IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 28, 2021, accepted May 18, 2021, date of publication May 24, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3082947

Calibration-Free Monocular Vision-Based Robot Manipulations With Occlusion Awareness

YONGLE LUO[1], KUN DONG[1], LILI ZHAO[1], ZHIYONG SUN[1], (Member, IEEE), ERKANG CHENG[1], (Member, IEEE), HONGLIN KAN[1], CHAO ZHOU[2,3], AND BO SONG[1], (Member, IEEE)

[1]Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China
[2]Institute of Plasma Physics, Chinese Academy of Sciences, Hefei 230031, China
[3]Faculty of Science and Technology, University of Twente, 7522 NB Enschede, The Netherlands

Corresponding author: Bo Song (songbo@iim.ac.cn)

This work was supported in part by the Key Research Development Program (KRDP) of Anhui Province under Grant 201904a05020086; in part by the NSFC under Grant 61804100, Grant 61973294, and Grant 61806181; and in part by the Chinese Academy of Science (CAS) under Grant GJTD-2018-15.

ABSTRACT Vision-based manipulation has been largely used in various robot applications. Normally, in order to obtain the spatial information of the operated target, a carefully calibrated stereo vision system is required. However, it limits the application of robots in the unstructured environment which limits both the number and the pose of the camera. In this study, a calibration-free monocular vision-based robot manipulation approach is proposed based on domain randomization and deep reinforcement learning (DRL). Firstly, a learning strategy combined domain randomization is developed to estimate the spatial information of the target from a single monocular camera arbitrarily mounted in a large area of the manipulation environment. Secondly, to address the monocular occlusion problem which regularly happens during robot manipulations, an occlusion awareness DRL policy has been designed to control the robot to avoid occlusions actively in the manipulation tasks. The performance of our method has been evaluated on two common manipulation tasks, reaching and lifting of a target building block, which show the efficiency and effectiveness of our proposed approach.

INDEX TERMS Monocular vision, reinforcement learning, reward shaping, robot manipulation.

I. INTRODUCTION

Obtaining the position of the target is the cornerstone of the general robotic manipulation for either regular motion planning or the modern learning-based operations such as the reinforcement learning approaches [1]. One of the most prevalent methods for target location estimation is the class of visual perception approaches. The traditional way for obtaining the 3D position of targets in pixel is to directly calculate the target location via the transition between the pixel coordinate and the 3D cartesian coordinates, precision of which depends heavily on the camera calibration [2]. To implement this type of method, one has to tell where the target is in advance through additional object detection methods, which is indirect and might be inefficient for practical implementations. Another direct way for the target position estimation is to regress the location of the target directly based on a given image containing the target by deep learning (DL) techniques. One challenge is that both of the mentioned methods are only applicable to the scenario that the cameras are fixed throughout the entire strategy-development-and-implementation process. Once the camera is moved and settled again, the calibration-based method has to repeat the calibration process, and the regression-based method has to collect new data from the altered camera perspective for the training purpose, this is because the target-location estimator trained with the data collected from a specific camera perspective usually performs poor when the camera is remounted to a new location. To tackle the camera remounting problem, one solution is to collect the data from various camera perspectives, which is apparently costly and time consuming. Instead, one can benefit from utilizing the data augmentation technique through simulation to produce a large amount of diverse data to train the DL-based estimator for obtaining target locations. However, this will introduce the problem of ''reality gap'' due to the domain difference between a simulation environment and a real scene.

Domain randomization [3] is a prevalent method to solve the sim-to-real problem by increasing the diversity of the simulated scenarios to cover the real scene. However, it is

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang.

pretty difficult to achieve satisfactory model transfer from a simulation setup to a real scene without significant expertise and tremendous manual adjustment to config the simulation parameters [4]. It is also noted that even with elaborate configuration of simulation parameters, the transfer errors are still inevitable. To tackle this problem, this paper proposes a calibration-free method that relaxes the requirement of the domain randomization, and further improves the accuracy of a transferred model (DL-based estimator) trained based on simulation data to a real scene. The additional cost of the fine tuning procedure includes only 5-10 more real images and a few fine training steps (as shown in the left panel of Fig. 1). As the 3D position of the end-effector of a robotic arm can be obtained easily, the sampling of the real images containing target and location information can be done automatically via several pick-and-place operations.

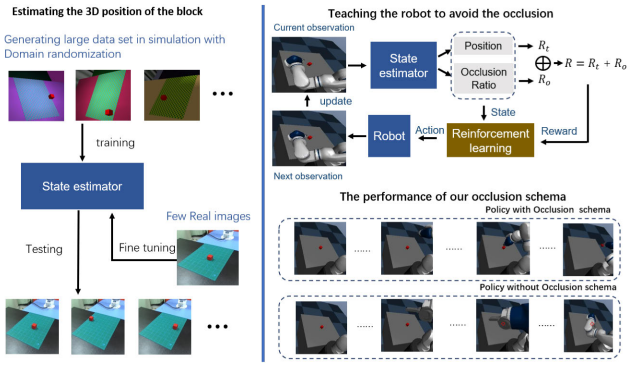


FIGURE 1. Left: Illustration of the framework for estimating 3D position of a target based on images from random camera angles, which utilizes a DL-based location estimator fine-tuned with a few real images. Right: The state estimator is used to extract the information such as the position and occlusion status of the target which are then feeded into the state and reward shaping of the DRL. The combined state-based DRL makes the agent learn a policy that avoids the self-occlusion actively, and thus complete the task more efficiently.

It is noted that fine-tuned state estimator can work well in the scenes that contain no unexpected objects. However, during robotic manipulations, one of the biggest challenges of applying a monocular camera to estimate states of a target is the self-occlusion which could be induced directly by the robot arm itself [5], [6]. In such a case, the predicted target position usually comes with large error especially when the target object is completely invisible in the camera's perspective [3]. To overcome this challenge, inspired by the human manipulation, which can avoid the vision self-occlusion by adjusting his or her action actively, this study proposes to incorporate the occlusion information into the deep reinforcement learning (DRL) policy to make the agent be able to aware and solve occlusion actively by itself (as shown in the right panel of Fig. 1). This paper firstly develops and validates the DL-based state estimator (without occlusions) systematically, and then establishes the DRL strategy with occlusion-awareness based on the developed state estimator. Evaluation of the entire DRL scheme is performed on two common robot manipulation tasks, reaching and lifting (with occlusions). The result shows that the DL-based state estimator can achieve a much higher accuracy compared to the raw randomization method. Besides, the manipulation results show that the DRL scheme with occlusion-awareness can achieve an excellent performance in the simulation environment. This study also evaluates the proposed scheme in the real robot manipulation tasks by transferring the state estimator from the simulation to the real scene. Though with a certain of accuracy reduction for the model transfer, the occlusion scheme can still complete the tasks with a high success rate.

There are two main contributions of this paper summarized as follows. Firstly, we propose an efficient DL-based state estimator that can estimate the 3D position of a target from random camera perspectives without camera calibration, which is based on the sim-to-real technique and fine-tuning approach with a few real scene images. Secondly, based on the state estimation method, the occlusion information that caused by the robot arm itself is taken into the DRL scheme through the estimated states and reward shaping during the policy training, which enables the robot to solve the self-occlusion problem during operations actively.

II. RELATED WORK

A. METHODS FOR TARGET POSE ESTIMATION AND MODEL TRANSFER

Pose estimation of target objects has been widely studied. A common method is based on the estimation of the corresponding relation between a 2D pixel and a 3D point. In this setting, traditional method [7]–[9] or learning-based method [10]–[12] are firstly used to extract the 2D features, and then the pose can be calculated with Perspective-n-Point (PnP) algorithms [13]. There are also methods that combine the object detection and 6D pose estimation together and directly regress the 6D object pose from RGB images [14], [15]. Most of those methods are tested on common datasets such as YCB [16] and T-LESS [17] which have rich label information. However, for a real scene, collecting enough labelled data to train the deep network is always costly. Besides, those methods can only work well in the scene that the camera is fixed throughout the entire strategy-development-and-implementation process, limiting applications in practices.

In contrast to the expensive real scene data collection and labelling scenario, a simulation environment can produce a large quantity of labelled data quickly and freely, and the camera can be simply mounted anywhere. However, it has been shown difficulties of directly applying the model trained using simulation data to a real scene for obtaining satisfactory results due to the "reality gap". To solve the sim-to-real problem, domain randomization is proposed and has been widely applied in vision system [18]–[20]. For example, in [18], a VAE-based affordance representation model is trained for following motion planning by random alteration of textures, clutter and lighting. Researchers also applied the domain

randomization to end-to-end policy training that directly maps the images to action, such as flying real quadrotor through indoor environments [19], lifting block [20] and manipulating a block from an initial configuration to a goal configuration [21]. Besides, domain randomization has been extended to dynamical system which is called dynamics randomization [22]. By randomizing the dynamics parameters such as link mass [22], static and dynamic friction [23], dynamic gap between a simulated robot and a real robot can be largely narrowed.

Although randomization is powerful for enhancing capability of the sim-to-real technique, it is still unclear how to select the form and parameters of those domain distributions [24], which requires significant expertise and tremendous manual adjustments [4]. If the choice is not good, the zero-shot performance in real scene may be poor. What's more, a generalized model that performs well across different domains might not exist [25]. Fine-tuning of a model that has been pretrained in ImageNet [26] can give well result in object recognition [27], but it requires a large amount of data for the fine tuning. Meta learning (MAML [28]) is another method that can improve the generalization ability of the model by utilizing a few data from a target domain, which has been widely applied in object detection [29], [30] and classification [31]. Inspired by the mentioned works, this paper proposes to combine real scene data-based fine tuning method with the domain randomization to relax the requirement of designing the simulation parameters to further improve the accuracy of the regular domain randomization. When performing this technique to estimate target location, we firstly construct the domain distribution with a large range of camera angles in the simulation environment and train a model based on the generated virtual data, and subsequently utilize a few number of real scene images to improve the model by fine-tuning or meta learning, which is capable of utilizing costly data efficiently.

B. VISION-GUIDED DRL METHODS FOR TACKLING ROBOTIC MANIPULATION WITH OCCLUSION

There is an increasing trend of studies that apply DRL to the decision-making problem from directly using the low-dimension physical characteristics [32]–[35], to the end-to-end policy training with high-dimension input (such as using images as states [36], [37]). Generally, compared with the high sample complexity of end-to-end policy training [38], [39], the policy trained with low-dimension information becomes more attractive. In manipulation, a common way of applying DRL is to firstly pre-train a representation model to convert the high-dimension information (such as images obtained from cameras) to the low-dimension information which serves as the state during the policy learning process. For instance, a representation model that combines the current state and target state is trained to get a target-driven visual navigation policy [39], [40]. In [41], a compact and multi-modal representation which combines the RGB image, force-torque and proprioception was pre-trained to get a peg insertion policy that is robust to external occlusion and perturbations. In [3], a detector that based on the VGG-16 [42] has been trained to regress 3D Cartesian coordinates of the target object. With the help of domain randomization, the model can keep a relatively high accuracy in a real scene by training in simulation. However, it also shows a large accuracy reduction when the occlusion is encountered. Actually, few attempts have been made to solve the occlusion problem in the area of reinforcement learning. A common method to tackle the occlusion is to apply the camera-in-hand configuration. For example, in [43] and [44], the camera is installed at the end of the robot, so that the robot will not shade the key information during moving. While the eye-in-hand method is good at solving occlusion problems, it is not easy for a commercial gripper to have a camera working with it, since the camera lens might be contaminated or even damaged in some scenarios. In this paper, we fixed the camera at a specific location to monitor the entire working area to guide robotic manipulations. It should also be noted that if the fixed-camera scheme performs well in our specific tasks, the eye-in-hand structure should be able to perform better.

Inspired by humanoid manipulation, active occlusion avoidance should be a fundamental ability of vision-based robot manipulations. Therefore, in this study, the occlusion information is used in both robot manipulation state estimator and the reward shaping to realize the occlusion avoidance automatically. To fulfill reliable robotic manipulation with calibration-free monocular vision guidance, the following contents are arranged: the DL-based state estimator will be built and analyzed in section III; the DRL with occlusion-awareness method and regarding manipulation settings will be introduced in section IV; experiment results and regarding analysis will be elaborated in section V, and conclusion will be summarized in section VI.

III. DL-BASED STATE ESTIMATOR FOR OBTAINING TARGET OBJECT LOCATION

To fulfill vision-guided DRL robotic manipulation of target objects, a state estimator that used to obtain the 3D position of the target needs to be established first. In this section, we firstly compare and evaluate performances of several general DL models that could serve as state estimators for regressing location of a target block, and then the influences of the occlusions are considered and analyzed.

A. GENERAL METHODS FOR REGRESSING POSITION OF A TARGET BLOCK

To perform efficient training while maintaining generalized capability of a DL model for estimating target location in 3D task space of a real scene, domain randomization [3] is applied in the simulation environment, and then the model is fine-tuned by a few real scene images. For the domain randomization, the color and texture of the background and table are randomly selected. The light of the scene and the position of the camera were also changed in a large range

TABLE 1. Parameter ranges for randomization.

Parameter description	Parameter range
Light intensity	30%-100% of the maximum light intensity
Distance between the camera focus and the table center	141 ± 5 cm
Shaft orientation of the camera	Vertical: $45 \pm 25^\circ$ Horizontal: $0 \pm 70^\circ$
Color of the table and the background	All colors except for the red

(see details in Table 1, it is noted that during altering the shaft orientation of the camera, the direction of the focus is kept in the center of the table). We totally collect 4,000 scenes and 10 images for each scene (totally 40,000 images) in the simulation environment as the training set. For the real scene, a random camera position is selected (without knowing the camera parameters) and 50 real scene images are collected as the testing set. VGG-16 [42] and ResNet34 [45] are two networks that serve as the main part of the state estimator (as shown in Fig. 2). The input of the estimator is $224 \times 224 \times 3$ RGB image which comes from a monocular camera. The loss function of estimator is the mean squared error between the predicted 3D position and the ground truth. In addition, the meta learning (MAML [28]) which aims to quickly adapt a trained DL model to a new scene through a few of samples-based fine tuning, is also taken into consideration.

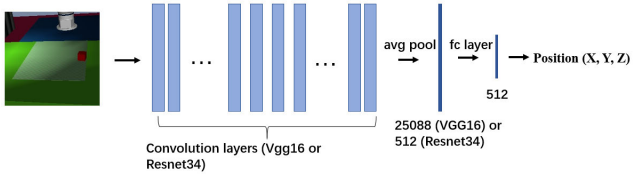


FIGURE 2. The network structure of the state estimator for target block location regression.

Fig. 3 illustrates the framework for training the state estimator. When performing the common DL model training method, all the scene images generated by simulation are directly used to train the state estimator. Before applied to a real scene task for performance evaluation, the state estimator is fine-tuned with a few real images by Adam optimizer. When performing the meta learning method for training the state estimator, every single scene (e.g. a series of images of a target object captured from a specific camera perspective with randomly selected render setting) in simulation is designed as the sub-task, and the support set contains 20% of images of this scene while the remaining 80% images make up the query set [28]. The fine tuning steps of the meta learning is identical to that of the common deep learning. The evaluation result is illustrated in Table 2, where "Fine-VGG" means that the VGG-based model is firstly trained on simulation data set, and then fewer real scene images (from 0 to 10) are used

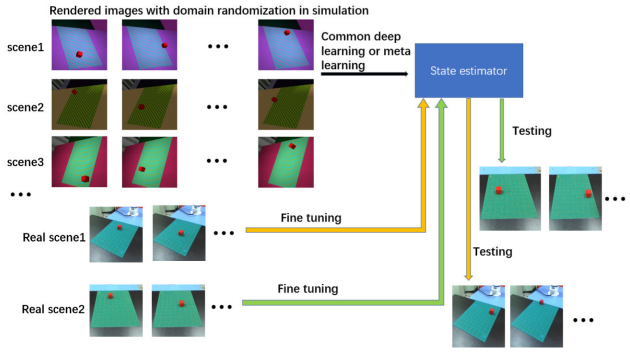


FIGURE 3. Framework of the training process for optimizing a DL state estimator.

TABLE 2. Performance comparison among three state estimators with different settings.

images used for fine tuning	0	5	10	20
Fine-VGG (mm)	124	20.3±3.5	12.8±0.4	8.1±1.7
Meta-VGG (mm)	143	32.9±3.5	18.1±4.0	10.0±2.8
Fine-ResNet(mm)	156	14.4±1.1	11.1±1.6	7.8±0.6

to fine tune the model; the "Fine-ResNet" is similar to the "Fine-VGG", which is based on the ResNet34 backbone; the "Meta-VGG" is based on VGG backbone and trained using Meta learning method. The values in the Table 2 is calculated by three random seeds.

From Table 2, it can be seen that though significant domain randomization has been applied to the simulation environment, the state estimators with different settings still show poor accuracy when there is no real scene data-based fine tuning process. The result is quite different from that in [3], and an explanation may be that the randomization parameters need expertise experience to adjust and to determine. Though there are poor performances of all three types of estimators for zero-shot transfer from a simulation to a real scene, the accuracy becomes higher as the number of images used to the fine tuning process increases. It is also noted that the Fine-VGG has a higher accuracy than that of the Meta-VGG, independent of the number of images used in the fine tuning process. The evaluation results indicates that the meta learning may not be suitable for the specific location estimation task compared with the common training method which can extract the common features more effectively. By comparing performance of the Fine-VGG with that of the Fine-ResNet, one can see all the results of the Fine-ResNet are better than the counterpart, therefore, the ResNet34 is determined as the backbone of the state estimator in the following study.

B. STATE ESTIMATOR CONSIDERING OCCLUSION INFORMATION

In order to realize the intelligence for avoiding occlusion during manipulation tasks, a robot has to know what is occlusion. Different from the data set in part A above, which is

sampled when the pose of a robot arm is frozen, a new data set is collected in the simulation environment when there is a randomly moving robot arm to train and to extend capability of the state estimator. To show occlusion influence on the performance of the state estimator, a range of camera settings where occlusions are easily encountered are chosen. Based on the Fine-ResNet, occlusion information is concatenated to the 3D position information of a target. The occlusion information is represented as the ratio of the target block's exposure area to its full surface area (as shown in Fig. 4).

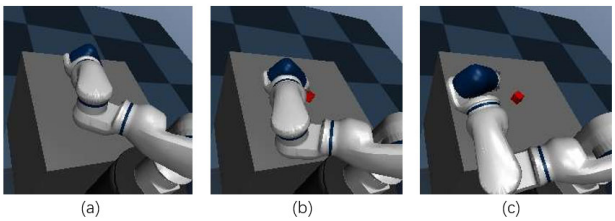


FIGURE 4. Illustration of the occlusion occurs during the robot manipulation: total occlusion (a), partial occlusion (b), and no occlusion (c). The occlusion proportion is evaluated by calculating the ratio between the exposed part and the full area of the target block.

The loss function of the state estimator considering occlusion is defined as follows:

$$L_{ResNet} = -\sum_{m=1}^{3}(d_m - \widetilde{d}_m)^2 - |(O - \widetilde{O})|, \qquad (1)$$

where d_m (m =1,2,3) represents the actual x, y and z value in the world coordinate and $\widetilde{d}_m$ (m =1,2,3) denotes the predicted x, y, z value generated by the state estimator. O and $\widetilde{O}$ represent the actual occlusion ratio and the one predicted by the state estimator, respectively.

The occlusion data proportion is controlled up to 30% during training. A number of 180, 000 images were captured and labeled automatically in the simulation environment to train the state estimator. Another 25, 000 data (20, 000 data with occlusion and 5, 000 data without occlusion) were used to test the obtained state estimator. By adding the randomization during the training process, the accuracy of the state estimator has been improved significantly (the mean error reduced from 10 cm to 0.84 cm for the data without the occlusion, and from 12 cm to 1.34 cm for the data with occlusion). Although the state estimator has tolerance of partial occlusion to some extent, the error could be unbounded in the case of full occlusion, which demonstrates the necessity of the occlusion avoidance ability for robot manipulation.

IV. DRL STRATEGY WITH OCCLUSION AWARENESS FOR ROBOT MANIPULATION

A. OVERVIEW OF THE POLICY LEARNING

The goal of this research is to control a robot in an unstructured environment with a robust policy that can complete tasks using vision information from a single monocular camera. Although some traditional controllers could also deal with the noisy and uncertain feedbacks, it has to redesign the control system once the environment changes. Therefore, in order to make the control system scalable to various environments, the robot should be able to learn the control policy by itself. In this study, we use the DRL to build a fully autonomous control algorithm that can complete the manipulation in an unstructured environment. The policy training process is shown in Fig. 5, the state estimator is trained in section III and frozen in policy training, and the proprioception contains the information of the angle and angle velocity of each joint of the robot arm.

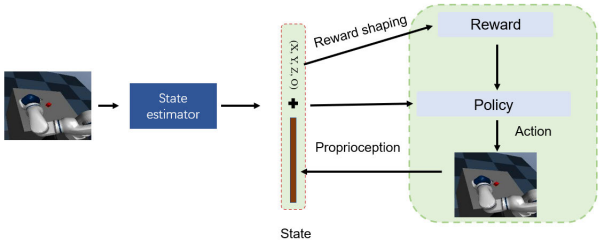


FIGURE 5. Illustration of the DRL structure and training process.

The policy learning can be considered as a standard finite-horizon, discounted Markov Decision Process (MDP). At time step t, the agent observes a state $s \in S$ and takes an action $a \in A$ with respect to the policy $\pi : S \rightarrow A$. Then the agent gets a reward r according to the mapping $S \times A \rightarrow R$, and observes the next state s'. Horizon T and discount factor $\gamma \in (0, 1]$ represent the maximal step in a single episode and the degree that rewards decay over time, respectively. The γ is set to 0.9 for all experiments in this paper. Our goal is to maximize the expected total reward $G(\pi)$ in the finite-horizon T, where $G(\pi)$ can be written as:

$$G(\pi) = E_\pi\left[\sum_{t=0}^{T}\gamma r(s_t, a_t)\right]. \qquad (2)$$

Particularly, S is the low-dimension information with clear physical meaning which contains two parts: the robot proprioception (such as the angle of joint) in the joint space, and the information of the target object (estimated by the state estimator learned from high-dimension image data) in the task space. A is defined as the continuous angle velocity of 7 joints and the grasping status of the end-gripper. The policy is represented by a neural network with parameters θ_π. In this paper, we select the Deep Twin Delayed Deep Deterministic policy gradient algorithm (TD3) [33] as the underlying RL algorithm.

B. CONFIGURATION OF THE POLICY LEARNING

1) Task setup. We evaluate our scheme in two common manipulations, reaching and lifting of a building block. Both tasks are conducted in the Robosuite environment [46], which is developed based on the MuJoCo physics engine [47]. All the simulative experiments are running on 2080Ti GPU with TensorFlow deep learning framework. The robot we used in

this study is a 7-DOF robot (SCR5, Siasun. Co, Shenyang, China) with a two-finger gripper. For the reaching task, the gripper is kept closed, and the task is completed only when the end of gripper touches the target block or within 3 cm from the center of the block. For the lifting task, the robot should grasp the block at first, and then lift it 4 cm away from the desktop to complete the task.

2) Reward shaping. To simplify the exploration and to improve the learning efficiency, we adopt the following staged reward function.

$$r_1 = 1 - tanh(10 \times d) \quad \text{(reaching)}$$

$$r_2 = \begin{cases} 0.25 & \text{if } block \text{ is } grasped, \\ 0 & \text{if } else. \end{cases} \quad \text{(grasping)}$$

$$r_3 = \begin{cases} 1 & \text{if } block \text{ is } lifted \text{ for } 4\,cm, \\ 0 & \text{if } else. \end{cases} \quad \text{(lifting)}$$

$$r_o = \begin{cases} -\alpha \times (1-o)^2 & \text{if } 1-o > \beta, \\ 0 & \text{if } else. \end{cases} \quad \text{(occlusion)}$$

where d is the Euclidean distance between the end gripper and the target block. $1 - o$ is the occlusion proportion which is detailed in section III. α is the coefficient of the occlusion reward. β is the occlusion threshold which means the occlusion reward makes effect only when the occlusion rate is large enough. For the reaching task, the reward for policy learning is set as $R_{reach} = r_1 + r_o$. For the lifting task, the reward for policy learning is $R_{lift} = r_1 + r_2 + r_3 + r_o$.

3) Evaluation metrics. The episode reward is used to evaluate the policy performance. Specifically, during the evaluation, the episode reward is calculated by the true state (obtained from the simulation environment directly) rather than the predicted state by the state estimator. For the reaching and lifting experiments, the policy was run for 5, 000 and 15, 000 episodes, respectively, with each episode lasting for 200 steps. We evaluate 5 episodes every 20 training episodes. Mean and standard deviation curves are drawn across 4 individual tests with different random seeds. Besides, the success rate of task completion is also used as an indicator of manipulation performance that each policy can achieve.

V. EXPERIMENTAL RESULTS AND ANALYSIS

Particularly, the experiments are designed to address the following questions: 1) How does the policy trained with our state estimator perform in the robot manipulation tasks? 2) Whether the agent trained with our method can solve the occlusion problem, and to what degree? To answer these questions, we make several ablations for both reaching and lifting task as follows:

Oracle: Using the ground truth as the state and to calculate the reward.

End-to-end pixel (ETE): Instead of using the state estimator, the system directly trains a policy to map the image to action with sparse reward.

Predicted state with no occlusion information (PNO): Using the position information predicted by the estimator as the state, which means the occlusion information is not taken into consideration of both the state and the reward.

Predicted state with occlusion information but no occlusion reward (PNR): Using the position and occlusion information predicted by the estimator as the state without involving the occlusion information into the reward shaping.

Predicted state with occlusion information and occlusion reward (POR): Using the position and occlusion information predicted by the estimator as the state, and using the occlusion information in reward shaping.

A. TRAINING RESULTS OF THE REACHING TASK

The training curves of the reaching task are shown in Fig. 6. It can be seen that the episode reward of the *ETE* nearly maintains zero during the training process, which means that the agent did not learn anything about how to complete the task at all. It is consistent with the fact that it is hard to train the policy directly from high-dimension input such as image due to the high sample complexity, especially when occlusion is involved.

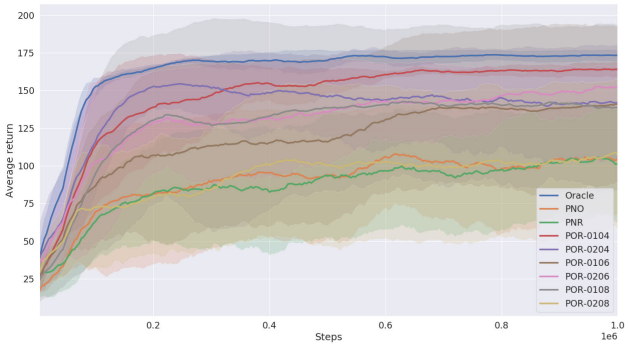


FIGURE 6. The training curves of the reaching task. The solid line and shade area represent the mean and standard deviation for different random seeds, respectively. α and β are two coefficients that can be adjusted in the ***POR***, for example, ***POR-0106*** means that α is 0.1 and β is 0.6, respectively.

Oracle represents the case using low-dimension information such as the position of the target object, and both the state and reward are just obtained from the environment directly without any noise or disturbance. That is the reason why it has the fastest convergence speed and the highest mean episode reward (170). In addition, since all of the state and reward come from the low-dimension observer, it is not necessary to take the occlusion into consideration. However, in the real practice of robot manipulation, it is difficult to obtain the low-dimension information directly, unless a number of position sensors are used in a structured environment. In such a condition, the *PNO* method can feed back the agent with both the estimated state and reward. Although this estimation is not as accurate as the low-dimension observer, it still provides a decent environment information that can make the agent converge (as shown in Fig. 6). However, almost all of the vision-based servo control methods experience the occlusion problem including the *PNO* network. When the occlusion

occurs, the agent will get less accurate state feedback (which has been proved in section III) and thus it will become difficult to maintain decent performance. This is a possible explanation for the phenomenon that the *PNO* converges relatively slow and obtains lower total reward (under 100) compared with other methods.

To solve the occlusion problem, an alternative way is to use multi-view information from the cameras at different locations. However, it will increase both the system complexity and hardware cost. Therefore, the best way to solve the occlusion problem could be to make the control system aware of the occlusion and avoid it actively. In order to verify this assumption, we designed an experiment, inside which the occlusion information is provided to the agent as the state and corresponding occlusion punishment (*POR*). Hopefully, the agent can solve the occlusion problem by avoiding occlusion actively but still can complete the desired manipulation task. The experimental results (as shown in Fig. 6) show that during all of the 6 tests (for different reward coefficient and occlusion threshold), all of the *POR* performed better than *PNO*. In specific setting ($\alpha = 0.1$ and $\beta = 0.4$), *POR* can even get 60 episodes reward higher than *PNO*. It means that the *POR* can solve the occlusion problem actively without any additional interference.

Interestingly, a different combination of α and β may come out with quite different performance. For the *POR-0104* ($\alpha = 0.1$ and $\beta = 0.4$), the episode reward reaches 160, but for the *POR-0208* ($\alpha = 0.2$ and $\beta = 0.8$), the episode reward is even lower than 100. It shows that a little change in the occlusion reward could lead to very different policy performance. A reasonable explanation for this is that when the end-effecter (the gripper) of the robot arm is getting close to the target block, it could produce new occlusion, and the occlusion reward term may prevent the policy from getting better performance. Therefore, the proper occlusion coefficient setting is important for the agent to solve the occlusion problem but does not affect the completion of the manipulation task. To further investigate the influence of the occlusion reward item, we designed another ablation experiment which set both α and β to zero (*PNR*). The experimental result shows that the policy performance of the *PNR* is similar to the *PNO*, which means that simply adding the occlusion to the state may not lead to better performance, and it can be concluded that it is important to make the use of the occlusion information in the reward shaping properly.

B. TRAINING RESULTS OF THE LIFTING TASK

To further evaluate our scheme in a more general manipulation scene, we conduct a series of experiments in lifting task which is a multi-stage task comprised of the reaching, grasping, and lifting process. The ablation experimental results (as shown in Fig. 7) illustrate that the *Oracle* still gives the best performance (the episode reward can reach as high as 320). It is consistent with the analysis in reaching task, because the agent can obtain the accurate state and reward

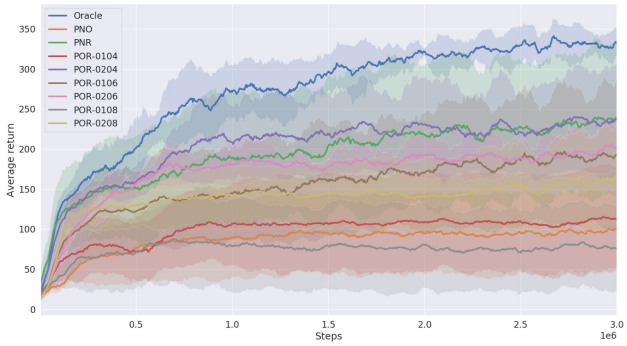


FIGURE 7. The training curves of the lifting task. The annotations are consistent with the Fig. 6.

without any noise or error. As for *PNO*, the final episode reward in the lifting task is similar to that in the reaching task (both are under 100). It means that in the lifting task, the agent trained with *PNO* can learn to reach the target block, but cannot complete the followed grasping and final lifting process. An intuitive reason is that compared to reaching stage, the grasping stage and final lifting stage will encounter more occlusions (in grasping stage and final lifting stage, the gripper must touch the block and thus lead to the occlusions). The occlusion problem reduces the accuracy of the predicted position of the target block, and thus finally makes a negative influence of the policy performance.

Compared to the *PNO*, the highest episode reward of the *POR* (*POR-0204*) can reach 260, which is about 160 higher than the *PNO*. It shows that our method can solve the occlusion problem in a large degree. It should be noted that for specific *POR* (*POR-0108*), the episode reward is even lower than the *PNO*. As there is plenty room to adjust the coefficient α and β, it is possible that there exist some typical combinations of α and β to produce relatively good or bad result which increases the complexity of parameter optimization work. An amazing result is that the performance of the *PNR* (the one uses the occlusion information for the state but not for the reward), is quite similar to the best of the *POR* (*POR-0108*). It illustrates that simply adding the occlusion information to the state can produce surprising effect which contradicts the result in the reaching task. A reasonable explanation is that when more occlusions occurred in the lifting task, it becomes more difficult to balance the occlusion reward item (r_o) and the task reward item (r_1, r_2, r_3) through adjusting the parameter α and β. For the *POR*, the agent may learn to utilize the occlusion information in the state even without the corresponding reward item, which performs more intelligent behavior.

C. SUCCESS RATE FOR REACHING AND LIFTING TASK

To study the influence of the occlusion on the state, the success rates of two ablations (*PNO* and *PNR*) are illustrated in Fig. 8. It can be seen that for the reaching task, the success rate of the *PNR* is 12.2% higher than that of *PNO*. For the lifting task, the success rate of the *PNR* is 69.2% while that of

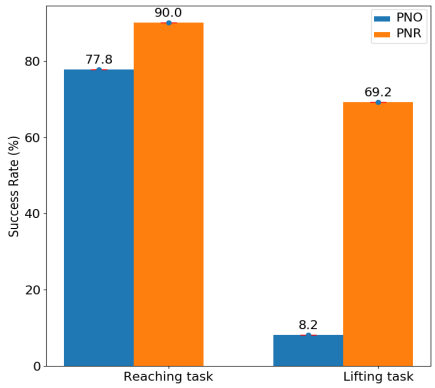


FIGURE 8. Success rate comparison between the policies with and without occlusion ratio in their state for completing the two tasks. 100 episodes are determined for each ablation, and it shows the mean value of the 100 episodes for each ablation with 4 random seeds.

the *PNO* is only 8.2%. It shows that considering the occlusion information into the state can produce positive effect on the policy, which enables the agent to utilize the occlusion information effectively. The reason for the *PNR* performing much better than the *PNO* in the lifting task compared to that in the reaching task should be that, for *PNO* policy in the lifting task, the agent could encounter more occlusions, which limits the performance. As discussed in part B of this section, if more occlusions are encountered, the accuracy of the predicted position of the target block reduces more, and it will finally make a negative influence on the policy performance.

To further study the influence of the occlusion in the reward, the success rates of all ablations in reaching and lifting tasks are collected and shown in Fig. 9. It can be seen that for the reaching task, the *POR-0104* can get 97.2% success rate, which means that the agent can almost complete the reaching task for all 4 random seeds. However, for the lifting task, the success rate of *PNR* is 69.2%, which is much higher than the best of *POR*. As the *PNR* is a special case of

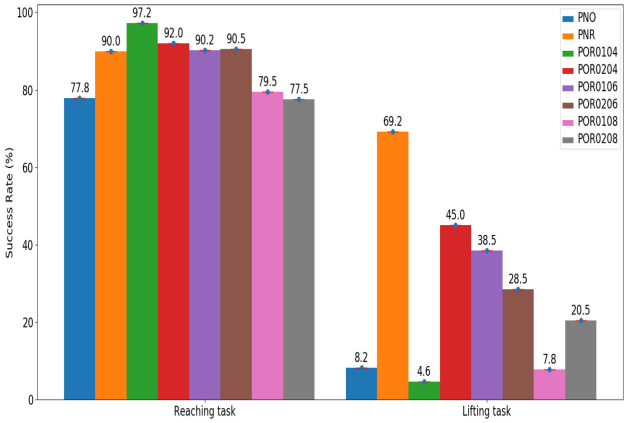


FIGURE 9. The success rate of the tasks with different α and β value.

the *POR* (when $\alpha = 0$ and $\beta = 0$), suggesting that it is better to let the agent utilize the occlusion information itself rather than to specify the parameters of reward shaping.

To quantitatively illustrate that the proposed scheme can solve the occlusion problem, a series of tests of occlusion proportion during manipulation tests were calculated. For each ablation experiment in previous subsection A and B, we recorded the average blocked-to-exposed proportion of 100 episodes, which is shown in Fig. 10. The maximal exposed proportion for each step is defined as 1.0 corresponding to the situation that the target block is completely exposed, and the maximal exposed value of each episode is 200 as there are 200 steps contained in each episode. For the *POR*, we select the policy that performs best in the ablation tests (*POR-0104* for reaching task and *POR-0204* for lifting task).

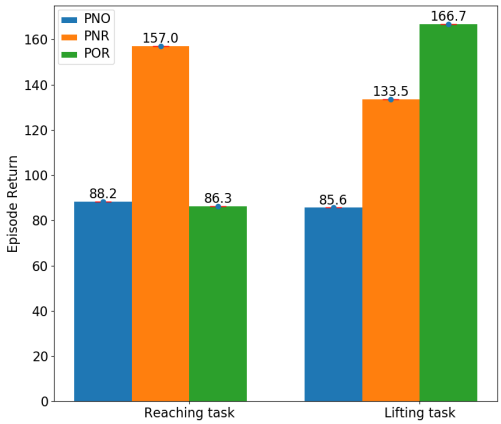


FIGURE 10. Average and standard deviation of blocked-to-exposed proportion of 100 episodes for different ablations.

The Fig. 10 shows that for both reaching and lifting tasks, the policy *POR* employing visual occlusion reward has a much higher average blocked-to-exposed proportion. In detail, 157 and 166.7 for reaching task and lifting task, respectively. It means that the *PNO* scheme can improve the visual occlusion problem significantly compared to the *PNO* and *PNR*. In addition, for the reaching task, both of *PNO* and *PNR* have a similar relatively low average blocked-to-exposed proportion (88.2 for *PNO* and 86.3 for *PNR*). It means that in the reaching task, introducing the occlusion information into the state only cannot solve the visual occlusion problem, which is consistent with the result and analysis in the subsection A. For the lifting task, we find that *PNR* has a superior performance for tackling the visual occlusion compared to *PNO* (133 for *PNR* and 85 for *PNO*), which explains why *PNR* performs much better than *PNO* in subsection B. A reasonable explanation for the different performances between *PNR* and *PNO* for tackling the visual occlusion in various tasks is that for the lifting task, the gripper touches the target block more frequently, and thus has a higher occlusion probability, which finally enhances the learning ability of *PNR* for solving the visual occlusion problem.

D. REACHING TASK WITH DIFFERENT TARGET BLOCKS

To evaluate the performance of our scheme on reaching target blocks with different shapes, we set up a scene that contains three types of building blocks with cube, rectangle and cylinder shape. Instead of training and testing the policy with each type of block independently, these blocks are mixed and utilized to train the policy together. In detail, for the initial state of each episode, one of three types of block is randomly selected as the target to interact with the robot arm and the environment. Typical training process images containing different targets are shown in Fig. 11.

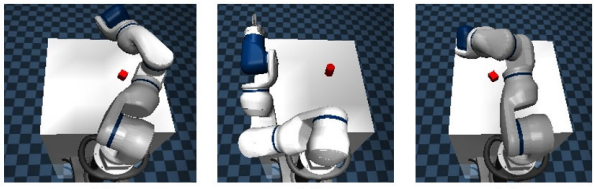


FIGURE 11. The scene with three different types of building blocks: The size of the cube block (a) is 5 cm×5 cm×5 cm, rectangle (c) is 4 cm×5 cm×6 cm. The height and radius of the cylinder (b) is 8 cm and 3 cm, respectively.

It has been shown in Fig. 6 that setting hyperparameter α and β to 0.1 and 0.4 can achieve the best performance among the existing parameters in the reaching task, thus we adopt this setting (*POR-0104*) and conduct another three ablation experiments ((*PNO*, *PNR* and *Oracle*) to illustrate the effectiveness of the proposed strategy. The training curves of the reaching task with different blocks are shown in Fig. 12.

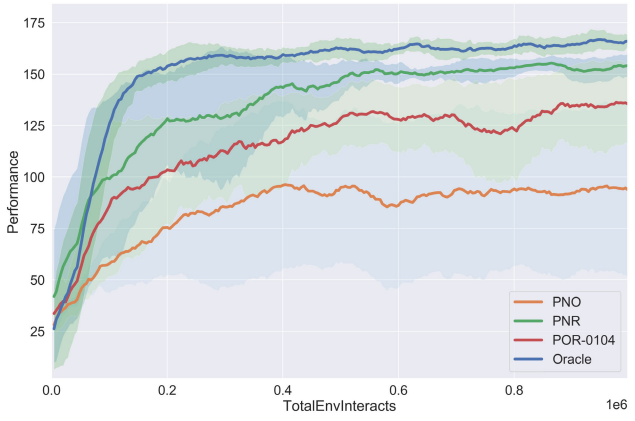


FIGURE 12. The training curves of reaching task with different blocks. The annotations are consistent with that of Fig. 6.

It can be seen from Fig. 12 that the *Oracle* still can obtain highest mean episode reward (170) when the accurate low-dimension information is used, which is consistent with the result shown in Fig. 6. In addition, the *PNO*, that does not utilize any occlusion information, performs worst, which illustrates the importance of incorporating occlusion information into the DRL learning scheme. Different from the results shown in Fig. 6, the mean episode reward of *POR-0104* can only reach up to 135 in this scenario, which is about 30 lower than that of Fig. 6. The most possible reason is that for the scenario with different types of blocks, the fixed parameter setting (α to 0.1 and β to 0.4) is not quite suitable, and there should be some other parameter configuration that fits this task better. It is also noted that the mean episode reward of the *PNR* is 20 higher than *POR-0104*, which is also inconsistent with the result of Fig. 6. The reason might be that targets with different shapes prefer different hyperparameter settings, therefore, it is better to only add the occlusion information into the state and let the agent learn how to utilize the information itself.

E. REAL ROBOT IMPLEMENTATION

An advantage of the proposed scheme is that it can achieve decent policy transfer from a simulation to a real scene by domain randomization technique. To test the effectiveness of our scheme in a real scene, we conducted two experiments of the reaching task. One is the *PNO* corresponding to the scene that only utilizes the position information of the target block. Another one is *POR* (*POR-0104*) corresponding to the scheme that involves the occlusion information in the state and reward shaping. We firstly trained the two policies above for *PNO* and *POR* in our Robosuite simulation environment for angle control instead of angle velocity as the real robot only supports angle control. After that the trained policies were transferred to the real robot control system directly. The hardware setup for this experimental test is shown in Fig. 13. The platform contains a monocular camera (ZED, Stereolabs, San Francisco, CA, US, only a single view was used in this study), a 7-DOF robot (SCR5, Siasun. Co, Shenyang, China) with a flexible gripper, and an operation table with a cubic building block on it. The size of horizontal table and building block are 80 cm×80 cm and 5 cm×5 cm×5 cm, respectively. At the beginning of each episode, the configuration of the robot and the position of the target block

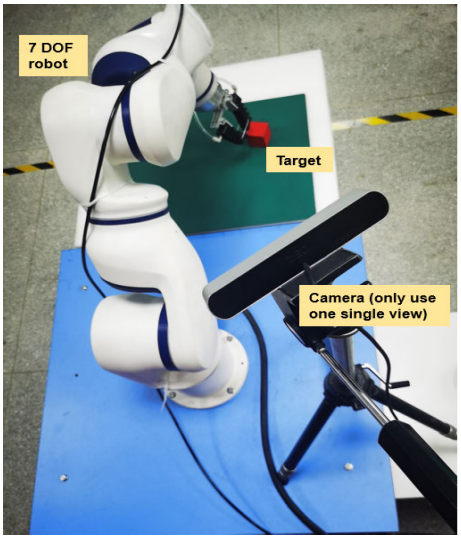


FIGURE 13. Experimental setup for the robot manipulation.

were randomly selected. For both simulation and real scene, we tested 10 episodes and recorded the success rate for the *PNO* and *POR*, which are summarized in Table 3.

TABLE 3. The comparison of success rate.

	PNO (success/total)	*POR* (success/total)
Simulation	7/10	9/10
Real scene	4/10	7/10

As is shown in Table 3, the POR policy has a better performance in the term of success rate compared with the *PNO* policy, which is consistent with our simulation result. It means that the occlusion information is important in the policy training and testing process. It should be also noted that both of the success rates of the *POR* and *PNO* in the simulation are higher than that in the real scene. An explanation for this phenomenon is that although domain randomization has been used in this experiment, the state estimator comes up with larger variance after transferring, which could be the main reason for this lower success rate. However, through further randomization and fine-tuning process, the gap should be smaller and thus the result in the real scene should be more consistent with that in the simulation.

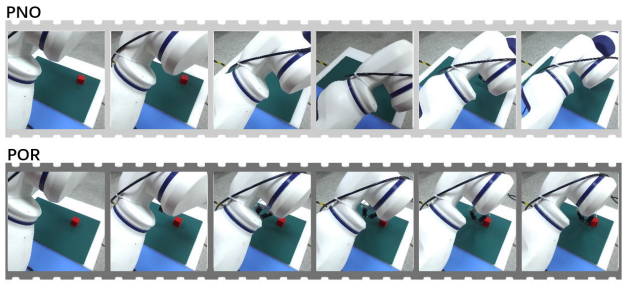


FIGURE 14. A rollout for the real robot manipulation tests of *PNR* and *POR*.

To illustrate the relation between the occlusion and effectiveness of manipulation, some frames in each testing episode of *PNO* and *POR* during the robot manipulation are shown in Fig. 14. It can be seen that, for the *POR*, the robot tries to avoid the occlusion actively when the robot is about to shield the target block, and thus complete the task effectively. However, for the *PNO*, although the robot tries to get close to the target block at the beginning, as the robot cannot perceive the occlusion (there is no occlusion information in state and reward), it is easy to encounter the occlusion problem which leads to an inaccurate state and reward to the agent. That is the reason why the success rate of the *PNO* is lower than *POR* in both simulation and real scene.

VI. CONCLUSION

In this study, we propose a universal method that can estimate the 3D position of a target object in random camera perspective without calibration. By fine tuning the state estimator with a few real scene images, the accuracy of the state estimator can reach up to 7.8 mm without occlusions which is the highest accuracy as we know. The occlusion, which is a common but neglected problem, is also taken into consideration of the DRL policy design. By introducing the occlusion information to states and reward (when $\alpha = 0.1$ and $\beta = 0.4$), we find that the policy is improved much in the reaching task (the success rate rises from 77.8% to 97.2). Besides, a pair of α and β performing well for all different tasks may not exist. But in the scene that occlusion is easily encountered, we find that adding the occlusion information to the state only can improve accuracy of the DRL policy (improve the success rate from 77.8% to 90.0% for the reaching task and improve the success rate from 8.2% to 69.2% for the lifting task). In addition, in the real robot implementation, our scheme can achieve 70.0% success rate in the reaching task by transferring the policy from simulation to the real scene directly without any further training, and that illustrates the effectiveness of the proposed policy with occlusion awareness. This method could also be easily extended to other manipulation tasks through modifying the reward function accordingly without altering the state estimator.

ACKNOWLEDGMENT

The authors would like to thank Hefei Advanced Computing Center for providing technical support. *(Yongle Luo and Kun Dong contributed equally to this work.)*

REFERENCES

[1] L. Roveda, N. Castaman, S. Ghidoni, P. Franceschi, N. Boscolo, E. Pagello, and N. Pedrocchi, "Human-robot cooperative interaction control for the installation of heavy and bulky components," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 339–344.
[2] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.
[3] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 23–30.
[4] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8973–8979.
[5] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, Jan. 2019, Art. no. eaau4984.
[6] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2786–2793.
[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.
[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[10] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.
[11] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2930–2939.

[12] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "Robust 3D object tracking from monocular images using stable parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1465–1479, Jun. 2018.

[13] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the PNP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, p. 155, 2009.

[14] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[15] F. Liu, P. Fang, Z. Yao, R. Fan, Z. Pan, W. Sheng, and H. Yang, "Recovering 6D object pose from RGB indoor image based on two-stage detection network with multi-task loss," *Neurocomputing*, vol. 337, pp. 15–23, Apr. 2019.

[16] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*. [Online]. Available: http://arxiv.org/abs/1711.00199

[17] T. Hodan, P. Haluza, V. S. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An RGB-D dataset for 6D pose estimation of texture-less objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 880–888.

[18] A. Hämäläinen, K. Arndt, A. Ghadirzadeh, and V. Kyrki, "Affordance learning for end-to-end visuomotor robot control," 2019, *arXiv:1903.04053*. [Online]. Available: http://arxiv.org/abs/1903.04053

[19] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," 2016, *arXiv:1611.04201*. [Online]. Available: http://arxiv.org/abs/1611.04201

[20] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," 2017, *arXiv:1710.06542*. [Online]. Available: http://arxiv.org/abs/1710.06542

[21] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, and J. Schneider, "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[22] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3803–3810.

[23] J. V. Baar, A. Sullivan, R. Cordorel, D. Jha, D. Romeres, and D. Nikovski, "Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6001–6007.

[24] Q. Vuong, S. Vikram, H. Su, S. Gao, and H. I. Christensen, "How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?" 2019, *arXiv:1903.11774*. [Online]. Available: http://arxiv.org/abs/1903.11774

[25] K. Arndt, M. Hazara, A. Ghadirzadeh, and V. Kyrki, "Meta reinforcement learning for sim-to-real domain adaptation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2725–2731.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[29] G. Wang, C. Luo, X. Sun, Z. Xiong, and W. Zeng, "Tracking by instance detection: A meta-learning approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6288–6297.

[30] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9925–9934.

[31] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Meta-learning for fast classifier adaptation to new users of signature verification systems," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1735–1745, 2020.

[32] A. A. Shahid, L. Roveda, D. Piga, and F. Braghin, "Learning continuous control actions for robotic grasping with reinforcement learning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 4066–4072.

[33] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[34] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2829–2838.

[35] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.

[36] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[37] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller, "Data-efficient deep reinforcement learning for dexterous manipulation," 2017, *arXiv:1704.03073*. [Online]. Available: http://arxiv.org/abs/1704.03073

[38] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2015.

[39] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," 2018, *arXiv:1807.04742*. [Online]. Available: http://arxiv.org/abs/1807.04742

[40] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3357–3364.

[41] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8943–8950.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

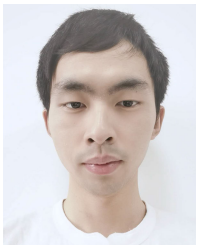
[43] R. Meyes, C. Scheiderer, and T. Meisen, "Continuous motion planning for industrial robots based on direct sensory input," *Procedia CIRP*, vol. 72, pp. 291–296, Jan. 2018.

[44] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning," 2018, *arXiv:1803.04996*. [Online]. Available: http://arxiv.org/abs/1803.04996

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

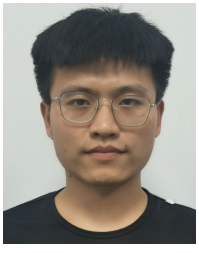
[46] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei, "Surreal: Open-source reinforcement learning framework and robot manipulation benchmark," in *Proc. Conf. Robot Learn.*, 2018, pp. 767–782.

[47] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.



YONGLE LUO received the B.E. degree in automation from Zhengzhou University, Zhengzhou, China, in 2017. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with the intelligence sensing research center, Institute of Intelligent Machines, Chinese Academic of Science, Hefei, China.

His research interests include deep reinforcement learning and robotics.



KUN DONG received the B.E. degree in electronic science from the Hefei University of Technology, China, in 2017. He is currently pursuing the M.E. degree in pattern recognition and intelligent system with the intelligence sensing research center, Institute of Intelligent Machines, Chinese Academic of Science, Hefei, China.

His research interests include programming by demonstration and robotic manipulations.

LILI ZHAO received the B.E. degree in information and computational science from Anqing Normal University, China, in 2019.

She is currently working as a Research Assistant with the intelligence sensing research center, Institute of Intelligent Machines, Chinese Academic of Science, Hefei, China. Her research interests include deep reinforcement learning and robotic manipulations.

ZHIYONG SUN (Member, IEEE) received the Ph.D. degree in mechatronics engineering from Northeastern University, Shenyang, China, in 2016.

He is currently working as an Associate Professor with the intelligence sensing research center, Institute of Intelligent Machines, Chinese Academic of Science, Hefei, China. His research interests include human–machine interaction, micro/nanorobotics, bioMEMS, and modeling/control of smart material sensors and actuators.

ERKANG CHENG (Member, IEEE) received the Ph.D. degree in computer science from Temple University, Philadelphia, PA, USA, in 2014.

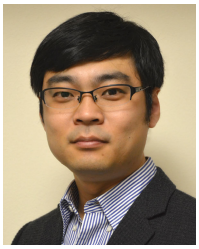
From 2014 to 2017, he worked as a Senior Imaging Research and Development Engineer with Broncus Medical Inc. From 2017 to 2020, he was with Nullmax, as the Director of computer vision. He is currently working as an Associate Professor with the intelligence sensing research center, Chinese Academic of Science, Hefei, China. His research interests include medical image analysis, computer vision, and machine learning.

HONGLIN KAN received the M.E. degree in mechanical and engineering from the Xi'an University of Technology, China, in 2017. He is currently pursuing the Ph.D. degree in detection technology and automation with the intelligence sensing research center, Institute of Intelligent Machines, Chinese Academic of Science, Hefei, China.

His research interests include artificial intelligence and robotic automation.

CHAO ZHOU received the M.Sc. degree in superconducting science and technology from the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, and the Ph.D. degree in superconducting science and engineering from the University of Twente, Enschede, The Netherlands, in 2010.

He is currently serving as a Full Professor with the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China, and also a Guest Scientist with the University of Twente. His main research interests include superconducting science, technology and applications, superconducting magnetic control, cryogenic electronics, and magnetic control-based robotics.



BO SONG (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2016.

He is currently working as a Full Professor with the intelligence sensing research center, Chinese Academic of Science, Hefei, China. His research interests include deep reinforcement learning, human–machine interaction, micro/nanorobotics and systems, and micro/nanomanufacturing.