Flash-Splat: 3D Reflection Removal with Flash Cues and Gaussian Splats

Mingyang Xie[1]* Haoming Cai[1]* Sachin Shah[1] Yiran Xu[1]
Brandon Y. Feng[2] Jia-Bin Huang[1] Christopher A. Metzler[1]

[1]University of Maryland [2]Massachusetts Institute of Technology
https://flash-splat.github.io/

Abstract. We introduce a simple yet effective approach for separating transmitted and reflected light. Our key insight is that the powerful novel view synthesis capabilities provided by modern inverse rendering methods (e.g., 3D Gaussian splatting) allow one to perform flash/no-flash reflection separation using *unpaired measurements*—this relaxation dramatically simplifies image acquisition over conventional paired flash/no-flash reflection separation methods. Through extensive real-world experiments, we demonstrate our method, Flash-Splat, accurately reconstructs both transmitted and reflected scenes in 3D. Our method outperforms existing 3D reflection separation methods, which do not leverage illumination control, by a large margin.

1 Introduction

We are often surrounded by scenes with transparent surfaces, most notably glass, which introduce specular reflections. When viewing such scenes, we see a superimposition of transmitted and reflected light. This work focuses on the unsupervised separation of a transmitted 3D scene and a reflected 3D scene.

Reflection removal and separation have received considerable attention in the computational photography community. In addition to enhancing image quality and appeal, effective reflection separation methods can improve the robustness of downstream computer vision systems used in various applications, including robot navigation, classification, and 3D surface reconstruction. Separating transmitted and reflected 3D scenes is vital for various virtual reality tasks, such as 3D object extraction or editing.

Unfortunately, separating transmitted and reflected light from the sum of their intensities is a highly under-determined problem. To address this challenge, prior works have relied on various assumptions to perform single-image reflection removal. For instance, they have assumed the reflection is out-of-focus [2,56] or there is a noticeable double reflection caused by two sides of the glass [42]. However, these assumptions are not always true in real life. Other works have leveraged videos or multi-view images for reflection removal [1,11,12,15,16,33,54]. Their advantages over single-image methods are (1) they can get "lucky" where

[*] Equal Contribution.

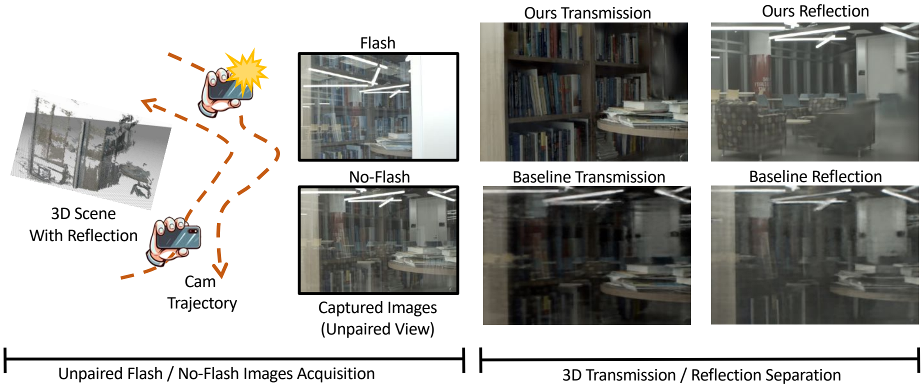


Fig. 1: Left: We separate the 3D transmitted and reflected scenes by capturing some views with camera flash and some views with no flash. **Right**: Our proposed Flash-Splat method achieves much better separation than the state-of-the-art unsupervised 3D separation method NeRFReN [13].

some views have weaker reflections than others, and (2) they can utilize multi-view consistency to regularize the separation. However, these methods still struggle to overcome the fundamental ill-posed problem, especially under strong reflection. For example, in Figure 1 the state-of-the-art unsupervised 3D reflection separation method, NeRFReN [13], fails to separate reflected and transmitted light from a collection of images captured under similar illumination conditions.

Introducing illumination control, i.e., flash/no-flash photography [24, 52, 53], can make the reflection separation problem significantly easier. Intuitively, the camera flash increases the intensities of the transmitted scene while leaving the reflected scene largely intact. Therefore, we can recover a reflection-free transmission scene by comparing images captured with and without flash. The core limitation is that it requires paired (tightly-aligned) flash/no-flash image captures—the camera cannot move between the captures. This paired measurement requirement represents a major barrier to effective in-the-wild reflection separation.

In this paper, we perform flash-based reflection separation without paired measurements by leveraging the powerful novel view synthesis capabilities of recently developed inverse differentiable rendering methods. Specifically, during acquisition, a user captures roughly half of the views with flash on and the other half with flash off. Then, by extending the powerful Gaussian Splatting [20] technique, we can construct 2D "pseudo-paired" flash/no-flash images, where one image in the pseudo flash/no-flash pair is captured, and the other one is synthesized with our 3D inverse rendering framework; we can also construct a 3D "pseudo-pair" of flash/no-flash 3D representations, where one 3D representation is reconstructed from only the flash images, and the other is reconstructed from only the no-flash images. The difference between the 2D pseudo-pair and the difference between the 3D pseudo-pair both serve as strong priors for the transmitted 3D scene, which significantly reduce the ill-posedness of the separation problem. As a byproduct of our 3D inverse differentiable rendering framework, our method, Flash-Splat, is also capable of performing novel view synthesis and depth estimation for each transmitted and reflected scene. We validate our proposed approach in real-world experiments and demonstrate its state-of-the-art performance.

Our contributions are:

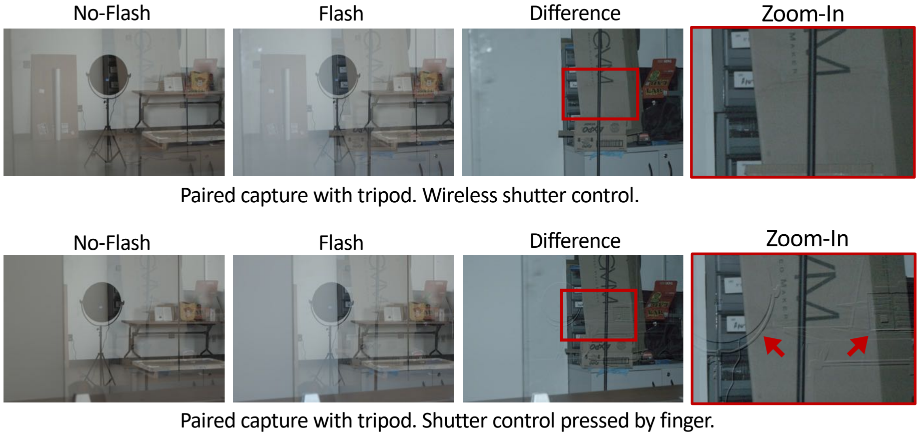
- We propose a robust strategy, Flash-Splat, for 3D transmission-reflection separation and scene reconstruction, using flash illumination as a physical cue without requiring paired flash/no-flash captures.
- We introduce novel modifications to make 3D Gaussian Splatting illumination-aware, enhancing the quality of each separated 3D scene.
- We show that Flash-Splat excels in separating reflection and transmission, even when baseline methods fail, over real-world scenes.
- We demonstrate that Flash-Splat can perform high-quality novel view synthesis and depth estimation for both the transmitted and reflected 3D scenes.

2 Related work

Reflection removal. Existing reflection removal methods can generally be divided into three categories: single-frame, multi-frame and polarization-based. Single-frame approaches [2,7,8,14,17–19,21,26–29,31,32,42,44–47,50,51,55–58, 60] only take a single image and remove the reflection. Multi-frame approaches [1, 6,9,11,12,15,16,31,33,54] use multiple input frames as cue and produce multi-view consistent results. Polarization-based approaches [22,23,25,30,34,37] leverage the fact that the transmission is unpolarized while the reflection component varies when rotating the polarization filter. However, none of those methods aim to recover a 3D representation of the transmitted or the reflected scene.

3D neural scene representations. To get more accurate 3D reconstruction for decomposition, we consider differentiable 3D neural representations. Neural Radiance Fields (NeRFs) [4,5,10,36] has received vast attentions in the past few years, for their accurate and consistent novel view synthesis results. Another line of works focuses on accurate 3D geometry, so they considers Signed Distance Function (SDF) [48,49] for better surface accuracy. Recently, 3D Gaussian Splatting (3DGS) [20,59] emerges for its fast training and inference speed.

Reflection removal by inverse rendering. Previous methods consider solving reflection removal using inverse 3D rendering. ReflectionsIBR [43] as a pioneer proposes to separate each frame into a transmission and reflection layer combined with a binary reflection mask, and tries to reconstruct the scene using an image-based rendering. Recently, NeRFReN [13] uses a NeRF to achieve better 3D reconstruction accuracy. NeuS-HSR [38], instead of focusing on reflection separation, uses Signed Distance Functions to achieve better surface reconstruction quality. Distinct from these 3D methods, our proposed method dramatically extends these approaches by incorporating variable illumination.



Paired capture with tripod. Wireless shutter control.

Paired capture with tripod. Shutter control pressed by finger.

Fig. 2: Flash/No-Flash For Reflection Removal. The difference between paired flash and no-flash images is equivalent to taking a photo with flash in a dark environment, which gives us a reflection-free image (top). This is because flash increases the transmission brightness, but not the reflection brightness. Notice pairs must be tightly aligned for this method to work. Even tiny vibrations such as pressing the shutter button even when using a tripod produce artifacts (bottom).

3 Method

3.1 Paired 2D Flash/No-Flash

Capturing a pair of flash and no-flash images of a scene from the same camera viewpoint allows one to reconstruct a reflection-free image.

Reflections exist because ambient light illuminates objects in the reflected scene and reflects off the glass onto the camera sensor. The captured composite scene with no flash $\mathbf{I}_N$ can be modeled as

$$\mathbf{I}_N = \mathbf{T}_N + \beta \circ \mathbf{R} \tag{1}$$

where $\mathbf{T}_N$ is the transmission scene with no flash, $\mathbf{R}$ is the reflection scene and β is the reflective fraction factor (in the extreme case where the reflection is caused by a mirror, then β could be interpreted as the mask of the mirror in the scene). Now consider the case where the scene is captured with a flash co-located on the camera. If we assume that the scene behind glass is diffuse and that the camera flash is uniform, the camera flash will increase the intensity of all pixels proportionally. Therefore, we can formulate the flash image $\mathbf{I}_F$ as following:

$$\mathbf{I}_F = (1+\alpha)\mathbf{T}_N + (\beta + \beta_F) \circ \mathbf{R}, \tag{2}$$

where α and β_F represent the intensity increase of the transmitted and reflected scene due to flash, respectively. Assuming the direct reflection of the flash is outside the camera's field of view (i.e., the specular surface is not orthogonal to the camera view), the flash will have little effect on the brightness of the reflected scene. In common cases like glass, the impact of secondary reflections is also usually very low. Therefore, we may approximate β_F as close to zero,

$$\mathbf{I}_F \approx (1+\alpha)\mathbf{T}_N + \beta \circ \mathbf{R}. \tag{3}$$

As such, one technique used among photographers is subtracting a no-flash image $\mathbf{I}_N$ [24] from a flash image $\mathbf{I}_F$,

$$\mathbf{I}_F - \mathbf{I}_N \approx \alpha \mathbf{T}_N. \tag{4}$$

The difference is effectively a reflection-free transmitted scene scaled by some constant. Figure 2 demonstrates the impressive performance of this simple method.

Unfortunately, this process only works when we capture **paired** flash and no-flash images at the *same* location and orientation. Any small movement between the image pair causes the approach to break down. As illustrated in the bottom row of Figure 2, even with a tripod, the slight motion caused by touching the exposure button (as opposed to using remote triggering) can introduce significant errors in the conventional flash/no-flash reflection separation process.

3.2 Unpaired 3D Flash/No-Flash

In this work, we extend the flash/no-flash idea to 3D and thus remove the requirement of capturing paired images, which makes flash-based reflection removal significantly easier and more practical. Instead of directly capturing paired multi-view images of a scene, we propose to first capture an arbitrary sequence of multi-view flash images of the scene, and then capture another sequence of multi-view no-flash images of the scene. These two sequences should be captured such that they approximately cover a similar range of perspectives.

Our 3D Flash/No-flash formulation is defined as follows. Following previous notations, we consider four 3D representations in total: transmission with flash $\mathbf{T}_F$, transmission without flash $\mathbf{T}_N$, reflection $\mathbf{R}$, and the reflective fraction factor β. To render a target pixel in a captured image, we blend the overlapping regions of the transmitted and reflected scenes. We then have,

$$\begin{aligned} \mathbf{I}_N &= \mathbf{T}_N + \beta \circ \mathbf{R} \\ \mathbf{I}_F &= \mathbf{T}_F + \beta \circ \mathbf{R} \end{aligned} \tag{5}$$

for flash (F) images and no-flash (N) images. Even though we are only capturing unpaired flash/no-flash views now, we can still associate them by creating 2 types of "pseudo-pairs" to aid reflection separation.

Firstly, we can construct **2D "pseudo-pairs"** via novel view synthesis of the missing flash/no-flash counterpart, as shown in Figure 3a. Consider a specific view where only the flash image is taken. Utilizing inverse rendering techniques, we are able to synthesize a no-flash image at this exact same view by using the no-flash images taken at neighboring views. This synthesized no-flash image and the captured flash image form a 2D pseudo-pair. The difference image between

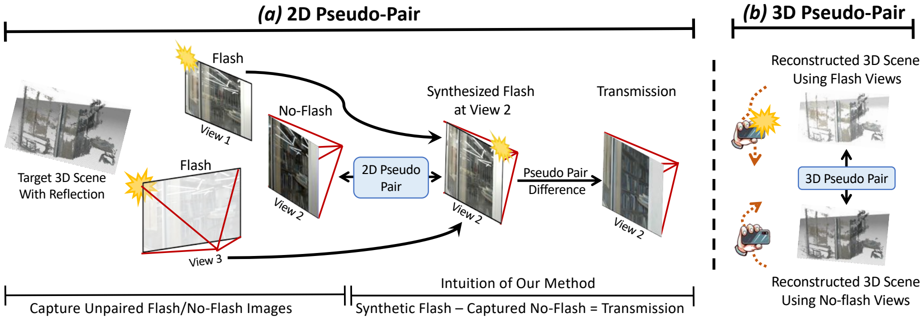

(a) 2D Pseudo-Pair
(b) 3D Pseudo-Pair
Target 3D Scene With Reflection
Flash
View 1
No-Flash
View 2
Flash
View 3
2D Pseudo Pair
Synthesized Flash at View 2
View 2
Pseudo Pair Difference
Transmission
View 2
Capture Unpaired Flash/No-Flash Images
Intuition of Our Method
Synthetic Flash – Captured No-Flash = Transmission
Reconstructed 3D Scene Using Flash Views
3D Pseudo Pair
Reconstructed 3D Scene Using No-flash Views

Fig. 3: Our Intuition: Construct 2D and 3D "pseudo-pairs" as Cues for Reflection Removal. Flash-Splat does not require *paired* flash/no-flash data. During the data capture stage, we collect *unpaired* flash/no-flash images from different views. In (a), if we captured a no-flash image at View 2, we can learn a 3D representation of the captured flash images at other views, and then synthesize a novel view of the flash image at View 2. As such, we have created a **2D pseudo-pair** of flash and no-flash images at View 2. If we then take the difference between the pseudo-pair as in Figure 2, we get the transmission component of View 2 that is free of reflection. In (b), we reconstruct a 3D scene with flash using only the flash images (top); we also reconstruct a 3D scene without flash using only the no-flash images (bottom). As such, we have created a **3D pseudo-pair** of flash/no-flash scenes.

this 2D pseudo-pair should be reflection-free, just like the difference between the 2D paired flash/no-flash images, as indicated in Equation (4).

Secondly, we can construct a **3D "pseudo-pair"** by elevating the problem to the 3D space, as shown in Figure 3b. More specifically, we can reconstruct a 3D scene with flash and another without flash, using only the views captured with flash and only the views captured without flash, respectively. We name these two reconstructed scenes as $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$, to differentiate them with the ground truth 3D flash/no-flash scenes $\mathbf{I}_F$ and $\mathbf{I}_N$. $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$ form a 3D pseudo-pair, as they are the same scene except that the transmitted part of $\mathbf{I}_F^{Rec}$ is brighter due to the flash. A 3D pseudo-pair difference can be used as a cue for the transmitted scene. Nevertheless, as $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$ are separately reconstructed from 2 unpaired sets of data, they will be misaligned, thus the word "pseudo".

As such, we obtain important "flash cues" from the 2D and 3D pseudo-pairs, and use them as the high-level intuitions for our proposed method.

3.3 Proposed Pipeline for 3D Reflection Separation

In this subsection, we first explain how to incorporate flash cues to guide our reconstruction, then describe our overall optimization framework, and finally discuss how to adapt the loss functions to accommodate the RAW input images.

Regularizing Reflection Separation using Flash Cues. While our high-level intuition is to construct pseudo-pairs as cues for reflection-free images, in this work, we want to reconstruct both the transmitted scene and the reflected scene. Therefore, we do not explicitly calculate the difference between the pseudo-pairs, but rather, use it as a regularization term to guide separation optimization. As shown in Equation (3), $\mathbf{T}_F$ is expressed as $\mathbf{T}_N$ multiplied by a scalar $(1+\alpha)$, which enforces a linear relationship between them. Therefore, we choose to enforce the linearity between $\mathbf{T}_F$ and $\mathbf{T}_N$, which is equivalent to enforcing the constraint that the flash/no-flash difference is reflection-free.

While in the ideal case, $\mathbf{T}_F$ and $\mathbf{T}_N$ should form a strictly linear relationship, in reality, the camera flash might not be perfectly uniform; there is also a chance the secondary reflection of the flash does hit the camera sensor. As a result, this relationship between $\mathbf{T}_F$ and $\mathbf{T}_N$ should be close to linear, but might not perfectly linear. Therefore, we chose not to use this hard constraint, but use the Pearson Coefficient, which measures the linearity between $\mathbf{T}_F$ and $\mathbf{T}_N$:

$$\mathcal{L}_{linearity} = -\frac{\operatorname{cov}\left(\mathbf{T}_N, \mathbf{T}_F\right)}{\sqrt{\operatorname{var}\left(\mathbf{T}_N\right)\operatorname{var}\left(\mathbf{T}_F\right)}} \tag{6}$$

By minimizing this loss term, we encourage the 3D Gaussians to learn a reflection-free transmission, therefore reducing the ill-posedness of the separation.

Notably, while the analysis above holds true for both the 3D pseudo-pair (see Figure 3b) and the 2D pseudo-pairs (see Figure 3a), we only apply this linearity regularization to the 2D pseudo-pairs, as it is more straightforward to measure the linearity of images than 3D representations.

Initializing 3D Representations Using Flash Cues. Now we show how to utilize the 3D pseudo-pair to aid reflection separation. As illustrated in Figure 2b, the 3D pseudo-pair, namely $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$, are 3D representations of the target scene reconstructed from the flash views and no-flash views, respectively. Their difference should be the reflection-free 3D transmitted scene. However, given the highly ill-posed nature of the 3D scene reconstruction problem, it is very likely that the contents in $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$ do not correspond with each other. As such, this difference between $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$ should be viewed as a very rough estimate of the transmitted scene. Therefore, we decide to only use it to initialize the 3D representations $\mathbf{T}_F$, $\mathbf{T}_N$, $\mathbf{R}$, and β for better convergence.

We use 3DGS [20] as the 3D representation architecture, which is normally initialized from sparse point clouds. We first use structure from motion, e.g., [41], to obtain the sparse point clouds of the 3D pseudo-pair $\mathbf{I}_F^{Rec}$ and $\mathbf{I}_N^{Rec}$. Then we roughly align them via linear transformation to compensate for the difference in the camera coordinate systems. Afterwards, we compare these two sets of point clouds: for points in regions with increased intensities, we classify them as "transmitted points"; for points in regions with unchanged intensities, we classify them as "reflected points". Finally, we initialize the 3DGSs for $\mathbf{T}_F$, $\mathbf{T}_N$, and β from the "transmitted points", and the 3DGS for $\mathbf{R}$ from the "reflected points".

Note that this way of initialization relies on the 3D representation's compatibility with point clouds. It does not work with implicit neural 3D representations like NeRF [36]. When using NeRF as our 3D representations, we just randomly initialize the neural network and only rely on the previously discussed linearity

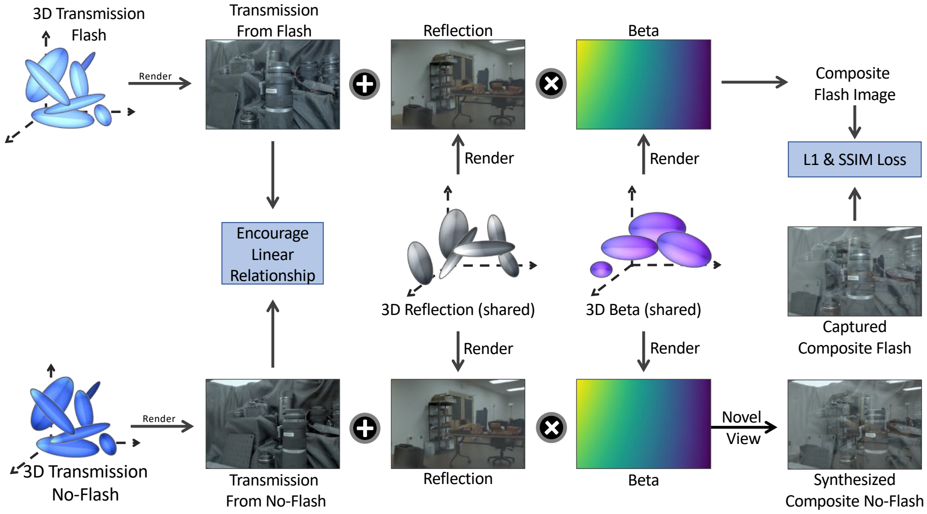


Fig. 4: Method Overview. We use 4 3DGSs [20] as our 3D representations for the transmitted scene with flash $\mathbf{T}_F$, the transmitted scene with no flash $\mathbf{T}_N$, the reflected scene $\mathbf{R}$ and the reflective fraction map β. Based on the Flash/No-flash technique, $\mathbf{R}$ and β are shared between the flash image and the no-flash image, while the relationship of $\mathbf{T}_F$ and $\mathbf{T}_N$ is close to linear. We initialize these 4 3DGSs using cues from the 3D pseudo-pair (see Figure 3b and Section 3.3). In each iteration of optimization, our method operates on a single view. This figure, for instance, shows a view where we captured a flash image. There is NO no-flash image captured at this view. As shown in the top row, we use $\mathbf{T}_F$, $\mathbf{R}$, and β to render a flash image of this particular view and calculate losses with the captured ground truth flash image. Additionally, based on the cues from 2D pseudo-pairs, we calculate the Pearson linearity loss between $\mathbf{T}_F$ and $\mathbf{T}_N$ to encourage the linearity between them (see Figure 3a and Section 3.3). We then back-propagate the gradients and update the weights of the 4 3DGSs.

regularization using 2D pseudo-pairs, which would still achieve better reflection removal performance than baselines, as will be shown in Section 6.

Overall Optimization Framework. As shown in Figure 4, in each iteration of optimization, Flash-Splat operates on a single view. If we captured a flash image at this view (meaning that NO no-flash image was captured at this view), we follow Equation (5) and use our 3D representations $\mathbf{T}_F$, $\mathbf{R}$, and β to render a flash image at this same view. Then we calculate the loss between the rendered flash image and the captured ground truth flash image (more on this in the next paragraph). Additionally, we also calculate the Pearson linearity loss between images rendered from $\mathbf{T}_F$ and $\mathbf{T}_N$ at this view (the 2D pseudo-pair). We then back-propagate the gradients and update the weights of the 4 3D representations $\mathbf{T}_F$, $\mathbf{T}_N$, $\mathbf{R}$, and β. In the next iteration, we perform similar computations with flash and no-flash swapped. By doing such alternative optimization, we are using the loss with the captured ground truth images to supervise the novel view synthesis ability of our 3D representations, while using the Pearson linearity loss to implicitly enforce the flash/no-flash prior.

Gamma Corrected Loss Function. Our complete loss function is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{DSSIM} + \lambda_3 \mathcal{L}_{linearity} + \lambda_4 \mathcal{L}_{depth} \,, \tag{7}$$

where $\mathcal{L}_1$ and DSSIM [3] are computed between the captured RGB images and the rendered images; $\mathcal{L}_{depth}$ is a depth smoothness regularization adopted from NeRFReN [13]; λ_{1-4} are weightings for these 4 loss terms, respectively.

It is imperative to understand that for the difference of the flash/no-flash discrepancy to work, Equations 1 through 6 must be operational within the RAW image space. Nevertheless, our objective is for the $\mathcal{L}_1$ loss and DSSIM loss to be applied to tone-mapped images, as [35] demonstrates that learning within the RAW space predisposes the model to bias in favor of brighter pixels while neglecting the darker ones. Furthermore, it is our intention for the 3DGS model to output tone-mapped images, a decision driven by empirical observations indicating a potential underperformance when learning is conducted in the RAW domain. Consequently, we modify our loss function to incorporate $\mathcal{L}_1$ and DSSIM calculations as follows:

$$\mathcal{L} = \mathcal{L}\left(\gamma\left(\mathbf{I}^{raw}\right), \gamma\left(\gamma^{-1}\left(\mathbf{T}\right) + \beta\gamma^{-1}\left(\mathbf{R}\right)\right)\right) \tag{8}$$

where $\mathcal{L}$ can be either $\mathcal{L}_1$ or DSSIM loss, and we chose $\gamma(\mathbf{x}) = \mathbf{x}^{0.22}$ as gamma correction to tone-map RAW images.

4 Experimental Details

4.1 Dataset Collection

We captured a flash/no-flash dataset using a Canon Rebel R7 camera. Our camera settings included a fixed 0.25-second exposure time, 200 ISO sensitivity, an f-number of 5.6, and fixed white balancing. For each scene, we collected 30 images equally split into two categories: 15 with the built-in flash and 15 without. The typical distance between a flash view and the closest no-flash view is $5-10$ centimeters, with $1-6$ meters separating the camera from the transmission scene objects. For a fair comparison with the baselines, we captured paired flash/no-flash data for a few scenes using a tripod, such that our method and the baselines use exactly the same views, with the only data difference being our method uses half flash and half no-flash views. Note that our method does not see any paired flash/no-flash views. We process the raw images through a standard image signal processor consisting of white balancing, tone-mapping and gamma correction. We run COLMAP [41] on each method's corresponding input images to obtain camera poses and sparse point cloud estimations.

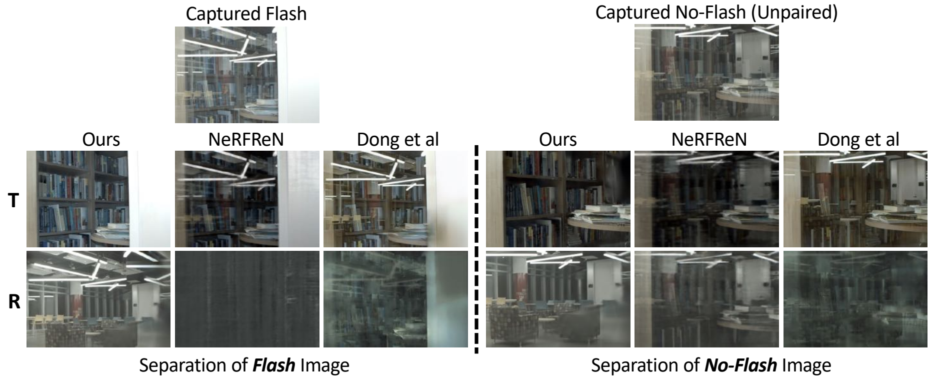

Captured Flash
Captured No-Flash (Unpaired)
Ours
NeRFReN
Dong et al
Ours
NeRFReN
Dong et al
T
R
Separation of *Flash* Image
Separation of *No-Flash* Image

Fig. 5: The Office scene. Top, middle, and bottom rows are the captured images, separated transmissions, and separated reflections, respectively. Our reflection separation approach is far more effective than NeRFReN [13] and Dong et al [7].

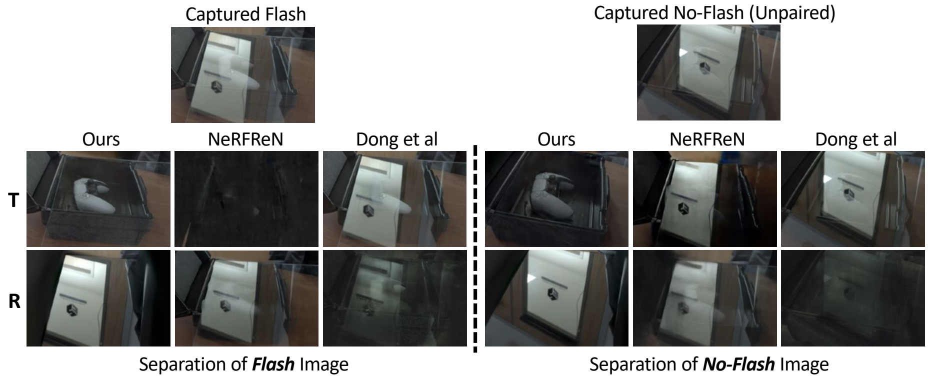

Captured Flash
Captured No-Flash (Unpaired)
Ours
NeRFReN
Dong et al
Ours
NeRFReN
Dong et al
T
R
Separation of *Flash* Image
Separation of *No-Flash* Image

Fig. 6: The Game Controller Scene. Top, middle, and bottom rows are the captured images, separated transmissions, and separated reflections, respectively. Our reflection separation approach is far more effective.

4.2 Baselines

We choose 2 baselines: NeRFReN [13], an unsupervised multi-view method based on 3D inverse rendering, and Dong et al. [7], a supervised deep learning approach. For fair comparison, we run both baselines twice, once on all flash images, and once on all no-flash images. This is to show that our method does not perform better because we use flash, but rather, because we use flash/no-flash cues.

4.3 Architecture and Optimization details.

Flash-Splat was implemented in PyTorch and run on an NVIDIA A6000 GPU. Our 4 3DGSs ($\mathbf{T}_F$, $\mathbf{T}_N$, $\mathbf{R}$, and β) have no shared parameters and are optimized

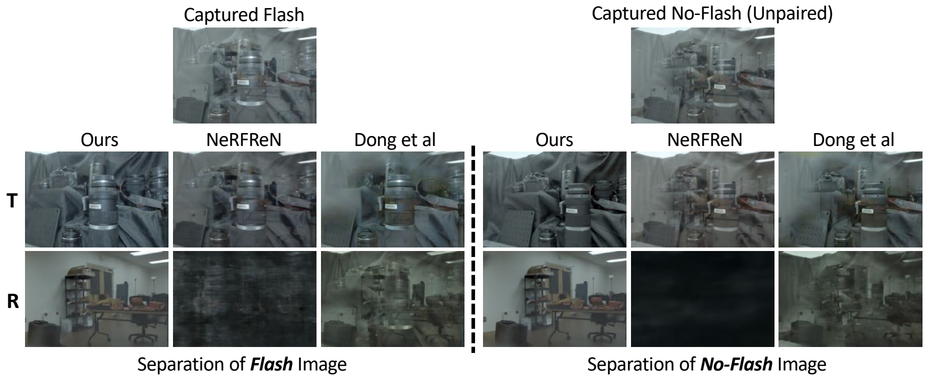

Captured Flash
Captured No-Flash (Unpaired)
Ours
NeRFReN
Dong et al
Ours
NeRFReN
Dong et al
T
R
Separation of *Flash* Image
Separation of *No-Flash* Image

Fig. 7: The Lens Stage Scene. Top, middle, and bottom rows are the captured images, separated transmissions, and separated reflections, respectively. Our reflection separation approach is far more effective.

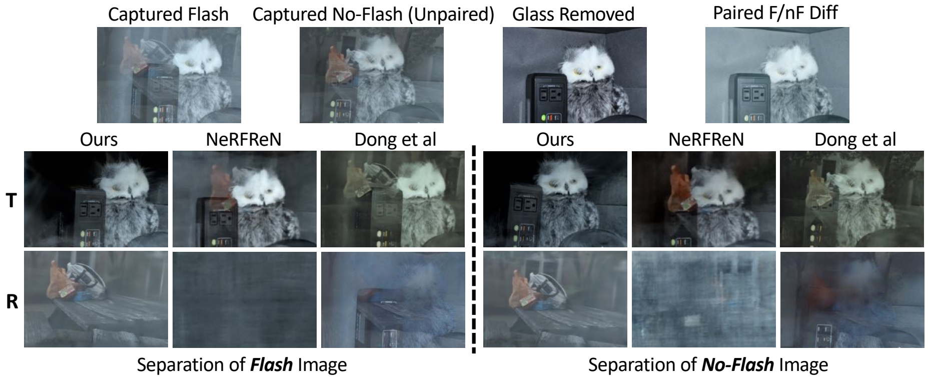


Fig. 8: The Outdoor Scene. The captured images, separated transmissions, and separated reflections are shown in the top, middle, and bottom rows, respectively. We also include two additional reference images in the top row as references. "Paired F/nF Diff" denotes the difference between flash and no-flash paired images. "Glass Removed" shows true transmission by directly removing the glass.

with the same hyperparameters. Our implementation of 3DGS follows FSGS [59], a variant of 3DGS that supports feedforward settings (the original 3DGS [20] is not intended for feedforward scenes). Each 3DGS is initialized with 350K Gaussians and grow up to 500K. We optimize all 3DGSs for 5000 iterations on our flash/no-flash images sized 1200×800 pixels. Total running time is about 10 minutes per scene. As an alternative to 3DGS, we also explored using NeRF for 3D representations, but empirically found that it inferior to 3DGS; see Sec. 6.2.

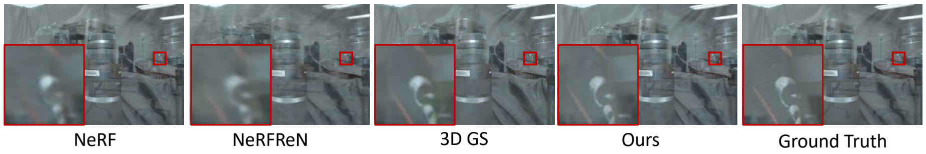


Fig. 9: Novel View Synthesis (NVS). Our method does not compromise NVS quality, even when compared to dedicated NVS methods like NeRF and 3DGS.

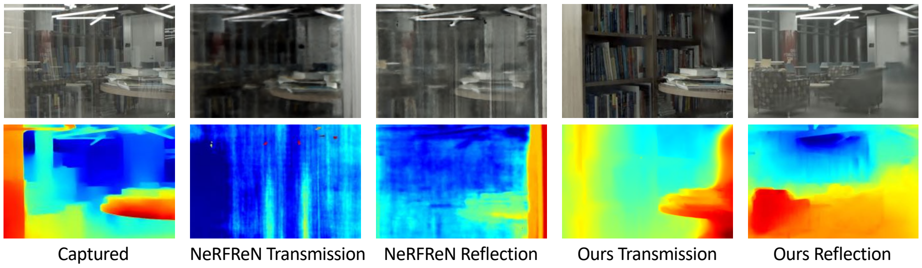


Fig. 10: Depth Estimation. The captured image's depth estimated by MiDaS [39, 40] is shown in the leftmost column, which cannot differentiate between transmitted and reflected scenes. Our depths are much better than NeRFReN's [13].

5 Results

We compare our method's transmission-reflection separation ability against the baseline state-of-the-art methods. We also demonstrate novel view synthesis and depth estimation capabilities through our 3D inverse rendering framework.

Transmission-Reflection Separation. Our method, Flash-Splat, outperforms both baselines in transmission-reflection separation on our real-world scenes. In fact, the baselines fail completely to produce a reasonable separation. Figures 5, 6, 7 show results on indoor scenes. 8 show results on an outdoor scene. Scene descriptions and results on 3 more scenes are included in the supplement.

Novel View Synthesis (NVS). Flash-Splat can perform novel view synthesis as it is based on inverse rendering. We compare our synthesis quality for scenes with reflections against NeRF [36], NeRFReN [13], and 3DGS (our implementation of 3DGS still follows FSGS [59], as explained in Section 4.3). Figure 9 shows Flash-Splat does not compromise NVS performance, even when compared to dedicated NVS methods like NeRF and 3DGS. Rendered videos of our separated transmitted and reflected 3D scenes are in our project webpage.

Depth Estimation. Benefitting from the 3DGS base representation, Flash-Splat can also perform depth estimation on both transmitted and reflected scenes. Flash-Splat outperforms NeRFReN for both components (see Figure 10).

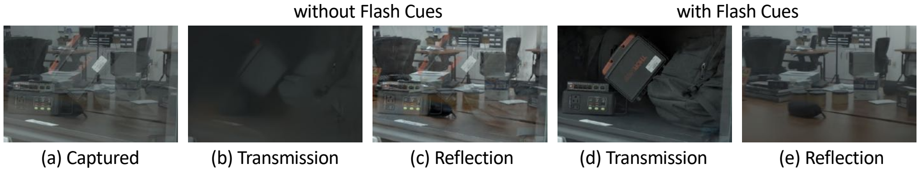


Fig. 11: With and Without the Flash Cues. (b,c) shows the reflection separation result if we do not utilize the flash cues in our proposed framework, details of which are discussed in Section 6.1. (d,e) shows the result using the full version our proposed framework — the separated reflection and transmission are significantly better. This highlights the importance of our proposed flash cues.

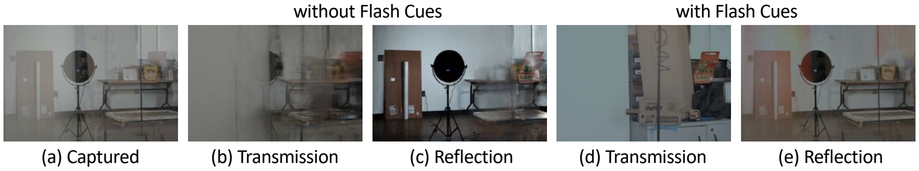


Fig. 12: Equip NeRFReN [13] With Flash Cues from 2D Pseudo-Pairs. (b,c) show separation results from the original NeRFReN; (d,e) show separation results from NeRFReN incorporated our proposed flash cues. Since NeRFReN is not compatible with the point clouds initialization using the 3D pseudo pair, we only equip NeRFReN with the Pearson linearity regularization from the 2D pseudo pairs. Our 2D pseudo pairs regularization alone can achieve better reflection separation performance than the original NeRFReN.

6 Ablation Studies

6.1 With and Without Flash Cues

To demonstrate the importance of flash cues, we design a “flashless” variant of our proposed framework where we remove the flash cues from the 2D and 3D pseudo-pairs. This “flashless” framework is still a 3DGS-based approach, but does not utilize flash/no-flash photography at all. Its detailed architecture is illustrated in Figure 19 in the supplement. For a fair comparison, this flashless framework is using the same number of total views as our proposed framework. Figure 11 shows that this flashless framework achieves significantly worse separation performance than our proposed Flash-Splat framework, which highlights the indispensability of the flash cues.

6.2 Replacing 3DGS with NeRF

To investigate the flash cues’ impact on other 3D representations, we design another framework named Flash-NeRF, where we replace the 3DGS [20] representation with NeRF [36] and keep everything else the same. Since NeRF is not compatible with our pseudo pair point clouds initialization, we only utilize the Pearson linearity regularization from the 2D pseudo pairs as the flash cue. The Flash-NeRF framework can be seen as NeRFReN [13] plus our 2D pseudo pairs regularization. Figure 12 shows that our 2D pseudo pairs regularization significantly enhances NeRFReN's reflection separation performance.

Nevertheless, while this Flash-NeRF framework obtains an almost perfect transmitted scene, we can still notice obvious artifacts and floaters in the reflected scene. We find that using 3DGS as 3D representation can notably mitigate this issue, and thus decide to use 3DGS in our proposed framework.

6.3 Discussions & Limitations

Reflections in COLMAP: By Law of Reflection, reflected objects are equivalently "virtual objects" superimposed into the transmitted scene. We empirically verified reflections do not decrease COLMAP's view matching performance.

View Coverage: Although Flash-Splat does not need paired flash/no-flash images, it does require the flash and no-flash scenes to cover a similar range of perspectives. For instance, Flash-Splat would not work if we take flash images of the scene from the left and no-flash images from the right, as we would not be able to accurately synthesize pseudo-pairs in this case.

Curved Reflection: Flash-Splat currently cannot deal with scenes with curved reflective surfaces, e.g., curved glasses, since curved reflective surfaces will cause severe deformation of the reflected scene. We believe this is an important yet under-explored corner case for reflection removal, which we leave for future work.

Double Reflection: While double reflections can be a powerful cue for removing reflections when dealing with thick or double-pane glass [42], Flash-Splat currently does not handle scenes with obvious double reflections. Incorporating double-reflection–based cues is an interesting research direction.

Flash Strength: If the flash is too weak to illuminate the transmission scene, Flash-Splat would not have the necessary flash cues to remove the reflection.

Dynamic Scene: Because our 3D representation is static, objects moving between captured images result in blurry reconstructions.

7 Conclusion

We present a novel approach for transmission-reflection separation of 3D scenes through flash cues, significantly mitigating the ill-posedness of the task. By synthesizing "pseudo-paired" flash/no-flash images within a 3D inverse rendering framework based on Gaussian Splatting, we demonstrate superior reflection separation capabilities, particularly under challenging conditions where traditional methods falter. We validate our method on a new real-world dataset, showcasing its effectiveness and robustness. Our method not only unlocks practical reflection-removal but also enables novel view synthesis and depth estimation separately for the transmitted and reflected 3D scene.

Acknowledgements

This work was supported in part by AFOSR Young Investigator Program Award no. FA9550-22-1-0208, ONR Award no. N00014-23-1-2752, NSF CAREER Award no. 2339616, the Joint Directed Energy Transition Office, and a gift from Dolby Labs. We thank Kevin Zhang and Yi-Ting Chen for helpful discussions.

References

1. Alayrac, J.B., Carreira, J., Zisserman, A.: The visual centrifuge: Model-free layered video representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2457–2466 (2019) 1, 3
2. Arvanitopoulos, N., Achanta, R., Susstrunk, S.: Single image reflection suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4498–4506 (2017) 1, 3
3. Baker, A.H., Pinard, A., Hammerling, D.M.: On a structural similarity index approach for floating-point data. IEEE Transactions on Visualization and Computer Graphics pp. 1–13 (2023) 9
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022) 3
5. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022) 3
6. Chugunov, I., Shustin, D., Yan, R., Lei, C., Heide, F.: Neural spline fields for burst image fusion and layer separation. CVPR (2024) 3
7. Dong, Z., Xu, K., Yang, Y., Bao, H., Xu, W., Lau, R.W.: Location-aware single image reflection removal. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5017–5026 (2021) 3, 10
8. Fan, Q., Yang, J., Hua, G., Chen, B., Wipf, D.: A generic deep architecture for single image reflection removal and image smoothing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3238–3247 (2017) 3
9. Farid, H., Adelson, E.H.: Separating reflections and lighting using independent components analysis. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). vol. 1, pp. 262–267. IEEE (1999) 3
10. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023) 3
11. Gandelsman, Y., Shocher, A., Irani, M.: " double-dip": unsupervised image decomposition via coupled deep-image-priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11026–11035 (2019) 1, 3
12. Guo, X., Cao, X., Ma, Y.: Robust separation of reflection from multiple images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2187–2194 (2014) 1, 3
13. Guo, Y.C., Kang, D., Bao, L., He, Y., Zhang, S.H.: Nerfren: Neural radiance fields with reflections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18409–18418 (2022) 2, 3, 9, 10, 12, 13, 14

14. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015) 3
15. Hong, Y., Zheng, Q., Zhao, L., Jiang, X., Kot, A.C., Shi, B.: Panoramic image reflection removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7762–7771 (2021) 1, 3
16. Hong, Y., Zheng, Q., Zhao, L., Jiang, X., Kot, A.C., Shi, B.: Par2 net: End-to-end panoramic image reflection removal. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 1, 3
17. Hu, Q., Guo, X.: Trash or treasure? an interactive dual-stream strategy for single image reflection separation. Advances in Neural Information Processing Systems **34**, 24683–24694 (2021) 3
18. Hu, Q., Guo, X.: Single image reflection separation via component synergy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13138–13147 (2023) 3
19. Kee, E., Pikielny, A., Blackburn-Matzen, K., Levoy, M.: Removing reflections from raw photos. ArXiv **abs/2404.14414** (2024) 3
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) 2, 3, 7, 8, 11, 13
21. Kim, S., Huo, Y., Yoon, S.E.: Single image reflection removal with physically-based rendering. arXiv preprint arXiv:1904.11934 (2019) 3
22. Kong, N., Tai, Y.W., Shin, J.S.: A physically-based approach to reflection separation: from physical modeling to constrained optimization. IEEE transactions on pattern analysis and machine intelligence **36**(2), 209–221 (2013) 3
23. Kong, N., Tai, Y.W., Shin, S.Y.: High-quality reflection separation using polarized images. IEEE Transactions on Image Processing **20**(12), 3393–3405 (2011) 3
24. Lei, C., Chen, Q.: Robust reflection removal with reflection-free flash-only cues. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14811–14820 (2021) 2, 5
25. Lei, C., Huang, X., Zhang, M., Yan, Q., Sun, W., Chen, Q.: Polarized reflection removal with perfect alignment in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1750–1758 (2020) 3
26. Levin, A., Weiss, Y.: 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) 3
27. Levin, A., Zomet, A., Weiss, Y.: Learning to perceive transparency from the statistics of natural scenes. Advances in Neural Information Processing Systems **15** (2002) 3
28. Levin, A., Zomet, A., Weiss, Y.: Separating reflections from a single image using local features. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. IEEE (2004) 3
29. Li, C., Yang, Y., He, K., Lin, S., Hopcroft, J.E.: Single image reflection removal through cascaded refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3565–3574 (2020) 3
30. Li, R., Qiu, S., Zang, G., Heidrich, W.: Reflection separation via multi-bounce polarization state tracing. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 781–796. Springer (2020) 3

31. Li, Y., Brown, M.S.: Exploiting reflection change for automatic reflection removal. In: Proceedings of the IEEE international conference on computer vision. pp. 2432–2439 (2013) 3
32. Li, Y., Liu, M., Yi, Y., Li, Q., Ren, D., Zuo, W.: Two-stage single image reflection removal with reflection-aware guidance. Applied Intelligence pp. 1–16 (2023) 3
33. Liu, Y.L., Lai, W.S., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Learning to see through obstructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14215–14224 (2020) 1, 3
34. Lyu, Y., Cui, Z., Li, S., Pollefeys, M., Shi, B.: Reflection separation using a pair of unpolarized and polarized images. Advances in neural information processing systems **32** (2019) 3
35. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16190–16199 (2022) 9
36. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 3, 7, 12, 13
37. Nayar, S.K., Fang, X.S., Boult, T.: Separation of reflection components using color and polarization. International Journal of Computer Vision **21**(3), 163–186 (1997) 3
38. Qiu, J., Jiang, P.T., Zhu, Y., Yin, Z.X., Cheng, M.M., Ren, B.: Looking through the glass: Neural surface reconstruction against high specular reflections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20823–20833 (2023) 3
39. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ArXiv preprint (2021) 12
40. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020) 12
41. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 501–518. Springer (2016) 7, 9
42. Shih, Y., Krishnan, D., Durand, F., Freeman, W.T.: Reflection removal using ghosting cues. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3193–3201 (2015) 1, 3, 14
43. Sinha, S.N., Kopf, J., Goesele, M., Scharstein, D., Szeliski, R.: Image-based rendering for scenes with reflections. ACM Transactions on Graphics (TOG) **31**(4), 1–10 (2012) 3
44. Wan, R., Shi, B., Duan, L.Y., Tan, A.H., Gao, W., Kot, A.C.: Region-aware reflection removal with unified content and gradient priors. IEEE Transactions on Image Processing **27**(6), 2927–2941 (2018) 3
45. Wan, R., Shi, B., Duan, L.Y., Tan, A.H., Kot, A.C.: Crrn: Multi-scale guided concurrent reflection removal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4777–4785 (2018) 3
46. Wan, R., Shi, B., Li, H., Duan, L.Y., Kot, A.C.: Reflection scene separation from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2398–2406 (2020) 3

47. Wan, R., Shi, B., Li, H., Duan, L.Y., Kot, A.C.: Face image reflection removal. International Journal of Computer Vision **129**, 385–399 (2021) 3
48. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS (2021) 3
49. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3295–3306 (2023) 3
50. Wei, K., Yang, J., Fu, Y., Wipf, D., Huang, H.: Single image reflection removal exploiting misaligned training data and network enhancements. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8178–8187 (2019) 3
51. Wen, Q., Tan, Y., Qin, J., Liu, W., Han, G., He, S.: Single image reflection removal beyond linearity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3771–3779 (2019) 3
52. Xia, Z., Gharbi, M., Perazzi, F., Sunkavalli, K., Chakrabarti, A.: Deep denoising of flash and no-flash pairs for photography in low-light environments. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2063–2072 (2020) 2
53. Xia, Z., Lawrence, J., Achar, S.: A dark flash normal camera. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 2410–2419 (2020) 2
54. Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM Transactions on Graphics (TOG) **34**(4), 1–11 (2015) 1, 3
55. Yang, J., Gong, D., Liu, L., Shi, Q.: Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In: Proceedings of the european conference on computer vision (ECCV). pp. 654–669 (2018) 3
56. Yang, Y., Ma, W., Zheng, Y., Cai, J.F., Xu, W.: Fast single image reflection suppression via convex optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8141–8149 (2019) 1, 3
57. Zhang, X., Ng, R., Chen, Q.: Single image reflection separation with perceptual losses. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4786–4794 (2018) 3
58. Zheng, Q., Shi, B., Chen, J., Jiang, X., Duan, L.Y., Kot, A.C.: Single image reflection removal with absorption effect. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13395–13404 (2021) 3
59. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. arXiv preprint arXiv:2312.00451 (2023) 3, 11, 12
60. Zou, Z., Lei, S., Shi, T., Shi, Z., Ye, J.: Deep adversarial decomposition: A unified framework for separating superimposed images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12806–12816 (2020) 3