International Journal of Innovative Computing, Information and Control
Volume **14**, Number **4**, August **2018**

ICIC International ©2018 ISSN 1349-4198
pp. **1545–1552**

POSE ESTIMATION OF DAILY CONTAINERS FOR A LIFE-SUPPORT ROBOT

Guang Yang[1], Shuoyu Wang[1], Junyou Yang[2], Bo Shen[1] and Peng Shi[3]

[1]School of Systems Engineering
Kochi University of Technology
185 Miyanokuchi, Tosayamada, Kami City, Tosayamada, Kochi 782-8502, Japan
218004i@gs.kochi-tech.ac.jp; { wang.shuoyu; shen.bo }@kochi-tech.ac.jp

[2]School of Electrical Engineering
Shenyang University of Technology
No. 111, Shenliao West Road, Shenyang 110870, P. R. China
junyouyang@sut.edu.cn

[3]School of Electrical and Electronic Engineering
The University of Adelaide
Adelaide, SA 5005, Australia
peng.shi@adelaide.edu.au

Received January 2018; revised May 2018

Abstract. *Caring for people who are elderly is now a global challenge given the shortage of nursing and healthcare providers. This makes life-support robots that could assist people to live independently highly significant. A frequent task during life support is the fetching of daily containers, which requires accurate six-dimensional pose estimation. In this paper, we develop a pipeline that is capable of providing such estimation. First, transfer learning is used to retrain an object-detection model based on a convolutional neural network to produce accurate rectangular masks. After extracting object clouds based on these masks, the iterative closest point algorithm is used to perform point-cloud registration, resulting finally in pose estimation. Several approaches are introduced to increase the registration accuracy and stability by providing adequate initial alignment and suitable model clouds considering both self-occlusions and partial occlusions. Through experiments involving actual daily scenarios, the effectiveness and accuracy of the proposed pipeline are verified.*

Keywords: Pose estimation, Life-support robot, Daily container

1. **Introduction.** Confronted with an ever-increasing and aging population and a shortage of nursing personnel, robots with life-support ability could play a vital role in assisting people who are elderly to live independently. Accompanied by intelligent robots, this approach could hopefully increase the quality of life for many people. The KUT-LSR robot pictured in Figure 1 could provide such assistance. Equipped with sensors of multiple types, this robot can perceive its environment accurately. It can also interact freely with the world by means of two manipulators and an omnidirectional movement base. Herein, we focus on the task of fetching containers, which is one of the services required regularly during daily life support.

Considerable research effort has been devoted to object fetching. One of the most effective approaches is based on a pipeline with two stages, namely (i) object detection based on a convolutional neural network (CNN) and (ii) point-cloud registration based on the iterative closest point (ICP) algorithm [1]. Our research benefits from this pipeline

DOI: 10.24507/ijicic.14.04.1545

FIGURE 1. Life-support robot KUT-LSR

while limiting the scope of objects to daily containers. Additionally, we modify both the two stages so that the pipeline could better serve the pose estimation of daily containers.

By daily containers, we mean the boxes and bottles that normally contain drink, food, or medicine. To realize the pipeline successfully, we confront two main challenges.

(1) Highly personalized containers: the daily containers used by different families vary considerably regarding brand, color, and shape. Therefore, the complexity of the model retraining process to provide accurate object detection should be considered seriously.

(2) Self- and partial occlusions: because of self- and partial occlusions, the scanned point clouds may not contain all the information about a given container. To perform highly accurate registration with the ICP algorithm, we must address problems such as initial alignment and model cloud processing properly.

In Section 2, we investigate related work in the field of object detection and pose estimation. Subsequently, we present a pose-estimation pipeline designed for daily containers in Section 3. Consequently, in Section 4, we show that the life-support robot KUT-LSR can fetch the target containers successfully with the proposed pipeline. Eventually, we conclude the work in Section 5.

2. **Related Work.**

2.1. **Object detection.**

Object detection has drawn increasing attention in recent years with the application of deep learning. The classical approaches to this challenge are based on algorithms such as shape matching [2] and histogram back projection [3]. However, these methods usually suffer from low recognition accuracy and limited tolerance for unstructured environments.

Deep learning for image classification was implemented successfully in 2012 [4] and was modified soon after to solve problems such as object detection. Delicately designed models such as MobileNets [5] and Faster R-CNN [6] allowed accurate rectangular masks (RMs) to be generated containing the target objects. Moreover, the use of deep learning has increased the performance of semantic segmentation dramatically, making highly accurate pixel-level segmentation available [7].

The choice among such approaches depends primarily on the needs of the given task. Traditional methods do not perform reliably in cluttered daily environments. Semantic segmentation requires large numbers of images labeled at pixel level as training data, but that is impractical because it is common to have to retrain the model for different users

or newly included containers when supporting the lives of people who are elderly. Therefore, object-detection approaches resulting in RMs are chosen given their low retraining complexity and steady performance in daily scenarios.

2.2. **Pose estimation.** Object detection can identify and localize daily containers in two-dimensional (2D) red-green-blue (RGB) images, pose estimation on the other hand is supposed to provide six-dimensional (6D) pose for each recognized container based on the object cloud extracted from the scanned scene cloud.

There are several widely used approaches to pose estimation. Local descriptors such as scale-invariant feature transform (SIFT) [8] have been applied successfully for objects with sufficient texture. As for the texture-less objects, three-dimensional (3D) template-matching-based methods including LINEMOD [9] prove effective, combining depth and color information. Nevertheless, this type of method normally performs less well than desired when confronted with an unstructured environment.

Moreover, ICP-based registration methods [10] can be used to align 3D point clouds, thereby producing accurate pose estimation. Because ICP is an iterative local optimizer, we must address problems such as initial alignment and point-cloud pre-processing to guarantee performance.

3. **Method.** Figure 2 illustrates the structure of our proposed pipeline. The original point cloud acquired from a Kinect 2 is regarded as the "scene cloud" containing the raw sensor information and can be registered with the RGB image via accurate camera calibration. With a fine-tuned CNN model in the object-detection phase, accurate RMs of the target containers can be obtained from the RGB images. Then, by projecting the RMs from the RGB-image frame to the point-cloud frame, the point cloud of the target container (object cloud) can be extracted from the scene cloud. Nevertheless, because the extraction is based on rectangles, there will be some background outliers.

Meanwhile, models of various containers are drawn with CAD software or scanned using a depth camera. A suitable model is retrieved from the model library based on the recognized container label. The "model cloud" results from multiple processing of container models such as down sampling and rotation. The ICP algorithm is then used to

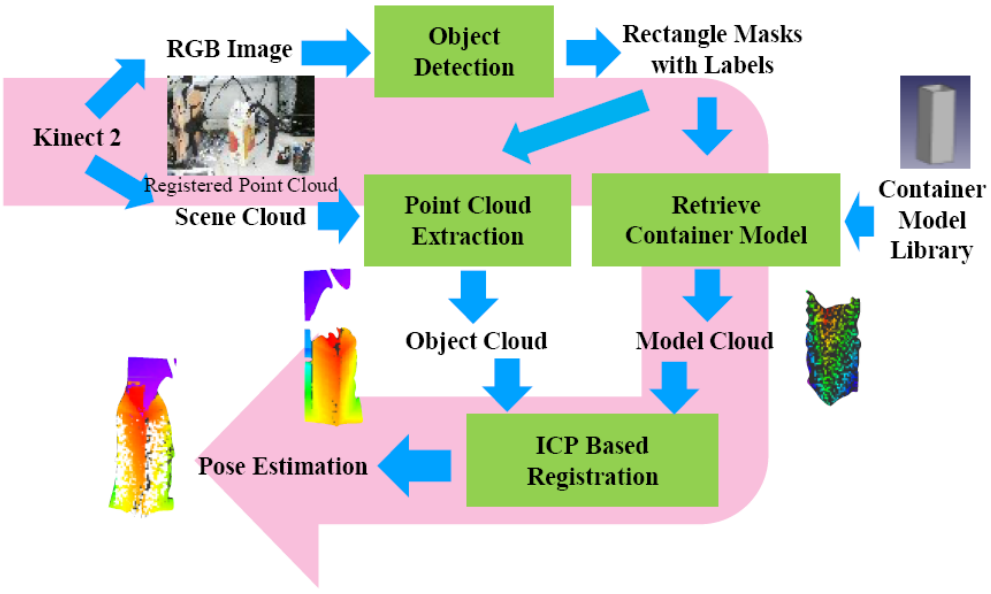

FIGURE 2. Proposed pose-estimation pipeline

register the model cloud with the object cloud. Upon successful registration, a 6D pose can finally be estimated.

3.1. **Object detection.** To select a CNN model that could keep accuracy, speed, and retraining complexity in ideal balance, we explored several models from the Tensorflow detection-model zoo. Ultimately, we chose the ssd_mobilenet_v1_coco model, which was pre-trained on the Microsoft COCO dataset [11]. Originally, the dataset consisted of only general container classes such as "bottle". By taking advantage of transfer learning, we fine-tuned the model with 200 labeled images (which were finished easily with software such as DarkLabel) for three containers and less than 10h of training on a PC (Intel i7 processor with no graphics-processing unit). Recognizing more containers in various scenarios requires more effort, but the overall process is relatively easy and simple, making it practical for daily use.

3.2. **Point cloud registration.** The ICP algorithm was designed initially to register two point clouds with the same or similar number of points and close initial poses. In our case, good initial alignment and suitable model clouds are crucial for acceptable performance.

Regarding pose initialization, without good initial alignment, the ICP algorithm can easily converge to an incorrect local minimum. We have observed experimentally that the initial orientation of a container does not influence the result dramatically because containers are usually placed vertically on platforms such as a desk.

A good initial position could be addressed based on prior knowledge. With a finely tuned CNN model, we can assume that the generated RMs surround the target container properly. For instance, in Figure 3, the container should be approximately centered in the corresponding RM marked with "Juice Box".

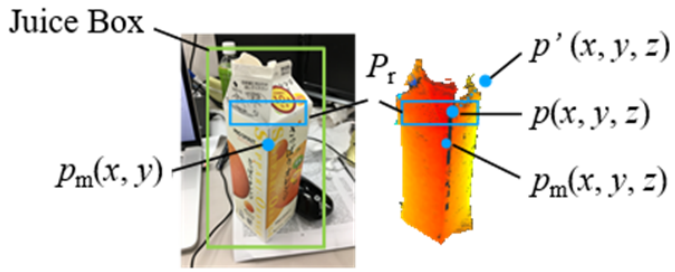

FIGURE 3. Iterative closest-point pose initialization

We then simply set the median point of the mask as $p_m(x, y)$ in 2D RGB space. Projecting the point into the point-cloud frame allows $p_m(x, y, z)$ to be obtained in 3D space. It should lie on the surface of the container, which is an initial pose close to the target cloud. Nevertheless, it often fails for two reasons: (i) the sensor data are noisy, meaning that the value of z could be either empty or inaccurate; (ii) if the median point in the RGB image is covered by other objects, $p_m(x, y, z)$ could be set close to an incorrect object.

Considering these factors, we modify the approach as follows. A rectangular area labeled with P_r is placed on the RM; P_r refers to the extracted point cloud with regard to this rectangle. The *CheckPointCloud* function in (1) iterates all the points in P_r, eliminates empty values, and reorganizes the remaining points, returning the median point as $p(x, y, z)$ in the 3D point-cloud space:

$$p(x, y, z) = CheckPointCloud(P_r) \tag{1}$$

Furthermore, we push the point back along the z axis by a distance d, which is half the z dimension of the model. Eventually, the initial-position guess $p'(x, y, z)$ can be calculated.

This newly introduced approach allows good initial poses to be calculated with higher accuracy and better tolerance to sensor noise and object occlusions.

Regarding model cloud processing, limited camera positions mean that only one side of the container can be scanned. Providing a full model of the container directly to the registration would result in poor performance because the external points could act as outliers.

Additionally, some containers such as the juice box shown in Figure 4 have movable parts that can influence the registration. Herein, we use only the most stable and reliable parts of the container when constructing the mesh model.

In [12], the self-occlusion issue was solved using a multi-hypothesis approach. Because most daily containers have relatively simple shapes (mainly cubes and cylinders), we can decrease the algorithmic complexity by having far fewer candidate crops (in [12], 30 crops were required for each object). As shown in Figure 5, because the object clouds of bottles scanned from different angles are similar, only one candidate crop is needed in that case.

The situation becomes slightly more complicated when cubic containers are involved. Two types of target cloud are possible, namely the "half" type (Figure 5-2) and the "one face" type (Figure 5-3). Two candidate model clouds that comprise only those parts labeled with solid lines are provided to the ICP registration, and for pose estimation we choose the one with the lower registration error.

Partial occlusions occur because the target container is partially hidden by other objects, making the scanned container surfaces incomplete. Herein, we consider only partial occlusions caused by other objects on the same surface with the target container.

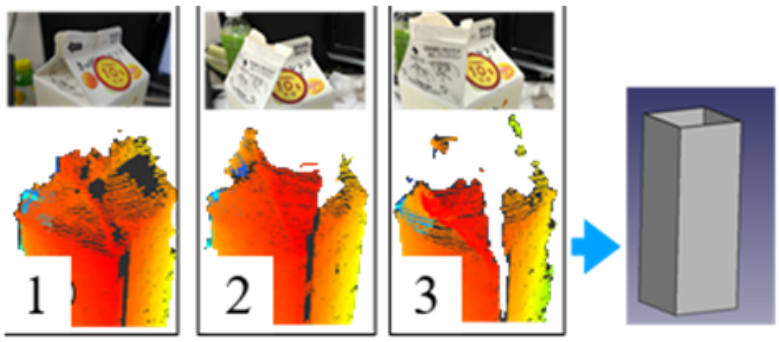

FIGURE 4. Movable parts of a container

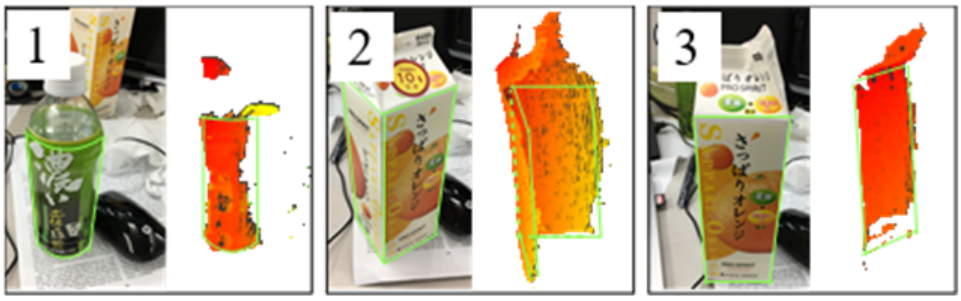

FIGURE 5. Self-occlusion examples, hand-drawn solid lines are used to highlight the container surfaces.

The multi-hypothesis method itself offers a certain degree of tolerance of partial occlusions. In Figure 6-1, the coffee can is causing an occlusion, two surfaces of the juice box are scanned, one incompletely so. In this case, the half-type model cloud would perform poor registration whereas the one-face model cloud would register the complete surface accurately. Therefore, no extra effort is required to deal with the occlusion in this situation.

However, if both scanned surfaces are incomplete because of the coffee can (Figure 6-2), the pose is estimated inaccurately. In this situation, a suitable model cloud could be provided by cropping the model upward from the bottom. In previous work, we cropped the model with a constant step length until the registration error ε decreased below a given threshold ε_t.

However, it is difficult to choose the step length because doing so involves a trade-off between speed and accuracy. Herein, we propose an approach involving a model cropped with a variable step size. During the experiment, we noticed that the ICP registration error ε could also be a metric for the degree of partial occlusion. Therefore, we calculate the model-cropping step length l as

$$l = \mu\varepsilon \tag{2}$$

where μ is an empirical value. Similar to the principle of a classical P controller, l can be calculated based on the degree of partial occlusion. Two successful registration scenarios are demonstrated in Figure 7.

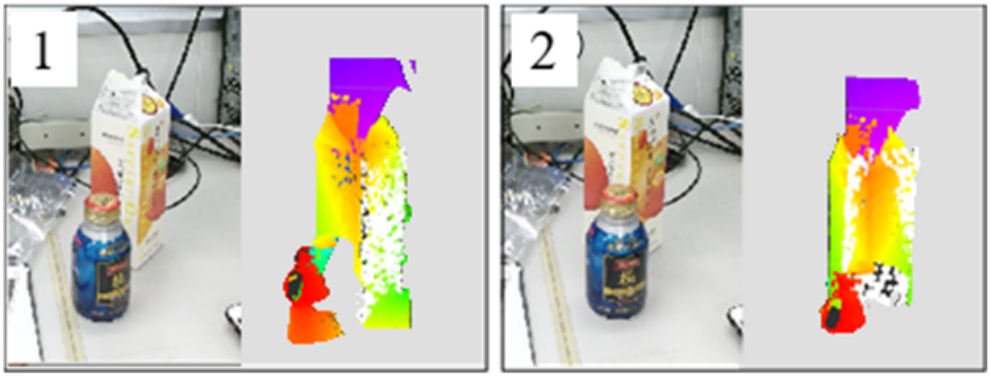

FIGURE 6. Examples of partial occlusions, the white point clouds are the model clouds used.

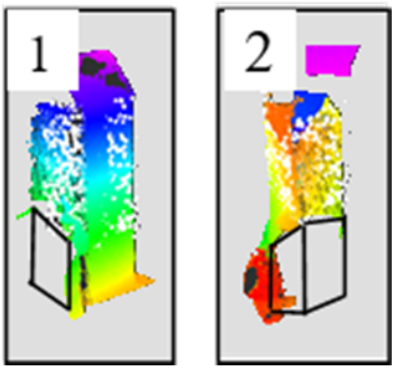

FIGURE 7. Successful registration scenarios with partial occlusions. The hand-drawn solid lines indicate the missing surface, and the white point cloud indicates the model cloud used for registration.

4. Experiments.

4.1. **Evaluation of effectiveness and accuracy.** In the first experiment, we used a Kinect 2 RGB-D camera as the data source. We placed several daily containers on a crowded desk and used a maker board to provide the ground truth of the target-container poses (Figure 8).

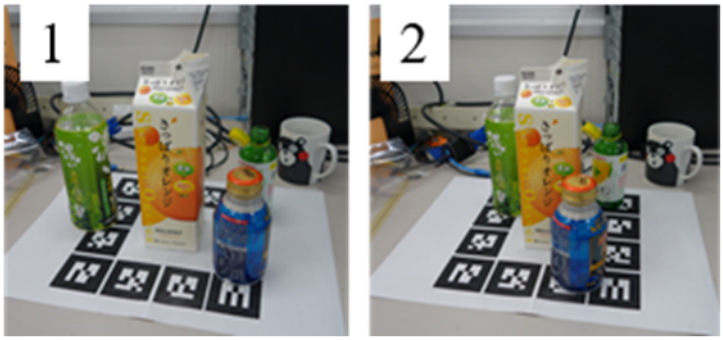

FIGURE 8. Scenarios used to evaluate accuracy

We began with pose estimation with no partial occlusion, taking a juice box as the target container (Figure 8-1). We conducted 20 estimations of various poses relative to the camera. The position error was less than 2 mm and the orientation error was less than 2°.

Next, we conducted 20 estimations of various poses with the juice box partially hidden (Figure 8-2). We note that the error bounds increased to 1 cm in position and 7° in orientation considering 18 successful registrations. The remaining two trials ended with failed information because ε did not decease below ε_t after multiple model cropping. Nevertheless, the performance fulfills the requirements of container picking.

4.2. **Container fetching with KUT-LSR.** The second experiment was to demonstrate various fetching tasks performed with a KUT-LSR robot. The scenario involved a working desk that was excluded from the model training phase to challenge the pipeline's generalization ability.

We placed several daily containers on the desk along with some tools. Figure 9-1 to Figure 9-4 show one of 10 fetching tasks conducted with the right arm, and Figure 9-5

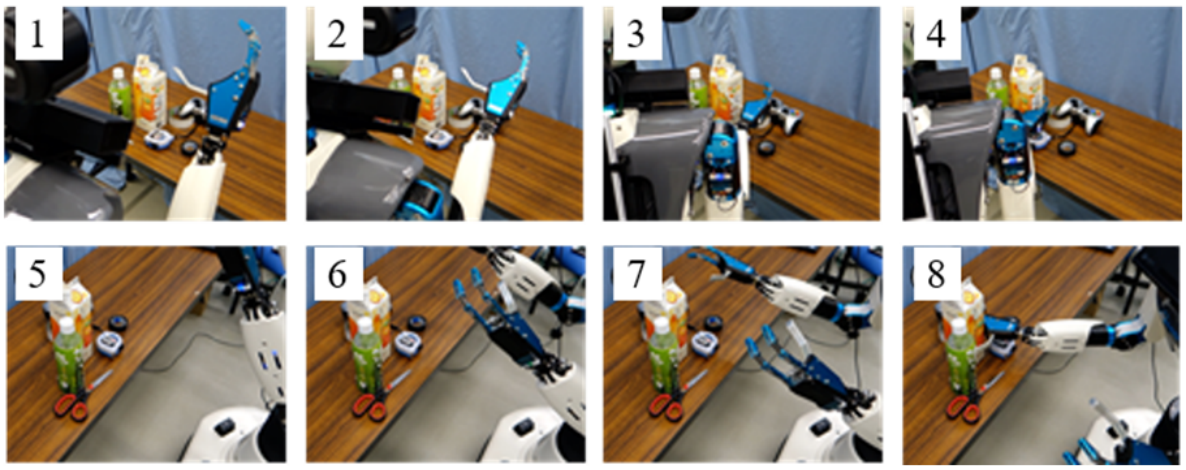

FIGURE 9. Container-fetching experiment with a KUT-LSR robot

to Figure 9-8 show the same task recorded from another angle. It could be concluded that the KUT-LSR is capable of reliable and accurate object picking in a cluttered daily environment, which is essential for the realization of various life-support tasks.

5. **Conclusion.** Assisted by the proposed pose-estimation pipeline, with small amount of re-training effort for the intended daily containers, a life-support robot can fetch the targets with promising accuracy and a certain degree of robustness to self- and partial occlusions. One main challenge remaining is the deformation of container surfaces when using RGB-D cameras such as Kinect 2. In future work, we aim to improve the stability of pose estimation given such deformations.

Acknowledgment. The research is supported by JSPS KAKENHI Grant No. 15H03951, the Canon Foundation, and the Casio Science Promotion Foundation.

REFERENCES

[1] A. Zeng, K. Yu, S. Song et al., Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge, *Proc. of IEEE Int. Conf. Robot. Autom.*, pp.1386-1393, 2017.
[2] M. Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks and R. Chellappa, Fast object localization and pose estimation in heavy clutter for robotic bin picking, *Int. J. Rob. Res.*, vol.31, no.8, pp.951-973, 2012.
[3] M. J. Swain and D. H. Ballard, Color indexing, *Int. J. Comput. Vis.*, vol.7, no.1, pp.11-32, 1991.
[4] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.*, vol.25, pp.1-9, 2012.
[5] A. G. Howard, M. Zhu, B. Chen et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv:1704.04861*, 2017.
[6] S. Ren, K. He, R. Girshick and J. Sun, 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.
[7] J. Long, E. Shelhamer and T. Darrell, Fully convolutional networks for semantic segmentation, *Proc. of IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol.7, pp.3431-3440, 2015.
[8] D. G. Lowe, Distinctive image features from scale invariant keypoints, *Int. J. Comput. Vis.*, vol.60, pp.91-110, 2004.
[9] S. Hinterstoisser, V. Lepetit, S. Ilic et al., Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes, *Lect. Notes Comput. Sci.*, pp.548-562, 2013.
[10] S. Rusinkiewicz and M. Levoy, Efficient variants of the ICP algorithm, *Proc. of the 3rd International Conference on 3-D Digital Imaging and Modeling*, pp.145-152, 2001.
[11] T. Y. Lin, M. Maire, S. Belongie et al., Microsoft COCO: Common objects in context, *Lect. Notes Comput. Sci.*, vol.869, pp.740-755, 2014.
[12] J. M. Wong, V. Kee, T. Le et al., SegICP: Integrated deep semantic segmentation and pose estimation, *IEEE Int. Conf. Intell. Robot. Syst.*, vol.2017, pp.5784-5789, 2017.