Heinz C. Steinhausen, Rodrigo Martín, Dennis den Brok,
Matthias B. Hullin and Reinhard Klein

"Extrapolation of Bidirectional Texture Functions using Texture Synthesis guided by Photometric Normals"

Proceedings of SPIE Volume 9398
Measuring, Modeling, and Reproducing Material Appearance 2015

Editor(s): Maria V. Ortiz Segovia; Philipp Urban; Francisco H. Imai

Volume Number: 9398

Article Number: 93980A

DOI:

http://dx.doi.org/10.1117/12.2075717

Extrapolation of Bidirectional Texture Functions using Texture Synthesis guided by Photometric Normals

Heinz C. Steinhausen, Rodrigo Martín, Dennis den Brok,
Matthias B. Hullin and Reinhard Klein

University of Bonn, Institute of Computer Science II,
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany

ABSTRACT

Numerous applications in computer graphics and beyond benefit from accurate models for the visual appearance of real-world materials. Data-driven models like photographically acquired bidirectional texture functions (BTFs) suffer from limited sample sizes enforced by the common assumption of far-field illumination. Several materials like leather, structured wallpapers or wood contain structural elements on scales not captured by typical BTF measurements. We propose a method extending recent research by Steinhausen et al. to extrapolate BTFs for large-scale material samples from a measured and compressed BTF for a small fraction of the material sample, guided by a set of constraints. We propose combining color constraints with surface descriptors similar to normal maps as part of the constraints guiding the extrapolation process. This helps narrowing down the search space for suitable ABRDFs per texel to a large extent. To acquire surface descriptors for nearly flat materials, we build upon the idea of photometrically estimating normals. Inspired by recent work by Pan and Skala, we obtain images of the sample in four different rotations with an off-the-shelf flatbed scanner and derive surface curvature information from these. Furthermore, we simplify the extrapolation process by using a pixel-based texture synthesis scheme, reaching computational efficiency similar to texture optimization.

Keywords: Reflectance, Bidirectional Texture Function, Texture Synthesis, Photometric Stereo

1. INTRODUCTION

Numerous applications in computer graphics and beyond benefit from accurate models for the visual appearance of real-world materials. Their common representation as bidirectional texture functions (BTFs) subsumes variations in geometry and non-local lighting effects into an "apparent" bidirectional reflectance distribution function (ABRDF) for each position on the surface. Unlike BRDFs, ABRDFs do not adhere to Helmholtz reciprocity and energy conservation and have too many degrees of freedom to be reasonably represented by parametric models.

When it comes to measuring BTFs, a discrete sampling of the function is typically obtained using image-based setups like camera domes, gonioreflectometers or cameras attached to kaleidoscopic arrangements of mirrors. One conceptual drawback of such data-driven models is the limited sample size. To justify the common assumption of far-field illumination, one has to maintain a certain ratio between the distance of the sample to the cameras and lights on one side and the sample size on the other side. Most acquisition setups compared by Schwartz et al.[1] thus support sample sizes not larger than 10 cm by 10 cm. Obviously, materials like leather, structured wallpapers or wood contain structural elements on several scales. While BTFs obtained by the aforementioned methods are able to capture small- and medium-scale structure, their measurements naturally lack information on the unseen large-scale structure.

Direct acquisition of a BTF for the larger sample, e.g. by sequentially treating different regions of the material, is prohibitive with regards to acquisition and postprocessing time as well as memory requirements. We therefore

Further author information: (Send correspondence to H.C.S.) WWW: http://cg.cs.uni-bonn.de
H.C.S.: E-mail: hc.steinhauen@uni-bonn.de, Telephone: +49 (0) 228 73-54187
R.M.: E-mail: rodrigo@cs.uni-bonn.de, Telephone: +49 (0) 228 73-54187
D.d.B.: E-mail: denbrok@cs.uni-bonn.de, Telephone: +49 (0) 228 73-54185
M.H.: E-mail: hullin@cs.uni-bonn.de, Telephone: +49 (0) 228 73-54169
R.K.: E-mail: rk@cs.uni-bonn.de, Telephone: +49 (0) 228 73-4201

propose a method that extends recent research presented by Steinhausen et al.[2] to extrapolate BTFs for large-scale material samples. The input is a compressed representation of a fully measured BTF for a small region of the material sample, complemented by a set of large-scale photographs that serve as additional constraints. We extend this work in several manners:

While the method by Steinhausen et al. relies on rectified and resampled photographs of the full sample, we propose using surface descriptors similar to normal maps as part of the constraints guiding the extrapolation process. Combined with further imagery and neighborhood information, these are used as distribution maps narrowing down the search space for suitable ABRDFs per texel to a large extent. In order to acquire surface descriptors for nearly flat materials of size in the range of 20 cm by 20 cm, we adapt the main idea of works like a recent paper by Pan and Skala[3] that makes use of an off-the-shelf flatbed scanner to scan the full sample in four different rotations. As flatbed scanners are available in a variety of sizes, this allows us to expand the range of covered scales at a moderate cost.

As another use case, we propose to utilize our method to obtain the same amount of detail with reduced measurement effort. This is achieved by reducing the size of the measured sample, enabling the acquisition of reflectance data for multiple material samples at once. The missing parts can then be reconstructed using our method.

We demonstrate that a constraint set consisting of only four color scans is sufficient to produce a visually appealing result. By evaluating the running times of our pixel-based texture synthesis scheme, we also indicate the improvement in computational efficiency compared to Steinhausen's original method based on texture optimization. In summary, the main contributions of this paper are

1. the utilization of a normals-like surface descriptor for BTF extrapolation,
2. an evaluation of the applicability of a simple pixel-based synthesis scheme for this task.

2. RELATED WORK

In this section, we give a short overview on previous research on BTFs in general (Sec. 2.1Bidirectional Texture Functionssubsection.2.1), followed by selected papers we are relying on in the fields of texture synthesis (Sec. 2.2Texture Synthesis and Completion of Sparse Datasetssubsection.2.2) and photometric stereo (Sec. 2.3Photometric Stereosubsection.2.3). As this paper is a direct successor to recent work by Steinhausen et al.,[2] we are basically building upon the same predecessors regarding representation and synthesis of reflectance datasets.

2.1 Bidirectional Texture Functions

A *bidirectional texture function (BTF)* can be defined as a six-dimensional function $\mathcal{B}(x, y, \theta_i, \phi_i, \theta_v, \phi_v)$, describing the reflectance of a surface depending on surface position (x, y), direction of incoming light θ_i, ϕ_i and viewing direction θ_v, ϕ_v. A seventh dimension is added when regarding color or, more general, wavelength. BTFs were first introduced by Dana et al.[4] They are related to *bidirectional reflectance distribution functions (BRDFs)* and their extension to the spatial domain, *spatially varying BRDFs (SVBRDFs)*, but account for non-local effects not captured with these representations like interreflections, self-shadowing and sub-surface scattering. To achieve this, BTFs discard properties like energy conservation and Helmholtz reciprocity, trading in physical accuracy for visual plausibility. Thus, the per-texel information encoded in a BTF is no longer a BRFD, but a so-called *apparent BRDF (ABRDF)*, a term that was coined by Wong et al.[5] An extensive taxonomy of reflectance models is given e.g. in the 2013 textbook by Haindl and Filip.[6]

Our work is based on a discrete approximation of BTFs captured using photographic devices like the camera domes proposed by Müller et al.[7] and Schwartz et al.[8] Thus, one can intuitively interpret a BTF as a "stack of textures", where each texel does not longer contain only one color value, but one for each combination of lighting and viewing direction. Typical sample sizes for these devices are in the magnitude of 10 cm × 10 cm while maintaining an average angular sampling of $9.4° \pm 1°$ for DOME I and $7.6° \pm 2.6°$ for DOME II. Overviews on methods for material appearance acquisition can be found in the textbook by Haindl and Filip[6] or, with more focus on devices for BTF capture, in the survey by Schwartz et al.[1]

As the captured datasets can reach the magnitude of terabytes, BTFs are most often used in the form of compressed representations. The BTF datasets used for our work are compressed using the *full matrix factorization (FMF)* scheme by Koudelka et al.[9] which is is based on a truncated *singular value decomposition (SVD)*. Here, a BTF is represented as a matrix A where each row represents one specific combination of incoming and outgoing light directions and color, while each column contains one ABRDF for a specific surface position. During compression, A is decomposed into three matrices that, when multiplied, approximate the original matrix:

$$A \approx \tilde{A} = U \cdot \Sigma \cdot V^T, \tag{1}$$

see also Figure 1Schema of SVD-based compression of a measured BTF, stored in matrix A. The columns of A contain ABRDFs, one for each surface coordinate (x, y), while the rows contain textures, one for each combination of lighting direction ω_l, viewing direction ω_v and bandwidth or color channel λ. A is decomposed into a matrix U of eigen-ABRDFs, a diagonal matrix Σ containing singular values and a matrix V of eigentextures. (Image taken from Steinhausen et al.[2])figure.caption.2. Being truncated means that only a limited number k of eigenvalues and thus of eigen-ABRDFs and eigentextures are kept. The choice of k directly influences both memory consumption and visual fidelity of the compressed BTF. The k columns of U are sometimes called the *eigen-ABRDFs* of A, and the k columns of V are the *eigentextures*. In our BTF files, Σ and V are stored in multiplied form as $V\Sigma$.

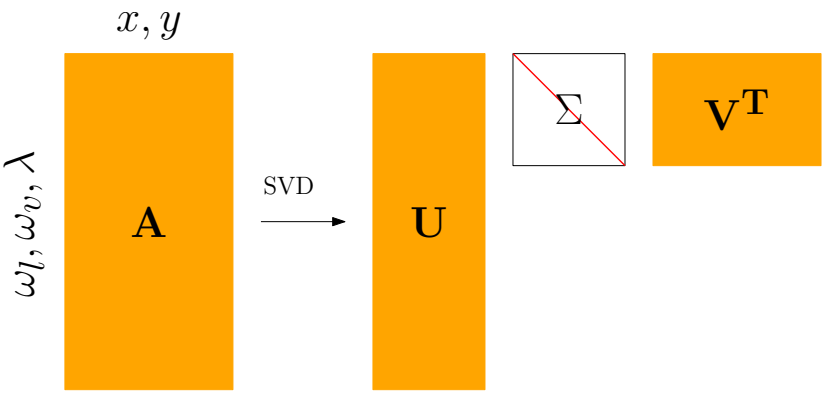


Figure 1: Schema of SVD-based compression of a measured BTF, stored in matrix A. The columns of A contain ABRDFs, one for each surface coordinate (x, y), while the rows contain textures, one for each combination of lighting direction ω_l, viewing direction ω_v and bandwidth or color channel λ. A is decomposed into a matrix U of eigen-ABRDFs, a diagonal matrix Σ containing singular values and a matrix V of eigentextures. (Image taken from Steinhausen et al.[2])

For our experiments, we are using a selection of the material samples contained in the database published by Weinmann et al.[10]

2.2 Texture Synthesis and Completion of Sparse Datasets

Example-based texture synthesis is the process of generating a new image which is, with regard to certain properties, similar to a given sample texture image. An overview of methods available by 2009 was given by Wei et al.[11] Several works have transferred this idea to bidirectional texture functions. While the method proposed by Tong et al.[12] aims at pixel-wise synthesis of a new BTF directly onto a surface, Zhou et al.[13] rely on image quilting. A special use case is the synthesis of realistic textures with complex geometry like fur as tackled by Furukawa et al.[14] Methods based on tilings were proposed by Haindl and Hatka[15] as well as by Leung et al.[16] Ruiters et al.[17] transfer the idea of texture interpolation[18] to BTFs.

These methods, especially those based on tilings, do not easily support the goal of faithfully expanding a measured BTF to one for a larger sample. For SVBRDFs, Dong et al.[19] devised an approach similar to ours in that regard that they also acquire data in two phases. Representative BRDF measurements for manually selected surface positions are distributed using a set of *key measurements* which provide a dense sampling of the surface, but are sparse in the angular domain. Reconstruction is then performed by fitting a manifold of

analytical BRDFs to the representative vectors, controlled by the key measurements. Due to the presence of non-local effects in ABRDFs, this method cannot be extended to BTF in a straightforward way.

Recent work by Filip et al.[20] proposes a new device for sparse reflectance measurement, together with a method to reconstruct an approximate BTF from such data. The authors emphasize the portability and scalability of the device which mainly consists of a gantry with a consumer camera and two LED lights. In its current form, samples of about 30 cm × 30 cm can be measured. For larger samples, they propose either the construction of a larger version or successive measurements of small regions of the sample. Acquisition times of only a few minutes should still enable a fast measurement, but at the price of new problems like proper, seamless alignment of the images. Thus, a method like ours where only a small sample is measured, accompanied by a reconstruction method using only a few images of the full sample, can still be regarded useful. One advantage of their method could be the larger sample size compared to our approach, but, on the other hand, the resolution achievable using a flatbed scanner is higher than with a consumer camera. Also, our method starts with fully measured BTF which should help reconstructing reflectance more accurately. Additionally, the BTFs generated using their approach are only of low dynamic range.

The present paper is a direct successor to the method presented by Steinhausen et al.[2] where *Texture Optimization* as proposed by Kwatra et al.[21] was utilized for the task of BTF extrapolation. We extend this research in several aspects: First, we incorporate a surface descriptor similar to normal maps into the set of constraints guiding the synthesis process. Furthermore, we evaluate the applicability of a simpler pixel-based synthesis scheme. Finally, reconstruction takes place on new data instead only of ground-truth data where all inputs are extracted from an already existing BTF. Our synthesis method is similar to methods like the one presented by Ashikhmin[22] in its pixel-wise manner and in the presence of a guiding constraint, but we refrain from neighborhood matching between the input and already synthesized pixels.

2.3 Photometric Stereo

In 1980, Woodham proposed the method of *Photometric Stereo*[23] which facilitates estimation of surface albedo and normals from only three images. For this purpose, images of an object are taken from a fixed viewpoint, but under varying illumination directions. Light sources are assumed to be point lights in a distance to the object placed at well-known angles. This motivates applicability to BTF images acquired by a camera dome where the LED or flash light sources, mounted at known positions on the hemispherical gantry, can be regarded as distant point lights.

The idea of photometric stereo has also been applied to images acquired using flatbed scanners. Pintus et al.[24] rely on scanners with two separate light bulbs to obtain four images with only two rotations of the material sample, thus reducing the chance of introducing misalignment artifacts. Pan and Skala[3] use a regular flatbed scanner with only one light bar to acquire four images of a sample, manually rotating the sample by approximately 90° between two scans. After image alignment, a set of three linear equations is derived from these images, defining x, y and z-components of the surface normal for each pixel. For accurate positioning of the sample on the scanner platen, as well as to simplify registration, they fix the sample into a clamp with checkerboard markers. To circumvent the necessity to construct a dedicated sample holder, we chose to align the scanned images manually using image manipulation tools. As another simplification, we do not estimate full normals, but employ a pair of difference images as a descriptor for surface curvature.

3. METHOD OVERVIEW

Figure 2Workflow of our BTF extrapolation method.figure.caption.3 depicts the workflow of our BTF extrapolation method. For a material sample under consideration, two kinds of data are generated: A BTF for a small area of the sample, and a set of synthesis constraints for the full-sized sample.

The full sample is scanned four times while being rotated about 90° clockwise between two successive scans, leading to a set of four images. These images are used both for estimating surface descriptors and directly as color constraints, guiding the extrapolation process.

BTF acquisition for a smaller piece of the material sample is performed using the camera dome available in our department. It is important to choose the region to be captured carefully such that the BTF contains all

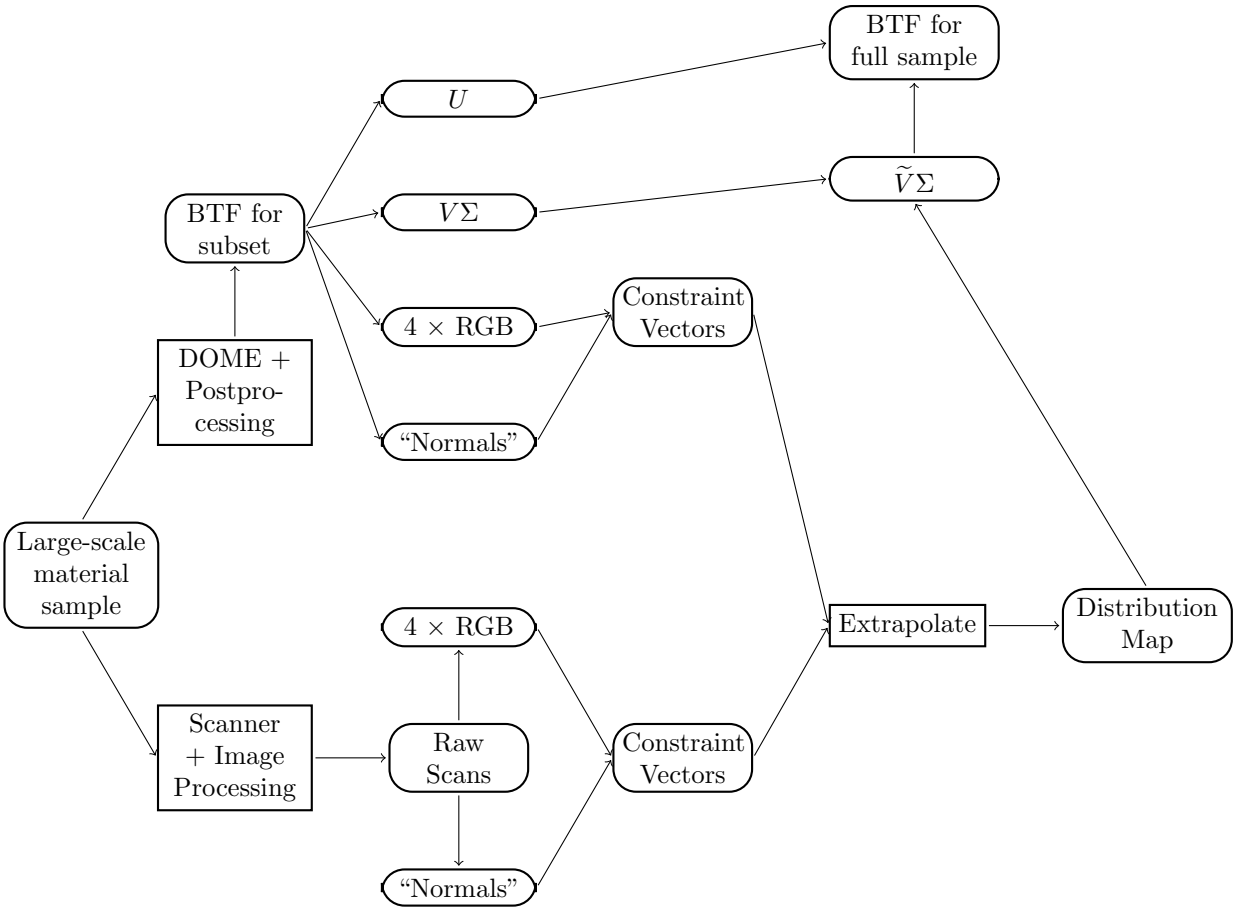


Figure 2: Workflow of our BTF extrapolation method.

variations in reflectance and structure which are desired in the extrapolation result. A compressed representation of the BTF dataset serves as input sample in the actual extrapolation step. A texture synthesis algorithm is applied, guided by the surface descriptors and color constraints.

In the following sections, the details of our method are explained. At first, the input image acquisition and constraint vector formation is described in Sec. 4Input Preparationsection.4, while Sec. 5EXTRAPOLATION METHODsection.5 is concerned with the actual extrapolation algorithm.

4. INPUT PREPARATION

4.1 BTF Acquisition

We rely on BTF measurements contained in the database published by Weinmann et al.[10] For each material, they took 22 801 images are taken from 151 camera positions under 151 different lighting directions each, using the DOME I acquisition device. For material patches of about 5 cm × 5 cm, the resulting images have a spatial extent of 512 × 512 texels. To reduce storage requirements, we extracted from these images a region spanning 128 × 128 pixels. By visual inspection, we selected these regions to contain all desired patterns and reflectance effects. One exception is the material sample "Cloth09" from which we chose an area of 256 × 256 pixels to account for all color variations.

The measured data set is arranged in a matrix $\mathcal{S}$ which is compressed into a factorized representation $\mathcal{S}' = U \cdot \Sigma \cdot V^T$ using a truncated singular value decomposition. ΣV^T, or rather $V\Sigma$, is later used as an input sample for the texture synthesis algorithm to extrapolate a larger set of eigentextures from it.

In the original measurements, four material samples were arranged together on the sample holder, saving about 75% acquisition time compared to measuring each material in a separate pass. With this work, we motivate BTF acquisition for even more material samples in one pass by arranging smaller pieces on the sample holder and reconstructing the full data set using our extrapolation method. Regarding BTF patches of size 128×128 texels, this would correspond to measuring up to 64 material patches of size $1.25\,\mathrm{cm} \times 1.25\,\mathrm{cm}$ in just one pass. A further reduction of acquisition effort could be reached by applying the method by den Brok et al.[25] who propose an angularly sparse measurement, followed by a sparse reconstruction step.

4.2 Extrapolation Constraints

To find a suitable input ABRDF for each output position, a constraint vector is used both on side of the BTF and on the output side. For each texel, a surface descriptor and some color information are combined to form a constraint vector upon which the synthesis algorithm makes its decisions. For this aim, the components of the surface descriptor and all channels of the color images are concatenated into one multi-channel image. We will now discuss the generation of image constraints and surface descriptors in more detail.

4.2.1 Color Images – Scanner

The images that form the extrapolation constraints and from which the surface descriptor is computed are acquired using an off-the-shelf flatbed scanner. One great advantage of this approach is the higher resolution of the scanner optics compared to the DOME images, such that the level of detail of the BTF sample is expected to be reached, if not exceeded, by our extrapolation method.

We used an Epson Perfection V550 Photo with an optical resolution of 6400 dpi for scanning the material samples. In the controlling software, we set resolution to 1200 dpi and deactivated any automatic color correction facilities. For a faithful color reproduction, we chose RGB color mode with 16 bit color per channel. We scanned each material sample four times, rotating the sample by 90° clockwise between two scans which resulted in a counterclockwise rotation of the images. The scanned images were manually rotated, aligned and cropped in order to represent identical parts of the material in identical orientations, providing us with an image set

$$\mathcal{I}^{S_{\mathrm{orig}}} = \left\{ I_{0^\circ}^{S_{\mathrm{orig}}}, I_{90^\circ}^{S_{\mathrm{orig}}}, I_{180^\circ}^{S_{\mathrm{orig}}}, I_{270^\circ}^{S_{\mathrm{orig}}} \right\}.$$

For color correction, we scanned the six grey fields in the *Classic Target* of an X-Rite ColorChecker Passport. We then searched for correction values for the red, green and blue channel such that the color values for the grey patches are most similar to the values measured by Robin Myers[26] for the percentual reflectance, see Tab. 1aReflectance values assumed for the ColorChecker grey patches.table.caption.4. By minimizing the ℓ_2-difference between corrected per-channel values and those measured by Myers, we determined the correction values given in Tab. 1bCorrection values applied to color channels of scanned images.table.caption.4. By applying these correction values to $\mathcal{I}^{S_{\mathrm{orig}}}$, we obtain

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

with

$$I_r^{S_{\mathrm{cc}}}(x, y, c) = \left(\left(I_r^{S_{\mathrm{orig}}}(x, y, c) - o(c) \right) \cdot m(c) \right)^{\gamma(c)}, \tag{2}$$

where $r \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$, (x, y) pixel positions in $I_r^{S_{\mathrm{orig}}}$, c an image channel in $I_r^{S_{\mathrm{orig}}}$.

The images in $\mathcal{I}^{S_{\mathrm{cc}}}$ are used both directly as color constraints and for determining the surface descriptors, see Sec. 4.2.3Surface Descriptorssubsubsection.4.2.3.

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

(a) Reflectance values assumed for the ColorChecker grey patches.

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

4.2.2 Color images – BTF

On the BTF side, corresponding constraints to those prepared for the extrapolation objective are required. For this purpose, we extract four images from a fixed viewing position from the BTF, choosing the top camera of the DOME and lights near positions $(-1, 0, z), (0, -1, z), (1, 0, z)$ and $(0, 1, z)$ in normalized Cartesian coordinates. By visual inspection, we chose a value of $z = 0.9272$, corresponding to an elevation angle of $\theta_i = 22°$. Using these positions, we decompressed a set of four RGB images

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

from the compressed BTF representation $\mathcal{S}'$.

Due to the different color characteristics of the DOME images compared to the scans, some alignment of color values between the two data sets $\mathcal{I}^{S_{\text{cc}}}$ and $\mathcal{I}^B$ has to be performed. We chose a simple linear scaling to align the color histograms of the image sets. For each color channel c, we determined a scaling factor

$$f_c = \frac{A(\mathcal{I}^B, c)}{A(\mathcal{I}^{S_{\text{cc}}}, c)}, \tag{3}$$

where

$$A(\mathcal{I}, c) = \frac{1}{4 \cdot w \cdot h} \sum_{I \in \mathcal{I}} \sum_{(x,y)} I(x, y, c) \tag{4}$$

denotes the arithmetic mean over all pixels in all images I contained in the set $\mathcal{I}$, with (x, y) being a pixel position in I and w, h width and height of I, $I \in \mathcal{I}$. These factors are then multiplied to the channel of the images in $\mathcal{I}^{S_{\text{cc}}}$, resulting in the final image set $\mathcal{I}^S$ with

$$I^S_r(x, y, c) = f_c \cdot I^{S_{\text{cc}}}_r(x, y, c), \tag{5}$$

r, x, y, c as above.

4.2.3 Surface Descriptors

Inspired by the idea of photometric stereo, we include into the constraint set photometrically acquired information on surface geometry. As a simplification compared to normal maps, we employ difference images to obtain a texture invariant descriptor for surface curvature of nearly flat objects.

We first transform the images of a set $\mathcal{I} = \{I_{0°}, I_{90°}, I_{180°}, I_{270°}\}$ to a set $\widetilde{\mathcal{I}} = \left\{ \widetilde{I}_{0°}, \widetilde{I}_{90°}, \widetilde{I}_{180°}, \widetilde{I}_{270°} \right\}$ of luminance images by converting them to CIELAB color space and keeping only the L-channel of each image.

Surface descriptors are then calculated as differences of luminance values:

$$\widetilde{n}_1 = \widetilde{I}_{0°} - \widetilde{I}_{180°}, \tag{6}$$

$$\widetilde{n}_2 = \widetilde{I}_{270°} - \widetilde{I}_{90°}. \tag{7}$$

As luminance values for images from different sources like a camera and a scanner may not be uniform, a scaling of these difference images takes place to obtain the final values for the surface descriptors:

$$n_i = \frac{\widetilde{n}_i - A(\widetilde{n}_i)}{\sigma(\widetilde{n}_i - A(\widetilde{n}_i))}, \qquad i = 1, 2, \tag{8}$$

where $M - s$ for a matrix M and a scalar s means subtracting s from all elements of M, and A, σ denote arithmetic mean and standard deviation:

$$A(\widetilde{n}_i) = \frac{1}{w \cdot h} \sum_{(x,y)} \widetilde{n}_i(x,y), \tag{9}$$

$$\sigma(\widetilde{n}_i) = \sqrt{\frac{1}{w \cdot h - 1} \sum_{(x,y)} \left(\widetilde{n}_i(x,y) - A(\widetilde{n}_i)\right)^2}, \tag{10}$$

with $i \in \{1,2\}$, (x,y) being a pixel position in $\widetilde{n}_i$ and w, h width and height of $\widetilde{n}_i$.

Applying equations 5Surface Descriptorsequation.4.5 to 7Surface Descriptorsequation.4.7 to $\mathcal{I}^B$ and $\mathcal{I}^S$ provides the surface descriptors (n_1^B, n_2^B) and (n_1^S, n_2^S) for the sample BTF and the scanned images.

4.3 Constraint Vector Assembly

After obtaining the color images and surface descriptors from both the measured BTF and the scanned material samples, we concatenate them to create the *constraint vectors* for texture synthesis. Weighting factors w_s and w_c are applied to the vector's components to level between the impact of surface descriptors and color constraints:

$$C_0^B(x,y) = \left[w_s \cdot n_1^B(x,y), w_s \cdot n_2^B(x,y), w_c \cdot I_{0^\circ}^B(x,y,:), w_c \cdot I_{90^\circ}^B(x,y,:), w_c \cdot I_{180^\circ}^B(x,y,:), w_c \cdot I_{270^\circ}^B(x,y,:)\right], \tag{11}$$

$$C_0^S(x,y) = \left[w_s \cdot n_1^S(x,y), w_s \cdot n_2^S(x,y), w_c \cdot I_{0^\circ}^S(x,y,:), w_c \cdot I_{90^\circ}^S(x,y,:), w_c \cdot I_{180^\circ}^S(x,y,:), w_c \cdot I_{270^\circ}^S(x,y,:)\right], \tag{12}$$

where $I(x,y,:)$ denotes the concatenation of the values of all channels of I in position (x,y).

We also evaluate the usefulness of neighborhood information without complicating the actual extrapolation process. For this purpose, we incorporate information on neighboring texels directly into the constraint vectors. For a neighborhood radius of R, and C_0^X a constraint vector as defined in Eq. 10Constraint Vector Assemblyequation.4.10 or 11Constraint Vector Assemblyequation.4.11 ($X = B$ or $X = S$), let $C_R^X(x,y)$ be the *extended constraint vector* for position (x,y), given by concatenating the constraint vectors C_0^X for (x,y)'s neighborhood:

$$\begin{aligned} C_R^X(x,y) = [&C_0^X(x-R,y-R), \ldots, C_0^X(x,y-R), \ldots, C_0^X(x+R,y-R), \\ &\vdots \\ &C_0^X(x-R,y), \ldots, C_0^X(x,y), \ldots, C_0^X(x+R,y), \\ &\vdots \\ &C_0^X(x-R,y+R), \ldots, C_0^X(x,y+R), \ldots, C_0^X(x+R,y+R)] \end{aligned} \tag{13}$$

As BTFs are stored as matrices, spatial indices (x,y) are linearized to row indices i in $V\Sigma$ or $\widetilde{V}\Sigma$, respectively:

$$i(x,y) = (y-1) \cdot w + x, \tag{14}$$

$$(x(i), y(i)) = \left(((i-1) \bmod w) + 1, ((i-1) \operatorname{div} w) + 1 \right), \tag{15}$$

with w, h width and height of $V\Sigma$ or $\widetilde{V}\Sigma$, depending on the context, mod modulus and div integer division. In the following, we will address constraint vectors for matrix rows in $V\Sigma$ or $\widetilde{V}\Sigma$ via $C_R^X(i) = C_R^X(x(i), y(i))$.

5. EXTRAPOLATION METHOD

The task of the extrapolation algorithm is to generate a new BTF $\widetilde{\mathcal{S}} = U\Sigma\widetilde{V}^T$ from the original BTF $\mathcal{S}' = U\Sigma V^T$. To be more specific, we rely on the presence of all important effects of reflectance in the original ABRDF set U and synthesize only a new set of eigentextures $\widetilde{V}\Sigma$. As each row i in $V\Sigma$ represents a position (x, y) in the BTF's image set, we thus need to find, for each row $\widetilde{i}$ in $\widetilde{V}\Sigma$ a suitable row i in $V\Sigma$.

In contrast to Steinhausen et al.,[2] we do not use texture optimization which performs a global optimization of local neighborhood similarities. Instead, we perform for each row i in $\widetilde{V}\Sigma$ a nearest neighbor search in the rows of $V\Sigma$, using the ℓ_2 difference of constraint vectors as similarity measure. Actual search is thus not performed on the $V\Sigma$-sets, but on the extended constraint vectors described in Sec. 4.3Constraint Vector Assemblysubsection.4.3:

$$i(\widetilde{i}) = \arg\min_{i'} \left\| C^B(i') - C^S(\widetilde{i}) \right\|. \tag{16}$$

In our implementation, this search is performed using the MATLAB function `knnsearch`, using a kd-tree as acceleration structure, and euclidean distance. After a corresponding row $i(\widetilde{i})$ has been found for all rows $\widetilde{i}$ in $\widetilde{V}\Sigma$, $\widetilde{V}\Sigma$ is assembled by just copying row $i(\widetilde{i})$ of $V\Sigma$ to row $\widetilde{i}$ in $\widetilde{V}\Sigma$. Finally, $\widetilde{V}\Sigma$ is stored to file together with U, forming the BTF $\widetilde{S}$ for the full material sample.

6. RESULTS

We evaluated our extrapolation method on the following materials from the BTF database:[10] Cloth09, Cloth10, Leather06, Wallpaper01, Wood04. For these material samples, we obtained compressed BTFs of size 128×128 texels, extracted from the full datasets spanning 512×512 texels. One exception is the material sample *Cloth09* for which a sample BTF of size 256×256 texels was chosen in order to capture all relevant color variations.

We ran our MATLAB implementation of the extrapolation pipeline described in the previous sections with different parameter sets. Figures 3Renderings of Leather06.figure.caption.5 to 7Renderings of Cloth10.figure.caption.9 show extrapolation results and sample BTFs as cut-outs of cylindrical renderings. As the reference BTF and the extrapolation results do not show the same area of the material sample and might have slightly different scales, direct comparison is not possible. Also for this reason, no numerical error values can be given, such that all quality evaluation takes place visually.

Computations took place using a desktop computer with Intel Core i7-2600K CPU at 3.4 GHz, equipped with 16 GB of RAM, using an un-parallelized MATLAB implementation operating in 64 bit. The running times for computing the surface structure constraints were in the range of a few seconds. Extrapolation took 30 sec for $R = 0$ up to about 150 sec for $R = 1$ in case of the 128×128 texel BTFs. For extrapolating Cloth09 from 256×256 texels to 512×512 texels, running times rose to nearly 2500 sec for $R = 0$ and about 3000 sec for $R = 1$, of which the biggest part was spent on the constraint vector generation for the scanned images. Compared to the runnnig time of 1h 20 min for the method by Steinhausen et al.,[2] this is a slight improvement.

6.1 Visual Quality

For the leather (Fig. 3Renderings of Leather06.figure.caption.5), one can see that both constraint types, color and surface structure, help preserving the distinctive look of the material. Also, a neighborhood radius of 1 helps removing noise.

Figure 4Renderings of Wallpaper01.figure.caption.6 illustrates a similar situation for the wallpaper. Note how a combination of color and surface structure constraints helps levelling between detail and smoothness, and how the addition of neighborhoods helps preserving the specular appearance in the flat lower parts of the sample.

The finished wood (Fig. 6Renderings of Wood04.figure.caption.8) makes it obvious that for materials with a rather flat surface, there is little to no benefit in the use of surface structure constraints.

Finally, we processed two pieces of cloth with quite different characteristics. For Cloth09, we show a reconstruction of a 512×512 texel region from a 256×256 texel BTF in Fig. 5Renderings of Cloth09.figure.caption.7. While some structure is captured by the surface structure constraints, it is not able to correctly extrapolate color. On the other hand, the result using color as the only constraint looks already rather convincing.

(a) Reconstruction ($R = 0, w_s = 0, w_c = 1$).
(b) Reconstruction ($R = 0, w_s = 1, w_c = 0$).
(c) Reconstruction ($R = 0, w_s = 1, w_c = 1$).
(d) Reconstruction ($R = 0, w_s = 1, w_c = 10$).
(e) Reconstruction ($R = 0, w_s = 1, w_c = 50$).
(f) Reconstruction ($R = 1, w_s = 1, w_c = 10$).
(g) Reference.

Figure 3: Renderings of Leather06.

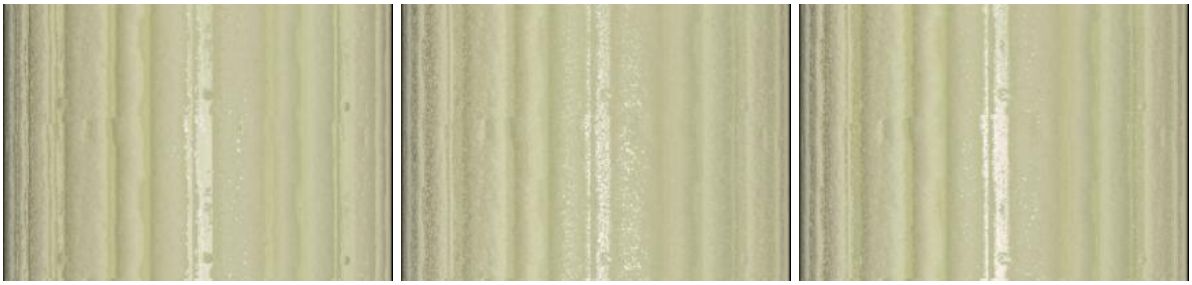
Cloth10, as depicted in Fig. 7Renderings of Cloth10.figure.caption.9, serves as a failure case. It exhibits a chaotic fibre structure and a strong specular behavior which are both not expressed well by the result of our extrapolation method.

7. DISCUSSION

The results presented in this paper demonstrate the usefulness of both color constraints and information on surface geometry. As we have shown, it highly depends on the specific material which constraint has the most influence on the reconstruction quality. Thus, the optimal parameters of operation have to be found either manually or by experience from similar materials. The speed of our algorithm for small extrapolation regions facilitates a nearly interactive use, running a 128×128 texel extrapolation to generate a preview for larger sizes.

As another aspect, we have proven that a simpler method than texture synthesis is also applicable to the problem of BTF extrapolation without sacrificing the chance to achieve visually pleasing results. Also, it is at least competitive to a texture optimization-based approach with regard to speed.

One challenge in our approach was that we wanted to avoid the necessity of a clamp or other form of sample holder with registration marks. Also, as the artifacts scanned by Pan and Skala are not easily deformable, only rigid transformations are needed to be taken into account on registration. In our setting, especially materials



(a) Reconstruction ($R = 0, w_s = 0, w_c = 1$).

(b) Reconstruction ($R = 0, w_s = 1, w_c = 0$).

(c) Reconstruction ($R = 0, w_s = 1, w_c = 1$).

(d) Reconstruction ($R = 0, w_s = 1, w_c = 10$).

(e) Reconstruction ($R = 0, w_s = 1, w_c = 50$).

(f) Reconstruction ($R = 1, w_s = 1, w_c = 10$).

(g) Reference.

Figure 4: Renderings of Wallpaper01.

(a) Reconstruction ($R = 0, w_s = 0, w_c = 1$).

(b) Reconstruction ($R = 0, w_s = 1, w_c = 0$).

(c) Reference.

Figure 5: Renderings of Cloth09.

(a) Reconstruction ($R = 0, w_s = 0, w_c = 1$).

(b) Reconstruction ($R = 0, w_s = 1, w_c = 0$).

(c) Reconstruction ($R = 0, w_s = 1, w_c = 1$).

(d) Reconstruction ($R = 0, w_s = 1, w_c = 10$).

(e) Reconstruction ($R = 0, w_s = 1, w_c = 50$).

(f) Reconstruction ($R = 1, w_s = 1, w_c = 10$).

(g) Reference.

Figure 6: Renderings of Wood04.

(a) Reconstruction ($R = 0, w_s = 0, w_c = 1$).

(b) Reconstruction ($R = 0, w_s = 1, w_c = 0$).

(c) Reference.

Figure 7: Renderings of Cloth10.

like leather or cloth can be locally bent or warped between two successive scans, such that global or even rigid registration methods might deliver unsatisfactory results. Thus, a powerful automatic image alignment method would be of great benefit for our problem setting, also enabling more accurate surface structure information.

Another question is the transferability of the proposed method to images acquired under less controlled conditions, i.e. using a camera on a tripod, together with a hand-held light source. To this end, one could rely on ideas given in the work of Wu et al.[27] which aims at improving geometry by combining multi-view stereo and photometric stereo. This is achieved by alternating between estimation of a spherical harmonic illumination model and surface reconstruction.

Furthermore, a systematic exploration of the influence of different parameter settings to both reconstruction quality and speed would be helpful, as well as an attempt to accelerate constraint vector construction and nearest neighbor search. In addition, results might be improved by using "real" photometrically acquired normals instead of our surface constraints.

Also, as an evaluation of the performance of several of the existing texture synthesis algorithms on this specific problem might be of interest.

ACKNOWLEDGMENTS

This work was developed in the graduate school on digital material appearance funded by X-Rite Inc. We would like to thank Gero Müller, Martin Rump and Roland Ruiters for helpful advice.

REFERENCES

[1] Schwartz, C., Sarlette, R., Weinmann, M., Rump, M., and Klein, R., "Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of bonn," *Sensors* **14** (apr 2014).

[2] Steinhausen, H. C., den Brok, D., Hullin, M. B., and 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).

[3] Pan, R. and Skala, V., "Normal map acquisition of nearly flat objects using a flatbed scanner," in [*Virtual Reality and Visualization (ICVRV), 2013 International Conference on*], 68–73, IEEE (2013).

[4] Dana, K. J., Van Ginneken, B., Nayar, S. K., and Koenderink, J. J., "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics (TOG)* **18**(1), 1–34 (1999).

[5] Wong, T.-T., Heng, P.-A., Or, S.-H., and Ng, W.-Y., "Image-based rendering with controllable illumination," in [*Proceedings of the Eurographics Workshop on Rendering Techniques*], **97**, 13–22, Citeseer (1997).

[6] Haindl, M. and Filip, J., [*Visual Texture: Accurate Material Appearance Measurement, Representation and Modeling*], Springer (2013).

[7] Müller, G., Meseth, J., Sattler, M., Sarlette, R., and Klein, R., "Acquisition, synthesis, and rendering of bidirectional texture functions," in [*Computer Graphics Forum*], **24**, 83–109, Wiley Online Library (2005).

[8] Schwartz, C., Sarlette, R., Weinmann, M., and Klein, R., "DOME II: A parallelized BTF acquisition system," in [*Eurographics Workshop on Material Appearance Modeling*], 25–31, The Eurographics Association (2013).

[9] Koudelka, M. L., Magda, S., Belhumeur, P. N., and Kriegman, D. J., "Acquisition, compression, and synthesis of bidirectional texture functions," in [*3rd International Workshop on Texture Analysis and Synthesis (Texture 2003)*], 59–64 (2003).

[10] Weinmann, M., Gall, J., and 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*], 156–171, Springer International Publishing (2014).

[11] 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*], 93–117 (2009).

[12] Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., and Shum, H.-Y., "Synthesis of bidirectional texture functions on arbitrary surfaces," in [*ACM Transactions on Graphics (TOG)*], **21**(3), 665–672, ACM (2002).

[13] Zhou, K., Du, P., Wang, L., Matsushita, Y., Shi, J., Guo, B., and Shum, H.-Y., "Decorating surfaces with bidirectional texture functions," *Visualization and Computer Graphics, IEEE Transactions on* **11**(5), 519–528 (2005).

[14] Furukawa, R., Harada, M., Nakamura, Y., and 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*], 77–82, Citeseer (2005).

[15] Haindl, M. and Hatka, M., "Btf roller," in [*Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*], 89–94 (2005).

[16] Leung, M.-K., Pang, W.-M., Fu, C.-W., Wong, T.-T., and Heng, P.-A., "Tileable btf," *Visualization and Computer Graphics, IEEE Transactions on* **13**(5), 953–965 (2007).

[17] Ruiters, R., Schwartz, C., and Klein, R., "Example-based interpolation and synthesis of bidirectional texture functions," *Computer Graphics Forum (Proceedings of the Eurographics 2013)* **32**, 361–370 (may 2013).

[18] Ruiters, R., Schnabel, R., and Klein, R., "Patch-based texture interpolation," in [*Computer Graphics Forum*], **29**, 1421–1429, Wiley Online Library (2010).

[19] Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., and Guo, B., "Manifold bootstrapping for svbrdf capture," *ACM Transactions on Graphics (TOG)* **29**(4), 98 (2010).

[20] Filip, J., Vávra, R., and Krupička, M., "Rapid material appearance acquisition using consumer hardware," *Sensors* **14**(10), 19785–19805 (2014).

[21] Kwatra, V., Essa, I., Bobick, A., and Kwatra, N., "Texture optimization for example-based synthesis," in [*ACM Transactions on Graphics (TOG)*], **24**, 795–802, ACM (2005).

[22] Ashikhmin, M., "Synthesizing natural textures," in [*Proceedings of the 2001 symposium on Interactive 3D graphics*], 217–226, ACM (2001).

[23] Woodham, R. J., "Photometric method for determining surface orientation from multiple images," *Optical engineering* **19**(1), 191139–191139 (1980).

[24] Pintus, R., Malzbender, T., Wang, O., Bergman, R., Nachlieli, H., and Ruckenstein, G., "Photo repair and 3d structure from flatbed scanners," in [*VISAPP International Conference on Computer Vision Theory and Applications*], (2009).

[25] den Brok, D., Steinhausen, H. C., Hullin, M., and 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).

[26] Myers, R. D., "Colorchecker passport technical review," *Robin Myers Imaging (www.rmimaging.com)* (2010).

[27] Wu, C., Liu, Y., Dai, Q., and Wilburn, B., "Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination," *Visualization and Computer Graphics, IEEE Transactions on* **17**(8), 1082–1095 (2011).