Robust Dynamic Radiance Fields
Supplementary Material

Yu-Lun Liu[2*] Chen Gao[1] Andreas Meuleman[3*] Hung-Yu Tseng[1] Ayush Saraf[1]
Changil Kim[1] Yung-Yu Chuang[2] Johannes Kopf[1] Jia-Bin Huang[1,4]
[1]Meta [2]National Taiwan University [3]KAIST [4]University of Maryland, College Park
https://robust-dynrf.github.io/

1. Overview

This supplementary material presents additional results to complement the main manuscript. First, we explain all the losses in our training process in Section 2. Then, we describe the experimental setup, including the datasets, compared methods, and evaluation metrics in Section 3. Next, we show the complete visual comparisons of the camera pose estimation results on the MPI Sintel Dataset in Section 4. In addition to this document, we provide an interactive HTML interface to compare our video results with state-of-the-art methods.

2. Losses

In this section, we explain all the training losses in detail.

Reconstruction loss. We train the static and dynamic radiance fields by minimizing the RGB reconstruction loss between the predicted and input images.

Reprojection loss. Based on the optimized camera poses and geometry, we project each 3D sampled point into the neighbor frame and compute the induced optical flow. Additionally, we enforce this induced flow to be similar to the optical flow estimated by RAFT. For the static radiance field, we calculate this loss only in the static regions indicated by the precalculated motion masks. For the dynamic radiance field, in addition to the relative camera pose and the geometry, we also take into account the scene flow estimated by the scene flow MLP when computing the induced flow.

Disparity loss. Similar to the reprojection loss that regularizes the consistency in the image's spatial domain, we also enforce the consistency in the depth domain. We project each volume-rendered 3D point into the neighbor camera and take the z component. We also calculate the z component of the corresponding (by RAFT flow) volume-rendered 3D point in the neighbor frame. Further, we enforce these two z values to be as close as possible. For the static radiance field, we calculate this loss only in the static regions indicated by the precalculated motion masks. For the dynamic radiance field, we also consider the scene flow estimated by the scene flow MLP when computing the z values.

Monocular depth loss. In order to deal with challenging camera motion, such as pure rotation or little translation and parallax, we use the state-of-the-art off-the-shell single-image depth estimation to guide the geometry during the optimization. Additionally, this loss helps resolve the ambiguity of the scale of a moving object. In accordance with MiDaS's training strategy, we apply a scale- and shift-invariant loss between the predicted depth and the depth determined by MiDaS. For the static radiance field, we only calculate this loss in the static regions indicated by the precalculated motion masks.

Motion mask loss. We enforce the volume-rendered nonrigid mask to be similar to the pre-calculated motion mask. Note that this way, the optimization can learn how to blend the static and dynamic parts instead of relying entirely on the pre-calculated motion masks.

Smooth scene flow loss. We regularize the prediction of the scene flow MLP to be smooth across time.

Small scene flow loss. As the time interval in the input monocular video is small, we regularize the prediction of the scene flow MLP to be small.

Distortion loss. We follow Mip-NeRF 360 to introduce distortion loss and eliminate the novel views' floaters.

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

Voxel TV loss. As our model is based on explicit voxel, which does not have smoothness regularization like MLPs, we enforce a TV loss on the voxel space to ensure the smoothness of the prediction.

Voxel density L1 loss. We encourage the voxel to be as sparse as possible by introducing a density L1 regularization loss.

3. Experimental Setup

Datasets. We evaluate our proposed method on four datasets: (1) the MPI Sintel dataset [1], (2) the Nvidia dynamic view synthesis dataset [15], (3) the iPhone dataset [4], and (4) the DAVIS dataset [10].

For the purpose of validating camera pose estimation, we use the MPI Sintel dataset, which contains ground truth camera trajectory data. The final version of the dataset is used, and we disregard sequences containing static or perfect line camera trajectory. We evaluated fourteen sequences in total.

The Nvidia dynamic view synthesis dataset contains nine dynamic scenes captured by a camera rig. Therefore we use this dataset to quantitatively and visually evaluate the performance of space-time synthesis methods.

Third, we quantitatively evaluate on the very recent iPhone dataset which contains seven dynamic scenes with ground truth novel view images.

Lastly, we use the DAVIS dataset, which contains fifty challenging sequences. All of the sequences in the DAVIS dataset contain dynamic moving objects such as animals or cars. COLMAP is not able to process most of the sequences as the camera movement is either too small or too challenging. COLMAP can only estimate the camera poses for six out of fifty sequences with *ground truth* object masks. Since COLMAP has difficulty processing this dataset, we use *our estimated camera poses* for all the compared methods. Note that we *do not* use the provided foreground masks for dynamic view synthesis.

Compared methods. We compare our method with state-of-the-art camera pose estimation and dynamic view synthesis methods.

- **Camera pose estimation**: We compare with learning-based pose estimation methods R-CVD [6], DROID-SLAM [12], and ParticleSfM [17]. We also compare with view synthesis methods that do not require camera poses as input: NeRF - - [14] and BARF [8]. We also try to use SCNeRF [5] but the trainings do not converge and result in NaN results. Therefore we exclude SCNeRF from our comparisons.

- **Dynamic view synthesis**: We compare with existing view synthesis methods that can handle dynamic scenes: D-NeRF [11], NR-NeRF [13], NSFF [7], DynamicNeRF [3], HyperNeRF [9], and TiNeuVox [2].

We obtain the results of all the methods using the official implementations with default configurations.

Evaluation metrics. For pose estimation, we report the three commonly used error metrics: RMSE of absolute trajectory error (ATE), translation, and rotation part of relative pose error (RPE trans and RPE rot). As the estimated trajectories are up to unknown scales, we scale, rotate, and align the predictions to the ground truth trajectories. For dynamic view synthesis, we evaluate the entire synthesized images using the peak signal-to-noise ratio (PSNR) and perceptual similarity using LPIPS [16].

4. Camera Pose Estimation Evaluation on the MPI Sintel Dataset

We show all the 14 sequences of the camera pose estimation comparisons on the Sintel dataset in Figure 1, Figure 2, and Figure 3.

References

[1] 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

[2] 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

[3] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2

[4] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022. 2

[5] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 2

[6] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, 2021. 2

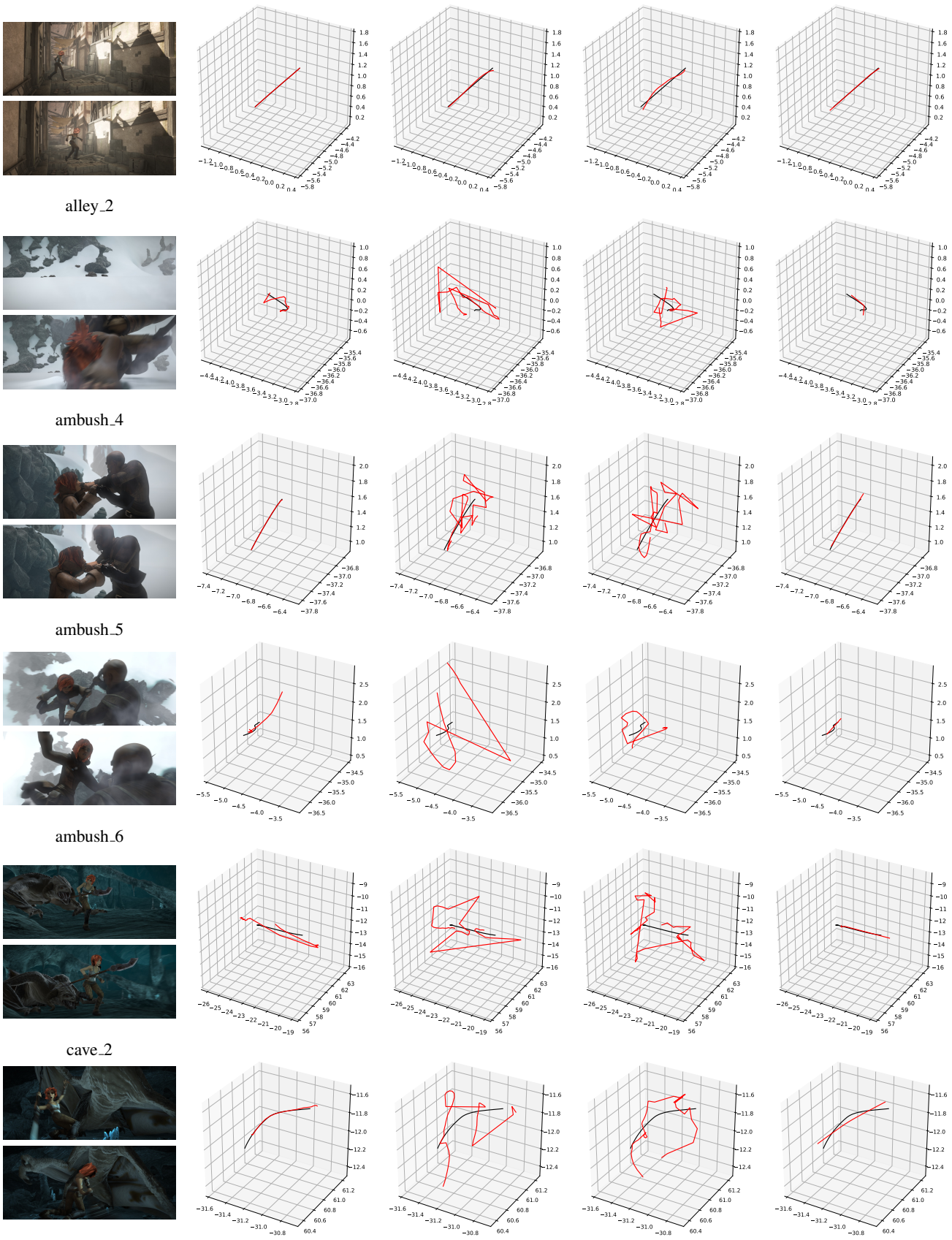

Figure 1. **Qualitative results of moving camera localization on the MPI Sintel dataset.**

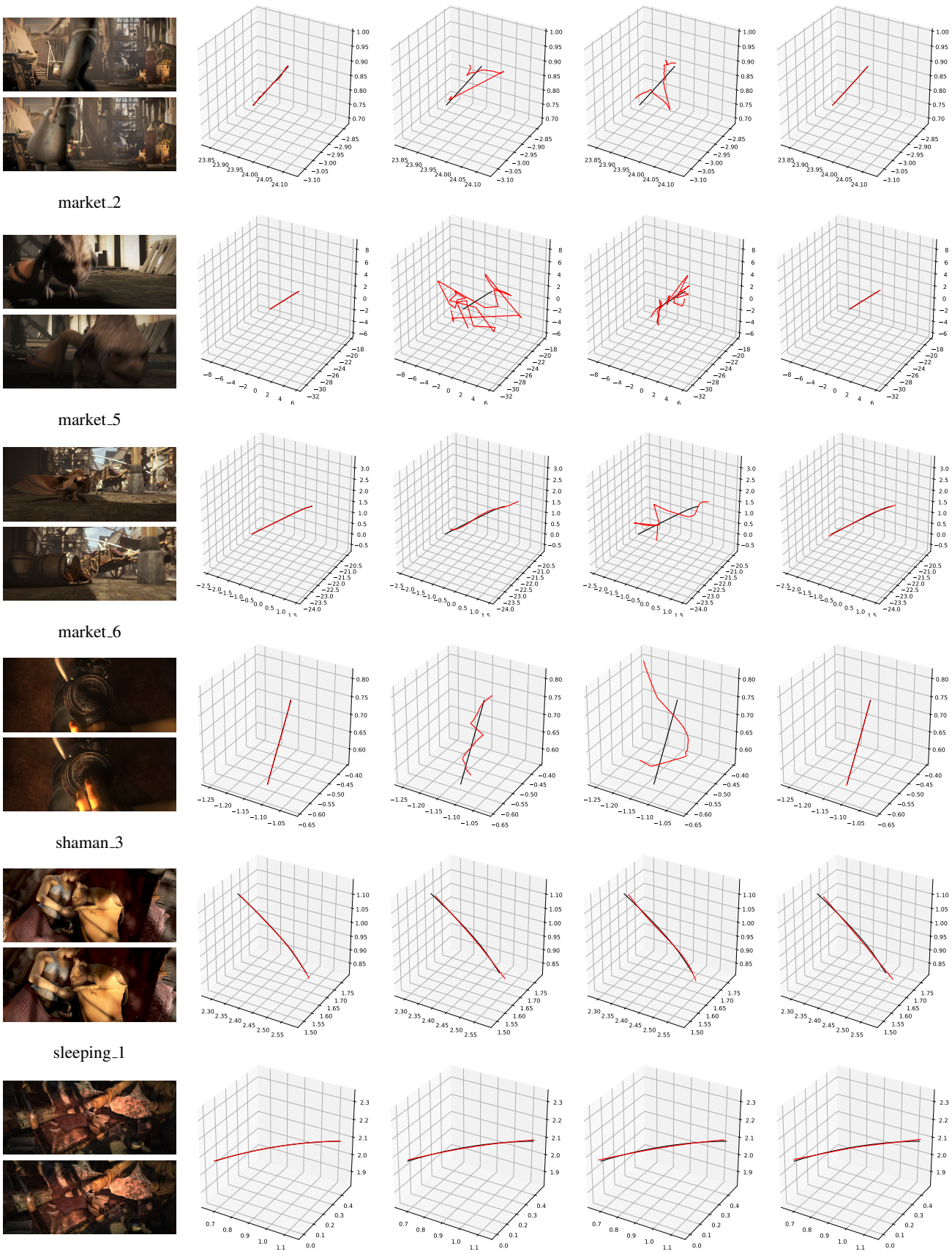

Figure 2. **Qualitative results of moving camera localization on the MPI Sintel dataset.**

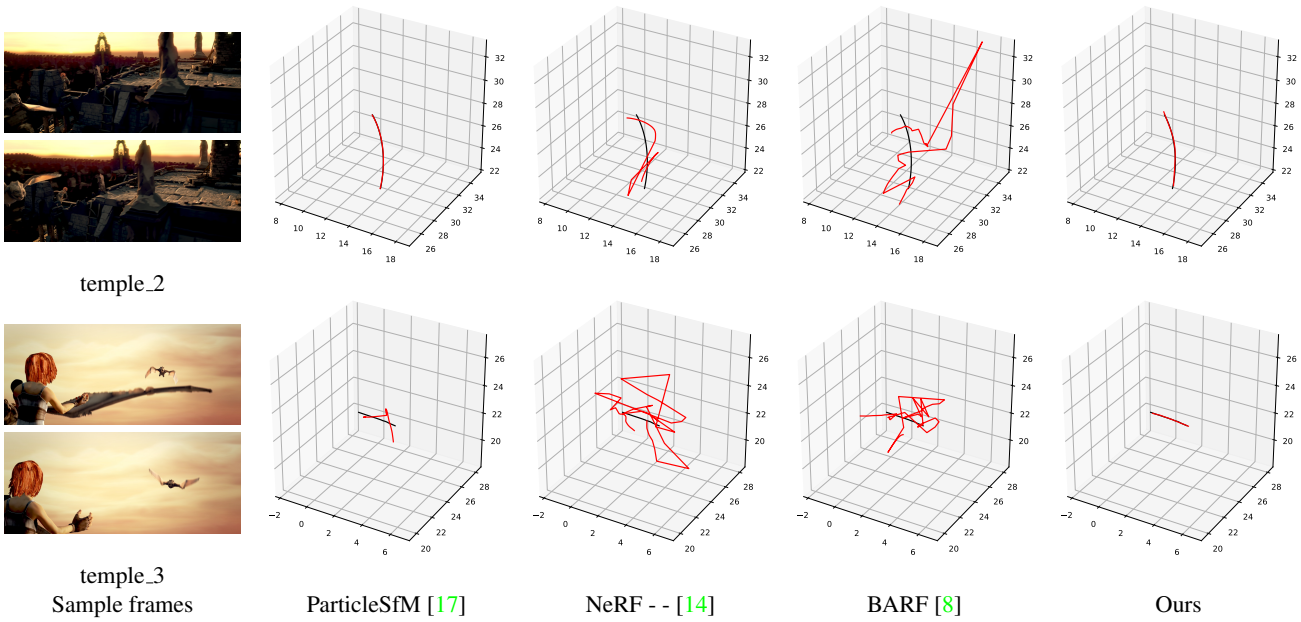

Figure 3. **Qualitative results of moving camera localization on the MPI Sintel dataset.**

[7] 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

[8] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2, 3, 4, 5

[9] 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

[10] 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

[11] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2

[12] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 2

[13] 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

[14] 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, 4, 5

[15] 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

[16] 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. 2

[17] 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. 2, 3, 4, 5