

Robust Dynamic Radiance Fields

Yu-Lun Liu^{2*} Chen Gao¹ Andreas Meuleman^{3*} Hung-Yu Tseng¹ Ayush Saraf¹

Changil Kim¹ Yung-Yu Chuang² Johannes Kopf¹ Jia-Bin Huang^{1,4}

¹Meta ²National Taiwan University ³KAIST ⁴University of Maryland, College Park

<https://robust-dynrf.github.io/>



Figure 1. **Robust space-time synthesis from dynamic monocular videos.** Our method takes a casually captured video as input and reconstructs the camera trajectory and dynamic radiance fields. Conventional SfM system such as COLMAP fails to recover camera poses even when using ground truth motion masks. As a result, existing dynamic radiance field methods that require accurate pose estimation do not work on these challenging dynamic scenes. Our work tackles this *robustness* problem and showcases high-fidelity dynamic view synthesis results on a wide variety of videos.

Abstract

Dynamic radiance field reconstruction methods aim to model the time-varying structure and appearance of a dynamic scene. Existing methods, however, assume that accurate camera poses can be reliably estimated by Structure from Motion (SfM) algorithms. These methods, thus, are unreliable as SfM algorithms often fail or produce erroneous poses on challenging videos with highly dynamic objects, poorly textured surfaces, and rotating camera motion. We address this robustness issue by jointly estimating the static and dynamic radiance fields along with the camera parameters (poses and focal length). We demonstrate the robustness of our approach via extensive quantitative and qualitative experiments. Our results show favorable performance over the state-of-the-art dynamic view synthesis methods.

1. Introduction

Videos capture and preserve memorable moments of our lives. However, when watching regular videos, viewers ob-

serve the scene from fixed viewpoints and cannot interactively navigate the scene afterward. Dynamic view synthesis techniques aim to create photorealistic novel views of dynamic scenes from arbitrary camera angles and points of view. These systems are essential for innovative applications such as video stabilization [33, 42], virtual reality [7, 15], and view interpolation [13, 85], which enable free-viewpoint videos and let users interact with the video sequence. It facilitates downstream applications like virtual reality, virtual 3D teleportation, and 3D replays of live professional sports events.

Dynamic view synthesis systems typically rely on expensive and laborious setups, such as fixed multi-camera capture rigs [7, 10, 15, 50, 85], which require simultaneous capture from multiple cameras. However, recent advancements have enabled the generation of dynamic novel views from a single stereo or RGB camera, previously limited to human performance capture [16, 28] or small animals [65]. While some methods can handle unstructured video input [1, 3], they typically require precise camera poses estimated via SfM systems. Nonetheless, there have been

*This work was done while Yu-Lun and Andreas were interns at Meta.

Table 1. Categorization of view synthesis methods.

	<i>Known camera poses</i>	<i>Unknown camera poses</i>
<i>Static scene</i>	NeRF [44], SVS [59], NeRF++ [82], Mip-NeRF [4], Mip-NeRF 360 [5], DirectVoxGO [68], Plenoxels [23], Instant-ngp [45], TensoRF [12]	NeRF - - [73], BARF [40], SC-NeRF [31], NeRF-SLAM [60]
<i>Dynamic scene</i>	NV [43], D-NeRF [56], NR-NeRF [71], NSFF [39], DynamicNeRF [24], Nerfies [52], HyperNeRF [53], TiNeuVox [20], T-NeRF [25]	Ours

many recent dynamic view synthesis methods for unstructured videos [24, 25, 39, 52, 53, 56, 71, 76] and new methods based on deformable fields [20]. However, these techniques require precise camera poses typically estimated via SfM systems such as COLMAP [62] (bottom left of Table 1).

However, SfM systems are not robust to many issues, such as noisy images from low-light conditions, motion blur caused by users, or dynamic objects in the scene, such as people, cars, and animals. The robustness problem of the SfM systems causes the existing dynamic view synthesis methods to be fragile and impractical for many challenging videos. Recently, several NeRF-based methods [31, 40, 60, 73] have proposed jointly optimizing the camera poses with the scene geometry. Nevertheless, these methods can only handle strictly static scenes (top right of Table 1).

We introduce RoDynRF, an algorithm for reconstructing dynamic radiance fields from casual videos. Unlike existing approaches, we do not require accurate camera poses as input. Our method optimizes camera poses and two radiance fields, modeling static and dynamic elements. Our approach includes a coarse-to-fine strategy and epipolar geometry to exclude moving pixels, deformation fields, time-dependent appearance models, and regularization losses for improved consistency. We evaluate the algorithm on multiple datasets, including Sintel [9], Dynamic View Synthesis [79], iPhone [25], and DAVIS [55], and show visual comparisons with existing methods.

We summarize our core contributions as follows:

- We present a space-time synthesis algorithm from a dynamic monocular video that does not require known camera poses and camera intrinsics as input.
- Our proposed careful architecture designs and auxiliary losses improve the robustness of camera pose estimation and dynamic radiance field reconstruction.
- Quantitative and qualitative evaluations demonstrate the robustness of our method over other state-of-the-art methods on several challenging datasets that typical SfM systems fail to estimate camera poses.

2. Related Work

Static view synthesis. Many view synthesis techniques construct specific scene geometry from images captured at various positions [8] and use local warps [11] to synthesize high-quality novel views of a scene. Approaches to

light field rendering use implicit scene geometry to create photorealistic novel views, but they require densely captured images [27, 37]. By using soft 3D reconstruction [54], learning-based dense depth maps [22], multiplane images (MPIs) [14, 21, 67], additional learned deep features [30, 58], or voxel-based implicit scene representations [66], several earlier work attempt to use proxy scene geometry to enhance rendering quality.

Recent methods implicitly model the scene as a continuous neural radiance field (NeRF) [4, 44, 82] with multi-layer perceptrons (MLPs). However, NeRF requires days of training time to represent a scene. Therefore, recent methods [12, 23, 45, 68] replace the implicit MLPs with explicit voxels and significantly improve the training speed.

Several approaches synthesize novel views from a single RGB input image. These methods often fill up holes in the disoccluded regions and predict depth [38, 49], additionally learned features [75], multiplane images [72], and layered depth images [34, 64]. Although these techniques have produced excellent view synthesis results, they can only handle static scenes. Our approach performs view synthesis of *dynamic scenes* from a single monocular video, in contrast to existing view synthesis techniques focusing on static scenes.

Dynamic view synthesis. By focusing on human bodies [74], using RGBD data [16], reconstructing sparse geometry [51], or producing minimal stereoscopic disparity transitions between input views [1], many techniques reconstruct and synthesize novel views from non-rigid dynamic scenes. Other techniques break down dynamic scenes into piece-wise rigid parts using hand-crafted priors [36, 61]. Many systems cannot handle scenes with complicated geometry and instead require multi-view and time-synchronized videos as input to provide interactive view manipulation [3, 7, 41, 85]. Yoon *et al.* [79] used depth from single-view and multi-view stereo to synthesize novel views of dynamic scenes from a single video using explicit depth-based 3D warping.

A recent line of work extends NeRF to handle dynamic scenes [20, 24, 39, 52, 53, 56, 71, 76]. Although these space-time synthesis results are impressive, these techniques rely on precise camera pose input. Consequently, these techniques are not applicable to challenging scenes where COLMAP [62] or current SfM systems fail. Our approach, in contrast, can handle complex dynamic scenarios without known camera poses.

Visual odometry and camera pose estimation. From a collection of images, visual odometry estimates the 3D camera poses [18, 19, 46–48]. These techniques mainly fall into two categories: direct methods that maximize photometric consistency [78, 84] and feature-based methods that rely on manually created or learned features [46, 47, 63]. Self-supervised image reconstruction losses have recently

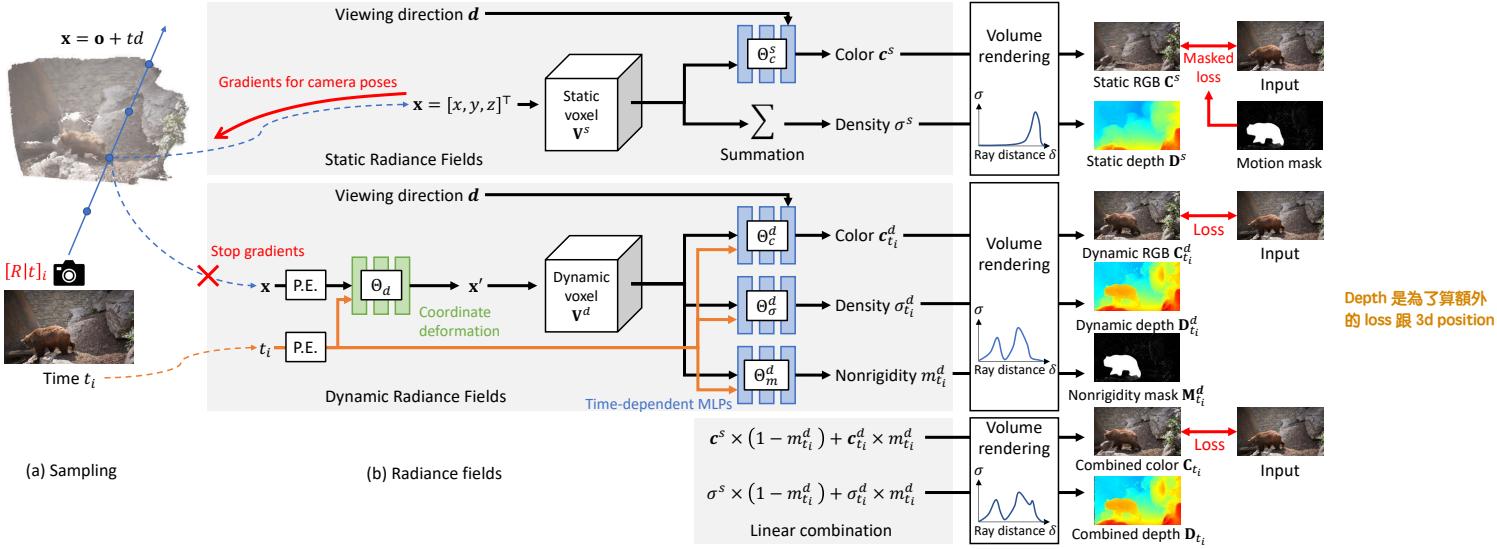


Figure 2. Overall framework. We model the dynamic scene with static and dynamic radiance fields. The static radiance fields take both the sampled coordinates (x, y, z) and the viewing direction d as input and predict the density σ^s and color c^s . Note that the density of the static part is invariant to time and viewing direction, therefore, we use summation of the queried features as the density (instead of using an MLP). We only compute the losses over the static regions. The computed gradients backpropagate not only to the static voxel field and MLPs but also to the camera parameters. The dynamic radiance fields take the sampled coordinates and the time t to obtain the *deformed coordinates* (x', y', z') in the canonical space. Then we query the features using these deformed coordinates from the dynamic voxel fields and pass the features along with the time index to a time-dependent shallow MLPs to get the color c^d , density σ^d , and nonrigidity m^d of the dynamic part. Finally, after the volume rendering, we can obtain the RGB images $C^{\{s,d\}}$ and the depth maps $D^{\{s,d\}}$ from the static and dynamic parts along with a nonrigidity mask M^d . Finally, we calculate the per-frame reconstruction loss. Note that here we only include per-frame losses.

been used in learning-based systems to tackle visual odometry [2, 6, 26, 35, 70, 77, 80, 81, 83, 84]. Estimating camera poses from casually captured videos remains challenging. NeRF-based techniques have been proposed to combine neural 3D representation and camera poses for optimization [31, 40, 60, 73], although they are limited to static sequences. In contrast to the visual odometry techniques outlined above, our system simultaneously optimizes camera poses and models *dynamic objects* models.

3. Method

In this section, we first briefly introduce the background of neural radiance fields and their extension of camera pose estimation and dynamic scene representation in Section 3.1. We then describe the overview of our method in Section 3.2. Next, we discuss the details of camera pose estimation with the static radiance field reconstruction in Section 3.3. After that, we show how to model the dynamic scene in Section 3.4. Finally, we outline the implementation details in Section 3.5.

3.1. Preliminaries

NeRF. Neural radiance fields (NeRF) [44] represent a static 3D scene with implicit MLPs parameterized by Θ and map

the 3D position (x, y, z) and viewing direction (θ, ϕ) to its corresponding color c and density σ :

$$(c, \sigma) = \text{MLP}_\Theta(x, y, z, \theta, \phi). \quad (1)$$

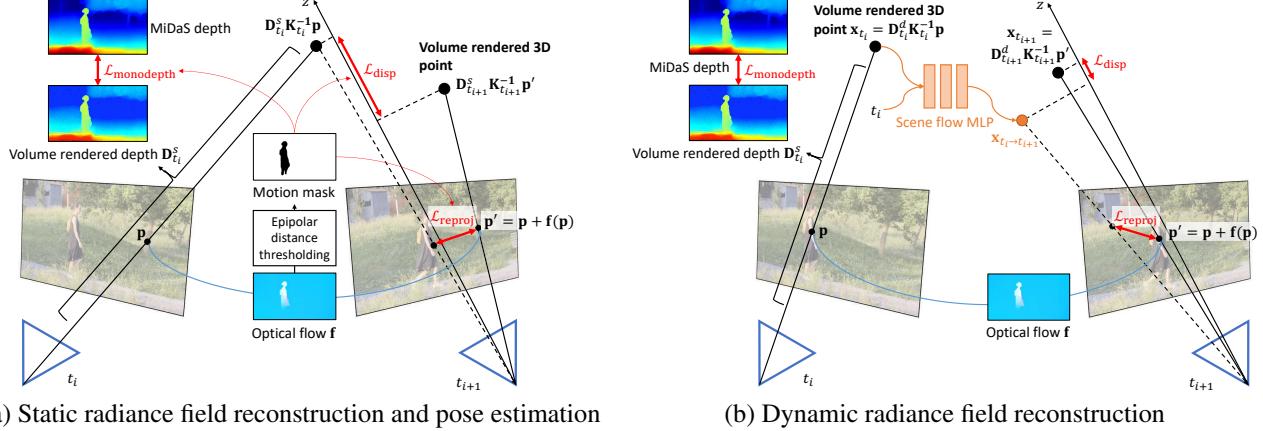
We can compute the pixel color by applying volume rendering [17, 32] along the ray r emitted from the camera origin:

$$\begin{aligned} \hat{C}(r) &= \sum_{i=1}^N T(i)(1 - \exp(-\sigma(i)\delta(i)))c(i), \\ T(i) &= \exp\left(-\sum_{j=1}^i \sigma(j)\delta(j)\right), \end{aligned} \quad (2)$$

where $\delta(i)$ represents the distance between two consecutive sample points along the ray, N is the number of samples along each ray, and $T(i)$ indicates the accumulated transparency. As the volume rendering procedure is differentiable, we can optimize the radiance fields by minimizing the reconstruction error between the rendered color \hat{C} and the ground truth color C :

$$\mathcal{L} = \left\| \hat{C}(r) - C(r) \right\|_2^2. \quad (3)$$

Explicit neural voxel radiance fields. Although with compelling rendering quality, NeRF-based methods model



(a) Static radiance field reconstruction and pose estimation

Figure 3. **Training losses.** For both the (a) static and (b) dynamic parts, we introduce three auxiliary losses to encourage the consistency of the modeling: reprojection loss, disparity loss, and monocular depth loss. The reprojection loss encourages the projection of the 3D volume rendered points onto neighbor frames to be similar to the pre-calculated flow. The disparity loss forces the volume rendered 3D points from two corresponding points of neighbor frames to have similar z values. Finally, the monocular depth loss calculates the scale- and shift-invariant loss between the volume rendered depth and the pre-calculated MiDaS depth. (a) We use the motion mask to exclude the dynamic regions from the loss calculation. (b) We use a scene flow MLP to model the 3D movement of the volume rendered 3D points.

the scene with implicit representations such as MLPs for high storage efficiency. These methods, however, are very slow to train. To overcome this drawback, recent methods [12, 23, 45, 68] propose to model the radiance fields with explicit voxels. Specifically, these methods replace the mapping function with voxel grids and directly optimize the features sampled from the voxels. They usually apply shallow MLPs to handle the view-dependent effects. By eliminating the heavy usage of the MLPs, the training time of these methods reduces from days to hours. We also leverage explicit representation in this work.

3.2. Method Overview

We show our proposed framework in Figure 2. Given an input video sequence with N frames, our method jointly optimizes the camera poses, focal length, and static and dynamic radiance fields. We represent both the static and dynamic parts with explicit neural voxels \mathbf{V}^s and \mathbf{V}^d , respectively. The static radiance fields are responsible for reconstructing the static scene and estimating the camera poses and focal length. At the same time, the goal of dynamic radiance fields is to model the scene dynamics in the video (usually caused by moving objects).

3.3. Camera Pose Estimation

Motion mask generation. Excluding dynamic regions in the video helps improve the robustness of camera pose estimation. Existing methods [39] often leverage off-the-shelf instance segmentation methods such as Mask R-CNN [29] to mask out the common moving objects. However, many moving objects are hard to detect/segment in the input video, such as drifting water or swaying trees. Therefore, in

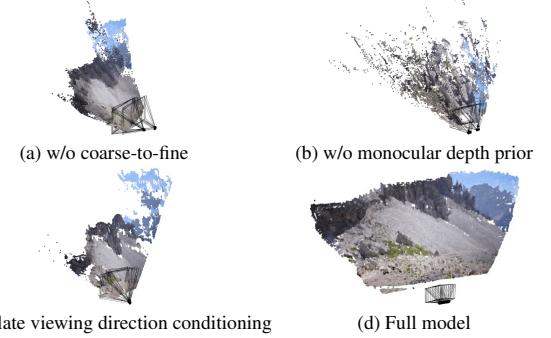


Figure 4. **The impact of design choices on camera pose estimation.** (a) No coarse-to-fine strategy leads to sub-optimal solutions. (b) No single-image depth prior results in poor scene geometry for challenging camera trajectories. (c) The absence of late viewing direction conditioning leads to wrong geometry and poses due to minimizing photometric loss instead of consistent voxel space using MLP. (d) Our proposed method incorporates all components and yields reasonable scene geometry and camera trajectory.

addition to the masks from Mask R-CNN, we also estimate the fundamental matrix using the optical flow from consecutive frames. We then calculate and threshold the Sampson distance (the distance of each pixel to the estimated epipolar line) to obtain a binary motion mask. Finally, we combine the results from Mask R-CNN and epipolar distance thresholding to obtain our final motion masks.

因為被觀察點 P 在兩張圖的 epipolar line 會因為 P 點的運動而改變，所以若 P 點跟 estimated epipolar line 的距離過大，代表 P 有在移動。

Coarse-to-fine static scene reconstruction. The first part of our method is reconstructing the static radiance fields along with the camera poses. We jointly optimize the 6D camera poses $[R|t]_i, i \in [1..N]$ and the focal length f shared by all input frames simultaneously. Similar to ex-

isting pose estimation methods [40], we optimize the static scene representation in a coarse-to-fine manner. Specifically, we start with a smaller static voxel resolution and progressively increase the voxel resolution during the training. This coarse-to-fine strategy is essential to the camera pose estimation as the energy surface will become smoother. Thus, the optimizer will have less chance of getting stuck in sub-optimal solutions (Figure 4(a) vs. Figure 4(d)).

Late viewing direction conditioning. As our primary supervision is the photometric consistency loss, the optimization could bypass the neural voxel and directly learn a mapping function from the viewing direction to the output sample color. Therefore, we choose to fuse the viewing direction only in the last layer of the color MLP as shown in Figure 2. This design choice is critical because we are reconstructing not only the scene geometry but also the camera poses. Figure 4(c) shows that without the late viewing direction conditioning, the optimization could minimize the photometric loss by optimizing the MLP and lead to erroneous camera poses and geometry estimation.

Losses. We minimize the photometric loss between the prediction $\hat{\mathbf{C}}^s(\mathbf{r})$ and the captured images in the static regions:

$$\mathcal{L}_c^s = \left\| (\hat{\mathbf{C}}^s(\mathbf{r}) - \mathbf{C}(\mathbf{r})) \cdot (1 - \mathbf{M}(\mathbf{r})) \right\|_2^2, \quad (4)$$

where \mathbf{M} denotes the motion mask.

To handle casually-captured but challenging camera trajectories such as fast-moving or pure rotating, we introduce auxiliary losses to regularize the training, similar to [24,39].

(1) Reprojection loss $\mathcal{L}_{\text{reproj}}^s$: We use 2D optical flow estimated by RAFT [69] to guide the training. First, we volume render all the sampled 3D points along a ray to generate a surface point. We then reproject this point onto its neighbor frame and calculate the reprojection error with the correspondence estimated from RAFT.

(2) Disparity loss $\mathcal{L}_{\text{disp}}^s$: Similar to the reprojection loss above, we also regularize the error in the z-direction (in the camera coordinate). We volume render the two corresponding points into 3D space and calculate the error of the z component. As we care more about the near than the far, we compute this loss in the inverse-depth domain.

(3) Monocular depth loss $\mathcal{L}_{\text{monodepth}}^s$: The two losses above cannot handle pure rotating cameras and often lead to the incorrect camera poses and geometry (Figure 4(b)). We enforce the depth order from multiple pixels of the same frame to match the order of a monocular depth map. We pre-calculate the depth map using MiDaSv2.1 [57]. The depth prediction from MiDaS is up to an unknown scale and shift. Therefore, we use the same scale- and shift-invariant loss in MiDaS to constrain our rendered depth values.

We illustrate these auxiliary losses in Figure 3(a). Since the optical flow and depth map may not be accurate, we

Table 2. Quantitative evaluation of camera poses estimation on the MPI Sintel dataset. The methods of the top block discard the dynamic components and do not reconstruct the radiance fields; thus they cannot render novel views. We exclude the COLMAP results since it fails to produce poses in 5 out of 14 sequences.

Method	ATE (m)	RPE trans (m)	RPE rot (deg)
R-CVD [35]	0.360	0.154	3.443
DROID-SLAM [70]	0.175	0.084	1.912
ParticleSfM [83]	0.129	0.031	0.535
NeRF - - [73]	0.433	0.220	3.088
BARF [40]	0.447	0.203	6.353
Ours	0.089	0.073	1.313

apply annealing for the weights of these auxiliary losses during the training. As the input frames contain dynamic objects, we need to mask out all the dynamic regions while applying all these losses and the L2 reconstruction loss. The final loss for the static part is:

$$\mathcal{L}^s = \mathcal{L}_c^s + \lambda_{\text{reproj}}^s \mathcal{L}_{\text{reproj}}^s + \lambda_{\text{disp}}^s \mathcal{L}_{\text{disp}}^s + \lambda_{\text{monodepth}}^s \mathcal{L}_{\text{monodepth}}^s. \quad (5)$$

3.4. Dynamic Radiance Field Reconstruction

Handling temporal information. To query the time-varying features from the voxel, we first pass the 3D coordinates (x, y, z) along with time index t_i to a coordinate deformation MLP. The coordinate deformation MLP predicts the 3D time-varying deformation vectors $(\Delta x, \Delta y, \Delta z)$. We then add these deformations onto the original coordinates to get the deformed coordinates (x', y', z') . This deformation MLP indicates that the voxel is a canonical space and that each corresponding 3D point from a different time should point to the same position in this voxel space. We design the deformation MLP to deform the 3D points from the original camera space to the canonical voxel space.

However, using a single compact canonical voxel to represent the entire sequence along the temporal dimension is very challenging. Therefore, we further introduce time-dependent MLPs to enhance the queried features from the voxel to predict time-varying color and density. Note that the time-dependent MLPs with only two to three layers are much shallower than the ones in other dynamic view synthesis methods [24,39] as the purpose of the MLPs here is further to enhance the queried features from the canonical voxel. Most of the time-varying effects are still carried out by the coordination deformation MLP. We show the above architecture at the bottom of Figure 2. And the photometric training loss for the dynamic part is:

$$\mathcal{L}_c^d = \left\| \hat{\mathbf{C}}^d(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2, \quad (6)$$

Scene flow modeling. We introduce three losses based on external priors to better model the dynamic movements.

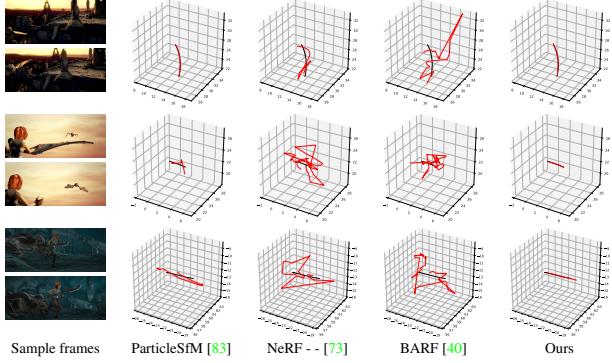


Figure 5. Qualitative results of moving camera localization on the MPI Sintel dataset.

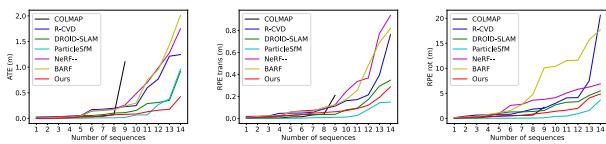


Figure 6. The sorted error plots showing both the *accuracy* and *completeness/robustness* in the MPI Sintel dataset.

The three losses are similar to the ones in the static part, but we need to model the movements of the 3D points. Therefore, we introduce a **scene flow MLP** to compensate the 3D motion.

$$(S_{i \rightarrow i+1}, S_{i \rightarrow i-1}) = \text{MLP}_{\theta_{sf}}(x, y, z, t_i), \quad (7)$$

where $S_{i \rightarrow i+1}$ represents the 3D scene flow of the 3D point (x, y, z) at time t_i . With the 3D scene flow, we can apply the losses for the dynamic radiance fields. We show the training losses in Figure 3(b).

(1) Reprojection loss $\mathcal{L}_{\text{reproj}}^d$: We induce the 2D flow using the poses, depth, and the estimated 3D scene flow from the **scene flow MLP**. And we compare the error of this induced flow with the one estimated by RAFT.

(2) Disparity loss $\mathcal{L}_{\text{disp}}^d$: Similar to the disparity loss in the static part, but here we additionally have the **3D scene flow**. We get the corresponding points in the 3D space, add the estimated 3D scene flow, and calculate the difference of the z components in the inverse-depth domain.

(3) Monocular depth loss $\mathcal{L}_{\text{monodepth}}^d$: We calculate scale- and shift-invariant loss between the rendered depth with the pre-calculated depth map using MiDaSv2.1.

We further regularize the 3D motion prediction from the MLP by introducing the **smooth and small scene flow loss**:

$$\mathcal{L}_{\text{sf}}^{\text{reg}} = \|S_{i \rightarrow i+1} + S_{i \rightarrow i-1}\|_1 + \|S_{i \rightarrow i+1}\|_1 + \|S_{i \rightarrow i-1}\|_1. \quad (8)$$

Note that the **scene flow MLP is not part of the rendering process but part of the losses**. By representing the 3D scene flow with an MLP and enforcing proper priors, we can make

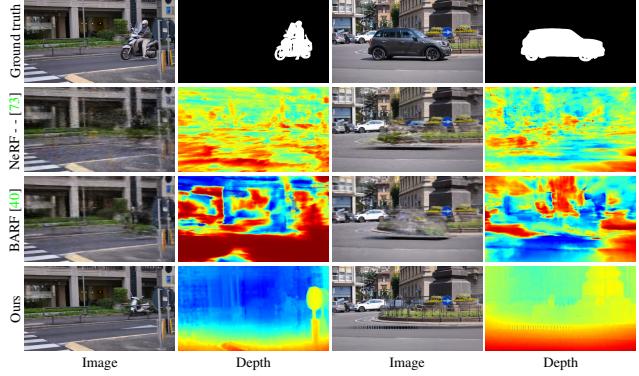


Figure 7. Qualitative results of static view synthesis on the DAVIS dataset from *unknown camera poses* and *ground truth foreground masks*.

the density prediction better and more reasonable. We also detach the gradients from the dynamic radiance fields to the camera poses. Finally, we supervise the nonrigidity mask \mathbf{M}^d with motion mask \mathbf{M} :

$$\mathcal{L}_m^d = \|\mathbf{M}^d - \mathbf{M}\|_1. \quad (9)$$

The overall loss of the dynamic part is:

$$\begin{aligned} \mathcal{L}^d &= \mathcal{L}_c^d + \lambda_{\text{reproj}}^d \mathcal{L}_{\text{reproj}}^d + \lambda_{\text{disp}}^d \mathcal{L}_{\text{disp}}^d + \\ &\quad \lambda_{\text{monodepth}}^d \mathcal{L}_{\text{monodepth}}^d + \lambda_{\text{sf}}^{\text{reg}} \mathcal{L}_{\text{sf}}^{\text{reg}} + \lambda_m^d \mathcal{L}_m^d. \end{aligned} \quad (10)$$

We then linearly compose the static and dynamic parts into the final results with the predicted nonrigidity m^d :

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N T(i)(m^d(1 - \exp(-\sigma^d(i)\delta(i)))\mathbf{c}^d(i) + (1 - m^d)(1 - \exp(-\sigma^s(i)\delta(i)))\mathbf{c}^s(i)). \quad (11)$$

Total training loss. The total training loss is:

$$\mathcal{L} = \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2 + \mathcal{L}^s + \mathcal{L}^d. \quad (12)$$

3.5. Implementation Details

We simultaneously estimate camera poses, focal length, static radiance fields, and dynamic radiance fields. For forward-facing scenes, we parameterize the scenes with **normalized device coordinates (NDC)**. To handle unbounded scenes in the wild videos, we parameterize the scenes using the **contraction parameterization** [5]. To encourage solid surface scene reconstruction and prevent floaters, we add the **distortion loss** [5, 68]. We set the finest voxel resolution to 262,144,000 and 27,000,000 for NDC and contraction, respectively. We also **decompose the voxel grid using the VM-decomposition in TensoRF** [12] for model compactness. The entire training process takes around **28 hours with one NVIDIA V100 GPU**. We provide the detailed architecture in the supplementary material.

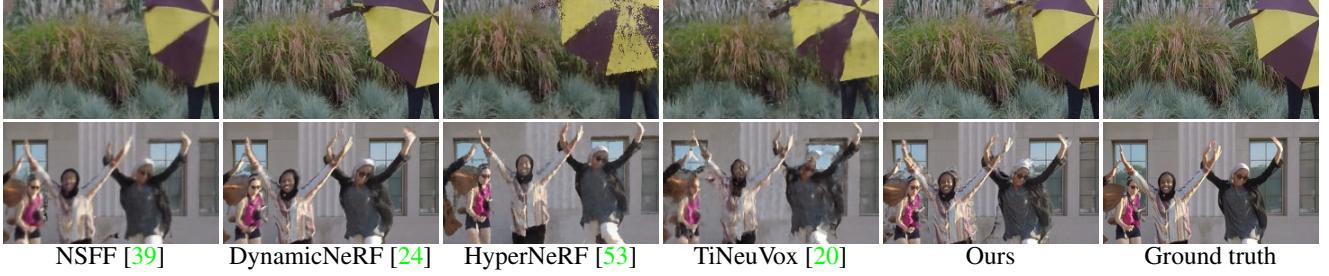


Figure 8. **Novel view synthesis.** Compared to other methods, our results are sharper, closer to the ground truth, and contain fewer artifacts.

Table 3. **Novel view synthesis results.** We report the average PSNR and LPIPS results with comparisons to existing methods on Dynamic Scene dataset [79]. *: Numbers are adopted from DynamicNeRF [24].

PSNR ↑ / LPIPS ↓	Jumping	Skating	Truck	Umbrella	Balloon1	Balloon2	Playground	Average
NeRF* [44]	20.99 / 0.305	23.67 / 0.311	22.73 / 0.229	21.29 / 0.440	19.82 / 0.205	24.37 / 0.098	21.07 / 0.165	21.99 / 0.250
D-NeRF [56]	22.36 / 0.193	22.48 / 0.323	24.10 / 0.145	21.47 / 0.264	19.06 / 0.259	20.76 / 0.277	20.18 / 0.164	21.48 / 0.232
NR-NeRF* [71]	20.09 / 0.287	23.95 / 0.227	19.33 / 0.446	19.63 / 0.421	17.39 / 0.348	22.41 / 0.213	15.06 / 0.317	19.69 / 0.323
NSFF* [39]	24.65 / 0.151	29.29 / 0.129	25.96 / 0.167	22.97 / 0.295	21.96 / 0.215	24.27 / 0.222	21.22 / 0.212	24.33 / 0.199
DynamicNeRF* [24]	24.68 / 0.090	32.66 / 0.035	28.56 / 0.082	23.26 / 0.137	22.36 / 0.104	27.06 / 0.049	24.15 / 0.080	26.10 / 0.082
HyperNeRF [53]	18.34 / 0.302	21.97 / 0.183	20.61 / 0.205	18.59 / 0.443	13.96 / 0.530	16.57 / 0.411	13.17 / 0.495	17.60 / 0.367
TiNeuVox [20]	20.81 / 0.247	23.32 / 0.152	23.86 / 0.173	20.00 / 0.355	17.30 / 0.353	19.06 / 0.279	13.84 / 0.437	19.74 / 0.285
Ours w/ COLMAP poses	25.66 / 0.071	28.68 / 0.040	29.13 / 0.063	24.26 / 0.089	22.37 / 0.103	26.19 / 0.054	24.96 / 0.048	25.89 / 0.065
Ours w/o COLMAP poses	24.27 / 0.100	28.71 / 0.046	28.85 / 0.066	23.25 / 0.104	21.81 / 0.122	25.58 / 0.064	25.20 / 0.052	25.38 / 0.079

4. Experimental Results

Due to the space limit, we leave the experimental setup, including datasets, compared methods, and the evaluation metrics to the supplementary materials.

4.1. Evaluation on Camera Poses Estimation

We conduct the camera pose estimation evaluation on the MPI Sintel dataset [9] and show the quantitative results in Table 2. Our method performs significantly better than existing NeRF-based pose estimation methods. Note that our method also performs favorably against existing learning-based visual odometry methods. We show some visual comparisons of the predicted camera trajectories in Figure 5, and the sorted error plots that show both the accuracy and completeness/robustness in Figure 6. Our approach predicts accurate camera poses over other NeRF-based pose estimation methods. Our method is a global optimization over the entire sequence instead of local registration like SLAM-based methods. Therefore, our RPE trans and rot scores are slightly worse than ParticleSfM [83] as consecutive frames’ rotation is less accurate.

To further reduce the effect of the dynamic parts, we use the ground truth motion masks provided by the DAVIS dataset to mask out the loss calculations in the dynamic regions for all the NeRF-based compared methods. We show the reconstructed images and depth maps in Figure 7. Our approach can successfully reconstruct the detailed content and the faithful geometry thanks to the auxiliary losses. On the contrary, other methods often fail to reconstruct consistent scene geometry and thus produce poor synthesis results.

4.2. Evaluation on Dynamic View Synthesis

Quantitative evaluation. We follow the evaluation protocol in DynamicNeRF [24] to synthesize the view from the first camera and change time on the NVIDIA dynamic view synthesis dataset. We report the PSNR and LPIPS in Table 3. Our method performs favorably against state-of-the-art methods. Furthermore, even without COLMAP poses, our method can still achieve results comparable to the ones using COLMAP poses.

We also follow the evaluation protocol in DyCheck [25] and evaluate quantitatively on the iPhone dataset [25]. We report the masked PSNR and SSIM in Table 4 and show that our method performs on par with existing methods.

Qualitative evaluation. We show some visual comparisons on the NVIDIA dynamic view synthesis dataset in Figure 8 and DAVIS dataset in Figure 9. COLMAP fails to estimate the camera poses for 44 out of 50 sequences in the DAVIS dataset. Therefore, we first run our method and give our camera poses to other methods as input. With the joint learning of the camera poses and radiance fields, our method produces frames with fewer visual artifacts. Other methods can also benefit from our estimated poses to synthesize novel views. With our poses, they can reconstruct consistent static scenes but often generate artifacts for the dynamic parts. In contrast, our method utilizes the auxiliary priors and thus produces results of much better visual quality.

Table 4. **Novel view synthesis results.** We compare the mPSNR and mSSIM scores with existing methods on the iPhone dataset [25].

mPSNR ↑ / mSSIM ↑	Apple	Block	Paper-windmill	Space-out	Spin	Teddy	Wheel	Average
NSFF [39]	17.54 / 0.750	16.61 / 0.639	17.34 / 0.378	17.79 / 0.622	18.38 / 0.585	13.65 / 0.557	13.82 / 0.458	15.46 / 0.569
Nerfies [52]	17.64 / 0.743	17.54 / 0.670	17.38 / 0.382	17.93 / 0.605	19.20 / 0.561	13.97 / 0.568	13.99 / 0.455	16.45 / 0.569
HyperNeRF [53]	16.47 / 0.754	14.71 / 0.606	14.94 / 0.272	17.65 / 0.636	17.26 / 0.540	12.59 / 0.537	14.59 / 0.511	16.81 / 0.550
T-NeRF [25]	17.43 / 0.728	17.52 / 0.669	17.55 / 0.367	17.71 / 0.591	19.16 / 0.567	13.71 / 0.570	15.65 / 0.548	16.96 / 0.577
Ours	18.73 / 0.722	18.73 / 0.634	16.71 / 0.321	18.56 / 0.594	17.41 / 0.484	14.33 / 0.536	15.20 / 0.449	17.09 / 0.534

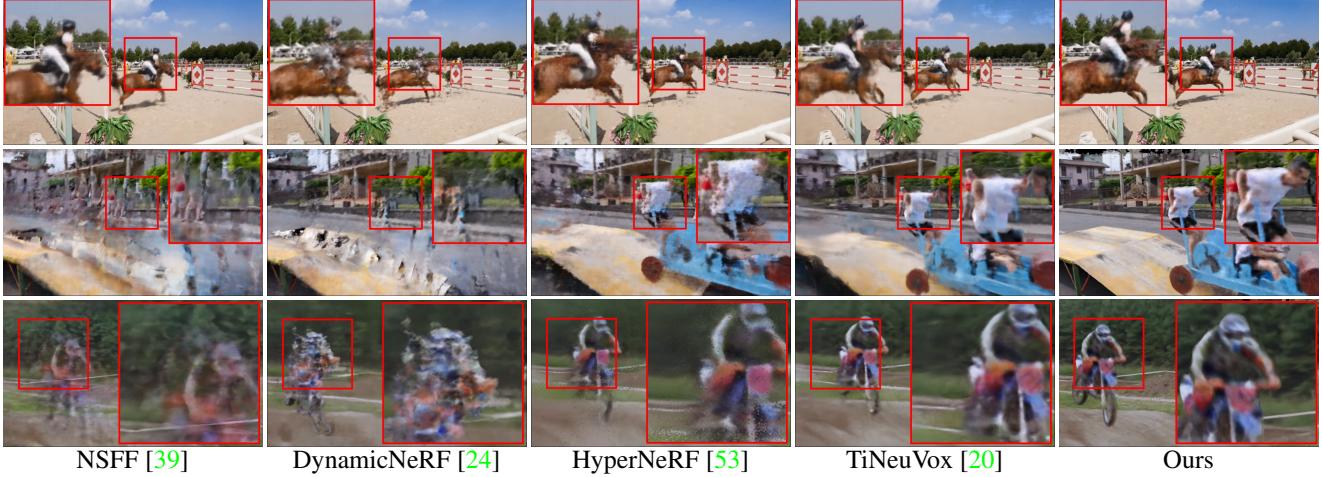


Figure 9. **Novel space-time synthesis results on the DAVIS dataset with our estimated camera poses.** COLMAP fails to produce reliable camera poses for most of the sequences in the DAVIS dataset. With the estimated camera poses by our method, we can run other methods and perform space-time synthesis on the scenes that are not feasible with COLMAP. Our method produces images with much better quality.

Table 5. **Ablation studies.** We report PSNR, SSIM and LPIPS on the Playground sequence.

(a) Pose estimation design choices

	PSNR ↑	SSIM ↑	LPIPS ↓
Ours w/o coarse-to-fine	12.45	0.4829	0.327
Ours w/o late viewing direction fusion	18.34	0.5521	0.263
Ours w/o stopping the dynamic gradients	21.47	0.7392	0.211
Ours	25.20	0.9052	0.052

(b) Dynamic reconstruction architectural designs

Dyn. model	Deform. MLP	Time-depend. MLPs	PSNR ↑	SSIM ↑	LPIPS ↓
✓	✓		21.34	0.8192	0.161
✓		✓	22.37	0.8317	0.115
✓	✓	✓	23.14	0.8683	0.083
			25.20	0.9052	0.052

4.3. Ablation Study

We analyze the design choices in Table 5. For the camera poses estimation, the coarse-to-fine voxel upsampling strategy is the most critical component. Late viewing direction fusion and stopping the gradients from the dynamic radiance field also help the optimization find better poses and lead to higher-quality rendering results. Please refer to Figure 4 for visual comparisons. For the dynamic radiance field reconstruction, both the deformation MLP and the time-dependent MLPs improve the final rendering quality.

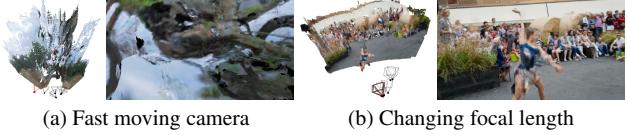


Figure 10. **Failure cases.** (a) In the cases that the camera is moving fast, the flow estimation fails and leads to wrong estimated poses and geometry. (b) Our method assumes a shared intrinsic over the entire video and thus cannot handle changing focal length well.

4.4. Failure Cases

Even with these efforts, robust dynamic view synthesis from a monocular video without known camera poses is still challenging. We show some failure cases in Figure 10.

5. Conclusions

We present *robust dynamic radiance fields* for space-time synthesis of casually captured monocular videos without requiring camera poses as input. With the proposed model designs, we demonstrate that our approach can reconstruct accurate dynamic radiance fields from a wide range of challenging videos. We validate the efficacy of the proposed method via extensive quantitative and qualitative comparisons with the state-of-the-art.

References

- [1] Luca Ballan, Gabriel J Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM TOG*, pages 1–11, 2010. [1](#), [2](#)
- [2] Irene Ballester, Alejandro Fontán, Javier Civera, Klaus H Strobl, and Rudolph Triebel. Dot: Dynamic object tracking for visual slam. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021. [3](#)
- [3] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4d visualization of dynamic events from unconstrained multi-view videos. In *CVPR*, 2020. [1](#), [2](#)
- [4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. [2](#)
- [5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. [2](#), [6](#)
- [6] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 2018. [3](#)
- [7] 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 TOG*, 39:86–1, 2020. [1](#), [2](#)
- [8] 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*, pages 425–432, 2001. [2](#)
- [9] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. [2](#), [7](#)
- [10] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM TOG*, 22:569–577, 2003. [1](#)
- [11] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM TOG*, 32:1–12, 2013. [2](#)
- [12] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. [2](#), [4](#), [6](#)
- [13] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993. [1](#)
- [14] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *ICCV*, 2019. [2](#)
- [15] 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*, 34:1–13, 2015. [1](#)
- [16] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escalano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM TOG*, 35:1–13, 2016. [1](#), [2](#)
- [17] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM TOG*, 22:65–74, 1988. [3](#)
- [18] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE TPAMI*, 40:611–625, 2017. [2](#)
- [19] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014. [2](#)
- [20] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. *ACM TOG*, 2022. [2](#), [7](#), [8](#)
- [21] 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. [2](#)
- [22] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *CVPR*, 2016. [2](#)
- [23] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. [2](#), [4](#)
- [24] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. [2](#), [5](#), [7](#), [8](#)
- [25] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022. [2](#), [7](#), [8](#)
- [26] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. [3](#)
- [27] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. [2](#)
- [28] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM TOG*, 38:1–17, 2019. [1](#)
- [29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. [4](#)
- [30] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM TOG*, 37:1–15, 2018. [2](#)
- [31] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. [2](#), [3](#)
- [32] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM TOG*, 18:165–174, 1984. [3](#)
- [33] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *ACM TOG*, 33:1–10, 2014. [1](#)

- [34] Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM TOG*, 39:76–1, 2020. 2
- [35] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, 2021. 3, 5
- [36] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *ICCV*, 2017. 2
- [37] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2
- [38] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019. 2
- [39] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2, 4, 5, 7, 8
- [40] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2, 3, 5, 6
- [41] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. In *ICCV*, 2021. 2
- [42] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Hybrid neural fusion for full-frame video stabilization. In *ICCV*, 2021. 1
- [43] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM TOG*, 38:65:1–65:14, 2019. 2
- [44] 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, 7
- [45] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41:102:1–102:15, 2022. 2, 4
- [46] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31:1147–1163, 2015. 2
- [47] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras. *IEEE transactions on robotics*, 33:1255–1262, 2017. 2
- [48] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2
- [49] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM TOG*, 38:1–15, 2019. 2
- [50] Sergio Orts-Escalano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 741–754, 2016. 1
- [51] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *ECCV*, 2010. 2
- [52] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *CVPR*, 2021. 2, 8
- [53] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM TOG*, 40, 2021. 2, 7, 8
- [54] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM TOG*, 36:1–11, 2017. 2
- [55] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2
- [56] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2, 7
- [57] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 44, 2020. 5
- [58] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2
- [59] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *CVPR*, 2021. 2
- [60] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 2, 3
- [61] Chris Russell, Rui Yu, and Lourdes Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *ECCV*, 2014. 2
- [62] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2
- [63] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2
- [64] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2
- [65] Samarth Sinha, Roman Shapovalov, Jeremy Reizenstein, Ignacio Rocco, Natalia Neverova, Andrea Vedaldi, and David Novotny. Common pets in 3d: Dynamic new-view synthesis of real-life deformable categories. *arXiv preprint arXiv:2211.03889*, 2022. 1
- [66] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

- [67] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019. 2
- [68] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 4, 6
- [69] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 5
- [70] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 3, 5
- [71] 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 dynamic scene from monocular video. In *ICCV*, 2021. 2, 7
- [72] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2
- [73] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 3, 5, 6
- [74] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humanerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 2
- [75] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2
- [76] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021. 2
- [77] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 2019. 3
- [78] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 2
- [79] 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. 2, 7
- [80] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 3
- [81] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. Vdo-slam: a visual dynamic object-aware slam system. *arXiv preprint arXiv:2005.11052*, 2020. 3
- [82] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2
- [83] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*, 2022. 3, 5, 6, 7
- [84] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2, 3
- [85] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM TOG*, 23:600–608, 2004. 1, 2