LAPIG: Language Guided Projector Image Generation with Surface Adaptation and Stylization

Yuchen Deng, Haibin Ling, and Bingyao Huang

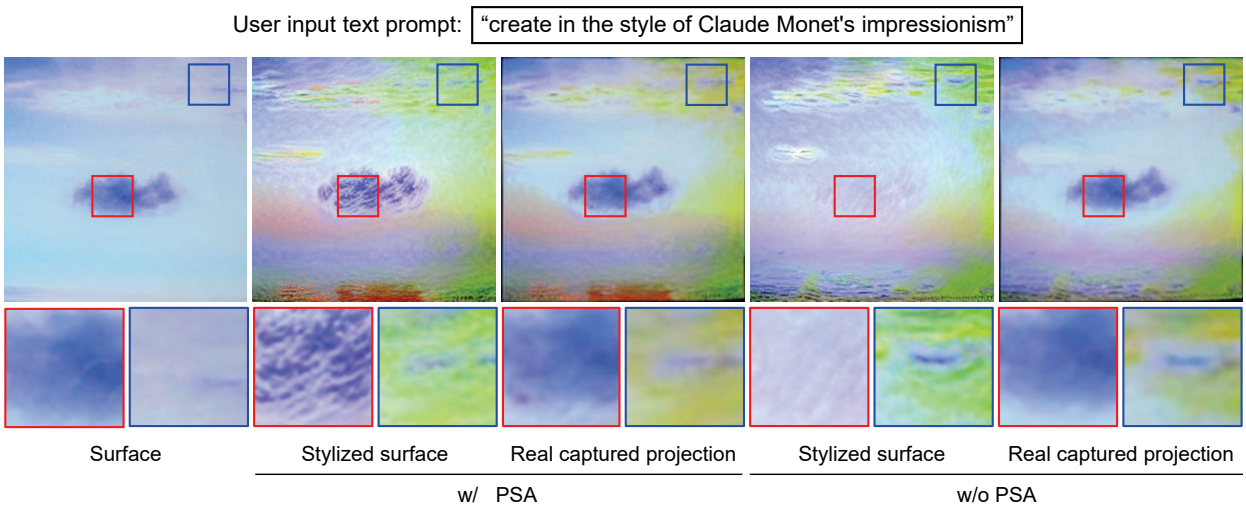

Fig. 1: Language Guided Projector Image Generation with Surface Adaptation and Stylization (**LAPIG**): The user input text prompt to stylize the surface is shown on the top of the figure. The 1st column shows the projection surface to be stylized. The 2nd and 4th columns show the stylized surface w/ or w/o our projection surface adaptation (**PSA**) approach. The 3rd and 5th columns present real captured scene under the compensated projection of the 2nd and 4th columns, respectively. Clearly, our LAPIG can adapt the stylization to the surface by showing fewer discrepancies between the target stylized surface (the 2nd column) and the real captured projection (the 3rd column). More results are provided in § 4 and **supplementary**.

Abstract— We propose LAPIG, a language guided projector image generation method with surface adaptation and stylization. LAPIG consists of a projector-camera system and a target textured projection surface. LAPIG takes the user text prompt as input and aims to transform the surface style using the projector. LAPIG's key challenge is that due to the projector's physical brightness limitation and the surface texture, the viewer's perceived projection may suffer from color saturation and artifacts in both dark and bright regions, such that even with the state-of-the-art projector compensation techniques, the viewer may see clear surface texture-related artifacts. Therefore, how to generate a projector image that follows the user's instruction while also displaying minimum surface artifacts is an open problem. To address this issue, we propose projection surface adaptation (PSA) that can generate compensable surface stylization. We first train two networks to simulate the projector compensation and project-and-capture processes, this allows us to find a satisfactory projector image without real project-and-capture and utilize gradient descent for fast convergence. Then, we design content and saturation losses to guide the projector image generation, such that the generated image shows no clearly perceivable artifacts when projected. Finally, the generated image is projected for visually pleasing surface style morphing effects. The source code and video are available on the project page: https://Yu-chen-Deng.github.io/LAPIG/.

Index Terms—Projector-camera systems, projection mapping, style transfer, projector compensation.

1 Introduction

Projection mapping (PM) [1,4,23,31,35,38,39,49,61] is a versatile technology that transforms the appearance of objects into dynamic displays by projecting digital images or videos onto them. This technique is utilized in various fields, including performing arts [6–8, 19, 33], product design [9, 12, 16, 17, 31, 44], and interactive experiences [2, 31, 43, 45], where it enhances the visual representation of objects with projected textures, colors, and other effects. By incorporating physical objects and environments into the display system, projection mapping increases user immersion and interaction, which is crucial for AR/VR applications.

Due to the complex environment, object material, and device optical properties, directly projecting desired patterns onto the target object may suffer from photometric and geometric distortions. Therefore, most projection mapping applications rely on projector compensation techniques [10, 21, 22, 24, 27, 34, 35, 41, 59] to cancel distortions. However, due to the projector's physical brightness limitation, not every projection surface is fully compensable, especially those extremely dark and bright regions, *e.g.*, the dark cloud in the surface (the 1st column) of Fig. 1. This is because the extremely dark/bright projection surface regions absorb/reflect more light, even when the projector brightness is set to the highest/lowest, the surface texture cannot be fully compensated.

- *Yuchen Deng is with Southwest University. E-mail: swudyc714@email.swu.edu.cn.*
- *Haibin Ling is with Stony Brook University. E-mail: hling@cs.stonybrook.edu.*
- *Bingyao Huang is with Southwest University. E-mail: bhuang@swu.edu.cn. Corresponding author.*

To address this issue, we propose LAPIG, a language guided projector image generation method with surface adaptation and stylization. LAPIG consists of a language guided style transfer module (LGST) for user controllable surface stylization and a projection surface adaptation (PSA) module for compensable stylization. The key difficulty is how to combine the language guided style transfer and the compensable stylization. A naive approach is to repetitively generate different projector input images using LGST, then project and capture the generated images until a compensable stylization is achieved. However, this approach is time-consuming because the real project-and-capture process is involved. A smarter method would be to simulate the real project-and-capture and projector compensation processes using a mathematical model, *e.g.*, a light transport matrix and its inverse, and then perform the similar random projector image generation as in the naive approach. However, this approach is still time-consuming due to exhaustive search and unstable convergence.

We propose to address this issue using gradient descent rather than random search, for efficient convergence. To allow for gradient descent-based optimization, we first design two differentiable metrics for surface adaptation quality, *i.e.*, projection consistency and color saturation losses. The first loss measures the similarity between the target stylization and the captured projection, and the second measures the projection and compensation color saturation losses in extremely dark and bright surface regions. Then, we pre-train two neural networks to simulate the projector compensation and project-and-capture processes, respectively. Finally, given an initial stylized image, we feed it to the compensation and project-and-capture modules to simulate the camera-captured compensation, which is used to calculate the projection consistency and color saturation losses. We iteratively backpropagate the loss gradient to update the LGST model parameters until the stylized surface is compensable.

The contributions of this work can be summarized in three aspects:

- The proposed LAPIG is the first language guided projector image generation method that can not only stylize the projection surface but also adapt to it.
- LAPIG nontrivially integrates user language guided style transfer, project-and-capture simulation, and projector compensation, enabling gradient-based optimization for projector image generation.
- We propose projection consistency and color saturation losses to measure compensation quality and guide PSA optimization, and they are expected to facilitate other projection mapping applications.

In the rest of the paper, we introduce the related work in § 2, and describe the problem formulation and the proposed LAPIG in § 3. We show our system configurations and experimental evaluations in § 4, discuss the paper in § 5, and conclude the paper in § 6.

2 Related Work

2.1 Projection mapping

Traditional projection mapping (PM) techniques aim to alter the appearance of physical objects by projecting carefully designed patterns onto their surfaces. By doing so, these methods effectively modify the color, texture, and brightness of the object surfaces, serving as a widely adopted tool for spatial augmented reality (SAR). To achieve satisfactory visual effects in PM, projector compensation and projector-camera systems (ProCams) relighting (also referred to as project-and-capture simulation) are commonly employed.

2.1.1 Projector compensation

Projector compensation aims to counteract distortions caused by environmental factors, equipment, and projection surfaces, thus improving the viewer's perception of projection effects by producing a compensation image [8–10, 21, 22, 24, 27, 34, 41, 42, 59]. Raskar *et al*. [52] achieved projection on non-flat objects using geometric corrections paired with camera radiometric calibration. In practice, while camera radiometric calibration is straightforward, recalibration becomes crucial when projector settings like brightness, contrast, or color profiles are altered, posing challenges for projection mapping applications that require frequent adjustment. To address these issues, Grundhöfer and Iwai [21] introduced a robust photometric compensation method using a pixel-wise thin plate spline (TPS) to directly estimate the photometric compensation function from RGB sampling images, and waives radiometric calibration. Huang *et al*. [28] achieved visually satisfactory projections by exploring properties of the human visual system such as chromatic adaptation and perceptual anchoring. They also employed gamut scaling to mitigate clipping artifacts from camera and projector sensor constraints. Luo *et al*. [42] also took advantage of the properties of the human visual system to generate a high-quality non-negative image, which may be applied to reduce projector compensation artifacts. Huang *et al*. [24, 27] proposed a learning-based formulation of the project-and-capture process, and applied deep neural networks to learn photometric and geometric compensation functions. Wang *et al*. [59] introduced CompenHR, a resource-efficient technique for high-resolution projector compensation. They also suggested a multi-threaded video compensation strategy to dynamically adjust the compensation parameters [60]. Kusuyama *et al*. [40] presented a projection system that optically eliminates shadows in PM, and Yasui *et al*. [61] proposed a novel approach using a mixed light field within the projector, enhancing PM efficiency in brightly lit settings. For more detailed reviews, see [10, 22, 30].

2.1.2 ProCams relighting/Project-and-capture simulation

ProCams relighting (or project-and-capture simulation) simulates the physical project-and-capture process of ProCams and allows adjustments to appearance editing based on inferred captured projection without actual project-and-capture. Early methods focus on light transport matrix (LTM) [13, 15, 58], which models the irradiance of each camera pixel as a linear combination of the radiances of all projector pixels. These methods can produce accurate global illumination, but are usually computationally intensive and require high initial conditions. To further improve relighting efficiency, Huang *et al*. [25] proposed an end-to-end trainable model that explicitly learns the photometric and geometric mappings involved in the project-and-capture process. Erel *et al*. [17] first formulated ProCams simulation using a neural radiance fields (NeRF) framework.

2.1.3 Interactive projection mapping (IPM)

Interactive projection mapping enables user feedback and control over the projected content. In scenarios such as art installations and museum displays, IPM not only enriches the presentation, but also provides the audience with the opportunity to participate in content creation. IPM can be categorized into three types.

(1) Sensor-based interaction captures user movements or expressions, allowing adjustment of projected content's appearance. Sensors [7, 48] and cameras [3, 18, 19, 31, 33] are used to capture the user's movements or expressions, allowing for the adjustment of the projected content's appearance. Amano [2] introduced a light field projection method that enhances material perception based on the direction from which it is viewed. Recently, Miyatake *et al*. [45] proposed a projection-based visuo-haptic AR system that allows independent rendering of visual and haptic content by embedding user-imperceptible tactile control signals in projected images. Erel *et al*. [18] presented a novel approach to dynamically projecting 3D content onto the user's hands in real time.

(2) Touch-based interaction provides users with direct control over projected content, allowing real-time adjustments via touch [43, 44, 50]. Sato *et al*. [56] proposed a high-speed projector-camera system that enables real-time interaction by creating optical illusions, dynamically altering the player's perception of a puck that is randomly hit during gameplay.

(3) Text-based interaction allows for dynamic and context-aware projection using verbal instructions or written descriptions. As diffusion models gain prominence, natural language also acts as a tool to alter and control projected content [17]. To our best knowledge, there is no

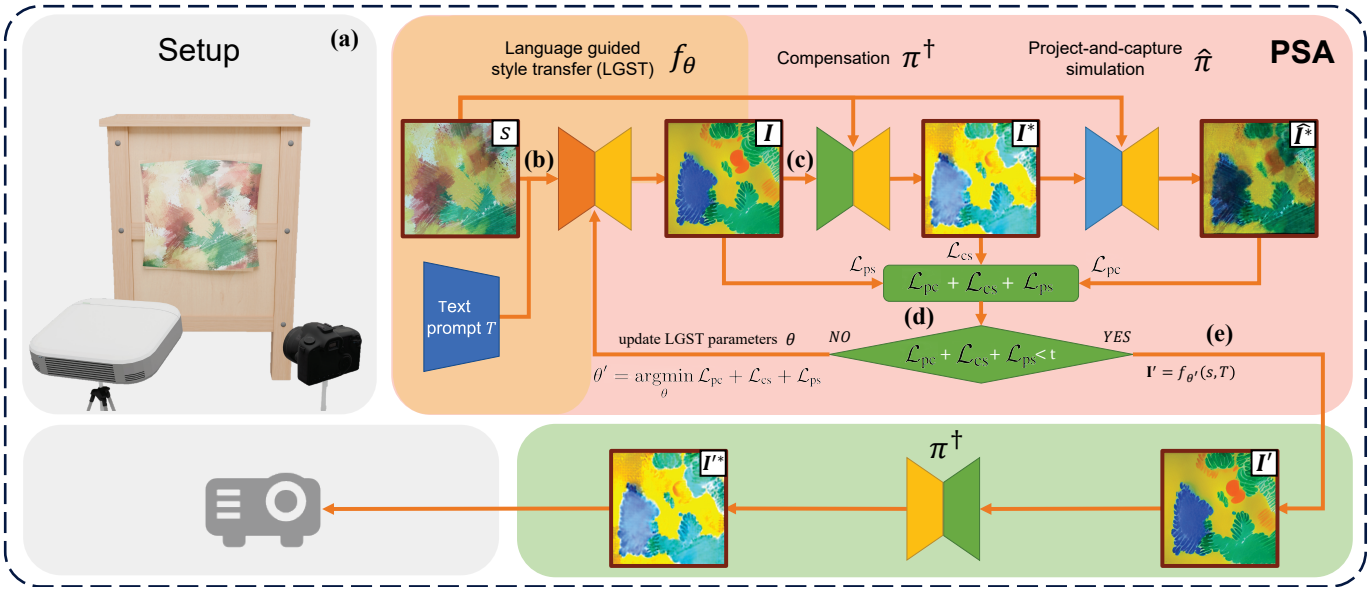

Fig. 2: System overview of our **LAPIG**. Note that we plot network in different colors to distinguish different modules utilized in PSA. (a) System setup consists of a projector, a camera, and a textured surface. (b) First, the surface s is captured by the camera, which is used in conjunction with text prompt T to generate a stylized surface image $\boldsymbol{I}$. (c) Then, the surface image s and its stylized image $\boldsymbol{I}$ are input into the compensation network $\pi^\dagger$ to infer the compensated stylized image $\boldsymbol{I}^*$. The compensated image is further input to the project-and-capture simulation network $\hat{\pi}$ to infer the camera-captured compensation effects $\hat{\boldsymbol{I}}^*$. The model architectures of $\hat{\pi}$ and $\hat{\boldsymbol{I}}^*$ are shown in Fig. 3. (d) Afterwards, the three compensation quality metrics/losses are calculated and backpropageted to update the LGST model parameters θ. (e) Finally, the optimal stylized surface image $\boldsymbol{I}'$ is generated using the updated LGST model, and after applying projector compensation, it is sent to the projector for real projection surface stylization.

method that can perform language guided projector image generation with surface adaptation and stylization.

2.2 Neural style transfer

Our LAPIG leverages text prompt guided neural style transfer for surface adaptation and stylization, making it essential to review related work in this field.

2.2.1 Non-text guided style transfer

Neural style transfer (NST) was initially introduced by Gatys *et al*. [20], who proposed a method to blend content and style features extracted through a deep convolutional neural network (CNN). Building on this, Johnson *et al*. [32] introduced a model that performs fast style transfer using a pre-trained perceptual loss function, significantly speeding up the style transfer process. Other related works include style transfer models based on generative adversarial networks (GAN), such as CycleGAN [63], which can perform image-to-image translation without paired data. These models are entirely image-driven and do not incorporate any text prompts.

2.2.2 Simple text guided style transfer

As text-to-image generation models advanced, researchers began exploring the use of simple text prompts to guide the style and content of the generated images. An early example is by Reed *et al*. [54], demonstrating the feasibility of generating image-matching natural language descriptions. Recently, transformer-based models [5, 53, 57] have further expanded this concept, enabling the generation of diverse images based on simple text descriptions. These methods allow users to generate images based on natural language prompts, but often offer limited guidance on fine-grained style and detail.

2.2.3 Fine-grained text guided style transfer

To enhance the guidance of text in image generation and editing, researchers have developed models capable of fine-grained guidance over image style and content based on text prompts. CLIP [51], a cross-modal contrastive learning model, significantly improved the alignment between text and images, laying the groundwork for subsequent text-driven image editing. Building on CLIP, models like StyleCLIP [47] have successfully combined CLIP with StyleGAN [36], enabling detailed style editing of images according to precise text prompts. Further advancements include InstructPix2Pix [11], which extends the concept by providing an instruction-based interface for text-guided image editing. InstructPix2Pix integrates text prompts with pix2pix [29] models, allowing users to apply fine-grained modifications to images according to textual instructions. These models can not only generate images that match textual descriptions, but also adjust specific details such as facial expressions, colors, and textures based on user input.

3 Methods

3.1 Problem formulation

We start by formulating the project-and-capture process of ProCams by expanding on the notation from [25]. Denote the project and capture function as π_p and π_c, respectively, and denote the projector input image as $\boldsymbol{I}$, the projection surface as s, then the camera-captured surface with superimposed projection $\tilde{\boldsymbol{I}}$ is given by:

$$\tilde{\boldsymbol{I}} = \pi_\mathrm{c}\left(\pi_\mathrm{p}(\boldsymbol{I}), s\right) \tag{1}$$

Denote the composite project-and-capture process and its inverse as π and $\pi^\dagger$, respectively. Then, the above equation can be formulated as:

$$\tilde{\boldsymbol{I}} = \pi\left(\boldsymbol{I}, s\right) \tag{2}$$

$$\boldsymbol{I} = \pi^\dagger\left(\tilde{\boldsymbol{I}}, s\right) \tag{3}$$

The inverse operation (Eqn. 3) can be applied to projector compensation [25, 27] by replacing the input image $\tilde{\boldsymbol{I}}$ with a desired viewer perceived image.

As shown in Fig. 2, the goal of our LAPIG is to achieve projection surface stylization, according to the user input text prompt T:

$$\boldsymbol{I} = f_\theta(s, T), \tag{4}$$

where $\boldsymbol{I}$ is the desired viewer's perceived surface stylization effect (the 2nd column of Fig. 1), and f_θ is a language guided neural style transfer

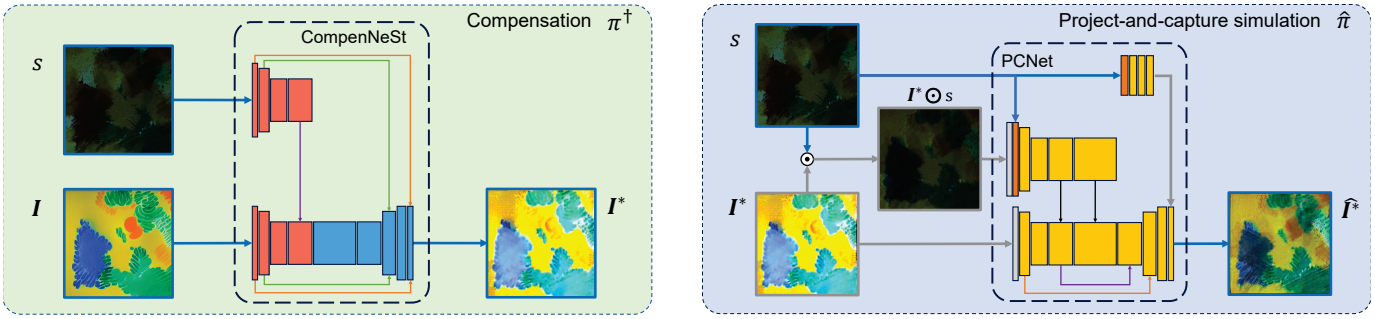

Fig. 3: Network architecture of projector compensation model $\pi^\dagger$ and project-and-capture simulation model $\hat{\pi}$. CompenNeSt (identified by the light green block on the left) features a siamese encoder (red blocks with shared weights) and a decoder (blue blocks), performing the inverse mapping from the desired perceived image $\boldsymbol{I}$ to the projector compensation image $\boldsymbol{I}^*$. PCNet (represented by the light blue block on the right) utilizes a dual-branch encoder-decoder architecture to capture complex photometric transformations. Specifically, it calculates the rough shading $\boldsymbol{I}^* \odot s$ using the camera-captured surface image s and compensation image $\boldsymbol{I}^*$, and feed them to the middle encoder branch. Likewise, $\boldsymbol{I}^*$ is passed to the backbone encoder branch. Skip connections between these two branches model photometric interactions across the three inputs at various levels. Additionally, s is passed through three convolutional layers to the output layer. Ultimately, the backbone decoder fuses the feature maps into $\hat{\boldsymbol{I}}^*$, which simulates the camera-captured scene under the projection of $\boldsymbol{I}^*$.

model (LGST) parameterized by θ, *e.g.*, InstructPix2Pix [11]. Then, we apply projector compensation (Eqn. 3) to $\boldsymbol{I}$:

$$\boldsymbol{I}^* = \pi^\dagger\left(\boldsymbol{I}, s\right), \tag{5}$$

and project the compensated image $\boldsymbol{I}^*$ to the projection surface, wishing the final camera-captured projection $\boldsymbol{I}^*$ matches the desired effect $\boldsymbol{I}$:

$$\tilde{\boldsymbol{I}}^* = \pi\left(\boldsymbol{I}^*, s\right) \approx \boldsymbol{I} \tag{6}$$

3.2 Projection surface adaptation (PSA)

A straightforward solution for $\tilde{\boldsymbol{I}}^*$ in Eqn. 6 relies on directly compensating the LGST (f_θ) stylized image $\boldsymbol{I}$. Although LGST (f_θ) can effectively edit or stylize an image based on text prompts, it lacks awareness of the actual ProCams configuration and may disregard the physical project-and-capture principles. As a result, the final captured stylized effect $\tilde{\boldsymbol{I}}^*$ often displays noticeable surface artifacts and diverges significantly from the intended effect $\boldsymbol{I}$, particularly due to the complex environment, surface material, and brightness limitations of the projector, despite using advanced projector compensation methods (refer to the last two columns of Fig. 1). To address this issue, we propose conditioning the stylization of the projection surface on both the user's input text prompt and the quality of compensation (Fig. 2).

We first design three metrics/losses for compensation quality measure by:

$$\mathcal{L}_{\text{pc}} : \|\hat{\boldsymbol{I}}^* - \boldsymbol{I}\| \tag{7}$$

$$\mathcal{L}_{\text{cs}} : \|\max(\boldsymbol{I}^* - 1, 0)\|^2 + \|\min(\boldsymbol{I}^*, 0)\|^2 \tag{8}$$

$$\mathcal{L}_{\text{ps}} : \|\max(\boldsymbol{I} - \boldsymbol{I}_+, 0)\|^2 + \|\min(\boldsymbol{I} - \boldsymbol{I}_-, 0)\|^2 \tag{9}$$

$$\hat{\boldsymbol{I}}^* = \hat{\pi}(\boldsymbol{I}^*, s) = \hat{\pi}(\pi^\dagger(\boldsymbol{I}, s), s), \tag{10}$$

where $\hat{\boldsymbol{I}}^*$ is the simulated camera-captured projection of $\boldsymbol{I}^*$, and $\hat{\pi}$ is a model that simulates the real project-and-capture process π (Fig. 3 right block). $\boldsymbol{I}_+$ and $\boldsymbol{I}_-$ are two projector surface images captured under the brightest and darkest projector illumination to measure the projector's physical brightness range. $\mathcal{L}_{\text{pc}}$, $\mathcal{L}_{\text{cs}}$ and $\mathcal{L}_{\text{ps}}$ are projection consistency, compensation saturation, and projection saturation, respectively. In particular, $\mathcal{L}_{\text{pc}}$ measures the similarity between the simulated and desired surface stylization effects. $\mathcal{L}_{\text{cs}}$ measures the color saturation error due to projector compensation, *i.e.*, pixel values outside the plausible RGB value range [0, 1] are counted as saturation errors. Similarly, $\mathcal{L}_{\text{ps}}$ measures the color saturation error due to the projection, *i.e.*, any pixel value outside the range of the darkest ($\boldsymbol{I}_-$) and brightest ($\boldsymbol{I}_+$) images captured by the camera is unattainable due to the projector's brightness limitations, and should also be regarded as saturation errors.

Algorithm 1: PSA: Projection Surface Adaptation.

Input:
s: camera-captured projection surface under gray illumination
T: user input text prompt
t: PSA loss threshold for optimization termination
β: step size in minimizing the PSA losses
Output : $\boldsymbol{I}'^*$: compensated surface stylization image

Initialize $\theta' \leftarrow \theta$
do
  $\boldsymbol{I} \leftarrow f_{\theta'}(s, T)$ ▷ `surface stylization`
  $\boldsymbol{I}^* \leftarrow \pi^\dagger(\boldsymbol{I}, s)$ ▷ `compensation`
  $\hat{\boldsymbol{I}}^* \leftarrow \hat{\pi}(\boldsymbol{I}^*, s)$ ▷ `simulate project-and-capture`
  $\mathcal{L}_{\text{pc}} \leftarrow \|\hat{\boldsymbol{I}}^* - \boldsymbol{I}\|$
  $\mathcal{L}_{\text{cs}} \leftarrow \|\max(\boldsymbol{I}^* - 1, 0)\|^2 + \|\min(\boldsymbol{I}^*, 0)\|^2$
  $\mathcal{L}_{\text{ps}} \leftarrow \|\max(\boldsymbol{I} - \boldsymbol{I}_+, 0)\|^2 + \|\min(\boldsymbol{I} - \boldsymbol{I}_-, 0)\|^2$
  $g \leftarrow \nabla_{\theta'}(\mathcal{L}_{\text{pc}} + \mathcal{L}_{\text{cs}} + \mathcal{L}_{\text{ps}})$ ▷ `minimize PSA loss`
  $\theta' \leftarrow \theta' + \beta * \frac{g}{\|g\|_2}$
while $\mathcal{L}_{pc} + \mathcal{L}_{cs} + \mathcal{L}_{ps} > t$
return $\boldsymbol{I}'^* \leftarrow \pi^\dagger(f_{\theta'}(s, T))$

We then propose a gradient-based iterative refinement method to optimize the language guided neural style transfer model parameters θ. This ensures that the stylized surface $\boldsymbol{I}'$ aligns with the input language guidance and remains compensable:

$$\theta' = \underset{\theta}{\text{argmin}}(\mathcal{L}_{\text{pc}} + \mathcal{L}_{\text{cs}} + \mathcal{L}_{\text{ps}}) \tag{11}$$

$$\boldsymbol{I}' = f_{\theta'}(s, T) \tag{12}$$

$$\boldsymbol{I}'^* = \pi^\dagger(\boldsymbol{I}') \tag{13}$$

Finally, the compensated surface stylization image $\boldsymbol{I}'^*$ is projected for surface stylization. The termination criterion for the gradient descent-based optimization in Eqn. 11 is determined by a loss threshold t. The detailed PSA algorithm is shown in Algorithm 1.

3.3 Project-and-capture simulation and compensation

The designs of the project-and-capture and the projector compensation models in Eqn. 2 and Eqn. 3 are crucial to our LAPIG, as their

simulation accuracy of the real project-and-capture and compensation processes directly affects the final surface adaptation quality and viewer's perceived effects.

Inspired by the previous study [26] that the project-and-capture process π can be modeled by a neural network named PCNet ($\hat{\pi}$), we modify PCNet [26] by removing the direct light mask requirement and applying it to our project-and-capture simulation, as shown in the right block of Fig. 3. In particular, PCNet utilizes a dual-branch encoder-decoder architecture to capture complex photometric transformations, allowing it to model the photometric changes introduced by the projection. By this characteristic, PCNet can accurately simulate the real-world project-and-capture process.

For the projector compensation network $\pi^{\dagger}$, we modified the photometric compensation subnet (CompenNeSt) of the full projector compensation model CompenNeSt++ [27]. Specifically, CompenNeSt can correct distortions in color and brightness. As depicted in the right section of Fig. 3, it utilizes a siamese encoder-decoder setup with two branches addressing the surface and projected images. The encoder (red blocks) in each branch shares weights and extracts features from the surface and projected images, merging them through skip connections. This alignment and combination enable the decoder (blue blocks) to reconstruct an image that compensates for photometric distortions such as color intensity, contrast, and brightness variances.

The effectiveness of the projector compensation network $\pi^{\dagger}$ and the project-and-capture simulation model $\hat{\pi}$ is shown in § 4.3 and § 4.4.

3.4 Training details

We implemented PSA using PyTorch [46] and optimized it using Adam optimizer [37]. For LGST, we used the pre-trained weights of InstructPix2Pix [11] and optimized its parameters with an initial learning rate of $\beta = 0.001$, and it decayed by a factor of 5 for every 50 iterations. For the compensation network *i.e.*, CompenNeSt $\pi^{\dagger}$ in PSA, the hyperparameters for the training model are: learning rate of 0.001, batch size of 8, and 800 iterations, which took about 3.5 minutes to finish training on an Nvidia GeForce RTX 4060 laptop GPU. For the project-and-capture simulation network *i.e.*, PCNet $\hat{\pi}$, the training hyperparameters are as follows: the learning rate is 0.001 with 1,500 training iterations. For training sizes of 8 and 48, the batch size is set to 8, and for a training size of 125, it is 24. Training on an Nvidia GeForce RTX 3090 GPU was completed in approximately 1 minute.

4 Experiments

4.1 System configuration

Our ProCams consists of a Canon 600D camera and an EPSON EB-C2050WN projector and their resolutions are 1280×720 and 640×480, respectively. Considering projector-camera synchronization, we manually set a 130-ms delay between projection and the capture operation, and in total it takes about 200 ms to capture a frame. We deploy LAPIG on a laptop with Intel Core i7-13650HX CPU, 16GB memory, and an Nvidia GeForce RTX 4060 Laptop GPU for both surface stylization and ProCams control.

4.2 Datasets

Our PSA involves both projector compensation $\pi^{\dagger}$ and project-and-capture simulation $\hat{\pi}$. Although there exist some datasets for projector compensation or project-and-capture simulation [25, 27, 41, 59], to the best of our knowledge, there is no public dataset to evaluate both methods simultaneously. Therefore, we captured a real dataset with 10 setups with real compensation images generated by [27]. In each setup, at least one of the texture, material, and surface geometry is different. Each setup has 180 image pairs for training and 20 for testing.

To improve the practicability of PSA, we also built a synthetic dataset using Blender [14] for model pre-training. Similarly to the real dataset, we also rendered 10 synthetic setups. For each setup, we provided 1,000 pairs of images for training and 200 for testing. Note that both the surface textures and the projector input differ from those in the real dataset. Some examples of images from our dataset are presented in Fig. 4. Our real and synthetic datasets can be leveraged in future work for model pre-training and network architecture exploration.

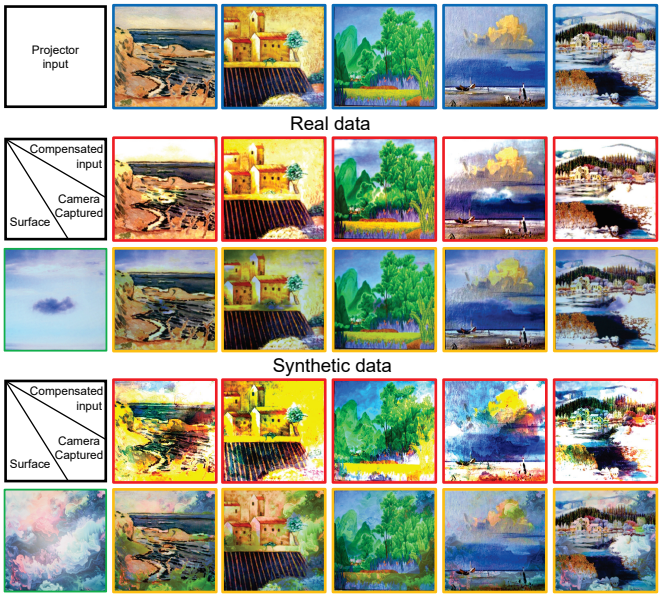

Fig. 4: Real and synthetic dataset. The 1st row shows five different projector input images. The 2nd row is the compensated images of the 1st row, given the surface in the 1st column of the 3rd row. The 2nd to the 4th columns of the 3rd row are the corresponding camera-captured projection of the 2nd row. Similarly, the last two rows are Blender synthesized images for another surface.

4.3 Simulated captured compensation

Since the accuracy of the real project-and-capture simulation $\hat{\pi}$ and the compensation model $\pi^{\dagger}$ directly affects the final surface adaptation quality and the viewer's perceived effects, we quantitatively and qualitatively evaluate the accuracy of the simulation by comparing the simulated camera-captured compensation, *i.e.*, $\hat{\boldsymbol{I}}^{*}$ (Eqn. 10) with the real camera-captured compensation, *i.e.*, $\tilde{\boldsymbol{I}}^{*}$ (Eqn. 6). The results are shown in Table 1 and Fig. 5.

To explore different project-and-capture models, we adapted CompenNeSt [27] to simulate the project-and-capture process by swapping its input and output, and we name it **CompenNeSt reversed**. The advantage of our PCNet over this adapted CompenNeSt model is demonstrated by comparing the two methods in Table 1 and Fig. 5. Clearly, our PCNet outperforms CompenNeSt reversed by a great margin in PSNR, RMSE, and SSIM, especially when the number of training images is smaller, *e.g.*, # Train = 8.

4.4 Comparison of different PCNet training losses

The loss function for PCNet training is crucial for our LAPIG performance. Although pixel-wise l_1 and l_2 losses are commonly used to penalize pixel errors in various image reconstruction tasks, in [25], Huang *et al*. demonstrated that SSIM loss effectively recovers structural details in image compensation tasks. The performance of different combinations of the three loss functions is shown in Table 3.

In summary, l_1 and l_2 losses alone produce suboptimal outcomes. Using l_2 loss alone may result in loss of detail, leading to lower visual quality compared to using l_1 loss. The combination of l_1, l_2, and SSIM losses yields the best performance during training, likely due to the balanced optimization it provides between fine-detail preservation, global structure alignment, and robustness to outliers. l_1 loss helps to retain pixel-wise fine details, l_2 loss suppresses large color errors, and SSIM loss improves structural details.

Given the potential biases and convergence issues associated with the use of individual loss functions [62], we employed a combined loss of l_1 and l_2 during the early training stages to improve convergence, and changed to $l_1 + l_2 +$ SSIM to further improve fine details and structural details.

Table 1: Quantitative evaluation of simulated captured compensation. Results are averaged over 10 real setups.

	# Train = 8			# Train = 48			# Train = 125		
Model	**PSNR ↑**	**RMSE ↓**	**SSIM ↑**	**PSNR ↑**	**RMSE ↓**	**SSIM ↑**	**PSNR ↑**	**RMSE ↓**	**SSIM ↑**
PCNet	**26.7746**	**0.0805**	**0.8782**	**30.7357**	**0.0505**	**0.9268**	**31.1363**	**0.0483**	**0.9304**
CompenNeSt reversed	25.7832	0.0903	0.8327	29.9323	0.0554	0.9089	30.5871	0.0514	0.9161

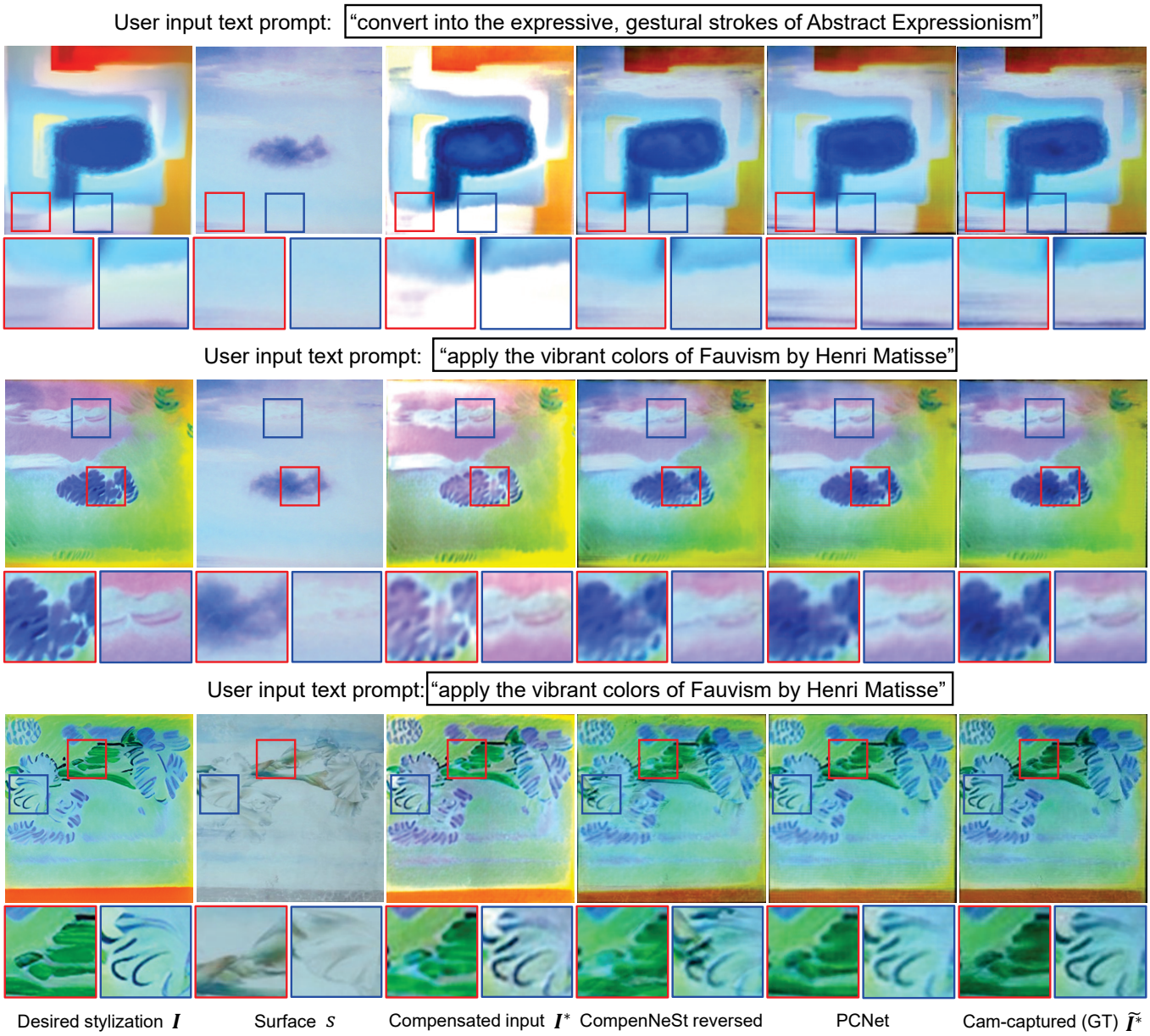

Fig. 5: Qualitative evaluation of simulated captured compensation. We provide three scenes in three rows, and each scene is under different projections generated by LGST. The first two rows share the same projection surface, but have different text prompts, while the last two rows share the same text prompt, but have different surface. The user input text prompts to stylize the surface are shown on the top of each row. The 1st column is the desired viewer perceived effects, *i.e.*, the 2nd column stylized by LGST. The 2nd column shows the original projection surfaces. The 3rd column displays the compensated result of the 1st column for projection. The 4th and the 5th columns are simulated captured projection by two methods. The last column presents real captured projection, *i.e.*, the 3rd column projected onto the 2nd column. Comparing the 4th and 5th columns with the final column, PCNet demonstrates superior performance in simulating the project-and-capture process, surpassing CompenNeSt reversed in both color and texture accuracy. Each image is provided with two zoomed-in patches for detailed comparison. More results are provided in **supplementary**.

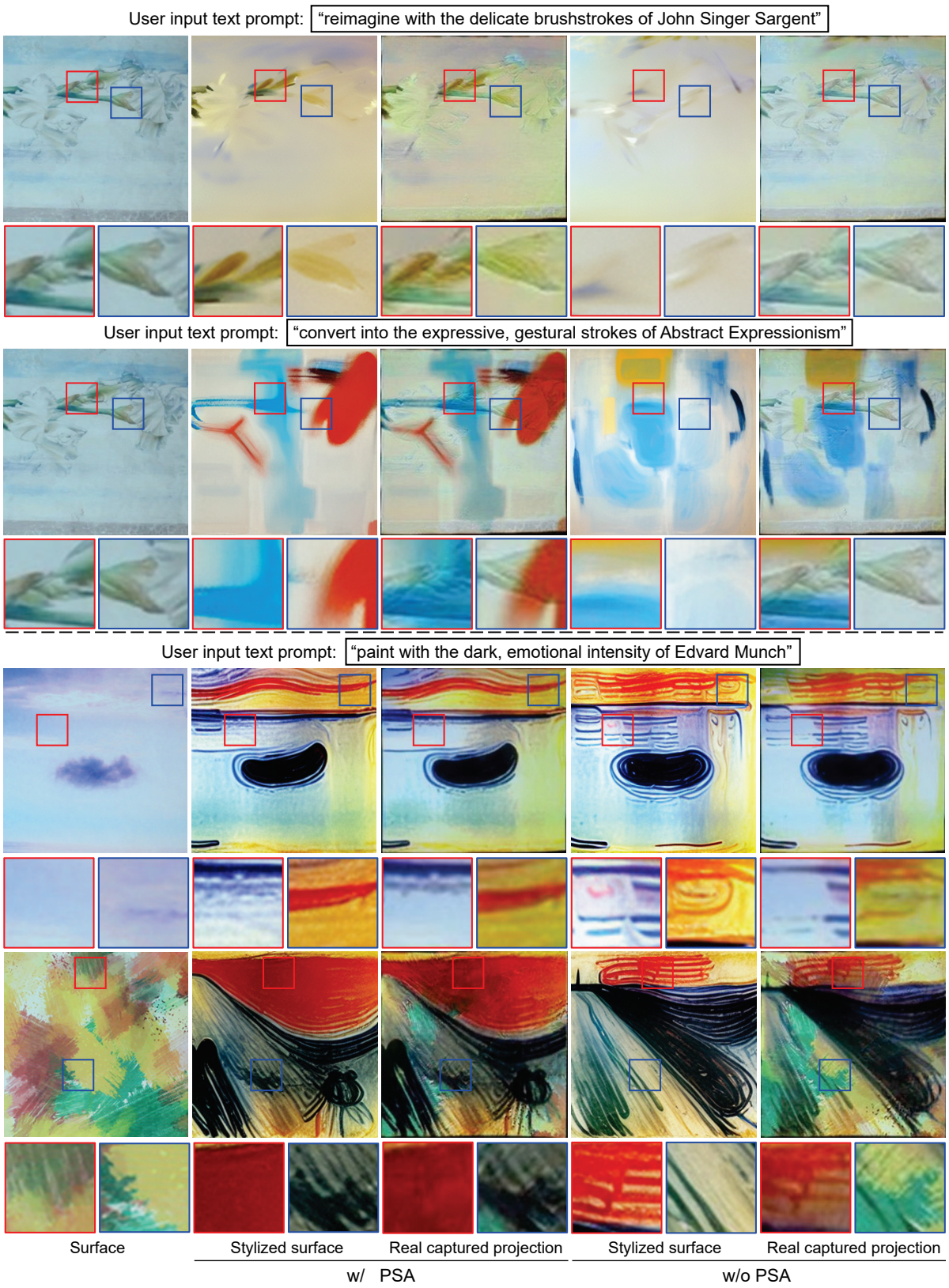

Fig. 6: Qualitative comparison between language guided projector image generation (LAPIG) w/ and w/o PSA. The first two rows show stylization of the *same projection surface* using two *different text prompts*, while the last two rows show stylization of *two different projection surfaces* using the *same text prompt*. The 1st column shows the original projection surfaces. The 2nd and 4th columns are stylized projection surfaces by LAPIG w/ or w/o PSA, given the user input text prompt on the top of each surface(s). The 3rd and 5th columns present real captured scenes under projection w/ or w/o PSA, *i.e.*, the 2nd or 4th column projected onto the 1st column after projector compensation. Clearly, our PSA can adapt the stylization to the surface by showing fewer discrepancies between the target stylized surface (the 2nd column) and the real captured projection (the 3rd column). See **supplementary** for more results.

Table 2: Quantitative comparison between language guided projector image generation (LAPIG) w/ and w/o PSA. Each evaluation involved selecting 5 different textured projection surfaces in the 1st column (images are shown in the bottom), then applying 10 same text prompts to guide surface stylization, and averaging the results from 50 trials.

	PSNR ↑		RMSE ↓		SSIM ↑	
Surface	w/ PSA	w/o PSA	w/ PSA	w/o PSA	w/ PSA	w/o PSA
Bench	**23.1286**	20.6997	**0.0728**	0.0980	**0.7013**	0.6910
Wood	**22.2555**	21.7816	**0.0789**	0.0860	**0.6869**	0.6599
Stripe	**20.1848**	18.5286	**0.1029**	0.1246	**0.7387**	0.7316
Sakura	**15.8081**	14.4061	**0.1672**	0.1964	**0.3942**	0.3772
Spray	**23.4177**	22.0880	**0.0708**	0.0825	**0.6654**	0.6228
Average	**20.9589**	19.5008	**0.0985**	0.1175	**0.6373**	0.6165

Table 3: Comparison of different PCNet training losses. The results are averaged over 5 real setups and 5 synthetic setups.

PCNet training loss	PSNR ↑	RMSE ↓	SSIM ↑
l_1	29.3504	0.0597	0.8783
l_2	29.2355	0.0603	0.8848
SSIM	32.0635	0.0435	0.9485
$l_1 + l_2$	32.4053	0.0418	0.9454
l_1+ SSIM	33.2556	0.0379	0.9586
l_2+ SSIM	33.7120	0.0360	0.9643
$l_1 + l_2$+ SSIM	**34.0953**	**0.0345**	**0.9662**

4.5 Effectiveness of PSA

To show the effectiveness of our PSA, we compared the stylized surfaces generated w/ and w/o PSA, and their corresponding real captured projections. The quantitative and qualitative results are shown in Table 2 and Fig. 6, respectively. Clearly, PSA can adapt the stylization to the surface by showing fewer discrepancies between the target stylized surface (the 2nd column) and the real captured projection (the 3rd column), due to the projection consistency and color saturation losses (Eqn. 7 to Eqn. 9). Comparing the 1st and 2nd columns with the 3rd and 4th columns of Fig. 6, it is obvious that our LAPIG (w/ PSA) tends to generate image content that fits/adapts the projection surfaces, *e.g.*, generates dark stylization around the hard-to-compensate dark surface regions, while without PSA, the LGST generated stylization tends to ignore these hard-to-compensate regions, resulting in artifacts.

The effectiveness of PSA is also demonstrated in Table 4, where w/ PSA shows superior PSNR, RMSE and SSIM compared to w/o PSA on 5 different surfaces with 10 different text prompts.

4.6 Effectiveness of different PSA losses

We evaluated the effectiveness of different PSA losses in Eqn. 7 to Eqn. 9. Specifically, we progressively removed or used a combination of projection consistency loss $\mathcal{L}_{\text{pc}}$, projection saturation loss $\mathcal{L}_{\text{ps}}$, and compensation saturation loss $\mathcal{L}_{\text{cs}}$. The quality of the stylized surfaces was evaluated by PSNR, RMSE, SSIM, and the average number of optimization iterations required for convergence (Mean iter.). The results are shown in Table 4, compared with w/o PSA, it is clear that projection saturation loss, projection consistency loss and compensation saturation loss can improve the quality of the stylized image. In particular, the compensation saturation loss and projection saturation loss can significantly improve convergence, *e.g.*, Mean iter. drops drastically from 86 to around 20. Moreover, combining all three losses achieves a balance between surface stylization quality and convergence. Note that besides LAPIG, the three losses can also be applied to general projection mapping applications, such as ProCams simulation and projector compensation, to improve both quality and convergence.

Table 4: Comparisons of different PSA losses. Each loss evaluation involved selecting 3 different textured projection surfaces, then performing 5 same text prompts guided surface stylization, and averaging the results from 15 trials.

PSA loss	PSNR ↑	RMSE ↓	SSIM ↑	Mean iter.
$\mathcal{L}_{\text{pc}}$	19.9299	0.1038	0.7447	86
$\mathcal{L}_{\text{cs}}$	**20.6526**	0.0948	0.7433	19
$\mathcal{L}_{\text{ps}}$	20.3378	0.0978	0.7487	**17**
$\mathcal{L}_{\text{ps}} + \mathcal{L}_{\text{cs}}$	20.5002	0.0958	0.7488	**17**
$\mathcal{L}_{\text{pc}} + \mathcal{L}_{\text{ps}} + \mathcal{L}_{\text{cs}}$	20.5831	**0.0948**	**0.7509**	21
w/o PSA	18.8891	0.1158	0.7149	0

5 Discussion

5.1 Applicability to other settings

For museums and galleries, LAPIG can be used to design the surfaces of exhibited objects in an artistic way that enhances the user experience. For LGST, we use InstructPix2Pix [11], which is based on the stable diffusion model [55]. In theory, other text-guided image-to-image models are also applicable to our LAPIG.

5.2 Limitations and future work

Failure cases of our LAPIG are shown in Fig. 7. To address these issues, future work could focus on improving the accuracy of project-and-capture simulation and projector compensation techniques, along with designing and training LGST specifically for projection mapping applications.

Low resolution. To achieve fast performance in our LAPIG tasks, we currently process images at low resolutions. Although our LAPIG can be directly applied to projectors and cameras with higher resolutions without modifying the neural network architecture, this could lead to

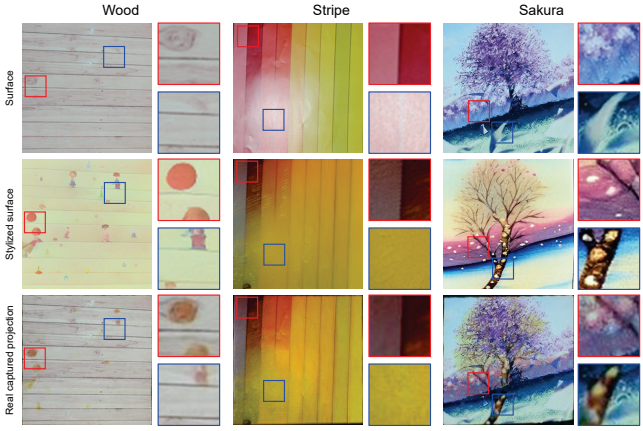

Fig. 7: Failure cases of our LAPIG. Three user input text prompts are used to stylize three projection surfaces: **Render it as a whimsical children's book illustration** for the *Wood* surface, and **Render it in the style of a Van Gogh oil painting** for the *Stripe* surface, and **Imagine it as a traditional Indian miniature painting** for the *Sakura* surface, respectively. The 1st row shows the original projection surfaces. The 2nd row is stylized projection surfaces by LAPIG. The 3rd row presents real captured scenes under projection, *i.e.*, the 2nd row projected onto the 1st row after projector compensation. Note that for *Wood* and *Sakura*, the real captured projections of in the 3rd look different from the 2nd row, indicating failed projection surface adaptation. For *Stripe*, although the 2nd and 3rd rows look similar, but the LGST generated stylization failed to represent "Van Gogh oil painting style".

increased processing times and GPU memory. Therefore, efficient network architectures, *e.g.*, CompenHR [59] may be applied to address this problem.

Basic language guidance. The current LAPIG framework is limited to basic text interaction and applying a global style transfer to the entire projection surface. Occasionally, it also minimally and locally adapts the stylization to the surface with slight transformations (Fig. 7). In addition, real-world use cases frequently include complex user interactions that integrate visual, auditory, and tactile modalities. Expanding LAPIG to accommodate multimodal interactions will align it more closely with the user's intention, resulting in more intuitive and engaging experiences.

User study. Our LAPIG lacks a direct assessment of viewer perceptual satisfaction. Future work will incorporate thorough user studies to assess and refine LAPIG.

Complex scenes. Our LAPIG was not tested in a wide variety of scenarios, and the current approach focuses mainly on modifying projection surface textures. However, real-world conditions typically involve more complexity, such as different depths, geometries, and lighting conditions. In future work, we plan to expand our method to tackle more complex scenes, where the difficult shape and indirect light conditions are involved. Extending LAPIG to dynamic projection mapping is also an interesting direction to explore.

6 Conclusion

In this paper, we introduced LAPIG, a novel language guided projector image generation approach for surface adaptation and stylization. Our LAPIG framework can generate a projector image that follows the user's instruction while reducing projection surface artifacts, providing support for extended applications in various fields such as artistic and animated projection mapping. We also introduce projection consistency and color saturation losses to guide projection surface adaptation (PSA), and they are expected to facilitate other projection mapping applications.

Acknowledgments

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

References

[1] R. Akiyama, G. Yamamoto, T. Amano, T. Taketomi, A. Plopski, C. Sandor, and H. Kato. Robust reflectance estimation for projection-based appearance control in a dynamic light environment. *IEEE TVCG*, 27(3):2041–2055, 2021. 1

[2] T. Amano. Manipulation of material perception with light-field projection. *Three-Dimensional Imaging, Visualization, and Display 2019*, 10997:1099706:1–1099706:13, 2019. 1, 2

[3] R. Asahina, T. Nomoto, T. Yoshida, and Y. Watanabe. Realistic 3d swept-volume display with hidden-surface removal using physical materials. In *IEEE VR*, pp. 113–121, 2021. 2

[4] H. Asayama, D. Iwai, and K. Sato. Fabricating diminishable visual markers for geometric registration in projection mapping. *IEEE TVCG*, 24(2):1091–1102, 2018. 1

[5] O. Avrahami, T. Hayes, O. Gafni, S. Gupta, Y. Taigman, D. Parikh, D. Lischinski, O. Fried, and X. Yin. Spatext: Spatio-textual representation for controllable image generation. In *CVPR*, pp. 18370–18380, 2022. 3

[6] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic shader lamps: painting on movable objects. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pp. 207–216, 2001. 1

[7] A. H. Bermano, M. Billeter, D. Iwai, and A. Grundhöfer. Makeup lamps: Live augmentation of human faces via projection. *Computer Graphics Forum*, 36(2):311–323, 2017. 1, 2

[8] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann, and T. Langlotz. Superimposing pictorial artwork with projected imagery. In *ACM SIGGRAPH 2005 Courses*, p. 6–es. ACM, 2005. 1, 2

[9] O. Bimber, A. Emmerling, and T. Klemmer. Embedded entertainment with smart projectors. *Computer*, 38(1):48–55, 2005. 1, 2

[10] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. *Computer Graphics Forum*, 27:2219–2245, 2008. 1, 2

[11] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pp. 18392–18402, 2023. 3, 4, 5, 8

[12] G. Cascini, J. O'Hare, E. Dekoninck, N. Becattini, J.-F. Boujut, F. Ben Guefrache, I. Carli, G. Caruso, L. Giunta, and F. Morosi. Exploring the use of ar technology for co-creative product and packaging design. *Computers in Industry*, 123:103308, 2020. 1

[13] N. Chiba and K. Hashimoto. Ultra-fast multi-scale shape estimation of light transport matrix for complex light reflection objects. In *IEEE ICRA*, pp. 6147–6152, 2018. 2

[14] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2024. 5

[15] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, p. 145–156. ACM Press/Addison-Wesley Publishing Co., 2000. 2

[16] X. Dong, H. Ling, and B. Huang. Adaptive color structured light for calibration and shape reconstruction. In *IEEE International Symposium on Mixed and Augmented Reality*, pp. 1240–1249, 2023. 1

[17] Y. Erel, D. Iwai, and A. H. Bermano. Neural projection mapping using reflectance fields. *IEEE TVCG*, 29(11):4339–4349, 2023. 1, 2

[18] Y. Erel, O. Kozlovsky-Mordenfeld, D. Iwai, K. Sato, and A. H. Bermano. Casper dpm: Cascaded perceptual dynamic projection mapping onto hands. In *SIGGRAPH Asia 2024 Conference Papers*. ACM, 2024. 2

[19] M. Flagg and J. M. Rehg. Projector-guided painting. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, p. 235–244. ACM, 2006. 1, 2

[20] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 3

[21] A. Grundhöfer and D. Iwai. Robust, error-tolerant photometric projector compensation. *IEEE TIP*, 24(12):5086–5099, 2015. 1, 2

[22] A. Grundhöfer and D. Iwai. Recent advances in projection mapping algorithms, hardware and applications. In *Computer graphics forum*, pp. 653–675. Wiley Online Library, 2018. 1, 2

[23] K. Hiratani, D. Iwai, Y. Kageyama, P. Punpongsanon, T. Hiraki, and K. Sato. Shadowless projection mapping using retrotransmissive optics. *IEEE TVCG*, 29(5):2280–2290, 2023. 1

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

[25] B. Huang and H. Ling. Deprocams: Simultaneous relighting, compensation and shape reconstruction for projector-camera systems. *IEEE TVCG*, 27(5):2725–2735, 2021. 2, 3, 5

[26] B. Huang and H. Ling. Spaa: Stealthy projector-based adversarial attacks on deep image classifiers. In *IEEE VR*, pp. 534–542, 2022. 5

[27] B. Huang, T. Sun, and H. Ling. End-to-end full projector compensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 3, 5

[28] 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 TIP*, 26(1):147–159, 2017. 2

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

[30] D. Iwai. Projection mapping technologies: A review of current trends and future directions. *Proceedings of the Japan Academy, Series B*, 100(3):234–251, 2024. 2

[31] D. Iwai and K. Sato. Limpid desk: see-through access to disorderly desktop in projection-based mixed reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, p. 112–115. ACM, 2006. 1, 2

[32] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[33] S. Kagami and K. Hashimoto. Animated stickies: Fast video projection mapping onto a markerless plane through a direct closed-loop alignment. *IEEE TVCG*, 25(11):3094–3104, 2019. 1, 2

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

[35] Y. Kageyama, D. Iwai, and K. Sato. Efficient distortion-free neural projector deblurring in dynamic projection mapping. *IEEE TVCG*, 30(12):7544–7557, 2024. 1

[36] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3

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

[38] P. Kurth, V. Lange, C. Siegl, M. Stamminger, and F. Bauer. Auto-calibration for dynamic multi-projection mapping on arbitrary surfaces. *IEEE TVCG*, 24(11):2886–2894, 2018. 1

[39] P. Kurth, V. Lange, M. Stamminger, and F. Bauer. Real-time adaptive color correction in dynamic projection mapping. In *IEEE International Symposium on Mixed and Augmented Reality*, pp. 174–184, 2020. 1

[40] H. Kusuyama, Y. Kageyama, D. Iwai, and K. Sato. A multi-aperture coaxial projector balancing shadow suppression and deblurring. *IEEE TVCG*, pp. 1–11, 2024. 2

[41] Y. Li, W. Yin, J. Li, and X. Xie. Physics-based efficient full projector compensation using only natural images. *IEEE TVCG*, 30(8):4968–4982, 2024. 1, 2, 5

[42] K. Luo, G. Yang, W. Xian, H. Haraldsson, B. Hariharan, and S. Belongie. Stay positive: Non-negative image synthesis for augmented reality. In *CVPR*, pp. 10050–10060, 2021. 2

[43] M. R. Marner, R. T. Smith, J. A. Walsh, and B. H. Thomas. Spatial user interfaces for large-scale projector-based augmented reality. *IEEE Computer Graphics and Applications*, 34(6):74–82, 2014. 1, 2

[44] K. Matsushita, D. Iwai, and K. Sato. Interactive bookshelf surface for in situ book searching and storing support. In *Proceedings of the 2nd Augmented Human International Conference*. ACM, 2011. 1, 2

[45] Y. Miyatake, T. Hiraki, D. Iwai, and K. Sato. Haptomapping: Visuo-haptic augmented reality by embedding user-imperceptible tactile display control signals in a projected image. *IEEE TVCG*, 29(4):2005–2019, 2023. 1, 2

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

[47] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, pp. 2085–2094, 2021. 3

[48] H.-L. Peng and Y. Watanabe. High-speed human arm projection mapping with skin deformation. In *SIGGRAPH Asia 2020 Emerging Technologies*. ACM, 2020. 2

[49] P. Pjanic, S. Willi, D. Iwai, and A. Grundhöfer. Seamless multi-projection revisited. *IEEE TVCG*, 24(11):2963–2973, 2018. 1

[50] P. Punpongsanon, D. Iwai, and K. Sato. Softar: Visually manipulating haptic softness perception in spatial augmented reality. *IEEE TVCG*, 21(11):1279–1288, 2015. 2

[51] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021. 3

[52] 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. 2

[53] M. D. M. Reddy, M. S. M. Basha, M. M. C. Hari, and M. N. Penchalaiah. Dall-e: Creating images from text. *UGC Care Group I Journal*, 8(14):71–75, 2021. 3

[54] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, vol. 48, pp. 1060–1069. PMLR, 2016. 3

[55] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022. 8

[56] K. Sato, H. Terashima, S. Nishida, and Y. Watanabe. E.s.p.: Extra-sensory puck in air hockey using the projection-based illusion. In *SIGGRAPH Asia 2022 Emerging Technologies*. ACM, 2022. 2

[57] M. Tao, B. Bao, H. Tang, and C. Xu. Galip: Generative adversarial clips for text-to-image synthesis. In *CVPR*, pp. 14214–14223, 2023. 3

[58] J. Wang, Y. Dong, X. Tong, Z. Lin, and B. Guo. Kernel nyström method for light transport. In *SIGGRAPH*. ACM, 2009. 2

[59] Y. Wang, H. Ling, and B. Huang. Compenhr: Efficient full compensation for high-resolution projector. In *IEEE VR*, pp. 135–145, 2023. 1, 2, 5, 9

[60] Y. Wang, H. Ling, and B. Huang. Vicomp: Video compensation for projector-camera systems. *IEEE TVCG*, 30(5):2347–2356, 2024. 2

[61] M. Yasui, R. Iwataki, M. Ishikawa, and Y. Watanabe. Projection mapping with a brightly lit surrounding using a mixed light field approach. *IEEE TVCG*, 30(5):2217–2227, 2024. 1, 2

[62] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE TCI*, 3(1):47–57, 2017. 5

[63] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pp. 2242–2251, 2017. 3