End-to-end Projector Photometric Compensation

Bingyao Huang[1,2,*] Haibin Ling[1,†]
[1]Department of Computer and Information Sciences, Temple University, Philadelphia, PA USA
[2]Meitu HiScene Lab, HiScene Information Technologies, Shanghai, China
{bingyao.huang, hbling}temple.edu

Abstract

Projector photometric compensation aims to modify a projector input image such that it can compensate for disturbance from the appearance of projection surface. In this paper, for the first time, we formulate the compensation problem as an end-to-end learning problem and propose a convolutional neural network, named CompenNet*, to implicitly learn the complex compensation function. CompenNet consists of a UNet-like backbone network and an autoencoder subnet. Such architecture encourages rich multi-level interactions between the camera-captured projection surface image and the input image, and thus captures both photometric and environment information of the projection surface. In addition, the visual details and interaction information are carried to deeper layers along the multi-level skip convolution layers. The architecture is of particular importance for the projector compensation task, for which only a small training dataset is allowed in practice.*

Another contribution we make is a novel evaluation benchmark, which is independent of system setup and thus quantitatively verifiable. Such benchmark is not previously available, to our best knowledge, due to the fact that conventional evaluation requests the hardware system to actually project the final results. Our key idea, motivated from our end-to-end problem formulation, is to use a reasonable surrogate to avoid such projection process so as to be setup-independent. Our method is evaluated carefully on the benchmark, and the results show that our end-to-end learning solution outperforms state-of-the-arts both qualitatively and quantitatively by a significant margin.

1. Introduction

Projectors are widely used in applications such as presentation, cinema, structured light and projection mapping [1, 3, 8, 9, 25, 28, 31, 32, 36]. To ensure high perception quality, existing systems typically request the projection surface (screen) to be white and textureless, under reasonable environment illumination. Such request, however, largely limits applicability of these systems. Projector photometric compensation [1, 3, 25, 28, 31, 32, 36], or simply *compensation* for short, aims to address this issue by modifying a projector input image to compensate for the projection surface as well as associated photometric environment. An example from our solution is illustrated in Fig. 1, where the compensated projection result (e) is clearly more visually pleasant than the uncompensated one (c).

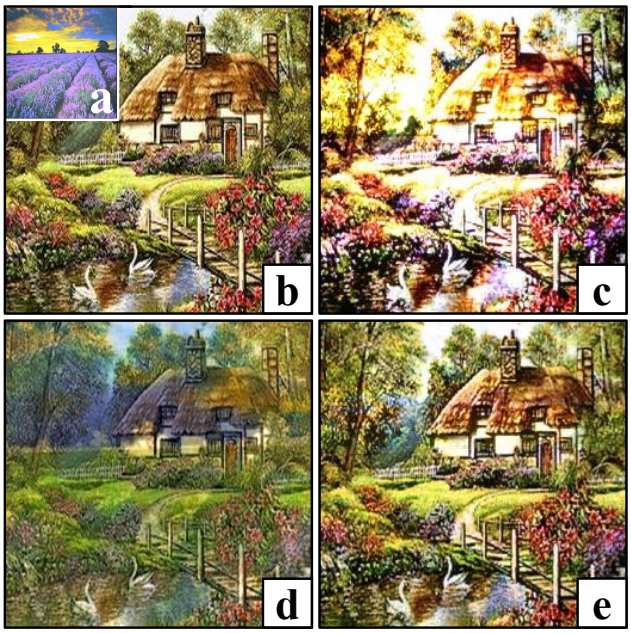


Figure 1: Projector photometric compensation. **(a)** Textured projection surface under normal illumination. **(b)** Input image (desired visual effect). **(c)** Camera-captured uncompensated projection result, *i.e.*, (b) projected onto (a). **(d)** Compensated image by the proposed CompenNet. **(e)** Camera-captured compensated projection result, *i.e.*, (d) projected onto (a). Comparing (c) and (e) we can see clearly improved color and details.

*Work partly done during internship with HiScene.
†Corresponding author.

A typical projector compensation system consists of a camera-projector pair and a projection surface placed at a fixed distance and orientation. Firstly, the projector projects a sequence of sampling input images to the projection surface, then the sampling images are absorbed, reflected or refracted according to the projection surface material. Once the camera captures all the projected sampling images, a composite radiometric transfer function is fitted that maps the input images to the captured images. This function (or its inverse) is then used to infer the compensated image for a new input image. Existing solutions (*e.g.*, [9, 11, 26, 30]) usually model the compensation function explicitly, with various simplification assumptions that allow the parameters to be estimated from samples collected. These assumptions, such as context independence (§2), however, are often violated in practice. Moreover, due to the tremendous complexity of the photometric process during projection, reflection and capturing, it is extremely hard, if not impossible, to faithfully model the compensation explicitly.

In this paper, for the first time, an end-to-end projector compensation solution is presented to address the above issues. We start by reformulating the compensation problem to a novel form that can be learned online, as required by the compensation task in practice. This formation allows us to develop a convolutional neural network (CNN), named *CompenNet*, to implicitly learn the complex compensation function. In particular, CompenNet consists of two subnets, a UNet-like [29] backbone network and an autoencoder subnet. Firstly, the autoencoder subnet encourages rich multi-level interactions between the camera-captured projection surface image and the input image, and thus captures both photometric and environment information of the projection surface. Secondly, the UNet-like backbone network allows the visual details and interaction information to be carried to deeper layers and the output using the multi-level skip convolution layers. The two subnets together make CompenNet efficient in practice and allow CompenNet to learn the complex backward mapping from camera captured image to projector input image. In addition, a pre-trained solution is designed that can further improve the training efficiency with a small tradeoff in precision.

Another issue addressed in this paper is the absence of evaluation benchmarks for projector compensation, due mainly to the fact that traditional evaluation is highly setup dependent. More specifically, to evaluate a compensation algorithm, theoretically, its experimental results need to be actually projected and captured and then quantitatively compared with ground truth. This process makes it impractical to provide a shared benchmark among different research groups. In this work, we tackle this issue by deriving a surrogate evaluation protocol that requests no actual projection of the algorithm output. As a result, this surrogate allows us to construct, for the first time, a sharable setup-independent compensation benchmark.

The proposed compensation network, *i.e.*, CompenNet, is evaluated on the proposed benchmark that is carefully designed to cover various challenging factors. In the experiments, CompenNet demonstrates clear advantages compared with state-of-the-art solutions. In summary, in this paper we bring the following contributions:

1. For the first time, an end-to-end solution is proposed for projector compensation. Such solution allows our system to effectively and implicitly capture the complex photometric process involved in the projector compensation process.
2. The proposed CompenNet is designed to have two important subnets that enable rich multi-level interactions between projection surface and input image, and to carry interaction information and structural details through the network.
3. A pre-train method is proposed to further improve the practical efficiency of our system.
4. For the first time, a setup-independent projector compensation benchmark is constructed, which is expected to facilitate future works in this direction.

The source code, benchmark and experimental results are available at https://github.com/BingyaoHuang/CompenNet.

2. Related Works

In theory, the projector compensation process is a very complicated nonlinear function involving the camera and the projector sensor radiometric responses [24], lens distortion/vignetting [20], defocus [35, 37], surface material reflectance and inter-reflection [33]. A great amount of effort has been dedicated to designing practical and accurate compensation models, which can be roughly categorized into context-independent [9, 11, 26, 30] and context-aware ones [1, 2, 24, 33]. Detailed reviews can be found in [4, 12].

Context-independent methods typically assume that there is an approximate one-to-one mapping between the projector and camera image pixels, *i.e.*, a camera pixel is only dependent on its corresponding projector pixel and the surface patch illuminated by that projector pixel. Namely, each pixel is roughly independent of its neighborhood context. The pioneer work by Nayar *et al.* [26] proposes a linear model that maps a projector ray brightness to camera detected irradiance with a 3×3 color mixing matrix. Grossberg *et al.* [9] improve Nayar's work and model the environment lighting by adding a 3×1 vector to the camera-captured irradiance. However, a spectroradiometer is required to calibrate the uniform camera radiometric response function. Moreover, as pointed out in [20], even with a spectroradiometer the assumption of uniform radiometric response is usually violated, let alone the linearity. Considering the nonlinearity of the transfer function, Sajadi *et al.*

[30] fit a smooth higher-dimensional Bézier patches-based model with 9^3=729 sampling images. Grundhöfer and Iwai [11] propose a thin plate spline (TPS)-based method and reduce the number of sampling images to 5^3=125 and further deal with clipping errors and image smoothness with a global optimization step. Other than optimizing the image colors numerically, some methods specifically focus on human perceptual properties, *e.g*. Huang *et al*. [15] generate visually pleasing projections by exploring human visual system's chromatic adaptation and perceptual anchoring property. Also, clipping artifacts due to camera/projector sensor limitation are minimized using gamut scaling.

Despite largely simplifying the compensation problem, the context-independent assumption is usually violated in practice, due to many factors such as projector distance-to-surface, lens distortion, defocus and surface inter-reflection [33, 35, 37]. Moreover, it is clear that a projector ray can illuminate multiple surface patches, a patch can be illuminated by the inter-reflection of its surrounding patches, and a camera pixel is also determined by rays reflected by multiple patches.

Context-aware methods compensate a pixel by considering information from neighborhood context. Grundhöfer *et al*. [10] tackle visual artifacts and enhance brightness and contrast by analyzing the projection surface and input image prior. Li *et al*. [24] reduce the number of sampling images to at least two by sparse sampling and linear interpolation. Multidimensional reflectance vectors are extracted as color transfer function control points. Due to the small size of sampling dots, this method may be sensitive to projector defocus and lens vignetting. A simple linear interpolation using those unreliable samples may add to the compensation errors. Besides computing an offline compensation model, Aliaga *et al*. [1] introduce a run time linear scaling operation to optimize multiple projector compensation. Takeda *et al*. [33] propose an inter-reflection compensation method using an ultraviolet LED array.

Context-aware methods generally improve over previous methods by integrating more information. However, it is extremely hard to model or approximate the ideal compensation process due to complex interactions between the global lighting, the projection surface and the input image. Moreover, most existing works focus on reducing pixel-wise color errors rather than jointly improve the color and structural similarity to the target image.

Our method belongs to the Context-aware one, and in fact captures much richer context information by using the CNN architecture. Being the first end-to-end learning-based solution, our method implicitly and effectively models the complex compensation process. Moreover, the proposed benchmark is the first one that can be easily shared for verifiable quantitative evaluation.

Our method is inspired by the successes of recently proposed **deep learning-based image-to-image translation**, such as pix2pix [18], CycleGAN [40], style transfer [7, 16, 19], image super-resolution [6, 21, 23, 34] and image colorization [5, 17, 38]. That said, as the first deep learning-based projector compensation algorithm, our method is very different from these studies and has its own special constraints. For example, unlike above CNN models that can be trained once and for all, the projector compensation model needs to be quickly retrained if the system setup changes. However, in practice, both capturing training images and training the model are time consuming. In addition, data augmentations such as cropping and affine translations are not available for our task, because each camera pixel is strongly coupled with a neighborhood of its corresponding projector pixel and the projection surface patch illuminated by those pixels. Furthermore, general image-to-image translation models cannot formulate the complex spectral interactions between the global lighting, the projector backlight and the projection surface. In fact, in our evaluation, the advantage of the proposed method over the classical pix2pix [18] algorithm is clearly demonstrated quantitatively and qualitatively.

3. Deep Projector Compensation

3.1. Problem formulation

Our projector compensation system consists of a camera-projector pair and a planar projection surface placed at a fixed distance and orientation. Let a projector input image be $\boldsymbol{x}$; and let the projector's and the camera's composite geometric projection and radiometric transfer functions be π_p and π_c, respectively. Let the surface spectral reflectance property and spectral reflectance functions be $\boldsymbol{s}$ and π_s, respectively. Let the global lighting irradiance distribution be $\boldsymbol{g}$, then the camera captured image $\tilde{\boldsymbol{x}}$,[1] is given by:

$$\tilde{\boldsymbol{x}} = \pi_c\Big(\pi_s\big(\pi_p(\boldsymbol{x}), \boldsymbol{g}, \boldsymbol{s}\big)\Big) \tag{1}$$

The problem of projector compensation is to find a projector input image $\boldsymbol{x}^*$, named *compensation image* of $\boldsymbol{x}$ such that the camera captured image is the same as the ideal desired viewer perceived image, *i.e*.,

$$\pi_c\Big(\pi_s\big(\pi_p(\boldsymbol{x}^*), \boldsymbol{g}, \boldsymbol{s}\big)\Big) = \boldsymbol{x} \tag{2}$$

However, the spectral interactions and responses formulated in the above equation are very complex and can hardly be solved by traditional methods. Moreover, in practice it is also hard to measure $\boldsymbol{g}$ and $\boldsymbol{s}$ directly. For this reason, we capture their spectral interactions using a camera-captured surface image $\tilde{\boldsymbol{s}}$ under the global lighting and the projector backlight:

$$\tilde{\boldsymbol{s}} = \pi_c\Big(\pi_s\big(\pi_p(\boldsymbol{x}_0), \boldsymbol{g}, \boldsymbol{s}\big)\Big), \tag{3}$$

[1] We use 'tilde' ($\tilde{\boldsymbol{x}}$) to indicate a camera-captured image.

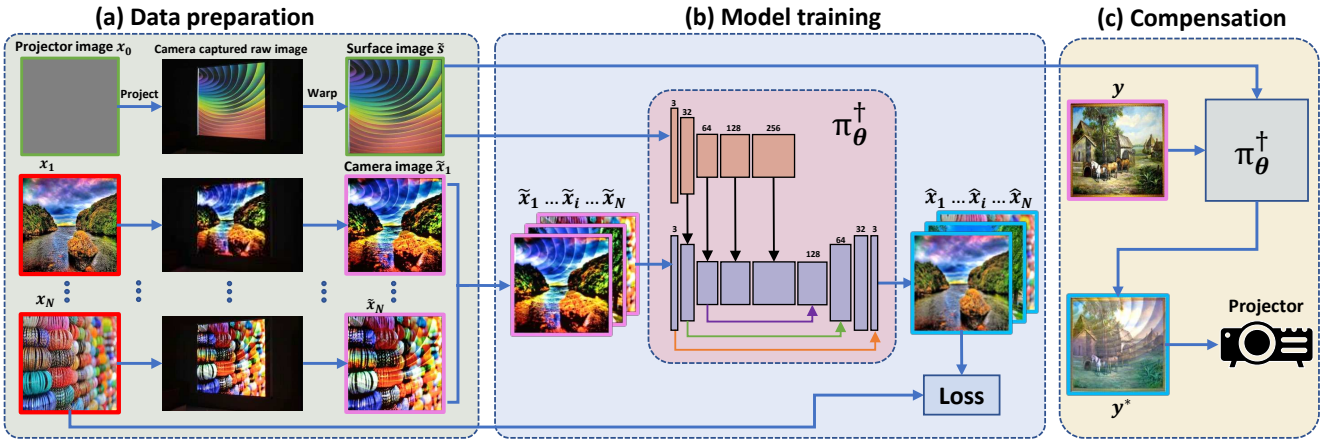


Figure 2: Flowchart of the proposed projector compensation pipeline consisting of three major steps. **(a)** Project and capture a surface image and a set of sampling images. **(b)** The proposed CompenNet, *i.e.*, $\pi_{\boldsymbol{\theta}}^\dagger$, is trained using the projected and captured image pairs. **(c)** With the trained model, an input image $\boldsymbol{y}$ can be compensated and projected.

where $\boldsymbol{x}_0$ is theoretically a black image. In practice, the projector outputs some backlight $\pi_p(\boldsymbol{x}_0)$ even when the input image is black, thus we encapsulate this factor in $\tilde{\boldsymbol{s}}$. When under low global illumination, $\tilde{\boldsymbol{s}}$ suffers from camera gamut clipping due to the limitation of camera dynamic range, thus we set $\boldsymbol{x}_0$ to a plain gray image to provide some illumination. Denoting the composite projector to camera radiometric transfer function in Eq. 2 as π and substituting $\boldsymbol{g}$ and $\boldsymbol{s}$ with $\tilde{\boldsymbol{s}}$, we have the compensation problem as

$$\pi(\boldsymbol{x}^*; \tilde{\boldsymbol{s}}) = \boldsymbol{x} \quad \Rightarrow \quad \boldsymbol{x}^* = \pi^\dagger(\boldsymbol{x}; \tilde{\boldsymbol{s}}), \tag{4}$$

where $\pi^\dagger$ is the pseudo-inverse function of π and, obviously, has no closed form solution.

3.2. Learning-based formulation

A key requirement for learning-based solution is the availability of training data. In the following we derive a method for collecting such data. Investigating the formulation in §3.1 we find that:

$$\tilde{\boldsymbol{x}} = \pi(\boldsymbol{x}; \tilde{\boldsymbol{s}}) \quad \Rightarrow \quad \boldsymbol{x} = \pi^\dagger(\tilde{\boldsymbol{x}}; \tilde{\boldsymbol{s}}) \tag{5}$$

This suggests that we can learn $\pi^\dagger$ over sampled image pairs like $(\tilde{\boldsymbol{x}}, \boldsymbol{x})$ and a surface image $\tilde{\boldsymbol{s}}$ as shown in Fig. 3. In fact, some previous solutions (*e.g.* [11, 30]) use similar ideas to fit models for $\pi^\dagger$, but typically under simplified assumptions and without modeling $\tilde{\boldsymbol{s}}$.

Instead, we reformulate the compensation problem with a deep neural network solution, which is capable of preserving the projector compensation complexity. In particular, we model the compensation process with an end-to-end learnable convolutional neural network, named *CompenNet* and denoted as $\pi_{\boldsymbol{\theta}}^\dagger$ (see (Fig. 2(b)), such that

$$\hat{\boldsymbol{x}} = \pi_{\boldsymbol{\theta}}^\dagger(\tilde{\boldsymbol{x}}; \tilde{\boldsymbol{s}}), \tag{6}$$

where $\hat{\boldsymbol{x}}$ is the compensation of $\tilde{\boldsymbol{x}}$ (not $\boldsymbol{x}$) and $\boldsymbol{\theta}$ contains the learnable network parameters. It is worth noting that $\tilde{\boldsymbol{s}}$ is fixed as long as the the setup is unchanged, thus only one $\tilde{\boldsymbol{s}}$ is needed in training and prediction.

By using Eq. 5, we can generate a set of N training pairs, denoted as $\mathcal{X} = \{(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i)\}_{i=1}^N$. Then, with a loss function $\mathcal{L}$, CompenNet can be learned by

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}'} \sum_i \mathcal{L}\big(\hat{\boldsymbol{x}}_i = \pi_{\boldsymbol{\theta}'}^\dagger(\tilde{\boldsymbol{x}}_i; \tilde{\boldsymbol{s}}),\ \boldsymbol{x}_i\big) \tag{7}$$

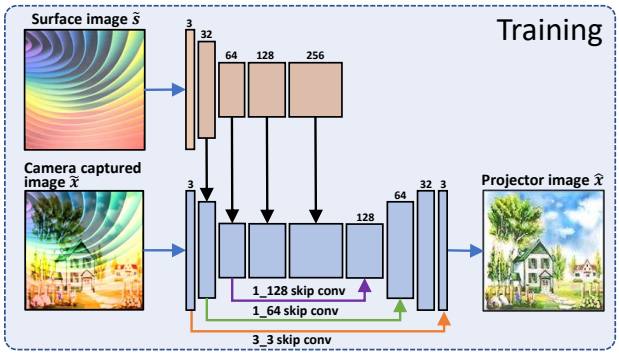
Our loss function is designed to jointly optimize the compensated image's color and structural similarity to the target image by combining the pixel-wise ℓ_1 and the SSIM loss:

$$\mathcal{L} = \mathcal{L}_{\ell_1} + \mathcal{L}_{\text{SSIM}} \tag{8}$$

The advantages of this loss function over the other loss functions are shown in [39] and in our comprehensive experimental comparisons in Table 3 and Fig. 5.

3.3. Network design

Based on the above formulation, our CompenNet is designed with two input images, $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{s}}$, corresponding to the camera-captured uncompensated image of $\boldsymbol{x}$ and the camera-captured surface image, respectively. The network architecture is shown in Fig. 3. Both two inputs and the output are 256×256×3 RGB images. Both input images are fed to a sequence of convolution layers to downsample and to extract multi-level feature maps. Note that in Fig. 3 we give the two paths different colors to indicate that the two branches do NOT share weights. The multi-level feature maps are then combined by element-wise addition, allowing the model to learn the complex spectral interactions between the global lighting, the projector backlight, the surface and the projected image.



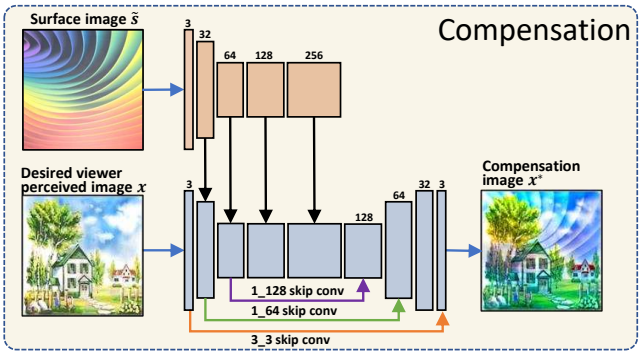


Figure 3: The architecture of CompenNet (ReLU layers omitted). All convolution layers are composed of 3×3 filters and all transposed convolution layers consist of 2×2 filters. Both upsample and downsample layers use a stride of two. The number of filters for each layer is labeled on its top. The skip convolution layers are shown in colored arrows, and the number of layers and the number of filters are labeled as **#layers_#filters** for conciseness. Learning the backward mapping from camera captured uncompensated image to the projector input image (left: $\tilde{\boldsymbol{x}} \mapsto \boldsymbol{x}$) is the same as learning the mapping from desired viewer perceived image to the compensation image (right: $\boldsymbol{x} \mapsto \boldsymbol{x}^*$).

We also pass low-level interaction information to high-level feature maps through skip convolution layers [14]. In the middle blocks, we extract rich features by increasing the feature channels while keeping the feature maps' width and height unchanged. Then, we use two transposed convolution layers to gradually upsample the feature maps to 256×256×32. Finally, the output image is an element-wise summation of the last layer's output and the three skip convolution layers at the bottom of Fig. 3. Note that we clamp the output image pixel values to [0, 1] before output. We find that deeper CNNs with more layers and filters, *e.g.* 512 filters can produce better compensation results, but suffer from overfitting on fewer sampling images and longer training and prediction time. However, if an application prefers accuracy to speed, it can add more convolution layers, increase the number of iterations and capture more training data accordingly. In this paper, we choose the architecture in Fig. 3 to balance training/prediction time and sampling data size.

To make the method more practical, we also provide a pre-trained model by projecting and capturing $N (N = 500)$ sampling images using a white projection surface. Once the setup, *e.g.*, the projection surface or the global lighting changes, rather than recapturing 500 training images, we use much fewer (*e.g.* 32) images to fine-tune the pre-trained model. This technique saves data preparation and training time and adds to the advantages over the existing solutions. We demonstrate the effectiveness of the pre-trained model in §5.3.

3.4. Training details

We implement CompenNet using PyTorch [27] and train it using Adam optimizer [22] with the following specifications: we set $\beta_1 = 0.9$ and fix ℓ_2 penalty factor to 10^{-4}. The initial learning rate is set to 10^{-3}, we also decay it by a factor of 5 every 800 iterations. The model weights are initialized using He's method [13]. We train the model for 1,000 iterations on two Nvidia GeForce 1080 GPUs with a batch size of 64, and it takes about 10min to finish training (for 500 samples). We report a comprehensive evaluation of different hyperparameters in supplementary material.

3.5. Compensation pipeline

To summarize, the proposed projector compensation pipeline consists of three major steps shown in Fig. 2. **(a)** We start by projecting a plain gray image $\boldsymbol{x}_0$ and N sampling images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ to the planar projection surface and capture them using the camera. Then each captured image is warped to the canonical view using a homography, and we denote warped camera images as $\tilde{\boldsymbol{x}}_i$. **(b)** Afterwards, we gather the N image pairs $(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i)$ and train the compensation model $\pi^{\dagger}_{\boldsymbol{\theta}}$. **(c)** Finally, with the trained model, we generate the compensation image $\boldsymbol{y}^*$ for an input image $\boldsymbol{y}$ and project $\boldsymbol{y}^*$ to the surface.

4. Benchmark

An issue left unaddressed in previous studies is the lack of public benchmarks for quantitative evaluation, due mainly to the fact that traditional evaluation is highly setup-dependent. In theory, to evaluate a compensation algorithm, its output compensation image $\boldsymbol{x}^*$ for input $\boldsymbol{x}$ should be actually projected to the projection surface, and then captured by the camera and quantitatively compared with the ground truth. This process is obvious impractical since it requests the same projector-camera-environment setup for fair comparison of different algorithms.

In this work, motivated by our problem formulation, we derive an effective surrogate evaluation protocol that re-

quests no actual projection of the algorithm output. The basic idea is, according to Eq. 5, we can collect testing samples in the same way as the training samples. We can also evaluate an algorithm in the similar way. Specifically, we collect the test set of M samples as $\mathcal{Y} = \{(\tilde{\boldsymbol{y}}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$, under the same system setup as the training set $\mathcal{X}$. Then the algorithm performance can be measured by averaging over similarities between each test input image $\boldsymbol{y}_i$ and its algorithm output $\hat{\boldsymbol{y}}_i = \pi_{\boldsymbol{\theta}}^{\dagger}(\tilde{\boldsymbol{y}}_i; \tilde{\boldsymbol{s}})$.

The above protocol allows us to construct a projector compensation evaluation benchmark, consisting of K system setups, each with a training set $\mathcal{X}_k$, a test set $\mathcal{Y}_k$ and a surface image $\tilde{\boldsymbol{s}}_k$, $k = 1, \ldots, K$.

System configuration. Our projector compensation system consists of a Canon 6D camera with image resolution of 960×640, and a ViewSonic PJD7828HDL DLP projector set to the resolution of 800×600. The distance between the camera and the projector is 500mm and the projection surface is around 1,000mm in front of the camera-projector pair. The camera exposure mode, focus mode and white balance mode are set to manual, the global lighting is fixed during the data capturing and system validation.

Dataset. To obtain the sampling colors and textures as diverse as possible, we download 700 colorful textured images from the Internet and use $N = 500$ for each training set $\mathcal{X}_k$ and $M = 200$ for each testing set $\mathcal{Y}_k$. In total $K = 24$ different setups are prepared for training and evaluation. Future works can replicate our results and compare with CompenNet on the benchmark without replicating our setups. For more camera perceived compensation results and the detailed configurations of the benchmark please refer to supplementary material.

5. Experimental Evaluations

5.1. Comparison with state-of-the-arts

We compare the proposed projector compensation method with a context-independent TPS model [11], an improved TPS model (explained below) and a general image-to-image translation model pix2pix [18] on our benchmark.

We first capture 125 pairs of plain color sampling image as used in the original TPS method [11]. We also fit the TPS method using our diverse textured training set $\mathcal{X}_k$, and name this method **TPS textured**. The experiment results in Table 1 and Fig. 4 show clear improvement of TPS textured over the original TPS method.

We then compare our method with pix2pix [18] to demonstrate the challenge of the projector compensation problem and to show the advantages of our formulation and architecture. We use the default implementation[2] of pix2pix with some adaptations for the compensation problem: (1) as mentioned in §2, data augmentation can break the strong coupling between the camera, the surface and the projector image, thus, we disable cropping, resizing and flipping. (2) We train the pix2pix model for 10,000 iterations and it takes about 10min with a batch size of one using the same hardware. The comparison results show that our method outperforms pix2pix by a significant margin on this task.

We find that TPS textured obtains slightly increased SSIM and slightly decreased PSNR when the data size increases. Pix2pix shows the lowest PSNR and SSIM when training data size is 250, and the highest PSNR and SSIM at 500. Only the proposed CompenNet achieves higher PSNR and SSIM when training data size increases from 125 to 500 (Table 1). Despite improving the performance of the CompenNet, a downside of large data size is increased data capturing time. In practice, taking hundreds of sampling images is time consuming, therefore, we proposed a pre-trained model that has improved performance than the default model when we only have limited training pairs and training time (§5.3).

Besides the state-of-the-arts above, we also tested the model-free "refinement by continuous feedback" method in [26] and find it work well. However, it has the disadvantage of needing several real projections, captures and iterations to converge for *each single* frame. Thus, it is impractical to evaluate it on the proposed setup-independent surrogate evaluation benchmark.

5.2. Effectiveness of the surface image

To show the effectiveness of our learning-based formulation and that the surface image $\tilde{\boldsymbol{s}}$ is a necessary model input, we compare with the proposed CompenNet that is without the input surface image and the corresponding autoencoder subnet, we name it **CompenNet w/o surf**. The results are shown in Table 1. Firstly, we can see a clear increase in PSNR and SSIM and a drop in RMSE when the $\tilde{\boldsymbol{s}}$ is included in the model input (**CompenNet**). This shows that our learning-based formulation has a clear advantage over the models that ignore the important information encoded in the surface image. Secondly, **CompenNet w/o surf** outperforms TPS, TPS textured and pix2pix on PSNR, RMSE and SSIM even $\tilde{\boldsymbol{s}}$ is not included. It is worth noting that for a new projection setting, simply replacing the surface image does not work well and it is necessary to train a new CompenNet from scratch. Fortunately, with the pre-trained model we can fine-tune from a reasonable initialization to reduce the number of training images and training time.

5.3. Effectiveness of the pre-trained model

We compare the default CompenNet model (using He's [13] initialization) with a model that is pre-trained with 500 training pairs projected to a **white** surface. Then we train and evaluate both models on each training set $\mathcal{X}_k$ and evaluation set $\mathcal{Y}_k$ of the 24 setups that the models have never

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

Figure 4: Comparison of TPS [11], TPS textured, pix2pix [18] and CompenNet on different surfaces. The 1st column is the camera-captured projection surface. The 2nd column is the camera-captured uncompensated projected image. The 3rd to 6th columns are the camera-captured compensation results of different methods. The last column is the ground truth input image. Each image is provided with two zoomed-in patches for detailed comparison. When trained with diverse textured images, TPS produces better results than its original version [11] that uses plain color images, though still suffers from hard edges, blocky effect and color errors. Compared with CompenNet, pix2pix generates unsmooth pixelated details and color errors.

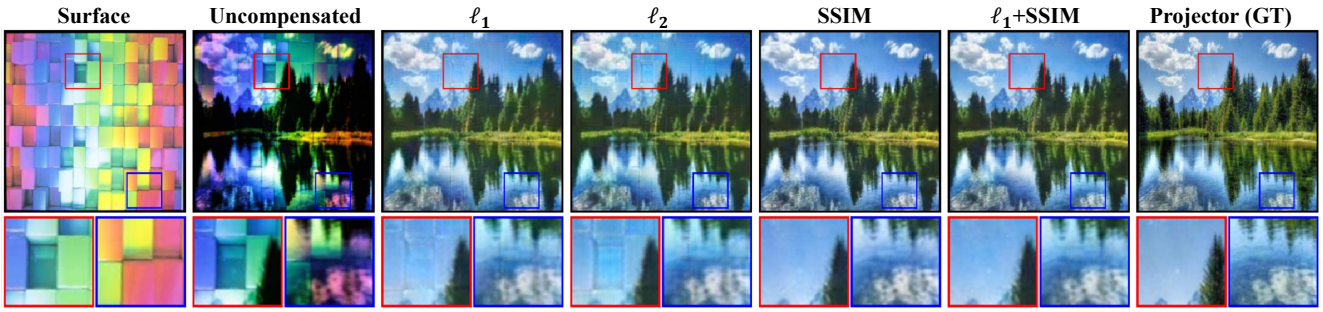


Figure 5: Qualitative comparison of CompenNet trained with ℓ_1 loss, ℓ_2 loss, SSIM loss and ℓ_1+SSIM loss. It shows that the ℓ_1 and ℓ_2 losses are unable to successfully compensate the surface patterns. The ℓ_1+SSIM and the SSIM losses produce similar results, but the water in the zoomed-in patch of SSIM is bluer than the ℓ_1+ SSIM and the ground truth.

been trained on. To demonstrate that pre-trained model obtains improved performance with limited training pairs and training time, we train the models for 500 iterations using only 32 training pairs. The results are reported in Table 2.

Clearly, we see that the pre-trained model outperforms the default counterpart even the 24 training and evaluation setups have different lightings and surface textures as the pre-trained setup. Our explanation is that despite the surfaces have different appearances, the pre-trained model has already learned partial radiometric transfer functions of the

Table 1: Quantitative comparison of compensation algorithms. Results are averaged over $K = 24$ different setups.

#Train	Model	PSNR↑	RMSE↓	SSIM↑
125	TPS [11]	17.6399	0.2299	0.6042
	TPS [11] textured	19.3156	0.1898	0.6518
	Pix2pix [18]	17.9358	0.2347	0.6439
	CompenNet w/o Surf.	19.8227	0.1786	0.7003
	CompenNet	**21.0542**	**0.1574**	**0.7314**
250	TPS [11] textured	19.2764	0.1907	0.6590
	Pix2pix [18]	16.2939	0.2842	0.6393
	CompenNet w/o Surf.	20.0857	0.1733	0.7146
	CompenNet	**21.2991**	**0.1536**	**0.7420**
500	TPS [11] textured	19.2264	0.1917	0.6615
	Pix2pix [18]	18.0923	0.2350	0.6523
	CompenNet w/o Surf.	20.2618	0.1698	0.7209
	CompenNet	**21.7998**	**0.1425**	**0.7523**
-	Uncompensated	12.1673	0.4342	0.4875

camera and the projector. This pre-trained model makes our method more practical, *i.e.*, as long as the projector and camera are not changed, the pre-trained model can be quickly tuned with much fewer training images and thus shortens the image capturing and training time. Another finding is even with 32 training pairs and 500 iterations, the proposed CompenNet, with or without pre-train, performs better than TPS [11], TPS textured and pix2pix [18] in Table 1. Furthermore, CompenNet has much fewer parameters (1M) than pix2pix's default generator (54M parameters). This further confirms that the projector compensation is a complex problem and is different from general image-to-image translation tasks, and carefully designed models are necessary to solve this problem.

5.4. Comparison of different loss functions

Existing works fit the composite radiometric transfer function by linear/nonlinear regression subject to a pixel-wise ℓ_2 loss and this loss function is known to penalize large pixel errors while oversmoothes the structural details. We investigate four different loss functions, i.e., pixel-wise ℓ_1 loss, pixel-wise ℓ_2 loss, SSIM loss, and ℓ_1+SSIM loss. The qualitative and quantitative comparisons are shown in Fig. 5 and Table 3, respectively. Compared with SSIM loss, pixel-wise ℓ_1 and ℓ_2 losses cannot well compensate surface patterns, notice the hard edges in the red zoomed-in patches in Fig. 5. Consistent to the qualitative results, the SSIM column in Table 3 also shows a clear disadvantage of both pixel-wise ℓ_1 and ℓ_2 losses. Although SSIM loss alone obtains the best SSIM value, its PSNR and RMSE are the 2nd worst. After comprehensive experiments on our benchmark, we find that ℓ_1+SSIM loss obtains the best PSNR/RMSE and the 2nd best SSIM, thus, we choose it as our CompenNet loss function. Moreover, even when trained with pixel-wise ℓ_1 loss, CompenNet outperforms TPS, TPS textured and pix2pix on PSNR, RMSE and SSIM, this again shows a clear advantage of our task-targeting formulation and architecture.

Table 2: Quantitative comparison between a pre-trained CompenNet and a CompenNet randomly initialized using He's method [13], both trained using **only 32 samples and 500 iterations** with a batch size of 32, and take about 170s.

Model	PSNR↑	RMSE↓	SSIM↑
CompenNet	19.6767	0.1863	0.6788
CompenNet pre-train	**20.3423**	**0.1688**	**0.7165**
Uncompensated	12.1673	0.4342	0.4875

Table 3: Quantitative comparison of different loss functions for the proposed CompenNet.

Loss	PSNR↑	RMSE↓	SSIM↑
ℓ_1	21.1782	0.1527	0.6727
ℓ_2	20.7927	0.1594	0.6453
SSIM	21.0134	0.1566	**0.7591**
ℓ_1+SSIM	**21.7998**	**0.1425**	0.7523
Uncompensated	12.1673	0.4342	0.4875

5.5. Limitations

We focus on introducing the first end-to-end solution to projector compensation, for planar surfaces with decent, not necessarily ideal, reflectance/geometric qualities. In addition, we have not experimented on surfaces with special reflectance transport properties, such as water, strong specular reflection, geometry inter-reflection and semi-gloss, thus it may not work well in these cases.

6. Conclusions

In this paper, we reformulate the projector compensation problem as a learning problem and propose an accurate and practical end-to-end solution named CompenNet. In particular, CompenNet explicitly captures the complex spectral interactions between the environment, the projection surface and the projector image. The effectiveness of our formulation and architecture is verified by comprehensive evaluations. Moreover, for the first time, we provide the community with a novel setup-independent evaluation benchmark dataset. Our method is evaluated carefully on the benchmark, and the results show that our end-to-end learning solution outperforms state-of-the-arts both qualitatively and quantitatively by a significant margin. To make our model more practical, we propose a pre-train method, which adds to the advantages over the prior works.

Acknowledgement. We thank the anonymous reviewers for valuable and inspiring comments and suggestions.

References

[1] Daniel G. Aliaga, Yu Hong Yeung, Alvin Law, Behzad Sajadi, and Aditi Majumder. Fast high-resolution appearance editing using superimposed projections. *ACM Tran. on Graphics*, 31(2):13, 2012.

[2] Mark Ashdown, Takahiro Okabe, Imari Sato, and Yoichi Sato. Robust content-dependent photometric projector compensation. In *Proceedings of IEEE International Workshop on Projector Camera Systems (PROCAMS) 2006*, 2006.

[3] Oliver Bimber, Andreas Emmerling, and Thomas Klemmer. Embedded entertainment with smart projectors. *Computer*, 38(1):48–55, 2005.

[4] Oliver Bimber, Daisuke Iwai, Gordon Wetzstein, and Anselm Grundhöfer. The visual computing of projector-camera systems. In *Computer Graphics Forum*, page 84, 2008.

[5] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David A Forsyth. Learning diverse image colorization. In *CVPR*, pages 2877–2885, 2017.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199. Springer, 2014.

[7] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[8] Jason Geng. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128, jun 2011.

[9] Michael D. Grossberg, Harish Peri, Shree K. Nayar, and Peter N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. In *CVPR*, June 2004.

[10] Anselm Grundhöfer and Oliver Bimber. Real-time adaptive radiometric compensation. *IEEE TVCG*, 14(1):97–108, 2008.

[11] Anselm Grundhöfer and Daisuke Iwai. Robust, error-tolerant photometric projector compensation. *IEEE TIP*, 24(12):5086–5099, 2015.

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

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Tai-Hsiang Huang, Ting-Chun Wang, and Homer H Chen. Radiometric compensation of images projected on non-white surfaces by exploiting chromatic adaptation and perceptual anchoring. *IEEE TIP*, 26(1):147–159, 2017.

[16] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017.

[17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Tran. on Graphics*, 35(4):110, 2016.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[20] Ray Juang and Aditi Majumder. Photometric self-calibration of a projector-camera system. In *CVPR*, pages 1–8. IEEE, 2007.

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, June 2016.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, volume 5, 2015.

[23] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[24] Yuqi Li, Aditi Majumder, Meenakshisundaram Gopi, Chong Wang, and Jieyu Zhao. Practical radiometric compensation for projection display on textured surfaces using a multidimensional model. In *Computer Graphics Forum*, volume 37, pages 365–375. Wiley Online Library, 2018.

[25] Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE TVCG*, (1):1–1, 2017.

[26] Shree K. Nayar, Harish Peri, Michael D. Grossberg, and Peter N. Belhumeur. A projection system with radiometric compensation for screen imperfections. In *ICCV-W PROCAMS*, volume 3, 2003.

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[28] Ramesh Raskar, Jeroen Van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. ilamps: geometrically aware and self-configuring projectors. *ACM Tran. on Graphics*, 22(3):809–818, 2003.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[30] Behzad Sajadi, Maxim Lazarov, and Aditi Majumder. Adict: accurate direct and inverse color transformation. In *ECCV*, pages 72–86. Springer, 2010.

[31] Christian Siegl, Matteo Colaianni, Marc Stamminger, and Frank Bauer. Adaptive stray-light compensation in dynamic multi-projection mapping. *Computational Visual Media*, 3(3):263–271, 2017.

[32] Christian Siegl, Matteo Colaianni, Lucas Thies, Justus Thies, Michael Zollhöfer, Shahram Izadi, Marc Stamminger, and Frank Bauer. Real-time pixel luminance optimization for dynamic multi-projection mapping. *ACM Tran. on Graphics*, 34(6):237, 2015.

[33] Shoichi Takeda, Daisuke Iwai, and Kosuke Sato. Inter-reflection compensation of immersive projection display by spatio-temporal screen reflectance modulation. *IEEE TVCG*, 22(4):1424–1431, 2016.

[34] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018.

[35] Liming Yang, Jean-Marie Normand, and Guillaume Moreau. Practical and precise projector-camera calibration. In *ISMAR*, pages 63–70. IEEE, 2016.

[36] Takenobu Yoshida, Chinatsu Horii, and Kosuke Sato. A virtual color reconstruction system for real heritage with light projection. In *Proceedings of International Conference on Virtual Systems and Multimedia*, volume 3, 2003.

[37] Li Zhang and Shree Nayar. Projection defocus analysis for scene capture and image display. In *ACM Tran. on Graphics*, volume 25, pages 907–915, 2006.

[38] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016.

[39] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE TCI*, 3(1):47–57, 2017.

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.