

Predict-Optimize-Distill: A Self-Improving Cycle for 4D Object Understanding

Supplementary Material

A. List of objects

We include the list of real objects that were used for qualitative results; all of them were captured using smart phone camera as well as the long monocular videos. A sample of objects and their corresponding groups are shown in the main paper.

The full list of objects and their names are:

1. Tractor
2. Redbox
3. Stapler
4. Scissors
5. Retractable knife
6. Folded lamp
7. Carrot knife
8. Vacuum
9. Barbarian
10. Switch
11. T-Rex
12. Pokeball
13. Wooden Drawer
14. Bike Pump

B. POD Implementation Details

B.1. Predictive Model

POD’s predictor extracts DINOv2 features from the input image, positionally encodes them with a sinusoidal embedding, and uses a 3-layer transformer decoder to decode 2 output tokens. The first token is passed through an MLP to predict object to camera pose, and the second is passed into a distinct MLP to predict a vector of all part poses. We represent neural network SE(3) outputs with the 6DoF Gram-Schmidt representation for SO(3) and a 3-vector for position. For the first cycle, we train the predictor model for 250 epochs on the synthetic data with a batch size of 1600 using L1 loss on its outputs. The predictor model is finetuned for 150 epochs in the following cycles. During inference, to match the training distribution of synthetic data we mask the object with SAMv2 [3].

B.2. Optimization

Before batch optimization, we optimize object’s global transformation for 15 steps for each frame while keeping the part poses fixed, to better align the rendered image with the ground truth signal. With the optimized global transformation, we combine it with the predicted camera pose to obtain the optimized camera, which serves as the initial camera pose for batch optimization. Temporal smoothness

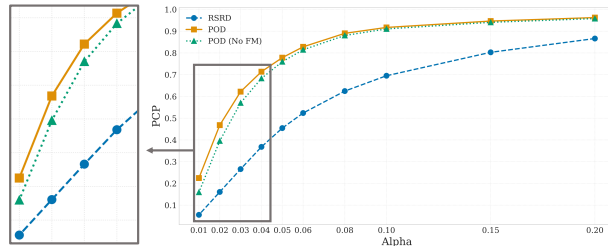


Figure 1. Exhaustive results for PCP thresholds. POD improves over RSRD in accuracy over all thresholds. We also compare against no frame matching (quasi-multiview), which results in worse performance at more strict thresholds where minor deviations are more noticeable.

losses alone do not completely remove temporal discontinuities in high dimensional pose space without data-driven priors [2, 4], hence we further apply DCT smoothing on our results (both ours and baselines) to remove high frequency jitter artifacts in the final results, following the DCT low-pass filtering method proposed in Akhter et al. [1].

B.3. PCP Metric

We compute PCP by measuring the percentage of points within a threshold of the correct corresponding point. A tighter threshold will yield more strict tolerance on error. We show exhaustive results for PCP at different thresholds here.

References

- [1] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. *Advances in neural information processing systems*, 21, 2008. 1
- [2] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *2017 international conference on 3D vision (3DV)*, pages 421–430. IEEE, 2017. 1
- [3] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1
- [4] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1