RGBD Objects in the Wild: Scaling Real-World 3D Object Learning from RGB-D Videos

Supplementary Material

A. Additional Details of WildRGB-D Dataset

Video types distribution Our dataset comprises objects paired with their respective videos, with each object being associated with three recorded videos. One of these videos is labeled as the *Single object video*, while the classification of the other two videos is contingent on the nature of the object. In our data collection strategy, we selectively opt for certain object categories to capture *Hand-object video*, excluding categories where objects, such as toy trains or buses, lack meaningful interactions with hands. Within our chosen categories, no more than 10 percent of objects feature a single clip of *Hand-object video*, and the remaining videos consistently fall into the category of *Multi-object video*.

Video recording details To ensure comprehensive 360-degree recording with minimal camera shaking and motion blurs in our videos, we implement a turntable hidden beneath the table, out of the camera's view. This turntable features an adjustable arm that can vary the pitch angle and distance. At the end of the arm, there's a mount for an iPhone, facilitating recording. During filming, we can rotate the arm, ensuring a smooth and uniform rotation for stable footage. To enhance dataset diversity, we deliberately choose various radii and pitch angles during recording. Additionally, for the background of objects, we deliberately select diverse scenarios to further enrich the variability of our dataset. The average video length is over 600 frames.

Dataset mask annotation details. With Grounding-DINO [8] to get prompts (bounding box) for SAM [6], every generated mask will be examined again by the annotators to ensure mask quality. If the mask is wrong, we will label the mask manually by explicit clicks in the image, which serve as click prompts for SAM, and generate the final correct mask. This generates almost the same quality of results as manually labeling all masks but is much more efficient.

Video quality examination We systematically exclude videos that fail to meet our predefined criteria. Specifically, any videos lacking a complete 360-degree recording are eliminated from consideration. For the quality of captured depths, errors of Apple's TrueDepth Camera only reach up to 5% of the target distance, so the quality of depths is quite good in our collected RGB-D videos. After applying Simultaneous Localization and Mapping (SLAM) processing to determine camera poses, we discard videos that exhibit an unreasonable camera trajectory. This evaluation is conducted by meticulous examination of point cloud reconstructions and visualized camera trajectories. Through this rigorous quality control process, we guarantee that our retained videos exhibit a consistently high level of quality.

COLMAP Result	RGB		RGB + Depth Sup	
	COLMAP	RGBD SLAM	COLMAP	RGBD SLAM
79/138 (**57.2%**) Fail	N/A	29.85	N/A	30.86
59/138 (42.8%) Success	31.10	31.33	31.75	31.38

Table 1. **The results of view synthesis** using different ways to estimate camera poses. We report the PSNR which is evaluated with NeuS2 [14].

Evaluation of pose quality from COLMAP and SLAM To evaluate the accuracy of camera poses estimated from COLMAP [11] and RGBD SLAM [4, 12], we run NeuS2 [14] with camera poses estimated from them, and report the PSNR of view synthesis in Tab. 1. Column 2 and 3 show results using only RGB supervision; Column 3 and 4 use both RGB and depth as supervision. We randomly choose three single object scenes from every category in our dataset, resulting in 138 scenes. We find COLMAP fails in 57.2% of the scenes on camera pose estimation, which RGBD SLAM can succeed in all cases. This shows *the significance of how much depth can help* in camera pose estimation. In the cases where COLMAP fails (1st row), adding depth supervision can boost view synthesis by a large margin with RGBD SLAM cameras. We report N/A with COLMAP cameras as the estimations fail. In 2nd row where COLMAP works, all ablations lead to similar results. This suggests depth is very useful in challenging scenes that cover a significant number of data.

Personal data and human subject During our dataset collection process, we enlisted the assistance of hired workers to record videos on our behalf. In a small fraction of these videos, the workers inadvertently appear, featuring their hands or partial bodies. Importantly, we have secured explicit consent from these individuals to include these specific portions of the videos in our dataset. This ensures that the inclusion of worker-related content is both intentional and authorized, maintaining transparency and adherence to ethical standards in our dataset compilation.

B. Additional Experiment Details

B.1. Novel view synthesis

Dataset splits In the context of Single-Scene Novel View Synthesis (NVS), where NeRF-based methods [2, 9, 10] are evaluated, we employ a randomized approach wherein we select ten scenes at random from each category. Subsequently, we uniformly subsample each video to a fixed length of 100 frames and further extract 20 percent of these frames for validation purposes. For Cross-Scene NVS experiments, we designate the same ten scenes chosen for the Single-Scene NVS as the test scenes within each category. The remaining scenes in each category are then utilized for training. During evaluations, we exclusively test on the 20 percent of images sampled from these test video clips, mirroring the methodology employed in Single-Scene NVS experiments. The remaining images serve as source views for synthesizing renderings in the context of Generalizable Neural Radiance Fields (NeRFs) [3, 13, 15]. Notably, for every method, we consistently employ three source views for the synthesis of renderings, which are chosen in a deterministic way. Difficulty level division in Cross-Scene NVS experiments is caused by various object shapes, backgrounds, and the number of training videos in each category.

Training details For every NeRF-based method, we largely follow its original training process. NeRF [9], Mip-NeRF 360 [2] and Instant-NGP [10] are all trained with 30k iterations using their default hyper-parameters. Pixel-NeRF [15], MVSNeRF [3] and IBRNet [13] are trained using default settings, but with different iterations and epochs. Pixel-NeRF [15] is trained in 200 epochs, MVSNeRF [3] and IBRNet [13] are trained with 100k iterations. We ensure enough training time for every method to be correctly evaluated.

B.2. Camera pose estimation

Dataset splits Our dataset, consisting of 46 categories, is systematically divided into 27 training categories and 19 test categories. During the training phase, we exclusively utilize 70 percent of the videos from the training categories, reserving the remaining 30 percent for validation. The metrics derived from this validation process are reported under the designation of "seen" categories. Evaluation results on the test categories are distinctly reported as "unseen" categories. This partitioning strategy ensures a robust assessment of model performance on both familiar and novel categories, contributing to a comprehensive evaluation framework. For every video, we uniformly subsample each video to a fixed length of 100 frames as well.

Training details We follow the training procedure and hyper-parameters in RelPose [17] and RelPose++ [7], with different iterations: RelPose [17] is trained with 100k iterations and RelPose++ [7] with 400k iterations.

B.3. Object surface reconstruction

Implementation details For Instant-NGP [10], we adopt its re-implementation in [1]. We apply an additional mask loss. To be more specific, for every casting ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ from the camera center $\mathbf{o}$ through the pixel center in direction $\mathbf{d}$, Instant-NGP samples a set of 3D points $\{\mathbf{x}_i\}$ along the ray. After querying respective density $\{\sigma_i\}$ of the points $\{\mathbf{x}_i\}$, we calculate the opacity of the ray by $\sum_i w_i$. We want to make it aligned with the mask m_r of that ray by the mask loss $\mathcal{L}_{\text{mask}} = \sum_{\mathbf{r}} \| \sum_i w_i - \mathbf{m}_r \|_2^2$. We additionally add depth loss to Instant-NGP in RGB-D surface reconstruction. For Neusfacto [16], we follow their implementations in SDFStudio, applying their proposed mask loss and sensor depth loss.

Training details We train Instant-NGP [10] with 30k iterations. For RGB surface reconstruction in Neusfacto [16], we train with a longer 60k iterations. RGB-D surface reconstruction in Neusfacto [16], we train with only 10k iterations because of its fast convergence speed.

B.4. Object 6D pose estimation

Training details We choose three common categories of our dataset and Wild6D [5] for category-level 6D pose estimation. We don't subsample our video here. We train every model following [18] for 20k iterations with default hyper-parameters.

References

[1] ngp-pl. https://github.com/kwea123/ngp_pl. 2

[2] 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

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2

[4] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 1

[5] Yang Fu and Xiaolong Wang. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset, 2022. 2

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[7] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 2

[8] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1

[9] 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

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 2

[11] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[12] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 1

[13] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2

[14] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction, 2023. 1

[15] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[16] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022. 2

[17] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision*, pages 592–611. Springer, 2022. 2

[18] Kaifeng Zhang, Yang Fu, Shubhankar Borse, Hong Cai, Fatih Porikli, and Xiaolong Wang. Self-supervised geometric correspondence for category-level 6d object pose estimation in the wild. *arXiv preprint arXiv:2210.07199*, 2022. 2