Single Image Reflection Removal with Physically-Based Training Images

Soomin Kim Yuchi Huo Sung-Eui Yoon
Korea Advanced Institute of Science and Technology (KAIST)

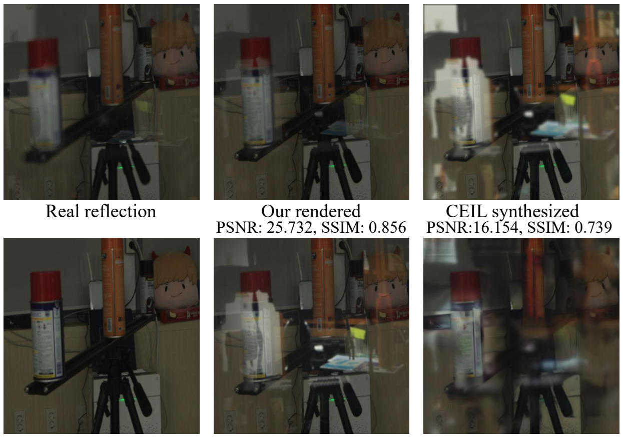
Abstract

Recently, deep learning-based single image reflection separation methods have been exploited widely. To benefit the learning approach, a large number of training image-pairs (i.e., with and without reflections) were synthesized in various ways, yet they are away from a physically-based direction. In this paper, physically based rendering is used for faithfully synthesizing the required training images, and a corresponding network structure and loss term are proposed. We utilize existing RGBD/RGB images to estimate meshes, then physically simulate the light transportation between meshes, glass, and lens with path tracing to synthesize training data, which successfully reproduce the spatially variant anisotropic visual effect of glass reflection. For guiding the separation better, we additionally consider a module, backtrack network (BT-net) for backtracking the reflections, which removes complicated ghosting, attenuation, blurred and defocused effect of glass/lens. This enables obtaining a priori information before having the distortion. The proposed method considering additional a priori information with physically simulated training data is validated with various real reflection images and shows visually pleasant and numerical advantages compared with state-of-the-art techniques.

1. Introduction

When taking a photo through a glass or a window, the front scene that is transmitted through the glass can be seen, but sometimes the reflection from the back scene is captured as well. These inevitable reflections and dim transmission can be annoying for some cases, for example, a case of taking a photo of a skyscraper from an indoor room. As a result, removing the reflections from the input images can help us to generate better images and various computer vision techniques to work robustly.

Physically, an image I with those reflections is a sum of the glass reflected back scene, $\tilde{R}$, and the glass transmitted front scene, $\tilde{T}$, as $I(x, y) = \tilde{T}(x, y) + \tilde{R}(x, y)$. Single image reflection removal problem is ill-posed, without using additional assumptions or priors.



Real reflection

Our rendered
PSNR: 25.732, SSIM: 0.856

CEIL synthesized
PSNR:16.154, SSIM: 0.739

Front scene

Zhang synthesized
PSNR:19.532, SSIM: 0.786

Wen synthesized
PSNR:22.085, SSIM: 0.827

Figure 1: Comparison between existing reflection synthesizing methods and our physically-based rendering method. The real reflection image is captured by a camera behind a glass. Our method can produce spatially variant visual effects that are most similar to the real-world reflection image. For example, the near bottle is blurred and the far bottle is focused in the transmission scene. Also, the reflection level of some back scene objects is properly considered. In contrast, previous methods assume the glass transmitted front scene is all clear, and the reflected back scene is spatially invariantly blurred, introducing biased information to the dataset. Sec. 6.1 has more details.

Previous methods utilize multiple images of reflection with different conditions for obtaining some priors [1, 23, 14, 20]. Especially, motion cue prior is widely used for separating the reflections from multi-images [8, 33, 9]. Although multiple-image reflection separation methods show reasonable results, it is not easy for users to capture constrained images as suggested in the prior approaches.

For single image reflection removal, natural image priors [16, 17, 18] or smoothness priors [19, 30] are used for formulating objective functions. Recent approaches started to utilize deep neural networks for removing the reflections on a single image. While training deep neural networks relies on a faithful dataset, most up-to-date methods synthesize datasets in an image space through a weighted addition between the front scene and the back scene [7, 36, 29, 34, 31], due to the difficulty of the physical simulation of the reflection and the transmission phenomena. Recently, Wen et al.[32] propose a method that generates reflection training images using deep learning architecture. However, these image-space methods ignore the physical fact that the visual effects of reflections are spatially variant, depending on the 3D positions of the visible points. Figure 1 shows the visual and numerical comparison of generated reflection images against the ground truth (Sec. 6.1).

In this paper, we present a data generation method to synthesize physically faithful training data. The method is based on modeling and rendering techniques, such as depth estimation, geometry synthesizing, and physically-based rendering. We utilize such physically-based training images, including the transmission and the reflection with or without glass/lens-effects, i.e., the attenuation, defocusing, blurring, and ghosting effects related to passing through a glass/camera lens (Sec. 4), for training our deep learning architectures. Especially, we train a backtrack network (BT-net) to fetch a priori information to improve separation quality.

In summary, our contributions are as follows:

- Propose a synthesizing method to physically render a faithful reflection image dataset for training.
- Use BT-net to transform the reflection image back to its prior-distortion status as a priori information of the separation problem.

2. Related Work

Single image-based methods with conventional priors. Since the single image methods lack information compared to the multi-image methods, they assume predefined priors. One of the widely used priors is the natural image gradient sparsity priors [17, 18]. These approaches decompose the layers with minimal gradients and local features. Levin et al. [16] proposed gradient sparsity priors with user labeling and showed reasonable results. Another widely used assumption is that reflection layers are more likely to be blurred because of the different distance to the camera [19, 30]. In addition to that, Arvanitopoulos et al. [2] proposed the Laplacian fidelity term and l_0-gradient sparsity term to suppress reflections. Shih et al. [24] suggested to examine ghosting effects on the reflection and model them by Gaussian Mixture Models (GMM) patch prior.

Single image based methods with deep learning. Recent studies start to adopt deep learning for the reflection removal problem. Fan et al. [7] proposed a two-step deep architecture utilizing edges of the image. Zhang et al. [36] adopt conditional GAN [11] with a combination of perceptual loss, adversarial loss, and exclusion loss for separating reflection. Wan et al. [29] suggested a concurrent deep learning-based framework for gradient inference and image inference. Yang et al. [34] proposed a cascaded deep network for estimating both the transmission and the reflection. Wei et al. [31] suggest to use misaligned real images for training and its corresponding loss term, and Wen et al. [32] proposes a learning architecture to produce reflection training images with a corresponding removal network.

Our method is also based on learning-based single image reflection removal, but with two main differentiations. First, we render a physically faithful dataset to reproduce lens focus and glass-effect realistically. These spatially variant anisotropic visual effects vary depending on the depth and viewing angle across the image space and were not supported faithfully by previous image-space data generation methods. Second, our method utilizes information not only after the images were distorted by the glass/lens (a posteriori information), but also before the glass/lens distortion (a priori information), to get better separation results.

Synthesizing training datasets with rendering. Monte Carlo (MC) rendering is widely used in various applications for high-quality image synthesis. Its theoretical foundation includes the physical simulation of light transportation and the unbiased integration of incident radiances [35]. In order to simulate the shading effect of complex geometry details, displacement mapping is proposed to reconstruct geometry from a depth map [6]. Because physically-based rendering can faithfully simulate the physical process of light transportation, it has been proven to be a promising way to synthesize deep learning datasets for various computer vision problems [37, 26, 21].

In this paper, we propose to use displacement mapping and path tracing to synthesize a physically plausible dataset for the reflection removal problem.

3. Overview

In this section, we present an overview of our method. There are two main components of our reflection removal technique. The first part is synthetically generating training images with physically-based rendering, and the second part is network training using the rendered training images as additional priori information.

To train the reflection removal network, a large amount of reflection and reflection-free image pairs are necessary. It is, however, quite troublesome to obtain such kinds of many image pairs. Most of the prior deep learning-based reflection removal methods [7, 36, 29, 34, 31] synthesize a reflection image by mixing two ordinary images, one as a reflection and another as a transmission, with different coefficients followed by applying Gaussian blurring and scaling down the brightness of the reflection. The technical details vary from one to the other, but they synthesize the reflection images in image space. Lately, Wen et al. [32] suggested to

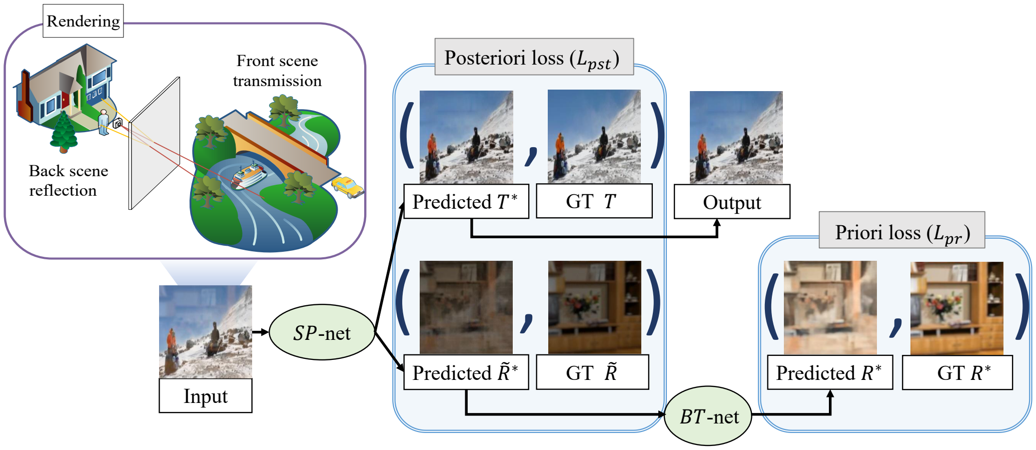


Figure 2: Overview of our method structure. From a given image with reflection (I), our SP-net first separates I to predicted front scene transmission, T^*, and back scene reflection with glass-effects, $\tilde{R}^*$. A posteriori loss (L_{pst}) is calculated with each of the predicted values and its ground truth. Our trained backtrack network, BT-net, removes the glass and lens effects of the predicted $\tilde{R}^*$ into R^*. Since R^* is released from complicated glass/lens-effects, we can better capture various image information, resulting in clearer error matching between the predicted image and its ground truth. To utilize this information, we use a new loss, a priori loss (L_{pr}), between R^* and its ground truth (GT). The entire separation network is trained with a loss combination of L_{pst} and L_{pr}.

use a network for generating reflection training image pairs, but still, they do not consider the spatially variant visual effects.

We find that instead of synthesizing the reflection images in the image space, rendering the reflection images in a 3D space would produce more realistic images for training, resulting in a higher removal accuracy. In order to achieve a physically faithful dataset, we adopt a series of modeling and rendering techniques, i.e., depth estimation, geometry synthesizing, and physically-based rendering technique (path tracing [13]).

From existing DIODE RGBD dataset [27] and PLACES365 RGB dataset [38], we randomly choose one image as a **front scene** transmission layer (*the side in front of the camera)* and another image as a **back scene** reflection layer (*the side behind the camera)*. With one front scene and one back scene as a **scene** setup, we extract the 3D model of the scene with depth and then render it with path tracing to synthesize a group of images with or without reflection for training; for the RGB dataset, we apply depth estimation [4] to extract the 3D model of the scene.

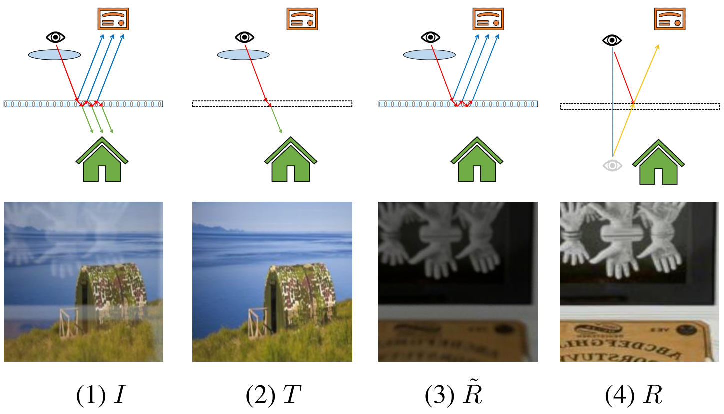
Figure 2 shows the overall pipeline of our network training algorithm using 4-image tuples as the training ground truth (GT). The algorithm contains a separation network (SP-net), which separates the input image into two layers with the help of the backtrack network (BT-net), which attempts to remove the glass/lens-effects (e.g., blurring, attenuation, and ghosting) of $\tilde{R}$ for better separation.

As shown in Figure 3, we can render 4-image tuples (I, T,$\tilde{R}$, R), and with those image tuples, we first train the BT-net, so that the $\tilde{R}$ can be backtracked into R and can be used for additional a priori information for separation. The table of Figure 3 summarizes these notations. We then train the main SP-net with rendered 4 tuples along with the pretrained BT-net.

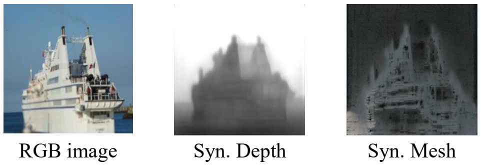
Intuitively, the algorithm makes use of additional a priori information (without glass/lens-effects) of separated $\tilde{R}$ along with widely used a posteriori information with glass/lens-effects. Specifically, those existing techniques try to calculate the error of separated reflection distorted by the glass-effect. However, the complicated glass-effects can hinder clear matching between predicted images and their GTs (e.g., feature loss), resulting in a low-quality loss generation. Interestingly, we find that the a priori information can provide additional clues for the separation problem. With the help of our BT-net, we can physically backtrack the physical process and remove the glass/lens-effects on an image.

4. Physically Faithful Dataset Generation

Compared with the classic image-space synthesized data, our physically faithful data is featured with anisotropic spatial variations that rely on physical simulation of light transportation within 3D space. In theory, the glass-effect and its physical light transmission effect are much more complex compared to the existing Gaussian blurring assumption adopted in prior techniques [7, 36, 34]. For a light path connecting a visible point $\mathbf{x}_k$ and the camera viewpoint



(1) I (2) T (3) $\tilde{R}$ (4) R



RGB image Syn. Depth Syn. Mesh

Symbol	Definition
T	Front scene image
R	Back scene image
$\tilde{R}$	Back scene image reflected by a glass
X^*	Predicted image of X
T/R	T or R

Figure 3: In this example, we set up a scene consisting of the front scene containing the house and back scene with indoor decorations. Suppose that we look at the front scene with a camera behind a glass. (1) I is the input image with reflection. (2) T is front scene transmission. (3) $\tilde{R}$ is the reflected back scene (reflection) image with lens/glass-effects, and it is computed by physically simulating the real-world attenuation and glass-effect, i.e., multiple bounces within the glass. (4) R is the back scene (reflection) image without any glass-effects.

$\mathbf{x}_0$ (Figure 3) bouncing through $k-1$ points, the contribution is computed as:

$$L(\mathbf{x}_0 \leftarrow \mathbf{x}_k) = \frac{L_e(\mathbf{x}_k, \mathbf{x}_{k-1})\hat{V}(\mathbf{x}_{k-1}, \mathbf{x}_k)}{prob(\mathbf{x}_0, \mathbf{x}_1, ...\mathbf{x}_k)} \prod_{i=1}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) f(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}) \hat{V}(\mathbf{x}_{i-1}, \mathbf{x}_i), \tag{1}$$

where $L_e(\mathbf{x}_k, \mathbf{x}_{k-1})$ is the outgoing radiance of point $\mathbf{x}_k$, $prob(\mathbf{x}_0, \mathbf{x}_1, ...\mathbf{x}_k)$ is the probability of sampling the path $\mathbf{x}_0, \mathbf{x}_1, ...\mathbf{x}_k$ from a given sampler, $\hat{V}(\mathbf{x}_{i-1}, \mathbf{x}_i)$ is the generalized visibility term between two points considering the medium attenuation factor, $G(\mathbf{x}_i, \mathbf{x}_{i+1})$ is the geometry term between two points, and $f(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$ is the bidirectional scattering function of point $\mathbf{x}_i$ from $\mathbf{x}_{i-1}$ to $\mathbf{x}_{i+1}$. Detailed explanations of these terms can be found in [5].

Simply speaking, a light path starting from a visible point is reflected/refracted multiple times by the glass and lens before contributing its brightness to the image, resulting in ghosting, blurring, defocusing, and attenuation. We call the visual effects resulting from passing through the lens or glass as lens/glass-effects. Lens-effect includes defocusing and attenuation. Glass-effect includes ghosting, blurring, and attenuation. When a path segment between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ passes through glass/lens, it will introduce glass/lens-effects. To remove those effects, we can render a scene without a glass or lens (Figure 4).

All these visual effects are spatially variant because the contribution function (Equation 1) is defined in 3D space rather 2D image space. In order to prepare such a dataset, we adopt a series of modeling and rendering techniques. Our physically-synthesized dataset not only improves the network performance but also provides a new perspective for understanding and exploring the reflection removal problem based on a physical ground.

4.1. Mesh Generation

Generating a variety of geometry meshes is the first block enabling physical simulation. Because modeling thousands of geometry scenes is economically prohibitive, we adapt the existing DIODE RGBD dataset [27]. In order to expand the diversity of the dataset, e.g., to add scenes with humans, we additionally use the labeled RGB dataset for scene recognition [38] and adopt a depth estimation technique [4] to synthesize the depth channel.

We choose 3000 image pairs (6 k in total) from the DIODE dataset, and 2000 image pairs (4 k in total) from the PLACES dataset. Specifically, we selected 34 categories of the scenes from the PLACES dataset. Because the depth estimation method predicts only normalized relative depth in a single image, we manually scaled each category of the scene with an appropriated depth range; e.g., 4 m depth on average for the bedroom scene. We mix 3000 scanned RGBD image pairs and 2000 synthesized RGBD pairs. Finally, the depth channel is fed into Blender [3] as a displacement map to export a geometry mesh from the input image. The figures in the top right corner of Figure 3 show an example.

4.2. Rendering Process

Given an RGB image and its corresponding mesh geometry, we attach the RGB channels of the image to the geometry surface to simulate the physical light transportation

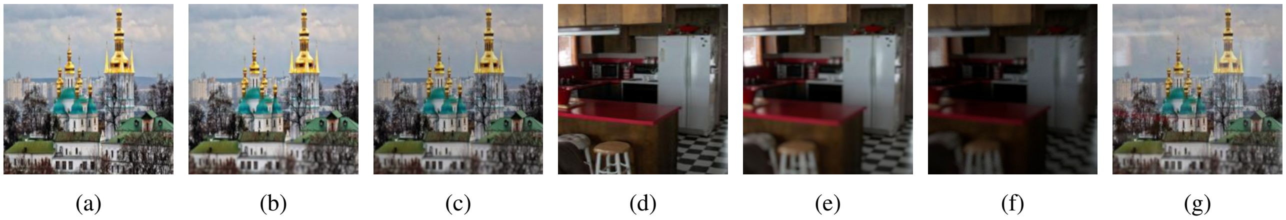


Figure 4: Images w/ and w/o lens- and glass-effects. (a) is a front scene w/o lens- and glass-effects; thus the whole image is sharp and clear. (b) is front scene w/ lens-effect, but w/o a glass-effect, where the corners are blurred since they are outside the focus range; the focus point is set to the center of the front scene and thus its effect is subtle. (c) is front scene w/ lens- and glass-effect, where the color is attenuated and image are even more blurred due to the glass. (d) is a back scene w/o lens- and glass-effects, so it is clean. (e) is back scene w/ lens-effect, but w/o glass-effect, where the whole image is blurred. (f) is a front scene w/ lens- and glass-effects, where the glass further introduces attenuation, blurring and ghosting effect. (g) is the sum of (c) and (f).

with path tracing [12]. For each scene setup, we randomly choose two images out of our image dataset, one for the front scene and the other for the back scene and render the entire scene with a glass model in the middle.

We study and decompose the physical process of light transportation and to fetch a posteriori and a priori information by rendering up to four different images for each scene. Figure 3 shows the illustrations of these four different images for a scene. These four different rendered images include:

- I: An input image containing transmission plus reflection, where both front scene and back scene are rendered with the glass-effect and lens-effect.
- T: The front scene image without any glass-effect. We simulate it with a **virtual glass**, instead of the real glass, that warps the light path as real glass, but does not cause any ghosting, blurring, and attenuation effect.
- $\tilde{R}$: The back scene image reflected by a glass with glass- and lens-effects.
- R: The back scene reflection image without any glass-effect and lens-effects. We simulate it also with the **virtual glass** to calculate the reflective direction.

Note that exact T and R are actually impossible to be captured by the real camera because taking away a real-world glass will certainly make image points shifted and thus misaligned with I anymore.

All images are rendered with a low-discrepancy sampler [12] with 256 samples per pixel, which is large enough to restrain visible noises. The glass is 10 millimeters of thickness with a common refractive index of 1.6, placed 30 centimeters in front of the camera. We use 55 millimeter thin lens model with a focus radius of 0.00893. In order to simulate the real application scenario, we set the focus distance to the center of the front scene. Overall, our synthetically generated dataset has 5000 image tuples for training and 200 image tuples for testing.

5. Proposed Network Architectures

Our model consists of two sub-networks. As illustrated in Figure 2, there is a backtrack network for the back scene reflection (BT-net) and a main separation network (SP-net). Initially, the input image I is separated into T^* and $\tilde{R}^*$(with glass-effect) using the SP-net, and then $\tilde{R}^*$ is fed into BT-net for removing the glass/lens-effect such as distortion, ghosting, attenuation, and defocusing. The output of BT-net is R^*, which is supposed to be devoid of the glass/lens-effect, and is used for providing additional error calculation for SP-net (a priori loss). Each of our network input is concatenated with extracted hypercolumn features [10] from the VGG-19 network [25] as an augmented input for better utilizing semantic information [36].

5.1. Loss function

Each sub-network has three loss terms: l_1**-loss, feature loss,** and **adversarial loss**. l_1-loss is used for penalizing pixel-wise difference in the predicted one, say, X^*, and its GT, X, via $l_1 = \|X^* - X\|$ for low-level information comparison for the results. Our feature loss and adversarial loss are based on [36]. The feature loss L_{ft} (Eq. 2) is used for considering semantic information, based on the activation difference of a pre-trained VGG-19 network Φ, which is a trained with the ImageNet dataset [22]. For obtaining realistic images, the adversarial loss is adopted too, as the many other recent studies [36, 34, 15, 39]. A conditional GAN [11] is utilized for this. For explanation, suppose that one of our sub-network's generator is f, its input is X, and its GT is Y. The feature loss L_{ft} is calculated as follows:

$$L_{ft}(f(X), Y) = \sum_l \gamma \|\Phi_l(Y) - \Phi_l(f(X))\|, \quad (2)$$

where Φ_l indicates the l-th layer of the VGG-19 network with the same layer selection of [36], which is ‘conv1_2’, ‘conv2_2’, ‘conv3_2’, ‘conv4_2’, and ‘conv5_2’. γ is the weighting parameter, which is empirically set to 0.2.

For the adversarial loss, the discriminator D of one sub-network is trained by:

$$\sum_{X,Y \in \mathcal{D}} \log D(X, f(X)) - \log D(X, Y), \tag{3}$$

where the discriminator tries to differentiate between the GT patches of Y and patches given by $f(X)$ conditioned on the input X. Adversarial loss is then defined as follows:

$$L_{adv}(X, f(X)) = \sum_{X \in \mathcal{D}} - \log D(X, f(X)). \tag{4}$$

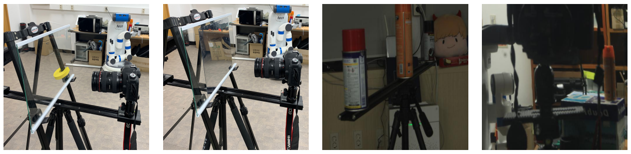
Loss for SP-net. The purpose of the SP-net is separating T^* and $\tilde{R}^*$ from the input I. The first loss we calculate for training SP-net on its output $(T^*, \tilde{R}^*)$ is *a posteriori loss* (L_{pst}) with the lens/glass-effect. It is the combination of l_1-loss, feature loss between predicted value and ground-truth, and adversarial loss for T^*. After using BT-nets removing glass/lens-effect of $\tilde{R}^*$ (so that it becomes R^*), we also calculate the second loss term called *a priori loss* (L_{pr}) without the glass/lens-effect between predicted R^* and ground-truth R.

$$\begin{aligned} L_{pst} &= L_{l_1}(T^*, T) + L_{ft}(T^*, T) + L_{adv}(I, T^*) \\ &\quad + L_{l_1}(\tilde{R}^*, \tilde{R}) + L_{ft}(\tilde{R}^*, \tilde{R}), \qquad (5a) \\ L_{pr} &= L_{l_1}(R^*, R) + L_{ft}(R^*, R). \qquad (5b) \end{aligned}$$

Combining the above loss terms, our complete loss for SP-net is $L_{SP} = L_{pst} + L_{pr}$.

Loss for BT-net. The goal of BT-net is removing the glass/lens effects from $\tilde{R}$, so that it can be recovered from darkening and blurring. To train the network, we formulate a combined loss function of l_1-loss, feature loss, and adversarial loss as follows:

$$L_{BT} = L_{l_1}(R^*, R) + L_{ft}(R^*, R) + L_{adv}(\tilde{R}, R^*). \tag{6}$$



With glass   With mirror   Front scene   Back scene

Figure 5: Experiment setup and taken images.

Implementation. Each of our two sub-nets shares the same structure based on the one proposed in [36], and they are fully convolutional networks for considering global information. For the training, we first train BT-net with rendered image pairs independently, and then pre-trained BT-net is connected to SP-net for SP-net training (BT-net is fine-tuned in this stage). SP-net is trained by minimizing the aforementioned loss terms between GTs and their predictions with a learning rate of 10^{-4}. The rendered training images have the 256 × 256 resolution.

6. Experiments with Real and Synthetic Data

We compare our approach with the state-of-the-art deep learning-based reflection removal methods, CEILNet [7], Zhang et al. [36], BDN [34], Wen et al. [32] across different test sets that work for a given single image.

For quantitative evaluation with real-world images, we utilize the well-known reflection removal benchmark, the SIR Wild dataset [28]. It consists of three images $(I, T, \tilde{R})$ under various capturing settings from controlled indoor scenes to wild scenes. Since the indoor dataset is designed for exploring the impact of various parameters, we test our results on their wild scenes. Also, we additionally capture 100 real reflection pair images for testing (denoted as real100). Also, we generate 200 rendered images for testing.

6.1. Dataset Evaluation

In order to validate our rendered dataset and its similarity to real-world reflection captured by a camera, we capture real image pairs with devices of Figure 5. We first capture the GT I with a glass (so that it contains reflection), then use a mirror to capture GT R and remove the glass to capture GT T as inputs of data synthesis. In order to match common RGB and RGBD datasets, the GT T and R are captured with F22 to minimize the defocusing effect. In addition, we capture and calibrate the depth map using a Kinect on each side of the slider across the glass. With the captured GT T and R, we generate reflection images with three different methods [7, 36, 32], and compare them with our rendered images. Figure 1 shows an example of the generated reflections. As shown, with depth information and a physically based rendering model, ours can generate lens- and glass-effects much similar to the real images.

Table 1 shows the numerical comparison of generated reflection images, and we use average PSNR and SSIM for

Method	PSNR	SSIM
CEIL [7]	14.466	0.737
Zhang [36]	20.379	0.842
Wen [32]	20.266	0.856
Ours	29.307	0.943

Table 1: The average similarity of synthesized reflections with 10 real camera-captured reflection images

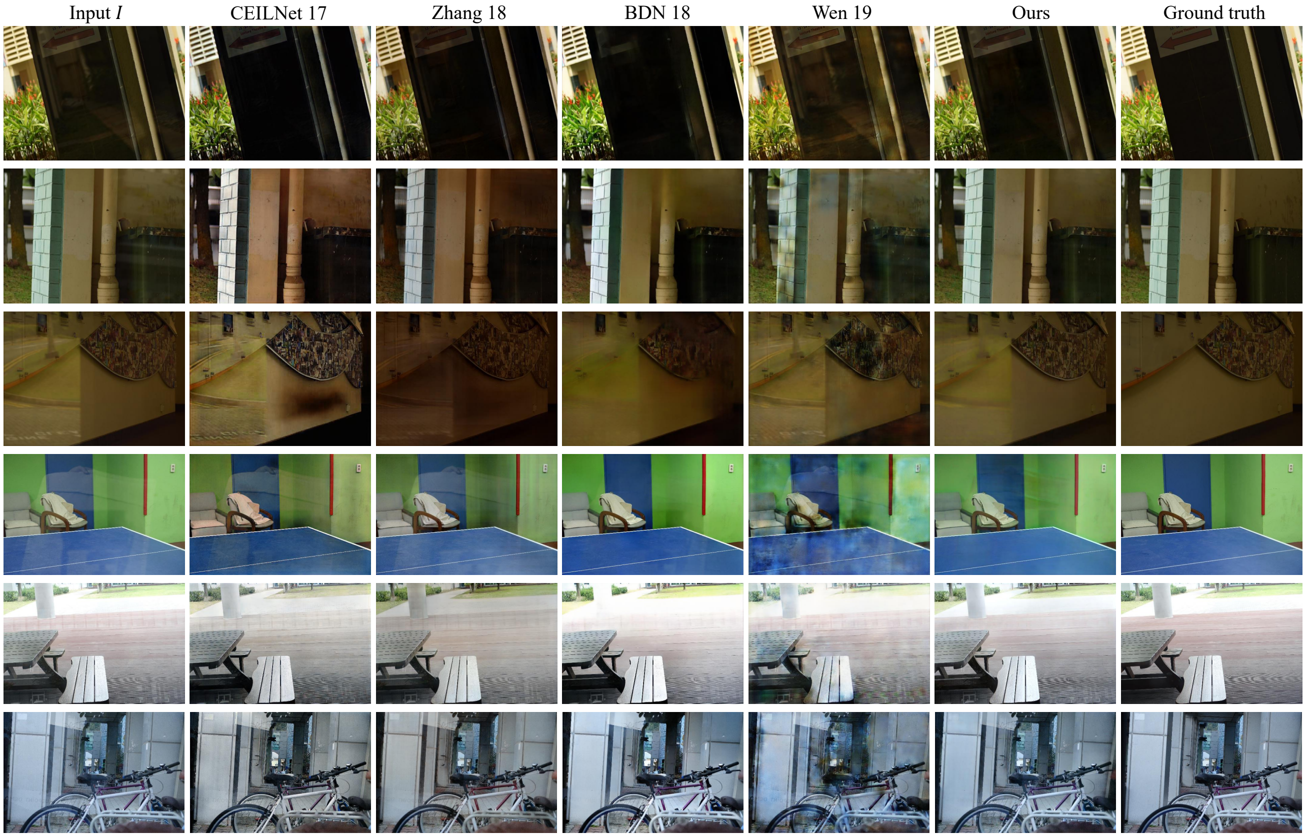


Figure 6: Examples of reflection removal results on the wild dataset (Rows 1-3) and our real 100 testset (Rows 4-6) visually.

Dataset	Index	Methods									
		Input	CEILNet [7]	CEILNet FW	CEILNet FR	Zhang [36]	Zhang FW	Zhang FR	BDN [34]	Wen [32]	Ours
SIR Wild [28]	PSNR	25.89	20.89	19.23	22.51	21.15	21.34	23.18	22.02	21.26	25.55
	SSIM	0.903	0.826	0.819	0.880	0.851	0.865	0.890	0.835	0.835	0.905
Real 100	PSNR	21.53	19.24	17.82	20.35	18.66	18.88	20.44	19.46	19.07	21.59
	SSIM	0.797	0.733	0.706	0.764	0.750	0.753	0.773	0.753	0.728	0.789
Rendered Testset	PSNR	23.27	19.31	20.23	23.46	22.21	21.83	24.43	21.66	21.79	27.90
	SSIM	0.846	0.745	0.777	0.829	0.829	0.828	0.854	0.819	0.804	0.894

Table 2: Quantitative results of different methods on SIR wild, our real 100, and rendered test set. Some result images of the SIR dataset can be found in Figure 6. CEILNet, Zhang, and BDN are the pre-trained networks. CEILNet-FR and Zhang-FR are fine-tuned with our rendered training images, and CEILNet-FW and Zhang-FW are fine-tuned with Wen's data generation method with the same source images with ours. Red numbers are the best, and blue numbers are the second best results.

measuring the similarity. We take two different scenes with 5 focus points, in total 10 real reflection images for comparison. Note that 10 real reflection images are different from the real 100 test set we captured because the real 100 test set does not have depth. For a fair comparison, we randomly synthesize 100 images using both CEIL [7] method and Zhang et al. [36] method for every 10 scenes and pick the best PSNR and SSIM synthesized image for each scenes. For the Wen [32] method, since their method utilizes pre-trained reflection synthesis network to produce 3 types of reflection, we generate 3 different images for each scene. Among them, we pick the best PSNR and SSIM synthesized result for each scenes. The report of their average values is listed in Table 1.

6.2. Ablation Study

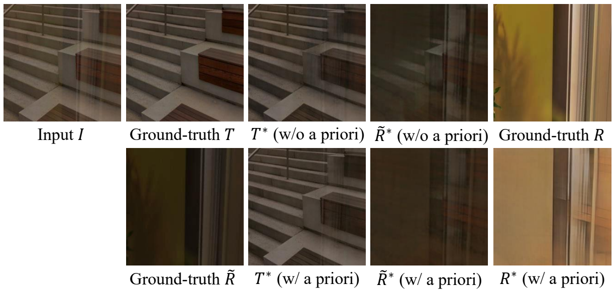
In order to validate the effectiveness of a priori loss from the BT-net, we evaluate each model (w/ and w/o a priori

Dataset	Index	Model	
		w/o a priori loss	w/ a priori loss (Ours)
SIR wild [28]	PSNR	24.31	**25.54**
	SSIM	0.874	**0.905**
Real 100	PSNR	20.86	**21.58**
	SSIM	0.772	**0.789**
Rendered Testset	PSNR	27.34	**27.90**
	SSIM	0.889	**0.894**

Table 3: Quantitative comparison of our ablated models

loss) on SIR wild, real 100 images, and 200 rendered images. Each model is trained from scratch with a denoted loss combination. Since we followed the other loss terms of Zhang et al. [36], we conduct an ablation study on the new a priori loss only.

The numerical results show that using the additional a priori loss can improve the separation quality both in the real and rendered test sets. Since BT-net backtracks the darken and distorted predicted $\tilde{R}^*$ into R^* to calculate a priori loss, this loss can provide a more robust signal of separation quality. Moreover, Figure 7 shows some visual results of our complete model and ablated model on the rendered test set. Since BT-net can backtrack the predicted $\tilde{R}^*$ into R^*, our complete model could figure out the reflection and transmission area better when separating.


Input I Ground-truth T T^* (w/o a priori) $\tilde{R}^*$ (w/o a priori) Ground-truth R

Ground-truth $\tilde{R}$ T^* (w/ a priori) $\tilde{R}^*$ (w/ a priori) R^* (w/ a priori)

Figure 7: Comparison between the rendered results of our complete model and an ablated model.

6.3. Comparision on Benchmarks

For comparison, we utilize pre-trained network weights provided by authors. Additionally, we also fine-tune the author's pre-trained network with our rendered dataset and a dataset generated by Wen's reflection synthesis network. Two generated datasets share the same source image pairs, and we use Wen's pre-trained weight and default setting for generating reflection training images. Both fine-tuned networks are tuned with the same epoch and learning rate. We name the models that are fine-tuned with our rendered data with a suffix '$-FR$', and the models finetuned with Wen's reflection synthesized images with a suffix '$-FW$'. Since BDN does not provide training code, and Wen's network needs additional alpha blending mask for training their separation network, we cannot fine-tune them.

Figure 6 shows some visual examples of reflection removal results on the SIR wild test set and our real 100 test set. All the compared methods do not work well in terms of removing strong regional reflection (row 3), but still, our method removes some of the reflection without significantly damaging the transmission area. In the last row, ours and BDN [34] could remove the reflection of banner in the below, while other methods do not remove, but darken the overall intensity.

Table 2 shows quantitative results on the real-world test sets (SIR wild and real 100) and our rendered test set. We utilize SSIM and PSNR as error metrics, which are widely used in prior reflection removal methods. Our method achieves the best or second-best numerical results in all the datasets. We also validate that our dataset can improve previous methods (pre-trained networks) by supplying more physically-based reflection training images (Table 2). However, for both real reflection testset, none of the existing methods, no matter how they are trained by classic synthesized dataset or our rendered dataset, outperforms the unseparated input in both error metrics. This suggests there is still room for further improvement.

7. Conclusion

We have proposed a novel learning-based single image reflection removal method, which utilizes reflection training images generated by physically-based rendering. The training images consist of different types, including transmission and reflection w/ and w/o the glass/lens-effects, and provide both classical a posteriori and novel a priori information. With the new dataset, we proposed SP-net to separate the input into two layers with the help of BT-net to remove the glass/lens-effect in the separated layers for error calculation (a priori loss). With a priori loss, the separation loss calculation is improved. Also, we validated that our physically-based training data can improve existing learning-based reflection removal methods as well with various real reflection test images.

Limitation. In this paper, we did not consider viewpoints that are not perpendicular to the glass. That is one possible extension for future research. Also, we did not consider the curved glass or glass with a special shape, while our rendering approach can accommodate these cases by replacing the plane glass model with a curved one in the future.

Acknowledgments

We would like to thank anonymous reviewers for constructive comments. Sung-Eui Yoon and Yuchi Huo are co-corresponding authors of the paper. This work was supported by MSIT/NRF (No. 2019R1A2C3002833) and SW Starlab program (IITP-2015-0-00199)

References

[1] Amit Agrawal, Ramesh Raskar, Shree K Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 828–835. ACM, 2005.

[2] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4506, 2017.

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2019.

[4] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016.

[5] Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. Scalable realistic rendering with many-light methods. In *Computer Graphics Forum*, volume 33, pages 88–104. Wiley Online Library, 2014.

[6] William Donnelly. Per-pixel displacement mapping with distance functions. *GPU gems*, 2(22):3, 2005.

[7] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3238–3247, 2017.

[8] Kun Gai, Zhenwei Shi, and Changshui Zhang. Blind separation of superimposed moving images using image statistics. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):19–32, 2012.

[9] Byeong-Ju Han and Jae-Young Sim. Reflection removal using low-rank matrix completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5438–5446, 2017.

[10] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[12] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[13] James T Kajiya. The rendering equation. In *ACM SIGGRAPH computer graphics*, volume 20, pages 143–150. ACM, 1986.

[14] Naejin Kong, Yu-Wing Tai, and Joseph S Shin. A physically-based approach to reflection separation: from physical modeling to constrained optimization. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):209–221, 2014.

[15] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[16] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647–1654, 2007.

[17] Anat Levin, Assaf Zomet, and Yair Weiss. Learning to perceive transparency from the statistics of natural scenes. In *Advances in Neural Information Processing Systems*, pages 1271–1278, 2003.

[18] Anat Levin, Assaf Zomet, and Yair Weiss. Separating reflections from a single image using local features. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[19] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2759, 2014.

[20] Abhijith Punnappurath and Michael S Brown. Reflection removal using a dual-pixel sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1556–1565, 2019.

[21] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24–37, 2015.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[23] Yoav Y Schechner, Nahum Kiryati, and Ronen Basri. Separation of transparent layers using focus. *International Journal of Computer Vision*, 39(1):25–39, 2000.

[24] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3193–3201, 2015.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[26] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[27] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.

[28] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal

algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3922–3930, 2017.

[29] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Crrn: multi-scale guided concurrent reflection removal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4777–4785, 2018.

[30] Renjie Wan, Boxin Shi, Tan Ah Hwee, and Alex C Kot. Depth of field guided reflection removal. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 21–25. IEEE, 2016.

[31] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019.

[32] Qiang Wen, Yinjie Tan, Jing Qin, Wenxi Liu, Guoqiang Han, and Shengfeng He. Single image reflection removal beyond linearity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3771–3779, 2019.

[33] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):79, 2015.

[34] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 654–669, 2018.

[35] Sung-eui Yoon. *Rendering*. 2018. First edition.

[36] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4786–4794, 2018.

[37] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5295, 2017.

[38] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[39] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.