Deep Appearance Maps

Maxim Maximov
Technical University Munich

Laura Leal-Taixé
Technical University Munich

Mario Fritz
CISPA Helmholtz Center for Information Security

Tobias Ritschel
University College London

Abstract

We propose a deep representation of appearance, i. e., the relation of color, surface orientation, viewer position, material and illumination. Previous approaches have used deep learning to extract classic appearance representations relating to reflectance model parameters (e. g., Phong) or illumination (e. g., HDR environment maps). We suggest to directly represent appearance itself as a network we call a Deep Appearance Map (DAM). This is a 4D generalization over 2D reflectance maps, which held the view direction fixed. First, we show how a DAM can be learned from images or video frames and later be used to synthesize appearance, given new surface orientations and viewer positions. Second, we demonstrate how another network can be used to map from an image or video frames to a DAM network to reproduce this appearance, without using a lengthy optimization such as stochastic gradient descent (learning-to-learn). Finally, we show the example of an appearance estimation-and-segmentation task, mapping from an image showing multiple materials to multiple deep appearance maps.

1. Introduction

The visual appearance of an object depends on the combination of four main factors: viewer, geometry, material and illumination. When capturing and processing appearance, one wishes to change one or more of those factors and predict what the new appearance is. This can be achieved using methods ranging from implicit image-based representations [6, 13] to explicit Computer Graphics-like representations [27]. Implicit methods take a couple of photos as input and allow to predict high-quality imagery in a limited set of conditions, but modest flexibility, e. g., interpolating an image between two photos but not extrapolating to new views. Explicit representations allow for more flexibility when acquiring Phong parameters and HDR illumination maps [27], but incur substantial acquisition effort, e. g., taking a large number of calibrated (HDR) photos.

DAMs propose a new direction to represent appearance: we move away from the pixel-based nature of implicit image-based representations into a deep representation, but without any explicit reconstruction, as we do not target a direct mapping to any explicit reflectance model or illumination either. Still, we show that such a representation can be used to solve actual tasks, such as image synthesis, appearance acquisition and estimation-and-segmentation of appearance. This is enabled by four contributions:

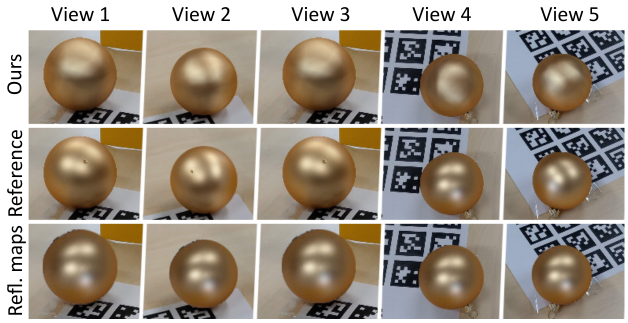

Figure 1. Frames from a video with a moving viewer **(columns)** comparing a re-synthesis using our novel deep appearance maps (DAMs) **(top)** and reflectance maps (RMs) **(bottom)** to a photo reference of a decorative sphere with a complex material under natural illumination **(middle)**.

First, we will introduce a generalization of 2D reflectance maps [13] to 4D, which we call an Appearance Map (AM). AMs represent appearance for varying geometry under varying views. This allows freely changing the viewer and surface geometry, which is not possible for classic reflectance maps that fix the relation between view and illumination (cf. Fig. 1).

Second, while classic Reflectance Maps (RM) can be simply tabulated using a single 2D image, the full appearance is a 4D phenomenon that is more challenging to represent and process. Storing 4D appearance as a table, modestly resolving 10 degree features, would require storing (and somehow also capturing) $36^4 = 17\,M$; impractical. Instead, we suggest using DAMs, neural networks that compactly represent AMs in only a couple of thousand parameters. This representation is efficient, does not require any look-ups and is differentiable. In addition, it can be learned effectively from images or video frames with a known viewer position and surface orientation. Applying this representation to new view positions and surface orientations can be done at speed competitive to classic rendering or RMs, i. e., within milliseconds for full images.

Acquiring a DAM requires learning a deep model for every new appearance in practice. This would incur substantial computational effort, i. e., running an optimization compared to capturing a RM image in seconds. Addressing this, our third contribution suggests to use another deep (convolutional) neural network to map images showing an appearance to a DAM ([17, 34, 14], one-shot learning [8], "life-long", or "continual" learning). This capture requires milliseconds instead of minutes.

Fourth, the DAM representation can be used for joint material estimation-and-segmentation, a generalization of the previous objective. Here the input is an image with a known number of n materials, and output is n different DAMs, and a segmentation network that maps every pixel to a n weights.

We train and quantitatively test all networks on a new dataset of photo-realistically rendered images as well as on a set of real photos.

2. Related Work

Inverse Rendering One of the main aims of inverse rendering is to recover material and illumination properties of a scene. It is a quite challenging, ill-posed and under-constrained problem that remains hard to solve for the general case. Related recent work can be roughly divided into data-driven and algorithmic approaches.

Algorithmic methods are based on optimizing appearance properties for a given input [23]. These methods are usually off-line and make simplifying assumptions about the world to reduce computation time and avoid ambiguity and allow for a mathematical derivation. Most recent works [38, 31] use a set of real RGBD images to estimate appearance that are based on a specific illumination model. More refined models use data-based statistical priors to optimize for illumination and reflectance explaining an image [24, 22].

Deep-learning based approaches make a similar assumption as to how humans can recognize materials based on previous experience. Recent work [25, 10, 21, 16, 7] uses CNNs to estimate explicit reflectance model parameters. Similarly, encoder-decoder CNNs are used to estimate reflectance maps [29] or illumination and materials [12, 10].

All of theses methods – data-driven or not – have in common that they rely on a specific illumination model to estimate its explicit parameters (such as Phong diffuse, specular, roughness, etc) and they represent lighting as an HDR illumination map. To the one hand, this is more that what we do as it factors out lighting, to the other hand our approach is more general as it makes no assumption on light or geometry and works on raw 4D samples. One of the other limitations of above mentioned CNN methods is limited feedback from a loss function: a change of estimated illumination or reflectance can only be back-propagated through the image synthesis method with suitable rendering layers. Our method does not involve a renderer, circumventing this problem.

Appearance synthesis Methods to synthesize appearance – or simply "rendering" –, can be classified as simulation-based or image-based.

Simulation-based methods require a complete explicit description of the environment that can be costly and difficult to acquire in practice [27]. A simple, yet powerful, method to represent appearance is a reflectance map [13], a 2D image that holds the color observed for a surface of a certain orientation and material under a certain illumination. In graphics, reflectance maps are known as pre-filtered environment maps [15], or spherical harmonics (SH) to capture the entire light transport [33]. A single 2D envmap or 2D SH however, cannot reproduce 4D appearance variation and furthermore requires a high pixel resolution or many SH coefficients to capture fine details (highlights) that are easy to reproduce using a NN. The entire reflectance field is a spatial map of these, as captured by Debevec [5].

Image-based rendering (IBR) uses a set of 2D images to reconstruct a 3D model and present it in a different view or different light [6]. These methods do geometry prediction, often with manual intervention, with prediction of rendered material on top of it. Recent methods [39, 28] address this problem by using CNN models to predict completely novel views. The method of Rematas et al. [29] and establish a relation between surface and light transport properties and appearance given by photos, generating images "without rendering". A simple data-driven approach to IBR is to learn a per-pixel fully-connected neural network to reproduce per-pixel appearance [30] depending on light. A generalization of this is to shade images with per-pixel positions, normals and reflectance constraints [26]. Our method stems from the same root but neither works on pixel-based image rendering, nor does it reconstruct an explicit appearance model. We will instead use a deep representation of appearance itself.

Light fields [19] also store 4D appearance, but are parametrized by spatial or surface position [37] and store 2D lumitexels. BRDFs 4D-capture reflectance, but not illumination. Our DAMs capture the 4D relation of surface orientation and view direction instead.

Segmentation Classic segmentation does not take materials into account [32]. Recent material segmentation work, such as Bell et al. [2] is mostly a form of extended semantic segmentation into material labels (23 in their case): Most arm chairs might be made of only three different kinds of materials that such approaches are successful in detecting. In our work, we have abstract objects (like photographs of spheres Fig. 1), that do not provide much semantics and require using a continuous appearance representation. For videos of view-dependent appearance, this is particularly difficult. With adequate capture equipment, spatially varying appearance is captured routinely now [18, 11]. In particular, with very dense light field that covers a tiny angular region, changes in appearance can be used to separate specular and diffuse [1]. Our work uses much sparser samples and goes beyond a specular-diffuse separation to support arbitrary 4D appearance extrapolated over all view directions.

Another challenge is multi-materials estimation. Some work [10, 35] has used multiple materials under the same illumination, but they require pre-segmented materials. In our method we perform joint multi-material segmentation and estimation.

Learning-to-learn Learning-to-learn is motivated by the observation that a general optimizer, such as the one used to find the internal parameters for a network, will never be much better than a random strategy for all problems [36]. At the same time, intelligent actors can learn very quickly, which obviously does not require a full optimization [17]. We hypothesize, after seeing a material for some time, that a human, in particular a trained artist, would be able to predict its appearance in a new condition. This requires the ability to refine the learned model with new observations [34]. For convolutional networks, this was done in dynamic filter networks [14], but we are not aware of applications to appearance modeling, such as we will pursue here.

3. Deep Appearance Processing

3.1. Appearance maps

We model RGB appearance L_o of a specific material f_r under a specific distant illumination L_i as a mapping from absolute world-space surface orientation $\mathbf{n}$ and viewer direction ω_o (jointly denoted as $\mathbf{x}$) as in

$$L_\mathrm{o}(\underbrace{\omega_\mathrm{o},\mathbf{n}}_{=\mathbf{x}}) = \int L_\mathrm{i}(\omega_\mathrm{i}) f_\mathrm{r}(\omega_\mathrm{i},\omega_\mathrm{o}) < \mathbf{n},\omega_\mathrm{i} >^{+} \mathrm{d}\omega_\mathrm{i}.$$

Essentially, L_o is a six-dimensional function. In the following, we denote the two three-dimensional parameters – outgoing direction ω_o and surface orientation $\mathbf{n}$ – as a joint parameter vector $\mathbf{x}$. The concept if visualized in Fig. 2: In a classic reflectance map, the normals vary (blue arrows), but the view direction is the same (orange arrows). In our generalization, both view and normals vary arbitrarily. We might even observe the same normal under different views. Classic reflectance maps [13], assume a view direction $\mathbf{z}$ along the z axis and hold the relation of light and surface

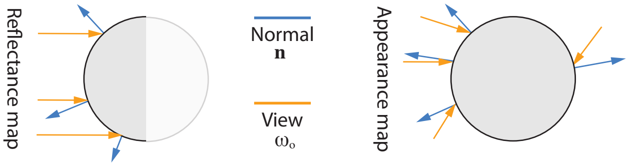

Figure 2. Reflectance and Appearance maps.

fixed, while also being limited to a single half-space:

$$L_\mathrm{RM}(\mathbf{n}) \text{ where } < \mathbf{n},\mathbf{z} > \leq \frac{\pi}{2}.$$

Covering the 4D sphere is motivated by our applications that allows to independently change view (2D) and surface orientation (2D). Note, no assumption on a BRDF is made as others do [12, 10].

3.2. Deep Appearance Maps

We use a deep neural network $\hat{L}_\mathrm{o}(\mathbf{x}|\theta)$ to approximate $L_\mathrm{o}(\mathbf{x})$ where θ denotes the networks internal parameters (Fig. 4, a). The input to such a network is the surface orientation and viewer direction parametrized as Euclidean vectors, i. e., a total of six numbers. This is followed by several fully-connected layers that are ultimately combined into a single RGB output value. Using 1×1 convolutions provide independence of image or object structure. Here, stochastic gradient descent (SGD) is used to minimize

$$\operatorname*{arg\,min}_{\theta,\delta} c_\mathrm{d}(\theta,W) + \alpha c_\mathrm{a}(\theta,\delta)$$

according to the α-weighted sum of a data cost c_d that depends on the DAM model parameters and an adversarial cost c_a that further includes the cost of the parameters of an adversarial model δ, that is biasing the solution to be plausible. W is a weight vector that is set to 1 for now, but will be required later for segmentation. We use $\alpha = .001$. The data cost is defined as

$$c_\mathrm{d}(\theta) = \frac{1}{n}\sum_{i=1}^{n} W_i||\hat{L}_\mathrm{o}(\mathbf{x}_i|\theta) - L_\mathrm{o}(\mathbf{x}_i)||, \qquad (1)$$

where $L_\mathrm{o}(\mathbf{x})$ are the observed appearance for normal and view direction $\mathbf{x}$ and $\hat{L}_\mathrm{o}(\mathbf{x}|\theta)$ is modeled appearance with parameter θ. The adversarial cost is defined as

$$c_\mathrm{a}(\theta,\delta) = \underbrace{\sum_{I\in\mathcal{I}''} \Delta_\mathrm{a}(R(\theta, I'_{\mathrm{n/v}})|\delta)}_{\text{Rendered appearance is fake}} + \underbrace{\sum_{I'\in\mathcal{I}} (1-\Delta_\mathrm{a}(I'_{\mathrm{rgb}})|\delta))}_{\text{Real appearance is real}}, \qquad (2)$$

where $\mathcal{I}$ is a set of images with per-pixel color ($\mathrm{I_{rgb}}$), normals($\mathrm{I_n}$) and view directions ($\mathrm{I_b}$) (detailed in Sec. 4), Δ_a is an adversarial network with parameters δ, classifying

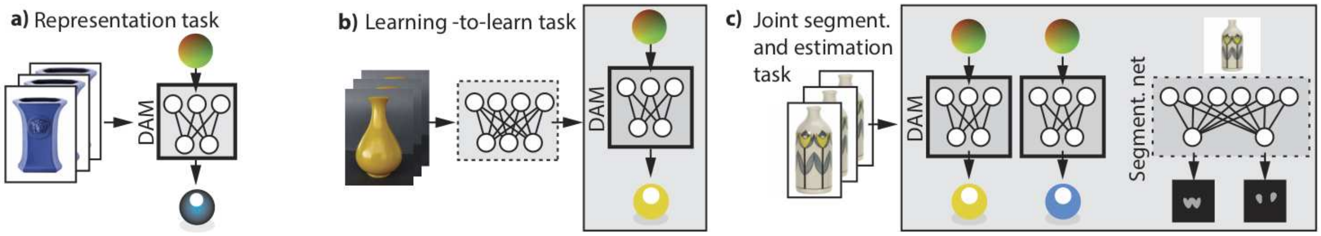
a) Representation task
DAM
b) Learning -to-learn task
DAM
c) Joint segment. and estimation task
DAM
Segment. net

Figure 3. Different appearance processing tasks that we address using our deep appearance maps. **a)** The first task simply reproduces a given appearance, i. e., it maps from normal and view directions to RGB values using a NN. **b)** In a learning-to-learn task a network maps an image to a DAM representation. **c)** Finally, in the segmentation-and-estimation task, a network maps an image to multiple DAMs and multiple segmentation networks.

its argument as fake when it is 1, and R is an operator that applies the appearance model with parameters θ to the per-pixel normals and view directions in image I (re-synthesis / rendering). The adversarial network Δ_a itself is a common encoder-style classifier as detailed in Fig. 4, b. It classifies the input image into a single value between 0 and 1. The GAN incentivizes the solution to produce plausible results in regions where no normals or view directions are observed during training.

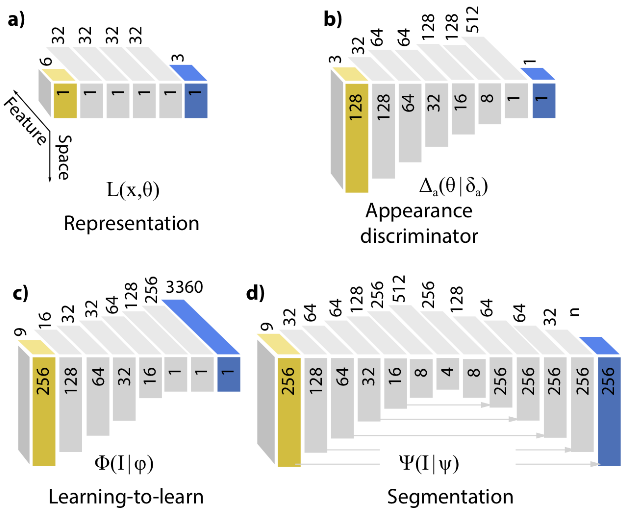

Figure 4. The four architectures used.

Learning a deep appearance model takes more than a minute while it is executed on a full image within one millisecond. We will now see how this representation enables two novel applications: learning-to-learn material appearance (Sec. 3.3) and material estimation-and-segmentation (Sec. 3.4).

3.3. Learning-to-learn Deep Appearance Models

Taking it a step further, we suggest to replace the learning procedure explained before by a network (learning-to-learn [34]). The main idea is to speed up the learning process, allowing acquisition of a deep appearance material on-the-fly at interactive rates (one millisecond) instead of an optimization requiring 71 seconds (for Tbl. 1 on a Nvidia GTX 1080). A general optimization process can find a good solution to all problems, but no learning approach that does well on all problems is much better than guessing [36]. However, we do not ask for "free lunch" here as we know that we do not want to solve all problems, but a specific subset: learning how to map normals and view directions to appearance.

To this end, we employ a convolutional neural network $\Theta(I|\phi)$ that is executed on an image I with a second set of internal parameters ϕ. Consequently, a network replaces a general learning algorithm [14]. This network can be compactly deployed and efficiently executed on arbitrary images to produce a DAM that can then be used for synthesis. In this sense, it is a network that predicts the weights (learned parameters) of network.

The input to the network is a 256×256 RGB, normal and view direction images I showing the appearance of a single material Fig. 4, c. Using an image instead of a plain list of samples allows the network to reason about spatial arrangement of values, e. g., detecting shapes and relations of highlights.

The output is a 3360-dimensional vector $\Theta(I|\phi)$ that describes the internal parameters of a network producing the appearance of I. The network Θ has eight layers, reducing resolution until a fully-connected layer. Training now minimizes for

$$\underset{\phi,\delta}{\arg\min} \sum_{I\in\mathcal{I}} c_\mathrm{d}(\Theta(I|\phi),\mathbf{w}) + \alpha c_\mathrm{a}(\Theta(I|\phi)|\delta), \qquad (3)$$

i. e., the same cost as in the previous section, but defined on the parameters ϕ of a network $\Theta(I|\phi)$ producing another network instead of the network parameters θ.

3.4. Appearance Estimation-and-Segmentation

A second application is joint appearance estimation and segmentation. Instead of holding a segmentation fix and estimating an appearance model for each segment or assuming an appearance to apply a segmentation, we jointly do both in an unsupervised way. We suggest using SGD itself as an optimization method to find the segmentation and appearance-predicting networks. Here, the DAM as well as a segmentation network are used as the latent variables to be inferred. The number of materials n is assumed to be known.

The appearance network parameters for all appearances are stacked into a matrix $\Theta(I) = (\Theta(I|\phi_1), \ldots, \Theta(I|\phi_n))^\mathsf{T}$.

Instead of directly inferring a per-pixel segmentation mask in the optimization, we suggest to learn a network $\Psi(\psi_i)$ with parameters ψ_i that jointly produces the all n segmentation masks (Fig. 4, d).

This network again is a simple encoder-decoder with skip connections that is shared among the materials in order to further reduce parameters. Input to this network is an image I with pixel color, normal, position, and output is a weight mask expressing how much a pixel belongs to a certain material i. There is one segmentation network parameter for each material i, and they are all stacked into a matrix $\Psi(I) = (\Psi(I|\psi_1), \ldots, \Psi(I|\psi_n))^\mathsf{T}$. The optimization now becomes

$$\begin{aligned}\operatorname*{arg\,min}_{\Theta,\Psi,\delta_\mathrm{a},\delta_\mathrm{m}} \sum_{i=1}^{n} & c_\mathrm{d}(\Theta(I|\phi_i), \Psi(I|\psi_i)) + \\ & \alpha c_\mathrm{a}(\Theta(I|\phi_i), \delta_\mathrm{a}) + \beta c_\mathrm{s}(W). \end{aligned} \tag{4}$$

Here, c_s is a sparsity term on the weight mask W that encourages a solution where most values for one pixel are zero, except one i. e., to have one unique material in most pixel. For every channel $\mathbf{w}$ in W it is $\sum_i \mathrm{abs}(\mathbf{w}_i - .5)$.

4. A Multi-view Multi-material Dataset

To work on arbitrary combinations for view, surface orientation and illumination for objects with multiple materials, we first produce a dataset. To our knowledge, no multi-material, multi-view dataset exists that allows for a controlled study. Examples from the dataset are shown in Fig. 5.

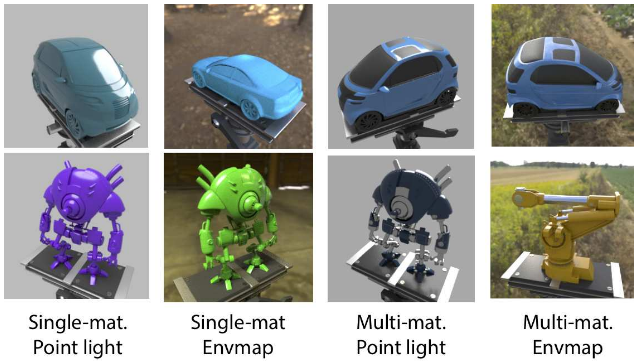

Figure 5. Two samples from four variants of our data set.

A seemingly obvious way to capture such a dataset from the real world is to take many photos at many exposure settings of many geometric objects under varying illuminations. Regrettably, this does not scale well to a high number of samples due to the curse of dimensionality encountered with so many parameters (the product of geometry, material, illumination and view). Also it would be difficult to manually decorate them with ground-truth material segmentation. Instead, we suggest to use photo-realistically rendered imagery.

We acquired five production-quality 3D objects from a model repository. As most of our architectures consider images simply as a list of 4D samples, without spatial layout, this comparatively low number of models appears adequate. Each model was assigned multiple (three) or one physically-plausible (layered GGX shading) materials organized on the objects surface in a complex and natural spatial arrangement. Before rendering, we randomize the hue of the diffuse component. For illumination, we used 20 different HDR environment maps. For each model, 32 different but equidistant view points on a circle around the object, with a random elevation, were used. Overall, this results in 5×20×32=3200 images. Note, the number of photos that would be required to exhaustively cover 4D; an order of magnitude higher. As the views and materials are randomized, no sharing between test and train sets exists. Geometry i. e., certain combinations of normals and view directions, might occur both in test and training data. We perform the same split into test and training for all tasks.

For rendering, we use Blender's [4] Cycles renderer with high-quality settings, including multiple indirect and specular bounces. Note that those light paths violate the model assumption. We add a virtual tripod to be closer to real photos, which typically also have local reflections which invalidate the model assumptions of distant illumination. The resulting images are linearly tone-mapped such that the 95th percentile maps to 1 and kept linear (non-gamma corrected). For each image I in the corpus $\mathcal{I}$, we store many channels: appearance as RGB colors I_c, position I_p, normals I_n, and a weight map I_w with n channels, where n is the number of materials.

Additionally to the ENVMAP version, we produce a variant with POINTLIGHT illumination (technically, a single, small area light) and split the set into flavors: MULTIMATERIAL and SINGLEMATERIAL. Using a single material, the material segmentation is ignored and one random material from the objects is assigned to the entire 3D objects. In the multi-material case, we proceed directly. Note, that such instrumentation would not be feasible on real photographs.

5. Results

5.1. Protocol

Here we evaluate our deep appearance representation (Sec. 5.2), as well as its application to learning-to-learn

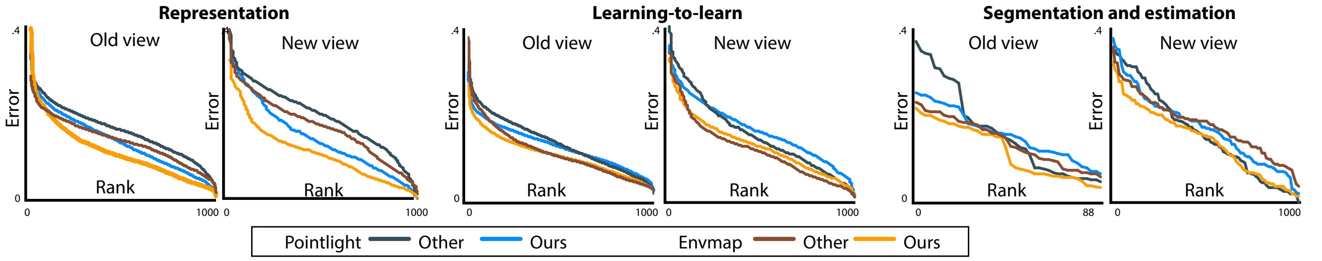

Figure 6. Pairs of error plots for each task. In each pair, the first is the old and the second the new view. Each curve is produced by sorting the DSSIM (less is better) of all samples in the data set. Blue colors are for point light illumination, red colors for environment maps. Dark hues are the competitor and light hues ours.

appearance (Sec. 5.3) and joint material estimation-and-segmentation (Sec. 5.4).

Instrumentation for all tasks is performed in a similar fashion using our multi-view, multi-material data set (Sec. 4). In particular, we consider its POINTLIGHT and ENVIRONMENTMAP variants. Depending on the task, we either use a SINGLEMATERIAL or MULTIMATERIAL. The main quantity we evaluate is image similarity error (DSSIM, less is better) with respect to a path-traced reference. We consider two tasks: re-synthesizing from the SAMEVIEW (training views) as well as from a NOVELVIEW (test views). We will use 10 of the 32 views for every sample for training and predict 22 novel views. The 10 views form a random but consecutive range of angles ca. 240 degree.

In each application we also consider one application-specific competitor to solve the task. We use perfect classic reflectance maps for appearance representation [13], an upper bound with what could be estimated [28]. SGD is the common solution to learn appearance. For testing on real data, we need to select a single 2D input image for the RM. We use an oracle that selects the 2D image resulting in the lowest error. This is an oracle, impossible in practice, as the selection would required knowing the reference. For material segmentation, no clear baseline exists. We experimented with the method of Bell et al. [2], but concluded it is trained on semantic shapes (chairs imply wood etc.) which do not transfer to the abstract shapes we study (see supplemental materials for examples). Therefore, inspired by intrinsic images that also find consistent patches of reflectance [9], we simply employ k-means clustering in RGB-Normal space to do joint material estimation-and-segmentation. We will now look into the three specific applications.

5.2. Appearance representation

We study how well our approach can represent appearance per-se. Most distinctly, we propose to use a 4D appearance map while other works use 2D image representations of a reflectance map. To quantify the difference, we represent the SINGLEMATERIAL variant of our dataset as a common reflectance map, as well as using our appearance map.

To emulate a common reflectance map, which is defined in view space, we take the input image from the closest view from the training set as a source image. Every normal in the new view is converted to camera space of the new view and the same is done for the normal in the old view. We then copy the RGB value from the old view image to the new-view image that had the most similar normal. Note, that such a multi-view extension of RMs already is more than the state of the art that would use a single view. We call this method RM++.

Tbl. 1, top part, shows results as mean error across the data set. We see that for all data sets our method is a better representation in terms of having a lower DSSIM error. The difference in error is more pronounced for NOVELVIEW than for SAMEVIEW. A detailed plot of error distribution is seen in Fig. 6, left. This is, as classic reflectance map captures appearance for a fixed viewer location, for changing geometry, but does not generalize when the viewer moves. Arguably, classic RMs look qualitatively plausible without a reference, but only have low quantitative similarity (SSIM) in novel views.

5.3. Learning-to-learn appearance models

Here, we also follow the protocol described in Sec. 5.1. After having established the superiority of deep appearance maps to classic reflectance maps in the previous section, we use it as a competitor (SGD) for learning-to-learn. At best, our learning-to-learn network produces a network which is as good as running a full SGD pass.

The middle part of Tbl. 1 summarizes the outcome when executing the resulting ϕ on the test data set. We see that both approaches reproduce the appearance faithfully. For point lights, the mean DSSIM is .144 for SGD while it is .165 for network-based (Tbl. 1, middle part). Naturally, letting a network do the learning degrades quality, but only by a marginal amount, in this case 14.1 %. For environment map illumination, the mean DSSIM is increased from .164 to .173, a decrease by only 5 %. While being marginally worse, it is two orders of magnitude faster. Fig. 6 show the error distribution across the data set.

Table 1. Quantitative results on synthetic data. Rows are different combination of tasks and methods (three applications, two view protocols, our two methods). Columns are different data. Error is measured as mean DSSIM across the data set (less is better).

Task	View	Method	Error	
			Pnt	Env
Representation (Sec. 3.2)	Same	Our	**.105**	**.123**
		RM++	.143	.160
	Novel	Our	**.144**	**.164**
		RM++	.181	.193
Learn-to-learn (Sec. 3.3)	Same	Our	.106	.131
		SGD	.105	.123
	Novel	Our	.165	.173
		SGD	.144	.164
Segmentation (Sec. 3.4)	Same	Our	**.113**	**.122**
		kMeans	.132	.136
	Novel	Our	**.161**	**.154**
		kMeans	.172	.164

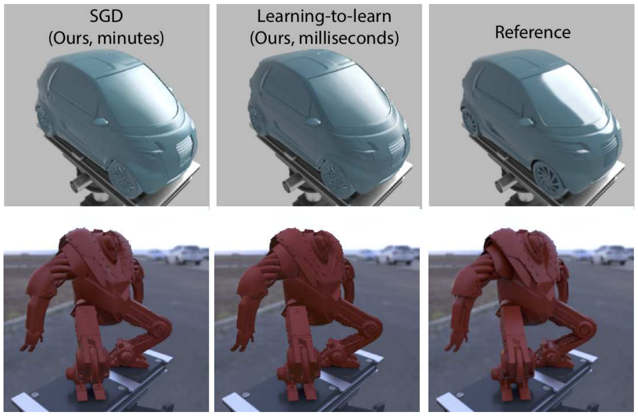

Figure 7. Results of our DAM representation trained using stochastic gradient descent **(1st column)**, our DAMs produced by our learning-to-learn network **(2nd column)** as well as a reference **(3rd column)** in a novel-view task.

A visual comparison is found in Fig. 7. We see that replacing the SGD computation of several minutes by a network, can produce a DAM that is qualitatively similar to both the SGD's result as well as to the reference. Overall, strength and sharpness of highlights that is already challenging for DAM per-se, seems to suffer a bit more by learning-to-learn, as also seen in Fig. 8.

5.4. Joint Material Estimation-and-Segmentation

Finally, we quantify the joint material-and-segmentation task from Sec. 3.4. We perform the same split as in the previous section, however, now on the MultiMaterial variant. For the re-synthesis to new views we use the ground truth segmentation in the new view (our method only produces the segmentation in all old views).

We here compare to a competitor, where the image is first segmented using k-means clustering on normals and RGB (same weight, as both have a similar range) and material is estimated for each segment in consecution.

Tbl. 1, bottom part shows the quantitative results and Fig. 9 the qualitative outcome. On average, we achieve an DSSIM error of .133 for PointLight and .122 for EnvironmentMap. The greedy method performs worse (.161 and .154), as it segments highlights into individual parts. While the method can understand that highlights belong "to the rest" of a material, sometimes they end up in different clusters, as seen in Fig. 8, middle and less in the bottom of Fig. 9.

5.5. Discussion

Typical failure modes are show in Fig. 8. For representation Fig. 8, left, the network can overshoot e. g., become darker than desired, for unobserved directions. More input images or a more effective GAN could suppress this. Sharp details cannot be represented by the cascade of functions of a small network. Fitting a network with more parameters might be required. For learning-to-learn Fig. 8, right, SGD might produce the right network, but the learned network overshoots. Similarly, highlights tend to be more blurry (not shown). For segmentation Fig. 8, middle, the rim highlight in the back of the character is purely white and apparently does not look enough like other highlights on blue to be understood. Consequently, it is assigned the metallic material, which is incorrect.

When the material approaches an arbitrarily complex illumination seen in a mirror, no network can capture all 4D variation anymore. This relation is shown in the inset plot (Fig. 10) where the vertical axis denotes error, which is decreasing when specular is decreased as well, along the horizontal axis.

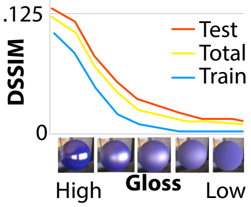

Figure 10. Relation of gloss and representation error.

5.6. Other Applications

DAMs can also be used for other tasks such as super-resolution, where we extract a DAM in low pixel resolution that can be transferred to a high resolution normal image and denoising of Monte Carlo path tracing, where we extract a DAM from noisy observations and re-generate the image from the DAM, removing the noise. Detailed evaluation, comparing to a state-of-the-art MC denoiser [3] and super-resolution [20] are found in the supplemental material.

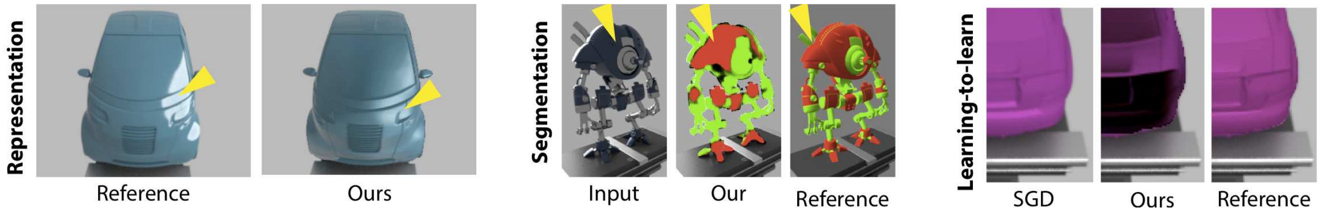

Figure 8. Failure modes for all three tasks: blurry highlights, split highlight segmentation and a overshooting DAM.

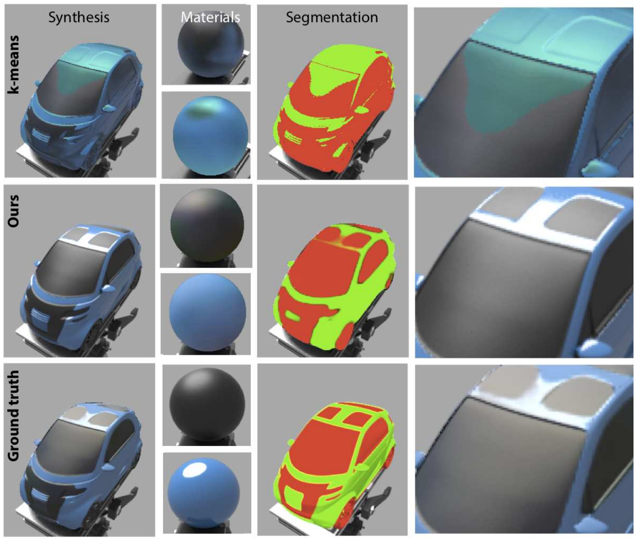

Figure 9. Results of joint material segmentation and estimation for two samples **(rows)**. In every part we show a re-synthesis, as well as two estimated materials and the resulting mask. The insets in the last row show that, while not all reflection details are reproduced, ours is free of color shifts around the highlights and mostly a low-frequency approximation of the environment reflected.

6. Real-world Evaluation

We have collected a second dataset of photographs of spherical objects with complex appearance (glossy objects under natural light). In particular we use 3 different materials and 5 different illuminations each with 5 images from registered views (Fig. 11) for training and 70 for testing. Please see the supplemental video for the animation. Transfer of appearance captured from a real world image sequence to other complex geometry is shown in Fig. 12.

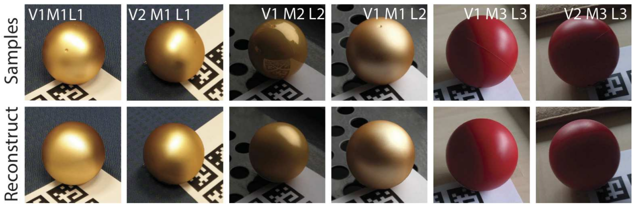

Figure 11. Real-world photo data and our reconstruction (from other views) of multiple materials (denoted M) in multiple illumination (L) from multiple views (V).

Table 2. DSSIM (less is better) error on real data.

	Same view		Novel view		
	OUR	RM++	OUR	RM++	RM
DSSIM Error	.069	.001	.079	.090	.127

Tbl. 2 summarizes the outcome for the representation task previously explored for synthetic data only. An example result is seen in Fig. 1, more are shown in the supplemental materials. Our method can estimate view-dependent appearance, unlike RM/RM++, from a small training set, but it can't fully reconstruct mirror-like reflections.

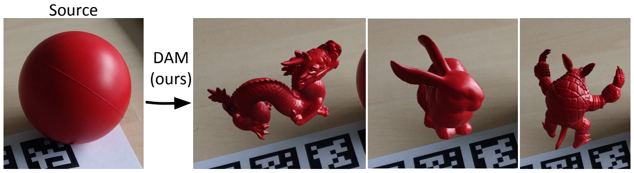

Figure 12. Transfer of appearance from a real video sequence (**left**) to new 3D shapes (**right**).

7. Discussion and Conclusion

We have proposed and explored a novel take on appearance processing that neither works on pixel-level IBR-like representations nor by extracting classic explicit reflectance and illumination parameters. Instead, we work on a deep representation of appearance itself, defined on a generalization of reflectance maps that works in world space where observations cover all directions. We have shown to enables effective reproduction, estimation by learning-to-learn and joint material estimation-and-segmentation.

In future work, we would like to generalize our approach to allow independent control of illumination and reflectance (BRDF) [10, 21, 16, 7], providing an improved editing experience. Equally, we have not yet explored the symmetric task of learning - how to learn segmenting appearance (Sec. 3.3).

References

[1] Anna Alperovich and Bastian Goldluecke. A variational model for intrinsic light field decomposition. In *ACCV*, pages 66–82, 2016. 3

[2] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *CVPR*, pages 3479–87, 2015. 2, 6

[3] Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A. Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. Nonlinearly weighted first-order regression for denoising monte carlo renderings. *Comp. Graph. Forum (Proc. EGSR)*, 35(4):107–17, 2016. 7

[4] Blender Foundation. *Blender - a 3D modelling and rendering package*, 2018. 5

[5] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, pages 145–156, 2000. 2

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proc. SIGGRAPH*, pages 11–20, 1996. 1, 2

[7] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37(128), 2018. 2, 8

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006. 2

[9] Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno, and Diego Gutierrez. Intrinsic images by clustering. *Comp. Graph. Forum (PROC. EGSR)*, 31(4):1415–24, 2012. 6

[10] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *ICCV*, 2017. 2, 3, 8

[11] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE PAMI*, 32(6):1060–71, 2010. 3

[12] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *CVPR*, page 6, 2017. 2, 3

[13] Berthold KP Horn and Robert W Sjoberg. Calculating the reflectance map. *Applied optics*, 18(11):1770–9, 1979. 1, 2, 3, 6

[14] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NIPS*, pages 667–675, 2016. 2, 3, 4

[15] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. A unified approach to prefiltered environment maps. In *Rendering Techniques*, pages 185–196. 2000. 2

[16] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias NieBner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation. In *ICCV*, 2017. 2, 8

[17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sci.*, 40, 2017. 2, 3

[18] Hendrik Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22(2):234–257, 2003. 3

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996. 2

[20] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, pages 136–144, 2017. 7

[21] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *ICCV*, 2017. 2, 8

[22] Stephen Lombardi and Ko Nishino. Reflectance and natural illumination from a single image. In *ECCV*, pages 582–95, 2012. 2

[23] Stephen R Marschner, Stephen H Westin, Eric PF Lafortune, Kenneth E Torrance, and Donald P Greenberg. Image-based BRDF measurement including human skin. In *Rendering Techniques*, pages 131–144. 1999. 2

[24] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. In *SIGGRAPH*, 2003. 2

[25] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[26] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. Deep shading: Convolutional neural networks for screen space shading. *Comp. Graph. Forum (Proc. EGSR)*, 36(4), 2017. 2

[27] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 1, 2

[28] Konstantinos Rematas, Chuong H. Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE PAMI*, 39(8):1576–1590, 2017. 2, 6

[29] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Efstratios Gavves, and Tinne Tuytelaars. Deep reflectance maps. In *CVPR*, 2016. 2

[30] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Trans. Graph.*, 34(4):111, 2015. 2

[31] Thomas Richter-Trummer, Denis Kalkofen, Jinwoo Park, and Dieter Schmalstieg. Instant mixed reality lighting from casual scanning. In *ISMAR*, 2016. 2

[32] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, 2000. 2

[33] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Trans. Graph.*, volume 21, pages 527–36, 2002. 2

[34] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 2, 3, 4

[35] Tuanfeng Y. Wang, Tobias Ritschel, and Niloy J. Mitra. Joint material and illumination estimation from photo sets in the wild. In *Proceedings of International Conference on 3DVision (3DV)*, 2018. selected for oral presentation. 3

[36] David H Wolpert and William G Macready. No free lunch theorems for search. Santa Fe Institute, 1995. 3, 4

[37] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proc. SIGGRAPH*, pages 287–96, 2000. 2

[38] Hongzhi Wu, Zhaotian Wang, and Kun Zhou. Simultaneous localization and appearance estimation with a consumer RGB-d camera. *IEEE TVCG*, 22(8):2012–2023, 2016. 2

[39] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, pages 286–301, 2016. 2