (right side). Within each side: original material in Mitsuba-Ward (left); remapped in Mitsuba-Ashikhmin-Shirley (center);. SSIM difference (right).

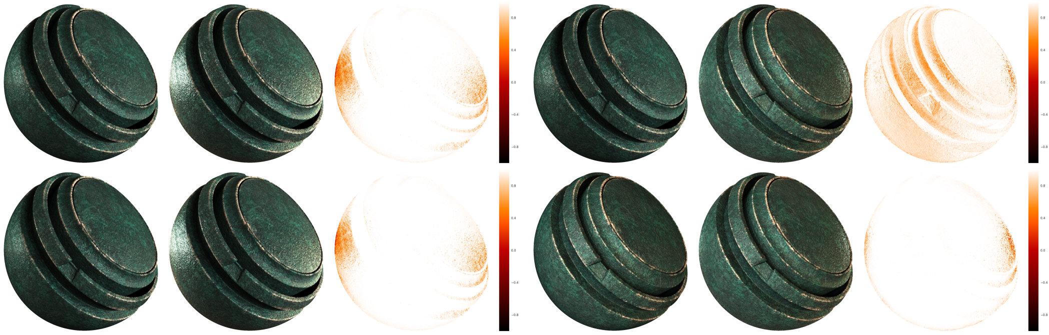


Figure 26. Parametric remappings of bronze statue with frontal (top) and non-frontal light (bottom). Illumination provided by the *Tabac Plant* environment map [Vogl 2010] at rotation angles 0° (left side) and 272° (right side). Within each side: original material in Mitsuba-Ward (left); remapped in Mitsuba-Ashikhmin-Shirley (center); SSIM difference (right).

the environment map illumination (note that we refer to a rotation of the illumination used for the renderings, *not* the one used in the remapping). Videos of remappings with rotating environment illumination can be found in the supplemental material.

10. Conclusions

We presented a method for translation of material appearance between different BRDF models and across different renderers, which uses an image-based metric for appearance comparison, delegating the interaction with the model to the renderer. We analyzed the performance of the method, both with respect to robustness and visual differences for multiple combinations of BRDF models. While it is effective for individual BRDFs, the computational cost does not scale well for spatially varying BRDFs. Hence, we also presented two regression schemes that are able to generate reduced representations of the transformations between BRDF models, that evaluate instantly and without further interaction with the renderer, allowing the remapping of SVBRDF texture maps. Moreover, the resulting transformations lend themselves to chaining, enabling effortless transitions between BRDF models. The first scheme is based on support vector regression and requires a relatively large dataset of uniform remapping data to train, covering the full domain of parameter space. The second regression method presented is based on a nonlinear parametric fit of the uniform remapping data which capitalizes on common properties observed in transformations between brdf models. Although the assumptions used to construct this parametric scheme make it potentially less general than SVR, we show that it provides a better extrapolation of parameters outside of sampled data. In addition, because the parametric scheme models the transformation of specular parameters as a linear relation, the sampling can be very sparse. As a real-world application we implemented the remapping between specular and metallic workflows, comparing the results with analytic transformations. Finally we analyzed the effect of the lighting used during uniform remapping in the quality of renderings of spatially varying materials, and confirmed that headlight illumination leads to a slight increase in visual differences.

A. Parametric Fits

Mitsuba (Ward) → Blender internal (Ward)

$$\alpha_B = 1.01\alpha_M - 0.13\alpha_M^2 - 0.41\alpha_M^3 + 0.24\alpha_M^4$$
$$s_B = s_M\left[-43.4 + 43.6e^{-0.0017\alpha_M} + 0.115e^{-2.21\alpha_M}\right]$$

Cycles (GGX) → Mitsuba (Ashikhmin-Shirley)

$$\alpha_M = 0.78\alpha_C + 1.31\alpha_C^2 - 1.86\alpha_C^3 + 0.61\alpha_C^4$$
$$s_M = s_C\left[-0.067 + 1.43e^{-0.692\alpha_C} - 0.547e^{-6.4\alpha_C} + 0.0002/\alpha_C\right]$$

Mitsuba (GGX) → Cycles (GGX)

$$\alpha_C = 0.98\alpha_M + 0.09\alpha_M^2 - 0.14\alpha_M^3 + 0.066\alpha_M^4$$
$$s_C = s_M\left[1.05 - 0.187e^{-23.45\alpha_M} + 0.036e^{-116.4\alpha_M}\right]$$

Mitsuba (Ward) → Mitsuba (GGX)

$$\alpha_G = 1.135\alpha_W - 0.766\alpha_W^2 + 0.102\alpha_W^3 + 1.08\alpha_W^4$$
$$s_G = s_W\left[-944 + 944.82e^{-0.00046\alpha_W} + 0.476e^{-4.786\alpha_W}\right]$$

Mitsuba (Ward) → Mitsuba (Ashikhmin-Shirley)

$$\alpha_{AS} = 1.04\alpha_W - 0.388\alpha_W^2 0.332\alpha_W^3 + 0.25\alpha_W^4$$
$$s_{AS} = s_W\left[-512.13 + 512.75e^{-0.00057\alpha_W} + 0.398e^{-3.12\alpha_W}\right]$$

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement № 642841. We would like to thank Cyrille Damez from Allegorithmic for his contribution to the project.

References

ALLEGORITHMIC, 2017. Allegorithmic: Substance designer and substance painter. https://www.allegorithmic.com. Last access Oct. 2017. 3

ASHIKHMIN, M., AND SHIRLEY, P. 2000. An anisotropic Phong BRDF model. *J. Graph. Tools 5*, 2 (Feb.), 25–32. URL: http://dx.doi.org/10.1080/10867651.2000.10487522. 3

BRADY, A., LAWRENCE, J., PEERS, P., AND WEIMER, W. 2014. genBRDF: Discovering new analytic BRDFs with genetic programming. *ACM Trans. Graph. 33*, 4 (July), 114:1–114:11. URL: http://doi.acm.org/10.1145/2601097.2601193. 2

Branch, M., Coleman, T., and Li, Y. 1999. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing 21*, 1, 1–23. URL: https://doi.org/10.1137/S1064827595289108. 7

Burley, B. 2012. Physically-based shading at disney. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12. ACM, New York, NY. 3

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2*, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 13

Corona, 2017. Corona renderer. https://corona-renderer.com. Last access Oct. 2017. 2, 3

Damez, C., Oct. 2017. Personal communication (CTO, Allegorithmic). 3

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00. ACM Press/Addison-Wesley Publishing Co., New York, NY, 145–156. URL: http://dx.doi.org/10.1145/344779.344855. 5

Guarnera, D., Guarnera, G., Ghosh, A., Denk, C., and Glencross, M. 2016. BRDF representation and acquisition. *Comput. Graph. Forum 35*, 2 (May), 625–650. URL: https://doi.org/10.1111/cgf.12867. 2, 3

Guarnera, D., Guarnera, G. C., Toscani, M., Glencross, M., Li, B., Hardeberg, J. Y., and Gegenfurtner, K. R. 2018. Perceptually validated analytical BRDFs parameters remapping. In *ACM SIGGRAPH 2018 Talks*, SIGGRAPH '18. ACM, New York, NY, 17:1–17:2. URL: http://doi.acm.org/10.1145/3214745.3214807. 3

Havran, V., Filip, J., and Myszkowski, K. 2016. Perceptually Motivated BRDF Comparison using Single Image. *Computer Graphics Forum*. doi:10.1111/cgf.12944. 4

Jakob, W., 2010. Mitsuba renderer. http://www.mitsuba-renderer.org. 7

Jones, E., Oliphant, T., Peterson, P., et al., 2001–. SciPy: Open source scientific tools for Python. [Online; accessed 04-09-2019]. URL: http://www.scipy.org/. 7

Krivanek, J., Jul. 2017. Personal communication (Head of Research, Corona Renderer). 3

Lafortune, E. P. F., Foo, S.-C., Torrance, K. E., and Greenberg, D. P. 1997. Non-linear approximation of reflectance functions. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, SIGGRAPH '97, 117–126. URL: http://dx.doi.org/10.1145/258734.258801. 3

Marschner, S. R. 1998. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University, Ithaca, NY. 3

NGAN, A., DURAND, F., AND MATUSIK, W. 2005. Experimental analysis of BRDF models. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05. Eurographics Association, Aire-la-Ville, Switzerland, 117–126. URL: http://dx.doi.org/10.2312/EGWR/EGSR05/117-126. 3

NGAN, A., DURAND, F., AND MATUSIK, W. 2006. Image-driven navigation of analytical brdf models. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, EGSR '06. Eurographics Association, Aire-la-Ville, Switzerland, 399–407. URL: http://dx.doi.org/10.2312/EGWR/EGSR06/399-407. 4, 5

NICODEMUS, F., C. RICHMOND, J., J. HSIA, J., W. GINSBERG, I., AND L. LIMPERIS, T. 1977. *Geometrical Considerations and Nomenclature for Reflection, NBS Monograph 160*. National Bureau of Standards, Washington, D.C. 3

PASZKE, A., GROSS, S., CHINTALA, S., CHANAN, G., YANG, E., DEVITO, Z., LIN, Z., DESMAISON, A., ANTIGA, L., AND LERER, A. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*. 13

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830. 13

PHARR, M., JAKOB, W., AND HUMPHREYS, G. 2016. *Physically Based Rendering, Third Edition: From Theory To Implementation*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 2, 3

SCHREGLE, R., DENK, C., SLUSALLEK, P., AND GLENCROSS, M. 2013. Grand Challenges: Material Models in the Automotive Industry. In *Eurographics Workshop on Material Appearance Modeling*. Eurographics Association, Aire-la-Ville, Switzerland. URL: http://dx.doi.org/10.2312/MAM.MAM2013.001-006. 2

SZTRAJMAN, A., KRIVANEK, J., WILKIE, A., AND WEYRICH, T. 2017. Image-based remapping of material appearance. In *Proc. 5th Workshop on Material Appearance Modeling (non-peer-reviewed presentation)*. The Eurographics Association, Aire-la-Ville, Switzerland, June, 5–8. URL: http://reality.cs.ucl.ac.uk/projects/reflectance-remapping/sztrajman17image-based.html. 3

2017. Unity. https://unity3d.com/. Last access Oct. 2017. 2

2017. Unreal engine. https://www.unrealengine.com. Last access Oct. 2017. 2

VOGL, B., 2010. Lightprobe of former tabac plant. http://dativ.at/lightprobes. [Online; accessed 15-April-2018]. 19, 23, 24, 25

WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07. Eurographics Association, Aire-la-Ville, Switzerland, 195–206. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206. 3

WEYRICH, T., MATUSIK, W., PFISTER, H., BICKEL, B., DONNER, C., TU, C., MCANDLESS, J., LEE, J., NGAN, A., JENSEN, H. W., AND GROSS, M. 2006. Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (July), 1013–1024. URL: http://doi.acm.org/10.1145/1141911.1141987. 5

WEYRICH, T., LAWRENCE, J., LENSCH, H., RUSINKIEWICZ, S., AND ZICKLER, T. 2008. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4*, 2, 75–191. URL: http://dx.doi.org/10.1561/0600000022. 3

Index of Supplemental Materials

The supplemental material can be found at http://jcgt.org/published/0008/04/01/supplemental.zip.

animations/index.html HTML page with animations of remapped materials from Section 9.

source_code/USAGE.txt Detailed description of source code files and usage.

source_code/mitsuba-ward2mitsuba-beck.csv Table with parameters from uniform remapping between Mitsuba-Ward and Mitsuba-Beckmann BRDFs.

source_code/parametric_remapping.py Python script that remaps the parameter textures (roughness.exr, specular.exr) from Mitsuba-Ward to Mitsuba-Beckmann. The code uses the uniform remapping data from mitsuba-ward2mitsuba-beck.csv to train a parametric fit and perform the remapping.

source_code/svr_remapping.py Similar to parametric_remapping.py, this script trains an SVR to perform the remapping.

source_code/pbr.cpp Plugin for Mitsuba PBR rendering. Fresnel function can use F0 as input (Equation (1)).

Author Contact Information

Alejandro Sztrajman
University College London
Malet Place, London WC1E 6BT, UK
a.sztrajman@ucl.ac.uk

Jaroslav Křivánek
Charles University
Malostranské náměsti 25
118 00 Praha 1, Czech Republic
jaroslav.krivanek@mff.cuni.cz

Alexander Wilkie
Charles University
Malostranské náměsti 25
118 00 Praha 1, Czech Republic
jwilkie@cgg.mff.cuni.cz

Tim Weyrich
University College London
66–72 Gower Street
WC1E 6EA, London, UK
t.weyrich@cs.ucl.ac.uk

A. Sztrajman, J. Křivánek, A. Wilkie, T. Weyrich, Image-based Remapping of Spatially Varying Material Appearance, *Journal of Computer Graphics Techniques (JCGT)*, vol. 8, no. 4, 1–30, 2019
http://jcgt.org/published/0008/04/01/

Received:	2019-03-13		
Recommended:	2019-06-03	Corresponding Editor:	Naty Hoffman
Published:	2019-10-31	Editor-in-Chief:	Marc Olano