Abstract

We present an algorithm for generating novel views at arbitrary viewpoints and any input time step given a monocular video of a dynamic scene. Our work builds upon recent advances in neural implicit representation and uses continuous and differentiable functions for modeling the time-varying structure and the appearance of the scene. We jointly train a time-invariant static NeRF and a time-varying dynamic NeRF, and learn how to blend the results in an unsupervised manner. However, learning this implicit function from a single video is highly ill-posed (with infinitely many solutions that match the input video). To resolve the ambiguity, we introduce regularization losses to encourage a more physically plausible solution. We show extensive quantitative and qualitative results of dynamic view synthesis from casually captured videos.

1. Introduction

Video provides a window into another part of the real world. In traditional videos, however, the viewer observes the action from a fixed viewpoint and cannot navigate the scene. Dynamic view synthesis comes to the rescue. These techniques aim at creating photorealistic novel views of a dynamic scene at arbitrary camera viewpoints and time steps, which enables free-viewpoint video and stereo rendering, and provides an immersive and almost life-like viewing experience. It facilitates applications such as replaying professional sports events in 3D [7], creating cinematic effects like freeze-frame bullet-time (from the movie “The Matrix”), virtual reality [11, 5], and virtual 3D teleportation [41].

Systems for dynamic view synthesis need to overcome challenging problems related to video capture, reconstruction, compression, and rendering. Most of the existing methods rely on laborious and expensive setups such as custom fixed multi-camera video capture rigs [8, 65, 11, 41, 5]. While recent work relaxes some constraints and can han-
dle unstructured video input (e.g., from hand-held cameras) [3, 4], many methods still require synchronous capture from multiple cameras, which is impractical for most people. Few methods produce dynamic view synthesis from a single stereo or even RGB camera, but they are limited to specific domains such as human performance capture [12, 20]. Recent work on depth estimation from monocular videos of dynamic scenes shows promising results [29, 62]. Yoon et al. [62] use estimated depth maps to warp and blend multiple images to synthesize an unseen target viewpoint. However, the method uses a local representation (i.e., per-frame depth maps) and processes each novel view independently. Consequently, the synthesized views are not consistent and may exhibit abrupt changes.

This paper presents a new algorithm for dynamic view synthesis from a dynamic video that overcomes this limitation using a global representation. More specifically, we use an implicit neural representation to model the time-varying volume density and appearance of the events in the video. We jointly train a time-invariant static neural radiance field (NeRF) [35] and a time-varying dynamic NeRF, and learn how to blend the results in an unsupervised manner. However, it is challenging for the dynamic NeRF to learn plausible 3D geometry because we have just one and only one 2D image observation at each time step. There are infinitely many solutions that can correctly render the given input video, yet only one is physically correct for generating photorealistic novel views. Our work focuses on resolving this ambiguity by introducing regularization losses to encourage plausible reconstruction. We validate our method’s performance on the Dynamic multi-view dynamic scenes dataset by Yoon et al. [62].

The key points of our contribution can be summarized as follows:

- We present a method for modeling dynamic radiance fields by jointly training a time-invariant model and a time-varying model, and learn how to blend the results in an unsupervised manner.

- We design regularization losses for resolving the ambiguities when learning the dynamic radiance fields.

- Our model leads to favorable results compared to the state-of-the-art algorithms on the Dynamic Scenes Dataset.

2. Related Work

View synthesis from images. View synthesis aims to generate new views of a scene from multiple posed images [51]. Light fields [26] or Lumigraph [19] synthesize realistic appearance but require capturing and storing many views. Using explicit geometric proxies allows high-quality synthesis from relatively fewer input images [6, 16]. However, estimating accurate scene geometry is challenging due to untextured regions, highlights, reflections, and repetitive patterns. Prior work addresses this via local warps [9], operating in the gradient domain [24], soft 3D reconstruction [45], and learning-based approaches [22, 15, 14, 21, 48]. Recently, neural implicit representation methods have shown promising view synthesis results by modeling the continuous volumetric scene density and color with a multilayer perceptron [35, 38, 61, 63].

Several methods tackle novel view synthesis from one single input image. These methods differ in their underlying scene representation, including depth [39, 57], multiplane images [56], or layered depth images [50, 25]. Compared with existing view synthesis methods that focus on static objects or scenes, our work aims to achieve view synthesis of dynamic scenes from one single video.

View synthesis for videos. Free viewpoint video offers immersive viewing experiences and creates freeze-frame (bullet time) visual effects [30]. Compared to view synthesis techniques for images, capturing, reconstructing, compressing, and rendering dynamic contents in videos is significantly more challenging. Many existing methods either focus on specific domains (e.g., humans) [8, 12, 20] or transitions between input views only [3]. Several systems have been proposed to support interactive viewpoint control watching videos of generic scenes [65, 11, 41, 4, 5, 1]. However, these methods require either omnidirectional stereo camera [1], specialized hardware setup (e.g., custom camera rigs) [65, 11, 5, 41], or synchronous video captures from multiple cameras [4]. Recently, Yoon et al. [62] show that one can leverage depth-based warping and blending techniques in image-based rendering for synthesizing novel views of a dynamic scene from a single camera. Similar to [62], our method also synthesizes novel views of a dynamic scene. In contrast to using explicit depth estimation [62], our implicit neural representation based approach facilitates geometrically accurate rendering and smoother view interpolation.

Implicit neural representations. Continuous and differentiable functions parameterized by fully-connected networks (also known as multilayer perceptron, or MLPs) have been successfully applied as compact, implicit representations for modeling 3D shapes [10, 59, 42, 18, 17], object appearances [40, 37], 3D scenes [52, 35, 44]. These methods train MLPs to regress input coordinates (e.g., points in 3D space) to the desired quantities such as occupancy value [33, 49, 44], signed distance [42, 2, 34], volume density [35], color [40, 52, 49, 35]. Leveraging differentiable rendering [31, 23], several recent works have shown training these MLPs with multiview 2D images (without using direct 3D supervision) [37, 60, 35].

Most of the existing methods deal with static scenes. Di-
Figure 2. Method overview. We propose to use two different models to represent the (a) static and (b) dynamic scene components. (a) Static NeRF: For static components, we train a NeRF model following [35], but excluding all the pixels marked as dynamic from training the model. This allows us to reconstruct the background’s structure and appearance without conflicting the moving objects. (b) Dynamic NeRF: Modeling a dynamic scene from a single video is highly ill-posed. To resolve the ambiguity, we leverage the multi-view constraints as follows: Our Dynamic NeRF takes both $r(u_k)$ and $t$ as input to predict 3D scene flow from time $t$ to $t + 1$ ($s_{bw}$) and from time $t$ to $t - 1$ ($s_{fw}$). Using the predicted scene flow, we can create a warped radiance field by resampling the radiance field modeled at the adjacent time instances and apply temporal consistency. Thus, at each instance, we can have multiple views associated with different time instances to train the model.

Concurrent work on dynamic view synthesis. Very recently, several methods concurrently to ours have been proposed to extend NeRF for handling dynamic scenes [58, 27, 54, 43, 46]. These methods either disentangle the dynamic scenes into a canonical template and deformation fields for each frame [54, 46, 43] or directly estimate dynamic (4D) spatiotemporal) radiance fields [58, 27]. Our work adopts the 4D radiance fields approach due to its capability of modeling large scene dynamics. In particular, our approach share high-level similarity with [27] in that we also regularize the dynamic NeRF through scene flow estimation. Our method differs in several important technical details, including scene flow based 3D temporal consistency loss, sparsity regularization, and the rigidity regularization of the scene flow prediction. For completeness, we include experimental comparison with one template-based method [54] and one 4D radiance field approach [27].
directly training the “NeRF + time” model leads to low visual quality.

The key contribution of our paper lies in resolving this ambiguity for modeling the time-varying radiance fields. To this end, we propose to use different models to represent static or dynamic scene components using the user-provided dynamic masks.

For static components of the scene, we apply the original NeRF model [35], but exclude all “dynamic” pixels from training the model. This allows us to reconstruct the background’s structure and appearance without conflicting reconstruction losses from moving objects. We refer to this model as “Static NeRF” (Figure 3).

For dynamic components of the scene (e.g., moving objects), we train an MLP that takes a 3D position and time \((x, y, z, t)\) as input to model the volume density and color of the dynamic objects at each time instance. To leverage the multi-view geometry, we use the same MLP to predict the additional three-dimensional scene flow from time \(t\) to the previous and next time instance. Using the predicted forward and backward scene flow, we create a warped radiance field (similar to the backward warping 2D optical flow) by resampling the radiance fields implicitly modeled at time \(t + 1\) and \(t - 1\). For each 3D position, we then have up to three multi-view observations to train our model. We refer to this model as “Dynamic NeRF” (Figure 4). Additionally, our Dynamic NeRF predicts a blending weight and learns how to blend the results from both the static NeRF and dynamic NeRF in an unsupervised manner. In the following, we discuss the detailed formulation of the proposed static and dynamic NeRF models and the training losses for optimizing the weights for the implicit functions.

3.2. Static NeRF

Formulation. Our static NeRF follows closely the formulation in [35] and is represented by a fully-connected neural network\(^1\). Consider a ray from the camera center \(o\) through a given pixel on the image plane as \(r(u_k) = o + u_kd\), where \(d\) is the normalized viewing direction, our static NeRF maps a 3D position \(r(u_k)\) and viewing direction \(d\) to volume density \(\sigma^s\) and color \(c^s\):

\[
(\sigma^s, c^s) = \text{MLP}_0(r(u_k)),
\]

where \(\text{MLP}_0\) stands for two cascaded MLP, detailed in Figure 2. We can compute the color of the pixel (corresponding the ray \(r(u_k)\)) using numerical quadrature for approximating the volume rendering interval [13]:

\[
C^s(r) = \sum_{k=1}^{K} T^s(u_k) \alpha^s(\sigma^s(u_k), \delta_k) c^s(u_k),
\]

\[
T^s(u_k) = \exp \left( -\sum_{k'=1}^{k-1} \sigma^s(u_{k'}) \delta_k \right),
\]

where \(\alpha(x) = 1 - \exp(-x)\) and \(\delta_k = u_{k+1} - u_k\) is the distance between two quadrature points. The \(K\) quadrature points \(\{u_k\}_{k=1}^{K}\) are drawn uniformly between \(u_s\) and \(u_f\) [35]. \(T^s(u_k)\) indicates the accumulated transmittance from \(u_s\) to \(u_k\).

Static rendering photometric loss. To train the weights \(\theta_s\) of the static NeRF model, we first construct the camera rays using all the pixels for all the video frames (using the associated intrinsic and extrinsic camera poses for each frame). Here we denote \(r_{ij}\) as the rays passing through the pixel \(j\) on image \(i\) with \(r_{ij}(u) = o_i + (u)d_{ij}\). We can then optimize \(\theta_s\) by minimizing the static rendering photometric loss for all the color pixels \(C(r_{ij})\) in frame \(i \in \{0, \ldots, N-1\}\) in

---

\(^1\)We refer the readers to [35] for implementation details.
the neighboring time instance to the current time. For every use the predicted scene flow to

\[ \text{Induced flow} \]

\[ \text{Ours} \]

\[ \text{Estimated flow} \]

\[ \text{Induced flow} \]

\[ \text{Ours} \]

Forward and backward scene flow training. We solve the multi-view reconstruction loss, the learned volume density helps render a more accurate flow than the estimated flow (e.g., the complex structures of the fence on the right).

3.4. Regularization Losses for Dynamic NeRF

While leveraging the multi-view constraint in the dynamic NeRF model reduces the amount of ambiguity, the model training remains ill-posed without proper regularization. To this end, we design several regularization losses to constrain the Dynamic NeRF.

Motion matching loss. As we do not have direct 3D supervision for the predicted scene flow from the motion MLP model, we use 2D optical flow (estimated from input image pairs using [53]) as indirect supervision. For each 3D point at time \( t \), we first use the estimated scene flow to obtain the corresponding 3D point in the reference frame. We then project this 3D point onto the reference camera so we can compute the scene flow induced optical flow and enforce it to match the estimated optical flow (Figure 5). Since we jointly train our model with both photometric loss and motion matching loss, our learned volume density helps render a more accurate flow than the estimated flow. Thus, we do not suffer from inaccurate optical flow supervision.

Motion regularization. Unfortunately, matching the ren-
**Sparsity regularization.** We render the color using principles from classical volume rendering. One can see through a particle if it is partially transparent. However, one can not see through the scene flow because the scene flow is not an intrinsic property (unlike color). Thus, we minimize the entropy of the rendering weights \( T^d \alpha^d \) along each ray so that few samples dominate the rendering.

**Depth order loss.** For a moving object, we can either interpret it as an object close to the camera moving slowly or an object far away moving fast. To resolve the ambiguity, we leverage the state-of-the-art single-image depth estimation [47] to estimate the input depth. As the depth estimates are up to shift and scale, we cannot directly use them.

We further regularize the scene flow to be **spatially smooth** by minimizing the difference between neighboring 3D points’ scene flow. To regularize the consistency of the scene flow, we have the scene flow cycle consistency regularization:

\[
\mathcal{L}_{\text{cyc}} = \sum_{ij} \| s_{fu}(r, t) + s_{bw}(r + s_{fu}(r, t), t + 1) \|^2_2 + \| s_{bw}(r, t) + s_{fu}(r + s_{bw}(r, t), t - 1) \|^2_2
\]

We render the color using principles from classical volume rendering. One can see through a particle if it is partially transparent. However, one can not see through the scene flow because the scene flow is not an intrinsic property (unlike color). Thus, we minimize the entropy of the rendering weights \( T^d \alpha^d \) along each ray so that few samples dominate the rendering.

**3.5. Combined model**

With both the static and dynamic NeRF model, we can easily compose them into a complete model using the predicted blending weight \( b \) and render full color frames at novel views and time steps.
Table 1. Novel view synthesis results. We report the average PSNR and LPIPS results with comparisons to existing methods on Dynamic Scene dataset [62]. The best performance is in **bold** and the second best is *underscored*.

|                | Jumping | Skating | Truck | Umbrella | Balloon1 | Balloon2 | Playground | Average  |
|----------------|---------|---------|-------|----------|----------|----------|------------|----------|
| NeRF           | 20.58 / 0.305 | 23.05 / 0.316 | 22.61 / 0.225 | 21.08 / 0.441 | 19.07 / 0.214 | 24.08 / 0.098 | 20.86 / 0.164 | **21.62 / 0.252** |
| NeRF + time    | 16.72 / 0.489 | 19.23 / 0.542 | 17.17 / 0.403 | 17.17 / 0.752 | 17.17 / 0.304 | 19.67 / 0.236 | 13.80 / 0.444 | **17.30 / 0.453** |
| Yoon et al. [62] | 20.16 / 0.148 | **21.75 / 0.135** | **23.93 / 0.109** | 20.35 / 0.179 | 18.76 / 0.178 | 19.89 / 0.138 | 15.09 / 0.183 | **19.99 / 0.153** |
| Tretschk et al. [55] | 19.38 / 0.295 | 23.29 / 0.234 | 19.02 / 0.453 | 19.26 / 0.427 | 16.98 / 0.353 | 22.23 / 0.212 | 14.24 / 0.336 | **19.20 / 0.330** |
| Li et al. [28]  | **24.12 / 0.156** | **28.91 / 0.135** | **25.94 / 0.171** | 22.58 / 0.302 | **21.40 / 0.225** | 24.09 / 0.228 | 20.91 / 0.220 | **23.99 / 0.205** |
| Ours           | **24.23 / 0.144** | **28.90 / 0.124** | **25.78 / 0.134** | 23.15 / 0.146 | **21.47 / 0.125** | 25.97 / 0.059 | 23.65 / 0.093 | **24.74 / 0.118** |

Figure 8. Novel view synthesis. Our model enables the free-viewpoint synthesis of a dynamic scene. Compared with Yoon et al. [62], our results appear slightly blurry (because we reconstruct the entire frame as opposed to warp and blend input images), but align with the ground truth image better and create smoother view-interpolation results. When compared to other NeRF-based methods, our results are sharper and closer to the ground truth. Please refer to the supplementary material for video results.

\[
C^{\text{full}}(r) = \sum_{k=1}^{K} T^{\text{full}} \left( \alpha^d (\sigma^d \delta_k)(1 - b) c_d + \alpha^s (\sigma^s \delta_k) b c_s \right)
\]

We predict the blending weight \(b_d\) using the dynamic NeRF to enforce the time-dependency. Using the blending weight, we can also render a dynamic component only frame where the static region is transparent (Figure 7).

**Full rendering photometric loss.** We train the two NeRF models jointly by applying a reconstruction loss on the composite results:

\[
L^{\text{full}} = \sum_{ij} \left\| C^{\text{full}}(r_{ij}) - C^{\text{gt}}(r_{ij}) \right\|_2^2
\]

4. Experimental Results

4.1. Experimental setup

**Dataset.** We evaluate our method on the Dynamic Scene Dataset [62], which contains 9 video sequences. The sequences are captured with 12 cameras using a static camera rig. All cameras simultaneously capture images at 12 different time steps \(\{t_0, t_1, \ldots, t_{11}\}\). The input twelve-frames monocular video \(\{I_0, I_1, \ldots, I_{11}\}\) is obtained by sampling the image taken by the \(i\)-th camera at time \(t_i\). Please note that a different camera is used for each frame of the video to simulate camera motion. The frame \(I_i\) contains a background that does not change in time, and a time-varying dynamic object. Like NeRF [35], we use COLMAP to estimate the camera poses and the near and far bounds of the scene. We assume all the cameras share the same intrinsic parameter. We exclude the DynamicFace sequence because COLMAP fails to estimate camera poses. We resize all the sequences to \(480 \times 270\) resolution.

4.2. Evaluation

**Quantitative evaluation.** To quantitatively evaluate the synthesized novel views, we fix the view to the first camera and change time. We show the PSNR and LPIPS [64] between the synthesized views and the corresponding ground truth views in Table 1. We obtain the results of Li et al. [28] and Tretschk et al. [55] using the official implementation with default parameters. Note that the method from Tretschk et al. [55] needs per-sequence hyper-parameter
We show that our proposed regularizations are the keys to better visual results.

Table 2. Ablation study on different losses. We report PSNR, SSIM and LPIPS on the Playground sequence.

|                | PSNR | SSIM | LPIPS |
|----------------|------|------|-------|
| Ours w/o $L_{\text{depth}}$ | 22.99 | 0.8170 | 0.117 |
| Ours w/o $L_{\text{motion}}$ | 22.61 | 0.8027 | 0.137 |
| Ours w/o rigidity | 22.73 | 0.8142 | 0.118 |
| Ours              | **23.65** | **0.8452** | **0.093** |

Figure 9. Comparison with [28]. We show that our proposed regularizations are the keys to better visual results.

The visual quality might be improved with careful hyper-parameter tuning. Our method compares favorably against the state-of-the-art algorithms.

**Qualitative evaluation.** We show the sample view synthesis results in Figure 8. With the learned neural implicit representation of the scene, our method can synthesize novel views that are never seen during training. Please refer to the supplementary video results for the novel view synthesis, and the extensive qualitative comparison to the methods listed in Table 1.

Figure 9 shows the comparison with Li et al. [28] on large motion sequences taken in the wild. Unlike [28] which predicts the blending weight using a static NeRF, we learn a time-varying blending weight. This weight helps better distinguish the static region and yields a clean background. Our rigidity regularization encourages the scene flow to be zero for the rigid region. As a result, the multi-view constraints enforce the background to be static. Without this regularization, the background becomes time-variant and leads to floating artifacts in [28].

4.3. Ablation Study

Table 2 analyzes the contribution of each loss quantitatively.

**Depth order loss.** For a complicated scene, we need additional supervision to learn the correct geometry. In Figure 10 we study the effect of the depth order loss. Since the training objective is to minimize the image reconstruction loss on the input views, the network may learn a solution that correctly renders the given input video. However, it may be a physically incorrect solution and produces artifacts at novel views. With the help of the depth order loss $L_{\text{depth}}$, our dynamic NeRF model learns the correct relative depth and renders plausible content.

**Motion regularization.** Supervising scene flow prediction with the 2D optical flow is under-constrained. We show in Figure 10 that without a proper motion regularization, the synthesized results are blurry. The scene flow may points to the wrong location. By regularizing the scene flow with to be slow, temporally and spatially smooth, and consistent, we obtain plausible results.

**Rigidity regularization of the scene flow.** The rigidity regularization helps with a more accurate scene flow prediction for the static region. The dynamic NeRF is thus trained with a more accurate multi-view constraint. We show in Figure 9 that the rigidity regularization is the key to a clean background.

4.4. Failure Cases

Dynamic view synthesis remains a challenging problem. We show and explain several failure cases in Figure 11.

5. Conclusions

We have presented a new algorithm for dynamic view synthesis from a single monocular video. Our core technical contribution lies in scene flow based regularization for enforcing temporal consistency and alleviates the ambiguity when modeling a dynamic scene with only one observation.
at any given time. We show that our proposed scene flow based 3D temporal consistency loss and the rigidity regularization of the scene flow prediction are the keys to better visual results. We validate our design choices and compare favorably against the state of the arts.

**References**

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryoshka: Real-time 6dof video view synthesis using multi-sphere images. In ECCV, 2020.

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In CVPR, 2020.

[3] Luca Ballan, Gabriel J Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM TOG (Proc. SIGGRAPH)*, 2010.

[4] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4d visualization of dynamic events from unconstrained multi-view videos. In CVPR, 2020.

[5] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Bourgarijan, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM TOG (Proc. SIGGRAPH)*, 39(4):86–1, 2020.

[6] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[7] Canon. Free viewpoint video system. https://global.canon/en/technology/frontier18.html, 2008.

[8] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM TOG (Proc. SIGGRAPH)*, 22(3):569–577, 2003.

[9] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM TOG (Proc. SIGGRAPH)*, 32(3):1–12, 2013.

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In CVPR, 2019.

[11] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM TOG (Proc. SIGGRAPH)*, 34(4):1–13, 2015.

[12] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM TOG (Proc. SIGGRAPH)*, 35(4):1–13, 2016.

[13] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988.

[14] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In CVPR, 2019.

[15] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In CVPR, 2016.

[16] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv preprint arXiv:2012.05903*, 2020.

[17] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *CVPR*, 2020.

[18] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, 2019.

[19] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the conference on Computer graphics and interactive techniques*, pages 43–54, 1996.

[20] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM TOG (Proc. SIGGRAPH)*, 38(2):1–17, 2019.

[21] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM TOG (Proc. SIGGRAPH)*, 37(6):1–15, 2018.

[22] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM TOG (Proc. SIGGRAPH)*, 35(6):1–10, 2016.

[23] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020.

[24] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *ACM TOG (Proc. SIGGRAPH Asia)*, 32(6), 2013.

[25] Johannes Kopf, Kevin Matzen, Suhib Alsian, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM TOG (Proc. SIGGRAPH)*, 39(4):76–1, 2020.

[26] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the annual conference on Computer graphics and interactive techniques*, 1996.

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

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

[29] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM TOG (Proc. SIGGRAPH)*, 2020.
[30] Marcus Magnor, Marc Pollefeys, German Cheung, Wojciech Matusik, and Christian Theobalt. Video-based rendering. In *ACM SIGGRAPH 2005 Courses*, 2005. 2

[31] R Mantiuk and V Sundstedt. State of the art on neural rendering. *STAR*, 39(2), 2020. 2

[32] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*, 2020. 3

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2

[34] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *ICCV*, 2019. 2

[35] 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, 2, 3, 4, 7

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019. 3

[37] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 2

[38] M. Niemeyer, Lars M. Mescheder, Michael Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. *CVPR*, 2020. 2

[39] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM TOG (Proc. SIGGRAPH Asia)*, 38(6):1–15, 2019. 2

[40] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *ICCV*, 2019. 2

[41] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, et al. Holoporation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016. 1, 2

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[43] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo-Martín Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020. 3

[44] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2

[45] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM TOG (Proc. SIGGRAPH)*, 36(6):1–11, 2017. 2

[46] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 3

[47] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 6

[48] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[49] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pixifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 2

[50] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2

[51] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13, 2000. 2

[52] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[53] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 5

[54] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radience fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *arXiv preprint arXiv:2012.12247*, 2020. 3

[55] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radience fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *arXiv preprint arXiv:2012.12247*, 2020. 7

[56] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[57] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[58] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021. 3

[59] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Fajer, Martin Brualla, David Parore, and Alex Kipman. Pifu: Pixel-aligned implicit differentiable renderer for implicit neural representations. *arXiv preprint arXiv:2003.09852*, 2020. 2

[60] 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. 2

[61] Lior Yariv, Matan Atzmon, and Yaron Lipman. Universal differentiable render for implicit neural representations. *arXiv preprint arXiv:2003.09852*, 2020. 2
[62] 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.

[63] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492, 2020.

[64] 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.

[65] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. ACM TOG (Proc. SIGGRAPH), 23(3):600–608, 2004.