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

Anonymous Author(s)
Affiliation
Address
email

1 Additional Results

We show animated results and comparisons for our sequences in the supplemental results website.

1.1 Static Synthetic Scene

Given the limited space in the main paper, we show additional qualitative results on *Bedroom* in Figure 2. Further, previous quantitative metrics were affected by a data error that has now been corrected; our quantitative metrics in Table 1 now align with the qualitative results.

Table 1: ***(Note to reviewers: We corrected a data error with quantitative numbers for* Bedroom *in the main paper's Table 1; these now align with the qualitative results. Updated numbers are in blue.)*** *Phasor supervision aids few-view reconstruction.* Each cell contains RGB image similarity measures, and metrics are computed on 10 hold-out views. TöRF significantly outperforms NeRF on both synthetic static scenes and produces more accurate depth estimates (Figure 2), particularly from just two input views. *Note that the metric depth error 'MSE (D)' in the Bathroom scene is affected by the large mirror, whose depth is defined by the plane of the mirror, not the reflected scene. The bedroom scene also has smaller mirrors, that also affect depth metrics.*

		Bathroom				*Bedroom*			
Views	Method	MSE (D) ▼	PSNR ▲	SSIM ▲	LPIPS ▼	MSE (D) ▼	PSNR ▲	SSIM ▲	LPIPS ▼
2	NeRF [7]	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	0.30	21.29	0.666	0.012
4	NeRF [7]	0.47	21.59	0.571	0.016	1.09	25.10	0.731	0.009
	TöRF (ours)	0.46	22.52	0.603	0.012	0.34	27.56	0.763	0.006

1.2 Dynamic Synthetic Scene

We compare the quality of view synthesis results on the synthetic dynamic sequence *DinoPear* in Table 2 with 30 ground-truth hold-out views and depth maps. For Video-NeRF, we use the ground truth depth, rather than the ToF-derived depth (an oversight on our part). It should thus be noted that it is not completely fair to compare ToRF performance to Video-NeRF performance in this case.

1.3 Dynamic Scene from iPhone ToF—*Dishwasher*

To evaluate a more practical camera setup than our prototype in Section 4, we captured one real-world sequence with a standard handheld Apple iPhone 12 Pro. This consumer smartphone contains a

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

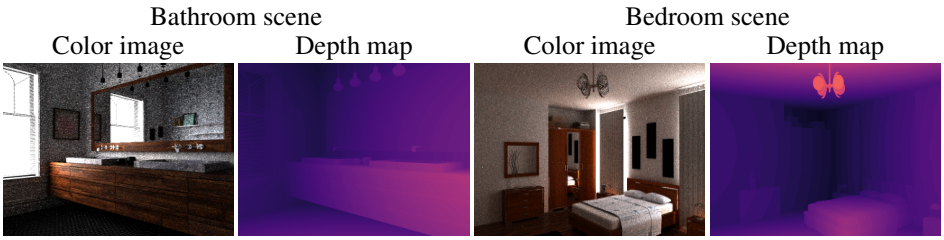


Figure 1: *Color and depth images for static bathroom and bedroom scenes.*

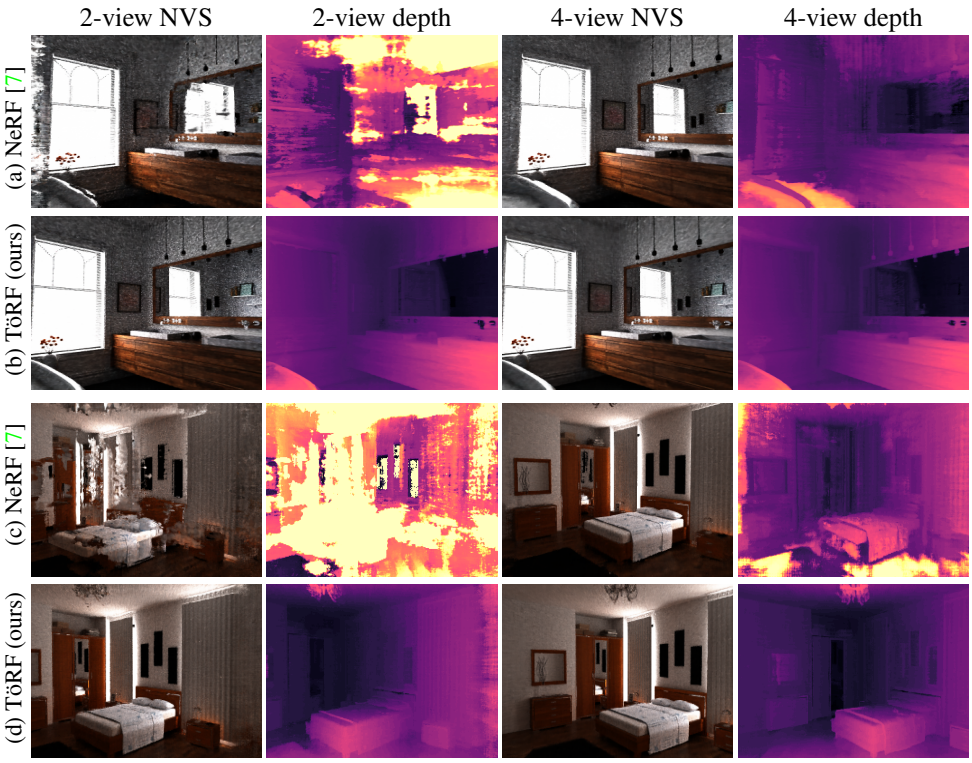


Figure 2: *Adding ToF aids reconstruction for low numbers of views for the bathroom scene (a) & (b) and bedroom scene (c) & (d).* In the classic multi-view static setting, NeRF quality suffers as the number of views decreases. For two RGB views, adding ToF data makes view synthesis possible. For four 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.

LIDAR ToF sensor for measuring sparse metric depth, which is processed by ARKit to provide a dense metric depth map video in addition to a captured RGB color video. Unfortunately, the raw measurements are not available from the ARKit SDK; however, if available, in principle our approach could apply.

Thus, for processing with TöRF, we convert the estimated metric depth maps to synthetic C-ToF sequences by assuming a constant infrared albedo everywhere. In this specific case, the RGB and ToF data are also collocated, as the depth maps are aligned with the color video.

1.4 Analysis of Dynamic Real scenes

The depth images of a C-ToF camera may not accurately represent scene geometry, for several reasons. Three such reasons include (1) the finite unambiguous range resulting of a C-ToF camera results depth wrapping, (2) the depth of specular regions captures the geometry of the reflections, and (3) the depth is noisy for dark objects. By modeling the raw phasor images directly, TöRF becomes

Table 2: Evaluation on ground-truth hold-out views for the dynamic *DinoPear* sequence shows improved RGB and depth results for our method. Note that the VideoNeRF results is given the groundtruth depth of the scene, whereas TöRF uses raw phasor images.

Method	MSE (D) ▼	PSNR ▲	SSIM ▲	LPIPS ▼
VideoNeRF [9]	0.0004 ± 0.0002	26.95 ± 0.95	0.670 ± 0.018	0.017 ± 0.012
NSFF [5]	0.021 ± 0.003	22.64 ± 1.46	0.554 ± 0.029	0.039 ± 0.010
TöRF (ours)	0.005 ± 0.001	22.19 ± 1.75	0.561 ± 0.052	0.028 ± 0.011

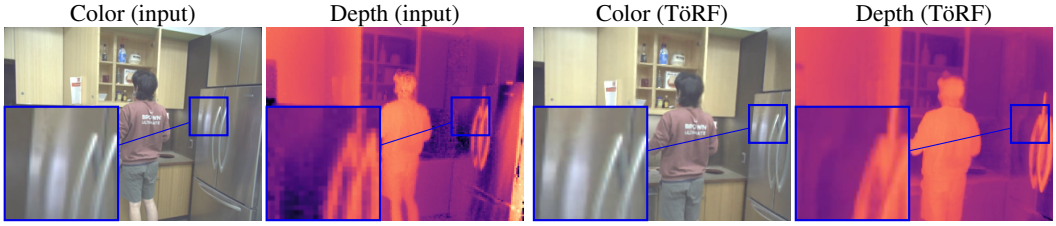


Figure 3: The specular reflections off of the metallic fridge door results in a C-ToF depth image does not accurately reflect the true geometry of the door itself. Instead, the C-ToF camera captures the distance travelled by light reflecting off the fridge door and hitting other objects within the scene. As a result, the reflection of the fridge's door handles appear closer to the camera than the reflection of the cabinets. While not representing the scene's true geometry, we believe that this information guides TöRF to more effectively predict the motion of the reflection from novel viewpoints.

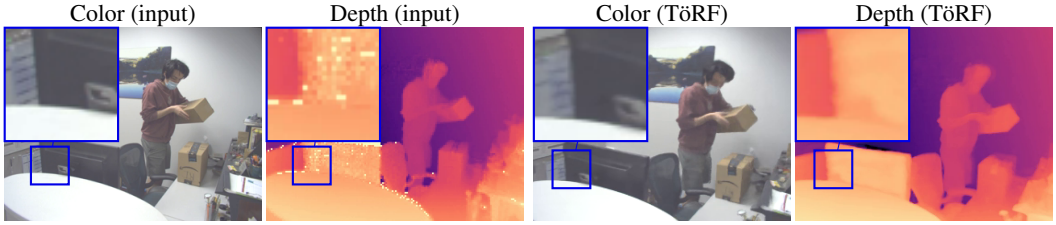


Figure 4: The weak signal reflected back by dark objects (e.g., the computer monitor) results in noisy depth measurements. However, because TöRF does not rely on depth explicitly and instead models the raw phasor image, our recovered depth map is a better representation of the scene geometry when compared to the depth extracted from a raw phasor image.

robust to these problems, as discussed and demonstrated in Figure 3 and Figure 4. For an example of the effect of depth wrapping, please refer to the main paper.

2 Dynamic Scene Image Formation Model

To support dynamic neural radiance fields, we model the measurements with two neural networks. The first, static network $F_{\boldsymbol{\theta}}^{\mathrm{s}} : (\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}) \rightarrow (\sigma^{\mathrm{s}}(\mathbf{x}_t), L_{\mathrm{r}}^{\mathrm{s}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}))$ is a 5D function of position and direction, while the second, dynamic network $F_{\boldsymbol{\theta}}^{\mathrm{d}} : (\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau) \rightarrow (\sigma^{\mathrm{d}}(\mathbf{x}_t, \tau), L_{\mathrm{r}}^{\mathrm{d}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau), b(\mathbf{x}_t, \tau))$ is a 6D function of position, direction, and time τ; the function $b(\mathbf{x}_t, \tau)$ is a temporally-varying blending weight, used to blend the static and dynamic terms. Instead of directly consuming a time τ, the dynamic network receives a latent code $\mathbf{z}_\tau$ which is optimized per frame, similar to Li et al. [4]. Following the approach of Li et al. [5], we blend the outputs of the static and dynamic networks to produce opacity and radiance values to pass into our image formation models:

$$L_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}, \tau) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{f}}} T_{\mathrm{r}}^{\mathrm{blend}}(\mathbf{x}, \mathbf{x}_t, \tau) \sigma^{\mathrm{blend}}(\mathbf{x}_t, \tau) L_{\mathrm{r}}^{\mathrm{blend}}(\mathbf{x}_t, \boldsymbol{\omega}_{\mathrm{o}}, \tau)\, dt \tag{1}$$

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

where the terms of these equations are the result of this blending operation with a learned blending weight.

We next explain how we model dynamic scenes using the RGB case; the ToF case is similar. In practice, we evaluate the integral in Equation 1 using quadrature [7] as follows:

$$L_{\text{RGB}}(\mathbf{x}, \boldsymbol{\omega}_{\text{o}}, \tau) = \sum_{k=0}^{N} \hat{T}_{\text{r}}^{\text{blend}}(\mathbf{x}, \mathbf{x}_k, \tau) \alpha^{\text{blend}}(\mathbf{x}_k, \tau) L_{\text{r}}^{\text{blend}}(\mathbf{x}_k, \boldsymbol{\omega}_{\text{o}}, \tau). \tag{3}$$

Here, $\hat{T}_{\text{r}}^{\text{blend}}$ is the blended transmittance for light propagating from $\mathbf{x}$ to $\mathbf{x}_k = \mathbf{x} - \boldsymbol{\omega}_{\text{o}} k$ at time τ:

$$\hat{T}_{\text{r}}^{\text{blend}}(\mathbf{x}, \mathbf{x}_k, \tau) = \prod_{j=0}^{k-1} \Big(1 - \alpha^{\text{blend}}(\mathbf{x}_k, \tau)\Big), \tag{4}$$

where α^{blend} is the blended opacity at position $\mathbf{x}_k$ and time τ. This blend combines the opacities

$$\alpha^{\text{s}}(\mathbf{x}_k) = 1 - \exp(-\sigma^{\text{s}}(\mathbf{x}_k)\Delta\mathbf{x}_k) \tag{5}$$

$$\alpha^{\text{d}}(\mathbf{x}_k, \tau) = 1 - \exp(-\sigma^{\text{d}}(\mathbf{x}_k, \tau)\Delta\mathbf{x}_k) \tag{6}$$

predicted by the static and dynamic networks, respectively, using the position- and time-dependent blending weight $b(\mathbf{x}_k, \tau)$ that is predicted by the dynamic network $F_{\boldsymbol{\theta}}^{\text{d}}$, as in Gao et al. [1]:

$$\alpha^{\text{blend}}(\mathbf{x}_k, \tau) = (1 - b(\mathbf{x}_k, \tau)) \cdot \alpha^{\text{s}}(\mathbf{x}_k) + b(\mathbf{x}_k, \tau) \cdot \alpha^{\text{d}}(\mathbf{x}_k, \tau). \tag{7}$$

The blended radiance $L_{\text{r}}^{\text{blend}}$, premultiplied by the blended opacity α^{blend}, is calculated using

$$\alpha^{\text{blend}}(\mathbf{x}_k, \tau) L_{\text{r}}^{\text{blend}}(\mathbf{x}_k, \boldsymbol{\omega}_{\text{o}}, \tau) = (1 - b(\mathbf{x}_k, \tau)) \cdot \alpha^{\text{s}}(\mathbf{x}_k) L_{\text{r}}^{\text{s}}(\mathbf{x}_k, \boldsymbol{\omega}_{\text{o}}) \tag{8}$$

$$+ b(\mathbf{x}_k, \tau) \cdot \alpha^{\text{d}}(\mathbf{x}_k, \tau) L_{\text{r}}^{\text{d}}(\mathbf{x}_k, \boldsymbol{\omega}_{\text{o}}, \tau), \tag{9}$$

where L_{r}^{s} and L_{r}^{d} are the radiance predicted by the static and dynamic networks, respectively.

3 Continuous-wave Time-of-Flight Image Formation Model

A continuous-wave time-of-flight (C-ToF) sensor is an active imaging system that illuminates the scene with a point light source. The intensity of this light source is modulated with a temporally-varying function $f(t)$, and the temporally-varying response at a camera pixel is

$$i(t) = \int_{-\infty}^{\infty} R(t - s) f(s)\, ds, \tag{10}$$

where $R(t)$ is the scene's temporal response function observed at a particular camera pixel (i.e., the response to a pulse of light emitted at $t = 0$). Note that Equation 10 is a convolution operation between the scene's temporal response function $R(t)$ and the light source modulation function $f(t)$.

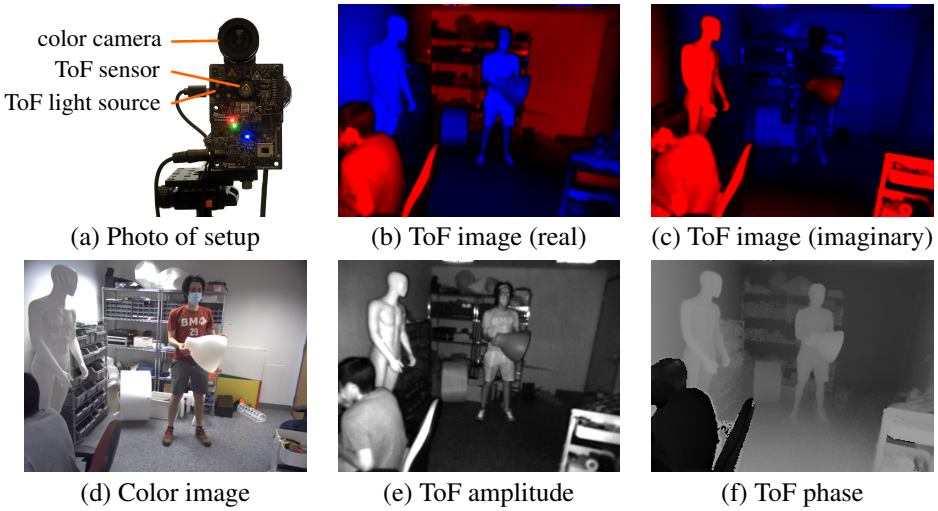
The operating principle of a C-ToF sensor is to modulate the exposure incident on the sensor with a function $g(t)$, and integrating the response over the exposure period. Suppose that $f(t)$ and $g(t)$ are periodic functions, the period is T, and there are N periods during an exposure. A C-ToF sensor would then measure the following:

$$L = \int_0^{NT} g(t) i(t)\, dt \tag{11}$$

$$= \int_0^{NT} g(t) \left(\int_{-\infty}^{\infty} R(t - s) f(s) ds \right) dt \tag{12}$$

$$= N \int_{-\infty}^{\infty} R(s) \underbrace{\left(\int_0^{T} f(t - s) g(t) dt \right)}_{=h(s)} ds, \tag{13}$$

where the function $h(t)$ is the convolution between the exposure modulation function $g(t)$ and the light source modulation function $f(t)$. This function $h(t)$ can be interpreted as a path length importance function, which weights the contribution of light path based on its path length.



(a) Photo of setup (b) ToF image (real) (c) ToF image (imaginary)

(d) Color image (e) ToF amplitude (f) ToF phase

Figure 5: **(a)** Photo of the proposed hardware setup, consisting of a single ToF and a color camera. **(b)** Real component of ToF phasor image (positive/negative values), captured with a modulation frequency $\omega = 30\,\text{MHz}$. **(c)** Imaginary component of ToF phasor image. **(d)** Color image from color camera. **(e)** Amplitude of the phasor image; represents the average amount of infrared light reflected by the scene. **(f)** Phase of the phasor image; values are approximately proportional to range.

In this work, we assume that the C-ToF camera produces phasor images [2], where $h(t) = \exp(\text{i}2\pi\omega t)$. To achieve this, suppose that $f(t) = \frac{1}{2}\sin(2\pi\omega t) + \frac{1}{2}$ and $g(t) = \sin(2\pi\omega t + \phi)$ for a modulation frequency $\omega = \frac{1}{T}$, where ϕ is a controllable phase offset between the two signals. The convolution between these two functions is then $h(t) = \frac{T}{4}\cos(2\pi\omega t + \phi)$. After capturing four images L_ϕ with different phase offsets $\phi \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, we can linearly recombine these measurements as follows:

$$L_{\text{ToF}} = (L_0 - L_\pi) - \text{i}(L_{\frac{\pi}{2}} - L_{\frac{3\pi}{2}}) = \frac{NT}{2}\int_{-\infty}^{\infty} R(s)\exp(\text{i}2\pi\omega s)\,ds. \tag{14}$$

The response at every pixel is therefore a complex phasor. Figure 5(b) and Figure 5(c) provide an example of the real and imaginary component of this phasor image, respectively. As discussed in the main paper, in typical depth sensing scenarios, the phasor's magnitude, $|L_{\text{ToF}}|$, represents the amount of light reflected by a single point in the scene (Figure 5(e)), and the phase, $\angle L_{\text{ToF}}$, is related to distance of that point.

4 Experimental Hardware Setup

The hardware setup shown in Figure 5(a) consists of a standard machine vision camera and a time-of-flight camera. Our USB 3.0 industrial color camera (UI-3070CP-C-HQ Rev. 2) from iDS has a sensor resolution of 2056×1542 pixels, operates at 30 frames per second, and uses a 6 mm lens with an $f/1.2$ aperture. Our high-performance 3D time-of-flight camera (OPT8241-CDK-EVM) from Texas Instruments has a sensor resolution of 320×240 pixels, and also operates at 30 frames per second (software synchronized with the color camera). Camera exposure was 10 ms. The illumination source wavelength of the time-of-flight camera is infrared (850 nm) and invisible to the color camera. The modulation frequency of the time-of-flight camera is $\omega = 30\,\text{MHz}$, resulting in an unambiguous range of 5 m. Both cameras are mounted onto an optical plate, and have a baseline of approximately 41 mm.

We use OpenCV to calibrate the intrinsics, extrinsics and distortion coefficients of the stereo camera system. We undistort all captured images, and resize the color image to 640×480 to improve optimization performance. In addition, the phase associated with the C-ToF measurements may be offset by an unknown constant; we recover this common zero-phase offset by comparing the measured phase values to the recovered position of the calibration target. For simplicity, we assume that the modulation frequency associated with the C-ToF camera is an approximately sinusoidal signal, and ignore any nonlinearities between the recovered phase measurements and the true depth.

Along with the downsampled 640×480 color images, the C-ToF measurements consist of the four 320×240 images, each representing the scene response to a different predefined phase offset ϕ. We linearly combine the four images into a complex-valued C-ToF phasor image representing the response to a complex light signal, as described in Equation 14. To visualize these complex-valued phasor images, we show the real component and imaginary component separately, and label positive pixel values as red and negative values as blue.

5 New and Existing Assets

5.1 Use of Existing Assets

We created synthetic data sequences using existing assets, including assets from the McGuire Computer Graphics Archive [6], 'Architectural Visualization' demo Blender scene by Marek Moravec (CC-0 Public Domain) [8], 'Rampaging T-Rex' from the 3D library of Microsoft's 3D Viewer, and 'Indoor Pot Plant 2' by 3dhaupt from Free3D (non-commercial) [3].

5.2 New Assets

This work includes newly-captured sequences from time-of-flight and RGB sensors. These were created using real-world scenes and people. All places and people consented to being captured and for their image to be released publicly. This data does contains personally-identifiable information—the appearances of people.

References

[1] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. arXiv:2105.06468, 2021.

[2] 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.

[3] Dennis Haupt. Indoor Pot Plant 2, November 2019. URL https://free3d.com/3d-model/indoor-pot-plant-77983.html. Non-commercial use only.

[4] 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.

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

[6] Morgan McGuire. Computer Graphics Archive, July 2017. URL https://casual-effects.com/data.

[7] 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.

[8] Marek Moravec. Architectural visualization—Blender demo scene, November 2019. URL https://www.blender.org/download/demo-files/. CC-0 Public Domain.

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