IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2022, accepted April 16, 2022, date of publication April 22, 2022, date of current version April 29, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169861

Projector Compensation Framework Using Differentiable Rendering

JINO PARK, DONGHYUK JUNG, AND BOCHANG MOON, (Member, IEEE)
School of Integrated Technology, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

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

This work was supported by the Ministry of Culture, Sports and Tourism and the Korea Creative Content Agency under Project R2020070004.

ABSTRACT Projection mapping is a widely adopted technique in various applications. A typically projector-camera system consists of a projector and camera pair where a projector emits an application-specific input image onto a surface, and the camera captures the projected output image. Given an ideal projection surface with white-colored planar geometries, the camera (or a viewer) can observe the projected output image without any visual distortion. The surface, however, can have arbitrary shapes and textured colors in practice, and it often introduces visually distracting artifacts to the output image. It results in lowering the viewing experience drastically. We propose a projector compensation framework that adjusts the projector input image so that the camera can see the projected output image with a much-reduced distortion. Our key contribution is to model the real projection mapping process with a virtual but controllable light simulation and optimize the projector input using differentiable rendering. We demonstrate that our new framework produces a more accurate output than state-of-the-art methods given complex projection surfaces.

INDEX TERMS Projector compensation, geometric calibration, photometric compensation, differentiable rendering.

I. INTRODUCTION

Projection mapping has been widely used in various applications such as medical applications [4], [8], [29], exhibition [21], [31], and spatial augmented reality [25], [30], [34]. A conventional (but ideal) scenario for such projection mapping is that a projector projects a target image onto a flat surface (screen) with white color, and in this setting, a viewer can observe the projected image without any geometric and photometric distortion. However, in practice, a projection surface often has arbitrary geometries or textured colors (e.g., not just a flat wall with white color). It often drastically lowers the viewing experience on projected images since a target image that we want to show on a projection surface can be distorted geometrically and photometrically.

Projector compensation tackles the challenge, introduced by a non-ideal projection surface, by adjusting the input image projected on the surface so that a viewer (or a camera) can see a target image without distortion. To this end, we should solve two major technical problems: geometric calibration and photometric compensation. In the geometric calibration, the shape of a projector input image should be amended (i.e., a warped image) by taking geometries of a projection surface into account. On the other hand, photometric compensation is a process that adjusts the color in the warped image so that the projected image on a projection surface can be seen from a viewer without color distortion.

A classical but common approach is to design hand-crafted mapping functions for the geometric calibration (e.g., [15], [22], [35]) or photometric compensation process (e.g., [9], [10], [23]) between a projector input and observed output image (e.g., a projected output captured by a camera). However, it can be technically challenging to accurately model a complex light interaction between the light (i.e., a projector) and a projection surface with arbitrary geometries and colors.

A recent alternative to the classical approaches is to employ a deep neural network that infers the optimal projector input image to match an output projection image with a target image [12], [16]. While the state-of-the-art learning-based methods demonstrated outstanding results compared to classical models by taking both geometric and photometric correction into account, their visual artifacts (i.e., a discrepancy

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi.

between the result and target image) can still become significant, especially when the geometries of a projection surface become complex, as shown in an example figure (Fig. 1).

We propose a novel projector compensation framework that seeks an optimal projector input to make the projected output similar to a target image for a typical projection mapping system with a projector-camera pair. Our key idea is to model a projection mapping process with a virtual but controllable rendering process and estimate the optimal input image using *differentiable rendering*. It allows us to optimize the projector input image by gradient descent while minimizing the difference between a target image and an observed output image without relying on a deep neural network. The technical contributions of our paper are summarized as follows.

- We simulate a projection mapping process in a virtual space so that a light transport algorithm can mimic the real projection mapping process. Technically, it allows us to explicitly model a functional relationship between a projector input and a projected output image as a rendering process.
- We formulate an optimization problem with a rendering function to estimate the optimal projector input and solve the problem using a multi-stage optimization process that utilizes differentiable rendering.

We demonstrate that our projection mapping framework using differentiable rendering can reproduce a target image more accurately than the state-of-the-art method, especially when a projection surface is non-planer or colored (e.g., Fig. 1).

FIGURE 1. Projection mapping results of the state-of-the-art (CompenNet++) [12] and our method (ours) given a non-planar and colored projection surface. The blue-colored box (in (a)) denotes the target display area where we want to show the target (ground truth) image (d) on the surface. Both methods estimate optimal projector input images ((b) and (c)) for this projection mapping task by adjusting its shape and colors. Our framework optimizes the input image using differentiable rendering and produces a more accurate output image (f) than the state-of-the-art result (e).

II. RELATED WORK

This section discusses existing methods that solve either geometric calibration or photometric compensation and explains comprehensive methods that solve both problems together like our method.

A. GEOMETRIC CALIBRATION

As a seminal work, Zhang [37] proposed a camera calibration method that can estimate intrinsic and extrinsic parameters of a camera by taking a few photos on a planar surface with a regular pattern. This initial work inspired geometric calibration methods for a projector-camera system. For example, Moreno and Taubin [22] presented a calibration procedure for structured-light systems, which estimates the correspondences between a projector input image and a camera-captured image using a checkerboard pattern. Yang *et al.* [35] used randomly distributed circular patterns instead of a checkerboard pattern for calibrating a projector-camera system. Recently, Huang *et al.* [15] addressed the imperfect planarity of the board used for calibration using a bundle adjustment algorithm.

B. PHOTOMETRIC COMPENSATION

Nayar *et al.* [23] is an early work that handled photometric compensation so that a non-ideal projection surface (e.g., a flat but textured surface) could be utilized as a projector display given a projector-camera system. Grossberg *et al.* [9] improved the initial work and presented a memory-efficient compensation framework. Also, Grundhöfer and Iwai [10] presented a non-linear color mapping between a projector input and camera-captured image using thin-plate spline (TPS) [6], [7].

It can be beneficial to directly model light interactions between surfaces for more accurate photometric compensation, especially given a non-planar projection surface. For example, Bimber *et al.* [1], Wetzstein and Bimber [33] and Sheng *et al.* [27] formulated the photometric compensation as an inverse light transport problem and estimated an optimal projector input that makes a projected output image similar to a target image. As a recent example, Huang and Ling [13] proposed an end-to-end neural network (CompenNet) that learns an optimal color mapping for the compensation.

C. GEOMETRIC AND PHOTOMETRIC CORRECTION

Raskar *et al.* [26] proposed a pioneer work that models a projection mapping with a rendering process and projects intensity-adjusted input images onto a real object while estimating the relative distances between projectors and the object. Illuminating a real object using multiple projectors was further studied, e.g., multi-projection mapping for

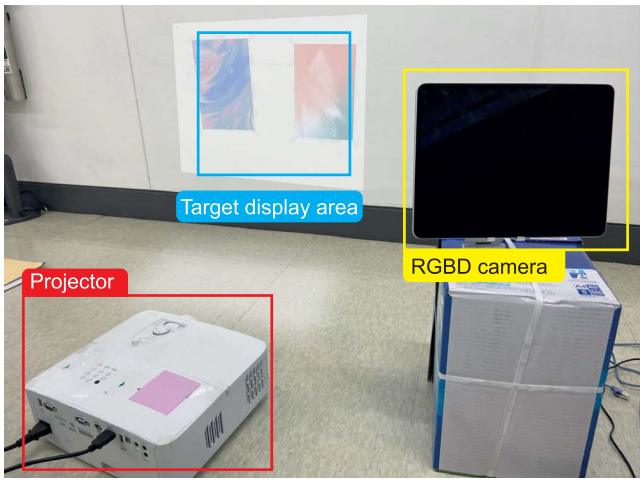


FIGURE 2. Our projector-camera system consisting of a projector and RGBD camera. Our framework estimates an optimal projector input so that a projected output image in the target display area is matched to a target image.

moving objects [19] and real-time mapping framework supporting multiple projectors [28].

Designing an end-to-end neural network for projector compensation was recently investigated. Huang and Ling [12] proposed a unified framework (CompenNet++) that handles both geometric and color correction by joining a newly designed submodule (WarpingNet) for geometric calibration into CompenNet [13] for photometric correction. Later, Huang *et al.* [16] optimized the CompenNet++ and presented a refined deep neural network (CompenNeSt++) with a reduced number of learnable parameters. Also, Huang and Ling [14] presented a sophisticated neural network that models light interactions with a projection surface and addressed image-based relighting, photometric correction, and estimation for depth and normal estimation simultaneously.

Our framework solves the geometric calibration and photometric compensation for a projector-camera system like the recent learning-based methods. However, the key difference of the proposed framework is that we employ a full light transport algorithm, path tracing [18], to model light interactions with a projection surface more accurately and estimate mapping functions for the calibrations via *differentiable rendering*. Our modeling of a projection mapping process with a differentiable rendering framework enables us to seek an optimal projector input close to a target image without training a deep neural network using a dataset, unlike the recent learning-based methods.

III. PROBLEM SPECIFICATION AND MOTIVATION

We configure a projector-camera system (Fig. 2) that consists of a projector and RGBD camera. We assume the projector and camera are static and located in front of a projection surface. We do not assume that the projection surface has a flat and white wall and will verify our projector compensation by varying the geometries and colors of the surface.

Let us formulate that the projected image y is generated by a projection mapping function f that takes a projector input image x and scene parameters Ω:

$$y = f(x, \Omega). \tag{1}$$

The parameters Ω includes all physical parameters that can affect the output image y. Examples are locations and orientations of the camera and projector and other information such as geometries and colors of a projection surface.

Our projector compensation problem is to find an optimal input image x^* that minimizes the squared difference between the observed image y and a target (ground truth) image y^t:

$$x^* = \underset{x}{\operatorname{argmin}} \left\| y^t - f(x, \Omega) \right\|^2. \tag{2}$$

A. TECHNICAL CHALLENGES AND OUR MOTIVATION

The major technical challenge in estimating the optimal projector input is to model the unknown light transport function f with an approximate but controllable (differentiable) function so that the optimal input can be estimated using the approximate function. For such approximation, existing approaches exploited simple hand-crafted functions such as matrix multiplication [3], [9], [23], [33], [36] and TPS [10], [11]. Also, the recent learning-based techniques [12], [16] approximate the unknown with a deep neural network. Nevertheless, their approximation quality can be degraded, as their approximation functions did not consider a complete light transport process. For example, the projected image can largely deviate from the target image when a complex projection surface is given, as shown in Fig. 1. Our main idea for tackling this challenge is to replace the unknown function with a light transport simulation, which can mimic the real light propagation in a virtual world. Note that a general light transport algorithm (e.g., path tracing [18]) can simulate the unknown function by taking multiple light bounces (i.e., global illumination) into account. Also, the light transport simulation can be differentiated with respect to its input parameters (e.g., virtual scenes and lights) thanks to differentiable rendering frameworks (e.g., Mitsuba renderer [24]). It enables us to consider arbitrary projection surfaces and estimate the optimal projector input more accurately.

IV. PROJECTOR COMPENSATION USING DIFFERENTIABLE RENDERING

Our projector compensation framework aims to infer an optimal projector input x^* that results in a projected image y (captured by a camera) close to a target image y^t. Our key idea for the compensation task is to model the unknown projector mapping function $f(x, \Omega)$ (Eq. 1) by a rendering function $\hat{f}(z, \hat{\Omega})$:

$$f(x, \Omega) = \hat{f}(z, \hat{\Omega}) - \theta, \tag{3}$$

where $\hat{\Omega}$ is virtual scene parameters fed into the rendering function $\hat{f}(\cdot)$ and θ is a bias image that compensates for the approximation error, i.e., $\hat{f}(\cdot) - f(\cdot)$. Table 1 summarizes the main notations used in the paper.

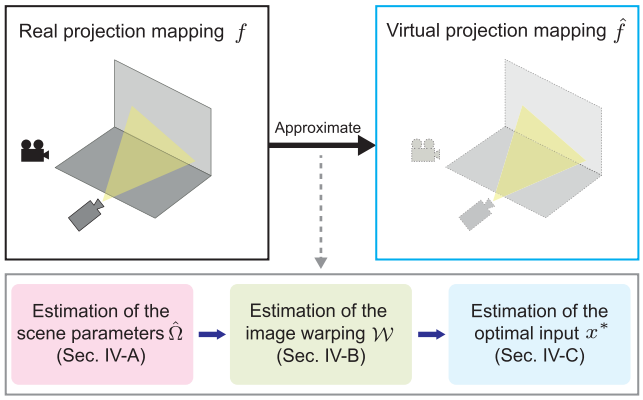


FIGURE 3. Our estimation framework for projector compensation. We approximate a projection mapping in the real space with a virtual rendering process and seek an optimal projector input x^*. Our estimation consists of the three main stages: estimation of scene parameters (Sec. IV-A), image warping function (Sec. IV-B), and optimal projector input (Sec. IV-C).

We also model the relation between the real projector input x and a virtual projector input z as

$$x = \mathcal{W}(z), \tag{4}$$

where $\mathcal{W}$ is an image warping function that compensates a geometric discrepancy between two projected images from the real and virtual space.

For the virtual scene parameters $\hat{\Omega}$, we only consider a subset of all potential scene parameters Ω since in practice it is not possible to model all physical parameters. Precisely, we define the virtual parameters as a set $\hat{\Omega} = \{\mathcal{C}_o, \mathcal{C}_d, \mathcal{P}_o, \mathcal{P}_d, \mathcal{S}_g, \mathcal{S}_c\}$ that includes the following:

- The position and direction of the camera ($\mathcal{C}_o$ and $\mathcal{C}_d$),
- The position and direction of the projector ($\mathcal{P}_o$ and $\mathcal{P}_d$),
- The geometries and colors of the projection surface ($\mathcal{S}_g$ and $\mathcal{S}_c$).

Given the functional relationship (Eq. 3) with the definitions of the virtual parameters, finding the optimal projector input x^* is technically equivalent to estimating the scene parameters $\hat{\Omega}$ (Sec. IV-A), warping function $\mathcal{W}(\cdot)$ (Sec. IV-B), and virtual projector input z and bias terms θ (Sec. IV-C). Unfortunately, estimating all the parameters simultaneously in a single step is technically challenging since this optimization can have many local minimums. As a practical route for tackling such a challenge, we present a multi-stage framework (Fig. 3) that estimates the parameters step-by-step using differentiable rendering and will be detailed in the subsequent sections.

A. ESTIMATION OF THE SCENE PARAMETERS

The first process (Fig. 4) in our framework is to compute the estimated scene parameters $\hat{\Omega}$ (in Eq. 3). To this end, we formulate an objective function as

$$\hat{\Omega} = \underset{\hat{\Omega}'}{\operatorname{argmin}} \left\| y - \hat{f}(z, \hat{\Omega}') \right\|^2, \tag{5}$$

TABLE 1. Notations used throughout the paper.

Notation	Description
$f(x, \Omega)$	Real-world projection mapping function (x: a projector input, Ω: real scene parameters)
$\hat{f}(z, \hat{\Omega})$	Simulated projection mapping function (z: a virtual projector input, $\hat{\Omega}$: estimated scene parameters)
y	Projected output image (i.e., a camera-captured image)
y^t	Target image (i.e., ground truth)
$\mathcal{W}(z)$	Image warping function (for considering errors in the scene estimates $\hat{\Omega}$)
θ	Bias of a virtual projection mapping, $\hat{f}(z, \hat{\Omega}) - f(x, \Omega)$

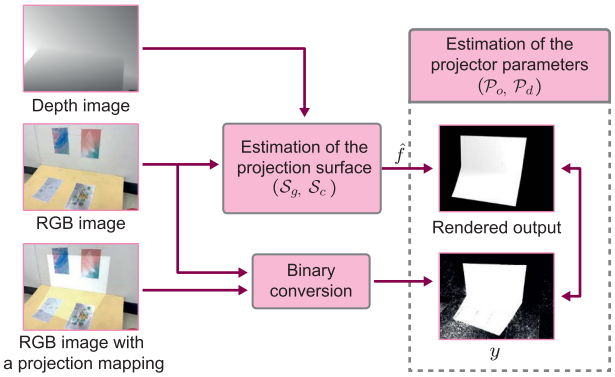


FIGURE 4. Our estimation of the scene parameters $\hat{\Omega}$. We set the origin $\mathcal{C}_o$ and direction $\mathcal{C}_d$ of a camera to [0, 0, 0] and [0, 0, 1] and estimate the geometries $\mathcal{S}_g$ and colors $\mathcal{S}_c$ of a projection surface using the camera-captured RGB and depth images. We then estimate the position $\mathcal{P}_o$ and direction $\mathcal{P}_d$ of a projector by comparing a rendered and projected output image via differentiable rendering.

where we obtain the estimated scene parameter $\hat{\Omega}$ that minimizes the squared difference between y and rendered image $\hat{f}(\cdot)$. We set the virtual input z by a constant image (e.g., white-colored). For the y, we generate a black-and-white image using the two camera-captured images, with and without a projection mapping. We use the constant image for the projection mapping. Specifically, we set one to the pixel colors in y if the color difference is higher than a threshold (e.g., 0.1) and set zero otherwise. We found that this binary conversion is beneficial for estimating the scene parameters more accurately since we can only consider an important image area directly affected by the projector.

Note that we omit the bias term θ (in Eq. 3) designed for the color discrepancy between the real and virtual projector output, and this will be optimized separately (in Sec. IV-C).

Intuitively, the minimization problem (Eq. 5) is for estimating relative positions and directions among the camera, projector, and surfaces. We set the camera position $\mathcal{C}_o$ and direction $\mathcal{C}_d$ to $[0, 0, 0]$ and $[0, 0, 1]$, and estimate the coordinates of the other parameters (i.e., relative distances in positions and directions from the camera parameters).

For the projection surface, we estimate its geometries and colors ($\mathcal{S}_g$ and $\mathcal{S}_c$) using an RGBD image captured

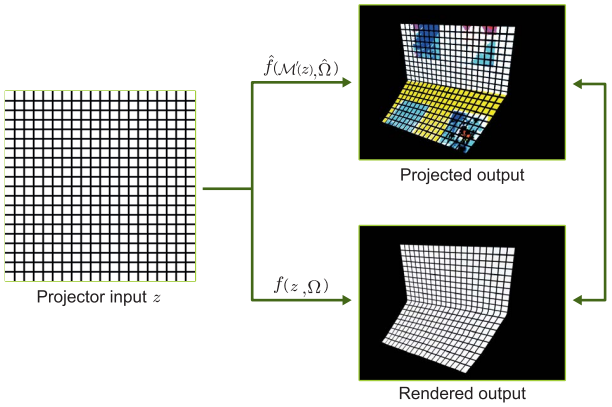


FIGURE 5. The estimation of the inverse warping function $\mathcal{M}$. We feed a checkerboard image z to the real projector and rendering function. Note that the virtual projection mapping (i.e., the rendering) takes a warped image (i.e., $\mathcal{M}'(z)$), unlike the real projection. We estimate an optimal inverse function $\mathcal{M}$ that minimizes the squared error between the rendered and projected output images.

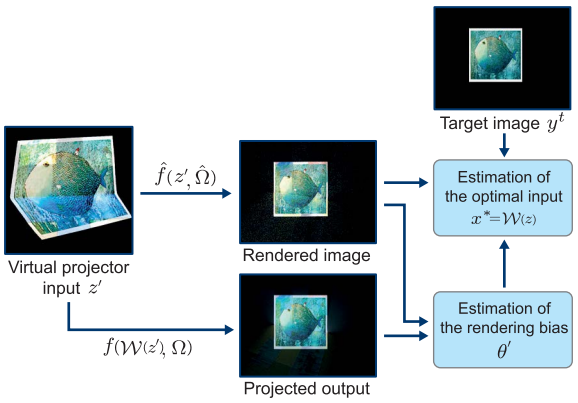


FIGURE 6. Our estimation of the optimal projector input x^* ($x^* = \mathcal{W}(z)$) that results in a projected output close to a target image. Once we estimate the scene parameters and image warping function, we perform this last process to optimize the projector input using differentiable rendering. We also consider a bias term θ for compensating a rendering bias for the final estimation.

by the camera. We assume that the surface has Lambertian reflectance, and we reconstruct a triangular mesh with a diffuse texture using the mesh reconstruction libraries, MeshLab [5] and Open3D [38].

Once we set the parameters for the camera and projection surface, we solve the optimization problem (Eq. 5) for the remaining scene parameters (i.e., $\mathcal{P}_o$ and $\mathcal{P}_d$ for the projector) using a differentiable rendering framework (Mitsuba [24]). Technically, the framework allows us to simulate the projection mapping in a virtual space via a general light transport algorithm (i.e., path tracing [18]) and outputs the optimal parameters that minimize the squared loss between the given image y and a rendered image $\hat{f}(z, \hat{\Omega}')$ via gradient descent.

B. ESTIMATION OF THE IMAGE WARPING FUNCTION

Once we find the estimated scene parameters $\hat{\Omega}$ (Sec. IV-A), we estimate an image warping operator $\mathcal{W}$:

$$\mathcal{W} = \underset{\mathcal{W}'}{\operatorname{argmin}} \left\| f(\mathcal{W}'(z), \Omega) - \hat{f}(z, \hat{\Omega}) \right\|^2, \tag{6}$$

where we use a checkerboard image for the virtual projector input z. Unfortunately, this optimization cannot be solved directly using differentiable rendering since we cannot differentiate the unknown function $f(\mathcal{W}'(z), \Omega)$ with respect to the parameter $\mathcal{W}'$.

As an alternative, we instead estimate a pseudo inverse $\mathcal{M}$ of the warping $\mathcal{W}$, and then compute the warping $\mathcal{W}$ from the estimated inverse $\mathcal{M}$. Fig. 5 illustrates our process for the estimation. Specifically, we reformulate the optimization problem (Eq. 6) into

$$\mathcal{M} = \underset{\mathcal{M}'}{\operatorname{argmin}} \left\| y - \hat{f}(\mathcal{M}'(z), \hat{\Omega}) \right\|^2. \tag{7}$$

Note that the parameter $\mathcal{M}'$ is an argument of a differentiable rendering function $\hat{f}(\cdot)$, and thus we can optimize it using a differentiable rendering framework. Specifically, we employ the thin-plate spline (TPS) [6], [7] for representing the $\mathcal{M}'$, which were also utilized in [12] and [16]. To generate the y, we project a checkerboard image onto a projection surface, and we apply a binary transform into each color channel in the captured image. We set one to the color channel if it is higher than a threshold (e.g., 0.7). Otherwise, we assign zero to the values.

After computing the estimated inverse warping $\mathcal{M}$, it is straightforward to find the image warping $\mathcal{W}$ by solving the simple optimization:

$$\mathcal{W} = \underset{\mathcal{W}'}{\operatorname{argmin}} \left\| \mathcal{W}'(\mathcal{M}(z)) - z \right\|^2. \tag{8}$$

C. ESTIMATION OF THE OPTIMAL PROJECTOR INPUT IMAGE

Once our estimation processes for the scene parameters $\hat{\Omega}$ and warping function $\mathcal{W}$ finish, our final task (Fig. 6) is to estimate the optimal projector input x^* by estimating the remain parameters (the virtual projector input z and bias image θ in Eq. 3).

To this end, we formulate an optimization:

$$z = \underset{z'}{\operatorname{argmin}} \left\| y^t - \left(\hat{f}(z', \hat{\Omega}) - \theta \right) \right\|^2. \tag{9}$$

As a solver for the optimization, we adopt the differentiable rendering framework (Mitsuba [24]) that iteratively optimizes the virtual projector input z using gradient descent. We also optimize the bias compensation term θ within the framework by the following update rule:

$$\begin{aligned} \theta_i &\leftarrow \theta_i - \alpha \frac{\partial \left(\hat{f}_i(z', \hat{\Omega}) - y_i - \theta_i \right)^2}{\partial \theta_i} \\ &\leftarrow \theta_i + 2\alpha \left(\hat{f}_i(z', \hat{\Omega}) - y_i - \theta_i \right), \end{aligned} \tag{10}$$

where the i-th pixel color θ_i in the bias image θ is updated iteratively by gradient descent. We use the subscript i to denote the i-th pixel color for the other images ($\hat{f}(z', \hat{\Omega})$ and y). In the equation (Eq. 10), y ($y = f(x, \Omega)$) is set to a projected output using a projector input image x ($x = \mathcal{W}(z')$) and the learning rate $\alpha = 0.01$. In our implementation, we update only the estimated z image without considering the bias θ for the two hundred iterations in the gradient descent, and then update both parameters simultaneously for the three hundred iterations. Once we obtain the virtual projector input z, we obtain the optimal project input x^* using the virtual input z, i.e., $x^* = \mathcal{W}(z)$.

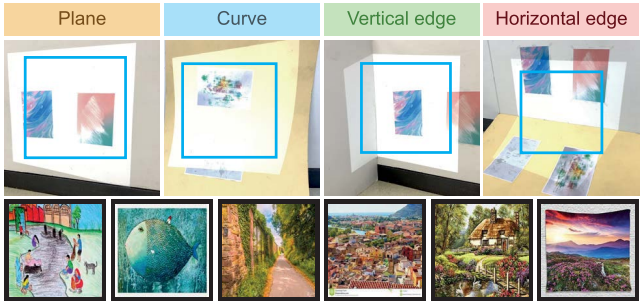


FIGURE 7. Tested projection surfaces (in the top row) and target images (in the bottom). The blue-colored boxes denote the target display areas where we want to produce projected output images close to the target images.

V. SYSTEM AND IMPLEMENTATION DETAILS

A. SYSTEM CONFIGURATION

We have used Optoma UHD50X and iPad Pro 12.9-inch (4th generation) for a projector and RGBD camera in the projector-camera system (Fig. 2). We have set the image resolutions for the RGB color and the projector to 640 × 480 and 800 × 600, which were used in CompenNet++ [12] and CompenNeSt++ [16], for a fair comparison with the existing methods (Sec. VI). In addition, we have used a LiDAR sensor on the tested camera to acquire a depth image of size 256 × 192 and then upscaled it using OpenCV [2] so that its resolution becomes the same as the color image, i.e., 640 × 480. The resolution of a projector input image has been set to 600 × 600, and the brightness of the projector has been set to 40 out of 100 levels. Also, we have implemented a simple iOS application that captures RGBD images and sends them to a desktop with Nvidia RTX 2080 Ti GPU via an ethernet connection and disabled the white balancing and autofocus for the mobile camera device.

B. MODIFICATIONS ON THE DIFFERENTIABLE RENDERING

We have modified the virtual projector implementation of the public differentiable rendering framework, Mitsuba [24]. Let us denote a virtual projector emits the area light (i.e., a projector input image) toward a direction $(u, v, 1)$, where $u \in [u_{min}, u_{max})$ and $v \in [v_{min}, v_{max})$. The original implementation assumes a vertically symmetric projection (i.e., $v_{max} = -v_{min}$), but the tested real projector (Optoma UHD50X) emits the light asymmetrically. We have amended the implementation to support a vertically asymmetric projection.

We observed that the differentiable rendering framework does not find the optimal scene parameters for the projector effectively when we feed the constant projector image (i.e., a white-colored image) (in Sec. IV-A) since it can introduce a discontinuity in the boundaries of a projected area. We apply a smoothing function that was exploited in [20] to the projected radiance to mitigate the discontinuity. Specifically, the output radiance $L_e(i, j)$ for the projector input color at a normalized coordinate $(i, j) \in [0, 1)$ is amended into

$$L'_e(i, j) = L_e(i, j)\Gamma(i)\Gamma(j), \tag{11}$$

where $\Gamma(\cdot)$ is a smoothing function:

$$\Gamma(t) = \begin{cases} t/\epsilon & 0 \le t < \epsilon \\ 1 & \epsilon \le t < 1-\epsilon \\ (1-t)/\epsilon & 1-\epsilon \le t < 1. \end{cases} \tag{12}$$

We set the ϵ to 0.01 so that only the boundary area can be smoothed.

C. COLOR CLAMPING ON THE VIRTUAL INPUT AND OUTPUT

The virtual projector and camera can handle HDR colors, but the tested real projector and camera can project and capture only LDR colors. For the consistency in the colors between the real and virtual space, we have clamped the colors of virtual projector input z and rendering output $\hat{f}(\cdot)$ so that those can be in the LDR range:

$$0 \le z \le 1, 0 \le \hat{f}(\cdot) \le 1. \tag{13}$$

We have also applied a clamping to the rendering bias term θ so that it can be in a range:

$$\hat{f}(\cdot) - 1 \le \theta \le \hat{f}(\cdot). \tag{14}$$

D. PARAMETERS FOR THE DIFFERENTIABLE RENDERING

We have adjusted the field of views for the virtual camera and projector using the hardware specifications of our tested camera and projector. Also, we have set the intensity parameter for the virtual projector so that a rendered image can have a brightness similar to a camera-captured image. Note that this pre-configuration is only required when configuring a specific projector-camera device, can be reused for different projection surfaces and target images. For the forward rendering algorithm (i.e., path tracing), we have disabled its indirect light simulation when we estimate the scene parameters and image warping functions (Sec. IV-A and IV-B) since those are affected mainly by direct lighting. We have simulated both direct and indirect illumination for the optimal projector input (Sec. IV-C).

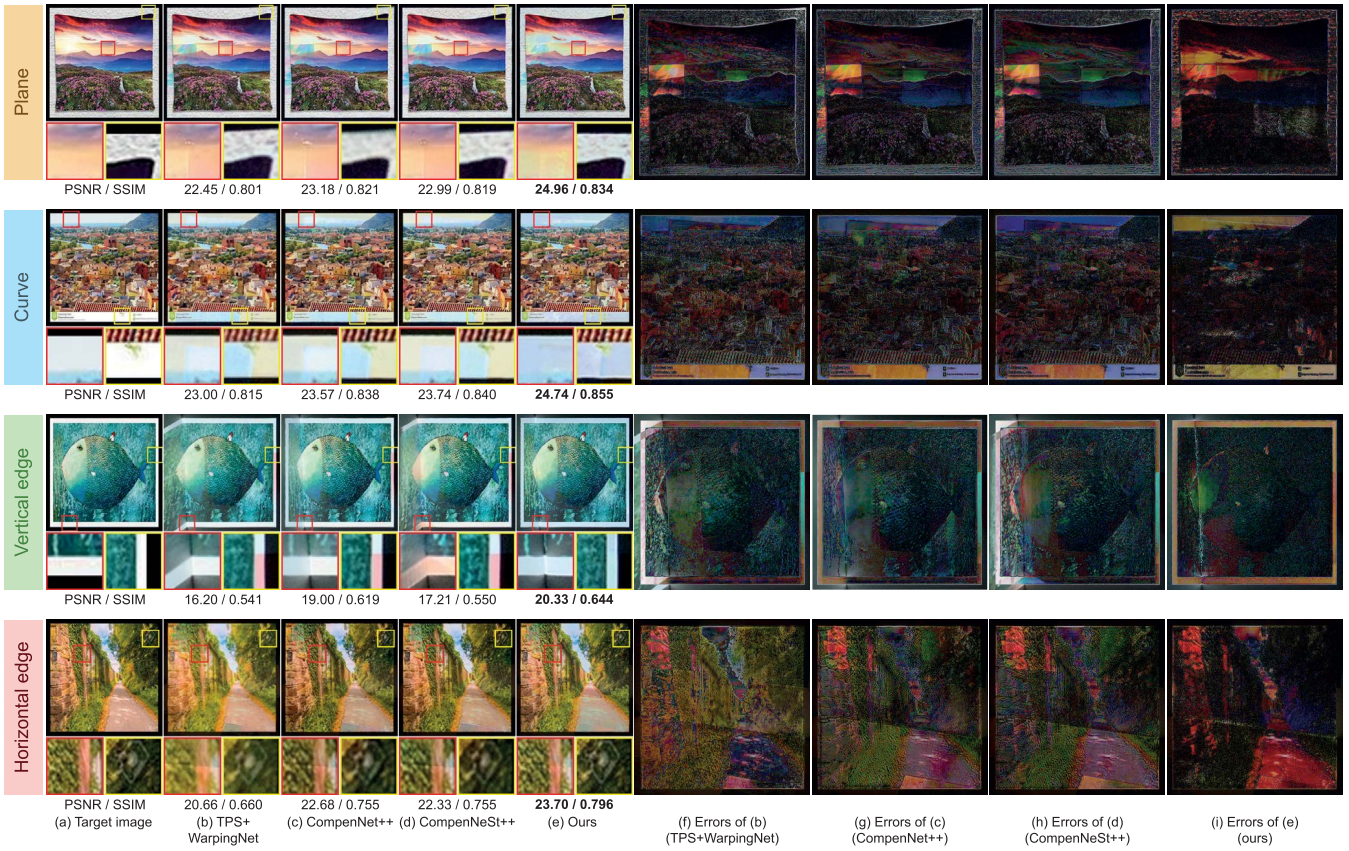


FIGURE 8. **Qualitative comparisons for our method (e) and the state-of-the-art methods, TPS+WarpingNet (b), CompenNet++ (c) and CompenNeSt++ (d). Compared to the existing methods, our framework produces high-quality output images with reduced visual artifacts. The images ((f) to (i)) are visualized errors of the tested methods, i.e., absolute color differences between the projected output and corresponding target images.**

TABLE 2. **Numerical comparisons of the tested methods.**

Method	Metric	Plane	Curve	Vert. edge	Hori. edge
TPS+WarpingNet	PSNR	23.20	23.37	17.31	18.53
	SSIM	0.801	0.793	0.542	0.647
CompenNet++	PSNR	23.41	23.93	19.49	20.37
	SSIM	0.819	0.825	0.614	0.749
CompenNeSt++	PSNR	23.26	24.03	18.39	20.27
	SSIM	0.818	0.824	0.554	0.748
Ours	PSNR	**25.92**	**25.55**	**21.16**	**21.11**
	SSIM	**0.855**	**0.845**	**0.643**	**0.784**

VI. RESULTS AND DISCUSSION

We compare our compensation framework with state-of-the-art methods, a TPS-based photometric compensation [10], CompenNet++ [12], and CompenNeSt++ [16]. Since the TPS-based method considered only the color correction, we have addressed its geometric calibration using WarpingNet, a subnetwork module in CompenNet++ for a fair comparison. We will refer to this method as TPS+WarpingNet.

Note that the compared techniques are learning-based methods requiring training deep neural networks, unlike our method. Hence, we have retrained their deep neural networks for a fair comparison. Specifically, we have retrained the neural networks of CompenNet++ and CompenNeSt++ for the tested projection surfaces using the training dataset used for the original papers. For TPS+WarpingNet, we have retrained WarpingNet (i.e., a subnetwork for CompenNet++ [12]) using the same training dataset and used it for the geometric calibration for the TPS method. Additionally, we have modified the public implementations of CompenNet++ and CompenNeSt++ (also the WarpingNet) to handle a projector input image of size 600×600, which is larger than their original image resolution (256×256). We have also implemented the TPS-based technique.

Given a projector-camera system (Fig. 2), our method and the previous methods are tested using four projection surfaces and six target images. Fig. 7 shows that the tested projection surfaces have different shapes and textures. A plane and a curved surface (the first two columns in Fig. 7) are tested for the relatively straightforward cases, and we use the surfaces with vertical and horizontal edges (the last two columns in

Fig. 7) for challenging scenarios. Also, the six target images (the bottom of Fig. 7) have been generated using the image benchmark released by the authors of CompenNet++ and CompenNeSt++.

We use the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [32] for measuring the numerical accuracy of the tested methods. To evaluate the numerical metrics, we crop the target display areas in the camera-captured output images and resize the ground truth target images to be the same as those of the cropped images.

A. NUMERICAL COMPARISONS

Table 2 reports the PSNR and SSIM values of the tested methods for the four projection surfaces. We test the methods using the six target images for each surface and show the average PSNR and SSIM numbers. As shown in the table, our technique produces more accurate results than state-of-the-art methods. For example, the PSNR numbers of our results are approximately 2.8 dB, 1.6 dB and 1.9 dB higher than the TPS+WarpingNet, CompenNet++, and CompenNeSt++ on average, respectively. Also, our results have lower perceptual errors (i.e., SSIM) than the previous methods.

B. QUALITATIVE COMPARISONS

In Fig. 8, we qualitatively compare our projector compensation and the previous methods. To easily identify their visual artifacts, we also visualize the errors (i.e., absolute differences between the projected outputs and target images). Overall, our method generates visually more pleasing results than TPS+WarpingNet, CompenNet++, and CompenNeSt++. In particular, the existing methods show noticeable visual artifacts for non-planar surface areas (see the zoomed area for the vertical edge in the figure). On the other hand, our method handles such complex geometries relatively well and shows much-reduced distortion. Also, the previous approaches show the artifacts propagated from the textured area (see the zoomed area in the horizontal edge), but our compensation reduces the errors by adjusting our projector input accordingly. It indicates that our method using a full light-transport simulation (i.e., path tracing) can be more robust against the geometric and color changes in the projection surface.

C. ABLATION STUDY

In our virtual light simulation, we use two rendering bias compensation terms, a color bias term θ (Eq. 9) and an image warping function $\mathcal{W}$ (Eq. 6) for a geometric discrepancy between the real and virtual projection mapping. In Fig. 9, we compare our results with and without the bias compensation terms. As shown in the figure, it is noticeable that our result without the color bias term θ shows a different color tone compared to the target image. It demonstrates that considering the bias term when estimating the optimal projector input (in Eq. 9) is necessary for resolving the color difference between the real and virtual projection mapping. Also, it can be a geometric discrepancy in the positions and directions between the real and virtual projector-camera pairs. When we omit the image warping function, i.e., setting the function to the identity function, our projected output produces a visually distracting artifact (e.g., see the striped patterns in Fig. 9 (d)). Our framework that considers both estimation errors shows much-enhanced results visually and numerically compared to the tested alternatives.

FIGURE 9. An ablation study for our projection mapping framework. We compare our projected output (b) with the alternatives that ignore either the color bias term θ (c) or image warping function $\mathcal{W}$ (d). It can be noticeable that the projected outputs become much more accurate when considering both estimation errors during our optimization.

D. LIMITATIONS AND FUTURE WORK

Our framework does handle non-planar projection surfaces with colors but still assumes that the surfaces have Lambertian reflectance. As the virtual light transport simulation can support more complex materials (not just the diffuse material), and thus estimating the general bidirectional reflectance distribution function (BRDF) of the projection surface can be an interesting future direction for handling more general projection surfaces (e.g., glossy surfaces).

Our technique optimizes projector input images once the target images are available and is an offline process. As a result, the proposed method is only applicable to offline projection mapping scenarios where we can record optimal projector input images in advance. For example, the computational overheads for estimating the scene parameters

(Sec. IV-A), image warping function (Sec. IV-B), and optimal input image (Sec. IV-C) are 10, 25, and 11 minutes, respectively. One can reuse the first two processes for a static scene configuration (i.e., a fixed projector-camera pair and projection surface) and only recompute an optimal projector input for each target image.

Nevertheless, it would be interesting to investigate a combination of an image to image learning (e.g., Pix2Pix [17]) and our method for interactive projection mapping scenarios. For example, we can generate a training dataset using our projection compensation and train an image to image learning to make an inference quickly instead of optimizing the optimal input from scratch.

VII. CONCLUSION

We have proposed a new projector compensation framework that estimates an optimal projector input image to prevent a visual distortion in a projected output, especially when a projector surface has colored non-planar geometries. As our main technical contribution, we approximate the real projection mapping with a light transport simulation and optimize the projector input image via differentiable rendering. It enables modeling the complex light transport in a controllable space and estimating the optimal input more accurately than the tested state-of-the-arts. Our compensation framework can be adopted for offline projection mapping scenarios where high-quality projection results are necessary.

REFERENCES

[1] O. Bimber, A. Grundhofer, T. Zeidler, D. Danch, and P. Kapakos, "Compensating indirect scattering for immersive and semi-immersive projection displays," in *Proc. IEEE Virtual Reality Conf. (VR)*, Mar. 2006, pp. 151–158.

[2] G. Bradski, "The OpenCV library," *Softw. Tools Prof. Programmer*, vol. 25, no. 11, pp. 120–123, Nov. 2000.

[3] X. Chen, X. Yang, S. Xiao, and M. Li, "Color mixing property of a projector-camera system," in *Proc. 5th ACM/IEEE Int. Workshop Projector Camera Syst.*, New York, NY, USA, Aug. 2008, pp. 1–6.

[4] T. F. Chen-Yoshikawa, E. Hatano, A. Yoshizawa, and H. Date, "Clinical application of projection mapping technology for surgical resection of lung metastasis," *Interact. CardioVascular Thoracic Surg.*, vol. 25, no. 6, pp. 1010–1011, Dec. 2017.

[5] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: An open-source mesh processing tool," in *Proc. Eurograph. Italian Chapter Conf.*, V. Scarano, R. De Chiara, and U. Erra, Eds., 2008, pp. 129–136.

[6] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *Computer Vision*, A. Heyden, G. Sparr, M. Nielsen, P. Johansen, Eds. Berlin, Germany: Springer, 2002, pp. 21–31.

[7] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory Functions Several Variables*, W. Schempp and K. Zeller, Eds. Berlin, Germany: Springer, 1977, pp. 85–100.

[8] R. Fukuhara, K. Kaneda, T. Tamaki, B. Raytchev, T. Higaki, S. Nishimoto, and Y. Sotsuka, "A projection mapping system onto a human body for medical applications," in *Eurographics*, A. Fusiello and O. Bimber, Eds. Genoa, Italy, Europe, The Eurographics Assoc., May 2019.

[9] 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 *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2004, pp. 1–8.

[10] A. Grundhöfer and D. Iwai, "Robust, error-tolerant photometric projector compensation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5086–5099, Dec. 2015.

[11] A. Grundhofer, "Practical non-linear photometric projector compensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 924–929.

[12] B. Huang and H. Ling, "CompenNet++: End-to-end full projector compensation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7164–7173.

[13] B. Huang and H. Ling, "End-to-end projector photometric compensation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6803–6812.

[14] B. Huang and H. Ling, "DeProCams: Simultaneous relighting, compensation and shape reconstruction for projector-camera systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 5, pp. 2725–2735, May 2021.

[15] 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 *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2018, pp. 15–20.

[16] B. Huang, T. Sun, and H. Ling, "End-to-end full projector compensation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 8, 2021, doi: 10.1109/TPAMI.2021.3050124.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[18] J. T. Kajiya, "The rendering equation," in *Proc. 13th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, New York, NY, USA, 1986, pp. 143–150.

[19] P. Lincoln, G. Welch, and H. Fuchs, "Continual surface-based multi-projector blending for moving objects," in *Proc. IEEE Virtual Reality Conf.*, Mar. 2011, pp. 115–118.

[20] G. Loubet, N. Holzschuch, and W. Jakob, "Reparameterizing discontinuous integrands for differentiable rendering," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–14, Dec. 2019.

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

[22] D. Moreno and G. Taubin, "Simple, accurate, and robust projector-camera calibration," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Visualizat. Transmiss.*, Oct. 2012, pp. 464–471.

[23] S. K. Nayar, H. Peri, M. D. Grossberg, and P. N. Belhumeur, "A projection system with radiometric compensation for screen imperfections," in *Proc. ICCV Workshop Projector-Camera Syst. (PROCAMS)*, Oct. 2003, pp. 1–8.

[24] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–17, Dec. 2019.

[25] P. Punpongsanon, D. Iwai, and K. Sato, "SoftAR: Visually manipulating haptic softness perception in spatial augmented reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 11, pp. 1279–1288, Nov. 2015.

[26] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay, "Shader lamps: Animating real objects with image-based illumination," in *Rendering Techn.*, S. J. Gortler and K. Myszkowski, Eds. Vienna Austria: Springer, 2001, pp. 89–102.

[27] Y. Sheng, T. C. Yapo, and B. Cutler, "Global illumination compensation for spatially augmented reality," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 387–396, May 2010.

[28] C. Siegl, M. Colaianni, 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. Graph.*, vol. 34, no. 6, pp. 1–11, Nov. 2015.

[29] M. Takada, M. Takeuchi, E. Suzuki, F. Sato, Y. Matsumoto, M. Torii, N. Kawaguchi-Sakita, H. Nishino, S. Seo, E. Hatano, and M. Toi, "Real-time navigation system for sentinel lymph node biopsy in breast cancer patients using projection mapping with indocyanine green fluorescence," *Breast Cancer*, vol. 25, no. 6, pp. 650–655, Nov. 2018.

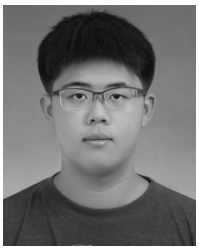
[30] B. H. Thomas, M. Marner, R. T. Smith, N. A. M. Elsayed, S. Von Itzstein, K. Klein, M. Adcock, P. Eades, A. Irlitti, J. Zucco, T. Simon, J. Baumeister, and T. Suthers, "Spatial augmented reality—A tool for 3D data visualization," in *Proc. IEEE VIS Int. Workshop 3DVis (3DVis)*, Nov. 2014, pp. 45–50.

[31] N. Tosa, R. Nakatsu, P. Yunian, and K. Ogata, "Projection mapping celebrating RIMPA 400th anniversary," in *Proc. Int. Conf. Culture Comput. (Culture Computing)*, Oct. 2015, pp. 18–24.

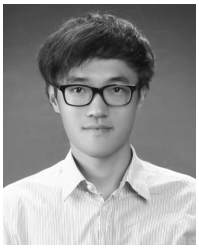
[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] G. Wetzstein and O. Bimber, "Radiometric compensation through inverse light transport," in *Proc. 15th Pacific Conf. Comput. Graph. Appl. (PG)*, Oct. 2007, pp. 391–399.
[34] A. D. Wilson and H. Benko, "Projected augmented reality with the RoomAlive toolkit," in *Proc. ACM Int. Conf. Interact. Surf. Spaces*, New York, NY, USA, Nov. 2016, pp. 517–520.
[35] L. Yang, J.-M. Normand, and G. Moreau, "Practical and precise projector-camera calibration," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2016, pp. 63–70.
[36] T. Yoshida, C. Horii, and K. Sato, "A virtual color reconstruction system for real heritage with light projection," in *Proc. Int. Conf. Virtual Syst. Multimedia (VSMM)*, 2003, pp. 1–7.
[37] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[38] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.

JINO PARK received the B.S. degree in computer science and engineering from Hanyang University, ERICA Campus, Ansan, South Korea, in 2019, and the M.S. degree from the School of Integrated Technology, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2022. His research interests include rendering and differentiable rendering.



DONGHYUK JUNG received the B.S. degree in electrical engineering and computer science (EECS) from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2021, where he is currently pursuing the M.S. degree with the School of Integrated Technology. His research interests include rendering and differentiable rendering.



BOCHANG MOON (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from KAIST, in 2010 and 2014, respectively. He is currently an Associate Professor with the 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. He served as a PC Member of international conferences, including EGSR, I3D, PG, and CGI.