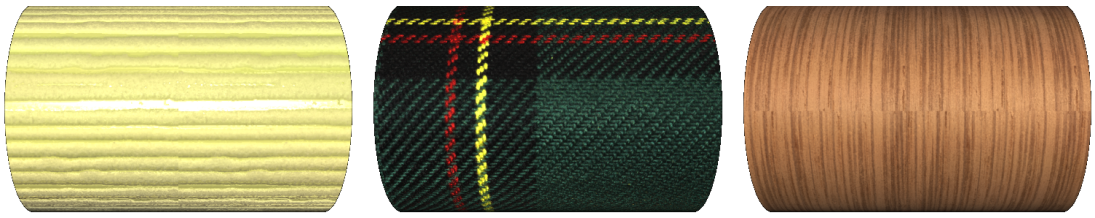
Vision, Modeling, and Visualization (2015)
D. Bommes, T. Ritschel, and T. Schultz (Eds.)

Extrapolating Large-Scale Material BTFs under Cross-Device Constraints

H. C. Steinhausen, D. den Brok, M. B. Hullin, R. Klein

Institut für Informatik II, Universität Bonn, Germany



(a) *Wallpaper1* 256 × 256 *(tiled).* **(b)** *Cloth9* 512 × 512. **(c)** *Wood4* 256 × 256 *(tiled).*

Figure 1: *Result renderings for extrapolation guided by scanned images.*

Abstract

In this paper, we address the problem of acquiring bidirectional texture functions (BTFs) of large-scale material samples. Our approach fuses gonioreflectometric measurements of small samples with few constraint images taken on a flatbed scanner under semi-controlled conditions. Underlying our method is a lightweight texture synthesis scheme using a local texture descriptor that combines shading and albedo across devices. Since it operates directly on SVD-compressed BTF data, our method is computationally efficient and can be implemented on a moderate memory footprint.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism — Color, shading, shadowing, and texture

1. Introduction

The important role of material models in computer graphics has been known for a long time; yet, to this day, the acquisition of data-driven models from real-world material samples still underlies many practical limitations. Photographically measured bidirectional texture functions (BTFs) as introduced by Dana et al. [DNvGK97] can recreate a wide range of optical phenomena but require long capture times and large measurement setups to satisfy the need for far-field illumination for large material samples. Most setups support sample sizes not larger than 10 cm × 10 cm, indicating the demand for alternative methods to capture the appearance of expanded samples.

In this work, we propose to combine full bidirectional measurements of a small sample with sparse measurements of a large-scale material sample in order to synthesize a fully relightable large-scale representation of the material. The proposed method relies on the observation that BTFs are highly redundant and that small samples are often representative of the overall appearance. In particular, research on BTFs has shown that they can be expressed efficiently in terms of a basis of eigentextures and eigen-BRDFs (to be precise, *apparent* BRDFs or ABRDFs incorporating non-local shading [WHON97]). Inspired by this insight, we devise a texture synthesis scheme that extrapolates an existing BTF with guidance from additional constraint images. Our constraints are provided by differently shaded images which we obtain by placing the material sample on a flatbed scanner in different orientations. After aligning these constraint images, we can extract, for each texel, a feature vector that combines shading and albedo terms. These features are laid

DOI: 10.2312/vmv.20151269

out such that they can be transferred between dissimilar capture devices. The texture synthesis then translates to a simple nearest-neighbor search that, for each texel in the stack of constraint images, finds a suitable ABRDF in the small-scale BTF dataset. Furthermore, we demonstrate compatibility of the feature vectors with other texture synthesis methods like Kwatra et al.'s texture optimization [KEBK05]. We give an evaluation against ground-truth data sets, supplemented with a demonstration of the practicability in real-world scenarios.

2. Related Work

In this section, we provide a short overview on literature related with BTFs in general, as well as on synthesis of larger textures or reflectance datasets from sparse ones.

2.1. Bidirectional Texture Functions

Bidirectional Texture Functions (BTFs), as introduced by Dana et al. [DNvGK97], are an extension of *bidirectional reflectance distribution functions (BRDFs)* to the spatial domain. They are not defined with respect to the true object surface but to some planar interface through which light is exchanged. Unlike surface-referenced *spatially varying BRDFs (SVBRDFs)*, this allows BTFs to account for non-local shading effects like interreflections, self-shadowing and sub-surface scattering. Formally, a BTF can be formulated as a six-dimensional function $\mathcal{B}(x,y,\theta_i,\phi_i,\theta_v,\phi_v) \mapsto r$ of surface position (x,y), lighting direction (θ_i,ϕ_i) and viewing direction (θ_v,ϕ_v); accounting for wavelength adds another dimension. A more thorough description and comparison of different reflectance models can be found in several textbooks and reports [MMS*05, WLL*09, HF13] .

Our research is centered around image-based BTF representations which are acquired using camera domes as proposed by Müller et al. [MMS*05] and Schwartz et al. [SSWK13]. For a comparison of several capturing devices, see e.g. the survey by Schwartz et al. [SSW*14]. The material samples used throughout our experiments belong to the database published by Weinmann et al. [WGK14]. To handle the datasets sized up to two terabytes, we apply the *full matrix factorization* compression scheme [KMBK03] based on truncated singular value decomposition (TSVD).

2.2. Expansion and Completion of (Sparse) Datasets

The intuition of BTFs as stacks of textures motivates the use of texture synthesis algorithms on BTFs. A survey by Wei et al. [WLK*09] provides an overview of example-based texture synthesis until the year 2009. Several attempts have been taken to apply such schemes to BTFs. For example, Tong et al. propose to synthesize a new BTF directly onto a surface in a pixel-wise manner [TZL*02]. Other methods rely on image quilting [ZDW*05] or tiling [HH05, LPF*07] or focus on special cases like textures with complex geometry such as fur [FHNK05].

The extent to which these methods support the goal of faithfully capturing the visual appearance of a large-scale sample is very limited. For SVBRDFs, an approach toward this goal was given by Dong et al. [DWT*10]. While their idea is similar to ours in that it combines spatially dense but angularly sparse *key measurements* with a small set of representative BRDFs, a straightforward extension of this method to BTFs is obstructed by the presence of non-local effects in the contained ABRDFs.

An approach specifically for BTFs was presented by Filip et al. [FVK14] who combine sparse reflectance measurements with a method to construct approximate BTFs. Their portable gonioreflectometer is capable of handling samples of size 30 cm × 30 cm; for even larger sample, the authors suggest to scale the device or perform a patch-wise measurement. On the other hand, flatbed scanners as employed in our approach are readily available in very large sizes.

Miandji et al. [MKU15] devise a compressed sensing framework to reconstruct images and higher-dimensional pendants from noisy image sets using pre-trained dictionaries from natural images. They demonstrate their method on 4D light fields, which suggests applicability to sparsely measured BTFs. However, not only is the method quite compute-intensive, it also requires a large number of samples,while our method operates on four differently lit top-view images.

In this work, we revisit the method recently proposed by Steinhausen et al. [SMdB*15], who use guided texture synthesis to extrapolate BTFs for large-scale material samples. Starting point is a fully measured, TSVD-compressed BTF for a cut-out of the sample. Additionally, four images of the full-size sample are acquired using a flatbed scanner. From these images, color constraints and normals-like surface descriptors are generated which then serve as the guiding constraints for a pixel-based texture synthesis algorithm. To this end, the authors borrow the idea of Woodham's *Photometric Stereo* [Woo80] and apply it to the flatbed scanner images [PMW*09, PS13]. Unlike these works, however, the goal is not to obtain accurate surface mesostructure which would suffer from nonlocal illumination effects. We argue that, even when looking at a single pixel, the presence of interreflections and shadowing provides cues about its neighborhood that can be a valuable source of guidance for synthesizing the local appearance. As an advantage over Filip's method, this input setting offers a higher angular and spatial resolution of the BTF data and a higher spatial resolution of the guiding constraints. We evaluate Steinhausen's method [SMdB*15] with special regard to error measures on ground-truth data.

3. Method Overview

In this section, we give an overview of our BTF acquisition which builds upon Steinhausen et al.'s method [SMdB*15]. It consists of two main phases: An acquisition of sparse input

data is followed by an extrapolation step. The following two subsections are dedicated to the preparation of input data, while Sec. 3.3 deals with the process of assembling a new BTF from the prepared data sets.

3.1. BTF Acquisition

We use BTF datasets captured using the Dome I device [MMS*05]. Each measurement comprises a set of 22801 images of a material patch sized about 5 cm × 5 cm, taken from 151 camera positions under 151 different lighting angles, leading to an average angular sampling of $9.4^\circ \pm 1^\circ$. The spatial extent of the resulting images is 512×512 texels, of which we select regions exhibiting all desired patterns and reflectance effects to serve as extrapolation inputs.

The image stack resulting from the measuring process is arranged in a matrix $\mathcal{S}$, where each row represents one specific combination of incoming and outgoing light directions and color, while the columns contain unrolled ABRDFs. Compression using truncated SVD yields a factorized representation approximating the original matrix: $\mathcal{S} \approx \mathcal{S}' = U\Sigma V^T$. The columns of U are usually referred to as the *eigen-ABRDFs* of $\mathcal{S}'$, while the columns of V are its *eigen-textures*. ΣV^T, stored as $V\Sigma$ in our BTF representation, serves as the input sample for the extrapolation step.

3.2. Constraint Acquisition

The search for a suitable ABRDF from the input BTF for each position in the output is based on comparing *constraint vectors*. The acquisition and assembly of these vectors is the topic of this section.

3.2.1. Color Images – Scanner

Images of full-size material samples are captured using an Epson Perfection V550 Photo flatbed scanner at a resolution of 1200 dpi and 16 bits per color channel and any automatic color correction facilities deactivated. From a scan of a color standard (X-Rite ColorChecker), we learn that the scanner applies a gamma exponent of roughly 1.8. We apply an offset, a linear factor and a gamma curve for each color channel in order to align the scanner's color space with the reference target (Table 1). This establishes radiometric linearity of our data.

After this color correction step, followed by manual alignment and cropping of the images, we obtain a set $\mathcal{I}^{S_{cc}}$ of four images

$$\mathcal{I}^{S_{cc}} = \left\{ I^{S_{cc}}_{0^\circ}, I^{S_{cc}}_{90^\circ}, I^{S_{cc}}_{180^\circ}, I^{S_{cc}}_{270^\circ} \right\}.$$

3.2.2. Color images – BTF

To obtain corresponding constraint images for the fully measured, compressed BTF $\mathcal{S}'$, four images are extracted taken from the topmost viewing position under lighting angles

Table 1: *Values used for color correction of material scans.*

(a) *Reflectance values assumed for the ColorChecker grey patches [Mye10].*

Black	N3.5	N5	N6.5	N8	White
3.10	9.11	19.54	37.20	60.90	94.76

(b) *Correction values applied to color channels of scanned images.*

	R	G	B
offsets (o)	0.01	0.01	0.00
multipliers (m)	1.18	1.14	1.10
gammas (γ)	1.71	1.83	1.89

$\theta_i = 22^\circ$ and $\phi_i \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. This provides us with a set $\mathcal{I}^B$ of four RGB images

$$\mathcal{I}^B = \left\{ I^B_{0^\circ}, I^B_{90^\circ}, I^B_{180^\circ}, I^B_{270^\circ} \right\}.$$

To account for the differences between the color characteristics of the BTF acquisition device and those of the scanner, a linear scaling between the two data sets $\mathcal{I}^{S_{cc}}$ and $\mathcal{I}^B$ is performed. A scaling factor f_c is multiplied to the intensity values of color channel c:

$$I^S_r(x,y,c) = f_c \cdot I^{S_{cc}}_r(x,y,c), \tag{1}$$

with

$$f_c = \frac{\mu(\mathcal{I}^B, c)}{\mu(\mathcal{I}^{S_{cc}}, c)}, \tag{2}$$

where $\mu(\mathcal{I}, c)$ denotes the arithmetic mean over all pixels in all images $I \in \mathcal{I}$. Figure 2 illustrates the effect of this histogram alignment step for images of material "Cloth9".

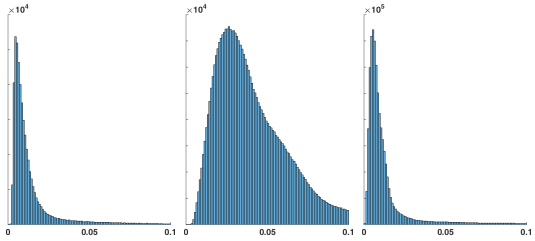


Figure 2: *Histogram alignment between scans and BTF images for "Cloth9": Cut-outs of histograms for red color channel of BTF image (left), scans before (middle) and after alignment (right).*

3.2.3. Surface Descriptors

To supply the extrapolation algorithm with stronger hints on a material's structure, the input constraints contain a descriptor for local surface curvature. To approximate "full" photometric stereo, difference images were chosen to serve as texture invariant curvature descriptors for nearly flat objects.

After reducing each image of an RGB image set $\mathcal{I} = \{I_{0^\circ}, I_{90^\circ}, I_{180^\circ}, I_{270^\circ}\}$ to its luminance channel in CIELAB color space, $\widetilde{\mathcal{I}} = \left\{\widetilde{I}_{0^\circ}, \widetilde{I}_{90^\circ}, \widetilde{I}_{180^\circ}, \widetilde{I}_{270^\circ}\right\}$, difference images are computed:

$$\widetilde{n} = \left[\widetilde{n}_1, \widetilde{n}_2\right] = \left[\widetilde{I}_{0^\circ} - \widetilde{I}_{180^\circ}, \widetilde{I}_{270^\circ} - \widetilde{I}_{90^\circ}\right]. \tag{3}$$

To compensate for a possibly non-uniform distribution of luminance values between images obtained by the scanner and the BTF acquisition device, $\widetilde{n}$ is scaled to $n = [n_1, n_2]$ with

$$n_i = \frac{\widetilde{n}_i - \mu(\widetilde{n}_i)}{\sigma(\widetilde{n}_i)}, \qquad i = 1, 2, \tag{4}$$

where the scalar value $\mu(\widetilde{n}_i)$ is subtracted from all entries of $\widetilde{n}_i$, and μ, σ are arithmetic mean and standard deviation.

Applying Eq. 3 and Eq. 4 to the image sets $\mathcal{I}^B$ and $\mathcal{I}^S$ finally yields the surface descriptors n^B and n^S for the sample BTF and the scans.

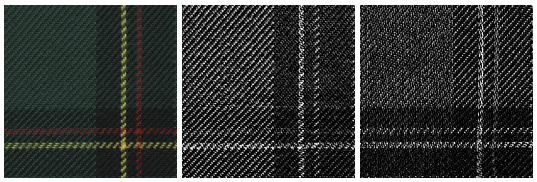
3.2.4. Constraint Vector Assembly

To form the *constraint vectors* guiding the extrapolation step, color images and surface descriptors are concatenated. In order to balance the influence of these components, weighting factors are applied:

$$C_0^X(x,y) = \left[w_s \cdot n^X(x,y,:), w_c \cdot I^X(x,y)\right], \tag{5}$$

where $X \in \{B, S\}$, $I^X(x,y)$ is the concatenation of $I^X_{0^\circ}(x,y,:)$ to $I^X_{270^\circ}(x,y,:)$, and $I(x,y,:)$ denotes enrolling of all channels of I for position (x,y). Figure 3 illustrates part of the information combined in the constraint vectors for material sample "Cloth9".

Intuitively, incorporating neighborhood information into the search criteria seems to be beneficial for texture synthesis. The *extended constraint vector* with radius R for position (x,y) is thus defined as the concatenation of the constraint vectors $C_0^X(x,y)$ for the respective local neighborhood of size $(2R+1) \times (2R+1)$. To match the positions in the rows of a BTF's eigentextures $V\Sigma$ or $\widetilde{V}\Sigma$, the $C_R^X(x,y)$ are unrolled into vectors $C_R^X(i)$.



(a) I_{0°. **(b)** $I_{0^\circ} - I_{180^\circ}$. **(c)** $I_{90^\circ} - I_{270^\circ}$.

Figure 3: *Extrapolation constraints for "Cloth9": scan* I_{0° *and the resulting surface descriptors, cropped to* 512×512 *pixels each.*

3.3. Extrapolation

The extrapolation step aims at creating a compressed representation $\widetilde{\mathcal{S}} = U\Sigma\widetilde{V}^T$ of the full material sample's BTF based on the input BTF $\mathcal{S}' = U\Sigma V^T$. Under the assumption that all relevant reflectance effects are modeled in the original set U of ABRDFs, only a new set $\widetilde{V}\Sigma$ of eigentextures needs to be synthesized. The task is now to find, for each row j in the output matrix $\widetilde{V}\Sigma$, a suitable row i in the input matrix $V\Sigma$.

We evaluate two approaches to this problem: applying texture optimization which searches for a global optimum of local neighborhood similarities [SdBHK14], and a simple pixel-based method [SMdB*15] which works as follows: For each row j in $\widetilde{V}\Sigma$, an index i into the rows of $V\Sigma$ is searched which minimizes the distance between the respective extended constraint vectors:

$$i(j) = \arg\min_{i'} \left\| C^B(i') - C^S(j) \right\|. \tag{6}$$

After the search process, the resulting eigentexture set is constructed by copying, for each j, the contents of row $i(j)$ of $V\Sigma$ into the j-th row of $\widetilde{V}\Sigma$.

4. Experiments and Results

In this section, we first evaluate Steinhausen's methods [SdBHK14, SMdB*15] by reconstructing ground truth data. This is followed by examples of the visual quality achievable using cross-device constraints.

4.1. Ground-truth Reconstruction

The evaluation of our MATLAB implementation of the pixel-based synthesis scheme was performed on a desktop computer built around an Intel Core i7-2600K CPU at 3.4 GHz with 16 GB of RAM. Due to limitations in hardware accessibility, running times for texture optimization [KEBK05] were measured using a CUDA-enabled C++ implementation running on an Intel Xeon E5645 at 2.4 GHz with 144 GB of RAM, supported by an NVidia GeForce GTX 570 graphics accelerator. When comparing the latter timing values to previously reported ones [SdBHK14], they appear similar to those achievable with the Core i7-2600K.

The six input data sets originate from a BTF database of fully measured BTFs [WGK14] with a spatial extent of 512×512 texels each, compressed to keep $k = 100$ eigenvalues. From these BTFs (*Cloth9*, *Cloth10*, *Leather4*, *Leather6*, *Wallpaper1*, *Wood4*), regions with a size of 128×128 texels constitute the BTF sample for the algorithm. This extraction was performed before compression, such that no information from the region to be reconstructed could bias the extrapolation input. For material sample "Cloth9", we exceptionally chose an area of 256×256 texels to capture the full range of color variations. As a substitute for scanned images, four textures of size 512×512 pixels were taken from the full BTF, just as described in Sec. 3.2.2.

Texture optimization was operated in a multi-scale fashion on three consecutive combinations of downsampling scales (4, 2, 1) with neighborhood radius $R = 8$. For all experiments involving pixel-wise synthesis, one pass on the full-scale data was performed. Four trials, with different neighborhood radii $R \in \{0,1,2\}$ and combinations of weights for color (w_c) and structure (w_s) constraints were applied.

For visual comparison of the reconstruction quality, see Figures 4 to 8 with renderings of reference BTFs and reconstructions for five material samples. Table 2 provides relative errors with respect to the decompressed reference BTFs in percent, together with timing values. To shed light on the qualitative variation even inside one semantic class of materials, values for another leather not depicted in the figures ("Leather6") are included. The error e between reference BTF matrix $\mathcal{S}'$ and reconstruction $\widetilde{\mathcal{S}}_{\text{rec}}$ was computed by

$$e = 100 \cdot \frac{\|\mathcal{S}' - \widetilde{\mathcal{S}}_{\text{rec}}\|_F}{\|\mathcal{S}'\|_F}.$$

Although the relative error is lowest for texture optimization except for "Wood4", the visual quality of the results is clearly inferior to that using the pixel-wise synthesis scheme. For leather, e.g. the global structure is recognizable, but the overall impression is blurred. For "Cloth9", severe color artifacts are introduced where the input contains an out-of-focus spot. Additionally, the running times are significantly larger (by a factor of 1.4 to 2) than for all tested variants of the pixel-wise method [SMdB*15]. Please note that although the computations did not take place on the same computer, at least approximate comparability is preserved due to the above-mentioned argument. One strategy to avoid the blur would be the addition of further optimization steps incorporating statistical synthesis [PS00, SdBHK14], but at the cost of a noticeable additional growth of time consumption.

Of the four parameter settings for the pixel-wise extrapolation, using color images as only constraints outperforms the others with regard to result quality as well as speed-wise. For "Wood4", a finished wood example with rather uniform surface, it seems reasonable that mostly color determines the overall appearance. But even for materials where one would expect a primacy of surface structure over color like for the leathers or the wallpaper, the benefit of the surface descriptors is less noticeable than expected. One possible explanation for this effect might be the fact that the shading information condensed into the surface descriptor is also encoded in the color images, hampering a separation of each components' effect. A rather a problematic case for modeling and reconstruction is "Cloth10" due to its chaotic fiber structure and bright highlights. Even this sample is reproduced at least recognizable, although its overall "fluffyness", as well as some contrast especially in the darker areas, is lost in all reconstructions.

4.2. BTFs for Extended Material Samples

The input constraints for the results presented in this subsection were achieved using the scanner-based workflow described in Sec. 3.2.1. To match the image resolution of the BTF's textures, the 1200 dpi scans were downsampled to 300 dpi. On the BTF-side, again regions of size 128×128 or 256×256 ("Cloth9") texels were extracted from a full measurement. Fig. 1 displays examples of the achievable BTF quality. For identical BTF inputs, one can see an advance in detail compared to the ground truth reconstructions. The quality of these results justifies the assumption that cross-device constraints are helpful in obtaining BTF data sets. All result BTFs were generated with $R = 0, w_s = 0, w_c = 1$, causing running times of about 300 seconds for wallpaper and wood, both sized 256×256 texels. The cloth example took about 3000 seconds. The increase in processing time is because part of the feature vector construction (surface descriptor generation) for this experiment took place before image downsampling to achieve a higher amount of detail.

5. Discussion

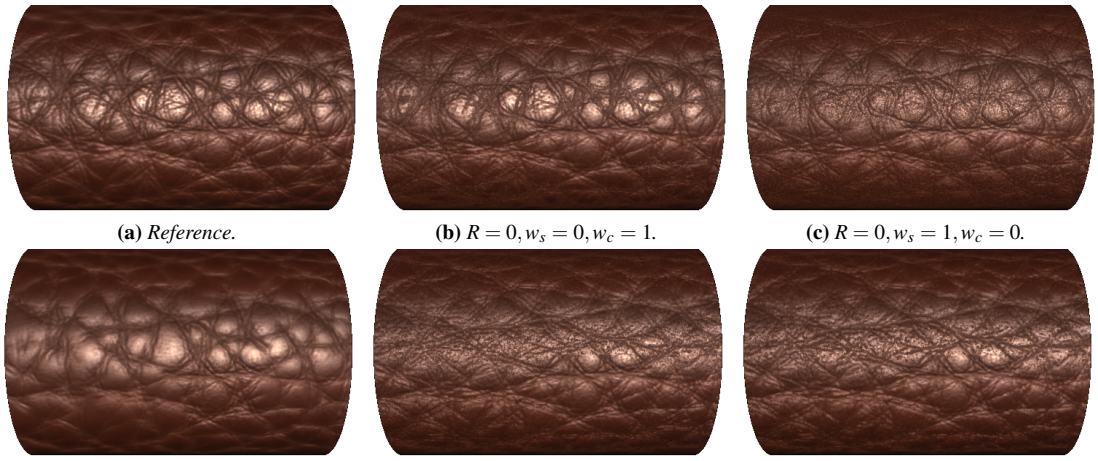
We have evaluated the applicability of two texture synthesis-based extrapolation schemes to achieve BTFs for large-scale material samples. Our results indicate that a fast and simple approach significantly outweighs more sophisticated ones. The findings in this paper also encourage to combine sparse data sets acquired in a cross-device fashion to obtain BTFs for sample sizes not achievable with current acquisition devices. Further reduction of acquisition effort might be possible by a combination with methods to reconstruct angularly sparse data sets, as provided by den Brok et al. [dBSHK14].

More elaborate methods to divide shading from albedo might help in further investigating the influence of structure-induced shading over color. Furthermore, fast and easy-to-use image alignment methods and extensions to constraints acquired under less controlled conditions, e.g. with a consumer camera and a hand-held light source, might be worth studying, see e.g. work by Wu et al. [WLDW11] who aim at improving geometry by combining multi-view stereo and photometric stereo.

Finally, some of the error values reported in Tab. 2 poorly reflect the visual impression, see e.g. the values for texture optimization compared to the pixel-wise synthesis scheme with $R = 0, w_s = 0, w_c = 1$. On the other hand, the available alternatives bear other disadvantages. As an example, the mean ABRDF RMSE used in other publications [RRK09, SdBHK14] does not allow a comparison between the reconstruction accuracy of different materials. Thus, further research in finding quality measures reflecting the perceived similarity of material representations would be of major importance.

Table 2: *Error values and timings for reconstruction of ground-truth material BTFs.*

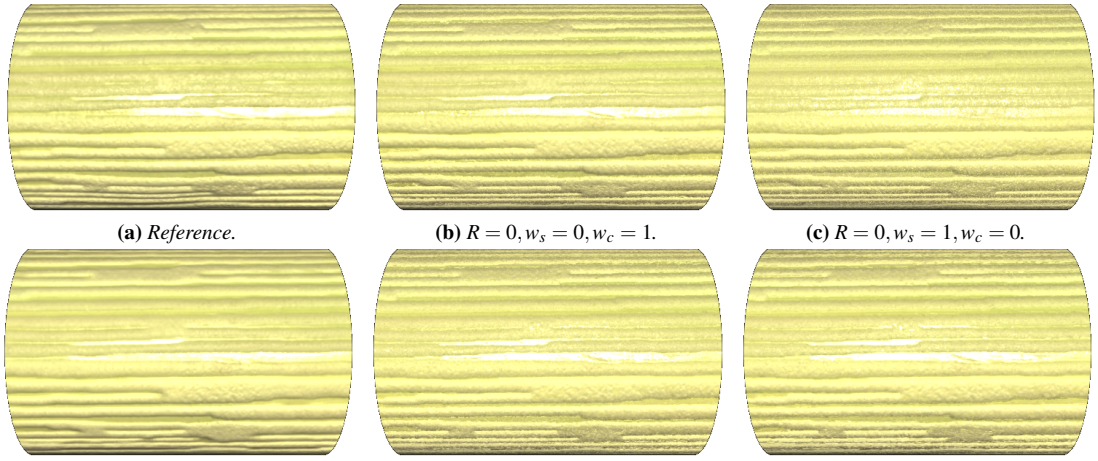
	Leather4		Leather6		Wallpaper1	
Method	Rel. error	Time (sec.)	Rel. error	Time (sec.)	Rel. error	Time (sec.)
$R=0, w_s=0, w_c=1$	51.547%	11.11	41.565%	13.59	33.273%	15.70
$R=0, w_s=1, w_c=0$	63.865%	30.73	48.380%	27.34	40.364%	29.86
$R=1, w_s=1, w_c=10$	63.035%	2193.89	50.059%	2253.04	36.465%	2196.28
$R=2, w_s=1, w_c=10$	61.917%	2264.74	48.443%	2299.53	34.887%	2269.43
Texture Optimization	46.547%	3120.00	35.838%	3180.00	29.159%	3120.00
	Cloth9		Cloth10		Wood4	
Method	Rel. error	Time (sec.)	Rel. error	Time (sec.)	Rel. error	Time (sec.)
$R=0, w_s=0, w_c=1$	35.400%	21.49	36.574%	17.26	12.438%	24.50
$R=0, w_s=1, w_c=0$	77.045%	38.12	42.175%	27.34	17.636%	28.62
$R=1, w_s=1, w_c=10$	55.875%	2326.81	44.723%	2203.31	16.979%	2203.10
$R=2, w_s=1, w_c=10$	48.973%	2521.43	43.742%	2269.75	16.684%	2246.71
Texture Optimization	26.758%	5100.00	31.530%	3240.00	14.336%	3180.00



(a) *Reference.* **(b)** $R=0, w_s=0, w_c=1$. **(c)** $R=0, w_s=1, w_c=0$.

(d) *Texture Optimization.* **(e)** $R=1, w_s=1, w_c=10$. **(f)** $R=2, w_s=1, w_c=10$.

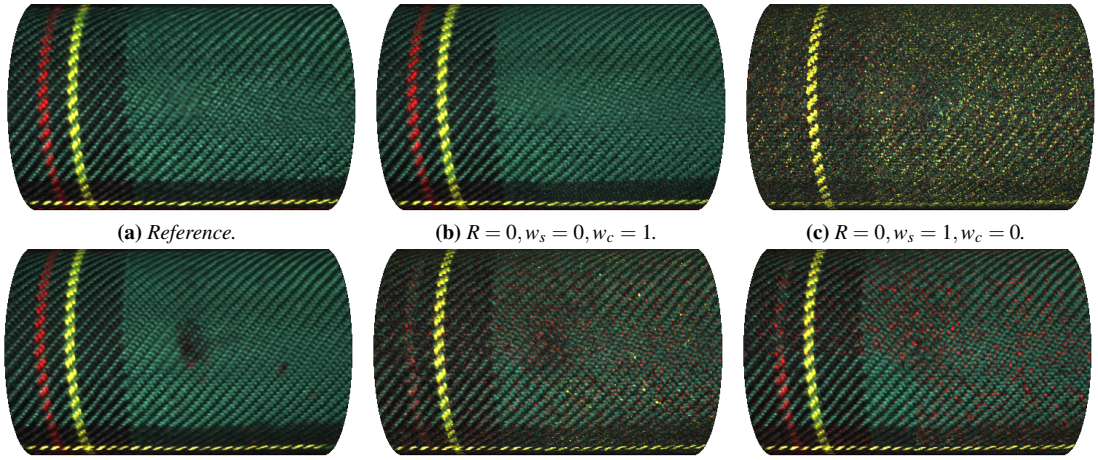
Figure 4: *Renderings of Leather4.*



(a) *Reference.* **(b)** $R=0, w_s=0, w_c=1$. **(c)** $R=0, w_s=1, w_c=0$.

(d) *Texture Optimization.* **(e)** $R=1, w_s=1, w_c=10$. **(f)** $R=2, w_s=1, w_c=10$.

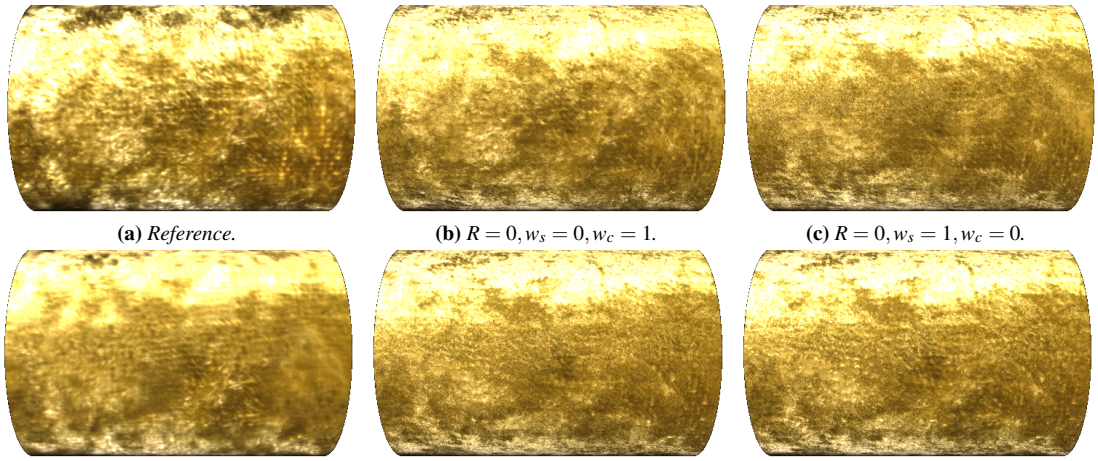
Figure 5: *Renderings of Wallpaper1.*



(a) *Reference.* **(b)** $R = 0, w_s = 0, w_c = 1.$ **(c)** $R = 0, w_s = 1, w_c = 0.$

(d) *Texture Optimization.* **(e)** $R = 1, w_s = 1, w_c = 10.$ **(f)** $R = 2, w_s = 1, w_c = 10.$

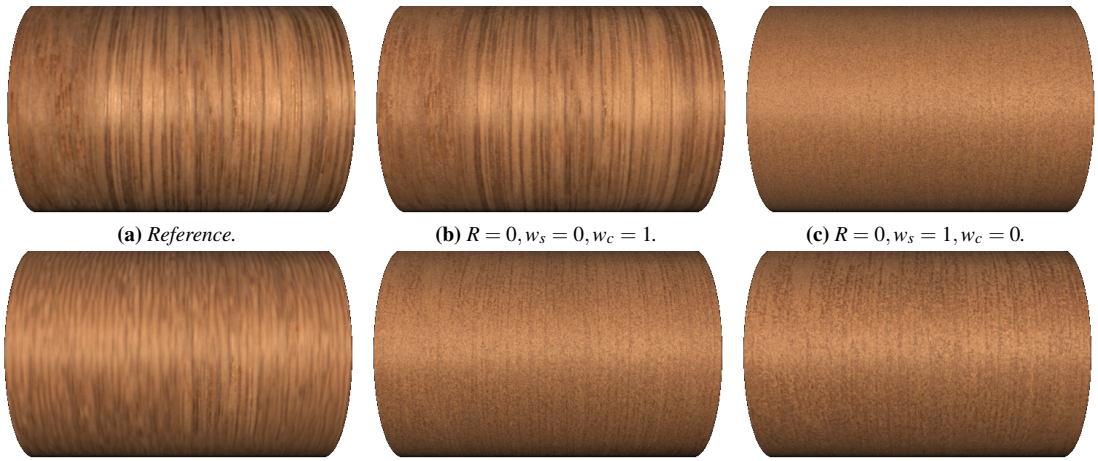
Figure 6: *Renderings of Cloth9.*



(a) *Reference.* **(b)** $R = 0, w_s = 0, w_c = 1.$ **(c)** $R = 0, w_s = 1, w_c = 0.$

(d) *Texture Optimization.* **(e)** $R = 1, w_s = 1, w_c = 10.$ **(f)** $R = 2, w_s = 1, w_c = 10.$

Figure 7: *Renderings of Cloth10.*



(a) *Reference.* **(b)** $R = 0, w_s = 0, w_c = 1.$ **(c)** $R = 0, w_s = 1, w_c = 0.$

(d) *Texture Optimization.* **(e)** $R = 1, w_s = 1, w_c = 10.$ **(f)** $R = 2, w_s = 1, w_c = 10.$

Figure 8: *Renderings of Wood4.*

Acknowledgments

This work was developed in the X-Rite Graduate School on Digital Material Appearance at the University of Bonn.

References

[dBSHK14] DEN BROK D., STEINHAUSEN H. C., HULLIN M., KLEIN R.: Patch-based sparse reconstruction of material BTFs. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2014)* (2014). 5

[DNvGK97] DANA K. J., NAYAR S. K., VAN GINNEKEN B., KOENDERINK J. J.: Reflectance and texture of real-world surfaces authors. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico* (1997), pp. 151–157. 1, 2

[DWT*10] DONG Y., WANG J., TONG X., SNYDER J., LAN Y., BEN-EZRA M., GUO B.: Manifold bootstrapping for SVBRDF capture. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 98. 2

[FHNK05] FURUKAWA R., HARADA M., NAKAMURA Y., KAWASAKI H.: Synthesis of textures with intricate geometries using btf and large number of textured micropolygons. In *Proc. of the 4th International Workshop on Texture Analysis and Synthesis* (2005), pp. 77–82. 2

[FVK14] FILIP J., VÁVRA R., KRUPIČKA M.: Rapid material appearance acquisition using consumer hardware. *Sensors 14*, 10 (2014), 19785–19805. 2

[HF13] HAINDL M., FILIP J.: *Visual Texture: Accurate Material Appearance Measurement, Representation and Modeling*. Springer, 2013. 2

[HH05] HAINDL M., HATKA M.: BTF roller. In *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis* (2005), pp. 89–94. 2

[KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 795–802. 2, 4

[KMBK03] KOUDELKA M. L., MAGDA S., BELHUMEUR P. N., KRIEGMAN D. J.: Acquisition, compression, and synthesis of bidirectional texture functions. In *3rd International Workshop on Texture Analysis and Synthesis (Texture 2003)* (2003), pp. 59–64. 2

[LPF*07] LEUNG M.-K., PANG W.-M., FU C.-W., WONG T.-T., HENG P.-A.: Tileable BTF. *Visualization and Computer Graphics, IEEE Transactions on 13*, 5 (2007), 953–965. 2

[MKU15] MIANDJI E., KRONANDER J., UNGER J.: Compressive image reconstruction in reduced union of subspaces. In *Eurographics 2015* (May 2015). 2

[MMS*05] MÜLLER G., MESETH J., SATTLER M., SARLETTE R., KLEIN R.: Acquisition, synthesis, and rendering of bidirectional texture functions. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 83–109. 2, 3

[Mye10] MYERS R. D.: Colorchecker passport technical review. *Robin Myers Imaging (www.rmimaging.com)* (2010). 3

[PMW*09] PINTUS R., MALZBENDER T., WANG O., BERGMAN R., NACHLIELI H., RUCKENSTEIN G.: Photo repair and 3d structure from flatbed scanners. In *VISAPP International Conference on Computer Vision Theory and Applications* (2009). 2

[PS00] PORTILLA J., SIMONCELLI E. P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision 40*, 1 (2000), 49–70. 5

[PS13] PAN R., SKALA V.: Normal map acquisition of nearly flat objects using a flatbed scanner. In *Virtual Reality and Visualization (ICVRV), 2013 International Conference on* (2013), IEEE, pp. 68–73. 2

[RRK09] RUITERS R., RUMP M., KLEIN R.: Parallelized matrix factorization for fast btf compression. In *Proceedings of the 9th Eurographics conference on Parallel Graphics and Visualization* (2009), Eurographics Association, pp. 25–32. 5

[SdBHK14] STEINHAUSEN H. C., DEN BROK D., HULLIN M. B., KLEIN R.: Acquiring bidirectional texture functions for large-scale material samples. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2014)* (2014). 4, 5

[SMdB*15] STEINHAUSEN H. C., MARTÍN R., DEN BROK D., HULLIN M. B., KLEIN R.: Extrapolation of bidirectional texture functions using texture synthesis guided by photometric normals. In *IS&T/SPIE Electronic Imaging* (2015), International Society for Optics and Photonics, pp. 93980A–93980A. 2, 4, 5

[SSW*14] SCHWARTZ C., SARLETTE R., WEINMANN M., RUMP M., KLEIN R.: Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of Bonn. *Sensors 14*, 5 (Apr 2014). 2

[SSWK13] SCHWARTZ C., SARLETTE R., WEINMANN M., KLEIN R.: DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling* (2013), The Eurographics Association, pp. 25–31. 2

[TZL*02] TONG X., ZHANG J., LIU L., WANG X., GUO B., SHUM H.-Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. In *ACM Transactions on Graphics (TOG)* (2002), vol. 21, ACM, pp. 665–672. 2

[WGK14] WEINMANN M., GALL J., KLEIN R.: Material classification based on training data synthesized using a BTF database. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III* (2014), Springer International Publishing, pp. 156–171. 2, 4

[WHON97] WONG T.-T., HENG P.-A., OR S.-H., NG W.-Y.: Image-based rendering with controllable illumination. In *Proceedings of the Eurographics Workshop on Rendering Techniques* (1997), vol. 97, Citeseer, pp. 13–22. 1

[WLDW11] WU C., LIU Y., DAI Q., WILBURN B.: Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. *Visualization and Computer Graphics, IEEE Transactions on 17*, 8 (2011), 1082–1095. 5

[WLK*09] WEI L.-Y., LEFEBVRE S., KWATRA V., TURK G., ET AL.: State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (2009), pp. 93–117. 2

[WLL*09] WEYRICH T., LAWRENCE J., LENSCH H., RUSINKIEWICZ S., ZICKLER T.: Principles of appearance acquisition and representation. *Foundations and Trends textregistered in Computer Graphics and Vision 4*, 2 (2009), 75–191. 2

[Woo80] WOODHAM R. J.: Photometric method for determining surface orientation from multiple images. *Optical engineering 19*, 1 (1980), 191139–191139. 2

[ZDW*05] ZHOU K., DU P., WANG L., MATSUSHITA Y., SHI J., GUO B., SHUM H.-Y.: Decorating surfaces with bidirectional texture functions. *Visualization and Computer Graphics, IEEE Transactions on 11*, 5 (2005), 519–528. 2