Deep Image-Based Relighting from Optimal Sparse Samples

ZEXIANG XU, University of California, San Diego
KALYAN SUNKAVALLI, Adobe Research
SUNIL HADAP, Adobe Research
RAVI RAMAMOORTHI, University of California, San Diego

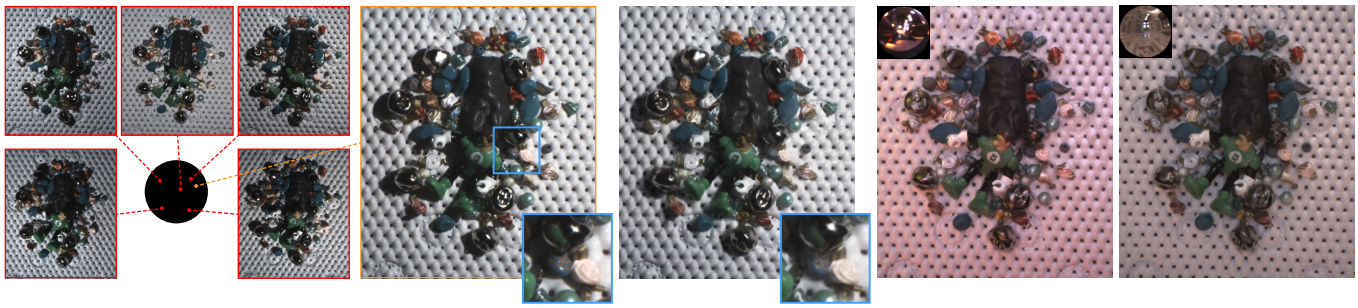


Fig. 1. We propose a learning-based method that takes only five images of a scene under directional lights (a, light directions marked on circle in red) and reconstructs its appearance (c) under a novel directional light in the upper hemisphere (marked in orange). Our method trains a fully-convolutional neural network to jointly learn the optimal input light directions and relighting function for any scene. The network can reconstruct even high-frequency patterns like specular shading and cast shadows (insets in c) and produces photorealistic relighting results that closely match the ground truth (b). Moreover, by generating images for every direction in the upper hemisphere, our method can be used to relight scenes under environment map illumination (d).

We present an image-based relighting method that can synthesize scene appearance under novel, distant illumination from the visible hemisphere, from only five images captured under pre-defined directional lights. Our method uses a deep convolutional neural network to regress the relit image from these five images; this *relighting network* is trained on a large synthetic dataset comprised of procedurally generated shapes with real-world reflectances. We show that by combining a custom-designed *sampling network* with the relighting network, we can jointly learn both the optimal input light directions and the relighting function. We present an extensive evaluation of our network, including an empirical analysis of reconstruction quality, optimal lighting configurations for different scenarios, and alternative network architectures. We demonstrate, on both synthetic and real scenes, that our method is able to reproduce complex, high-frequency lighting effects like specularities and cast shadows, and outperforms other image-based relighting methods that require an order of magnitude more images.

CCS Concepts: • **Computing methodologies** → **Rendering**;

Additional Key Words and Phrases: Image-based relighting, Illumination, Convolutional Neural Network, Sparse sampling, Appearance capture

Authors' addresses: Zexiang Xu, University of California, San Diego, zexiangxu@cs.ucsd.edu; Kalyan Sunkavalli, Adobe Research, sunkaval@adobe.com; Sunil hadap, Adobe Research, hadap@adobe.com; Ravi Ramamoorthi, University of California, San Diego, ravir@cs.ucsd.edu.

0730-0301/2018/8-ART126 $15.00
https://doi.org/10.1145/3197517.3201313

ACM Reference Format:
Zexiang Xu, Kalyan Sunkavalli, Sunil hadap, and Ravi Ramamoorthi. 2018. Deep Image-Based Relighting from Optimal Sparse Samples. *ACM Trans. Graph.* 37, 4, Article 126 (August 2018), 13 pages. https://doi.org/10.1145/3197517.3201313

1 INTRODUCTION

Rendering a scene under novel lighting is a long-studied vision and graphics problem with applications in visual effects, virtual and augmented reality, and product visualization for e-commerce. One approach to relighting is to reconstruct the geometry and material properties of the scene, and render this reconstruction under novel lighting. However, reconstruction is an extremely challenging problem, especially for scenes with complex geometry and reflectance.

Image-based relighting methods bypass reconstruction by directly modeling the scene's light transport function. Assuming distant illumination, the light transport function, $\mathbf{T}(\mathbf{x}, \boldsymbol{\omega})$, maps incident illumination from direction $\boldsymbol{\omega}$ to outgoing radiance at pixel x (towards the camera), and allows for the scene to be rendered under novel distant lighting as:

$$\mathbf{I}(\mathbf{x}) = \int_{\mathcal{L}} \mathbf{T}(\mathbf{x}, \boldsymbol{\omega}) L(\boldsymbol{\omega}) d\boldsymbol{\omega}, \tag{1}$$

where $L(\boldsymbol{\omega})$ is the radiance of the incident illumination from direction $\boldsymbol{\omega}$. The light transport function can be sampled by capturing images under different lighting conditions; for example, an image of the scene under a single directional light from direction $\boldsymbol{\omega}_j$, yields the sample: $\mathbf{I}_j(\cdot) = \mathbf{T}(\cdot, \boldsymbol{\omega}_j)$. Image-based relighting methods use a set of such samples, $\{(\mathbf{I}_j, \boldsymbol{\omega}_j) \mid j = 1, 2, \ldots, k\}$, to reproduce scene appearance, $\mathbf{I}_n$, under a novel light, $\boldsymbol{\omega}_n$. Because the light transport

function already combines all the interactions of incident illumination with scene geometry and materials, these methods can reproduce photorealistic lighting effects that are difficult to reconstruct and render.

Brute-force image-based relighting methods [Debevec et al. 2000] densely sample the light transport function by capturing hundreds to thousands of images of a scene; they can then relight the scene by interpolating these dense samples. However, the light transport function is known to be highly coherent [Mahajan et al. 2007; Nayar et al. 2004; Sloan et al. 2003], and this has been used to reconstruct it from a smaller number of images [Reddy et al. 2012; Wang et al. 2009] and relight images using lower-dimensional functions [Malzbender et al. 2001; Ren et al. 2015]. However, these methods still require tens to hundreds of images; this in turn requires considerable acquisition time, and often, specialized hardware.

In this work, we present a technique to render scene appearance under novel illumination from only five images. Previous image-based relighting methods have exploited coherence in a *single* light transport function. Instead, we leverage commonalities between *different* light transport functions, and estimate a single, non-linear, high-dimensional function that maps the appearance of *any* scene under a *sparse* set of pre-defined directional lights to the appearance of that scene under *any* directional light (in the upper hemisphere). Inspired by the success of deep learning at challenging appearance analysis tasks [Gardner et al. 2017; Kalantari et al. 2016; Rematas et al. 2016], we represent this function using a deep convolutional neural network (Sec. 3.1). We train this network — that we refer to as Relight-Net — using a large, synthetically rendered dataset consisting of scenes with procedurally generated shapes and real-world BRDFs (Sec. 3.3). Given five images of a scene under directional lights, Relight-Net can reproduce scene appearance under any directional light lying in the visible hemisphere.

The visual quality of Relight-Net's output is a function of the input directions used. Therefore, we design Sample-Net, a custom layer that chooses a sparse subset of a dense set of images. We prepend Sample-Net to Relight-Net to construct an end-to-end network that we train to *jointly learn the optimal input lighting directions and the relighting function* (Sec. 3.2). We present an extensive evaluation of our method, including an empirical analysis of reconstruction quality, optimal lighting configurations for different ranges of incident illumination, and alternative network architectures (Sec 4.1). We also propose a refinement method that makes Relight-Net robust to small deviations from the optimal input light directions that are likely to occur in real-world capture scenarios (Sec. 4.2).

As shown in Figs. 1, 12 and 17, our method generates photorealistic results for real scenes with complex high-frequency effects like cast shadows and sharp specularities. The visual quality of our results — generated from just five images — is better than those from previous image-based relighting methods that require an order of magnitude more images (see Figs. 12 and 13). Thus, our method significantly reduces the acquisition time and complexity for image-based relighting methods and takes a step towards making them more practical.

2 RELATED WORK

Dimensionality of Light Transport. While changes in illumination can lead to drastic changes in the images of a scene, previous work has shown that these images often lie in low-dimensional subspaces. For example, images of a Lambertian scene are known to lie on a low-dimensional manifold [Basri and Jacobs 2003; Belhumeur and Kriegman 1998; Ramamoorthi and Hanrahan 2001; Shashua 1997; Sunkavalli et al. 2010]. Even scenes with complex geometry and reflectance have been shown to have low-dimensional light transport in local regions [Mahajan et al. 2007], a fact that has been exploited for fast rendering [Ng et al. 2003; Sloan et al. 2003] and relighting [Nayar et al. 2004]. These techniques use linear analysis (globally or in local regions) to show the low dimensionality of light transport for a single scene. By exploiting correlations in light transport across scenes using a non-linear CNN-based representation, our work dramatically reduces the number of images required for scene relighting.

Relighting from Sparse Samples. While brute-force image-based relighting methods densely sample the light transport function [Debevec et al. 2000], recent methods have leveraged the coherence of the light transport function to reconstruct it using fewer samples. One such approach is to use specially designed illumination patterns during capture [Matusik et al. 2004; Peers and Dutré 2005; Peers et al. 2009; Reddy et al. 2012]. Other methods reconstruct the light transport matrix from a sampled subset of rows or columns [Fuchs et al. 2007; Wang et al. 2009]. However, these methods still require hundreds of images, and special acquisition systems to create the desired illumination. In contrast, our method can relight scenes from only five images under directional lighting.

Polynomial texture maps (PTM) [Malzbender et al. 2001] model per-pixel radiance values as polynomial functions of lighting directions. These functions are fit to (approximately 50) captured images and used to render the scene under novel lighting. Ren et al. [2015] use a similar scheme, with the difference that they use shallow neural networks instead of polynomials. They demonstrate impressive results for scenes with complex light transport, but require hundreds of images to achieve this. In contrast, we exploit spatial and angular coherence in light transport across scenes, and learn a more complex, non-linear relighting function to achieve image relighting with only five samples. Figure 12 shows that our results have better visual quality than PTM run on 60 images.

BRDF estimation techniques reconstruct BRDFs from images captured under varying illumination. Nielsen et al. [2015] and Xu et al. [2016] use a linear data-driven BRDF model to derive optimal light directions for BRDF estimation from sparse samples. We propose a technique to learn optimal lighting directions for high-quality scene relighting with a non-linear CNN-based reconstruction method.

Photometric Stereo-based Scene Reconstruction. Our acquisition setup is similar to Photometric Stereo methods which reconstruct surface geometry (and reflectance) from images of a scene under varying illumination [Woodham 1980]. While recent techniques can handle non-Lambertian BRDFs [Chandraker 2016; Goldman et al. 2010; Hui and Sankaranarayanan 2017; Oxholm and Nishino 2016] they either assume homogeneous BRDFs or require hundreds of image to reconstruct shape and spatially-varying BRDFs. In addition, Photometric Stereo methods often do not consider cast shadows, global illumination, and other light transport effects. In contrast, our method is able to reproduce these effects from fewer samples and outperforms Photometric Stereo-based methods (see Fig. 12).

Deep Learning for Appearance Analysis and Synthesis. Recently, deep learning-based methods have been successfully applied to

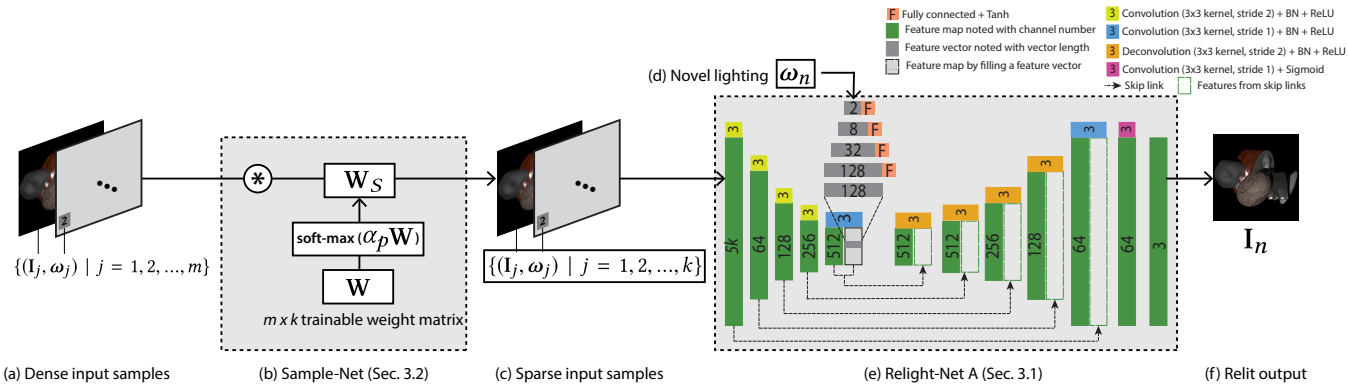


Fig. 2. An overview of our network. We stack a dense set of m input images and light directions (a) into a $5m$-channel input that is passed to Sample-Net (b, Sec. 3.2). Sample-Net consists of a trainable weight matrix, $\mathbf{W}$, that is multiplied by temperature parameter, α_p and passed through a softmax layer. This constructs a sparse sampling matrix, $\mathbf{W}_s$, that multiplies the dense input to produce the sparse $5k$-channel sample set (c) that is the input to Relight-Net A (e, Sec. 3.1). Relight-Net A is a fully-convolutional encoder-decoder; the encoder downsamples the input samples to an intermediate representation. We pass the output light direction, ω_n, (d) through fully-connected layers and replicate and concatenate it to the intermediate representation. The decoder upsamples this to recover the output relit image (f). We use skip links to introduce high-frequency features into the output. We train Sample-Net and Relight-Net jointly to learn both the optimal samples and the relighting function for relighting any scene (Sec. 3.4). At test time, we only use Relight-Net to relight the input sparse samples (c) under the input novel lighting (d).

inverse rendering and scene reconstruction problems such as reflectance map and illumination estimation [Gardner et al. 2017; Georgoulis et al. 2017; Hold-Geoffroy et al. 2017; Rematas et al. 2016], reflectance capture [Li et al. 2017; Liu et al. 2017], and depth and normal estimation [Bansal et al. 2016; Eigen and Fergus 2015]. These methods make simplifying assumptions about the scene (for example, considering only a single object) to make the reconstruction tractable. We bypass reconstruction, and directly generate relit images for complex scenes. Deep networks have also been used for view interpolation for relatively unstructured cameras [Flynn et al. 2016] and light field cameras [Kalantari et al. 2016]. Our work assumes a fixed viewpoint and attempts to interpolate/extrapolate lighting.

3 LEARNING IMAGE-BASED RELIGHTING

Given a small set of images of a scene under individual light sources, we want to render the scene under novel lighting. We assume that the scene is imaged from a fixed viewpoint and that the illumination is distant. We also assume that illumination from behind the scene makes a minimal contribution to appearance and can be ignored. Under these assumptions, the light transport matrix, $\mathbf{T}(\mathbf{x}_i, \omega_j)$, represents the proportion of incident radiance from direction, ω_j (sampled from the upper hemisphere, $\mathcal{L}$) that reaches pixel $\mathbf{x}_i$. Images of the scene under single directional lights represent column-wise samples of the light transport matrix, i.e., $\mathbf{I}_j = \mathbf{T}(:, \omega_j)$.

Given a set of k such samples — images of the scene, $\mathbf{I}_1, \mathbf{I}_2, ...\mathbf{I}_k$, captured under predefined directional lights, $\omega_1, \omega_2, ..., \omega_k$ respectively — the goal of our work is to reconstruct the image, $\mathbf{I}_n$, that would be produced by a novel directional light, ω_n, via a relighting function, $\Phi(\cdot)$:

$$\mathbf{I}_n = \Phi(\omega_n; \mathbf{I}_1, \omega_1; \mathbf{I}_2, \omega_2; ...; \mathbf{I}_k, \omega_k) = \Phi(\omega_n, \mathbf{S}_1, \mathbf{S}_1, ..., \mathbf{S}_k). \quad (2)$$

We hypothesize that $\Phi(\cdot)$ is *scene-agnostic* and can transform sparse input samples, $\mathcal{S} = \{\mathbf{S}_j\} = \{(\mathbf{I}_j, \omega_j)\}$, of *any* scene into a rendering of that scene in novel lighting. We believe this is possible because light transport is highly coherent; our formulation enables this by allowing the radiance at a pixel to potentially be a function of the entire scene under all the sparse light directions.

We model the relighting function, $\Phi(\cdot)$, as a deep convolutional neural network (CNN) that we refer to as **Relight-Net**. We train Relight-Net using a large synthetic dataset consisting of procedurally generated shapes rendered with complex spatially-varying BRDFs, and demonstrate that it can reconstruct high-frequency light transport effects like specularities and cast shadows. Relight-Net is illustrated in the right half of Fig. 2 and described in Sec. 3.1.

The quality of the reconstruction from Relight-Net depends on the pre-defined lighting directions that we use as input samples. Intuitively, the ability to generalize to new lighting will be limited if the input directions all lie very close together, and will improve as they span the full incident hemisphere. Therefore, we also propose a scheme to learn the optimal input sample directions that lead to the best relighting results. Specifically, we densely sample the space of input light directions, and design a layer that selects a sparse set of these directions. We call this layer **Sample-Net** and describe it in Sec. 3.2 (also see left half of Fig. 2). We prepend Sample-Net to Relight-Net to construct an end-to-end network that is trained jointly to estimate both the optimal light directions and the corresponding relighting function.

3.1 Learning to Relight: Relight-Net

At the core of our method is Relight-Net, a deep fully-convolutional neural network that approximates Eqn. 2. We explore two architectures for Relight-Net — the first is a conventional encoder-decoder architecture, while the second disentangles the direct and global illumination components of the scene.

Relight-Net A. Our first network architecture, Relight-Net A is designed to directly generate a relit image from sparse input samples, $\mathcal{S} = \{(\mathbf{I}_j, \omega_j) \mid j = 1, ..., k\}$. To pass the input light directions $\omega_j = (s_j, t_j)$ (2D coordinates of the direction vector projected to the

$z = 0$ disk) to the network, we construct 2-channel constant images with the same resolution as the input images and s and t in each channel respectively. Concatenating this to the input RGB image yields a 5-channel input per-sample; stacking the k samples leads to a $5k$-channel input to Relight-Net A.

As illustrated in Fig. 2, Relight-Net A uses a U-net-style encoder-decoder architecture [Ronneberger et al. 2015]; the encoder takes the $5k$-channel input, passes it through a series of convolutional layers (with stride 2 for downsampling), each followed by batch normalization (BN) and ReLU layers. The target lighting direction, $\boldsymbol{\omega}_n$, is passed through fully-connected layers (with tanh activation layers after each linear operation) to expand the 2-dimensional vector $\boldsymbol{\omega} = (s, t)$ into a 128-dimensional feature vector. We replicate this feature vector spatially to construct a 128-channel feature map that is concatenated with the encoder output. The decoder convolves the concatenated encoder output and upsamples the features with deconvolution (transpose convolution) layers, where both convolution and deconvolution are followed by BN and ReLU layers. We use skip connections from the encoder to the decoder to improve per-pixel details in the output. The decoder ends with 2 convolution layers followed by a sigmoid activation to produce the relit image. We train the network using an L_2 loss on the output images, $L_A = \|\mathbf{I}_n - \mathbf{I}_n^{gt}\|_2$, where $\mathbf{I}_n^{gt}$ is the ground truth image rendered with a directional light source at $\boldsymbol{\omega}_n$.

The structure of Relight-Net A allows it to leverage two forms of coherence in a transport matrix: the convolutional layers exploit spatial coherence by aggregating over the network's receptive field, and combining feature maps across channels exploits correlations over lighting directions. This allows it to handle diffuse and specular reflectance, shadowing and other global illumination effects.

Relight-Net B. Relight-Net A is designed to directly regress a relit image from input samples, and does not have any rendering-specific constraints. We design an alternative architecture — Relight-Net B — to evaluate if the explicit inclusion of rendering priors can improve the relighting results. Specifically, we know that the appearance of a scene under a single directional light, $\boldsymbol{\omega}_n$, can be represented as a sum of direct and global illumination components: $\mathbf{I}_n = \mathbf{I}_n^d * \mathbf{V}_n + \mathbf{I}_n^g$, where $\mathbf{I}_n^d$, $\mathbf{V}_n$, and $\mathbf{I}_n^g$ are the direct component, per-pixel visibility map w.r.t. $\boldsymbol{\omega}_n$, and global illumination components respectively. We train Relight-Net B to explicitly decode each of these components.

As shown in Fig. 3, Relight-Net B consists of a single encoder, connected with three separate decoders. The three decoders generate $\mathbf{I}_n^d$, $\mathbf{V}_n$, and $\mathbf{I}_n^g$, which are then combined to reconstruct $\mathbf{I}_n$. The encoder and decoders are identical to those used in Relight-Net A.

Since we use synthetic rendered data to train the network, we can generate ground truth data for direct illumination images, visibility maps, and final relit images and use them as supervision. We render direct component, $\mathbf{I}_n^{d,gt}$, by computing local per-pixel shading without considering visibility. We construct the visiblity map, $\mathbf{V}_n^{gt}$, using shadow ray casting. The final loss of Relight-Net B is the sum of three L_2 losses of the three supervised terms:
$L_B = \|\mathbf{I}_n^d - \mathbf{I}_n^{d,gt}\|_2 + \|\mathbf{V}_n - \mathbf{V}_n^{gt}\|_2 + \|\mathbf{I}_n - \mathbf{I}_n^{gt}\|_2$.

3.2 Learning Optimal Light Samples: Sample-Net

Relight-Net produces relit images from a set of sparse input samples captured under *fixed, predefined* directions, and this form of structured input contributes to the quality of the results. However, the specific choice of directions to use can have a substantial bearing on relighting quality, and it is not clear apriori what the optimal lighting configuration is. One choice for learning the optimal lighting directions is to regress these parameters using the network. However, changes in light direction can lead to complex changes in scene appearance that are challenging to model and are not differentiable w.r.t. to the lighting (e.g., changes in shadows). Instead, we densely sample the domain of incident illumination (i.e., the upper hemisphere, $\mathcal{L}$), pre-render images of the training scenes under these lights, and pose the problem of estimating the optimal samples as one of *selecting* a sparse subset of these dense samples.

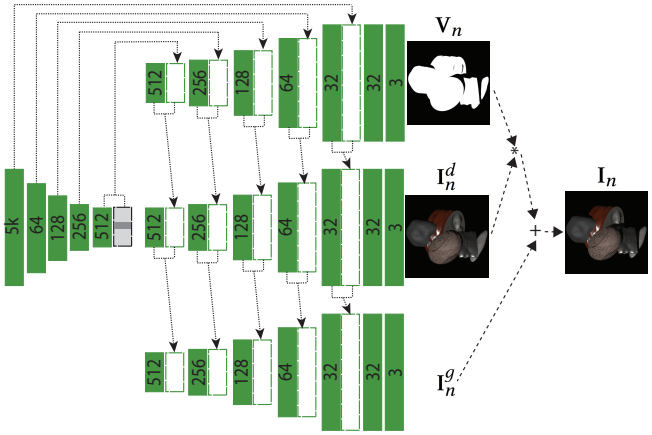


Fig. 3. Relight-Net B. Similar to Relight-Net A, we use an encoder-decoder architecture. However, Relight-Net B has a single encoder and three separate decoders to reconstruct the direct illumination, visibility map, and indirect illumination images, that are then combined to reconstruct the relit output. We use skip links from the encoder to all the decoders to recover high-frequency details.

Let the dense set of input samples be given by $\mathcal{D} = \{(\mathbf{I}_j, \boldsymbol{\omega}_j) \mid j = 1, \ldots, m\}$. By vectorizing each $(\mathbf{I}_j, \boldsymbol{\omega}_j)$ pair and stacking these samples, we can construct the $5p \times m$ dense sample matrix $\mathbf{D}$, where p is the number of pixels in the input images. Selecting a subset of these samples can be done as:

$$\mathbf{S} = \mathbf{D}\,\mathbf{W}_S, \tag{3}$$

where $\mathbf{W}_S$ is a $m \times k$ binary matrix ($k << m$), where each column has a single 1 entry (corresponding to the sample from $\mathbf{D}$ that is "selected").

Sample-Net is a trainable $m \times k$ $\mathbf{W}$ matrix that post-multiplies the dense input samples to create a sparse set of samples. However, we need to enforce that this matrix is binary and each column only has a single 1. Inspired by Chakrabarti et al. [2016], we do this by applying a `softmax` layer to each column of $\mathbf{W}$:

$$\mathbf{W}_S = \text{softmax}(\alpha_p \mathbf{W}), \tag{4}$$

where α_p is a scalar parameter that gradually increases from 1 to ∞ during each epoch, p, of the training process. Because of the form of the softmax layer, $\sigma(\mathbf{z})_j = \exp(z_j)/\sum \exp(z_k)$, using a larger α_p makes each column of $\mathbf{W}_S$ sparser. We initialize $\mathbf{W}$ with all 1s. As a result, in the early stages of training, samples in $\mathbf{S}$ are a linear combination of samples in $\mathbf{D}$, but as α_p goes to infinity, each column of $\mathbf{W}_S$ gradually converges to a single non-zero element that

corresponds to the chosen optimal sample (see Fig. 6 in Section 4.1). We use a quadratic model $\alpha_p = \beta p^2$, where β is a tunable hyper parameter.

While Eqn. 3 vectorizes each input sample into a $5p \times 1$ vector, in practice we represent them as 5-channel inputs and apply the same value of $\mathbf{W}_s$ to each channel of the sample. Figure 2 illustrates how we combine Sample-Net and Relight-Net. In Sec. 3.4 we describe how we train them jointly.

3.3 Generating training data

We train our networks with a synthetic rendered dataset; this allows us to control all aspects of our training scenes and also lets us create ground truth data for relit images and intermediate results (like those needed by Relight-Net B).

We generate primitive shapes (cubes, ellipsoids, and cylinders) with random parameters, and apply height-fields with varying frequencies of variation. This gives us a large set of random shapes; we construct the scene geometry by combining multiple (1 to 9) shapes after applying random translations and rotations. We create 600 scenes using this method with 500 for training and 100 for testing.

We use material definitions from the Adobe Stock 3D material dataset[1] — a dataset of 694 realistic spatially-varying BRDFs (SVBRDFs). This dataset uses a physically-based microfacet BRDF model [Burley 2012] (that uses the GGX distribution [Walter et al. 2007]) and each material is represented by high resolution (4096×4096) diffuse maps, roughness maps and normal maps. We texture the shapes with random crops from these SVBRDFs (see Fig. 4). We separate the SVBRDFs into training set (594 materials) and test set (100 materials); the materials used in the synthetic test scenes are thus never seen during training.

We render 512×512-resolution training and testing images with Mitsuba [Jakob 2010] using bidirectional path tracing [Lafortune and Willems 1993] with 196 samples. To make our method applicable to low dynamic range images captured by conventional cameras, we apply a gamma of 2.2 and clip the images at 1. We use a combination of Mitsuba's `mixturebsdf`, `roughconductor`, `diffuse`, and `normalmap` plugins to render the materials.

Figure 4 illustrates some of our rendered scenes. While the composition of these scenes may not be realistic, note that they locally exhibit the kinds of complex light transport that are present in the real world, including complex surface reflections, cast shadows, and inter-reflections. As we show in Figs. 1, 12 and 17, this allows us to learn a relighting function that generalizes well to real scenes.

3.4 Training Relight-Net and Sample-Net

As discussed in Sec. 3.2 (and illustrated in Fig. 2), Sample-Net is designed to be jointly trained with Relight-Net; given a dense set of scene samples, Sample-Net selects a sparse subset that can be input to Relight-Net to produce the relit result. To train them jointly, we start by densely sampling the incident illumination domain, $\mathcal{L}_\theta$ — a θ-degree cone towards the viewpoint as shown in Fig. 5. This gives us a large set of discrete lights $\Omega_\theta = \{\boldsymbol{\omega}_j | j = 1, 2, .., m_\theta\}$. We render each training scene, i, under every light in this set to create the dense input samples $\mathcal{D}_{i,\theta} = \{(\mathbf{I}^{gt}_{i,j}, \boldsymbol{\omega}_j) | \boldsymbol{\omega}_j \in \Omega_\theta\}$. We train the combined Sample-Relight-Net in an end-to-end fashion to minimize the Relight-Net loss function across all scenes and all output light

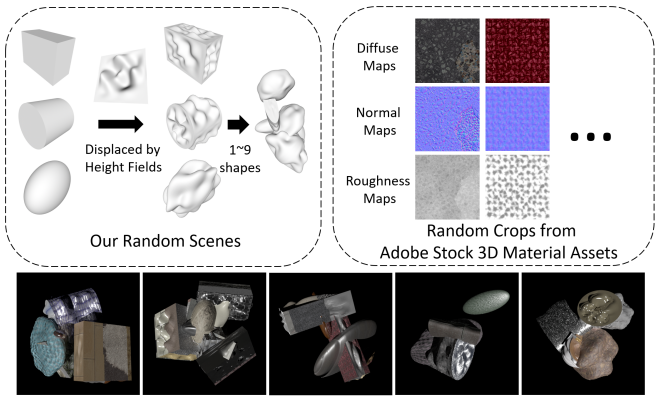


Fig. 4. Training data. Our scenes consist of multiple (1−9) random primitive shapes that are augmented with varying levels of height fields (top left). We texture these shapes with SVBRDFs from the Adobe Stock 3D Material dataset (top right) and render them using Mitsuba (bottom).

directions:

$$L(\mathbf{W}_s, \Phi) = \sum_i \sum_{\boldsymbol{\omega}_j \in \Omega_\theta} \|\Phi(\boldsymbol{\omega}_j; \mathbf{D}_{i,\theta}\ \mathbf{W}_S) - \mathbf{I}^{gt}_{i,j}\|_2, \tag{5}$$

where $\mathbf{D}_{i,\theta}$ is constructed from $\mathcal{D}_{i,\theta}$ as described in Sec. 3.2. This loss function evaluates the error of reconstructing images under *every* light in Ω_θ from images under only k lights from Ω_θ.

We crop 10 128×128 patches from each rendered image $\mathbf{I}^{gt}_{i,j}$ giving us 5000 scene-patches for our training. Each training scene-patch has a corresponding $\mathcal{D}_{i,\theta}$. Since training Sample-Net requires loading $\mathcal{D}_{i,\theta}$ completely, we are only able to train with a small batch size. This in turn implies swapping $\mathcal{D}_{i,\theta}$ out repeatedly and can lead to significant I/O overheads. Instead we organize training as follows: in each batch, we load $\mathcal{D}_{i,\theta}$ for 4 random scenes and randomly pick 18 images $(\mathbf{I}^{gt}_{i,j}, \boldsymbol{\omega}_j)$ from each $\mathcal{D}_{i,\theta}$ as targets for Relight-Net to reconstruct. This forms a batch of 4 scenes for Sample-Net training and 72 images for Relight-Net training. We use ADAM with 0.0001 as the learning rate for joint training. β from 5 to 8 generally works well, and we use $\beta = 6$. We find that our networks typically converge after 16 epochs. Our final learned models, scenes, rendered images and the code for generating them are released on the project website.[2]

Since Relight-Net is fully convolutional, at test time we can apply it to arbitrary resolution images, although it only considers appearance within a 128×128 window size. Moreover, while Relight-Net has been trained using only the discrete lights in Ω_θ, we show that it can be used to relight using any directional light on the continuous domain $\mathcal{L}_\theta$.

4 ANALYSIS ON SYNTHETIC DATA

4.1 Analysis of Relight-Net and Sample-Net

In this section, we present analysis and empirical evaluations of the different components of our network. Unless otherwise specified, we use the Relight-Net A for testing. Later in the section, we compare Relight-Net A and Relight-Net B.

[1] https://stock.adobe.com/3d-assets

[2] http://viscomp.ucsd.edu/projects/SIG18Relighting

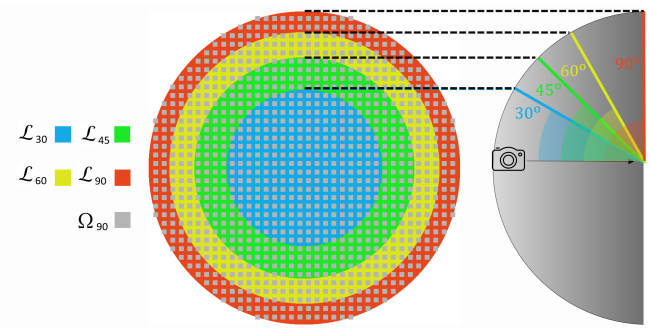


Fig. 5. Incident illumination domains and samples. As shown on the right, we model scene appearance under directional lights that lie on $\mathcal{L}_\theta$, a θ degree cone on the hemisphere pointing towards the viewing direction. On the left we illustrate the hemisphere and different light domains, $\mathcal{L}_{30}$, $\mathcal{L}_{45}$, $\mathcal{L}_{60}$, $\mathcal{L}_{90}$, in a 2D plane. We also show Ω_{90}, the densely sampled 1052 discrete directions from $\mathcal{L}_{90}$. Other Ω_θ are constructed from Ω_{90} as $\Omega_{30} = \Omega_{90} \cap \mathcal{L}_{30}$, $\Omega_{45} = \Omega_{90} \cap \mathcal{L}_{45}$, $\Omega_{60} = \Omega_{90} \cap \mathcal{L}_{60}$.

Light domain vs. number of sparse samples. By training Sample-Net and Relight-Net to minimize Eqn. 5, we can learn to relight a scene from any light direction in $\mathcal{L}_\theta$. To investigate the effect of the size of this domain (in terms of cone angle, θ) on the performance of our network, we train our network on four light domains: $\theta = \{30, 45, 60, 90\}$. We create Ω_{90} by uniformly sampling 38 values for the (s, t) coordinates of light directions in the domain $(-0.952, 0.952)$ and rejecting samples outside the unit disk. This leads to $m_{90} = 1052$ distinct light directions in $\mathcal{L}_{90}$. Ω_{30}, Ω_{45} and Ω_{60} are subsets of Ω_{90}, with $m_{30} = 256$, $m_{45} = 540$ and $m_{60} = 804$ directions, respectively. Details about $\mathcal{L}_\theta$ and Ω_θ are shown in Fig. 5.

In addition to the size of the illumination domain, the quality of our reconstructions depends on the number of sparse samples that are input to Relight-Net. In particular, as the size of the illumination domain increases, we expect that we would need more samples to preserve reconstruction quality. Therefore, we analyze the performance of our full Sample-Net-Relight-Net architecture for 12 different light domain size/sample configurations: $\theta = 30, k = \{2, 3, 4\}$; $\theta = 45, k = \{4, 5\}$; $\theta = 60, k = \{5, 6, 7\}$; and $\theta = 90, k = \{5, 6, 7, 8\}$.

Learnt optimal samples. Over the course of our joint training process, Sample-Net gradually converges to k optimal samples. In Fig. 6, we illustrate this for $\theta = 90$ and $k = 5$ samples. $\mathbf{W}_S$ starts off as a mixing of many samples and gradually converges to 5 optimal samples. As we would expect intuitively, these samples are distributed over $\mathcal{L}_{90}$.

Figure 7 indicates the learnt optimal directions for 4 representative networks, one for each light angle setting. We can see that when $k = 3, 4$, the optimal directions are spread in a circle around the center of the cone; setting $k = 5$ adds a direction near the center of the cone, i.e., nearly collocated with the viewpoint. Also note that all optimal directions are not placed at the edge of lighting domain, indicating that Sample-Net chooses directions that allow Relight-Net to both interpolate and extrapolate the input samples to produce relit results. Note that Sample-Net could have converged to a local minima, as is often the case with deep networks. However, in practice we have found that these directions lead to better reconstructions than arbitrarily chosen directions and other sampling

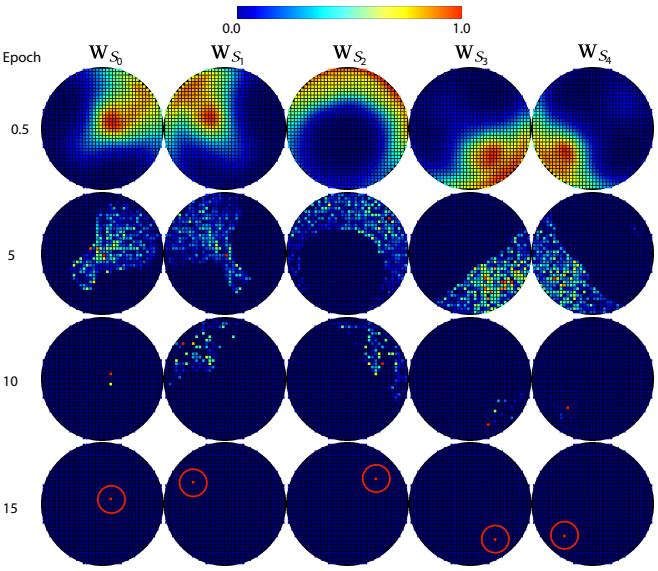


Fig. 6. Evolution of the optimal sparse samples during joint training for ($\theta = 90$, $k = 5$) ; each column represents the values from one column of $\mathbf{W}_S$. Starting from a flat distribution, each column of $\mathbf{W}_S$ gradually becomes peakier, till it converges to a single sample at epoch 15.

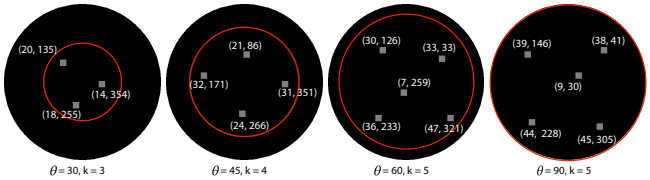


Fig. 7. Learnt optimal directions for several lighting configurations. We represent directions using the standard (θ, ϕ) spherical parameterization.

strategies based on heuristics, as we will discuss shortly. Optimal directions for the remaining 8 (θ, k) configurations are shown in the supplementary document.

Reconstruction quality. We use our trained networks to relight the 100-scene test set under all the lighting directions in the trained light domain, and aggregate the errors. We perform this analysis for different choices of (θ,k) and illustrate the results in Fig. 8. From these error distributions, we can make the following observations about Relight-Net: 1) it produces very low reconstruction errors for lights close to the input samples; 2) it produces high-quality results for interpolated light samples, i.e., output light directions that lie within the convex hull of the input light directions; 3) while it is able to do extrapolate to relight scenes under lights that lie outside the convex hull of the sparse input samples, the errors are larger that those for interpolation. While each scene might have its own optimal sampling directions based on its geometry and reflectance properties, the directions in Fig. 7 are optimal for all scenes. In addition, the optimal directions are chosen to let Relight-Net trade-off errors in interpolation and extrapolation scenarios.

As expected, the reconstruction error is lower for smaller light domains; for example, the network trained for ($\theta = 45, k = 5$)

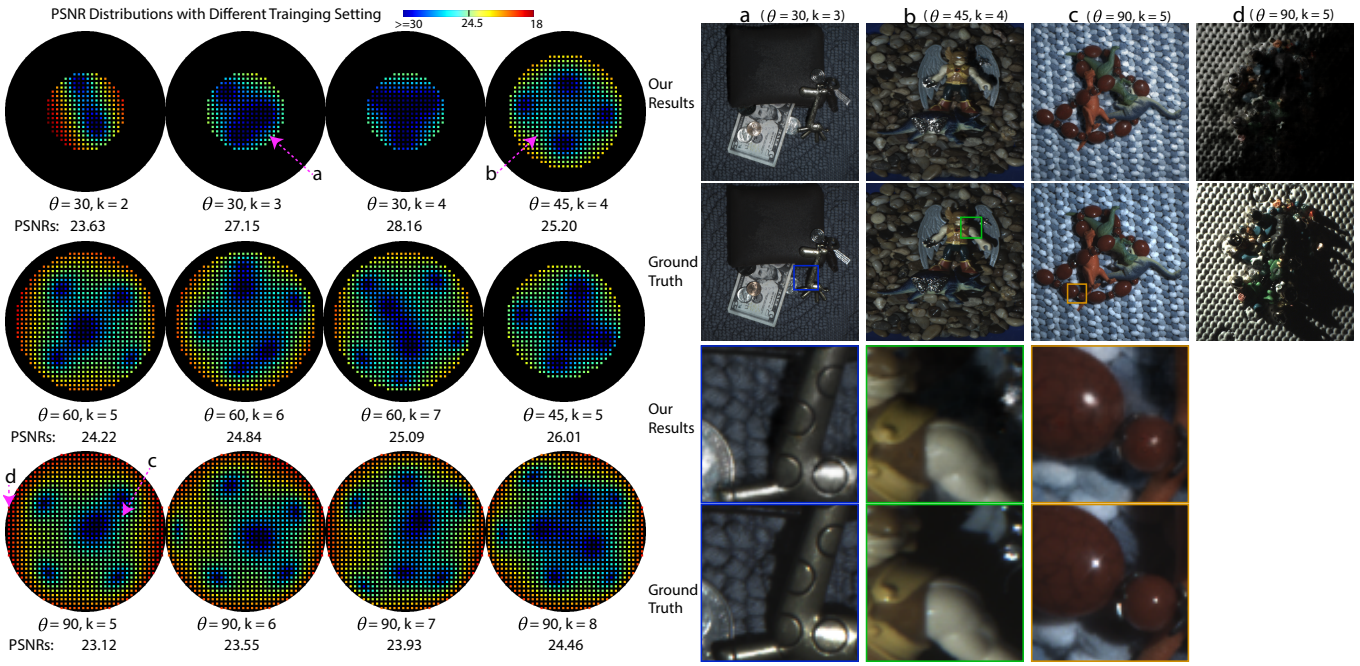


Fig. 8. Reconstruction errors for different lighting configurations. On the left, we visualize reconstruction quality (in PSNR) for each test light direction aggregated over all our test scenes. Errors are lowest for directions close to the input samples, and are quite low in the convex hull of the input samples even for directions away from input samples. The average PSNR across all light directions is listed under each figure. On the right, we show relighting results for 4 real scenes (a,b,c,d) captured with different light setups and compare them against ground truth. The corresponding output directions are marked on the left. Our reconstructions are very accurate for ($\theta = 30$, $k = 3$), ($\theta = 45$, $k = 4$), and most light directions in ($\theta = 90$, $k = 5$). Note that the ($\theta = 90$, $k = 5$) setup can fail for single directional lights at extreme grazing angles (d), but this has minimal impact when integrating over an environment map as seen in Figs. 1 and 17.

has a PSNR of 26.01 vs. 23.12 for ($\theta = 90, k = 5$). Our network with ($\theta = 45, k = 4$) produces near-photorealistic results for most directions in $\mathcal{L}_{45}$. For $\theta = 90$, i.e., the entire upper hemisphere, the lighting setup we have showcased in this paper, ($\theta = 90, k = 5$), produces accurate results across much of the light domain. This network might blur some high-frequency effects like sharp shadows and small specularities. However, these issues are most evident when we render the scene under high-frequency directional lights; rendering the scene under environment map illumination leads to results that are perceptually indistinguishable from ground truth images (see Fig. 15). Moreover, using additional samples, e.g., $k = 8$, also improves performance. These experiments suggest that we can use Sample-Net to further optimize the sample configuration for specific scenes or capture scenarios.

Comparisons against alternative sampling strategies. To evaluate the quality of our learnt samples, we compare them against heuristics-based strategies that produce well-distributed samples. We choose two methods — random dart throwing and k-means clustering, and evaluate them on the ($\theta = 90, k = 5$) configuration.

To ensure that random dart throwing leads to well-distributed samples, we specify a minimal threshold on the distance between two samples. We also apply the same minimal threshold on the distance from the boundary of the light domain, i.e., the grazing angle. Without this condition, we found that samples tend to converge towards the boundary of the domain (which has more samples) which leads to poor relighting performance. Since we do not know the threshold distance apriori, we generate sample sets with gradually increasing thresholds and pick the value, 40°, which allows only five samples. Our other baseline uses k-means clustering to group the 1052 samples in Ω_{90} into 5 clusters. We found that k-means clustering generally converges to two types of distributions: either one central direction with four directions distributed around it, or five directions around the center of the light domain (see Fig. 9(c,d)). We randomly select two results of random dart throwing and two of k-means clustering (from the two representative distributions). We train the Relight-Net by using these samples as the input and using the same 5000 scene-patches $\mathcal{D}_{i,\theta}$ as training data.

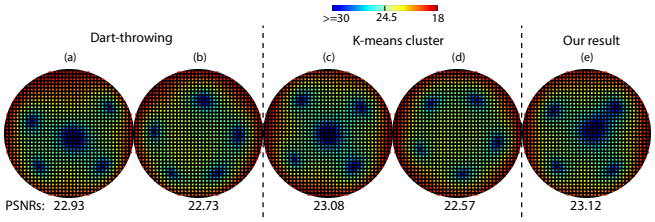


Fig. 9. Comparison with two random dart throwing sample sets (a,b), two representative k-Means clustering samples (c,d), and our samples (e). Our samples produce higher average PSNR.

We test these trained networks on the 100-scene test and compare the reconstruction error. As shown in Fig. 9, our Sample-Net samples significantly outperform the random dart throwing ones (Fig. 9 (a,b)).

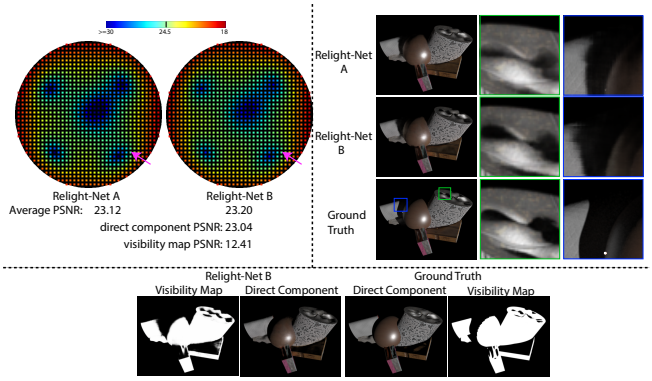
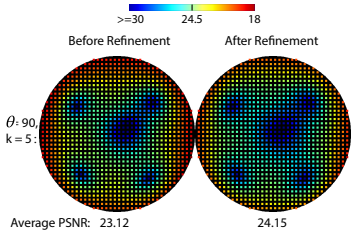


Fig. 10. Relight-Net A vs. Relight-Net B. We compare relighting errors across all light directions for our two Relight-Net architectures for the ($\theta = 90$, $k = 5$) lighting configuration using the same optimal light directions (top left). Overall PSNR and PSNR for each Relight-Net B component are also listed. Relight-Net B has marginally better performance but visual inspection (top right) for one scene and light direction (marked by the arrow on the left), shows that it produces shadows that are better sometimes (green inset) and inaccurate at other times (blue inset). We also show the visibility map and direct component from Relight-Net B (bottom).



Average PSNRs for inputs away from the optimal

All	1-5 degree	5-10 degree
23.82	24.02	23.59

Fig. 11. Evaluation of network refinement. On the left we compare reconstruction errors for optimal inputs for Relight-Net before and after refinement with ($\theta = 90$, $k = 5$). Refinement improves the results because it has been trained on a larger dataset. Moreover, the refined network also performs well on two scenarios of non-optimal input directions that are 1–5° and 5–10° off the optimal. As the average PSNR (computed across our entire test set) for these scenarios shows, the refined network is quite robust to these deviations, and in fact outperforms the non-refined network's performance on optimal inputs.

k-means clustering is not reliable either; while the first distribution (Fig. 9 (c)) approaches our performance, the second distribution is significantly worse (Fig. 9 (d)). Moreover, it is not easy to predict what the relighting performance of any of these sampling strategies would be, without training Relight-Net for each of them. In contrast, using Sample-Net in conjunction with Relight-Net allows us, in a single training pass, to jointly learn the optimal samples and the relighting function that *maximizes relighting performance*.

Relight-Net A vs. Relight-Net B. We evaluate the effect of introducing rendering constraints into the relighting function, by comparing the performance of Relight-Net A and Relight-Net B for the same ($\theta = 90, k = 5$) configuration. For this experiment, we first trained Sample-Net+Relight-Net A to learn the optimal input directions and then trained Relight-Net B with the same directions. Fig. 10 shows a comparison of the relighting error for these two networks. While Relight-Net B has marginally better average performance (PSNR of 23.20 vs. 23.12 for Relight-Net A), it does not consistently outperform Relight-Net A in terms of visual quality. This might be attributable to the difficulty of learning the high-frequency visibility function (PSNR of 12.41). In contrast, the reconstruction of the direct illumination component is quite accurate (PSNR of 23.04).

We chose Relight-Net A for all the experiments in this paper because it matches Relight-Net B's *relighting* performance and is faster to evaluate. However, Relight-Net B's scene decomposition results suggest that using this technique for scene *reconstruction* could be an interesting direction of future work.

4.2 Refining Relight-Net

After joint training, Relight-Net can relight a scene from the k sparse samples from the learnt optimal directions. However, this would require an acquisition system to recreate these optimal light directions exactly, which can be challenging in practice. In order to reduce this requirement, we refine Relight-Net by training it to handle input light directions in the local neighborhood of the optimal directions. Note that we are able to do this because the input light directions are one of the inputs to Relight-Net.

To refine Relight-Net, we generate a new training dataset comprised of the original 500 training scenes as well as a new set of 5000 scenes (for a total of 5500 scenes). For each scene, we render a new set of k random input samples (sampled within a 10° cone around the learnt optimal light directions) and another 50 output images under random light directions over the entire $\mathcal{L}_\theta$ cone. These images are generated as described in Sec. 3.3. As before, we refine Relight-Net using 10 random 128×128 crops from each image.

Figure 11 compares Relight-Net error distributions before and after refinement on the same test dataset. We can see that refinement improves reconstruction quality (23.12 before refinement vs. 24.15 after) even when we use the optimal input directions that the joint optimization selected. This is a consequence of refining Relight-Net on a larger (5500 vs. 500) scene dataset; while we could have trained our combined Sample-Net+Relight-Net on this dataset, the large computational requirements to train Sample-Net make this intractable. More importantly, the refined network is able to handle inputs that are away from the optimal directions. To evaulate this, we randomly select two sets of 10 input directions that are 1–5° and 5–10° away from the optimal directions, and test relighting performance for these directions on our entire test dataset. The average PSNRs of these 2 settings are shown on the right in Fig. 11. We can see that even when the inputs vary by 5–10° from the optimal directions, the refined network achieves 23.59 average PSNR, and in fact, outperforms the non-refined network's results with the optimal input samples. As we show in our experiments on real data, this robustness to the input directions allows our method to produce high-quality results even for datasets captured by acquisition setups that do not exactly meet our specifications.

5 RESULTS AND EVALUATION

We now present an evaluation and comparisons of our method on both synthetic and real data. Unless otherwise specified, the results in this section were generated using our refined Relight-Net A model trained for the ($\theta = 90, k = 5$) setting. We also provide additional examples and results (including videos under moving lights) in the

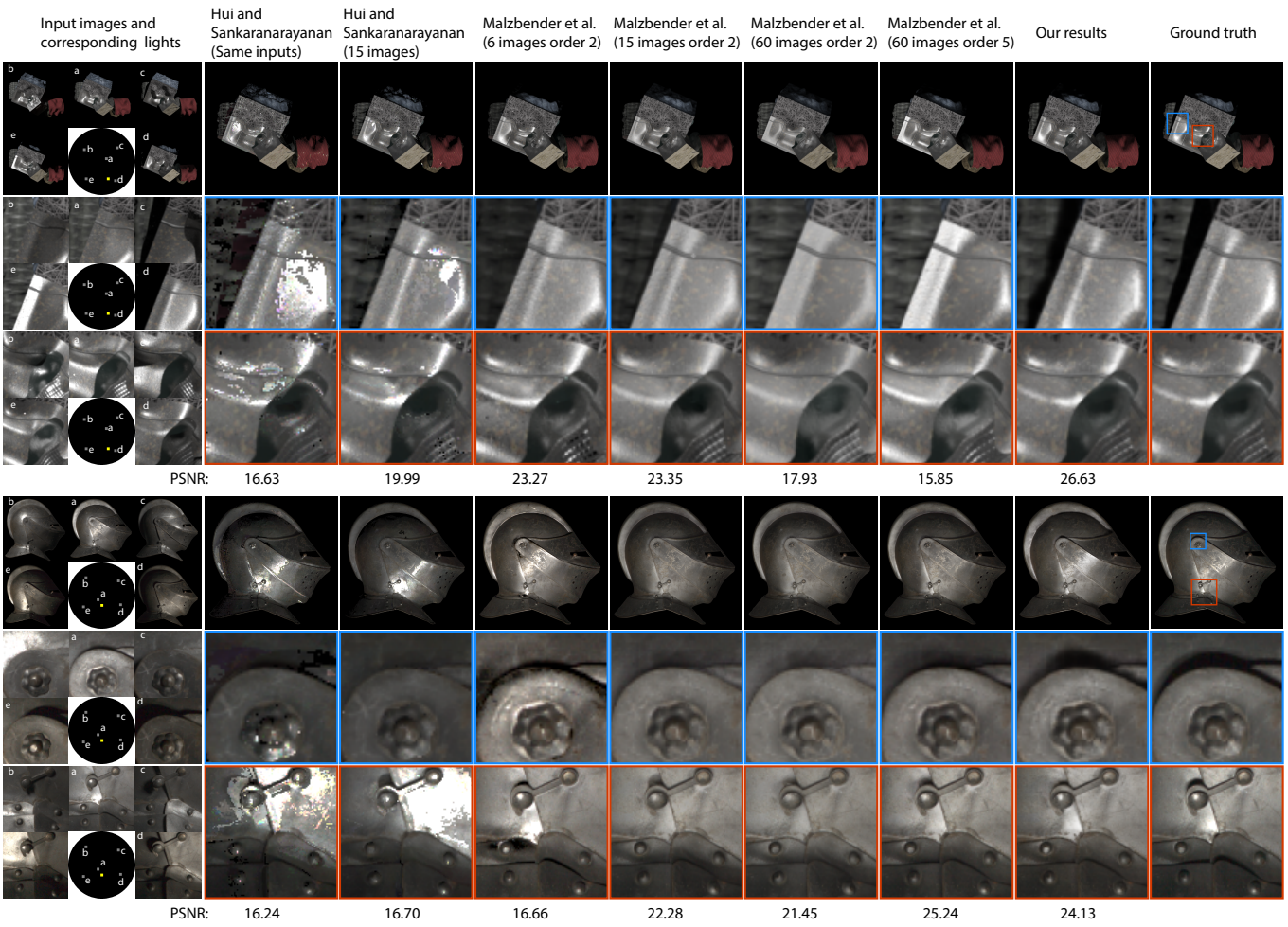


Fig. 12. Comparisons with Photometric Stereo [Hui and Sankaranarayanan 2017] (with 5 and 15 samples), and Polynomial Texture Maps [Malzbender et al. 2001] (with 6, 15, and 60 samples) on a synthetic test scene (top) and a real scene captured by [Einarsson et al. 2006] (bottom, the input light directions here deviate from our optimal directions by 3-7°). The input images and corresponding lights are shown on the left and the PSNR of the result is listed below the images. Our results have some of the highest PSNR scores even when compared to methods with more input images. Moreover, our results have better visual quality and reproduce cast shadows and specularities better (insets).

accompanying supplementary video. We encourage readers to zoom into the images in the paper to look at image details.

Datasets. We evaluate our method on the synthetic scenes from our 100-scene test data, as well as three different real datasets — one that we captured ourselves using a gantry-based acquisition system (Figs. 8 and 17), and additional scenes captured with a light stage setup by Einarsson et al. [2006] (Figs. 12 and 17) and Schwartz et al. [2011] (supplementary video and Fig. 14). While we could control our gantry to capture images under our learnt optimal directions, the latter two datasets do not have these directions, and the closest directions deviate by an average of 4°. Our results for these scenes rely on the refined network's robustness to input light directions.

Timing. We can relight a scene under a directional light using a forward pass through our network model; this takes 0.03 seconds on a NVIDIA Geforce 1080Ti for a 512x512-resolution image.

Comparison with Photometric Stereo-based reconstruction. As mentioned in Sec. 1, one approach to image relighting is to reconstruct the scene and re-render it under novel lighting. To compare against this approach, we use a state-of-the-art Photometric Stereo method that can handle spatially-varying BRDFs [Hui and Sankaranarayanan 2017]. Fig. 12 shows comparisons with this method when run on 5 and 15 images. Even a state-of-the-art Photometric Stereo method has large errors when reconstructing a scene from a small number of images, resulting in significant artifacts in the relit results. Moreover, this method does not handle non-local effects like cast shadows and inter-reflections. In comparison, our results are significantly better in terms of both PSNR and visual quality.

Comparison with image-based relighting methods. Most image-based relighting methods are designed for dense input samples captured using specialized hardware. Therefore, we choose two representative methods —Polynomial Texture Mapping (PTM) [Malzbender et al. 2001] and barycentric interpolation — and compare their performance on different sample sets to our results (Fig. 12 and Fig. 13). For PTM, we fit order-2 polynomials to 6, 15, and 60 input images, and order-5 polynomials to 60 input images. Our network outperforms PTM in most of these settings on both synthetic and real data. Using order-5 polynomials and 60 samples (12× as many as we use) allows PTM to outperform our PSNR on a real scene, but, unlike our network, it can't reconstruct specularities and completely blurs shadows. Moreover, PTM's performance does not consistently improve when we add more samples or use higher-order polynomials. This is possibly because polynomials are poor approximations of light transport, especially in the presence of cast shadows, and using an L_2 error to fit them can lead to unstable results. In general, PTM was designed for largely planar scenes with minimal cast shadows. In contrast, our method can handle more complex scenes with a fraction of the number of samples.

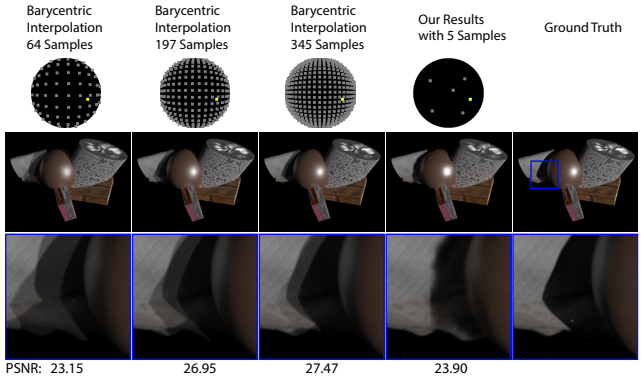


Fig. 13. Comparisons with barycentric interpolation with increasing sampling resolution (top row, samples shown as gray dots and relighting direction in yellow). Our method has a better PSNR than barycentric interpolation with 64 samples. At higher resolutions, barycentric interpolation produces better PSNR, but the subjective visual quality is worse. For example, there are significant ghosting artifacts at shadow boundaries (inset, bottom row).

We analyze how many samples are required for barycentric interpolation to approach our result quality in Fig. 13. We uniformly sample the upper hemisphere with 64, 197 and 345 directions. We render a synthetic scene at these directions, and use these images to do barycentric interpolation for the frontal hemisphere. As shown in Fig. 13, our model produces a reasonable result with plausible (though slightly jagged) shadows even for a novel relighting direction that is outside the convex hull of the input samples. Our method, with 5 samples, has a PSNR of 23.90 vs. 23.15 for barycentric interpolation with 64 images. As the resolution of the sampling is increased, barycentric interpolation starts to outperform our result quantitatively (PSNR of 26.95 and 27.47 for 197 and 345 images). However, the visual quality of our result is still superior; the barycentric interpolation results have significant ghosting artifacts even at 345 input images. These issues are exacerbated in animations with a moving light (please refer to the supplementary video); our reconstructed shadows and specularities move smoothly and intuitively, while the barycentric interpolation results exhibit significant spatial ghosting and temporal aliasing.

Directional and environment map relighting. Using a network trained to handle directional lights in $\mathcal{L}_{90}$ allows us to relight a scene under (the upper hemisphere of) environment lighting by rendering images under every direction in the environment map and summing them using weights based on the environment map radiance values. We use a 64×64 hemispherical environment map (3000 output directions). Note that our network design allows us to pre-compute the encoder features for the input images once, and only process the decoder for different light directions.

Figure 15 shows a comparison of our results for both directional and environment map illumination with ground truth images for four synthetic test scenes. Note that our results under directional lighting match the ground truth images closely with some minor artifacts along sharp shadow boundaries. Moreover, our results under all-frequency environment lighting — generated from only 5 sparse samples — are visually imperceptible from ground truth results. This indicates that, when rendering under environment illumination, the accuracy of our method at relighting most of the directions in $\mathcal{L}_{90}$ is sufficient to compensate for the errors that occur at grazing angles (as shown in Fig. 8).

As Fig. 17 demonstrates, we observe similar behavior in real scenes exhibiting a wide range of materials (diffuse to highly specular), geometries (arbitrary shapes arranged in complex ways), and scene scales and layouts (small to medium to large objects). Our network faithfully reproduces appearance under novel directional lights, and creates photorealistic results under environment map illumination. The bottom two results in this figure come from the [Einarsson et al. 2006] whose light samples deviate from our exact optimal light directions by 3° to 7°. Yet, we obtain high-quality results illustrating the ability of our network to relight using light directions that are not perfectly optimal.

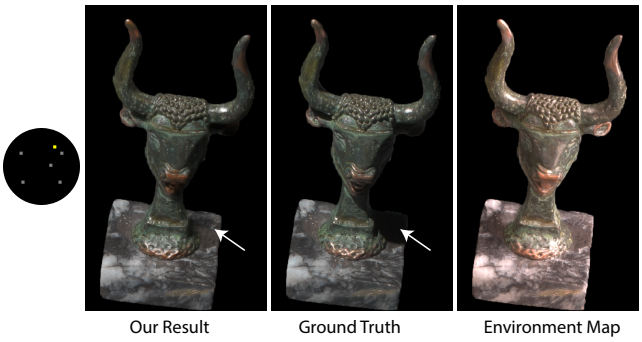


Fig. 14. Limitations. Our method fails to recover cast shadows caused by highly non-convex geometry. However renderings under environment map illumination are still plausible as shown in the rightmost column.

Limitations. While our method produces results of a high quality, some artifacts still remain. While we capture the general shape of cast shadows, the edges can have artifacts (Fig. 13). We train our network on 128×128 image patches; this determines the receptive field of the features and the spatial scale at which we can analyze scene

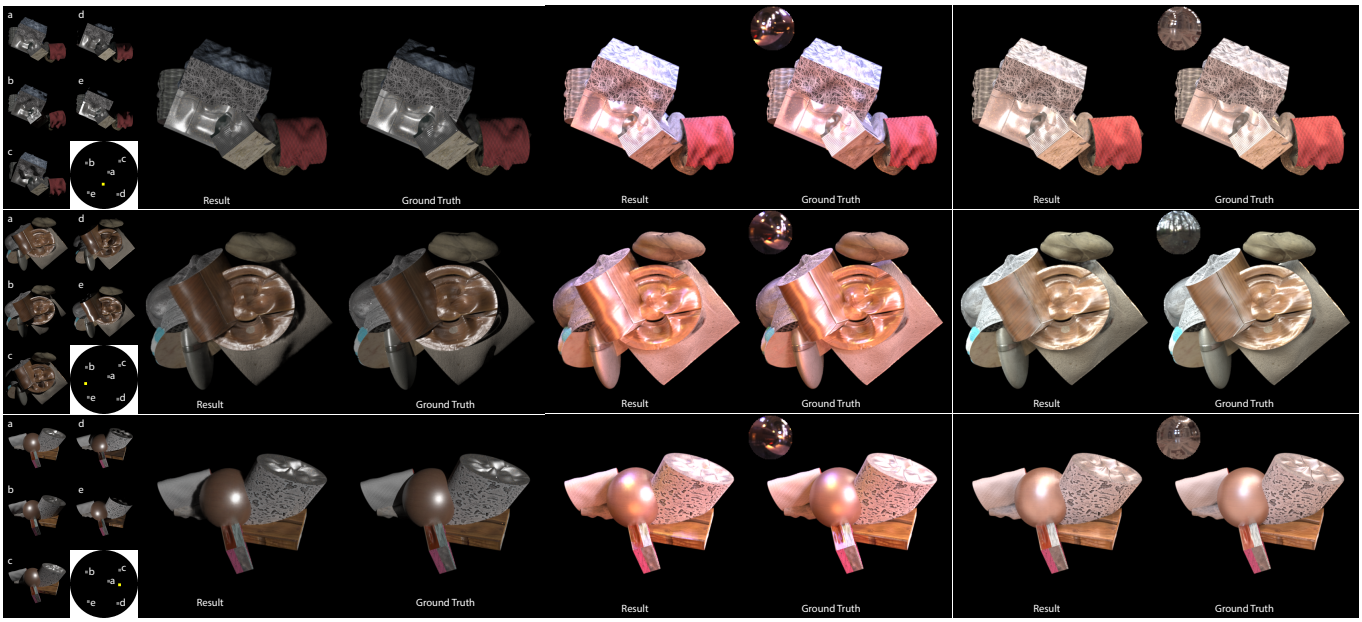


Fig. 15. Relighting results from five samples for synthetic data (first column, input/output light directions marked on the black circle in gray/yellow). Our results match the ground truth images quite faithfully (left vs. right of second column) with some errors near hard shadow boundaries. Moreover, these errors are visually imperceptible under all-frequency hemispherical environment map illumination (third and fourth columns, environment maps in inset).

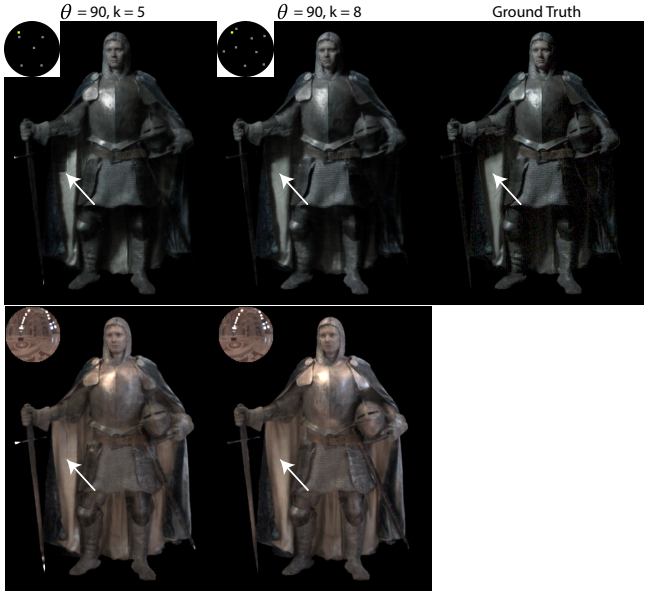


Fig. 16. Comparison between $k = 5$ vs $k = 8$ for $\theta = 90$. In this case, the incident light causes very significant cast shadows, leading to a shadow artifact with sharp corners (noted by the white arrow) in the $k = 5$ result. The artifact goes away when we use $k = 8$ samples. The artifact is also mitigated under environment lighting (bottom).

appearance. Consequently, our method cannot handle non-local appearance changes that happen at a larger scale, for example shadows caused by grazing angle lighting (Fig. 8) or highly non-convex geometry (Fig. 14). Our method might blur very sharp specularities (Fig. 17). As shown in Figs. 14 and 16, these issues can be ameliorated using more samples or when rendering under environment lighting.

Our results are also limited by our training data, and the assumptions made to generate it. For example, we assume that objects in the scene are opaque and don't model complex effects like glints. Increasing the diversity of the shapes, materials, and composition of our scenes could help mitigate this.

6 CONCLUSION AND FUTURE WORK

We have presented a novel approach to relighting a scene from a sparse set of input images. We are able to accomplish this by training a CNN to take 5 images of a scene under single directional lights and render the scene under a novel directional light (in the upper hemisphere). Moreover, we present a scheme to learn the optimal directions for these sparse samples in conjunction with the relighting function by jointly training a combined sampling-relighting network. Extensive evaluations and comparisons to previous state-of-the-art image-based relighting approaches show that we are able to achieve the same (if not better) performance as them, except with an order of magnitude fewer input samples.

This paper suggests a number of interesting directions for future work. At a high-level, most previous scene appearance analysis has relied on simple linear analysis tools. On the other hand, deep networks have been extremely successful at learning good representations for images; can we use them similarly to learn representations for scene appearance? How can we use such representations to reduce the memory and time to relight (or render) a scene? In this work, we learn the optimal set of *directional* lights; how would this change if we also allowed non-directional, *general* illumination? While we have avoided explicit scene reconstruction in this

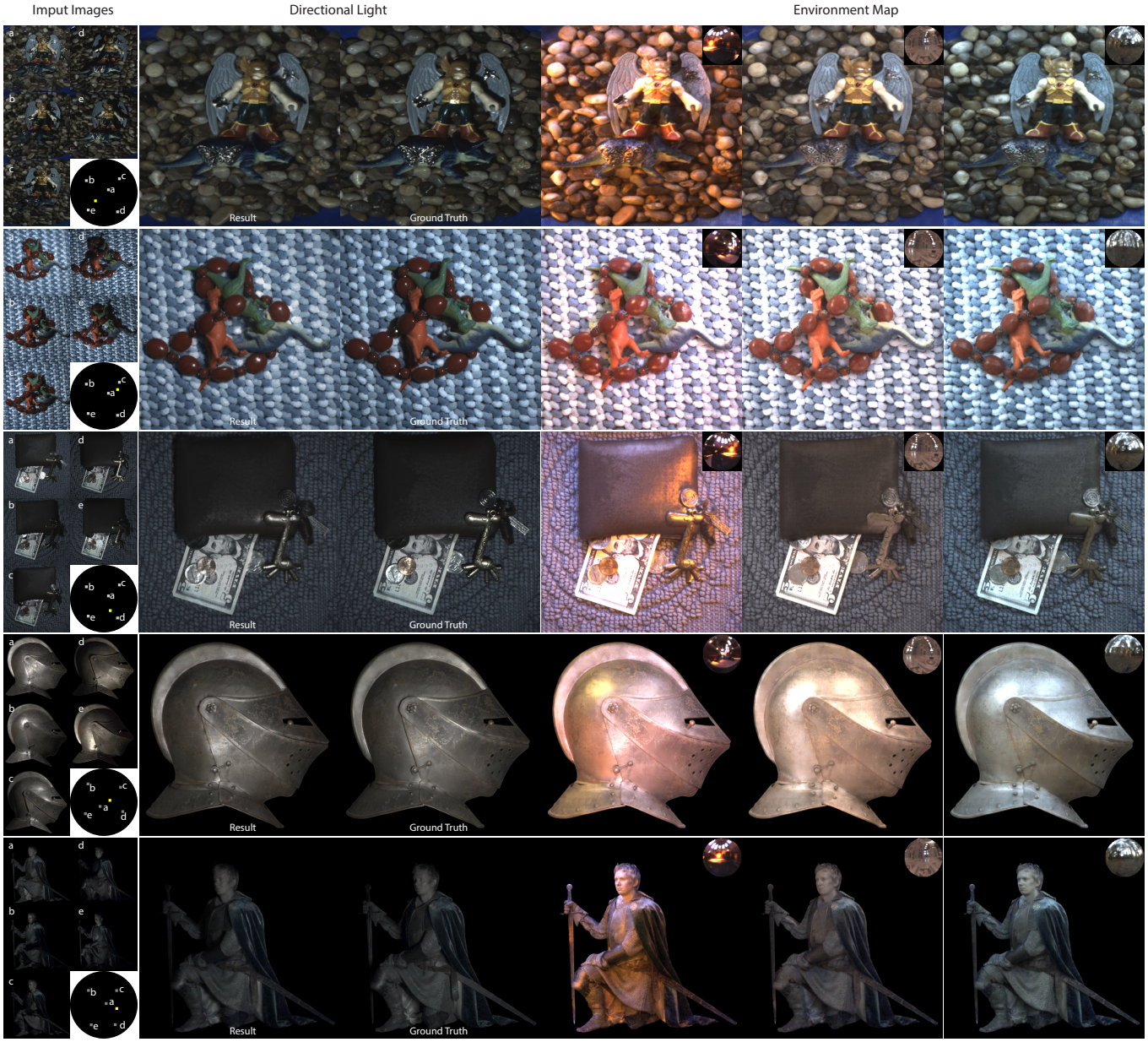


Fig. 17. Real scenes (top 3 captured by us, bottom 2 from [Einarsson et al. 2006]) rendered with environment map lighting. These scenes contain objects with complex reflectances, intricate geometries and span a wide range of scene size and layout. Yet, our method produces accurate relighting results for a single directional light (second column and third column; output direction marked in yellow) and under environment lighting (fourth to sixth columns).

work, the results from training Relight-Net B (Fig. 10) indicate that a network could learn to decompose scene factors from input samples. Combining this with learning the lighting that gives the best reconstruction could be another interesting extension.

ACKNOWLEDGEMENTS

We thank the reviewers for many helpful suggestions. This work was supported in part by NSF grants 1451830 and 1703957, Adobe, a Powell Bundle Fellowship, the Ronald L. Graham endowed Chair and the UC San Diego Center for Visual Computing. We also acknowledge computing resources from the San Diego Supercomputer Center, and NSF Chase-CI 1730158.

REFERENCES

Aayush Bansal, Bryan Russell, and Abhinav Gupta. 2016. Marr Revisited: 2D-3D Model Alignment via Surface Normal Prediction. *CVPR* (2016).

Ronen Basri and David W. Jacobs. 2003. Lambertian Reflectance and Linear Subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 2 (Feb. 2003), 218–233.

Peter N. Belhumeur and David J. Kriegman. 1998. What Is the Set of Images of an Object Under All Possible Illumination Conditions? *International Journal of Computer Vision* 28, 3 (01 Jul 1998), 245–260.

Brett Burley. 2012. Physically-based shading at Disney. In *ACM SIGGRAPH 2012 Courses*.

Ayan Chakrabarti. 2016. Learning sensor multiplexing design through back-propagation. In *Advances in Neural Information Processing Systems*. 3081–3089.

Manmohan Chandraker. 2016. The information available to a moving observer on shape with unknown, isotropic BRDFs. *IEEE transactions on pattern analysis and machine intelligence* 38, 7 (2016), 1283–1297.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 145–156.

David Eigen and Rob Fergus. 2015. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. *ICCV* (2015).

Per Einarsson, Charles-Felix Chabert, Andrew Jones, Wan-Chun Ma, Bruce Lamond, Tim Hawkins, Mark T Bolas, Sebastian Sylwan, and Paul E Debevec. 2006. Relighting Human Locomotion with Flowed Reflectance Fields. *Rendering techniques* 2006 (2006), 17th.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. DeepStereo: Learning to Predict New Views From the World's Imagery. In *CVPR*.

Martin Fuchs, Volker Blanz, Hendrik Lensch, and Hans-Peter Seidel. 2007. Adaptive sampling of reflectance fields. *ACM Transactions on Graphics (TOG)* 26, 2 (2007), 10.

Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. 2017. Learning to Predict Indoor Illumination from a Single Image. *ACM Transactions on Graphics (SIGGRAPH Asia)* 9, 4 (2017).

Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. 2017. What Is Around The Camera?. In *ICCV*.

Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. 2010. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 6 (2010), 1060–1071.

Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. 2017. Deep Outdoor Illumination Estimation. In *CVPR*.

Z. Hui and A. C. Sankaranarayanan. 2017. Shape and Spatially-Varying Reflectance Estimation from Virtual Exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 10 (Oct 2017), 2060–2073. https://doi.org/10.1109/TPAMI.2016.2623613

Wenzel Jakob. 2010. Mitsuba renderer. (2010). http://www.mitsuba-renderer.org.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 193.

Eric P Lafortune and Yves D Willems. 1993. Bi-directional path tracing. (1993).

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 45.

Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. 2017. Material Editing Using a Physically Based Rendering Network. In *ICCV*. 2261–2269.

Dhruv Mahajan, Ira Kemelmacher Shlizerman, Ravi Ramamoorthi, and Peter Belhumeur. 2007. A Theory of Locally Low Dimensional Light Transport. *ACM Trans. Graph.* 26, 3, Article 62 (July 2007).

Tom Malzbender, Dan Gelb, and Hans Wolters. 2001. Polynomial Texture Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 519–528.

Wojciech Matusik, Matthew Loper, and Hanspeter Pfister. 2004. Progressively-Refined Reflectance Functions from Natural Illumination. In *Eurographics Workshop on Rendering*, Alexander Keller and Henrik Wann Jensen (Eds.).

Shree K. Nayar, Peter N. Belhumeur, and Terry E. Boult. 2004. Lighting Sensitive Display. *ACM Trans. Graph.* 23, 4 (Oct. 2004), 963–979.

Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. 2003. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 376–381.

Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. On optimal, minimal BRDF sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 186.

Geoffrey Oxholm and Ko Nishino. 2016. Shape and reflectance estimation in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38, 2 (2016), 376–389.

Pieter Peers and Philip Dutré. 2005. Inferring reflectance functions from wavelet noise. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*. Eurographics Association, 173–182.

Pieter Peers, Dhruv K Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. 2009. Compressive light transport sensing. *ACM Transactions on Graphics (TOG)* 28, 1 (2009), 3.

Ravi Ramamoorthi and Pat Hanrahan. 2001. On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. *J. Opt. Soc. Am. A* 18, 10 (Oct 2001), 2448–2459.

Dikpal Reddy, Ravi Ramamoorthi, and Brian Curless. 2012. Frequency-space Decomposition and Acquisition of Light Transport Under Spatially Varying Illumination. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI (ECCV'12)*. Springer-Verlag, Berlin, Heidelberg, 596–610. https://doi.org/10.1007/978-3-642-33783-3_43

Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Efstratios Gavves, and Tinne Tuytelaars. 2016. Deep reflectance maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4508–4516.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 111.

O. Ronneberger, P.Fischer, and T. Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (LNCS)*, Vol. 9351. Springer, 234–241.

Christopher Schwartz, Michael Weinmann, Roland Ruiters, and Reinhard Klein. 2011. Integrated High-Quality Acquisition of Geometry and Appearance for Cultural Heritage. In *VAST*. Eurographics Association, 25–32.

Amnon Shashua. 1997. On Photometric Issues in 3D Visual Recognition from a Single 2D Image. *International Journal of Computer Vision* 21, 1 (01 Jan 1997), 99–122.

Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. 2003. Clustered Principal Components for Precomputed Radiance Transfer. *ACM Trans. Graph.* 22, 3 (July 2003), 382–391.

Kalyan Sunkavalli, Todd Zickler, and Hanspeter Pfister. 2010. Visibility Subspaces: Uncalibrated Photometric Stereo with Shadows. In *ECCV*. 251–264.

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 195–206.

Jiaping Wang, Yue Dong, Xin Tong, Zhouchen Lin, and Baining Guo. 2009. Kernel Nyström method for light transport. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 29.

Robert J. Woodham. 1980. Photometric Method For Determining Surface Orientation From Multiple Images. *Optical Engineering* 19 (1980), 19 – 19 – 6.

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 Transactions on Graphics (TOG)* 35, 6 (2016), 188.