Neural Transmitted Radiance Fields

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural radiance fields (NeRF) has brought tremendous progress to novel view synthesis. Though NeRF enables the rendering of subtle details in a scene by learning from a dense set of images, it also reconstructs the undesired reflections when we capture images through glass. As a commonly observed interference, the reflection would undermine the visibility of the desired transmitted scene behind glass by occluding the transmitted light rays. In this paper, we aim at addressing the problem of rendering novel transmitted views given a set of reflection-corrupted images. By introducing the transmission encoder and motion inconsistency as guidance, our neural transmitted radiance fields can resist such reflection interference during rendering and reconstruct high-fidelity results even under sparse views. The proposed method achieves superior performance from the experiments on a newly collected dataset compared with state-of-the-art methods.

1 Introduction

Novel view synthesis plays a vital role in various computer vision applications. Tremendous progress has been witnessed recently with the introduction of neural radiance fields (NeRF) [1]. By modeling the emitted radiance and density of a scene from a set of images with a multilayer perceptron (MLP), it demonstrates an unprecedented level of fidelity on a range of challenging scenes. Despite its success in diverse scenarios, such a framework still faces challenges when working with glass, a medium which is frequently observed in photography and daily life [2] to provide see-through protection for desired scenes. However, glass's reflectivity may reflect undesired scenes on images captured through it. People always feel interested in the transmitted scenes behind the glass and want to remove the undesired reflection. This traps neural radiance fields into a dilemma: Its well-designed framework is capable of modeling every detail in a scene, including those *undesired* ones.

One plausible way to resolve this dilemma is opting for a post-processing tool to remove the undesired components before the captured images are fed into NeRF [1]. For example, existing reflection removal methods [3, 4, 5] can be adopted to suppress the undesired components and provide better visibility for transmitted scenes. However, the processed images from different views may violate the assumption of NeRF that the scene is photometrically static. Besides, the reflection cannot be completely removed in most cases by existing single-image-based reflection removal approaches [6], and the remaining reflection may be modeled by NeRF as a part of the scenes, undermining the quality of rendered results.

An alternative solution is to render with undesired components like NeRF-W [7] by separating the whole scene into static and transient elements with exclusive properties. However, the reflection and transmission usually present a highly "twisted" relationship since they may influence each other during light transportation and reflect such *entanglement* in captured images. From results shown in Figure 1(left), it may challenge the applicability of a purely unsupervised strategy in NeRF-W [7], thus showing degraded performance. Besides, NeRF-W [7] also partially relies on a *dense* set of multi-view images to conduct the separation. As an impractical demand inherited from NeRF [1],

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

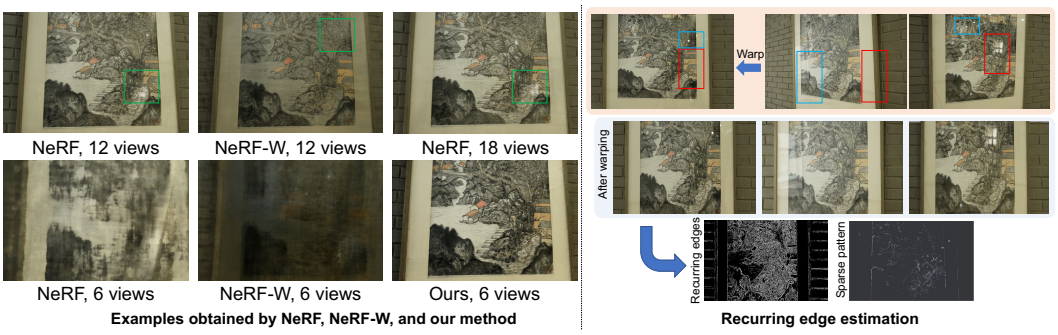

Figure 1: Left: The results obtained by NeRF [1] and NeRF-W [7] with 6 and 12 training views, the result obtained by NeRF [1] with 18 training views, and our result. The rendered results are with low reconstruction fidelity for NeRF [1] and NeRF-W [7] only with 6 and 12 training views. For NeRF [1] with 18 training views, the result shows higher fidelity, but the undesired reflection is also finally rendered (labeled by green box). Right: The recurring edge estimation process. By warping images from neighboring views into a predefined view, the recurring edges belonging to the transmission can be extracted to guide the novel transmitted view synthesis. The blue and red boxes denote the repeated transmission and sparse reflection pattern, respectively.

it contradicts the casual photographing experience, where only limited inputs with sparse views are available. The results in Figure 1(left) illustrate that NeRF-W [7] cannot perform high-fidelity synthesis with limited views, let alone effective separation between the transmitted scene and the reflected scene. Thus, how to render *novel transmitted scenes* with *sparse views* poses unique challenges.

In contrast to the exclusive properties required by NeRF-W [7], we employ motion inconsistency, a phenomenon verified by pioneering reflection removal methods [8, 9], to link the above issues by warping images with different viewpoints into a predefined target view. As shown in Figure 1 (right), given an aligned image sequence, the transmitted scenes appear in different views with a large overlapping area while the reflected scenes only have a sparse presence [8]. Besides, the aligned image features from the neighboring views can also be used for a more high-fidelity reconstruction of the target view. We, therefore, design a method supported by two pillars: 1) Edges of the repeated patterns, denoted as the recurring edges shown in Figure 1(right), that could be estimated as a pilot to guide the differentiation during rendering; 2) features from reference views, which are incorporated into rendering for high-fidelity reconstruction under sparse views.

Several modifications have to be conducted to accommodate the above designs. Since the transmission/reflection edges only show their significance in a certain area, the classical "pixel-wise" rendering framework cannot capture such internal physical properties. We employ a patch-based rendering framework, which aggregates the pixels from a certain area into a patch as the training batch. Besides, due to the entanglement between the transmission and reflection, the features extracted by a network pre-trained for general tasks [10] may introduce interference from feature level, which deteriorates the transmission/reflection disambiguation described before. We, therefore, propose a transmission encoder to penetrate the feature-level interference caused by the reflection and render the transmission structure more accurately. At last, since the strength of reflection tends to rely on the viewing direction, the weighting coefficients for the two layers are obtained from the viewing direction to combine transmission and reflection layers.

In summary, the above considerations lead to our Neural Transmitted Radiance Fields as depicted in Figure 2, and its main contribution can be concluded as follows:

- An approach to render novel transmitted views, which is capable of penetrating the reflection interference.
- A unified consideration derived from motion inconsistency for removing reflection/transmission ambiguity and increasing rendering fidelity of transmission.
- A dataset collected from the real world for evaluation.

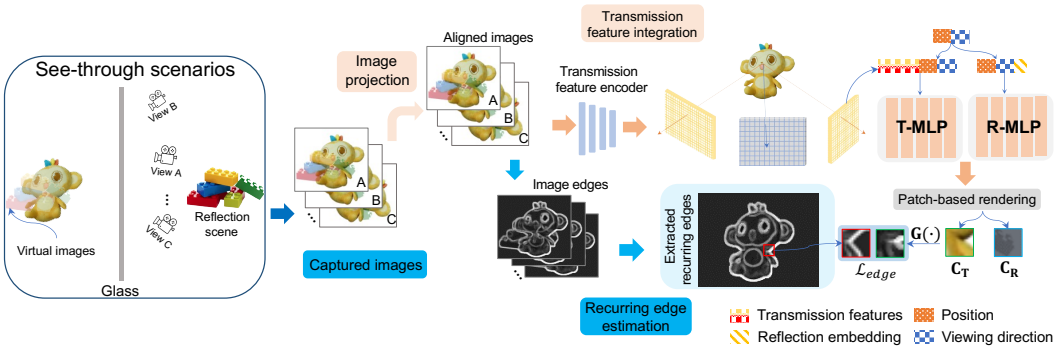

Figure 2: The proposed framework. Given images captured through glass from several different views, our method first projects the images at neighboring views to their target view (image projection). Next, a transmission feature encoder extracts the transmission features from the reference images (transmission feature integration). The extracted transmission features are then passed into the transmission MLP (T-MLP) for transmission rendering ($\mathbf{C_T}$). For the reflection rendering ($\mathbf{C_R}$), we feed the viewing direction and corresponding reflection embeddings into the reflection MLP (R-MLP). The recurring edges repeatedly appearing in the aligned image sequence are extracted to further differentiate the transmission and reflection based on the edge loss $\mathcal{L}_{edge}$ (recurring edges estimation).

2 Related work

2.1 Novel view synthesis with NeRF

Novel view synthesis aims at generating images from another viewpoint using a set of established views. The classical approaches rely on the explicit representations like multi-plane images [11, 12], meshes [13], or point cloud [14] to render novel views. Despite their rendering efficiency at test time, they usually suffer from the limited expressiveness within their representation. In contrast to the explicit representations, Neural Radiance Fields (NeRF) [1] utilize the implicit representation to represent 3D scenes using neural networks. For example, NeRF [1] approximates a continuous 3D MLP by mapping an input 5D coordinate to its corresponding scene properties. Given its promising representation ability, NeRF [1] has been extended to solving problems like relighting and scene editing. Recently, NeRF-W [7] proposes to adapt the original NeRF [1] to wild scenes by introducing an additional MLP to model the undesired transient element, which grants the original NeRF [1] extended ability to model the scenes with photometric variations and transient components. Since NeRF requires a large number of images for training, several methods [10, 15] are proposed to make it feasible with sparse views by utilizing the feature information from neighboring views. In this work, we aim to extend NeRF to see-through scenarios by eliminating negative interference caused by reflection.

2.2 Transmission recovery from reflection-corrupted images

How to recover the transmitted scene from reflection-corrupted images has been discussed for decades. Early methods [16, 17] make use of the edge difference to reduce the ambiguity between the transmission and reflection through optimization. Recent methods mainly adopt the deep learning framework to solve this problem. For example, Fan *et al.* [18] proposed a two-stage deep learning approach to learn the mapping between the mixture images and the estimated clean images. Wan *et al.* [19, 20] proposed a cooperative model to better preserve the background details. Zhang *et al.* [21] proposed a perceptual reflection removal method based on the generative adversarial network. Wei *et al.* [5] proposed another method to solve the reflection removal problem with non-aligned image pairs. Besides, some recently proposed [4] methods also try to focus on the reflection-corrupted regions by localizing the reflection first. The recently proposed method [22] also removed reflections under the guidance of an additional polarizer. Instead of solely relying on single images, plenty of methods [8, 23, 24, 25] also leverage advantages from the priors provided by multiple images to

remove reflections based on motion inconsistency. Such inconsistency observed from different views also allows our framework to differentiate transmission and reflection during rendering.

3 Preliminaries about Neural Radiance Fields

We first briefly review the settings in NeRF [1] and NeRF-W [7]. A NeRF represents a scene as a continuous volumetric radiance field f, which maps the position of any given point $\mathbf{x} = (x, y, z)$ and a viewing direction $\mathbf{d} = (\theta, \phi)$ into a color $\mathbf{c} = (r, g, b)$ and density σ as

$$f(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma). \tag{1}$$

Then, the expected color $\hat{\mathbf{C}}$ corresponding to the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ emitted from the center of projection of a camera at position $\mathbf{o}$ can be obtained as

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt, \tag{2}$$

where t_n and t_f denote the near and far bounds of rendering, and $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$ denotes the accumulated transmittance. NeRF-W [7] extends NeRF [1] with two MLPS to model the desired static (f_s) and undesired transient (f_t) components as

$$f_s(\mathbf{x}, \mathbf{d}, \ell_a) = (\mathbf{c}_s, \sigma_s) \text{ and } f_t(\mathbf{x}, \mathbf{d}, \ell_t) = (\mathbf{c}_t, \sigma_t). \tag{3}$$

Due to the introduction of the additional MLP for transient elements, as well as the embeddings ℓ_a and ℓ_t adapting to the various appearances and transient objects, NeRF-W [7] is capable of decomposing the desired static components during rendering.

For an image $\mathbf{I}$ captured in see-through scenarios, the relationship between the reflection $\mathbf{R}$ and transmission $\mathbf{T}$ can be formulated as $\mathbf{I} = \beta\mathbf{T} + \alpha\mathbf{R}$ [6], where β and α denotes the weighting coefficients. Applying NeRF-W [7] to achieve novel transmitted view synthesis is not directly applicable due to the following issues:

1. Dense view requirement: Both NeRF-W [7] and NeRF [1] inherit the substantial demand for dense multi-view images, and their performances degenerate under real-world scenarios where only limited input with sparse views are available.
2. Transmission and reflection entanglement: Due to the absorption, reflection, and refractive effect, $\mathbf{T}$ and $\mathbf{R}$ presents a highly entangled relationship, which cannot be simply disentangled by using an unsupervised model.

We propose to construct Neural Transmitted Radiance Fields to address these issues.

4 Neural Transmitted Radience Fields

Our goal is to learn radiance fields as a representation for the transmitted scene behind glass. To avoid the high separation ambiguity for transmission/reflection and low reconstruction fidelity under sparse views, we build a unified approach by warping pixels to the desired target views. By extracting the recurring edges after warping, our approach can get rid of the interference from the reflection. Meanwhile, a transmission feature integration scheme is introduced to merge the transmission features from the warped images, which enables higher reconstruction fidelity even under sparse views.

The pixel-wise rendering scheme in NeRF [1] and NeRF-W [7] lacks the ability to reflect the transmission/reflection correlation, since some statistical properties can only be observed in a certain area. To better consider the spatial correlation among neighboring pixels, we employ the patch-based rendering scheme. Specifically, during training we sample a $\mathcal{K} \times \mathcal{K}$ patch $\mathcal{P}(\mathbf{u}, s)$ in a given image following the way proposed in [26] as follows:

$$\mathcal{P}(\mathbf{u}, s) = \left\{ (sx + u, sy + v) \mid x, y \in \left\{ -\frac{\mathcal{K}}{2}, \ldots, \frac{\mathcal{K}}{2} - 1 \right\} \right\}, \tag{4}$$

where $\mathbf{u} = (u, v)$ denotes the center position of the patch, x and y range from $-\frac{\mathcal{K}}{2}$ to $\frac{\mathcal{K}}{2} - 1$ to create a sampling grid, and s denotes the scale of the image patch $\mathcal{P}$. In our setting, we apply a

combined sampling of grid sampling with $s = 1$ over the entire image and random sampling with $s > 1$, allowing image patches to capture spatial information of different scales. The aggregated pixels from the image patch are then used for training, during which the model can not only learn the colors and densities through one ray, but also get a thorough understanding towards the context.

4.1 Transmission feature integration

In contrast to NeRF [1] and NeRF-W [7], which solely rely on point coordinates and viewing direction as the input, we propose to address the fidelity issue by means of the additional guidance from reference images. Given a scene only with sparse views, the feature guidance from neighboring views can help to reconstruct novel target views with higher fidelity. In general, such image features are extracted by using a 2D CNN network [10] pretrained for general tasks. However, since the images under neighboring views are also captured through glass, a general CNN network for view synthesis like previous methods [15] may lead to feature-level interference during rendering.

As shown in Figure 2, we first project the neighboring views to their target view. To resist such interference during feature integration, we propose to obtain the transmission features by means of a transmission encoder η obtained by pre-training a reflection removal network, *e.g.*, ERRNet [5]. As a feature encoder pretrained for reflection removal task, η aims at extracting the transmission features as $\mathbf{W} = \eta(\mathbf{I})$, where $\mathbf{W}$ denotes the extracted feature volume. Instead of solely relying on the single-scale feature like pixel-NeRF [27], which suffers from view misalignment in our setting, we propose to build a feature pyramid as

$$\mathbf{W} = \{\mathbf{W}_g, \mathbf{W}_l\}, \tag{5}$$

where $\mathbf{W}_g$ and $\mathbf{W}_l$ denote the global and local features with different scales, respectively. In general, as CNNs naturally store local information in shallow layers and encode global information in deeper layers with a large receptive field, we extract $\mathbf{W}_g$ and $\mathbf{W}_l$ from the bottleneck layer and second last layer of η, respectively.

The extracted global and local transmission features for the i-th target view are then passed into the MLP network for the transmission rendering (T-MLP) $f_{\mathbf{T}}$, along with the position and viewing direction, as

$$f_{\mathbf{T}}(\mathbf{x}, \mathbf{d}, \mathbf{W}_g^{(i)}, \mathbf{W}_l^{(i)}) = (\mathbf{c}_t^{(i)}, \sigma_t^{(i)}), \tag{6}$$

where $c_t^{(i)}$ and $\sigma_t^{(i)}$ are the estimated color and density of the transmitted scene for the i-th view. As for reflection, a learnable reflection embedding $\ell_r^{(i)}$ is introduced for each view i, and the color $c_r^{(i)}$ and density $\sigma_r^{(i)}$ of the reflection scene are estimated using an additional MLP network for the reflection rendering (R-MLP) as

$$f_{\mathbf{R}}(\mathbf{x}, \mathbf{d}, \ell_r^{(i)}) = (\mathbf{c}_r^{(i)}, \sigma_r^{(i)}), \tag{7}$$

where the reflection embedding is optimized along with the weights of $f_{\mathbf{R}}$.

Similar to existing NeRFs [1, 7], we construct a self-consistent loss as follows:

$$\mathcal{L}_{sc} = \sum_{p(\mathbf{r}) \in \mathcal{P}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_1, \tag{8}$$

where $p(\cdot)$ denotes the mapping from a ray to its corresponding pixel position, and $\hat{\mathbf{C}}(\mathbf{r})$ denotes the observed color and $\mathbf{C}$ denotes the composite of $\mathbf{C_T}$ and $\mathbf{C_R}$ physically defined as

$$\mathbf{C} = \beta \mathbf{C_T} + \alpha \mathbf{C_R}. \tag{9}$$

Here α denotes the weighting coefficients to balance $\mathbf{C_T}$ and $\mathbf{C_R}$. For simplicity, we directly set $\beta = 1 - \alpha$ like previous reflection removal methods [28].

Several factors during light transportation can influence the weighting coefficients α. In our setting, since the strength of reflection largely depends on viewing direction [29] rather than the exact camera position, we propose to model such influence by a simple MLP that outputs the weighting coefficients given corresponding viewing direction encoding as $f_\alpha(\mathbf{x}, \mathbf{d}) = \alpha$.

4.2 Recurring edge constraints

The settings described above ensure a high-fidelity novel view reconstruction even under sparse views. However, since no reflection removal model can reach a completely reflection-free status, the reflection residuals left by the transmission feature encoder may further deteriorate the ambiguity issue. We then utilize the benefits provided by the aligned image sequence to further disentangle their twisted relationship.

As pointed by pioneer methods [9, 28], when multiple images with different viewpoints are captured through glass, the transmission appears frequently across all samples, while the reflection may only have sparse presence. This property makes the recurring transmission edges become a "pilot" to relieve ambiguity between the transmission and reflection. In our method, we first transform aligned images into their corresponding edge domain and measure the frequency of edge occurrence as

$$\phi(\mathbf{z}) = \frac{\sum_{i=1}^{k} G_i(\mathbf{z})^2}{(\sum_{i=1}^{k} G_i(\mathbf{z}))^2}, \tag{10}$$

where $G_i(\mathbf{z})$ denotes the gradient magnitude at pixel location $\mathbf{z}$ of view i, and k denotes the total number of pixels. It measures the sparsity of the vector which achieves its maximum value of 1 (when only one non-zero item exists) and achieves its minimum value of $1/k$ (when all items are non-zero and have identical values). Using this measurement, each edge pixel is assigned with two probabilities regarding whether it belongs to the transmitted scene or not.

Recognizing the transmission and reflection poses a great challenge for a vanilla NeRF network aiming at multi-view consistency, we provide our network with a likelihood estimation of the transmission and reflection, following the strategy proposed in [28]:

$$\begin{aligned} l_{\mathbf{T}_i}(\mathbf{z}) &= s(-(\phi(\mathbf{z}) - \frac{1}{k})), \\ l_{\mathbf{R}_i}(\mathbf{z}) &= s(\phi(\mathbf{z}) - \frac{1}{k}), \end{aligned} \tag{11}$$

where s is a sigmoid function to facilitate the separation. Then, the final likelihood for transmission edges can be defined as follows:

$$E_{\mathbf{T}}(\mathbf{z}) = \begin{cases} 1, & l_{\mathbf{T}_i/\mathbf{R}_i}(\mathbf{z}) > 0.6 \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

From the results shown in Figure 1(right), the likelihood of target transmitted scene effectively labels the transmission edges, which provides a clear guidance during rendering.

Based on the binarized edge map obtained in Equation (12), we propose to constrain rendering using a simple edge loss as follows:

$$\mathcal{L}_{edge} = \sum_{p(\mathbf{r}) \in \mathcal{P}} \| E_{\mathbf{T}}(p(\mathbf{r})) \odot (\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})) \|_2^2, \tag{13}$$

which means we constrain the rendered image more on wherever the transmission edges are present, encouraging a more faithful reconstruction towards the mixture image.

Besides, the transmission edges and those of the reflection tend to be different, as prior works pointed out [21]. We inherit the exclusion assumption between the gradient of transmission and reflection to differentiate the two parts in the gradient domain as follows:

$$\begin{aligned} \mathcal{L}_{\text{excl}}(\theta) &= \| \Psi(\mathbf{C_T}, \mathbf{C_R}) \|_F, \\ \Psi(\mathbf{C_T}, \mathbf{C_R}) &= \tanh(\lambda_{\mathbf{T}} |G(\mathbf{C_T})|) \odot \tanh(\lambda_{\mathbf{R}} |G(\mathbf{C_R})|), \end{aligned} \tag{14}$$

where $\lambda_{\mathbf{T}} = \sqrt{\frac{\|G(\mathbf{C_R})\|_F}{\|G(\mathbf{C_T})\|_F}}$ and $\lambda_{\mathbf{R}} = \sqrt{\frac{\|G(\mathbf{C_T})\|_F}{\|G(\mathbf{C_R})\|_F}}$. The downsampling operation in its original setting is discarded since our patch has been small enough.

By considering the above settings and constraints, our loss functions for the whole optimization can be concluded as follows:

$$\mathcal{L} = \gamma \mathcal{L}_{sc} + \delta \mathcal{L}_{edge} + \omega \mathcal{L}_{excl}, \tag{15}$$

where $\gamma = 2$, $\delta = 0.002$, and $\omega = 1$ are the weighting coefficients to balance the influence of each term. All the calculations and optimization processes are based on the patch sampled from the image.

4.3 Implementation details

We implement our framework using PyTorch. In the training and testing phase, two eight-layer MLPs with 256 channels are used to predict colors c and densities σ corresponding to the transmitted and reflection scenes. We train a "coarse" network along with a "fine" network network for importance sampling. We sample 64 points along each ray in the coarse model and 64 points in the fine model. A batch contains an image patch of 32×32 pixels, equivalent to 1024 rays. Similar to the settings in NeRF [1], positional encoding is applied to input location before they are passed into the MLPs. We use the Adam optimizer with defaults values $\beta_1 = 0.999$, $\beta_2 = 0.9$, $\epsilon = 10^{-8}$, and a learning rate 10^{-4} that decays following the cosine scheduler during the optimization. We optimize a single model for about 100K iterations on two NVIDIA V100 GPUs.

Our transmission encoder can extract transmission features of multiple resolutions by leveraging advantages from ERRNet [5] and U-Net [30]. Based on our experiments, even a simple U-Net [30] can also help to resist the reflection interference during rendering. The transmission encoder is first integrated with a feature decoder to compress the information to a lower dimension, and the output is fed into the transmission MLP.

For the pre-training of the transmission encoder, we follow the common way adopted by most image restoration methods, where the $\mathcal{L}_1$ and adversarial loss between the estimation and its ground truth are employed to optimize the solution space.

5 Experiments

5.1 Experimental Settings

Dataset. Our experiments are all based on a real-world dataset we collect. This dataset contains 8 different real-world scenes, each consisting of 20 to 30 mixture images with different poses. Specifically, four scenes are with the ground truth for quantitative evaluations in the experiments. We follow the setup proposed in [6] by first capturing the mixture image through transparent glass and then its corresponding transmitted scene by removing the glass.

Baselines. We compare our method with five NeRF-based methods: 1) NeRF [1]: the original NeRF method; 2) NeRF-W [7]: an unofficial implementation of NeRF in the Wild[1]; 3) MVSNeRF [10]: a NeRF-based multi-view stereo model for novel view synthesis which also relies on reference views; 4) Reflection removal (RR) + NeRF: the most direct way for novel transmitted view synthesis, namely training a NeRF network based on the results of a reflection removal method (without losing generality, we use ERRNet [5], a state-of-the-art method inspiring our design); 5) NeRFReN [31]: a NeRF-based method designed for see-through scenarios released recently[2]. All the results in Section 5 are obtained using six views for training. Due to the page size limitation, the results of other methods trained under more views can be found in the supplementary material.

Metrics. We report quantitative performance using PSNR, SSIM and LPIPS [32]. For PSNR and SSIM, higher value indicates better performance. For LPIPS [32], lower value indicates better performance. By measuring image fidelity using high-level features, LPIPS [32] can better match human judgements of image similarity.

5.2 Qualitative results

The qualitative results of the rendered novel transmitted views are presented in Figure 3 and Figure 4. Due to the effectiveness of the proposed transmission feature integration scheme, our method can reliably reconstruct novel transmitted views under only six views. Besides, the feature integration scheme and its cooperative recurring edge constraints further reduce the ambiguity between reflection and transmission during rendering. For other methods, though NeRF [1] can partially reconstruct the desired transmitted views, it still suffers from the interference caused by the reflection; NeRF-W [7] and NeRFReN [31] output degenerated results as they cannot separate reflection from transmission

[1] https://github.com/kwea123/nerf_pl

[2] We have only found the non-peer-reviewed version of this paper, so there is no publicly released code and we implement it by ourselves.

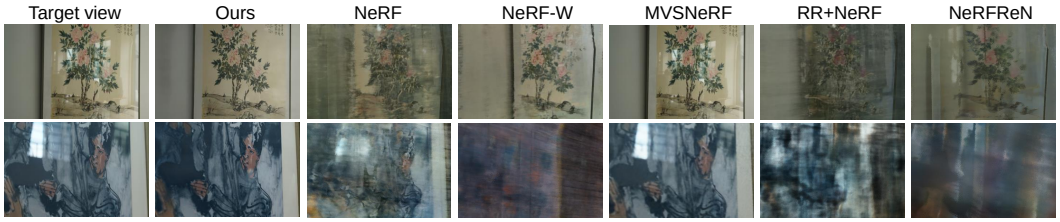

Figure 3: From left to right: the target view, our results, the results obtained by NeRF [1], NeRF-W [7], MVSNeRF [10], RR+NeRF, and NeRFReN [31]. The target views in the two scenarios (scenes with non-detachable glass, *e.g.*, painting with fixed frames) are captured through glass for reference only (no ground truth transmission). Please zoom in for more details.

Table 1: Quantitative evaluation compared with NeRF [1], NeRF-W [7], MVSNeRF [10], RR+NeRF, and NeRFReN [31]. Higher PSNR and SSIM values denote better results (↑), while lower LPIPS values denote better results (↓).

	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**22.75**	**0.841**	**0.205**
NeRF [1]	16.72	0.644	0.510
NeRF-W [7]	15.44	0.624	0.529
MVSNeRF [10]	17.83	0.685	0.400
RR+NeRF	18.42	0.691	0.539
NeRFReN [31]	16.69	0.672	0.545

with as few inputs as our method; MVSNeRF [10] can keep high fidelity of the rendering results like our methods, but obvious reflection residuals still remain in their results since their feature extractor cannot resist the reflection interference.

5.3 Quantitative results

The quantitative results of the rendered novel transmitted view in Table 1 also validate the observation in Figure 4. Higher PSNR values show that our method can render novel transmitted views and recover the color information with higher accuracy. Higher SSIM values indicate that our method can preserve the structural information with high-frequency details. Lower LPIPS values show that recovered images by our method better aligns with human perception. Specifically, for the results obtained by other methods, their higher LPIPS values indicate that they cannot reconstruct realistic results with high fidelity when only six views are provided. Specifically, MVSNeRF [10] achieves the second best LPIPS results among all methods, which shows that the feature integration scheme can indeed improve the rendering fidelity. On the other hand, RR+NeRF setting achieves the second best PSNR and SSIM result, which shows the effectiveness of this simple strategy to some degrees.

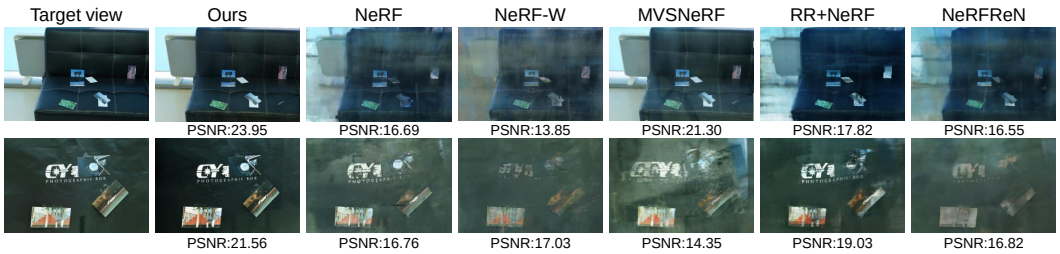

Figure 4: From left to right: the target view, our results, the results obtained by NeRF [1], NeRF-W [7], MVSNeRF [10], RR+NeRF, and NeRFReN [31]. The target views in the two scenarios (scenes with detachable glass, captured with the similar setup as [6]) are obtained by removing the glass, and are compared with the results as ground truth for PSNR values shown below each image. Please zoom in for more details.

Table 2: Ablation study on the model without the transmission encoder (η), the recurring edge constraints (REC), the patch-based rendering scheme ($\mathcal{P}$), and either the transmission encoder or patch-based rendering scheme. Higher PSNR and SSIM values denote better results (↑), while lower LPIPS values denote better results (↓).

	PSNR ↑	SSIM ↑	LPIPS ↓
Complete model	**22.75**	**0.841**	**0.205**
(w/o) η	15.60	0.512	0.565
(w/o) REC	22.48	0.836	0.265
(w/o) $\mathcal{P}$	18.17	0.769	0.271
(w/o) η & $\mathcal{P}$	15.65	0.588	0.594
NeRF	16.72	0.644	0.510

Figure 5: From left to right: the target view (with reflection), the results of our complete model, without the recurring edge constraints (REC), without the transmission encoder (η), without patch-based rendering scheme ($\mathcal{P}$), and with neither transmission encoder nor patch-based rendering scheme.

5.4 Ablation study

Our network consists of two parts: the transmission encoder to disentangle the transmitted and reflected scenes, and the patch-rendering scheme to utilize the physical constraints. We conduct several experiments to evaluate the benefits of these two parts. We first remove the transmission encoder and directly feed the MLP with the position and viewing direction. From the results shown in Figure 5, without the support of reference views, the position and viewing directions cannot effectively reconstruct the desired transmitted views. The errors shown in Table 2 also become closer to those of NeRF in Table 1. We then remove the recurring edge constraints, and more reflection is observed in the rendered target view since the network attaches equal importance to every pixel in the patch. We further remove the patch-based rendering scheme entirely. In this situation, reflection evidently remains on the generated target view due to a lack of physical constraint, as we discussed before. At last, both the patch-based rendering scheme and the transmission encoder are removed, and we achieve a degenerated result similar to that of NeRF-W [7] shown in Table 1.

6 Conclusions

We solve the problem of novel view synthesis for see-through scenarios in this paper. We introduce a transmission encoder to address the fidelity issue caused by sparse views and ambiguity issues led by the reflection interference in a unified framework. Specifically, to further disentangle the twisted relationship between transmission and reflection during the rendering process, we introduce a recurring edge constraint by counting the frequency of edge occurrence among the aligned image sequence. Experimental evaluation on a newly collected dataset demonstrates the promising performance for novel transmitted view synthesis our method could achieve.

Limitations. Our method still faces the challenge led by the occlusion issue, since we rely on features from neighboring views to complement the missing information given a sparse set of input views. However, when features are not consistent with a certain view, occlusions may negatively undermine the performance of our proposed method, due to the inaccurate feature information for synthesizing novel views in that area. We will deal with this issue in our future work.

References

[1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

[2] Teju Cole. The history of photography is a history of shattered glass. *New York Times*, 2017.

[3] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, Wen Gao, and Alex C Kot. Region-aware reflection removal with unified content and gradient priors. *IEEE Transactions on Image Processing*, 2018.

[4] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson WH Lau. Location-aware single image reflection removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[5] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[6] Renjie Wan, Boxin Shi, Haoliang Li, Yuchen Hong, Lingyu Duan, and Alex Kot Chichung. Benchmarking single-image reflection removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[7] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[8] Yu Li and Michael S Brown. Exploiting reflection change for automatic reflection removal. In *Proceedings of the IEEE international conference on computer vision*, 2013.

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

[10] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[12] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Styles Overbeck, Noah Snavely, and Richard Tucker. DeepView: High-quality view synthesis by learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[13] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D photography using context-aware layered depth inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[14] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

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

[18] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proc. International Conference on Computer Vision (ICCV)*, 2017.

[19] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C. Kot. CRRN: Concurrent multi-scale guided reflection removal network. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Renjie Wan, Boxin Shi, Haoliang Li, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. CoRRN: Cooperative reflection removal network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(12):2969–2982, 2019.

[21] 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*, 2018.

[22] Youwei Lyu, Zhaopeng Cui, Si Li, Marc Pollefeys, and Boxin Shi. Reflection separation using a pair of unpolarized and polarized images. *Advances in neural information processing systems*, 2019.

[23] 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*, 2017.

[24] Abhijith Punnappurath and Michael S Brown. Reflection removal using a dual-pixel sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[25] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[26] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[28] 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*, 2014.

[29] H.-C. Lee, E.J. Breneman, and C.P. Schulte. Modeling light reflection for computer color vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(4):402–409, 1990.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

[31] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. NeRFReN: Neural radiance fields with reflections. *arXiv preprint arXiv:2111.15234*, 2021.

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not think this work will have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 4.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] (during peer review). The code and data will be released once the paper is accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We think error bars are unnecessary in our work.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [No] We only use the dataset captured by ourselves for demonstration.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] (during peer review). The data will be released once the paper is accepted.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]