2016 IEEE-RAS 16th International Conference on
Humanoid Robots (Humanoids)
Cancun, Mexico, Nov 15-17, 2016

3D Object Segmentation for Shelf Bin Picking by Humanoid with Deep Learning and Occupancy Voxel Grid Map

Kentaro Wada[1] and Masaki Murooka[1] and Kei Okada[1] and Masayuki Inaba[1]

***Abstract*— Picking objects in a narrow space such as shelf bins is an important task for humanoid to extract target object from environment. In those situations, however, there are many occlusions between the camera and objects, and this makes it difficult to segment the target object three dimensionally because of the lack of three dimentional sensor inputs. We address this problem with accumulating segmentation result with multiple camera angles, and generating voxel model of the target object. Our approach consists of two components: first is object probability prediction for input image with convolutional networks, and second is generating voxel grid map which is designed for object segmentation. We evaluated the method with the picking task experiment for target objects in narrow shelf bins. Our method generates dense 3D object segments even with occlusions, and the real robot successfully picked target objects from the narrow space.**

I. INTRODUCTION

Robotic picking capability, which is fundamental for manipulation of objects, has progressed in recent years so that which can be applied for various objects. But it is still difficult to pick objects in a narrow space such as shelf bin picking environment as shown in Fig.1 with uncertainty of object pose, brightness, and occlusions. Even with the recent progress of machine 2D object segmentation technology with deep learning [1] [2], the lack of three dimentional sensor inputs because of occlusions is still the problem on segmenting 3D object for robotic manipulation.

Compared to the picking task in an environment where previous studies were conducted like a tabletop, picking task in a narrow space has difficulty of occlusions. This is because the objects is located inside the narrow space like bins shown in Fig.1, and the appropriate camera angles for object localization is restricted by the wall and roof parts as well as by the objects themselves. In general, it is hard task for robot to pick target object with many occlusions, because the lack of 3D information of the object makes both localization and size estimation difficult, which are crucial for generating picking motion.

In this study, we propose a method for three dimentional object segmentation in a condition with difficulty on acquiring three dimensiontal sensor inputs, point cloud. Our proposing method consists of 2D object segmentation and 3D voxel grid mapping. In addition to the standard method to convert 2D segmentation to 3D voxels, which uses the correspondence between image and point cloud (upper left in Fig.1), we accumulate each segmentation result in different viewpoints as a 3D map (lower left in Fig.1). Our method enables the robot to acquire a dense three dimentional information on segmenting the target object, and expands the scope of activity of humanoids for manipulation in narrow space.

K. Wada, M. Murooka, K. Okada, and M. Inaba are with the Department of Mechano-Informatics. The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. wada@jsk.imi.i.u-tokyo.ac.jp.

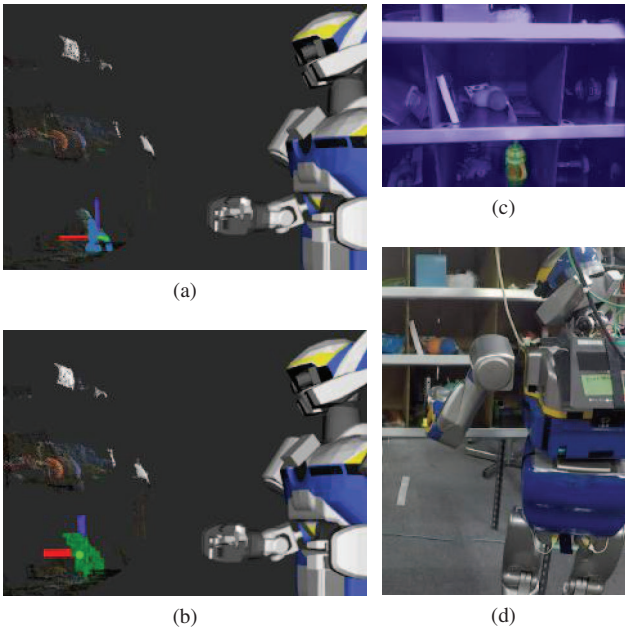


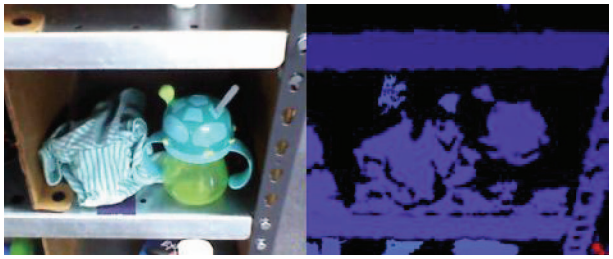
Fig. 1: Shelf bin picking by humanoid robot.
Each image is a result of segmentation with usual method (Fig.1a), that with our method (Fig.1b) robot camera view at recognizing (Fig.1c) with red region representing high probability of the target, and picking behavior by real robot (Fig.1d). The target object, green cup, is placed at deep position in the bin, and it is hard to segment it three dimentionally with usual method because of the occlusions. Our method segments the target object densely compared to the usual method refering the point cloud.

II. 3D OBJECT SEGMENTATION FOR PICKING BY HUMANOID

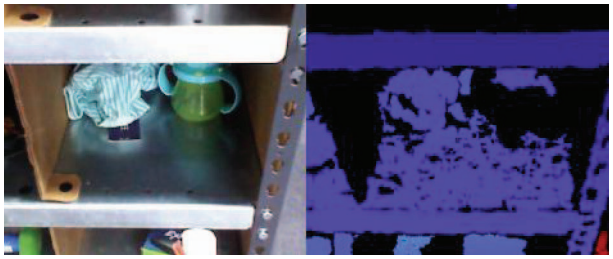
A. Related Works

Robotic picking is one of the fundamental robotic manipulation behavior, and studied in a many previous works. In previous works, however, the robot workspace is mostly collision and occlusion free, and they conduct the picking task in a situation with given location of target object [3], no target in various objects [4] [5], and in situation with single object on a tabletop [6]. Even in a previous work for picking task in a narrow space, the target object is located at the front side of the shelf bin [7], whose 3D information can be more easily sensed by RGB-D sensor: Fig.2b is the

harder situation than Fig.2a with objects located in inside back of the shelf bin. In Fig.2b, the objects are occluded by the part of the shelf, and the depth of transparent part of the cup is not sensed because of the camera angle. In this study, we tackle the difficulty of picking task execution in a condition with occlusions of target object and lack of 3D sensor inputs.



(a)



(b)

Fig. 2: 3D information of objects in a narrow space.
It shows RGB (left) and depth image (right). Fig.2a shows the situation where objects are located in front, and Fig.2b shows one with objects in back.

As the previous works on object segmentation, it is tackled using a large-scale dataset with thousands of images and dozes of object classes [1]. In contrast to this 2D segmentation work, segmenting three dimensionally is required for robot to localize object in order to conduct tasks in real world. The usual approach for this is registering the 2D segmentation result to the 3D point cloud with correspondence relationship between image and point cloud [7]. In another work [8], mapping-based segmentation is tackled using probabilistic 2D image segmentation and updating voxels with a Bayesian framework. That work was tackled in order to improve the segmentation accuracy on 2D image by accumulating the probability in multiple views. Our approach also includes this component to accumulate 2D segmentation result to acquire dense 3D information of the target object in an environment with occlusions. And the main difference in the approach between [8] and our work is the 2D segmentation method: random decision forests in previous work and deep convolutional network in our work.

B. Proposed Method

Our proposing method for object segmentation consists of 2 components: first is the object probability prediction $I_t^{proba} = f(I_t)$, which predicts pixelwise label occupancy probability for candidate objects I_t^{proba} at time t from input image I_t, and the second is 3D voxel grid map generation

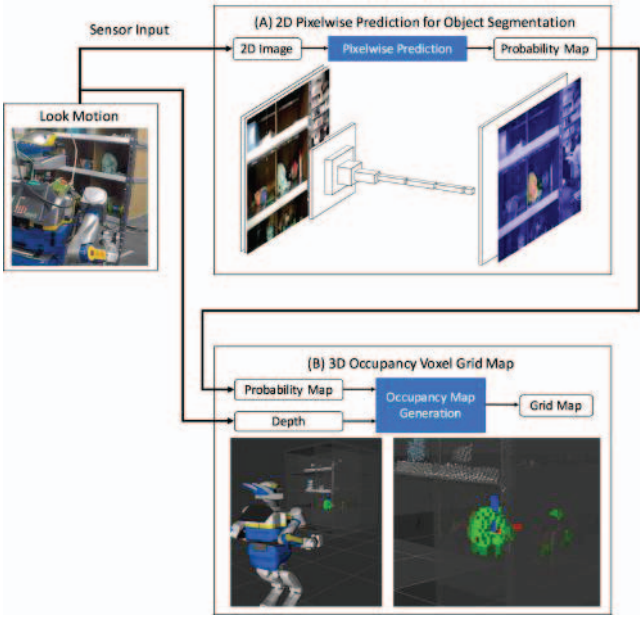


Fig. 3: Proposed system for 3D object segmentation.
The main component of this system is the pixelwise object probability prediction based on convolutional network, and the voxel grid map generation with updating the occupancy probability in each grid.

$M_t = g(M_{t-1}, I_t^{proba}, C_t)$ which updates the voxel grid map M_{t-1} with the probability map I_t^{proba} and point cloud C_t. In the second component, the probability 2D prediction result is registered to the point cloud to generate the 3D probability prediction result, and the 3D map is used to update the voxel grid map.

We construct system shown in Fig.3 by integrating our 3D segmentation method with looking around motion of humanoid. By accumulating the object probability prediction result three dimensionally as a map, we realize the target object segmentation in a narrow space.

In the following sections, we introduce the method to predict object probability from input image in Section III, and the map generation method in Section IV. In Section V, we validate the efficiency of our proposed method with a shelf bin picking task execution in the real world.

III. 2D Object Segmentation with Fully Convolutional Networks

In this section, we introduce the function $I_t^{proba} = f(I_t)$ which receives RGB image I_t as the input and outputs object probability image I_t^{proba}. We construct deep neural network for this function, and we describe the network architecture and the way of training in the following subsections.

A. Network Architecture

For object segmentation with deep learning, Fully Convolutional Networks (FCN) architecture is usually used [2]. In this network model, the convolutional operation is applied to the input image with half resizing in max pooling layers, and the inverse operation of convolution (deconvolution) is applied and the hidden layers output is resized to the same

size with the input image as the output layer. The network architecture is shown in Fig.4: the number in the figure represents the number of channels, the box size represents the image size, H represents input image height, W represents width, N represents the number of object labels. In general, object segmentation is a problem to assign object labels to each pixels, and it is requested to output label image, whose size is same as the input image and each pixel has object label values. In previous works [1] [2], convolution neural networks extract features from image with keeping the 2D construction, and it is confirmed that FCN is the powerful architecture for this kind of task, in which both input and output are image.

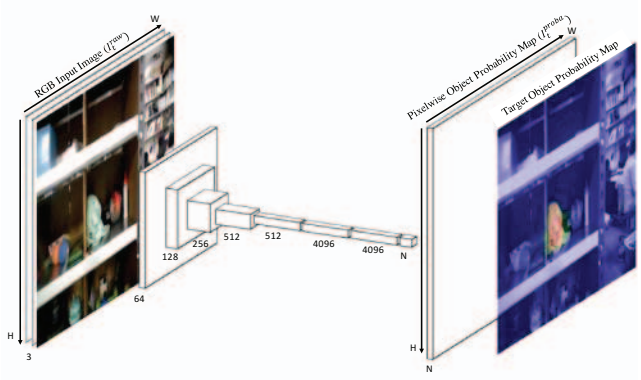


Fig. 4: Fully Convolutional Networks architecture.
Our network architecture for the 2D object segmentation with 40 object labels.

In this study, we also used the similar FCN architecture shown in Fig.4 for object segmentation as the previous work [2]. The network consists of 16 convolution layers, 5 max pooling layers, and 1 deconvolutional layer, and as the activation function Relu [9] is used. We used the softmax function to output probability map I_t^{proba} from the output of deconvolutional layer I^{deconv} for $i = 0 \cdots H$ and $j = 0 \cdots W$ as below:

$$I_{ij}^{proba} = \sigma(I_{ij}^{deconv}) \tag{1}$$

$$\sigma(I_{ij}^{deconv}) = \frac{exp(I_{ij}^{deconv})}{\sum_{k=0}^{N} exp(I_{ijk}^{deconv})} \tag{2}$$

B. Training the Network with Dataset

We handle items shown in Fig.5, so the whole object labels for segmentation is 40 labels: 39 labels for the items and 1 background label. Compared to the previous researches [1] [2], which handle 21 object classes, the number of that we are handling is larger. But in our picking senario the objects are only located inside the shelf, and the location where objects is located is limited.

Considering the restriction of the objects location, we collected dataset with images in which the objects are located inside the shelf bins as shown in Fig.6. The dataset generation process consists of the image collection and human-handed annotation, and we automated the image collection

Fig. 5: Objects and environment for the picking sernario.
We use 39 items for the experiment, and a shelf which has 12 bins. The number of object classes is larger than that of previous works, but the environment where objects exist is limited in the shelf.

by using a robot which had RGB-D camera sensor on its hand. The size of the dataset is 218 sets of sensor image and label image generated with annotation by human. We splitted the dataset in 8:2 for training and validation, and used them for network training and evaluation of segmentation accuracy.

Fig. 6: Process for generating image segmentation dataset.
The process consists of automated image collection by a robot, and human annotation.

Following the previous work we used the parameters of VGG16 network [10] to initialize parameters of convolutional layers in FCN, by fine-tuning of the object recognition network for object segmentation as its efficiency is reported in previous work [11]. The loss function is the softmax cross entropy of deconvolutional layer's output and ground truth label image, and the optimization is conducted by Adam [12] with parameters of step size: $\alpha = 1\text{e}{-5}$, and exponential decay rates for the moment estimates: $\beta_1 = 0.9$, $\beta_2 = 0.99$. We compute the 2D segmentation accuracies defined below:

- pixelwise accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean IU: $(1/n_{cl}) \sum_i n_{ii} / (t_i + \sum_j n_{ji} - n_{ii})$

where n_{ij} is the number of pixels of class i predicted as class j with n_{cl} number of classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels predicted as class i. Fig.7 shows the loss, pixelwise accuracy, and mean IU change with iterations: the left side figures show the result with training dataset and the right side ones show that with validation dataset. This figure shows that the segmentation accuracies are enhanced with both training and validation dataset, however, after the iteration 6000 the mean IU with validation dataset decreases. We used the trained network at training iteration 6000 for the actual prediction in the object segmentation task. As for the segmentation accuracy at the iteration, pixelwise accuracy is 0.906 and mean IU is 0.283 for validation dataset, and this performance is as good as that by previous work [2] on a segmentation dataset NYUDv2 [13]: pixelwise accuracy

0.600 and mean IU 0.292.

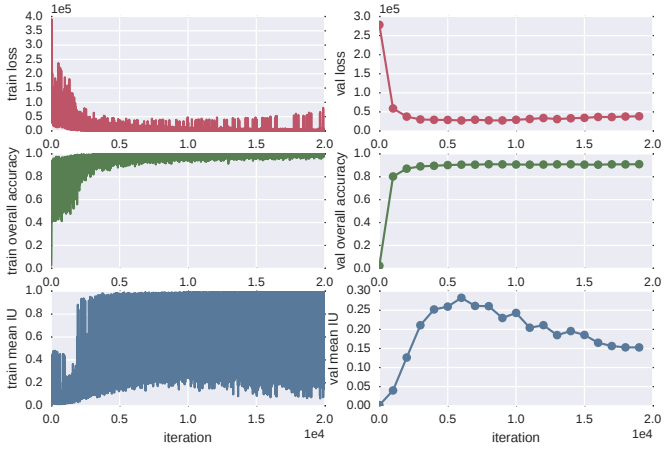


Fig. 7: Learning curve on training of the segmentation network.
Learning curve for loss, pixelwise accuracy, and mean IU for both training (left) and validation (right) dataset. The x axis in all figures represent training iterations.

IV. Map Generation for 3D Object Segmentation

In this section, we introduce the function $M_t = g(M_{t-1}, I_t^{proba}, C_t)$ to generate 3D voxel grid map for object segmentation, which consists of the registration of 2D probability map to 3D, and map updates with the map representation with label occupancy grid.

A. 3D registeration of 2D probability map

For extracting 2D object segmentation result three dimensionally, the segmented region in the image is registered to the corresponding 3D point using the correspondence relationship between image and point cloud. This is a standard approach for the conversion of 2D image segmentation to 3D, and in a occlusion-free environment enough number of 3D points of target object are extracted with this operation for the shape and centroid estimation. In environment with many occlusions, however, it is hard to estimate the 3D shape and centroid with this approach because of the lack of point cloud in some viewpoints. It causes the displacement of the estimated centroid and different estimated size of the object. Our method also includes this registration method, but we segment the target object by generating a voxel grid map by sensing with different camera angles.

Robot motion is required for the sensing of environment with different camera angles, but there is displacement between RGB image and the point cloud sensed by the camera when the camera is moving faster than a certain velocity. This is because the RGB sensor and depth sensor activate in a different cycle, and even with the synchronization of nearest two sensor input timestamps, there is displacement in the index of pixels between RGB image and point cloud. For this synchronization problem, we limited the map updates only when the camera is not moving, and we guaranteed the correspondence between image pixels and depth points with suspension of map updates until the velocity become 0 after camera movements.

B. Map Generation with Label Occupancy Grid

In this section, we introduce Label Occupancy Grid Map, which is map updating method for object segmentation, and the variation of Occupancy Grid Map [14], where the environment is splitted into cells with static grid size and the occupancy of each cell is represented probabilistically. The map is usually used for path planning for mobile robot navigation [14] and humanoid reaching behavior [15], but label occupancy grid map is map generation method for object segmentation by replacing occupancy probability in original occupancy grid map with object probability.

In standard occupancy grid map, the i-th cell m_i holds the probability of occupancy $p(m_i)$. $p(m_i)$ becomes close to 1 for the occupied cell, close to 0 for the free cell, and around 0.5 for the unknown cell. In our occupancy grid map, however, each grid probability means the occupancy by the target object. For object segmentation, the object class is represented by label value with 0 for background and 1 to N for other objects. So we call this occupancy grid map as Label Occupancy Grid Map, in which each cell holds the probability of label occupancy $p_{lbl}(m_i)$: $p(m_i) = 1$ means occupied by the label, $p(m_i) = 0$ means occupied by other label. Our segmentation model includes background label and all pixel is labeled as each cell, so $p(m_i) > 0.5$ means occupied by the label.

We explain the method of updating $p(m_i)$ from sensor information z_t; t is the index of series of sensor input which represents the period of the measurement time, z_t means the set of sensor information of t-th period, and $z_{1:t}$ means the set of the sensor information from the first period to the t-th period. Notice that z_t is set of the 3D point cloud and the result of object probability prediction in 2D image. By computing $p(m_i|z_{1:t})$ with $p(m_i|z_{1:t-1})$ and z_t, we can integrate new sensor information and update the map periodically.

We can derive the following formula from Bayes' theorem and independence of the conditional probability.

$$\begin{aligned} p(m_i|z_{1:t}) &= \frac{p(z_t|m_i, z_{1:t-1})\ p(m_i|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(m_i|z_t)\ p(z_t)\ p(m_i|z_{1:t-1})}{p(m_i)\ p(z_t|z_{1:t-1})} \end{aligned} \tag{3}$$

The ratio of occupied probability $p(m_i|z_{1:t})$ and free probability $p(\neg m_i|z_{1:t})$ becomes simple as follows.

$$\begin{aligned} \frac{p(m_i|z_{1:t})}{p(\neg m_i|z_{1:t})} = \frac{p(m_i|z_{1:t-1})}{p(\neg m_i|z_{1:t-1})} \frac{p(m_i|z_t)}{p(\neg m_i|z_t)} \frac{p(\neg m_i)}{p(m_i)} \\ \text{where} \quad p(\neg m_i) = 1 - p(m_i) \end{aligned} \tag{4}$$

We can deform the formula by introducing log-odds (logit) as follows.

$$\begin{aligned} l(m_i|z_{1:t}) = l(m_i|z_{1:t-1}) + l(m_i|z_t) - l(m_i) \\ \text{where} \quad l(x) = \text{logit}(p(x)) = \log \frac{p(x)}{1 - p(x)} \end{aligned} \tag{5}$$

Presuming that we have no prior knowledge of the environment,

$$l(m_i) = 0 \quad (\because \quad p(m_i) = 0.5 \ ) \tag{6}$$

Therefore,

$$l(m_i|z_{1:t}) = l(m_i|z_{1:t-1}) + l(m_i|z_t). \tag{7}$$

The probabilistic map is updated using this formula, and the object probability $p(m_i|z_{1:t})$ is obtained from the log-odds $l(m_i|z_{1:t})$ as follows:

$$p(m_i|z_{1:t}) = 1 - \frac{1}{1 + \exp l(m_i|z_{1:t})} \tag{8}$$

V. Experiment

A. Experiment Setup

We validated our segmentation method with picking task experiments using a life-sized humanoid robot and octomap [16] as the efficient implementation of the 3D occupancy grid map. To use original occupancy grid map for our label occupancy grid map, we added the function of map updates using the object probability 2D map to the octomap. The occupancy grid map is generated with 1[mm] resolution and the grids are limited to the region inside a bin, with assumption that the shelf position and shape is known. In the experiment, the robot localize the shelf by detecting the checkerboard, and input point cloud to the shelf is segmented before being passed to the octomap with known region clipping of the bin inside. The given information about objects in the experiment is the target object and its located bin. The set of candidate objects for object segmentation which can exist in the target bin is unknown, so the object segmentation must be conducted with 40 classes exactly.

B. Picking Task Execution based on Our Approach

We evaluated the method of object segmentation by generating the label occupancy map with picking target objects from shelf bins. Fig.8 shows the sequential images which represents the state of real robot (left side of each image) and visualization fo recognition (right side). In the visualization, upper right image shows the RGB image input from camera which exists in the head of the robot, lower right image shows the pixelwise probability of target object, and left region shows the point cloud and generated 3D map.

The task is conducted as follows: firstly robot looks around the target bin to generate label occupancy grid map, secondly picks the object based on the generated map, and finally places the object at the ordered place. The camera poses with the looking around motion by robot are decided by the target bin and given shelf geometry and its pose with a checkerboard, setting fixed offset from shelf to the standing point and fixed 3 different camera angles in 4 different heights of standing. If the voxels for target object are generated with this recognition stage, the picking motion of humanoid robot to the object is generated based on depth value of the centroid of computed from the generated 3D voxels afterwards. There can be some optimization in looking around motion, but the result with this motion shows the ability of generating object voxels even with few good views for the target object by using the few good views.

We conducted the picking task for 3 objects (t-shirts, cup and dumbbell) from 3 bins. Fig.8 shows the picking sequence for t-shirts, and Fig.9 shows the segmentation accuracy and camera velocity while the looking around motion. The segmentation accuracy is measured with IU between the 3D box annotated by human and generated voxels. Here, IU is defined as $V_{tp}/V_{tp} + V_{fp} + V_{fn}$: V_{tp} is overlap volume between the box and voxels (true positive), V_{fp} is difference between V_{tp} and voxels volume (false positive), V_{fn} is one between V_{tp} and box volume (false negative). Fig.9 shows the rise of accuracy (IU) with different camera views. From the transition of camera velocity it is supposed that IU after first view is around 0.05 at time $1[sec]$, and after 4 views (at time $12[sec]$) the IU is around 0.11 and it increases twofold. These result shows our proposed method is effective to segment object densely using multiple views. With the task sequence, the robot successfully picked the 3 objects from shel bin and placed them into the ordered place.

VI. Conclusions

We presented an approach to segment object three dimensionally in a narrow space, acquiring dense 3D information with deep learning and voxel grid map. The contributions of this paper are the following:

1) we proposed a novel approach to segment object three dimentionally with integration of deep learning and occupancy voxel grid map.
2) we evaluated the effectiveness our approach with shelf picking task by life-sized humanoid.

By introducing map generation method which uses object probability, we extended the applicability of object segmentation in environment with lack of point cloud of objects because of occlusions. We verified the effectiveness of our method with experiments on picking task execution for target objects in narrow bins. With our method, the object 3D model is generated densely in an environment with occlusions, and picking task is completed successfully using the centroid estimation. For future work, it is considered that our method is usable for object exploration by updating the label occupancy grid map until the object is detected.

References

[1] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, Vol. 111, No. 1, pp. 98–136, January 2015.

[2] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015.

[3] Johan Tegin, Staffan Ekvall, Danica Kragic, Jan Wikander, and Boyko Iliev. Demonstration-based learning and control for automatic grasping. *Intelligent Service Robotics*, Vol. 2, No. 1, pp. 23–30, 2008.

[4] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *In ISER*, 2016.

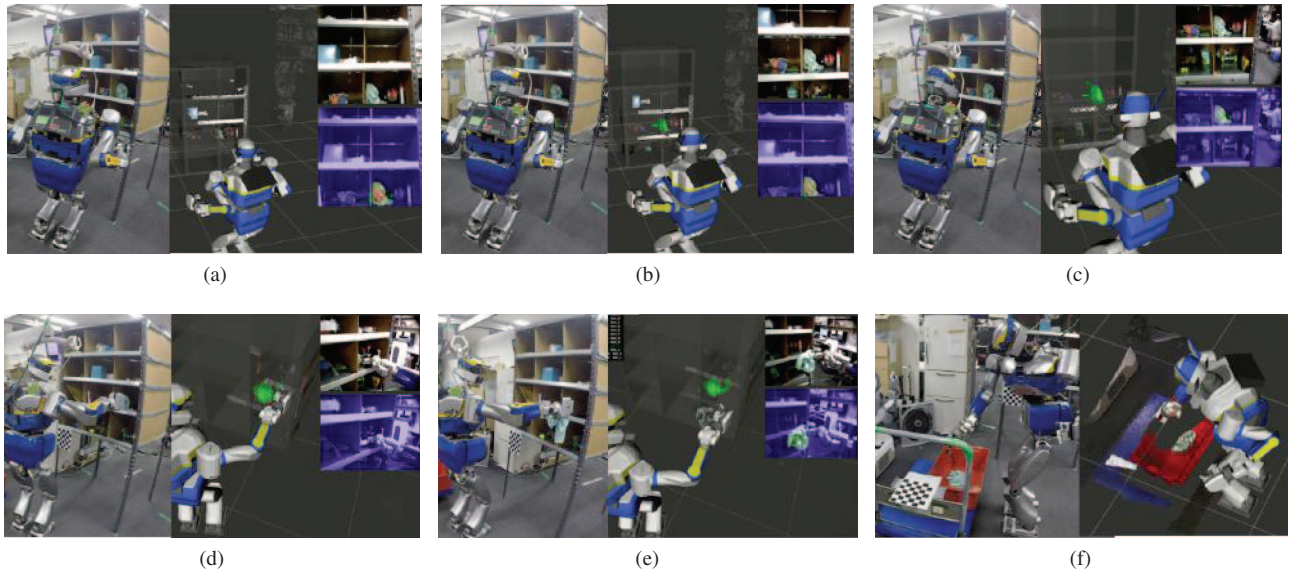


Fig. 8: Task sequence of robot picking t-shirts from a bin and place to the ordered bin.
The left side of each image represent the state of real robot, and the right side shows the result of visualization. In the experiment, the robot segment target object with looking around motion, and reach the hand to the centroid of generated 3D object map.

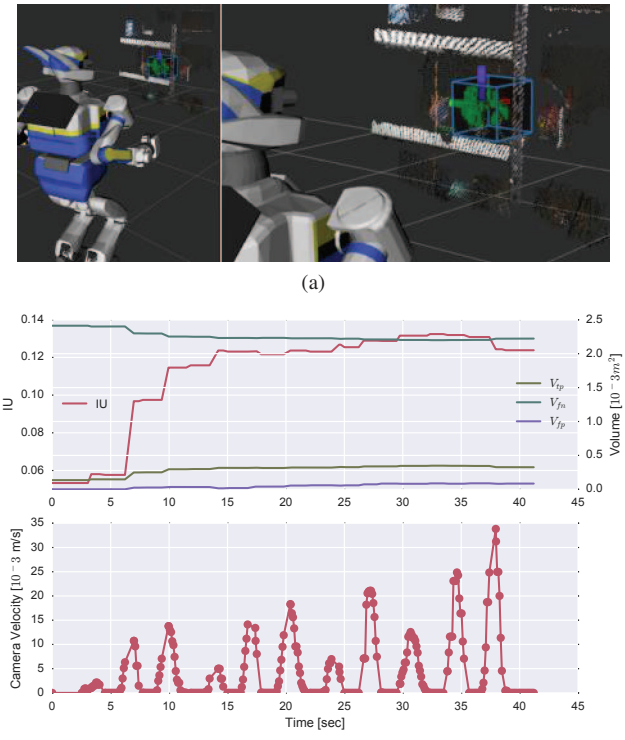


Fig. 9: Transition graph of segmentation accuracy and camera velocity while looking around to pick t-shirts.
Horizontal axis indicates the time as an elapsed time in second, and vertical axis does IU, volumes, and camera velocity. The accuracy is computed as the IU between voxels and 3D box annotated by human as shown in Fig.9a.

[5] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413, May 2016.

[6] Kei Okada, et al. Multi-cue 3D Object Recognition in Knowledge-based Vision-guided Humanoid Robot System. In *IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA*, 2007.

[7] C. Eppner, et al. Lessons from the Amazon Picking Challenge: Four Aspects of Building Robotic Systems. In *Proceedings of Robotics: Science and Systems*, 2016.

[8] Jorg Stuckler, Nenad Biresev, and Sven Behnke. Semantic mapping using object-class segmentation of rgb-d images. In *IROS*, 2012.

[9] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Frnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814. Omnipress, 2010.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[11] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pp. 3320–3328. Curran Associates, Inc., 2014.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *In ICLR*, 2015.

[13] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. *Indoor Segmentation and Support Inference from RGBD Images*, pp. 746–760. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, Vol. 22, No. 6, pp. 46–57, June 1989.

[15] M. Murooka, R. Ueda, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Planning and execution of groping behavior for contact sensor based manipulation in an unknown environment. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3955–3962, May 2016.

[16] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, Vol. 34, No. 3, pp. 189–206, 2013.