Weakly-supervised Single-view Image Relighting

Renjiao Yi[*], Chenyang Zhu[*], Kai Xu[†]
National University of Defense Technology

Abstract

We present a learning-based approach to relight a single image of Lambertian and low-frequency specular objects. Our method enables inserting objects from photographs into new scenes and relighting them under the new environment lighting, which is essential for AR applications. To relight the object, we solve both inverse rendering and re-rendering. To resolve the ill-posed inverse rendering, we propose a weakly-supervised method by a low-rank constraint. To facilitate the weakly-supervised training, we contribute Relit, a large-scale (750K images) dataset of videos with aligned objects under changing illuminations. For re-rendering, we propose a differentiable specular rendering layer to render low-frequency non-Lambertian materials under various illuminations of spherical harmonics. The whole pipeline is end-to-end and efficient, allowing for a mobile app implementation of AR object insertion. Extensive evaluations demonstrate that our method achieves state-of-the-art performance. Project page: https://renjiaoyi.github.io/relighting/.

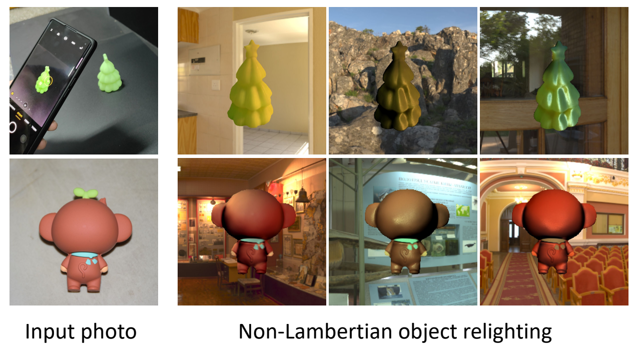

Figure 1. Our method relights real objects into new scenes from single images, which also enables editing materials from diffuse to glossy with non-Lambertian rendering layers.

1. Introduction

Object insertion finds extensive applications in Mobile AR. Existing AR object insertions require a perfect mesh of the object being inserted. Mesh models are typically built by professionals and are not easily accessible to amateur users. Therefore, in most existing AR apps such as SnapChat and Ikea Place, users can use only built-in virtual objects for scene augmentation. This may greatly limit user experience. A more appealing setting is to allow the user to extract objects from a photograph and insert them into the target scene with proper lighting effects. This calls for a method of inverse rendering and relighting based on a single image, which has so far been a key challenge in the graphics and vision fields.

Relighting real objects requires recovering lighting, geometry and materials which are intertwined in the observed image; it involves solving two problems, inverse rendering [17] and re-rendering. Furthermore, to achieve realistic results, the method needs to be applicable for non-Lambertian objects. In this paper, we propose a pipeline to solve both problems, weakly-supervised inverse rendering and non-Lambertian differentiable rendering for Lambertian and low-frequency specular objects.

Inverse rendering is a highly ill-posed problem, with several unknowns to be estimated from a single image. Deep learning methods excel at learning strong priors for reducing ill-posedness. However, this comes at the cost of a large amount of labeled training data, which is especially cumbersome to prepare for inverse rendering since ground truths of large-scale real data are impossible to obtain. Synthetic training data brings the problem of domain transfer. Some methods explore self-supervised pipelines and acquire geometry supervisions of real data from 3D reconstruction by multi-view stereo (MVS) [34, 35]. Such approaches, however, have difficulties in handling textureless objects.

To tackle the challenge of training data shortage, we propose a *weakly-supervised inverse rendering pipeline* based on a novel low-rank loss and a re-rendering loss. For low-rank loss, a base observation here is that the material reflectance is invariant to illumination change, as an intrinsic property of an object. We derive a low-rank loss for inverse rendering optimization which imposes that *the reflectance maps of the same object under changing illuminations are linearly correlated.* In particular, we constrain

[*]Co-first authors.
[†]Corresponding author: kevin.kai.xu@gmail.com.

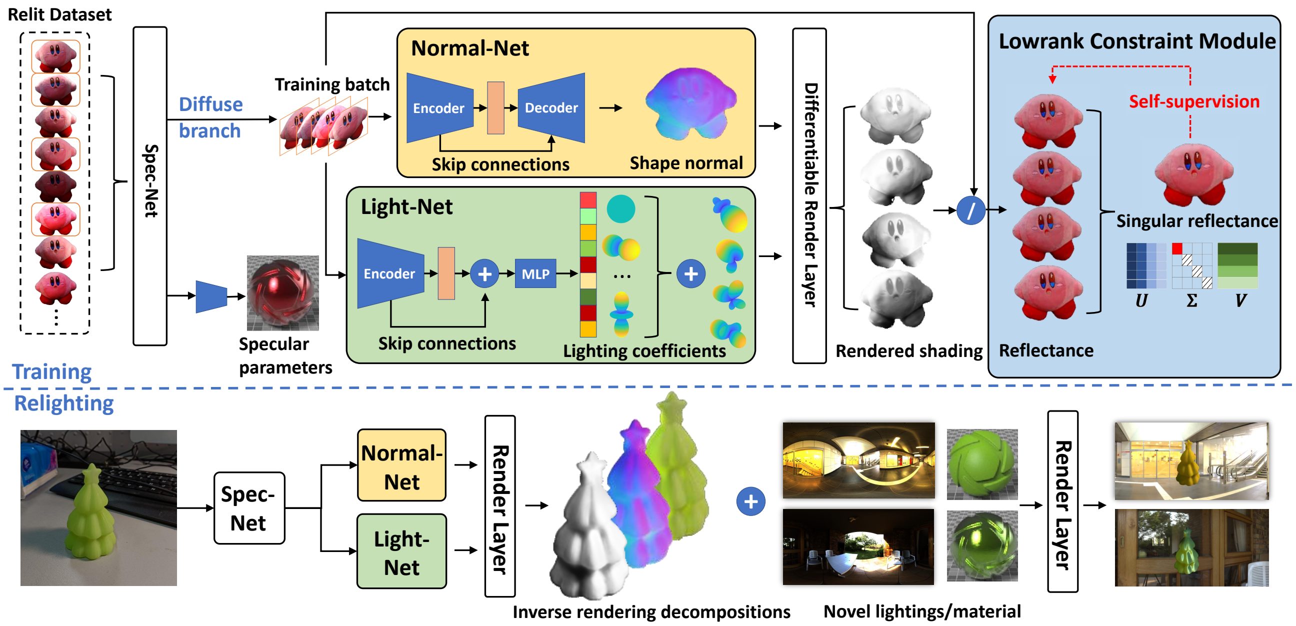

Figure 2. Overview of our method. At training time, Spec-Net separates input images into specular and diffuse branches. Spec-Net, Normal-Net and Light-Net are trained in a self-supervised manner by the Relit dataset. At inference time, inverse rendering properties are predicted to relight the object under novel lighting and material. The non-Lambertian render layers produce realistic relit images.

the reflectance matrix with each row storing one of the reflectance maps to be rank one. This is achieved by minimizing a low-rank loss defined as the Frobenius norm between the reflectance matrix and its rank-one approximation. We prove the convergence of this low-rank loss. In contrast, traditional Euclidean losses lack a convergence guarantee.

To facilitate the learning, we contribute Relit, a large-scale dataset of videos of real-world objects with changing illuminations. We design an easy-to-deploy capturing system: a camera faces toward an object, both placed on top of a turntable. Rotating the turntable will produce a video with the foreground object staying still and the illumination changing. To extract the foreground object from the video, manual segmentation of the first frame suffices since the object is aligned across all frames.

As shown in Figure 2, a fixed number of images under different lighting are randomly selected as a batch. We first devise a Spec-Net to factorize the specular highlight, trained by the low-rank loss on the chromaticity maps of diffuse images (image subtracts highlight) which should be consistent within the batch. With the factorized highlight, we further predict the shininess and specular reflectance, which is self-supervised with the re-rendering loss of specular highlight. For the diffuse branch, we design two networks, Normal-Net and Light-Net, to decompose the diffuse component by predicting normal maps and spherical harmonic lighting coefficients, respectively. The diffuse shading is rendered by normal and lighting, and diffuse reflectance (albedo) is computed by diffuse image and shading. Both networks are trained by low-rank loss on diffuse reflectance.

Regarding the re-rendering phase, the main difficulty is the missing 3D information of the object given a single-view image. The Normal-Net produces a normal map which is a partial 3D representation, making the neural rendering techniques and commercial renderers inapplicable. The existing diffuse rendering layer for normal maps of [20] cannot produce specular highlights. Pytorch3D and [9, 11] render specular highlights for point lights only.

To this end, we design a *differentiable specular renderer* from normal maps, based on the Blinn-Phong specular reflection [5] and spherical harmonic lighting [6]. Combining with the differentiable diffuse renderer, we can render low-frequency non-Lambertian objects with prescribed parameters under various illuminations, and do material editing as a byproduct.

We have developed *an Android app* based on our method which allows amateur users to insert and relight arbitrary objects extracted from photographs in a target scene. Extensive evaluations on inverse rendering and image relighting demonstrate the state-of-the-art performance of our method.

Our contributions include:

- A weakly-supervised inverse rendering pipeline trained with a low-rank loss. The correctness and convergence of the loss are mathematically proven.
- A large-scale dataset of foreground-aligned videos collecting $750K$ images of 100+ real objects under different lighting conditions.

- An Android app implementation for amateur users to make a home run.

2. Related Work

Inverse rendering. As a problem of inverse graphics, inverse rendering aims to solve geometry, material and lighting from images. This problem is highly ill-posed. Thus some works tackle the problem by targeting a specific class of objects, such as faces [27, 29] or planar surfaces [1]. For inverse rendering of general objects and scenes, most prior works [4, 8, 12, 13] require direct supervisions by synthesized data. However, networks trained on synthetic data have a domain gap for real testing images. Ground truths of real images are impossible to obtain, and it calls for self-supervised methods training on real images. Recently, self-supervised methods [34, 35] explore self-supervised inverse rendering for outdoor buildings, where the normal supervision is provided by reconstructing the geometry by MVS. However, they do not work well for general objects, which is reasonable because object images are unseen during training. However, applying the pipelines for objects meet new problems. Textureless regions on objects are challenging for MVS due to lack of features. It motivates our work on weakly-supervised inverse rendering for general objects. To fill the blank of real-image datasets on this topic, we capture a large-scale real-image datasets Relit to drive the training.

There are also many works addressing inverse rendering as several separated problems, such as intrinsic image decomposition [10, 14, 25, 32], specularity removal [24, 25, 31] or surface normal estimation [12]. In order to compare with more related methods, we also evaluate these tasks individually in experiments.

Image relighting. Most prior methods in image-based relighting require multi-image inputs [2, 30]. For example, in [30], a scene is relit from a sparse set of five images under the optimal light directions predicted by CNNs. Single-image relighting is highly ill-posed, and needs priors. [18, 34] target outdoor scenes, and benefit from priors of outdoor lighting models. [15, 22, 26–28] target at portrait images, which is also a practical application for mobile AR. Single image relighting for general scenes have limited prior works. Yu et al. [34] takes a single image as inputs, with the assumption of Lambertian scenes. In this work, we propose a novel non-Lambertian render layer, and demonstrate quick non-Lambertian relighting of general objects.

3. Overview

We propose a deep neural network to solve single-image inverse rendering and object-level relighting. The overall pipeline is shown in Figure 2. The whole pipeline is weakly-supervised with a supervised warm-up of NormalNet, and self-supervised training of the whole pipeline. The self-supervised training is driven by the Relit Dataset. The details of Relit Dataset is intoduced in Section 4. In Section 5, we introduce the proposed pipeline following the order from single-image inverse rendering to differentiable non-Lambertian relighting. The weakly-supervised inverse rendering, including the proofs of theoretical fundamentals and convergence of the low-rank loss, are introduced in Section 5.2. The differentiable non-Lambertian rendering layers are introduced in Section 5.3.

4. The Relit Dataset

To capture foreground-aligned videos of objects under changing illuminations, we design an automatic device for data capture, as shown in Figure 3 (left). The camera and object are placed on the turntable, and videos are captured as the turntable rotating. The target object stays static among the frames in captured videos, with changing illuminations and backgrounds. In summary, the Relit dataset consists of 500 videos for more than 100 objects under different indoor and outdoor lighting. Each video is 50 seconds, resulting in 1500 foreground-aligned frames under various lighting. In total, the Relit dataset consists of $750K$ images. Selected objects are shown in Figure 3 (right). The objects cover a wide variety of shapes, materials, and textures. In Section 5, we introduce how to leverage Relit dataset to drive the self-supervised training. It can facilitate many tasks, such as image relighting and segmentation.

5. Our Method

5.1. Image formation model

A coarse-level image formation model for inverse rendering is intrinsic image decomposition (IID), which is a long-standing low-level vision problem, decomposing surface reflectance from other properties, assuming Lambertian surfaces. For non-Lambertian surfaces, the model can be improved by adding a specular highlight term:

$$I = I_d + H, \quad I_d = \mathcal{A} \odot S, \tag{1}$$

where H is the specular highlight, $\mathcal{A}$ is the surface reflectance map, i.e. albedo map in IID, and S is a term describing the shading related to illumination and geometry. Here $\odot$ denotes the Hadamard product. To be more specific, according to the well-known Phong model [19] and Blinn-Phong model [5], the image can be formulated as the sum of a diffuse term and a specular term:

$$\begin{aligned} I(p) =& I_d(p) + H(p), \\ I_d(p) =& \mathcal{A}(p)S(p) = \mathcal{A}(p)\sum_{\omega\in\mathcal{L}} l_\omega(L_\omega \cdot n(p)), \\ H(p) =& \sum_{\omega\in\mathcal{L}} s_p l_\omega\left(\frac{L_\omega + v}{\|L_\omega + v\|}\cdot n(p)\right)^\alpha, \end{aligned} \tag{2}$$

Figure 3. Left: The data capture set-up. Right: Selected objects in Relit dataset. The last row shows selected frames from one video.

where $I(p)$ is the observed intensity and $n_p = (x, y, z)$ is the surface normal at pixel p. $\mathcal{L}$ is a set of sampled point lights in the lighting environment. L_ω and l_ω describe lighting direction and intensity of one point light ω in $\mathcal{L}$ respectively. $\mathcal{A}(p)$ and s_p are defined as the diffuse and specular reflectance at pixel p, respectively. The specular term is not view independent, view direction v is needed to calculate the reflectance intensity and α is a shininess constant. The differentiable approximation for Equation (2) is introduced in Section 5.3.1-5.3.2.

5.2. Inverse rendering from a single image

For relighting, we first inverse the rendering process to get 3D properties including geometry, reflectance, shading, illumination and specularities, then we can replace the illumination and re-render the objects. Following this order, we firstly introduce inverse rendering.

For non-Lambertian object, we can perform specular highlight separation first by the Spec-Net. The specular parameters are then predicted in the specular branch, which is introduced in Section 5.2.2.

For diffuse branch, adopting separate networks to predict normal, lighting, shading, reflectance is the most straightforward choice. However, in this way, the diffuse component in the rendering equation (Equation (2)) is not respected, since relations between these properties are not constrained. Thus, we design a lightweight physically-motivated inverse rendering network, respecting the rendering equation strictly, as shown in Figure 2. There are only two learnable network modules in our end-to-end diffuse inverse rendering pipeline. Here we adopt spherical harmonics [20] to represent illumination $\mathcal{L}$ in Equation (2)), which is calculated more efficiently than Monte Carlo integration of point lights:

$$\mathcal{L} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} C_{l,m} Y_{l,m}, \tag{3}$$

where $Y_{l,m}$ is the spherical harmonic basis of degree l and order m, $C_{l,m}$ is the corresponding coefficient. Each environment lighting can be represented as the weighted sum of spherical harmonics. The irradiance can be well approximated by only 9 coefficients, 1 for $l = 0, m = 0$, 3 for $l = 1, -1 \leq m \leq 1$, and 5 for $l = 2, -2 \leq m \leq 2$.

Normal-Net predicts surface normal maps n, and Light-Net regresses lighting coefficients $C_{l,m}$ in spherical harmonic representation. A total of 12 coefficients are predicted by Light-Net, where the last 3 coefficients present the illumination color. The shading S is then rendered from the predicted normal and lighting, by a hard-coded differentiable rendering layer (no learnable parameters) in Section 5.3.1, following Equation (2). The reflectance $\mathcal{A}$ is computed by Equation (1) after rendering shading. The pipeline design is based on the physical rendering equation (Equation (2)), where relations among terms are strictly preserved.

5.2.1 Self-supervised low-rank constraint

We have foreground-aligned videos of various objects under changing illuminations in Relit dataset. The target object is at a fixed position in each video, which enables pixel-to-pixel losses among frames.

For each batch, N images I_1, I_2, ..., I_N are randomly selected from one video. Since the object is aligned in N images under different lighting, one observation is that the reflectance should remain unchanged as an intrinsic property, and the resulting reflectance $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_N$ should be identical. However, due to the scale ambiguity between reflectance and lighting intensities, i.e., estimating reflectance as $\mathcal{A}$ and lighting as $\mathcal{L}$, is equivalent to estimating them as $w\mathcal{A}$ and $\frac{1}{w}\mathcal{L}$. A solution for supervised methods is defining a scale-invariant loss between ground truths and predictions. However the case is different here, there are no predefined ground truths. While adopting traditional Euclidean losses between every pair in $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_N$, it leads to degenerate results where all reflectance are converged to zero. To solve the problem, here we enforce $\mathcal{A}_1$, $\mathcal{A}_2$,..., $\mathcal{A}_N$ to be linearly correlated and propose a rank constraint as training loss. Therefore, a scaling factor w does not affect the loss.

We can compose a matrix R with each reflectance $\mathcal{A}_i$ storing as one row. Ideally, rows in R should be linearly correlated, i.e., R should be rank one. We formulate a self-supervised loss by the distance between R and its rank-one approximation. We introduce Theorem 1 below.

Theorem 1. ***Optimal rank-one approximation.*** *By SVD, $R = U\Sigma V^T$, $\Sigma = diag(\sigma_1, \sigma_2, ...\sigma_k)$, $\Sigma' = diag(\sigma_1, 0, ...)$, $\bar{R} = U\Sigma' V^T$ is the optimal rank-one approximation for R, which meets:*

$$||\bar{R} - R||_F^2 = \min_{b \in \mathcal{R}^N, c \in \mathcal{R}^d} ||bc^T - R||_F^2, \tag{4}$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix.

The proof of Theorem 1 can be found in the supplementary material.

Therefore, we define the low-rank loss as:

$$f(R) = ||\bar{R} - R||_F^2. \tag{5}$$

Its convergence is proven as below, fitting the needs of learning-based approaches training by gradient descents.

Since the gradient of $\bar{R}$ is detached from the training, the derivative of $f(R)$ can be accomplished as $\nabla f(R) = -2(\bar{R} - R)$. According to the gradient descent algorithm, with a learning rate η, the result $R^{(n+1)}$ (R after $n+1$ training iterations), can be deduced as:

$$R^{(n+1)} = R^{(n)} + 2\eta(\bar{R} - R^{(n)}), \tag{6}$$

Theorem 2. ***Convergence of $f(R)$.*** *The loss $f(R)$ would converge to a fixed point, which is $\bar{R}$ while $0 < \eta < 0.5$,*

$$\lim_{n\to\infty} R^{(n)} = \bar{R} \Leftarrow 0 < \eta < 0.5, R^{(0)} = R. \tag{7}$$

Proof. According to Equation (6), $R = U\Sigma V^T$ and $\bar{R} = U\Sigma' V^T$, we have:

$$\begin{aligned} R^{(1)} &= R + 2\eta(\bar{R} - R) \\ &= U\mathrm{diag}\{\sigma_1, (1-2\eta)\sigma_2, \dots, (1-2\eta)\sigma_k\}V^T. \end{aligned} \tag{8}$$

Since $0 < \eta < 0.5$, we have $1 - 2\eta < 1$, $\{\sigma_1, (1-2\eta)\sigma_2, \dots, (1-2\eta)\sigma_k\}$ are still descending. Therefore, Equation (8) is the SVD form for $R^{(1)}$. Similarly, we have:

$$R^{(2)} = U\mathrm{diag}\{\sigma_1, (1-2\eta)^2\sigma_2, \dots, (1-2\eta)^2\sigma_k\}V^T. \tag{9}$$

Repeat n iterations, we have the expression for $R^{(n)}$:

$$R^{(n)} = U\mathrm{diag}\{\sigma_1, (1-2\eta)^n\sigma_2, \dots, (1-2\eta)^n\sigma_k\}V^T. \tag{10}$$

Since $|1 - 2\eta| < 1$, Equation (10) can be reduced to:

$$\begin{aligned} \lim_{n\to\infty} R^{(n)} &= U\mathrm{diag}\{\sigma_1, 0, \dots, 0\}V^T \\ &= U\Sigma' V^T = \bar{R}. \end{aligned} \tag{11}$$

□

In our diffuse branch, the low-rank loss of reflectance back-propagates to Normal-Net and Light-Net, and trains both in self-supervised manners.

5.2.2 Specularity separation

To deal with the specular highlights, we add a Spec-Net, to remove the highlights before diffuse inverse rendering. On highlight regions, pixels are usually saturated and tends to be white. Based on it, we automatically evaluate the percentage of saturated pixels on the object image. If the percentage exceeds 5%, Spec-Net will be performed, otherwise the object is considered as diffuse and Spec-Net will not be performed. We found that under this setting the results are better than performing Spec-Net on all images, since learning-based highlight removal methods tend to overextract highlights on diffuse images. The training of Spec-Net is initialized from the highlight removal network of Yi et al. [32], enhanced with images of non-Lambertian objects in our Relit Dataset by self-supervised finetuning. From the Di-chromatic reflection model [23], if illumination colors remain unchanged, the rg-chromaticity of Lambertian reflection should be unchanged as well. Thus the finetuning can be driven by the low-rank constraint on rg-chromaticity of diffuse images after removing specular highlights, following the image formation model in Equation (1).

With the separated specular highlight, we can further predict specular reflectance s_p and shininess (smoothness) α in Equation (2). The training is self-supervised by re-rendering loss between the separated highlight by Spec-Net, and the re-rendered specular highlight by the predicted s_p, α, lighting coefficients $C_{l,m}$ from Light-Net via the specular rendering layer in Section 5.3.2.

5.2.3 Joint training

Firstly, the Spec-Net is trained to separate input images into specular highlight and diffuse images, as the first phase. Since training to predict specular reflectance and smoothness requires lighting coefficients from Light-Net, Light-Net and Normal-Net in the diffuse branch are trained as the second phase. Training to predict specular reflectance and smoothness is the last phase.

In the second phase, Light-Net predicts spherical harmonic lighting coefficients $C_{l,m}$ corresponding to each basis $Y_{l,m}$. There is an axis ambiguity between Normal-Net and Light-Net predictions. For example, predicting a normal map with the x-axis pointing right with positive coefficients of the bases related to x, is equivalent to predicting a normal map with x-axis pointing left with corresponding coefficients being negative. They would render the same shading results. Normal-Net and Light-Net are in a chicken-and-egg relation and cannot be tackled simultaneously. We employ a joint training scheme to train Normal-Net and Light-Net alternatively. To initialize the coordinate system in Normal-Net, we use a small amount of synthetic data (50k images) from LIME [16] to train an initial Normal-Net. Then we freeze Normal-Net and train Light-Net from scratch by our low-rank loss on reflectance, as the 1^{st} round joint training. Then Light-Net is frozen and Normal-Net is trained from the initial model by the same low-rank loss on reflectance. The joint training is driven by the Relit dataset, using 750k unlabeled images. Normal-Net is weakly-supervised due to the pretraining and all other nets are self-supervised. The joint training scheme effectively avoids the axis ambiguity and the quantitative ablation studies are shown in Section 6.1.

5.3. Non-Lambertian object relighting

After inverse rendering, an input photo is decomposed into normal, lighting, reflectance, shading and a possible specular component by our network. With these predicted properties, along with the lighting of new scenes, the object is re-rendered and inserted into new scenes. We propose a specular rendering layer in Section 5.3.2. Given specularity parameters (specular reflectance and smoothness), we can relight the object in a wide range of materials.

Both diffuse and specular render layers take spherical harmonic coefficients as lighting inputs, which present low-frequency environment lighting. The transformations from HDR lighting paranomas to SH coefficients are precomputed offline. We also implement a mobile App, whose details are in the supplementary material.

5.3.1 Diffuse rendering layer

In order to encode the shading rendering while keeping the whole network differentiable, we adopt a diffuse rendering layer respecting to Equation (2)-(3), based on [20]. The rendering layer takes the spherical harmonic coefficients as lighting inputs. Combining Equation (2)-(3), introducing coefficients $\hat{A}_l$ from [21], and incorporating normal into the spherical harmonic bases, the shading and the diffuse component of relit images are rendered by:

$$I_d(p) = \mathcal{A}(p) \sum_{\omega \in \mathcal{L}} l_\omega(L_\omega \cdot n(p)) = \mathcal{A}(p) \sum_{l,m} \hat{A}_l C_{l,m} Y_{l,m}(\theta, \phi), \tag{12}$$

where (θ, ϕ) is the spherical coordinates where $(x, y, z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ and $n(p) = (x, y, z)$.

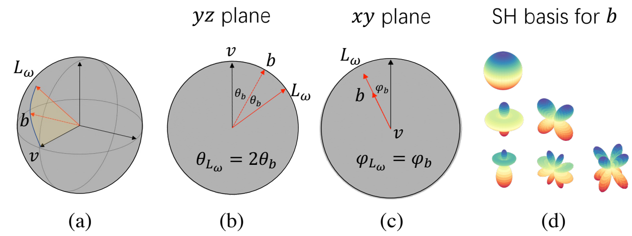

Figure 4. The visual illustration of b, L_ω and v. (a) The view point v is set as $[0, 0, 1]$ in our inverse rendering problem. b is the bisector of L_ω and v. (b) Observe b, L_ω and v in the yz plane view. We find that the polar angle $\theta_{L_\omega} = 2\theta_b$. (c) We find that the azimuth angle $\phi_{L_\omega} = \phi_b$ from the xy plane view. (d) Then we get spherical harmonic basis $Y_{l,m}(\theta_b, \phi_b)$ for differentiable rendering of the specular component.

5.3.2 Specular rendering layer

Since the specular componet is view dependent, which can not be simply parameterized with $Y_{l,m}$ and $C_{l,m}$ as in the diffuse renderer. With the assumption of distant lighting, the view point is fixed. As shown in Figure 4, $b = \frac{L_\omega + v}{\|L_\omega + v\|}$ is the bisector of light direction L_ω and view point v. Note that, b has the same azimuth angle ϕ as L_ω while polar angle θ is only a half under spherical coordinate system as shown in Figure 4. Since the predicted normal map has a pixel-to-pixel correspondence to the input image, which means the normal map is projected perspectively. We only need to apply orthogonal projection in the re-rendering step by assuming viewing the object the z direction, which means $v = [0, 0, 1]$. The re-rendered images share a pixel-wise correspondence to observed images, following a perspective projection.

Now we can modify $Y_{l,m}$ into $\hat{Y}_{l,m}$, and use $\hat{A}_l \hat{Y}_{l,m}$ to describe the distribution of all possible b as well, keeping lighting coefficients $C_{l,m}$ unchanged for sharing between both renderers.

$$\begin{aligned}
\hat{Y}_{0,0}(\theta, \phi) &= Y_{0,0}(2\theta, \phi) = c_0 \\
\hat{Y}_{1,1}(\theta, \phi) &= Y_{1,1}(2\theta, \phi) = c_1 \sin 2\theta \cos\phi = 2c_1 xz \\
\hat{Y}_{1,-1}(\theta, \phi) &= Y_{1,-1}(2\theta, \phi) = c_1 \sin 2\theta \sin\phi = 2c_1 yz \\
\hat{Y}_{1,0}(\theta, \phi) &= Y_{1,0}(2\theta, \phi) = c_1 \cos 2\theta = c_1(2z^2 - 1) \\
\hat{Y}_{2,-2}(\theta, \phi) &= Y_{2,-2}(2\theta, \phi) = 4c_2 xyz^2 \\
\hat{Y}_{2,1}(\theta, \phi) &= Y_{2,1}(2\theta, \phi) = c_2(4xz^3 - 2xz) \\
\hat{Y}_{2,-1}(\theta, \phi) &= Y_{2,-1}(2\theta, \phi) = c_2(4yz^3 - 2yz) \\
\hat{Y}_{2,0}(\theta, \phi) &= Y_{2,0}(2\theta, \phi) = c_3(3(4z^4 - 4z^2 + 1) - 1) \\
\hat{Y}_{2,2}(\theta, \phi) &= Y_{2,2}(2\theta, \phi) = c_4(4x^2z^2 - 4y^2z^2) \\
c_0 &= 0.282095, c_1 = 0.488603 \\
c_2 = 1.092548&, c_3 = 0.315392, c_5 = 0.546274
\end{aligned}$$

Hence, we can write the differentiable rendering approximation for the specular component similar as Equation (12):

$$H(p) = s_p \sum_{\omega \in \mathcal{L}} l_\omega \left(\frac{L_\omega + v}{\|L_\omega + v\|} \cdot n(p) \right)^\alpha \approx s_p \sum_{l,m} C_{l,m} (\hat{A}_l \hat{Y}_{l,m}(\theta, \phi))^\alpha. \tag{13}$$

6. Experiments

In this section, we evaluate the performance of inverse rendering and image relighting. The inverse rendering evaluation with a series of state-of-the-art methods is presented in Section 6.1, along with several ablations. For image relighting, we provide quantitative evaluations on a synthetic dataset in Section 6.2, and real object insertion is demonstrated in Figure 1 and the project page.

6.1. Inverse rendering

Many prior works address surface normal estimation or intrinsic image decomposition but not both, and there are no benchmark datasets for inverse rendering, we evaluate these two tasks individually. Evaluations on lighting and specularity are in the supplementary material. The end-to-end inverse rendering takes 0.15 seconds per image at 256×256 on a Titan T4 GPU.

Intrinsic image decomposition. We compare our self-supervised intrinsic image decomposition to several inverse rendering methods (InverseRenderNet [35], RelightNet [13], ShapeAndMaterial [13]), and intrinsic image decomposition methods [10, 14, 25, 32] on MIT Intrinsics dataset, which is a commonly-used benchmark dataset for IID. To evaluate the performances and cross-dataset generalizations, all methods are not finetuned on this dataset. We adopt scale-invariant MSE (SMSE) and local scale-invariant MSE (LMSE) as error metrics, which are designed for this dataset [7]. As shown in Table 1 (visual comparisons are in the supplementary material), our method outperforms all unsupervised and self-supervised methods and has comparable performance with supervised ones. Note that the assumptions of white illumination and Lambertian surfaces in this dataset fit the cases of synthetic data, which benefit supervised methods. However, self-supervised and unsupervised methods enable training on unlabeled real-image datasets, which produce better visual results on unseen natural images. As shown in Figure 5, SIRFS [4], a method based on scene priors, fails to decompose reflectance colors. InverseRenderNet [35] and RelightingNet [34] tend to predict a similar color of shading and reflectance, leading to unnatural reflectance colors. ShapeAndMaterial [13] generates visually good results but has artifacts on reflectance due to specular highlights. Our method decomposes these components by considering non-Lambertian cases.

Table 1. Quantitative comparisons with state-of-the-art alternatives and ablation study of intrinsic image decomposition on MIT intrinsic dataset.

Methods	Supervision	Data type	SMSE	LMSE
Shi et al. [25]	Sup.	Synthetic	0.0194	0.0318
Li et al. [10]	Sup.	Synthetic	0.0186	**0.0259**
Shape&Material [13]	Sup.	Synthetic	**0.0150**	0.0309
RelightingNet [34]	Self-sup.	Real	0.0368	0.1077
Yi et al. [32]	Unsup.	Real	0.0231	0.0422
InverseRenderNet [35]	Self-sup.	Real	0.0299	0.0855
Liu et al. [14]	Unsup.	Real	0.0193	0.0428
Ours	Self-sup.	Real	**0.0186**	**0.0369**
1^{st} round training	Self-sup.	Real	0.0224	00420
w/o joint training	Self-sup.	Real	0.0216	0.0399
loss^{+} (σ_2)	Self-sup.	Real	0.0357	0.0513
loss* (σ_2/σ_1)	Self-sup.	Real	0.0808	0.2137

Table 2. Quantitative comparisons with state-of-the-art alternatives and ablation study of surface normal estimation on the dataset from Janner et al. [8].

Methods	MSE	DSSIM
SIRFS [3]	0.0230	0.0243
SVBRDF [12]	0.0144	0.0278
InverseRenderNet [35]	0.0084	0.0272
RelightNet [34]	0.0080	0.0265
ShapeAndMaterial [13]	0.0060	0.0228
Ours	**0.0054**	**0.0201**
1^{st} round training	0.0061	00219
w/o joint training	0.0065	0.0228
loss^{+} (σ_2)	0.0059	0.0213
loss* (σ_2/σ_1)	0.0083	0.0309

Normal estimation. We compare our method with several inverse rendering methods [4, 12, 13, 34, 35] on synthetic dataset from Janner et al. [8]. Since the dataset is too large (95k), and SIRFS [4] takes one minute for each data, a testing set of 500 images is uniformly sampled, covering a wide variety of shapes. In Table 2, the evaluations are reported with two error metrics, MSE and DSSIM, measuring pixel-wise and overall structural distances. Our method yields the best performance. Qualitative comparisons are shown in the supplementary material.

Ablations. We present ablations in the last four rows in Table 1-2. 1^{st} round training denotes the networks of initial Normal-Net and self-supervised Light-Net. "w/o joint training" denotes training Normal-Net and Light-Net simultaneously, rather than alternatively. Previous works propose different formulations of low-rank loss, as the second singular value (σ_2) [33, 36] or the second singular value normalized by the first one ($\frac{\sigma_2}{\sigma_1}$) [32] to enforce a matrix to be rank one. As discussed in the original papers, these losses are unstable in training and would degenerate to local optima. The proposed low-rank constraint is more robust as proven, not suffering from local optimas. More discussions and visual comparisons of these low-rank losses are in the supplementary material.

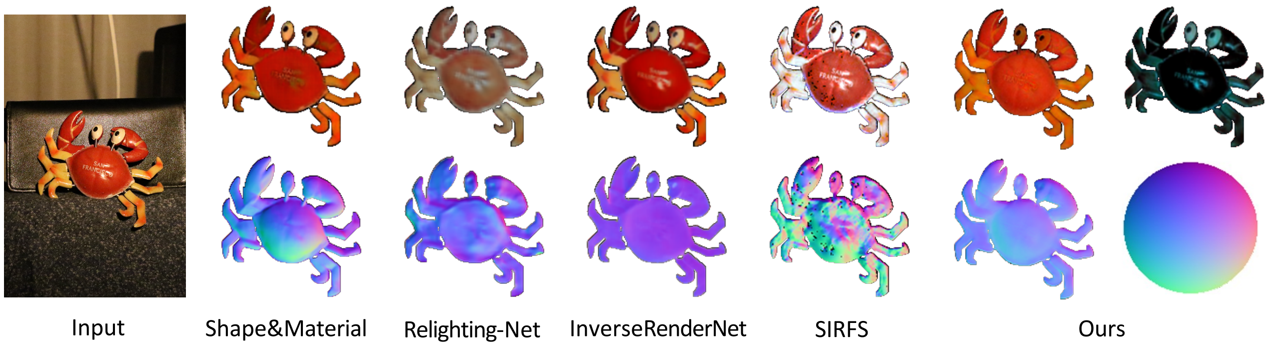

Figure 5. Qualitative comparisons on an unseen image, comparing with state-of-the-art methods. The first row shows the reflectance of all methods and specular highlights of our method. The second row shows estimated normal maps and the colormap for reference.

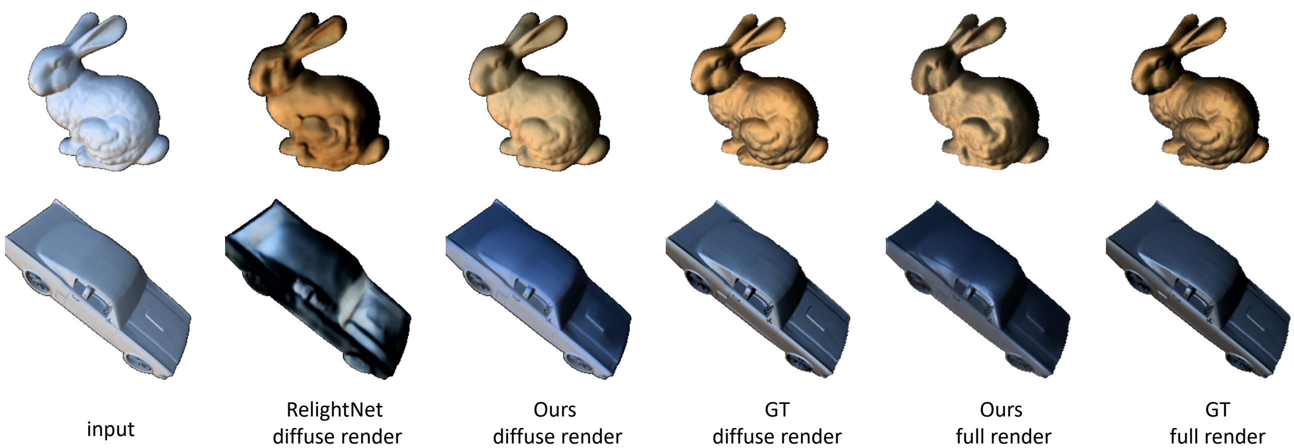

Figure 6. Comparisons of object relighting with RelightNet [34] and ground truths.

Table 3. Quantitative evaluation on relighting.

	Baseline*		RelightNet		Ours	
	MSE	DSSIM	MSE	DSSIM	MSE	DSSIM
Diffuse	0.2210	0.1350	0.1144	0.0788	**0.0926**	**0.0616**
With specularity	0.2152	0.1272	-	-	**0.0876**	**0.0720**

6.2. Image relighting

After inverse rendering, a differentiable non-Lambertian renderer is used to relight the object under new lighting. The rendering is efficient, taking 0.35 seconds per image at 256×256 on a single Tesla T4 GPU. For quantitative evaluations, we rendered an evaluation set of 100 objects under 30 lighting environments, with various materials. For each object, we use one image under one lighting as input, and relight it under the other 29 lighting for evaluation. We compare our method with a state-of-the-art method RelightNet [34], which only provides diffuse relighting. To be fair, we compare them on diffuse relighting only. Ours is evaluated for both diffuse and non-Lambertian relighting. Comparisons are shown in Figure 6 and Table 3, more in the supplementary material. Baseline* in the table denotes naive insertions without relighting. Object insertion and App demos are on the project page, where our method relights and inserts objects into new scenes realistically.

7. Conclusions

We present a single-image relighting approach based on weakly-supervised inverse rendering, driven by a large foreground-aligned video dataset and a low-rank constraint. We propose the differentiable specular renderer for low-frequency non-Lambertian rendering. Limitations including shadows and parametric models are discussed in the supplementary material.

Acknowledgements. We thank anonymous reviewers and AC for their valuable comments. We thank Kun Xu for the helpful discussions. We thank Sisi Dai and Kunliang Xie for data capturing. This work is supported in part by the National Key Research and Development Program of China (2018AAA0102200), NSFC (62002375, 62002376, 62132021), Natural Science Foundation of Hunan Province of China (2021JJ40696, 2021RC3071, 2022RC1104) and NUDT Research Grants (ZK22-52).

References

[1] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 3

[2] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2447–2456, 2019. 3

[3] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 7

[4] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Trans Pattern Anal Mach Intell (PAMI)*, 37(8):1670–1687, 2015. 3, 7

[5] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977. 2, 3

[6] Robin Green. Spherical harmonic lighting: The gritty details. In *Archives of the game developers conference*, volume 56, page 4, 2003. 2

[7] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, pages 2335–2342, 2009. 7

[8] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. In *NIPS*, pages 5936–5946, 2017. 3, 7

[9] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 2

[10] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *ECCV*, 2018. 3, 7

[11] Zhen Li, Lingli Wang, Xiang Huang, Cihui Pan, and Jiaqi Yang. Phyir: Physics-based inverse rendering for panoramic indoor images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12713–12723, 2022. 2

[12] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 3, 7

[13] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W. Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6123–6133, June 2021. 3, 7

[14] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised learning for intrinsic image decomposition from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3248–3257, 2020. 3, 7

[15] Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, et al. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[16] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6315–6324, 2018. 6

[17] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. In *Computer graphics forum*, volume 22, pages 663–687. Wiley Online Library, 2003. 1

[18] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3

[19] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 3

[20] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001. 2, 4, 6

[21] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001. 6

[22] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6296–6305, 2018. 3

[23] S. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985. 5

[24] Hui-Liang Shen and Zhi-Huan Zheng. Real-time highlight removal using intensity ratio. *Applied optics*, 52(19):4483–4493, 2013. 3

[25] Jian Shi, Yue Dong, Hao Su, and Stella X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 7

[26] Zhixin Shu, Sunil Hadap, Eli Shechtman, Kalyan Sunkavalli, Sylvain Paris, and Dimitris Samaras. Portrait lighting transfer using a mass transport approach. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017. 3

[27] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017. 3

[28] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single

image portrait relighting. *ACM Trans. Graph.*, 38(4):79–1, 2019. 3

[29] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017. 3

[30] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 3

[31] Takahisa Yamamoto and Atsushi Nakazawa. General improvement method of specular component separation using high-emphasis filter and similarity function. *ITE Transactions on Media Technology and Applications*, 7(2):92–102, 2019. 3

[32] Renjiao Yi, Ping Tan, and Stephen Lin. Leveraging multiview image sets for unsupervised intrinsic image decomposition and highlight separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12685–12692, 2020. 3, 5, 7

[33] Renjiao Yi, Chenyang Zhu, Ping Tan, and Stephen Lin. Faces as lighting probes via unsupervised deep highlight extraction. In *ECCV*, September 2018. 7

[34] Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William AP Smith. Self-supervised outdoor scene relighting. In *European Conference on Computer Vision*, pages 84–101. Springer, 2020. 1, 3, 7, 8

[35] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 1, 3, 7

[36] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. Adacoseg: Adaptive shape co-segmentation with group consistency loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8543–8552, 2020. 7