Eurographics Symposium on Rendering (2007)
Jan Kautz and Sumanta Pattanaik (Editors)

# Procedural Editing of Bidirectional Texture Functions

Gero Müller, Ralf Sarlette and Reinhard Klein

Institute for Computer Science II, University of Bonn †

**Abstract**

*Measured material representations like Bidirectional Texture Functions or Reflectance Fields offer very realistic appearance but the user is currently not capable of changing this appearance in an effective and intuitive way. Such editing operations would require a low-dimensional but expressive model for appearance that exposes only a small set of intuitively editable parameters (1D-sliders, 2D-maps) to the user but preserves all visually relevant details. In this paper we present a novel editing technique for complex spatially varying materials. It is based on the observation that we are already good in modeling the basic geometric structure of many natural and man-made materials but still have not found effective models for the detailed small-scale geometry and the interaction of light with these materials. Our main idea is to use procedural geometry to define the basic structure of a material and then to enrich this structure with the BTF information captured from real materials. By employing recent algorithms for real-time texture synthesis and BTF compression our technique allows interactive editing.*

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

## 1. Introduction

Despite their photo-realistic appearance image-based material representations like the Bidirectional Texture Function (BTF) [DvGNK97] are still far from being used as an ordinary modeling resource like textures or scanned geometry. The main reasons for that are the still elaborate measurement procedures, the demanding storage and rendering requirements and last but not least the very limited possibilities for effective user-guided modification and editing. This last point deserves special attention. In fact a material model with an intuitive editing interface *and* the visual quality of image-based representations would be the holy grail of appearance modeling. Unfortunately, the space of physically valid materials seems to be extremely complex and there is strong evidence that even for the space of diffuse textures alone no fixed dimensionality can be found [HH99].

As a consequence the parametric appearance models currently used in computer graphics have been developed for more or less comprehensive subclasses of materials only. E.g., opaque and uniform materials like plastic or metal are reasonably represented by simple analytical BRDF models [NDM05]. Many natural and man-made materials like wood, marble or brick walls can be modeled well by procedural texturing techniques [EMP*94].

While similar techniques became quasi standards in appearance modeling over the last almost 30 years their limitations in terms of photo-realism and intuitive editing are well known and have promoted the development of non-parametric or so-called data-driven editing techniques like constrained texture synthesis [Ash01], non-linear manifold analysis [MPBM03] or warping-based texture interpolation [MZD05] that provide modification of measured appearance data like textures or BRDFs while retaining their superior visual quality.

Unfortunately, these techniques cannot be easily applied to BTFs. The main reason is the data explosion which prohibits the practical application of the aforementioned methods because they where designed for BRDFs or diffuse texture images while even a single BTF consists of thousands of (apparent) BRDFs and images respectively. A compact factorization into a few BRDFs and material maps as introduced in [LBAD*06] helps to alleviate the problem for very

† e-mail: {gero, sarlette, rk}@cs.uni-bonn.de

thin materials (spatially varying BRDFs) but can not be applied to materials with significant height variation.

We assume in this work that the material's spatial appearance variation results primarily from its 3D meso-structure, i.e., the material could basically be represented by only a few BRDFs and its meso-scale geometry (meso-structure). In this case the measured per-texel data

$$b_{\mathbf{x}}(\omega_i, \omega_o) := BTF(\mathbf{x}, \omega_i, \omega_o)$$

is a so-called *Apparent* BRDF which contains not only the micro-scale reflectance but also effects resulting from the 3D meso-structure like shadows, masking and inter-reflections [WHON97]. If now the known meso-structure is edited all these ABRDFs have to be changed accordingly to reproduce the characteristic look of the material. If the exact BRDFs are known this could in principle be achieved by simulating the light-transport on the edited meso-structure using a global-illumination algorithm, what is currently not a practical solution. Another approach would be to warp and interpolate between two BTFs by warping their meso-structures and interpolating the corresponding per-texel ABRDFs as proposed for rgb-textures in [MZD05]. Unfortunately, there are currently only very few BTFs publicly available which are suitable for pairwise morphing which fairly limits the range of possible edits. Another problem with this approach is the feature matching and the ABRDF interpolation itself because a realistic smooth morph restricts the search range for matching of the meso-structure which in turn will lead to matchings between rather incompatible ABRDFs that cannot be linearly interpolated without artifacts.

Based on these observations and assuming that the meso-structure of the given BTF is known at least approximately we propose the following novel procedure for editing the BTF's meso-structure and reflectance. To edit the meso-structure while keeping the micro-scale reflectance constant, the meso-structure itself is edited and the resulting corresponding ABRDFs are generated by a constraint synthesis approach, i.e. are copied from the original BTF. Although this procedure only rearranges the measured ABRDFs impressive edits can be achieved with interactive feedback (cf. Figure 5). To edit the reflectance of the BTF a data-driven approach is taken. Given an additional BTF with resembling meso-structure the individual ABRDFs of the two BTFs are merged by first identifying ABRDFs with most similar meso-structure and then interpolating them. Of course this process can be used for interpolating not only between two but also several BTFs. This enables meaningful interpolation of BTFs which was not demonstrated before. In combination with a procedural model for meso-structure we use this kind of interpolation to create a combined procedural and data-driven model for leather BTFs which for the first time offers a fully parameterized model for the BTF of a particular material class and allows for smooth navigation in the space of the measured samples (cf. Figures 8 and 10).

The remainder of this paper is organized as follows. We discuss previous work in Section 2 and give an overview of our editing technique in Section 3. The algorithmic details and results are given in Section 4 (meso-structure constrained synthesis) and Section 5 (hybrid procedural and data-driven leather model). Conclusions and directions for future work are presented in Section 6.

## 2. Previous Work

The editing of measured appearance is a relatively new research area but it is based on a great deal of work from the fields of appearance modeling as well as texture synthesis and transfer from which we will only mention the most related work.

**Appearance modeling and editing** Measuring reflectance and fitting analytical models to the data as introduced by Ward [War92] has a long time tradition in computer graphics. Such a representation can easily be edited by changing the parameters of the analytical model. Matusik et al. [MPBM03] questioned this measure-fit paradigm by claiming that analytical reflectance models only describe a restricted class of materials and that their parameters often have no intuitive physical meaning. Consequently, they proposed a data-driven reflectance model based on a large collection of measured BRDFs and applied data (manifold) analysis and perceptual studies to find perceptually meaningful directions in the space of these measured BRDFs.

While such an approach seems to be feasible in the case of BRDFs no analogue model has been proposed for spatially varying reflectance yet. Instead researchers concentrated on the aspects of acquisition, synthesis, compression and rendering of spatially varying appearance as summarized in the report of Müller et al. [MMS*05]. Only recently the editing aspect came into play. E.g., Weyrich et al. [WMP*06] proposed a measurement-based reflectance model for human skin based on the Torrance-Sparrow BRDF model. The spatial variation of the skin is captured in simple albedo maps which can be transfered between faces using the histogram equalization technique of Heeger and Bergen [HB95]. Unfortunately, such an approach is infeasible for materials with depth variation because in this case analytical reflectance models perform poorly [MMK04].

The non-parametric Inverse Shade Tree approach by Lawrence et al. [LBAD*06] allows editing of measured spatially varying materials but the used low-term factorizations again restrict the method to flat materials which can be described by a few BRDFs only. Recently, Kautz et al. [KBD07] introduced photo-editing-like editing of BTFs. They operate on the full BTF data and achieve interactive performance by employing a sophisticated out-of-core data management. While their method offers a set of general editing operators our technique is designed for creating specific data-driven models for BTFs (like the leather model we introduce in Section 5) that then can be manipulated using a few sliders.

**Texture synthesis and transfer** The inherent complexity of texture has prevented the development of a general parametric texture model. Instead models for special types of textures have been invented. Procedural models [EMP*94] are suited for natural textures like wood or stone. Parametric models based on the Markov Random Field model (e.g., [FH03]) are only capable of reproducing mainly stochastic textures and are expensive to evaluate. In either case the selection of appropriate parameters might be non-trivial and many textures can not be reproduced using procedural models. Therefore, researchers started to focus on non-parametric models based on example textures [EL99, WL00]. The idea is to generate a larger but perceptually similar version of a small texture sample by iteratively choosing pixels or patches from the sample which "fit well" (in terms of a specific metric) into the already synthesized part of the output texture.

While apart from avoiding repetition artifacts the generation of larger versions of input textures might not sound that useful in itself the significance of texture synthesis for material editing became apparent with the idea of constrained texture synthesis [Ash01, HJO*01]. In this case constraints, typically given as images or flow fields (e.g., [KEBK05]), are used to guide the synthesis process and thus enable some kind of user control for texture synthesis. The question remains how to generate these constraints. Zhou et al. [ZDW*05] presented a method which uses constraint graph-cut synthesis to allow the interactive placement of BTFs on surfaces. The synthesis is used for seamless blending and not for editing the BTF itself. In the work of Mertens et al. [MKC*06] a method is presented that transfers the texture of a scanned object to arbitrary geometry which then acts as the constraint.

## 3. Method Overview

As sketched in the introduction we propose to edit and interpolate BTFs by using a given meso-structure as guidance for finding compatible ABRDFs. Building a practical BTF editing system based on this fundamental concept poses several practical problems.

First, the search for ABRDFs with compatible meso-structure requires to reconstruct the meso-structure of measured BTFs and has to be as fast and as accurate as possible to allow for interactive editing. We describe the algorithms to reach these goals in Section 4.

Then we need to discuss how the meso-structure constraint should be defined and edited. Several techniques like image-editing or warping can be used here but we propose to use procedural models for this purpose. This way we are able to present the user a fully parameterized model for a specific class of materials much like a material shader in a modeling and rendering software package. We present a procedural model for leather meso-structures in Section 5.1.

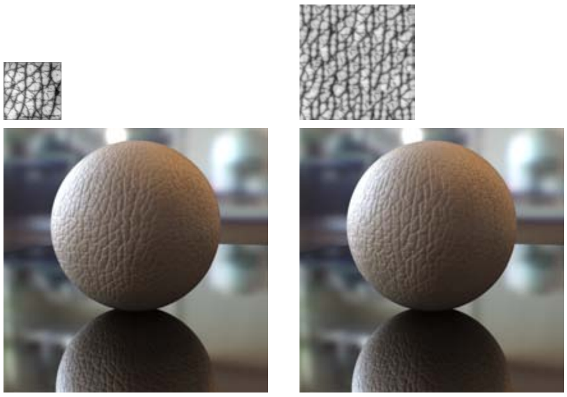

**Figure 1:** *Synthesizing BTFs by synthesizing only their meso-structure (top) and copying the corresponding reflectance values works well for many materials because occlusions, shadows and shading typically correlate strongly with meso-structure.*

Last but not least the meso-structure guided BTF interpolation requires to hold several BTFs in memory. Despite the recent advances in BTF compression these memory demands can introduce a serious performance penalty. Therefore, we describe in Section 5.2 an algorithm for compressing several BTFs simultaneously which offers significant memory savings.

## 4. Meso-structure constraint BTF synthesis

Synthesizing BTFs using reconstructed meso-structure as constraint was introduced by Liu et al. [LYS01]. Their main goal was to synthesize a continuous BTF from the sparse BTF samples offered in the CUReT database [DvGNK97]. To synthesize a novel view/light configuration they rendered the synthesized meso-structure with the corresponding viewing and lighting parameters and used the resulting image as guidance for copying appropriate sample blocks from the captured images. Thereby, they were able to enforce consistency across different view/light configurations although they synthesized each image of the BTF independently. This synthesis scheme is extremely slow, since thousands of images have to be synthesized which usually takes several hours.

Fortunately, due to recent progress in BTF measurement methodology much more densely sampled BTF data than the original CUReT data is now available. These datasets can be used as is without synthesis of novel view/light configurations. Instead, as shown in Figure 1, it often suffices to synthesize only the meso-structure and copy the corresponding reflectance values into the synthesized BTF.

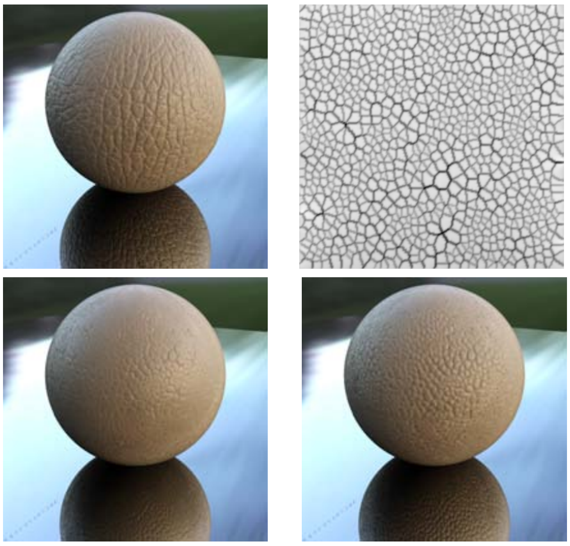

**Figure 2:** *Using geometrical features improves constraint synthesis. Top left: Original BTF. Top right: Heightfield constraint (generated with the procedural model from Section 5). Bottom left: Result with constraint synthesis using only the heightfield as constraint. Bottom right: Result using also normals and view-dependent occlusion.*

If we now provide another meso-structure to constrain the synthesis this process fits right into the image analogies framework of Hertzmann et al. [HJO*01] with the reconstructed meso-structure and the constraint acting as the unfiltered and filtered source images and the measured BTF as the unfiltered target image. Unfortunately, this approach is too slow for interactive editing because the high-dimensional BTF data is involved in the synthesis process. Therefore, we simply propose to perform the neighborhood matching without the BTF data. As expected, this simplification does not come without cost. If the constraint differs significantly from the reconstructed meso-structure it can happen that visually important structures brake up and are not correctly transferred especially for non-frontal viewing angles. As shown in Figure 2 this problem can be alleviated by extracting additional features like normals or view-dependent occlusion and including them into the synthesis. Of course this additional dimensionality again slows down the synthesis process. This problem can be solved by computing appearance space textures as in [LH06] and using them for the synthesis.

The whole interactive meso-structure constraint BTF synthesis process is illustrated in Figure 3. The building blocks of the algorithm are consequently:

- Meso-structure reconstruction
- Feature extraction
- Appearance space transformation and projection
- Constraint synthesis

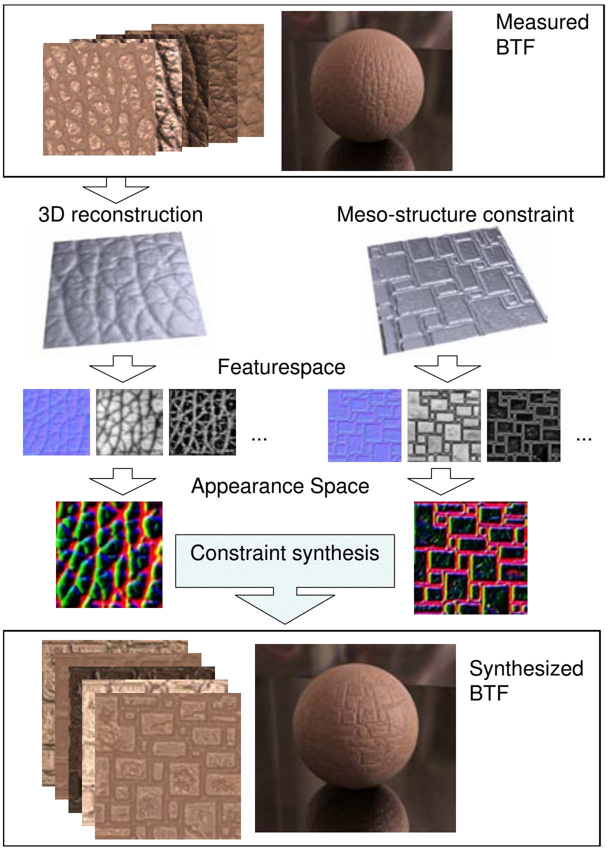


**Figure 3:** *Overview of meso-structure constraint synthesis: The reconstructed meso-structure is transformed into an appearance space texture. The features of the given constraint are then projected into the appearance space and used to constrain the BTF synthesis.*

We will give technical and implementation details of these parts in the following subsections.

### 4.1. Meso-structure reconstruction

Reconstructing the meso-geometry of textures is an active research topic in its own right. Since our samples are only moderately glossy and our method does not depend on a perfectly accurate reconstruction we found classical photometric stereo techniques [RTG97] and normal integration [FC89] to be sufficient for our experiments. To increase the reconstruction quality for more specular materials we plan to investigate more accurate reconstruction techniques like that of Chen et al. [TCS06] or Neubeck et al. [NZG05] in future work.

Figure 4 shows results of the photometric meso-structure reconstruction for some of the materials used in this paper. Please note that the BTF transferring method is robust against low-frequency errors typically present in photometric reconstructions since it is based on qualitative local feature similarity.

### 4.2. Feature Extraction

As mentioned above the heightfield alone often does not sufficiently constrain the synthesis. Therefore, we extract additional features from the heightfield like normals and view-dependent occlusion which we then use for multi-channel constraint texture synthesis.

To compute the normals we simply convert the heightfield into a triangular mesh. Computing occlusions is a little bit more difficult because doing it straightforward, e.g., using raytracing would be far to costly for interactive editing. We propose two solutions for that. If the constraint is an arbitrary heightfield given by the user we employ graphics hardware to rapidly compute occlusions via shadow mapping. Our optimized implementation is able to compute about 400 directions per second for a 60k vertices mesh on a NVidia 7900. This sampling resolution typically suffices to reliably compute a 5th order Spherical Harmonics expansion of the view-dependent occlusion. For procedurally generated constraints like the crack pattern texture which is described in Section 5 it is also possible to generate the occlusion map procedurally with almost no runtime overhead.

### 4.3. Appearance Space

As said in the previous subsection we extract several features from the given heightfields in order to drive the synthesis. If we use height, normals and a 5th order Spherical Harmonics expansion for view-dependent occlusion our textures actually have 29 channels. And while this feature set seems to suffice for the materials we have tested, it is likely, that for other materials we have not measured and tested yet additional features are necessary. If we now employ a pixel-based synthesis technique like k-coherance search [TZL*02] with a neighborhood search window of radius 3 the neighborhood vectors dimension is 1421! In fact, synthesizing a 256x256 texture then takes about 4 seconds with our only mildly optimized C++ implementation (Intel Core 2 Duo, 2.4 GHZ). A 3-channel texture (neighborhood space dimension 147) is finished in already 1 second.

A typical optimization in this case is to project all source neighborhood vectors on a linear subspace using PCA. But also in this case the runtime depends on the number of channels, since the synthesized neighborhoods must be projected onto the linear subspace which means a costly matrix-vector multiplication with neighborhood dimensionality for every synthesized pixel. Therefore, we adopt the appearance space synthesis strategy from Levebvre and Hoppe [LH06]. The idea is to synthesize directly the neighborhood subspace texture. As in [LH06] we typically use 8 channels for the subspace texture and 5x5 or 3x3 neighborhoods which leaves us with neighborhood vectors of dimensions 200 or 81 only.

Please note, that the meso-structure features and their appearance space have to be computed only once for each BTF. This can be done in a pre-process which then stores the results on disk for later access during the constraint synthesis.

### 4.4. Constraint Synthesis

After pre-computing the appearance space for the source BTF all what is left is to compute the features for the meso-structure constraint and project the neighborhoods of the resulting feature texture onto the appearance space of the source BTF. Then principally any exemplar based texture synthesis algorithm can be used to perform the constraint synthesis. The result of the synthesis process is then a coordinate map which is used to index into the original BTF.

Our current CPU implementation uses k-coherance texture synthesis [TZL*02] and synthesizes about 50k texels per second for $k = 8$ and neighborhood vectors with dimension 200. To make the editing even more interactive we plan to implement the synthesis process on the GPU using [LH06] which reports about 50 times faster synthesis times.

### 4.5. Results and Discussion

The results shown in Figure 5 demonstrate the effectiveness of the transfer technique for some sample materials. It is also clearly visible that the constraint should have at least roughly similar feature statistics than the source. If the constraint describes geometry that is not present in the source, then the results become more or less arbitrary. This can be seen in the bottom-right image of Figure 5 where the dots and the lines of the constraint texture are arbitrarily colored bright and dark. Furthermore, high-frequency details are not transferred because the constraint has no high-frequency information to guide the synthesis of the cloth details. This latter problem could be reduced by matching multi-scale statistical feature distributions of source and target as proposed by Mertens et al. [MKC*06].

A question that immediately arises in the context of BTFs is how they compare to standard techniques like normal-mapping. And since we only use meso-structural details to guide the transfer one might wonder why it should not suffice to render this meso-geometry using normal mapping probably in combination with the average BRDF of the material. Figure 6 shows clearly that it does not suffice but delivers a different kind of appearance. The reason is that normal maps give an impression of meso-structure but since important effects like meso-structural shadowing and masking and inter-reflections are not captured the appearance does not compare to real images.

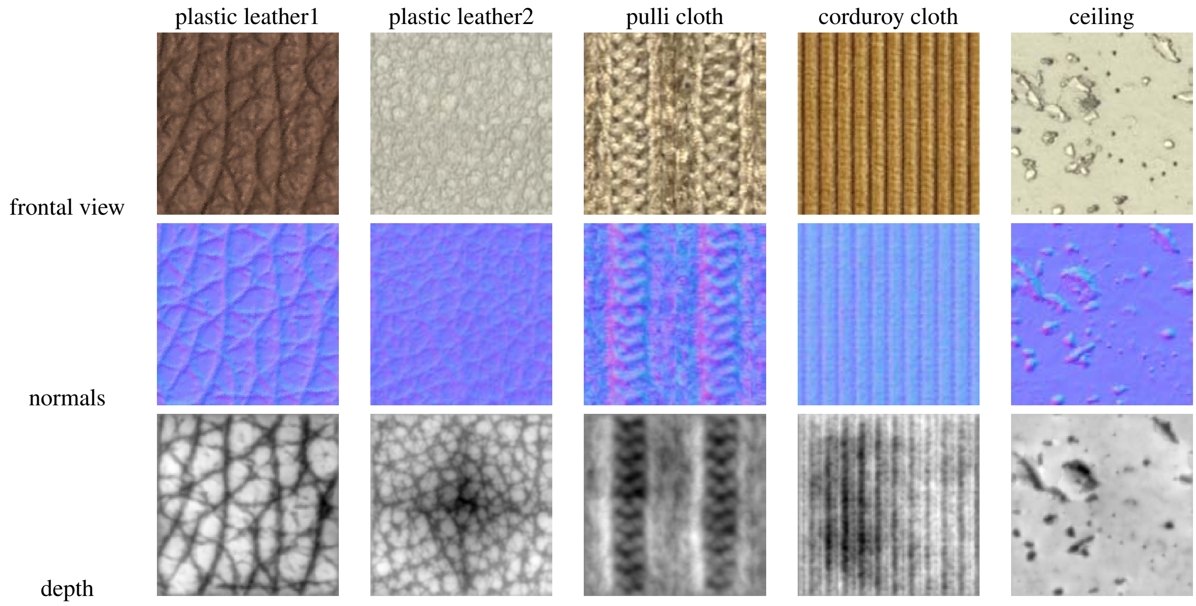


**Figure 4:** *Some meso-structure reconstructions of the materials used in this paper. All results where generated using photometric stereo and normal integration. The normal maps are shown using the usual RGB encoding. The cloth and ceiling materials are from the BTF database Bonn [SSK03].*

## 5. A hybrid procedural and data-driven leather model

The meso-structure constraint BTF synthesis introduced in the previous section is already a powerful tool to edit the appearance of measured BTFs. But it allows only the manipulation of meso-structural details. And creating or editing these details can still be a very cumbersome task. Therefore, we prefer to regard the appearance transfer technique as a tool to support the construction of specialized material models which combine the advantages of procedural textures and measured appearance like BTFs. In this section we will exemplify this idea by developing a data-driven model for leather-like materials which will be based on a database of leather-like measured BTFs, a procedural model for leather meso-geometry and the meaningful BTF interpolation using the meso-structure constraint BTF synthesis technique from Section 4.

### 5.1. Procedural leather meso-geometry

The main purpose of our procedural leather generator is the interactive and intuitive generation of the meso-structure of characteristical crack patterns which can be used to constrain the appearance transfer. Of course one of the many algorithms trying to simulate biological or physical crack pattern generation processes can be used here (the recent paper of Iben et al. [IO06] contains a good overview) but these methods are typically not interactive. Therefore, commercial leather shaders are often based on on Worley's cellular texture basis function [Wor96] which generates cracks by computing the Voronoi diagram of randomly distributed sites and can be evaluated very efficiently. The drawback of this method is that the generated Voronoi sites have an angled, not very realistic shape. Furthermore, we need a method to generate cracks of different depth.

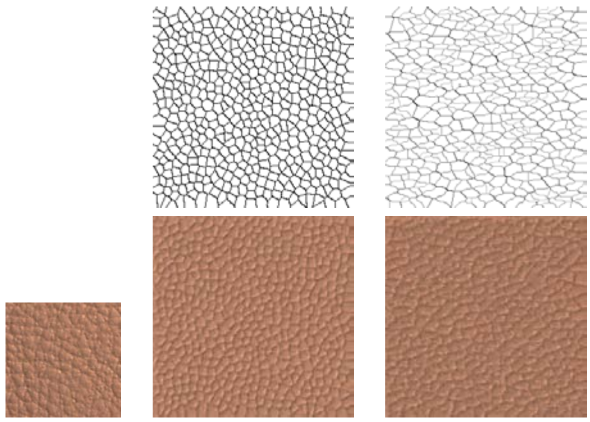

**Figure 7:** *Left: Real leather texture. Transferring to a typical Voronoi crack pattern gives the typical Voronoi-look with angle-shaped sites. Our modified crack pattern with polyline based sites and depth variation offers a much greater and more natural variation of site shapes.*

We propose two simple but effective solutions for these

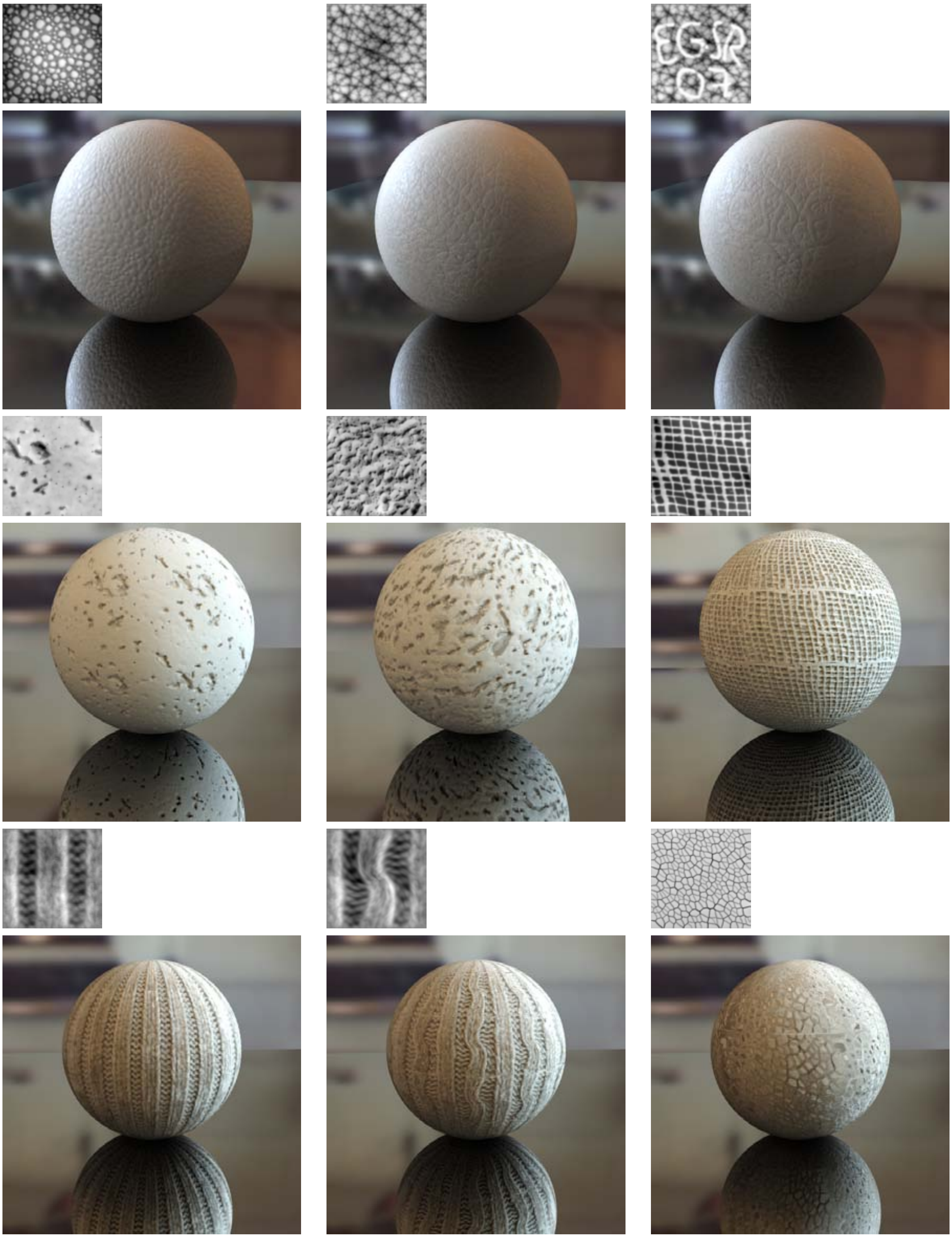

**Figure 5:** *Some results of our meso-structure constraint synthesis. The first material is a plastic leather imitation with round spots. The transfer to a more skin-like meso-structure works well and allows even the believable integration of artificial structures like text. The plaster ceiling can be safely transferred to more wild and more regular structures. Notice the BTF effect, which allows to "look into holes". To edit the cloth BTF a warping was applied to the source meso-structure (middle). Transfer fails, if source and target meso-structure feature distributions are too different like the cracks and the cloth in the lower right.*

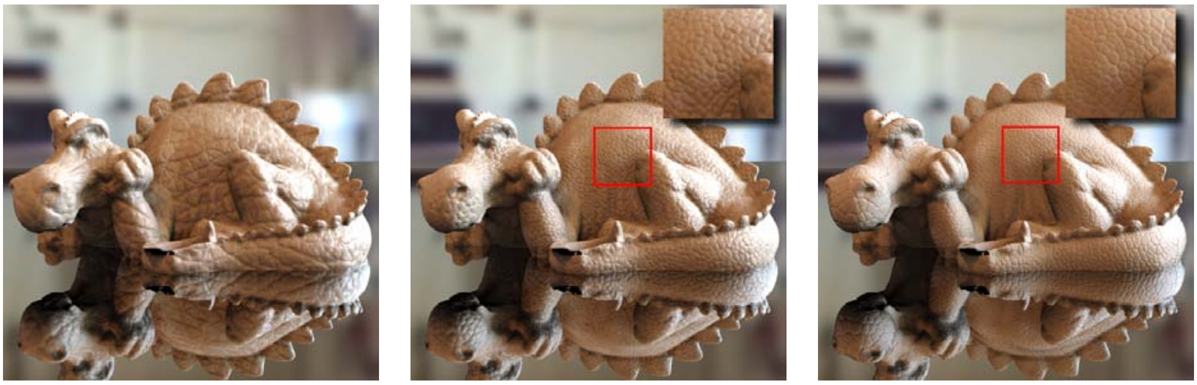


**Figure 6:** *Normal-mapping gives a much more artificial impression then meso-structure constraint synthesis. View-dependent occlusions and filtering, shadowing and global illumination phenomena like inter-reflections are completely lost.*

shortcomings. The first idea is to compute the generalized Voronoi diagram of polylines instead of points. This allows to generate more realistically shaped sites. The second trick is to base the crack depth on the distance to the center of the neighboring sites.

The distribution of the sites can be controlled with arbitrary 2D-density functions. By default we use a randomly generated Perlin turbulence texture as density. The degree of randomness can be set by the user and controls the "spotiness" of the crack texture. The shape of the sites, i.e. their "roundness" and "anisotropy" can be controlled by setting average length, preferred orientation and regularity of the polylines. As illustrated in Figure 7 these modifications offer more variety in the shape of the sites and thus enable a more realistic transfer.

#### 5.1.1. Crack Pattern Computation

The computation of a crack pattern from the aforementioned parameters can be done very efficiently. We use the algorithm of Kopf et al. [KCODL06] to distribute the sites given a 2D-density. To compute the crack pattern from the sites we employ the GPU-based drawing algorithm for generalized Voronoi diagrams by Hoff et al. [HKL*99]. After assigning a depth to each crack we use a depth-dependent gaussian crack profile to draw the crack heightfield into the z-Buffer.

#### 5.1.2. Morphing Procedural Crack Patterns

A great advantage of our procedural meso-geometry model is that natural morphs between different instances of the model can be computed easily. In our case this is not done by interpolating the model parameters but by finding appropriate correspondences between the sites. In particular we compute correspondences that minimize a morphing distance. Then we morph the structure by interpolating position and structure of the corresponding sites.

Our morphing distance is defined as follows: $A = \{a_i\}_{0<i\leq M}$ and $B = \{b_i\}_{0<i\leq N}$ shall be the polyline based Voronoi sites of two crack patterns with $M \leq N$. Then the morphing distance we try to minimize is simply

$$D(A,B) = min_f \sum_{i=0}^{M} d_\lambda(a_i, b_{f(i)})$$

where $f$ is an injective mapping between $\{1 \ldots M\}$ and $\{1 \ldots N\}$. Finding the assignment $f$ which minimizes $D$ is a classical combinatorial optimization problem which can be solved efficiently using the Hungarian algorithm [Kuh55]. The distance $d_\lambda$ between two sites $a_i$, $b_j$ is defined as a weighted sum of the euclidean distance between the centers of gravity $cog(\cdot)$ of the Voronoi sites and a shape distance $d_s(\cdot,\cdot)$ between the site borders of the Voronoi sites:

$$d_\lambda(a_i, b_j) = \lambda|cog(a_i) - cog(b_j)| + (1-\lambda)d_s(a_i, b_j)$$

There are numerous ways to define the 2D shape distance $d_s$. We used the approach of Sederberg and Greenwood [SG92] which is based on dynamic programming. This term can be used enforce that the shapes of two sites that are blended together are roughly similar. Typically we choose $\lambda$ close to 1 since a morph with a minimal movement of sites uses to be the visually the most pleasant.

The morphing itself is then based on interpolating the centers of of gravity of the sites and morphing the sites' polylines also based on Sederberg and Greenwood. Since the number of sites typically does not match, the number of sites is equalized by generating new sites which are smoothly blended in during the morphing. Figure 8 shows a morphing example which smoothly interpolates structure and reflectance of two leathers. Please take also a look at the accompanying video which gives an even better impression of the naturalness of the computed warp.

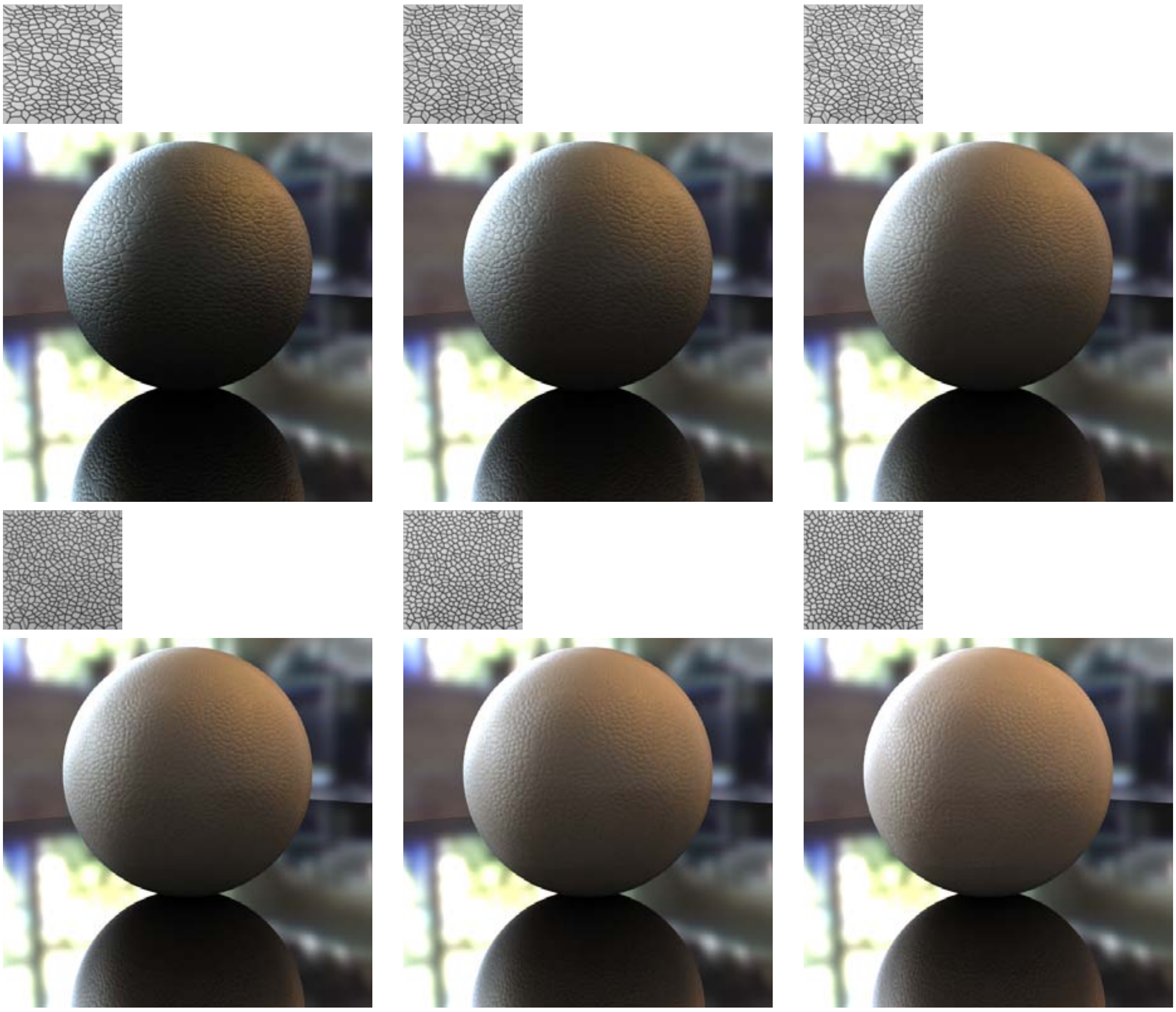

**Figure 8:** *Morphing between real gray leather and a beige imitated leather with regular structure. The morphed constraint texture is shown above each rendered image. The reflectance interpolates the measurements from the materials stored in the compressed BTF database.*

#### 5.1.3. Fitting Crack Patterns to Meso-Geometry

Fitting the crack pattern texture to real meso-geometry is desirable because it allows to represent real measured materials fully within our model and then to edit this representation.

In the case of crack patterns the fitting can be reduced to a computer vision task. First we detect the cracks in the meso-structure heightfield using a variant of a watershed image segmentation algorithm [RM00]. Then we compute a medial axis transform of the found crack pattern to get the corresponding Voronoi sites. Since we currently only allow simple polylines as Voronoi sites we loose a bit of information here. As illustrated in Figure 9 this procedure reconstructs the overall shape and distribution of cracks and sites well. Comparing both images a loss of high-frequency informationn can be noted in the reconstruction. This kind of information cannot be captured by a simple crack pattern. Again we can use the strategy of Mertens et al. [MKC*06] to match the feature statistics across different scales.

#### 5.1.4. Discussion

Of course numerous techniques that generate or modify heightfields can be used for guiding the meso-structure constrained synthesis. Figure 5 shows some examples generated using image editing and warping. Nevertheless, we have chosen to use procedural models as our editing paradigm because they perfectly complement the measurements in the sense that the user is guided to generate only constraints that are reasonable for the specific material class captured in the

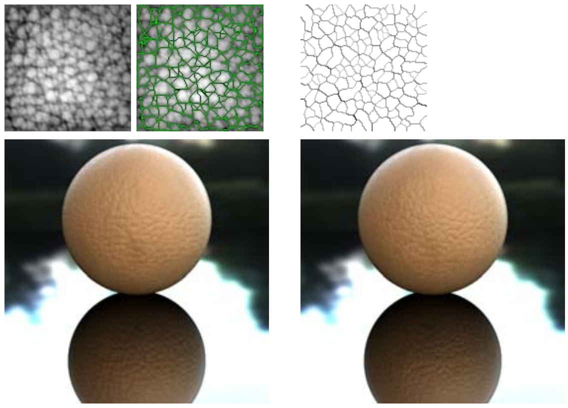

**Figure 9:** *Fitting the procedural crack pattern. Left: Original BTF with depth map and cracks (in green) found by watershed algorithm. Right: Resulting Voronoi cracks and transferred result.*

BTF database. We believe that this a passable way to deal with the complexity of textures and beyond because of the data explosion problem which forces us to find extremely compact models which most probably will be procedural or generative in some kind.

### 5.2. Data-Driven Modeling of Leather Reflectance with Meso-Geometry

With the procedural model for leather meso-structure and the meaningful BTF interpolation based on the meso-structure constraint BTF synthesis we have almost all necessary ingredients to built a completely parameterized data-driven model of leather BTFs. What is left is the data itself and an effective compression algorithm that exploits the coherence between the different leather BTFs.

Our leather BTF database consists of 15 different real and imitated leather materials of which some are shown in the paper (the others will be made available later on through our project web-page). The measurements all have been cut to a spatial resolution of 128x128 texels. Since about 20000 HDR images (RGB) are captured per material the uncompressed storage size of this database is about 30 gigabytes. Using BTF compression the storage requirements for each material can be reduced to about 10 megabytes leaving us with 150 megabytes for the whole database which is already manageable. We can half these requirements without reducing visual quality by applying the compression to all materials simultaneously.

To make the compression of such a large dataset practical we used a multi-step approach. First we compute the PCA of each material and keep the first 150 eigenimages. This reduces the data to a size that fits in main memory. Then we perform an inter-material spatial clustering based on these eigenimages. We typically used about 250 clusters. Then the PCA of the original data in each cluster is computed (using online SVD [Bra03] if the data does not fit in main memory) keeping only 8-10 eigenvectors per cluster to allow for fast evaluation during run-time.

Now we can afford doing meaningful interpolation during editing. Beforehand we define a set of perceptual meaningful directions for the material class. In the case of our leather database we used parameters like roughness, glossiness etc. Then we assign each material a score along the particular trait vector. These scores then determine the interpolation weights for the respective measurements when the user chooses particular values for the different traits.

During editing, for each material in the database which has been assigned a weight larger than a small threshold, we perform the meso-structure constraint synthesis. This gives a set of spatial positions which are used to lookup and reconstruct the respective reflectance values from the compressed database. Then these are interpolated using the given weights. Figures 10 and 8 show examples of this user defined navigation within the database.

## 6. Conclusions and Future Work

In this paper we presented a method to edit measured BTFs in a meaningful and effective way. The main idea is to perform a kind of lazy factorization of the geometry and reflectance properties of the material and to use meso-structure constraint synthesis to recombine the measurements and the edited geometry. This avoids the need for simulating the whole complexity of light transport on the edited meso-structure. By using meso-structure constraint interpolation of ABRDFs we demonstrated data-driven editing of the reflectance of BTFs. We then proposed using procedural models to represent the meso-structure constraints for a specific material class in an intuitive way and demonstrated the advantages of this approach by means of a procedural crack pattern texture which allows, e.g., smooth morphs between different leather textures.

Directions for future research are numerous. We plan to investigate materials with volumetric meso-structure [MK06] and to build models for other interesting material classes like stone, wood or cloth where powerful procedural models are already available. In order to deal with materials consisting of several different substrates, e.g., with different colors, it will be fruitful to combine our technique with the BTF selection operators introduced in [KBD07]. It will also be interesting to compare our results with simulated data or even to incorporate simulated materials to increase the expressiveness of the material database without the need for additional expensive measurements.

**Figure 10:** *Editing the "dragons skin": Left: Measured BTF, middle: making the meso-structure more regular giving it a more spotty lizard-like look, right: increasing glossiness and shifting color makes the dragon looking a little bit more evil.*


## 7. Acknowledgments

Special thanks belong to our colleague Dirk Koch for performing most of the leather BTF measurements. The high dynamic range environments used in the rendered images are from Paul Debevec's website. We also thank the anonymous reviewers for their helpful comments.


## References


[Ash01] ASHIKHMIN M.: Synthesizing natural textures. In *Symposium on Interactive 3D Graphics* (2001), pp. 217–226.

[Bra03] BRAND M.: Fast online svd revisions for lightweight recommender systems. In *SIAM International Conference on Data Mining (SDM)* (May 2003).

[DvGNK97] DANA K. J., VAN GINNEKEN B., NAYAR S. K., KOENDERINK J. J.: Reflectance and texture of real-world surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition* (1997), pp. 151–157.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2* (1999), IEEE Computer Society, p. 1033.

[EMP*94] EBERT D., MUSGRAVE K., PEACHEY D., PERLIN K., WORLEY: *Texturing and Modeling: A Procedural Approach*. Academic Press, Oct. 1994. ISBN 0-12-228760-6.

[FC89] FRANKOT R. T., CHELLAPPA R.: A method for enforcing integrability in shape from shading algorithms. In *Shape from Shading*, Horn B. K. P., Brooks M. J., (Eds.). MIT Press, Cambridge, MA, 1989, pp. 89–122.

[FH03] FILIP J., HAINDL M.: Fast btf texture modelling. In *Proceedings Texture 2003* (2003), pp. 47–52.

[HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In *Proceedings of SIGGRAPH* (1995), ACM Press, pp. 229–238.

[HH99] HEAPS C., HANDEL S.: Similarity and features of natural textures. *Journal of Experimental Psychology: Human Perception and Performance*, 25 (1999), 1–24.

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *SIGGRAPH 2001, Computer Graphics Proceedings* (2001), Fiume E., (Ed.), ACM Press / ACM SIGGRAPH, pp. 327–340.

[HKL*99] HOFF III K. E., KEYSER J., LIN M., MANOCHA D., CULVER T.: Fast computation of generalized Voronoi diagrams using graphics hardware. *Computer Graphics 33*, Annual Conference Series (1999), 277–286.

[IO06] IBEN H. N., O'BRIEN J. F.: Generating surface crack patterns. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Sept 2006), pp. 177–185.

[KBD07] KAUTZ J., BOULOS S., DURAND F.: Interactive editing and modeling of bidirectional texture functions. In *Proceedings of SIGGRAPH* (August 2007), p. To appear.

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics 25*, 3 (Proc. SIGGRAPH 2006) (2006), 509–518.

[KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. *ACM Transactions on Graphics, SIGGRAPH 2005* (August 2005).

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2 (1955), 83–97.

[LBAD*06] LAWRENCE J., BEN-ARTZI A., DECORO

C., Matusik W., Pfister H., Ramamoorthi R., Rusinkiewicz S.: Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (July 2006).

[LH06] Lefebvre S., Hoppe H.: Appearance-space texture synthesis. *ACM Trans. Graph. 25*, 3 (2006), 541–548.

[LYS01] Liu X., Yu Y., Shum H.-Y.: Synthesizing bidirectional texture functions for real-world surfaces. In *Proceedings of SIGGRAPH* (2001), ACM Press, pp. 97–106.

[MK06] Magda S., Kriegman D.: Reconstruction of volumetric surface textures for real-time rendering. In *Proceedings of Eurographics Symposium on Rendering* (2006), pp. 19–29.

[MKC*06] Mertens T., Kautz J., Chen J., Bekaert P., Durand F.: Texture transfer using geometry correlation. In *Proceedings of Eurographics Symposium on Rendering* (2006).

[MMK04] Meseth J., Müller G., Klein R.: Reflectance field based real-time, high-quality rendering of bidirectional texture functions. *Computers and Graphics 28*, 1 (February 2004), 103–112.

[MMS*05] Müller G., Meseth J., Sattler M., Sarlette R., Klein R.: Acquisition, synthesis and rendering of bidirectional texture functions. *Computer Graphics Forum 24*, 1 (March 2005), 83–109.

[MPBM03] Matusik W., Pfister H., Brand M., McMillan L.: A data-driven reflectance model. *ACM Trans. Graph. 22*, 3 (2003), 759–769.

[MZD05] Matusik W., Zwicker M., Durand F.: Texture design using a simplicial complex of morphable textures. *ACM Trans. Graph. 24*, 3 (2005), 787–794.

[NDM05] Ngan A., Durand F., Matusik W.: Experimental analysis of brdf models. In *Proceedings of the Eurographics Symposium on Rendering* (2005), Eurographics Association, pp. 117–226.

[NZG05] Neubeck A., Zalesny A., Gool L. V.: 3d texture reconstruction from extensive btf data. In *Texture 2005 Workshop in conjunction with ICCV 2005* (October 2005), pp. 13–19.

[RM00] Roerdink, Meijster: The watershed transform: Definitions, algorithms and parallelization strategies. *FUNDINF: Fundamenta Informatica 41* (2000).

[RTG97] Rushmeier H. E., Taubin G., Guéziec A.: Applying shape from lighting variation to bump map capture. In *Proceedings of EGRW '97* (1997), Springer-Verlag, pp. 35–44.

[SG92] Sederberg T. W., Greenwood E.: A physically based approach to 2-D shape blending. In *Proceedings of SIGGRAPH* (1992), vol. 26, pp. 25–34.

[SSK03] Sattler M., Sarlette R., Klein R.: Efficient and realistic visualization of cloth. *Proceedings of the Eurographics Symposium on Rendering 2003* (2003).

[TCS06] Tongbo Chen M. G., Seidel H.-P.: Mesostructure from specularity. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), pp. 1825–1832.

[TZL*02] Tong X., Zhang J., Liu L., Wang X., Guo B., Shum H.-Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. In *Proceedings of SIGGRAPH* (2002), ACM Press, pp. 665–672.

[War92] Ward G. J.: Measuring and modeling anisotropic reflection. In *Proceedings of SIGGRAPH* (1992), ACM Press, pp. 265–272.

[WHON97] Wong T.-T., Heng P.-A., Or S.-H., Ng W.-Y.: Image-based rendering with controllable illumination. In *Proceedings of EGRW '97* (1997), Springer-Verlag, pp. 13–22.

[WL00] Wei L.-Y., Levoy M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 479–488.

[WMP*06] Weyrich T., Matusik W., Pfister H., Bickel B., Donner C., Tu C., McAndless J., Lee J., Ngan A., Jensen H. W., Gross M.: Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph. 25*, 3 (2006), 1013–1024.

[Wor96] Worley S.: A cellular texture basis function. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM Press, pp. 291–294.

[ZDW*05] Zhou K., Du P., Wang L., Matsushita Y., Shi J., Guo B., Shum H.-Y.: Decorating surfaces with bidirectional texture functions. *IEEE Transactions on Visualization and Computer Graphics 11*, 5 (2005), 519–528.