NeRF-Pose: A First-Reconstruct-Then-Regress Approach for Weakly-supervised 6D Object Pose Estimation Supplementary

Fu Li[1,2*] Shishir Reddy Vutukur[2,3*] Hao Yu[2] Ivan Shugurov[2,3] Benjamin Busam[2]
Shaowu Yang[1] Slobodan Ilic[2,3]
[1] National University of Defense Technology [2] Technical University of Munich
[3] Siemens AG, Munich.

We add some additional training details and results in the following sections.

1. Implementation and Runtime Analysis

We implement our object-centric NeRF network based on the original version of NeRF in[15], and train the network from scratch. For the 2D detector, we use the standard YOLOv3 [4] detector in stage two. An ImageNet [12] pretrained ResNet34 [6] network is leveraged as the backbone of our pose regression network. All networks are trained until convergence.

The training is done on a machine with a Titan RTX GPU with 24GB Memory, an Intel(R) i7-8700K CPU and 24GB RAM. During inference, for a single image with 640×480 resolution, our approach takes about 0.25s for one object, including about 0.03s for YOLOV3 2D detector, 0.01s for pose regression and 0.21s for our pose solver.

2. Limitations

Though we present **NeRF-Pose** in a weakly-supervised way, considering the training difference in OBJ-NeRF network and our pose regression net, failing to enable end-to-end optimization sometimes leads to local minima.

As indicated in Tab. 1, when trained on *pbr* images rendered by *Blender* with high quality from BOP [10, 9], GDR [19] and SO-Pose[2] gain about 10% improvement on ADD(-S) metric. Those fully-supervised methods benefit from *pbr* images that cover more poses and have more realistic occlusion under various light conditions. It inspires us to generate more synthetic training data using our well-trained OBJ-NeRF for better performance.

3. Experiments

We add object-wise results in this section for the Linemod dataset(LM) in Table 2 and Linemod Occlusion(LMO) datasets in Table 1. We also present additional results on T-Less [8] dataset in Table 3. **T-Less.** We evaluate our pipeline on T-Less dataset. The T-Less dataset comprises 30 objects with real training images. We train our model using relative camera poses and real training images. In Tab. 3 we report the AR of VSD, MSSD, MSPD metrics on the BOP challenge test set. We achieve closer to benchmark accuracy despite not using a CAD model. It shows that Nerf can learn accurate geometry and render correspondences which are usually extracted from the CAD model. SurfEmb performs better than our approach as their approach is tailored for symmetric objects and also employs an inference pipeline with 2.2s. However, the results compared to other regression-based, Dpod and DpodV2, show that our approach can perform equally better employing NeRF.

[*] The first two authors contributed equally to this work

Table 1. Comparisons with state-of-the-art methods on LMO. We report the Average Recall(%) of ADD(-S) without refinement. *real* denotes the same real data as LM. *syn* denotes self-generated synthetic data, and *pbr* denotes blender rendered synthetic data from BOP[9]. * denotes the symmetric objects. **Our-pose** denotes our results with accurate pose labels and **Our-weak** is with relative pose labels. *w/o NeRF* denotes our results using original PnP+RANSAC and *w/ NeRF* is our method with our NeRF-enabled PnP+RANSAC.

Object	PVNet	Single-Stage	HybridPose	GDR	SO-Net	GDR	SO-Net	Cai.	**Our-pose**	**Our-pose**	**Our-weak**	**Our-weak**
	[16]	[11]	[18]	[19]	[2]	[19]	[2]	[1]	*w/o NeRF*	*w/ NeRF*	*w/o NeRF*	*w/ NeRF*
CAD	*w/ CAD*					*w/ CAD*		*w/o CAD*			*w/o CAD*	
training	*real+syn*					*real+pbr*		*real*			*real*	
Ape	15.8	19.2	20.9	39.3	46.3	46.8	48.4	7.10	46.8	46.9	48.3	**49.7**
Can	63.3	65.1	75.3	79.2	81.1	**90.8**	85.8	40.6	79.1	86.2	81.4	86.4
Cat	16.7	18.9	24.9	23.5	18.7	**40.5**	32.7	15.6	20.7	27.1	28.8	26.9
Driller	65.7	69.0	70.2	71.3	71.3	**82.6**	77.4	43.9	58.9	65.8	60.4	66.2
Duck	25.2	25.3	27.9	44.4	43.9	46.9	**48.9**	12.9	25.3	29.9	32.8	36.9
Eggbox	50.2	52.0	52.4	**58.2**	46.6	54.2	52.4	46.4	19.6	24.9	22.8	24.4
Glue	49.6	51.4	53.8	49.3	63.3	75.8	**78.3**	51.7	61.0	66.3	69.8	70.9
Holep.	36.1	45.6	54.2	58.7	62.9	60.1	**75.3**	24.5	41.0	46.4	41.8	49.8
Mean	40.8	43.3	47.5	53.0	54.3	62.2	**62.3**	30.3	44.1	49.2	48.2	51.4(↑ 21.1)

Table 2. **LM** results in on ADD-10 metric. *denotes that the objects is symmetric and is evaluated in ADD-S. **Our-pose** denotes our results trained on 6D pose labels, and **Our-weak** denotes training on camera relative pose labels. **Ours-sam** denotes our results trained on 6D pose labels and segmentation masks extracted using SegmentAnything. *w/o NeRF* denotes our results using original PnP+RANSAC and *w/ NeRF* is our method with our NeRF-enabled PnP+RANSAC

Object	PVNet	CDPN	GDR	SO-Pose	LieNet	Cai.	**Ours-sam**	**Ours-pose**	**Our-pose**	**Our-weak**
	[16]	[14]	[19]	[2]	[3]	[1]	*w/o NeRF*	*w/o NeRF*	*w/ NeRF*	*w/ NeRF*
CAD	*w/ CAD*				*w/o CAD*					
Ape	43.6	64.4	-	-	38.8	52.9	50.1	69.4	89.1	**93.1**
Bvise	99.9	97.8	-	-	71.2	96.5	99.4	99.4	99.3	**99.6**
Cam	86.9	91.7	-	-	52.5	87.8	97.7	98.3	98.7	**98.9**
Can	95.5	95.9	-	-	86.1	86.8	98.7	97.8	99.1	**99.7**
Cat	79.3	83.8	-	-	66.2	67.3	77.2	77.8	97.1	**98.1**
Drill	96.4	96.2	-	-	82.3	88.7	99.1	99.6	97.4	**98.7**
Duck	52.6	66.8	-	-	32.5	54.7	57.4	69.7	90.3	**94.2**
Eggbox*	99.2	99.7	-	-	79.4	94.7	89.1	99.9	99.6	**99.9**
Glue*	95.7	**99.6**	-	-	63.7	100	98.9	91.9	98.1	99.3
Holep	81.9	85.8	-	-	56.4	75.4	90.3	89.4	94.3	**96.5**
Iron.	98.9	97.9	-	-	65.1	94.5	100	99.89	**98.1**	97.8
Lamp	99.3	97.9	-	-	89.4	96.6	98.7	99.8	97.9	**98.7**
Phone	92.4	90.8	-	-	65.0	89.2	90.2	94.8	96.4	**97.3**
Mean	86.3	89.9	93.7	96.0	65.2	82.9	88.3(↑ 5.4)	91.8(↑ 8.9)	96.6(↑ 13.7)	**97.8**(↑ 14.9)

Table 3. Comparisons with state-of-the-art methods on T-Less. We report the VSD, MSPD, MSSD, AR metrics as described in the BOP challenge without refinement. CAD refers to the approaches assuming that the CAD model is available for training

Approach	Dv2	SurfEmb	EP	CP	Dv2	CDPN	Ours
	[17]	[5]	[7]	[13]	[17]	[14]	
CAD	Y	Y	Y	Y	N	N	N
VSD	0.57	0.5		0.57	0.46	0.49	0.45
MSSD	0.62	0.53		0.59	0.49	0.67	0.49
MSPD	0.76	0.83	0.63	0.76	0.59	0.41	0.66
AR	0.65	0.62	0.47	0.64	0.51	0.37	0.54

References

[1] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3153–3163, 2020.

[2] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021.

[3] T Do, Trung Pham, Ming Cai, and Ian Reid. Real-time monocular object instance 6d pose estimation. 2019.

[4] Joseph Redmon Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. *Retrieved September*, 17:2018, 2018.

[5] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings, 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. *arXiv preprint arXiv:2004.00605*, 2020.

[8] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[9] Tomas Hodan and Antonin Melenovsky. Bop: Benchmark for 6d object pose estimation: `https://bop.felk.cvut.cz/home/`, 2019.

[10] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[11] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[13] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[14] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7678–7687, 2019.

[15] 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 *European conference on computer vision*, pages 405–421. Springer, 2020.

[16] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[17] Ivan Shugurov, Sergey Zakharov, and Slobodan Ilic. Dpodv2: Dense correspondence-based 6 dof pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[18] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

[19] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.