TöRF: Time-of-Flight Radiance Fields for Dynamic Scene View Synthesis

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural networks can represent and accurately reconstruct radiance fields for static 3D scenes (e.g., NeRF). Several works extend these to dynamic scenes captured with monocular video, with promising performance. However, the monocular setting is known to be an under-constrained problem, and so methods rely on data-driven priors for reconstructing dynamic content. We replace these priors with measurements from a time-of-flight (ToF) camera, and introduce a neural representation based on an image formation model for continuous-wave ToF cameras. Instead of working with processed depth maps, we model the raw ToF sensor measurements to improve reconstruction quality and avoid issues with low reflectance regions, multi-path interference, and a sensor's limited unambiguous depth range. We show that this approach improves robustness of dynamic scene reconstruction to erroneous calibration and large motions, and discuss the benefits and limitations of integrating RGB+ToF sensors now available on modern smartphones.

1 Introduction

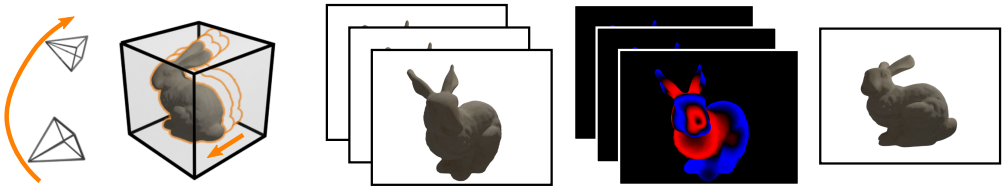
Novel-view synthesis (NVS) is a long-standing problem in computer graphics and computer vision, where the objective is to photorealistically render images of a scene from novel viewpoints. Given a number of images taken from different viewpoints, it is possible to infer both the geometry and appearance of a scene, and then use this information to synthesize images at novel camera poses. One of the challenges associated with NVS is that it requires a diverse set of images from different perspectives to accurately represent the scene. This might involve moving a single camera around a static environment [4, 12, 25, 26, 30], or using a large multi-camera system to capture dynamic events from different perspectives [2, 6, 19, 32, 38, 50]. Techniques for dynamic NVS from a monocular video sequence have also demonstrated compelling results, though they suffer from various visual artifacts due to the ill-posed nature of this problem [21, 31, 36, 43, 46]. This requires introducing priors, often deep learned, on the dynamic scene's depth and motion.

In parallel, mobile devices now have camera systems with both color and depth sensors, including Microsoft's Kinect and HoloLens devices, and the front and rear RGBD camera systems in the iPhone and iPad Pro. Depth sensors can use stereo or structured light, or increasingly the more accurate time-of-flight principle for measurement. Although depth sensing technology is more common than ever, many NVS techniques currently do not exploit this additional source of visual information.

To improve NVS performance, we propose TöRF[1], an implicit neural representation for scene appearance that leverages both color and time-of-flight (ToF) images, as depicted in Figure 1. This reduces the number of images required for static NVS problem settings, compared with just using a color camera. Further, the additional depth information makes the monocular dynamic NVS problem more tractable, as it directly encodes information about the geometry of the scene. Most importantly,

[1] TöRF = ToF + NeRF. Pronounced just like 'NeRF' but starts with a 'T'.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) Acquisition process (b) Color images (c) Phasor images (d) Novel Image

Figure 1: *Illustration of Time-of-Flight Radiance Fields.* **(a)** We move a handheld imaging system around a dynamic scene, capturing **(b)** color images and **(c)** raw phasor images from a continuous-wave time-of-flight (C-ToF) camera. **(d)** Then, we optimize for a continuous neural radiance field of the scene that predicts the captured color and phasor images. This allows novel view synthesis.

rather than using depth directly, we show that using 'raw' ToF data—in the form of phasor images [10] that are normally used to derive depth—is more accurate as it allows the optimization to consider cases out of depth range, with low reflectance, or with multiple single-scattering events, leading to better dynamic scene view synthesis. The contributions of our work include:

- A physics-based neural volume rendering model of the raw measurements formed by a continuous-wave ToF sensor;
- A method to optimize a neural radiance field of dynamic scenes with information from color and continuous-wave ToF sensors;
- Quantitative and qualitative evaluation on synthetic and real scenes showing better view synthesis than NeRF [26] for few input views, and than two dynamic scene baselines [21, 46].

2 Related Work

While NVS is a long-standing problem in computer graphics and vision [7–9, 18], recent developments in neural scene representations for NVS have enabled state-of-the-art results for a wide variety of settings [42]. The common thread across many of these works is to bring learnable elements together with physics-based models and classical rendering processes.

The designs for neural scene representations often build on standard computer graphics data structures, including voxel grids [28, 40], multiplane images (MPIs) [25, 44, 47, 52], multi-sphere images (MSIs) [3, 6], point clouds [24], and even implicit functions of scene geometry and appearance [26, 41, 48]. For example, DeepVoxels [40] represent a scene as a discrete volume of embedded features to encode view-dependent effects, and enable wide baselines that may not be possible with other representations; however, the cost associated with this volumetric representation is that the memory requirements scale cubically with resolution. Alternatively, MPIs can be used to encode scene appearance from a single stereo pair [52]; the key benefit of this representation is the fast rendering speeds (ideal for interactive VR applications), though it performs best for forward-facing scenes.

Implicit neural representations of a scene provide similar flexibility to voxel grids, but circumvent the high memory requirements. These implicit networks therefore have the capacity to better represent the appearance of a scene. For example, scene representation networks (SRNs) [41] encode the geometry in a single neural network, which takes 3D points as input and outputs a feature representation of local scene properties (e.g., surface color or reflectance); rendering an image requires a differentiable ray marching procedure, that intersects rays with the implicit volume. Neural radiance fields (NeRF) [26] encode 5D radiance fields (3D position with 2D viewing direction), to offer higher-fidelity geometry and visual appearance. While these implicit neural representations initially assumed a static scene, recent approaches also demonstrate the ability to perform dynamic NVS from monocular video [21, 31, 36, 43, 46], despite this being a highly ill-posed problem.

The inclusion of depth maps has proven beneficial for improving NVS results for a long time [35]. Surprisingly few NVS methods exploit the availability of depth sensors today. One reason is that explicitly reconstructing depth maps for NVS [23, 49] may prove problematic for certain scenes (e.g., thin structures, depth edges, complex reflectance, noisy depth). In this work, we circumvent these issues by proposing a new implicit neural representation that models raw ToF data and offers better view synthesis results for both static and dynamic scenes.

3 Neural Volume Rendering of ToF images

A neural radiance field (NeRF) [26] is a neural network optimized to synthesize novel views. Assuming a static scene, the neural network $F_{\boldsymbol{\theta}} : (\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o}) \to (\sigma(\mathbf{x}_t), L_s(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o}))$ takes as input a position $\mathbf{x}_t$ and a direction $\boldsymbol{\omega}_\mathrm{o}$, and outputs both the density $\sigma(\mathbf{x}_t)$ at point $\mathbf{x}_t$ and the radiance $L_\mathrm{s}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o})$ of a light ray passing through $\mathbf{x}_t$ in direction $\boldsymbol{\omega}_\mathrm{o}$. After optimizing the network to predict a set of input images, the network can be used to generate images from novel perspectives. The key insight of our work is that NeRFs can also be extended to model (and learn from) the raw images of a ToF camera.

The volume density function $\sigma(\mathbf{x}_t)$ controls the opacity at every point—large values of $\sigma(\mathbf{x}_t)$ represent opaque regions and small values represent transparent ones, which allows representation of 3D structures. The radiance function $L_\mathrm{s}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o})$ represents the light scattered by a surface at point $\mathbf{x}_t$ in direction $\boldsymbol{\omega}_\mathrm{o}$, and characterizes the visual appearance of different materials (e.g., shiny or matte). Together, these two functions can be used to render images of a scene from any given camera pose.

Neural volume rendering is the procedure for generating an image from these two functions. Given the pose of a camera, the procedure generates an image by tracing rays through the volume, and computing the radiance observed along each ray by ray marching the following integral [26]:

$$L(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}) = \int_{t_\mathrm{n}}^{t_\mathrm{f}} T_\mathrm{r}(\mathbf{x}, \mathbf{x}_t)\sigma(\mathbf{x}_t)L_\mathrm{s}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o})\, dt, \quad \text{where} \quad T_\mathrm{r}(\mathbf{x}, \mathbf{x}_t) = e^{-\int_{t_\mathrm{n}}^{t} \sigma(\mathbf{x}-\boldsymbol{\omega}_\mathrm{o}s)\, ds} \tag{1}$$

describes the transmittance for light propagating from position $\mathbf{x}$ to $\mathbf{x}_t = \mathbf{x} - \boldsymbol{\omega}_\mathrm{o}t$ for $t \in [t_\mathrm{n}, t_\mathrm{f}]$. In practice, this integral is evaluated using quadrature [26]:

$$L(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}) \approx \sum_{k=1}^{N} \hat{T}_\mathrm{r}(\mathbf{x}, \mathbf{x}_k)(1 - e^{-\sigma(\mathbf{x}_k)\Delta\mathbf{x}_k})L_\mathrm{s}(\mathbf{x}_k, \boldsymbol{\omega}_\mathrm{o}), \quad \text{where} \quad \hat{T}_\mathrm{r}(\mathbf{x}, \mathbf{x}_k) = \prod_{j=1}^{k-1} e^{-\sigma(\mathbf{x}_j)\Delta\mathbf{x}_j} \tag{2}$$

The value for $\Delta\mathbf{x}_j = \|\mathbf{x}_{j+1} - \mathbf{x}_j\|$ is the distance between two quadrature points.

Generalizing the neural volume rendering procedure for ToF cameras requires two changes. First, because ToF cameras use an active light source to illuminate the scene, we must consider the fact that the lighting conditions of the scene changes with the position of the camera. In Section 3.1, we derive the scene's appearance in response to collocating a point light source with a camera, which follows a similar derivation to that of Bi et al. [5]. Second, in Section 3.2, we extend the typical volume rendering integral to model images captured with a ToF camera. Similar to the approach taken in transient rendering frameworks [13, 33], we incorporate a path length importance function into our integral, which can model different types of ToF cameras.

To keep notation simple, our discussion assumes that the function $L(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o})$ is monochromatic, i.e., it outputs radiance at a single wavelength. In practice, the function outputs values for red, green, blue, and infrared light. The red, green, and blue values correspond to the three channels of a color camera, and the infrared value corresponds to the measurements made by a ToF camera.

3.1. Collocated Point Light Source. An ideal ToF camera responds only to the light from a collocated point source. We therefore model radiance $L_\mathrm{s}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o})$ as a function of the source position [5]:

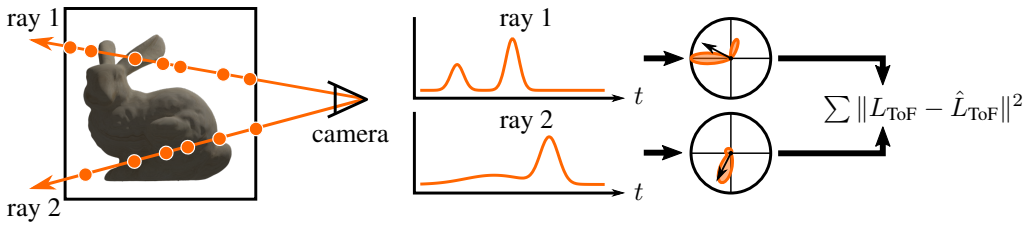
$$L_\mathrm{s}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o}) = \int_{\mathcal{S}^2} f_\mathrm{p}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}, \mathbf{n}(\mathbf{x}_t))L_\mathrm{i}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i})\, d\boldsymbol{\omega}_\mathrm{i}, \tag{3}$$

where the function $L_\mathrm{i}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i})$ represents the incident illumination, and the scattering phase function $f_\mathrm{p}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}, \mathbf{n}(\mathbf{x}_t))$ describes the light scattered at a point $\mathbf{x}_t$ in the volume. Note that the scattering phase function also depends on the local surface shading normal $\mathbf{n}(\mathbf{x}_t)$. For a point light source with center $\mathbf{x}$ (i.e., collocated with the camera), the incident radiance $L_\mathrm{i}(\mathbf{x}_t)$ is given by

$$L_\mathrm{i}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i}) = \frac{I}{\|\mathbf{x}_t - \mathbf{x}\|^2}\delta\left(\frac{\mathbf{x}_t - \mathbf{x}}{\|\mathbf{x}_t - \mathbf{x}\|} - \boldsymbol{\omega}_\mathrm{i}\right)T_\mathrm{r}(\mathbf{x}, \mathbf{x}_t), \tag{4}$$

where the scalar I represents the emitted radiant intensity of the source, and $\delta(\cdot)$ is the Dirac delta distribution. This model ignores forward scattering, which is reasonable if the scene consists mostly of completely opaque surfaces. When combined with Equation 1, the resulting forward model is

$$L(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}) = \int_{t_\mathrm{n}}^{t_\mathrm{f}} \frac{T_\mathrm{r}(\mathbf{x}, \mathbf{x}_t)^2}{\|\mathbf{x}_t - \mathbf{x}\|^2}\sigma(\mathbf{x}_t)L_\mathrm{r}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o})\, dt \quad \text{where} \quad L_\mathrm{r}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{o}) = f_\mathrm{p}(\mathbf{x}_t, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}, \mathbf{n}(\mathbf{x}_t))I. \tag{5}$$



(a) Ray marching (b) Integrand of Eq. (5) (c) Phasor (d) Loss function

Figure 2: *Neural volume rendering of C-ToF images.* **(a)** We start with ray marching to evaluate the radiance and opacity at different points along the ray. **(b)** These samples represent the continuous integrand of Equation 5, which describes the contribution of every point $\mathbf{x}_t$ along the ray. For example, ray 1 grazes both ears of the bunny, producing two distinct responses. **(c)** As described by Equation 6, we multiply the integrand with a complex exponential, with the result represented here in the complex plane. Integrating this result produces a single complex phasor, represented here by a vector with a magnitude corresponding to reflectance and a direction corresponding to the phase (or distance). **(d)** The loss function compares rendered phasors with the raw measurements of a C-ToF camera.

This expression is similar to Equation 1 with two key differences: the squared transmittance term, and the inverse square falloff induced by the point light source. Similar to NeRF [26], we can once again numerically approximate the above integral using quadrature, and recover the volume parameters $(\sigma(\mathbf{x}_t), L_{\text{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\text{o}}))$ by training a neural network which depends only on position and direction.

3.2. Continuous-Wave ToF Model. ToF cameras use the travel time of light to compute distances [11]. The collocated point light source sends an artificial light signal into an environment, and a ToF sensor measures the time required for light to reflect back in response. Given the constant speed of light $\text{c} = 3 \cdot 10^8$ m/s, this temporal information determines the distance traveled. These devices have found widespread applications from autonomous vehicles [20] to mobile AR applications [15, 16].

Photorealistic simulations of ToF cameras involve introducing a path length importance function to the rendering equation [13, 33], and can be just as easily applied to the volume integral in Equation 5:

$$L_{\text{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}}) = \int_{t_{\text{n}}}^{t_{\text{f}}} \frac{T_{\text{r}}(\mathbf{x}, \mathbf{x}_t)^2}{\|\mathbf{x}_t - \mathbf{x}\|^2} \sigma(\mathbf{x}_t) L_{\text{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\text{o}}) W(2\|\mathbf{x}_t - \mathbf{x}\|)\, dt, \tag{6}$$

where the function $W(d)$ weights the contribution of a light path of length d. Note that light travels twice the distance between the camera's origin $\mathbf{x}$ and the scene point $\mathbf{x}_t$. As described by Pediredla et al. [33], the function $W(d)$ can be used to represent a wide variety of ToF cameras, including both pulsed ToF sensors [14] and continuous-wave ToF (C-ToF) sensors [10, 15, 29]. Here, as our proposed system uses a C-ToF sensor for imaging, the images are modeled using the phasor $W(d) = \exp\left(\text{i}\frac{2\pi d f}{\text{c}}\right)$, where f is the modulation frequency of the signal emitted by the C-ToF camera. Note that, because the function $W(d)$ is complex-valued, the radiance $L_{\text{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}})$ will also produce a complex-valued phasor image [10]. In practice, these images are typically created by capturing four real-valued images that are linearly combined to produce the phasor image; please see the supplement for additional details. In Figure 1c, we show the real component component of the phasor image, with positive pixel values as red and negative values as blue.

ToF cameras recover depth by assuming only one point $\mathbf{x}_{\text{s}}$ reflects light for every ray, i.e., the integrand of Equation 6 is assumed to be zero for all other points $\mathbf{x} \neq \mathbf{x}_{\text{s}}$ (points in front of $\mathbf{x}_{\text{s}}$ reflect no light, and point behind $\mathbf{x}_s$ are hidden). Equation 6 thus simplifies to the phasor:

$$L_{\text{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}}) = a \cdot W(2\|\mathbf{x}_s - \mathbf{x}\|) = a \cdot \exp\left(\text{i}\frac{4\pi f}{\text{c}}\|\mathbf{x}_s - \mathbf{x}\|\right), \tag{7}$$

where the phasor's magnitude, $|L_{\text{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}})| = a$, represents the amount of light reflected by this single point, and the phase, $\angle L_{\text{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}}) = \frac{4\pi f}{\text{c}}\|\mathbf{x}_s - \mathbf{x}\| \bmod 2\pi$, is related to the distance $\|\mathbf{x}_s - \mathbf{x}\|$.

In real-world scenarios, it is also possible for multiple points along a ray to contribute to the signal, resulting in a linear combination of phasor radiance values—known as multi-path interference. Typically, this degrades the quality of depth measurements for a C-ToF camera. For example, around depth edges, a pixel integrates the signal from surfaces at two different distances from the camera

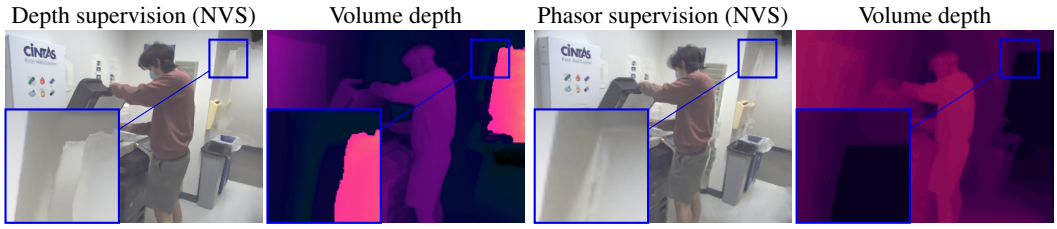


Figure 3: *Raw phasor supervision improves reconstruction.* Wrap-around phase bounds the range of useful ToF measurements when depth is derived first and used as supervision (left). Our approach of modeling raw phasor measurements within the neural volume alleviates this problem (right).

(e.g., Figure 2), resulting in 'flying pixel' artifacts [37] (i.e., 3D points not corresponding to either distance). Similar artifacts occur when imaging semi-transparent or specular objects, where two or more surfaces contribute light to a pixel.

Optimizing NeRFs with phasor images therefore has three distinct advantages over using processed depth maps. (i) For ranges that span values larger than $\frac{c}{2f}$, the true range is ambiguous, as there are multiple depth values that produce the same phase. For example, a typical modulation frequency of $f = 30\,\mathrm{MHz}$ for a C-ToF camera corresponds to an unambigous range of $\frac{c}{2f} \approx 5\,\mathrm{m}$. By modeling the phasor images directly, we avoid the issues associated with recovering depth images for scenes that exceeds this range (see Figure 3). (ii) Depth values become unreliable (noisy) when the amount of light reflected to the camera is small. Modeling the phasor images directly makes the solution robust to sensor noise. (iii) For scenes with complicated reflectance properties, the light detected may not travel along a single path; this results in mixtures of phasors, which can result in phase values that do not correspond to a single depth. This can occur near depth edges as well as for specular reflections and transparent surfaces. Equation 6 models the response from multiple single-scattering events along a ray, and provides us with a better handle over such scenarios.

4 Optimizing Dynamic ToF + NeRF = TöRF

4.1. Dynamic Neural Radiance Fields. One key advantage of working with phasor images is that it can capture scene geometry in a single measurement, which enables higher-fidelity novel-view synthesis of dynamic scenes from a potentially moving color camera and C-ToF camera pair. To support dynamic neural radiance fields, we model the measurements with two neural networks. The first, static network $F^{\mathrm{s}}_{\boldsymbol{\theta}} : (\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}) \rightarrow (\sigma^{\mathrm{s}}(\mathbf{x}_t), L^{\mathrm{s}}_{\mathrm{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}))$ is a 5D function of position and direction, while the second, dynamic network $F^{\mathrm{d}}_{\boldsymbol{\theta}} : (\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau) \rightarrow (\sigma^{\mathrm{d}}(\mathbf{x}_t, \tau), L^{\mathrm{d}}_{\mathrm{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau))$ is a 6D function of position, direction, and time τ. Instead of directly consuming a time τ directly, the dynamic network receives a latent code $\mathbf{z}_\tau$ which is optimized per frame, similar to Li et al. [19]. Following the approach of Li et al. [21], we blend the outputs of the static and dynamic networks to produce opacity and radiance values to pass into our image formation models:

$$L_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{f}}} T^{\mathrm{blend}}_{\mathrm{r}}(\mathbf{x}, \mathbf{x}_t, \tau) \sigma^{\mathrm{blend}}(\mathbf{x}_t, \tau) L^{\mathrm{blend}}_{\mathrm{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau)\, dt \tag{8}$$

$$L_{\mathrm{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{f}}} \frac{T^{\mathrm{blend}}_{\mathrm{r}}(\mathbf{x}, \mathbf{x}_t, \tau)^2}{\|\mathbf{x}_t - \mathbf{x}\|^2} \sigma^{\mathrm{blend}}(\mathbf{x}_t, \tau) L^{\mathrm{blend}}_{\mathrm{r}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau) W(2\,\|\mathbf{x}_t - \mathbf{x}\|)\, dt, \tag{9}$$

where the terms of these equations are the result of this blending operation with a learned blending weight; see the supplemental document for an explicit definition of the blending terms.

4.2. Loss Function. Given a set of color images and phasor images captured of a scene at different time instances, we sample a set of camera rays from the set of all pixels, and minimize the following total squared error between the rendered images and measured pixel values:

$$\mathcal{L} = \sum_{(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau)} \|L_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau) - \hat{L}_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau)\|^2 + \lambda \|L_{\mathrm{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau) - \hat{L}_{\mathrm{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau)\|^2, \tag{10}$$

where the scalar $\lambda \geq 0$ controls the relative contribution of both loss terms, $\hat{L}_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau)$ represents the measurements of a color camera, and $\hat{L}_{\mathrm{ToF}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau)$ represents the phasor measurements of a C-ToF camera. At training time, we reduce the weight λ in later iterations to prioritize the color loss.

4.3. Camera Pose Optimization. In past works, COLMAP [39] has been used to recover camera poses for NVS. However, COLMAP fails to recover accurate camera poses for many real scenes even if we masked dynamic regions. Further, COLMAP only recovers camera poses up to an unknown scale factor, whereas our ToF image formation model assumes a known scene scale. As such, for real-world scenes, we optimize camera poses from scratch within the training loop. First, we optimize the weights of the static neural network $F_{\boldsymbol{\theta}}^{\mathrm{s}}$, as well the camera poses for each video frame and the relative rotation and translation between the color and C-ToF sensor, with a learning rate of 10^{-3}. After 5000 iterations, we decrease the pose learning rate to $5 \cdot 10^{-4}$, and optimize our full model.

4.4. Ray Sampling. Many physical camera systems do not have collocated color and ToF cameras. As such, to train our model, we must trace separate rays through the volume for color and ToF. We alternate using the color loss and the ToF loss for every iteration. Further, like NeRF [26], we use stratified random sampling when sampling points along a ray.

5 Experiments

5.1. Hardware. We use an iDS UI-3070CP-C-HQ machine vision color camera (640×480 @ 30 fps), and a Texas Instruments OPT8241 sensor to provide phasor measurements (320×240 @ 30 fps) with an unambiguous range of 5 m. Both cameras are mounted with a baseline of 41 mm. We use OpenCV to calibrate intrinsics, extrinsics, and lens distortion. See the supplement for details.

For optimization, we use an NVIDIA GeForce RTX 2080 Ti with 11 GB RAM. Our model takes 12–24 hours to converge, and 3–5 seconds per frame to generate a novel view (256×256).

5.2. Data. We captured *PhoneBooth*, *Cupboard*, *Photocopier*, and *DeskBox* sequences with our handheld camera setup. Each is indoors in an office with a person performing a dynamic action, and includes view dependence from real-world materials. *PhoneBooth* includes multi-path interference effects from a glass door, and *Photocopier* includes wrap-around phase effects in the distance. For comparison, we also captured the *Dishwasher* sequence on an iPhone 12 Pro, which uses a ToF sensor to capture depth (raw measurements are not available). Finally, we create synthetic raw C-ToF sequences *Bathroom* and *Bedroom* by adapting the physically-based renderer PBRT [34] to generate phasor images with multi-bounce and scattering effects via path tracing (rendered at 512×512 pixels).

5.3. Metrics. For synthetic scenes with ground-truth hold-out views, we use PSNR and SSIM [45] metrics and the recent neural LPIPS measure [51]. For real-world sequences, capturing hold-out reference views was not practically possible as we only had one hardware setup; instead, we assess novel view quality numerically using the opinion-aware no-reference metric BRISQUE [27]. This indicates the noise and blur in an image (lower scores are better), and complements our qualitative supplemental videos. We use the BRISQUE implementation and model available in OpenCV contrib.

5.4. Few-View Reconstruction of Static Scenes. We demonstrate that integrating raw ToF measurements in addition to RGB enables TöRF to reconstruct static scenes from fewer input views, and to achieve higher visual fidelity and depth accuracy than standard NeRF [26] for the same number of input views. Table 1 contains a quantitative comparison on two synthetic sequences, *Bathroom* and *Bedroom*, for reconstructions from just 2 and 4 input views. To enable the comparison on 10 hold-out views, we use ground-truth camera poses for both methods. With just two input views, TöRF's added phasor supervision better reproduces the scene than NeRF, as one might expect. This closely resembles a camera system that might exist on a smartphone, and shows the potential value of ToF supervision for dynamic scenes if we consider a static scene as one time step of an animation. For four views, NeRF and TöRF produce comparable RGB results, though our depth reconstructions are significantly more accurate (Figure 4).

5.5. Real Dynamic Scenes. We show results and comparisons on our real-world sequences in Figure 5, and compare them quantitatively in Table 2 using the BRISQUE metric [27]. VideoNeRF [46] shows stronger disocclusion artifacts and warped edges near depth boundaries, and cannot recover from depth maps with wrapped range. NSFF's results [21] suffer from severe ghosting and stretching artifacts that negatively impact the quality of results. Our results show the highest visual quality and most accurate depth maps. The quantitative evaluation in Table 2 using the BRISQUE metric [27] supports this general trend. Please see our supplemental video for in-motion NVS.

Table 1: *Phasor supervision aids few-view reconstruction.* On two input views, TöRF significantly outperforms NeRF on both synthetic static scenes; RGB results are on-par for four views but our disparity estimates are qualitatively more accurate (Figure 4). Metrics computed on 10 hold-out views. Each cell contains disparity MSE or RGB image similarity measures. *Note: Metric error in the bathroom scene is affected by the large mirror, which has depth at the first surface intersection of the mirror.*

		Bathroom				*Bedroom*			
Views	Method	MSE (D) ▼	PSNR ▲	SSIM ▲	LPIPS ▼	MSE (D) ▼	PSNR ▲	SSIM ▲	LPIPS ▼
2	NeRF [26]	1.11	13.41	0.333	0.046	1.28	11.86	0.280	0.053
	TöRF (ours)	0.48	23.38	0.628	0.014	0.07	18.54	0.516	0.018
4	NeRF [26]	0.47	21.59	0.571	0.016	1.09	25.10	0.731	0.004
	TöRF (ours)	0.46	22.52	0.603	0.012	1.10	22.03	0.634	0.017

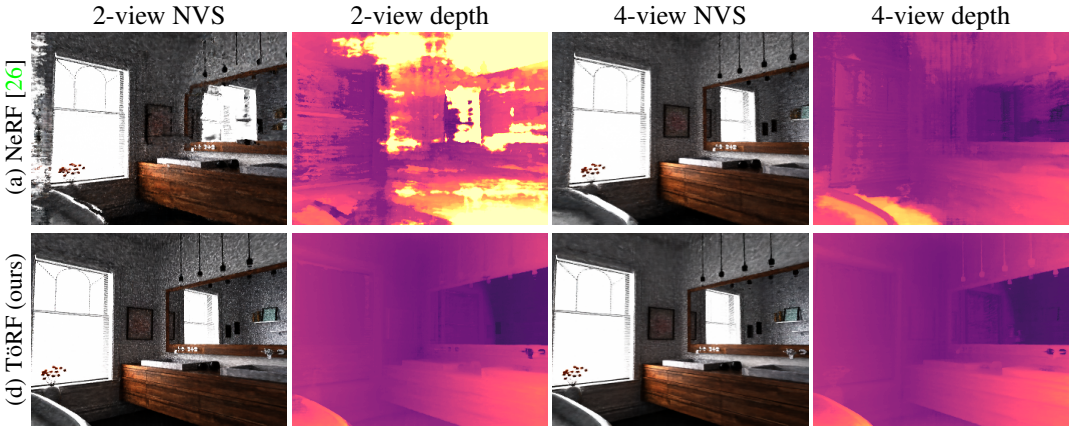


Figure 4: *Adding ToF aids reconstruction for low numbers of views.* In the classic multi-view static setting, NeRF quality suffers as the number of views decreases. For two RGB views, adding ToF data makes view synthesis possible. For four RGB views, ToF data increases depth quality over NeRF. Note the clean depth with sharp edges and fine geometric details such as the lamps above the mirror.

Table 2: *Adding ToF data provides higher quality view synthesis*. We compare our method to two neural dynamic reconstruction methods using the BRISQUE no-reference image quality metric [27]. †*Dishwasher* uses processed LIDAR depth from an iPhone 12 Pro as a comparison sequence instead of ToF phasor image supervision. *Note:* As an opinion aware no-reference metric, these numbers are more suggestive than PSNR, and we recommend viewing our supplemental video in addition.

Method / BRISQUE ▼	*PhoneBooth*	*Cupboard*	*Photocopier*	*DeskBox*	*Dishwasher*†
VideoNeRF [46]	48.7 ± 0.6	48.4 ± 1.5	41.6 ± 0.7	49.4 ± 0.6	31.2 ± 0.8
NSFF [21]	42.2 ± 1.3	32.7 ± 1.5	42.0 ± 2.3	46.4 ± 2.4	21.9 ± 3.3
TöRF (ours)	28.4 ± 1.2	25.7 ± 1.4	28.6 ± 2.5	27.9 ± 2.3	25.8 ± 1.9

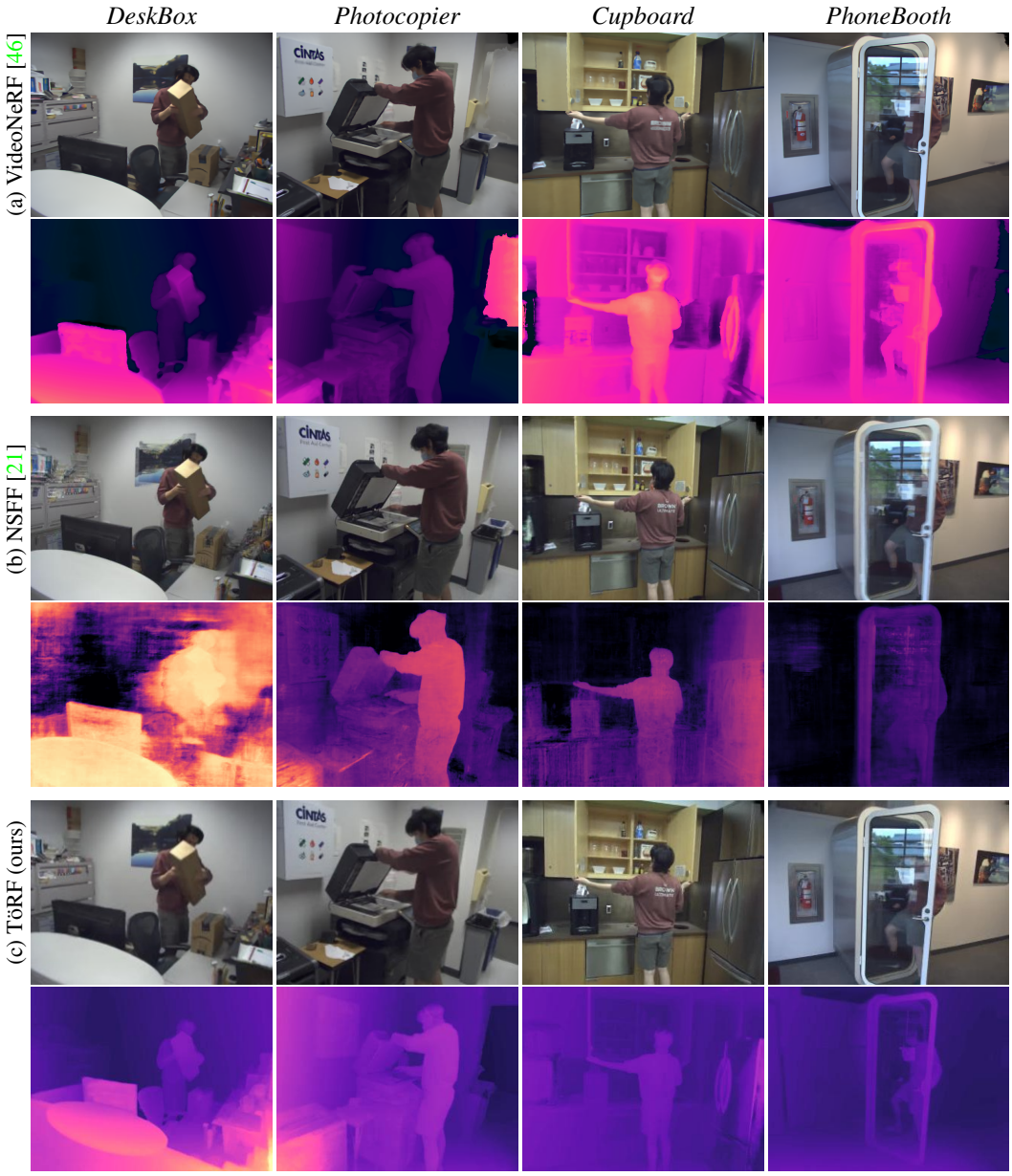


Figure 5: *Adding ToF supervision helps improve quality.* Compared to two video baselines, Video-NeRF [46] and NSFF [21], our approach reduces errors in dynamic scene components and reflections on the glass door. All models were trained for comparable times. Please see our supplemental video for additional comparisons in motion.

6 Discussion

6.1. Limitations. Introducing ToF sensors into RGB neural radiance fields aims to improve quality by merging the benefits of both sensing modes; but, some limitations are also brought in. C-ToF sensing can struggle on larger-scale scenes; using multiple different modulation frequencies can, however, extend the unambiguous range [10]. While C-ToF sensors typically struggle outdoors, EpiToF [1] has demonstrated the ability to perform 15 m ranging under strong ambient illumination. Further, for each measurement, C-ToF sensors require capturing four or more images quickly at different times, which can cause artifacts for fast-moving objects.

Even with ToF data, objects imaged at grazing angles or objects that both are both dark (low reflectance) and dynamic remain difficult to reconstruct, e.g., black hair in our scenes (Figure 6). Further, neural networks have limited capacity to model dynamic scenes, resulting in relatively short dynamic sequences. This is a limitation of many current neural dynamic scene methods.

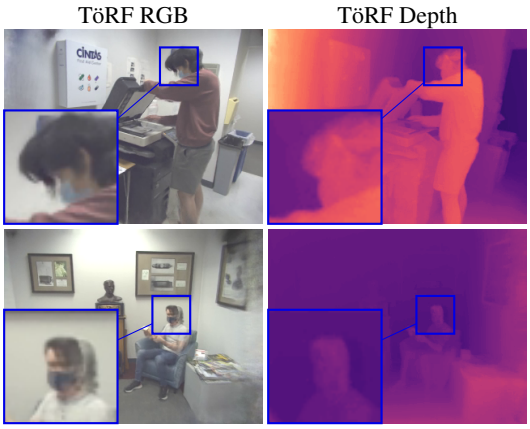


Figure 6: RGB and depth NVS show that dark and dynamic objects pose difficulties. Black hair incorrectly extends in front or to the side of the face due to failure to reconstruct dynamic motion. *Brightened for visibility.*

6.2. Potential Social Impact. Scene reconstruction and view synthesis are core problems in visual computing that attempt to determine the shape and appearance of objects and scenes. Neural approaches to these tasks hold promise to increase accuracy and fidelity. At the methodological level, integrating ToF data, as we propose, improves accuracy, but restricts use to scenarios where active illumination is detectable. Knowing the shape and appearance of objects is useful for many applications; negative impact may include synthesizing images from perspectives or time instances that were never captured (falsifying media), extending surveillance through higher-fidelity reconstructions, remote capture of forms for copying physical objects or 'ripping-off' designs (security, industrial espionage), or providing new sensory experiences that distract people from their real-world obligations (VR rabbithole).

Practically, current neural approaches are more computationally expensive in both optimization and rendering than classic image-based rendering. Our work required GPUs to optimize for many hours (12–24 h). Without renewable energy sources, this use will generate CO_2 emissions, requiring 1.5–3 kg CO_2-equivalents per scene for optimization and 0.01–0.02 kg CO_2-equivalents per sequence for rendering (numbers generated by ML CO_2 Impact [17]). Concurrent work in neural radiance fields reduces this cost using caching, spatial acceleration structures, and more efficient parameterizations, and real-world deployment should exploit these approaches to reduce the CO_2 emission impact.

7 Conclusion

Modern camera systems integrate multiple modes of sensing, and our reconstruction methods should exploit this information to improve quality. To this end, we formulate a neural model for time-of-flight radiance fields based on physical RGB+ToF image formation. We demonstrate an optimization method to recover ToRF volumes, and show that it improves novel-view synthesis for few-view scenes and especially for dynamic scenes. Further, we demonstrate that using raw ToF phasor supervision leads to better performance than using derived depth directly, allowing both sensing modes to help resolve errors, limitations, and ambiguities. Future work may extend the combination of additional sensors into neural radiance fields, e.g., dynamic vision sensors [22] or *event cameras* may be used to measure scenes at higher speeds. Further, a collocated point light source has been shown to be able to render photos of scenes under non-collocated illumination conditions [5]. As a result, we believe ToF images may also serve to support relighting applications.

References

[1] Supreeth Achar, Joseph R. Bartels, William L. 'Red' Whittaker, Kiriakos N. Kutulakos, and Srinivasa G. Narasimhan. Epipolar time-of-flight imaging. *ACM Trans. Graph.*, 36(4):37:1–8, 2017. doi:10.1145/3072959.3073686.

[2] Robert Anderson, David Gallup, Jonathan T. Barron, Janne Kontkanen, Noah Snavely, Carlos Hernandez, Sameer Agarwal, and Steven M. Seitz. Jump: Virtual reality video. *ACM Trans. Graph.*, 35(6):198:1–13, 2016. doi:10.1145/2980179.2980257.

[3] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *ECCV*, 2020. doi:10.1007/978-3-030-58452-8_26.

[4] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. OmniPhotos: Casual 360° VR photography. *ACM Trans. Graph.*, 39(6):267:1–12, 2020. doi:10.1145/3414685.3417770.

[5] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. arXiv:2008.03824, 2020.

[6] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Trans. Graph.*, 39(4):86:1–15, 2020. doi:10.1145/3386569.3392485.

[7] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432, 2001. doi:10.1145/383259.383309.

[8] Paul Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Proceedings of the Eurographics Workshop on Rendering*, pages 105–116, 1998. doi:10.1007/978-3-7091-6453-2_10.

[9] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, 1996. doi:10.1145/237170.237200.

[10] Mohit Gupta, Shree K. Nayar, Matthias B. Hullin, and Jaime Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Trans. Graph.*, 34(5):156:1–18, 2015. doi:10.1145/2735702.

[11] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Patrice Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer, 2012. doi:10.1007/978-1-4471-4658-2.

[12] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3D photography. *ACM Trans. Graph.*, 36(6):234:1–15, 2017. doi:10.1145/3130800.3130828.

[13] Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Trans. Graph.*, 33(6):177:1–10, 2014. doi:10.1145/2661229.2661251.

[14] Walter Koechner. Optical ranging system employing a high power injection laser diode. *IEEE Transactions on Aerospace and Electronic Systems*, 4(1):81–91, 1968. doi:10.1109/TAES.1968.5408936.

[15] Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-flight cameras in computer graphics. *Comput. Graph. Forum*, 29(1):141–159, 2010. doi:10.1111/j.1467-8659.2009.01583.x.

[16] George Alex Koulieris, Kaan Akşit, Michael Stengel, Rafał K. Mantiuk, Katerina Mania, and Christian Richardt. Near-eye display and tracking technologies for virtual and augmented reality. *Comput. Graph. Forum*, 38(2):493–519, 2019. doi:10.1111/cgf.13654.

[17] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. arXiv:1910.09700, 2019.

[18] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42, 1996. doi:10.1145/237170.237199.

[19] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3D video synthesis. arXiv:2103.02597, 2021.

[20] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020. doi:10.1109/MSP.2020.2973615.

[21] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[22] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. doi:10.1109/JSSC.2007.914337.

[23] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Trans. Graph.*, 39(4):71:1–13, 2020. doi:10.1145/3386569.3392377.

[24] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *CVPR*, 2019. doi:10.1109/CVPR.2019.00704.

[25] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):29:1–14, 2019. doi:10.1145/3306346.3322980.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. doi:10.1007/978-3-030-58452-8_24.

[27] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. doi:10.1109/TIP.2012.2214050.

[28] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *ICCV*, 2019. doi:10.1109/ICCV.2019.00768.

[29] Matthew O'Toole, Felix Heide, Lei Xiao, Matthias B. Hullin, Wolfgang Heidrich, and Kiriakos N. Kutulakos. Temporal frequency probing for 5D transient analysis of global light transport. *ACM Trans. Graph.*, 33(4):87:1–11, 2014. doi:10.1145/2601097.2601103.

[30] Ryan Styles Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. A system for acquiring, compressing, and rendering panoramic light field stills for virtual reality. *ACM Trans. Graph.*, 37(6):197:1–15, 2018. doi:10.1145/3272127.3275031.

[31] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo-Martin Brualla. Deformable neural radiance fields. arXiv:2011.12948, 2020.

[32] Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyse Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. An integrated 6DoF video camera and system design. *ACM Trans. Graph.*, 38(6):216:1–16, 2019. doi:10.1145/3355089.3356555.

[33] Adithya Pediredla, Ashok Veeraraghavan, and Ioannis Gkioulekas. Ellipsoidal path connections for time-gated rendering. *ACM Trans. Graph.*, 38(4):38:1–12, 2019. doi:10.1145/3306346.3323016.

[34] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Elsevier Science, 3rd edition, 2016. ISBN 9780128007099. URL `https://www.pbr-book.org/`.

[35] Kari Pulli, Michael F. Cohen, Tom Duchamp, Hugues Hoppe, Linda Shapiro, and Werner Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In *Proceedings of the Eurographics Workshop on Rendering*, pages 23–34, 1997. doi:10.1007/978-3-7091-6858-5_3.

[36] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[37] Malcolm Reynolds, Jozef Doboš, Leto Peel, Tim Weyrich, and Gabriel J Brostow. Capturing time-of-flight data with confidence. In *CVPR*, pages 945–952, 2011. doi:10.1109/CVPR.2011.5995550.

[38] Christopher Schroers, Jean-Charles Bazin, and Alexander Sorkine-Hornung. An omnistereoscopic video pipeline for capture and display of real-world VR. *ACM Trans. Graph.*, 37(3):37:1–13, 2018. doi:10.1145/3225150.

[39] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. doi:10.1109/CVPR.2016.445.

[40] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning persistent 3D feature embeddings. In *CVPR*, pages 2437–2446, 2019. doi:10.1109/CVPR.2019.00254.

[41] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *NeurIPS*, 2019.

[42] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhöfer. State of the art on neural rendering. *Comput. Graph. Forum*, 39(2):701–727, 2020. doi:10.1111/cgf.14022.

[43] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. arXiv:2012.12247, 2020.

[44] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. doi:10.1109/CVPR42600.2020.00063.

[45] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.

[46] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. arXiv:2011.12950, 2020.

[47] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Trans. Graph.*, 38(4):76:1–13, 2019. doi:10.1145/3306346.3323007.

[48] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020.

[49] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020.

[50] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Trans. Graph.*, 2021. doi:10.1145/3450626.3459756.

[51] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. doi:10.1109/CVPR.2018.00068.

[52] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4):65:1–12, 2018. doi:10.1145/3197517.3201323.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   [Yes]
   (b) Did you describe the limitations of your work?
   [Yes] – See Section 6.1.
   (c) Did you discuss any potential negative societal impacts of your work?
   [Yes] – See Section 6.2.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them?
   [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?
   [N/A] – We do not include theoretical results.
   (b) Did you include complete proofs of all theoretical results?
   [N/A] – We do not include theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   [No] – We aim to release materials for reproduction later on.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
   [Yes] – See Sections 4.3 and 4.4 and the supplemental document for details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   [Yes] – In Table 2. Our method does not have random seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?
   [Yes] – See Section 5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators?
   [Yes] – Please refer to Section 'New and Existing Assets' in the supplemental document.
   (b) Did you mention the license of the assets?
   [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL?
   [No] – No new assets are introduced in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?
   [Yes] – Please refer to Section 'New and Existing Assets' in the supplemental document.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?
   [Yes] – Please refer to Section 'New and Existing Assets' in the supplemental document.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable?
   [N/A] – We did not use crowdsourcing or perform experiments with human participants.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?
   [N/A] – We did not perform experiments with human participants.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?
   [N/A] – We did not perform experiments with human participants.