Sim-to-Real 6D Object Pose Estimation via Iterative Self-training for Robotic Bin Picking Supplementary Materials

Kai Chen[1], Rui Cao[1], Stephen James[2], Yichuan Li[1],
Yun-Hui Liu[1], Pieter Abbeel[2], and Qi Dou[1]

[1] The Chinese University of Hong Kong
[2] University of California, Berkeley

In this supplementary material, we provide additional contents that are not included in the main paper due to the space limit:

- Training details of the proposed self-training method for sim-to-real 6D object pose estimation (Section A)
- More details about ROBI dataset and SIBP dataset (Section B)
- Results with different amount of real data for self-training (Section C)
- More qualitative comparison results (Section D)

A. Details of Network Output and Loss Function

Our proposed iterative self-training method for sim-to-real object pose estimation is designed to be scalable and network-agnostic. The specific output format for representing the 6D object pose and the loss function are dependent on the architecture of the adopted object pose estimation backbone network. In our experiments, we mainly take DC-Net [9] as the backbone network to test the performance of the proposed self-training method. In this case, the network output and the loss function used for self-training are similar with the ones proposed in [9].

For the rotation, we use discrete-continuous formulation for regressing rotation. Specifically, based on the icosahedral group, we generate 60 rotation anchors to uniformly sample the whole $SO(3)$ space. For each rotation anchor, the network would predict a rotation deviation in the form of quaternion and an uncertainty value σ to indicate the confidence of each rotation anchor. For the translation, the network would predict a unit vector v' for each input point that represents the direction from the point to the object center. Then, the translation is estimated based on a RANSAC-based voting. The loss function used for rotation estimation is based on the ShapeMatch-Loss [10]. For symmetric objects, it is defined as:

$$L = \frac{1}{M} \sum_{x_1 \in \mathcal{M}} \min_{x_2 \in \mathcal{M}} \|\widetilde{R}x_1 - R'x_2\|_2, \tag{1}$$

where M denotes the total number of points of the object CAD model. $\widetilde{R}$ and R' denote rotations correspond to the pseudo label and the network output

respectively. For asymmetric objects, the loss function is defined as:

$$L = \frac{1}{M} \sum_{x \in \mathcal{M}} \|\widetilde{R}x - R'x\|_2. \tag{2}$$

The specific probabilistic loss used for rotation estimation that considers the uncertainty value predicted by the network is defined as:

$$L_R = \sum_i = \ln\sigma_i + \frac{L_i}{d \times \sigma_i}, \tag{3}$$

where d denotes the diameter of the object. In addition, we use smooth L1 loss for translation estimation:

$$L_t = \begin{cases} \sum_i \frac{1}{M} \sum_j 0.5 \times \|\widetilde{v_{ij}} - v'_{ij}\|_2^2, & \|\widetilde{v_{ij}} - v'_{ij}\| < 1.0 \\ \sum_i \frac{1}{M} \sum_j \|\widetilde{v_{ij}} - v'_{ij}\| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

where M denotes the total number of points of the object CAD model. $\widetilde{v}$ and v' denote unit vectors computed from the pseudo label and the network prediction respectively. We integrate loss functions for rotation estimation and translation estimation as in [9] for the network training.

B. Detailed Information about Experiment Dataset

As mentioned in the main paper, we extend ROBI with 7000 synthetic scenes based on the provided CAD models shown in Fig. 1. We further build the SIBP dataset. It provides both synthetic and real data for 6 objects with different materials and sizes. When synthesizing virtual data for both ROBI and SIBP, we follow [2,6] and leverage multiple strategies to narrow the sim-to-real gap: (a) add realism surface texture based on the provided texture-less CAD model; (b) simulate virtual cluttered scene under the physical constraints; (c) employ ray-tracing rendering engine in Blender [1] to generate photo-realistic images; (d) use copy-paste strategy to randomize background. Fig. 2 presents some examples of our extended ROBI dataset.

Fig. 3 depicts 6 objects that are adopted in constructing the SIBP dataset. We use a fixed industrial stereo camera to collect the required RGB-D data. To make the collected real data of SIBP closer to the practical bin-picking scenario, we follow the scheme in [5] to collect RGB-D data for each object. We first put tens of objects in the scene. Then, we carefully remove the objects from the scene one by one and keep the remaining objects unchanged. Fig. 4 depicts some examples of the SIBP dataset. Tab. 1 further provides detailed statistics information of SIBP.

In addition, before object pose estimation, we trained a Mask-RCNN [4] with the synthetic data for instance segmentation. Thanks to photo-realistic

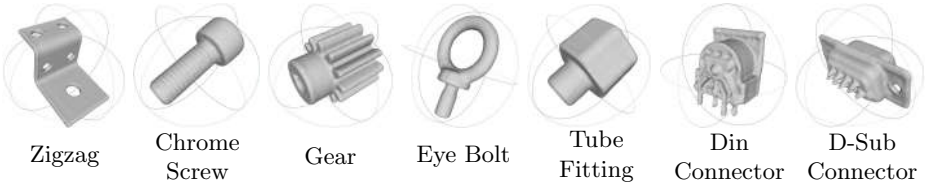

Fig. 1. Seven object models provided by ROBI [11]. Zigzag, Din Connector, and D-Sub Connector are three asymmetric objects, and the remains are symmetric objects. Din Connector and D-Sub Connector are composed of two different materials.

Fig. 2. Examples of ROBI dataset. The first row presents the synthetic data. The second and third rows show real data in low-bin and full-bin scenarios.

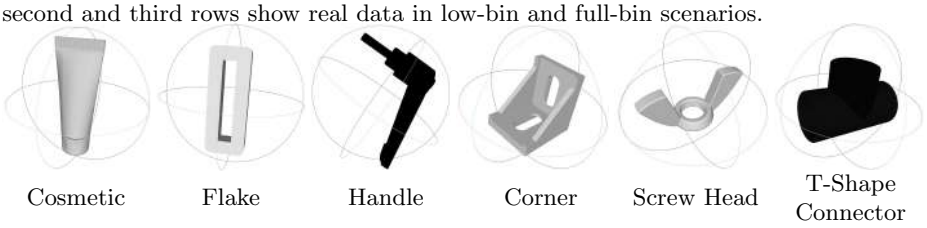

Fig. 3. Six objects used in SIBP. Handle is an asymmetric object, and remains are symmetric objects.

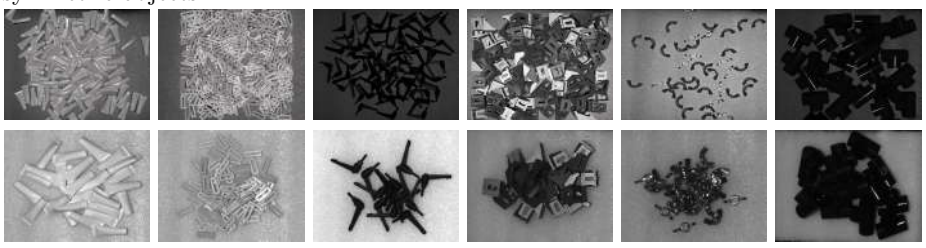

Fig. 4. Examples of SIBP dataset. The first row presents the synthetic data. The second row shows the real data.

Table 1. Statistics information of SIBP dataset.

	Cosmetic	Flake	Handle	Corner	Screw Head	T-Shape Connector
Diameter (mm)	67.9	34.5	77.4	66.6	54.5	108.4
Surface material	plastic	plastic	metallic	alloyed	metallic	plastic
Geometric symmetry	yes	yes	no	yes	yes	yes
Synthetic data	1000	1000	1000	1000	1000	1000
Real data	529	588	418	480	449	279

Fig. 5. Segmentation results on ROBI with a Mask-RCNN trained on synthetic data.

Fig. 6. Segmentation results on SIBP with a Mask-RCNN trained on synthetic data.

and physically plausible rendering techniques, as well as many off-the-shelf but effective 2D augmentation techniques [3], the predicted instance masks on real data could be very accurate to be used for self-training. Fig. 5 and Fig. 6 depict some qualitative examples for instance segmentation.

C. Self-training with Different Amounts of Real Data

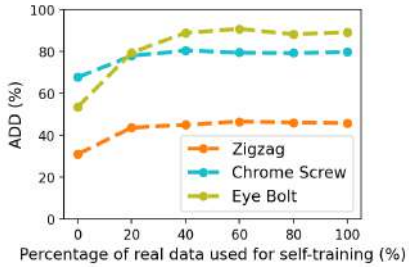

Fig. 7. Self-training results with different amount of real data for self-training.

We further studied the self-training performance with different amounts of real data on three objects of ROBI. We trained the object pose estimation network on 20%, 40%, 60%, 80% and 100% of training data with the proposed self-training method, and tested the resulting model on the same testing dataset. Fig. 7 presents the experiment results. With only 20% real data, the self-training model has significantly outperformed the model trained with only synthetic data. With about 40% real data, the model achieved comparable performance with the model trained with 100% real data. These results demonstrate the data efficiency of our proposed self-training method for sim-to-real object pose estimation in industrial bin-picking scenario.

D. More Qualitative Results

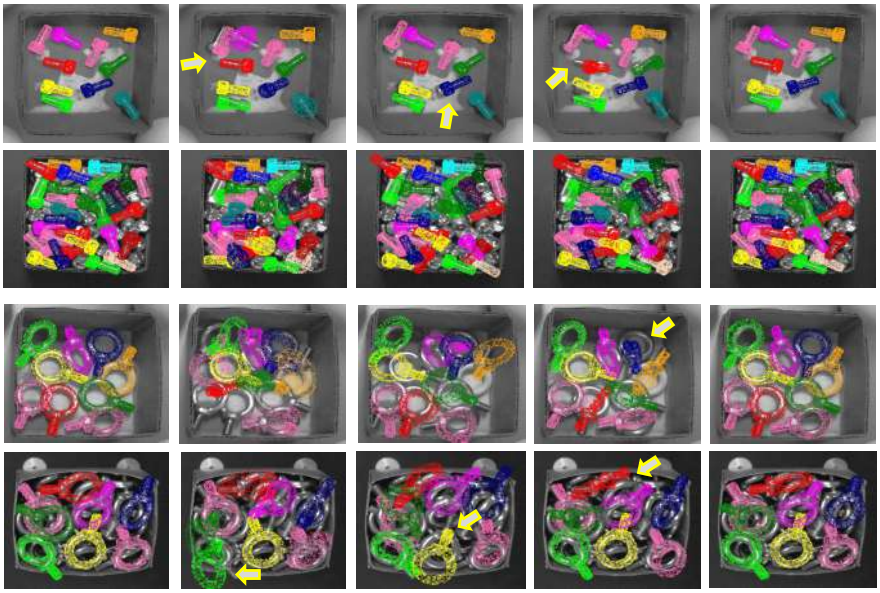

Fig. 8. More qualitative comparisons with state-of-the-art methods on ROBI Dataset.

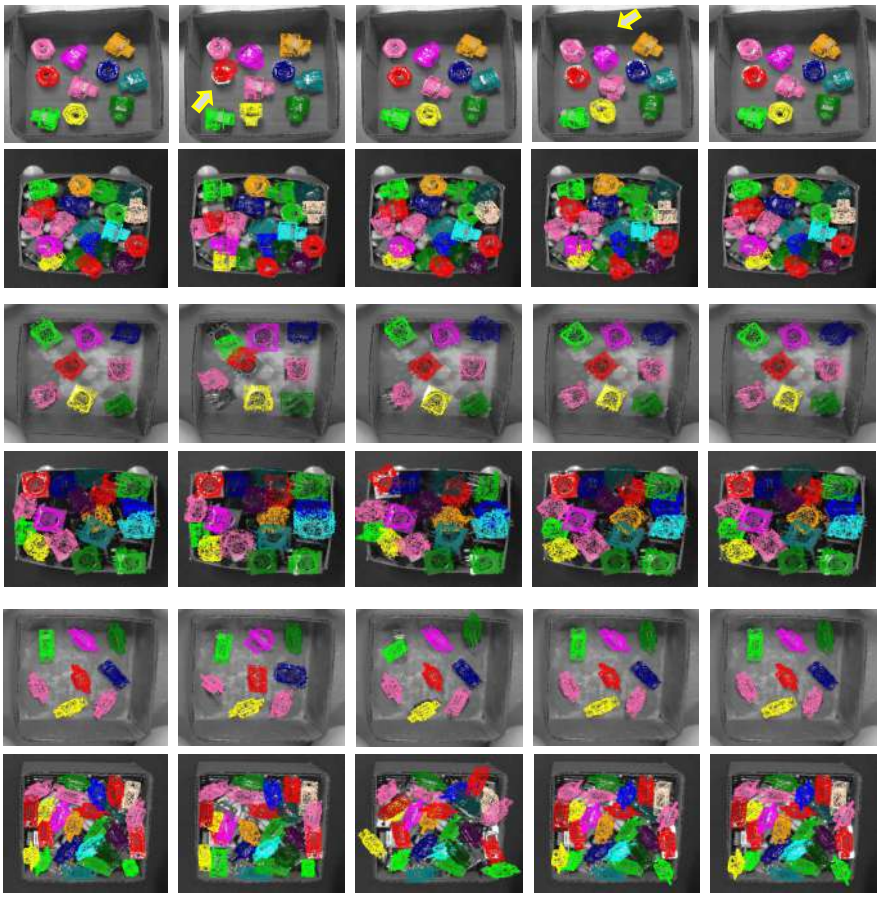

Fig. 9. More qualitative comparisons with state-of-the-art methods on ROBI Dataset.

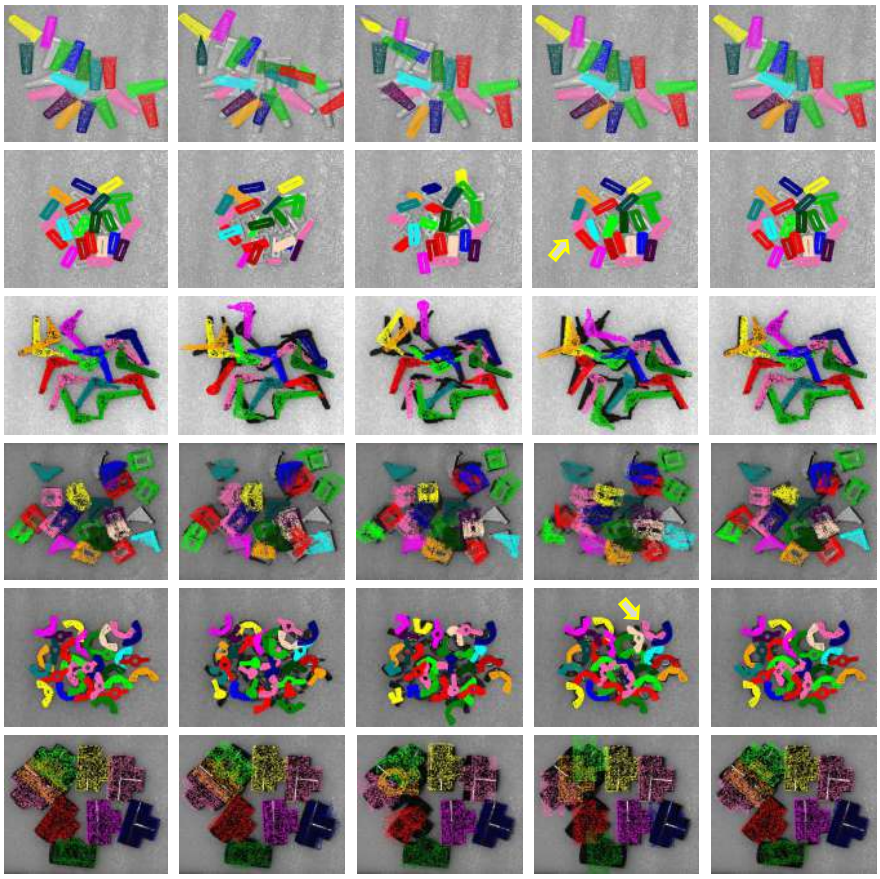

Fig. 10. More qualitative comparisons with state-of-the-art methods on SIBP Dataset.

References

1. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), http://www.blender.org 2
2. Denninger, M., Sundermeyer, M., Winkelbauer, D., Olefir, D., Hodan, T., Zidan, Y., Elbadrawy, M., Knauer, M., Katam, H., Lodhi, A.: Blenderproc: Reducing the reality gap with photorealistic rendering. In: RSS (2020) 2
3. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: CVPR (2021) 4
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 2
5. Kleeberger, K., Landgraf, C., Huber, M.F.: Large-scale 6d object pose estimation dataset for industrial bin-picking. In: IROS (2019) 2
6. Li, X., Cao, R., Feng, Y., Chen, K., Yang, B., Fu, C.W., Li, Y., Dou, Q., Liu, Y.H., Heng, P.A.: A sim-to-real object recognition and localization framework for industrial robotic bin picking. RAL (2022) 2
7. Sundermeyer, M., Durner, M., Puang, E.Y., Marton, Z.C., Vaskevicius, N., Arras, K.O., Triebel, R.: Multi-path learning for object pose estimation across domains. In: CVPR (2020) 5, 6, 7
8. Sundermeyer, M., Marton, Z.C., Durner, M., Triebel, R.: Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. IJCV (2020) 5, 6, 7
9. Tian, M., Pan, L., Ang, M.H., Lee, G.H.: Robust 6d object pose estimation by learning rgb-d features. In: ICRA (2020) 1, 2, 5, 6, 7
10. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In: RSS (2018) 1
11. Yang, J., Gao, Y., Li, D., Waslander, S.L.: Robi: A multi-view dataset for reflective objects in robotic bin-picking. In: IROS (2021) 3