IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Lighting Model-Guided Initialization for Gradient Descent-Based Projector Compensation

WOOCHAN SUN, GEONU NOH, KWANGHEE KO, AND BOCHANG MOON, (Member, IEEE)
Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Bochang Moon (bmoon@gist.ac.kr).

This work was supported in part by the National Research Foundation of Korea (NRF) and Institute of Information & Communications Technology Planning & Evaluation (IITP) through Grants RS-2023-00207939 and 2021-0-00315, funded by the Ministry of Science and ICT (MSIT).

ABSTRACT Projection mapping, which maps a projector input (i.e., an image) onto a physical surface, is widely used to display user-specified visual content. However, the projected content can become distorted when an ideal surface, such as a flat and white plane, is unavailable. As a result, adjusting the projector input using a projector compensation technique becomes necessary to ensure that the projected image matches the user-provided target image. While various techniques have been proposed for projector compensation, a recent approach models projection mapping as a simulatable process in virtual space, where a light transport algorithm simulates the projection mapping. In such rendering frameworks, projector compensation is typically achieved by iteratively adjusting the projector input via a gradient descent-based optimizer, starting from an initial guess (often chosen arbitrarily). In this paper, we investigate how to set the initial values more effectively rather than choosing them arbitrarily. As the main contribution of the paper, we propose a new initialization scheme that determines the starting values based on a lighting model. We then integrate this model-guided initialization into gradient descent-based optimization and demonstrate that it improves projection mapping results, particularly for non-planar and colored surfaces.

INDEX TERMS Projector compensation, lighting model-guided initialization, physically based rendering for projector compensation

I. INTRODUCTION

Projection mapping has been employed in various applications, such as augmented reality [1]–[3], entertainment [4], [5], and digital signage [6], [7], where visual content is displayed on physical surfaces. A technical challenge in projection mapping arises when these surfaces are non-planar or colored. Such complex surfaces can introduce visual distortions to the projected content. Projector compensation techniques aim to eliminate these distortions by adjusting the projector input so that the projected output (i.e., camera-captured images) visually matches user-provided target images.

One approach to projector compensation is to address the two main sources of distortion separately. For example, non-planar geometry can be handled through geometric compensation, which estimates pixel correspondences between projector input images [2], [8]–[10] and camera-captured images. In contrast, non-constant surface color can be addressed through photometric compensation, which adjusts pixel values in the projector input based on the variations in surface colors [10]–[12]. This separate handling is effective when the projection surface is either non-planar with nearly constant colors (e.g., white) or planar with some color variations.

However, when projection surfaces are both non-planar and have spatially varying colors, employing a unified framework that performs both geometric and photometric compensation can be more desirable. For instance, learning-based methods using neural networks have been proposed to infer adjusted projector inputs that account for both types of distortion [13]–[15]. An alternative to these approaches is employing a differentiable rendering-based framework that models the camera-projector system and simulates projection mapping using a light transport algorithm [16]. This method typically employs on-the-fly learning, iteratively updating the projector input starting from an initial guess while minimizing the error between the simulated projector output and a user-provided target image using a gradient descent-based optimizer such as Adam [17].

A straightforward way to initialize this optimization is to use arbitrary values (e.g., a constant input as in [16]) as the initial guess. However, in this paper, we demonstrate that

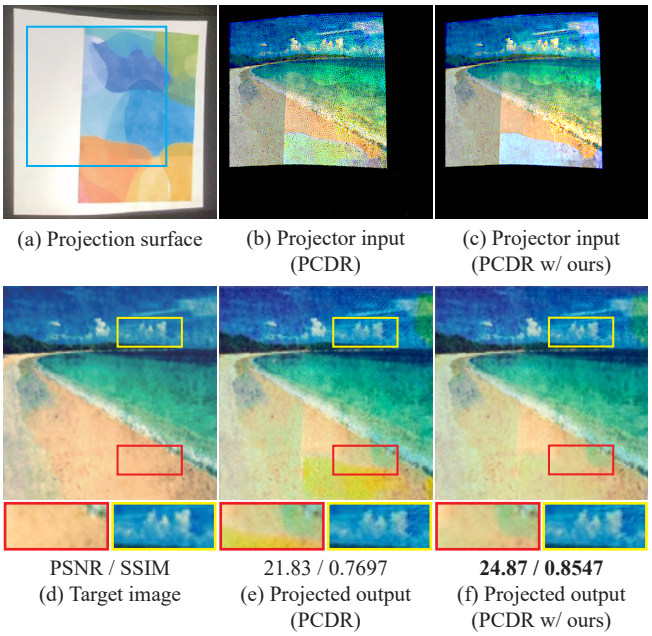


FIGURE 1. Projection mapping results of the gradient descent-based projector compensation framework, PCDR [16], on a non-planar curved surface with textured colors (a). PCDR adjusts the projector input images ((b) and (c)) so that the projected outputs ((e) and (f)) within the target display area (denoted by a cyan box in (a)) match the user-provided target image (d). We compute an initial guess of the projector input using a lighting model and pass this to PCDR, allowing it to begin its iterative optimization from our output rather than its default initialization (i.e., a constant input). This modification enables PCDR to produce visually and numerically improved results without altering its underlying algorithm.

initializing with carefully selected values can significantly improve the final outcome. Our contributions are summarized as follows:

- We propose a new initialization scheme that enables a differentiable rendering-based framework to begin optimization from properly selected initial parameters.
- We determine the initial values by considering the functional relationship between the projector input and its projected output (i.e., the camera-captured image) using a well-established lighting model.

We integrate this lighting model-guided initialization into an existing projector compensation framework as a plug-in module and demonstrate that it guides the optimization toward a more accurate local minimum than arbitrary initialization, as illustrated in Fig. 1.

II. RELATED WORK

Identifying geometric correspondences between a projection surface and a camera, as well as between the surface and a projector, requires estimating the intrinsic and extrinsic parameters of both devices [10]. In a seminal work, Zhang [9] introduced a camera calibration system that utilizes a plane with a printed checkerboard pattern, which was later extended for projector calibration (e.g., [18]).

A popular alternative approach employs well-designed structured light patterns to jointly estimate the parameters of a camera and a projector [8], [19], [20], or a camera and multiple projectors [21]. Scalable calibration methods for scenarios involving multiple cameras and projectors have also been proposed [22], [23].

In addition to the geometric calibration techniques mentioned above, photometric compensation, which adjusts the intensity of projected images, is also required when the projection surface exhibits spatially varying colors. For example, Grossberg *et al.* [11] estimated the radiometric relationship between projector illumination and the intensity observed by a camera. Chen *et al.* [12] proposed estimating a single color mixing matrix for a projector-camera system and demonstrated its application for radiometric compensation. Furthermore, thin-plate spline-based modeling of a nonlinear mapping between projector and camera images was presented in [24], [25]. This mapping was also approximated using polynomials [26] and clustering-based interpolation methods [27]. Additionally, Pjanic *et al.* [28] estimated an optimal color mapping for a more complex scenario involving multiple projectors and a learning-based approach. A further learning method employing a neural network trained on pairs of projector and captured images was also introduced [13].

An alternative to the separate handling of geometric and photometric compensation is to adjust the projector input for a camera-projector system through a unified framework. Huang and Ling [15] proposed a neural framework consisting of two sub-networks, each responsible for geometric and photometric correction, respectively. This framework was later further improved in subsequent work [14]. Additionally, Wang *et al.* [29] introduced an efficient neural framework capable of compensating projector input at high image resolutions. Li *et al.* [30] formulated full compensation as an optimization problem based on a physics-based model that captures the relationship between the camera and projector images.

As additional examples of leveraging neural networks, Huang and Ling [31] modeled the mappings between camera and projector images as image-based shadings using convolutional neural networks, and Erel *et al.* [32] estimated a neural reflectance field from multi-view camera images to perform projector compensation for novel-view synthesis.

A recent alternative to using neural networks for full projector compensation is to approximate projector mapping using physically based light transport in a simulatable rendering space [16], [33], where a (modeled) projector input image is iteratively optimized through a differentiable rendering framework (e.g., [34]).

Recent differentiable rendering-based approaches iteratively refine a projector input image using a gradient descent-based optimizer. However, such optimization typically converges to a local minimum rather than a global one. The accuracy of the result, i.e., the difference between the user-provided target and the camera-captured image, can depend on the starting point of the optimization (i.e., the initial guess for the projector input). This paper proposes a lighting model-guided initialization method to steer the optimization toward a desirable minimum rather than relying on arbitrary values

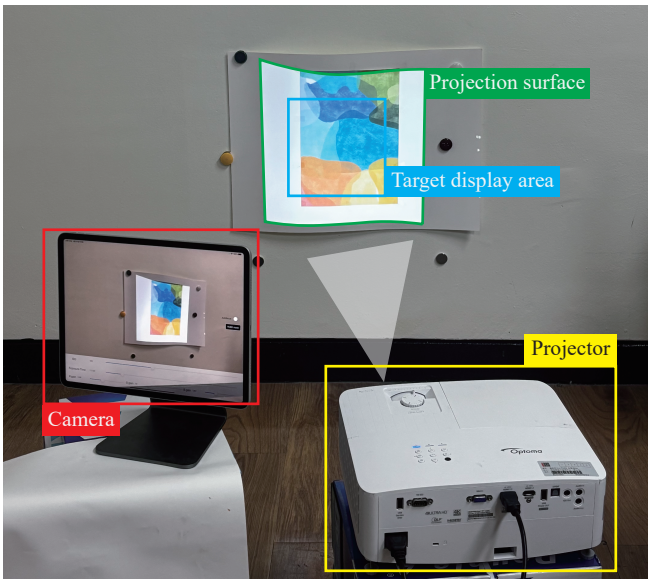


FIGURE 2. **Illustration of our experimental projector-camera system, consisting of a projector (Optoma UHD50X), a camera mounted on an iPad Pro, and a non-planar textured projection surface. The goal of projector mapping is to optimize the projector input image so that the projected output within the target display area closely matches a user-provided target image. A custom iOS application was used to capture RGB-D images at a resolution of 640 × 480 and transmit them via Ethernet to a desktop computer equipped with an NVIDIA RTX 2080 Ti GPU for projector compensation.**

TABLE 1. **List of main notations used in this paper.**

Symbol	Description
I^{target}	User-provided target image
I	Camera-captured image
$f(\theta^{\text{in}}, \Omega)$	Real-world mapping function (θ^{in}: projector input, Ω: real-world scene parameters)
$\hat{f}(\theta, \hat{\Omega})$	Rendering function (θ: virtual projector input, $\hat{\Omega}$: estimated scene parameters)
$\mathcal{W}$	Image warping function to correct geometric discrepancies between $\hat{\Omega}$ and Ω
$\mathbb{B}$	Bias correction term for photometric discrepancies between $\hat{f}(\theta, \hat{\Omega})$ and $f(\theta^{\text{in}}, \Omega)$
$L(x, \omega_i)$	Incident radiance at surface point x from incident light direction ω_i
$\rho(x, \omega_o, \omega_i)$	Bidirectional scattering distribution function (BSDF)
$\theta(u, v)$	Texel value at texture coordinate (u, v)

(e.g., initializing to zeros as in [16]). We demonstrate that this simple modification enables a differentiable rendering-based projector compensation method to achieve improved outputs without altering the underlying gradient descent-based algorithm.

III. BACKGROUND AND PROBLEM SPECIFICATION

Let us consider a projector-camera system in which a projector projects an input image θ^{in} onto a surface, and a camera captures the resulting image (see an example configuration in Fig. 2).

A projector compensation framework adjusts the projector input image θ^{in} so that the camera-captured image I matches a user-provided target image I^{target}. The notations used throughout the paper are summarized in Table 1.

We represent the relationship between the input θ^{in} and the output I using a mapping function f as follows:

$$I = f(\theta^{\text{in}}, \Omega), \tag{1}$$

where Ω denotes all scene parameters that influence the mapping f, such as the geometries and materials of the surface, the position and orientation of the camera, and the position, direction and intensity of the projector. The scene parameters Ω are assumed to be fixed.

To obtain a desired projector input that produces an output image I close to the target image I^{target}, the unknown function f can be approximated by a tractable model $\hat{f}$ using estimated scene parameters $\hat{\Omega}$.

We provide a brief overview of PCDR [16], a gradient descent-based projector compensation framework that approximates the unknown real-world mapping f using a virtual model based on physically based light transport:

$$f(\theta^{\text{in}}, \Omega) = \hat{f}(\theta, \hat{\Omega}) - \mathbb{B}, \tag{2}$$

where the rendering function $\hat{f}$ takes a virtual projector input θ, which is transformed into the real projector input θ^{in} via an image warping function $\mathcal{W}$ (i.e., $\mathcal{W}(\theta) = \theta^{\text{in}}$). Specifically, the warping function $\mathcal{W}$ is computed based on the difference between the rendered output $\hat{f}(\theta, \hat{\Omega})$ and the camera-captured image I. The term $\mathbb{B}$ is a bias correction component that accounts for the discrepancy between the real output image $I = f(\theta^{\text{in}}, \Omega)$ and the virtual output image $\hat{f}(\theta, \hat{\Omega})$.

It is also necessary to compute the estimated scene parameters $\hat{\Omega}$ in order to evaluate the virtual function $\hat{f}(\theta, \hat{\Omega})$. Since modeling all scene parameters in real space is challenging, PCDR estimates only a subset of them. Specifically, it assumes that the projection surfaces have Lambertian reflectance and estimates their geometries and colors from an RGB-D image. The positions and orientations of both the camera and the projector are then determined by minimizing the squared difference between rendered and captured images [16].

This process can be performed only once as a pre-processing step when the scene parameters Ω are static, as it is independent of the user-specified target image I^{target}.

After this pre-processing step, the framework minimizes the following cost function for each user-provided image I^{target}:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left\| I^{\text{target}} - \hat{f}(\theta, \hat{\Omega}) + \mathbb{B} \right\|^2. \tag{3}$$

Specifically, the optimized virtual projector input θ^* is computed via an iterative optimization, implemented within a differentiable rendering framework [34]. During this optimization, the bias correction term $\mathbb{B}$ is also updated iteratively, based on the difference between the virtual rendering output $\hat{f}(\theta, \hat{\Omega})$ and the camera-captured image I obtained using the real projector input $\theta^{\text{in}} = \mathcal{W}(\theta)$ at each gradient step.

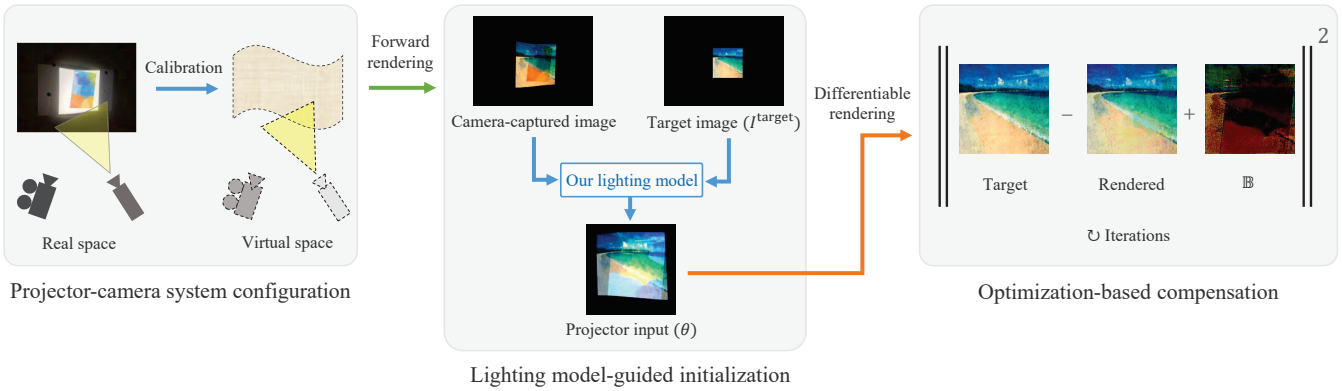


FIGURE 3. Overview of the projector compensation framework (PCDR) with our lighting model-guided initialization. We leverage a functional relationship between the camera-captured image and the target image through a lighting model to compute a projector input θ. This computed input is then provided as the starting point for the gradient descent-based optimization in PCDR, replacing its original initialization (e.g., a constant input).

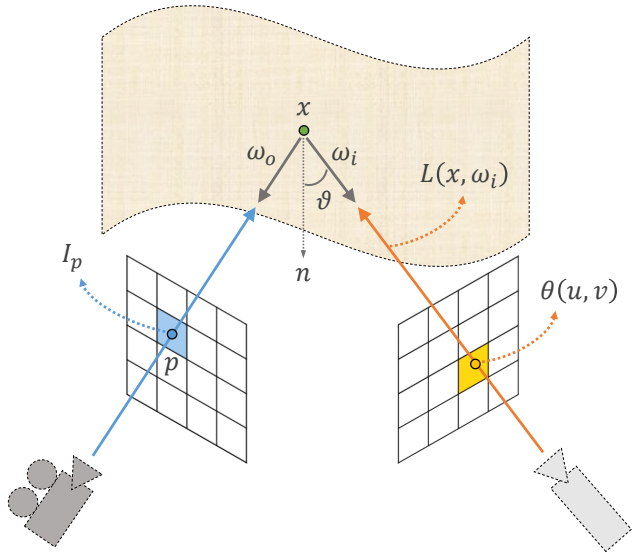


FIGURE 4. Illustration of our method for computing the projector input θ. To model the pixel value I_p at pixel p in the camera-captured image, we first generate a ray for p and compute its intersection point x on the projection surface. The point x is then projected into the projector input image θ, and the incident radiance $L(x, \omega_i)$ is approximated using the corresponding texel value $\theta(u, v)$. We formulate the direct lighting model (Eq. 5) using this incident radiance (Eq. 6), which enables the determination of the projector input image by substituting the pixel value I_p with a desired target value I_p^{target} through Eq. 7.

Once the iterative optimization converges, the final projector input is set as $\theta^{\text{in}} = \mathcal{W}(\theta^*)$.

A. OUR PROBLEM

The main technical advantage of modeling the unknown function f with a rendering function $\hat{f}$ is that it enables mimicking the real-world process f through physically based light simulation such as path tracing [35] and allows differentiation of $\hat{f}$ with respect to its input parameter θ, using a differentiable rendering framework [34]. This makes it possible to apply gradient descent-based optimization (e.g., Adam [17]) to minimize the cost function (Eq. 3).

However, to employ such an iterative optimization method effectively, it is important to initialize the parameter θ appropriately. Note that the optimized parameter θ^* (i.e., a local minimum) obtained through this process does not necessarily correspond to the global minimum.

A simple strategy is to set the initial value of θ manually. For example, PCDR initializes θ to zero. Other straightforward alternatives include assigning the colors of the target image to the initial value or using a random initialization. However, such naïve strategies can degrade the performance of projection compensation, leading to a large discrepancy between the camera-captured image I and the target image I^{target}, since they do not take into account the relationship between the projector input and output.

In this paper, we propose a new initialization scheme that improves the performance of a gradient descent-based projector compensation framework by replacing manual initialization with a lighting model-guided initialization, without altering the underlying optimization algorithm. The details of this scheme are presented in the subsequent section.

IV. LIGHTING MODEL-GUIDED INITIALIZATION

This section presents our initialization method for determining the starting point of the iterative optimization in the projector compensation framework (see Sec. III). Fig. 3 provides an overview of the proposed initialization process. The method takes the estimated scene parameters $\hat{\Omega}$ from the framework as input and determines the starting point of the optimization under two assumptions: (1) the projection surfaces have Lambertian reflectance, and (2) the projector light is the sole light source.

Note that the optimization operates in a virtual space using the rendering function $\hat{f}$. Therefore, our initialization process focuses on determining a suitable virtual projector input θ before the optimization begins. To this end, we model the p-th pixel value I_p of the camera-captured image using the direct lighting integral [36]:

$$I_p \approx \int_{\mathcal{H}} \rho(x, \omega_o, \omega_i) L(x, \omega_i) \cos\vartheta d\omega_i, \tag{4}$$

which integrates the incident radiance $L(x, \omega_i)$ from direction ω_i over a hemisphere $\mathcal{H}$ at a surface point x, i.e., the intersection between the projection surface and a primary ray generated from the virtual camera. $\rho(x, \omega_o, \omega_i)$ is the bidirectional scattering distribution function (BSDF), and $\cos\vartheta$ is computed using the dot product between ω_i and the surface normal n at x. Fig. 4 illustrates this process.

We further simplify the direct lighting formulation by assuming that the scene contains only a single point light source, specifically a light source located at the projector position. Under this assumption, the direct lighting equation reduces to:

$$I_p \approx \rho(x, \omega_o, \omega_i) L(x, \omega_i) \cos\vartheta. \tag{5}$$

We use the projector light model implemented in Mitsuba [37], which determines the incident radiance $L(x, \omega_i)$ based on a texture image (θ in our case):

$$L(x, \omega_i) = \begin{cases} \frac{\pi\theta(u_x, v_x) SV(x, x_{\text{proj}})}{\|x - x_{\text{proj}}\|^2 \cos^3\alpha} & \text{if } (u_x, v_x) \in [0, 1]^2 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where x_{proj} is the position of the virtual projector, and $\theta(u_x, v_x)$ is the texel value sampled at (u_x, v_x) in the texture image θ. The projected coordinates (u_x, v_x) corresponding to the surface point x are computed by projecting x onto the texture image θ using a perspective transformation [37]. In addition, $\cos\alpha = -\omega_i \cdot n'$, where n' denotes the direction of the projector. The visibility term $V(x, x_{\text{proj}})$ is assumed to be 1, which indicates that no occluders exist between the surface point x and the projector position x_{proj}. The parameter S is a user-defined scalar that controls the intensity of the projector light and should be adjusted according to the hardware configuration of the tested projector. In our experiments, we set $S = 2 \times 10^4$ based on the hardware setup shown in Fig. 2, and we used this value for all tests reported in the paper.

The lighting model (Eq. 5 together with Eq. 6) defines a functional relationship between the camera-captured image I and the virtual projector input θ (i.e., the texture image). Since our goal is to determine the input θ such that the resulting image I closely matches the target image I^{target}, this relationship enables the computation of $\theta(u_x, v_x)$ by substituting I_p with I_p^{target} in Eq. 5 and expressing it in terms of $\theta(u_x, v_x)$ using Eq. 6:

$$\theta(u_x, v_x) = \frac{I_p^{\text{target}} \|x - x_{\text{proj}}\|^2 \cos^3\alpha}{\pi S \rho(x, \omega_o, \omega_i) \cos\vartheta}, \tag{7}$$

where I_p^{target} is the p-th pixel value in the target image I^{target}. Once $\theta(u_x, v_x)$ is computed for each surface point x, which is identified by a primary ray through pixel p, the texture values $\theta(u, v)$ are assigned at coordinates (u, v) that are spatially close to (u_x, v_x). In our experiments, unless otherwise specified, sixteen primary rays are generated per pixel p.

However, generating primary rays per pixel does not guarantee that every texel will receive at least one value since the sampling density in texture space can be non-uniform, unlike image space, where primary rays uniformly sample pixels. As a result, some texels can receive no samples (i.e., zero texel values), as illustrated in Fig. 5(b).

Algorithm 1 Lighting model-guided initialization

Input: $\hat{\Omega}$, I^{target}
Output: Initialized projector input θ

$\theta \leftarrow 0$, $W \leftarrow 0$
for each pixel p
  for $n = 1$ to N *// generate N rays per pixel*
    Generate a primary ray for pixel p
    $x \leftarrow$ ray-surface intersection
    $(u_x, v_x) \leftarrow$ project x to the projector's texture space
    Compute $\theta(u_x, v_x)$ *// Eq. 7*
    for each neighbor (u, v) of (u_x, v_x)
      $w \leftarrow G(u_x - u, v_x - v)$ *// Eq. 9*
      $\theta(u, v) \leftarrow \theta(u, v) + w \cdot \theta(u_x, v_x)$
      $W(u, v) \leftarrow W(u, v) + w$
    end for
  end for
end for
$\theta \leftarrow \theta \oslash W$ *// element-wise division*
Return θ

To mitigate this problem, for each projected surface point (u_x, v_x), we update not only the nearest texel value but also its spatial neighborhood (u, v) using a kernel G:

$$\theta(u, v) = \frac{\sum_{x \in X} \theta(u_x, v_x) G(u_x - u, v_x - v)}{\sum_{x \in X} G(u_x - u, v_x - v)}, \tag{8}$$

where X is the set of all surface points obtained from the intersections between the surfaces and primary rays, and $\theta(u_x, v_x)$ is computed according to Eq. 7. The kernel G is defined as a truncated Gaussian:

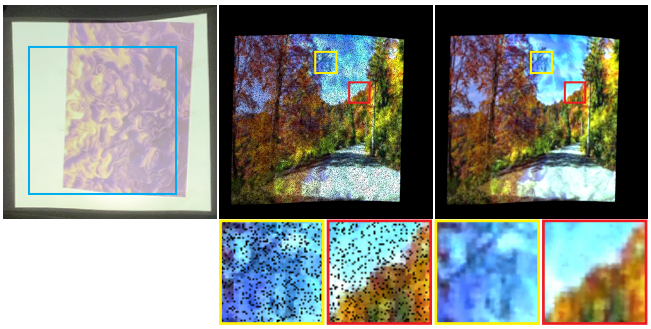
$$G(u_x - u, v_x - v) = \exp\left(-\frac{(u_x - u)^2 + (v_x - v)^2}{2\sigma^2}\right), \tag{9}$$

where the bandwidth σ is set to $\sqrt{3}$ and the kernel $G(u, v)$ is truncated such that it becomes zero when $|u_x - u| > 2$ or $|v_x - v| > 2$. As shown in Fig. 5(c), it drastically reduces the number of the texels that do not receive any update values, compared to using only the nearest-neighbor texel as in Fig. 5(b).

Once the texture image θ is computed, it is used as the initial parameter in the gradient descent-based projector compensation framework for the optimization in Eq. 3. The pseudocode of the proposed initialization process is given in Algorithm 1.

V. RESULTS AND DISCUSSION

We integrated our initialization scheme into a recent projector compensation framework, PCDR [16], replacing its original initialization method (i.e., setting the parameter θ to zero) with our lighting model-guided scheme.



(a) Projection surface (b) Nearest neighbor (c) Gaussian kernel

FIGURE 5. **Initialized projector input images θ for the projection surface (a). If we update only the texel $\theta(u, v)$ at the nearest position (u, v) to the projected point (u_x, v_x) in the projector input image, some texel values remain uninitialized (i.e., appear as zero pixels), as shown in (b). To mitigate this, we apply a Gaussian kernel (c) to update the neighboring texels around (u_x, v_x), effectively reducing the number of uninitialized pixels. For this test, we use 16 rays per pixel.**

Specifically, PCDR employs an on-the-fly learning strategy that performs a two-stage optimization process: the first stage adjusts only the virtual projector input θ over 200 iterations, excluding the bias term $\mathbb{B}$ in Eq. 3; the second stage jointly optimizes both θ and $\mathbb{B}$ for an additional 300 iterations. When using our initialization in place of the constant-zero initialization, we observed that the first stage becomes unnecessary due to the improved starting point. Consequently, we skip the initial stage and directly perform joint optimization for 500 iterations. We also tested two deep learning-based projector compensation methods, CompenNet++ [15] and CompenNeSt++ [14]. We trained the two learning-based methods using the training dataset provided by CompenNet++ on our hardware configuration for fair comparison.

a: Comparisons

We compare the camera-captured results from PCDR with and without our initialization, along with the outputs of two learning-based methods, CompenNet++ and CompenNeSt++. For these comparisons, we used the first 100 target images from the CompenNet++ test dataset and evaluated performance on two projection surfaces. Fig. 6(a) shows the projection surfaces that have two complex texture patterns on a curved surface in Fig. 2.

Under this test configuration, each compensation method is expected to adjust its projector input such that the resulting projector output (i.e., the captured camera image) closely matches the user-provided target image shown in Fig. 6(b). To quantitatively evaluate the quality of these output images, we report peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [38], computed based on the differences between the target and output images.

Table 2 reports the numerical accuracy, i.e., the average PSNR and SSIM values computed from 100 target images, while Fig. 6 presents qualitative comparisons for four target images. Fig. 7 shows the PSNR and SSIM values of all tested methods across the target images.

TABLE 2. **PSNR and SSIM comparisons of projector compensation methods for the two projection surfaces (see Fig. 6(a)). We tested 100 target images for each surface and report the average PSNR and SSIM values, respectively.**

Surface	Method	PSNR↑	SSIM↑
1st	CompenNet++	20.33	0.7781
	CompenNeSt++	20.02	0.7846
	PCDR	21.33	0.7910
	PCDR w/ ours	**22.47**	**0.8414**
2nd	CompenNet++	22.08	0.7891
	CompenNeSt++	22.16	0.7925
	PCDR	26.02	0.8338
	PCDR w/ ours	**27.08**	**0.8875**

As shown in the table and figures, the on-the-fly learning method (PCDR), which leverages a physically based rendering framework, generally outperforms the two learning-based alternatives. It indicates that modeling the camera-projector system using a tractable light transport algorithm enables more accurate adjustment of the projector input by explicitly accounting for variations in surface materials and geometries. When PCDR incorporates our initialization scheme, its visual results and numerical accuracy improve significantly. For example, as reported in Table 2, integrating our initialization module into PCDR increases its numerical accuracy by approximately 1 dB in PSNR and 0.05 in SSIM.

These results highlight the practical importance of properly setting the initial parameters for gradient descent-based optimization. Notably, this performance gain is achieved through a simple modification to PCDR: replacing its default initialization with our lighting model-guided scheme.

b: Convergence comparisons of PCDR with and without our initialization

Fig. 8 compares the numerical convergence of PCDR when initialized either with the original method (setting the projector input image θ to zeros) or with our lighting model-guided initialization. With the original initialization, PCDR optimizes θ alone for the first 200 iterations, then jointly optimizes both θ and the bias term $\mathbb{B}$ (as defined in Eq. 3) for the remaining 300 iterations.

In contrast, when using our initialization, we let PCDR begin with joint optimization from the start, as the accuracy of our initial guess is similar to one of PCDR with the original one after 200 iterations, as shown in Fig. 8. As illustrated, PCDR with the original initialization starts from significantly lower PSNR values due to the zero input and requires more iterations to reach convergence than PCDR with our initialization. Furthermore, our initialization enables PCDR to achieve higher final PSNR values after 500 iterations.

c: Analysis of reconstruction kernels G

To reduce the number of uninitialized texels in the projector input image θ, we employ a reconstruction kernel G in Eq. 9. Fig. 9 compares our chosen Gaussian kernel with an alternative: bilinear interpolation. As shown, both kernels (bilinear

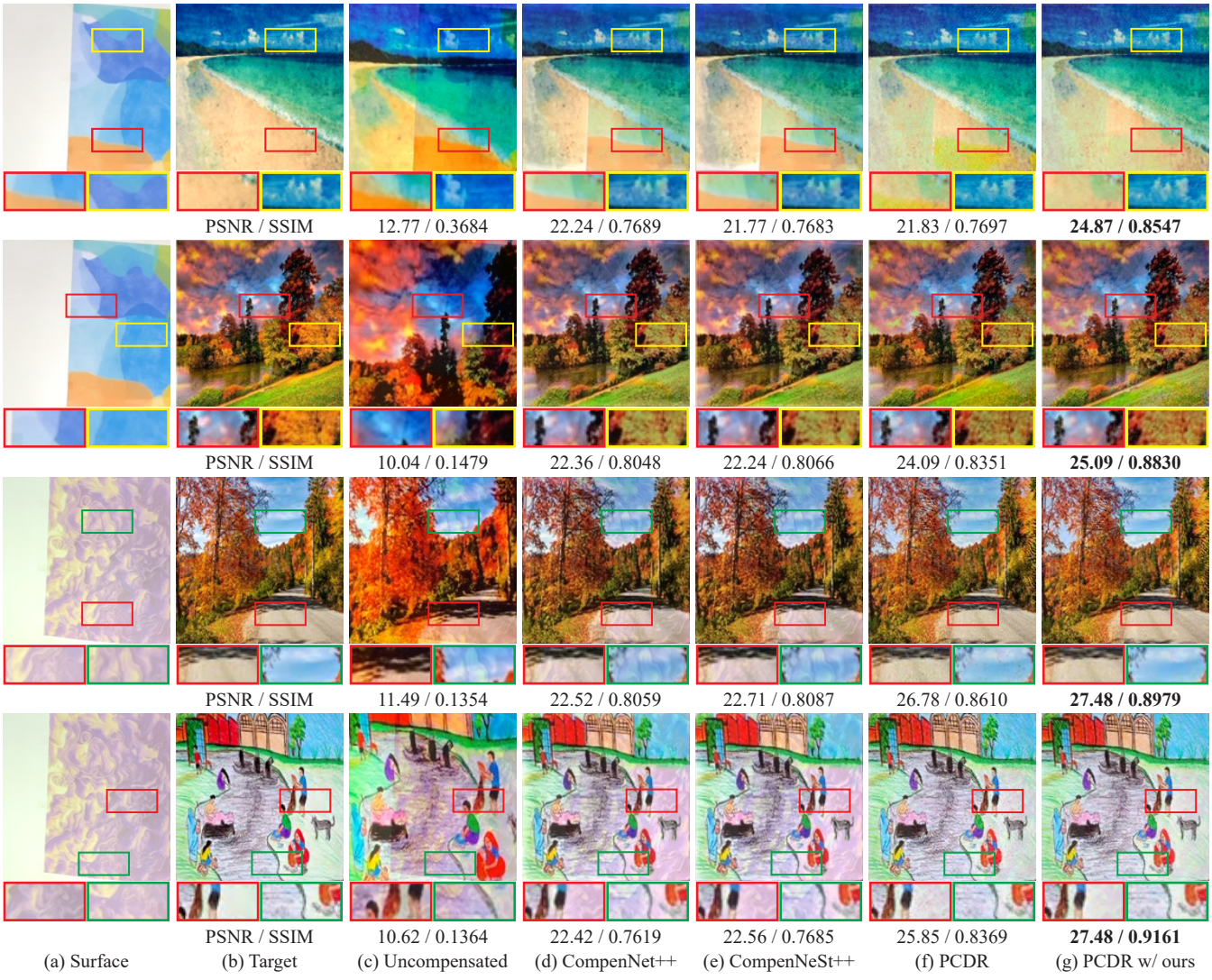


FIGURE 6. Comparison of projector compensation frameworks for a projector-camera setup with a curved projection surface (as shown in Fig. 2). We evaluate two printed textured patterns (shown in (a)) under varying target images. The projected outputs (i.e., camera-captured images) within the target display area are shown, and we report PSNR and SSIM values based on the differences between the projected outputs and the corresponding target images. When the projector input is directly set to the target image without any compensation (as in (c)), strong visual artifacts caused by the surface texture are observed, resulting in low numerical accuracy. Learning-based methods (d) and (e), as well as the differentiable rendering-based approach PCDR (f), mitigate these distortions to some extent. However, when our initialization technique (g) is integrated into PCDR, both the visual quality and numerical accuracy further improve. This highlights the practical importance of proper initialization in gradient descent-based optimization for accurate projector compensation.

and Gaussian) significantly reduce the number of uninitialized texels compared to the nearest-neighbor approach that does not use spatial filtering. However, our Gaussian kernel demonstrates greater robustness in initializing texels, particularly when the number of rays per pixel is small.

d: Analysis of sample counts and computational overheads

Fig. 10 shows the projector input θ initialized by our technique before optimization, along with their corresponding camera-captured images. We vary the number of primary rays per pixel used during initialization. When a relatively small number of rays is used (e.g., 1 or 4 samples per pixel), some pixels in the projector input remain uninitialized (visible as black pixels), resulting in slightly lower quality in the corresponding camera-captured images compared to those using more samples. Increasing the sample count from 16 to 64 does not yield noticeable accuracy improvements. However, the computational overhead increases from 0.44 seconds to 1.86 seconds. Based on this trade-off, we selected 16 samples per pixel for tests unless otherwise mentioned. Moreover, the computational cost of 0.44 seconds for our initialization, given the selected sample count, is relatively small compared to the total optimization time of PCDR, which is 495 seconds.

It is worth noting that PCDR is not the only framework to which our initialization can be applied. Since our method serves as a plug-in module for gradient descent-based projector compensation without altering the underlying algorithm, it can also be integrated into potentially faster frameworks than

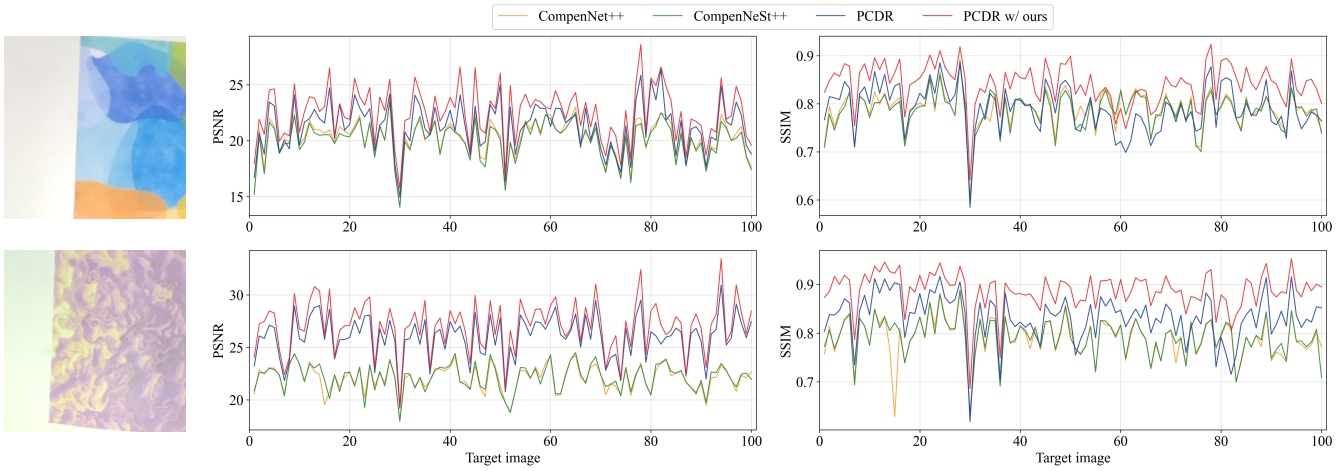


FIGURE 7. For the two projection surfaces (a), we plot the PSNR (b) and SSIM (c) values of the four tested methods for each target image (from the 1st to the 100th). PCDR provides more accurate compensation than the learning-based approaches (CompenNet++ and CompenNeSt++), except for a small number of cases (e.g., lower SSIM values around the 60th target image for the first surface). When our initialization module is integrated into PCDR, its numerical accuracy is consistently improved, yielding higher PSNR and SSIM values than PCDR with its original initialization (which sets the initial values to zero).

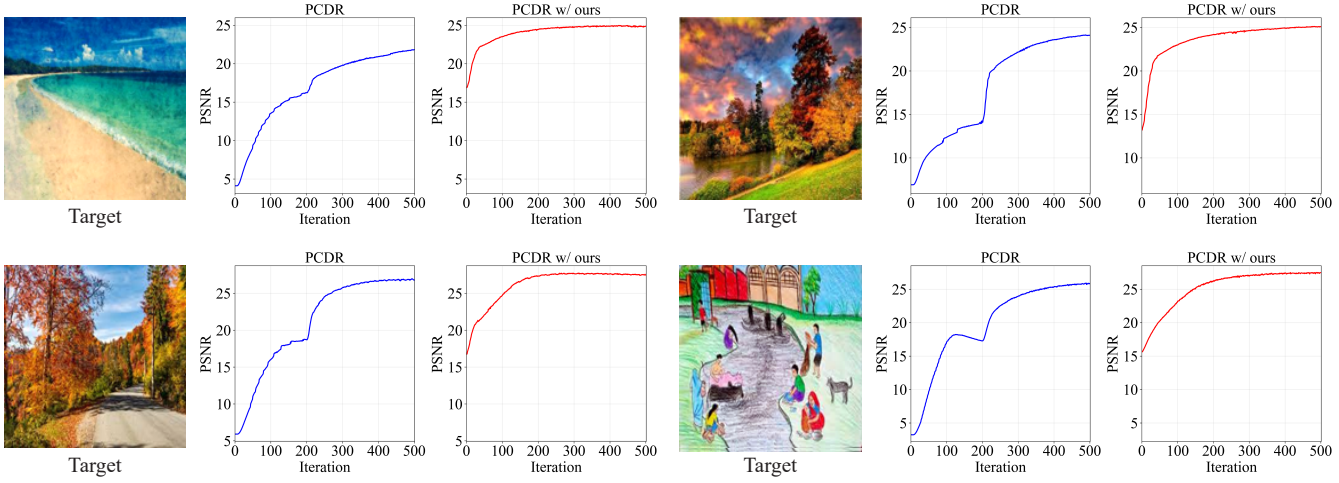


FIGURE 8. Numerical convergence of PCDR when initialized with its original setting versus our lighting model-guided initialization, evaluated across four target images. The test conditions match those used in Fig. 6. Our initialization enables PCDR to begin gradient descent-based optimization from a significantly better starting point, as evidenced by the higher PSNR values of PCDR w/ ours compared to those of the original PCDR at the first iteration. Furthermore, our technique leads to improved final results, enabling the existing framework to achieve higher numerical accuracy by the final iteration.

PCDR. In such cases, it may be desirable to use a smaller sample count than 16 samples per pixel, e.g., 1 sample per pixel, which reduces the computational overhead to 33 ms.

e: Analysis of different initialization schemes

In Fig. 11, we compare our lighting model-guided initialization with two simpler alternatives: 1) target-based initialization, where the projector input θ is directly set using the target image, and 2) random initialization, where θ is initialized with values drawn from a uniform distribution. For both alternatives, we directly start the joint optimization of θ and $\mathbb{B}$ in PCDR (Eq. 3), as we do when applying our initialization.

As shown in Fig. 11(b), target-based initialization produces visual artifacts due to discontinuities in the textured projection surfaces (see Fig. 6(a)). It indicates that directly using the target image as the optimization starting point can lead to an undesirable local minimum. Initializing θ randomly (Fig. 11(c)) makes the optimization more robust to such artifacts, but introduces residual noise. Consequently, neither alternative outperforms the original PCDR initialization. For example, the original PCDR configuration (i.e., setting the parameters to zero tested in Fig. 6(f)) achieves higher accuracy than both target-based and random initialization when tested on the four target images.

In contrast, our method enables PCDR to produce more accurate and visually pleasing results through a lighting model-guided initialization. This demonstrates that proper initialization in gradient descent-based optimization is crucial, as it can

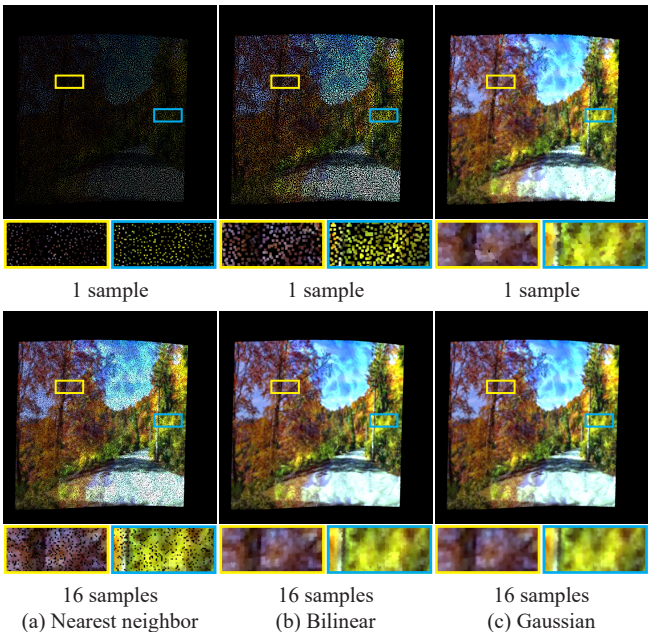


FIGURE 9. **Analysis of the reconstruction kernel G used in our lighting model-guided initialization. We compare the initialized projector input images using three settings for G: nearest neighbor without spatial filtering, bilinear interpolation (b), and a Gaussian kernel (c). Two sampling rates are tested: 1 ray per pixel (top row) and 16 rays per pixel (bottom row). Applying spatial filtering ((b) and (c)) helps reduce the number of uninitialized texels (i.e., zero-value pixels in the resulting images). Among the tested approaches, the Gaussian kernel (c) yields the most effective results, particularly at low sampling rates (top row).**

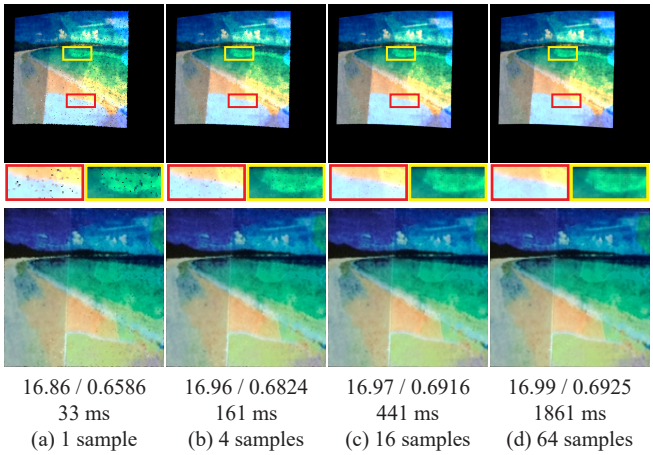


FIGURE 10. **Analysis of our lighting model-based initialization using varying numbers of rays per pixel, ranging from 1 to 64. The first row shows the initialized projector input images for each sampling rate, while the second row displays the corresponding projected outputs (prior to PCDR optimization). Increasing the number of samples improves the accuracy of the initial projector input; however, the improvement becomes marginal beyond 16 samples. Given the increasing computational overhead with higher sample counts, we select 16 rays per pixel as the default setting, balancing performance and efficiency.**

guide the process toward a better local minimum.

f: Limitations and future work

A technical limitation of our technique is that it relies on estimated scene parameters, $\hat{\Omega}$, which are also used as input to the projector compensation framework (PCDR). Specifically, this framework assumes that projection surfaces have only Lambertian reflectance, and our method inherits the same assumption. We also assume that the surfaces are illuminated solely by the projector light when estimating the virtual projector input θ.

In projection mapping scenarios involving glossy surfaces or external lighting, these assumptions are violated. In such cases, the gap between the unknown real projection mapping f and the rendering function $\hat{f}$ can become large. Although the optimization in PCDR includes a bias term ($\mathbb{B}$ in Eq. 2) to compensate for this gap, heavy reliance on this term is undesirable because its estimation can be inaccurate, as it is modeled simply as the difference between the virtual rendering output and the camera-captured image. Consequently, when these assumptions are not met (e.g., due to glossy projection surfaces or external lights), both our initialization results and the optimization results of PCDR may suffer reduced accuracy.

As an important direction for future work, it would be valuable to extend our initialization to handle more complex materials (i.e., non-Lambertian surfaces) and external lighting without depending on the inputs required by a specific framework.

Another potential research direction is to investigate integrating our initialization method into other projector compensation frameworks that rely on physically based rendering (e.g., [33]).

VI. CONCLUSION

In this paper, we presented a new scheme for determining the starting point of a gradient descent–based optimization that adjusts the projector input through physically based rendering. We employ a well-established lighting (rendering) model and derive a functional relationship between the observed camera-captured image and the projector input image. This relationship allows us to compute the projector input directly in a virtual space, enabling a given projector compensation framework to begin its optimization from our output rather than relying on a manually selected initialization (e.g., constant colors). We demonstrated that incorporating our method into an existing projector compensation framework leads to substantial improvements without modifying the underlying algorithm.

ACKNOWLEDGMENT

We thank the reviewers for their insightful comments, which have helped improve this work.

REFERENCES

[1] O. Bimber and R. Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*. USA: A. K. Peters, Ltd., 2005.

[2] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay, "Shader lamps: animating real objects with image-based illumination," in *Proceedings of the 12th Eurographics Conference on Rendering*, ser. EGWR'01. Goslar, DEU: Eurographics Association, 2001, p. 89–101.

FIGURE 11. Projected outputs of PCDR using different initialization schemes for its optimization. We compare our lighting model-guided initialization (d) with two alternatives: 1) target-based initialization (b) that assigns the projector input directly from the target image (a) and 2) random initialization (c) that sets each texel in the projector input image randomly using a uniform distribution. The surface configurations used in this test are the same as those in Fig. 6. After optimization, alternative initializations result in visual artifacts (b) or residual noise (c). In contrast, PCDR, with our initialization, achieves visually more pleasing and numerically more accurate results.

[3] A. D. Wilson and H. Benko, "Projected augmented reality with the roomalive toolkit," in *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ser. ISS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 517–520.

[4] M. R. Mine, J. van Baar, A. Grundhofer, D. Rose, and B. Yang, "Projection-based augmented reality in disney theme parks," *Computer*, vol. 45, no. 7, pp. 32–40, 2012.

[5] N. Tosa, R. Nakatsu, P. Yunian, and K. Ogata, "Projection mapping celebrating rimpa 400th anniversary," in *2015 International Conference on Culture and Computing (Culture Computing)*, 2015, pp. 18–24.

[6] S. Murayama, I. Torii, and N. Ishii, "Development of projection mapping with utility of digital signage," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 895–900.

[7] Y. Hosomizo, D. Iwai, and K. Sato, "A flying projector stabilizing image fluctuation," in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 2014, pp. 31–32.

[8] B. Huang, S. Ozdemir, Y. Tang, C. Liao, and H. Ling, "A single-shot-per-pose camera-projector calibration system for imperfect planar targets," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2018, pp. 15–20.

[9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[10] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer, "The visual computing of projector-camera systems," in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH '08. New York, NY, USA: Association for Computing Machinery, 2008.

[11] M. Grossberg, H. Peri, S. Nayar, and P. Belhumeur, "Making one object look like another: controlling appearance using a projector-camera system," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. 1–8.

[12] X. Chen, X. Yang, S. Xiao, and M. Li, "Color mixing property of a projector-camera system," in *Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems*, ser. PROCAMS '08. New York, NY, USA: Association for Computing Machinery, 2008.

[13] B. Huang and H. Ling, "End-to-end projector photometric compensation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019, p. 6803–6812.

[14] B. Huang, T. Sun, and H. Ling, "End-to-end full projector compensation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2953–2967, 2022.

[15] B. Huang and H. Ling, "CompenNet++: End-to-end full projector compensation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7164–7173.

[16] J. Park, D. Jung, and B. Moon, "Projector compensation framework using differentiable rendering," *IEEE Access*, vol. 10, pp. 44 461–44 470, 2022.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[18] M. Sugimoto, D. Iwai, K. Ishida, P. Punpongsanon, and K. Sato, "Directionally decomposing structured light for projector calibration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, p. 4161–4170, Nov. 2021.

[19] D. Moreno and G. Taubin, "Simple, accurate, and robust projector-camera calibration," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 2012, pp. 464–471.

[20] C. Xie, H. Shishido, Y. Kameda, and I. Kitahara, "A projector calibration method using a mobile camera for projection mapping system," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2019, pp. 261–262.

[21] P. Kurth, V. Lange, C. Siegl, M. Stamminger, and F. Bauer, "Auto-calibration for dynamic multi-projection mapping on arbitrary surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 11, pp. 2886–2894, 2018.

[22] M. A. Tehrani, M. Gopi, and A. Majumder, "Automated geometric registration for multi-projector displays on arbitrary 3d shapes using uncalibrated devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 4, pp. 2265–2279, 2021.

[23] M. Son and K. Ko, "Multiple projector camera calibration by fiducial marker detection," *IEEE Access*, vol. 11, pp. 78 945–78 955, 2023.

[24] A. Grundhöfer, "Practical non-linear photometric projector compensation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 924–929.

[25] A. Grundhöfer and D. Iwai, "Robust, error-tolerant photometric projector compensation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5086–5099, 2015.

[26] P. Kurth, V. Lange, M. Stamminger, and F. Bauer, "Real-time adaptive color correction in dynamic projection mapping," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2020, pp. 174–184.

[27] Y. Li, A. Majumder, M. Gopi, C. Wang, and J. Zhao, "Practical radiometric compensation for projection display on textured surfaces using a multidimensional model," *Computer Graphics Forum*, vol. 37, pp. 365–375, 05 2018.

[28] P. Pjanic, S. Willi, D. Iwai, and A. Grundhöfer, "Seamless multi-projection revisited," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 11, pp. 2963–2973, 2018.

[29] Y. Wang, H. Ling, and B. Huang, "CompenHR: Efficient full compensation for high-resolution projector," in *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 2023, pp. 135–145.

[30] Y. Li, W. Yin, J. Li, and X. Xie, "Physics-based efficient full projector compensation using only natural images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 8, pp. 4968–4982, 2024.

[31] B. Huang and H. Ling, "DeProCams: Simultaneous relighting, compensation and shape reconstruction for projector-camera systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2725–2735, 2021.

[32] Y. Erel, D. Iwai, and A. H. Bermano, "Neural projection mapping using reflectance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 11, p. 4339–4349, Nov. 2023.

[33] J. Li, Q. Deng, H. Ling, and B. Huang, "DPCS: Path tracing-based differentiable projector-camera systems," *IEEE Transactions on Visualization and Computer Graphics*, 2025.
[34] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 38, no. 6, Dec. 2019.
[35] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '86. New York, NY, USA: Association for Computing Machinery, 1986, p. 143–150.
[36] M. Pharr, W. Jakob, and G. Humphreys, "Physically based rendering: From theory to implementation," 2023.
[37] W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang, "Mitsuba 3 renderer," 2022, https://mitsuba-renderer.org.
[38] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

WOOCHAN SUN received the B.S. degree from the Department of Applied Artificial Intelligence, Ajou University in 2022, and the M.S. degree from the Department of AI Convergence, College of Information and Computing, Gwangju Institute of Science and Technology (GIST) in 2025. His research interests include inverse rendering and computer vision.

GEONU NOH received the B.S degree from the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST) in 2020. He is currently pursuing the integrated M.S. and Ph.D. degree in Department of AI Convergence, College of Information and Computing, Gwangju Institute of Science and Technology. His research interests are rendering and denoising.

KWANGHEE KO received the B.S. degree in naval architecture and ocean engineering from Seoul National University, in 1995, and the M.S. degree in mechanical and ocean engineering and the Ph.D. degree in ocean engineering from MIT, in 2001 and 2003, respectively. From 2003 to 2004, he was a Postdoctoral Researcher with the Seagrant College Program, MIT. From 2004 to 2005, he was a Research Associate with the Stevens Institute of Technology. He joined the Gwangju Institute of Science and Technology, in 2006. He was an assistant professor, from 2006 to 2010, an associate professor, from 2010 to 2016, and has been a professor, since 2016. He is currently a Professor with the Gwangju Institute of Science and Technology, Republic of Korea. His research interests include CAD/CAM/CAE, geometric modeling, digital twin application, projection mapping, virtual reality, and augmented reality.

BOCHANG MOON (Memeber, IEEE) received the M.S and Ph.D. degrees in computer science from KAIST, in 2010 and 2014, respectively. He is an associate professor at Gwangju Institute of Science and Technology (GIST). Before joining GIST, he was a postdoctoral researcher at Disney Research. His research interests include rendering, denoising, and augmented and virtual reality.