

Efficient Reflectance Capture Using an Autoencoder

KAIZHANG KANG, State Key Lab of CAD&CG, Zhejiang University, China

ZIMIN CHEN, State Key Lab of CAD&CG, Zhejiang University, China

JIAPING WANG, Sinovation Ventures, USA

KUN ZHOU, State Key Lab of CAD&CG, Zhejiang University, China

HONGZHI WU, State Key Lab of CAD&CG, Zhejiang University, China

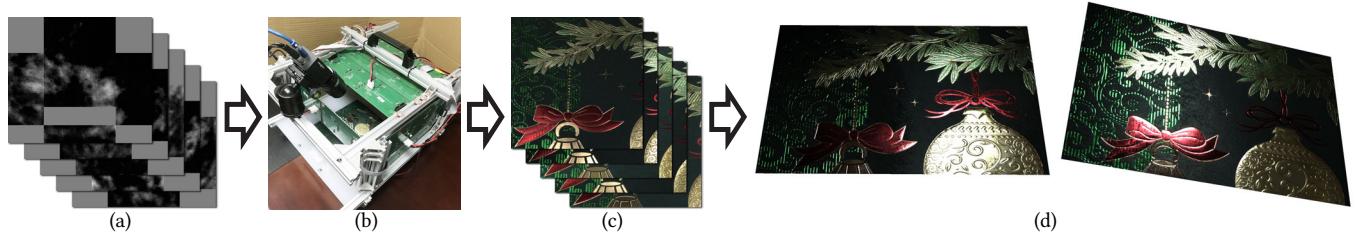


Fig. 1. Using as few as 16 ~ 32 lighting patterns (a) learned from a large amount of reflectance data, we efficiently capture photographs (c) of a planar physical sample in a mini, near-field lightstage (b), and faithfully reconstruct its SVBRDF that can be rendered under novel lighting and view conditions (d).

We propose a novel framework that automatically learns the lighting patterns for efficient reflectance acquisition, as well as how to faithfully reconstruct spatially varying anisotropic BRDFs and local frames from measurements under such patterns. The core of our framework is an asymmetric deep autoencoder, consisting of a nonnegative, linear encoder which directly corresponds to the lighting patterns used in physical acquisition, and a stacked, nonlinear decoder which computationally recovers the BRDF information from captured photographs. The autoencoder is trained with a large amount of synthetic reflectance data, and can adapt to various factors, including the geometry of the setup and the properties of appearance. We demonstrate the effectiveness of our framework on a wide range of physical materials, using as few as 16 ~ 32 lighting patterns, which correspond to 12 ~ 25 seconds of acquisition time. We also validate our results with the ground truth data and captured photographs. Our framework is useful for increasing the efficiency in both novel and existing acquisition setups.

CCS Concepts: • Computing methodologies → Reflectance modeling;

Additional Key Words and Phrases: reflectance acquisition, optimal sampling, lighting patterns, SV-BRDF

ACM Reference Format:

Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. 2018. Efficient Reflectance Capture Using an Autoencoder. *ACM Trans. Graph.* 37, 4, Article 127 (August 2018), 10 pages. <https://doi.org/10.1145/3197517.3201279>

Hongzhi Wu (hwu@acm.org) is the first corresponding author, and Kun Zhou (kunzhou@acm.org) is the second. Kaizhang Kang and Zimin Chen contributed equally. Authors' addresses: Kaizhang Kang, State Key Lab of CAD&CG, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, 310058, China; Zimin Chen, State Key Lab of CAD&CG, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, 310058, China; Jiaping Wang, Sinovation Ventures, 4158 Interdale Wy, Palo Alto, CA, 94306, USA; Kun Zhou, State Key Lab of CAD&CG, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, 310058, China; Hongzhi Wu, State Key Lab of CAD&CG, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, 310058, China.

© 2018 Association for Computing Machinery.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, <https://doi.org/10.1145/3197517.3201279>.

1 INTRODUCTION

Digitally acquiring high-quality material appearance from the real-world is a crucial and challenging problem, with applications in visual effects, e-commerce, product design and entertainment. One fundamental difficulty here is the sheer complexity of the material appearance. It can be modeled as a Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF), a 6D function that varies with location, lighting and view directions. Directly capturing the unknown, general SVBRDF of a planar sample with a conventional gonioreflectometer [Dana et al. 1999; Lawrence et al. 2006] requires taking thousands or even millions of photographs from all possible combinations of lighting and view directions, in order to sufficiently sample high-frequency features such as sharp highlights. This makes the approach prohibitively expensive both in time and storage.

Significant research efforts have been made to improve the efficiency of reflectance acquisition over the past years. One popular class of methods for high-quality SVBRDF acquisition are based on complex lighting patterns. Instead of using a single light source at a time, such a system illuminates a planar physical sample with many intensity-programmable light sources simultaneously, effectively forming different lighting patterns. It then takes corresponding photographs with a camera, and reconstructs the reflectance from these measurements. Representative work includes the lightstage [Ghosh et al. 2009; Tunwattanapong et al. 2013], the linear light source reflectometry [Chen et al. 2014; Gardner et al. 2003], and the LCD-based setup [Aittala et al. 2013]. However, hundreds of photographs are still needed in the general setting [Chen et al. 2014], limiting the practicality in real-world applications like e-commerce, where the physical acquisition efficiency is critical in digitizing the appearance of a large number of products.

In this paper, we observe two fundamental problems in lighting-pattern-based reflectance acquisition:

- (1) What are the optimal lighting patterns to use, given a highly limited number of them?
- (2) How to faithfully recover the reflectance properties from photographs captured under such patterns?

To answer the above questions, existing work conducts sophisticated, hand-crafted derivations, for each type of illumination-multiplexing setup; the number of lighting patterns depends on the theories used in the derivations and cannot be arbitrarily specified; the optimality of the derived patterns and the reflectance reconstruction algorithm is not verified over a large number of test samples with considerable variations. In comparison, we propose a novel framework that **automatically** learns the lighting patterns as well as the reflectance reconstruction algorithm, for any given number of patterns, using the state-of-the-art deep learning techniques in a data-driven fashion.

Our key insight is that illumination-multiplexing reflectance acquisition can be viewed as an autoencoder: for each point on the sample, its BRDF is physically multiplied with different lighting patterns and encoded in the corresponding photographs; then the BRDF information is computationally decoded. This motivates our design of an asymmetric deep autoencoder: it has a nonnegative, linear encoder that directly corresponds to lighting patterns used in the acquisition hardware, and a stacked, nonlinear decoder to harness the powerful learning ability of modern deep neural networks to map the measurements to the BRDF information. After training with a large number of synthetic reflectance data, our deep autoencoder can faithfully recover a wide variety of BRDFs, ranging from sharp isotropic or narrow anisotropic specular lobes to broad diffuse ones. Furthermore, our approach is highly flexible as the training is performed in a data-driven fashion that can adapt to various factors, including the geometry of the setup and the properties of appearance.

The effectiveness of our framework is demonstrated with a mini, near-field lightstage to physically acquire a wide range of material appearance, using a very small number of lighting patterns ($16 \sim 32$), which correspond to a short acquisition time ($12 \sim 25$ seconds). We achieve the **lowest** number of lighting patterns in a general setting (near-field lighting, anisotropic results), compared with 240 patterns used in [Chen et al. 2014] (near-field lighting, anisotropic results), 128 patterns used in [Aittala et al. 2013] (near-field lighting, isotropic results) and 44 patterns used in [Tunwattanapong et al. 2013] (distant lighting, anisotropic results), which are the three techniques most similar to ours. We also compare our results with the ground truth data, obtained by densely sampling 2,560 lights.

2 RELATED WORK

2.1 Optimal BRDF Sampling

The canonical way to model an unknown, general BRDF requires a dense sampling over its 4D domain. To improve the efficiency, Matusik et al. [2003b] use 800 samples to model a BRDF, assuming that it lies in the subspace of pre-captured, isotropic ones [Matusik et al. 2003a]. With a similar assumption, Nielsen et al. [2015] reduce the number of samples to about 20, via an improved algorithm that optimizes lighting and view sampling directions. Xu et al. [2016] further reduce the number to 2, by exploiting the view direction

variations in a near-field camera. They also propose a simple extension to isotropic SVBRDFs, assuming no normal variations, and only a small number of basis materials, each of which has a good coverage over the field of view.

2.2 Measurement-Based Reflectance Acquisition

Extensive work has been published on reflectance acquisition using images captured under controlled / uncontrolled lighting. Please refer to [Weyrich et al. 2009] and [Weinmann and Klein 2015] for excellent surveys on recent acquisition techniques. Below we review some of the previous work that is most relevant to our approach.

2.2.1 Direct Sampling. Methods in this category probe the material appearance in its 6D domain. The most straightforward, general approach is to exhaustively sample a large number of lighting and view directions by mechanically positioning a camera and/or a light source, and capturing photographs for every possible combination of the two factors [Dana et al. 1999; Lawrence et al. 2006]. These approaches are typically time-consuming.

To reduce the acquisition cost, various approaches have been proposed to reconstruct the reflectance from a lower number of images, by assuming priors over the reflectance data. The reflectance of a homogeneous convex object can be recovered with a single view direction, by exploiting the normal variations to sufficiently sample the angular domain [Marschner et al. 1999]. Spatially varying reflectance on a known shape can be reconstructed from a sparse number of photographs, assuming that the appearance is a linear combination of basis materials [Lensch et al. 2003]. Zickler et al. [2005] share the reflectance information over a 6D domain and reconstruct the reflectance via the scattered-data interpolation. Wang et al. [2008] exploit the spatial similarity of reflectance and the spatial variation of local frames, to complete the microfacet distributions of BRDFs from single-view measurements. An efficient two-phase reflectance acquisition method is proposed in [Dong et al. 2010], assuming that the reflectance lies on a low-dimensional manifold. Recently, Aittala et al. [2015] use only two photographs to model the appearance of stochastic-texture-like materials.

In comparison, none of the above priors is explicitly assumed in our framework. In particular, we reconstruct the reflectance at each point **independently**, despite the low number of measurements. On the other hand, additional material properties can be easily exploited without involved manual derivations, by training our autoencoder with specific samples (Sec. 8.2).

2.2.2 Complex Lighting Patterns. Our work is most similar to this class of methods, which record the responses of a material sample under different lighting patterns, and recover the reflectance properties from the measurements.

The lightstage systems [Ghosh et al. 2009; Tunwattanapong et al. 2013] capture photographs of a material sample under spherical harmonics (SH) lighting patterns, and recover the reflectance from a manually derived inverse lookup table, which maps the observed radiance to anisotropic BRDF parameters. Recently, Nam et al. [2016] propose a similar system that reconstructs micro-scale reflectance via an alternating optimization, with the assumption of a small

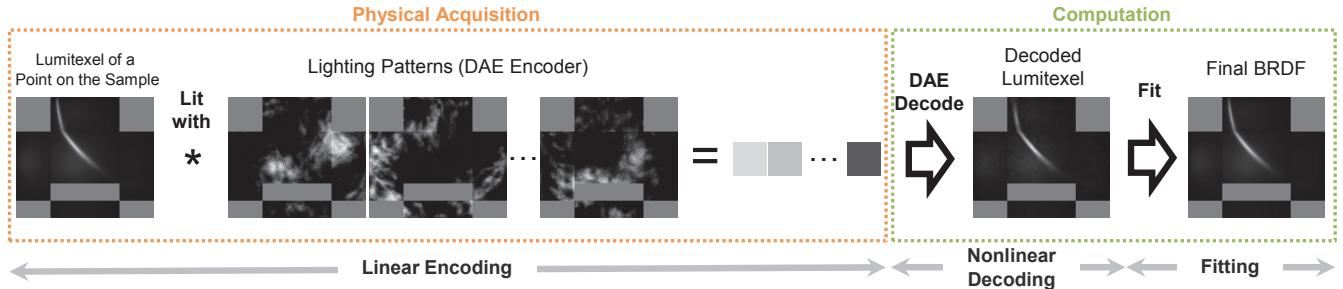


Fig. 2. Our acquisition pipeline is composed of three stages. First, we illuminate the physical sample with the learned lighting patterns in our setup, and capture a small number of corresponding photographs, essentially performing physical linear encoding of the lumitexels. Next, for a point on the sample, the encoding result is fed to the nonlinear decoder in our DAE to recover the lumitexel as if captured by looping over each individual light source. Finally, we fit a 4D BRDF along with a local frame to the reconstructed lumitexel as the result.

number of basis materials. All these systems assume that the incident lighting is distant with respect to the size of the sample.

The setup in [Gardner et al. 2003; Ren et al. 2011] scans a linear light source over a planar, isotropic material sample. The system is extended to handle anisotropic reflectance in [Chen et al. 2014], by modulating the intensity along the linear light source, assuming a low rank appearance subspace. Aittala et al. [2013] propose a system with a single camera and a tilted, near-field LCD panel as a programmable planar light source, to acquire isotropic reflectance based on a hand-derived frequency domain analysis.

Our framework uses as few as 16 ~ 32 near-field lighting patterns, to efficiently and faithfully acquire spatially varying anisotropic BRDFs and the local frames. While related work heavily relies on manual derivations, our framework automatically determines both the lighting patterns and the method to reconstruct reflectance from measurements, using machine learning techniques.

2.3 Deep-Learning-Assisted Reflectance Modeling

Despite their success in many fields in computer vision and graphics, deep learning techniques so far are only applied to the single-image-based problem in reflectance modeling. Aittala et al. [2016] model the isotropic SVBRDF and surface normals, by synthesizing from a single flash image of a stationary textured material. The challenging, precise point-to-point correspondences are avoided by adopting a texture descriptor based on a convolutional neural network (CNN). Recently, Li et al. [2017] present a CNN-based solution for modeling SVBRDF from a single photograph of a planar sample with unknown natural illumination, using a self-augmentation training process.

We refer the readers to [Hinton and Salakhutdinov 2006] for more details on deep autoencoders, and [Goodfellow et al. 2016] for an introduction to general deep learning techniques.

3 PRELIMINARIES

We formulate our SVBRDF acquisition problem before introducing our framework. Without loss of generality, we assume a single-camera acquisition setup with effectively independently controlled, near-field or distant light sources. No polarization filter is used. In addition, we assume a planar sample of interest, whose appearance can be modeled as an anisotropic SVBRDF. No spatial coherence is

exploited; the reflectance at each point is reconstructed independently.

Measurement Equation. The reflected radiance B observed by the camera can be modeled as follows:

$$B(I, p) = \int \frac{1}{||x_l - x_p||^2} I(l) \Psi(x_l, -\omega_i) f_r(\omega_i'; \omega_o', p) (\omega_i \cdot n_p) (-\omega_i \cdot n_l) dx_l. \quad (1)$$

We model each light as a locally planar source. x_p/n_p are the position / normal of a point p on the physical sample, and x_l/n_l are the position / normal of a point on a light source l . ω_i/ω_o are the lighting / view directions in the world space, while ω_i'/ω_o' are their counterparts expressed in the local frame of p . ω_l can be computed as $\omega_l = \frac{x_l - x_p}{||x_l - x_p||}$. $I(l)$ is the programmable intensity for the light l over its maximum intensity, in the range of [0, 1]. The array $\{I(l)\}_l$ corresponds to a lighting pattern. $\Psi(x_l, \cdot)$ describes the angular distribution of the light intensity when fully on. $f_r(\cdot; \omega_o', p)$ is a 2D BRDF slice, which is a function of the lighting direction only. The above integral is computed over all light sources.

BRDF Representation. Our framework is not tied to any specific BRDF model. In this paper, we use the anisotropic GGX BRDF model [Walter et al. 2007] to efficiently represent f_r :

$$f_r(\omega_i; \omega_o, p) = \frac{\rho_d}{\pi} + \rho_s \frac{D_{GGX}(\omega_h; \alpha_x, \alpha_y) F(\omega_i, \omega_h) G_{GGX}(\omega_i, \omega_o; \alpha_x, \alpha_y)}{4(\omega_i \cdot n)(\omega_o \cdot n)}, \quad (2)$$

where ρ_d/ρ_s are the diffuse / specular albedo, α_x/α_y are the roughnesses parameters, and ω_h is the half vector. D_{GGX} is the microfacet distribution function, F is the Fresnel term and G_{GGX} accounts for shadowing / masking effects, all of which are detailed in the supplemental material for brevity. We choose the GGX model for three reasons: first, it is a compact parametric model that can represent a wide range of materials; second, it is the de-facto industry standard for physically-based BRDFs [McAuley et al. 2012]; finally, it allows efficient real-time rendering.

Lumitexel. Due to the linearity with respect to I in Eq. 1, B can be expressed as the dot product between I and a lumitexel m :

$$B(I, p) = \sum_l I(l)m(l; p). \quad (3)$$

Similar to [Lensch et al. 2003], m is a function of the light source j , defined on each point \mathbf{p} of the physical sample:

$$m(j; \mathbf{p}) = B(\{I(l=j) = 1, I(l \neq j) = 0\}, \mathbf{p}). \quad (4)$$

Each element of m is the reflected radiance B with only one light source turned on and set to the maximum intensity, and the remaining lights off. The lumitexel will be used as a key data structure in our framework.

Problem Formulation. From Eq. 1 & 2, for a point \mathbf{p} on the physical sample, reflectance acquisition is essentially to solve for the unknown BRDF f_r and its local frame, parameterized as $\{\rho_d, \rho_s, \alpha_x, \alpha_y, \mathbf{n}, \mathbf{t}\}$, from the photographs $\{B(I, \mathbf{p})\}_I$ captured with pre-determined lighting patterns $\{I(l)\}_l$. All other variables involved in Eq. 1 can be pre-calibrated.

4 OUR FRAMEWORK

From the aforementioned insight into our problem, we propose a deep autoencoder for lumitexels (L-DAE) that automatically learns to encode and decode the lumitexel m (Eq. 4) for each point \mathbf{p} on the physical sample, for a given number of lighting patterns (Sec. 5). We then fit a 4D BRDF along with the local frame to the lumitexel (Sec. 7). The procedure is performed for every point on the sample, which yields texture maps that describe the 6D SVBRDF as the final results. An illustration of our pipeline is shown in Fig. 2. Below we briefly discuss the major considerations that lead to the design of our framework.

4.1 Design Considerations

At first glance, one natural way of applying machine learning to our problem is to construct a linear encoder that corresponds to the lighting patterns, followed by a regression network that directly outputs BRDF parameters. Although this end-to-end approach is straightforward, a major issue is that there are one-to-many relationships from certain BRDFs to their BRDF parameters, which are difficult to learn by regression. For example, a pure Lambertian BRDF can map to various sets of parameters as long as they share the same ρ_d and \mathbf{n} , and $\rho_s = 0$, regardless of other parameters.

To avoid this issue, we propose an autoencoder-based framework instead. An autoencoder maps the input to itself in a one-to-one fashion, which is more amenable for deep learning techniques. Now the question becomes what our autoencoder should learn. One intuitive answer is to learn the 2D BRDF slice $f_r(\cdot; \omega_0, \mathbf{p})$, from the camera view. However, this choice undesirably complicates the decoder, which is required to take away the complex near-field lighting effects baked in the input physical measurements (Eq. 1), in order to produce a BRDF slice as output.

Therefore, we choose to learn the lumitexel (Eq. 4), because it assimilates all near-field-lighting-related terms such as the cosine and the inverse squared distance one (Eq. 1) and it is parameterized over light sources. These two properties result in a **spatially invariant** linear relationship among the lighting pattern I , the lumitexel m and the measurements B (Eq. 3); such a simple relationship is amenable for modeling and training. In addition, sufficient training lumitexels can be easily generated, by virtually rendering synthetic BRDFs with a variety of parameters (Sec. 7). A large number of varied training samples are critical for applying deep learning techniques.

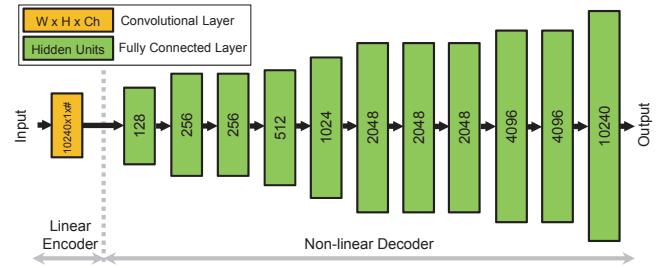


Fig. 3. The structure of our asymmetric DAE. We use a linear encoder which is implemented as a convolutional layer that represents all lighting patterns, followed by a nonlinear encoder consisting of 11 fully connected layers, which recovers the lumitexel from the encoding result. Here # indicates the number of lighting patterns.

Finally, to bridge the gap between the decoded lumitexel from our autoencoder and the final 4D BRDF result, we perform a separate BRDF fitting step. There are two reasons for applying this decoupled fitting. First, it can easily handle near-field lighting in a nonlinear optimization, without adding undesired complexity to the autoencoder, which in the current design is not aware of \mathbf{p} . Second, it makes our autoencoder independent of the underlying BRDF model; no re-training or tuning is needed, when switching to a different BRDF model.

5 L-DAE

We introduce the single-channel L-DAE with details in this section. In acquisition, it is applied to each of the RGB channels to obtain an RGB lumitexel as the result. On a high level, the DAE consists of two parts: a nonnegative, linear encoder, and a stacked, nonlinear decoder (Fig. 2). The encoding is conducted physically by projecting the lighting patterns to the physical sample in the acquisition setup, and then taking measurements of the reflected radiances, essentially performing dot products between the lumitexel and the lighting patterns, according to Eq. 3. The lighting patterns $\{I(l)\}_l$ directly correspond to weights in our encoder. For decoding, measurements under the lighting patterns are fed to a stacked, width-increasing network, which produces a lumitexel as output.

More specifically, the encoder network is implemented as a convolutional (conv) layer with no padding. We treat all lighting patterns as a single convolution kernel of $c \times 1 \times \#$, where c is the dimension of a lumitexel and $\#$ is the number of lighting patterns. The decoder network has 11 fully connected (fc) layers. We use fc layers over conv ones in the decoder, to avoid making assumptions on the spatial relationships between different elements in the lumitexel. Each fc layer is preceded by a batch normalization (bn) layer, and followed by a leaky ReLU activation layer. One exception is that the first fc layer has no preceding bn layer and is directly connected with the conv layer in the encoder. Please refer to Fig. 3 for an illustration of the network structure.

Our asymmetric L-DAE is different from conventional autoencoders in a couple of ways. First, its encoder needs to be directly mapped to the physical acquisition process on the hardware. This excludes various complex operations, and leaves only nonnegative

multiplication and addition between the lumitexel and the lighting patterns (Eq. 3) as viable options. Second, even though our encoder is physically limited to the nonnegative linear form, the decoder runs on a computer and is not subject to such limitation. So we use a stacked nonlinear neural network as the decoder, to harness the powerful learning ability of modern deep neural networks to faithfully map the encoded result to the original lumitexel.

5.1 Loss Function

The loss function L for training our L-DAE consists of two terms:

$$L = L_{\text{auto}}(m) + \lambda \sum_{w \in \text{enc.}} L_{\text{barrier}}(w). \quad (5)$$

Here the first term measures the error of the reconstructed lumitexel m with respect to its ground truth m_{gt} :

$$L_{\text{auto}}(m) = \sum_j [\log(1 + m(j)) - \log(1 + m_{\text{gt}}(j))]^2. \quad (6)$$

Note that we apply the log transform to alleviate the problem of possible large values in the specular lobe dominating this term, similar to [Nielsen et al. 2015]. The second term is a barrier function to ensure physical plausibility of the computed lighting patterns. It penalizes any weight w in the encoder that is beyond the range of $[0, 1]$, as w corresponds to the ratio of the lighting intensity over its maximum intensity for each source (Sec. 3):

$$L_{\text{barrier}}(w) = \tanh\left(\frac{w - (1 - \epsilon)}{\epsilon}\right) + \tanh\left(\frac{-w + \epsilon}{\epsilon}\right) + 2. \quad (7)$$

We find that $\lambda = 0.03, \epsilon = 0.005$ works well in our experiments.

5.2 Training Data

To train the L-DAE to faithfully reconstruct a wide range of possible lumitexels in the real-world, we synthesize the training data by evaluating Eq. 1, using a large number of randomly generated f_r , the local frame and the location on the physical sample. All three factors have an impact over the lumitexel m (Eq. 4).

Specifically, for the local frame, we randomly sample \mathbf{n} in the upper hemisphere of the sample plane, and then \mathbf{t} as a random unit vector that is orthogonal to \mathbf{n} . Similarly, for the location on the physical sample, we randomly choose a point from the valid region of the sample plane. For the BRDF f_r , we use the anisotropic GGX model and randomly sample ρ_d/ρ_s uniformly in the range of $[0, 1]$, and α_x/α_y uniformly on the log scale in the range of $[0.006, 0.5]$. The calibration data of the acquisition setup (Sec. 6) are used when evaluating Eq. 4 for training lumitexel generation.

Despite that the anisotropic GGX model works well for training data generation in all our experiments, we would like to emphasize that our framework at this step is again not tied to any specific BRDF model. Moreover, we do not require that the sampled BRDF in training data generation here and the final fitting results in Sec. 4 share the same model. In fact, any BRDF model that well covers the material variation in the physical samples of interest can be used for generating the training data.

6 ACQUISITION SETUP

Our automatic acquisition setup can be viewed as a mini, near-field lightstage. The size of the setup is approximately 420mm × 360mm

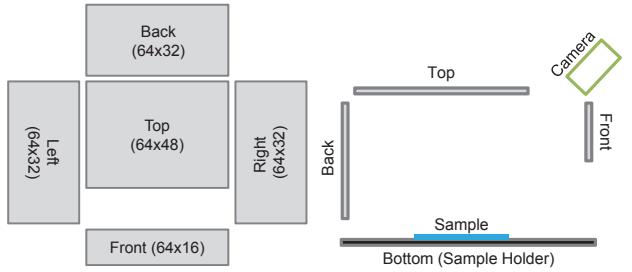


Fig. 4. Our LED layout and our acquisition device. The LED layout is shown on the left, with all 5 LED boards unfolded to the same 2D plane. The number of LEDs on each board are marked as (#columns × #rows). A side view of our device is illustrated on the right.

× 210mm. A single machine vision camera, PointGrey Grasshopper 3, is installed to capture photographs of the physical sample at approximate 45 degrees from the sample plane, with a resolution of 2,736 × 2,192. Please refer to Fig. 4 for an illustration. The camera has a narrow field of view and is focused on the sample.

To acquire reflectance properties, we illuminate a planar physical sample with 10,240 white LEDs ($c = 10,240$), each of which emits light from a rectangular region of about 1.4mm × 1.4mm. The LEDs are grouped as boards and mounted to the left, right, front, back and top sides of our setup, while the sample is placed on the bottom. The maximum size of the sample is 120mm × 120mm. Please refer to Fig. 4 for a visualization of the LED layout. The intensity of each LED can be independently controlled via Pulse Width Modulation (PWM) by an Altera Cyclone IV FPGA. We use 8 bits to quantize the LED intensity in our experiments. Note that since the size of our setup and that of the sample is on the same order of magnitude, we can no longer assume distant lighting as in traditional light-stages (e.g., [Ghosh et al. 2009]) and must take near-field effects into consideration in reflectance reconstruction.

The intrinsic and extrinsic parameters of the camera, as well as the positions, orientations and angular intensity distribution of LEDs, are all calibrated before acquisition experiments. Color correction is performed based on photographs of an X-Rite ColorChecker under different lighting patterns. The scale ambiguity of diffuse / specular albedo is resolved with the help of a planar diffuse patch of a uniform albedo, similar to [Gardner et al. 2003].

7 IMPLEMENTATION DETAILS

Training. Our DAE is implemented with the TensorFlow framework. For back propagation, we use RMSProp [Tieleman and Hinton 2012] with mini-batches of 50 and a momentum of 0.9. For the linear encoder, the initial weights are drawn i.i.d. from a normal distribution ($\mu = 0, \sigma = 0.01$) and processed as follows: we flip the sign for a negative sampled weight; the weight is also clamped to 1 if needed, though this rarely happens due to the small σ . For all weights in the decoder, Xavier initialization is applied. For training, we run 250K iterations with a learning rate of 1×10^{-4} , followed by 50K iterations with a learning rate of 1×10^{-6} . We generate 1 million synthetic lumitexels using the method described in Sec. 5.2, from which 80% are used for training, and 20% for validation.



Fig. 5. Visualization of different lighting patterns. From the top row to the bottom: photographs of a physical sample lit with corresponding lighting patterns in the second row, our lighting patterns computed from anisotropic training samples ($\# = 32$), our lighting patterns from isotropic samples ($\# = 24$), and lighting patterns computed with PCA on anisotropic samples ($\# = 32$). Only a subset of all lighting patterns are shown here due to the limited space.

Mean Subtraction. For training, we perform mean subtraction for each input lumitexel passing to our DAE, and add the mean back to the final output, as common in deep neural networks [Goodfellow et al. 2016]. In physical acquisition, we perform an equivalent operation by feeding the mean lumitexel to the linear encoder only, and subtracting the result from the measurements for each point on the physical sample. The mean lumitexel is also added back to the output of L-DAE.

BRDF Fitting. We fit an anisotropic GGX model and a local frame to the output lumitexel from our L-DAE as final results. First, box-constrained, nonlinear least squares fitting is performed using the Levenberg-Marquardt algorithm [Lafortune et al. 1997], by minimizing the squared differences between the lumitexel computed with current estimates of parameters and the output from L-DAE, in a single, gray-scale channel. We then fix the normal, tangent and roughnesses parameters, and fit the chromatic diffuse and specular albedos with nonnegative, linear least squares.

8 RESULTS AND DISCUSSIONS

We conduct our experiments on a PC with an Intel Core i7-7700 CPU, 32GB memory, and a GeForce GTX 1080 Ti video card. It takes approximately 5 hours to train our DAE from scratch. In experiments, we merge 2 low-dynamic-range (LDR) photographs of the physical sample with different exposures into HDR ones using bracketing. The typical acquisition time using 32 learned lighting patterns is 25 seconds. This time scales linearly with respect to the number of lighting patterns. The size for all captured HDR photographs is about 1.8GB. The decoding time of L-DAE from measurements is 4 minutes for processing 1 million lumtexels. For the ground truth

data, we uniformly subsample a quarter of all independently controlled LEDs ($c/4 = 2,560$) and image the physical sample with one LED fully on at a time, essentially capturing a subsampled version of the lumitexel m . The reason for subsampling is that we find 2,560 LEDs are sufficient to capture the most specular materials among our datasets. So we do not sample over all lights to save the acquisition time. Due to the relatively lower power of a single LED, we have to increase the exposure time to get reliable measurements. Consequently, the acquisition time for the ground truth data is around 14,000 seconds. The total size of all photographs is about 145GB. The subsampled lumtexels are then fitted to the GGX BRDF models, which is the same as the final step of our pipeline. Using our unoptimized code, it takes 1.6 hours to conduct BRDF fitting on 1 million lumtexels, for decoded ones from our framework and the ground truth ones. The performance is comparable to previous work with a similar setup [Aittala et al. 2013].

Fig. 5 visualizes the lighting patterns learned by our L-DAE. We show a subset of all lighting patterns obtained by training with anisotropic samples ($\# = 32$) and isotropic samples ($\# = 24$). In the same figure, we also show captured photographs of a physical sample lit with the lighting patterns (anisotropic samples, $\# = 32$): rich material variations in the angular domain are revealed under our computed lighting patterns. As a comparison, we visualize the lighting patterns obtained by performing Principal Component Analysis (PCA) over the same set of anisotropic training samples. The results look similar to SH patterns used in previous work with a distant lighting assumption.

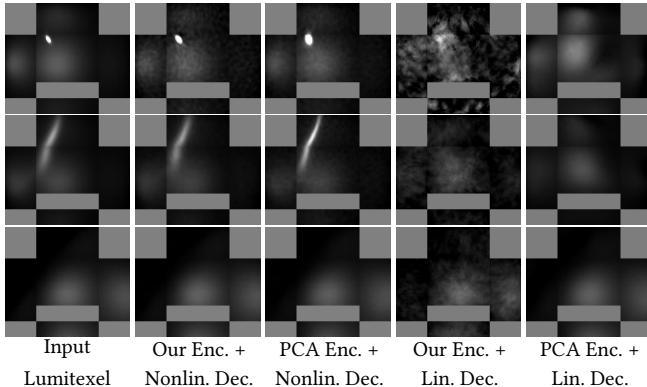


Fig. 6. Reconstruction results using different encoding / decoding strategies. For encoding, we use the lighting patterns computed using either our framework (anisotropic training samples, # = 32) or PCA (# = 32). For decoding, we use either the nonlinear structure illustrated in Fig. 3, or the pseudo inverse of the matrix corresponding to all lighting patterns.

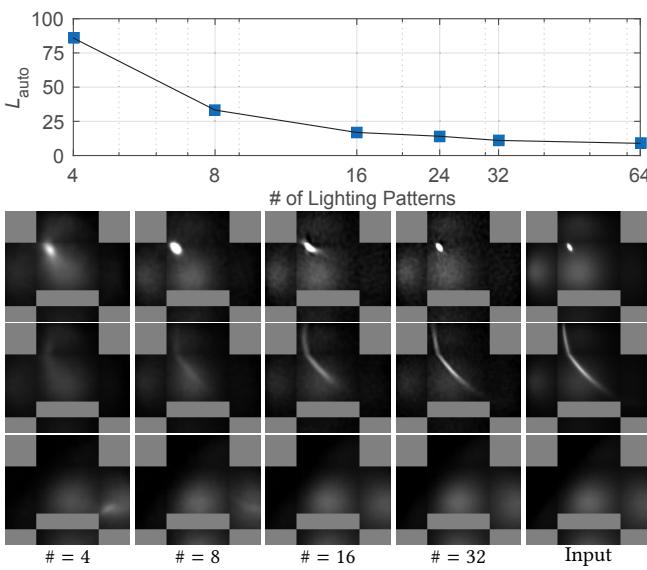


Fig. 7. The impact of the number of lighting patterns over lumitexel reconstruction quality. Top chart: L_{auto} as a function of the number of lighting patterns. Bottom three rows: the output of DAEs that differs in the number of lighting patterns for different input lumitexels in the rightmost column.

8.1 Modeling Results and Comparisons

Fig. 12 shows the ability of our L-DAE (anisotropic samples, # = 32) to faithfully reconstruct a wide range of randomly generated lumitexels, which vary in albedos, normals, sample locations, anisotropy axes and roughnesses. For fairness, the input lumitexels are not used in the training of the DAE.

We demonstrate the effectiveness and generality of our framework over 6 physically captured SVBRDF datasets, which cover a wide range of material appearance. In Fig. 11, the rendering results of SVBRDFs reconstructed with our L-DAE using only 32 light patterns closely match the ground truth computed using 2,560

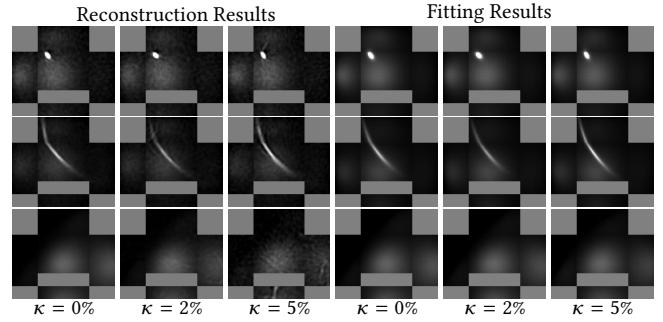


Fig. 8. The impact of perturbation over the encoding results using our DAE (anisotropic samples, # = 32). For each component η of the encoding, we add a Gaussian noise with a zero mean and a standard deviation that equals $\kappa|\eta|$. The reconstruction results using our DAE are shown in the left three columns, while the corresponding fitting results in the right three columns.

lights, as well as photographs taken by our setup. The rendering results with a novel medium-sized disk light source and novel view conditions, are also shown in the figure. Moreover, in Fig. 11, we report quantitative errors of our reconstruction with respect to the ground truth, which are comparable to one state-of-the-art technique [Nam et al. 2016]. Please refer to the accompanying video for animated results. In Fig. 13, we show the texture maps of BRDF parameters, after fitting the GGX model to the output lumitexels by L-DAE (anisotropic samples, # = 32).

8.2 Evaluations

In Fig. 6, we evaluate the impact of different encoding / decoding strategies over lumitexel reconstruction. First, we fix and set the linear encoder in our DAE structure to the SH-like lighting patterns computed with PCA over anisotropic training samples, and only allow the weights in the decoder to be optimized in the training process. The resulting DAE produces lumitexels of lower quality, compared with our DAE with both the encoder and decoder trained in an end-to-end fashion. Next, we investigate the theoretically optimal linear decoder: the pseudo-inverse of the matrix that corresponds to the linear encoder. We conduct experiments using the L-DAE encoder and the PCA encoder. Except for the case where the diffuse lobe is dominant, the reconstruction results are of considerably lower quality, due to inability of small-sized linear decoder to accurately recover high frequency features in the lumitexels (e.g., tiny or highly narrow specular lobes).

Next, we evaluate in Fig. 7 the impact of the number of lighting patterns over the reconstruction quality of lumitexels. We plot L_{auto} as a function of the light pattern number. As more light patterns are used, the reconstruction error L_{auto} decreases, since more information about the input is passed down to the decoder, which makes it possible to produce a more accurate output.

We also perform sensitivity tests in Fig. 8 over our DAE by adding a Gaussian noise to each component of the encoding result, with a standard deviation proportional to the magnitude of the component, to simulate possible noise / factors not modeled in the acquisition process. Both reconstructed lumitexels and fitting results are shown in the figure.

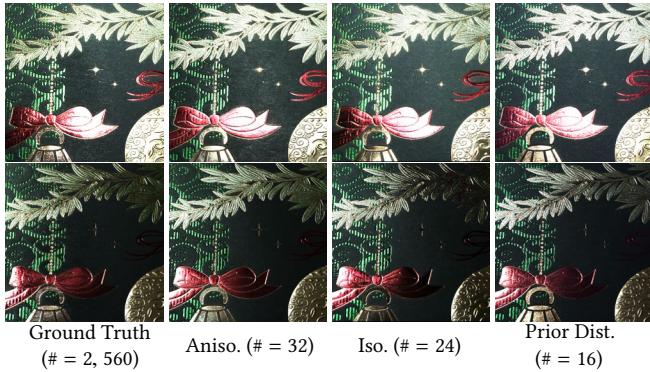


Fig. 9. The impact of the distribution of training samples. From the left column to the right: SVBRDF reconstruction results from the ground truth data (#=2,560), and from DAEs trained with anisotropic BRDF samples (#=32), isotropic BRDF samples (#=24), and BRDF samples drawn i.i.d. from the precaptured SVBRDF of the same physical subject (#=16). A distribution of BRDF training samples which is closer to that of the physical subject permits fewer lighting patterns to be used in the acquisition, for comparable reconstruction quality.

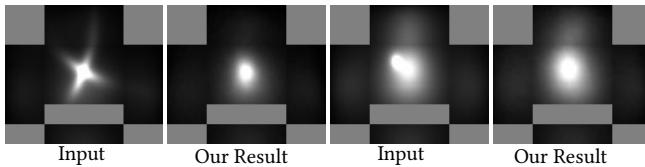


Fig. 10. Failure cases. For each pair of images, the left one is the input lumitexel and the right one is the reconstruction using the DAE trained with anisotropic GGX samples (# = 32). As our framework is based on DAE, we cannot reconstruct BRDFs that substantially differ from the training samples.

Finally, we study the impact of training data distribution over the lighting pattern number needed for reconstructing SVBRDFs of decent quality in Fig. 9. We test L-DAEs trained with anisotropic samples, isotropic samples, and samples with all parameters drawn i.i.d. from the ground truth data, except that the sample locations are randomly determined. As more knowledge about the SVBRDF of interest is exploited in the training of DAE, the amount of information that needs to be determined from the measurements reduces, resulting in a decrease in the number of light patterns for reconstructions of similar quality, as listed in the figure.

9 LIMITATIONS AND FUTURE WORK

Our work is subject to a number of limitations. First, as a data-driven approach, our L-DAE cannot recover lumitexels that substantially deviate from training samples, as shown in Fig. 10. Second, our framework is limited to handle mostly planar samples, whose geometric variations can be well modeled by normal maps. Also, for certain materials, our setup may not observe key reflectance features from only one fixed view direction.

In the future, it will be interesting to apply our framework to existing setups, such as the linear light source reflectometry and the distant lightstage, to automatically learn what efficient lighting

patterns are and how to recover SVBRDFs from the measurements lit with these patterns. It will also be intriguing to extend our framework to learn optimal view sampling as well, similar to [Nielsen et al. 2015]. Finally, to apply our framework to image-based relighting (e.g., [Peers et al. 2009]), a topic closely related to reflectance acquisition, is a promising future direction.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers, Yue Dong, Min H. Kim, Giljoo Nam, Xin Tong, Cihui Xie, Minyi Gu, Jingke Wang, Shuyu Jiang, Yuan Wan, Minmin Lin and Meng Yuan for generous help to this project. This work is partially supported by the National Key Research & Development Program of China (2016YFB1001403), NSF China (61772457 & U1609215), and the Fundamental Research Funds for the Central Universities (2017XZZX009-03).

REFERENCES

- Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4, Article 65 (July 2016), 13 pages.
- Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2013. Practical SVBRDF Capture in the Frequency Domain. *ACM Trans. Graph.* 32, 4, Article 110 (July 2013), 12 pages.
- Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Trans. Graph.* 34, 4, Article 110 (July 2015), 13 pages.
- Guojun Chen, Yue Dong, Pieter Peers, Jiawan Zhang, and Xin Tong. 2014. Reflectance Scanning: Estimating Shading Frame and BRDF with Generalized Linear Light Sources. *ACM Trans. Graph.* 33, 4, Article 117 (July 2014), 11 pages.
- Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. 1999. Reflectance and Texture of Real-world Surfaces. *ACM Trans. Graph.* 18, 1 (Jan. 1999), 1–34.
- Yue Dong, Jiaping Wang, Xin Tong, John Snyder, Yanxiang Lan, Moshe Ben-Ezra, and Baining Guo. 2010. Manifold Bootstrapping for SVBRDF Capture. *ACM Trans. Graph.* 29, 4, Article 98 (July 2010), 10 pages.
- Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. 2003. Linear light source reflectometry. *ACM Trans. Graph.* 22, 3 (2003), 749–758.
- Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul Debevec. 2009. Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination. *Computer Graphics Forum* 28, 4 (2009), 1161–1170.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507.
- Eric P. F. Laforte, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. 1997. Non-linear Approximation of Reflectance Functions. In *Proc. SIGGRAPH*, 117–126.
- Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. 2006. Inverse Shade Trees for Non-parametric Material Representation and Editing. *ACM Trans. Graph.* 25, 3 (July 2006), 735–745.
- Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. 2003. Image-based Reconstruction of Spatial Appearance and Geometric Detail. *ACM Trans. Graph.* 22, 2 (April 2003), 234–257.
- Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling Surface Appearance from a Single Photograph Using Self-augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4, Article 45 (July 2017), 11 pages.
- Stephen R. Marschner, Stephen H. Westin, Eric P. F. Laforte, Kenneth E. Torrance, and Donald P. Greenberg. 1999. Image-based BRDF Measurement Including Human Skin. In *Proc. EGWR*. 131–144.
- Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003a. A Data-driven Reflectance Model. *ACM Trans. Graph.* 22, 3 (July 2003), 759–769.
- Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. 2003b. Efficient Isotropic BRDF Measurement. In *Proc. EGWR*. 241–247.
- Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. 2012. Practical Physically-based Shading in Film and Game Production. In *ACM SIGGRAPH 2012 Courses*. Article 10, 7 pages.
- Giljoo Nam, Joo Ho Lee, Hongzhi Wu, Diego Gutierrez, and Min H. Kim. 2016. Simultaneous Acquisition of Microscale Reflectance and Normals. *ACM Trans. Graph.* 35, 6, Article 185 (Nov. 2016), 11 pages.
- Jannik Boll Nielsen, Henrik Wan Jensen, and Ravi Ramamoorthi. 2015. On Optimal, Minimal BRDF Sampling for Reflectance Acquisition. *ACM Trans. Graph.* 34, 6, Article 186 (Oct. 2015), 11 pages.

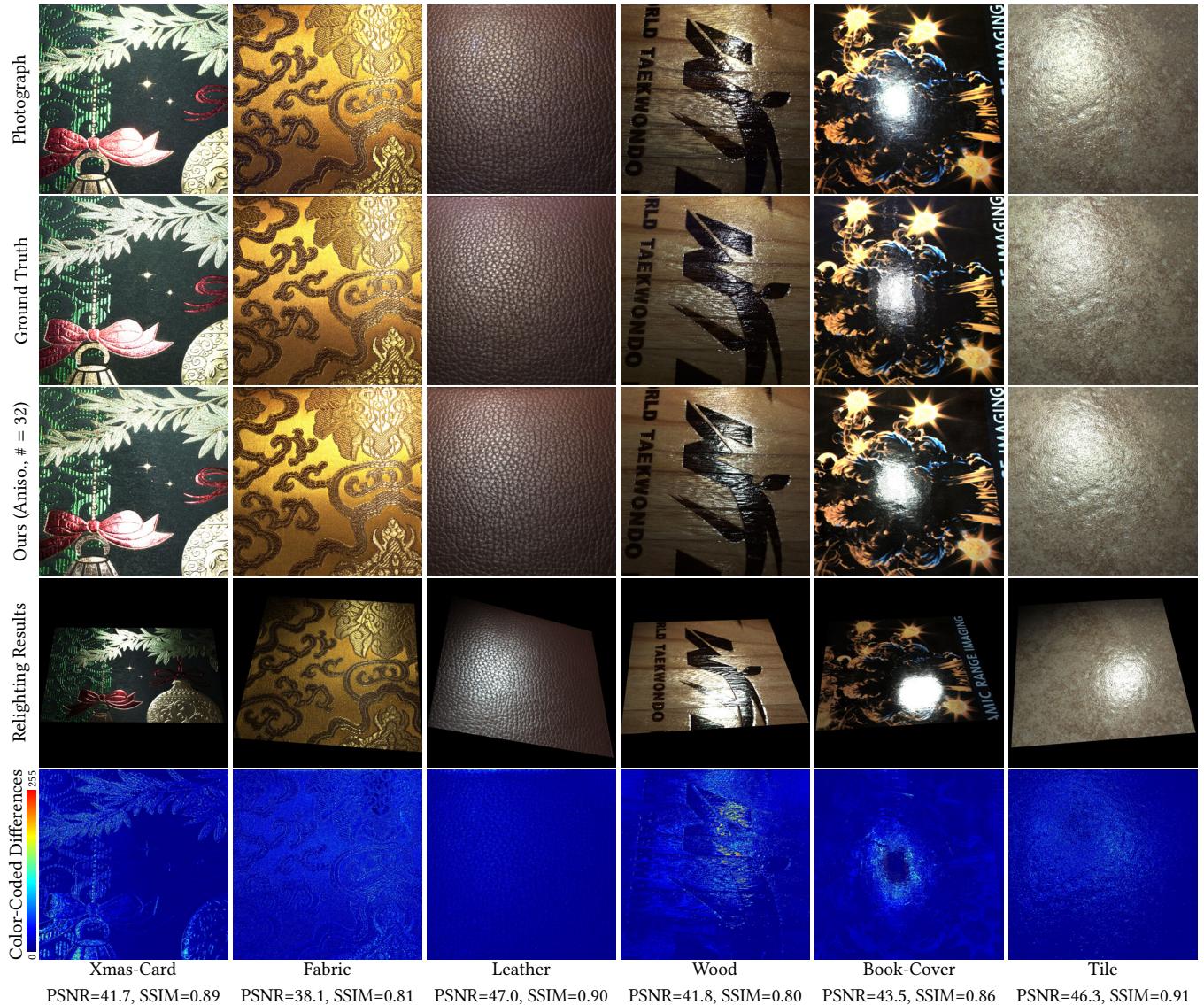


Fig. 11. Main acquisition results. For images from the top row to the bottom in each column: a photograph of the physical sample, the rendering of ground-truth SVBRDF (# = 2, 560), the rendering of SVBRDF reconstructed using our lighting patterns (anisotropic training samples, # = 32), the rendering of our result with novel lighting and view conditions, and the color-coded difference image between our result and the photograph. The last row reports quantitative errors (PSNR & SSIM) of our results with respect to the ground truth, with PSNR measured in decibel. Please refer to the accompanying video for animated results.

- Pieter Peers, Dhruv K. Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. 2009. Compressive Light Transport Sensing. *ACM Trans. Graph.* 28, 1, Article 3 (Feb. 2009), 18 pages.
- Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. 2011. Pocket reflectometry. *ACM Trans. Graph.* 30, 4 (2011), 1–10.
- Tijmen Tielenman and Geoffrey Hinton. 2012. Lecture 6.5-RMSProp: Divide the Gradient by a Running Average of Its Recent Magnitude. *Neural Networks for Machine Learning* 4 (2012), 26–31.
- Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. 2013. Acquiring Reflectance and Shape from Continuous Spherical Harmonic Illumination. *ACM Trans. Graph.* 32, 4, Article 109 (July 2013), 12 pages.
- Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Rendering Techniques (Proc. EGWR)*.

- Jiapeng Wang, Shuang Zhao, Xin Tong, John Snyder, and Baining Guo. 2008. Modeling Anisotropic Surface Reflectance with Example-based Microfacet Synthesis. *ACM Trans. Graph.* 27, 3, Article 41 (Aug. 2008), 9 pages.
- Michael Weinmann and Reinhard Klein. 2015. Advances in Geometry and Reflectance Acquisition. In *SIGGRAPH Asia Courses*. Article 1, 71 pages.
- Tim Weyrich, Jason Lawrence, Hendrik P. A. Lensch, Szymon Rusinkiewicz, and Todd Zickler. 2009. Principles of Appearance Acquisition and Representation. *Found. Trends. Comput. Graph. Vis.* 4, 2 (2009), 75–191.
- Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF Sampling for Two-shot Near-field Reflectance Acquisition. *ACM Trans. Graph.* 35, 6, Article 188 (Nov. 2016), 12 pages.
- Todd Zickler, Sebastian Enrique, Ravi Ramamoorthi, and Peter Belhumeur. 2005. Reflectance Sharing: Image-based Rendering from a Sparse Set of Images. In *Proc. EGSR*. 253–264.

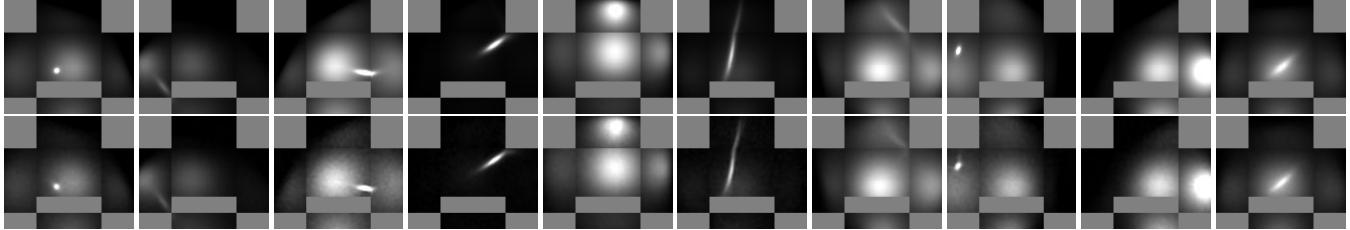


Fig. 12. Reconstruction results of various lumitexels of different materials, local frames and sample locations, using our L-DAE. The top row: randomly sampled input lumitexels (not used in training); the bottom row: the output of L-DAE, before the fitting step.

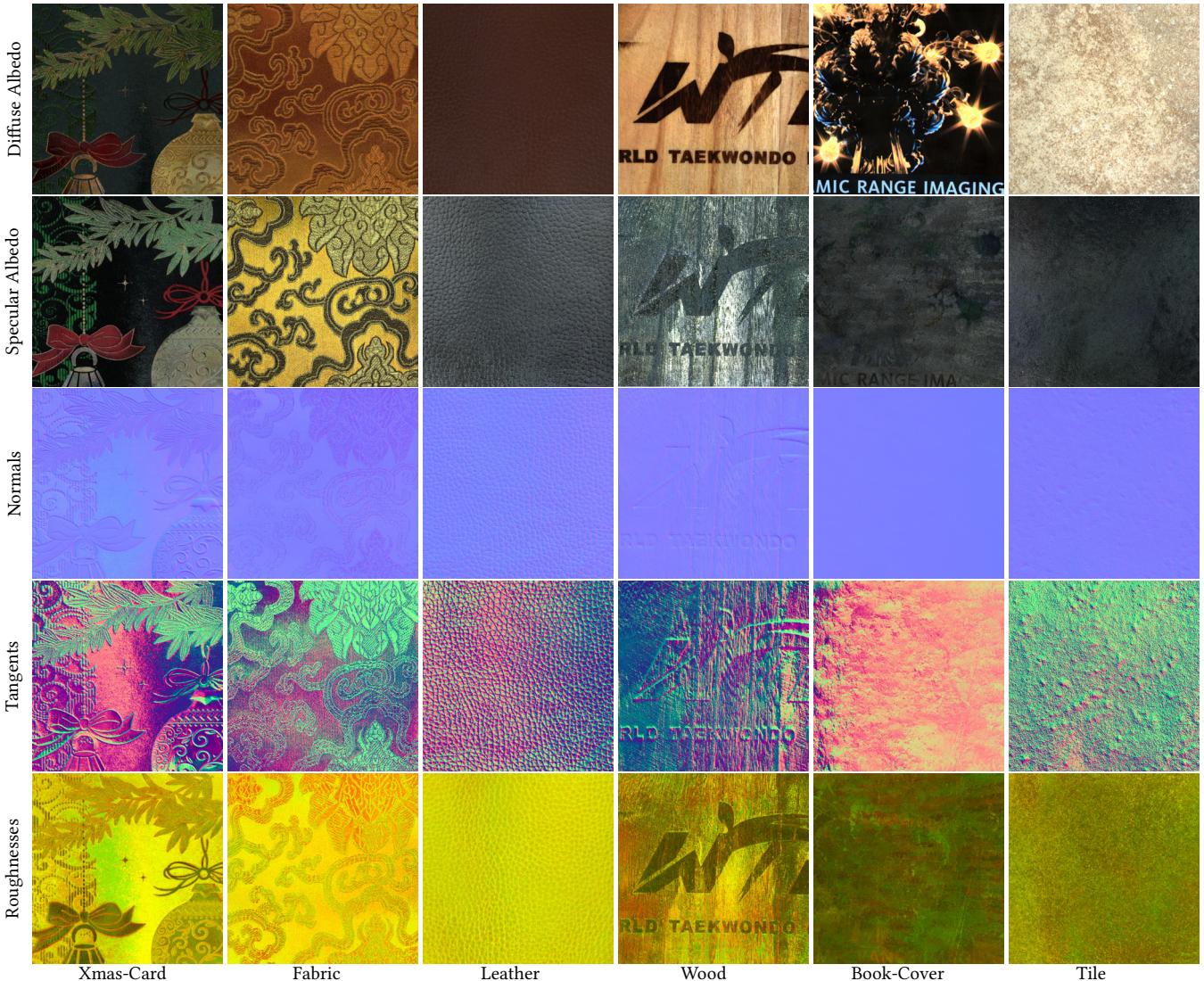


Fig. 13. GGX model fitting results. Each normal is added with $(1, 1, 1)$ and then divided by 2 to fit to the range of $[0, 1]^3$ for visualization. The tangents are visualized in the same manner. For roughnesses, α_x/α_y are visualized in the red / green channel.