# Single Image Reflection Removal Beyond Linearity

Qiang Wen[1], Yinjie Tan[1], Jing Qin[2], Wenxi Liu[3], Guoqiang Han[1], Shengfeng He[1*]

[1] School of Computer Science and Engineering, South China University of Technology
[2] Department of Nursing, Hong Kong Polytechnic University
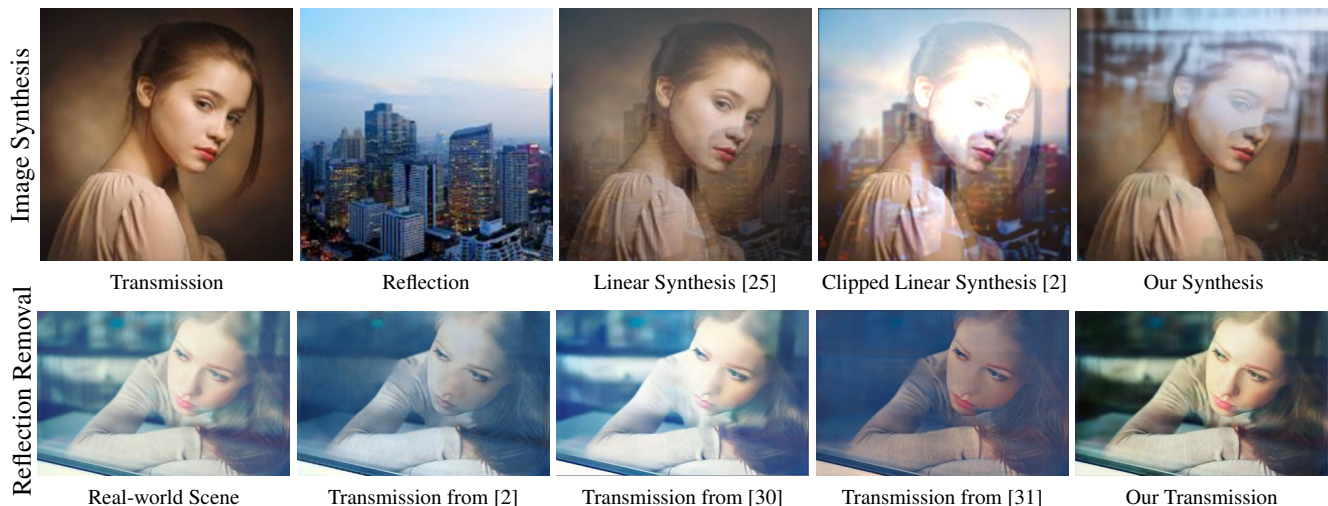[3] College of Mathematics and Computer Science, Fuzhou University

Figure 1: Existing reflection removal methods rely heavily on the linearly synthesized data, which, however, cannot simulate the real-world reflections. We propose to synthesize and remove reflection beyond linearity, leading to the controllable synthesis and clean reflection removal.

## Abstract

*Due to the lack of paired data, the training of image reflection removal relies heavily on synthesizing reflection images. However, existing methods model reflection as a linear combination model, which cannot fully simulate the real-world scenarios. In this paper, we inject non-linearity into reflection removal from two aspects. First, instead of synthesizing reflection with a fixed combination factor or kernel, we propose to synthesize reflection images by predicting a non-linear alpha blending mask. This enables a free combination of different blurry kernels, leading to a controllable and diverse reflection synthesis. Second, we design a cascaded network for reflection removal with three tasks: predicting the transmission layer, reflection layer, and the non-linear alpha blending mask. The former two tasks are the fundamental outputs, while the latter one being the side output of the network. This side output, on the other hand, making the training a closed loop, so that the separated transmission and reflection layers can be recombined together for training with a reconstruction loss. Extensive quantitative and qualitative experiments demonstrate the proposed synthesis and removal approaches out-*

*performs state-of-the-art methods on two standard benchmarks, as well as in real-world scenarios.*

## 1. Introduction

Undesired reflection from glasses not only damages the image quality, but also influences the performance of computer vision tasks like image classification. To remove reflection, early researches design hand-crafted priors [11, 13, 18, 26], while recent works [2, 31, 25, 29] train deep models to remove reflection patterns.

Notwithstanding the demonstrated success of deep reflection removal models, the arguably most critical challenge is to obtain sufficient paired training data, which includes the reflection images and their corresponding clean transmission images. To synthesize reflection data, existing methods [2, 31, 25, 29] simply blend two images with a linear model. Particularly, they express the reflection image $S \in [0,1]^{m \times n \times 3}$ as the linear combination of the transmission layer $T \in [0,1]^{m \times n \times 3}$ and the reflection layer $R \in [0,1]^{m \times n \times 3}$:

$$S = \alpha T + (1 - \alpha)(K \otimes R),  \qquad (1)$$

where $\alpha \in (0.5, 1)$ is the combination factor, $\otimes$ denotes a

*Corresponding author (hesfe@scut.edu.cn).

convolution operator, and $K$ represents a Gaussian blurring kernel. However, blending two images with a constant $\alpha$ does not simulate the complex real-world reflection. The formation of reflection image depends on the relative position of the camera to the image plane and on the lighting conditions [9].

In this paper, we revisit this challenging synthesis problem. We observe that the synthesis of reflection images should be non-linearly combined with an alpha blending mask. To this end, we propose a deep synthesis network *SynNet* to predict the alpha blending mask of two input images. This mask can be freely combined with a user defined kernel to simulate different types of reflections. This is indispensable for generating the controllable and diverse reflection data. To properly train the network, we selectively collect a large number of real data with different types of reflections.

On the other end, we involve the non-linear alpha blending mask into our reflection removal process. We design a cascaded reflection removal network *RmNet* which has three tasks: predicting the reflection layer, transmission layer, and the alpha blending mask. The first two are the essential outputs of the reflection separation, while the last one is treated as the side output that aids the network training. We use the predicted mask to re-combine the separated transmission and reflection layers. In this way, the re-combined reflection should be consistent with the original input, and the entire network is a closed loop thus can be guided with a reconstruction loss.

In summary, our contributions are:

- We revisit the single image reflection removal problem, and synthesize reflection data beyond linearity. In particular, we propose to predict a non-linear alpha blending mask, which enables a controllable and diverse reflection data synthesis.

- We present a reflection removal network with the aid of the predicted alpha blending mask. This mask serves as the side output of our network, so that the predictions in the first stage can be re-combined together, and the network can be supervised with a reconstruction loss.

- The proposed networks outperform existing reflection synthesis and removal methods, on two standard benchmarks and in real-world scenarios.

## 2. Related Work

**Single-image reflection removal.** Reflection removal/separation has been a long-standing problem. Early researches many address this problem by proposing different image priors [11, 10, 19, 1, 13, 26, 12, 22]. For example, sparse gradient priors [11] are used to distinguish the reflection and transmission layers. Li and Brown [13] extracts the reflection and transmission layers using a smooth gradient prior by assuming that reflections are often less in focus. These priors may work well on the specific cases, but cannot generalize to different types of reflections.

As a consequence, deep models are adopted for removing reflection. Fan *et al.*[2] propose the first attempt to solve this ill-posed problem with a deep network. They use an edge map as the additional cue to guide the layer separation. Wan *et al.*[25] develop a two-stage network, while the first one inferences the gradient of the transmission layer, and the second one associates with the gradient output to predict the final transmission layer. Without augmented cues, Yang *et al.*[29] propose a bidirectional network that separately predicts reflection and transmission layers, and then uses the predicted reflection to estimate the transmission layer in the second step. To exploit low-level and high-level image information, Zhang *et al.*[31] introduce two perceptual losses and an exclusion loss into a fully convolutional network. However, all the above methods suffer from the linear image synthesis model, preventing these methods from generalizing to different real-world scenarios. Although two reflection datasets [24, 31] have been proposed, they are too small and captured with a conditioned environment, *i.e.*, only one type of reflection. These problems motivate us to synthesize realistic reflections beyond linearity.

**Physically-based reflection models.** Some previous works [9, 17, 27, 21] explore the physical model of reflection. They regard the reflection model as a non-linear combination of the light reflected off the glass surface and the light transmitted through the surface. The imaging process is also determined by the angle between the incoming light and the surface. In the same time, according to the Malus's law [6], when an reflection image is taken by the polarizer, the amount of light is also changed by the angle between the polarization direction of the incoming light and the transmission axis of the polarizer. All these factors result in different type of reflections.

Following the above rules, some physically-based reflection removal methods use simplified models for specific scenarios. Kong *et al.*[9] require a series of three polarized images in the same scene, each captured with a different polarizer angle. However, they assume that thickness of the medium is thin enough, thus ghosting effect is not considered in this model. In contrast, Shih *et al.*[18] focus on the ghosting reflection. They regard the reflection as double layers. One layer is the primal reflection layer, and the other one is a spatially shifted and attenuated image of the former. Due to this characteristic, it models the ghosting effect using a double-impulse convolution kernel and removes the reflection with a Gaussian Mixture Model. Although the physically-based methods model reflections accurately, they are limited to specific cases. On the con-

| (a) Real Focused | (b) Real Defocused | (c) Real Ghosting |
| (d) Our Focused | (e) Our Defocused | (f) Our Ghosting |

Figure 2: Three types of reflection examples. The first row shows the real examples and the second row shows our synthesized images. The proposed synthesis method is able to simulate all the three types of reflections.

trary, we leverage the physical reflection model from a data synthesis aspect, creating realistic and diverse training data.

## 3. SynNet: the Synthesis Network

To inject non-linearity into the synthesis process, we rewrite Eq. (1) as follows:

$$S = W \circ T + (1 - W) \circ (K \otimes R), \qquad (2)$$

where $\circ$ is a element-wise multiply operator. $W \in [0, 1]^{m \times n \times 3}$ denotes the alpha blending mask, which is a non-constant matrix that weighs the contribution of the transmission layer at each pixel. Each layer combined with the alpha blending mask is to simulate the intensity of the physical light from the corresponding objects. In the real world, the kernel $K$ have different forms according to the thickness of the glass or the angle between the incoming light and the surface [24]. According to different scenarios, reflection can be roughly categorized into three types, focused reflection, defocused reflection, and ghosting reflection. These criteria are the principles to our data collection and synthesis processes.

### 3.1. Reflection Types and Data Collection

**Focused Reflection.** When the object behind the glass and the reflected object are in the same focal plane, the reflection layer will be as sharp as the transmission layer in the reflection image. In this case, the kernel $K$ is considered as a one-pulse kernel. To prevent intensity overflow, some works [25, 28] scale down the light of the two layers linearly by a constant. Both two layers of this type of reflection look sharp, and thus they are difficult to separate

by human eyes. Fig. 2 (a) shows an example of focused reflection.

**Defocused Reflection.** In most reflection images, they are captured from a certain distance to the camera, and therefore the reflected objects are usually out of focus when the object behind the glass is in the focal plane. In this case, the reflection layer is blurry and smoother than the transmission layer. Most linear blending methods [29, 31, 2] model this type of reflection images by setting the kernel $K$ as a Gaussian kernel to simulate the blurry reflection. A real defocused reflection image is shown in Fig. 2 (b).

**Ghosting Reflection.** For the two above types, we assume the thickness of the medium, such as a glass, is thin enough to regard it as single-surface. However, when the thickness is non-negligible, we should take the refraction into consideration, as it will cause the quadric reflection with shifting. To model this ghosting reflection, Shih et al. [18] set the kernel $K$ as a two-pulse kernel which is called the ghosting kernel. Fig. 2 (c) shows an real ghosting reflection example.

To simulate the above typical types of reflections, we collect 1109 real-world reflection images (each type has 306, 672, 131, respectively) for training the synthesis network *SynNet*. We denote this real reflection dataset as $\Re$.

### 3.2. Network Structure

The network structure of the proposed SynNet is shown in Fig. 3. *SynNet* has an encoder-decoder structure [14, 15]. It takes a six-channel image as input, by concatenating two real-world clean images. The former three channels are treated as the transmission layer, and the latter are the reflection layer that preprocessed by the kernel $K$. Both encoder and decoder contain three convolutional layers. In the middle of them, we add nine residual blocks [5] to enrich the reflection features representations. All convolution layers are followed by an InstanceNorm layer [23] and ReLU activation, except the last layer followed by the Sigmoid activation function to scale the output into [0, 1]. The network outputs a three-channel alpha blending mask.

### 3.3. Objective Function

The objective function of *SynNet* contains two terms: an adversarial loss and a smoothness loss.

**Adversarial Loss.** As there is no paired data for the synthesis training process, involving an adversarial loss is arguably the best solution. We use the collected real data of three reflection types as the real samples for training the discriminator. Note that we do not directly synthesize the reflection data, as it is impossible to control the output reflection type. Instead, we synthesize the alpha blending mask, so that all the three types can be generated and they can be used to train the network properly. The loss for the discrim-
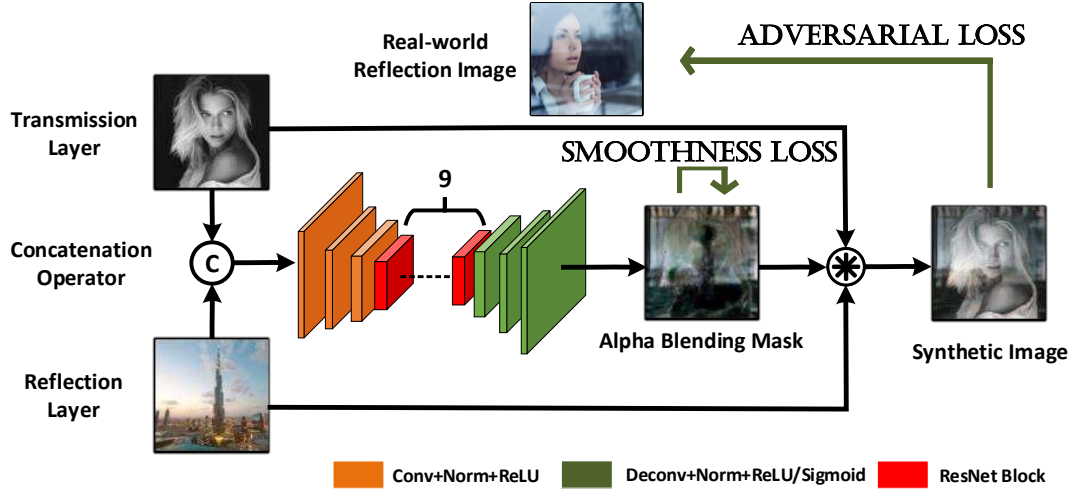
Figure 3: The proposed synthesis network *SynNet*. The symbol ⊛ denotes the blending operator of Eq. (2). Instead of directly synthesizing the reflection image, we synthesize the alpha blending mask. With a different choice of the blurring kernel, the proposed network can be controlled to generate different types of reflections.

inator $D$ is defined as:

$$L_D = \sum_{I,S \in \Re} log D(I) + log(1 - D(S)), \qquad (3)$$

where $D(x)$ is the probability that $x$ is a real reflection image, $I$ denotes the real-world image and $S$ is our synthesized images. According to [3], we optimize the network with only the first term in Eq. (3). The adversarial loss is then defined as:

$$L_{adv} = \sum_{S \in \Re} -log(D(S)). \qquad (4)$$

**Smoothness Loss.** We add a smoothness loss $L_{smooth}$ as an augmented loss to avoid the value mutation in the alpha blending mask, which will cause the unexpected color change in the synthetic image. This loss is to encourage the spatial smoothness, which is also used some other image processing applications like super-resolution [7]. The smoothness loss is defined as:

$$L_{smooth} = \sum_{S \in \Re} \sum_{i,j} \|W_{i+1,j} - W_{i,j}\|_1 + \|W_{i,j+1} - W_{i,j}\|_1,$$
$$(5)$$

where $W_{i,j}$ denotes the pixel value of the alpha blending mask.

Overall, our objective function of *SynNet* is:

$$L_{syn} = w_1 L_{adv} + w_2 L_{smooth}. \qquad (6)$$

We heuristically set $w_1 = 1$ and $w_2 = 10$ to balance the contribution of each term.

During training, the kernel $K$ is selected among three reflection types according to the statistic (2.34 : 5.13 : 1)

of the collected data. In this way, the proposed network generates a $W$, which can be combined with a kernel $K$ to simulate Eq. (2) with different types of reflections. Fig. 2 shows a comparison for the synthesized reflections and the real ones. We can see that the generated images show similar reflection features to the real reflection images.

## 4. RmNet: the Removal Network

Given a large amount of synthetic data $\mathfrak{S}$, we propose a cascaded network for reflection removal.

### 4.1. Network Structure

The architecture of the proposed network is shown in Fig. 4. It is a three-stream structure with one encoder and three decoders, and each layer of the encoder have skip-connections to the corresponding layers of all the three decoders. For the decoder, there are six convolutional layers with the kernel size of $4 \times 4$ and stride-2. Each layer is followed by the InstanceNorm layer and a Leaky ReLU activation function (slope is 0.2). For the decoders, there are six (de)convolutional layers with the kernel size of $4 \times 4$ and stride-$\frac{1}{2}$. Similarly, they are followed by the InstanceNorm layer and ReLU activation function.

Each decoder corresponds to predict a different output image. Two of them estimate the transmission and reflection images. These are the basic elements of the reflection removal task. On the other hand, our synthesized images are constructed by the alpha blending masks, and therefore they can be treated as the ground truth for supervising the *RmNet* to produce the alpha blending mask as an additional output. In this way, all the three outputs can be united to reconstruct the input reflection image. This makes the network an closed loop, and allowing a new reconstruction loss
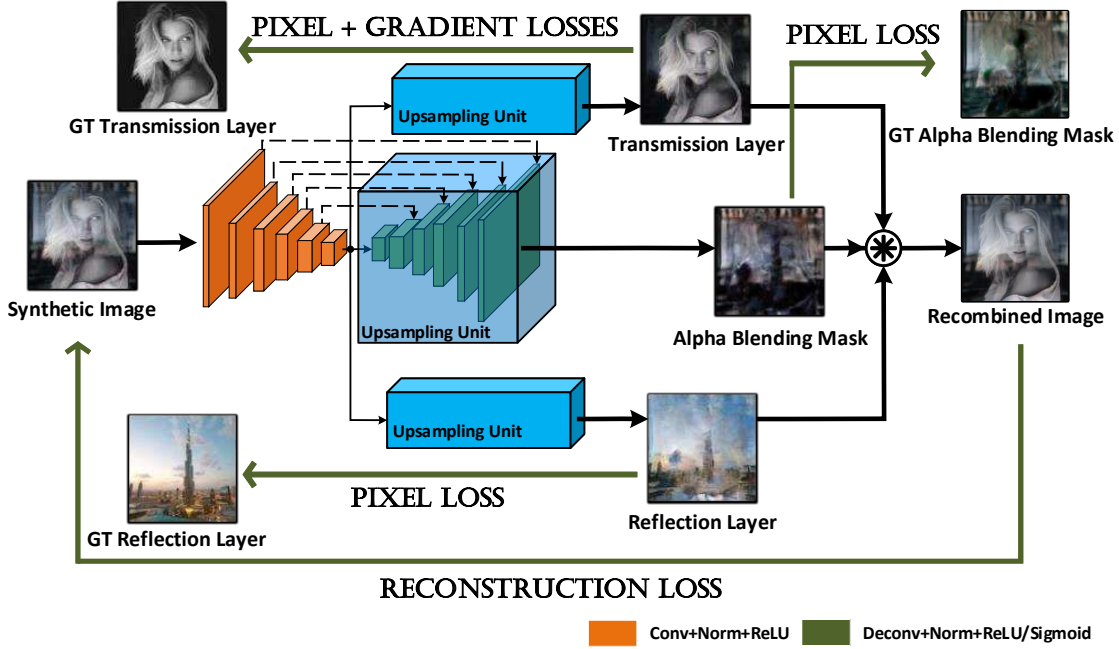
Figure 4: The proposed reflection removal network *RmNet*. The symbol ⊛ denotes the blending operator of Eq. (2). We involve the alpha blending mask as the side output, making the training a closed loop. Therefore, our network can be trained with a reconstruction loss.

for training.

## 4.2. Objective Function

The objective function of *RmNet* contains three terms: a pixel loss, a gradient loss, and a reconstruction loss.

**Pixel Loss.** To ensure the outputs as similar to the ground truth as possible, we utilize $L1$ loss to measure the pixel-wise distance between them. Our pixel loss is defined as:

$$L_{pixel} = \sum_{T,R,W \in \mathfrak{S}} \|T - T^*\|_1 + \|R - R^*\|_1 + \|W - W^*\|_1 ,$$

(7)

where the $T^*, R^*, W^*$ are the predicted transmission, reflection layer, and the alpha blending mask, respectively.

**Gradient Loss.** For an reflection image, the gradients are consistent in the transmission images, while the gradients vary in the reflection images [13, 4]. We use the *Sobel* operator to extract the gradient images for the transmission layer. Then we compare the predicted transmission with its ground truth in gradient domain to keep the same gradient distribution. We obtain both the horizontal and vertical gradients, and our gradient loss is defined as:

$$L_{grad} = \sum_{T \in \mathfrak{S}} \|G_x(T) - G_x(T^*)\|_1 + \|G_y(T) - G_y(T^*)\|_1 ,$$

(8)

where $G_x()$ and $G_y()$ are vertical and horizontal *Sobel* operators respectively.

**Reconstruction Loss.** Different from existing reflection removal networks, we introduce a new loss to the proposed network, named reconstruction loss. Due to the additional predicted alpha blending mask, we can re-compose the three outputs of *RmNet* according to Eq. (2). It is intuitive that the re-composed reflection image should be similar to the original input, if the network is trained properly. For the reconstruction loss, we measure the perceptual distance between the recombined image and the input image. Both the images are fed to a *VGG19* network $F(\cdot)$, and the reconstruction loss is defined as:

$$L_{reconstr} = \sum_{S \in \mathfrak{S}} \|F(S) - F(S^*)\|_1 ,$$

(9)

where $S^*$ is the recombined image.

Overall, our object function of *RmNet* is:

$$L_{rm} = w_1 L_{basic} + w_2 L_{grad} + w_3 L_{reconstr}.$$

(10)

We heuristically set $w_1 = 100$, $w_2 = 50$ and $w_3 = 100$.

## 5. Experiments

We implement the proposed two networks in Pytorch on PC with a Nvidia Geforce GTX 1080 Ti GPU. Every networks are trained for 130 epoches with a batch size of 10, using the Adam optimizer [8] with a learning rate of 0.0002. To generate our synthetic training data, we collect 4000 images from Flickr randomly to form the transmission and reflection layers, and they are randomly blended to generate

| | Focused | Defocused | Ghosting |
| Input | GT Transmission | CEILNet [2] | Zhang *et al.*[31] | BDN [29] | Ours |

Figure 5: Qualitative comparisons on our synthetic testing set. We show three types of reflection images generated by our SynNet. State-of-the-art models cannot address all the scenarios.

Table 1: Comparison of the generated reflection images with respect to the inception score.

| | [31] | [2] | [29] | Real | **Ours** |
|---|---|---|---|---|---|
| Inception | 1.138 | 1.194 | 1.134 | - | **1.272** |
| Score | ± 0.034 | ± 0.042 | ± 0.044 | - | **± 0.037** |
| Accuracy | 78.00% | 86.67% | 74.00% | 97.33% | **93.33%** |

the reflection images. We also involve the real-world paired training data from Zhang *et al.*[31] and $SIR^2$ datasets [24]. Note that when we train *RmNet* on these two real-world datasets, we discard the losses of the reflection layer and alpha blending mask. This is because they cannot capture the ground truth reflection layers.

We evaluate the proposed network on four testing sets. Two real-world testing sets from Zhang *et al.*[31] and $SIR^2$. The former one contains 20 testing reflection images, and the latter one we select 55 testing reflection images from the wild scene subset (this is the only subset without overlapped images). Furthermore, we construct another synthetic testing set, which includes 300 reflection images in three different types (1:1:1). The above three testing sets contain ground truth transmission layers. We also construct

a real-world reflection dataset with 25 images from the Internet, without any transmission ground truth, for qualitative evaluations and user study.

## 5.1. Comparison to State-of-the-arts

We compare the proposed *RmNet* to three state-of-the-art deep models: CEILNet [2], Zhang *et al.*[31], and BDN [29].

### 5.1.1 Quantitative Evaluations

**Reflection generation.** Evaluating images generated by a GAN is challenging. Here we use the inception score [16] to assess the quality of our synthesis method and existing linear combination methods. In particular, we re-train the inception_v3 network [20] on the real-world reflection images and non-reflection images. Then we randomly select 300 images from Flickr, and these images are served as the foreground and background image pairs for data synthesis using different methods (*i.e.*CEILNet [2], Zhang *et al.*[31], BDN [29], and ours). All the methods use the same 150 pairs of images for synthesis, and they are evaluated by the re-trained inception_v3 network. Table 1 shows the inception score and the accuracy of each synthetic set. Note that

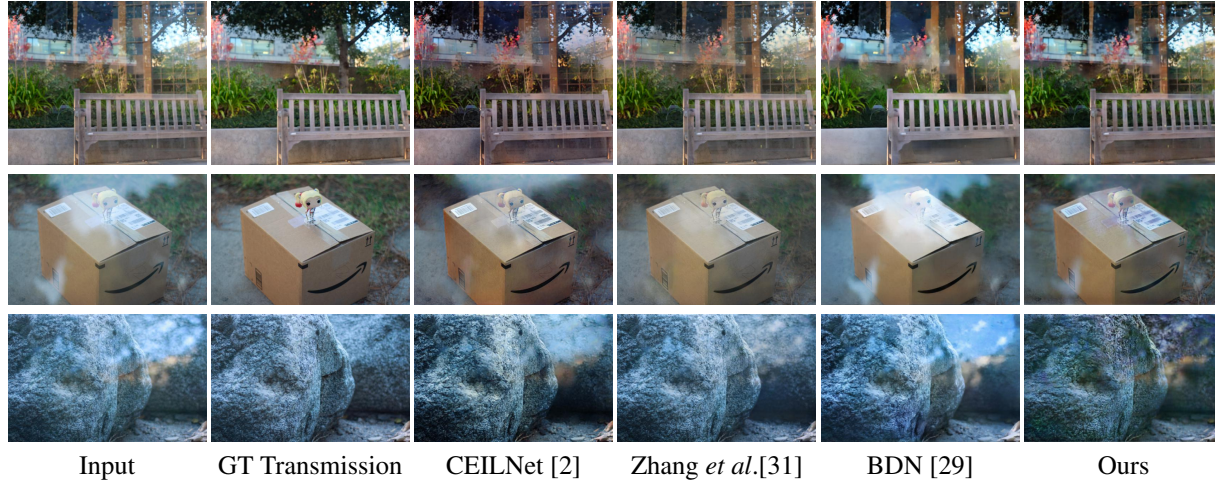| Input | GT Transmission | CEILNet [2] | Zhang *et al.*[31] | BDN [29] | Ours |

Figure 6: Qualitative comparisons on the dataset collected by Zhang *et al.*[31].

Table 2: Quantitative evaluations on three testing sets. We further show the performances on three different types of synthetic images. The proposed method is able to achieve the best (marked in red) or the second best (marked in blue) performances on either the real-world or synthetic data.

| Model | Zhang *et al.*[31] PSNR | SSIM | $SIR^2$ [24] PSNR | SSIM | Syn. Focused PSNR | SSIM | Syn. Defocused PSNR | SSIM | Syn. Ghosting PSNR | SSIM | Syn. All PSNR | SSIM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CEILNet [2] | 18.555 | 0.725 | 19.727 | 0.781 | 14.365 | 0.636 | 14.171 | 0.664 | 14.721 | 0.662 | 14.339 | 0.656 |
| [2] fine-tuned | 16.753 | 0.711 | 17.663 | 0.740 | **19.524** | **0.742** | **20.122** | **0.735** | **19.685** | **0.753** | **19.777** | **0.743** |
| Zhang *et al.*[31] | **20.744** | **0.784** | **24.271** | **0.868** | 12.345 | 0.602 | 11.317 | 0.570 | 12.909 | 0.635 | 12.231 | 0.605 |
| [31] fine-tuned | 18.032 | 0.737 | 20.265 | 0.835 | 17.090 | 0.712 | 18.108 | 0.758 | 17.882 | 0.738 | 17.693 | 0.736 |
| BDN [29] | 18.136 | 0.726 | 20.866 | 0.806 | 14.258 | 0.632 | 14.053 | 0.639 | 14.786 | 0.660 | 14.301 | 0.652 |
| Ours | **21.283** | **0.818** | **23.707** | **0.855** | **21.064** | **0.770** | **22.896** | **0.840** | **21.008** | **0.780** | **21.656** | **0.796** |

the accuracy shows the percentage of the synthetic images classified as the real-world ones in each set. We can see that the proposed synthetic method outperforms all the linear methods, achieving a closer performance to the real images. Some reflection synthesis examples can be found in Fig. 5.

**Reflection removal.** We also evaluate the proposed removal method on three testing sets with respect to PSNR and SSIM. The quantitative results are shown in Table 2. First, we compare the proposed method on two standard benchmarks, $SIR^2$ [2] and Zhang *et al.*[31]. These two datasets are collected in a similar way, and thus the reflection images show similar reflection type. The proposed method achieves the best on Zhang *et al.*[31], and the second best on $SIR^2$ [2]. For our synthetic dataset, we further separate it to three different reflection types. Interestingly, we can see all the methods perform better on the defocused type. This is similar to human, as the transmission and reflection layers show different imaging features (clear vs. blurry), so that they can be easier separated. The proposed method performs the best on all the three scenarios. These results demonstrate that the proposed method is able to handle different types of reflections, either they are real

or synthetic.

In Table 2, we also show the fine-tuned results of CEIL-Net [2] and Zhang *et al.*[31] (BDN [29] does not provide training codes) on our synthetic training set. Surprisingly, fine-tuned with our synthetic data decreases the performance on the datasets of $SIR^2$ [2] and Zhang *et al.*[31]. This is mainly because these two datasets constructed with a similar type of reflections, and training with the other types may leading to learning ambiguity. On the other hand, state-of-the-art methods cannot achieve as good performance as ours in the synthetic test set. This demonstrates the importance of predicting alpha blending mask and the supervision of the reconstruction loss.

Table 3: User study on the removal results. The preference rate shows the percentage of users that prefer our results over the competitor.

| | Preference rate |
|---|---|
| Ours > CEILNet [2] | 84.6% |
| Ours > Zhang *et al.*[31] | 73.8% |
| Ours > BDN [29] | 78.7% |

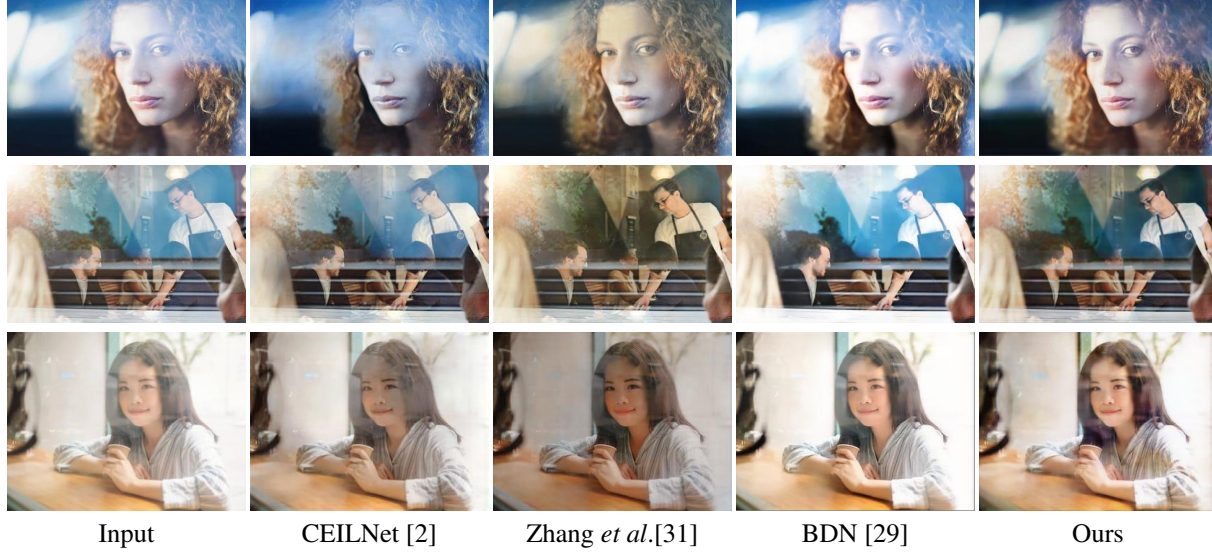|   Input   |   CEILNet [2]   |   Zhang *et al*.[31]   |   BDN [29]   |   Ours   |

Figure 7: Qualitative comparisons on real-world reflection images collected from the Internet.

### 5.1.2  Qualitative Evaluations

Fig. 5 shows results on our synthetic dataset. We show three types of reflections and their corresponding reflection-free images. It can be seen that the proposed method is able to handle different types of reflections, while state-of-the-arts fail to remove reflections on all the three types. In Fig. 6, we also show the results on the dataset collected from Zhang *et al*.[31]. This dataset mainly simulates the focused reflection, and we achieve comparable performance to the others on this conditioned scenario.

We also examine the proposed method on the real-world reflection images collected from the Internet. These results are mainly used for conducting a user study. For each evaluation, we compare our method to one competitor (three in total) on these real-world reflection images following the set of Zhang *et al*.[31]. Each user is presented with an original reflection image, our predicted transmission layer and the transmission layer by the competitor. The user needs to choose the image which is more like the reflection-free image. There are 25 real-world reflection images presented in the comparisons. The user study results are shown in Table. 3. The results are statistically significant with $p < 10^{-3}$ and 30 users participate in the user study. Some examples are also shown in Fig. 7.

### 5.2. Ablation Study

For better analysing the objective function of *RmNet*, we remove three losses one by one. We re-train new models with the modified losses. The ablation study is shown in 4. We observe that $L_{reconstr}$ and $L_{grad}$ enhance the generality of *RmNet* in both the real-world and synthetic cases, and both the losses show different contributions to the removal performance. Our complete objective function show the best results.

Table 4: Ablation studies on three testing sets. Each loss contributes to the reflection performance, while combining all of them achieves the best result.

| Model | Zhang *et al*.[31] | | $SIR^2$ [24] | | Syn. All | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| $L_{pixel}$ only | 18.684 | 0.727 | 20.833 | 0.780 | 19.413 | 0.762 |
| w/o $L_{reconstr}$ | 19.029 | 0.752 | 21.011 | 0.805 | 20.274 | 0.774 |
| w/o $L_{grad}$ | 19.303 | 0.748 | 21.276 | 0.813 | 20.229 | 0.766 |
| Complete | **21.283** | **0.818** | **23.707** | **0.855** | **21.656** | **0.796** |

## 6. Conclusion

In this paper, we revisit the linear combination problem of single image reflection removal. Particularly, we develop a reflection synthesis network to predict a non-linear alpha blending mask. In this way, it is able to generate images with different reflection types. Based on the synthesized diverse data, we propose a multi-branch reflection removal network. This network predicts the alpha blending mask as the side output, which makes the training a closed loop, so that it can be supervised by the reconstruction loss. Quantitative and qualitative evaluations on four datasets show that the proposed method is able to handle different types of reflections and outperform the state-of-the-arts in both the real-world and synthetic scenarios.

# References

[1] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Süsstrunk. Single image reflection suppression. In *CVPR*, pages 1752–1760, 2017.

[2] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David P Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *ICCV*, pages 3258–3267, 2017.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[4] Byeong-Ju Han and Jae-Young Sim. Reflection removal using low-rank matrix completion. In *CVPR*, volume 2, 2017.

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

[6] Eugene Hecht. *Optics (4th Edition)*. Addison Wesley, 4 edition, Aug. 2001.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

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

[10] Anat Levin, Assaf Zomet, and Yair Weiss. Learning to perceive transparency from the statistics of natural scenes. In *NeurIPS*, pages 1271–1278, 2003.

[11] Anat Levin, Assaf Zomet, and Yair Weiss. Separating reflections from a single image using local features. In *CVPR*, volume 1, pages 306–313, 2004.

[12] Yu Li and Michael S. Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, pages 2432–2439, 2013.

[13] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *CVPR*, pages 2752–2759, 2014.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[15] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MIC-CAI*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

[16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, pages 2234–2242, 2016.

[17] Yoav Y. Schechner, Joseph Shamir, and Nahum Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *ICCV*, volume 2, pages 814–819, 1999.

[18] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T. Freeman. Reflection removal using ghosting cues. In *CVPR*, June 2015.

[19] Ofer Springer and Yair Weiss. Reflection separation using guided annotation. In *ICIP*, pages 1192–1196, 2017.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[21] T. Tsuji. Specular reflection removal on high-speed camera for robot vision. In *ICRA*, pages 1542–1547, 2010.

[22] M. A. C. Tuncer and A. C. Gurbuz. Ground reflection removal in compressive sensing ground penetrating radars. *ICIP*, 9(1):23–27, 2012.

[23] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[24] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. In *ICCV*, 2017.

[25] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Crrn: Multi-scale guided concurrent reflection removal network. In *CVPR*, pages 4777–4785, 2018.

[26] Renjie Wan, Boxin Shi, Tan Ah Hwee, and Alex C Kot. Depth of field guided reflection removal. In *ICIP*, pages 21–25, 2016.

[27] R. Wan, B. Shi, T. A. Hwee, and A. C. Kot. Depth of field guided reflection removal. In *ICIP*, pages 21–25, 2016.

[28] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T. Freeman. A computational approach for obstruction-free photography. *ACM TOG*, 34(4):1–11, 2015.

[29] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *ECCV*, pages 654–669, 2018.

[30] Jiaolong Yang, Hongdong Li, Yuchao Dai, and Robby T Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *CVPR*, pages 1410–1419, 2016.

[31] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. *CVPR*, 2018.