Pseudo-keypoint RKHS Learning for Self-supervised 6DoF Pose Estimation (Supplementary Material)

Yangzheng Wu and Michael Greenspan

RCVLab, Dept. of Electrical and Computer Engineering, Ingenuity Labs,
Queen's University, Kingston, Ontario, Canada
{y.wu, michael.greenspan}@queensu.ca

S.1 Overview

We document here some addition implementation details and results. The detailed structures of keypoint radial voting network M_v and Convolutional RKHS Adapter M_A are described in Sec. S.2 and shown in Figs. S.2 and S.1. Sec. S.3 describes the visualized radial pattern (shown in Fig. S.3) of the dragon object in TUDL, which inspired us to use a CNN (M_v) to simulate the voting process. The AR_{VSD}, AR_{MSSD}, AR_{MSPD}, and AR results of all datasets we used for all ablation studies are summarized in Sec. S.4 and listed in Fig. S.4 and Tables S.1, S.2, S.3, and S.4. The detailed ADD(S) results for each category of object on LM and LMO are listed in Tables S.6 and S.5, and ADD-S AUC results on YCB are shown in Table S.7.

S.2 Network Diagrams

The structure of Adapter network M_A is shown in Fig. S.1. Each conv block comprises 3×3 convolution, batch normalization, and ReLu layers. Inputs of M_A, *i.e.* x8s, x4s, x2s, upx2s, upx4s, and upx8s are feature maps extracted from each step of M_r. x8s stands for the 8th step of the encoding ResNet block, and upx8s is the 8th step of the decoding block, etc. Outputs of each column of conv blocks are concatenated and mapped into RKHS by a linear layer.

The keypoint radial voting network M_v structure is shown in Fig. S.2. The input of M_v is the inversed radial map $\widehat{V}_r^{-1}$, defined in the main paper. M_v comprises 5 convolution layers with a kernel size of 3 and stride of 1. A linear layer maps the feature map into the shape of $n{\times}4$ comprising $n{\times}2$ projected 2D keypoints K, n classification labels C, and n confidence scores S.

Estimated keypoints are organized into clusters based on geometric constraints when multiple instances of the same object appear within an image. More precisely, the estimated keypoints are grouped into instance sets by sorting the mean absolute differences of the Euclidean distances between keypoints defined on the CAD model, and those being estimated.

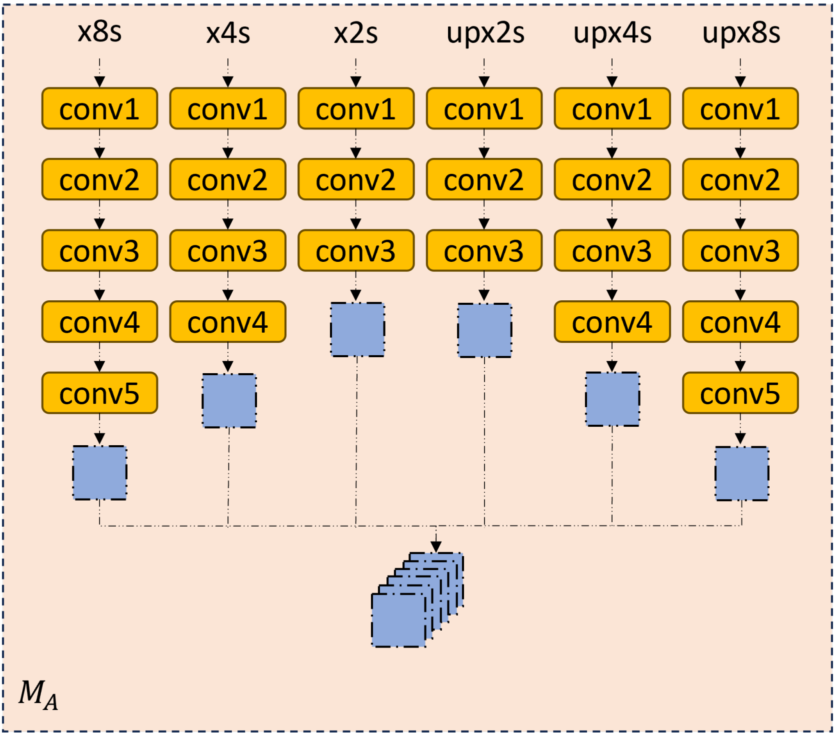


Fig. S.1: Convolutional RKHS Adapter M_A detailed structure.

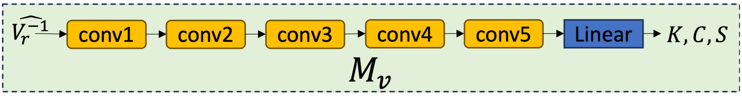
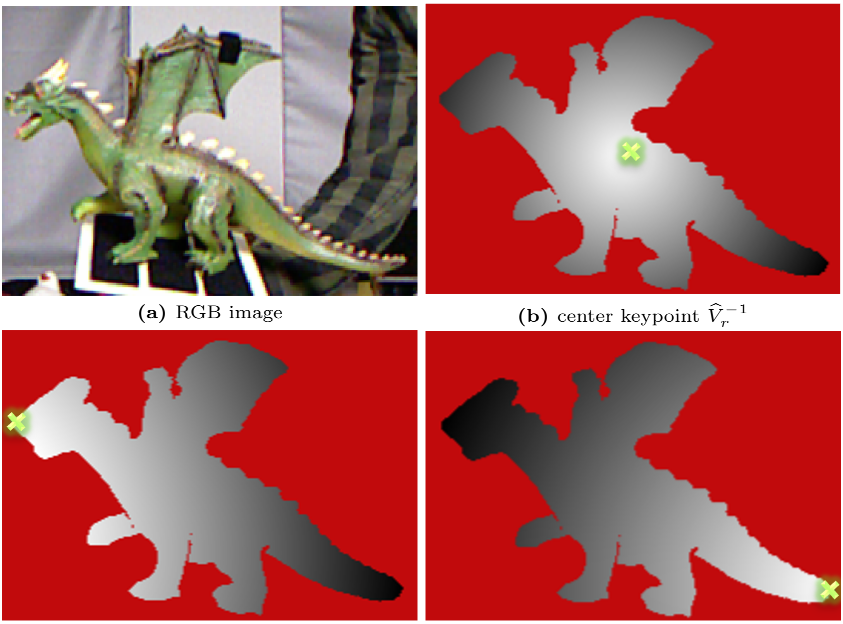


Fig. S.2: Keypoint radial voting network M_v detailed structure. Each conv block comprises a 3×3 convolution, batch normalization, and ReLu layer. The final linear layer reshapes the feature map into the output K, C, S.

S.3 Radii Pattern for Keypoint Voting

As shown in Fig. S.3, the estimated radial map $\widehat{V}_r$ is an inverse heat map of the candidate keypoints' locations , distributed in a radial pattern centered at the keypoints. The further away from the pixel to the keypoint, $\widehat{V}_r$ has a greater value. Pixels with value -1 do not lie on an object. This inspired us to use a CNN (M_v) to detect the peak, thereby localizing the keypoint.



(a) RGB image **(b)** center keypoint $\widehat{V}_r^{-1}$

(c) left keypoint $\widehat{V}_r^{-1}$ **(d)** right keypoint $\widehat{V}_r^{-1}$

Fig. S.3: TUDL [S.3] dataset dragon object (a) RGB image and (b)-(d) inverse radial maps $\widehat{V}_r^{-1}$. The segmentation mask is applied to filter out the background, shown in red. Keypoints (pink stars) are located at the (b) center, (c) left, and (d) right.

S.4 Ablation Studies

The AR_{VSD}, AR_{MSSD}, AR_{MSPD}, and AR results for all datasets used for all ablation studies are provided here.

Table S.1 shows the dataset-wise performance when sparse M_A^S and dense M_A RKHS Adapters are used to reduce the sim2real domain gap. M_A performs better on all datasets than M_A^S.

Table S.2 demonstrates the impact of different kernels used in M_A including linear and RBF kernels, both with and without trainable weights. M_A with trainable linear kernel performs the best compared to others.

Table S.3 compares different RKHS metrics including MMD, KL Div, and Wass Distances. M_A evaluated by MMD outperforms the other metrics.

Table S.4 illustrates the training strategies of the regression and voting model M_{rv} and the adapter M_A, whether sequentially or mixed. The mixed training is shown to lead to better (+2.5%) performance.

Lastly, the correlation between object size and accuracy is also evaluated on LM, LMO, and YCB datasets, and the results are shown in Fig. S.4. The object size impact on the accuracy has no significant impact except when there are extreme occlusions in LMO.

Table S.1: The impact of sparse (M_A^s) and dense (M_A) Adapters, defined in Sec. 5.1 in the main paper, on LM and LMO, and five BOP core datasets.

Adapter	Dataset	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
M_A	LM	96.0	95.7	95.7	95.8
	LMO	68.7	68.2	68.2	68.4
	TLESS	85.9	85.1	85.8	85.6
	TUDL	97.6	96.4	95.4	96.2
	ITODD	69.2	68.5	68.4	68.7
	HB	92.7	91.6	92.5	92.3
	YCB	83.9	83.4	84	83.8
	average	84.9	84.1	84.3	**84.4**
M_A^s	LM	93.4	92.7	92.6	92.9
	LMO	59.7	59.3	59.2	59.4
	TLESS	78.8	78.7	78.8	78.8
	TUDL	95.6	95.3	95.5	95.5
	ITODD	56.7	56.7	56.5	56.6
	HB	85.7	85.3	85.6	85.5
	YCB	76.5	76.4	76.6	76.5
	average	78.1	77.8	77.8	77.9

Table S.2: The impact of different RKHS kernels defined in Sec. 5.3 in the main paper on LM and five BOP core datasets.

Kernels	Trainable Weights (w)	Dataset	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
Linear	✗	LM	85.4	84.2	84.4	84.7
		LMO	56.9	56.3	55.4	56.2
		TLESS	74.3	73.8	74.2	74.1
		TUDL	88.2	87.3	84.3	86.6
		ITODD	45.2	44.7	45.3	45.1
		HB	79.3	78.6	79.2	79.0
		YCB	72.2	70.5	71.3	71.3
		average	71.6	70.8	70.6	71.0
RBF	✗	LM	85.3	84.1	84.3	84.6
		LMO	57.1	56.3	55.2	56.2
		TLESS	73.7	73.1	73.6	73.5
		TUDL	90.3	89.7	89.9	90.0
		ITODD	52.7	51.9	52.5	52.4
		HB	80.2	79.6	79.3	79.7
		YCB	72.6	71.8	71.5	72.0
		average	73.4	72.9	73.2	73.2
Linear	✓	LM	96.0	95.7	95.7	95.8
		LMO	68.7	68.2	68.2	68.4
		TLESS	85.9	85.1	85.8	85.6
		TUDL	97.6	96.4	95.4	96.2
		ITODD	69.2	68.5	68.4	68.7
		HB	92.7	91.6	92.5	92.3
		YCB	83.9	83.4	84	83.8
		average	84.9	84.1	84.3	**84.4**
RBF	✓	LM	94.7	94.5	94.4	94.5
		LMO	57.1	56.3	55.2	56.2
		TLESS	83.7	82.9	83.3	83.3
		TUDL	97.4	95.8	94.9	95.1
		ITODD	66.7	63.6	64.2	64.7
		HB	85.6	83.9	84.8	84.8
		YCB	82.8	82.6	82.8	82.7
		average	82.5	81.3	81.5	81.6

Table S.3: The impact of different metrics defined in Sec. 5.3 in the main paper for M_A on LM and five BOP core datasets.

Metrics	Dataset	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
MMD	LM	96.0	95.7	95.7	95.8
	LMO	68.7	68.2	68.2	68.4
	TLESS	85.9	85.1	85.8	85.6
	TUDL	97.6	96.4	95.4	96.2
	ITODD	69.2	68.5	68.4	68.7
	HB	92.7	91.6	92.5	92.3
	YCB	83.9	83.4	84	83.8
	average	84.9	84.1	84.3	**84.4**
KL Div	LM	90.2	90.3	90.5	90.3
	LMO	63.4	62.2	63.6	63.1
	TLESS	79.8	79.7	80.2	79.9
	TUDL	93.4	93.2	92.9	93.2
	ITODD	62.1	62.2	62.1	62.1
	HB	84.6	84.3	84.7	84.5
	YCB	72.6	72.5	72.3	72.4
	average	78.0	77.8	78.0	77.9
Wass Distance	LM	92.3	91.7	92.2	92.1
	LMO	64.7	64.2	64.7	64.5
	TLESS	82.3	82.2	81.9	82.1
	TUDL	95.7	95.6	95.5	95.6
	ITODD	65.7	64.9	65.6	65.4
	HB	89.3	89.1	89.5	89.3
	YCB	76.5	76.3	76.6	76.4
	average	80.9	80.6	80.9	80.8

Table S.4: The impact of different training strategies described in Sec. 5.2 in the main paper on LM and five BOP core datasets.

Training Type	Dataset	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
Mixed	LM	96.0	95.7	95.7	95.8
	LMO	68.7	68.2	68.2	68.4
	TLESS	85.9	85.1	85.8	85.6
	TUDL	97.6	96.4	95.4	96.2
	ITODD	69.2	68.5	68.4	68.7
	HB	92.7	91.6	92.5	92.3
	YCB	83.9	83.4	84	83.8
	average	84.9	84.1	84.3	**84.4**
Sequential	LM	95.4	95.2	95.3	95.3
	LMO	65.3	65.2	65.2	65.2
	TLESS	85.7	85.3	85.6	85.5
	TUDL	97.6	96.4	95.4	96.2
	ITODD	63.4	62.2	62.3	62.6
	HB	88.6	88.4	88.3	88.4
	YCB	79.8	79.5	80.0	79.8
	average	82.3	81.7	81.7	81.9

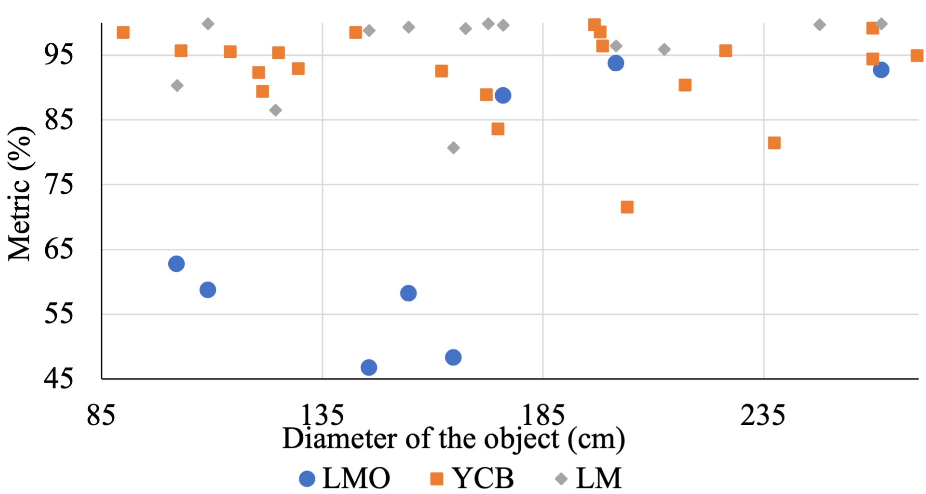


Fig. S.4: Impact of object size (diameter) on accuracy, for three datasets (LM, LMO, and YCB). There is no noticeable performance change for different object sizes, except for small heavily occluded objects in the LMO dataset.

S.5 Category Wise Performance

The category-wise ADD(S), ADD-S AUC, and ADD(S) AUC results for LINEMOD-Occlusion, LINEMOD, and YCB-Video are shown in Tables S.5, S.6, and Table S.7. RKHSPose outperforms on average against all self-supervised methods and most of the object categories.

Table S.5: LMO accuracy results of self-supervised 6DOF PE methods: accuracy of RKHSPose for non-symmetric objects is evaluated with ADD, and for symmetric objects (annotated with *) is evaluated with ADD-S. All the syn + real image methods use real images without GT labels. The methods annotated with ** are the TexPose [S.1] re-implementation.

Mode	Method	Object								Mean
		ape	can	cat	driller	duck	eggbox*	glue*	hole-puncher	
syn	GDR [S.12]	44	83.9	49.1	88.5	15	33.9	75	34	52.9
syn + real images	Self6D [S.11]	13.7	43.2	18.7	32.5	14.4	**57.8**	54.3	22	32.1
	Sock et al. [S.7]	12	27.5	12	20.5	23	25.1	27	35	22.8
	DSC [S.14]	9.1	21.1	26	33.5	12.2	39.4	37	20.4	24.8
	SMOC-Net [S.9]	60.0	<u>94.5</u>	<u>59.1</u>	**93.0**	37.2	48.3	**89.3**	25.0	63.3
	Self6D++ ** [S.1, S.10]	59.4	**96.5**	**60.8**	92	30.6	<u>51.1</u>	88.6	38.5	64.7
	TexPose [S.1]	<u>60.5</u>	93.4	56.1	92.5	<u>55.5</u>	46	82.8	<u>46.5</u>	66.7
	Ours	**62.7**	93.5	58.2	92.5	**58.7**	48.2	<u>88.7</u>	<u>46.5</u>	<u>68.6</u>
	Ours+ICP	**62.7**	93.7	58.2	<u>92.7</u>	**58.7**	48.3	<u>88.7</u>	**46.7**	**68.7**

Table S.6: LINEMOD Accuracy Results of self-supervised 6DOF PE methods: Non-symmetric objects are evaluated with ADD, and symmetric objects (annotated with *) are evaluated with ADD-S . All the syn + real image methods use real images without GT labels. The methods annotated with ** are the TexPose [S.1] re-implementations. DeepIM annotated with # is the self6D++ re-implementation.

		Object													
Mode	Method	ape	bench-vise	camera	can	cat	driller	duck	eggbox*	glue*	hole-puncher	iron	lamp	phone	mean
syn	AAE [S.8]	4.0	20.9	30.5	35.9	17.9	24.0	4.9	81.0	45.5	17.6	32.0	60.5	33.8	31.4
	MHP [S.5]	11.9	66.2	22.4	59.8	26.9	44.6	8.3	55.7	54.6	15.5	60.8	-	34.4	38.8
	DeepIM# [S.4, S.10]	85.8	93.1	99.1	**99.8**	98.7	**100.0**	61.9	93.5	93.3	32.1	**100.0**	99.1	94.8	88.0
syn + real image	DSC [S.14]	31.2	83.0	49.6	56.5	57.9	73.7	31.3	96.0	63.4	38.8	61.9	64.7	54.4	58.6
	Self6D [S.11]	38.9	75.2	36.9	65.6	57.9	67.0	19.6	99.0	94.1	16.2	77.9	68.2	50.1	58.9
	GDR** [S.1, S.12]	85.0	**99.8**	96.5	99.3	93.0	**100.0**	65.3	**99.9**	98.1	73.4	86.9	**99.6**	86.3	91.0
	Sock et al. [S.7]	37.6	78.6	65.6	65.6	52.5	48.8	35.1	89.2	64.5	41.5	80.9	70.7	60.5	60.6
	Self6D++** [S.1, S.10]	75.4	94.9	97.0	99.5	86.6	98.9	68.3	99.0	96.1	41.9	99.4	98.9	94.3	88.5
	SMOC-Net [S.9]	85.6	96.7	97.2	99.9	95.0	**100.0**	76.0	98.3	99.2	45.6	99.9	98.9	94.0	91.3
	TexPose [S.1]	80.9	99	94.8	99.7	92.6	97.4	83.4	94.9	93.4	79.3	99.8	98.3	78.9	91.7
	DPODv2 [S.6]	80.0	99.7	**99.2**	99.6	95.1	98.9	79.5	99.6	99.8	72.3	99.4	96.3	96.8	93.5
	Ours	90.2	99.7	99.1	**99.8**	96.2	99.2	86.3	99.8	**99.8**	80.3	99.6	98.8	**97.2**	95.8
	Ours+ICP	**90.3**	99.7	99.1	**99.8**	**96.4**	99.3	**86.5**	99.8	**99.8**	**80.7**	99.6	98.8	**97.2**	**95.9**
syn + real labels	SO-Pose [S.2]	-	-	-	-	-	-	-	-	-	-	-	-	-	96.0
	Ours	92.3	99.7	**99.3**	**99.8**	**97.5**	99.2	90.2	99.8	**99.8**	84.3	99.6	98.8	97.2	96.7
	Ours+ICP	**92.7**	99.7	**99.3**	**99.8**	**97.5**	99.2	**90.7**	99.8	**99.8**	**84.5**	99.6	98.8	**97.3**	**96.8**

Table S.7: YCB ADD-S and ADD(S) AUC [S.13] results of self-supervised 6DoF PE methods: Non-symmetric objects are evaluated with ADD AUC, and symmetric objects (annotated with *) are evaluated with ADD-S AUC. To the best of our knowledge, self6d++ [S.10] is the only self supervision method that provided YCB results.

Mode	Metric	Method	002 master chef can	003 cracker box	004 sugar box	005 tomato soup can	006 mustard bottle	007 tuna fish can	008 pudding box	009 gelatin box	010 potted meat can	011 banana	019 pitcher base	021 bleach cleanser	024 bowl*	025 mug	035 power drill	036 wood block*	037 scissors	040 large marker	051 large clamp*	052 extra large clamp*	061 foam brick*	Mean
syn+ real images	ADD-S AUC	Self6D++ [S.10]	88.8	94.2	95.8	90.8	98.6	97.5	98.4	94.0	89.3	98.5	98.9	93.5	89.1	94.1	95.2	78.3	69.2	87.5	79.2	87.3	95.5	91.1
		Ours	88.7	94.7	96.2	92.2	99.5	98.2	98.3	95.2	92.7	98.4	99.1	94.2	92.3	95.2	95.5	81.2	71.3	89.2	83.4	90.2	95.5	92.4
		Ours+ICP	**88.9**	**94.9**	**96.4**	**92.3**	**99.7**	**98.5**	**98.5**	**95.5**	**92.9**	**98.6**	**99.2**	**94.4**	**92.5**	**95.4**	**95.7**	**81.4**	**71.5**	**89.4**	**83.6**	**90.4**	**95.7**	**92.6**
	ADD(S) AUC	Self6D++ [S.10]	8.4	84.9	88.0	79.4	92.7	89.7	93.9	83.9	75.7	91.8	92.1	84.5	89.1	81.4	84.2	78.3	45.2	74.6	79.2	87.3	95.5	80.0
		Ours	13.7	86.2	91.3	83.2	92.7	92.3	94.3	84.2	**76.3**	93.7	94.3	86.0	92.3	83.2	86.3	81.2	62.3	75.6	83.4	90.2	95.5	82.8
		Ours+ICP	**13.8**	**86.5**	**91.5**	**83.3**	**93.6**	**92.5**	**94.5**	**84.5**	**76.3**	**93.9**	**94.5**	**86.1**	**92.5**	**83.4**	**86.5**	**81.4**	**62.5**	**75.7**	**83.6**	**90.4**	**95.7**	**83.0**
syn+ real labels	ADD-S AUC	Self6D++ [S.10]	93.8	98.8	**99.6**	95.4	**100.0**	99.9	63.3	92.9	91.1	93.0	99.3	91.2	87.2	**96.4**	99.7	68.6	78.9	93.0	81.7	86.9	94.3	90.7
		Ours	95.4	98.8	99.2	96.3	99.6	99.8	67.2	93.5	94.3	95.2	99.5	93.7	92.3	96.3	99.6	71.2	81.2	94.3	84.2	90.2	94.8	92.2
		Ours+ICP	**95.7**	99.0	99.5	**96.5**	99.8	**100.0**	**67.5**	**93.7**	**94.5**	**95.5**	**99.7**	**93.9**	**92.5**	**96.4**	**99.8**	**71.4**	**81.4**	**94.4**	**84.3**	**90.4**	**95.2**	**92.4**
	ADD(S) AUC	Self6D++ [S.10]	56.7	92.8	95.0	90.5	94.7	97.0	42.1	84.7	78.2	80.5	98.7	81.9	87.2	86.6	93.6	68.6	61.3	81.7	81.7	86.9	94.3	82.6
		Ours	62.3	95.3	94.9	93.2	95.2	97.0	50.2	87.2	81.2	83.2	99.1	83.2	92.3	87.2	95.2	71.2	74.3	82.7	84.2	90.2	94.8	85.4
		Ours+ICP	**62.7**	**95.6**	**95.2**	**93.4**	**95.5**	**97.2**	**50.5**	**87.5**	**81.4**	**83.3**	**99.2**	**83.4**	**92.5**	**87.3**	**95.4**	**71.4**	**74.4**	**82.8**	**84.3**	**90.4**	**95.2**	**85.6**

References

[S.1] Chen, H., Manhardt, F., Navab, N., Busam, B.: Texpose: Neural texture learning for self-supervised 6d object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4841–4852 (2023) 8, 9

[S.2] Di, Y., Manhardt, F., Wang, G., Ji, X., Navab, N., Tombari, F.: So-pose: Exploiting self-occlusion for direct 6d pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12396–12405 (October 2021) 9

[S.3] Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European conference on computer vision (ECCV). pp. 19–34 (2018) 3

[S.4] Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: Deepim: Deep iterative matching for 6d pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 683–698 (2018) 9

[S.5] Manhardt, F., Arroyo, D.M., Rupprecht, C., Busam, B., Birdal, T., Navab, N., Tombari, F.: Explaining the ambiguity of object detection and 6d pose from visual data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 9

[S.6] Shugurov, I., Zakharov, S., Ilic, S.: Dpodv2: Dense correspondence-based 6 dof pose estimation. IEEE transactions on pattern analysis and machine intelligence **44**(11), 7417–7435 (2021) 9

[S.7] Sock, J., Garcia-Hernando, G., Armagan, A., Kim, T.K.: Introducing pose consistency and warp-alignment for self-supervised 6d object pose estimation in color images. In: 2020 International Conference on 3D Vision (3DV). pp. 291–300. IEEE (2020) 8, 9

[S.8] Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3d orientation learning for 6d object detection from rgb images. In: Proceedings of the european conference on computer vision (ECCV). pp. 699–715 (2018) 9

[S.9] Tan, T., Dong, Q.: Smoc-net: Leveraging camera pose for self-supervised monocular object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21307–21316 (2023) 8, 9

[S.10] Wang, G., Manhardt, F., Liu, X., Ji, X., Tombari, F.: Occlusion-aware self-supervised monocular 6d object pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 8, 9, 10

[S.11] Wang, G., Manhardt, F., Shao, J., Ji, X., Navab, N., Tombari, F.: Self6d: Self-supervised monocular 6d object pose estimation. In: European Conference on Computer Vision. pp. 108–125. Springer (2020) 8, 9

[S.12] Wang, G., Manhardt, F., Tombari, F., Ji, X.: 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. pp. 16611–16621 (2021) 8, 9

[S.13] Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes (2018) 10

[S.14] Yang, Z., Yu, X., Yang, Y.: Dsc-posenet: Learning 6dof object pose estimation via dual-scale consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3907–3916 (2021) 8, 9