Two-Shot SVBRDF Capture for Stationary Materials

Miika Aittala
Aalto University

Tim Weyrich
University College London

Jaakko Lehtinen
Aalto University, NVIDIA

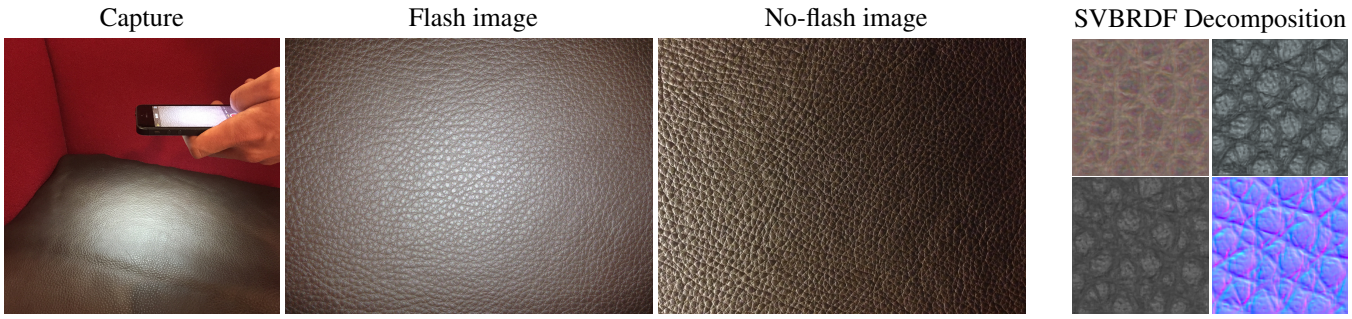


Figure 1: *Given an flash-no-flash image pair of a "textured" material sample, our system produces a set of spatially varying BRDF parameters (an SVBRDF, right) that can be used for relighting the surface. The capture (left) happens in-situ using a mobile phone.*

Abstract

Material appearance acquisition usually makes a trade-off between acquisition effort and richness of reflectance representation. In this paper, we instead aim for both a light-weight acquisition procedure and a rich reflectance representation simultaneously, by restricting ourselves to one, but very important, class of appearance phenomena: texture-like materials. While such materials' reflectance is generally spatially varying, they exhibit self-similarity in the sense that for any point on the texture there exist many others with similar reflectance properties. We show that the texturedness assumption allows reflectance capture using only two images of a planar sample, taken with and without a headlight flash. Our reconstruction pipeline starts with redistributing reflectance observations across the image, followed by a regularized texture statistics transfer and a non-linear optimization to fit a spatially-varying BRDF (SVBRDF) to the resulting data. The final result describes the material as spatially-varying, diffuse and specular, anisotropic reflectance over a detailed normal map. We validate the method by side-by-side and novel-view comparisons to photographs, comparing normal map resolution to sub-micron ground truth scans, as well as simulated results. Our method is robust enough to use handheld, JPEG-compressed photographs taken with a mobile phone camera and built-in flash.

CR Categories: I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Reflectance;

Keywords: appearance capture, reflectance, SVBRDF, texture synthesis

1 Introduction

Spatial variation in surface reflectance plays an immense role in the perceived realism of imagery. Traditionally, reflectance modeling has concentrated on the angular aspects of scattering (BSDF), building models that faithfully reproduce the shape and color of reflection and transmission from a single, uniform, material. However, subtle changes in reflectance, for instance those induced by wear and tear, scratches, and the like, are required to really bring a surface to life. Even simple angular reflectance models can produce results of impressive realism when their parameters are carefully varied across surface points. This observation has relatively recently led to the search for efficient and light-weight methods for capturing models that reconstruct such variation, as spatially-varying BRDF (SVBRDF), from real surfaces [Dong et al. 2011; Wang et al. 2011; Aittala et al. 2013].

In this paper, we describe a combination of acquisition setup design, simplifying material assumptions, and reconstruction algorithm that leads, we believe, to the lightest-yet setup for photographic capture of a full SVBRDF model including albedos, glossiness, and normals. This is made possible by assuming that the SVBRDF is part of a large and important class of "texture-like" materials that consist of elements or structures that (randomly) repeat over the surface, also known as *stationary materials*. This allows us to combine information across the material examplar.

Our input consists of two approximately fronto-parallel images of a relatively flat material sample, one taken in the ambient illumination, and another taken with a flash located close to the lens, that is, under headlight illumination (see Figure 1); in practice, we use a handheld Apple iPhone 5 and perform no calibration of any kind. The flash image provides local illumination whose relative direction of incidence varies across the surface, thus providing us with lighting-dependent samples of the texture's reflectance. The second image serves as a guide that allows us to identify similarities across the texture, under the more homogenous ambient illumination.

We present a multi-stage reconstruction pipeline that starts with gathering reflectance observations from across the image to augment information content at one representative image region, followed by structural regularization and subsequent texture statistics transfer, before we perform a non-linear optimization to fit a SVBRDF to the resulting data. From here, we propagate back this partial solution

from the reference region to the remainder of the input image. The final result is a photo-realistic SVBRDF that describes spatially-varying, diffuse and specular, anisotropic reflectance over a detailed normal map across the entire input image.

Many of our design decision are driven by the general goal of enabling photorealitic SVBRDF capture with a minimal acquisition effort. To maximize usability, we further decided to use consumer-grade mobile-phone photography. This choice introduces additional error sources, which is why we paid particular attention to robustness to errors in the data. This pays off in a reconstruction pipeline that, so far, has not failed in a significant way; failure cases are limited to materials that break our model assumptions, and even then, the method fails gracefully. We will show representative reconstruction results and provide, as supplemental material, results for the full set of any materials we acquired so far, to demonstrate the reliable robustness of the method.

2 Related Work

In principle, capturing a representation of surface reflectance of an arbitrary object requires densely sampling its high-dimensional reflectance function [Weyrich et al. 2009], which is, in its generality, a formidable task that requires extensive hardware and effort. In this section, we focus on various approaches to drastically reduce the acquisition complexity.

Reflectance Sample Fusion Many approaches reduce acquisition effort by combining reflectance measurements from multiple observations, in turn requiring fewer input images. Marschner [1998] showed that for a known (convex) object of constant BRDF, a single image contributes many BRDF samples, as each pixel observes a local surface frame under a different relative orientation to light source and camera. Lensch et al. [2003] and Goldman et al. [2005] carry this further by assuming that each surface point be a linear combination of very few unique basis BRDFs. These were important steps toward practical SVBRDF reconstruction, but still require tens of input images. Reflectance Sharing [Zickler et al. 2006] treats SVBRDF reconstruction as a scattered-data interpolation in the mixed spatio-angular domain. The method assumes spatial coherence of reflectance and fuses angular information across neighboring points on known geometry. While only requiring a single input image, the method trades spatial for angular resolution. Dong et al. [2010] combine separate, dense, 4D BRDF measurements of few representative points with spatially dense, angularly sparse, observations across a material sample. Wang et al. [2008] combine partial observations of microfacet distributions from different surface points based on similarity of their overlapping parts. All of the above methods rely on additional geometry or reflectance information — a requirement we wish to avoid.

Strong Model Assumptions Another avenue toward light-weight appearance acquisition is to restrict the range of reflectance phenomena supported. For instance, the intrinsic-image approach [Barrow and Tenenbaum 1978] aims at reconstructing diffuse albedo and normals for every pixel in a single input image. While great progress has been made in terms of stability and accuracy (e.g. [Barron and Malik 2015]), the Lambertian assumption makes these methods ill-suited for our purpose.

Glencross et al. [2008] use pairs of flash and no-flash images as input and use the additional information to drive a dark-is-deep heuristic to reconstruct plausible depth maps. While their input data are virtually identical to ours, their method is limited to Lambertian materials. CrazyBump and AppGen [Clark 2010; Dong et al. 2011] combine intrinsic-image methods with user edits to create plausible spatially varying reflectance models. In contrast, we do not require user input to specify lighting, material similarities, or specular parameters.

Texture Synthesis There is a rich literature on texture models and synthesis, a full review of which is beyond our scope. We combine the strengths of both statistics-based and exemplar-based synthesis algorithms by first employing ideas similar to "guided synthesis" for establishing a rough relighting, and then refining the result using a statistics-based approach [Heeger and Bergen 1995; Efros and Freeman 2001; Hertzmann et al. 2001].

Appearance Capture of Texture Interestingly, only few works directly exploit properties of textured surfaces for acquisition. Wang et al. [2011] take advantage of the notion of stationarity of the spatial structure to drastically simplify the acquisition of purely specular surfaces: from a single image of a the reflection of a step function off the surface, they reconstruct statistical properties of meso and microgeometry, which allows them to synthesise novel instances of the stochastic material. However, their approach can only handle a very narrow class of materials and spatial structures.

Ngan and Durand [2006] present a light-weight setup for BTF capture [Haindl and Filip 2013]. In their system, they use the Heeger-Bergen algorithm [1995] to transfer the statistics of one texture sample onto a low-quality version of another realization of the assumedly same texture. We borrow from that approach to transfer the detailed image statistics of a textural sample onto a coarse approximation of a similar texture.

Schröder et al. [2015] reconstruct the visual appearance of woven cloth by reverse-engineering its physical structure from a single image. They explicitly exploit the repetitive texture to fuse information across regions of similar structure. This has analogies to our structure-preserving reflectance transport; however, in contrast to our approach, their detection of recurring information is strictly limited to regular patterns and does not exploit variations in the local view and light vectors.

3 Designing an Acquisition System

Our goal is to maximize convenience of acquisition and yet obtain rich SVBRDF reconstructions. We identify the following desirables:

1. Light, mobile acquisition with a single, off-the-shelf device.
2. Convenient capture: few images and no need for calibration.
3. Under these constraints, unprecedented range of surfaces that can be captured.

We will now derive a system design that optimizes these constraints.

3.1 Hardware

Desirable 1. leads us to choose a mobile phone for acquisition, which offers an on-board flash for controlled illumination[1]. This choice has convenient properties. First, mobile-phone flashes are small enough to allow a point light assumption, which allows for a much simpler image formation model than that required by using an non-point-like emitter. Second, a phone's flash is mounted close to the lens, so that its effect approximates headlight illumination very well.

[1] An alternative would have been to use the phone's screen for illumination and to image using its front-facing camera, as proposed by Wang et al. [2011], but our tests led us to conclude the brightness of the screen is insufficient and leads to extremely low-quality, noisy images on current devices.

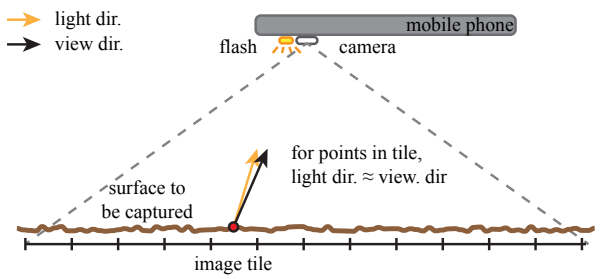


Figure 2: *The imaging setup. A camera with a flash attached near its center of projection (top) images an approximately flat material sample (bottom) in a roughly fronto-parallel projection. The pixels in input images are binned in constant-sized tiles. The surface points in each tile are lit and imaged from roughly the same direction because the flash is mounted close to the camera.*

Note, however, that the use of a stock mobile phone implies inferior data quality compared to, for instance, cameras with RAW support, which requires a robust reconstruction pipeline.

3.2 Acquisition Procedure

In order to obtain an undistorted image of the material sample (Desirable 2), we require the phone to be held roughly parallel to the material sample, see Figure 2. We take one image with the flash turned on (the *flash image*). This results in a range of viewing and lighting directions symmetric around the material sample's (macroscopic) normal, and due to the geometry of the imaging setup (Figure 2), each pixel in the flash image is an approximate retro-reflective measurement of the local reflectance function that combines the effect of surface normal and BRDF. Note that the imaging geometry remains identical regardless of the shooting distance; the range of observed angles is determined by the camera field of view (up to 33 degrees from the pole on iPhone 5). This also sets a rough upper limit on the width of the specular lobes we can reliably observe.

Even if the material is stationary, its appearance in the flash image may vary dramatically with angle of incidence, making it hard to reliably identify surface points with similar local reflectance. To aid reliable identification of points across the sample further on, we require a second image, taken from the same position, with the flash turned off, using only ambient illumination (the *guide image*).

Some representative flash and guide images are shown in Figure 3. We allow the images to be taken handheld (most of our results are), and register the two images before processing using a simple homography computed from manually specified point correspondences. The user also specifies the approximate characteristic size of the repeating texture with a few mouse clicks, and the image is then organised in regular tiles of approximately that size, as described in Section 4.1.

3.3 Material Assumptions

As our input is constrained to the retroreflective slice of the BRDF, we observe little information about Fresnel effects, or shadowing and masking. We hence choose a reflectance model that assumes typical behavior for these effects, while maintaining sufficient generality to fit to a wide range of observed microfacet distributions. In particular, we choose a microfacet BRDF model that includes both diffuse and specular reflectance, long-tailed reflectance lobes (kurtosis) and anisotropy in both meso- and microgeometry (Desirable 3). These effects typically show up well in retro-reflective point measurements, and their spatial variation is often the key defining characteristic

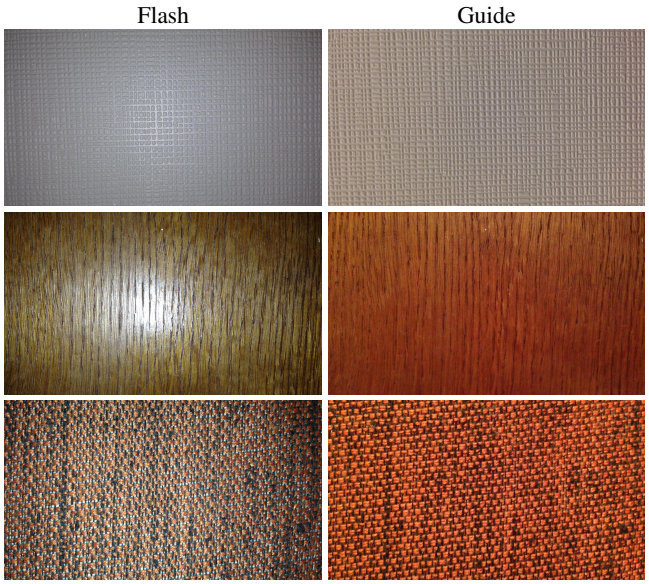


Figure 3: *Representative input data.*

of a given surface material. Full model details are described in the technical sections below.

Given that we have only one observation per pixel, we inevitably need to combine observations across multiple surface points (viewed and lit from different directions), of which we have to assume that they share similar reflectance properties. Luckily, most real-world spatially-varying materials exhibit a large degree of redundancy, that is, for each point on the material, there are others on the same material sample that exhibit identical, or very similar, surface properties (i.e., normal and BRDF). Such materials fall into the category of *stationary textures*, which cover the spectrum from repetitive patterns to stochastic textures. This assumption allows us to combine multiple observations into joint reflectance hypotheses. Specifically, we require that the material contains sufficient self-similarity so that within any large enough region, small neighborhoods of high similarity re-occur. Furthermore, we assume that the image can be divided to such regions by regular tiling, and that similar points can be identified by the images of their local neighborhoods under ambient illumination.

Lastly, we assume an approximately planar material sample, to eliminate the need for geometry reconstruction.

4 Reconstruction

Our goal is to reconstruct an independent parameterized reflectance function (normal plus anisotropic BRDF) at each pixel observed in the guide image from reflectance data in the flash image.

As our input data do not contain enough information to fit an SV-BRDF to all of the pixels directly (there is only one measurement per pixel!), we first concentrate information from across the flash image at a representative user-chosen *master tile* of the image, synthesizing additional reflectance samples at each of its pixel locations. This is tantamount to relighting the tile, that is, to creating novel versions of the master tile with altered illumination. Following the terminology of Lensch et al. [2006], it can also be seen as synthesizing *lumitexels* of reflectance samples for each pixel location on the master tile.

We create these lumitexels in a three-stage procedure. The first stage exploits that, according to our assumptions about the material's

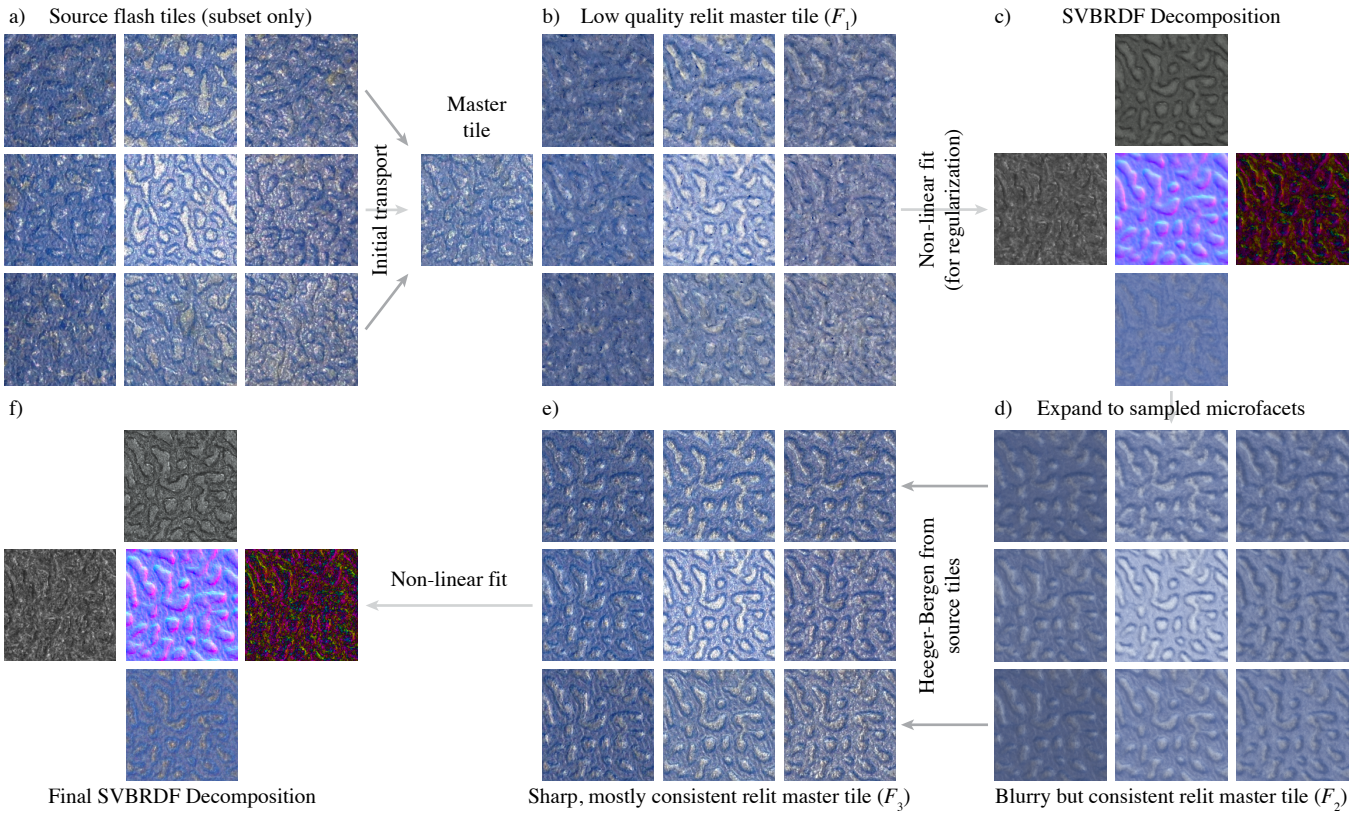


Figure 4: *Steps in the algorithm.* 3×3 *grids of tiles are representative of the full set of tiles as organised in the flash image (each corresponding to a unique half-way vector). Note that the choice of master tile is arbitrary; it does not even have to correspond to one of the input tiles.*

composition and the tile sizes, each tile contains roughly the same "material points", just rearranged in a potentially random manner. This means that, by identifying corresponding points between the master tile and each of the other tiles (not necessarily a one-to-one correspondence), we can share measurements across the image, i.e., relight the master tile by *transporting reflectance samples* from other, correspondingly lit, tiles (see Section 4.2.1). This already leads to coarsely accurate lumitexels for the master tile, but noise, jitter and other artifacts remain. We hence regularize the result using a preliminary SVBRDF fit, followed by recreating the lumitexels from the fitted SVBRDF, which removes most artifacts but introduces blur.

In the second stage, we refine the relit versions of the master tile by *transferring texture statistic* from other, similarly lit, regions in the original flash image (Section 4.2.2). Roughly speaking, this amounts to copying the high-frequency detail from the source tile onto similar-looking blurry structures in the master tile. As we will show, this results in high-quality lumitexels that are consistent in both spatial and angular domain and faithfully represent the master tile's appearance. Once the master tile has been augmented with lumitexels that way, we use a non-linear optimizer to fit an analytic SVBRDF model.

In the last stage, we then appropriately *reverse-propagate the solution* back to the full image using a guided texture transfer approach, yielding a full SVBRDF decomposition for all pixels in the input image (Section 4.2.3).

We will describe our SVBRDF model in more detail in Section 4.3; the fit by non-linear optimization, which plays a central role in keeping our reconstruction robust, is described in 4.4.

4.1 Data Organization and Flow

To simplify both data representation and problem formulation, we first divide the images up into a regular $n \times m$ grid of square tiles. We require each tile to be large enough to fully capture the texture's statistic and structural characteristics. In our experiments, we use a tile size of 192×192 pixels. As the frequency of textural repetition varies depending on the material and the shooting distance, we allow the user to specify the suitable number of tiles. We typically use around 12×8 tiles. The input image is scaled to match number of tiles and the tile size.

The tiling and imaging setup together suggest a natural data-driven representation of reflectance. As seen in Figure 2, each tile in the flash image sees the surface under a roughly constant light and view direction, i.e., at a unique half-vector. If we denote the guide image by $G(x,y)$ and the flash image by $F(x,y)$, where x and y are integer pixel indices, the tiling allows us to re-index all pixels in the input images as $G(i,j,s,t)$ and $F(i,j,s,t)$, where i,j identify a tile (and hence half-vector) and s,t identify the pixel within the tile. In this representation, each sub-image $F(i,j,\cdot,\cdot)$ provides, for all surface points imaged in the tile, a single retroreflective measurement of the local reflectance for that particular half-vector.

An example input dataset is illustrated in Figure 4a. The three-by-three image matrix shows a subset of the flash image tiles. It can be seen that although the tiles evidently consist of the same material, the spatial structures within the tiles are not in a similar arrangement. On the other hand, we can observe how the lighting condition is roughly equivalent within a tile.

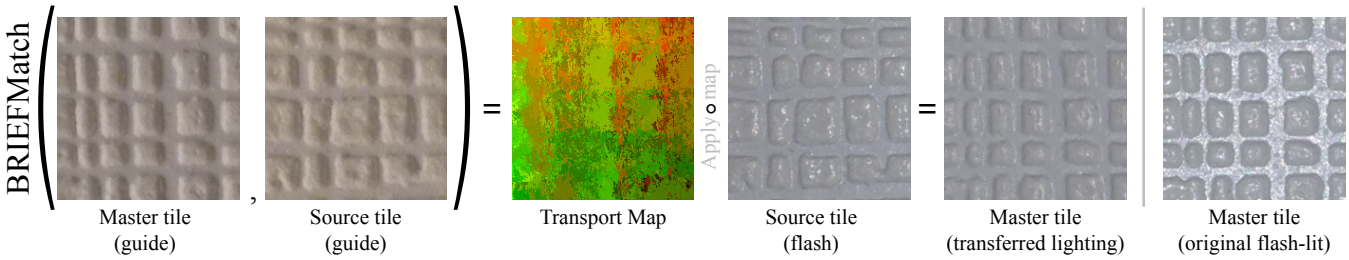


Figure 5: *Matching source tiles and the master tile. For each pixel in the master tile guide image (left), the best neighborhood match in the source tile guide image is computed by greedily selecting the pixel with the closest BRIEF descriptor distance. When taken over all pixels, this forms a transport mapping, visualized here by color-coding the pixel offset to the match in red and green channels. When the transport map is applied to the* flash *image of the source tile, the result is a an image that has the structure of the master tile, but illumination roughly matches that of the source flash tile. Despite some smoothing, the transferred appearance matches the source flash tile well, and the change in appearance compared to the original flash-lit master tile can be dramatic. The smoothing effect and how it is subsequently removed is visualized in Figures 4 and 6.*

Algorithm Steps Using this notation, the data flow and individual steps of our algorithm can be outlined as follows (cf. Figure 4).

1.1 Establish pixel-to-pixel correspondences between the master tile and every other image tile (i, j).

1.2 Using the correspondence, reshuffle each tile's pixels $F(i, j, \cdot, \cdot)$ to populate a new $F_1(i, j, \cdot, \cdot)$. Now all the lumitexels $F_1(\cdot, \cdot, s, t)$ are coarsely consistent with the master tile.

1.3 As a regularization, fit an SVBRDF to the data from the previous step. This results in a spatially and angularly consistent but blurry SVBRDF. Expand the result of the fit back into a sampled image $F_2(i, j, s, t)$.

2.1 For each tile (i, j), transfer the image statistics from the input photo $F(i, j, \cdot, \cdot)$ to $F_2(i, j, \cdot, \cdot)$, and denote the result by $F_3(i, j, \cdot, \cdot)$. This restores much of the "gist" of the original flash image tile into the relit master tile, while accurately maintaining its spatial structure.

2.2 Perform a final SVBRDF fit to the lumitexel array $F_3(i, j, s, t)$.

3 Optionally, either transport the SVBRDF parameters from the master tile back onto the entire guide image by effectively reversing the first step, or use texture synthesis to generate an arbitrary amount of new texture based on the master tile.

Figure 4 and Figure 6 illustrate the operations on an example dataset. The remainder of the section describes these steps in detail.

4.2 Method

4.2.1 Reflectance Sample Transport

In the first step, we put the pixels of the master tile in correspondence with the pixels of each input tile. This allows us to relight the master tile, thus synthesizing dense lumitexels at every point. To be successful, the measure of correspondence between points will have to consider the similarly of the surrounding neighborhoods, as matching single pixels by intensity alone would be meaningless. Because matching distant regions in the flash image is difficult due to significant differences in shading, we perform it in the guide image, which is mostly unaffected by lighting changes, and assume a strong structural match between two neighborhoods implies the points share a common reflectance function. In broad terms, this closely resembles guided texture synthesis [Hertzmann et al. 2001], but without considering neighborhood constraints in the target image.

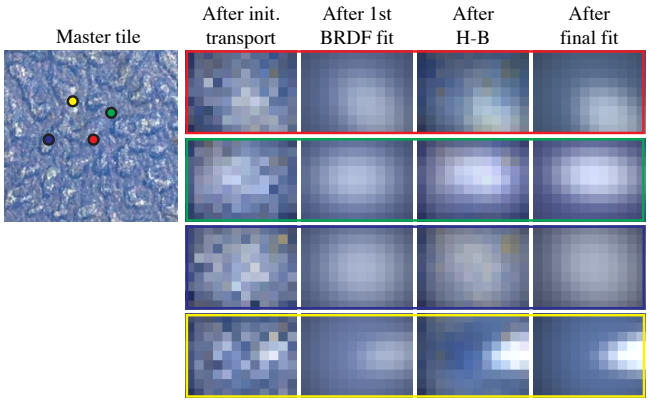


Figure 6: *Sampled lumitexel representation of reflectance functions for four representative points over the course of the algorithm. For each point, the reflectance data are represented as a sampling over microfacet orientations.*

A straightforward implementation would compare neighborhoods using squared pixelwise differences in a window. In our experiments, however, this approach turned out too brittle: small neighborhood window sizes were easily thrown off by noise, blur, color variation and random irregularities in texture, whereas larger windows tend to localize small details poorly, and struggle with non-rigidly distorted features.

Instead, we perform matching by comparing local feature (neighborhood) descriptors. We use the BRIEF descriptor [Calonder et al. 2010], for which details are given in Appendix A. It is insensitive to small differences in lighting and image quality, but respond sharply to different spatial arrangements of pixels. Furthermore, it is fast to compute and compare.

To further increase the quality of the relighting, we also copy image gradients in addition to the pixel values, and perform final reconstruction by solving a screened Poisson equation that balances between matching intensities and integrated gradients [Agarwala et al. 2004].

The entire process is illustrated in Figure 5. Because the matches are computed for all master tile pixels independently, this problem parallelizes perfectly. Conretely, denoting the descriptor by BRIEF, we loop over all tiles i, j, and for each pixel s, t in the master tile, seek the pixel i, j, s', t' whose descriptor $\mathrm{BRIEF}(i, j, s', t')$ best matches that of the master pixel $\mathrm{BRIEF}(s, t)$. Once found, we copy the intensity from the source flash tile into a new array to the corresponding position, i.e., $F_1(i, j, s, t) \leftarrow F(i, j, s', t')$.

The result of the matching is a set of coarsely-relit master tiles, denoted $F_1(i,j,s,t)$. This is visualized in Figure 4b. As can be seen upon close inspection, the tiles in the result all have the same spatial structure — that of the master tile — and a lighting roughly in line with the source tiles. However, perfect matches between the neighborhoods cannot be expected, not the least due to the fixed pixel grid, and artifacts remain everywhere due to imperfect matching. The combined effect of such error sources produces spatial and angular jitter. The angular nature of the error is visualized in Figure 6.

Regularization by SVBRDF Fit After the transport, we regularize the relighting result by fitting an SVBRDF to the enriched master tile, using a non-linear optimizer that yields spatially consistent decompositions (Section 4.4). The spatio-angular jitter present in the data blurs the SVBRDF both spatially and angularly, suppressing fine details (Figure 4d); however, we observe that the resulting model is structurally very consistent, that is, normals correlate well with the master tile's mesostructure and reflectance lobes capture the relevant trends. This is illustrated in Figure 6.

Before further processing, we convert back the SVBRDF to the original sampled representation of multiple relit master tiles and denote the result by $F_2(i,j,s,t)$.

4.2.2 Texture Statistics Transfer

In order to further improve the relighting result, faithfully matching the target appearance, we borrow from Ngan and Durand [2006], who, too, acquire lighting- and viewing-dependent images of (literal!) material tiles. Drawing on their richer data set, they employ linear interpolation between lighting directions to approximately relight the master tile. To mitigate the resulting ghosting artifacts, they use *texture statistics transfer* to modify the (approximately) relit master tile to exhibit texture statistics that they interpolated from other tile's statistics, using Heeger and Bergen's histograms-of-steerable-pyramids approach [1995].

We follow a very similar approach, by seeding Heeger-Bergen texture synthesis [1995] with our coarsely relit master tile, letting the algorithm iterate over that tile until its histograms of multi-scale steerable-filter responses match the respective histograms of the target appearance. In order to maintain coherence across color channels, we use a method by Rabin et al. [2011], who propose a computationally efficient simplification of the Wasserstein optimal transport metric and use it to formulate an efficient multichannel version of the Heeger-Bergen algorithm. Concretely, we run, for each half vector i,j, this multi-channel Heeger-Bergen algorithm that uses the $F_2(i,j,\cdot,\cdot)$ as the seed, and matches its pyramid statistics with the corresponding tile in the input flash image $F(i,j,\cdot,\cdot)$. The result is denoted $F_3(i,j,\cdot,\cdot)$.

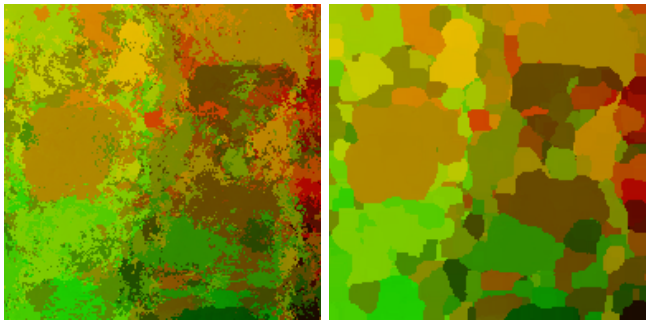
In theory, Heeger-Bergen histogram matching does not guarantee to stay close to its seed image; however, for seeds whose histograms are sufficiently close to the target statistics, the histogram-matched output preserves the input structure. Because, furthermore, our seeds for the different lighting conditions all draw from the same, slightly blurry but spatially and angularly consistent, SVBRDF, the different independent texture syntheses maintain mutual consistency. Accordingly, the resulting per-point reflectance samples exhibit much less jitter. In particular, the texture synthesis "recovers" characteristic high-frequency details while preserving spatial structure, see Figure 6. The result is shown in Figure 4e: the dull appearance apparent in Figure 4d has been transformed into a new, crisp one that matches the feel of the input flash tiles (Figure 4a) very well, while maintaining the spatial structure of the master tile.

Together, this allows us to fit an SVBRDF, for the second time, which now leads to a crisp reconstruction consistent with the master tile. Our goal of obtaining an SVBRDF decomposition of the master tile is now achieved. (In the example in Figure 4, the resulting relit tiles are hardly distinguishable from the Heeger-Bergen results and not shown.)

4.2.3 Reverse Transport

At this stage, we have resolved the material parameters for all points in the master tile. To recover the parameters for the rest of the surface, we essentially perform the reflectance sample transport in reverse direction, this time transporting the parameter values from the master tile to the other tiles. Note that we cannot use the transport maps obtained in the first stage of the algorithm because they are not one-to-one, but must recompute them in the opposite direction.

As the resulting transport map may be somewhat noisy and irregular, we first apply a simple windowed median-like filter to increase coherence and clear up the boundaries between the copied regions. Each pixel in the transport map is a 2D shift vector. We first compute, for each pixel, the sum of its ℓ_1 distances to all pixels in a 7×7 sliding window. In a second pass, we replace each pixel with the one from its neighborhood that has smallest summed distance stored in the first pass. An example is shown in Figure 7. The resulting map copies consistent patches with clear boundaries. Like in the forward transport pass, we perform the transport in the gradient domain to ensure better continuity.



Transport map, initial | Transport map, filtered

Figure 7: *The transport mapping used in initial transport (left) is produced by the nearest-neighbor BRIEF matching. For final backpropagation of the SVBRDF parameters onto the entire source image, we use a filtered map (right) that yields more consistent results by keeping spatial structure more intact.*

Alternatively, we can produce an arbitrary amount of the texture by using the master tile as an exemplar for a classical texture synthesis method, such as Image Quilting of Efros and Freeman [2001]. While this is straightforward to do, we leave this outside our scope.

4.3 Material Model

Our BRDF model is inspired by "BRDF Model A" of Brady et al. [2014]. Their model is simple but shown to fit accurately to a wide range of measured BRDFs. It also contains a kurtosis-like parameter that controls the pointiness of the specular lobe – an effect that is often clearly visible in our input data. We extend it with a simple anisotropy model.

Specifically, our model consists of the parameters

$\rho_d \in \mathbb{R}^3_+$	diffuse albedo (RGB)
$\rho_s \in \mathbb{R}^3_+$	specular albedo (RGB)

$\mathbf{S} \in \mathbb{SPD}^2$	specular glossiness and anisotropy
$\alpha \in \mathbb{R}_+$	specular pointiness
$\mathbf{n} \in \Omega_+$	surface normal unit vector

where Ω_+ is the upper unit hemisphere and $\mathbb{SPD}^2$ is the set of 2×2 symmetric positive definite matrices. In addition, we use a global (not spatially varying) parameter $\sigma_f \in \mathbb{R}_+$ to model vignetting effects due to camera and light. The parameters, with the exception of the glossiness and anisotropy $\mathbf{S}$ and specular pointiness α, are entirely standard. Let $\mathbf{h}$ be the tangent plane parameterized version of the half-vector. We define our microfacet distribution (NDF) as

$$D(\mathbf{h}) = \exp\left[-(\mathbf{h}^\top \mathbf{S}\mathbf{h})^{\alpha/2}\right]. \quad (1)$$

Notice that if $\mathbf{S}$ is a multiple of the identity matrix, this reduces to a tangent-plane parameterized version of the NDF of "BRDF Model A" of Brady et al. [2014]. However, with $\mathbf{S}$ a general SPD matrix, the NDF is stretched, and the BRDF becomes anisotropic. This parameterization of anisotropy has the advantage of being unique: for every stretch (linear transformation) of the lobe, there is exactly one SPD matrix, and vice versa. This makes it well-behaved in optimization and interpolation.

Note also that shadowing/masking effects of "BRDF Model A" vanish at backscattering angles, and the Fresnel term reduces to constant, leaving the NDF as the only term in the specular component. Our measurements carry no information about these effects. Hence, during fitting we evaluate the BRDF model as a sum of this NDF and a constant diffuse term, weighted by the respective albedos. The rendered value is modulated by the inverse square distance, cosine foreshortening, and a Gaussian vignetting term of width σ_f. Appendix B provides details on coordinate systems and other details necessary for evaluating the full image formation model during optimization.

4.4 Fitting SVBRDFs by Optimization

At two stages in our algorithm, we are presented with the problem of robustly fitting a parametric SVBRDF model to given input data represented as sampled lumitexels. To encourage spatially consistent solutions, we perform the data fitting jointly between all pixels and use priors that bind the solutions of neighboring pixels together, much like Aittala et al. [2013]. Like them, we use the Levenberg-Marquardt algorithm for jointly minimizing the data fit error and the prior penalties.

Even though conceptually simple, the nature of our input data makes the fitting task very difficult. The input contains

- unknown vignetting from both non-uniform flash emission and the sensor, making the observations at the sides of the image highly unreliable;
- unknown non-linear color correction and contrast enhancement, and possibly other image processing operations, distorting the observed profiles of the reflectance lobes and modifying their relative intensities;
- overexposure (clipping), i.e., no precise relative brightness information for some pixels.

This necessitates priors other than simple spatial smoothness. On the highest level, our data fitting procedure aims to decompose the input into a sum of very wide diffuse lobe and a relatively narrow specular peak while tolerating the strong distortions present. Our reflectance model is co-designed with priors that encourage plausible separations; for instance, we favor a diffuse explanation to a wide specular one when the two are ambiguous. We also use robust error metrics (the Huber loss function) to downweight the effect of outlier values in the data. Complete details are given in Appendix C.

5 Results

5.1 Acquisition

In all of our experiments, we used an iPhone 5 camera with the default camera app (iOS 7.1.1). Having been a reasonably high-end camera two years ago, we believe it to be very representative of a generic mobile phone camera today. It features an 8 mega-pixel sensor and a white LED flash, 9.45 mm off the camera lens. We work directly on JPEGs shot with the built-in interface. The only image processing we apply from our end is inverse gamma correction (exponent 2.2) of the 8-bit JPEG images after reading; note that this still does no guarantee truly linear data. We have no control over white balance and ISO, and all images are selectively, automatically denoised (especially under flash conditions) and sharpened [DPReview 2012].

All of our results were obtained from either hand-held photographs or by supporting the phone on a level surface. No tripods or other mechanical supports were used. Thus we were able to casually capture a variety of materials *in situ* without any accessibiliy constraints beyond having to reach out to the material (see Figure 1). In particular, little care was taken to control the illumination in the environment, aside from avoiding strong ambient lighting and direct shadows. This is apparent in the input data (included in supplemental material).

5.2 SVBRDF Reconstructions

We captured 72 flash/no-flash images for a wide range of materials. Here, we present and discuss a representative subset of our results; animated renderings of the complete set of our reconstructions are provided as supplemental material. Figure 8 shows a subset.

Overview Our inputs feature both metals and dielectrics. The broad categories include metal, wood, plastic, papers, fabrics, and leathers. Many materials feature significant anisotropy (e.g. *fabric_orange*, *fabric_chair*), strong, sharp spatial variations (e.g. *wood_dark*), and stretch our BRDF+normals assumption due to significant interreflection and volumetric scattering (e.g. *plastic_cutting* and, again, fabrics). Apart from doublets, the results include every capture we have ever performed, and include materials that strongly violate our assumptions (e.g. *seed*).

Despite the above, we observe that our method is stable, and produces excellent to reasonable results for most inputs. Overall, we feel the results are photorealistic and convey the look and feel of most input materials faithfully. We find this surprising given the low quality and limited amount of input data. The quality of the input data varies greatly across the datasets; we invite the reader to examine the unprocessed input images and the videos in the supplemental material.

Analysis The results feature plausible spatial variation in all parameters. Even though difficult in our imaging setup, diffuse-specular separation often succeeds well (cf. *fabric_orange*, *fabric_chair*), although sometimes crosstalk is observed (*paint_black*). At times, extremely fine spatial detail may suffer. For instance, the photographs of *book_brown* show a tiny regular structure on top of the otherwise stochastic material, which is mostly lost in the reconstruction. Despite this, the resulting decomposition is reasonable. Some inputs, such as lacquered wood (e.g. *wood_door*), feature

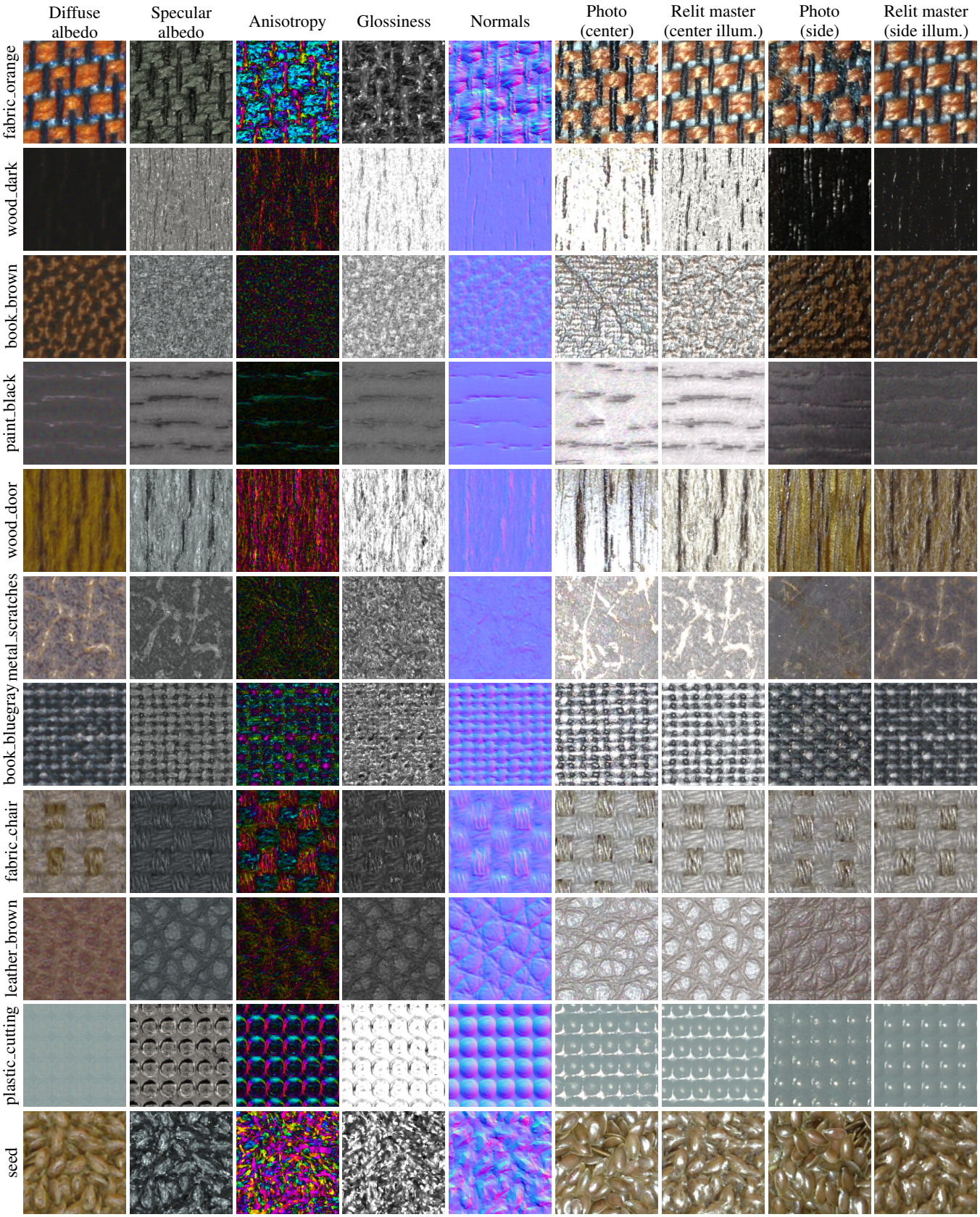


Figure 8: *Model fits and relit master tile for head-on and side views. All images, including many more results, can be found in the supplemental material. The glossiness-anisotropy matrix* $\mathbf{S}$ *is symmetric positive definite, and hence uniquely encodes an anisotropic scaling. We visualize the direction and strength of the anisotropy as hue and intensity of the anisotropy map, and the overall scaling factor as the glossiness map. Higher values indicate a glossier material.*

strong dual-lobed reflectance with a sharp specular on top of the anisotropic reflection from the polished wood fibers. This subtlety is lost in our model, but the resulting rendering still conveys much of the feel of the material.

We chose to keep anisotropy enabled for all datasets regardless of whether they seem to exhibit it.

The optimizer often introduces small amounts of anisotropy even for isotropic materials. Typically this occurs in regions of rapid normal variations or occlusions, where no unique normal direction sufficiently characterizes a single pixel, and the sub-pixel detail is absorbed into the microgeometry as anisotropy. This behavior appears neutral or beneficial. Note that these effects are also responsible for "true" anisotropy, as can be seen in Figure 11: clearly oriented microgeometry visible in the ground truth scan is appropriately captured in our anisotropy maps.

Our algorithm is, by design, robust to small-to-moderate violations of the stationarity assumption. With increasing non-stationarity, the results gradually lose their expressive power, but we have not observed badly unstable results. Figure 9 below shows an example (*metal_gritty*) whose structure has larger-scale variations than those captured in the tiles. The result is reasonable, but does not generalize well to regions that are clearly different from the rest. This behavior is typical for our results.

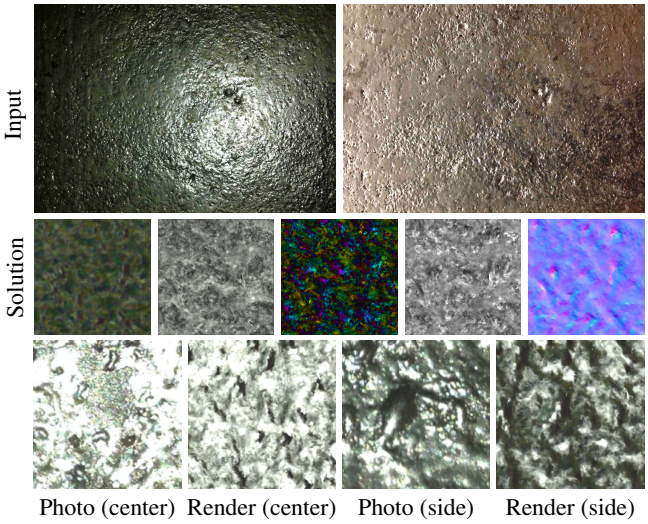


Figure 9: *In some of our inputs, the stationarity assumption ("tiles feature same material points") is not satisfied at the scale imaged. The overall structure in* metal_gritty *is roughly homogeneous, but there is significant low-frequency variation in smoothness and bumpiness. The solution tolerates this and captures the average appearance, but cannot reproduce the nonstationary parts (bottom row).*

Failures We have categorized some datasets as clear failure cases in the supplemental material videos. Typically they strongly violate our assumptions. The violations include significantly non-planar geometry, interreflections, and non-stationarity at the scale of the tiling. The *seed* set features all of these. While the decomposition may look reasonable at first sight, it contains a strong specular "film" on top of the entire tile. Furthermore, the normals, while somewhat resembling the seeds, are not captured well, and the relighting result (cf. video) is not faithful.

In practice, our method is rather robust against shading changes in the guide image caused by non-uniform illumination. However, occasionally the illumination in the guide image is non-uniform enough to significantly change the appearance of the exemplar and throw the transport step off. This results in visibly dull, not very faithful reconstructions (e.g. *leather_black*, cf. videos), or rough spatial variations (e.g. *plastic_weave_silver*). We encourage examination of the flash-guide pairs in the supplemental material.

Implementation All results have been computed using the parameters given in Appendix C. No per-input tweaking of prior weights etc. has been performed. Our implementation combines a CUDA implementation of the transport step with unoptimized and non-parallelized Matlab implementations of the Levenberg-Marquardt fitter and Heeger-Bergen synthesis. Processing takes roughly three hours per dataset on a fast quad-core desktop PC equipped with an NVIDIA Quadro K6000 GPU.

5.3 Verification

Side-by-side Figure 10 shows side-by-side comparisons between the input flash images, relit flash images, and flash images taken under novel view and novel illumination. We observe qualitatively good matches between the inputs from the capture view. At times, the relative strengths of the specular and diffuse components have been misidentified by the optimizer due to clipping, vignetting, and so on. While the original view may yield a faithful match, a less pointy, broader specular lobe may become apparent when observed under novel angles (e.g. the *tape_silver2* set). In such cases, simple global adjustments of the resulting material parameters (reweighing diffuse vs. specular and adjusting the specular pointiness) often result in a good match (rightmost column).

Ground Truth for Normal Maps We obtained submicron-accurate heightmap scans from GelSight, Inc. for a set of input materials [Johnson et al. 2011]. We converted the height maps to normal maps, and compare to our reconstructions in Figure 11. As can be observed, our method captures the normals generally well.

The *plastic_cutting* set features highly glossy bumps. The bumps' sides are so steep that the specular reflection off them is not entirely captured in the flash photo, i.e., our input contains no information about a part of the bumps; furthermore, the sample includes interreflections and subsurface scattering not consistent with the model. Our algorithm fails gracefully and produces a flat normal field for the outer rim of the bumps. However, most of the bumps *are* recovered, and the resulting relightings are rather faithful. The physical scale of the bumps is approx. 1 mm.

The *fabric_zigzag* set demonstrates capture of anisotropy due to the strongly oriented microgeometry in the weaves (visible in the inset). While our input definitely does not have sufficient resolution to capture this behavior in the normal map, the resulting anisotropic reflectance is captured well by our per-point anisotropy term (bottom right).

Stability Analysis by Synthetic Input To test the stability of the reconstruction algorithm, we simulated different classes of error sources. We first generated a hypothetical SVBRDF decomposition (Figure 12, left column) in a paint program. We then rendered simulated flash and no-flash images from the data, applying various corruptions, and running the decomposition algorithm on the inputs to study their effect on the result.

Given an HDR flash-no-flash pair rendered using the ground truth SVBRDF, the reconstruction algorithm produces the result in the second column. The top rows show crops of the flash and guide images used as an input to the algorithm. Although not quite identical, the output is faithful to ground truth. A slight amount of crosstalk

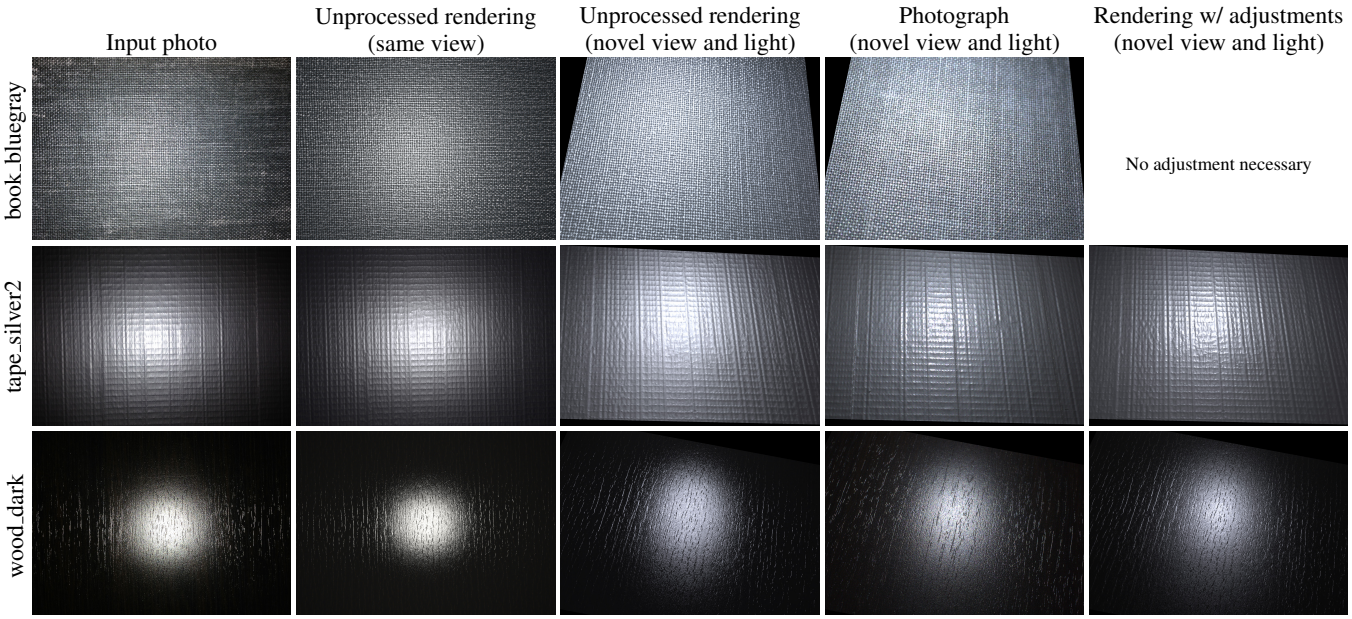


Figure 10: *Relighting vs. photograph for novel view and illumination. See text for description.*

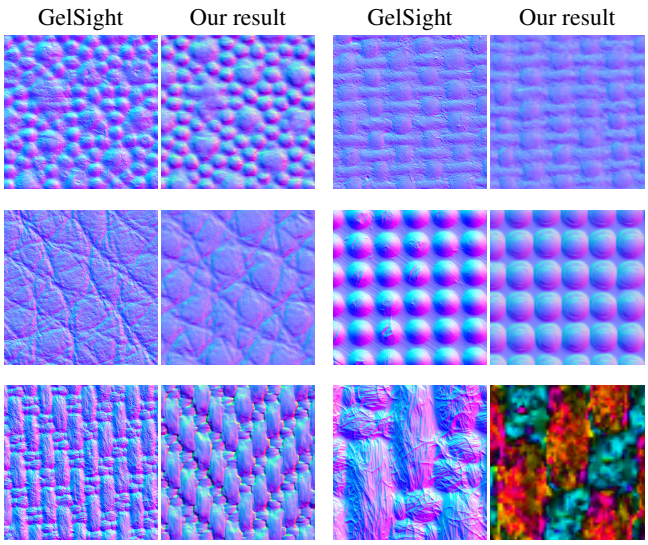


Figure 11: *Comparison of normal recovery to submicron-accurate GelSight™ scans for* book_black, book_wine, leather_brown, plastic_cutting, *and* fabric_zigzag. *The pitch of the bumps in* plastic_cutting *is about 1 mm. See text for analysis of* plastic_cutting. *Bottom row: The fibers in* fabric_zigzag *are strongly oriented alternatively in 90 degree angles at different points in the weave (blow-up of GelSight scan, 2nd from right). While our resolution is insufficient to reproduce this microscopic detail in the normal map, the resulting large-scale anisotropic reflectance is captured by the spatially varying anisotropy map (right). Cf. caption of Figure 8 for a description of the anisotropy visualization.*

is observed between diffuse and specular, and some anisotropy has been introduced.

The third column shows a reconstruction from a color-processed noisy version of the original input. The input images have been processed with a non-linear contrast-boosting S-curve, typical to

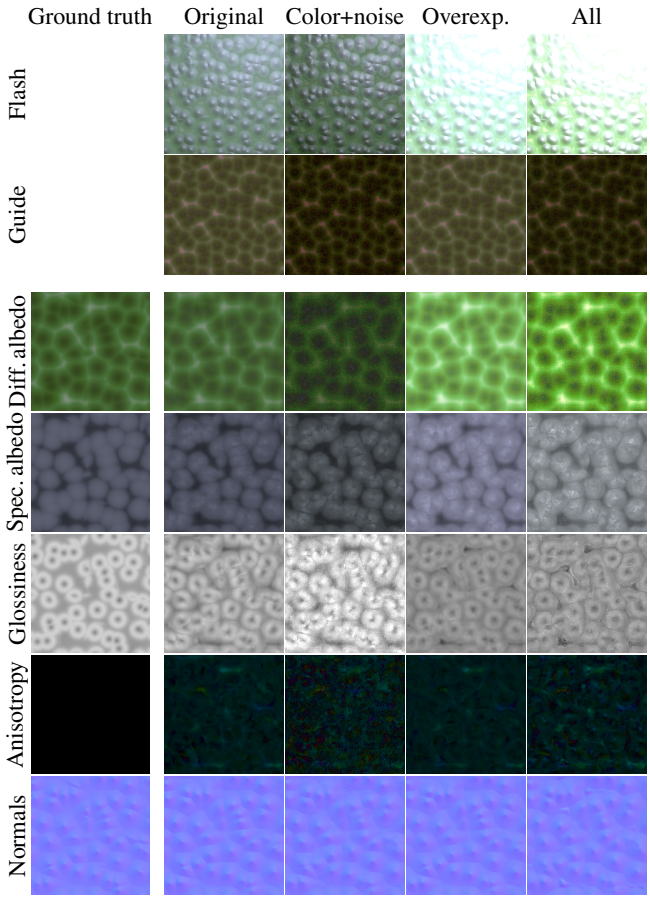


Figure 12: *Resiliency to input corruption.* Top two rows: *close-ups of synthetic input data. The images were rendered from ground truth SVBRDF parameters* (leftmost column). *Input renderings are shown as-is, with color processing and added noise, with overexposure, and with all corruptions applied at once (including also near-field illumination, ambient lighting and misalignment, not shown separately).* Bottom five rows: *SVBRDF decompositions computed using different versions of the inputs. The results show the solver is stable.*

camera processing pipelines. Besides distorting albedo levels, the camera's internal processing tightens the specular highlight somewhat, leading to higher glossiness in solution. Nevertheless, the overall visual appearance and spatial structure of the material is preserved well.

The fourth column shows the effect of severe overexposure of the flash image, which sometimes occurs in LDR photos of glossy materials. We find that the reconstruction tolerates the partially missing data without introducing notable artifacts, and absorbs the higher intensity into albedos.

Finally, the rightmost column shows the combined effect of these and several other distortions: spatially mismatched flash and no-flash images due to slight offset and zooming, ambient light in the flash photo, and near-field illumination in the guide photo. While naturally adapting to the changed inputs, the reconstruction still resembles the ground truth well, and we conclude the optimizer is relatively robust. The full set of input images and rendered results of these experiments are included in the supplemental material (*synth_**).

Miscellaneous Experiments. We have tested the method with linear RAW image input from a Canon 5D Mk III SLR camera. Figure 13 shows a comparison between iPhone and DSLR results for two datasets. The DSLR results are somewhat cleaner and sharper, and show different albedo levels due to lack of color processing. Both effects are expected. Otherwise, the results agree well. We also shot the same dataset from two distances, near and far, and found that the solutions match well, resolution notwithstanding. Please see the supplemental material for results on this experiment.

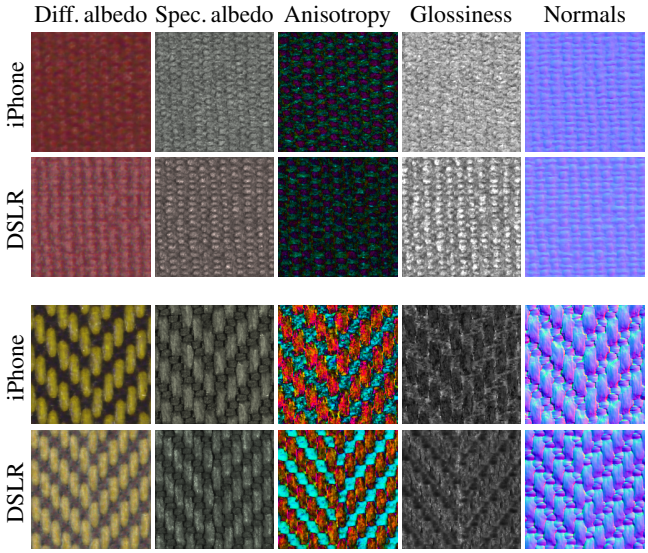


Figure 13: *Comparison to raw DSLR input data. Top rows show two reconstructions for* book_wine*, shot with iPhone 5 and Canon EOS 5D Mk III. Bottom rows show similar results for* fabric_zigzag.

6 Conclusion

We have described an algorithm that is able to recover rich models of spatially-varying reflectance (SVBRDF) from flash-no-flash image pairs for a large class materials that are spatially stationary (self-similar) in a loosely defined sense. The method is demonstrated on dozens of real-world data sets, with input photos taken using an off-the-shelf mobile phone. The results are often excellent, generally good, and remain mostly reasonable even when our model assumptions are violated. We find this surprising given the *ad hoc*, uncalibrated and low-dynamic-range nature of our setup. We believe we offer the lightest-yet method for capturing interesting material models directly suitable for content creation.

Acknowledgments

This work was supported by the Academy of Finland (grant 277833), the Helsinki Doctoral Programme in Computer Science (HeCSE), and the UK Engineering and Physical Sciences Research Council (grant EP/K023578/1).

A Feature Descriptor

We use BRIEF features [Calonder et al. 2010] to determine similarity between points. To consider similarity on multiple scales, we concatenate together three BRIEF G II style descriptors with window sizes S of 33, 17 and 5 and Gaussian blurs of standard deviations of 4, 2 and 0. We compute 96, 128 and 32 bits of these descriptors, respectively. This is repeated for all color channels, resulting in a 768-bit descriptor for each pixel.

The feature distance is the Hamming distance of the descriptor strings, normalized by 768. On top of this, we add the absolute values of pairwise pixel color differences, multiplied by 0.01. This process defines the descriptor distance $\|B(i,j,s,t) - B(i',j',s',t')\|$ from Section 4.2.1.

Before computing the descriptors for each pixel, we first normalize the guide image by subtracting from it a strongly Gaussian-blurred version of itself, so that any sufficiently large region will have approximately zero mean. We then pointwise-divide the resulting image by the square root of a blurred squared image, so that any large-enough region will have unit variance.

B Image Formation Model

This section defines the coordinate systems used in capture, and how precisely the BRDF model is evaluated. Set the coordinate system as follows. The planar sample is lying on the XY-plane with its normal pointing towards positive Z-direction. As the global scale of the geometry is inconsequential, the camera is without loss of generality assumed to face the sample perpendicularly at position $\mathbf{E} = [0\ 0\ 1]^\top$, and the point light source is assumed to be exactly coincident with the camera. In practice, as the flash highlight might not be exactly centered in the input photo, we place the origin at the centroid of the 5% of the brightest pixels in the image, and establish the plane coordinates from the known field of view of the camera. At a given sample plane position $\mathbf{p} \in \mathbb{R}^2$ (and the interchangeable 3D position $\mathbf{P} \in \mathbb{R}^3$), the rendered pixel intensity is determined as follows.

To be able to flexibly specify the NDF (in particular, its stretching due to anisotropy), let us transform the half-vector direction into the local normal-centered coordinate system. We do this by using the Rodriguez rotation formula to find the shortest-path rotation matrix $\mathbf{R}$ that takes $\mathbf{n}$ to $[0\ 0\ 1]^\top$. Then the half-vector in NDF coordinates is $\tilde{\mathbf{h}} := \mathbf{R}(\mathbf{E}-\mathbf{P})/\|\mathbf{E}-\mathbf{P}\|$. Finally, let $\mathbf{h} := \tilde{\mathbf{h}}_{xy}/\tilde{\mathbf{h}}_z$ be the tangent plane parameterized version of $\tilde{\mathbf{h}}$.

The final pixel color value is now computed as

$$v(\mathbf{p})I(\mathbf{P})c(\mathbf{P})\left(\rho_d + \rho_s D(\mathbf{h})\right),$$

where $c(\mathbf{P}) := \max\left(0, \mathbf{n}^\top \frac{\mathbf{E}-\mathbf{P}}{\|\mathbf{E}-\mathbf{P}\|}\right)$ is the cosine foreshortening term, $I(\mathbf{P}) := \|\mathbf{E}-\mathbf{P}\|^{-2}$ is the inverse square distance attenuation, and the NDF $D(\mathbf{h})$ is defined as in Equation 1. We also include a simple vignetting term $v(\mathbf{p}) := \exp(-\|\mathbf{p}\|^2/\sigma_f^2)$ that partially models the

uncontrolled darkening of the image at edges by a Gaussian of standard deviation σ_f.

C Fitting the SVBRDF

The Levenberg-Marquardt algorithm takes in a function $\mathbf{R} : \mathbb{R}^K \mapsto \mathbb{R}^L$ that maps K parameters (unknowns) to L residuals, and finds a parameter vector $\mathbf{X}^* \in \mathbb{R}^K$ that locally minimizes the sum of squared residuals $\|\mathbf{R}(\mathbf{X})\|^2$. Let us build a residual function that describes our desired solution. Although in practice the L-M algorithm requires that we supply the non-squared vector $\mathbf{R}(\mathbf{X})$, for the purposes of exposition we describe here the sum of squared residuals that we aim to minimize. Each of the summed partial residuals addesses a specific concern; at the highest level, let $\|\mathbf{R}(\mathbf{X})\|^2 = \|\mathbf{R}_{\text{fit}}(\mathbf{X})\|^2 + \|\mathbf{R}_{\text{priors}}(\mathbf{X})\|^2$.

Unknown variables. In our problem, the vector $\mathbf{X}$ of unknowns contains the per-pixel parameter values $\mathbf{x}_{(s,t)}$ for the BRDF model (9 per pixel), and four global unknown parameters $\mathbf{g}$ that do not vary per pixel. Most of the variables discussed in Section 4.3 are constrained (e.g. to positive values), and we also sometimes wish to limit the range of values they can assume. However, explicitly constrained nonlinear optimization is in general difficult. Instead, we choose to reparameterize our optimization variables in a way that any choice of real numbers will yield a feasible set of parameters. Our optimization variables, their mappings to the actual model parameters, and the initial guesses for optimization are as follows:

variable	mapping to parameters	initial
$\tilde{\rho}_d \in \mathbb{R}^3$	$\rho_d = \exp(\tilde{\rho}_d)$	
$\tilde{\rho}_s \in \mathbb{R}^3$	$\rho_s = \exp(\tilde{\rho}_{s,1})\mathbf{C}_{\text{YUV}}[1\ \tilde{\rho}_{s,2}\ \tilde{\rho}_{s,3}]^\top$	[-1 0 0]$^\top$
$\mathbf{s} \in \mathbb{R}^3$	$\mathbf{S} = \left[\exp\begin{pmatrix}\mathbf{s}_1 + 10^{-3} & \mathbf{s}_3 \\ \mathbf{s}_3 & \mathbf{s}_2 + 10^{-3}\end{pmatrix}\right]^{-1}$	[-3 -3 0]$^\top$
$\tilde{\mathbf{n}} \in \mathbb{R}^2$	$\mathbf{n} = [\tilde{\mathbf{n}}; 1]/\sqrt{\|\tilde{\mathbf{n}}\|^2 + 1}$	[0 0]$^\top$
$\tilde{\sigma}_f \in \mathbb{R}$	$\sigma_f = \exp(\tilde{\sigma}_f) + 0.3$	-1
$\tilde{\alpha} \in \mathbb{R}$	$\alpha = \exp(\tilde{\alpha}) + 0.5$	0.4

Of these variables, $\tilde{\rho}_{s,2}$, $\tilde{\rho}_{s,3}$, $\tilde{\sigma}_f$ and $\tilde{\alpha}$ are global, and rest are per-pixel. $\mathbf{C}_{\text{YUV}}$ is the YUV to RGB conversion matrix. The initial guess for ρ_d is the average color of the flash photo.

The variable $\tilde{\rho}_s$ encodes the specular albedo in YUV. The reason for this is that we make the intensity of the specular spatially varying, but force the chroma to be constant over our image. This is typical behavior for materials, and enforcing it improves the stability of the optimization by reducing unnecessary degrees of freedom. Similarly, for stability reasons we make the kurtosis parameter α global. The additions of constants restrict the ranges of values that the transformed parameters can take.

Data fit residual. The data fit residual $\mathbf{R}_{\text{fit}}(\mathbf{X})$ measures the difference between the input lumitexel values $Z(i,j,s,t)$, and their rendered predictions given the current parameters $\mathbf{X}$. It is evaluated independently at each combination of pixel (s,t) and half-vector (i,j), and the residuals are summed.

Let $Q(\mathbf{x},\mathbf{g};i,j)$ be the function that renders a pixel at half-vector (i,j) with local BRDF parameters $\mathbf{x}$ and global parameters $\mathbf{g}$, according to the model in Section 4.3. The data fit residual for the component (i,j,s,t) of the lumitexel array is

$$\frac{150}{n} H_{0.1}\left\{\text{clamp}[Q(T(\mathbf{x}_{(s,t)},\mathbf{g});i,j)] - Z(i,j,s,t)\right\}, \qquad (2)$$

where $T(\mathbf{x},\mathbf{g})$ performs the domain mappings described above, clamp(a) saturates the pixel value at 1, and $H_\gamma(a) = 2\gamma^2\left(\sqrt{1+\frac{a^2}{\gamma^2}} - 1\right)$ is pseudo-Huber loss function, which smoothly approximates the robust ℓ_1-norm. The leading multiplier is a weighting, where n is the number of half-vector directions in the data. The residual is computed separately for each color channel.

Priors We use three types of priors: pointwise priors that specify the desired ranges for individual variables, smoothness priors that encourage spatially continuous solutions, and an integrability prior that encourages the normal map to be consistent. The full prior residual is a weighted sum of these residuals: respectively, $\|\mathbf{R}_{\text{priors}}(\mathbf{X})\|^2 := 10^{-4}\|\mathbf{R}_{\text{p}}(\mathbf{X})\|^2 + 0.5\|\mathbf{R}_{\text{s}}(\mathbf{X})\|^2 + 25\|\mathbf{R}_{\text{i}}(\mathbf{X})]\|^2$. All of the priors operate on the raw parameter values, without the domain mappings described above.

The pointwise prior $\|\mathbf{R}_p(\mathbf{X})\|^2$ is a sum of separate residuals for each local variable of each pixel. Let $\tilde{\rho}_d, \tilde{\rho}_{s,1}, \mathbf{s}$ and $\tilde{\mathbf{n}}$ be the current parameters at pixel (s,t). The residual is then computed as $4\|\tilde{\rho}_d\|^2 + 16\|\tilde{\rho}_{s,1} + 6\|^2 + 0.25\|\mathbf{s}_1 + 6\|^2 + 0.25\|\mathbf{s}_2 + 6\|^2 + 0.25\|\mathbf{s}_3\|^2 + \|\tilde{\mathbf{n}}\|^2$. The main purpose of these priors is to discourage gross outliers, and to favor the use of diffuse component for wide lobes.

The smoothness prior $\|\mathbf{R}_s(\mathbf{X})\|^2$ is computed by considering each spatially varying variable as a separate image and evaluating its (unnormalized) finite differences in both x- and y-direction. The difference maps are weighted by a variable-wise weighting, after which we compute the Huber loss as above. We use the weight 0.2 for $\tilde{\rho}_d$ and 0.1 for the other variables. All difference residuals at each point are summed together. The Huber norm penalizes spatial variations of these variables in a manner similar to the ℓ_1 norm: smoothness is generally preferred, but occassional abrupt jumps are allowed.

Finally, as consistent normal maps have zero curl, the integrability prior $\|\mathbf{R}_i(\mathbf{X})\|^2$ is computed by evaluating the square norm of the finite difference curl of the normal vector field: $\|\mathbf{R}_i(\mathbf{X})\|^2 = \|\nabla_y \tilde{\mathbf{n}}_x - \nabla_x \tilde{\mathbf{n}}_y\|^2$, where $\tilde{\mathbf{n}}_x$ and $\tilde{\mathbf{n}}_y$ are suitably vectorized images of the normal map components.

Construction and differentiation of residuals. Aside from the residual function $\mathbf{R}$, the Levenberg-Marquardt algorithm requires that we supply its Jacobian matrix of partial derivatives against all the optimization variables. This matrix is very large but sparse, as all residuals only depend on a few variables. In practice we form the residual vector and the Jacobian by hierarchically applying simple operations on the input variables in bulk, while simultaneously keeping track of the partial derivatives using the chain rule. We compute analytic derivatives for all operations except for the rendering function, which we differentiate by finite differences. For L-M, we use MATLAB's `lsqnonlin` without Jacobian-based diagonal scaling, with initial trust region size parameter of 1.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3 (Aug.), 294–302.

AITTALA, M., WEYRICH, T., AND LEHTINEN, J. 2013. Practical SVBRDF capture in the frequency domain. *ACM Transactions on Graphics (Proc. SIGGRAPH) 32*, 4 (July), 110:1–110:12.

BARRON, J., AND MALIK, J. 2015. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)*, 1–19.

BARROW, H. G. B., AND TENENBAUM, J. M. 1978. Recovering intrinsic scene characteristics from images. *Computer Vision Systems* (Apr.), 3–26.

BRADY, A., LAWRENCE, J., PEERS, P., AND WEIMER, W. 2014. genBRDF: Discovering new analytic BRDFs with genetic programming. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4 (July), 114:1–114:11.

CALONDER, M., LEPETIT, V., STRECHA, C., AND FUA, P. 2010. BRIEF: binary robust independent elementary features. In *Proc. European Conference on Computer Vision*, Springer-Verlag, Berlin, Heidelberg, 778–792.

CLARK, R., 2010. Crazybump. `http://www.crazybump.com`, Last access: 7 May 2015.

DONG, Y., WANG, J., TONG, X., SNYDER, J., LAN, Y., BEN-EZRA, M., AND GUO, B. 2010. Manifold bootstrapping for SVBRDF capture. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (July), 98:1–98:10.

DONG, Y., TONG, X., PELLACINI, F., AND GUO, B. 2011. Appgen: interactive material modeling from a single image. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 30*, 6 (Dec.), 146:1–146:10.

DPREVIEW, 2012. Quick review: Apple iPhone 5. `www.dpreview.com/articles/6867454450/quick-review-apple-iphone-5-camera`, Last access: 7 May 2015.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proc. SIGGRAPH*, ACM, New York, NY, USA, 341–346.

GLENCROSS, M., WARD, G., JAY, C., LIU, J., MELENDEZ, F., AND HUBBOLD, R. 2008. A perceptually validated model for surface depth hallucination. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (Aug.), 59:1–59:8.

GOLDMAN, D., CURLESS, B., HERTZMANN, A., AND SEITZ, S. 2005. Shape and spatially-varying BRDFs from photometric stereo. In *Proc. IEEE International Conference on Computer Vision*, vol. 1, 341–348.

HAINDL, M., AND FILIP, J. 2013. *Visual Texture*. Advances in Computer Vision and Pattern Recognition. Springer.

HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proc. SIGGRAPH*, ACM, New York, NY, USA, 229–238.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proc. SIGGRAPH*, 327–340.

JOHNSON, K., COLE, F., RAJ, A., AND ADELSON, E. 2011. Microgeometry capture using an elastomeric sensor. *ACM Transactions on Graphics (Proc. SIGGRAPH) 30*, 4 (July), 46:1–46:8.

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics 22*, 2 (Apr.), 234–257.

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2006. Image-based reconstruction of spatially varying materials. In *Proc. Eurographics Symposium on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, 31–40.

MARSCHNER, S. 1998. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University.

NGAN, A., AND DURAND, F. 2006. Statistical acquisition of texture appearance. In *Proc. Eurographics Symposium on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, 31–40.

RABIN, J., PEYRÉ, G., DELON, J., AND BERNOT, M. 2011. Wasserstein barycenter and its application to texture mixing. In *Proc. Intl. Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, Springer, vol. 6667 of *Lecture Notes in Computer Science*, 435–446.

SCHRÖDER, K., ZINKE, A., AND KLEIN, R. 2015. Image-based reverse engineering and visual prototyping of woven cloth. *IEEE Transactions on Visualization and Computer Graphics 21*, 2 (Feb.), 188–200.

WANG, J., ZHAO, S., TONG, X., SNYDER, J., AND GUO, B. 2008. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (Aug.), 41:1–41:9.

WANG, C.-P., SNAVELY, N., AND MARSCHNER, S. 2011. Estimating dual-scale properties of glossy surfaces from step-edge lighting. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 30*, 6 (Dec.), 172:1–172:12.

WEYRICH, T., LAWRENCE, J., LENSCH, H., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4*, 2, 75–191.

ZICKLER, T., RAMAMOORTHI, R., ENRIQUE, S., AND BELHUMEUR, P. N. 2006. Reflectance sharing: predicting appearance from a sparse set of images of a known shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 8, 1287–1302.