2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Object Pose Estimation by Camera Arm Control Based on the Next Viewpoint Estimation

Tomoki Mizuno[1], Kazuya Yabashi[2], Tsuyoshi Tasaki[3]

***Abstract*— We have developed a new method to estimate a Next Viewpoint (NV) which is effective for pose estimation of simple-shaped products for product display robots in retail stores. Pose estimation methods using Neural Networks (NN) based on an RGBD camera are highly accurate, but their accuracy significantly decreases when the camera acquires few texture and shape features at a current view point. However, it is difficult for previous mathematical model-based methods to estimate effective NV which is because the simple shaped objects have few shape features. Therefore, we focus on the relationship between the pose estimation and NV estimation. When the pose estimation is more accurate, the NV estimation is more accurate. Therefore, we develop a new pose estimation NN that estimates NV simultaneously. Experimental results showed that our NV estimation realized a pose estimation success rate 77.3%, which was 7.4pt higher than the mathematical model-based NV calculation did. Moreover, we verified that the robot using our method displayed 84.2% of products.**

I. INTRODUCTION

Currently, the labor shortage in retail stores is becoming severe, and there is a growing expectation for the automation of product display using robots [1]–[3]. Automating product display requires the estimation of object poses, typically using color and depth images acquired with RGBD cameras [4][5]. Recently, methods using Neural Networks (NNs) with color and depth images as inputs have been developed to estimate poses with high accuracy [6]–[8]. Among them, PYNet [6] accurately estimates the poses of simple shaped objects commonly found in retail stores, such as rectangular prisms, cylinders and triangular prisms. However, general pose estimation methods decrease accuracy depending on the position of the camera relative to the object, especially for unknown objects not included in the training data. The pose estimation accuracy decreases when there is less information available from the camera due to the product's pose. Especially for simple-shaped objects, it is difficult to estimate the pose when only images like Fig. 1 are obtained. Therefore, estimating and moving to the next viewpoint that increases texture information improve the pose estimation accuracy.

In SLAM (Simultaneously Localization And Mapping) and SfM (Structure from Motion), there are many studies on determining the next viewpoint to increase the amount of information obtained from sensors [9][10]. For example, Simon Kriegel et al. increased the amount of information by searching for areas where long edges could be obtained in images, efficiently conducting SfM [9] . However, as shown in Fig. 1, it is often impossible to uniquely determine the direction in which information increases for simple-shaped objects. Therefore, this study tackles the novel challenge of estimating an effective next viewpoint for pose estimation of simple-shaped objects with the goal of automating product display by robots. To address the challenge, this study focuses on the relationship between object pose and effective next viewpoints. If the pose can be correctly estimated, it is possible to correctly estimate the effective viewpoint. Conversely, the direction of the effective viewpoint can also work as a guideline for pose estimation. This study develops a new method that allows the pose estimation NN to simultaneously estimate a next viewpoint, realizing the automation of product display.

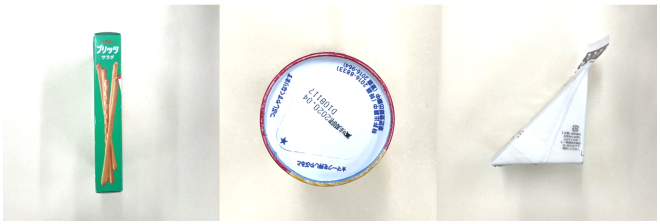

Fig. 1. difficult example of pose estimation

The academic contributions of this study are as follows:

- Developed a new NN that estimates a next viewpoint simultaneously with the pose estimation.
- Demonstrated that pose estimation accuracy improves by simultaneously estimating a next viewpoint.
- Showed that the next viewpoint estimation by the NN considering the pose is effective compared to the next viewpoint estimation based on the mathematical model.
- Implemented the developed next viewpoint estimation method into a robot, demonstrated improving the number of displaying products successfully.

II. RELATED RESEARCH

A. RGBD Camera Pose Estimation

With the release of the LINEMOD dataset [11], many NNs for the pose estimation using RGBD cameras have been developed, such as DeepIM [12], DenseFusion [7] and FFB6D

*This work was not supported by any organization
[1]Meijo University, 1-501 Shiogamaguchi, Tenpaku-ku, Nagoya, Aichi, Japan, 200442160@ccmailg.meijo-u.ac.jp
[2]Meijo University, 1-501 Shiogamaguchi, Tenpaku-ku, Nagoya, Aichi, Japan, 190442146@ccmailg.meijo-u.ac.jp
[3]Meijo University, 1-501 Shiogamaguchi, Tenpaku-ku, Nagoya, Aichi, Japan, tasaki@meijo-u.ac.jp

[8]. Methods with the high accuracy in the LINEMOD dataset [11] estimate the poses of complex-shaped objects accurately. However, for simple-shaped objects with few shape features, as shown in Fig. 1, the accuracy decreases.

PYNet [6] has been more accurate than FFB6D [8], which once achieved the highest accuracy in the LINEMOD dataset, in the pose estimation of simple-shaped objects. The representative shapes of simple-shaped objects are rectangular prisms, cylinders and triangular prisms as shown in Fig. 2. PYNet [6] estimates the surface on which the object is grounded and the angle around the axis perpendicular to the grounding surface. PYNet [6] calls the surface "poseclass" and the angle "yaw angle". By estimating the poseclass, the 3D pose estimation problem is simplified to a 1D yaw angle estimation problem, which improves the pose estimation accuracy. The poseclass is estimated by solving a classification problem with NNs, dividing rectangular prisms, cylinders, and triangular prisms into 6, 8 and 5 classes, respectively, as shown in Fig. 2. PYNet [6] divides the sides of cylinders into 6 classes because it thinks 60deg is the necessary resolution for product display by a robot. The left side of Fig. 2 shows the net of each simple shape product and the assigned poseclasses to the unfolded product surfaces. The right side of Fig. 2 shows images of the products taken from above when the assigned poseclasses are grounded. By estimating the poseclass and outputting the yaw angle as shown in Fig. 3, the 3D pose is determined. In this study, we refer to PYNet [6] architecture for the purpose of display simple-shaped objects by robots.

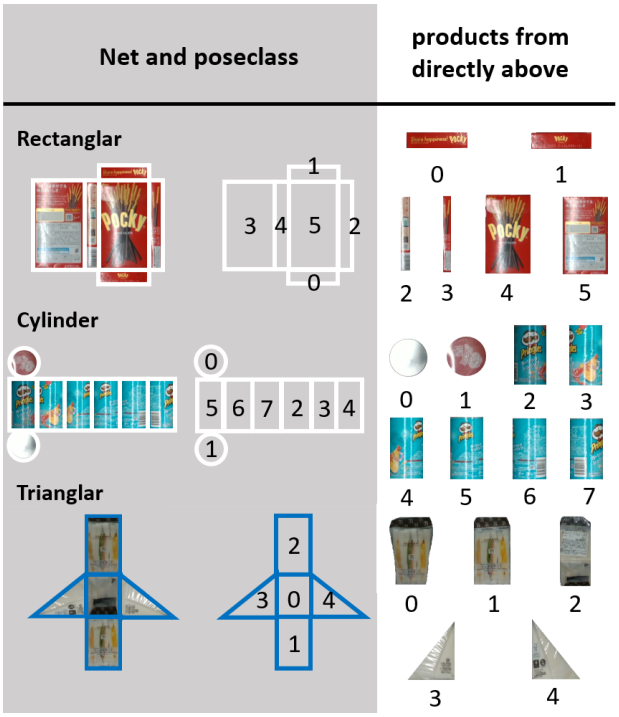

Fig. 2. poseclass of the simple-shaped product

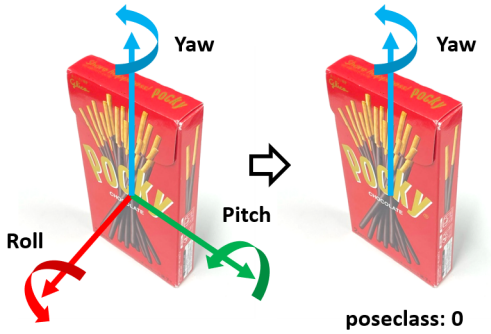

Fig. 3. pose estimation using poseclass

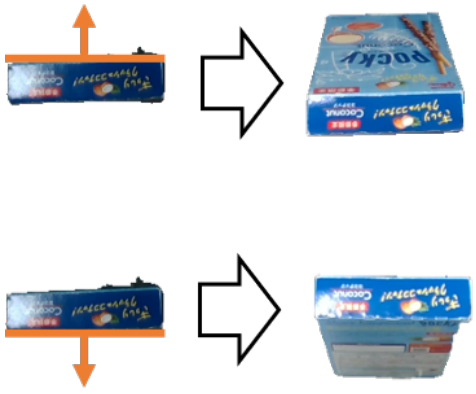

Fig. 4. example of camera motion

B. Next Viewpoint Estimation

The next viewpoint estimation has been extensively researched in SLAM and SfM. A simple edge-based method uses the direction of long edges, where information is abundant [9]. The edge-based method [9] detects the longest edge observable from the current viewpoint and searches for the next viewpoint in the direction of the longest edge. However, the simple-shaped objects often have multiple long edge. As shown in Fig. 4, moving the camera in the direction of the product's front may increase accuracy, but there are cases where moving to the backside. Therefore, even if the viewpoint is moved in the direction of long edges, the accuracy may not always improve in pose estimation.

In SfM, there is an outlier-based method [10] that moves the viewpoint to areas with many outliers in the 3D point cloud. The outlier-based method is utilized for 3D modeling objects in SfM, because the area with many outliers is the complex shape area and requires many observations. However, it is difficult to apply the outlier-based method to the pose estimation of simple-shaped objects because simple-shaped objects have fewer outliers.

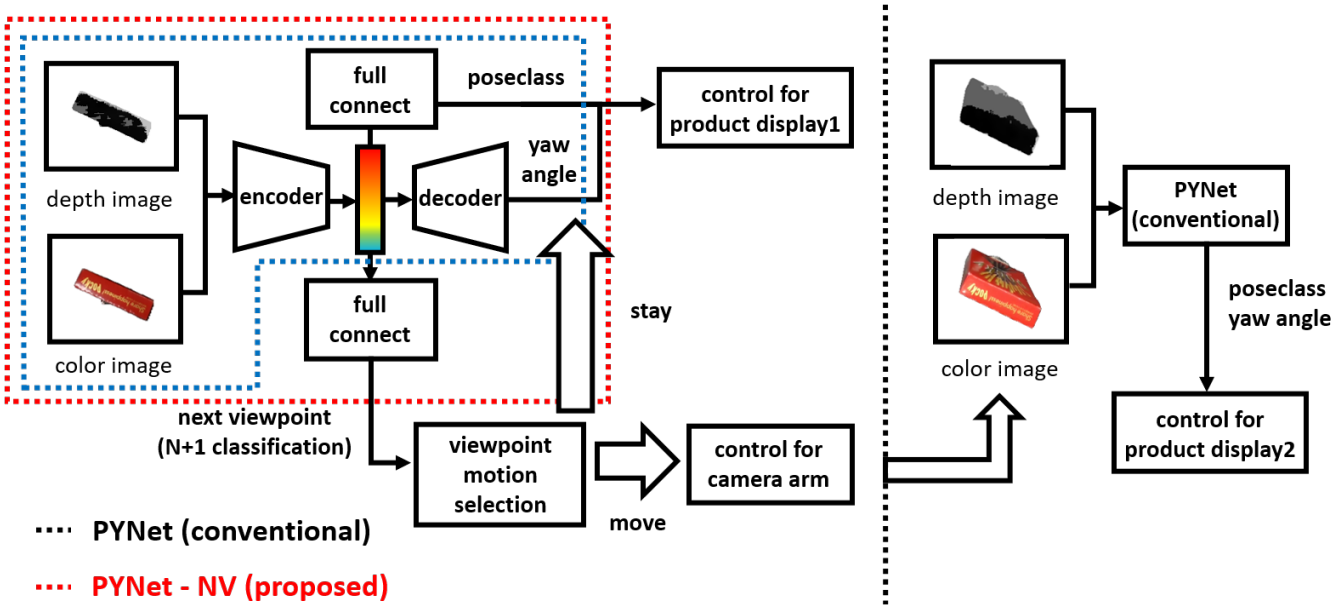

Fig. 5. system using PYNet estimating Next Viewpoint (PYNet-NV)

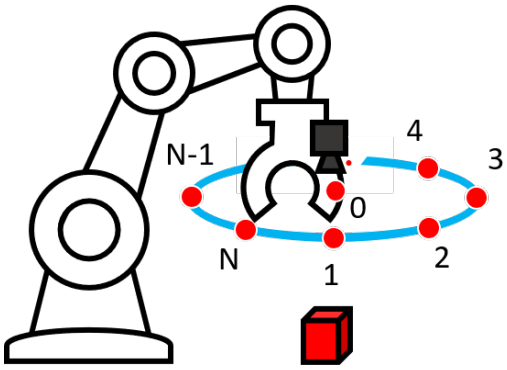

Fig. 6. next viewpoint candidates

C. Positioning this study

There are many studies on the pose estimation and next viewpoint estimation [6]–[10]. However, there has been no research on the next viewpoint estimation to improve the pose estimation accuracy of simple-shaped objects. This study develops a new NN that simultaneously estimates the pose and next viewpoint.

III. Proposed Method

The system structure that uses our proposed method is shown in Fig. 5. In this study, the viewpoint is estimated by the pose estimation NN to control the camera arm. This chapter explains the proposed NN and the training method for viewpoints in Section Ⅲ-A and Ⅲ-B, respectively.

A. PYNet-NV and Arm Control

The new method making PYNet [6] estimate the viewpoint is called PYNet-NV (PYNet [6] estimating Next Viewpoint). The architecture of PYNet-NV and the structure of the developed system are shown in Fig. 5. In this study, we refer to the PYNet's architecture for the pose estimation of simple-shaped objects because of its high accuracy. In our system, PYNet-NV estimates the effective viewpoint for the pose estimation, and after moving to the estimated viewpoint, the second pose estimation is performed. The conventional PYNet [6] is used for the second pose estimation. When the viewpoint that the robot doesn't have to move is estimated, the first estimated pose is used. To shorten the operation time by the robot, the viewpoint movement is limited to once. The robot controls the arm based on the results of the first or second pose estimation, respectively, in "control for product display 1 or 2" in Fig. 5, which performs product display. In this study, the initial viewpoint for the pose estimation is directly above the object which is detected by object detection methods [13][14]. If candidates of next viewpoints involve positions that require complex three-dimensional movement, only robots with high degrees of freedom can work. Therefore, the next viewpoint candidates are N viewpoints which set evenly on the circle with radius r that can be moved flatly as shown in Fig. 6. Note that, the candidate also includes the initial viewpoint directly above the object, as shown in Fig. 6 "0". Therefore, the next viewpoint estimation can be solved as an $N+1$ classification problem.

PYNet-NV leverages features effective for the pose estimation by branching from the encoder layer of PYNet [6]. The branched viewpoint estimation part is realized with a single fully connected layer.

B. Training Method for Viewpoint Estimation Branch

PYNet-NV is trained to select a viewpoint effective for the pose estimation. In the case of simple shapes, the information effective for pose estimation is the texture. Therefore, PYNet-NV trained to select a viewpoint where the robot can see surfaces with complex textures, namely those with many edges in the images. Teaching data for the next viewpoint is created based on the edge amount. Specifically, an $N+1$ dimensional one-hot vector, where only the element v becomes 1, is used as the teaching data for the next viewpoint estimation, as shown in (1).

$$v = argmax_{i \in V} e_i \tag{1}$$

Here, e_i denotes the number of pixels indicating edges in the image obtained at the next viewpoint i, and V denotes the set of next viewpoints. Note that, for each shape, the direct-above viewpoint is selected regardless of the edge amount when the difference of an object area in the image between before and after moving is small. Specifically, when the object grounding surface is poseclass p that satisfies (2), a one-hot vector with $v = 0$ is used as the teaching data.

$$p = argmin_{k \in P}(max_{i \in V} a_i^{(k)} / a_0^{(k)}) \tag{2}$$

Here, P denotes the set of poseclass, and $a_i^{(k)}$ denotes the area of the detection rectangle of the object that can be observed from a viewpoint i when the object is grounded in poseclass k. An example of the teaching data for $N = 4$ is shown in Fig. 7. It can be seen that the image obtained from viewpoints where the front of the product has many edges, which is more effective for the pose estimation than the direct-above viewpoint. Since a next viewpoint is estimated as a classification problem, the cross-entropy L_v is used as the loss function, as shown in (3).

$$L_v = -\sum_j \mathbf{t}(j) \log \mathbf{z}(j) \tag{3}$$

In (3), $\mathbf{t}$ denotes the teaching data, and $\mathbf{z}$ denotes the vector indicating the estimated next viewpoint.

IV. Evaluation on Pose Estimation Accuracy

A. Experimental Setup

To confirm the effect of the viewpoint estimation with the pose estimation, we compare following 3 methods.

- PYNet-NV
- the edge-based method [9]
- PYNet [6]

PYNet-NV and the edge-based method change a view point maximum once, and candidates of viewpoints of both methods are same. PYNet [6] estimates a pose from only direct-above viewpoint. The products used for training by PYNet-NV and PYNet [6] are shown in Fig. 8. Similar to PYNet [6]'s evaluation, this study uses rectangular prisms, cylinders and triangular prisms. The test is performed by leave-one-out cross-validation. That is, three types of products shown in the train or test column of Fig. 8 are used for training, and the remaining one type is used for test. Therefore, the test products are unknown products. The products shown in the validation column of Fig. 8 are used for validation during training. The NN parameters with the highest pose estimation accuracy on the validation data are used for test. The data used for training, test and validation is obtained by capturing the RGBD images of the products with an RGBD camera (RealSense) from each viewpoint. During capturing, each product is rotated from 0 deg to 359 deg in 1-degree increments for each poseclass. As a result, we can obtain 360 sets of color and depth images of each product for each poseclass. The 360 sets of images are obtained for all viewpoints. The obtained image is cropped to a size of 224×224 by using object detection [13][14]. Therefore, the center of the image corresponds to the center of the product, approximately. Considering the manipulation range of the robot used in Chapter 5, the number of viewpoints N and the radius r are set to 4 and 0.15m, respectively. The evaluation index uses the angle error e of the 3D pose. Using the angle error e, the proportion of outputs that satisfies (4) for all test data was evaluated as the success rate. ϕ is the arbitrarily set as the permissible angle error.

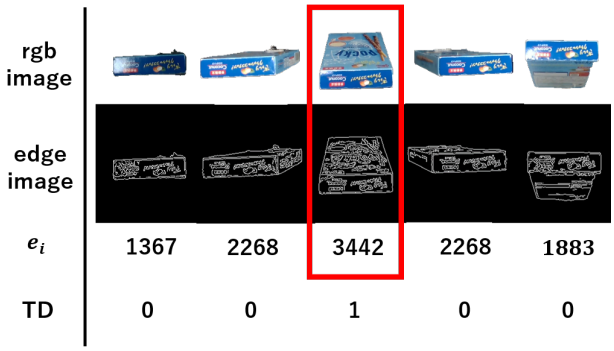

Fig. 7. example teaching data (TD)

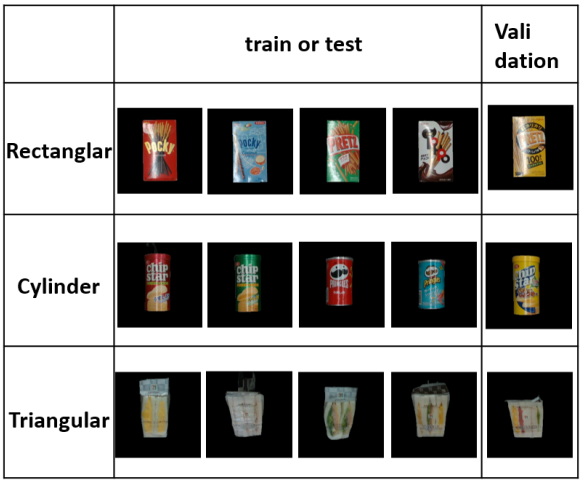

Fig. 8. products used in the experiment

$$e <= \phi \tag{4}$$

In this study, ϕ is set from 0 deg to 30 deg in 0.1-degree increments to calculate the success rate.

B. Experimental Results and Discussion

The success rates of PYNet [6], the edge-based method [9] and PYNet-NV are shown in Fig. 9. Following the World Robot Summit [15], the discussion is based on the accuracy rate Φ of 30 deg. As shown in Fig. 9, the success rates of PYNet [6], edge-based method and PYNet-NV were 68.0%, 69.9%, and 77.3%, respectively. The pose estimation accuracy improved by changing viewpoint because the accuracy of edge-based method and PYNet-NV was higher than that of PYNet [6]. Moreover, PYNet-NV improved accuracy by 7.4 points compared to the edge-based method. Therefore, it was found that estimating the next viewpoint by NN improved the pose estimation accuracy. Especially, the accuracy was higher when poseclass 0 and 1 of rectangular prisms and cylinders were grounded, which had low features of the shape. Furthermore, the success rates of PYNet-NV and PYNet [6] are shown in Fig. 10 when the evaluation was performed only with the test data for poseclass satisfying (2). In this case, PYNet-NV does not change the viewpoint and estimates pose just once. The poseclass satisfying (2) represents surfaces with little shape change even if the viewpoint change, such as the front of the object, where relatively large surfaces are visible. As shown in Fig. 10, even without changing a viewpoint, the success rate of PYNet-NV was 82.3%, improving by 3.5 points over PYNet [6]. Therefore, it is considered that there is an effect improving the pose estimation accuracy by estimating the next viewpoint and pose simultaneously.

V. Evaluation of Product Display by Robots

A. Experimental Setup

To examine the impact of the improved pose estimation accuracy on the product display robot, we compare the robot using PYNet-NV with the robot using PYNet [6]. The training setup for PYNet-NV and PYNet [6] is the same as in Chapter 4. The products used in this evaluation are those in the first column of each shape shown in Fig. 7. A yaw angle is determined randomly, and evaluations are conducted twice for each poseclass. The process of displaying the product to the shelf by the robot is shown in Fig. 11. In Fig. 11(a), the first pose estimation and next viewpoint estimation are performed. Then, in Fig. 11(b), the robot moves to the estimated next viewpoint and performs the second pose estimation. From Fig. 11(c) to (d), the product is displayed to the shelf so that the front of the product is visible based on the second pose estimation result. The evaluation is based on whether the product is successfully displayed facing the front on the shelf, as shown in Fig. 12.

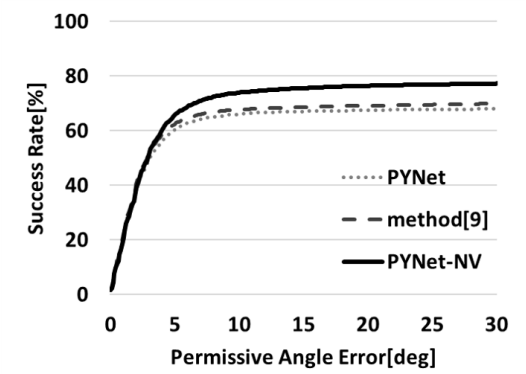

Fig. 9. success rate

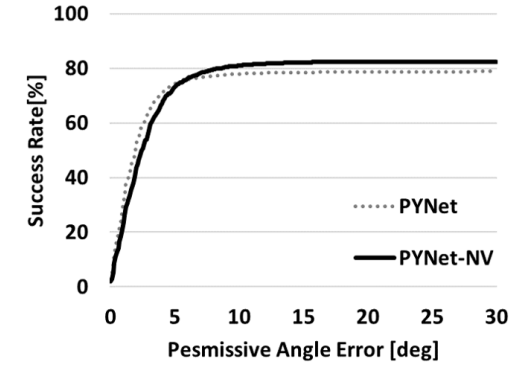

Fig. 10. success rate in non-moving poseclass

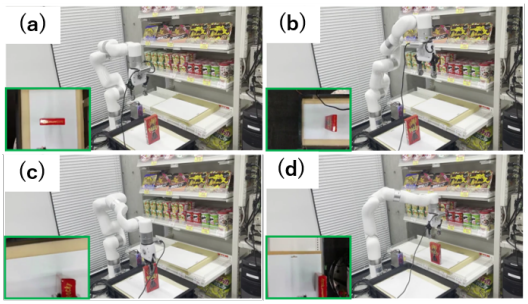

Fig. 11. product display flow

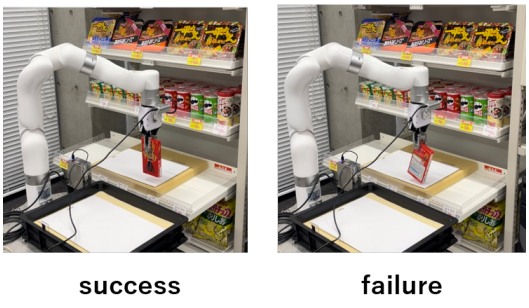

Fig. 12. example of success and failure

B. Products used in the experiment

The display results of PYNet-NV and PYNet [6] are shown in Table.1. The product display was successful in 29

Fig. 13. poseclass 1 of all products

Fig. 14. poseclass 3 of rectanglar

TABLE I
RESULTS OF DISPLAYING PRODUCTS

	Rectangular	Cylinder	Trianglar	Total
PYNet	66.7%	87.5%	70.0%	76.3%
PYNet-NV	83.3%	87.5%	80.0%	84.2%

out of 38 times (76.3%) by using PYNet [6]. The product display was successful in 32 out of 38 times (84.2%) by using PYNet-NV. Therefore, changing viewpoint based on the viewpoint estimation result is considered to contribute to the improvement of displaying performance.

Next, we want to discuss the number of success for each poseclass of the product. The number of successful product display for poseclass 1 of all products (Fig. 13) increased by using PYNet-NV compared to by using PYNet [6]. Since poseclass 1 is a surface with little information, the changing viewpoint is considered to contribute to the product display performance. However, as shown in Fig. 14, when poseclass 3 of the rectangular prism was grounded, the number of successful product display was zero. As shown in Fig. 14, products grounded in poseclass 3 have significantly different designs depending on the products, such as the presence or absence of barcodes. Even if an effective surface for the pose estimation becomes visible due to changing a viewpoint, pose estimation errors occur. This is because the most visible surface close to the camera includes distinctive designs and affects the pose estimation. Therefore, in the future, we plan to develop a pose estimation NN that focuses on the newly visible surfaces after changing a viewpoint.

VI. CONCLUSIONS

This study tackled the challenge of estimating an effective viewpoint for the pose estimation of simple-shaped objects with the goal of developing product display robots. We have developed PYNet-NV that estimates the product pose and next viewpoint simultaneously. The accuracy of the pose estimation based on a viewpoint estimated by using PYNet-NV was 77.3%, improving by 7.4 points compared to the previous mathematical model-based method. Moreover, we have developed a product display robot using PYNet-NV. Our robot has successfully displayed 84.2% of the products placed randomly. In the future, we plan to develop NNs that focus on surfaces that become newly visible after changing the viewpoint.

ACKNOWLEDGMENT

A part of this work was supported by JSPS KAKENHI (grant number JP23K11157)

REFERENCES

[1] R. B. Rusu, N. Blodow and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in 2009 IEEE International Conference on Robotics and Automation, pp. 3212-3217, 2009.
[2] L. S. F. Tombari and S. Salti, “Unique signatures of histograms for local surface description,” in European Conference on Computer Vision, pp. 356-369, 2010.
[3] F. Tombari, S. Salti and L. Di Stefano, “A combined texture-shape descriptor for enhanced 3d feature matching,” in 2011 18th IEEE International Conference on Image Processing, pp. 809-812, 2011.
[4] G. A. Garcia Ricardez, S. Okada, N. Koganti, A. Yasuda, P. M. Uriguen, Eljuri, T. Sano, P.-C. Yang, L. El Hafi, M. Yamamoto, J. Takamatsu and T. Ogasawara, “Restock and straightening system for retail automation using compliant and mobile manipulation,” Advanced Robotics, pp. 235-249, 2019.
[5] R. Sakai, S. Katsumata, T. Miki, T. Yano, W. Wei, Y. Okadome, N. Chihara, N. Kimura, Y. Nakai, I. Matsuo and T. Shimizu, “A mobile dual-arm manipulation robot system for stocking and disposing of items in a convenience store by using universal vacuum grippers for grasping items,” Advanced Robotics, pp. 219-234, 2019.
[6] K. Fujita and T. Tasaki, “PYNet: Poseclass and Yow Angle Output Network for Object Pose Estimation,” Journal of Robotics and Mechatronics, Vol. 35, No. 1, pp.8-17, 2023.
[7] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei and S. Savarese, “DenseFusion: 6D object Pose Estimation by Iterative Dense Fusion,” International Conference on Computer Vision and Pattern Recognition, pp. 3343-3352, 2019.
[8] Y. He, H. Huang, H. Fan, Q. Chen and J. Sun, “FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Estimation,” International Conference on Computer Vision and Pattern Recognition, pp. 3002-3012, 2021.
[9] S. Kriegel, C. Rink, T. Bodenmuller and M. Suppa: “Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects,” J Real-Time Image Proc, Vol. 10, No. 4, pp. 611-631, 2015.
[10] J. Hu and P. R.Pagila, “View Planning for Object Pose Estimation Using Point Clouds: An Active Robot Perception Approach,” IEEE Robotics and Automation Letters, Vol 7, No. 4, pp9248-9255, 2022
[11] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige and N. Navab, “Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes,” Asian Conference on Computer Vision, pp. 548-562, 2012.
[12] Y. Li, G. Wang, X. Ji, Y. Xiang and D. Fo, “DeepIM: Deep Iterative Matching for 6D Pose Estimation,” Vol. 11210, pp. 695-711, 2018.
[13] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane and M. Jagersand, “U2-net:Going deeper with nested u-structure for salient object detection,” Pattern Recognition, Vol. 106, pp. 107404, 2020.
[14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollar and R. Girshick, “Segment Anything,” In International Conference on Computer Vision, 2023.
[15] H. Okada, T. Inamura and K. Wada, “What competitions were conducted in the service categories of the world robot summit?,” Advanced Robotics, Vol. 33, pp. 900-910, 2019.