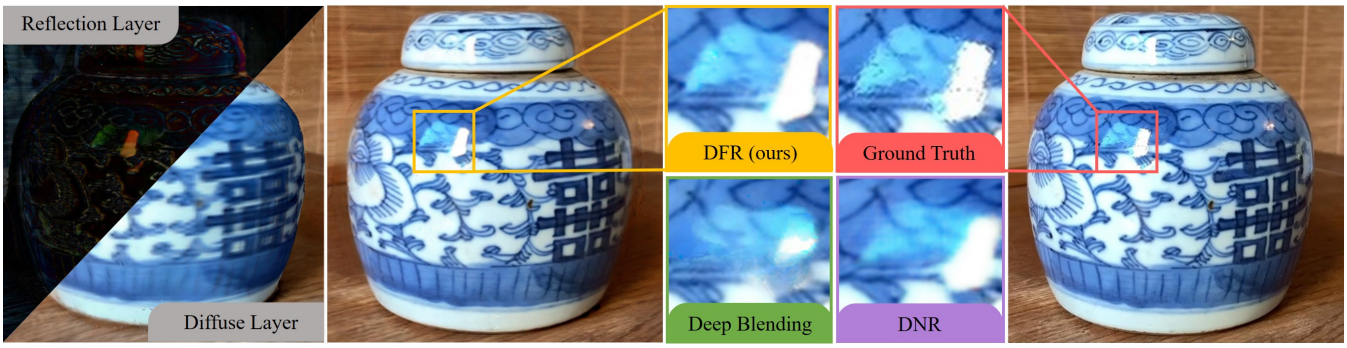
DOI: 10.1111/cgf.14593
Eurographics Symposium on Rendering 2022
A. Ghosh and L.-Y. Wei
(Guest Editors)

Volume 41 (2022), Number 4

Deep Flow Rendering: View Synthesis via Layer-aware Reflection Flow

Pinxuan Dai and Ning Xie †

Center for Future Media, School of Computer Science and Engineering,
University of Electronic Science and Technology of China, 611731, China



(a) Two-layer representation (b) Synthesis result (c) Close-up comparison (d) Ground truth

Figure 1: *We synthesize images for novel views using a two-layer representation (a) where view-dependent and view-independent features are predicted respectively. Our algorithm renders more accurate reflections (b) compared to existing methods that generate blurred and incorrect results (c).*

Abstract

Novel view synthesis (NVS) generates images from unseen viewpoints based on a set of input images. It is a challenge because of inaccurate lighting optimization and geometry inference. Although current neural rendering methods have made significant progress, they still struggle to reconstruct global illumination effects like reflections and exhibit ambiguous blurs in highly view-dependent areas. This work addresses high-quality view synthesis to emphasize reflection on non-concave surfaces. We propose Deep Flow Rendering that optimizes direct and indirect lighting separately, leveraging texture mapping, appearance flow, and neural rendering. A learnable texture is used to predict view-independent features, meanwhile enabling efficient reflection extraction. To accurately fit view-dependent effects, we adopt a constrained neural flow to transfer image-space features from nearby views to the target view in an edge-preserving manner. Then we further implement a fusing renderer that utilizes the predictions of both layers to form the output image. The experiments demonstrate that our method outperforms the state-of-the-art methods at synthesizing various scenes with challenging reflection effects.

CCS Concepts

• ***Computing methodologies*** → ***Image-based rendering; Neural networks;***

1. Introduction

To render more realistic images has always been the key goal in the computer graphics community. Rendering techniques have evolved enormously during the last few decades due to the growth of various applications and the rapid development of modern computing hardware. Photo-realistic images can be rendered through various approaches including widely used rasterization and physically based ray tracing. Indirect lighting contributes to the sense of reality of an image in a tremendous way, methods like precomputed radiance transfer [SKS02; SHHS03; SLS05; LSSS04] and radi-

† Corresponding author: seanxiening@gmail.com

ance regression functions [RWG*13; QX14] achieve promising results and largely reduce run-time computations of traditional ray tracing [Whi80]. However, these methods rely on extensive well-defined scene parameters (e.g. geometry, lighting, and camera). In order to overcome the trivial parameter configuration based on the empirical knowledge, *image-based rendering* (IBR) infers geometry and lighting from input images and uses them to render the results for novel views. General 3D reconstruction algorithms like structure-from-motion [SF16] and multi-view stereo [JP11; FP10; SZPF16] are often adopted for camera calibration and coarse geometry reconstruction.

Deep neural networks improve IBR performance by coding high-dimensional scene parameters of geometry and appearance implicitly. Neural rendering methods are able to understand and interpolate these parameters more reasonably. Most of them try to learn a high-dimensional mapping from the spatial positions and view directions to the output colors, which is described by the *Light Field* (LF) function. LF is defined as the radiance at a point in a given direction [LH96]. Classic LF rendering methods follow a direct sampling-and-rendering process, which suffers from either inaccuracy due to insufficient samples or overwhelming storage of dense sampling. Recently emerged methods boosted with deep neural networks [MST*20; TZN19; CWZ*18] have made significant progress on accurate synthesis based on limited source images.

Methods based on LF optimization vary in different geometry representations (e.g. mesh, voxel, and point cloud). Voxel-based methods [MST*20; ZSD*21; BBJ*21; GKB*21] regress on both geometry and appearance, while mesh-based methods [CWZ*18; TZT*20; CDSD13; TZN19; HRDB16; HPP*18] usually require explicit geometry representations. Our algorithm tackles the novel view synthesis problem by LF optimization based on meshed geometry. We only apply the LF function to mesh surfaces as *Surface Light Fields* (SLF) [WAA*00] that can be regarded as a simplified version of the rendering equation [Kaj86]:

$$\mathrm{SLF}(P,D_o) = \int_{\Omega} f(D_i,P,D_o)L_i(P,D_i)|\cos\theta|dD_i, \qquad (1)$$

where $\mathrm{SLF}(P,D_o)$ is the outcoming radiance from surface point P along direction D_o, $L_i(P,D_i)$ denotes incoming light that hits model surface at point P from direction D_i within the corresponding hemisphere Ω, $f(D_i,P,D_o)$ represents the bidirectional reflectance distribution function (BRDF), and θ is the angle between D_i and D_o.

Most existing algorithms for high quality novel view synthesis are either **blending-based** [BBM*01; HRDB16; HPP*18] or **generation-based** [CWZ*18; ZFT*21; MST*20; TZT*20; TZN19]. Blending-based methods learn a strategy to blend images from nearby viewpoints, while generation-based methods synthesize target images from neural scene representations. Despite well-fitted diffuse color, these methods are prone to generate either blurred or clear but wrong reflections. Flow-based implementations are limited due to the entanglement of the view-dependent and view-independent features, which should follow different rules when sampling from reference images. In this work, we solve this problem by involving a reflection extraction step to reference images and adopt a neural flow to predict the reflection layer for each target view. Thus, the SLF function is treated as two individual components:

$$\mathrm{SLF}(P,D_o) = \mathrm{SLF}_{vi}(P) + \mathrm{SLF}_{vd}(P,D_o), \qquad (2)$$

$\mathrm{SLF}_{vi}(P)$ represents the view-independent part and $\mathrm{SLF}_{vd}(P,D_o)$ represents the view-dependent part. We coarsely extract these two layers from reference images, render each layer for the target views respectively, and fuse the results to form the output images. We adopt a learnable color texture to predict the view-independent parts and extract reflection layers from source images, considering the high efficiency of texture mapping in both rendering and optimization. For the view-dependent part, the appearance flow model [ZTS*16] is adopted to refine the extracted reflection layers by exploiting the pixel coherence in 2D image space. At last, predictions of both layers are combined using a CNN-based neural renderer.

In short, we proposed a mesh-based novel view synthesis algorithm *Deep Flow Rendering* (DFR) that achieves precise reflection reconstruction, geometry correction, and consistent results for continuous frames. Our algorithm produces photo-realistic results in multiple test scenes with challenging reflection effects while running at interactive frame rates.

2. Related work

We first introduce the general geometry inference step involved in novel view synthesis algorithms and different approaches facing reconstruction imperfections in Sec. 2.1. Then, we introduce recent works of novel view synthesis in two categories: generation-based (Sec. 2.2) and blending-based (Sec. 2.3) methods. We also discuss the appearance flow model and its advantages we leveraged to predict the neural reflection flow in Sec. 2.4.

2.1. 3D Reconstruction for novel view synthesis

IBR methods for view synthesis first infer geometry information from input images and render target images based on the geometry. 3D reconstruction methods like multi-view stereo [JP11; FP10; SZPF16] and those enhanced with RGB-D scanners [CZK15; HDGN17] still cannot meet the demanding requirements of high-quality rendering due to unavoidable loss of details.

Novel view synthesis algorithms implement various techniques to correct the coarsely reconstructed geometry. Some of them [CDSD13; CDD15; HRDB16; HPP*18] involve per-view meshes to enhance the visibility at each view meanwhile keeping the global structure as consistent as possible. Recent progress in neural rendering [TZN19; HPP*18; RK20] has also proved the effectiveness of deep neural networks in correcting inaccurate geometry, which is a struggle for classic rendering. We adopt a deep neural renderer that takes geometry information as input to correct geometry imperfections.

2.2. Generation-based novel view synthesis

Rather than shading from manually designed parameters, neural rendering methods [MST*20; ZFT*21; TZT*20; TZN19; CWZ*18] generate features or images from implicit neural scene representations. Neural Radiance Fields (NeRF) [MST*20] learns

a mapping from spatial position and view direction (p,v) to color and density (c,d) using multi-layer perceptrons and renders images through a classic volume renderer. Various improvements to NeRF are proposed including relighting [BBJ*21; ZSD*21] and reflection synthesis [GKB*21]. NeRFReN [GKB*21] reproduces mirror reflections at planar surfaces by decomposing the NeRF representation into transmitted and reflected parts. Recent methods of *multiplane image* [FBD*19; WPYS21] also achieve high-quality synthesis of view-dependent effects. For mesh-based methods, Deep Surface Light Fields [CWZ*18] fits the SLF function by mapping viewpoints (x,y,z) and texture coordinates (u,v) to output colors (r,g,b), Deferred Neural Rendering (DNR) [TZN19] uses a U-Net for image generation. Park *et al.* [PHS20] achieve extrapolation of views, which is often considered difficult. It is worth noting that, most methods [TZN19; TZT*20; PHS20; MST*20; CWZ*18] design two-stream architectures to process spatial positions and view poses separately, for such divided structures help reproduce view-dependent effects in higher accuracy. In our work, we apply a more explicit separation using reflection extraction to achieve accurate reflection synthesis.

Texture mapping is a mature and flexible technique to fit complex view-independent information. DNR [TZN19] proposed *Neural Texture*, which extends the channels of RGB color texture and leaves some of the channels unconstrained. In DNR, neural textures are trained end-to-end through a CNN renderer instead of exhausting manual configurations for different feature maps. DNR deals with view-dependent effects by feeding the neural renderer with view positions projected to Spherical Harmonic bases. It struggles to reproduce highly view-dependent effects in experiments. We improve the performance by directly feeding the neural renderer with the predicted reflection layer at each target view to provide more detailed view-dependent information.

2.3. Blending-based novel view synthesis

The other approach for novel view synthesis aims to find proper weighting functions to blend nearby images smartly. Unstructured Lumigraph Rendering (ULR) [BBM*01] proposed an effective method to calculate per-pixel blending weights for nearby images by considering the differences of angle and distance terms between the target and reference views. However, some regions visible in target views are not ensured to be found in nearby images due to occlusions. Thus obvious artifacts are often observed around geometry boundaries. Inside-Out [HRDB16] improves ULR by involving a band-width selection that adjusts parameters in the weighting function adaptively based on different blending candidates. Methods using optical flow to align image features [EDM*08; BYLR20] also outperform ULR at handling occlusions. Lipski *et al.* [LLB*10] blend images in a tetrahedralized navigation space with temporal registration and enable interpolation of both space and time. Recent approaches [CDSD13; CDD15; HRDB16; HPP*18; RK20] perform the per-view geometry refinement and generate different meshes for each view that help better align geometry edges. Deep Blending [HPP*18] utilizes a CNN structure to calculate blending weights for a fixed number of "mosaics" which are pixel-wise selections of nearby images ranked by the IBR cost [HRDB16]. It achieves accurate and stable results at boundary regions.

Based on the implementation of Deep Blending, Rodriguez *et al.* [RPHD20] improve synthesis quality at car windows using semantic labels and approximated reflection flow. They explicitly calculate the reflection flow assuming that the car windows are slightly curved cylinder surfaces and achieve fast convergence. Xu *et al.* [XWZ*21] handle reflections on planar surfaces in a geometry-guided way. They follow the work of Sinha *et al.* [SKG*12] to model the geometry of reflected objects by multi-layer stereo algorithms. Our method utilizes a neural flow predictor to code the geometry in reflection space and relax the harsh constraints for reflector surfaces to fit more general cases.

2.4. Appearance flow

Appearance flow represents a per-pixel mapping between images observed from different views. It was introduced by Zhou *et al.* [ZTS*16] based on the spatial transformer [JSZ*15]. Various usages of appearance flow are implemented to predict visibility [ZTS*16; SHL*18], create bullet time effect [JLJ*18], and inpaint corrupted images [RYZ*19].

In the original implementation of Zhou *et al.* [ZTS*16], a convolutional encoder-decoder architecture takes both nearby images and their view poses as input, and outputs a full-resolution dense flow. The final output image is synthesized by sampling the input image according to the predicted dense flow. The core of this system is a differentiable sampler that can backpropagate gradients from image-space loss to previous stages. For an example of the differentiable bilinear sampler:

$$I_t^i(I_r, f(r,t)^i) = \sum_{q \in \Psi} I_r^q (1 - |x^{f(r,t)^i} - x^q|)(1 - |y^{f(r,t)^i} - y^q|), \quad (3)$$

where I_t^i denotes pixel i in the target image I_t, which is sampled from reference image I_r around a given sampling center $f(r,t)^i$. $f(r,t)^i$ describes the pixel mapping from I_r to I_t and q is each pixel inside valid sampling area Ψ around $f(r,t)^i$. The weights are designed in negative correlation with the distance terms along x and y axes. During optimization, sampling centers are dragged towards optimal positions to obtain larger weights for pixels in desire. The target of the appearance flow model is to predict the full-resolution flow $f(r,t)$ for every possible pair of (I_r, I_t). For cases of multiple input reference images, the appearance flow model blends each prediction using another output channel as per-pixel blending weights.

Such a neural flow network exploits the coherence of features in image space and reuses existing information in nearby views. This pixel-level manipulation enables interpolating and transferring features of interest across 2D pixel coordinates with comparably low loss of details. In this work, we construct a constrained neural flow predictor to synthesize the reflection layers at novel views.

3. Method

The motivation of this work is to synthesize photo-realistic reflection for novel views. Due to the entanglement of the reflection and diffuse layers in reference images, it is difficult to obtain accurate reflection effects by direct blending (Fig. 3). We propose Deep

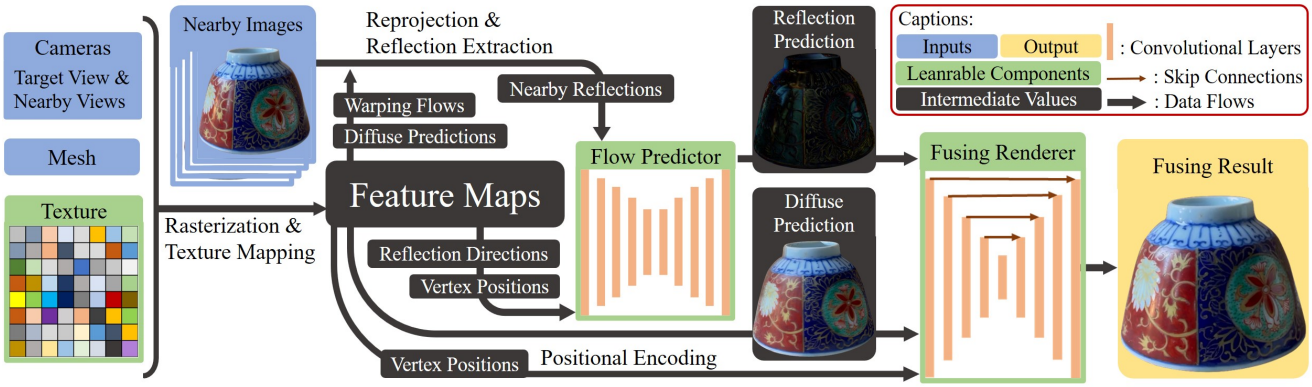


Figure 2: *An overview of the Deep Flow Rendering framework. The feature maps (including warping flows for reprojection, diffuse predictions sampled from texture, reflection directions representing view information, and vertex positions) for the reference and target views are calculated in a differentiable rasterization pipeline. The flow predictor takes the reflection extraction (Sec. 3.1) result, the reflection directions of the target and nearby views, and vertex positions of the target view as input to synthesize the reflection prediction (Sec. 3.2) under a sampling constraint (Sec. 3.3). The Fusing renderer (Sec. 3.4) combines predictions of both layers and the positionally encoded vertex positions of the target view to form the output result.*

Flow Rendering to tackle this problem. An overview of our algorithm is presented in Fig. 2, and we introduce each component in this section.

3.1. Reflection extraction

To achieve higher synthesis quality of reflections, we first perform layer extraction to images being referenced, following the classic assumption in the reflection removal problem: an image is the linear combination of a diffuse layer and a reflection layer. We apply the extraction at the target view, so a nearby image I_r is warped from the reference view V_r to the target view V_t by bilinear sampling according to a reprojection warping flow F_t^r that maps pixels from V_r to V_t.

We train a learnable texture from scratch for diffuse predictions leveraging recent progress on differentiable rendering [LHK*20]. The texture can represent the average appearance color stably after adequate training iterations. Underfitting the view-dependent effects enables the extraction for reflection layers of images. Denoting $\tilde{D}_t$ as the diffuse prediction at the target view V_t and forcing the linear combination assumption, we extract the reflection layer R_t^r at V_t by:

$$R_t^r = I_t^r - \tilde{D}_t, \tag{4}$$

where I_t^r is the reference image warped from V_r to V_t using the corresponding flow F_t^r computed in the rasterization step. We also calculate a visibility mask to select visible regions at the reference view from the extracted reflection layer to enhance the robustness when facing geometry occlusions.

As the reprojection warping aligns diffuse features of images, reflection features $(R_t^{r_0}, R_t^{r_1}, ..., R_t^{r_i})$ obtained from nearby views $(V_{r_0}, V_{r_1}, ..., V_{r_i})$ are often misaligned (Fig. 3). Thus we further implement a neural flow predictor (Sec. 3.2) to obtain clear and correct reflection effects.

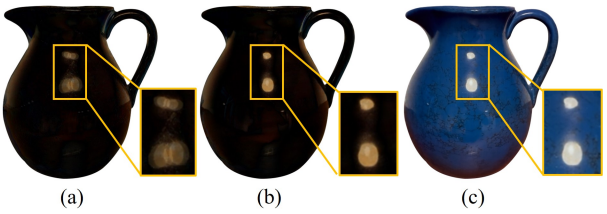


Figure 3: *Refine misaligned reflections: (a) warped and blended reflection, (b) refined prediction, and (c) fusing result combining diffuse prediction.*

3.2. Reflection synthesis

We adopt a neural flow predictor to align and refine extracted reflection features for target views by pixel-wise replacement. Explicit computation for reflection flows requires geometry in reflection space, which is difficult to infer from curved reflector surfaces. To this end, our flow prediction network is designed to learn the high-dimensional mapping from the 3D geometry in reflection space to the movements of reflected features in 2D images through a data-driven approach.

As reflection extraction is considered an ill-posed problem, it is impossible to obtain perfectly extracted results for all input images, whether in linear or gamma color space. As a result, the appearance flow model is prone to learn extraction errors in reflection layers and often overfits reference views. In experiments, it fails to make globally consistent predictions and results in non-smooth changes between continuous frames. We design a modified appearance flow [ZTS*16] network with a robust sampling strategy to counter the challenge of processing reflection layers with errors. Our insights lie in two points: (1) utilizing an additional channel in the output layer, which performs as a pixel-wise confidence mask to

suppress non-reflection errors in the extracted reflection layers, and (2) setting a sampling constraint on the predicted flow to improve the generalization ability of the system across different views.

Unlike the original appearance flow described in Sec. 2.4, we provide geometry information as input, together with the pixel-wise difference of the reflection direction feature maps at target and reference views. As shown in Fig. 4, our network outputs a pair of re-scaling values $[x_i, y_i]$ to apply the sampling constraint (Sec. 3.3), a blending weight $W_t^{r_i}$, and a confidence mask $C_t^{r_i}$ that learns to suppress undesired values in non-reflection regions. All blending weights are normalized by a softmax function and used to aggregate reflection predictions $\tilde{R}_t^{r_i}$ sampled from $R_t^{r_i}$. The refined reflection layer at the target view V_t referring to a set $\mathbf{N}$ of n nearest reference views can be calculated as:

$$\tilde{R}_t = \sum_{r_i \in \mathbf{N}} \tilde{R}_t^{r_i} C_t^{r_i} W_t^{r_i}. \tag{5}$$

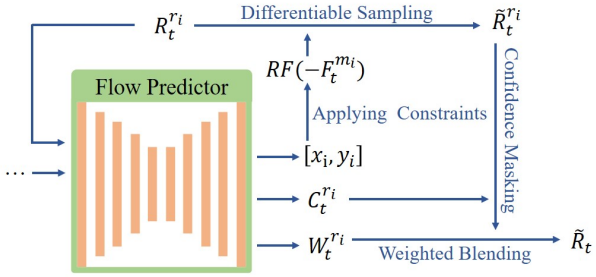


Figure 4: *Steps of reflection synthesis. All intermediate values are in full-resolution size as the input and output images.*

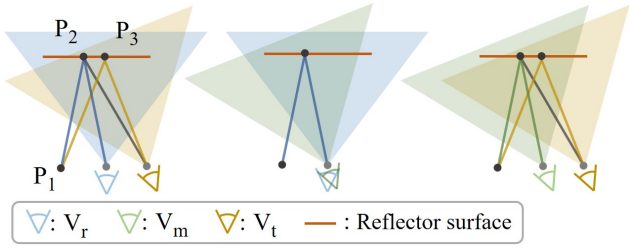


Figure 5: *Two-step decomposition of camera transformations. Left: the reflected ray from P_1 and a ray from P_2 are originally overlapped when observing from V_r. They separate when camera moves to V_t as the ray from P_1 is now reflected at P_3. This deviation happens with translation (right) and is irrelevant to rotation (middle).*

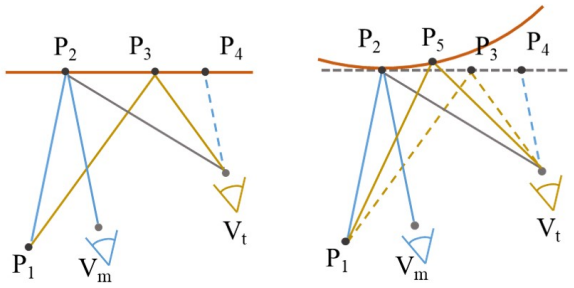


Figure 6: *Reflection movements on non-concave surfaces during camera translations. The reflected ray originating from P_1 hits the camera at V_m with directions $P_2 \to V_m$. $P_4 \to V_t$ is parallel to $P_2 \to V_m$ denoting the incident direction of $P_2 \to V_m$. When the camera translates to V_t, the reflection feature at P_1 moves to P_3 in the planar case (left) and P_5 in the convex case (right).*

3.3. Sampling constraint

As an unconstrained flow model is vulnerable to local optimums, we propose a regularization method to constrain the reflection flow used for sampling. This sampling constraint significantly improves the frame-to-frame consistency as well as synthesis quality. We first decompose the reprojection warping process into two independent steps and apply the constraint using the decomposed features.

The relative change of two camera poses can be described as a rotation and a translation. We treat the rotation and translation that transform the image from the reference view V_r to the target view V_t separately by involving a mid-stage view V_m as: $V_m = \text{rotate}(V_r)$, $V_t = \text{translate}(V_m)$. The warped image at V_t can be obtained by two-step warping using F_m^r and F_t^m instead of direct warping using F_t^r. Such reprojection warpings align diffuse features in images, but reflections are misaligned. As shown in Fig. 5, the relative position of the reflection and diffuse layers only changes with camera translations, while camera rotations keep these two layers visually "glued" together. We leverage this information to sample reflection features by modifying the second-step flow F_t^m that is responsible for the camera translation.

In Fig. 6, we visualize how the incident rays from a reflected point P_1 and a diffuse point P_2 change when the camera translates from V_m to V_t. Note that P_1 and P_2 are originally overlapped when observing from V_m. In the planar case, cameras at V_m and V_t see the reflection of P_1 from directions $P_2 \to V_m$ and $P_3 \to V_t$, and the angular difference is $\angle P_4 V_t P_3$. Likewise, cameras at V_m and V_t see the diffuse point P_2 from directions $P_2 \to V_m$ and $P_2 \to V_t$, and the angular difference is $\angle P_4 V_t P_2$. We have: $0 \le \angle P_4 V_t P_3 \le \angle P_4 V_t P_2$ and similar relations can also be found in the convex case (the point reflecting P_1 moves from P_3 to P_5): $0 \le \angle P_4 V_t P_3 \le \angle P_4 V_t P_5 \le \angle P_4 V_t P_2$. So the incident direction of P_3 or P_5 at V_t is within $\angle P_4 V_t P_2$.

We know that the incident directions of rays determine the projected positions of the corresponding features in image planes. The pixel mapping from P_4 to P_2 observing from V_t (corresponding to the angular change of $\angle P_4 V_t P_2$) is described by the previously extracted warping flow F_t^m. Thus, the desired P_3 or P_5 can be sampled using a flow between $[0 \times F_t^m, 1 \times F_t^m]$ and it can be done by re-scaling F_t^m. To further broadcast this assumption to more general cases, we predict two individual scale factors for each dimension of F_t^m. Therefore we derive our sampling constraint and form the predicted reflection flow as Eq. 6, where $\text{RF}(F_t^m)$ denotes the reflection flow constrained with F_t^m and $[x, y]$ is a pair of scale factors

outputted by the neural flow predictor. Parameters ε and γ together control the range of the output flow, we set $\varepsilon = 0.05$ and $\gamma = 0.8$ in all experiments.

$$\mathrm{RF}(F_t^m) = \varepsilon + \gamma \cdot \mathrm{sigmoid}([x,y]) \cdot F_t^m, \tag{6}$$

We sample the reflection layer for the target view V_t based on a reference view V_r by Eq. 7a, where R_m^r is the reflection layer at V_m warped from V_r in the first-step warping. While following Eq. 7a needs an additional rasterization step for V_m, we simplify the computations and approximate the result by sampling the reflection layer R_t^r at V_t according to the re-scaled $-F_t^m$ (Eq. 7c) instead. Then the refined reflection can be formed as the weighted-sum of all masked $\tilde{R}_t^{r_i}$ following Eq. 5.

$$\begin{aligned} \tilde{R}_t^r &= \mathrm{sampling}(R_m^r, \mathrm{RF}(F_t^m)) && (7a)\\ &\approx \mathrm{sampling}(\mathrm{sampling}(R_m^r, F_t^m), \mathrm{RF}(-F_t^m)) && (7b)\\ &\approx \mathrm{sampling}(R_t^r, \mathrm{RF}(-F_t^m)) && (7c) \end{aligned}$$

3.4. Fusing rendering

With both the diffuse and reflection predictions, it is common to simply add them together to obtain the output image. However, the summed results suffer from severe artifacts (Fig. 9). We adopt a U-Net with skip connections [HZRS16] to fuse the diffuse and reflection layers into the output image meanwhile applying geometry correction. We feed the neural fusing renderer with vertex positions of the target view along with the two-layer predictions. Instead of directly using the 3D vertex positions, we found mapping them into a higher-dimensional space produces better results (Fig. 9). So we apply a high-frequency *positional encoding* ($L = 10$) to vertex positions P (Eq. 8) and generate the final output image following Eq. 9.

$$\begin{aligned} \mathrm{PE}(P) = [&\sin(2^0\pi P), \cos(2^0\pi P),\\ &\sin(2^1\pi P), \cos(2^1\pi P),\\ &\ldots,\\ &\sin(2^{L-1}\pi P), \cos(2^{L-1}\pi P)] \end{aligned} \tag{8}$$

$$\tilde{I}_t = \mathrm{fusingRenderer}(\tilde{R}_t, \tilde{D}_t, \mathrm{PE}(P_t)) \tag{9}$$

4. Implementation details

We build the system on TensorFlow [MAP*15], access differentiable rasterization through nvdiffrast [LHK*20], and use the Adam optimizer [KB14] with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\varepsilon = 10^{-6}$. Convolutional layers in the flow predictor and the fusing renderer are configured with kernel sizes of 4, strides of 2, and ReLU activations except for the output layers. The RGB output of the fusing renderer is constrained by a sigmoid function. Instance normalization is applied to the fusing renderer except for the last layer. The whole system is jointly trained by 200K iterations of stochastic gradient descent with a learning rate of 5×10^{-4} on an NVIDIA RTX TITAN graphic card. We also apply random cropping of size 256×256 to avoid overfitting.

4.1. Loss function

Our objective function includes four parts, an overall image **generation** loss, individual losses for the **diffuse** and **reflection** layers, and a **temporal** loss to enhance frame-to-frame consistency (Eq. 10).

$$L = L_G + L_D + L_R + L_T. \tag{10}$$

We first employ an l_1 loss on the final output image $\tilde{I}_t$ and the ground truth image I_t. To enhance the visual similarity, we also apply a perceptual loss [JAF16] defined on the first and second ReLU outputs of a pre-trained VGG-19 network. This helps the model to converge by matching high-level perceptual features rather than pixel similarity. We randomly select one reference view as the prediction target at each training iteration. Our generation loss is defined as:

$$\begin{aligned} L_G(I_t, \tilde{I}_t) = &\mid I_t - \tilde{I}_t \mid\\ &+ \mid VGG_{relu1}(I_t) - VGG_{relu1}(\tilde{I}_t) \mid\\ &+ \mid VGG_{relu2}(I_t) - VGG_{relu2}(\tilde{I}_t) \mid . \end{aligned} \tag{11}$$

We guide the two-layer representation to make stable predictions respectively using two individual objective terms:

$$L_D(I_t, \tilde{D}_t) = \mid I_t - \tilde{D}_t \mid, \tag{12}$$

$$L_R = \mid R_t - \tilde{R}_t \mid + \sum_{r_i \in \mathbf{N}} \frac{\mid R_t - C_t^{r_i} \tilde{R}_t^{r_i} \mid}{\mid \mathbf{N} \mid}, \tag{13}$$

where $R_t = I_t - \tilde{D}_t$ is the ground truth reflection layer at V_t. L_D forces diffuse texture to learn the average color of all references. L_R valuates reflection prediction both before and after the weighted blending, which helps the model learn to not only synthesize each reflection layer but also blend them smartly.

Eq. 11, 12, and 13 guarantee photo-realistic synthesis for each single target view, but the results still suffer from unstable flickering between continuous frames. To this end, we design two temporal loss terms to obtain more natural inter-frame transitions. Firstly, we generate two close temporal views V_{temp0} and V_{temp1} between two adjacent reference views and force their predictions to be as similar as possible. Besides, as geometry recovered from images usually contains highly fragmented faces that can only be seen at very limited views, diffuse predictions from texture mapping often run into an undersampling problem. We solve it by warping an adjacent reference image to V_{temp0} and use it to optimize the texture. So our temporal loss is set to be:

$$L_T = L_G(\tilde{I}_{temp0}, \tilde{I}_{temp1}) + L_D(I^r_{temp0}, \tilde{D}_{temp0}). \tag{14}$$

4.2. Data acquisition

We record unstructured videos with a hand-held iPhoneX and extract reference images with an averaged angular difference of 6°. These reference images are then used to reconstruct the coarse geometry using COLMAP [SF16; SZPF16]. Then we apply automatic *uv*-unwrapping, HC Laplacian smoothing [VMM99], and quadric edge collapse simplification [GH97] to the reconstructed mesh. Synthesis data are rendered through Mitsuba2 [NVZJ19]. We define $\mathbf{N}$ as a set of the nearest n reference views to the target view and set $n = 4$ as default (see comparisons in Fig. 10).

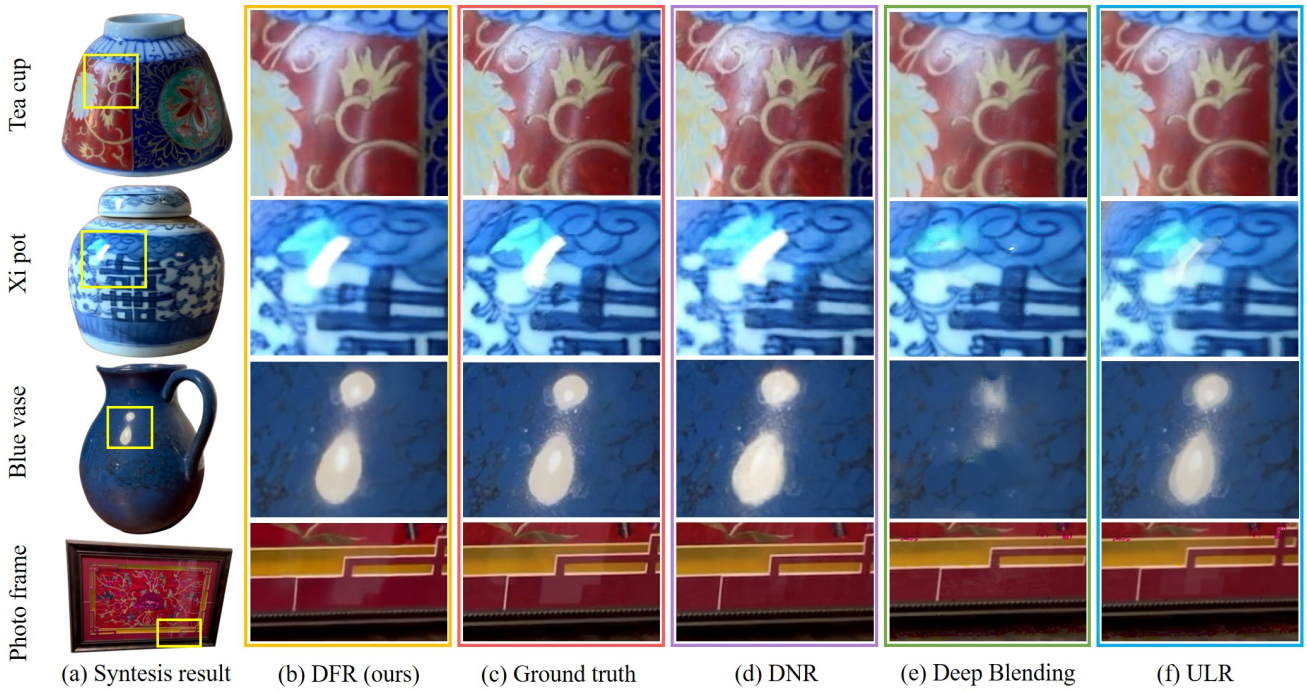


Figure 7: *Comparisons for real data: (a) synthesis result of our method, and (b)-(f) close-ups for reflection regions. Our method generates correct shapes for reflections in the Tea cup and Xi pot scene. We also reproduce clear light bulbs in the Blue vase scene and the reflected corner in the Photo frame scene, where other methods failed.*

Testing images are extracted from the same video as those for training and are registered to the existing reconstruction project. This local registration step generates a set of camera calibrations for testing, while keeping the 3D model recovered from training images unchanged. We manually select the target objects from rendered images to better present some example figures in this paper.

5. Results

In this section, we compare our algorithm with the state-of-the-art methods on both real-captured and synthetic data. Then, we analyze the validity of each component in our system.

5.1. Comparision

For real data, we compare with Deferred Neural Rendering (DNR) [TZN19], Deep Blending [HPP*18], and the classic approach of Unstructured Lumigraph Rendering (ULR) [BBM*01]. Our method synthesizes more accurate reflections (Fig. 7), and also achieves higher scores under multiple metrics (Table 1).

Table 1: *Average scores across all testing scenes*

Metrics	DFR (ours)	DNR	Deep Blending	ULR
L1 ↓	**0.0192**	0.0230	0.0720	0.0783
VGG ↓	**0.0028**	0.0033	0.0088	0.0098
PSNR ↑	**32.905**	30.214	20.574	19.759

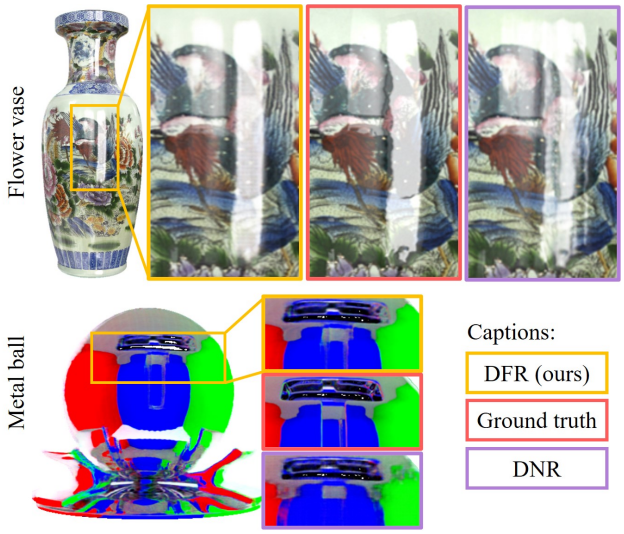


Figure 8: *Comparisons for synthetic data: the china vase and metal ball reflect the surrounding environment. Our algorithm generates more accurate reflection boundaries than DNR.*

For the synthetic data, known meshes are employed. We challenge the algorithms with highly reflective materials (china and metal), and ours produces clearer results compared to DNR (Fig. 8).

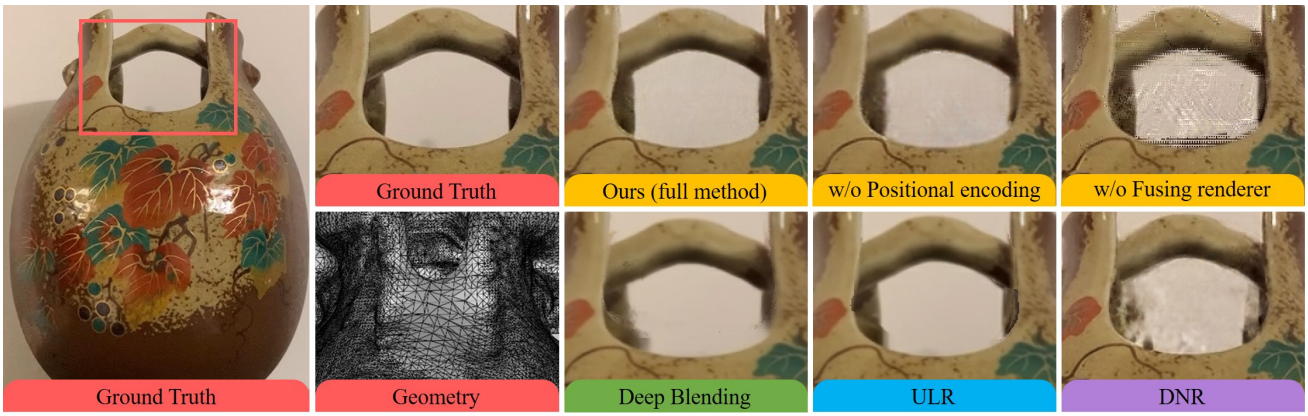


Figure 9: *Geometry correction test. In this scene, the reconstructed geometry falsely fills a big hole. Our full method produces the clearest result compared to our ablated models and other methods.*

Our method is able to reproduce mirror reflections on curved surfaces based on accurate geometry. However, it is still challenging in real scenes because the 3D reconstruction method we adopted fails to recover mirror-like surfaces. As our system is agnostic to a specific reconstruction algorithm, it will benefit from future improvements in geometry inference techniques.

5.2. Ablation study

We conduct an ablation study to examine the effectiveness of the key components in overall synthesis quality and geometry correctness. Then we investigate the influence of the number n of nearby reference images at rendering.

Table 2: *Ablation scores for the Tea cup scene.*

Methods	L1 ↓	VGG ↓	PSNR ↑
Full method	**0.0206**	**0.0027**	**32.128**
w/o Reflection flow	0.0267	0.0052	29.512
w/o Sampling constraint	0.0239	0.0047	30.770
w/o Confidence mask	0.0256	0.0049	30.072
w/o Fusing renderer	0.0213	**0.0027**	31.602
w/o Temporal loss	**0.0196**	0.0037	**32.483**

Table 2 shows that all components help produce more accurate predictions except the temporal loss. However, the temporal loss is essential to maintain multi-frame consistency that significantly improves the sense of reality (please refer to supplementary videos for details). Thus we consider the slight drop in accuracy acceptable. We also set the gap between the two temporal views in Eq. 14 user-configurable to balance the trade-off between single-frame accuracy and multi-frame consistency.

We show a case with severe geometry errors in Fig. 9. Inaccurate geometry involves conflicts in depth estimations when observing from multiple views, and results in flickering and blurring in output images. The results in Fig. 9 show that models with the neural flow (yellow label) better align geometry edges, the fusing renderer

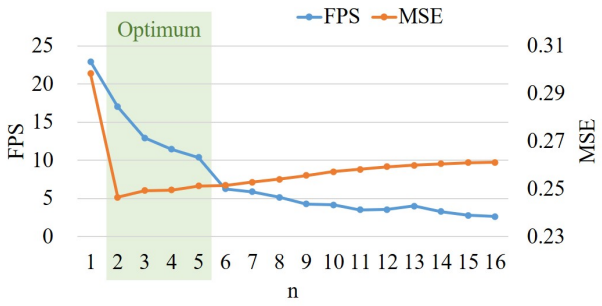


Figure 10: *A test of referring to different numbers of nearby images when rendering the Xi pot scene. The model contains around* 5×10^4 *triangles, and the output resolution is* 640×640.

significantly alleviates noise, and positional encoding further helps recover more details. As DNR relies on the neural texture which is highly correlated to the reconstructed mesh, it results in artifacts around regions with inaccurate geometry.

Another experiment is conducted to test how the number of reference images n influences the rendering speed and synthesis quality (Fig. 10). We set $n = 4$ at training and test different values for n at rendering. The result shows that increasing n slows down rendering speed and increases the error rate. We consider the optimal values for n are between 2 and 5.

6. Conclusion

In this work, we have presented Deep Flow Rendering for interactive novel view synthesis with accurate reflections on curved surfaces. Our method outperforms multiple state-of-the-art methods by exploiting image-space coherence using the constrained reflection flow. It is also robust at repairing geometry errors and preserving frame-to-frame consistency.

For limitations, this algorithm relies on the reconstructed mesh;

it fails when the inferred geometry is heavily corrupted. Like many IBR algorithms, ours is good at interpolating views, but extrapolation quality is not guaranteed. Complex reflections on concave surfaces are not included. Besides, this is a scene-specific algorithm that should be trained for each particular scene. In future works, we look forward to exploring improvements that expand the generalization ability across multiple scenes and involve more degrees of freedom for scene parameters to enable interactive scene editing.

Acknowledgements

This work was supported by Development and Application Demonstrations of Digitalized Governance of Local Society for Future Cities Research Program (2021-JB00-00033-GX), and the National Natural Science Foundation of China under Grant NO. 61976156.

References

[BBJ*21] BOSS, MARK, BRAUN, RAPHAEL, JAMPANI, VARUN, et al. "NeRD: Neural Reflectance Decomposition From Image Collections". *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, 12684–12694 2, 3.

[BBM*01] BUEHLER, CHRIS, BOSSE, MICHAEL, MCMILLAN, LEONARD, et al. "Unstructured Lumigraph Rendering". *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 425–432. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383309 2, 3, 7.

[BYLR20] BERTEL, TOBIAS, YUAN, MINGZE, LINDROOS, REUBEN, and RICHARDT, CHRISTIAN. "OmniPhotos: Casual 360° VR Photography with Motion Parallax". *SIGGRAPH Asia 2020 Emerging Technologies*. SA '20. event-place: Virtual Event, Republic of Korea. New York, NY, USA: Association for Computing Machinery, 2020. ISBN: 978-1-4503-8110-9. DOI: 10.1145/3415255.3422884 3.

[CDD15] CAYON, RODRIGO ORTIZ, DJELOUAH, ABDELAZIZ, and DRETTAKIS, GEORGE. "A Bayesian Approach for Selective Image-Based Rendering Using Superpixels". *2015 International Conference on 3D Vision*. 2015, 469–477. DOI: 10.1109/3DV.2015.59 2, 3.

[CDSD13] CHAURASIA, GAURAV, DUCHENE, SYLVAIN, SORKINE-HORNUNG, OLGA, and DRETTAKIS, GEORGE. "Depth Synthesis and Local Warps for Plausible Image-Based Navigation". *ACM Trans. Graph.* 32.3 (July 2013). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/2487228.2487238 2, 3.

[CWZ*18] CHEN, ANPEI, WU, MINYE, ZHANG, YINGLIANG, et al. "Deep Surface Light Fields". *Proc. ACM Comput. Graph. Interact. Tech.* 1.1 (July 2018). Place: New York, NY, USA Publisher: Association for Computing Machinery. DOI: 10.1145/3203192 2, 3.

[CZK15] CHOI, SUNGJOON, ZHOU, QIAN-YI, and KOLTUN, VLADLEN. "Robust reconstruction of indoor scenes". *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, 5556–5565. DOI: 10.1109/CVPR.2015.7299195 2.

[EDM*08] EISEMANN, M., DE DECKER, B., MAGNOR, M., et al. "Floating Textures". *Computer Graphics Forum* (2008). Publisher: The Eurographics Association and Blackwell Publishing Ltd. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2008.01138.x 3.

[FBD*19] FLYNN, JOHN, BROXTON, MICHAEL, DEBEVEC, PAUL, et al. "DeepView: View Synthesis With Learned Gradient Descent". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 3.

[FP10] FURUKAWA, YASUTAKA and PONCE, JEAN. "Accurate, Dense, and Robust Multiview Stereopsis". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.8 (2010), 1362–1376. DOI: 10.1109/TPAMI.2009.161 2.

[GH97] GARLAND, MICHAEL and HECKBERT, PAUL S. "Surface Simplification Using Quadric Error Metrics". *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, 209–216. ISBN: 0-89791-896-7. DOI: 10.1145/258734.258849 6.

[GKB*21] GUO, YUANCHEN, KANG, DI, BAO, LINCHAO, et al. "NeRFReN: Neural Radiance Fields with Reflections". *CoRR* abs/2111.15234 (2021). arXiv: 2111.15234 2, 3.

[HDGN17] HUANG, JINGWEI, DAI, ANGELA, GUIBAS, LEONIDAS, and NIESSNER, MATTHIAS. "3Dlite: Towards Commodity 3D Scanning for Content Creation". *ACM Trans. Graph.* 36.6 (Nov. 2017). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3130800.3130824 2.

[HPP*18] HEDMAN, PETER, PHILIP, JULIEN, PRICE, TRUE, et al. "Deep blending for free-viewpoint image-based rendering". *ACM Transactions on Graphics (TOG)* 37.6 (2018). Publisher: ACM New York, NY, USA, 1–15 2, 3, 7.

[HRDB16] HEDMAN, PETER, RITSCHEL, TOBIAS, DRETTAKIS, GEORGE, and BROSTOW, GABRIEL. "Scalable inside-out image-based rendering". *ACM Transactions on Graphics (TOG)* 35.6 (2016). Publisher: ACM New York, NY, USA, 1–11 2, 3.

[HZRS16] HE, KAIMING, ZHANG, XIANGYU, REN, SHAOQING, and SUN, JIAN. "Deep Residual Learning for Image Recognition". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 6.

[JAF16] JOHNSON, JUSTIN, ALAHI, ALEXANDRE, and FEI-FEI, LI. "Perceptual losses for real-time style transfer and super-resolution". *European conference on computer vision*. Springer, 2016, 694–711 6.

[JLJ*18] JIN, SHI, LIU, RUIYNAG, JI, YU, et al. "Learning to Dodge A Bullet: Concyclic View Morphing via Deep Learning". *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018 3.

[JP11] JANCOSEK, MICHAL and PAJDLA, TOMAS. "Multi-view reconstruction preserving weakly-supported surfaces". *CVPR 2011*. 2011, 3121–3128. DOI: 10.1109/CVPR.2011.5995693 2.

[JSZ*15] JADERBERG, MAX, SIMONYAN, KAREN, ZISSERMAN, ANDREW, et al. "Spatial transformer networks". *Advances in neural information processing systems* 28 (2015), 2017–2025 3.

[Kaj86] KAJIYA, JAMES T. "The Rendering Equation". *SIGGRAPH Comput. Graph.* 20.4 (Aug. 1986). Place: New York, NY, USA Publisher: Association for Computing Machinery, 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902 2.

[KB14] KINGMA, DIEDERIK P. and BA, JIMMY. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980 6.

[LH96] LEVOY, MARC and HANRAHAN, PAT. "Light field rendering". *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, 31–42 2.

[LHK*20] LAINE, SAMULI, HELLSTEN, JANNE, KARRAS, TERO, et al. "Modular Primitives for High-Performance Differentiable Rendering". *ACM Trans. Graph.* 39.6 (Nov. 2020). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3414685.3417861 4, 6.

[LLB*10] LIPSKI, CHRISTIAN, LINZ, CHRISTIAN, BERGER, KAI, et al. "Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time". *Computer Graphics Forum* 29 (Dec. 2010), 2555–2568. DOI: 10.1111/j.1467-8659.2010.01824.x 3.

[LSSS04] LIU, XINGUO, SLOAN, PETER-PIKE J, SHUM, HEUNG-YEUNG, and SNYDER, JOHN. "All-Frequency Precomputed Radiance Transfer for Glossy Objects." *Rendering Techniques* 2004 (2004) 1.

[MAP*15] MARTÍN ABADI, ASHISH AGARWAL, PAUL BARHAM, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015 6.

[MST*20] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". *Computer Vision – ECCV 2020*. Ed. by VEDALDI, ANDREA, BISCHOF, HORST, BROX, THOMAS, and FRAHM, JAN-MICHAEL. Cham: Springer International Publishing, 2020, 405–421. ISBN: 978-3-030-58452-8 2, 3.

[NVZJ19] NIMIER-DAVID, MERLIN, VICINI, DELIO, ZELTNER, TIZIAN, and JAKOB, WENZEL. "Mitsuba 2: A Retargetable Forward and Inverse Renderer". *ACM Trans. Graph.* 38.6 (Nov. 2019). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3355089.3356498 6.

[PHS20] PARK, JEONG JOON, HOLYNSKI, ALEKSANDER, and SEITZ, STEVEN M. "Seeing the World in a Bag of Chips". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 3.

[QX14] QIN, XUE and XIAO, SHUANGJIU. "Transparent-Supported Radiance Regression Function". *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '14. event-place: Shenzhen, China. New York, NY, USA: Association for Computing Machinery, 2014, 197–200. ISBN: 978-1-4503-3254-5. DOI: 10.1145/2670473.2670498 2.

[RK20] RIEGLER, GERNOT and KOLTUN, VLADLEN. "Free View Synthesis". *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*. event-place: Glasgow, United Kingdom. Berlin, Heidelberg: Springer-Verlag, 2020, 623–640. ISBN: 978-3-030-58528-0. DOI: 10.1007/978-3-030-58529-7_37 2, 3.

[RPHD20] RODRIGUEZ, SIMON, PRAKASH, SIDDHANT, HEDMAN, PETER, and DRETTAKIS, GEORGE. "Image-Based Rendering of Cars using Semantic Labels and Approximate Reflection Flow". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3.1 (May 2020) 3.

[RWG*13] REN, PEIRAN, WANG, JIAPING, GONG, MINMIN, et al. "Global Illumination with Radiance Regression Functions". *ACM Trans. Graph.* 32.4 (July 2013). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/2461912.2462009 2.

[RYZ*19] REN, YURUI, YU, XIAOMING, ZHANG, RUONAN, et al. "StructureFlow: Image Inpainting via Structure-Aware Appearance Flow". *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 3.

[SF16] SCHÖNBERGER, JOHANNES L. and FRAHM, JAN-MICHAEL. "Structure-from-Motion Revisited". *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, 4104–4113. DOI: 10.1109/CVPR.2016.445 2, 6.

[SHHS03] SLOAN, PETER-PIKE, HALL, JESSE, HART, JOHN, and SNYDER, JOHN. "Clustered Principal Components for Precomputed Radiance Transfer". *ACM Trans. Graph.* 22.3 (July 2003). Place: New York, NY, USA Publisher: Association for Computing Machinery, 382–391. ISSN: 0730-0301. DOI: 10.1145/882262.882281 1.

[SHL*18] SUN, SHAO-HUA, HUH, MINYOUNG, LIAO, YUAN-HONG, et al. "Multi-view to Novel View: Synthesizing Novel Views with Self-Learned Confidence". *European Conference on Computer Vision*. 2018 3.

[SKG*12] SINHA, SUDIPTA N., KOPF, JOHANNES, GOESELE, MICHAEL, et al. "Image-Based Rendering for Scenes with Reflections". *ACM Trans. Graph.* 31.4 (July 2012). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/2185520.2185596 3.

[SKS02] SLOAN, PETER-PIKE, KAUTZ, JAN, and SNYDER, JOHN. "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments". *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '02. event-place: San Antonio, Texas. New York, NY, USA: Association for Computing Machinery, 2002, 527–536. ISBN: 1-58113-521-1. DOI: 10.1145/566570.566612 1.

[SLS05] SLOAN, PETER-PIKE, LUNA, BEN, and SNYDER, JOHN. "Local, Deformable Precomputed Radiance Transfer". *ACM Trans. Graph.* 24.3 (July 2005). Place: New York, NY, USA Publisher: Association for Computing Machinery, 1216–1224. ISSN: 0730-0301. DOI: 10.1145/1073204.1073335 1.

[SZPF16] SCHÖNBERGER, JOHANNES LUTZ, ZHENG, ENLIANG, POLLEFEYS, MARC, and FRAHM, JAN-MICHAEL. "Pixelwise View Selection for Unstructured Multi-View Stereo". *European Conference on Computer Vision (ECCV)*. 2016 2, 6.

[TZN19] THIES, JUSTUS, ZOLLHÖFER, MICHAEL, and NIESSNER, MATTHIAS. "Deferred Neural Rendering: Image Synthesis Using Neural Textures". *ACM Trans. Graph.* 38.4 (July 2019). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3306346.3323035 2, 3, 7.

[TZT*20] THIES, JUSTUS, ZOLLHÖFER, MICHAEL, THEOBALT, CHRISTIAN, et al. "Image-guided Neural Object Rendering". *International Conference on Learning Representations*. 2020 2, 3.

[VMM99] VOLLMER, J., MENCL, R., and MÜLLER, H. "Improved Laplacian Smoothing of Noisy Surface Meshes". *Computer Graphics Forum* 18.3 (1999). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00334, 131–138. DOI: 10.1111/1467-8659.00334 6.

[WAA*00] WOOD, DANIEL N., AZUMA, DANIEL I., ALDINGER, KEN, et al. "Surface Light Fields for 3D Photography". *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., 2000, 287–296. ISBN: 1-58113-208-5. DOI: 10.1145/344779.344925 2.

[Whi80] WHITTED, TURNER. "An Improved Illumination Model for Shaded Display". *Commun. ACM* 23.6 (June 1980), 343–349. ISSN: 0001-0782. DOI: 10.1145/358876.358882 2.

[WPYS21] WIZADWONGSA, SUTTISAK, PHONGTHAWEE, PAKKAPON, YENPHRAPHAI, JIRAPHON, and SUWAJANAKORN, SUPASORN. "NeX: Real-Time View Synthesis With Neural Basis Expansion". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, 8534–8543 3.

[XWZ*21] XU, JIAMIN, WU, XIUCHAO, ZHU, ZIHAN, et al. "Scalable Image-Based Indoor Scene Rendering with Reflections". *ACM Trans. Graph.* 40.4 (July 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3450626.3459849 3.

[ZFT*21] ZHANG, XIUMING, FANELLO, SEAN, TSAI, YUN-TA, et al. "Neural Light Transport for Relighting and View Synthesis". *ACM Trans. Graph.* 40.1 (Jan. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3446328 2.

[ZSD*21] ZHANG, XIUMING, SRINIVASAN, PRATUL P., DENG, BOYANG, et al. "NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination". *ACM Trans. Graph.* 40.6 (Dec. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/3478513.3480496 2, 3.

[ZTS*16] ZHOU, TINGHUI, TULSIANI, SHUBHAM, SUN, WEILUN, et al. "View synthesis by appearance flow". *European conference on computer vision*. Springer, 2016, 286–301 2–4.