CompenHR: Efficient Full Compensation for High-resolution Projector

Yuxi Wang*
Hangzhou Dianzi University

Haibin Ling†
Stony Brook University

Bingyao Huang‡
Southwest University

ABSTRACT

Full projector compensation is a practical task of projector-camera systems. It aims to find a projector input image, named compensation image, such that when projected it cancels the geometric and photometric distortions due to the physical environment and hardware. State-of-the-art methods use deep learning to address this problem and show promising performance for low-resolution setups. However, directly applying deep learning to high-resolution setups is impractical due to the long training time and high memory cost. To address this issue, this paper proposes a practical full compensation solution. Firstly, we design an attention-based grid refinement network to improve geometric correction quality. Secondly, we integrate a novel sampling scheme into an end-to-end compensation network to alleviate computation and introduce attention blocks to preserve key features. Finally, we construct a benchmark dataset for high-resolution projector full compensation. In experiments, our method demonstrates clear advantages in both efficiency and quality.

Index Terms: Projector compensation—;—Spatial augmented reality—; Projector-camera system

1 INTRODUCTION

As an essential device for spatial augmented reality, projectors are usually combined with cameras to form smart projector-camera systems, and are used in many scientific experiments and real-world applications [4, 5, 14, 24, 34, 35, 38, 42, 43, 46, 48, 53, 54, 65, 70]. However, projection onto non-planar and textured surfaces is still a challenging problem, which limits the applicability of projector-camera systems. As a typical solution, full projector compensation neutralizes geometric and photometric distortions caused by sensor radiometric variation, lens distortion, and surface material reflectance [3, 15, 19, 21, 28, 30, 44, 55, 61, 62, 64, 71]. In particular, a composite function of full projector compensation is estimated from projector input and the corresponding camera-captured images, and then, the compensation image is generated based on the estimated parameters.

Traditional projector compensation methods assume that geometric and photometric distortions are independent. Thus, they formulate these two tasks separately. For geometric correction, a common solution is finding the pixel-to-pixel correspondences with structured light, then generating the corrected image by inverse mapping. For photometric compensation, conventional methods define a per-pixel color mapping function for each camera and projector pixel pair. Recently, with the successful application of deep learning, some works are devoted to modeling full compensation using deep neural networks [26, 29]. Although these end-to-end algorithms overcome the drawbacks of two-step methods, the memory usage and computation cost increase rapidly with image resolutions, and thus they are less practical for high-resolution setups. Therefore, how to compensate for the high-resolution input of a projector under constrained conditions has yet to be studied.

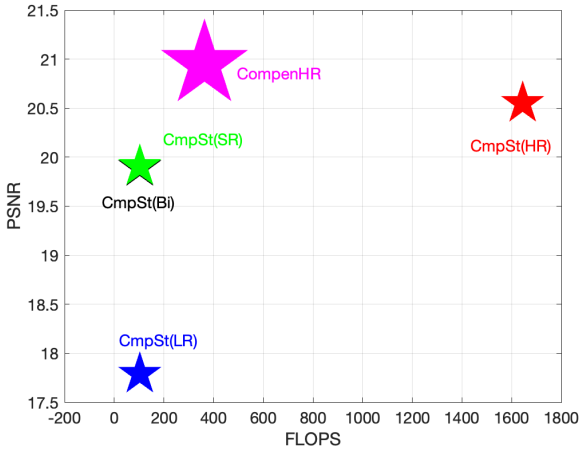

Figure 1: Comparison of state-of-the-art end-to-end full compensation algorithms. FLOPS are calculated on an Nvidia GeForce 1080 GPU with input size $1024 \times 1024 \times 3$. The star sizes are proportional to the number of parameters. The proposed CompenHR (the magenta star) achieves the highest PSNR with a mediate FLOPS. Note that CmpSt(Bi) and CmpSt(SR) are highly overlapped.

By taking into account the memory and time limitations, this paper builds an efficient end-to-end trainable framework named CompenHR for full high-resolution projector compensation. After reformulating the problem by integrating a sampling scheme, we design efficient subnets to model compensation functions. For geometric correction, a novel subnet named GANet (short for Attention-based Geometry Correction Network) is utilized to warp high-resolution input with corrected geometry. We design a grid refinement network to improve the accuracy of sampling grid estimation. Then, for photometric compensation, an efficient subnet named PANet (short for Attention-based Photometric Compensation Network) is exploited to recover the high-resolution images. We employ shuffle/unshuffle [60], instead of traditional downsample/upsample, to improve PANet and enable it to be trained more efficiently with a small amount of information loss. Moreover, we integrate the attention mechanism into both subnets and thus allow CompenHR to extract more important features from the high-resolution input. In addition, due to the lack of high-resolution compensation datasets for evaluation, we construct a dataset with 20 different projector-camera system setups. In experiments, the proposed CompenHR is clearly more efficient than state-of-the-art methods. Our contributions can be summarized as follows:

- We reformulate the full compensation problem for high-resolution projectors and propose a memory and time efficient solution named CompenHR.
- We design an efficient sampling grid refinement subnet for geometric correction, owing to which our CompenHR can achieve more accurate image warping than the state-of-the-art methods.

*e-mail: yxwang@hdu.edu.cn
†e-mail: hling@cs.stonybrook.edu
‡e-mail: bhuang@swu.edu.cn. Corresponding author

- Instead of simple downsampling/upsampling, we apply novel pixel unshuffle/shuffle operations in photometric compensation. Such a design not only avoids information loss but also improves network training and inference efficiency. Moreover, the pixel attention mechanism is integrated into both subnets to focus on key features, which further improves our CompenHR performance.
- A real high-resolution projector full compensation benchmark dataset with 25 setups is constructed and is expected to facilitate future work in this direction. In addition, a synthetic high-resolution dataset with 100 setups is proposed to pre-train models.

2 Related works

2.1 Compensation methods

Projector compensation is an important task for spatial augmented reality, and it has been studied extensively. Existing methods can be divided into three categories: geometry correction, photometric compensation, and full compensation. Detailed reviews can be found in [6, 17].

2.1.1 Geometric correction

For conventional applications, where projection surfaces are planar or multi-planar, traditional methods estimate geometric relations between the camera, the projector, and the projection surface. Projections can be simply corrected by homographies [54, 57]. However, curved surfaces increase the intricacy of geometric correction in many applications.

A surge of work estimates the pixel mappings between the projector input and camera-captured images using structured light [9, 40, 56, 63, 68]. These methods project landmarks onto the surface and capture them with a synchronized camera. Then the 3D geometry of the surface is reconstructed given the pixel mappings and the geometric relationships between the cameras, the projectors, and the surfaces. To reduce the computational complexity, Boroomand *et al*. [9] propose a geometric correction method based on local surface saliency that selects a small set of points rather than dense samples. Tardif *et al*. [63] decompose the mapping function from the camera to the projector into two orientations and determine its parameters by the correspondence of each pixel without surface reconstruction, then construct the corrected image by inverse mapping. Tehrani *et al*. [64] study an automatic method to estimate all device parameters and the surface geometry for a multi-projector system without prior calibration. Particularly, some efforts track the dynamic non-planar surface by marking patterns with invariant topologies [20,46]. Narita *et al*. [46] design fiducial markers that consist of four types of dot clusters, and track non-rigid surfaces by identifying these dot cluster IDs in real-time.

2.1.2 Photometric compensation

Photometric compensation aims to cancel the photometric distortion caused by the textured projection surface and the radiometric response functions, with the assumption that captured images have already been geometrically corrected. Previous methods estimate the color transformation by 1-to-1 mapping from the camera to the projector pixels. Nayar *et al*. [47] define the mapping function with a 3×3 color mixing matrix and estimate it using the correspondence between the captured image and the projected image. On this basis, Grossberg *et al*. [16] reduce the number of calibration patterns to six. Grundhöfer and Iwai. [18, 19] propose a method for an uncalibrated projector and camera system. The compensation process is modeled by a non-linear color mapping function that is defined by a per-pixel thin plate spline interpolation. Considering the pixel redundancies of surface reflectance and the input coherence of the transfer function, Li *et al*. [39] employ sparse sampling and multidimensional interpolation techniques to improve compensation efficiency.

However, the limitation of dynamic ranges and gamuts of the projector and camera system results in clipping artifacts in compensation images. To address this issue, some studies take human vision system properties into consideration. For instance, Wang *et al*. [66] employ the perceptually-based physical error metric, which incorporates the threshold sensitivity, contrast sensitivity, and visual mask to minimize achromatic artifacts in compensation images. Huang *et al*. [31] adjust the brightness and hue of the image by manipulating the reference white of the CIECAM02 Color Appearance Model according to the anchoring property. Pjanic *et al*. [52] propose an adaptive color gamut acquisition to generate a color-prediction model, and then optimize the framework in the RLab color space. Akiyama *et al*. [2] generate the compensation image by minimizing the perceptual distance between its projection and the desired image.

Besides, for a dynamic environment, Fujii *et al*. [15] present an adaptive photometric model under the assumption that the global light is approximately unchanged. In their method, parameters are first estimated by projecting four uniform calibration images, and then the surface reflectance matrix is updated using the error between the captured and desired images when the surface reflectance change exceeds the threshold. Bokaris *et al*. [7, 8] generate images using a linear transformation matrix for dynamic surfaces with one-frame delay. Hashimoto *et al*. [22] estimate the offset of the adjacent moment to update the inter-pixel correspondence and optimize the current reflectance using the present and the sum of past correspondence. Considering the effect of inter-pixel coupling, Shih *et al*. [61] calculate the gamma function and the inter-pixel coupling matrix using two constant grayscale patterns and a ramp grayscale one in the initial calibration, then estimate the dynamic reflectance using the projected image as calibration patterns.

Inspired by the successful application of deep learning to image-to-image translation tasks, Huang *et al*. [27] explore an end-to-end photometric compensation method that learns the inverse mapping from the camera image to the projector image using convolution neural networks and achieves outstanding performance in static projector-camera systems. For white diffuse surfaces, Kageyama *et al*. [33] propose an effective deblurring technique using a convolutional neural network for dynamic projection mapping scenarios. It employs an extractor to estimate defocus blur and luminance attenuation maps and then feeds them to a generator to compute compensation images.

2.1.3 Full compensation

Full compensation techniques perform geometry correction and photometric compensation jointly. Park *et al*. [49] present spatial and temporal encoding techniques that compensate images via embedding patterns. In temporal encoding, pattern images for geometric and radiometric calibration are projected and embedded separately; in spatial encoding, a single pattern that incorporates the information for both geometric and radiometric calibration is designed for simultaneous compensation. Shahpaski *et al*. [58] also design a special projected pattern using squares with mixed blue and red colors for geometric and radiometric calibration. They project this special pattern onto a printed pattern with a standard checkerboard. Benefiting from this design, printed and projected corners are able to be detected from the blue channel and the red channel of captured images respectively using automatic checkerboard detectors.

Recently, Huang *et al*. [26, 29] reformulate the physical process of full compensation and learn the geometric correction and photometric compensation functions using deep neural networks. Park *et al*. [50] simulate the full projection process under virtual light and optimize the compensation image using differentiable rendering.

2.2 Our method

Our method, named CompenHR, belongs to the category of full compensation and is inspired by CompenNeSt++ [29]. While Com-

penNeSt++ has achieved promising performance on low-resolution setups, its memory and training time grow dramatically with the increase of image sizes, making it impractical to compensate for high-resolution inputs. To address this issue, we reformulate the full compensation process for high-resolution projectors and propose to reduce the feature map sizes in the photometric compensation module. After that, we design networks by combining a variety of effective schemes to further improve accuracy. For geometric correction, an attention-based network is designed to refine the sampling grid, produces accurate image warping; for photometric compensation, novel sampling operations are introduced to rearrange images and feature maps. Furthermore, attention mechanisms are employed to preserve key features from images and their linear transformations. Benefiting from these schemes, our CompenHR shows great advantages in memory and time efficiency, with even slightly improved projection quality.

2.3 Attention Mechanism

The attention mechanism used in our CompenHR is inspired by its popularity in computer vision. In the following, we discuss some most related works. A pioneer channel-wise attention mechanism is the squeeze-and-excitation (SE) module [25], which emphasizes the channels with key information of feature maps. On this basis, Hui *et al*. [32] construct the contrast-aware channel attention block by replacing the pooling with a contrast operation. The Squeeze-and-Attention (SA) module [74] replaces the full convolutional layers in SE with the pooling and upsampling operations. Zhang *et al*. [72] design the residual channel attention network (RCAN) using residual channel attention blocks. Dai *et al*. [12] propose the second-order attention network (SAN) by considering the high-order channel feature correlations.

Additionally, a surge of methods incorporate both channel-wise attention and spatial attention. Features in Long *et al.* [10] are weighted by the cascaded channel-wise attention and spatial attention modules. Woo *et al*. [69] arrange the channel-wise and spatial attention modules in parallel and sequentially respectively. Liu *et al*. [41] propose the enhanced spatial attention (ESA) blocks which aggregate local features into more representative features. Muqeet *et al*. [45] make ESA blocks more efficient by employing dilated convolutions. Zhao *et al*. [73] explores an effective pixel attention scheme that learns attention coefficients for all pixels.

3 Deep High-resolution Projector Compensation

3.1 Problem formulation

3.1.1 Projector compensation

Our full projector compensation system consists of a pair of uncalibrated high-resolution projector and camera as well as a fixed non-planar textured surface. Let the input image be $\boldsymbol{x}_h$ (h stands for high-resolution), and the function that geometrically warps a high-resolution input image to the camera view be $\mathcal{T}$, and the photometric function that transforms the high-resolution warped image to the camera-captured image be $\mathcal{F}$, then the image $\tilde{\boldsymbol{x}}_h$ captured by camera[1] can be formulated as:

$$\tilde{\boldsymbol{x}}_h = \mathcal{T}(\mathcal{F}(\boldsymbol{x}_h; \boldsymbol{l}, \boldsymbol{s})) \tag{1}$$

where $\boldsymbol{l}$ stands for the environment lighting and $\boldsymbol{s}$ stands for the surface reflection parameters.

The purpose of full projector compensation is to find the high-resolution compensation image $\boldsymbol{x}_h^*$, so that the camera-captured projection $\tilde{\boldsymbol{x}}_h^*$ is close to the ideal viewer-perceived image $\boldsymbol{x}_h'$:

$$\tilde{\boldsymbol{x}}_h^* = \mathcal{T}(\mathcal{F}(\boldsymbol{x}_h^*; \boldsymbol{l}, \boldsymbol{s})) \approx \boldsymbol{x}_h' \tag{2}$$

[1] Following [29], we use $\tilde{}$ for the camera-captured image.

We assume that $\boldsymbol{s}$ and $\boldsymbol{l}$ are implicitly captured by the camera-captured surface image $\tilde{\boldsymbol{s}}$, then the compensation process can be formulated as:

$$\boldsymbol{x}_h^* = \mathcal{F}^\dagger(\mathcal{T}^{-1}(\boldsymbol{x}_h'); \mathcal{T}^{-1}(\tilde{\boldsymbol{s}})) \tag{3}$$

3.1.2 Compensation with a sampling scheme

For high-resolution setups, directly learning Equ. (3) using deep neural networks is impractical, due to the high memory consumption and training time. To address this issue, we propose a more memory and time-efficient method with a novel sampling scheme below.

Let the low-resolution version of $\boldsymbol{x}_h$ be $\boldsymbol{x}_l$ and plug it into Equ. (3), projector compensation for low-resolution input can be formulated as:

$$\boldsymbol{x}_l^* = \mathcal{F}^\dagger(\mathcal{T}^{-1}(\boldsymbol{x}_l'); \mathcal{T}^{-1}(\tilde{\boldsymbol{s}}_l)) \tag{4}$$

Clearly, $\boldsymbol{x}_l'$ can be easily obtained by sampling $\boldsymbol{x}_h'$. Define $\downarrow$ and $\uparrow$ as downsampling and upsampling operations, respectively, and let $k \in \{1, 2, 3, \ldots, M\}$ be the scale factor, then, according to Equ. (3) $\boldsymbol{x}_h^*$ is given by:

$$\boldsymbol{x}_h^* = (\mathcal{F}^\dagger(\mathcal{T}^{-1}(\boldsymbol{x}_h' \downarrow_k); \mathcal{T}^{-1}(\tilde{\boldsymbol{s}}_h \downarrow_k))) \uparrow_k \tag{5}$$

where $\downarrow_k$ reduces the dimension of the image by $1/k$. This allows us to perform full compensation on the low-resolution images $\boldsymbol{x}_l'$ and $\tilde{\boldsymbol{s}}_l$ rather than $\boldsymbol{x}_h'$ and $\tilde{\boldsymbol{s}}_h$. Finally, the compensated high-resolution image $\boldsymbol{x}_h^*$ is reconstructed from $\boldsymbol{x}_l^*$ by an upsampling operation $\uparrow_k$. However, reconstructing high-resolution images from low-resolution ones is an ill-posed problem [37]. Thus, to preserve more information from $\boldsymbol{x}_h'$ and $\tilde{\boldsymbol{s}}_h$, we employ pixel unshuffle $\mathcal{D}_k$ and pixel shuffle $\mathcal{U}_k$ operations instead of $\downarrow_k$ and $\uparrow_k$, and Equ. (3) becomes

$$\boldsymbol{x}_h^* = \mathcal{U}_k\Big(\mathcal{F}^\dagger\big(\mathcal{T}^{-1}(\mathcal{D}_k(\boldsymbol{x}_h')); \mathcal{T}^{-1}(\mathcal{D}_k(\tilde{\boldsymbol{s}}_h))\big)\Big) \tag{6}$$

Note that $\mathcal{T}$ performs image warping and the most intensive computation is performed in photometric compensation, thus we only need to perform pixel shuffle on the geometrically corrected image, therefore we swap the positions of $\mathcal{D}_k$ and $\mathcal{T}^{-1}$.

$$\boldsymbol{x}_h^* = \mathcal{U}_k\Big(\mathcal{F}^\dagger\big(\mathcal{D}_k(\mathcal{T}^{-1}(\boldsymbol{x}_h')); \mathcal{D}_k(\mathcal{T}^{-1}(\tilde{\boldsymbol{s}}_h))\big)\Big) \tag{7}$$

We model the above equation using a deep neural network named CompenHR $\pi_{\boldsymbol{\theta}}^\dagger(\cdot,\cdot) \equiv \mathcal{U}_k\Big(\mathcal{F}^\dagger\big(\mathcal{D}_k(\mathcal{T}^{-1}(\cdot)); \mathcal{D}_k(\mathcal{T}^{-1}(\cdot))\big)\Big)$ for conciseness, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_\mathcal{F}, \boldsymbol{\theta}_\mathcal{T}\}$ are CompenHR's learnable parameters. Clearly, it can be trained using image pairs like $\{\boldsymbol{x}_{h,i}^*, \boldsymbol{x}_{h,i}'\}$ and a captured surface image $\tilde{\boldsymbol{s}}_h$. However, the ground truth of $\boldsymbol{x}_h^*$ is hard to obtain. Therefore, following [29] we generate a surrogate training set $\mathcal{X} = \{(\tilde{\boldsymbol{x}}_{h,i}, \boldsymbol{x}_{h,i})\}_{i=1}^N$ by projecting the sampling images $\boldsymbol{x}_{h,i}$ and capturing their projections $\tilde{\boldsymbol{x}}_{h,i}$, then CompenHR can be trained by

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}'} \sum_i \mathcal{L}\big(\hat{\boldsymbol{x}}_{h,i} = \pi_{\boldsymbol{\theta}'}^\dagger(\tilde{\boldsymbol{x}}_{h,i}; \tilde{\boldsymbol{s}}_h),\ \boldsymbol{x}_{h,i}\big) \tag{8}$$

In our approach, we define the loss function $\mathcal{L}$ using a combination of pixel-wise l_1, l_2 and structural similarity (SSIM) [67]:

$$\mathcal{L} = \mathcal{L}_{l_1} + \mathcal{L}_{l_2} + \mathcal{L}_{\text{ssim}} \tag{9}$$

3.2 Network design

Based on Equ. (7), our CompenHR integrates two subnets **GANet** and **PANet**, which model $\mathcal{T}^{-1}$ and the combination of $\mathcal{U}_k$, $\mathcal{F}^\dagger$ and $\mathcal{D}_k$, respectively. The architecture is shown in Fig. 2(a). It takes a surface image $\tilde{\boldsymbol{s}}_h$ and some captured sampling images $\tilde{\boldsymbol{x}}_{h,i}$ of resolution 1920×1080 as input, and then generates the inferred projector input/compensation images with a resolution of 1024×1024. Next, we will introduce the subnets in detail.

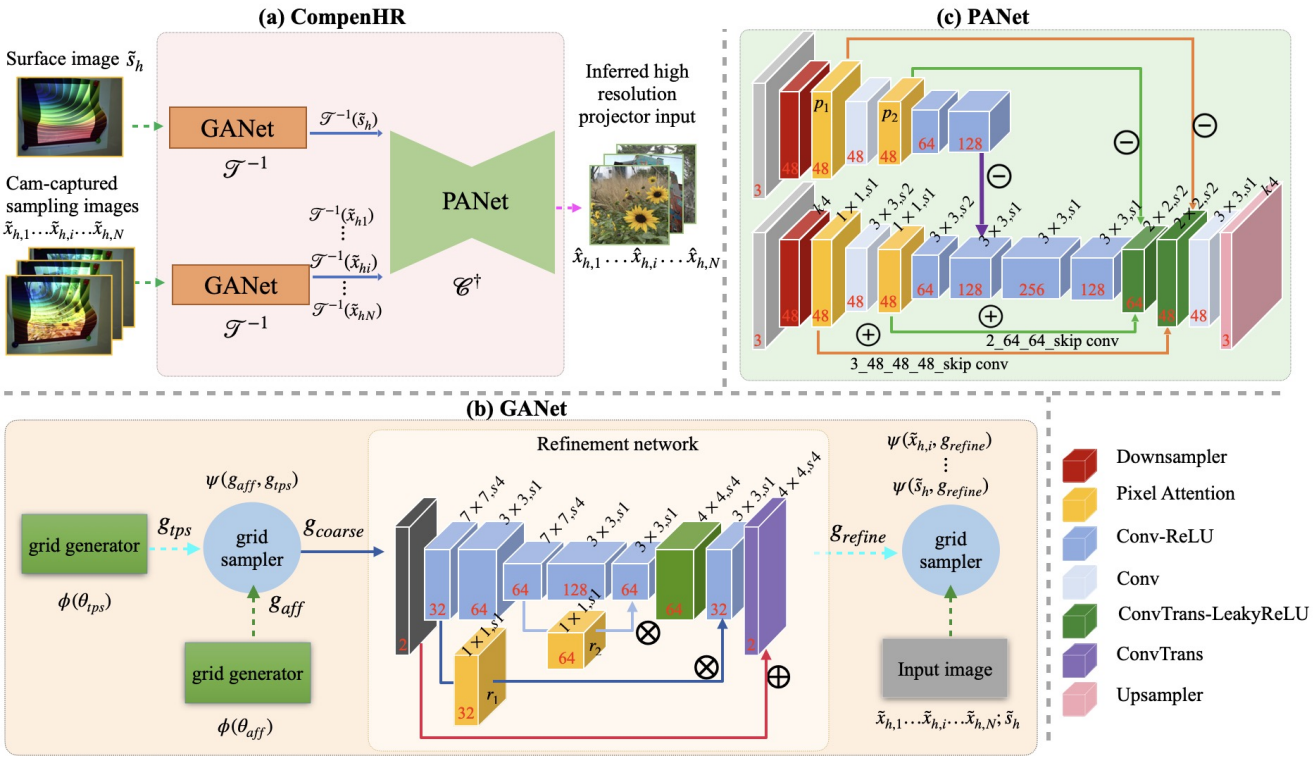

Figure 2: An overview of our **CompenHR**. (a) During training, **CompenHR** takes the captured surface image $\tilde{\boldsymbol{s}}$ and sampling images $\tilde{\boldsymbol{x}}_{h,1},\dots,\tilde{\boldsymbol{x}}_{h,i},\dots,\tilde{\boldsymbol{x}}_{h,N}$ as input, and outputs the inferred high-resolution projector input. It consists of two subnets: **GANet** and **PANet**. (b) **GANet** aims to warp the input camera-captured image to the projector's canonical frontal view. It uses a coarse-to-fine architecture that integrates two grid generators, two grid samplers, and a novel refinement network. (c) **PANet** is applied to compensate for the warped image. It incorporates a siamese encoder with a downsampler and a decoder with an upsampler.

3.2.1 GANet

Our GANet is inspired by WarpingNet [29], a coarse-to-fine architecture for geometric correction. As shown in Fig. 2(b), GANet consists of two grid generators, two grid samplers, and a grid refinement network. Let $\boldsymbol{\theta}_{\text{aff}} \in \mathcal{R}^{2\times 3}$ be the learnable parameters of the affine matrix used to roughly warp $\tilde{\boldsymbol{x}}_h$ and $\tilde{\boldsymbol{s}}_h$ to the front view, $\boldsymbol{\theta}_{\text{tps}}$ be the learnable parameters of thin plate spline (TPS) [13] with five control points used to roughly model the nonlinear warping from the affine warped image to the desired view. GANet employs grid generators $\phi(\boldsymbol{\theta}_{\text{aff}})$ and $\phi(\boldsymbol{\theta}_{\text{tps}})$ to generate affine grid $\boldsymbol{g}_{\text{aff}}$ and TPS grid $\boldsymbol{g}_{\text{tps}}$, and then injects them into the first grid sampler $\psi(\boldsymbol{g}_{\text{aff}}, \boldsymbol{g}_{\text{tps}})$ that samples 2D coordinates using bilinear interpolations. This process can be summarized as:

$$\boldsymbol{g}_{\text{coarse}} = \psi(\phi(\boldsymbol{\theta}_{\text{aff}}), \phi(\boldsymbol{\theta}_{\text{tps}})) \tag{10}$$

Then, we design a neural network $\mathcal{W}$ to further refine the coarse grid $\boldsymbol{g}_{\text{coarse}}$:

$$\boldsymbol{g}_{\text{refine}} = \mathcal{W}(\boldsymbol{g}_{\text{coarse}}) \tag{11}$$

The refinement network contains six convolutional layers followed by ReLU activation and two transpose convolutional layers. To extract useful information from a large input coarse grid efficiently, we use the first and third convolutional layers to downsample large-scale feature maps, and others to extract multi-level features. Then we place two transposed convolutional layers to upsample feature maps and generate the refined output. The detailed parameters are listed in Fig. 2(b). In addition, we employ the attention module that consists of a 1×1 convolutional layer followed by sigmoid activation for efficient feature extraction. This strategy brings better performance on the geometric correction compared with [29].

After refining the grid, the second grid sampler is used for warping the input image using the finer sampling grid $\boldsymbol{g}_{\text{refine}}$.

$$\mathcal{T}^{-1}(\tilde{\boldsymbol{x}}_h) = \psi(\tilde{\boldsymbol{x}}_h, \boldsymbol{g}_{\text{refine}}) \tag{12}$$

3.2.2 PANet

In PANet, we model $\mathcal{F}^{\dagger}$ with a combination of an encoder and a decoder, and model the downsample/upsample-like operations $\mathcal{U}_k/\mathcal{D}_k$ with shuffle/unshuffle operations.

As shown in Fig. 2(c), to reduce computation cost and memory usage, PANet employs a pixel unshuffle operation $\mathcal{D}_k$ instead of spatial bilinear interpolation. It reshapes the input $\tilde{\boldsymbol{x}}_h \in \mathcal{R}^{1024\times 1024\times 3}$ and $\tilde{\boldsymbol{s}}_h \in \mathcal{R}^{1024\times 1024\times 3}$ to the first feature maps $\tilde{\boldsymbol{M}}_x^0 \in \mathcal{R}^{256\times 256\times 48}$ and $\tilde{\boldsymbol{M}}_s^0 \in \mathcal{R}^{256\times 256\times 48}$, without losing pixel information. Benefiting from it, useful information from the original high-resolution image can be preserved for extracting subsequent features.

The photometric compensation function $\mathcal{F}^{\dagger}$ is modeled by an encoder and a decoder. The encoder is a siamese network with shared weights. Each branch stacks three convolutional layers, two of which are followed by ReLU activation. The decoder extracts and upsamples multi-level feature maps from the difference between surface feature maps and captured image feature maps. It consists of three convolutional layers and two transposed convolutional layers. The detailed parameters are noted in Fig. 2(c). Besides, two skip convolution connections are used to capture interaction among multi-level information. The first skip convolution connection (yellow line in Fig. 2(c)) consists of a 1×1 convolutional layer and two 3×3 convolutional layers. The second skip convolution connection (green line in Fig. 2(c)) consists of a 1×1 convolutional layer and a 3×3 convolutional layer. The stride of all convolutional layers is 1.

After generating the compensated feature maps, we employ a pixel shuffle operation $\mathcal{U}_k$ to recover the high-resolution image. It reshapes the multi-channel output of decoder $\tilde{\boldsymbol{M}}_n \in \mathcal{R}^{256\times256\times48}$ to $\tilde{\boldsymbol{x}}_h \in \mathcal{R}^{1024\times1024\times3}$. The use of pixel unshuffle and shuffle operations reduces memory usage and time computation, but it may also lead to lower precision. To alleviate this issue, we place two pixel attention modules [73] (the yellow box in Fig. 2(c)) after the unshuffle layer and the first convolutional layer to preserve more important information from the reshaped image and the low-level feature maps. This pixel attention module also consists of a 1×1 convolutional layer followed by sigmoid activation. Furthermore, it operates on the input directly by using a skip connection. Denote the input feature map as $\boldsymbol{M} \in \mathcal{R}^{C\times H\times W}$, a 1×1 convolution operation as $\mathcal{C}$, the Sigmoid function as σ, respectively, then the output of the pixel attention layer is given by:

$$\boldsymbol{M}' = \sigma(\mathcal{C}(\boldsymbol{M})) \otimes \boldsymbol{M} \tag{13}$$

where $\otimes$ is the element-wise multiplication. Owing to the unshuffle/shuffle operations and the attention mechanism, the memory and time efficiency of PANet is significantly improved with little accuracy degradation.

3.3 Training details

We implement CompenHR using PyTorch [51] and optimize parameters using Adam optimizer [36]. The model is trained on an Nvidia GeForce 1080 GPU with 2000 iterations.

For CompenHR, the initial learning rate is set to 10^{-3} and is decayed by a factor of 5 for every 1500 iteration. For parameter initialization, the weights of the refinement network in GANet are initialized using a normal distribution with a mean of 0 and a standard deviation of 1; the weights of PANet are initialized using He's method [23]. The batch size is set to 4 for all experiments.

4 Benchmark

To evaluate the compensation methods for high-resolution projectors, we build a benchmark dataset with high-resolution image pairs following [29].

4.1 System configuration

Our projector compensation system consists of a Sony α6400 camera and an EPSON CB-X05 projector whose resolutions are set to 1920×1080 and 1024×768 respectively. An Elgato Cam Link 4K video capture card is used to capture the camera frames.

The projector is placed about 1 meter in front of the surface, and the camera is placed within a range of 0.3 to 1 meter around the projector. In each setup, the camera settings such as exposure, focus, and white balance are adjusted manually based on the ambient light and surface material, and fixed during each setup data capturing.

4.2 Datasets

4.2.1 Real data

To the best of our knowledge, there is no public high-resolution dataset for quantitative evaluation. Thus we construct a real dataset with 25 setups, and 5 of them have specular surfaces. For each setup, at least one of the ambient lighting, camera parameters, non-planar projector surface, etc. is different.

We collect $N = 700$ colorful high-resolution (1920×1080 or higher) images taken in real life and resize them to 1024×1024 as projector input. During data collection, all these images and a gray image are projected to the projection surface and captured by the camera. Thus, the sets consist of image pairs $\mathcal{X}_{\text{train}} = \{(\tilde{\boldsymbol{x}}_{h,i}, \boldsymbol{x}_{h,i}) | i = 1 \ldots N_{\text{train}}\}$, $\mathcal{Y} = \{(\tilde{\boldsymbol{y}}_{h,i}, \boldsymbol{y}_{h,i}) | i = 1 \ldots N_{\text{test}}\}$ and the uncompensated surface $\tilde{\boldsymbol{s}}_h$. Among them, $N_{\text{train}} = 500$ image pairs are for training and $N_{\text{test}} = 200$ for testing. Fig. 3 shows three image pairs with different setups and surfaces.

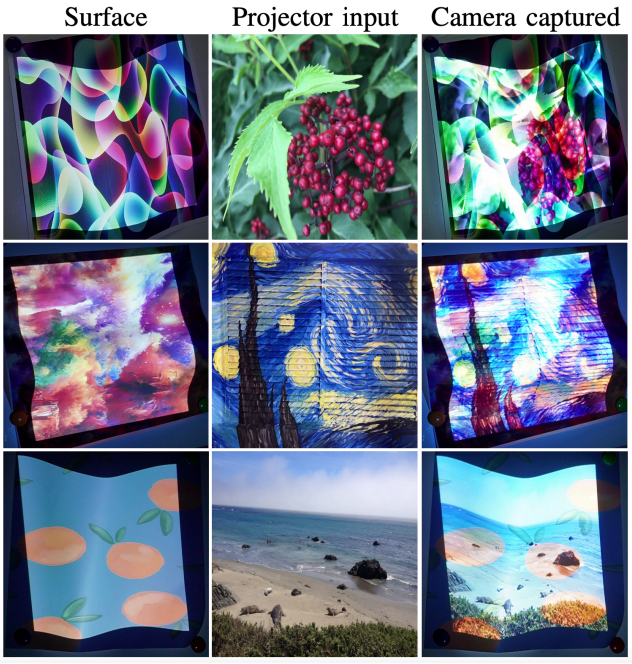

Figure 3: Samples of the real dataset. From left to right: textured surfaces, projector input images, and camera-captured projections.

4.2.2 Synthetic data

Following [29], to improve the practicability of our full compensation method, we build a synthetic high-resolution dataset to pre-training the photometric compensation module by rendering 100 setups with different projector-camera-surface poses, materials, exposures, and lightings in Blender [11].We use 100 surface patterns provided by [29] and 500 projected sampling images in this synthetic dataset are selected from DIV2K training dataset [1] and resized to 1024×1024.

4.3 Metrics

We use the surrogate evaluation protocol presented in [29] for quantitative comparisons, and four metrics are used: PSNR, RMSE, SSIM, and ΔE (CIE standard for perceptual color differences [59]).

5 Experiments

In experiments, the proposed CompenHR is trained on our high-resolution full compensation dataset with 500 images and tested with 200 images in each setup, and the final results are averaged over K = 20 setups.

5.1 Comparison with state-of-the-arts

We compare our CompenHR with an end-to-end trainable method CompenNeSt++. The original CompenNeSt++ is proposed as the solution for low-resolution input, we trained it with both high-resolution and low-resolution image pairs, and we name the two baselines *CmpSt(HR)* and *CmpSt(LR)*, respectively. Besides, an intuitive way to reconstruct high-resolution images is using super-resolution. Thus, we also use CompenNeSt++ to generate low-resolution compensation images and then reconstruct the corresponding high-resolution images by bicubic interpolation, and we name this method *CmpSt(Bi)*. We also combine CompenNeSt++ with the state-of-the-art deep learning-based super-resolution method named Residual Local Feature Network (RLFN) [37]. We pre-train it with DIV2K training dataset [1] and use it as an upsampler without fine-tuning, we name this baseline as *CmpSt(SR)*.

We train *CompenHR* and *CmpSt(HR)* using high-resolution image pairs (1024×1024) and then train *CmpSt(LR)* using the low-resolution image pairs (256×256). In testing, all methods' input and output resolutions are set to 1024×1024, but the intermediate resolutions of the two two-step methods (*CmpSt(Bi)* and *CmpSt(SR)*) are 256×256. The quantitative comparisons are shown in Tab. 1. To verify the efficiency of algorithms, in Tab. 3 we further compare the time and memory consumption of *CmpSt(LR)*, *CmpSt(HR)*, and our *CompenHR* during training. We ignore the consumption during testing since it is negligible compared with the training phase. *CmpSt(Bi)* and *CmpSt(SR)* share the same trained module *CmpSt(LR)*, thus they have the same parameters, FLOPS, training time, and memory consumption.

Table 1: Quantitative comparison of full compensation algorithm on image quality. Results are averaged over 20 different setups.

Model	PSNR	RMSE	SSIM	ΔE
CmpSt(HR)	20.5508	0.1628	0.5980	7.6076
CmpSt(LR)	17.7894	0.2250	0.4673	9.8970
CmpSt(Bi)	19.8987	0.1761	0.5495	8.1626
CmpSt(SR)	19.9099	0.1758	0.5498	8.1719
CompenHR	**20.9468**	**0.1554**	**0.6011**	**7.5746**
Uncompensated	11.5984	0.4619	0.2414	21.4257

Table 2: Quantitative comparison of full compensation algorithm on image quality. Results are averaged over 5 different setups with specular highlight surfaces.

Model	PSNR	RMSE	SSIM	ΔE
CmpSt(HR)	20.6376	0.1614	0.6049	7.3914
CmpSt(LR)	18.3643	0.2104	0.4772	9.1279
CmpSt(Bi)	20.1215	0.1710	0.5514	7.8007
CmpSt(SR)	20.1094	0.1712	0.5513	7.8371
CompenHR	**21.0135**	**0.1544**	**0.6094**	**7.3994**
Uncompensated	11.4566	0.4695	0.2327	20.5489

Table 3: Quantitative comparison of full compensation algorithm on the amount of computation. Results are averaged over 20 different setups.

Model	Params(M)	FLOPS(G)	Mem.(M)	Time(s)
CmpSt(HR)	**833,145**	1645.5460	10003	6200.05
CmpSt(LR)	**833,145**	**102.8466**	**1779**	**367.70**
CompenHR	1327,586	364.1476	4927	1484.45

In Tab. 1, Tab. 2 and Tab. 3, comparing our *CompenHR* with *CmpSt(Bi)* and *CmpSt(SR)*, because our *CompenHR* is trained with high-resolution images, it has a higher FLOPS, number of parameters and memory usage in training. But clearly, it also has a better compensation quality than others in testing Besides, compared with *CmpSt(HR)*, which also uses high-resolution images for training, *CompenHR* achieves a little better quality and trains much faster. Benefiting from the usage of shuffle/unshuffle operations, the feature map size is reduced to a quarter of the original sizes so that our *CompenHR* consumes less computation and memory than *CmpSt(HR)*. Fig. 4 shows qualitative comparisons of all methods. For all samples, the color and brightness of *CompenHR* are better than the others, and the results of *CompenHR* and *CmpHR* are sharper than the other three methods. In particular, the left two columns also show that *CompenHR*'s geometric correction is more accurate than the others. These results further demonstrate that preserving the input image resolution has a significant influence on the quality of compensation images. Methods trained with high-resolution images learn more details than low-resolution ones. In particular, these methods can handle slight specular highlights to some extent but do not work well on the area with strong specular reflection.

5.2 Ablation study

In this section, we first validate the proposed CompenHR using different numbers of training datasets, and then considering the major ingredients in CompenHR: (1) a novel GANet with sampling grid refinement; (2) shuffle/unshuffle based sampling operations for both input and output of PANet; (3) pixel attention mechanisms for important feature extraction; (4) the loss function combining l_1, l_2 and *SSIM*, we explore the effectiveness of our sampling grid refinement network, attention mechanism, and loss function.

5.2.1 The effect of the number of training images

To validate the practicability of the proposed method further, we train our network with different numbers of training images and evaluate them on our real dataset with 20 setups. The results are reported in Tab. 4. The image quality of the default CompenHR using 48 training images is much higher than that using 8 training images, and then the image quality improves slightly as the number of training datasets increases.

Furthermore, following [29], we also pre-train CompenHR using the proposed synthetic dataset with 100 setups, and then *fine-tune it using only 8 images with* 1000 *iterations*. The initial learning rate for fine-tuning is set to 10^{-3} and is decayed by a factor of 5 for every 600 iteration. In Tab. 5, the pre-trained CompenHR outperforms the default CompenHR.

Table 4: Quantitative comparisons of CompenHR with different numbers of training datasets. Results are averaged over 20 different setups.

Model(-#Train)	PSNR	RMSE	SSIM	ΔE
CompenHR-8	19.3906	0.1876	0.5155	9.1814
CompenHR-48	20.7544	0.1589	0.5933	7.8020
CompenHR-125	20.7540	0.1589	0.5945	8.0139
CompenHR-250	20.7894	0.1583	0.5987	7.7242
CompenHR-500	**20.9468**	**0.1554**	**0.6011**	**7.5746**

Table 5: Quantitative comparisons between the default CompenHR and the pre-trained CompenHR. Results are averaged over 20 different setups.

Model(-#Train)	PSNR	RMSE	SSIM	ΔE
CompenHR-8	19.3828	0.1888	0.5345	9.3571
CompenHR-pretrain-8	**19.7082**	**0.1800**	**0.5532**	**8.8882**

Table 6: Quantitative comparisons of CompenHR with different sampling grid refinement networks.

Model	PSNR	RMSE	SSIM	ΔE
CmpHR(WPGA)	20.4888	0.1639	0.5743	7.8431
CmpHR w/o r_1, r_2	20.6910	0.1603	0.5854	7.6807
CompenHR	**20.9468**	**0.1554**	**0.6011**	**7.5746**

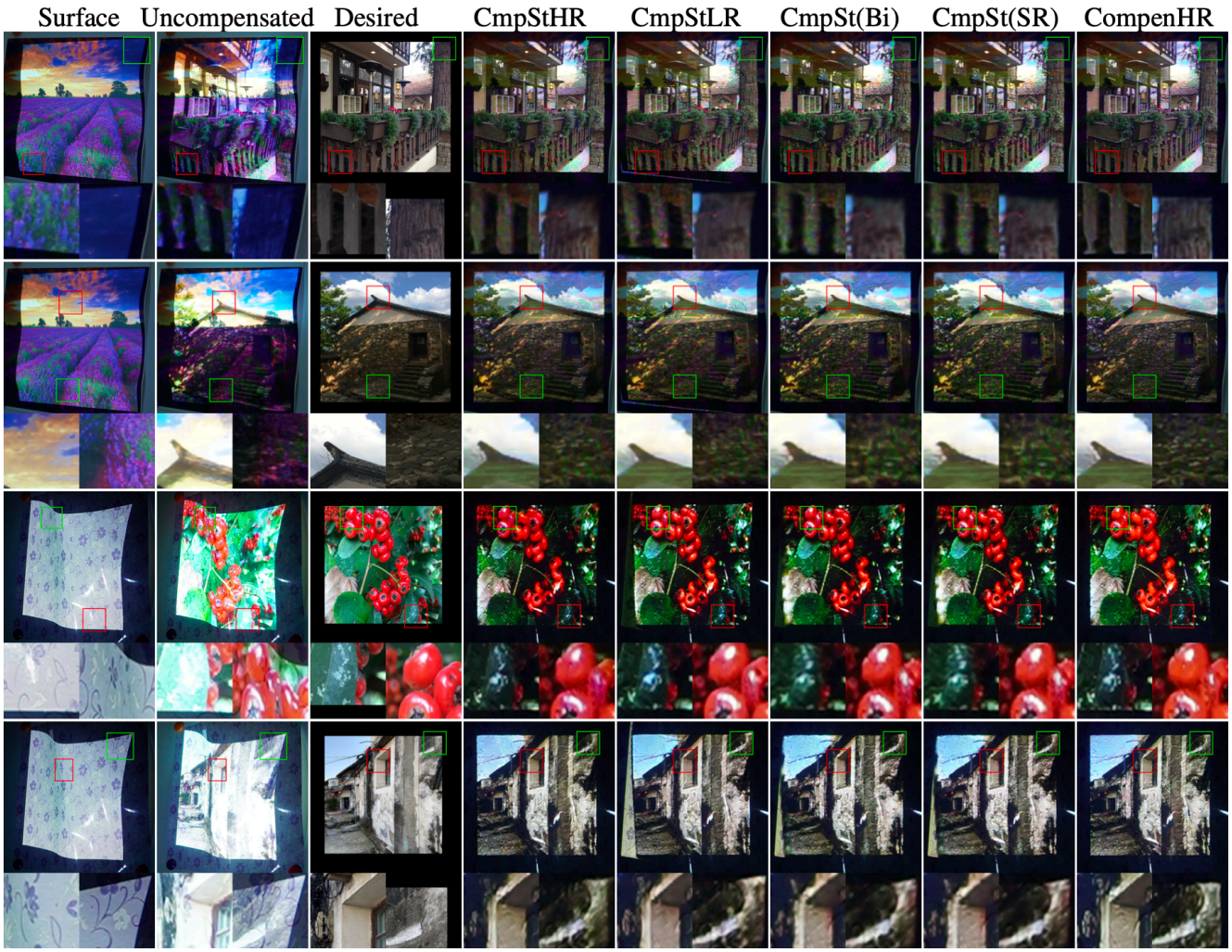

Figure 4: Qualitative comparison on real camera-captured compensation. From top to bottom: surface, uncompensated, desired image, *CmpSt(HR)*, *CmpSt(LR)*, *CmpSt(Bi)*, *CmpSt(SR)*, and our *CompenHR*. See high-resolution figures in the supplementary.

5.2.2 Effectiveness of the refinement network in GANet

To show the effectiveness of GANet, we replace GANet with WarpingNet in [29] and name the compensation model *CmpHR(WPGA)*. As the sampling grid refinement network contains two attention modules (yellow blocks in Fig. 2(b)), to further explore the role of this architecture, we also compare *CompenHR* with the model without r_1 and r_2 (short for *CmpHR w/o* r_1, r_2). The quantitative comparisons in Tab. 6 show that *CompenHR* and *CmpHR w/o* r_1, r_2 outperform *CmpHR(WPGA)* on all metrics. Additionally, the fact that the image quality of *CompenHR* is better than *CmpHR w/o* r_1, r_2 also indicates the effectiveness of the attention modules. In Fig. 5, *CompenHR* generates the sharpest images for all examples. Besides, in the two right-most columns, *CmpHR(WPGA)* cannot work well for the cluttered surface texture, as it can not warp the image correctly.

5.2.3 Effectiveness of pixel attention blocks in PANet

To improve the performance of CompenHR, two attention modules are also introduced to extract key features in PANet. We explore their effectiveness in this section. After removing each pixel attention layer respectively, we build new models named *CmpHR w/o* p_1, *CmpHR w/o* p_2, and *CmpHR w/o* p_1, p_2, respectively. In the comparison experiment, all methods are trained using the proposed full compensation datasets with 2,000 iterations. The quantitative comparisons are listed in Tab. 7 and the qualitative comparisons are shown in Fig. 6.

Table 7: Quantitative comparisons of CompenHR with different pixel attention layers in PANet.

Model	PSNR	RMSE	SSIM	ΔE
CmpHR w/o p_1	20.6688	0.1606	0.5932	7.7296
CmpHR w/o p_2	20.8284	0.1576	0.5953	7.6685
CmpHR w/o p_1, p_2	20.4553	0.1646	0.5915	8.1839
CompenHR	**20.9468**	**0.1554**	**0.6011**	**7.5746**

In the quantitative comparison, their SSIM scores are very close, while PSNR scores gradually increase, and RMSE and ΔE scores decrease with the number of pixel attention layers. In particular, the image quality of *CmpHR w/o* p_2 is slightly better than *CmpHR w/o* p_1. In Fig. 6, the compensation images of *CompenHR* have the closest color and detail to the desired effects. Besides, the detail of images generated by *CompenHR* is close to *CmpHR w/o* p_2, and is slightly better than *CmpHR w/o* p_1, p_2, because the first pixel attention module extracts key features from the reshaped colorful image directly and the second one further extracts key features

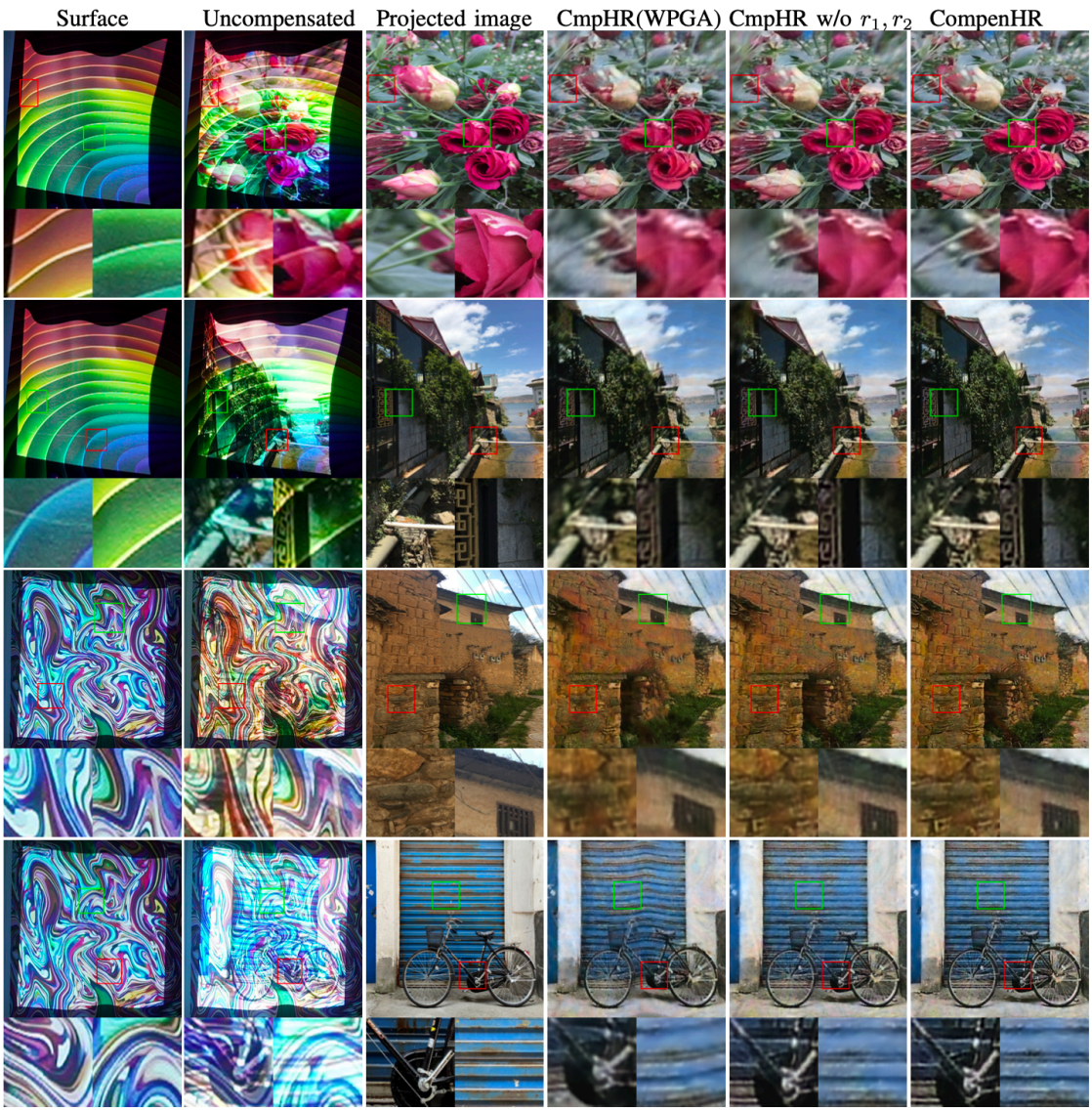

Figure 5: Qualitative comparison of models with different geometric correction methods. From top to bottom: surface, uncompensated, projected image, *CmpHR(WPGA)*, *CmpHR w/o* r_1, r_2 and *CompenHR*. See high-resolution figures in the supplementary.

from its linear transformation results. Benefiting from the attention mechanism, *CompenHR* preserves more image color information in photometric compensation. The experiment demonstrates that the pixel attention mechanism with a few additional parameters can help the model achieve better performance on image color. But adding more attention blocks only brings a small performance improvement. Thus, we finally use two attention blocks in our method.

5.2.4 Comparison of different loss functions

The pixel-wise l_1 and l_2 losses are widely used to penalize the pixel errors in many image reconstruction tasks. In [29], Huang *et al.* verify that *SSIM* loss can be used for image compensation tasks to help recover the structural details. Therefore, we compare the performance of methods that use different combinations of these three loss functions in Tab. 8.

When using three losses separately, l_1 loss achieves the best scores on all metrics. Using l_2 or *SSIM* alone achieves suboptimal results in this task, while loss functions with added *SSIM* loss achieve higher structure similarity, and loss functions with added l_l loss achieve better color quality. More qualitative comparisons are listed in the supplementary material. The results confirm that l_1 loss contributes to compensating image color in this task, while *SSIM* loss tends to recover image structural details. In addition, l_2 loss also helps improve the image quality slightly. As a result, we employ the combination of all three losses and achieve the best performance.

Table 8: Quantitative comparisons of CompenHR with different loss functions.

Loss	PSNR	RMSE	SSIM	ΔE
l_1	20.7247	0.1595	0.5571	7.7462
l_2	20.3679	0.1663	0.5465	8.1671
SSIM	18.9372	0.1964	0.5471	11.3003
$l_1 + l_2$	20.7946	0.1582	0.5555	7.7017
$l_1 + SSIM$	20.8897	0.1564	0.5984	7.6284
$l_2 + SSIM$	20.3312	0.1670	0.5917	8.7418
$l_1 + l_2 + SSIM$	**20.9468**	**0.1554**	**0.6011**	**7.5746**

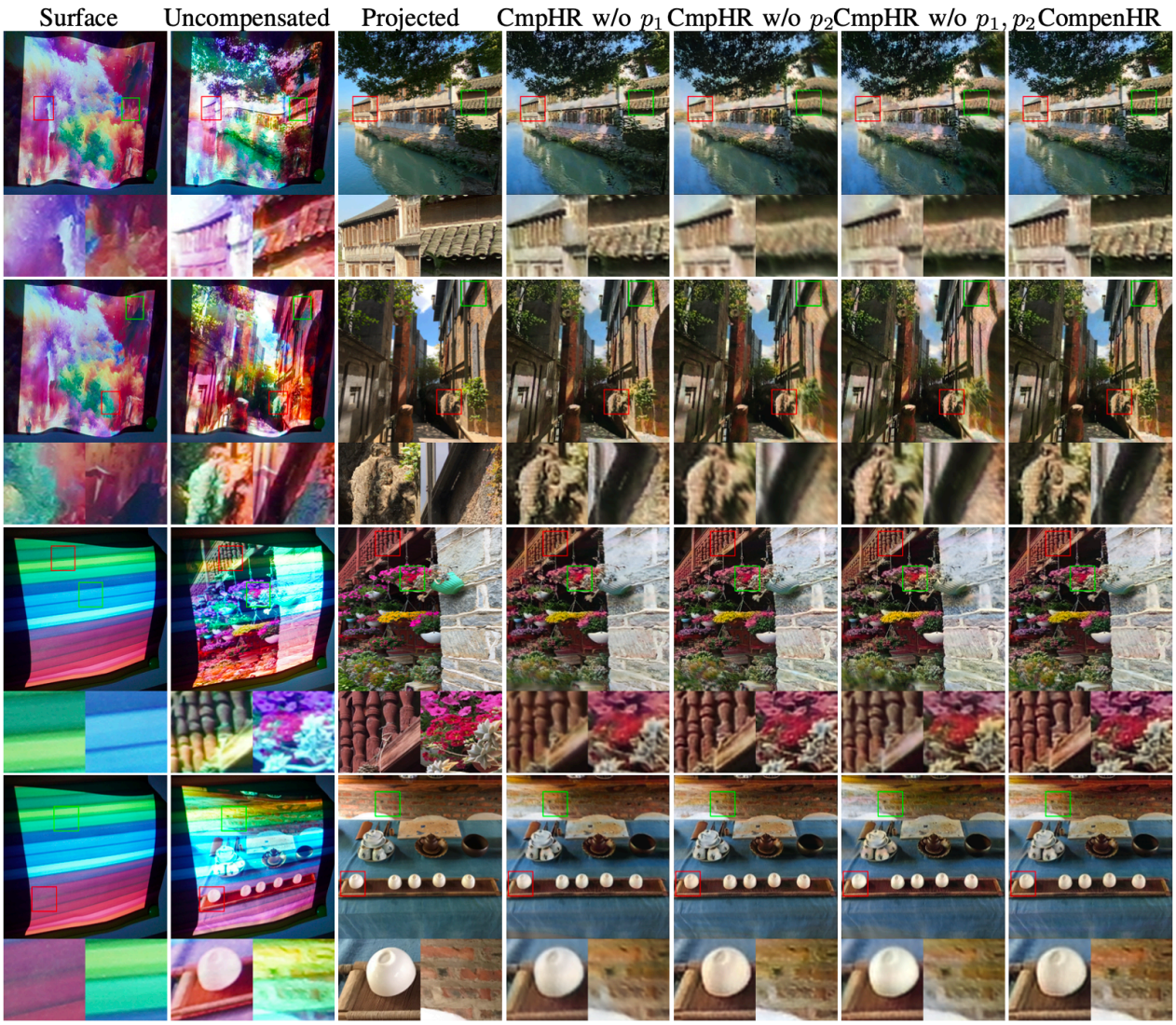

Figure 6: Qualitative comparison of models with the different number of attention layers. From top to bottom: surface, uncompensated, projected image, *CmpHR w/o* p_1, *CmpHR w/o* p_1, *CmpHR w/o* p_1, p_2 and *CompenHR*. See high-resolution figures in supplementary.

6 Discussion

Our method improves the efficiency of the deep learning-based method and achieves competitive performance on the high-resolution compensation task, but it still has some limitations. First, our model is designed for static projector-camera systems, and in future work, we will explore online learning methods for dynamic projector compensation. Second, like [29], our GANet does not work for surfaces with sharp edges and occlusions, and a multi-projector setup may better address this issue. Third, similar to CompenNeSt++ [29], our method can handle slight specular highlights but does not work well on the area with strong specular reflections.

7 Conclusion

In high-resolution full projector compensation, memory usage and time cost increase sharply with the image resolution. This paper proposes an efficient end-to-end solution by first reformulating the full compensation problem by integrating the sampling process, then an attention-based sampling grid refinement network is designed for better geometric correction. Moreover, unshuffle/shuffle operations and pixel attention mechanisms are applied to improve quality and efficiency. Finally, a high-resolution full compensation benchmark dataset is constructed, and experiments demonstrate the advantages of the proposed method.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1122–1131, 2017.

[2] R. Akiyama, T. Fukiage, and S. Nishida. Perceptually-based optimization for radiometric projector compensation. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*, pp. 750–751, 2022.

[3] H. Asayama, D. Iwai, and K. Sato. Fabricating diminishable visual markers for geometric registration in projection mapping. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1091–1102, 2018.

[4] O. Bimber. Multi-projector techniques for real-time visualizations in everyday environments. In *IEEE Virtual Reality Conference*, pp. 320–320, 2006.

[5] O. Bimber, A. Emmerling, and T. Klemmer. Embedded entertainment with smart projectors. *Computer*, 38(1):48–55, 2005.

[6] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. *Computer Graphics Forum*, 27:2219–2245, 2008.

[7] P.-A. Bokaris, M. Gouiffès, C. Jacquemin, and J.-M. Chomaz. Photometric compensation to dynamic surfaces in a projector-camera system. In *European Conference on Computer Vision Workshops*, pp. 283–296, 2014.

[8] P.-A. Bokaris, M. Gouiffès, C. Jacquemin, J.-M. Chomaz, and A. Trémeau. One-frame delay for dynamic photometric compensation in a projector-camera system. In *2015 IEEE International Conference on Image Processing*, pp. 2675–2679, 2015.

[9] A. Boroomand, H. Sekkati, M. Lamm, D. A. Clausi, and A. Wong. Saliency-guided projection geometric correction using a projector-camera system. In *2016 IEEE International Conference on Image Processing*, pp. 2951–2955, 2016.

[10] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6298–6306, 2017.

[11] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022.

[12] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11057–11066, 2019.

[13] G. Donato and S. Belongie. Approximate thin plate spline mappings. In *European Conference on Computer Vision*, pp. 21–31, 2002.

[14] J. Ehnes and M. Hirose. Projected reality - enhancing projected augmentations by dynamically choosing the best among several projection systems. In *IEEE Virtual Reality Conference*, pp. 283–284, 2006.

[15] K. Fujii, M. Grossberg, and S. Nayar. A projector-camera system with real-time photometric adaptation for dynamic environments. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 814–821, 2005.

[16] 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.

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

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

[19] A. Grundhöfer and D. Iwai. Robust, error-tolerant photometric projector compensation. *IEEE Transactions on Image Processing*, 24(12):5086–5099, 2015.

[20] I. Guskov. Efficient tracking of regular patterns on non-rigid geometry. In *2002 International Conference on Pattern Recognition*, vol. 2, pp. 1057–1060, 2002.

[21] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay. Practical methods for geometric and photometric correction of tiled projector. In *2006 Conference on Computer Vision and Pattern Recognition Workshop*, 2006.

[22] N. Hashimoto and K. Yoshimura. Radiometric compensation for non-rigid surfaces by continuously estimating inter-pixel correspondence. *The Visual Computer*, 37(1):175–187, 2021.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

[24] K. Hiratani, D. Iwai, P. Punpongsanon, and K. Sato. Shadowless projector: Suppressing shadows in projection mapping with micro mirror array plate. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces*, pp. 1309–1310, 2019.

[25] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

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

[27] B. Huang and H. Ling. End-to-end projector photometric compensation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6803–6812, 2019.

[28] B. Huang and H. Ling. DeProCams: Simultaneous relighting, compensation and shape reconstruction for projector-camera systems. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2725–2735, 2021.

[29] B. Huang, T. Sun, and H. Ling. End-to-end full projector compensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2953–2967, 2022.

[30] B. Huang, Y. Tang, S. Ozdemir, and H. Ling. A fast and flexible projector-camera calibration system. *IEEE Transactions on Automation Science and Engineering*, 18(3):1049–1063, 2021.

[31] T.-H. Huang, T.-C. Wang, and H. H. Chen. Radiometric compensation of images projected on non-white surfaces by exploiting chromatic adaptation and perceptual anchoring. *IEEE Transactions on Image Processing*, 26(1):147–159, 2017.

[32] Z. Hui, X. Gao, Y. Yang, and X. Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia (ACM MM)*, pp. 2024–2032, 2019.

[33] Y. Kageyama, D. Iwai, and K. Sato. Online projector deblurring using a convolutional neural network. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2223–2233, 2022.

[34] Y. Kemmoku and T. Komuro. AR tabletop interface using a head-mounted projector. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 288–291, 2016.

[35] A. Kenyon, J. van Rosendale, S. Fulcomer, and D. Laidlaw. The design of a retinal resolution fully immersive VR display. In *2014 IEEE Virtual Reality*, pp. 89–90, 2014.

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

[37] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, and L. Fu. Residual local feature network for efficient super-resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 765–775, 2022.

[38] D. M. Krum, S.-H. Kang, T. Phan, L. C. Dukes, and M. Bolas. Head mounted projection for enhanced gaze in social interactions. In *2016 IEEE Virtual Reality*, pp. 209–210, 2016.

[39] 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*, 37(2):365–375, 2018.

[40] H. Lin, L. Nie, and Z. Song. A single-shot structured light means by encoding both color and geometrical features. *Pattern Recognition*, 54:178–189, 2016.

[41] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu. Residual feature aggregation network for image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2356–2365, 2020.

[42] K.-L. Low, A. Ilie, G. Welch, and A. Lastra. Combining head-mounted and projector-based displays for surgical training. In *IEEE Virtual Reality*, pp. 110–117, 2003.

[43] A. Majumder, D.-Q. Lai, and M. A. Tehrani. A multi-projector display system of arbitrary shape, size and resolution. In *2015 IEEE Virtual Reality*, pp. 339–340, 2015.

[44] I. Miyagawa, Y. Sugaya, H. Arai, and M. Morimoto. An iterative compensation approach without linearization of projector responses for multiple-projector system. *IEEE Transactions on Image Processing*, 23(6):2676–2687, 2014.

[45] A. Muqeet, J. Hwang, S. Yang, J. Kang, Y. Kim, and S.-H. Bae. Multi-attention based ultra lightweight image super-resolution. In *European Conference on Computer Vision Workshops*, pp. 103–118, 2020.

[46] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1235–1248, 2017.

[47] S. Nayar, H. Peri, M. Grossberg, and P. Belhumeur. A projection system with radiometric compensation for screen imperfections. In

Proceedings of the IEEE International Conference on Computer Vision Workshop Projector-Camera Syst. (PROCAMS), 2003.

[48] K. Ozacar, T. Hagiwara, J. Huang, K. Takashima, and Y. Kitamura. Coupled-clay: Physical-virtual 3D collaborative interaction environment. In *2015 IEEE Virtual Reality*, pp. 255–256, 2015.

[49] H. Park, M.-H. Lee, B.-K. Seo, J.-I. Park, M.-S. Jeong, T.-S. Park, Y. Lee, and S.-R. Kim. Simultaneous geometric and radiometric adaptation to dynamic surfaces with a mobile projector-camera system. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(1):110–115, 2008.

[50] J. Park, D. Jung, and B. Moon. Projector compensation framework using differentiable rendering. *IEEE Access*, 10:44461–44470, 2022.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[52] P. Pjanic, S. Willi, D. Iwai, and A. Grundhöfer. Seamless multi-projection revisited. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2963–2973, 2018.

[53] P. Punpongsanon, D. Iwai, and K. Sato. Flexeen: Visually manipulating perceived fabric bending stiffness in spatial augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 26(2):1433–1439, 2020.

[54] R. Raskar. Immersive planar display using roughly aligned projectors. In *Proceedings IEEE Virtual Reality*, pp. 109–115, 2000.

[55] R. Raskar and P. Beardsley. A self-correcting projector. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[56] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: geometrically aware and self-configuring projectors. *ACM Transactions on Graphics*, 22(3):809–818, 2003.

[57] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Rendering Techniques 2001*, pp. 89–102, 2001.

[58] M. Shahpaski, L. R. Sapaico, G. Chevassus, and S. Süsstrunk. Simultaneous geometric and radiometric calibration of a projector-camera pair. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3596–3604, 2017.

[59] G. Sharma, W. Wu, and E. N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research and Application*, 30:21–30, 2005.

[60] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.

[61] K.-T. Shih, J.-S. Liu, F. Shyu, and H. H. Chen. Enhancement and speedup of photometric compensation for projectors by reducing inter-pixel coupling and calibration patterns. *IEEE Transactions on Image Processing*, 30:418–430, 2021.

[62] 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 Transactions on Graphics*, 34:1–11, 2015.

[63] J.-P. Tardif, S. Roy, and M. Trudeau. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. In *Fourth International Conference on 3-D Digital Imaging and Modeling*, pp. 217–224, 2003.

[64] 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*, 27(4):2265–2279, 2021.

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

[66] D. Wang, I. Sato, T. Okabe, and Y. Sato. Radiometric compensation in a projector-camera system based properties of human vision system. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2005.

[67] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[68] S. Willi and A. Grundhöfer. Robust geometric self-calibration of generic multi-projector camera systems. In *2017 IEEE International Symposium on Mixed and Augmented Reality*, pp. 42–51, 2017.

[69] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, pp. 3–19, 2018.

[70] T. Yoshida, Y. Hirobe, H. Nii, N. Kawakami, and S. Tachi. Twinkle: Interacting with physical surfaces using handheld projector. In *2010 IEEE Virtual Reality Conference*, pp. 87–90, 2010.

[71] J. Yu, F. Da, and W. Li. Calibration for camera–projector pairs using spheres. *IEEE Transactions on Image Processing*, 30:783–793, 2021.

[72] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, pp. 294–310, 2018.

[73] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong. Efficient image super-resolution using pixel attention. In *European Conference on Computer Vision Workshops*, pp. 56–72, 2020.

[74] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong. Squeeze-and-attention networks for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13062–13071, 2020.