Distilling Reflection Dynamics for Single-Image Reflection Removal

Quanlong Zheng[1] Xiaotian Qiao[2] Ying Cao[2] Shi Guo[3,4] Lei Zhang[3,4] Rynson Lau[2]
[1]OPPO Research [2]City Unversity of Hong Kong
[3]The Hong Kong Polytechnic University [4]DAMO Academy, Alibaba Group

(a) Multi-view sequence (b) IBCLN [17] (c) Ours (d) GT

Figure 1: Given an image sequence of a static scene captured by moving a camera (a), observing the appearance change in the reflection regions (highlighted by red boxes) will make it easier to separate the transmission layer from the reflection layer. Such appearance change can be regarded as a result of moving the transmission layer and reflection layer along different trajectories. We learn such reflection dynamics from multi-view image sequences for single-image reflection removal (SIRR), enabling our method to remove the reflection more accurately (c) than the state-of-the-art method (b).

Abstract

*Single-image reflection removal (SIRR) aims to restore the transmitted image given a single image shot through glass or window. Existing methods rely mainly on information extracted from a single image along with some predefined priors, and fail to give satisfying results on real-world images, due to inherent ambiguity and lack of large and diverse real-world training data. In this paper, instead of reasoning about a single image only, we propose to distill a representation of reflection dynamics from multi-view images (*i.e.*, the motions of reflection and transmission layers over time), and transfer the learned knowledge for the SIRR problem. In particular, we propose a teacher-student framework where the teacher network learns a representation of reflection dynamics by watching a sequence of multi-view images of a scene captured by a moving camera and teaches a student network to remove reflection from a single input image. In addition, we collect a large real-world multi-view reflection image dataset for reflection dynamics knowledge distillation. Extensive experiments show that our model yields state-of-the-art performances.*

1. Introduction

Reflection often appears when taking photos through glass or windows. It can severely ruin the visual quality of a photograph. The reflection formation process is formulated as $I = T + R \otimes k$, where I is the reflection image, T is the reflection layer, R is the transmission layer and k is the blurriness kernel. The objective of single-image reflection removal (SIRR) task is to recover the desired transmission layer, given an observed reflection image, which is an ill-posed problem. The content of a reflection may also be affected by many factors, such as scene content, glass property, camera angle and lighting condition, which make the problem very challenging.

Due to the ill-posed nature of SIRR, most of the existing works utilize priors that are either hand-crafted [28, 20, 16] or learned from large datasets [32, 33, 41, 34]. Nevertheless, they still largely rely on information extracted from single images and may generate degraded results when applied to real-world images.

We observe that humans often look at a scene from multiple viewpoints in order to disambiguate the transmitted content from the reflected content. For example, if we just look at the the middle image in Figure 1(a), it may be difficult to separate the transmitted/reflected contents. However, if we have access to the sequence of images captured by moving a camera from left to right, and compare their local appearance changes, the layer separation task will become much easier. For example, by looking at the contents of the red regions in Figure 1, we can observe that the reflected content shifts a lot to the left while the transmitted content only shifts a little to the left. We refer to the motion of the reflection and transmission layers over time as *reflection dynamics*. This implies that given a *static* scene and *small* camera motion, reflection dynamics captured by a sequence of multi-view images can provide useful cues for reflection removal. Through learning such reflection dynamics, we can recover the transmission layer more reliably, as shown in Figure 1(c).

Inspired by this observation, we propose a learning framework for SIRR, by learning and transferring the

knowledge of reflection dynamics from multi-view image sequences. While some prior works have explored the idea of using multiple images for reflection removal/layer separation [14, 1, 8, 19], they require multiple images as input during inference and thus are not applicable to the SIRR problem. In contrast, despite being trained on multi-view image sequences, our method only observes a single input image at test time. Our framework is built upon a teacher-student framework where we use multi-view image sequences to train a teacher network to learn a representation of reflection dynamics and a student network to hallucinate the representation from a single input image for SIRR. At the crux of the teacher network is a newly proposed multi-view knowledge learning module, which encourages a representation that encodes reflection dynamics to emerge by explicitly predicting the motions of reflection and transmission layers across different views without direct supervision. Compared to traditional single-view methods, our model can leverage motion information from the hallucinated representation, even only given a single test image.

To train our model, we collect a large, real multi-view reflection image dataset, named Seq1K, containing 1015 image sequences from different real-world scenes. Each sequence comprises three frames of a scene captured by a slowly moving camera, with the ground truth transmission layer of the center frame. To learn reflection dynamics more effectively, we further propose an approach to generate synthetic multi-view images. Our experimental results show that learning from the combination of real and synthetic images can lead to significant performance improvement.

To sum up, we make the first attempt to distill reflection dynamics knowledge from *multi-view* images for *single-image* reflection removal, and propose a framework that learns a representation of reflection dynamics from multi-view image sequences and transfers it to single static images. Moreover, we contribute a large-scale dataset of multi-view reflection image sequences, which can be used to train and evaluate both our model and traditional SIRR. Extensive evaluations on several benchmarks and our newly collected dataset show that our proposed method outperforms existing methods, achieving state-of-the-art results.

2. Related Work

Single-Image Reflection Removal (SIRR). Early methods usually handle it with pre-defined priors, *e.g.*, ghosting cues of the reflection layer [28], sparsity [16], and smoothness [20]. These methods depend highly on the asymmetry prior of the two layers, and their applicability is limited by their specific assumptions. Recently, learning-based methods are popular for reflection removal. These methods learn powerful features from a large amount of data with well-designed networks [41, 34, 5, 32, 41, 31, 17]. To ease learning, some auxiliary cues are used. For example, edge information is explored by [5, 32], which is either jointly predicted with the transmission layer [32] or predicted using two stages [5]. Auxiliary semantic features are also utilized in [41, 34, 17], and usually extracted from a VGG network. Later, auxiliary layer information is widely studied by [39, 35, 22, 11, 17]. This information can be learned by one network first [39], or jointly predicted with the target image to be constraint by a well-designed reconstruction loss [35, 22, 11], or used to iteratively refine the target image [17]. Some methods used gated network [26] or hybrid network [25] for image dehazing or image enhancement. Although learning-based methods achieve good performances, they all try to address the problem from a single-image perspective. In contrast, we train our model on image sequences to learn reflection dynamics knowledge and use it for the SIRR problem.

Multi-Image Reflection Removal. Another line of work exploits multiple input images to alleviate the ill-posed nature of reflection removal. Specific multiple images are explored, *e.g.*, flash/unflash pair images [1], polarized images [14, 36, 15, 12] and etc. Special physical cues from multiple images [27, 1, 8, 19] are also widely studied, including the depth information from multiple images [27], the different motion information of reflection and transmission layers [8], "defocus-disparity" cues [36] and etc. In general, multi-image-based methods can achieve good performances. However, all these methods require multiple images as input, which renders them not applicable to the SIRR problem. In contrast to these works, our framework uses multiple images only in the training stage, and takes as input a single image during inference.

Video Reflection Removal. Motion cues are often used to tackle video reflection removal [7, 38, 6, 2, 24]. They are used under the assumption that the motion fields of the two layers are different. Hence, they can help separate the two layers and reduce the artifacts via temporal coherency constrains. Similar in spirit to these works, we also take advantage of the motion information (*i.e.*, reflection dynamics in our work) captured from image sequences to help separate an image into different layers. Unlike real videos that contain both object and camera motions, we assume that only the camera is moving in our image sequences. This means that we factor out object motion and learn to exploit camera motion only for layer separation. As such, these methods cannot be easily adapted to address our problem, as their input is a video, instead of a single image.

Knowledge Distillation. Knowledge distillation was originally for compressing a large model (teacher model) to a compact one (student model) [9, 30]. It has been widely used in the image classification task via the student model to mimic the output distribution of the teacher model. In addition, knowledge distillation is also used in other vision tasks, *e.g.*, pedestrian re-identification [3], semantic segmentation [37, 21] and object detection [18]. In our work, we aim to distill reflection dynamics from image sequences

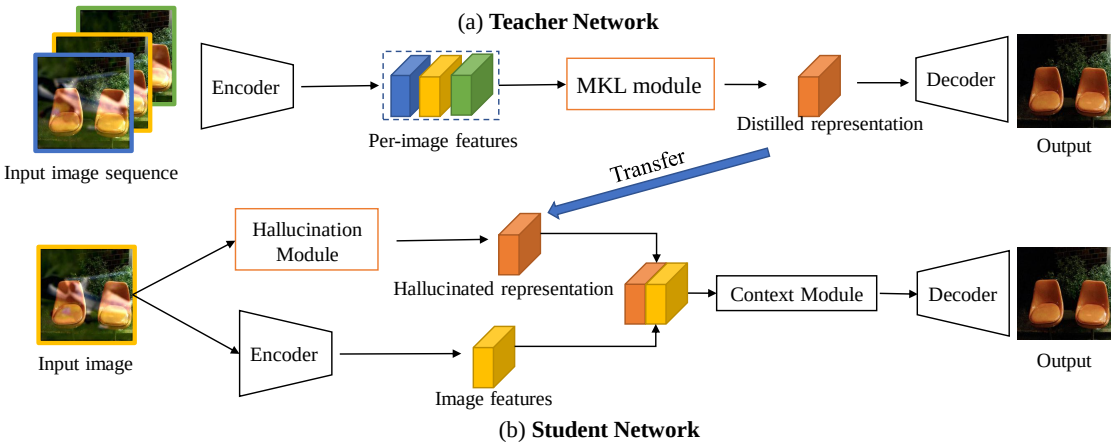

Figure 2: Overview of our learning framework. (a) Teacher Network. The input is a multi-view image sequence and the output is the estimated transmission layer of the middle image of the input sequence. (b) Student Network. The input is a single image (or the middle image of the sequence during training) and the output is the estimated transmission image.

and apply it to single images for reflection removal via a teacher-student paradigm.

3. Learning Framework

Given an input image $\mathcal{I}^{H\times W\times 3}$, where H and W are the image height and width, reflection removal aims to recover its transmission layer $\mathcal{T}^{H\times W\times 3}$. Our key idea is to capitalize on information about reflection dynamics (*i.e.*, the motions of the transmission and reflection layers) contained in multi-view image sequences to tackle the SIRR problem. Our framework is based on a teacher-student paradigm as shown in Figure 2. The teacher model observes a multi-view image sequence and explicitly learns a reflection dynamics representation that is useful for separating the transmission layer from the center image of the sequence. The learned representation is then used to teach a single-view student model to remove reflection from a single input image. During training, the teacher and student networks are coupled. However, during inference, the student network will be used as a stand-alone SIRR model.

3.1. Teacher Network

Our teacher network takes as input a sequence of N multi-view images and outputs a transmission image. We use $N = 3$ in our work, which we empirically find to work the best. It is composed of one encoder, a multi-view knowledge learning (MKL) module, and a decoder. The encoder is used to extract image-wise visual features from the input sequence of multi-view images. The MKL module learns a representation that captures reflection dynamics from the sequence of image features. The decoder then decodes the representation to output a transmission image.

3.1.1 Encoder

The encoder is to extract deep image features. Given an input image sequences (I_{t-1}, I_t, I_{t+1}), it outputs per-image features (f_{t-1}, f_t, f_{t+1}). The encoder is composed of 4 convolutional layers, each of which has 256 kernels of size 3 x 3. Note that, unless otherwise stated, all convolutional layers in our model are followed by a ReLU nonlinearity. To avoid color attenuation/shifting problems, we do not use any normalization layers as in [34].

3.1.2 Multi-view Knowledge Learning (MKL) Module

The MKL module is the core module to distill reflection dynamics knowledge from the encoded image features. Its output is a representation. To enforce that the learned representation encodes the reflection dynamics, we explicitly predict the motions of the reflection and transmission layers across the input sequence. However, a key challenge is that we do not have direct supervision on the motions. To address this, we design the MKL module to learn the layer motions jointly with feature disentanglement of the reflection and transmission layers in each image. This allows us to encourage the learned layer motion to be plausible by performing temporal alignment of the corresponding layers adjacent in time in the learned feature space.

Figure 3 shows the overview of the module, which comprises a decouple net, a flow net and a fusion net. The per-image features are first fed into the decouple Net, producing two groups of features, $\{\Phi_R, \Psi_R\}$ and $\{\Phi_T, \Psi_T\}$, for the transmission and reflection layers, respectively, of each image. For a pair of adjacent images (*e.g.*, I_{t-1} and I_t), the disentangled features are passed through the flow net to predict two motion fields from one image to another, one for the reflection layer and the other for the transmission layer. The features from the two flow nets, one for each

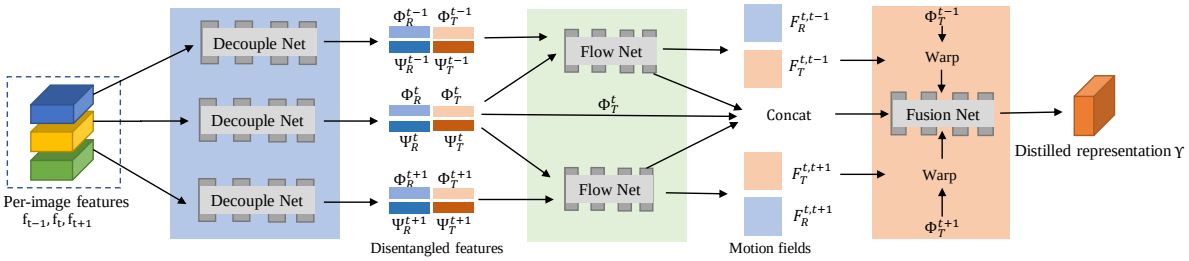

Figure 3: Multi-view Knowledge Learning Module.

adjacent image pair, are then fed into the fusion net. The fusion net outputs a representation, by combining the flow net features, the transmission features of the center image, and the warped transmission features of the two nearby images according to the estimated motion fields.

Decouple Net. The decouple net aims to decouple an input image into two sets of features for the transmission and reflection layers of the image. It follows an encoder-decoder architecture. The image features are equally split to two sets, Φ_R and Φ_T, which capture the visual semantics of the reflection and transmission layers, respectively. The features are then fed into two decoders, G_T and G_R, to reconstruct the reflection and transmission layers. The two decoders share the same architecture, which contains 1 deconvolutional layer (128 kernels of size 4×4) and 3 convolutional layers (128 kernels of size 3×3).

Let Φ_T^t and Φ_R^t be the transmission and reflection features of image I_t. To learn disentangled features, we introduce a feature decoupling loss that measures how close the reconstructed layers are to their ground truth O_T^t and O_R^t:

$$\mathcal{L}_{\mathcal{FD}}^t = l(G_T(\Phi_T^t), O_T^t) + l(G_R(\Phi_R^t), O_R^t), \quad (1)$$

where $l(x, y) = ||x-y||_1 + ||\bigtriangledown x - \bigtriangledown y||_1$ is the reconstruction loss in terms of both image and gradient domains. We may also apply this loss to the other two images, I^{t-1} and I^{t+1}, if the ground truth transmission and reflection layers for them are available. In this case, the feature decoupling loss of the sequence is defined as:

$$\mathcal{L}_{\mathcal{FD}} = \mathcal{L}_{\mathcal{FD}}^{t-1} + \mathcal{L}_{\mathcal{FD}}^t + \mathcal{L}_{\mathcal{FD}}^{t+1}. \quad (2)$$

Flow Net. The flow net aims to predict how the transmission and reflection layers move between two adjacent views (*i.e.*, $\{I_t, I_{t-1}\}$ or $\{I_t, I_{t+1}\}$). In order to predict the layer motion well, the flow net should learn features that carry rich information of reflection dynamics from one view to another. The input to it is the concatenation of the disentangled high- and low-level features from the two views along the channel dimension. It has five convolutional layers: the first layer has 512 kernels of size 3×3, the following three layers have 128 kernels of size 3×3, and the last layer has 2 kernels of size 3×3, to output two 1-channel motion fields F_T and F_R for the transmission and reflection layers.

We apply a view synthesis loss to the predicted flow maps to ensure that they are able to model meaningful motion fields. We evaluate the quality of each motion field by measuring how well it can be used to warp one view to reconstruct another. Formally, let Φ_T^{t-1} and Φ_T^t be the transmission layer features of images I_{t-1} and I_t, respectively. The quality measure of the predicted flow map $M_T^{t,t-1}$ between the two transmission layers is defined as: $l_T^{t,t-1} = ||\Phi_T^t - warp(\Phi_T^{t-1}, M_T^{t,t-1})||_2$, where $warp(\cdot)$ is a warping process via bilinear interpolation. Similarly, we define a quality measure for $M_R^{t,t-1}$ as: $l_R^{t,t-1t} = ||\Phi_R^t - warp(\Phi_R^{t-1}, M_R^{t,t-1})||_2$. The view synthesis objective is then defined as:

$$\mathcal{L}_{\mathcal{VS}} = l_T^{t,t-1} + l_R^{t,t-1} + l_T^{t,t+1} + l_R^{t,t+1}, \quad (3)$$

where $l_T^{t,t+1}$ and $l_R^{t,t+1}$ are defined in the same way as $l_T^{t,t-1}$ and $l_R^{t,t-1}$, but for $M_T^{t,t-1}$ and $M_R^{t,t-1}$, respectively.

Fusion Net. It aims to fuse the reflection dynamics features and the transmission features of the three input images to learn a representation, which encodes not only the information of all the input images but also the reflection dynamics across them. To combine the features from the images at time steps $t-1$ and $t+1$, their transmission features are warped to align with the center image t based on the estimated corresponding motion fields. The fusion net first fuses the reflection dynamics features and transmission features of the center image by concatenating them along the channel dimension and passing them through three convolutional layers with 640, 128, 128 kernels of size 3×3. The output features are then combined with the warped transmission features and sent into two convolutional layers with 384, 128, 128 kernels, to output the distilled representation to be transferred to the student network.

3.1.3 Decoder

The decoder aims to transform the distilled representation into an output image, corresponding to the transmission layer of the center input image. It is composed of 2 up-sampling blocks. The first up-sampling block has one de-

convolutional layer, one convolutional layer and one pyramid pooling layer [42]. The pyramid pooling layer is used to obtain multi-scale spatial context and has been shown to be effective in the semantic segmentation task [42]. It is composed of 4 average pooling layers with kernel sizes 4, 8, 16, 32. The second up-sampling block has one deconvolutional layer with 256 kernels of size 4×4 followed by two convolutional layers with 256 kernels of size 3×3.

We supervise the output image using a reconstruction loss and a perceptual loss. The reconstruction loss is used to measure the pixel-wise difference between the predicted transmission layer $\hat{L}$ and the ground truth in the gradient domain and image domain:

$$\mathcal{L}_{\mathcal{R}} = \sum ||L - \hat{L}||_1 + ||\nabla L - \nabla \hat{L}||_1. \quad (4)$$

The perceptual loss measures the difference between the prediction and the ground truth in the feature space. Similar to [41, 34], our perceptual loss is computed based on "conv5_3" of VGG-19 [29]:

$$\mathcal{L}_{\mathcal{P}} = \sum ||\phi(L) - \phi(\hat{L})||_1, \quad (5)$$

where ϕ is the feature map of the "conv5_3" layer.

The full objective of the teacher network is then:

$$\mathcal{L}_{teacher} = \alpha_t \mathcal{L}_{\mathcal{R}} + \beta_t \mathcal{L}_{\mathcal{P}} + \gamma_t \mathcal{L}_{\mathcal{FD}} + \zeta_t \mathcal{L}_{VS}. \quad (6)$$

3.2. Student Network

The student network takes as input the center image in the input sequence and learns to hallucinate a representation of reflection dynamics from the MLK module, so that it can be used to reconstruct the transmission layer of the input image well. We feed the input image into a hallucination module and a feature encoder to predict the distilled representation and extract visual features. The hallucination module consists of 5 convolutional layers of size 3×3 and (256, 256, 256, 256, 128) kernels. The predicted representation and visual features are then concatenated and send into a context module, which aims to fuse them by exploiting global context information across the channels of its input as in [34, 10]. It is composed of 10 residual channel-wise blocks, each having two convolutional layers followed by a channel-wise attention layer. The input features v are first fed into two convolutional layers of the residual channel-wise block to obtain features $v^{'}$, which are then fed to a channel-wise attention block [34, 10] to generate a channel-wise attention vector u. $v^{'}$ is multiplied by u element-wise to obtain new features $v^{''}$, which are added to v. The output from the context block is converted to the output image via a decoder that has the same architecture as the decoder in the teacher network.

To train the student network, we introduce a transfer loss and an adversarial loss. The transfer loss aims to minimize the difference between the output of the hallucination module in the student network f_s and the MKL module in the teacher network f_t:

$$\mathcal{L}_T = \sum ||f_s - f_t||_1. \quad (7)$$

The adversarial loss aims to reduce the gap between the hallucinated and distilled representations. To stabilize the training process, we use the least-square loss [23]:

$$\mathcal{L}_{\mathcal{G}} = E_{x \sim p_{fake}(x)}[(D(x) - 1)^2], \quad (8)$$

where D is a discriminator that distinguishes between real and fake representations. We adopt the discriminator setup in PatchGAN [43], which operates on overlapping patches from the input. The loss for D is formulated as:

$$\mathcal{L}_{\mathcal{D}} = E_{x \sim p_{fake}(x)}[(D(x))^2] + E_{y \sim p_{real}(y)}[(D(y) - 1)^2]. \quad (9)$$

We train the student network by minimizing the following objective function:

$$\mathcal{L}_{student} = \alpha_s \mathcal{L}_{\mathcal{R}} + \beta_s \mathcal{L}_{\mathcal{P}} + \gamma_s (\mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\mathcal{D}}) + \zeta_s \mathcal{L}_T. \quad (10)$$

4. Datasets

To train our network, we collect a large-scale multi-view reflection image dataset by capturing real-world scenes with commodity cameras. We also propose a method to generate synthetic multi-view reflection images to facilitate the feature disentanglement learning of the MKL module.

4.1. Real-world Multi-view Reflection Dataset

To construct our real dataset (*i.e.*, Seq1K), we use two cameras, Nikon D810 and Google Pixel2, to capture images of 521 indoor and 494 outdoor scenes. The cameras are set to have ISO 100-3200 and apertures f/4.0 - f/16. Since we are interested in image sequences of stationary scenes (without object motion), whose appearance change is dominated only by camera movement, we carefully select the scenes so that they contain as few moving foreground objects as possible. To make our dataset cover a wider range of lighting conditions, we also capture at different times of a day to obtain both normal-light and low-light images. We end up with a real dataset of 1,015 multi-view image sequences $D_{real} = \{(I_{t-1}, I_t, I_{t+1}, T_t)\}$, resulting in a total of 4,060 images of resolution $1,760 \times 1,160$. We randomly select 715 sequences for training and the rest for testing. Note that since the ground truth transmission image of the center view in every sequence is available, our dataset can also be thought of as containing individual reflection images with the corresponding ground truth. Hence, it can be used to train and evaluate existing SIRR methods. However, our dataset contains $1,015$ reflection images, instead of just tens or hundreds of images in existing SIRR datasets [41, 34, 31].

4.2. Synthetic Multi-view Reflection Dataset

Our real dataset contains only true transmission layers. To make our MKL more effectively learn the disentanglement of transmission and reflection features, we need to

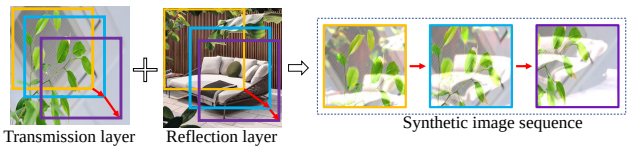

Figure 4: Synthesis of a multi-view image sequence.

have both true transmission and reflection layers for every image, as indicated by Eq. (1). For this, we propose to synthesize multi-view image sequences with realistic reflection dynamics. As shown in Figure 4, given a transmission layer and a reflection layer, we put a pair of fixed size windows on the two images, and move the windows in a synchronized manner along different trajectories, giving a sequence of cropped region pairs. We then blend each region pair using the reflection image synthesis method in [5] to form a sequence of reflection images. We end up with a synthetic dataset $D_{syn} = \{S_{t-1}, S_t, S_{t+1}\}$, where $S_t = (\widetilde{I}_t, T_t, R_t)$, and $\widetilde{I}_t$, T_t, R_t are the reflection image, its ground truth transmission and reflection layers, respectively.

Specifically, we uniformly split the VOC training dataset [4] into a transmission set and a reflection set. To generate an image sequence, we randomly sample two images from the two sets, respectively. The sliding windows are first placed randomly, and then shifted at each time step using a 2D offset vector whose x and y components are uniformly sampled from a range. We empirically use [0, 2] for the transmission layer and [10, 20] for the reflection layer, so that the synthetic sequences contain modest camera motion while still temporally stable. Note that our synthesis method may not be physically accurate since it does not consider geometry constraints, e.g., parallax. However, our synthetic sequences contain layer motions that can help separate different layers, and hence suffice for our purpose of learning to disentangle transmission and reflection features. Further, we also train on our real dataset to learn reflection removal, which can alleviate the potential physical inaccuracy. Some example synthetic sequences are shown in the supplemental.

5. Experiments

We introduce the experimental setup in Section 5.1. We then compare our proposed method with existing methods in Section 5.2, and evaluate the effectiveness of the MKL and network variants in Section 5.3.

5.1. Experimental Setup

For training, we empirically set the loss weights $\alpha_t, \beta_t, \gamma_t, \zeta_t$ to $1, 1, 1, 0.1$ and $\alpha_s, \beta_s, \gamma_s, \zeta_s$ to $1, 1, 1, 1$. We adopt the Adam solver [13] to optimize the model parameters. We set the initial learning rate as 0.0002 and decay it by 0.1 every 40 epochs. We first train the teacher model for 80 epoches with a batch size of 10. We keep the weights of the teacher networks fixed, and then train the student model for 80 epochs with a batch size of 1. We train our network using a mixture dataset like [41, 34, 17], *i.e.*, using our synthetic and Seq1K dataset to train the teacher network, and the synthetic dataset, Seq1K, Real20 [41] and Nature [17] to train the student network. The training images are horizontally flipped in a random manner. We quantitatively evaluate our results using SSIM and LMSE as [41, 5, 35] on several popular SIRR datasets, including SIR^2 [31], Real20 [41], Nature [17], and our real dataset (*i.e.*, Seq1K).

5.2. Comparison with State-of-the-art Methods

We compare our method with leading SIRR methods, including CEILNet [5], Zhang *et al.* [41], BDN [39], ERRNet [34], RmNet [35], Yang *et al.* [40], CoRRN [33], and IBCLN [17]. For fair comparison, we use the results generated via their provided codes. We re-train CEILNet, Zhang *et al.*, ERRNet and IBCLN on our training dataset. Since RmNet and CoRRN require ground truth reflection layers as supervision, we can only train them on our synthetic dataset. We note that the performance of CoRRN degrades after the re-training, and thus do not include these re-training results. Yang *et al.* [40] is not learning-based.

Results. Table 1 shows the quantitative results. In general, our method outperforms the other methods in most cases. Specifically, our method achieves the best SSIM and LMSE performances on Nature, SIR^2 and Seq1K datasets, the best SSIM performances on Real20 dataset and the second best LMSE performance on Real20 dataset. Our results are better than all other methods by a large margin on the SIR^2 dataset. Moreover, as shown in Table 1, for most of the previous methods, our datasets are able to offer huge performance gain on various real-world benchmarks, compared with the training datasets used in their own respective papers (*e.g.*, Real20 for Zhang *et al.*). This suggests that our training datasets have the benefit of improving the generalization ability of the conventional methods.

Figure 5 shows some qualitative results. We find that CoRRN, ZHANG *et al.* and CEILNet suffer from color attenuation or shifting problem. For example, in the 5th - 6th rows, CoRRN, ZHANG *et al.* and CEILNet predict a brighter/darker color for the building. Yang *et al.* tends to give overly smooth results. For example, in the 4th row, it smoothens the texture of the table. The possible reason is that their optimization uses the edge cue, but edge detection errors may lead to undesirable results. RmNet may ignore sharp reflections (the 5th row). RmNet removes reflection by predicting blending factors. Nevertheless, in the presence of sharp reflection, the blending factors are hard to estimate. BDN and IBCLN may fail to sharp reflections (the 5th rows) or strong reflections (the 1st - 2rd rows). This may be because they iteratively estimate the reflection and transmission layers, but initial reflection layers are hard to

Figure 5: Visual comparisons of our method against prior methods on real-world images.

	Real20		Nature		SIR^2		SIR^2 - Object		SIR^2 - Postcard		SIR^2 - Wild		Seq1K	
	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓
CEILNet [5]	0.7185	0.0294	0.7048	0.0218	0.8520	0.0080	0.8764	0.0054	0.8388	0.0088	0.8113	0.0148	0.8130	0.0179
Zhang *et al*. [41]	0.7988	0.0204	0.7401	0.0159	0.8454	0.0072	0.8847	0.0054	0.8082	0.0073	0.8369	0.0132	0.8522	0.0123
BDN [39]	0.7464	0.0297	0.7449	0.0190	0.8616	0.0062	0.8637	0.0054	0.8686	0.0054	0.8290	0.0120	0.8176	0.0186
RmNet [35]	0.7194	0.0306	0.7433	0.0169	0.8328	0.0097	0.8246	0.0108	0.8375	0.0081	0.8459	0.0119	0.8273	0.0154
ERRNet [34]	0.8036	0.0210	0.7590	0.0150	0.8807	0.0062	0.8872	0.0040	0.8786	0.0056	0.8644	0.0169	0.8711	0.0104
CoRRN [33]	0.7140	0.0299	0.7400	0.0154	0.8392	0.0055	0.8678	0.0045	0.8119	0.0054	0.8343	0.0096	0.7809	0.0158
Yang *et al*. [40]	0.7084	0.0287	0.7415	0.0162	0.8570	0.0065	0.8502	0.0065	0.8651	0.0059	0.8526	0.0087	0.7885	0.0189
IBCLN [17]	0.7816	0.0224	0.7845	0.0126	0.8948	0.0050	0.9020	0.0038	0.8880	0.0052	0.8934	0.0085	0.8738	0.0105
CEILNET (F)	0.7284	0.0267	0.7506	0.0151	0.8689	0.0057	0.8737	0.0044	0.8737	0.0053	0.8339	0.0118	0.8525	0.0117
Zhang (F)	0.7998	0.0202	0.8026	0.0118	0.8952	0.0044	0.9009	0.0036	0.8884	0.0043	0.8994	0.0071	0.8873	0.0083
RmNet (F)	0.7559	0.0245	0.7649	0.0136	0.8862	0.0049	0.8845	0.0041	0.8868	0.0053	0.8904	0.0065	0.8682	0.0104
ERRNet (F)	0.8120	0.0184	0.7950	0.0121	0.8940	0.0046	0.9028	0.0032	0.8810	0.0057	0.9090	0.0057	0.8934	0.0074
IBCLN (F)	0.7759	0.0231	0.7746	0.0131	0.8946	0.0043	0.9026	0.0032	0.880	0.0048	0.8892	0.0046	0.8672	0.0106
Ours	0.8196	0.0187	0.8213	0.0104	0.9009	0.0041	0.9089	0.0031	0.8908	0.0045	0.9084	0.0064	0.9015	0.0072

Table 1: Quantitative comparisons. We compare our method with state-of-art single image reflection removal methods, CEILNet [5], Zhang *et al*. [41], BDN [39], ERRNet [34], RmNet [35], Yang *et al*. [40], CoRRN [33], IBCLN [17] and the retrained versions of some methods on our training dataset (denoted as '(F)'). The performances are reported on four real datasets, Real20, SIR^2, Nature and Seq1K. We also present the results on the three subsets of SIR^2 (SIR^2 - Objects, SIR^2 - Postcard and SIR^2 - Wild). The best results are highlighted in red.

estimate for sharp and strong reflections. ERRNet works better than other existing methods in most cases due to the well-engineered network, but it may fail to remove reflections completely sometimes (*e.g*., the reflection on the toy

	Real20		Nature		SIR^2		Seq1K	
	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓	SSIM ↑	LMSE ↓
w/o MKL	0.7936	0.0210	0.8055	0.0121	0.8861	0.0050	0.8911	0.0079
w/o Decouple Net	0.8039	0.0195	0.8080	0.0115	0.8945	0.0049	0.8953	0.0076
w/o Flow Net	0.8069	0.0221	0.8096	0.0113	0.8946	0.0046	0.8967	0.0078
w/o Fusion Net	0.8108	0.0203	0.8185	0.0111	0.8957	0.0043	0.8976	0.0075
w/o KT	0.7944	0.0205	0.8068	0.0129	0.8794	0.0052	0.8859	0.0079
w/o Hallucination	0.8075	0.0200	0.8153	0.8111	0.8918	0.0047	0.8966	0.0074
w/o Encoder	0.8153	0.0194	0.8162	0.0106	0.8920	0.0048	0.8968	0.0075
Ours	**0.8196**	**0.0187**	**0.8213**	**0.0104**	**0.9009**	**0.0041**	**0.9015**	**0.0072**

Table 2: Results of the ablation study. The best results are in bold.

in the first row). Further, all the other methods fail to handle the situation with a dark background (3rd and 4th rows). In contrast, our method can favorably deal with various types of reflections, avoid color attenuation/shifting, and produce sharper outputs. The hallucinated reflection dynamics information provides extra useful cues that are lacked in a single static image, and thus enables our model to more accurately separate the reflection and transmission layers. Further, the use of the context block can help our recovered transmission layers to be more natural.

5.3. Ablation Study

In this section, we compare our method with the following variants to evaluate the effectiveness of our MKL module, its components, our knowledge transfer strategy and our student network design.

- w/o MKL. We remove the MKL and fuse the per-image features into a representation via a convolutional layer.
- w/o Decouple Net. We remove the decouple net in the MKL and send the per-image features to the flow net.
- w/o Flow Net. We remove the flow net in the MKL and directly concatenate and send the transmission features to the fusion net.
- w/o Fusion Net. We remove the fusion net and generate the distilled representation by directly concatenating the transmission features.
- w/o Knowledge Transfer (KT). We train the student network alone without the knowledge transfer loss.
- w/o Hallucination. We remove the hallucination module from the student network and train it alone with the knowledge transfer loss.
- w/o Encoder. We remove the student network's encoder.

From Table 2, we observe that our final model outperforms the alternative without the MKL module and the ablations of the MKL, indicating the effectiveness of our MKL module and necessity of all its components. Without the knowledge transfer from the teacher network to the student network, the performance significantly degrades, suggesting that distilling reflection dynamics knowledge from multi-view image sequences can greatly help SIRR. We also observe that without the hallucination module or encoder, the performance also degrades, demonstrating the effectiveness of our student network design.

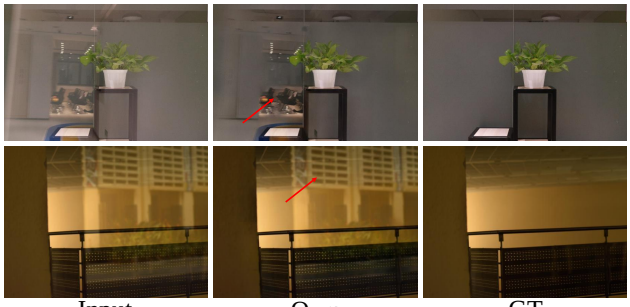

Figure 6: Failure cases. Our method may fail when a transmission layer is textureless and the reflection layer has strong texture.

6. Conclusion

In this paper, we propose to distill reflection dynamics knowledge from image sequences of scenes captured by a moving camera to address the SIRR problem. To this end, we propose a teacher-student framework, where the teacher network learns a reflection dynamics representation from multi-view image sequences with a newly proposed multi-view knowledge learning module and teaches a student network to remove reflection from single images. We also construct a large-scale real-world dataset of multi-view reflection image sequences for reflection dynamics distillation and for SIRR evaluation. Extensive experiments demonstrate the effectiveness of our method and the usefulness of the newly collected dataset for SIRR. Although our method works well in most situations, it may fail in the presence of a clear reflection caused by mixing a strongly textured reflection layer with a textureless transmission layer, as shown in Figure 6. For example, in the second row, given the input image only, it is rather difficult to tell which layer the building-like texture (*i.e.*, the reflection) belongs to. In this case, decoupling transmission and reflection layers is hard, resulting in inaccurate reflection dynamics prediction.

References

[1] Amit Agrawal, Ramesh Raskar, Shree K Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *TOG*, 2005. 2

[2] Jean-Baptiste Alayrac, Joao Carreira, and Andrew Zisserman. The visual centrifuge: Model-free layered video representations. In *CVPR*, 2019. 2

[3] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *AAAI*, 2018. 2

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 6

[5] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image

reflection removal and image smoothing. In *ICCV*, 2017. 2, 6, 7

[6] Kun Gai, Zhenwei Shi, and Changshui Zhang. Blind separation of superimposed moving images using image statistics. *PAMI*, 2011. 2

[7] Xiaojie Guo, Xiaochun Cao, and Yi Ma. Robust separation of reflection from multiple images. In *CVPR*, 2014. 2

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

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS*, 2015. 2

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5

[11] Soomin Kim, Yuchi Huo, and Sung-Eui Yoon. Single image reflection removal with physically-based training images. In *CVPR*, June 2020. 2

[12] Soomin Kim, Yuchi Huo, and Sung-Eui Yoon. Single image reflection removal with physically-based training images. In *ECCV*, 2020. 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[14] Naejin Kong, Yu-Wing Tai, and Joseph S Shin. A physically-based approach to reflection separation: from physical modeling to constrained optimization. *PAMI*, 2014. 2

[15] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *CVPR*, June 2020. 2

[16] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *PAMI*, 2007. 1, 2

[17] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E Hopcroft. Single image reflection removal through cascaded refinement. In *CVPR*, 2020. 2, 6, 7

[18] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017. 2

[19] Yu Li and Michael S Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, 2013. 2

[20] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *CVPR*, 2014. 1, 2

[21] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *CVPR*, 2019. 2

[22] Daiqian Ma, Renjie Wan, Boxin Shi, Alex C. Kot, and Ling-Yu Duan. Learning to jointly generate and separate reflections. In *ICCV*, 2019. 2

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 5

[24] Ajay Nandoriya, Mohamed Elgharib, Changil Kim, Mohamed Hefeeda, and Wojciech Matusik. Video reflection removal through spatio-temporal optimization. In *ICCV*, 2017. 2

[25] Wenqi Ren, Sifei Liu, Lin Ma, Qianqian Xu, Xiangyu Xu, Xiaochun Cao, Junping Du, and Ming-Hsuan Yang. Low-light image enhancement via a deep hybrid network. *TIP*, 2019. 2

[26] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *CVPR*, 2018. 2

[27] Yoav Y Schechner, Nahum Kiryati, and Ronen Basri. Separation of transparent layers using focus. *IJCV*, 2000. 2

[28] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *CVPR*, 2015. 1, 2

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 5

[30] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv:1603.05691*, 2016. 2

[31] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. In *ICCV*, 2017. 2, 5, 6

[32] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Crrn: Multi-scale guided concurrent reflection removal network. In *CVPR*, 2018. 1, 2

[33] Renjie Wan, Boxin Shi, Haoliang Li, Ling-Yu Duan, Ah-Hwee Tan, and Alex Kot Chichung. Corrn: Cooperative reflection removal network. *PAMI*, 2019. 1, 6, 7

[34] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7

[35] Qiang Wen, Yinjie Tan, Jing Qin, Wenxi Liu, Guoqiang Han, and Shengfeng He. Single image reflection removal beyond linearity. In *CVPR*, 2019. 2, 6, 7

[36] Patrick Wieschollek, Orazio Gallo, Jinwei Gu, and Jan Kautz. Separating reflection and transmission images in the wild. In *ECCV*, 2018. 2

[37] Jiafeng Xie, Bing Shuai, Jian-Fang Hu, Jingyang Lin, and Wei-Shi Zheng. Improving fast segmentation with teacher-student learning. *arXiv:1810.08476*, 2018. 2

[38] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *TOG*, 2015. 2

[39] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *ECCV*, 2018. 2, 6, 7

[40] Yang Yang, Wenye Ma, Yin Zheng, Jian-Feng Cai, and Weiyu Xu. Fast single image reflection suppression via convex optimization. In *CVPR*, 2019. 6, 7

[41] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *CVPR*, 2018. 1, 2, 5, 6, 7

[42] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 5

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 5