2022 IEEE 18th International Conference on Automation
Science and Engineering (CASE)
August 20-24, 2022. Mexico City, Mexico

2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-9042-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CASE49997.2022.9926599

Flexible 3D Object Appearance Observation Based on Pose Regression and Active Motion

Shaohu Wang, Fangbo Qin, Fei Shen, Zhengtao Zhang*, *Member, IEEE*

***Abstract*—3D object appearance inspection plays an important role in manufacturing industry. To observe clear images of different parts of a 3D object in a semi-structured scene, camera pose should be properly adjusted to several different viewpoints. In this paper, we propose a flexible appearance observation framework for 3D-shaped objects with 3-DoF pose (2D position and 1D angle) uncertainty. First, we propose 3-DoF Pose Regression Network (PR3Net) based on convolutional neural network (CNN), to estimate the 3-DoF pose of a target 3D object placed on a platform. Considering the data scarcity problem in practical application and the variety of object types, we utilize data synthesis to automatically generate training samples from only one annotated image sample, so that the pose learning can be conducted conveniently. Besides, a semi-supervised fine-tuning method is used to improve the generalization ability by leveraging plenty of unlabeled images. Second, the teachable active motion strategy is designed to enable the inspection robot to observe a 3D object from multiple viewpoints. The human user teaches the standard viewpoints once beforehand. The robot actively moves its camera multiple times according to both the predefined viewpoints and the regressed 3-DoF pose, so that the images of multiple parts of object are collected. The effectiveness of the proposed methods is validated by a series of experiments.**

I. Introduction

Appearance inspection means using camera to observe the surface of the object for defect inspection, which is widely applied in industrial manufacturing[1][2]. Automated appearance inspection consists of three main steps: appearance observation, image processing and defect inspection. The quality of the surface images acquired in the first step highly influences the final inspection results[3].

Currently there are two main architectures for appearance observation, one of which is to place the camera at a fixed viewpoint for image acquisition, such as [3], which used a line-scan camera at a fixed angle to capture flat optical elements. In [4], multiple vision devices were placed at fixed viewpoints to achieve surface defect detection for optical spheres. However, fixed viewpoints are not adaptive to the various types of objects. The second architecture is based on the robotic system to achieve multi-view observation[5]. In this architecture, the viewpoints generation plays a key role in high-quality observation. [6] used a UR5 (Universal Robots) robotic arm with an end-mounted 3D scanner for 3D object observation and proposed a scanning viewpoint generation algorithm based on CAD. However, these methods require that the object is placed at a known pose. In semi-structured industrial scenarios, although the uncertainty is limited, there still exist unknown changes of pose, background and illumination during the repeated observations.

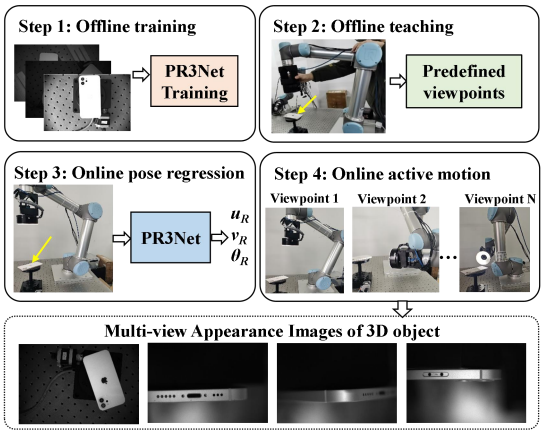

Fig. 1. 3D object appearance observation framework. The object pointed by the yellow arrow is the observation object.

To deal with the pose uncertainty, pose measurement can be used to obtain the actual object pose after the object is placed. In many cases, objects are placed on a plane, such as stage, platform, and conveyor, so that the degree of freedoms of object pose is partially constrained and 6-DoF pose estimation[7][8] is not necessary. Typically, 3-DoF pose measurement is adequate when object only translates on a plane and rotates around a vertical axis. Traditional 3-DoF pose estimation methods include binary segmentation, template matching, etc. However, these methods' performances are limited when the scene is not fully structured. With the development of deep learning, the use of DCNN for visual measurement has significantly surpassed traditional methods, among them, object detection and object segmentation are widely used [9][10]. However, standard object detection cannot provide directional information. [11] designed an orientated object detection network using BBAVectors to describe rotating target frames, achieving faster speed and better accuracy. Some works attempted to combine deep learning with traditional methods[12]. In [13], a circular fitting algorithm was used to obtain the exact location based on the segmentation results of the UNet. [14]

This work is supported by the National Natural Science Foundation of China (U21A20482, 62103413, U1909218), the Project of Test of Specific Area (Beijing Automobile Industry Cluster) Industrial Internet Platform Experiments (DX201103XL01).

Authors are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Science, Beijing 100049, China. Z. Zhang is also with Binzhou Institute of Technology, Binzhou City 256601, Shandong Province, China. {wangshaohu2020; qinfangbo2013; zhengtao.zhang@ia.ac.cn}.

designed an end-to-end affine transformation parameter estimation network to obtain the affine transformation by simultaneously inputting a current image and a template image. The above methods all require post-processing to finally obtain the object pose. End-to-end pose estimation is more efficient [15]. However, deep learning methods usually rely on a large amount of labeled data, which is expensive to obtain in actual industry. In order to solve this problem, self-supervised and semi-supervised methods were proposed[17][18].

In this paper, an end-to-end pose regression network PR3Net and a teachable active motion strategy are proposed to achieve appearance observation for 3D-shaped objects with 3-DoF pose uncertainty. Our contributions are as follows:

1) A 3-DoF pose regression network PR3Net is proposed to obtain the object pose in image space. The coordinate information is fused with the input image, so that the 3-DoF parameters can be efficiently regressed in the end-to-end manner and no post-processing is required.

2) To avoid the costly burden of the manual annotation while utilizing real unlabeled images, a one-to-many sample generation strategy and a consistency-based semi-supervised fine-tuning method are designed to train CNN, which requires only one labeled image and numerous unlabeled images.

3) The teachable active motion strategy is designed to enable robotic inspection system to observe a 3D object from multiple viewpoints, which is adaptive to the object's uncertain pose variation.

II. Overview

As shown in Fig.1, the appearance inspection system mainly consists of a UR5 robot arm and an industrial camera. The camera is mounted on the robot's end. The robot and camera coordinate frames are denoted by {R} and {C}, respectively. An object is manually or automatically placed on the platform with uncertainty. As mentioned before, the 3-DoF pose of object is uncertain after placement. Besides, the 3D object has multiple key parts to be inspected. Only one image usually cannot cover all the key parts on object surface.

To realize the multi-viewpoint inspection of object with 3-DoF pose uncertainty, we deign the following appearance observation pipeline:

Step 1: Offline training. An object of the target type is used for image collection. A series of images $\{I_U\}$ are captured in various conditions. Only one image I_A is manually annotated, namely, the object's 3-DoF pose (u_A, v_A, θ_A) is determined by the human user with a UI software. I_A and the unlabeled $\{I_U\}$ are used to train our PR3Net as introduced in Section III.

Step 2: Offline teaching. The object is put on the platform. The robot moves the camera to the measurement viewpoint P_0 and the camera captures a top-view image I_T of the object when the camera optical axis is approximately vertical. Then PR3Net is used to obtain the 3-DoF pose (u_T, v_T, θ_T) in I_T. Afterwards, the human user manually dragged the UR5 robot arm to N predefined viewpoints $\{P_{T1}, P_{T2}, \ldots, P_{TN}\}$.

Step 3: Online pose regression. In online deployment, after an object of the target type is placed on the platform, the robot firstly moves the camera to the measurement viewpoint P_0, then the camera captures a top-view image I_R of the object. PR3Net is used to obtain the 3-DoF pose (u_R, v_R, θ_R) in I_R.

Step 4: Online active motion. With the aforementioned (u_T, v_T, θ_T), (u_R, v_R, θ_R) and $\{P_{T1}, P_{T2}, \ldots, P_{TN}\}$, the adapted viewpoints $\{P_{A1}, P_{A2}, \ldots, P_{AN}\}$ are calculated using the method in Section IV. The robot actively moves the camera to these N adapted viewpoints. At each viewpoint, the camera captures an image. Finally, the image series $\{I_1, I_2, \ldots, I_N\}$ are gathered and sent to the inspection model.

III. 3-DoF Pose Estimation Network

PR3Net is proposed to obtain the 3-DoF pose (u, v, θ) of object from the top-view image I. To achieve the brief and efficient pose estimation, PR3Net directly outputs the three elements u, v, and θ without using any post-processing.

A. Network Architecture

***Image-coordinate fusion* (*ICF*)**: Traditional CNNs cannot effectively encode the positional information of object due to the translation-invariant property. Inspired by [16], which used normalized coordinate map for CNN training, we involve the horizontal and vertical normalized coordinate maps C_u and C_v into the model's input.

$$C_{u,i,j} = \frac{j}{H}, C_{v,i,j} = \frac{i}{W} (i = 1,2,...,H, j = 1,2,...,W) \quad (1)$$

where W and H represent the width and height of the image, respectively. i and j are pixel indices.

Instead of stacking C_u and C_v with I, we propose to fuse the image with the coordinates by hadamard production and pixel-wise summation between image and normalized coordinates map. Thus, the fused input I_F is the concatenation of $\{I, I \odot C_u, I+C_u, I \odot C_v, I+C_v\}$, which is a 5-channel map, where $\odot$ is elementwise multiplication.

***Encoder*:** The fused input was fed to the encoder, which consists of five blocks. Each encoder block is formed by two convolutional layers. The output channels of convolutional layers are shown in Fig .2. Thus, the multiscale feature maps are obtained from the input, containing both the shape features and the coordinates information.

***Regression branch*:** This branch uses four convolutional layers to squeeze the spatial size and expand the feature channels, as shown in Fig. 2. Two convolutional block attention modules (CBAMs) [20] are embedded, to make the regression focus on relative features. Finally, the feature map is reshaped as a feature vector, and the high-dimension vector is processed by three fully convolutional layers to obtain the estimated (u, v, θ), representing the 3-DoF object pose in the input image frame.

***Auxiliary branch*:** Inspired by [10][11], this branch conducts the auxiliary tasks of foreground (FG) segmentation and keypoint detection, which is only used in the training stage. By leveraging the FG and keypoint information, the model is guided to learn the FG and keypoint related features, which can assist the main task. The auxiliary branch consists of four decoder blocks. Each block has three convolutional layers. The skip connections from the encoder are used to provide multi-scale features. The lower-level feature maps

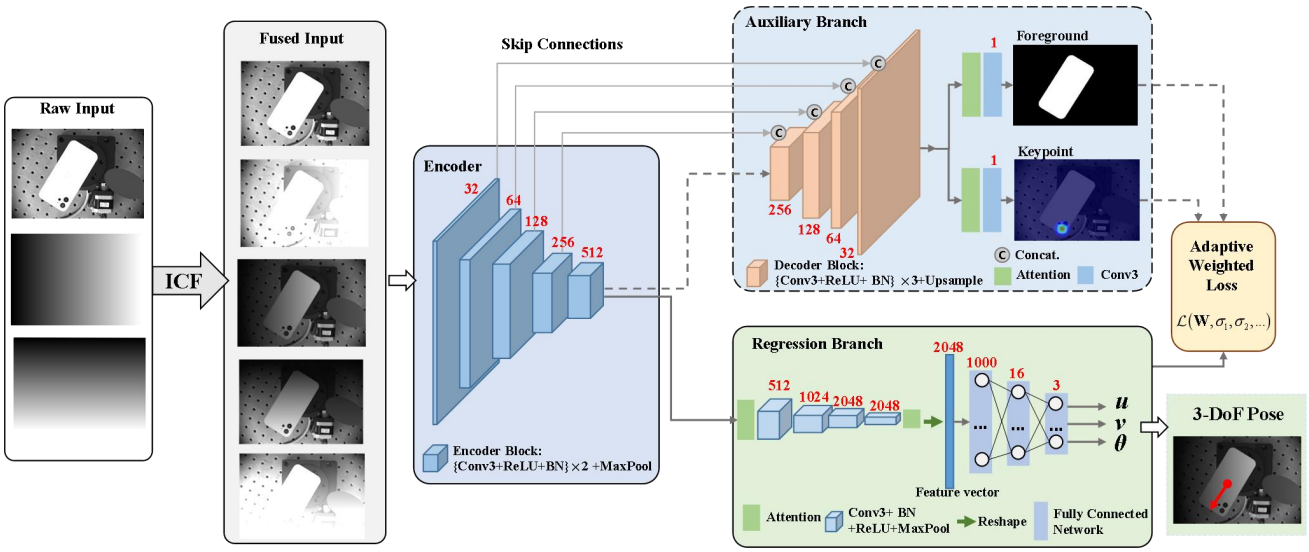

Fig. 2. Network Architecture. The output channels of convolutional layers are labeled by red digits.

are concatenated to the corresponding feature maps in the decoder block. The bilinear interpolation based up-sampling is used to expand the spatial size. Finally, the last feature map is processed by a CBAM and a convolutional layer to predict a FG map M and the keypoint map H.

Each convolutional layer above is with 3×3 kernel size and followed by the ReLU and batchnorm (BN) layers. Each max-pooling layer shrinks the feature map size by ×2.

Adaptive weighted loss (AWL): The training loss includes pose regression loss L_P, FG segmentation loss L_S and keypoint detection loss L_K, which are implemented by smooth-L1, mean squared error (MSE), and MSE, respectively:

$$L_P = \text{SmoothL1}(u,u_{GT}) + \text{SmoothL1}(v,v_{GT}) + \text{SmoothL1}(\theta,\theta_{GT}) \quad (2)$$

$$L_S = \text{MSE}(M,M_{GT}) \quad (3)$$

$$L_K = \text{MSE}(H,H_{GT}) \quad (4)$$

The total loss is the weighted sum of the above three losses. We utilize the adaptive weighted loss in [19]. The weights σ_1, σ_2 and σ_3 can be automatically optimized in backpropagation, and are all initialized as 1.0. The total loss is formed by,

$$L_{Total} = \frac{1}{\sigma_1^2}L_P + \frac{1}{\sigma_2^2}L_S + \frac{1}{\sigma_3^2}L_K + \sum_i \log(1+\sigma_i^2) \quad (5)$$

B. One-to-Many Sample Generation

To avoid the costly burden of the manual annotation of numerous images for CNN training, we design a one-to-many sample generation strategy inspired by [12]. For an object type, only one image I_A is annotated. The annotations include the object center position (u_A, v_A), orientation angle θ_A, FG map M_A, and the heatmaps H_A of one or more keypoints on the object. Besides, a background image I_B is obtained.

The sample generation method is shown in Fig. 3(a). Firstly, a random translation (Δu, Δv), rotation angle $\Delta\theta$ around center (u_A, v_A) and scaling factor k are generated and applied to I_A, M_A and H_A to obtain I_{AS}, M_{AS} and H_{AS}, respectively. The values of the 3-DoF object pose in the transformed image turns to $u_{AS}=u_A+\Delta u$, $v_{AS}=v_A+\Delta v$, $\theta_{AS}=\theta_A+\Delta\theta$. The background image I_B is also processed to be I_{BS} by another random similar transformation. Then the object is embedded in the transformed background image by,

$$I_G = M_{AS} \odot I_{AS} + (1-M_{AS}) \odot I_{BS} \quad (6)$$

However, the real images contain many other interferences as shown in Fig. 3(b). Thus, random data augmentations are applied to I_G, including adding noise, blurring, brightness change, and contrast change. Besides, random blocks are added to I_G, which might partially occlude the object.

With the above three steps, one annotated image can be randomly transformed to numerous images denoted by T_G to train CNN. And the generalization ability is satisfactory when the CNN is used in semi-structured scenes.

C. Fine-Tunning via Semi-Supervised Learning

In order to utilize unlabeled images of real scenes, a consistency-based semi-supervised training method inspired by [18] is used to fine-tune PR3Net. We use random translation of image to obtain a translated image I_1 from a raw image I_0. Both I_0 and I_1 are input to PR3Net, and the outputs are (u_0, v_0, θ_0) and (u_1, v_1, θ_1), respectively. Because the random translation is known as (Δu, Δv), PR3Net is expect to provide the results satisfying $u_0=u_1-\Delta u$, $v_0=v_1-\Delta v$, $\theta_0=\theta_1$. Thus, the pose regression consistency loss is given by,

$$L_C = \text{SmoothL1}(u_0,u_1-\Delta u) + \text{SmoothL1}(v_0,v_1-\Delta v) + \text{SmoothL1}(\theta_0,\theta_1) \quad (7)$$

In the fine-tuning stage, the auxiliary branch is not used. We mix the labeled synthetic images and unlabeled real images together to fine-tune the network. The total loss is

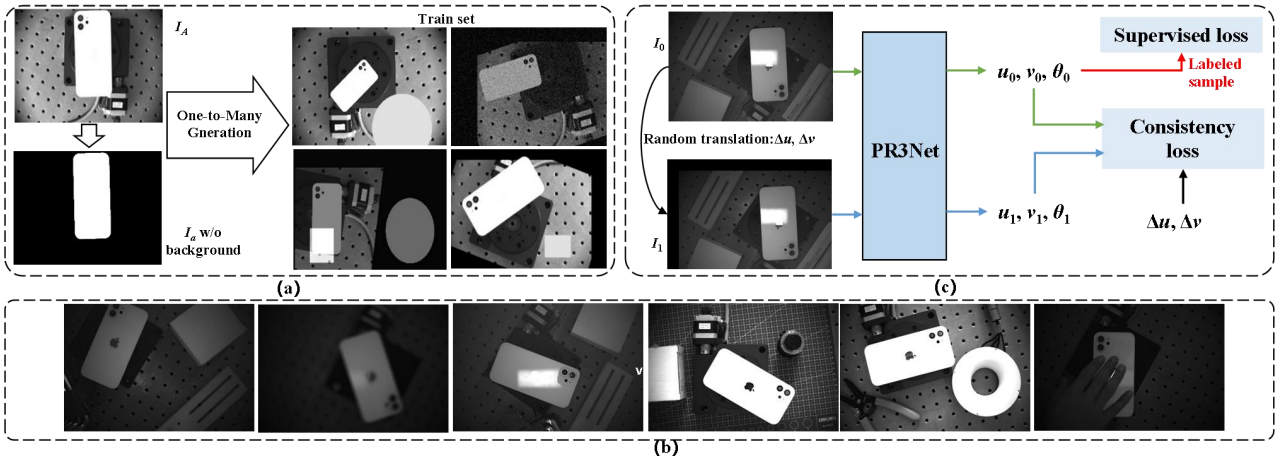

Fig.3. Training and evaluation. (a) One-to-Many Sample Generation. (b) Real test images of iPhone Case. (c) Semi-supervised fine-tuning strategy.

composed of pose regression loss L_P and consistency loss L_C.

$$L_{fine} = L_P + \lambda L_C \tag{8}$$

IV. Active Motion for Adaptive Observation

After PR3Net is trained, the inspection system is able to measure the actual 3-DoF pose of object. As introduced in Section II, the human user firstly put an object on the platform, the camera is moved to the viewpoint P_0 and measures the object's current 3-DoF pose (u_T, v_T, θ_T) in image view using PR3Net. Then the human user moves the UR5 robot to N predefined viewpoints $\{P_{T1}, P_{T2}, \ldots, P_{TN}\}$. As shown in Fig. 4, in the online inspection stage, the object has uncertainty in its 3-DoF pose. The camera is firstly moved to the initial view point P_0, and measures the object's current 3-DoF pose (u_R, v_R, θ_R). By comparing (u_T, v_T, θ_T) and (u_R, v_R, θ_R), the relative 3-DoF pose can be obtained, based on which the predefined viewpoints can be adapted to the actual object pose.

In our system, when the camera pose is at the initial view point P_0, the camera's optical axis is approximately perpendicular to the platform plane. The object's depth is constrained by the platform. Ignoring the object's depth change, the camera model is simplified as affine projection,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{9}$$

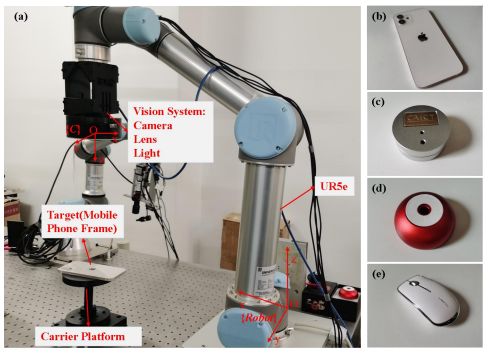

Fig. 4. (a) Robotic arm system. (b)iPhone case. (c) Ornament base 1. (d) Ornament base 2. (e) Mouse.

where x and y are coordinates in the robot frame. k_{11}, $k_{22} \neq 0$, $k_{11}k_{22}-k_{12}k_{21} \neq 0$.

Because the rotation axis of object is also perpendicular to the platform. The angle in Cartesian space can be obtained as

$$\alpha = \begin{cases} \arctan(-\dfrac{k_{11}}{k_{12}}), & u_n = 0 \\ 90^\circ, & k_{22}u_n = k_{12}v_n \\ \arctan(\dfrac{-k_{21} + k_{11}\tan\theta}{k_{22} - k_{12}\tan\theta}), & else \end{cases} \tag{10}$$

where (u_n, v_n) is the direction vector of the target in image, $\theta, \alpha \in [0,180°)$. The object's 3-DoF pose (x_T, y_T, α_T) and (x_R, y_R, α_R) in Cartesian space can be obtained. Approximating that $k_{11}=k_{22}=f$, $k_{12}=k_{21}=0$, we have $\alpha=\theta$, $\Delta x=\Delta u/f$, $\Delta y=\Delta v/f$, $\Delta\alpha=\alpha_R-\alpha_T$. The rotation and translation transformation matrix of the viewpoint in $\{C\}$ can be obtained as follows:

$$R = \begin{bmatrix} \cos(\Delta\alpha) & -\sin(\Delta\alpha) & 0 \\ \sin(\Delta\alpha) & \cos(\Delta\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; t = \begin{bmatrix} x_R \\ y_R \\ 0 \end{bmatrix} - R \begin{bmatrix} x_T \\ y_T \\ 0 \end{bmatrix} \tag{11}$$

Since the relative pose of viewpoint and the object are fixed, the adapted viewpoint $P_A=<p_A, n_A>$ in $\{C\}$ can be obtained:

$$p_A = Rp_T + t, n_A = Rn_T \tag{12}$$

Then P_A can be transformed to $\{R\}$, then the robot actively moves the camera to these adapted viewpoints $\{P_{A1}, P_{A2}, \ldots, P_{AN}\}$. At each viewpoint, the camera captures an image. The images are finally used to inspect the surface defects.

V. Experiments

A. Experiment Setup

The experiment system is shown in Fig. 4(a), which has a UR5e robot and a Daheng Mecury2 industrial camera with 25mm lens. Four 3-D objects are used to verify the

effectiveness of our appearance observation framework, as shown in Fig. 4(b-e). Using one-to-many sample generation method in Section III.C, 5000 images were generated as the training set, 500 images were generated as the validation set. Besides, 1268 unlabeled images are used for fine-tuning. For evaluation, 200 real images were collected and annotated. The image is resized to 300×448 before entering the network. The batch size is set to 2. The number of epochs is 50 (40 for supervised training, 10 for fine-tuning). The learning rates are 0.0005 for supervised training and 0.0002 for fine-tuning. The λ in (8) is set to 0.5. The deep learning framework is Pytorch, running on NVIDIA GeForce GTX2080.

B. 3-DoF Pose Regression

Ablation Experiments: As shown in Table I, 1) by adding Coordinate Maps, the accuracy is significantly improved as they provide explicit coordinate information. 2) Adding Auxiliary Branch alone has no obvious influence on positioning. When using Auxiliary Branch together with AWL, the pose regression accuracy is significantly improved, especially the angle accuracy, because Auxiliary Branch provides more shape features like contours and keypoints. 3) After semi-supervised fine-tuning, the PR3Net's performance is further enhanced, by learning from numerous real images.

Comparative Experiments: PR3Net is compared with three related methods: 1) UNet-R [13] uses UNet to segment the object and fits the foreground shape of it with RANSAC to obtain the centroid and angle. 2) DIHE [14] feeds two foreground images of object to the network together to predict a transformation matrix, which can be transferred to position and angle. 3) BBAVNet [11] estimates object's orientated bounding box, whose center and rotation angle can be obtained. The evaluation results are reported in Table II.

TABLE I
Ablation Study on Adaptive Weight Loss (AWL), Auxiliary Branch (AB), Coordinate Maps (CM) and Semi-Supervised Fine-Tuning (SSF). APE(pixel) and AAE(degree) are average absolute position and angle errors. SR(%) is success rate (threshold=10).

AWL	AB	CM	SSF	APE(p)	AAE(d)	SR(%)
○	○	○	○	9.02	9.81	50.2
○	○	✓	○	7.01	7.35	56.1
○	✓	○	○	9.16	7.26	55.1
✓	✓	○	○	6.61	4.50	83.9
○	✓	✓	○	6.87	5.75	74.6
✓	✓	✓	○	3.74	3.45	88.3
✓	✓	✓	✓	**2.73**	**2.13**	**92.7**

Overall, the pose regression accuracy is higher on validation sets than on test sets, because validation set is generated and is more similar to training set, while test set is gathered from the real scene. In comparison, PR3Net presented the best pose regression accuracies and the highest success rates on almost all the test sets. Moreover, PR3Net has the fastest inference speed, because the regression branch is compact and light-weighted, and no post-processing is required. The 3-DoF pose regression results is visualized in Fig. 5. As shown by the first row, PR3Net regressed the object pose with a reasonable accuracy even when the object is partially occluded. As shown by the second row, PR3Net regressed the object pose with a reasonable accuracy even when the object image has a very large contrast caused by lighting change. Using guided backpropagation, we found that the pixels on the object's salient parts contribute the most to the output.

TABLE II
Comparative Experiment Results with Different Models

Method	Dataset	iPhone Case			Ornament Base 1			Ornament Base 2		Mouse			Inference speed(fps)
		APE(p)	AAE(d)	SR(%)	APE(p)	AAE(d)	SR(%)	APE(p)	SR(%)	APE(p)	AAE(d)	SR(%)	
UNetR[13]	Val.	1.01	1.23	98.0	2.74	81.65	99.4	2.49	100	3.67	6.87	93.3	8.7
	Test	4.99	2.8	84.9	3.64	71.04	42.7	3.49	100	3.64	8.66	94.1	
DIHE[14]	Val.	1.51	**1.10**	99.0	1.47	3.84	95.6	1.74	100	2.63	4.97	96	39.83
	Test	5.99	4.22	82.0	3.86	10.18	67.7	3.4	97.6	4.08	4.19	90.8	
BBAVNet [11]	Val.	**0.63**	3.09	**100**	**0.37**	5.12	96.0	**0.76**	100	**1.22**	2.14	**100**	15.14
	Test	5.30	4.91	88.3	2.53	24.95	75.8	1.66	100	2.79	4.64	**100**	
Ours	Val.	1.52	1.23	99.2	1.57	**2.92**	**98.0**	1.25	100	2.33	**1.93**	98.4	**59.67**
	Test	**2.73**	**2.13**	**92.7**	**1.97**	**3.92**	**91.2**	**1.61**	100	**2.28**	**2.91**	94.6	

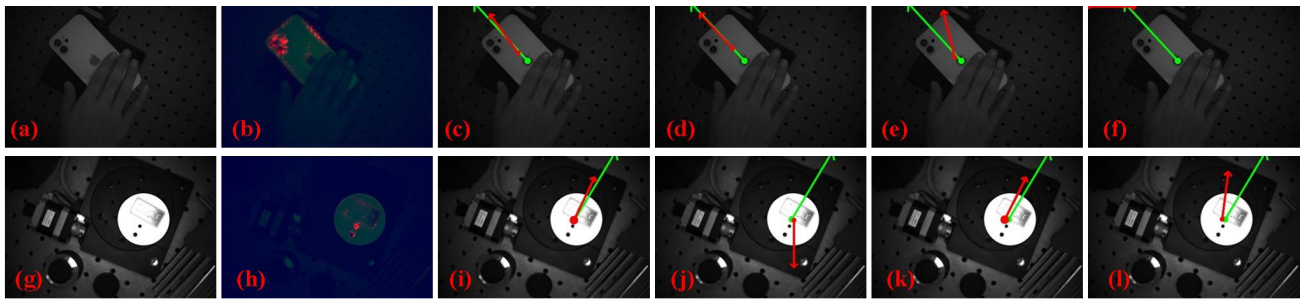

Fig. 5. Visualization of 3-DoF pose regression results. Each row shows an example of an object. (a) and (g) are input image. (b) and (h) are gradient maps of PR3Net obtained by guided backpropagation. (c) and (i) are given by PR3Net. (d) and (j) are given by UNetR. (e) and (k) are given by DIHE. (f) and (l) are given by BBAVNet. The ground truth and regression result are shown by green and red arrows, respectively.

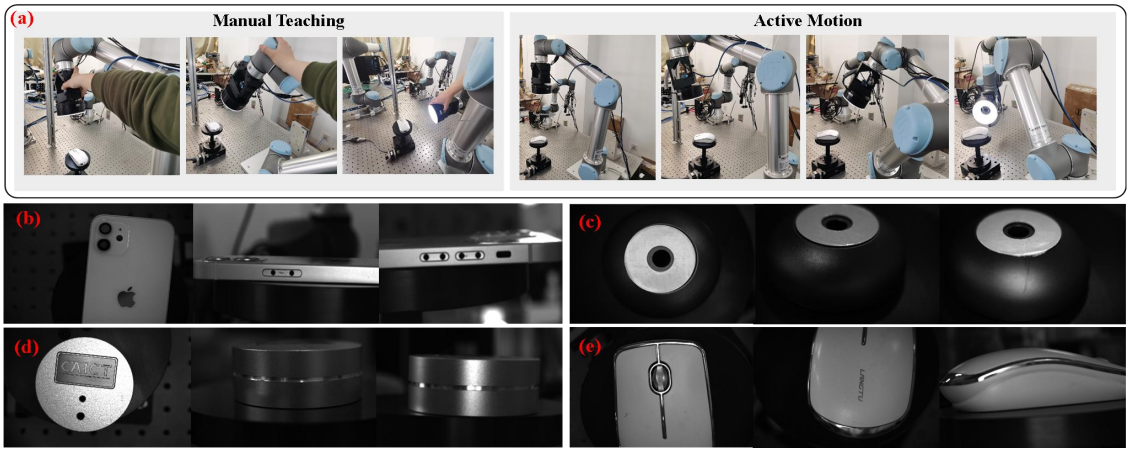

Fig. 6. 3D Object Appearance Observation. (a) Manual teaching and automatic observation. (b-e) Observed images of four objects.

C. 3D Object Appearance Observation

In this experiment, we demonstrate the proposed 3D object appearance observation framework with the aforementioned four objects. As shown in Fig. 6(a), for each object, the viewpoints have been predefined by manual teaching. After an object of the same type is placed on the platform. The UR5 robot first moves the camera to the initial viewpoint to measure the object's actual pose with PR3Net. Then the camera is moved to the adapted viewpoints sequentially, and the images of multiple parts of the object are captured at each viewpoint. These images are clear enough for surface inspection, as shown in Fig. 6(b-e).

VI. Conclusion

Appearance observation is an essential step in appearance inspection. Towards this target, a flexible 3D object appearance observation framework is proposed. To obtain 3-DoF pose of the object in semi-structured environments, we propose an end-to-end pose estimation network PR3Net. A one-to-many sample generation strategy and a semi-supervised fine-tuning method are designed to avoid the costly burden of the manual annotation. The teachable active motion strategy is designed to enable the robot to observe a 3D object from multiple viewpoints. The proposed framework will help to reduce the cost of manual teaching and path planning in industry, but it does not take into account problems such as collisions that may occur after changes of the object's pose, and thus the future work will be focused on obtaining more reliable observation viewpoints by imitation learning.

References

[1] J. Yang, C. Wang, B. Jiang, *et al*., "Visual perception enabled industry intelligence: state of the art, challenges and prospects," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2204–2219, 2021.

[2] R. Almadhoun, T. Taha, L. Seneviratne, *et al*., "A survey on inspecting structures using robotic systems," *Int. J. Adv. Robotic Syst.*, vol. 13, no. 6, p. 172988141666366, 2016.

[3] F. Wang, Y. Yang, D. Sun, *et al*., "Digital realization of precision surface defect evaluation system," *Proc. SPIE*, pp. 61500F-1-61500F-5, 2006.

[4] X. Tao, Z. Zhang, F. Zhang, *et al*., "A novel and effective surface flaw inspection instrument for large-aperture optical elements," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2530–2540, 2015.

[5] Z. Lončarević, A. Gams, S. Reberšek, *et al*., "Specifying and optimizing robotic motion for Visual Quality Inspection," *Robot. Comput.-Integr. Manuf.*, vol. 72, p. 102200, 2021.

[6] I. D. Lee, J. H. Seo, Y. M. Kim, *et al*., "Automatic pose generation for robotic 3-D scanning of mechanical parts," *IEEE Trans. Robotics*, vol. 36, no. 4, pp. 1219–1238, 2020.

[7] W. Kehl, F. Manhardt, F. Tombari, *et al*., "SSD-6d: Making RGB-based 3D detection and 6D pose estimation Great again," *IEEE/CVF Int. Conf. Comput. Vis.*, pp. 1530-1538 2017.

[8] F. Qin, F. Shen, D. Zhang, *et al*., "Contour primitives of interest extraction method for microscopic images and its application on pose measurement," *IEEE Trans. Syst., Man, Cybern.*: *Syst*, vol. 48, no. 8, pp. 1348–1359, 2018.

[9] S. Ren, K. He, R. Girshick, *et al*., "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Int. Conf. Comput. Vis*, pp. 2980-2988, 2017.

[11] J. Yi, P. Wu, *et al*., "Oriented object detection in aerial images with box boundary-aware vectors," *IEEE Winter Conf. Appl. Comput. Vis,* pp. 2149-2158, 2021.

[12] F. Qin, J. Qin, S. Huang, *et al*., "Contour primitive of interest extraction network based on one-shot learning for object-agnostic vision measurement," *IEEE Int. Conf. Robot. and Autom*, pp. 4311-4317, 2021.

[13] S. Yan, X. Tao, and D. Xu, "High-Precision Robotic Assembly system using three-dimensional vision," *Int. J. Adv. Robotic Syst.*, vol. 18, no. 3, p. 172988142110270, 2021.

[14] T. Nguyen, S. W. Chen, S. S. Shivakumar, *et al*., "Unsupervised deep homography: A fast and robust homography estimation model," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2346–2353, 2018.

[15] J. Yang, J. Man, M. Xi, *et al*., "Precise measurement of position and attitude based on convolutional neural network and visual correspondence relationship," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2030–2041, 2020.

[16] Islam. Md Amirul, Sen Jia, *et al*. "How much position information do convolutional neural networks encode?" *Int. Conf. Learn. Represent.,* pp. 1-11, 2020.

[17] Y. Zhao, G. Wang, *et al.*, "Self-supervised visual representations learning by Contrastive Mask Prediction," *IEEE/CVF Int. Conf. Comput. Vis.*, pp. 10140-10149, 2021.

[18] Jeong, Jisoo, et al. "Consistency-based semi-supervised learning for object detection." *Adv. Neural Inform. Process. Syst.*, pp. 10759–10768, 2019.

[19] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, pp. 7482-7491, 2018.

[20] S. Woo, J. Park, J.-Y. Lee, *et al*., "CBAM: Convolutional Block Attention Module," *Europ. Conf. Comput. Vis*, pp. 3–19, 2018.