

# DeProCams: Simultaneous Relighting, Compensation and Shape Reconstruction for Projector-Camera Systems

Bingyao Huang, Haibin Ling

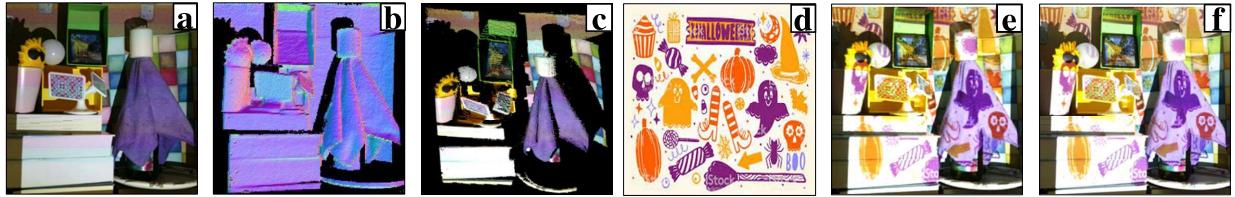


Fig. 1. Deep Projector-Camera systems (DeProCams): (a) scene under plain white illumination, (b) estimated normal map, (c) estimated colored point cloud viewed from a different pose, (d) a new projector input pattern, (e) inferred relit result, and (f) camera-captured ground truth, *i.e.*, (d) projected to (a).

**Abstract**— Image-based relighting, projector compensation and depth/normal reconstruction are three important tasks of projector-camera systems (ProCams) and spatial augmented reality (SAR). Although they share a similar pipeline of finding projector-camera image mappings, in tradition, they are addressed independently, sometimes with different prerequisites, devices and sampling images. In practice, this may be cumbersome for SAR applications to address them one-by-one. In this paper, we propose a novel end-to-end trainable model named DeProCams to explicitly learn the photometric and geometric mappings of ProCams, and once trained, DeProCams can be applied simultaneously to the three tasks. DeProCams explicitly decomposes the projector-camera image mappings into three subprocesses: shading attributes estimation, rough direct light estimation and photorealistic neural rendering. A particular challenge addressed by DeProCams is occlusion, for which we exploit epipolar constraint and propose a novel differentiable projector direct light mask. Thus, it can be learned end-to-end along with the other modules. Afterwards, to improve convergence, we apply photometric and geometric constraints such that the intermediate results are plausible. In our experiments, DeProCams shows clear advantages over previous arts with promising quality and meanwhile being fully differentiable. Moreover, by solving the three tasks in a unified model, DeProCams waives the need for additional optical devices, radiometric calibrations and structured light.

**Index Terms**—Projector-camera systems, spatial augmented reality, projector compensation, depth and normal reconstruction

## 1 INTRODUCTION

Projector-camera systems (ProCams) have many applications in projection mapping/spatial augmented reality (SAR) [20, 35, 49, 62, 71–74, 77], which usually involves three important tasks: image-based relighting [21, 22, 44, 55, 57, 63, 67], projector compensation [1, 2, 5, 7, 18, 19, 28, 29, 38, 49, 51, 60, 61, 76, 85], and structured light shape reconstruction [14, 46, 54, 78]. An example is shown in Fig. 1, where image-based relighting and depth/normal estimation are performed using the proposed DeProCams.

Image-based relighting aims to simulate the camera-captured image when a novel projector image is projected to the scene. It can synthesize realistic projection mapping effects without actual project-and-capture operations, which can be useful for SAR testing/debugging and virtual storing and sharing SAR setups. A typical solution for image-based relighting is light transport matrix (LTM) [9, 10, 53, 54, 67, 79, 90], which starts by projecting and capturing various sampling images and then estimate an LTM from the collected images, finally a camera-captured image can be inferred by applying the LTM to a novel projector input image. Although LTM produces accurate relighting results, it may be time-consuming [67], or requires additional optical devices such as lasers and a second camera [79] and beamsplitters [53]. Moreover, projector-camera radiometric calibration [11, 17] is required and thus

may be inconvenient for some projection mapping/SAR applications that frequently adjust the projector settings.

Projector compensation aims to modify a projector image such that when projected to the scene it can compensate for the disturbance from the scene geometry and photometry, and create desired viewer-perceived effects. This technique is often applied in SAR applications, *e.g.*, appearance editing [1, 71] and projection-based transparent effects [36, 86]. A typical pipeline [5, 19, 51] is to first find pixel geometric mappings using structured light, then estimates pixel-wise photometric transfer functions from the geometrically registered sampling images, finally a projector input image can be compensated by applying the inverse photometric transfer functions.

Shape reconstruction aims to estimate the scene depth and normal using a different set of sampling images other than the image-based relighting and the projector compensation ones. Knowing the scene shape can significantly improve the accuracy of projected augmentations, and aid depth-aware interactive SAR applications [4, 27]. A typical shape reconstruction method is structure light [14] that projects and captures hand-crafted patterns, then estimates depth (and normal) using the matched pattern pixels and geometric calibrations.

Because the three tasks have different prerequisites, optical devices and sampling images, some SAR applications solve them independently, although they share a similar pipeline of projecting and capturing sampling images and then finding projector-camera image mappings.

In this paper, we show that it is possible to solve the three tasks simultaneously without additional optical devices, radiometric calibrations or different sampling images (*e.g.*, structured light patterns). In particular, we explicitly model projector-camera image mappings as a geometry- and attributes-aware shading problem and solve it using an end-to-end trainable model named **DeProCams** (Deep Projector-Camera systems), which consists of two modules: *DepthToAttribute*

• Bingyao Huang and Haibin Ling are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794 USA.  
E-mail: {bihuang, hling}@cs.stonybrook.edu.

and *ShadingNet*.

From a learnable depth map, the **DepthToAttribute** module explicitly calculates the shading attributes such as projector direct light rays, surface normal, view and reflection directions, *etc*. To address the *occlusion* challenge in shading, a novel *differentiable* projector direct light mask is derived by exploring the related epipolar geometry, and can be efficiently computed using differentiable image warping [37], tensor sorting and image gradient. Then, initial rough direct light components are estimated to constrain photometry and geometry and to provide a good initialization for *ShadingNet*'s photorealistic direct and indirect light rendering.

Afterwards, the **ShadingNet** module is designed to aggregate intermediate shading attributes and rough direct light, and to render the camera-captured image with photorealistic direct and indirect light. Moreover, to improve model robustness and convergence, we incorporate multiple photometric and geometric constraints, such as projector direct light mask consistency, rough diffuse shading consistency and depth/normal/pixel mapping smoothness. Because of the explicit modeling of light-geometry-material interactions and sufficient constraints, our depth map is learned from the sampling images without its ground truth.

Finally, after a few minutes of data capturing and training, three SAR tasks, *i.e.*, *image-based relighting*, *projector compensation* and *depth/normal estimation* can be simultaneously performed by the proposed DeProCams.

Our contributions can be summarized as follows:

- The proposed DeProCams is, to our best knowledge, the first to simultaneously perform three SAR tasks with one learned model: *image-based relighting*, *projector compensation* and *depth/normal estimation*.
- The proposed DeProCams does not require additional optical devices, radiometric calibrations and structured light patterns.
- Multiple projector-camera photometric and geometric constraints are incorporated into DeProCams to improve model convergence and performance, and we propose a *differentiable* projector direct light mask to explicitly address the occlusion issue. These techniques are expected to facilitate future works in deep learning-based SAR application.
- We construct a simultaneous *image-based relighting* and *shape reconstruction* benchmark for projector-camera systems, which allows future works to compare with DeProCams using our training and testing dataset.

For the benefit of the society, the source code, training and evaluation benchmark dataset and experimental results are made publicly available at <https://github.com/BingyaoHuang/DeProCams>.

## 2 RELATED WORK

### 2.1 Image-based relighting

In tradition, *image-based relighting* methods rely on estimating a light transport matrix (LTM) [9, 10, 54, 63, 67, 79, 90] from projected and captured sampling image pairs, such that each camera pixel's irradiance is modeled as a linear combination of *all* projector pixels' riances.

Early methods [10, 82] directly estimate LTM using brute force sampling that requires plenty of HDR images, memory and time. To improve LTM estimation efficiency, later methods exploit the sparsity and coherence of LTM [9, 44, 52, 57, 63, 67, 79, 90], leading to reduced sampling images and computation. For example, [67] leverage adaptive sampling to reduce LTM reconstruction time to two hours. With additional RGB lasers and a second camera, Wang *et al.* [79] use kernel Nyström to approximate LTM. Ren *et al.* [63] model each LTM entry as a nonlinear function of pixels and light locations, then use multilayer neural networks to approximate it. Once reconstructed, LTM can be applied to a novel projector input image to simulate the relit scene.

Typical LTM-based relighting methods focus on accurate indirect light estimation and usually a camera cannot directly see projected

patterns [79]. While other SAR applications focus on direct light such as structured light [14], projection-based appearance editing [18, 62, 71] and augmentation [35, 36, 49], where camera directly sees projected patterns (Fig. 2), *i.e.*, *projector direct light dominates*. For those applications, estimating a full LTM is time consuming and it may be simplified. For example, some approaches [18, 19, 51] assume that each camera pixel is only coupled with a *single* projector pixel that contributes most of the irradiance. Then projector-camera pixel-wise mappings can be applied to efficiently solve *projector compensation* and *attributes from shading*. Following this kind of work in SAR applications, in this paper, we focus on a single projector-camera pair (Fig. 2) where projector direct light dominates, but unlike [18, 19, 51], we do not assume pixel-wise mapping and thus can estimate indirect light.

### 2.2 Projector compensation

*Projector compensation* estimates a projector input image given its corresponding desired camera-captured image and can be roughly categorized into full compensation, *i.e.*, both geometric and photometric compensation and partial ones.

Wetzstein *et al.* [83] use an inverse LTM to obtain accurate indicate light compensation, *e.g.*, glasses, and thus projector-camera radiometric calibration is required as a prerequisite of LMT. In practice, the camera radiometric calibration is simple but the projector calibration is nontrivial. Moreover, recalibration is needed when the projector settings change, such as brightness, contrast and color profiles. Thus it may be inconvenient for some SAR applications that frequently adjust projector settings. To waive radiometric calibrations, Grundhöfer and Iwai [19] encode projector-camera radiometric responses and photometric transfer function using a pixel-wise thin-plate-spline (TPS), such that the composite transfer function can be estimated using RGB sampling images. To further waive the structured light patterns, Huang and Ling [28] propose a full compensation method that jointly learns 2D geometry and photometry from the sampling images using a deep neural network. More detailed reviews can be found in [5, 20], and some recent representative works can be found in [19] and [28–30, 38].

### 2.3 Attributes from shading

*Attributes from shading* [3] aims to estimate scene attributes from camera-captured images, such as surface normal, reflectance, light directions and depth, *etc*. To avoid complex indirect light modeling, some simplifications are imposed, such as assuming distant light in photometric stereo (PS) or direct light dominance in structured light (SL), respectively. In the following we briefly review studies in PS and SL that relate most to our work. The two differences between PS and SL are (1) PS assumes light sources with fixed riances and varying directions placed at infinity while SL assumes a projector with varying output patterns placed at the fixed near field. (2) PS estimates depth and normal from light and surface interactions/shadings, while SL computes depth using 2-D pixel mappings and triangulation.

**Photometric stereo** (PS) [8, 66, 70, 75, 84] estimates surface normal by varying the light directions while fixing the view direction. In its earliest form [84] the surface is assumed Lambertian and the surface normal is obtained by solving linear equations. More complex BRDFs are considered later [15, 69]. Recently, Santo *et al.* [66] leverage deep neural networks to solve this problem using supervised training data. To avoid creating normal ground truth, Taniai *et al.* [75] propose a self-supervised photometric stereo network that uses image reconstruction loss. To deal with uncalibrated cases, Chen *et al.* [8] propose a self-calibrating method to jointly estimate unknown light directions and surface normal from non-Lambertian surfaces.

**Structured light** (SL) [14, 40, 46, 54, 78] estimates scene depth by matching projected and captured hand-crafted patterns, then matched 2-D pixel coordinates are triangulated [23] to recover the 3-D point cloud. However, this two-step strategy heavily depends on projector direct light and thus 2-D pixel matching may fail in strong indirect light regions [9, 28, 50, 54]. To address this issue, O'Toole *et al.* [54] combine LTM with SL to jointly estimate scene depth, albedo and indirect light.

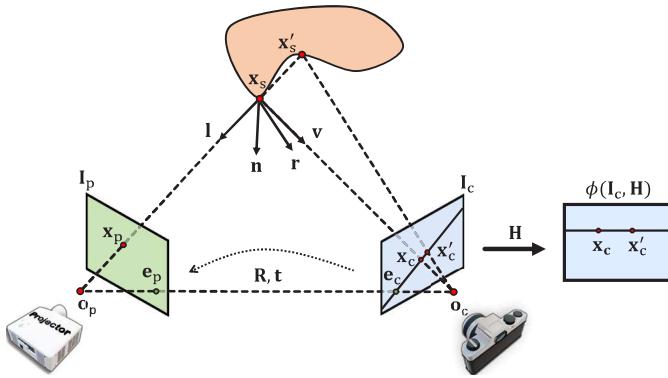


Fig. 2. **Coordinate system of our projector-camera setup.** With the geometric calibration, a rectification transformation  $\mathbf{H}$  can be calculated to transform the camera-captured image such that all epipolar lines are parallel.

Although PS and SL can recover the geometry of the scene, they alone are unable to solve image-based relighting or projector compensation.

#### 2.4 Relationship to DeProCams

Recently, deep learning-based solutions have been proposed for related works, such as deep shading/neural rendering [26, 41, 48], projector compensation [28, 29, 38] and attributes from shading [12, 42, 43, 59, 87]. However, these methods either require scene modeling or have simple assumptions on scene geometry, or need ground truth attributes supervision.

The question is, without scene modeling or ground truth attributes, can we learn a model from projector-camera sampling images, and apply it *simultaneously* to three tasks, *i.e.*, image-based relighting, projector compensation and depth/normal reconstruction? It is shown in CompenNet++ [28] to be possible for projector compensation, but restricted to smooth occlusion-free surfaces. Moreover, the depth and normal involved in the projector-camera system are ignored. Both limitations make it unable to perform image-based relighting in general scenes with occlusions or depth/normal reconstruction. Thus, directly applying CNNs to projector-camera systems is hard without modeling the light-geometry-material interactions. By contrast, our DeProCams explicitly models light and geometry (depth/normal) interactions, and leverages additional photometric and geometric constraints, meanwhile performing three tasks simultaneously.

### 3 DEEP PROJECTOR-CAMERA SYSTEMS

#### 3.1 Problem formulation

As shown in Fig. 2, we model the projector-camera image formation as a mapping from a projector input RGB image  $\mathbf{I}_p$  to a camera-captured RGB image  $\mathbf{I}_c$ :

$$\mathbf{I}_c = \pi(\mathbf{I}_p) \quad (1)$$

where  $\pi$  is the composite image transfer function that contains nonlinear projector and camera radiometric responses, image projection functions and projector light ray, scene geometry and material interactions, *etc.* Apparently Eq. 1 has no closed form solution and one intuition is to directly learn  $\pi$  from sampling image pairs like  $\mathbf{I}_c$  and  $\mathbf{I}_p$  using a general image-to-image translation CNN, *e.g.*, U-Net [65] or Pix2pix [34]. It may work if the training set is large or data augmentation is available. However, SAR applications require online data capturing and thus it is impractical to capture a large sampling set. Moreover, data augmentation such as rotation and random cropping cannot be applied due to strong couplings between projector-camera pixels. As a result, only a small number of sampling images are available for model training. To address this issue, we leverage domain knowledge of projector-camera image formation, such as light-geometry-material interactions and photometric and geometric constraints.

Note that our method is inspired by CompenNet++ [28], which disentangles 2D pixel geometric mappings and photometry from projector-camera systems. Compared with [28], our DeProCams explicitly separates shading attributes and depth from the 2D pixel geometric mappings; and we explicitly compute rough shadings before applying photometric neural rendering. More specifically, we decompose  $\pi$  into a geometry-aware module **DepthToAttribute**  $\mathcal{T}$  to compute rough shading attributes and a photometry module **ShadingNet**  $\mathcal{F}$  for photorealistic neural rendering. In particular, DepthToAttribute contains a learnable depth map (in the canonical camera frontal view)  $\mathbf{d}_c \in \mathbb{R}^{h \times w \times 1}$  and ShadingNet  $\mathcal{F}$  is a neural network that consists of learnable convolutional filters. The learnable parameters of DeProCams are  $\{\mathbf{d}_c, \mathcal{F}\}_\theta$ . Note that  $\mathbf{d}_c$  and  $\mathcal{F}$  are jointly optimized during DeProCams training, thus the term “learnable”.

#### 3.2 DepthToAttribute $\mathcal{T}$

##### 3.2.1 Shading attributes

As shown in Fig. 2, we assume that the world origin is at the camera optical center  $\mathbf{o}_c$ , and denote the camera and the projector intrinsics as  $\mathbf{K}_c$  and  $\mathbf{K}_p$ , and denote their relative rotation and translation as  $\mathbf{R}$  and  $\mathbf{t}$ , respectively. For a surface point  $\mathbf{x}_s \in \mathbb{R}^3$  in the scene, let  $\mathbf{x}_c, \mathbf{x}_p \in \mathbb{R}^2$  be its pixel coordinates in the camera image  $\mathbf{I}_c$  and the projector image  $\mathbf{I}_p$ , respectively. Then the depth of  $\mathbf{x}_s$  is given by  $\mathbf{d}_c = \mathbf{d}_c(\mathbf{x}_c)$ . The geometric relationships between  $\mathbf{x}_s, \mathbf{x}_c$  and  $\mathbf{x}_p$  are given by:

$$\mathbf{x}_s = \mathbf{K}_c^{-1} \bar{\mathbf{x}}_c \mathbf{d}_c, \quad \bar{\mathbf{x}}_p = \mathbf{K}_p [\mathbf{R} | \mathbf{t}] \bar{\mathbf{x}}_s \quad (2)$$

where  $\bar{\mathbf{x}}$  is the homogeneous coordinate of  $\mathbf{x}$ . Given the 3-D coordinates  $\mathbf{x}_s$  in Eq. 2, shading attributes such as the projector direct light ray direction  $\mathbf{l}$ , the reflection ray direction  $\mathbf{r}$  and the camera view direction  $\mathbf{v}$  are computed by:

$$\begin{aligned} \mathbf{n} &= \partial_x \mathbf{x}_s \times \partial_y \mathbf{x}_s & \mathbf{l} &= \mathbf{o}_p - \mathbf{x}_s \\ \mathbf{r} &= 2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l} & \mathbf{v} &= \mathbf{o}_c - \mathbf{x}_s \end{aligned} \quad (3)$$

where  $\times$  is vector cross product. Note that in above equation  $\mathbf{n}, \mathbf{l}, \mathbf{r}, \mathbf{v} \in \mathbb{R}^3$  are shadings attributes of a single surface point  $\mathbf{x}_s$ . To extend Eq. 3 to all surface points, we abuse the notation and refer them as  $\mathbb{R}^{h \times w \times 3}$  maps in the following text. Meanwhile, given the projector and camera pixel geometric mappings in Eq. 2, we construct an image sampling/warping grid  $\Omega \in \mathbb{R}^{h \times w \times 2}$  such that  $\mathbf{x}_p = \Omega(\mathbf{x}_c)$ . Therefore, the projector input image warped to the canonical camera frontal view is given by:

$$\mathbf{I}'_p = \phi(\mathbf{I}_p, \Omega) \odot \mathbf{M} \quad (4)$$

where  $\phi$  is a differentiable image interpolator [37];  $\odot$  is element-wise multiplication, and  $\mathbf{M} \in \mathbb{R}^{h \times w \times 1}$  is a mask and more details are given in Sect. 3.2.3.

##### 3.2.2 Rough shadings

Afterwards, we can compute light-geometry-material interactions using a rough direct light shading model (Eq. 5) under the direct light dominance assumption of SAR applications. *It is worth noting that our rough shading model is inspired but different from the Phong model [6, 58, 68] (thus the name “rough”).* Rather than using the actual diffuse and specular reflectance maps and shininess of Phong model [6, 58, 68], we use a rough approximation  $\mathbf{s}$  instead and set shininess  $\alpha = 1$ , because (1) estimating actual reflectance maps are difficult and unnecessary for our tasks; and (2) even with accurate reflectance maps, the empirical model (Eq. 5) alone cannot produce photorealistic rendering. Therefore, *the rough shadings only provide a simple yet plausible initial point for ShadingNet (Eq. 9)*, meanwhile, leverage photometric and geometric constraints (Sect. 4.1).

$$\mathbf{I}_c^{\text{abnt}} = \mathbf{s}, \quad \mathbf{I}_c^{\text{diff}} = \mathbf{I}'_p \odot \mathbf{s} \odot (\mathbf{n} \cdot \mathbf{l}), \quad \mathbf{I}_c^{\text{spec}} = \mathbf{I}'_p \odot \dot{\mathbf{s}} \odot (\mathbf{r} \cdot \mathbf{v})^\alpha \quad (5)$$

where  $\mathbf{s}$  models rough ambient and surface reflectance, and is approximated by a scene image captured under plain white illumination (Fig. 4);  $\odot$  and  $\cdot$  are element-wise multiplication and dot product, respectively; and  $\dot{\mathbf{s}}$  is the grayscale image of  $\mathbf{s}$ .

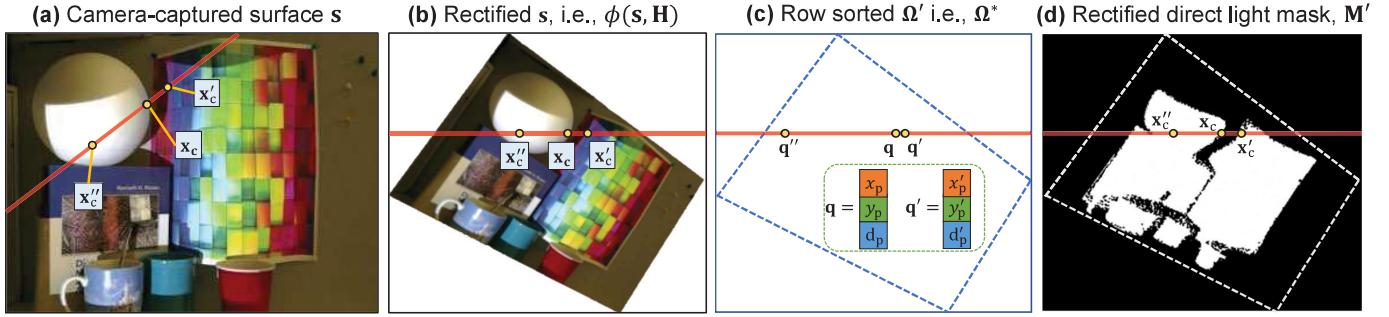


Fig. 3. **Proposed differentiable projector direct light mask.** (a) Three points are on the same red epipolar line and finding occlusion requires 2-D searching. (b) Instead, we perform 1-D search in stereo rectified image. (c) To speed up pair-wise comparison, we row sort  $\Omega'$  in lexicographic order. After sorting,  $\mathbf{x}_c$ 's and  $\mathbf{x}'_c$ 's corresponding points  $\mathbf{q}, \mathbf{q}' \in \Omega^*$  become neighbors, then we apply Eq. 7 and find that  $\mathbf{x}'_c$  is occluded, since  $|\mathbf{x}'_p - \mathbf{x}_p| + |\mathbf{y}'_p - \mathbf{y}_p| < 1$  and  $|\mathbf{d}'_p - \mathbf{d}_p| > 0$ . All operations are differentiable and parallelizable, thus are efficient for training on GPU.

### 3.2.3 Differentiable projector direct light mask

Occlusion is a particular challenge to depth and normal reconstruction because it may make two camera pixels map to the same projector pixel (Fig. 2 and Fig. 3), causing false projector direct light rays, while the depth-based image warping  $\phi$  (Eq. 4 and Fig. 4) is unable to address it. Some previous methods [13, 16, 32, 45, 80] leverage forward-backward consistency to find occlusions, assuming that only slight motion and lighting changes across views. However, for a projector-camera system such an assumption may be violated. Therefore, we exploit epipolar geometry and propose a *differentiable* projector direct light mask  $\mathbf{M}$  to address occlusions below.

As shown in Fig. 2 and Fig. 3, two surface points  $\mathbf{x}_s$  and  $\mathbf{x}'_s$  lie on the same projector ray, therefore their projections  $\mathbf{x}_c$  and  $\mathbf{x}'_c$  must lie on the same epipolar line. From Fig. 2 and Fig. 3, we see that  $\mathbf{x}'_s$  is occluded by  $\mathbf{x}_s$ , and thus only  $\mathbf{x}_c$  is illuminated by the projector direct light. This type of occlusion can be modeled by a projector direct light mask  $\mathbf{M}$ , such that for any camera pixel  $\mathbf{x}_c$  on the epipolar line  $\overrightarrow{\mathbf{e}_c \mathbf{x}_c}$  ( $\mathbf{e}_c$  is the camera epipole), its pixel value in  $\mathbf{M}$  is given by:

$$\mathbf{M}(\mathbf{x}_c) = \begin{cases} 1, & \forall \mathbf{x}'_c \in \overrightarrow{\mathbf{e}_c \mathbf{x}_c}, \mathbf{x}_p \not\approx \mathbf{x}'_p \text{ OR } \mathbf{d}_p < \mathbf{d}'_p \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where  $\mathbf{x}_p = \Omega(\mathbf{x}_c)$  and  $\mathbf{x}'_p = \Omega(\mathbf{x}'_c)$  are the corresponding projector pixel coordinates (Fig. 2);  $\mathbf{d}_p = ([\mathbf{R}|\mathbf{t}] \bar{\mathbf{x}}_s)_z$  and  $\mathbf{d}'_p = ([\mathbf{R}|\mathbf{t}] \bar{\mathbf{x}}'_s)_z$  are the depths of  $\mathbf{x}_c$  and  $\mathbf{x}'_c$  in the projector view space, respectively. Then, we apply a stereo rectification [23]  $\mathbf{H}$  to reduce the 2-D search to 1-D (Fig. 3(b)). However, even with 1-D search, the pair-wise comparison in Eq. 6 for each pixel on each epipolar line is still expensive. The logical OR operator in Eq. 6 implies that depth comparison is only needed when  $\mathbf{x}_c$  and  $\mathbf{x}'_c$  map to the same projector pixel, *i.e.*,  $\mathbf{x}_p \approx \mathbf{x}'_p$  (Fig. 2). Therefore, for each *rectified* epipolar line, if we sort each pixel's corresponding projector pixel coordinates  $\mathbf{x}_p$  in lexicographical order, the pair-wise comparisons in Eq. 6 can be performed only in a small neighborhood, *e.g.*,  $\pm 1$  pixel.

To make the above operations *differentiable*, we concatenate the sampling/warping grid  $\Omega$  and the projector-view depth map  $\mathbf{d}_p$  along the channel and rectify them using the differentiable image interpolator [37] by  $\Omega' = \phi([\Omega, \mathbf{d}_p], \mathbf{H})$ ,  $\Omega' \in \mathbb{R}^{h \times w \times 3}$ . Next, we sort each 3  $\times$  1 element in each row of  $\Omega'$  (*i.e.*, along epipolar line) and denote the sorted tensor  $\Omega'$  as  $\Omega^*$  (Fig. 3(c)). Afterwards, we compute the forward difference in the row direction, then the rectified direct light mask  $\mathbf{M}'$  is given by:

$$\mathbf{M}' = \begin{cases} 1, & |\partial_x \Omega_x^*| + |\partial_x \Omega_y^*| > 1 \text{ or } \partial_x \Omega_z^* < 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Note that the non-differentiable unit step function above is approximated by a differentiable clamp and ReLU [47]:  $\min(\sigma * \text{ReLU}(\cdot), 1)$ ,  $\sigma \gg 1$ . Then, we unrectify mask  $\mathbf{M}'$  (Fig. 3(d)) back to the camera view by  $\mathbf{M} = \phi(\mathbf{M}', \mathbf{H}^{-1})$ . Finally, the projector direct light mask is applied to mask out false direct light pixels as shown in Fig. 4, and to enforce direct light mask consistency in Sect. 4.1.1.

We implement the above geometric transformations using differentiable operations (Kornia [64]), and then gather them in **DepthToAttribute**  $\mathcal{T}$  (Fig. 4). It is worth noting that, unlike the WarpingNet in [28] that only estimates a sampling/warping grid  $\Omega$ , DepthToAttribute learns depth map and computes shading attributes and rough shadings:

$$\{\mathbf{n}, \mathbf{l}, \mathbf{r}, \mathbf{v}, \Omega, \mathbf{M}, \mathbf{I}'_p, \mathbf{I}_c^{\text{diff}}, \mathbf{I}_c^{\text{spec}}\} = \mathcal{T}(\mathbf{K}_c, \mathbf{K}_p, \mathbf{R}, \mathbf{t}) \quad (8)$$

Note that we omit  $\mathbf{d}_c$  in above equation because it is inherently included in  $\mathcal{T}$ . These attributes are then fed to ShadingNet to further infer photorealistic shadings and used as additional constraints (Sect. 4.1).

### 3.3 ShadingNet

To infer photorealistic direct and indirect light, we design an image-to-image translation CNN named **ShadingNet**  $\mathcal{F}$  to aggregate intermediate results and refine rough direct light shadings by:

$$\hat{\mathbf{I}}_c = \mathcal{F}(\mathbf{I}'_p, \mathbf{I}_c^{\text{abnt}}, \mathbf{I}_c^{\text{diff}}, \mathbf{I}_c^{\text{spec}}) \quad (9)$$

ShadingNet architecture is shown in Fig. 4 and Table 1, and it takes four inputs: (1) rough ambient, *i.e.*, the camera-captured surface image  $\mathbf{I}_c^{\text{abnt}} = \mathbf{s}$ ; (2) rough diffuse  $\mathbf{I}_c^{\text{diff}}$ ; (3) rough specular  $\mathbf{I}_c^{\text{spec}}$ ; and (4) projector direct light image warped to the canonical camera frontal view  $\mathbf{I}'_p$ . ShadingNet consists of three subnets: (1) a surface skip branch sk1 with three convolution layers (blue blocks in the top) to pass rough ambient  $\mathbf{s}$  to the output layer; (2) an encoder-like rough shading branch (green blocks) that uses four convolutional layers (Conv1r-Conv4r) to downsample and extract rich features of the rough shadings; and (3) an encoder-decoder backbone network (red blocks) to capture rich direct and indirect light interactions involved in the projector-camera image formation. The backbone network consists of an encoder (Conv1-Conv4) and a decoder (Conv5, TConv1, TConv2 and Conv6).

Firstly, in the rough shading branch, the input rough shadings ( $\mathbf{s}, \mathbf{I}_c^{\text{diff}}, \mathbf{I}_c^{\text{spec}}$ ) are concatenated along the channel. Then, the 9-channel input is fed to the encoder (Conv1r-Conv4r) and sequentially downsampled to 1/4 of the original sizes, while rich features are extracted by increasing the feature map channels:  $9 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ . Similarly, the encoder part of the backbone (Conv1-Conv4) has the same structure as the rough shading branch, except that the input is a 3-channel warped projector direct light image  $\mathbf{I}'_p$ . The four vertical black skip connections between the rough shading branch and the backbone indicate that rough shading branch feature maps are added to the backbone ones. Moreover, the two black horizontal skip connections under the backbone network add low-level feature maps (Conv1 and Conv2) to the *first half* of the high-level feature maps (Conv5 and TConv1).

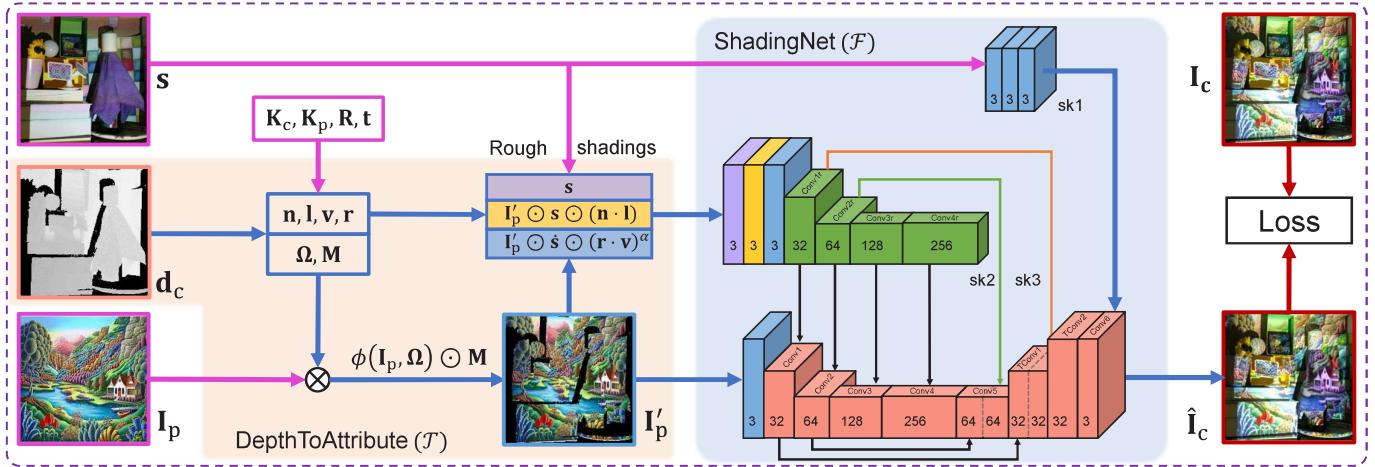


Fig. 4. Deep projector-camera systems (DeProCams) architecture and training pipeline. The inputs (magenta) are  $I_p$ ,  $s$ ,  $K_c$ ,  $K_p$ ,  $R$  and  $t$ , and output is  $\hat{I}_c$ ;  $\phi$  is an image warper. In the forward/inference stage, using the depth map  $d_c$  and calibration data  $K_c$ ,  $K_p$ ,  $R$  and  $t$ , DepthToAttribute  $\mathcal{T}$  computes the shading attributes, i.e.,  $n$ ,  $l$ ,  $r$ ,  $v$ ,  $\Omega$ ,  $M$  and warps projector direct light image to the camera view as  $I'_p$ . Then, we compute three rough shadings. Finally, ShadingNet  $\mathcal{F}$  gathers rough direct light shadings and  $I'_p$  to infer photorealistic camera-capture image  $\hat{I}_c$ . During backpropagation, the loss (between  $\hat{I}_c$  and  $I_c$ ) gradient is backpropagated to update  $d_c$  and ShadingNet  $\mathcal{F}$  parameters jointly.

Table 1. **ShadingNet  $\mathcal{F}$  parameters.** **Conv** and **TConv** are convolutional and transposed convolutional layers of a backbone subnet, respectively. Note that Conv1r is the first convolutional layer of rough shading branch, and Conv2r-Conv4r shares the same structure as backbone Conv2-Conv4. **Sk** stands for skip convolutional layers and Sk1 has three identical layers (weights are different), thus we only show one layer here. ReLU layers after each Conv/TConv/Sk layer are omitted. **Ker.**, **Str.**, **Pad.** are kernel size, stride and padding size, respectively.

Layer	Input size	Output size	Ker.	Str.	Pad.
Conv1r	[h, w, 9]	[h/2, w/2, 32]	3×3	2	1
Conv1	[h, w, 3]	[h/2, w/2, 32]	3×3	2	1
Conv2	[h/2, w/2, 32]	[h/4, w/4, 64]	3×3	2	1
Conv3	[h/4, w/4, 64]	[h/4, w/4, 128]	3×3	1	1
Conv4	[h/4, w/4, 128]	[h/4, w/4, 256]	3×3	1	1
Conv5	[h/4, w/4, 256]	[h/4, w/4, 128]	3×3	1	1
TConv1	[h/4, w/4, 128]	[h/2, w/2, 64]	2×2	2	0
TConv2	[h/2, w/2, 64]	[h, w, 32]	2×2	2	0
Conv6	[h, w, 32]	[h, w, 3]	3×3	1	1
Sk1	[h, w, 32]	[h, w, 32]	3×3	1	1
Sk2	[h/2, w/2, 32]	[h/2, w/2, 32]	1×1	1	0
Sk3	[h/4, w/4, 64]	[h/4, w/4, 64]	1×1	1	0

Meanwhile, the two low-level feature maps of the rough shading branch (Conv1r and Conv2r) are transformed by two skip convolutional layers sk2 and sk3, and then added to the *second half* of Conv5 and TConv1. The two types of skip connections allow ShadingNet to carry low-level visual details to deeper layers, and to better learn complex interactions between the direct light and the scene geometry in the feature space. Afterwards, the backbone decoder gradually upsamples the fused feature maps back to the original image sizes, meanwhile, decreases the feature map channels by  $256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 3$ . Finally, the backbone network and sk1 output images are added through the blue skip connection. Detailed network ShadingNet parameters are shown in Table 1, where **Input size** and **Output size** are sizes of input and output feature map/image in the [height, width, channel] format. Note that for each layer, the number of convolutional filters is equal to the number of output channel.

It is worth noting that the input and design of **ShadingNet  $\mathcal{F}$**  are particularly important to the model convergence: (1) all the inputs

are geometrically aligned, obviating the need for  $\mathcal{F}$  to learn geometry, and (2) ShadingNet output image  $\hat{I}_c$  shares some structural details with the rough ambient image ( $s$  in Fig. 4), which can be leveraged as a good initial point. Thus, we use a shallow surface skip branch to pass  $s$  to the backbone output, allowing  $\mathcal{F}$  to focus on learning the residual [25] light on top of  $s$ , instead of directly inferring the final result  $\hat{I}_c$  from random guess. The two techniques and the photometric and geometric constraints below together reduce the chance of falling into local minima early on and improve convergence.

We find that a more complex ShadingNet (e.g., a deeper network with more learnable parameters) and longer training time may increase accuracy of relighting, compensation and shape reconstruction. However, considering the practicability of DeProCams in SAR applications, in this paper we choose this architecture and 1,000 iterations to balance training/inference efficiency and accuracy.

#### 4 IMPLEMENTATION AND TRAINING DETAILS

We implement DeProCams  $\pi$  using PyTorch [56] and Kornia [64]. The learnable parameters of DeProCams are  $\pi_\theta = \{d_c, \mathcal{F}\}_\theta$ . In particular, the depth map  $d_c$  contains  $h \times w \times 1$  learnable parameters, where  $h$  and  $w$  are camera image height and width, respectively;  $\mathcal{F}$  contains learnable convolutional filters/kernels (Table 1) and with a carefully chosen loss function, constraints and an optimizer (e.g., stochastic gradient descent and Adam [39]), both  $\mathcal{F}$  and  $d_c$  can be learned jointly.

##### 4.1 Loss function and constraints

As shown in Fig. 4, given a new projector input image  $I_p$ , a well trained DeProCams  $\pi$  should be able to reconstruct/infer a relit image  $\hat{I}_c = \pi(I_p)$ , which is very close to the real camera-captured one  $I_c$ . Thus, during network training, we encourage the similarity between DeProCams inferred and the ground truth images by minimizing the pixel-wise  $L_1 + \text{SSIM}$  [81] loss:

$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_i^N (|\hat{I}_c^{(i)} - I_c^{(i)}| + 1 - \text{SSIM}(\hat{I}_c^{(i)}, I_c^{(i)})) \quad (10)$$

where  $N$  is the number of training image pairs ( $I_p, I_c$ ). The advantages of this image reconstruction loss are shown in [28–30, 88].

Note that directly optimizing  $\mathcal{F}$  and  $d_c$  using  $\mathcal{L}_{\text{recon}}$  may be difficult and as a result, the final depth map  $d_c$  is coarse and DeProCams inferred relit image  $\hat{I}_c$  is blurry, because  $\mathcal{L}_{\text{recon}}$  alone does not impose sufficient

constraints. To address this issue, besides modeling rough direct light shadings and projector direct light mask in DepthToAttribute, we introduce three additional photometric and geometric constraints to ensure accurate inference of the shading attributes (Eq. 8) and prevent  $\mathcal{F}$  from overfitting. Their effectiveness is shown in ablation study Sect. 5.6.

#### 4.1.1 Direct light mask consistency constraint

In Sect. 3.2.3, DeProCams explicitly infers a projector direct light mask  $\mathbf{M}$  (Eq. 6) to address occlusion. It also imposes an additional constraint that  $\mathbf{M}$  should match the projector field of view (FOV)  $\mathbf{s}^* \in \mathbb{R}^{h \times w \times 1}$ . This geometric constraint can reduce depth searching space by early rejecting geometrically impossible depths and is implemented as a pixel-wise  $L_2$  loss:

$$\mathcal{L}_{\text{mask}} = \|\mathbf{M} - \mathbf{s}^*\|_2^2 \quad (11)$$

where  $\mathbf{s}^*$  is a thresholded camera-captured surface for projector FOV.

#### 4.1.2 Rough shading consistency constraint

Although our rough direct light shading (Sect. 3) is different from an actual Phong model, it is useful to encourage photometric similarities between rough diffuse shading  $\mathbf{I}_c^{\text{diff}}$  (Eq. 5) and camera-captured ground truth  $\mathbf{I}_c$ . We hence use the following pixel-wise  $L_2$  loss:

$$\mathcal{L}_{\text{rough}} = \frac{1}{2N} \sum_i^N \left\| \mathbf{I}_c^{\text{diff}(i)} - \mathbf{I}_c^{(i)} \right\|_2^2 \quad (12)$$

This loss forces DeProCams to infer plausible light rays and surface normals. Since the two intermediate attributes are derived from the depth map  $\mathbf{d}_c$ , this constraint also improves depth map accuracy.

#### 4.1.3 Smoothness constraint

In most SAR setups, the depth map  $\mathbf{d}_c$  (Fig. 4) and intermediate attributes such as the sampling/warping grid  $\Omega$  (Eq. 4) and the normal map  $\mathbf{n}$  (Eq. 3) are locally smooth. This constraint can be implemented as minimizing the absolute value of image gradient:

$$\mathcal{L}_{\text{smooth}}(\mathbf{I}) = |\partial_x \mathbf{I}| e^{-||\partial_x \mathbf{s}||} + |\partial_y \mathbf{I}| e^{-||\partial_y \mathbf{s}||} \quad (13)$$

where  $\mathbf{I}$  is an image (e.g.,  $\mathbf{d}_c$ ,  $\Omega$  and  $\mathbf{n}$ ) whose smoothness is constrained by its partial derivatives in  $x$  and  $y$  directions. To preserve sharp edges, we use the gradient of camera-captured scene  $\mathbf{s}$  as smoothness loss weights [13, 16, 89]. Combining smoothness losses of  $\mathbf{d}_c$ ,  $\Omega$  and  $\mathbf{n}$ , we have:

$$\mathcal{L}_{\text{smooth}}^{\text{all}} = 2\mathcal{L}_{\text{smooth}}(\mathbf{d}_c) + \mathcal{L}_{\text{smooth}}(\Omega) + 0.01\mathcal{L}_{\text{smooth}}(\mathbf{n}) \quad (14)$$

The smoothness constraint forces  $\mathbf{d}_c$ ,  $\Omega$  and  $\mathbf{n}$  to be geometrically plausible and prevents  $\mathcal{F}$  from overfitting on coarse depth map and intermediate attributes in early training stages.

## 4.2 Network training

We combine image reconstruction loss  $\mathcal{L}_{\text{recon}}$  with the three constraints above and jointly minimize them using Adam optimizer [39]:

$$\{\mathbf{d}_c, \mathcal{F}\}_{\theta} = \underset{\{\mathbf{d}_c', \mathcal{F}'\}_{\theta}}{\text{argmin}} \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{rough}} + \mathcal{L}_{\text{smooth}}^{\text{all}} \quad (15)$$

The performance of the gradient-based optimization above depends on  $\{\mathbf{d}_c, \mathcal{F}\}_{\theta}$  initializations and a reasonable initial point can greatly improve convergence. Thus, to initialize  $\mathbf{d}_c$ , we start by assuming the initial sampling/warping  $\Omega$  as affine, and then we estimate the affine transformation using the projector input image size and the bounding rectangle of camera-captured projector FOV mask  $\mathbf{s}^*$  (similar to WarpingNet [28] affine parameters initialization). Afterwards, we compute the depths of  $\Omega$  using triangulation [23] and set the initial  $\mathbf{d}_c$  to it. ShadingNet  $\mathcal{F}$  filter weights are initialized using He's method [24].

The initial learning rate of the depth map  $\mathbf{d}_c$  and the ShadingNet  $\mathcal{F}$  are set to  $10^{-2}$  and  $10^{-3}$ , respectively. Then, using training image pairs like  $\{(\mathbf{I}_p^{(i)}, \mathbf{I}_c^{(i)})\}_{i=1}^N$  ( $N$  is #Train in Table 2), we train the model for 1,000 iterations on an Nvidia GeForce 1080Ti GPU with a batch size of 24, and it takes about 5 minutes to finish.

## 4.3 One stone three birds

Once trained, DeProCams can be used for three SAR tasks: image-based relighting, projector compensation and depth/normal estimation.

### 4.3.1 Image-based relighting

Image-based relighting infers the camera-captured image  $\tilde{\mathbf{I}}_c$  given a novel projector input lighting image  $\tilde{\mathbf{I}}_p$  without actual projection or capture. Since DeProCams  $\pi$  has already learned the forward projector-camera image mappings (Eq. 1), relighting is performed by a single forward pass of DeProCams by  $\tilde{\mathbf{I}}_c = \pi(\tilde{\mathbf{I}}_p)$ .

### 4.3.2 Projector compensation

Given a new projector input image  $\mathbf{I}_p$ , projector compensation aims to infer a compensation image  $\mathbf{I}_p^*$ , such that when projected to the textured surface, the camera-captured image is ideally the same as  $\mathbf{I}_p$  (*i.e.*, without geometric or photometric disturbances from the surface). In practice, the optimal displayable area of the projection surface is restricted to a subregion of the projector FOV, and following [28] we first compute an optimal desired camera-captured image  $\mathbf{I}_c'$ , which is a scaled version of  $\mathbf{I}_p$ . Then, projection compensation is to find  $\mathbf{I}_p^*$  such that  $\mathbf{I}_c' = \pi(\mathbf{I}_p^*)$  and it can be solved using the following objective function:

$$\begin{aligned} \mathbf{I}_p^* = \underset{\mathbf{I}_p}{\text{argmin}} \left( \underbrace{|\pi(\mathbf{I}_p) - \mathbf{I}_c'| + 1 - \text{SSIM}(\pi(\mathbf{I}_p), \mathbf{I}_c')}_\text{image reconstruction loss} \right. \\ \left. + \underbrace{\mathcal{L}_{\text{smooth}}(\mathbf{I}_p)}_\text{smoothness loss} + \underbrace{10 (\|\max(\mathbf{I}_p - 1, 0)\|^2 + \|\min(\mathbf{I}_p, 0)\|^2)}_\text{color saturation loss [19]} \right) \end{aligned} \quad (16)$$

### 4.3.3 Depth and normal

The depth  $\mathbf{d}_c$  is inherently learned by DeProCams. To be robust against setup scale variances, we normalize both  $\mathbf{d}_c$  and the projector-camera translation vector  $\mathbf{t}$  by dividing  $\|\mathbf{t}\|$ , the actual scene depth is given by denormalization:  $\hat{\mathbf{d}}_c = \mathbf{d}_c * \|\mathbf{t}\|$ . Moreover, following the convention of learning-based depth estimation [89], we estimate the inverse depth  $1/\hat{\mathbf{d}}_c$  instead to bound the learnable parameter values to  $[0, 1]$ . The normal is then given by Eq. 3.

## 5 EXPERIMENTAL EVALUATIONS

### 5.1 System configuration

Our setup consists of a single Canon 6D camera and a single ViewSonic PJD7828HDL DLP projector, and they are geometrically calibrated using [31]. Their resolutions are  $320 \times 240$  and  $800 \times 600$ , respectively. The distance between the projector-camera system and the target scene is around 1,000mm to 2,000mm, depending on different setups. To improve data capturing efficiency, we set the camera to live view mode and connect an Elgato Cam Link 4K video capture card to the camera HDMI output. Considering the the video capture card delay and the projector-camera synchronization, we manually set a 200ms delay between the projection and the capture operation, and in total it takes about 360ms to capture a frame.

### 5.2 Evaluation benchmark

As mentioned in Sect. 4.3 that DeProCams can perform three tasks, *i.e.*, image-based relighting, projector compensation and depth/normal estimation. In this section, we quantitatively and qualitatively evaluate and compare DeProCams with other methods on the three tasks.

Following [28], we captured 23 different setups with various lightings, poses, focal lenses and projector settings. Using the sampling patterns in [28], for each setup, we captured a surface image  $\mathbf{s}$  and  $N$  (#Train) training image pairs  $\{(\mathbf{I}_p^{(i)}, \mathbf{I}_c^{(i)})\}_{i=1}^N$  (took about 3 min) for training and another 200 image pairs for testing. Note that depth and normal are learned by the network without their ground truth images.

We then captured projector compensation setups in complex scenes with glasses and slight occlusions (see Fig. 6) to show that the proposed DeProCams outperforms the other methods. However, these scenarios may cause missing pixels and thus it is unfair to use the surrogate

Table 2. **Quantitative comparison of relighting and depth reconstruction.** Results are averaged over 23 different setups.  $d_{\text{err}}$  is the mean  $L_2$  distance between the inferred and the ground truth point cloud (mm). For SL [46] without manual cleaning and interpolation,  $d_{\text{err}} = 1.0858$ . Note that SL  $d_{\text{err}}$  is much smaller because the ground truth is a manually cleaned and interpolated version of SL point cloud.

# Train	Model	SL Masked			Whole Image			$d_{\text{err}}(\text{mm}) \downarrow$
		PSNR $\uparrow$	RMSE $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$	RMSE $\downarrow$	SSIM $\uparrow$	
50	TPS + SL [19, 46]	27.6703	0.0731	0.9407	17.8285	0.2345	0.4336	-
	CompenNet++ [28]	24.2998	0.1093	0.8991	23.3141	0.1220	0.8437	-
	DeProCams (ours)	<b>27.9819</b>	<b>0.0719</b>	<b>0.9408</b>	<b>26.1807</b>	<b>0.0883</b>	<b>0.8919</b>	<b>8.5653</b>
	No const. (ours)	27.5575	0.0751	0.9365	25.8954	0.0910	0.8874	9.7733
	No mask (ours)	27.5614	0.0759	0.9346	25.8146	0.0927	0.8851	13.1531
	No rough (ours)	27.0641	0.0795	0.9313	25.4962	0.0951	0.8798	10.4029
100	TPS + SL [19, 46]	27.6165	0.0736	0.9439	17.8261	0.2346	0.4365	-
	CompenNet++ [28]	26.7337	0.0817	0.9329	25.3195	0.0961	0.8849	-
	DeProCams (ours)	<b>29.6620</b>	<b>0.0586</b>	<b>0.9569</b>	<b>27.3590</b>	<b>0.0768</b>	<b>0.9100</b>	<b>4.9520</b>
	No const. (ours)	29.1676	0.0621	0.9535	27.0458	0.0797	0.9064	6.0782
	No mask (ours)	29.5997	0.0592	0.9564	27.2395	0.0782	0.9100	7.2205
	No rough (ours)	28.8089	0.0649	0.9498	26.7679	0.0822	0.9005	5.7721
250	TPS + SL [19, 46]	27.4265	0.0753	0.9444	17.8056	0.2352	0.4369	-
	CompenNet++ [28]	26.7593	0.0810	0.9322	25.3926	0.0947	0.8860	-
	DeProCams (ours)	30.5640	0.0522	0.9655	28.0801	0.0702	0.9224	<b>2.9408</b>
	No const. (ours)	30.3678	0.0537	0.9642	27.9490	0.0714	0.9209	3.4319
	No mask (ours)	<b>30.7901</b>	<b>0.0509</b>	<b>0.9676</b>	<b>28.1284</b>	<b>0.0701</b>	<b>0.9253</b>	4.1483
	No rough (ours)	30.0289	0.0557	0.9603	27.7054	0.0733	0.9144	3.2261
500	TPS + SL [19, 46]	27.2926	0.0766	0.9436	17.7908	0.2356	0.4362	-
	CompenNet++ [28]	27.4843	0.0743	0.9386	26.0587	0.0879	0.8948	-
	DeProCams (ours)	30.8368	0.0506	0.9674	28.3392	0.0682	0.9251	<b>2.5087</b>
	No const. (ours)	30.7392	0.0512	0.9666	28.2660	0.0688	0.9242	2.8683
	No mask (ours)	<b>31.1380</b>	<b>0.0488</b>	<b>0.9699</b>	<b>28.4636</b>	<b>0.0675</b>	<b>0.9286</b>	3.5542
	No rough (ours)	30.3418	0.0535	0.9626	27.9812	0.0710	0.9176	2.7140

protocol in [28]. Instead, we compared on actual camera-captured compensations. Afterwards, we captured the ground truth point cloud using Gray-coded SL [46], followed by manual cleaning and interpolation. Future works on projector-camera image-based relighting and depth/normal reconstruction can be evaluated and compared with DeProCams without actual projections.

### 5.3 Image-based relighting

Existing LTM-based relighting methods usually rely on radiometric calibrations and additional optical devices, such as lasers and a second camera [79] and beam splitters [53], thus we cannot compare with them using our radiometrically uncalibrated single projector-camera setup. Instead, we modified the recently proposed learning-based projector compensation method CompenNet++ [28] by swapping its input and output, such that forward projector-camera image mappings can be learned. We train it with the same batch size of 24 and 1,000 iterations as DeProCams, and it also takes about 5 minutes.

Another method that works for radiometrically uncalibrated single projector-camera setup is Grundhöfer and Iwai [19] that applies pixel-wise thin-plate-spline (TPS) to projector compensation, we combine a Gray-coded SL [46] geometric correction with it and name the method **TPS+SL**. Since SL can only recover direct light pixels as shown in Fig. 5, for fair comparison, we compare with TPS+SL only in SL masked regions and name it **SL Masked** in Table 2. DeProCams has a similar performance as TPS+SL when using 50 training images. However, TPS+SL uses 42 extra SL images to find pixel mappings. Moreover, TPS+SL cannot estimate indirect light and thus has worse performance in Table 2 **Whole Image** columns.

The qualitative comparisons are shown in Fig. 5. The first two columns are projector input light patterns and camera-captured scenes, respectively. The 3<sup>rd</sup> to 5<sup>th</sup> columns are relit results of different methods. The last column is camera-captured ground truth, *i.e.*, the 1<sup>st</sup> column projected to the 2<sup>nd</sup>. The setups consist of challenging direct and

indirect light effects, *e.g.*, in the 1<sup>st</sup> row, we present diffuse (plywood board, paper boxes and books), translucency (light bulb), subsurface scattering (wax, towel), specular highlight (mug) and shadow; and in the 2<sup>nd</sup> row, we show diffuse (plywood board, paper boxes, foam ball), subsurface scattering (wax), mirror (metal ball), specular highlight (metal can), translucency (light bulb) and shadow.

For TPS+SL [19, 46], indirect light regions such as shadow, specular highlight and mirror are black because SL [46] rely on direct light and is unable to find correct pixel mappings in those regions (see zoomed-in patches). Moreover, it fails to estimate image blur, thus the images look much sharper than the ground truth. CompenNet++ [28] can estimate some indirect light but it generates much blurry relit results, *e.g.*, the purple skull in the 1<sup>st</sup> row red zoomed-in patch and the yellow wheel in the blue zoomed-in patch. By contrast, our DeProCams can estimate more accurate indirect light, *e.g.*, in the 2<sup>nd</sup> row red zoomed-in patch, the bottom-right corner of the metal ball shows reflected orange color from the white foam ball. See supplementary material for more results.

The results in Fig. 5 and Table 2 show that the proposed DeProCams significantly outperforms CompenNet++ [28] because (1) DeProCams explicitly models light-geometry-material interactions using rough direct light (Eq. 5) and it imposes additional photometric and geometric constraints (Sect. 4.1). (2) Without depth, CompenNet++'s 2-D geometric correction is unable to address occlusions, while DeProCams predicts a differentiable projector direct light mask to explicitly deal with occlusions (Sect. 3.2.3). (3) DeProCams's ShadingNet  $\mathcal{F}$  refines rough shadings and has a novel structure that allows modeling more complex light-attribute interactions (Fig. 4).

### 5.4 Projector compensation

We compare DeProCams with TPS+SL [19, 46] and CompenNet++ [28] using the same sampling image pairs as image-based relighting, except that both TPS+SL's and CompenNet++'s input and output are swapped to learn the backward mapping  $\mathbf{I}_c \mapsto \mathbf{I}_p$ . Then, we train CompenNet++

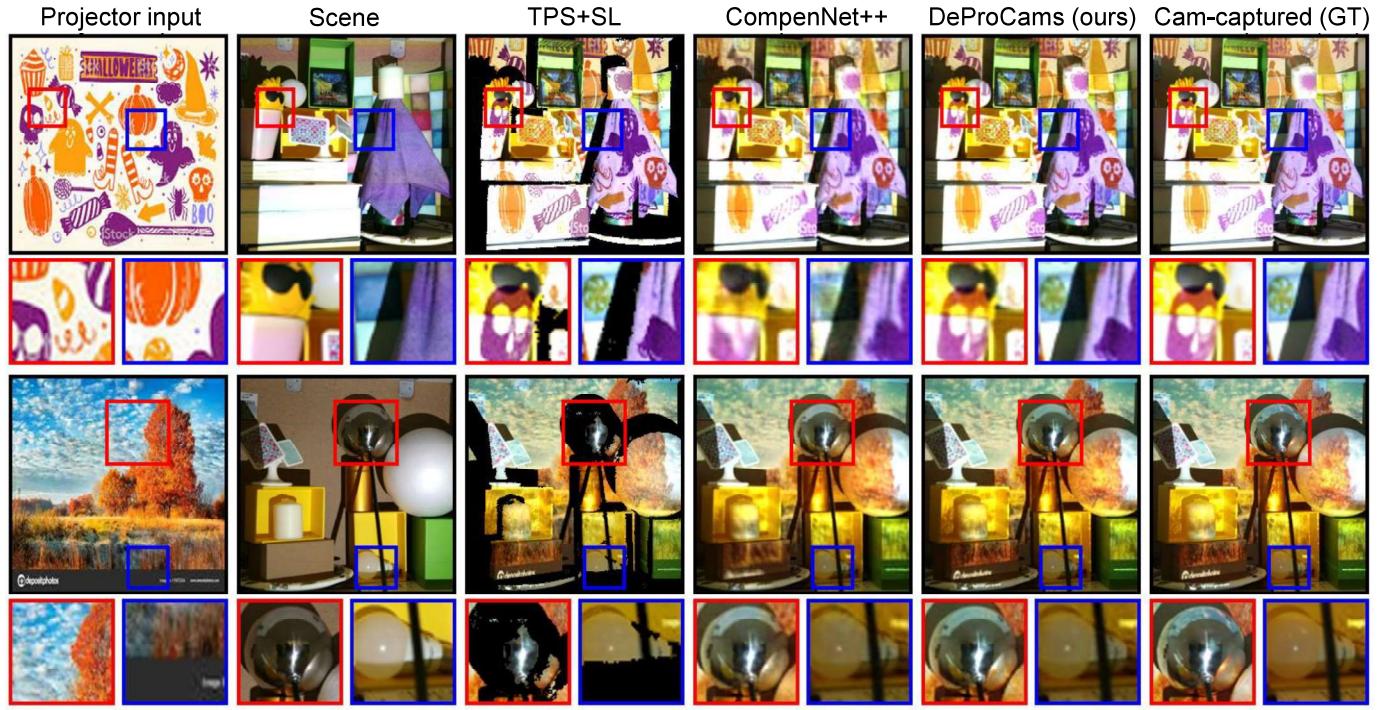


Fig. 5. **Image-based relighting.** We provide two scenes (the first two rows and the last two rows) and each scene is under different projector lightings. Each image is provided with two zoomed-in patches for detailed comparison. See supplementary material for more results.

Table 3. **Quantitative comparison of projector compensation.** Results are averaged over 10 different setups.

Model	PSNR $\uparrow$	RMSE $\downarrow$	SSIM $\uparrow$
TPS + SL [19, 46]	21.3601	0.1541	0.8695
CompenNet++ [28]	22.2046	0.1421	0.8838
DeProCams (ours)	<b>23.8245</b>	<b>0.1132</b>	<b>0.8885</b>
Uncompensated	9.3374	0.5979	0.1089

for 1,500 iterations to match the original settings in [28].

The quantitative results are shown in Table 3. Note that DeProCams outperforms both TPS+SL and CompenNet++, and has much smaller pixel-wise color errors (PSNR/RMSE). Note that we capture another 10 different setups for compensation with slight or no occlusions.

The qualitative comparisons are shown in Fig. 6, where a challenging wine glass is placed in front of the textured surface, and some projector pixels are occluded by the wine glass stem and the bottom of the textured surface. Note that DeProCams generates better compensation in the sky region, while the other two methods cannot well compensate the surface texture, and thus we can see clear green cube-like artifacts. Moreover, CompenNet++ [28]’s result looks blurry and contains clear sawtooth patterns in the top, because it assumes a smooth projection surface and WarpingNet may fail to learn accurate pixel mappings around occluded regions. On the other hand, TPS+SL can handle occlusions, but in the top-left region some SL decoding errors deteriorate the compensation effect. See supplementary material for more results.

In summary, DeProCams outperforms TPS+SL [19, 46] and CompenNet++ [28] on this projector compensation benchmark because (1) geometry- and attributes-aware formulation (Sect. 3) and the novel architecture (Fig. 4) let DeProCams better model complex projector-camera image mappings. (2) Occlusion is explicitly addressed by a differentiable projector direct light mask. (3) We explicitly deal with saturation errors due to projector’s physical limitation and encourages compensation image smoothness (Eq. 16). (4) The fully differentiable

architecture allows the combined photometric and geometric constraints to optimize the compensation image through backpropagation.

### 5.5 Depth and normal estimation

To evaluate shape reconstruction, we capture the ground truth point cloud using Gray-coded SL [46], followed by manual cleaning and interpolation. Then, we convert DeProCams learned depth map  $\hat{d}_c$  to a point cloud using Eq. 2 and calculate depth error  $d_{err}$  as the mean Euclidean distance between the estimated and ground truth point cloud. We also compare with the SL’s raw point cloud before it is manually cleaned and interpolated. Note this comparison gives SL [46] great advantages, thus its depth error  $d_{err} = 1.0858$  mm is much smaller than those of DeProCams.

The quantitative and qualitative results are shown in Fig. 7 and Table 2 and the proposed DeProCams shows good depth and normal reconstruction capability. Moreover, unlike SL that assumes no geometric or photometric constraints between the surfaces points, DeProCams uses local weighted smoothness and rough shading consistency to generate high fidelity shapes. Thus, as shown in Fig. 8, our point cloud and normal map have lighter aliasing artifacts than SL.

### 5.6 Ablation study

To show the effectiveness of the differentiable projector direct light mask (Sect. 3.2.3) and the photometric and geometric constraints (Sect. 4.1), we performed comprehensive ablation studies. We created degraded DeProCams with the respective constraint disabled, *i.e.*, **No rough**, **No mask** and **No const.** are w/o rough diffuse and specular shadings (Eq. 5), w/o direct light mask, and w/o rough shading consistency (Sect. 4.1), respectively.

By comparing DeProCams and its degraded versions in Table 2, we find that DeProCams obtained the best relighting results when  $\#Train \leq 100$  and the best depth reconstruction results ( $d_{err}$ ) for all  $\#Train$ . To visually interpret  $d_{err}$ , we show the reconstructed point cloud and normal in Fig. 8. For each section, we compare different number of training images, *i.e.*,  $\#Train = 50$  and  $250$ . We repeat the ground truth of manually cleaned and interpolated SL depth and nor-

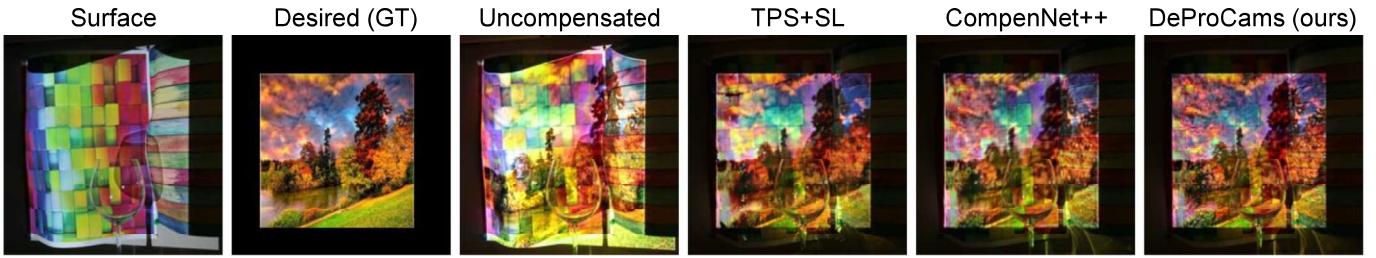


Fig. 6. **Projector compensation.** The 1<sup>st</sup> to 3<sup>rd</sup> columns are the projection surface, optimal desired viewer perceived image and uncompensated projection, respectively. The rest columns are real camera-captured results of different compensation methods. See supplementary for more results.

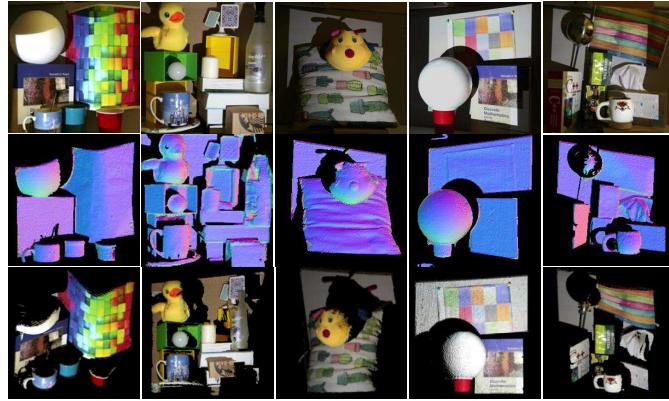


Fig. 7. **Depth and normal reconstruction.** We show five different setups. The 1<sup>st</sup> row is the scene under plain illumination. The 2<sup>nd</sup> row is estimated normal. The 3<sup>rd</sup> row is reconstructed point cloud viewed from a different pose. See supplementary material for more results.

mal (named **SL (GT)**) for each row for convenient visual comparison. **When #Train is large**, the quantitative and qualitative results of DeProCams and its degraded versions look similar, indicating that the hidden light and geometry interactions and constraints may also be learned without explicit modeling, as long as sufficient training data is provided. However, unlike general image-to-image translation tasks that data can be collected offline, SAR applications require online data capturing and thus are restricted to smaller training datasets. In this case, the proposed light and geometry interaction and constraints become particularly important. **When #Train is small**, DeProCams point cloud and normal are smoother than its degraded versions, showing that explicitly modeling rough light and geometry interactions (even different from an empirical Phong model) and adding photometric and geometric constraints can improve model convergence when training samples are limited. See supplementary material for more setups.

## 6 DISCUSSION

**Applicability to other settings.** The proposed DeProCams can be applied to higher resolution projectors and cameras without modifying the network structure and filter sizes, as long as the machine has sufficient GPU memory and computing power. Theoretically, a higher resolution projector-camera pair can provide finer training image details, which may improve DeProCams performance quality. Note that for new resolutions sizes not divisible by 4, we can change the padding size in Table 1 to compensate the input/output feature map sizes.

For miniaturized projector-camera systems with limited computing power, we can use model compression techniques [33] and half-precision floating-point to create a lighter version of DeProCams to improve efficiency with some sacrifice in quality. One concern about miniaturized projector-camera systems is limited dynamic ranges,

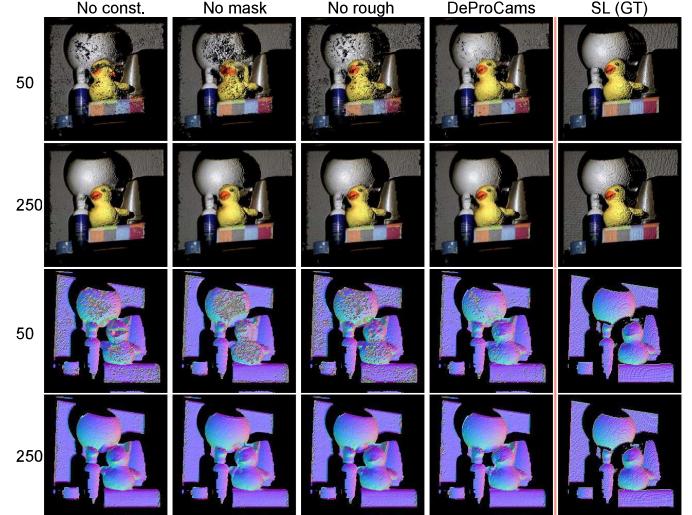


Fig. 8. **Ablation study of DeProCams constraints.** The first two rows are reconstructed point cloud viewed from a new pose and the last two rows are reconstructed normal. See supplementary for more results.

which may cause dropped DeProCams performance.

**Limitations and future work.** DeProCams is designed for SAR applications and has a different projector-camera setup from LTM, and its image-based relighting in indirect light regions still have room for improvement, as shown in Fig. 5, the 1<sup>st</sup> row wax and light bulb and Fig. 7 and the 2<sup>nd</sup> column frosted glass bottle. Incorporating domain knowledge of LTM and adding an additional step of radiometric calibration may improve the results and it is definitely an interesting direction to explore in future work.

## 7 CONCLUSIONS

In this paper, we propose the first end-to-end trainable solution, DeProCams, to solve the three important SAR tasks: image-based relighting, projector compensation and depth/normal reconstruction simultaneously. In addition, by solving the three tasks in a unified model, DeProCams waives the need for potential additional optical devices, radiometric calibrations and structured light patterns. To improve convergence and performance, especially when sampling images are limited, we leverage rich photometric and geometric constraints such as projector direct light mask, rough direct light shading and smoothness constraint. In our thorough experiments, DeProCams shows clear advantages over previous approaches with promising quality. Moreover, by constructing the first simultaneous image-based relighting and depth/normal estimation benchmark for projector-camera systems, our work is expected to facilitate future SAR studies in this direction.

**Acknowledgments.** This work was supported in part by US National Science Foundation (No. 2006665 and 1814745).

## REFERENCES

[1] D. G. Aliaga, Y. H. Yeung, A. Law, B. Sajadi, and A. Majumder. Fast high-resolution appearance editing using superimposed projections. *ACM Tran. on Graphics*, 2012. 1

[2] M. Ashdown, T. Okabe, I. Sato, and Y. Sato. Robust content-dependent photometric projector compensation. In *CVPRW PROCAMS*, 2006. 1

[3] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *T-PAMI*, 37(8):1670–1687, 2014. 2

[4] H. Benko. Beyond flat surface computing: challenges of depth-aware and curved interfaces. In *ACM Multimedia*, pp. 935–944, 2009. 1

[5] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. In *Computer Graphics Forum*, p. 84, 2008. 1, 2

[6] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH*, number 2, pp. 192–198. ACM, 1977. 3

[7] A. Boroomand, H. Sekkati, M. Lamm, D. A. Clausi, and A. Wong. Saliency-guided projection geometric correction using a projector-camera system. In *ICIP*, 2016. 1

[8] G. Chen, K. Han, B. Shi, Y. Matsushita, and K.-Y. K. Wong. Self-calibrating deep photometric stereo networks. In *CVPR*, pp. 8739–8747, 2019. 2

[9] N. Chiba and K. Hashimoto. Ultra-fast multi-scale shape estimation of light transport matrix for complex light reflection objects. In *ICRA*, pp. 1–5. IEEE, 2018. 1, 2

[10] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, pp. 145–156. ACM Press/Addison-Wesley Publishing Co., 2000. 1, 2

[11] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, p. 31. ACM, 2008. 1

[12] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Tran. on Graphics*, 37(4):128, 2018. 3

[13] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pp. 740–756. Springer, 2016. 4, 6

[14] J. Geng. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, 2011. 1, 2

[15] A. S. Georghiades. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In *ICCV*, vol. 3, p. 816, 2003. 2

[16] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pp. 270–279, 2017. 4, 6

[17] M. D. Grossberg and S. K. Nayar. Modeling the space of camera response functions. *T-PAMI*, 26(10):1272–1282, 2004. 1

[18] M. D. Grossberg, H. Peri, S. K. Nayar, and P. N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. In *CVPR*, 2004. 1, 2

[19] A. Grundhöfer and D. Iwai. Robust, error-tolerant photometric projector compensation. *IEEE TIP*, 2015. 1, 2, 6, 7, 8

[20] A. Grundhöfer and D. Iwai. Recent advances in projection mapping algorithms, hardware and applications. In *Computer Graphics Forum*. Wiley Online Library, 2018. 1, 2

[21] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escalano, R. Pandey, J. Dourgarian, et al. The relightables: volumetric performance capture of humans with realistic relighting. *ACM Tran. on Graphics*, 38(6):1–19, 2019. 1

[22] S. Han, I. Sato, T. Okabe, and Y. Sato. Fast spectral reflectance recovery using dlp projector. *IJCV*, 110(2):172–184, 2014. 1

[23] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 4, 6

[24] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[26] P. Hermosilla, S. Maisch, T. Ritschel, and T. Ropinski. Deep-learning the latent space of light transport. In *Computer Graphics Forum*, vol. 38, pp. 207–217. Wiley Online Library, 2019. 3

[27] J. Hochreiter, S. Daher, A. Nagendran, L. Gonzalez, and G. Welch. Touch sensing on non-parametric rear-projection surfaces: A physical-virtual head for hands-on healthcare training. In *IEEE VR*, pp. 69–74. IEEE, 2015. 1

[28] B. Huang and H. Ling. Compennet++: End-to-end full projector compensation. In *IJCV*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[29] B. Huang and H. Ling. End-to-end projector photometric compensation. In *CVPR*, 2019. 1, 2, 3, 5

[30] B. Huang, T. Sun, and H. Ling. End-to-end full projector compensation. *arXiv preprint arXiv:2008.00965*, 2020. 2, 5

[31] B. Huang, Y. Tang, S. Ozdemir, and H. Ling. A fast and flexible projector-camera calibration system. *IEEE Transactions on Automation Science and Engineering*, 2020. doi: 10.1109/TASE.2020.2994223. 6

[32] J. Hur and S. Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *ICCV*, pp. 312–321, 2017. 4

[33] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *ICLR*, 2017. 9

[34] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 3

[35] D. Iwai, M. Aoki, and K. Sato. Non-contact thermo-visual augmentation by ir-rgb projection. *IEEE TVCG*, 25(4):1707–1716, 2018. 1, 2

[36] D. Iwai and K. Sato. Document search support by making physical documents transparent in projection-based mixed reality. *Virtual reality*, 15(2-3):147–160, 2011. 1, 2

[37] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 2, 3, 4

[38] Y. Kageyama, M. Isogawa, D. Iwai, and K. Sato. Prodebnnet: projector deblurring using a convolutional neural network. *Optics Express*, 28(14):20391–20403, 2020. 1, 2, 3

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 6

[40] C. Li, Y. Monno, H. Hidaka, and M. Okutomi. Pro-cam ssfm: Projector-camera system for structure and spectral reflectance from motion. In *ICCV*, pp. 2414–2423, 2019. 2

[41] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. In *SIGGRAPH Asia*, p. 222. ACM, 2018. 3

[42] X. Li, Y. Dong, P. Peers, and X. Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Tran. on Graphics*, 36(4):45, 2017. 3

[43] 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*, p. 269. ACM, 2018. 3

[44] V. Masselus, P. Peers, P. Dutré, and Y. D. Willems. Relighting with 4d incident light fields. *ACM Tran. on Graphics*, 22(3):613–620, 2003. 1, 2

[45] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 4

[46] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. In *3DIMPVT*, 2012. 1, 2, 7, 8

[47] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 4

[48] O. Nalbach, E. Arabadzhyska, D. Mehta, H.-P. Seidel, and T. Ritschel. Deep shading: convolutional neural networks for screen space shading. In *Computer Graphics Forum*, vol. 36, pp. 65–78. Wiley Online Library, 2017. 3

[49] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE TVCG*, 2017. 1, 2

[50] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. In *ACM Tran. on Graphics*, vol. 25, pp. 935–944. ACM, 2006. 2

[51] S. K. Nayar, H. Peri, M. D. Grossberg, and P. N. Belhumeur. A projection system with radiometric compensation for screen imperfections. In *ICCVW PROCAMS*, vol. 3, 2003. 1, 2

[52] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM Tran. on Graphics*, vol. 22, pp. 376–381. ACM, 2003. 2

[53] M. O’Toole and K. N. Kutulakos. Optical computing for fast light transport analysis. *ACM Tran. on Graphics*, 29(6):164, 2010. 1, 7

[54] M. O’Toole, J. Mather, and K. N. Kutulakos. 3d shape and indirect appearance by structured light transport. In *T-PAMI*, pp. 1298–1312, 2016. 1, 2

[55] S. Oya and T. Okabe. Image-based relighting with 5-d incident light fields.

In *ICCV*, pp. 3031–3038, 2017. 1

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[57] P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. E. Debevec. Compressive light transport sensing. *ACM Trans. on Graphics*, 28(1):3, 2009. 1, 2

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

[59] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, pp. 283–291, 2018. 3

[60] R. Raskar and P. Beardsley. A self-correcting projector. In *CVPR*, 2001. 1

[61] R. Raskar, J. Van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: geometrically aware and self-configuring projectors. *ACM Trans. on Graphics*, 2003. 1

[62] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Rendering Techniques*. Springer, 2001. 1, 2

[63] P. Ren, Y. Dong, S. Lin, X. Tong, and B. Guo. Image based relighting using neural networks. *ACM Trans. on Graphics*, 34(4):111, 2015. 1, 2

[64] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, 2019. 4, 5

[65] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 2015. 3

[66] H. Santo, M. Samejima, Y. Sugano, B. Shi, and Y. Matsushita. Deep photometric stereo network. In *ICCV*, pp. 501–509, 2017. 2

[67] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. Lensch. Dual photography. In *ACM Trans. on Graphics*, vol. 24, pp. 745–755. ACM, 2005. 1, 2

[68] S. A. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985. 3

[69] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi. Bi-polynomial modeling of low-frequency reflectances. *T-PAMI*, 36(6):1078–1091, 2013. 2

[70] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *T-PAMI*, pp. 271–284, 2019. 2

[71] C. Siegl, M. Colaiani, L. Thies, J. Thies, M. Zollhöfer, S. Izadi, M. Stamminger, and F. Bauer. Real-time pixel luminance optimization for dynamic multi-projection mapping. *ACM Trans. on Graphics*, 2015. 1, 2

[72] T. Sueishi, H. Oku, and M. Ishikawa. Robust high-speed tracking against illumination changes for dynamic projection mapping. In *IEEE VR*, pp. 97–104. IEEE, 2015. 1

[73] H. Takatori, M. Hiraiwa, H. Yano, and H. Iwata. Large-scale projection-based immersive display: The design and implementation of largespace. In *IEEE VR*, pp. 557–565. IEEE, 2019. 1

[74] T. Takezawa, D. Iwai, K. Sato, T. Hara, Y. Takeda, and K. Murase. Material surface reproduction and perceptual deformation with projection mapping for car interior design. In *IEEE VR*, pp. 251–258. IEEE, 2019. 1

[75] T. Taniai and T. Maehara. Neural inverse rendering for general reflectance photometric stereo. In *ICML*, pp. 4864–4873, 2018. 2

[76] J.-P. Tardif, S. Roy, and M. Trudeau. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. In *3DIM*, 2003. 1

[77] T. Ueda, D. Iwai, T. Hiraki, and K. Sato. Illuminated focus: Vision augmentation using spatial defocusing via focal sweep eyeglasses and high-speed projector. *IEEE TVCG*, 26(5):2051–2061, 2020. 1

[78] M. Vo, S. G. Narasimhan, and Y. Sheikh. Texture illumination separation for single-shot structured light reconstruction. *T-PAMI*, 38(2):390–404, 2015. 1, 2

[79] J. Wang, Y. Dong, X. Tong, Z. Lin, and B. Guo. Kernel nyström method for light transport. *ACM Trans. on Graphics*, 28(3):29, 2009. 1, 2, 7

[80] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pp. 4884–4893, 2018. 4

[81] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 5

[82] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. In *ACM Trans. on Graphics*, vol. 24, pp. 756–764. ACM, 2005. 2

[83] G. Wetzstein and O. Bimber. Radiometric compensation through inverse light transport. In *15th Pacific Conference on Computer Graphics and Applications*, 2007. 2

[84] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980. 2

[85] T. Yoshida, C. Horii, and K. Sato. A virtual color reconstruction system for real heritage with light projection. In *Proceedings of International Conference on Virtual Systems and Multimedia*, vol. 3, 2003. 1

[86] T. Yoshida, K. Jo, K. Minamizawa, H. Nii, N. Kawakami, and S. Tachi. Transparent cockpit: Visual assistance system for vehicle using retro-reflective projection technology. In *IEEE VR*, pp. 185–188. IEEE, 2008. 1

[87] Y. Yu and W. A. Smith. Inverserendernet: Learning single image inverse rendering. In *CVPR*, pp. 3155–3164, 2019. 3

[88] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE TCI*, 2017. 5

[89] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pp. 1851–1858, 2017. 6

[90] D. E. Zongker, D. M. Werner, B. Curless, and D. Salesin. Environment matting and compositing. In *SIGGRAPH*, vol. 99, pp. 205–214, 1999. 1, 2