Acquiring Bidirectional Texture Functions for Large-Scale Material Samples

Heinz Christian Steinhausen Dennis den Brok Matthias B. Hullin Reinhard Klein
Universität Bonn
Institute of Computer Science II
Friedrich-Ebert-Allee 144
53113 Bonn, Germany
{steinhau, denbrok, hullin, rk}@cs.uni-bonn.de

Abstract

Most current acquisition setups for bidirectional texture functions (BTFs) are incapable of capturing large-scale material samples. We propose a method based on controlled texture synthesis to produce BTFs of appealing visual quality for such materials. Our approach uses as input data a complete measurement of a small fraction of the sample, together with few images of the large-scale structure controlling the synthesis process. We evaluate the applicability of our approach by reconstructing sparsified ground truth data and investigate the consequences of choosing different kinds and numbers of constraint images.

Keywords

bidirectional texture functions - texture synthesis - material appearance.

1 INTRODUCTION

To produce realistically looking scenes in computer graphics, suitable material representations beyond two-dimensional textures are needed. Data-driven models like photographically acquired *bidirectional texture functions* (*BTFs*) are capable of capturing a wide range of optical effects. However, due to practical constraints the capture of full BTFs is limited to small sample sizes. The goal of this work is to capture materials with large-scale structure such as wood (see Fig. 1), ornamented cloth, or structured wallpapers, for which tiling or uncontrolled synthesis fails to produce realistic results.

Figure 1: Example of a piece of wood for which a BTF measurement using typical sample sizes, bordered green, would not be able to capture all of its structure.

BTFs are usually acquired in a data-driven fashion by taking and combining series of photographs shot using different lighting and viewing directions. Not only is the capture volume of available setups typically much smaller than a material's characteristic structure, complete measurements are also often prohibitively expensive regarding both the time needed for acquisition and postprocessing (hours to days) and the storage requirements (terabytes). In agreement with literature [Don13], [Hai13], we observe that BTF data is highly redundant. In particular, disjoint regions typically share very similar visual properties.

We therefore propose to use a texture synthesis approach to capture the visual appearance of materials with large-scale structure. Our method performs the following steps:

1. complete measurement of a small representative spatial region,
2. acquisition of few images of the full sample, and
3. synthesis of a complete BTF.

The term "representative" in this context means that the region to be measured is chosen in such a way that it contains all effects regarding reflectance and surface structure which can be observed throughout the complete material sample. The modeling of appearance as BTFs gives rise to large amounts of data that need to be managed efficiently. We therefore propose to perform all steps in a compressed representation, similar to the work on material interpolation by Ruiters *et al.* [Rui13].

The focus of this work lies on surveying the applicability of texture synthesis to the task of completing sparse BTF measurements, widely disregarding the further challenges mentioned above. Evaluation takes place on sparsified ground truth data for which our method is able to produce visually pleasing results. We start by providing an overview of related work in Section 2, followed by a detailed description of our technique in Section 3, including considerations on the right choice of controlling constraints. We then deliver some experimental results in Section 4.

2 RELATED WORK

2.1 Bidirectional Texture Functions

Bidirectional texture functions (BTFs) as introduced by Dana *et al.* [Dan99] are one model to describe the reflection behavior of materials, closely related to *bidirectional reflectance distribution functions (BRDFs)* and their extension to the spatial domain, *spatially varying BRDFs (SVBRDFs)*. For a taxonomy of reflectance models, see *e.g.* the 2013 textbook by Haindl and Filip [Hai13]. While plain BRDFs describe the reflectance of a material depending on in- and outgoing light directions only, the other two also take into account spatial variations in the reflection behaviour of a material. Both models share properties like energy conservation and Helmholtz reciprocity. SVBRDFs may be a sufficient representation for materials without complex surface structure, but they are not able to capture non-local effects such as self-shadowing, interreflections and subsurface scattering.

For modeling materials exhibiting such effects, BTFs are regarded to be more suitable. A BTF can be interpreted as a generalization of SVBRDFs where the per-texel BRDFs are not true BRDFs anymore but *apparent BRDFs (ABRDFs)* which account for the effects ignored by (SV)BRDFs. Formally, a *bidirectional texture function (BTF)* is a six-dimensional function $\mathcal{B}(\mathbf{x}, \theta_i, \phi_i, \theta_v, \phi_v)$, parametrized over surface position $\mathbf{x} = (x, y)$, direction of incoming light θ_i, ϕ_i and viewing direction θ_v, ϕ_v. Commonly, a discrete approximation of this function is captured using photographic devices like the camera domes proposed by Müller *et al.* [Mül05] and Schwartz *et al.* [Sch13]. This leads to a more intuitive interpretation of 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. The devices mentioned above provide a good tradeoff between sample sizes and sampling density in spatial as well as in angular domain and support sample sizes of approximately $10\,\text{cm} \times 10\,\text{cm}$. For an overview on methods for material appearance acquisition, see again the textbook by Haindl and Filip [Hai13] or the recent survey by Schwartz *et al.* [Sch14] focusing on BTFs.

Several methods have been developed to reduce the large amounts of data resulting from a BTF measurement. In our work, we utilize compression based on *singular value decomposition* as it was first used by Suen and Healey [Sue00] for dimensionality analysis on BTFs. Applied to a matrix A, the result is a decomposition into three matrices that, when multiplied, approximate the original matrix:

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

The decomposition of a BTF is sketched in Fig. 2. The number of eigenvalues and corresponding eigenvectors which are kept determines both memory consumption and visual fidelity of the compressed BTF.

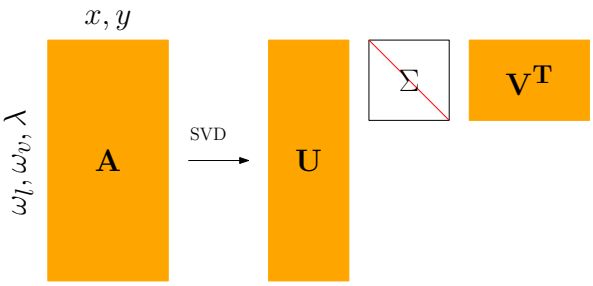


Figure 2: Schema of SVD-based compression of a measured BTF, stored in matrix A. The columns of A contain ABRDFs, one for each surface coordinate (x, y), while the rows contain textures, one for each combination of lighting direction ω_l, viewing direction ω_v and bandwidth or color channel λ. A is decomposed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

2.2 Synthesis and Reconstruction

Texture synthesis has been an active field of research for several years now. An elaborate survey of the example-based texture synthesis methods from the 1990s to 2009 is given by Wei *et al.* [Wei09].

Also, the idea of synthesizing bidirectional texture functions has been around for quite some years now. Tong *et al.* [Ton02] propose a pixel-wise method to synthesize a new BTF directly onto a surface. Another approach, based on image quilting, is given by Zhou *et al.* [Zho05]. Synthesis of realistic textures with complex geometry like fur is the goal of the approach by Furukawa *et al.* [Fur05]. Haindl and Hatka [Hai05] propose a tiling method for BTFs. Leung *et al.* [Leu07] also rely on tiling, generating a set of seamless Wang tiles from a compressed representation of measured BTF data. A recent approach by Ruiters *et al.* [Rui13] transfers the idea of texture interpolation [Rui10] to BTFs, allowing to generate a new texture which is perceived as lying in-between the appearance of two given input examples.

What these methods for BTF synthesis lack is the possibility to generate a BTF for a large material sample

from a measurement of a smaller portion of it. There are two main reasons for choosing *Texture Optimization* as proposed by Kwatra *et al.* [Kwa05] for this task. On the one hand, their algorithm allows for controlled texture synthesis, and on the other hand, it was already successfully utilized for the synthesis of BTFs by Ruiters *et al.* in their interpolation method. The algorithm is regarded lying in-between pixel- and patch-based approaches, as it optimizes the output texture pixelwise, minimizing a global energy function of neighborhood similarities.

An approach for reconstructing spatially varying appearance with reduced measurement effort was proposed by Dong *et al.* [Don10], but only for SVBRDFs. They also acquire data in two phases. First, they take a set of representative BRDF measurements for manually selected surface positions using a one-pixel camera. In a second step, they capture a set of *key measurements*, measuring reflectance densely over the surface, but angularly sparse. The actual reflectance vector for a surface point is then constructed by fitting a manifold of analytical BRDFs to the representative vectors, controlled by the key measurements. It is not obvious how to adopt their method for BTF enlargement, as ABRDFs contain non-local effects not captured by (SV)BRDFs.

3 DESCRIPTION OF METHOD

Our method for generating BTFs for large-scale material samples can be divided into three main parts:

1. Completely measure a BTF $\mathcal{S}$ for a small representative region,
2. acquire a set $\mathcal{C}$ of images of the large-scale structure using as few lighting and viewing directions as possible,
3. and finally use controlled texture synthesis to combine the gathered information resulting in a BTF for the large sample.

In this work, we focus on the third step. We demonstrate how texture synthesis can be applied to the problem of completing sparse measurements to obtain a BTF for the full sample. Synthesis does not take place on the raw data of a measurement, but on the $\Sigma \cdot V^T$-part of its compressed and logarithmic range reduced version (see Eq. 1 in Section 2.1). This approach saves computation time and memory and was already successfully used for BTF synthesis and even interpolation by Ruiters *et al.* [Rui13]. If the region for $\mathcal{S}$ is carefully chosen to contain all aspects of reflectance and surface structure present in the full sample, the ABRDFs stored in U can be expected to contain all information needed for a faithful completion.

We evaluate our method by reconstructing ground truth data from BTF measurements, applying the workflow depicted in Fig. 3a. In the following subsections, we explain our reconstruction method as well as the preparation of input data in more detail.

3.1 Reconstruction Method

The main part of our reconstruction method is based on Texture Optimization [Kwa05]. The algorithm performs pixelwise optimization of the output texture, minimizing a global energy function assembled from local similarity measures of neighborhoods in the input and output image. The energy function is constructed as follows:

$$E_t(\mathbf{x};\{\mathbf{z}_p\}) = \sum_{p\in X^\dagger} \|\mathbf{x}_p - \mathbf{z}_p\|^2, \tag{2}$$

where $\mathbf{x}$ is the vectorized version of the texture X being optimized, $\mathbf{x}_p$ the subvector of $\mathbf{x}$ containing pixels in a neighborhood of fixed size around p, $\mathbf{z}_p$ a vectorized neighborhood in the example texture Z which is most similar to $\mathbf{x}_p$ under the Euclidean norm, and $X^\dagger \subset X$ the set of neighborhood centers to consider. Optimization takes place in an Expectation-Maximization-like manner, alternating between optimizing the $\mathbf{x}_p$ and $\mathbf{z}_p$. For constrained synthesis, E_t is augmented by a control term E_c, depending on $\mathbf{x}$ and a control vector $\mathbf{u}$, weighted by a coefficient λ:

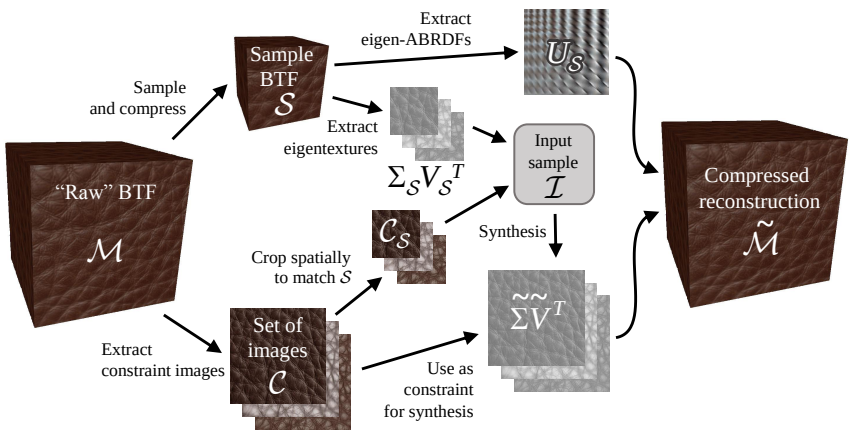
$$E(\mathbf{x}) = E_t(\mathbf{x};\{\mathbf{z}_p\}) + \lambda E_c(\mathbf{x};\mathbf{u}). \tag{3}$$

In this paper, we use a value of $\lambda = 20\ 000$, leading to a ratio of nearly one-to-one between constraint satisfaction and neighborhood similarity on the $151 \cdot 151 = 22\ 801$ views in $\mathcal{S}$, see also Subsection 3.2.

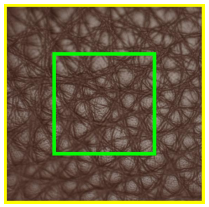
To capture material structure at different scales, Kwatra *et al.* propose to perform synthesis in a multi-level fashion, starting from coarse image resolution and large neighborhoods, successively reducing neighborhood size and downsampling factor. We use a neighborhood size of 8 ABRDFs and downsampling factors $1/4, 1/2$ and 1 for our experiments.

Results from works like those by Ruiters *et al.* [Rui10] have shown that during texture synthesis, high-dimensional details are often lost. Therefore, they propose to apply an additional step of *Statistical Synthesis* as introduced by Portilla and Simoncelli [Por00] to compensate for this effect. Their method uses a wavelet transform of a sample image to generate constraints which are then used to generate new images obeying the same statistics. We follow this approach, but incorporate it into a second optimization step applied to the result of Texture Optimization.

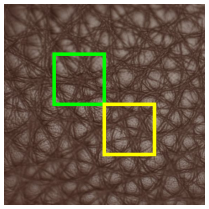
Statistical Synthesis applies its modifications uniformly over the Texture Optimization output, leading to the



(a) Workflow for evaluating our methods on ground-truth data.



(b) $256 \times 256 \rightarrow 512 \times 512$.



(c) $128 \times 128 \rightarrow 128 \times 128$.

Figure 3: Our experimental set-up: (a) illustrates the workflow for input generation and ground-truth reconstruction, while (b) and (c) denote the regions in $\mathcal{M}$ from which the inputs are cropped: From the region bordered green, the complete measurement $\mathcal{S}$ is taken, while the images for $\mathcal{C}$ are taken from the area bordered yellow (outer region in (b)).

visible effect that sharpness in the region corresponding to the sample gets exaggerated. To overcome this issue, we then search for an optimal trade-off between the results before and after Statistical Synthesis while still obeying the constraints. Formally, we minimize for each position p on the material surface the function

$$f(p) = \alpha||x^p - b_1^p||^2 + \beta||x^p - b_2^p||^2 \tag{4}$$

under the constraint $A \cdot x^p = b^p$, where x^p are the color values of the ABRDF to optimized, b_1^p, b_2^p the ABRDF's color values before and after Statistical Synthesis, α and β constant factors levelling between the two objectives, A those rows of the basis $U_\mathcal{S}$ of $\mathcal{S}$ which correspond to the constraint images, and b^p being the color values in the corresponding positions in the synthesis constraints unrolled as vectors. From now on, we will omit the superscript p in favor of readability.

By transforming Equation 4, we get

$$\begin{aligned} f(p) &= 1/2\, x^T \cdot \big(2(\alpha+\beta)I\big) \cdot x \qquad (5) \\ &\quad + \big(-2(\alpha b_1^T + \beta b_2^T)\big) \cdot x + C \\ &= 1/2\, x^T \cdot Q \cdot x + c^T \cdot x + C \qquad (6) \end{aligned}$$

with c, C constants, which is known to be solvable using quadratic programming.

We implemented this optimization in a MATLAB script, iteratively alternating between applying Statistical Synthesis to the result of the previous step and optimizing for x that minimizes Equation 6 until the result does not change significantly anymore. In our experiments, two to five iterations were sufficient to reach convergence. Algorithm 1 summarizes our complete approach. The result of this algorithm, applied to the inputs $\mathcal{I}$ and $\mathcal{C}$ as constructed in the following subsection, is a reconstruction $\widetilde{\Sigma} \cdot \widetilde{V}^T$ of the eigentextures of $\mathcal{M}$. Combined with $U_\mathcal{S}$, it forms an approximation $\widetilde{\mathcal{M}}$ of $\mathcal{M}$:

$$\mathcal{M} \approx \widetilde{\mathcal{M}} = U_\mathcal{S} \cdot \widetilde{\Sigma} \cdot \widetilde{V}^T. \tag{7}$$

Algorithm 1 BTF Enlargement

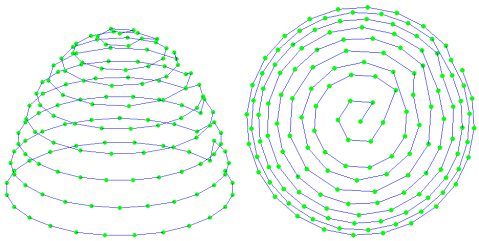
Input: Preprocessed sample BTF $\mathcal{I}$, constraint set $\mathcal{C}$.
Output: Synthesized $\widetilde{\Sigma} \cdot \widetilde{V}^T$ for $\widetilde{\mathcal{M}}$.

```
R ← Texture Optimization on I and C;
R' ← Statistical Synthesis on R;
while R ≠ R' do
    R' ← R;
    R' ← Statistical Synthesis on R;
    R' ← Quadratic Programming on R and R';
end while
ΣV ← Remove constraint information from R';
return ΣV.
```

3.2 Input Generation

All inputs for ground truth reconstruction are extracted from a matrix $\mathcal{M}$ containing the raw data of an angularly and spatially complete BTF measurement captured using the Dome I device described by Müller *et al.* [Mül05]. Materials are sampled using 151 lighting and viewing directions, resulting in 22 801 textures stored in $\mathcal{M}$. The full sampling is sketched in Fig. 4, where each green dot denotes the position of a camera with built-in flashlight.

From $\mathcal{M}$, we extract a subset $\mathcal{S}$ cropped in the spatial domain, taking the role of the representative measure-



(a) Viewed from the side. (b) Viewed from top.

Figure 4: Positions of the 151 cameras and light sources in the Dome I BTF acquisition device.

ment. It is compressed using singular value decomposition, as sketched in Fig. 2:

$$\mathcal{S} \approx U_{\mathcal{S}} \cdot \Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}^T. \tag{8}$$

The input sample for Texture Optimization is then given by

$$Z = \Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}^T. \tag{9}$$

Also from $\mathcal{M}$, a set $\mathcal{C}$ is extracted, consisting of images in full spatial resolution, but only from few lighting and viewing directions. These are used as the constraint images. For the synthesis algorithm, the input sample also has to contain a constraint set $\mathcal{C}_{\mathcal{S}}$ corresponding to $\mathcal{C}$. In a real-world scenario, the images constituting $\mathcal{C}_{\mathcal{S}}$ should ideally be taken under the same conditions as those in $\mathcal{C}$. For our experiments, $\mathcal{C}_{\mathcal{S}}$ is extracted from $\mathcal{C}$ by spatial cropping and then stacked together with $\Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}$ to form a multi-channel image serving as the input sample $\mathcal{I}$ in Algorithm 1.

The cropping of $\mathcal{M}$ used to extract $\mathcal{S}$, and also of $\mathcal{C}$ to extract $\mathcal{C}_{\mathcal{S}}$ is illustrated in Fig. 3b. The inner region, bordered green, depicts the spatial region from which $\mathcal{S}$, spanning 256×256 ABRDFs, is taken, while the yellow bordered outer region is the area to reconstruct. From this region, sized 512×512 ABRDFs, the images in $\mathcal{C}$ are extracted.

Naturally, one would not want to take too many photographs, so we only considered samplings containing no more than ten images:

1. Top light, top view image as only constraint, and
2. a sampling consisting of the top light, top view perspective plus nine randomly chosen combinations of lighting and viewing directions.

To simulate the situation of taking the constraint photographs under diffuse daylight, we also evaluate the use of a *mean color image* as constraint. One motivation for this idea is the possible effect that from some lighting directions, highlights might mask the material's structure such that, for a bad choice of constraint images, the Texture Optimization algorithm might still lack information for a faithful reproduction of these features. The constraint image c is constructed from the material measurement by fixing one viewing position (top view in our case) while averaging between the color values for all viewing directions:

$$c(x,y) = \sum_{(\theta_l,\phi_l \in \Omega)} \mathcal{M}(x,y,\theta_l,\phi_l,\theta_v,\phi_v), \tag{10}$$

with θ_v, ϕ_v fixed, and Ω the set of all lighting and viewing angles from which $\mathcal{M}$ is sampled. An example image generated for a piece of wood is shown in Fig. 5.

Figure 5: Example of a mean color image generated from a BTF measurement for a piece of wood. (Brightness enhanced for viewing.)

4 EXPERIMENTAL RESULTS

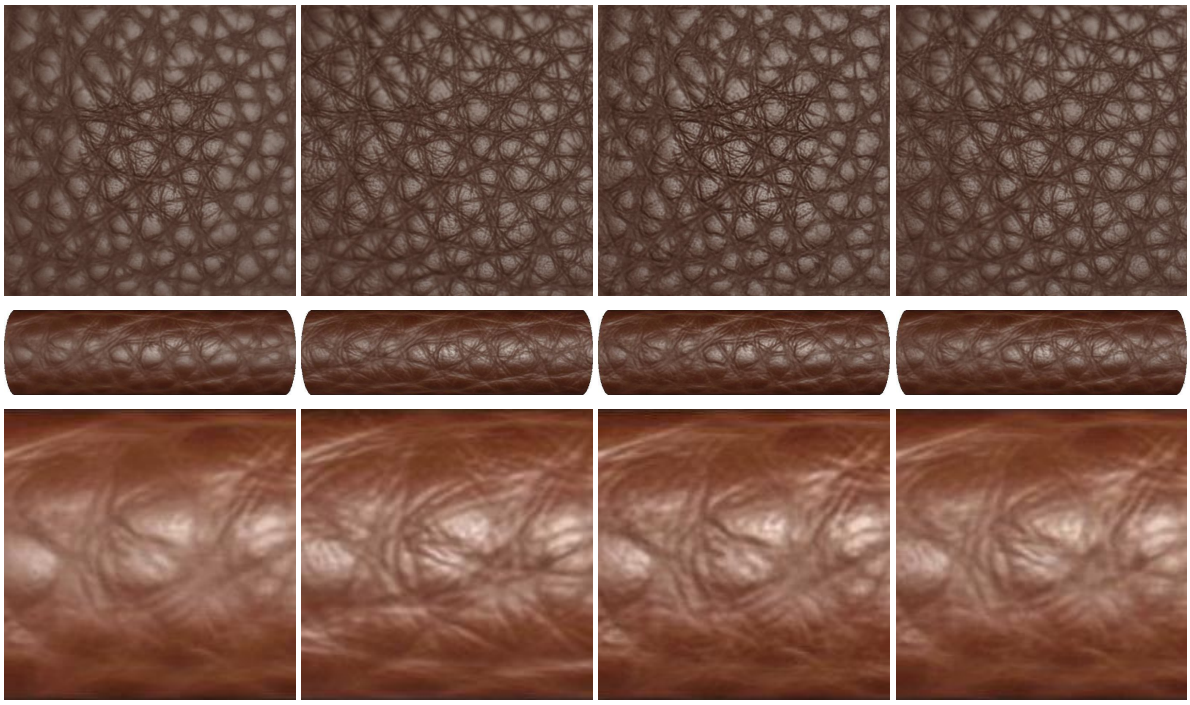
4.1 Single Constraint

All computations were performed on a desktop computer with Intel Core i7-2600K CPU at 3.4 GHz and 16 GB of RAM, using a C++ implementation of Texture Optimization, a MATLAB MEX implementation of Statistical Synthesis and a MATLAB script for iterative optimization, all operating in 64 bit. Texture Optimization and statistical synthesis contain CUDA-accelerated computations performed on an NVIDIA GeForce GTX 560 Ti graphics adapter.

Our first experiments took place using only the three color channels of the top light, top view image as constraints. An example for the reconstruction of a leather measurement is shown in Fig. 6a, with the compressed reference in Fig. 6b. One observation from these images is the severe blurring outside of the measured region of the reconstructed BTF. Fig. 6c shows that the additional step of statistical synthesis after texture optimization indeed helps deblurring the result, but at the cost of oversharpening in the sample region. After performing the additional optimization step, the result appears to be visually much closer to the ground truth data, as can be observed in Fig. 6d.

Running times for these results are about one hour for Texture Optimization plus additional time of approximately 20 minutes for the iterative optimization.

As one can see in the renderings onto a cylinder in the middle row of Fig. 6, details of the material's structure are lost for flat viewing and lighting angles. The effect is best visible when comparing the bottom thirds of



(a) Reconstruction (Texture Optimization only) (b) Reference (c) Reconstruction after Statistical Synthesis (d) Reconstruction after iterative optimization

Figure 6: Comparison of the reconstruction results for a sample of leather, rendered both onto a flat surface (top row) and onto a cylinder (middle row). The bottom row shows a detail of the cylinder with enhanced contrast to make differences more recognizable.

the zoomed and contrast enhanced images in the bottom row. This behaviour is exactly what one would expect, as the synthesis algorithm lacks information on how to choose the correct color values for those angles when only one constraint image is provided.

4.2 Multiple and Alternative Constraints

An obvious idea is to add more constraint images, which corresponds to taking more photographs in a real-world setting. Thus, this subsection demonstrates the change in result quality when rising the number of constraint images from one to ten.

To enable a faster evaluation, we switched to the setting in Fig. 3c when producing the results in the remainder of this paper. Here, the sample area as well as the region to be reconstructed span 128×128 ABRDFs each, leading to computation times of about half an hour for Texture Optimization and iterative optimization together.

Result images are given in Fig. 7c. Comparison with the images rendered using only one constraint image (see Fig. 7b) suggest that additional constraints can indeed help improve details of the structure, but not without cost. Some aspects of the structure are still not met correctly, *e.g.* the results appear to be noisier now, especially the carpet.

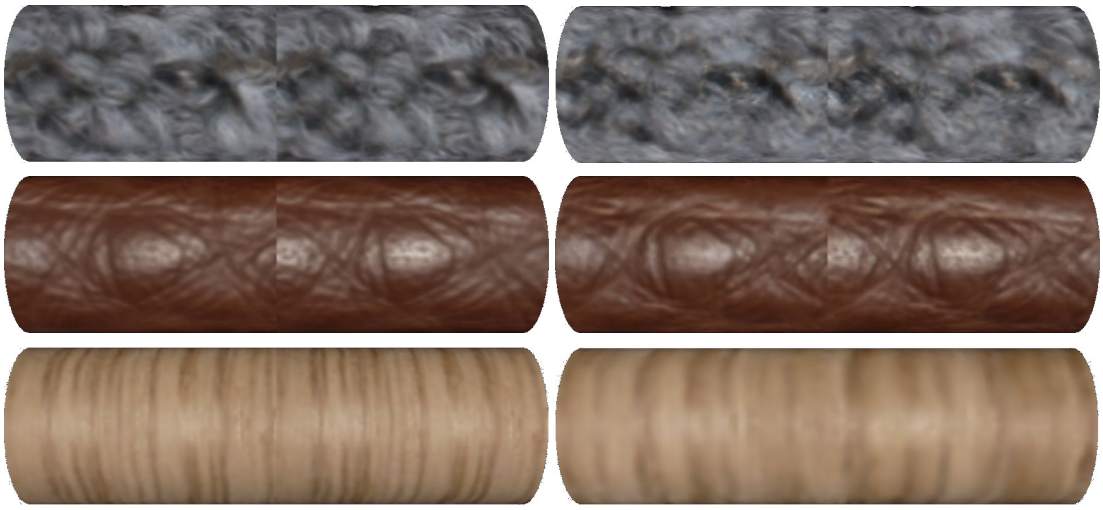
One approach to avoid these drawbacks is the use of a mean color image as a constraint, as proposed in Section 3.2. Here, the contribution of a single light is only $1/151$, such that theoretically, a highlight in an unfortunate position cannot introduce a too large error onto the overall reconstruction quality. When inspecting the result images in Fig. 7d, the results again look blurred, and a change in color or brightness has occurred. Thus, using mean color images as only constraint might not be the right choice, but it might still prove helpful in combination with different kinds of constraints.

As a numeric measure for the reconstruction error, we use the *average ABRDF RMSE* as also used by Ruiters *et al.* [Rui09]:

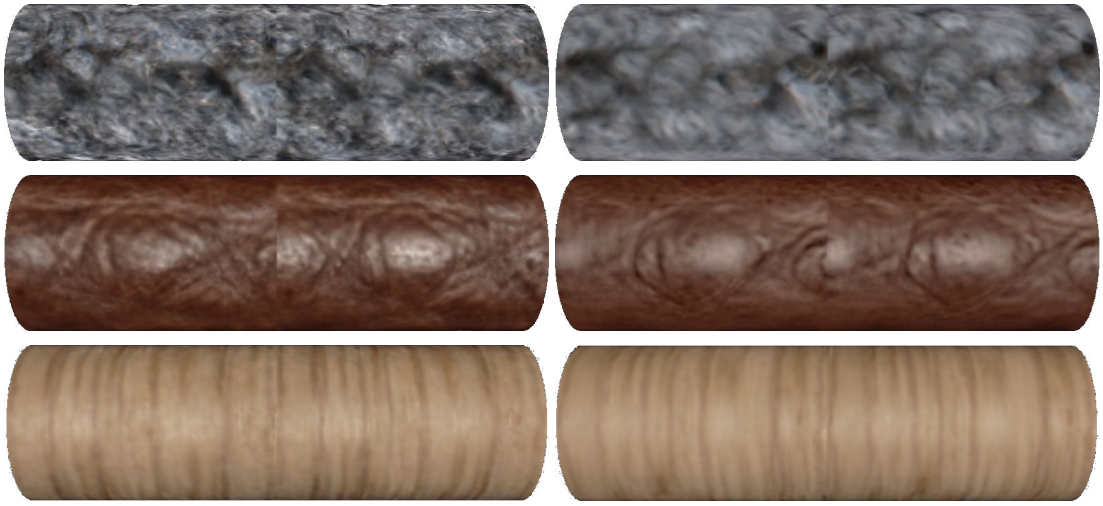
$$E = \frac{1}{n} \cdot \sum_{i=1}^{n} \sqrt{\frac{||M_i - \widetilde{M}_i||^2}{m}}, \tag{11}$$

with M_i the i-th ABRDF of the reference BDI matrix, $\widetilde{M}_i$ the i-th ABRDF of the reconstructed BTF matrix, n the number of ABRDFs of M (and $\widetilde{M}$, respectively) and m the number of components of each ABRDF vector. It must be pointed out that this error measure cannot be used to compare between reconstruction quality for *different* materials.

Error values for the reconstruction results illustrated in this section are given in Table 1. Note that the reference also exposes an error $E_{\text{ref}} > 0$, as we are comparing compressed BTFs with 64 components to the uncompressed data. We observe that in most cases, the



(a) Reference. (b) Reconstruction (1 constraint).



(c) Reconstruction (10 constraints). (d) Mean color image as only constraint.

Figure 7: Comparison of reconstruction results for carpet, leather and wood (from top to bottom). Constraints were: Top light, top view (Fig. 7b), random sampling (Fig. 7c) and mean color image (Fig. 7d). All images show the results after two phases of iterative optimization.

value of reconstruction error increases with rising effort in constraint generation. For the results with ten constraint images, this might be surprising, but it suits to the observation that they look less blurred, but noisier than those produced using only one constraint image.

5 DISCUSSION AND OUTLOOK

In this paper, we have demonstrated how a measurement of only a subset of a material's surface can be completed to obtain a BTF for the full material sample. It is obvious that even with our method working only on compressed data, one will soon reach the limitations of existing computer systems in terms of memory consumption and computational power.

Thus, one step toward the applicability of our ideas to larger problem instances has to be the development of faster reconstruction methods, *e.g.* based on fitting linear models. Ideally, all information needed would be preprocessed and stored in a more compact fashion, allowing fast on-demand reconstruction, ideally directly on the graphics hardware, instead of precomputing several mega- to gigabytes of reflectance data.

Also, the concept of *level-of-detail* should be considered: If only one BTF is visible, probably the object onto which it is mapped is viewed from a nearby viewing point, demanding all available details to be rendered. On the other hand, if multiple BTFs are visible in one scene, it is not implausible to assume that neither of them needs to be kept in memory in full detail.

When sticking to Texture Optimization as a method for BTF enlargement, there are also several entry points for further evaluation. A deeper understanding of the reasons for the blurring and the exaggerated sharpness

Table 1: Average ABRDF RMSE values for carpet, leather and wood reconstructions. Column legend: (1): Texture Optimization only, (2): Texture Optimization plus Statistical Synthesis, (3): Optimized.

(a) Carpet: $E_{\text{ref}} = 0.0095$.

	(1)	(2)	(3)
1 Constr.	0.0253	0.0306	0.0303
Random	0.0215	0.0262	0.0423
Mean Color	1.0030	1.0097	1.0079

(b) Leather: $E_{\text{ref}} = 0.0069$.

	(1)	(2)	(3)
1 Constr.	0.0160	0.0182	0.0185
Random	0.0153	0.0175	0.0238
Mean Color	1.0022	1.0017	1.0020

(c) Wood: $E_{\text{ref}} = 0.0029$.

	(1)	(2)	(3)
1 Constr.	0.0132	0.0161	0.0158
Random	0.0117	0.0151	0.0155
Mean Color	1.0072	1.0058	1.0064

might help to simplify the reconstruction method even to the point where the additional optimization step becomes obsolete. A systematical evaluation of the influence of different values for the weighting coefficient λ in Eq. 3 might be another aspect.

It is also not clear if a different method for preparing the representative measurement could provide improvements. This also includes questioning the use of logarithmic range reduction during the compression step.

As we have seen from the results in section 4, the choice of constraint images has a large impact on the visual quality of the reconstruction results. Thus, an extensive investigation on the choice of suitable samplings, or, more general, methods for constraint generation, might help improving result quality or reduce the number of images needed to be taken. Aspects under consideration should contain convenience of acquisition as well as a reduction of processing effort and memory consumption.

A major challenge which still has to be tackled is the step from our idealized input data to a more realistic setting. Currently, all inputs originate from existing BTF measurements. Especially the constraint images are already resampled and reprojected, such that all pixels are really taken from the same viewing angle and illuminated from the same lighting angle, which is not true for real photographs. Thus, one would either have to implement a constraint acquisition method exhibiting exactly these properties, or to perform preprocessing on real-world data to match the requirements.

6 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.

7 REFERENCES

[Dan99] Dana, K. J., Van Ginneken, B., Nayar, S. K., and Koenderink, J. J. Reflectance and texture of real-world surfaces. In ACM Transactions on Graphics (TOG), 18(1):pp. 1–34, 1999.

[Don10] Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., and Guo, B. Manifold bootstrapping for SVBRDF capture. In ACM Transactions on Graphics (TOG), 29(4):p. 98, 2010.

[Don13] Dong, Y., Lin, S., and Guo, B. *Material Appearance Modeling: A Data-Coherent Approach*. Springer-Verlag New York Incorporated, 2013. ISBN 9783642357763.

[Fur05] 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*, pp. 77–82. Citeseer, 2005.

[Hai05] Haindl, M. and Hatka, M. BTF roller. In *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pp. 89–94. 2005.

[Hai13] Haindl, M. and Filip, J. *Visual Texture: Accurate Material Appearance Measurement, Representation and Modeling*. Springer, 2013.

[Kwa05] Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 795–802. ACM, 2005.

[Leu07] Leung, M.-K., Pang, W.-M., Fu, C.-W., Wong, T.-T., and Heng, P.-A. Tileable BTF. In Visualization and Computer Graphics, IEEE Transactions on, 13(5):pp. 953–965, 2007.

[Mül05] Müller, G., Meseth, J., Sattler, M., Sarlette, R., and Klein, R. Acquisition, synthesis, and rendering of bidirectional texture functions. In *Computer Graphics Forum*, vol. 24, pp. 83–109. Wiley Online Library, 2005.

[Por00] Portilla, J. and Simoncelli, E. P. A parametric texture model based on joint statistics of complex wavelet coefficients. In International Journal of Computer Vision, 40(1):pp. 49–70, 2000.

[Rui09] Ruiters, R., Rump, M., and Klein, R. Parallelized matrix factorization for fast BTF compression. In *Proceedings of the 9th Eurographics conference on Parallel Graphics and Visualization*, pp. 25–32. Eurographics Association, 2009.

[Rui10] Ruiters, R., Schnabel, R., and Klein, R. Patch-based texture interpolation. In *Computer Graphics Forum*, vol. 29, pp. 1421–1429. Wiley Online Library, 2010.

[Rui13] Ruiters, R., Schwartz, C., and Klein, R. Example-based interpolation and synthesis of bidirectional texture functions. In Computer Graphics Forum (Proceedings of the Eurographics 2013), 32(2):pp. 361–370, 2013.

[Sch13] Schwartz, C., Sarlette, R., Weinmann, M., and Klein, R. DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling*, pp. 25–31. The Eurographics Association, 2013.

[Sch14] 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. In Sensors, 14(5), 2014. ISSN 1424-8220.

[Sue00] Suen, P.-h. and Healey, G. The analysis and recognition of real-world textures in three dimensions. In Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(5):pp. 491–503, 2000.

[Ton02] 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)*, vol. 21, pp. 665–672. ACM, 2002.

[Wei09] 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*, pp. 93–117. 2009.

[Zho05] Zhou, K., Du, P., Wang, L., Matsushita, Y., Shi, J., Guo, B., and Shum, H.-Y. Decorating surfaces with bidirectional texture functions. In Visualization and Computer Graphics, IEEE Transactions on, 11(5):pp. 519–528, 2005.