Differentiable Appearance Acquisition from a Flash/No-flash RGB-D Pair

Hyun Jin Ku[†] Hyunho Ha[†] Joo Ho Lee[†,*] Dahyun Kang[†] James Tompkin[§] Min H. Kim[†]

[†] KAIST [*] Sogang University [§] Brown University

Abstract—Reconstructing 3D objects in natural environments requires solving the ill-posed problem of geometry, spatially-varying material, and lighting estimation. As such, many approaches impractically constrain to a dark environment, use controlled lighting rigs, or use few handheld captures but suffer reduced quality. We develop a method that uses just two smartphone exposures captured in ambient lighting to reconstruct appearance more accurately and practically than baseline methods. Our insight is that we can use a flash/no-flash RGB-D pair to pose an inverse rendering problem using point lighting. This allows efficient differentiable rendering to optimize depth and normals from a good initialization and so also the simultaneous optimization of diffuse environment illumination and SVBRDF material. We find that this reduces diffuse albedo error by 25%, specular error by 46%, and normal error by 30% against single- and paired-image baselines that use learning-based techniques. Given that our approach is practical for everyday solid objects, we enable photorealistic relighting for mobile photography and easier content creation for augmented reality.

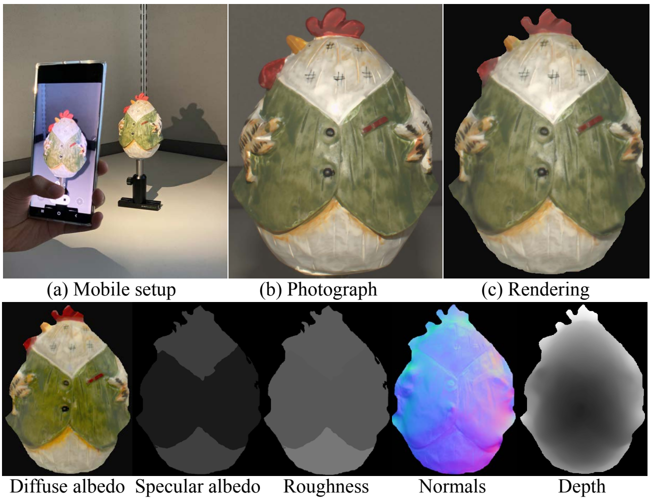
Index Terms—Appearance acquisition, inverse rendering, SVBRDF, flash photography.

1 Introduction

Modeling object appearance with geometry and spatially-varying bidirectional reflectance distribution functions (SVBRDFs) can create photorealistic renderings and allow simple appearance editing and relighting for photography and augmented reality. Acquiring high-quality SVBRDFs of real-world 3D objects requires dense sampling of view and light angles using a camera and active illumination on a mechanical gantry [1], [2], [3], [4]. Approaches that capture SVBRDFs with everyday devices like smartphones or DSLRs often restrict objects to planar geometry [5], [6], [7], [8], [9], [10] or require hundreds of multi-view input images [11], [12]. Multiple views requires accurate structure-from-motion and multiview stereo [13] to estimate camera parameters and build initial base geometry before starting any material estimation process.

To reduce the number of input views, learning-based approaches train to infer SVBRDF parameters from a small number of input images [16], [17], [18], [19]. These require a dataset of thousands to a hundred thousand synthetic rendering images to tackle the under-constrained problem. However, overfitting often limits the prediction accuracy of reflectance characteristics on unseen objects. Given the limitations of gantry, many view, and learning-based methods, the problem of practical appearance acquisition still stands.

One particular problem is specular reflections. These are difficult to separate from diffuse reflection when the number of views is low, and due to their high radiance cause material estimation to be less successful, especially when the geometry and normal are unknown. State of the art approaches use a flash/no-flash pair to separate specular reflections, but the unconstrained geometry still causes ambiguity. This again suggests deep learned priors, but in experiments we find that general reconstruction accuracy is still lacking. As learning approaches do not solve the problem, we must look to integrate any additional information to help solve practical appearance reconstruction.


(a) Mobile setup (b) Photograph (c) Rendering

Diffuse albedo Specular albedo Roughness Normals Depth

Fig. 1: **High-quality appearance reconstruction from a pair of exposures.** We use a flash/no-flash RGB-D pair from a smartphone (a) to recover geometry, lighting, and Cook-Torrance SVBRDF material properties (d). Re-renderings (c) are close to the original photograph (b) without using impractical lighting rigs or multiple capture positions.

We propose appearance acquisition by using additional sensors available on modern smartphones to capture one flash/no-flash RGB-D pair, using an efficient differentiable rendering optimization and material clustering. This has three benefits:

1) Depth is simple to capture with structured light or time of flight sensors in camera systems, and takes the place of geometry reconstructed from multi-view captures. However, captured depth is low quality, with noise,

		Nam et al. [11]	Schmitt et al. [12]	Barron et al. [14]	Cao et al. [15]	Sang et al. [16]	Boss et al. [17]	Ours
Setup	# views	Multi view	Multi view	Single view	Two views	Single view	Single view	Two views
	Active lights	Flash	Multiple lights	-	Flash/no-flash×2	Flash	Flash/no-flash	Flash/no-flash
	# images	100–400	10–40	1	3	1	2	2
	Geometry	MVS	MVS	Optional	Stereo	-	-	Depth sensor
Output	Depth	✓	✓	✓	✓	✓	✓	✓(refined)
	Normal	✓	✓	✓	✓	✓	✓	✓
	Diffuse albd.	✓	✓	✓	✓	✓	✓	✓
	Specular albd.	✓	✓	-	-	-	✓	✓
	Roughness	✓	✓	-	-	✓	✓	✓

TABLE 1: **Existing research on object appearance reconstruction has more limited practicality or quality than our method.** Nam et al. [11] and Schmitt et al. [12] capture SVBRDFs and complete 3D geometry, but require 10–400 input images with active illumination. Barron et al. [14] and Cao et al. [15] recover diffuse reflectance only, which limits flexibility. Sang et al. [16] capture a single image and use deep learned priors but do not estimate specular albedo. Boss et al. [17] is the closest related work as it estimates a complete SVBRDF from a pair of images at a single view. However, this uses deep learned priors that do not overcome the ill-posed reconstruction problem (Figure 10). Our method overcomes this using additional depth data and differentiable rendering to reduce errors by at least 25% (Table 2).

artifacts from specular reflections, and quantization errors. Accurate normals are critical for material reconstruction, but deriving normals from low-quality depth leads to poor results. Differentiable rendering and an incremental material clustering lets us integrate photometric constraints to jointly refine depth and normal geometry and estimate SVBRDF materials.

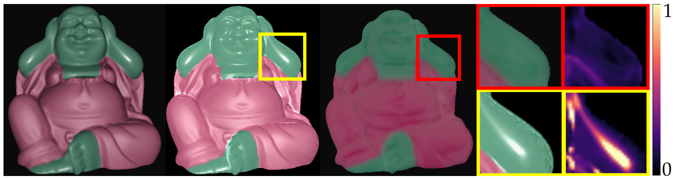
2) Physically-based differentiable rendering is slow. Using a flash/no-flash pair lets us reduce the hemispherical light integration problem to direct light from a known point, leading to a more efficient formulation.
3) We can optimize diffuse environment illumination via an alternating strategy, avoiding the need to capture objects in a dark room and providing additional diffuse albedo constraints across image formation models.

These benefits lead to higher-quality reconstruction: our approach reduces diffuse albedo error by 25%, specular error by 46%, and normal error by 30% against single- and paired-image baselines that use learning-based techniques. These improvements are gained without loss of practicality, allowing capture in lit indoor environments from an RGB-D camera at a single viewpoint. Collectively, our work moves toward 'point and shoot' digitization of object appearance.

2 Related Work

Object appearance reconstruction surveys cover progress up until 2016 [20], [21], often using special hardware like light stage acquisition platforms. Given our approach, we focus on recent practical acquisition systems including those using differentiable rendering (Table 1).

Diffuse Intrinsic Imaging. Intrinsic image decomposition [14], [22], [23] aims to separate a single image into diffuse reflectance and illumination-dependent shading. Since this problem is ill-posed, Bousseau et al. [24] use user scribbles to constrain diffuse albedo. Research has also used stereo or depth camera reconstructions as additional cues [25], [26], [27], [28]. For example, Cao et al. [15] use a stereo depth map and flash/no-flash images to reconstruct 3D geometry and diffuse albedo, and Haefner et al. [29] super-resolve depth using shape from shading via a diffuse surface assumption. Unfortunately, diffuse-only models rarely describe the real world as they ignore specularity. Our work does not assume diffuse-only reflectance as it recovers specular albedo and roughness based on the Cook-Torrance SVBRDF model [30], which improves quality (Figure 2).



(a) Flash-only (b) Diffuse albedo from Barron et al. [14] and ours (c) Error

Fig. 2: **Modeling specular appearance is essential.** Diffuse albedo recovered from Barron et al. [14] (b left) using flash-only image (a) and depth as input. Our approach with specular modeling given added no-flash input (b right). Since Barron et al. compute diffuse reflectance only, specular highlights cannot be factored from diffuse albedo. (c) shows difference from ground truth.

SVBRDFs on Planar Surfaces. To simplify the ill-posed inverse rendering problem, many works restrict objects to *planar* geometry [31]. Ren et al. [6] use a BRDF chart to reconstruct SVBRDFs under a moving light from video. Riviere et al. [8] and Hui et al. [7] reconstruct normals and SVBRDFs of a near-planar surface from multiple views. Aittala et al. [32] use an LCD and a DSLR camera to acquire multiple reflectance images for SVBRDF estimation, then later on [5] use a flash/no-flash pair to reconstruct SVBRDFs on planar objects. Given the ill-posed problem, deep learning may help. Deschaintre et al. [10] learn to recover normal and SVBRDF from a single image, Li et al. [33] use a self-augmented convolutional neural network (CNN) for SVBRDF estimation, and Gao et al. [9] use deep inverse rendering from an arbitrary number of images (single to many) to estimate SVBRDFs. Unlike these works, our approach does not assume that the object is planar, leading to a more general method.

SVBRDFs from Multi-view Images. Multi-view images from light stages and gantries help us recover highly accurate material appearance of real-world objects [1], [3], [4], [34], [35], [36], [37]. Given the size and expense of gantries, research has also investigated practical hand-held data acquisition, such as from smartphones [11] or compact custom imagers with multiple lights [12]. Ha et al. [38]

use an omnidirectional environment capture and many RGB-D captures to progressively update object shape and SVBRDFs parameters in a signed distance field. However, the authors state that their method is not robust to severe depth noise or errors. All these methods use many images—up to hundreds—to collect angular appearance samples and reconstruct good base geometry via structure from motion (SFM) and multi-view stereo (MVS) methods, typically for whole objects. Even then, this geometry may be in error and can be difficult to refine by optimization. Some current approaches attempt expensive multi-view differentiable path tracing for highly accurate reconstruction [39], [40]. We focus on accurate reconstruction from a single view using captured depth, which is easier and faster to refine than a mesh or SDF within a simultaneous geometry, material, and lighting optimization to help overcome depth errors.

SVBRDFs from a Single View Flash/No-flash Pair. With only one flash image [16], [18], [41] or a flash/no-flash pair [17], SVBRDF reconstruction of non-planar objects is highly ill-posed, leading to the use of deep-learned methods. Current methods are based on supervised learning that, given the large space of material appearance, require thousands of labeled or synthetically-rendered images for model training [16], [17], [18], [19], [37]). Models may not generalize depending on the characteristics of the training data, causing artifacts or failures in unseen test data [17], or causing accuracy drops with fewer input flash images [19]. Our technical novelty of using flash/no-flash captures with the noisy depth information now available on camera systems, via differentiable rendering, allows us to improve accuracy and maintain practicality without relying on a learned prior.

3 Method: Image Formation Model

We explain our method via the models and optimization terms, and defer implementation details to Section 5.

Image formation models approximate real-world image appearance from varying surface geometry, material, and lighting. Rendering SVBRDFs from complex environment lighting typically requires integrating over a hemisphere with a high-frequency lighting representation, which is difficult and slow to optimize. Instead, we separate lighting into low-frequency ambient diffuse illumination and direct illumination only from a point light—our flash.

SVBRDF Reflectance Model. We use the Cook-Torrance BRDF [30] that has diffuse albedo $\boldsymbol{\rho}$, specular albedo s, and surface roughness r terms to model material appearance. Object surface reflectance is a function of surface normal $\mathbf{n}$ as derived from surface geometry (for us, depth), incoming light vector $\boldsymbol{\omega}_i$, and view vector $\boldsymbol{\omega}_o$:

$$f(\mathbf{n},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) = \frac{\boldsymbol{\rho}}{\pi} + s\frac{\Psi(\mathbf{h})G(\mathbf{h},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)F(\mathbf{h},\boldsymbol{\omega}_i)}{4(\mathbf{n}\cdot\boldsymbol{\omega}_i)(\mathbf{n}\cdot\boldsymbol{\omega}_o)}. \tag{1}$$

Reflectance also depends upon the microfacet distribution Ψ for which we use GGX [42], the halfway vector $\mathbf{h} = (\boldsymbol{\omega}_i + \boldsymbol{\omega}_o)/||\boldsymbol{\omega}_i + \boldsymbol{\omega}_o||$, and the Smith geometric attenuation factor $G(\mathbf{h},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) \approx G_1(\mathbf{h},\boldsymbol{\omega}_i) \cdot G_1(\mathbf{h},\boldsymbol{\omega}_o)$. We assume the Fresnel term F is pre-integrated within the specular albedo term, which eases optimization without significant loss of accuracy for common objects [11].

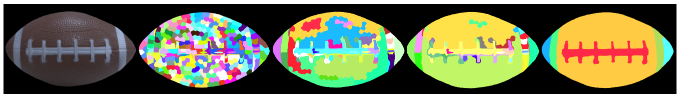


Fig. 3: **Specular parameter clustering.** *Left to right:* We compute initial superpixels of a no-flash image and then iteratively cluster according to chromaticity distance and boundary strength.

Unlike diffuse albedo, estimating accurate specular albedo s and roughness r requires dense angular samples. To avoid capturing multiple views, we assume that: 1) the 3D object surface orientation varies across the image, and 2) the object is made of a small number of materials that share specular parameters [11], [36], [43]. This lets us cluster materials, accumulate samples from different ray directions in the normal space of each material, and only estimate specular parameters per material: s^c and r^c.

Environment Lighting Model. We model diffuse illumination from the environment surrounding the object using 2nd-order spherical harmonics (SH), assuming that the environment illumination has a smooth shading effect across the object surface [44]. We parameterize diffuse illumination shading S as product of nine SH coefficients $\mathbf{g} \in \mathbb{R}^9$ and their basis functions with respect to normals $\mathbf{n}$: $\mathbf{H}(\mathbf{n})$. This lets us estimate diffuse reflection from illumination using diffuse albedo $\boldsymbol{\rho}$, produceing image I:

$$I = \boldsymbol{\rho} \sum_{k=0}^{8} \mathbf{g}_k \, \mathbf{H}_k(\mathbf{n}), \tag{2}$$

To compute diffuse shading S, we simply remove $\boldsymbol{\rho}$. Note that this model ignores specular reflection appearance, which we include in our next model.

Flash Lighting Model. Under a point light source, we model direct illumination only including specular reflection and produce image I via:

$$I(\mathbf{n},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,f) = f(\mathbf{n},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)(\mathbf{n}\cdot\boldsymbol{\omega}_i)L, \tag{3}$$

where our point light source has intensity L. Given the relatively long distance between the light and our object with respect the object size, we ignore the distance attenuation factor (d^2 law) over the object surface.

Discussion. This image formation model benefits differentiable rendering in two ways: 1) Modeling the flash as direct illumination only from a point light source removes an integral over incident light directions, leaving us with a simplified Eq. (3) that is easier and faster to optimize. 2) Modeling diffuse-only appearance via a separate environment lighting model lets us penalize the difference between the diffuse albedo recovered from both flash and environment lighting renderings (Eq. (7)), which improves quality.

4 Method: Differentiable Rendering Optim.

As input, we capture a pair of linear RAW RGB images without flash $\bar{I}$ and with flash $\dot{I}$, and a depth map D. To recover the effect on appearance of direct flash illumination, we obtain a *flash-only* image I from the RGB pair by subtracting the no-flash image from the flash image: $I = \dot{I} - \bar{I}$ (see Figure 5a for an example).

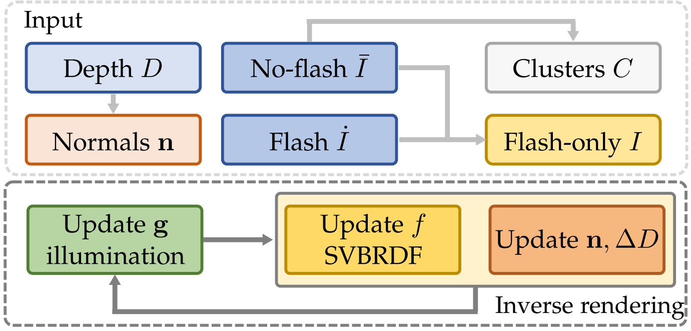


Fig. 4: **Normal, depth, SVBRDF, and illumination optimization.** From a captured depth map D, we create initial normal map $\mathbf{n}$ and no-flash image $\bar{I}$ to estimate initial environment illumination $\mathbf{g}$. Then, we use the flash image $\dot{I}$, normal map, and illumination for inverse rendering through iterative multi-scale joint optimization, yielding SVBRDFs f and refined depth and normals.

Given these inputs, our optimization proceeds in a two-step alternating strategy. In Step 1, we optimize the environment lighting via SH coefficients $\tilde{\mathbf{g}}$. In Step 2, we jointly optimize five elements: geometry as depth offsets ΔD and normals $\mathbf{n}$ as differential linearized rotation matrices $\Delta\mathbf{R}$ (twist rotation matrices [45]), and appearance as diffuse albedo $\boldsymbol{\rho}$, specular albedo per material s^c, and roughness per material r^c: $\boldsymbol{\chi} = \{\Delta D, \Delta\mathbf{R}, \boldsymbol{\rho}, s^c, r^c\}$. Figure 4 presents all inputs and the optimization strategy.

4.1 Initialization

Depth. The depth to optimize $\tilde{D}$ is initially set to the sensor depth D. The depth offsets are set to zero: $\Delta D = 0$.

Normals. We initialize $\mathbf{n}$ from D by finite differences.

SVBRDF. We initialize diffuse albedo $\boldsymbol{\rho}$ as the no-flash image. We initialize specular albedo $s = 0.05$ and roughness $r = 0.15$ for all material clusters.

Environment shading. We set diffuse shading S from luminance levels of no-flash image $\bar{I}$, assuming that initial diffuse albedo is 50% of the reflectance: $S = Y(\bar{I}/0.5)$, where $Y(\cdot)$ is a luminance function.

4.2 Step 1: Environment Illumination

Given normal image $\mathbf{n}$ and diffuse shading image S, we minimize a least-squares objective:

$$\tilde{\mathbf{g}} = \underset{\mathbf{g}}{\operatorname{argmin}} \sum_{(i,j)\in M} \|S - \mathbf{g}\,\mathbf{H}(\mathbf{n})\|_2^2 , \tag{4}$$

where $\tilde{\mathbf{g}}$ denotes the optimized SH illumination coefficients and M indicates valid pixels. This is similar to the method of Wu at al. [46], which also uses 2nd-order SH diffuse illumination within an alternating shape from shading optimization for normal refinement.

4.3 Step 2: Joint SVBRDF & Normal Optimization

Our joint objective has geometric and photometric terms:

$$\begin{aligned}\tilde{\boldsymbol{\chi}} = \underset{\boldsymbol{\chi}}{\operatorname{argmin}}\ & \lambda_P\psi_P + \lambda_{\boldsymbol{\rho}}\psi_{\boldsymbol{\rho}} + \lambda_{\boldsymbol{\rho}}^{\text{reg}}\psi_{\boldsymbol{\rho}}^{\text{reg}} + \\ & \lambda_D\psi_D + \lambda_{\mathbf{n}}\psi_{\mathbf{n}} + \lambda_{\mathbf{n}}^{\text{reg}}\psi_{\mathbf{n}}^{\text{reg}} .\end{aligned} \tag{5}$$

The first three terms are radiometric: a photometric term ψ_P, a diffuse albedo term $\psi_{\boldsymbol{\rho}}$, and a diffuse albedo regularization term $\psi_{\boldsymbol{\rho}}^{\text{reg}}$. The second three terms are geometric: a depth term ψ_D, a normal term $\psi_{\mathbf{n}}$, and a normal regularization term $\psi_{\mathbf{n}}^{\text{reg}}$. Each term has a corresponding hyperparameter λ.

Photometric Consistency. Our photometric term ψ_P encourages similarity between the input flash-only image I and the rendered flash-only image $\tilde{I}$ via Equation (3):

$$\psi_P = \sum_{(i,j)\in M} \left\| I - \tilde{I}(\tilde{\mathbf{n}}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, f)\right\|_1 , \tag{6}$$

where we use $\tilde{D}$ to calculate light $\boldsymbol{\omega}_i$ and view $\boldsymbol{\omega}_o$ vectors.

Diffuse Albedo. The origin of bright regions can be ambiguous: are they from bright diffuse albedo, large specularity, or smooth object surfaces? We can add an additional cue from diffuse albedo computed with SH illumination shading to help mitigate this ambiguity (Figure 5).

Given SH illumination coefficients $\mathbf{g}$ from Step 1, we approximate diffuse shading $S = \mathbf{g}\,\mathbf{H}(\mathbf{n})$ (Equation (4)). Then, we divide each color channel of the no-flash image $\bar{I}$ by the diffuse shading S to approximate diffuse albedo: $\bar{\boldsymbol{\rho}}_{\{r,g,b\}} = \bar{I}_{\{r,g,b\}}/S$.

We also approximate diffuse albedo a second way by computing shading from direct illumination S_d via Equation (3) by ignoring SVBRDF f, then dividing the flash-only image by this shading: $\boldsymbol{\rho}_{\{r,g,b\}} = I_{\{r,g,b\}}/S_d$. Then, we encourage both estimates to be similar:

$$\psi_{\boldsymbol{\rho}} = \sum\nolimits_{(i,j)\in M} \|\boldsymbol{\rho}(i,j)/\tau - \bar{\boldsymbol{\rho}}(i,j)\|_1 . \tag{7}$$

As the scales of these diffuse albedos are different due to intensity differences of the flash/no-flash images, we normalize using factor $\tau = I^\mu + 3 \cdot I^\sigma$, where I^μ and I^σ are the flash-only image mean and standard deviation. This normalization factor also guides the albedo to lie in the physically-meaningful range of [0,1].

Diffuse albedo can spatially vary, and so we enforce edge-aware smoothness with regularizer $\psi_{\boldsymbol{\rho}}^{\text{reg}}$:

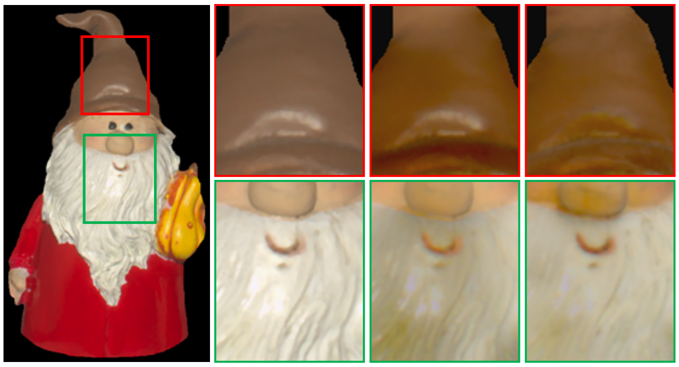
$$\begin{aligned}\psi_{\boldsymbol{\rho}}^{\text{reg}} = & \sum_{(i,j)\in M} w_{\text{i}} \,\|\boldsymbol{\rho}(i+1,j) - \boldsymbol{\rho}(i,j)\|_2^2 + \\ & \sum_{(i,j)\in M} w_{\text{j}} \,\|\boldsymbol{\rho}(i,j+1) - \boldsymbol{\rho}(i,j)\|_2^2 ,\end{aligned} \tag{8}$$

where w_{i} and w_{j} are bilateral weights computed from the SH illumination diffuse albedo:

$$\begin{aligned} w_{\text{i}} &= \exp\left(-\frac{||\bar{\boldsymbol{\rho}}(i+1,j) - \bar{\boldsymbol{\rho}}(i,j)||_2^2}{2\sigma^2}\right) , \\ w_{\text{j}} &= \exp\left(-\frac{||\bar{\boldsymbol{\rho}}(i,j+1) - \bar{\boldsymbol{\rho}}(i,j)||_2^2}{2\sigma^2}\right) , \end{aligned} \tag{9}$$

and where Gaussian standard deviation is $\sigma = 0.01$.

Depth. We optimize depth values $\tilde{D}$ via the depth offsets ΔD: $\tilde{D} = D + \Delta D$. Our depth term ψ_D encourages similarity



Flash-only image (a) Flash-only (b) Without $\psi_{\boldsymbol{\rho}}$ (c) With $\psi_{\boldsymbol{\rho}}$

Fig. 5: **Constraining diffuse albedo from SH illumination to be close to diffuse albedo from direct illumination improves quality.** For an input flash-only image (a), the results in (b,c) compare the effect of the diffuse albedo term $\psi_{\boldsymbol{\rho}}$ (Eq. (7)). This term helps to separate SVBRDF parameters.

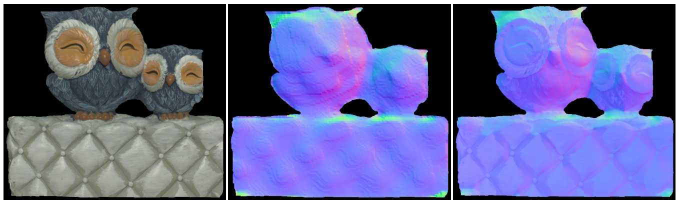


Fig. 6: **Differentiable rendering refines noisy normals.** Input normals (middle) show low detail and quantization artifacts. Our refined normals (right) recover detail while maintaining accuracy.

between the current depth $\tilde{D}$ and the sensor depth D:

$$\psi_D = \sum\nolimits_{(i,j)\in M} ||\tilde{D}(i,j) - D(i,j)||^2. \tag{10}$$

Normal. As we optimize depth and normal as separate parameters, we encourage integrability via normal term $\psi_{\mathbf{n}}$:

$$\psi_{\mathbf{n}} = \sum\nolimits_{(i,j)\in M} ||\tilde{\mathbf{n}}(i,j) - \dot{\mathbf{n}}(i,j)||_2^2, \tag{11}$$

where $\tilde{\mathbf{n}}$ denotes the current normal rotated by the optimized rotation $\tilde{\mathbf{n}} = \Delta\mathbf{R}(\mathbf{n})$, and $\dot{\mathbf{n}}$ denotes normals obtained from depth $\tilde{D}$ following Wu et al. [46].

To mitigate depth sensor noise (Figure 6), we use normal regularization term $\psi_{\mathbf{n}}^{\text{reg}}$ to make sure normals have edge-aware smooth changes with respect to neighboring pixels:

$$\begin{aligned}\psi_{\mathbf{n}}^{\text{reg}} = &\sum_{(i,j)\in M} w_{\text{i}} \left\|\tilde{\mathbf{n}}(i+1,j) - \tilde{\mathbf{n}}(i,j)\right\|_2^2 + \\ &\sum_{(i,j)\in M} w_{\text{j}} \left\|\tilde{\mathbf{n}}(i,j+1) - \tilde{\mathbf{n}}(i,j)\right\|_2^2,\end{aligned} \tag{12}$$

where w_{i} and w_{j} are as in Equation (9) with $\sigma = 0.1$.

5 Method: Implementation Details

Calibration. To align color and depth camera pixels, we recover intrinsic and extrinsic camera parameters using Zhang's method [47]. For extrinsic calibration, we use a pair of color images and an infrared image from the depth camera. Then, we warp the depth camera pixels to the color camera space to use the higher-resolution color information.

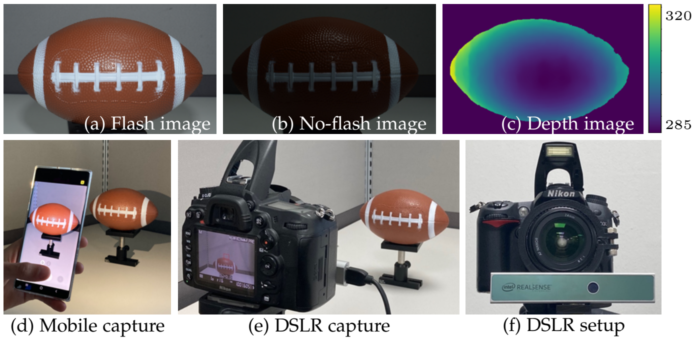


Fig. 7: **Real-world data capture.** Flash/no-flash pair (a,b) and a depth map (c) from either a smartphone with an RGB-D sensor (d) or a DSLR with a depth camera (e) & (f).

After projecting the depth map, some color pixels do not have depth values due to occlusion by parallax. Thus, we build a visibility mask M for each color image pixel (i,j), and compute our optimization loss only for visibile pixels.

Material Clustering. To model specular albedo s^c and roughness r^c per material, we formulate a graph simplification problem based on Kruskal's algorithm [48], where nodes are material clusters and where edges join adjacent clusters. We initialize material clusters using SLIC superpixels [49] from the no-flash image (Figure 3). Edge weights are set as the average L2 chromaticity (UV) difference between the constituent pixels of node superpixels, and we compute a boundary strength score between adjacent clusters from the range of pixel intensities. Then, for each edge in increasing weight order, we merge two clusters if they have a weak boundary score and similar average color values. Upon merging, we update the graph and recompute edge and boundary scores. Then, we iterate until either 1) no clusters are merged such as when adjacent color distances are larger than a preset threshold value, or 2) we reach a minimum number of clusters per object (manually specified).

Over our various test objects, this process consistently produces large homogenous regions without merging superpixels of similar albedo but that are separated by strong features (Figure 3). Since the material clustering algorithm depends on chromaticity and adjacent pixel values, similar but distant materials are not classified as the same material, and nearby materials with similar colors cannot be classified as different materials.

Coarse-to-fine Algorithm. To aid the optimization, we take a

Algorithm 1 Coarse-to-fine optimization.

```
Initialize parameters $\boldsymbol{\chi} = \{\Delta D_K, \Delta\mathbf{R}_K, \boldsymbol{\rho}_K, s_K^c, r_K^c\}$
for level $k = K \ldots 0$ do
    $I_k \leftarrow I_k / \tau_k$
    Step 1: Optimize $\mathbf{g}$ using no-flash image $\bar{I}$
    Step 2: Optimize with flash-only image $I_k$: $\tilde{\boldsymbol{\chi}} = \{\Delta D_k, \Delta\mathbf{R}_k, \boldsymbol{\rho}_k, s_k^c, r_k^c\}$
    if $k > 1$ then
        $\{D_{k-1}, \mathbf{n}_{k-1}, \boldsymbol{\rho}_{k-1}\} \leftarrow \text{UPSAMPLE}(\{D_k + \Delta D_k, \Delta\mathbf{R}_k(\mathbf{n}_k), \boldsymbol{\rho}_k \cdot \tau_k\})$
        $s_{k-1} \leftarrow s_k^c \cdot \tau_k$
        $r_{k-1} \leftarrow r_k^c$
    end if
end for
```

multiscale approach (Algorithm 1). We use the optimization results from the previous coarse level for the initialization of the current level optimization. We optimize four hierarchy levels (k=0...K, K=3) and perform $100{\cdot}k/(K{+}1)$ iterations at each level, using area interpolation for initial downsampling and bilinear interpolation for upsampling after every level.

After each iteration, we clip specular albedo s and roughness r to be at least $\epsilon{=}1{\times}10^{-7}$. Also, since we update our normal at every iteration $\tilde{\mathbf{n}} = \mathbf{\Delta R}(\mathbf{n})$, we set the differential rotation matrix to the identity $\mathbf{\Delta R} = \mathbf{I}$ after every iteration. Finally, when upsampling images between levels, we renormalize diffuse albedos by τ.

Optimization Hyperparameters. We use Adam [50] with a learning rate of 0.01 and a weight decay rate of 0.6 over every 30 iterations. Each optimization function is composed of different optimization parameters with different unit ranges. Thus, we use hyperparameters both to align these ranges and to determine the relative importance of each subobjective. To find good hyperparameters, we employ grid search. For synthetic scenes, we use $\lambda_P = 6 \times 10^4$, $\lambda_\rho = 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^3$, $\lambda_D = 1$, $\lambda_{\boldsymbol{n}} = 10^5$, $\lambda_{\boldsymbol{n}}^{\text{reg}} = 5 \times 10^4$. For real scenes with more imaging noise and greater variety in size and material type, we fine tune the hyperparameters for each object. Please refer to the supplement for hyperparameters of each object. For owls, we use $\lambda_P = 5 \times 10^4$, $\lambda_\rho = 4 \times 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^6$, $\lambda_D = 10^3$, $\lambda_{\boldsymbol{n}} = 10^5$, $\lambda_{\boldsymbol{n}}^{\text{reg}} = 10^5$.

Software, Hardware, and Computation Time. We implement our method using TensorFlow2 [51]. When executing upon an NVIDIA GPU Titan V and Intel CPU i7-9700 with an input image size of 0.5 Mpx (Figure 8) and with four hierarchical levels, our implementation takes 150 seconds for material clustering and 88 seconds for optimization.

6 Experiments

Baselines. We compare our method to the closest two practical SVBRDF works that use only one or two input images (Table 1): Sang et al. [16] and Boss et al. [17]. Both use deep learned priors whereas we forego these and use additional sensor depth information. While this means these works are not directly comparable, we judged these comparisons to be informative given new smartphone capabilities.

Sang et al. [16] use a flashlight to estimate depth, normal, diffuse albedo, and roughness, and Boss et al. [17] use a pair of flash/no-flash images to estimate appearance and geometry. We provide a flash-only image to Sang et al. [16], a pair of flash and no-flash images to Boss et al. [17], and a depth map to our method.

Datasets: Synthetic and Real World. We use the synthetic dataset of 20 test scenes provided by Boss et al. [17] (Figure 10). Using the perfect depth from these scenes would be unfair to other methods as we expect noisy sensor depth as input, so we add Gaussian random noise of varying standard deviation $\sigma = 0.001, 0.005, 0.01$ to the input depth. For environment illumination, we extract nine SH coefficients of monochromatic illuminance from the Grace HDR environment map (Grace). Then, we render flash and no-flash images with an additional point light source, which is collocated with the camera position.

For real-world scenes, we use two setups to show a range of quality (Figure 7): (1) a handheld smartphone (Samsung Galaxy Note 10+) with an RGB sensor resolution of 4000×3000 and a depth sensor resolution of 640×480 (Figures 1 and 8), and (2) a tripod-mounted Nikon DSLR camera (D7000) with a sensor resolution of 4928×3264 and a 24 mm lens and a depth sensor (Intel RealSense SR305) with a resolution of 640×480 (Figure 9). Given limited dynamic range, we assume that flash intensity is stronger than environment illumination intensity to obtain a high signal-to-noise ratio (SNR) flash-only image. To ensure this, we fix ISO at 100 and vary exposure time.

Metrics. For depth, we use scale and shift invariant (affine similarity) mean-squared error (MSE) between the ground truth and the predicted value [52]. For normals, we use average angular error. For diffuse albedo, we compute MSE. Finally, owing to model rendering equation differences in that Sang et al. [16] does not estimate specular albedo, we compute a combined specular reflection image error as MSE rather than individual specular albedo and roughness, and refer to qualitative results for comparisons to Boss et al. [17] otherwise. Since the reflectance image has the same value range over different methods, direct comparison of MSE makes more sense than using scale-invariant MSE.

Qualitative Assessment and Comparison. Our approach improves depth and normal estimates over their initializations within the optimization process. The final depth maps and normals show less noise and higher resolution, up to the flash/no-flash image size (Figures 6 and 10), for both the smartphone (Figure 8) and the DSLR setup (Figure 9).

Table 2 quantitatively compares the accuracy of refined depth and normal, along with estimated diffuse albedo and specular reflection on the synthetic data. Our algorithm outperforms the two learning-based methods on average. Qualitatively, in Figure 10, our normal estimates occasionally lack high frequency detail present in the ground truth, but our depth input (even when noisy) and differentiable rendering lead them to rarely have large error. In contrast, the method of Sang et al. erroneously flattens the normals, and the method of Boss et al. erroneously introduces sharp boundaries. Material clustering leads our method to produce more accurate specular albedo, but all methods struggle to reconstruct surface roughness—learning or not.

For real-world scenes (Figures 8 and 9), we see similar trends. Glossy objects like the orange and Santa's hat cause the method of Boss et al. [17] bake variation into diffuse albedo rather than specular. Sang et al. [16] show better results for diffuse albedo, but again suffer geometrically with flattened normals and normals that are inconsistent with the predicted depth. Concerning relighting, Sang et al. both underestimate and overestimate specular appearance (Santa, missing specular on hat; orange, too bright and sharp specular lobe), and Boss et al. show more limited results due to the baking issue. While our result suffers some normal inaccuracy (Santa beard), overall our results improve quality without risking overfitting to data and notably improve normal and depth quality over the input (please zoom in).

Limitations. For materials, the Cook-Torrance SVBRDF model can only represent certain materials. For instance, transmissive materials that show subsurface scattering will cause inaccurate reconstruction, such as skin or rock crystals like quartz. Also, to ease the inverse rendering problem,

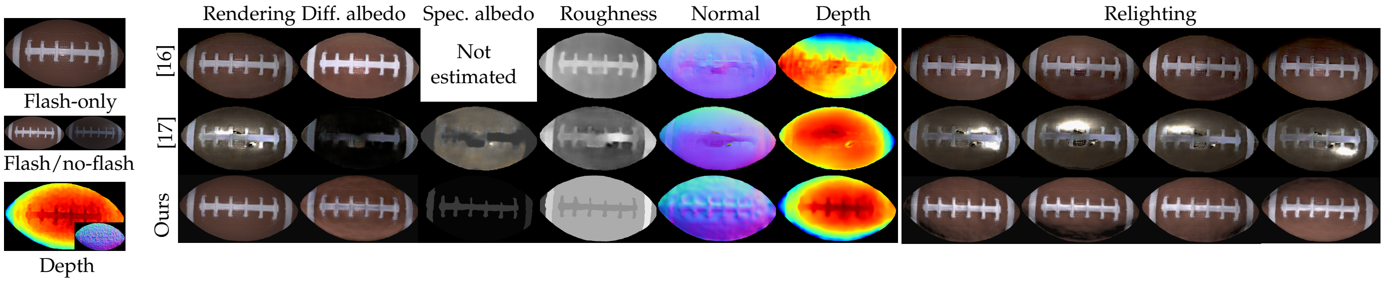


Fig. 8: **Real-world object smartphone capture.** The method of Sang et al. [16] produces reasonable appearance but with flattened normals and erroneous depth, and the method of Boss et al. [17] struggles to produce plausible material decomposition. Our method provides better results given low-quality depth and normals from the smartphone as its differentiable renderer can refine them (please zoom).

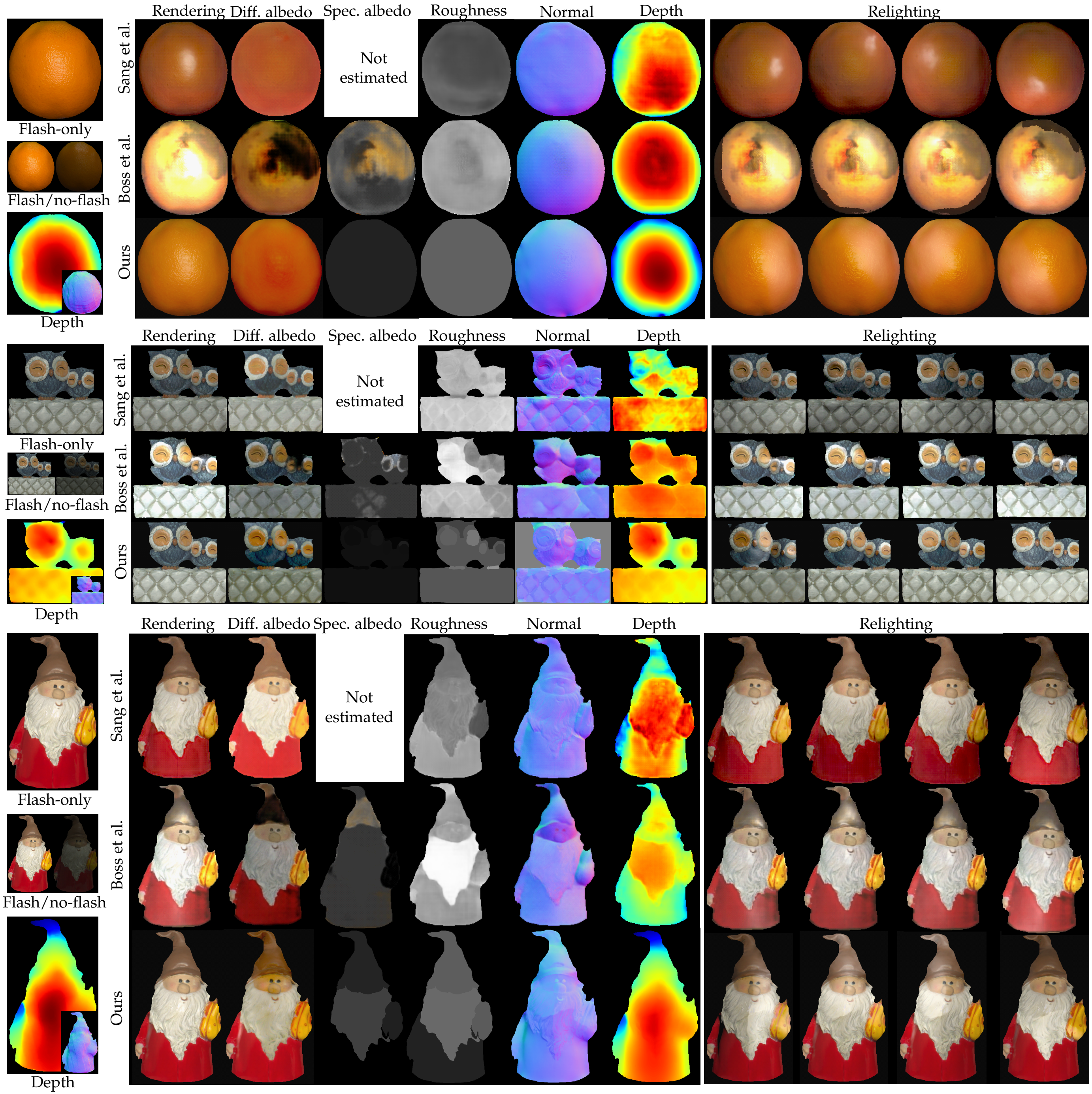


Fig. 9: **Real-world object DSLR + depth capture.** Normals estimated by Sang et al. [16] appear flat, and they do not estimate specular albedo. For Boss et al. [17], often diffuse albedo and specular albedo are not separated properly. Our method more successfully factorizes appearance and normals. Relighting results are illuminated with a point light at different positions in a counter-clockwise direction.

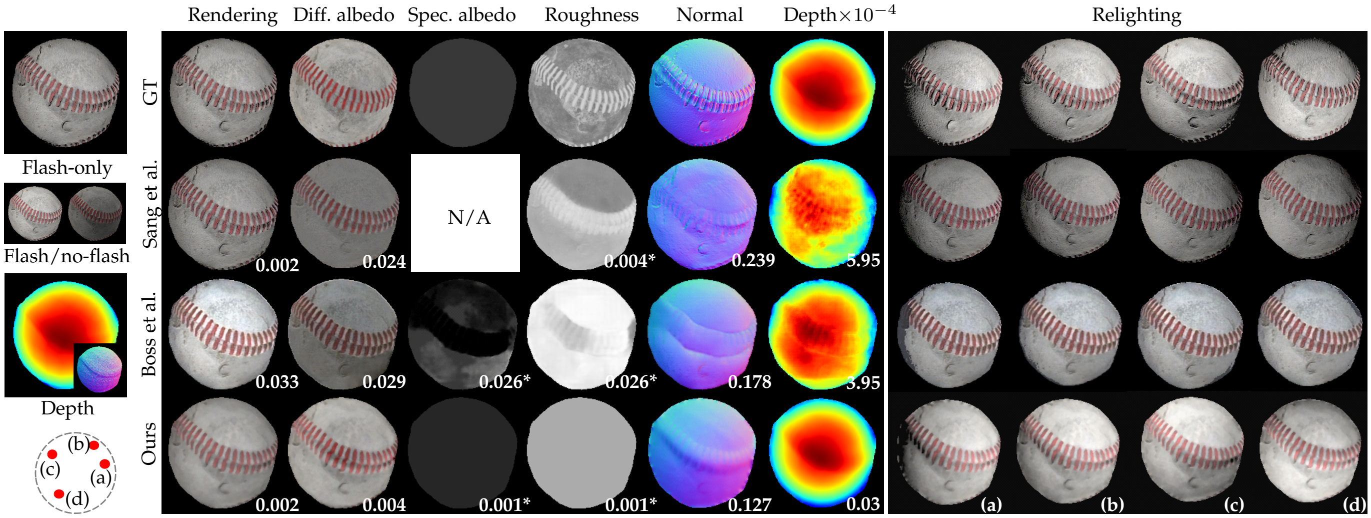
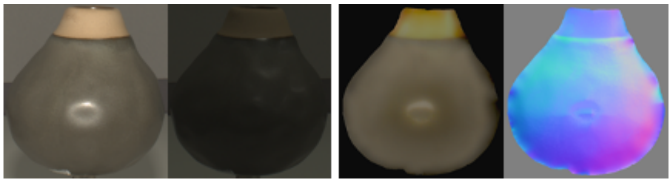


Fig. 10: **Our approach qualitatively improves over two learning-based single view flash/no-flash methods.** Synthetic baseball scene from the 20 scenes of Boss et al. [17]. Our method outperforms both Sang et al. [16] and Boss et al. [17], showing better SVBRDFs estimation and consistent 3D shape results. This lets our relighting be closer to the ground truth (point light positions shown in bottom left circle).

Method	Diffuse albedo	Specular reflection	Normal angle diff.	Depth ($\times 10^{-4}$)
Sang et al. [16]	0.024	0.013	0.446	6.082
Boss et al. [17]	0.038	0.176	0.353	3.162
Ours (avg. over σ)	**0.018**	**0.007**	**0.240**	**2.676**
Ours ($\sigma = 0.001$)	0.017	0.007	0.232	1.898
Ours ($\sigma = 0.005$)	0.018	0.007	0.243	3.054
Ours ($\sigma = 0.01$)	0.018	0.007	0.246	3.082

TABLE 2: **Our approach lowers quantitative error over two learning-based single view flash/no-flash methods.** We compute average MSE over 20 synthetic scenes from Boss et al. [17]. Bold marks lowest error. Varying Gaussian noise with standard deviation σ added to our input depth does not increase material error.



(a) Flash and no-flash images (b) Diffuse albedo (c) Normals

Fig. 11: **Limitation: Saturated pixels cause errors.** If input flash image has saturated pixels, the quality of the optimized diffuse albedo and normals is degraded.

we assume that material surfaces are dielectric: That is, we assume that the specular albedo should be monochromatic and have the same color as the illumination. Clustering for specular parameters may fail to group materials with high-frequency patterns, leading to slightly different reconstructions across the same material. Further, in our application scenario, specularity is optimized by reducing the rendered image loss, and so specular albedo and roughness are bound together. This can lead to small rendering errors, e.g., Figure 9 red dress has lower specular albedo but higher roughness.

Our algorithm requires that the input flashlight image should not be saturated. At saturated pixels, the albedo cannot be guided well by the no-flash image and normals can be overfit to inaccurate results (Figure 11). Further, for lighting, complex environment illumination causing object or scene shadows cannot be represented with only nine SH coefficients, nor any illumination that causes specular reflections. Finally, outdoor scenes remain difficult because the maximum intensity of the flashlight is relatively low compared to the sun, requiring high signal to noise ratios and high precision sensors.

7 Conclusion

Practical geometry and SVBRDF reconstruction is an ill-posed problem that has previously been tackled with deep learned priors. However, these can fail to generalize to unseen data, causing inaccuracy or even catastrophic failure (Boss et al. [17] on the orange, Figure 9). Given the challenge, we show that better accuracy is possible by integrating information from depth cameras, so long as the low input accuracy can be overcome. We show how to do this efficiently via differentiable rendering and a flash/no-flash pair that lets us estimate environment lighting and enforce additional diffuse albedo constraints. This approach allows capture in lit environments without any setup more complex than a smartphone, helping us move toward 'point and shoot' digitization of real-world object appearance.

Acknowledgements

Min H. Kim acknowledges the MSIT/IITP of Korea (RS-2022-00155620 and 2017-0-00072) and the Samsung Research Funding Center (SRFC-IT2001-04) for developing partial 3D imaging algorithms, in addition to the support of the NIRCH of Korea (2021A02P02-001), Samsung Electronics, and Microsoft Research Asia. James Tompkin acknowledges US NSF CAREER-2144956.

References

[1] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Trans. Graph.*, 2003.

[2] A. Ghosh, T. Hawkins, P. Peers, S. Frederiksen, and P. Debevec, "Practical modeling and acquisition of layered facial reflectance," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, p. 139, 2008.

[3] M. Holroyd, J. Lawrence, and T. Zickler, "A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance," in *ACM SIGGRAPH 2010 Papers*. New York, NY, USA: Association for Computing Machinery, 2010.

[4] C. Schwartz, R. Sarlette, M. Weinmann, and R. Klein, "Dome ii: A parallelized btf acquisition system," in *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition*. Eurographics Association, Jun. 2013.

[5] M. Aittala, T. Weyrich, and J. Lehtinen, "Two-shot svbrdf capture for stationary materials," *ACM Trans. Graph.*, 2015.

[6] P. Ren, J. Wang, J. Snyder, X. Tong, and B. Guo, "Pocket reflectometry," *ACM Trans. Graph.*, 2011.

[7] Z. Hui, K. Sunkavalli, J. Lee, S. Hadap, J. Wang, and A. C. Sankaranarayanan, "Reflectance capture using univariate sampling of brdfs," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[8] J. Riviere, P. Peers, and A. Ghosh, "Mobile surface reflectometry," *Computer Graphics Forum*, vol. 35, no. 1, pp. 191–202, 2016.

[9] D. GAO, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019.

[10] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Single-image svbrdf capture with a rendering-aware deep network," *ACM Trans. Graph.*, 2018.

[11] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim, "Practical svbrdf acquisition of 3d objects with unstructured flash photography," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2018)*, 2018.

[12] C. Schmitt, S. Donne, G. Riegler, V. Koltun, and A. Geiger, "On joint estimation of pose, geometry and svbrdf from a handheld scanner," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[14] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[15] X. Cao, M. Waechter, B. Shi, Y. Gao, B. Zheng, and Y. Matsushita, "Stereoscopic flash and no-flash photography for shape and albedo recovery," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[16] S. Sang and M. Chandraker, "Single-shot neural relighting and svbrdf estimation," in *ECCV*, 2020.

[17] M. Boss, V. Jampani, K. Kim, H. P. Lensch, and J. Kautz, "Two-shot spatially-varying brdf and shape estimation," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Learning to reconstruct shape and spatially-varying reflectance from a single image," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 269.

[19] D. Lichy, J. Wu, S. Sengupta, and D. W. Jacobs, "Shape and material capture at home," in *CVPR*, 2021.

[20] M. Weinmann and R. Klein, "Advances in geometry and reflectance acquisition (course notes)," in *SIGGRAPH Asia 2015 Courses*. ACM, 2015, pp. 1:1–1:71.

[21] D. Guarnera, G. Guarnera, A. Ghosh, C. Denk, and M. Glencross, "Brdf representation and acquisition," *Computer Graphics Forum*, vol. 35, pp. 625–650, 2016.

[22] M. F. Tappen, W. T. Freeman, and E. H. Adelson, "Recovering intrinsic images from a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[23] S. Bi, X. Han, and Y. Yu, "An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Trans. Graph.*, 2015.

[24] A. Bousseau, S. Paris, and F. Durand, "User-assisted intrinsic images," in *ACM SIGGRAPH Asia 2009 Papers*, ser. SIGGRAPH Asia '09. New York, NY, USA: Association for Computing Machinery, 2009.

[25] K. J. Lee, Q. Zhao, X. Tong, M. Gong, S. Izadi, S. U. Lee, P. Tan, and S. Lin, "Estimation of intrinsic image sequences from image+depth video," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ser. ECCV'12, 2012.

[26] Q. Chen and V. Koltun, "A simple model for intrinsic image decomposition with depth cues," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[27] L. Yu, S. Yeung, Y. Tai, and S. Lin, "Shading-based shape refinement of rgb-d images," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[28] J. Jeon, S. Cho, X. Tong, and S. Lee, "Intrinsic image decomposition using structure-texture separation and surface normals," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, 2014.

[29] B. Haefner, S. Peng, A. Verma, Y. Quéau, and D. Cremers, "Photometic depth super-resolution," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[30] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Trans. Graph.*, 1982.

[31] Y. Guo, C. Smith, M. Ha✓san, K. Sunkavalli, and S. Zhao, "Materialgan: Reflectance capture using a generative svbrdf model," *ACM Trans. Graph.*, vol. 39, no. 6, nov 2020. [Online]. Available: https://doi.org/10.1145/3414685.3417779

[32] M. Aittala, T. Weyrich, and J. Lehtinen, "Practical svbrdf capture in the frequency domain," *ACM Trans. Graph.*, 2013.

[33] X. Li, Y. Dong, P. Peers, and X. Tong, "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks," *ACM Transactions on Graphics*, vol. 36, 2017.

[34] J. Riviere, P. Gotardo, D. Bradley, A. Ghosh, and T. Beeler, "Single-shot high-quality facial geometry and skin appearance capture," *ACM Trans. Graph.*, vol. 39, no. 4, Jul. 2020.

[35] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec, "Acquiring reflectance and shape from continuous spherical harmonic illumination," *ACM Transactions on Graphics (TOG)*, vol. 32, 07 2013.

[36] G. Nam, J. H. Lee, H. Wu, D. Gutierrez, and M. H. Kim, "Simultaneous acquisition of microscale reflectance and normals," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2016)*, 2016.

[37] S. Bi, Z. Xu, K. Sunkavalli, D. Kriegman, and R. Ramamoorthi, "Deep 3d capture: Geometry and reflectance from sparse multi-view images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] H. Ha, S. Baek, G. Nam, and M. H. Kim, "Progressive acquisition of svbrdf and shape in motion," *Computer Graphics Forum*, 2020.

[39] P. Goel, L. Cohen, J. Guesman, V. Thamizharasan, J. Tompkin, and D. Ritchie, "Shape from tracing: Towards reconstructing 3d object geometry and svbrdf material from images via differentiable path tracing," in *International Conference on 3D Vision (3DV)*, 2020.

[40] F. Luan, S. Zhao, K. Bala, and Z. Dong, "Unified shape and svbrdf recovery using differentiable monte carlo rendering," *Computer Graphics Forum*, vol. 40, no. 4, pp. 101–113, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14344

[41] Z. Li, K. Sunkavalli, and M. Chandraker, "Materials for masses: Svbrdf acquisition with a single mobile phone image," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 74–90.

[42] B. Walter, S. Marschner, H. Li, and K. Torrance, "Microfacet models for refraction through rough surfaces." 01 2007, pp. 195–206.

[43] Z. Zhou, G. Chen, Y. Dong, D. Wipf, Y. Yu, J. Snyder, and X. Tong, "Sparse-as-possible svbrdf acquisition," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 189, 2016.

[44] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001.

[45] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[46] C. Wu, M. Zollhöfer, M. Nießner, M. Stamminger, S. Izadi, and C. Theobalt, "Real-time shading-based refinement for consumer depth cameras," 2014.

[47] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[48] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[50] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations (ICLR)*, 12 2014.

[51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[52] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

APPENDIX

In the supplemental material, we provide quantitative evaluation results for the 20 unseen test dataset [17]. We use Mean Square Error (MSE) to compute diffuse albedo L2 difference in the linear domain (Table 3). For specular albedo and roughness, since Sang et al. [16] do not compute specular albedo, instead we calculate specular reflectance MSE as our specular error metric (Table 4). To evaluate surface normals, computing direct L2 difference is not appropriate since normals from learning-based methods are not guaranteed to have unit length magnitude. Thus, we compute the average angular difference between predicted normals and ground truth normals (Table 5) by following the method of Schmitt et al. [12]. In a similar sense, depth computed from different methods has different scales. Thus, we compute scale and shift-invariant (affine similarity) error metric (Table 6) by following Ranftl et al. [52].

For hyperparameters, once a moderately-robust interval is found, optimal hyperparameters can be further refined by sweeping a hyperparameter table. We tested these parameters: $\lambda_P = [10^4, 10^5, 10^6]$, $\lambda_\rho = [10^4, 10^5, 10^6]$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = [10^5, 10^6, 10^7]$, $\lambda_D = [1, 10, 10^2, 10^3]$, $\lambda_{\mathbf{n}} = [10^4, 10^5, 10^6]$, $\lambda_{\mathbf{n}}^{\text{reg}} = [10^4, 10^5, 10^6, 10^7]$.

Specifically, for *rugby* we use, $\lambda_P = 4 \times 10^4$, $\lambda_\rho = 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^6$, $\lambda_D = 10^3$, $\lambda_{\mathbf{n}} = 10^5$, $\lambda_{\mathbf{n}}^{\text{reg}} = 10^5$. For *chicken* we use, $\lambda_P = 6 \times 10^4$, $\lambda_\rho = 5 \times 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 7 \times 10^5$, $\lambda_D = 10^3$, $\lambda_{\mathbf{n}} = 10^5$, $\lambda_{\mathbf{n}}^{\text{reg}} = 2 \times 10^6$. For *orange* we use, $\lambda_P = 10^5$, $\lambda_\rho = 6 \times 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^7$, $\lambda_D = 10^3$, $\lambda_{\mathbf{n}} = 10^5$, $\lambda_{\mathbf{n}}^{\text{reg}} = 2 \times 10^6$. For *owls* we use, $\lambda_P = 5 \times 10^4$, $\lambda_\rho = 4 \times 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^6$, $\lambda_D = 10^3$, $\lambda_{\mathbf{n}} = 10^5$, $\lambda_{\mathbf{n}}^{\text{reg}} = 10^5$. For *santa* we use, $\lambda_P = 10^5$, $\lambda_\rho = 6 \times 10^4$, $\lambda_{\boldsymbol{\rho}}^{\text{reg}} = 10^6$, $\lambda_D = 10^3$, $\lambda_{\mathbf{n}} = 10^5$, $\lambda_{\mathbf{n}}^{\text{reg}} = 2 \times 10^5$.

Diffuse albedo MSE			
Test scene	Sang [16]	Boss [17]	Ours
1	0.002	0.003	0.004
2	0.002	0.010	0.006
3	0.001	0.001	0.005
4	0.027	0.044	0.038
5	0.001	0.003	0.007
6	0.002	0.002	0.022
7	0.002	0.002	0.017
8	0.007	0.008	0.056
9	0.024	0.027	0.004
10	0.003	0.004	0.007
11	0.102	0.282	0.010
12	0.042	0.019	0.005
13	0.050	0.058	0.004
14	0.092	0.042	0.019
15	0.006	0.008	0.005
16	0.002	0.033	0.004
17	0.038	0.100	0.007
18	0.011	0.035	0.004
19	0.062	0.046	0.004
20	0.006	0.036	0.015
Avg.	0.024	0.038	0.017

TABLE 3: Diffuse albedo MSE for the 20 test dataset [17], where $\sigma = 0.001$ for our method (cf. Table 2 for varying σ.)

Specular reflectance MSE			
Test scene	Sang [16]	Boss [17]	Ours
1	0.004	0.002	0.002
2	0.006	0.021	0.003
3	0.004	0.002	0.002
4	0.044	1.455	0.025
5	0.005	0.010	0.004
6	0.014	0.054	0.012
7	0.009	0.038	0.008
8	0.059	0.164	0.037
9	0.004	0.026	0.001
10	0.006	0.005	0.005
11	0.027	0.087	0.010
12	0.008	0.015	0.002
13	0.005	0.020	0.001
14	0.024	0.374	0.010
15	0.004	1.097	0.004
16	0.004	0.022	0.002
17	0.007	0.030	0.002
18	0.006	0.049	0.001
19	0.002	0.011	0.001
20	0.016	0.034	0.009
Avg.	0.013	0.176	0.007

TABLE 4: Specular reflectance MSE for the 20 test dataset [17], where $\sigma = 0.001$ for our method (cf. Table 2 for varying σ.)

Normal angle difference			
Test scene	Sang [16]	Boss [17]	Ours
1	0.258	0.177	0.164
2	0.437	0.261	0.192
3	0.630	0.256	0.149
4	0.342	0.371	0.174
5	0.924	0.569	0.332
6	0.459	0.274	0.093
7	0.863	0.252	0.217
8	0.417	0.439	0.106
9	0.240	0.178	0.127
10	0.306	0.411	0.087
11	0.755	0.451	0.128
12	0.174	0.161	0.096
13	0.148	0.162	0.100
14	0.186	0.132	0.115
15	0.464	0.545	0.215
16	0.715	0.545	0.253
17	0.512	0.272	0.108
18	0.119	0.510	0.084
19	0.598	0.115	0.054
20	0.285	0.243	0.266
Avg.	0.4461	0.3529	0.232

TABLE 5: Normal angular difference for the 20 test dataset [17], where $\sigma = 0.001$ for our method (cf. Table 2.)

Depth affine MSE (10^{-4})			
Test scene	Sang [16]	Boss [17]	Ours
1	12.31	0.26	0.05
2	43.22	0.36	0.27
3	70.62	0.81	0.38
4	11.29	3.63	0.24
5	50.94	3.52	15.48
6	133.87	5.22	2.38
7	88.24	1.56	0.05
8	29.32	2.87	1.69
9	5.10	0.40	0.03
10	26.58	4.78	0.25
11	97.09	6.79	3.46
12	10.77	0.35	0.22
13	12.26	0.37	0.23
14	32.58	3.47	6.70
15	67.52	2.72	0.64
16	109.41	11.46	4.34
17	58.24	3.17	0.12
18	2.27	0.95	0.06
19	300.46	9.64	0.66
20	53.56	0.93	0.60
Avg.	60.82	3.16	1.89

TABLE 6: Depth MSE up to an affine transform for the 20 test dataset [17], where $\sigma = 0.001$ for our method (cf. Table 2).

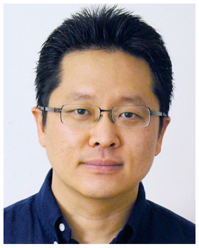
Hyun Jin Ku Hyun Jin Ku received her B.Sc. (2019) and M.Sc. (2021) degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST). Her research interests include light field, 3D retrievals, and computational photography.

Hyunho Ha is a PhD student in the Visual Computing Lab, KAIST and received his Master degree in Computer Science from KAIST in 2019. Prior to coming to the KAIST-VCLAB, he obtained his BS degree in Computer Science from KAIST. His research interests include various applications of computer graphics and vision, including real-time 4D scanning of dynamic objects.

Joo Ho Lee is an assistant professor of Sogang university and supervises the Visual Computing Laboratory. He worked as a postdoctoral researcher at the University of Tuebingen and Max Planck Institute before. He received his Ph.D. in computer science from KAIST in 2020. He served reviewer of conference programs such as SIGGRAPH and CVPR. His research interests include computer graphics, 3D reconstruction, and computer vision.

Dahyun Kang received her B.Sc. (2019) and M.Sc. (2021) degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST). Her research interests include light field, 3D retrievals, and computational photography.

James Tompkin is the John E. Savage Assistant Professor of Computer Science at Brown University. His research at the intersection of computer vision, computer graphics, and human-computer interaction helps develop new visual computing tools and experiences. He completed doctoral work at University College London on large-scale video processing and interaction techniques, and postdoctoral work at the Max-Planck Institute for Informatics and Harvard University on new methods to edit content within images and videos. Recent research has developed new machine learning techniques for low-level scene reconstruction, view synthesis for VR, and content generation.



Min H. Kim is a professor of computer science at the Korea Advanced Institute of Science and Technology (KAIST), leading the Visual Computing Laboratory. Prior to KAIST, he worked as a postdoctoral researcher at Yale University. He received his Ph.D. in computer science from University College London (UCL) in 2010. In addition to serving on many conference program committees, such as SIGGRAPH and CVPR, he has been working as an associate editor in various journals: ACM Transactions on Graphics and IEEE Transactions on Visualization and Computer Graphics. His research interests include computational imaging, computational photography, 3D imaging, and hyperspectral imaging, in addition to color and visual perception.