Proceedings of the 2019 IEEE/SICE
International Symposium on System Integration
Paris, France, January 14-16, 2019

Toward 6 DOF Object Pose Estimation with Minimum Dataset

Kota Suzui[1,2], Yusuke Yoshiyasu[1], Antonio Gabas[1,3], Fumio Kanehiro[1], Eiichi Yoshida[1]

Abstract— **In this research, we propose a method for estimating 6 DOF object pose (3D orientation and position), based on convolutional neural networks (CNN). We propose RotationCNN that predicts 3D orientation of the object. The position of the object is estimated using an object detection CNN that predicts the class of the object and bounding box around it. Unlike the method that trains CNNs using a large-scale database, the proposed system is trained with minimum dataset obtained in a local environment that is similar to where the robot is used. With the proposed semi-automated dataset collection techniques based on a web camera and AR markers, users in different environment will be able to train the network suited for their own environment relatively easily and quickly. We believe that this approach is suitable for a practical robotic application. The results on 3D orientation prediction using RotationCNN show the average error of 18.9 degrees, which we empirically found that it is low enough as an initial solution to successfully run the iterative closest point (ICP) algorithm that uses depth data to refine the pose obtained with CNNs. The effectiveness of the proposed method is validated by applying the method to object grasping by a robot manipulator.**

I. INTRODUCTION

In recent years, industrial robots have led to automation in factory lines. The next avenue for this trend is such that a robot is expected to play a role in a less structured environment such as shops, households and construction. In such cases, robots will need to be more autonomous, having good perception to recognize 6 DOF pose (3D position and orientation) of an object that is randomly placed in environment. Since a certain object such as a tool has a certain orientation to be grasped and used, knowing 3D orientation of the object accurately is a very important issue in robotic manipulation, which will be the focus of this paper. In the past years, methods using AR markers and 3D laser scan has been developed to estimate object pose. In commercial areas, however, object changes repeatedly, which makes it difficult to manage AR markers. 3D laser scanners also have a disadvantage with its long acquisition time which makes it difficult to use it online.

In this paper, we propose a novel 6 DOF object pose estimation algorithm as the base function of robotic manipulation for object detection and localization. Our method uses RGB images as an input to the convolutional neural network (CNN). In particular, we propose RotationCNN that predicts 3D orientation from a single RGB image such that the robot can grasp an object from a specific orientation. Considering robotics applications in various environments, we propose a new method to easily obtain training data using web cameras. The pose estimated with RotationCNN is refined using the iterative closest point (ICP) algorithm to align with 3D point cloud data in order to be used for robotic manipulation.

The main contributions of this work are summarized as follows:

1 We develop a system that integrates object detection, 6 DOF object pose estimation and robotic manipulation, which can achieve grasping of an object from RGB-D data.

2 We propose a method for easily capturing and annotating an image dataset for using a web camera and AR marker based on the 3D reconstruction method called visual hull to semi-automate the annotation process.

II. RELATED WORK

Iterative closest point (ICP) is a widely-used algorithm in object localization and pose estimation [1], which minimizes the distance between two sets of point cloud data. ICP iteratively alternates the optimization of a rigid transformation and the closest point search from one set of point cloud data to the other. This process is continued until the two data is aligned. However, ICP tends to converge to local minima, making it heavily dependent on the initial state. If the initial state is largely dislocated, not only will the computational cost increase, but the possibility of not being able to converge to the correct position also increases. Yang t al. [2] proposed Globally Optimal ICP (Go-ICP) as a solution for this problem, which is an algorithm that enables ICP to converge to global minima. Nevertheless, the calculation cost of Go-ICP is high, which makes it difficult for practical usage such as robot grasping.

Methods for detecting an object inside 2D image using CNN have been introduced with fast and robust detection. These methods use a single CNN to detect multiple objects simultaneously, giving a bounding box for each object while also classifying them into categories. You only look once (YOLO) by Redmon et al. [3] and Single shot multi-box detector (SSD) by Liu et al. [4] are examples of such object detectors with robust online performance.

In computer vision and computer graphics fields, techniques have been proposed for object pose estimation and viewpoint estimation using CNN. In robotics field, deep learning has been used to improve perception capability of robots, especially in the field of grasping and manipulation.

1 Kota Suzui, Yusuke Yoshiyasu, Antonio Gabas, Fumio Kanehiro and Eiichi Yoshida are with CNRS-AIST JRL (Joint Robotics Laboratory) UMI3218/RL, National Institute of Advanced Industrial Science and Technology (AIST), Japan.

2 Kota Suzui is with the Department of Mechanical Systems Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan.

3 Antonio Gabas is with the University of Tsukuba, Ibaraki, Japan.

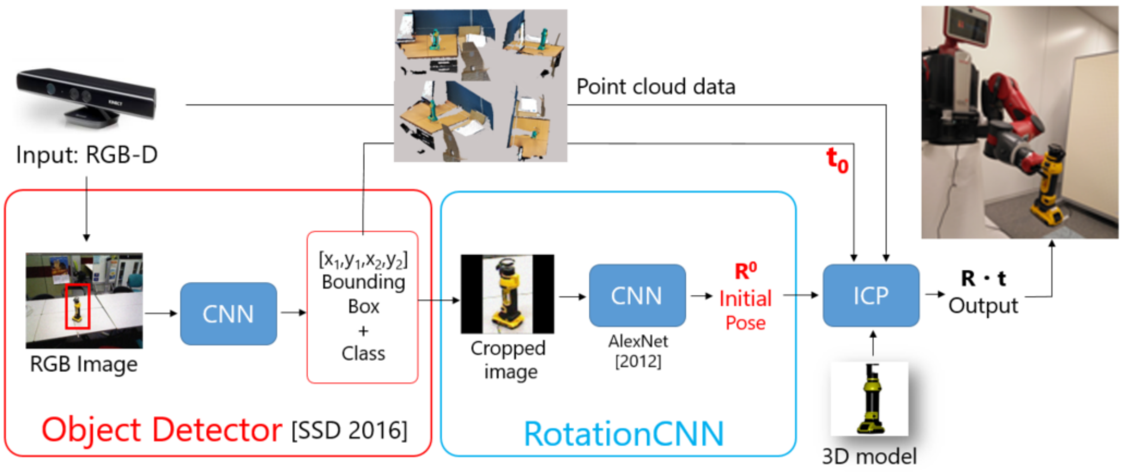

Fig. 1. Overview of the proposed method. The red box indicates the object detector which gives the location and the class of the target object. The blue box is RotationCNN that estimates the 3D rotation of the object. The final 3D rotation of the object is obtained with the ICP algorithm, using the initial pose, point cloud data, and CAD model.

The early techniques have tried to detect the graspable points or parts in a general manner without knowing the class of a object[6][7][8][9][10].However, actually using these techniques for robot grasping an object seems to be still challenging. Pavlakos et al. [11] estimated object poses by showing semantic key points on heat maps. This method is robust in conditions such as when the object is occluded. However, the key points have to be defined manually, which requires a more efficient way to obtain training data. SSD-6D by Kehl et al. [12] and poseCNN by Xiang et al. [13] are also examples of pose estimation methods using CNN. They use large dataset to train CNN, while our method is enabling pose estimation using minimum dataset. Smaller dataset is easier to collect, which will be an advantage.

III. OVERVIEW OF THE SYSTEM

The overview of the system, which integrates from object detection to robot grasping is shown in Fig. 1. Six DOF object pose comprises 3D orientation and 3D position. In this paper, we use rotation to represent 3D orientation. We first use RGB-D sensors such as Kinect to obtain data of the target object. The RGB image is used as input to the object detector to get the bounding box and class of the object. Using the bounding box, the original RGB image is cropped out so that only the object region remains. The cropped images are then fed as input for the rotation regressor to calculate the 3D orientation. We use the calculated pose as the initial pose for the Iterative Closest Point (ICP) algorithm, which we use for refinement to obtain the final pose of the object. The robot will use grasping points defined on the 3D model as guide to generate and execute the grasping motion. Note that our main focus in this paper is to create a rotation regressor using CNN, aiming to get initial pose inside the permissible range quickly so that the ICP algorithm will converge to the correct solution. We would also like to mention that all of the 3D models used for ICP are built beforehand, and thus users are expected to construct 3D models for their target objects. Note that the proposed method can be fitted to detect multiple class of object. When classifying the object with the object detector, the result will be used to select the corresponding 3D model. This will allow the system to estimate and visualize object pose of the targeted object accordingly.

IV. METHOD

A. *RotationCNN network architecture*

In our research, we modified AlexNet [16], which is originally proposed for object classification, into object orientation estimation task. The original network consists of five convolutional layers and three fully connected layers, containing the total of eight layers. We changed the softmax loss function used for classification to the l2-norm regression loss function in order to output nine parameters. We chose to use the shallower network than the recent ones, such as ResNet, because our dataset is relatively small for learning deeper networks and we prefer fast prediction performance. When testing time, we further perform polar decomposition using the singular value decomposition (SVD) to eliminate shear/scaling. We also remove the symmetrical flip, which makes the coordinate frame to left-hand instead of the standard right-hand system, by checking the determinant of the matrix and changing the sign to make it always positive. These operations will make the output transformation to become a perfect rotation.

B. *Iterative closest point*

With SSD and RotationCNN, we can estimate the 6 DOF pose of the object relatively accurately. The pose is further improved by using iterative closest point (ICP) to optimize a transformation (rotation + translation). The first step searches the closest points from the model to the point clouds. Then, the transformation is optimized using the closed form formulation.

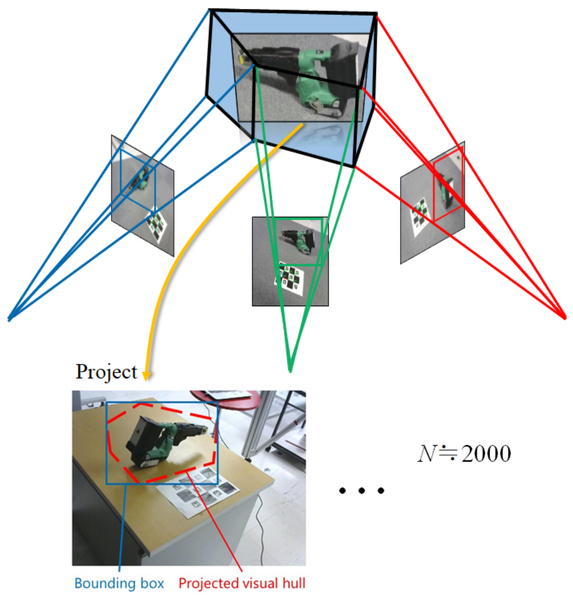

Fig. 2. Visual hull was used to provide bounding boxes for our training images. Only 8-10 manually annotated bounding boxes are needed to obtain bounding boxes for all of the rest of the images.

C. *Object detection using single shot multi box detector*

Prior to inputting an image into RotationCNN, we detect bounding boxes surrounding objects and crop the original images so that only the region of a certain object is in the image. To do so, we use the single shot multi box detector (SSD) framework. Here, the original image is first resized to the size of 300-by-300 and input into SSD. SSD outputs the bounding boxes, classes and probabilities of those class predictions for the objects in the image.

V. TOOL DATASET CONSTRUCTION

A. *Dataset acquisition methods*

Three types of techniques were used for constructing image datasets in this research.

1) Semi-automatic dataset annotation technique: First dataset was obtained using web cameras (webcam). Measuring ground truth 3D orientation data was done using ArUco markers developed by Garrido-Jurado et al.[14]. This setup eases the user from time consuming manual 6 DOF annotations using such as the Pascal annotation tool to align the 3D model to images.

To provide the bounding box annotation without efforts, we provide a semi-automatic method using 3D reconstruction method called visual hull, shown in Fig. 2. From the extrinsic camera calibration results, i.e., camera pose, we plot all the camera positions and select approximately 10 views which are distributed uniformly using the farthest sampling methods. The user is only required to annotate the bounding boxes for these 8-10 views. Next, the visual hull algorithm is used to reconstruct a volume around the object using these bounding boxes. The visual hull algorithm was originally proposed to carve the volume around the object within the image silhouettes, whereas we use the bounding boxes to roughly determine the volume around the object. Finally, the volume is projected onto all other images to automatically annotate bounding boxes of e.g., 2000 images.

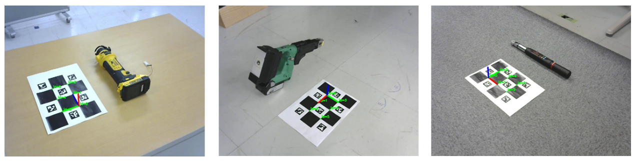

Fig. 3. Some examples of training dataset. The dataset was taken at different environments.

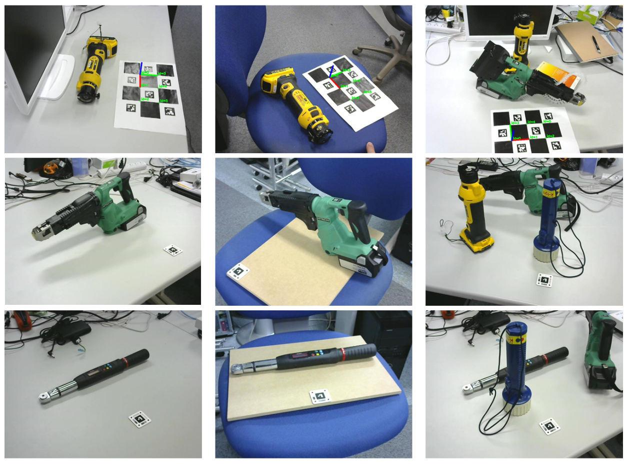

Fig. 4. Examples of images of testing dataset. AR markers were used here also for ground truth.

2) Other techniques: The second dataset was obtained using multi-view photography machine (3D MFP) to capture images of objects from multiple angles. The third dataset was images taken at random environments with different brightness (Random). The ground truth data for this dataset was measured by using PASCAL 3D annotation.

When training these datasets, random cropping was used for data augmentation. This was done to train partial parts of the objects, as well as the full image of the object.

B. *Dataset splits*

1) Other techniques: Webcam dataset were taken at three different backgrounds, desk, tile, and carpet, while the object was standing and laying. Some examples are shown in Fig. 3. For images when the object is laying, the object was taken laying on both sides. However, we did not take any data sets for unrealistic positions, such as objects standing upside down. The images were taken from around 360 degrees, 200 images in one lap. Approximately 2000 images were taken in total. 3D MFP dataset were taken by rotating the object 10 degrees each, from 3 different elevation angles, resulting in total of 108 images. The Random dataset were taken at 50 random environments with different brightness. Furthermore, 15 different noises were added to each image to create a total of 750 images.

Different combinations of these dataset were made to evaluate the effectiveness of each dataset. The dataset was

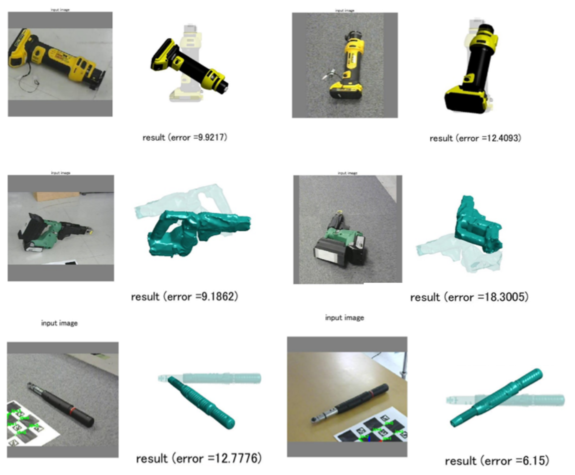

Fig. 5. The validation results for the 3 objects are shown above.

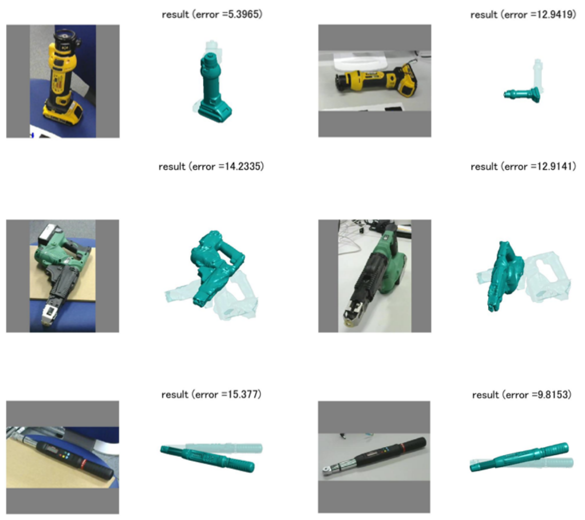

Fig. 6. The results of pose estimation using the test dataset are shown. The Webcam dataset was used for training.

split for training and validation in the proportion of 9:1 respectively.

2) Testing dataset: Testing dataset was obtained to quantitatively evaluate the effectiveness of training data. Examples of test dataset are shown in Fig. 4 These images were taken at different environments than the webcam training dataset. The test dataset features include different colored background and occluded conditions. Testing dataset included 253 images for yellow tool, 280 images for green tool, and 223 images for black tool. To obtain the ground truth pose angles of the yellow tool, ArUco markers were used again. However, with the test dataset for green and black tools, we used LentiMark, developed by Tanaka et al.[15]. LentiMark is smaller than the ArUco markers which prevents markers from showing in the input images. Evaluation was done by comparing the average degree error when training the network with various combinations of training data and using the testing data as input.

VI. RESULTS AND DISCUSSION

We ran our method on i7-6700K@4.00GHz with NVIDIA GTX 1080Ti. In order to quantitatively evaluate the accuracy of the network, we compared the error average of various combinations of dataset. The error average was first calculated using relative rotation between the estimated rotation $\mathbf{R}_{est}$ and the label $\mathbf{R}_{correct}$, $\mathbf{R}_{error} = \mathbf{R}_{est}\mathbf{R}^{T}{}_{correct}$. Then $\mathbf{R}_{error}$ was converted to axis angle. Finally, we take the norm of axis angle and convert it to degrees.

1) Pose prediction results: Some result examples of rotation estimation on validation dataset are shown in Fig. 5. Since the input images are validation dataset with similar background as training data, the estimation is possible with low error rates. Even for the image with different background and partial occlusion, the proposed method works relatively fine as shown in Fig.6. We estimated poses of three different types of tools, having different colors and shapes. The error average of the pose estimation using the test dataset is shown in Table I.

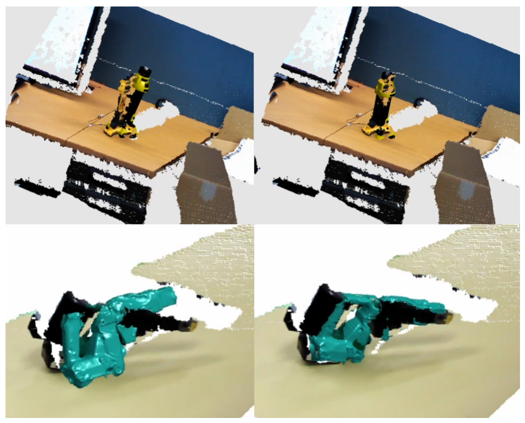

Fig. 7. The result when applying ICP after the RotationCNN. The left images are before ICP. The right images are after the ICP is executed. The 3D model is aligned with the point cloud by ICP.

After estimating the initial pose of the object, ICP is used to get the final pose. Fig. 7 compares the pose of the 3D model, before and after ICP is executed. The initial pose is accurate enough, which allows ICP to converge to the same pose as the point cloud.

2) Computational preformance: The average estimation time for RotationCNN was approx. 0.035 seconds. As mentioned in section IV, the drawback of representing the rotation with 9 parameters needs to be considered. As a result, the difference of calculation time during the testing time between axis angle and rotation matrix was insignificant, which only took 1.7% longer than when using axis angle. Hence, the advantage of getting higher accuracy received by using rotation matrix is more beneficial than the disadvantage of longer calculation time. SSD takes approx. 0.04 seconds. Combining SSD and RotationCNN, online pose prediction can be done with 15 fps. ICP however takes approx. 0.5 seconds. We wish to improve the performance of ICP using GPU parallelization in future work.

TABLE I

ERROR AVERAGE OF POSE ESTIMATION WITH TEST DATASETS FOR THE THREE DIFFERENT TYPE OF TOOLS WE USED.

Dataset	Error average [Degrees]
tool_yellow	25.6
tool_green	30.6
tool_black	58.4

TABLE II

ERROR AVERAGE OF ROTATION ESTIMATION WITH TEST DATASET. DIFFERENT COMBINATION OF DATASET WAS USED TO EVALUATE THE EFFECTIVENESS OF EACH DATASET.

Dataset	Error average [Degrees]
Webcam	26.9
3D	66.1
Random	71.5
Webcam + 3D MFP	22.0
Webcam + Random	24.3
Webcam + Random + 3D MFP	18.9
Webcam + Random + 3D MFP (No augmentation)	20.8
Webcam + Random + 3D MFP (Not pretrained)	54.5

3) Dataset evaluation: In order to further more investigate the effectiveness of dataset, we prepared various type of dataset mentioned in Section 5 for the yellow tool. The evaluation of training dataset is shown in Table II. The results with the webcam dataset only already achieves accurate prediction with the average error of 27 degrees. On the other hand, the results of network trained with 3D MFP dataset only and Random images only both had poor scores. For both of these dataset, the number of images is small, which makes a network difficult to learn the features of the object to correctly estimate the pose.

The results with the lowest error average was obtained with the training dataset combination of webcam, Random, and 3D MFP. The various information included in Random and 3D MFP images have helped with achieving better scores. Random images include images with different brightness and noises, while 3D MFP images include a tool taken from different elevation angles. Images from different elevation angles seem to have some effect on accuracy, as we can see that the combination of webcam and 3D MFP marked better score than the combination of webcam and Random.

4) Effect of augmentation and pretraining: We also evaluated the effectiveness of augmentation and pre-training. The results show that data augmentation was effective as well. The pose estimation in occluded conditions were superior when the network was trained with webcam, Random, and 3D MFP, then trained by the same combination of dataset without the augmentations. We also learned that the estimated results show lower score when the network is untrained. We randomly changed the weights of the network while having the same structure as the trained network. This result showed that network trained with various features, in this case with ImageNet images, have higher performance level.

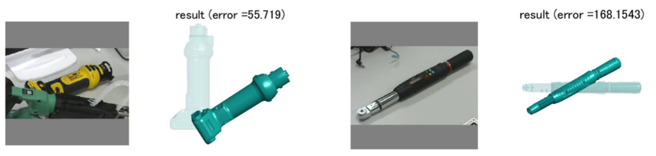

Fig. 8. When the object was placed in an occluded condition, the network failed to obtain accurate pose estimation. For the black tool, the network had difficulty recognizing the roll angle. As in the figure, the tool is facing up, but the network recognizes it as if the tool is facing down.

5) Failure cases: Some examples of failure case is shown in Fig.8. The most critical issue is when the object is partially occluded. This may be caused by the lack of visual features extractable from the image. For better pose estimation, either collecting more dataset in occluded conditions or modifying the data augmentation process may be required. For the black tool, estimating the roll pose sometimes failed. The lack of characteristic of the tool may have caused this.

VII. APPLICATION TO ROBOT GRASPING

In order to demonstrate the usefulness of the proposed method, we performed tests of our algorithm in a real robot grasping scenario. The first setup consists in a Baxter robot with a RGB-D sensor. The sensor has been calibrated and its position in the world with respect to the robot is known. The robot has a tool in front of him and the goal is to grasp the tool by a point and orientation that have been predefined using the object 3D model. The robot will obtain the RGB image and point cloud of the scene using the sensor and will input them together with the object 3D model to the aforementioned algorithm.

Once the position and orientation of the tool are known, we can select any point of the tool in the 3D object model and find the point's position in the real scene. This is especially useful because even if the point that we want to grasp is not present in the RGB image or point cloud, its position can be known. We select one point along the handle of the tool to be the grasping point and define the grasping orientation to be the same as the direction of the point's normal vector. The robot will first move the end effector to a pre-grasp position situated at 10 cm from the grasping point in the normal direction and then move in a straight line towards the grasping point.

Fig. 10 shows the results. In the first row we see 3 cases of detection of the tool's bounding box at different orientations. Second row shows the obtained point cloud and, in red, the initial rotation and translation provided from the neural network to the ICP algorithm. The model in blue is the final transformation resulting from the ICP algorithm. The green line shows the path that the end effector will cover from the pre-grasp point to the grasp point. Fig. 9 shows the robot performing the grasping motion with the yellow tool.

During this grasping motion generation, we gave order to the robot which arm to use. However, it is ideal to have the robot decide which arm to use for grasping by itself, which will be a subject for future work.

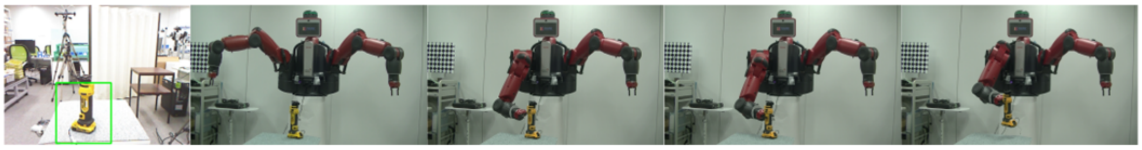

Fig. 9. Baxter robot performing the grasping action.

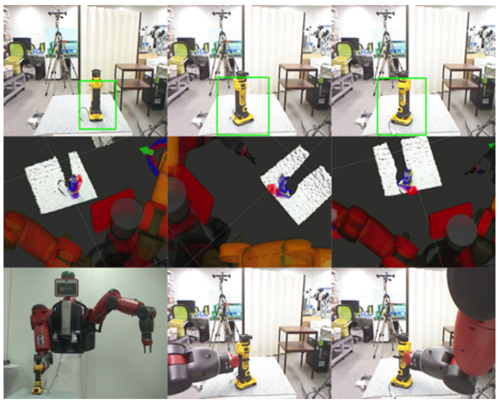

Fig. 10. Top row: RGB images of different tool poses with the detected bounding boxes. Middle row: Tool point cloud with the initial transformation obtained from the CNN (Red), the final transformation obtained from the ICP algorithm (Blue) and a green line representing the trajectory that the robot end effector will follow from the pre-grasp point to the grasp point. Bottom row: A real robot performing the grasping action on the item from the correct direction which illustrates the strength of our system.

VIII. CONCLUSION

We presented RotationCNN which can accurately estimate the rotation of objects from a single RGB image. A simple method to obtain training data was also introduced. Only minimum dataset is required for this system, allowing users to easily suit it to one's own environment. Representing the rotation with rotation matrix and applying orthogonality constraints made the accuracy increase.

Our system is currently applicable to images with similar background as training data. The issue is that the object detector (SSD) is not trained on images with a wide variety of background. On the other hand, by training the network with multiple background, RotationCNN were able to estimate the pose in new environment with different background. Furthermore, state-of-the-art object detectors are powerful enough to classify objects with similar shape, therefore makes the proposed algorithm feasible to detect objects with similar shape.

In future work, we will explore marker-less solution to 3D orientation annotations. This may be achieved by the techniques, such as structure from motion, so that the camera pose is known automatically. This approach may also provide a solution to the issue of 3D model construction.

As the dataset evaluation results showed, images with noise and with different elevation angles also contribute to high accuracy. Attempting to reproduce these features with just the web camera images will be part of future research. In order to enhance the completeness of this system, avoiding problems with occluded conditions is also a challenge left to solve in the future.

IX. ACKNOWLEDGMENT

We would like to thank Gentiane Venture from Tokyo University of Agriculture and Technology, and the members of GVlab for the discussion and suggestions given toward this project. Also, this work was partly supported by JSPS Kakenhi No.17K18420.

REFERENCES

[1] Besl, P., et al. "A Method for Registration of 3-D Shapes" Proc. of the IEEE Transactions on Pattern Analysis and Machine Intelligence, (1992), Vol. 14, pp.239-256.

[2] Yang, J., et al. "Solving 3D Registration Efficiently and Globally Optimally" Proc. of the IEEE International Conference on Computer Vision, (2013), pp.1457-1464.

[3] Redmon, J., et al. "You Only Look Once: Unified, Real-Time Object Detection" Proc. of the IEEE Computer Vision and Pattern Recognition, (2016), pp. 779-788.

[4] Liu, W., et al. "SSD: Single Shot MultiBox Detector" Proc. of the European Conference on Computer Vision, (2016), Vol. 9905, pp. 1-17.

[5] Kanezaki, A. et al., "RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints" arXiv: 1603.06208v3 [cs.CV], Nov. 2017.

[6] Ian Lenz, Honglak Lee, Ashutosh Saxena. "Deep learning for detecting robotic grasps" Proc. of the International Journal of Robotics Research Volume: 34 issue: 4-5, pp.705-724, 2015.

[7] Redmon, J., et al. "Real-Time Grasp Detection Using Convolutional Neural Networks" arXiv: 1412.3128v2 [cs.RO], Feb. 2015.

[8] Varley, J., "Generating multi-fingered robotic grasps via deep learning" Proc. of the IEEE International Conference on Intelligent Robots and Systems, (2015).

[9] Hatori, J., et al. "Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions" arXiv: 1710.06280v2[cs.RO], Mar. 2018.

[10] Mahler, J., et al. "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics" 2017.circuit devices (Patent style), U.S. Patent 3 62412, July 16, 1990.

[11] Pavlakos, G., et al. "6-DoF Object Pose from Semantic Keypoints" Proc. of the IEEE International Conference on Robotics and Automation, (2017), pp. 2011-2018.

[12] Kehl, W., et al. "SSD-6D:Making RGB-Based 3D Detection and 6D Pose Estimation Great Again" Proc. of the International Conference on Computer Vision, (2017), pp. 1521-1529.

[13] Xiang, Y., et al. "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes" Proc. of Robotics: Science and Systems, June 2018.

[14] Garrido-Jurado, S., et al. "Automatic Generation and Detection of Highly Reliable Fiducal Markers Under Occlusion" Pattern Recognition 47, Elsevier Ltd., (2014), pp.2280-2292.

[15] Tanaka, H., et al. "A Portable 6-DOF Motion Tracker Using High-Accuracy AR Markers - First Report on the Feasibility" Proc. of the International Conference on Machine Vision Applications, (2015), pp.563-566.

[16] Krizhevsky A., et al. "ImageNet Classification with Deep Convolutional Neural Networks" Proc. of the 25th International Conference on Neural Information Processing Systems, (2012), pp.1097-1105.