The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

Learning-based Selective Dual-arm Grasping for Warehouse Picking

Shingo Kitagawa Kentaro Wada Kei Okada Masayuki Inaba

the University of Tokyo, Graduate School of Information Science and Technology

We propose a learning-based system of selective dual-arm grasping and use Convolutional Neural Networks (CNN) for grasping point prediction and semantic segmentation. First, the network learns grasping points with the automatic annotation. and the grasping points are automatically calculated based on the shape of an object and annotated for both single-arm and dual-arm grasping. The robot then samples various grasping points with both grasping ways and learns optimal grasping points and grasping way. As a result of multi-stage learning, the robot learns to select and execute optimal grasping way depending on the object status. In the experiments with the real robot, we demonstrated that our system worked well in warehouse picking task.

1. Introduction

Recently, learning-based approaches have become popular in robot grasping, and self-supervised method and simulation-based one are both commont. However, the main difficulty of the learning-based approaches is the small variety of the grasping motion. These approaches deal only with one arm, but a robot can grasp more various objects with two arms, and many studies show the advantage of dual-arm manipulation [Harada 12, Edsinger 07]. Recent dual-arm robots such as Baxter[*1] are expected to manipulate various objects with two arms. However, learning dual-arm grasping is difficult because a robot needs to sample with both grasping ways for self-supervised approach [Pinto 16, Levine 16], and simulation-based one [Mahler 17, Viereck 17] needs to reproduce the grasping condition of dual-arm grasping, which is more complex than single-arm one. In this paper, we tackle the difficulties of learning to grasp with two arms and propose to combine supervised learning with the automatic annotation and self-supervised learning.

This paper focuses on how to learn to selectively grasp with two arms as shown in Fig.1, and we propose a learning-based system of selective dual-arm grasping. In the system, we use Convolutional Neural Networks (CNN) for grasping point and semantic segmentation. The network first learns grasping points with automatic grasping point annotation, and the automatic annotation is based on the geometric constraints of objects. The robot then samples various grasping points, tries to grasp by both grasping way and learns where to grasp and which grasping way is optimal. In detail, the robot executes both grasping way with various grasping points and collect its grasping data with the trained network, and we train the network again with the sampled data for the real world adaptation. Finally, the robot selects and executes optimal grasping from single-arm and dual-arm ones with the adapted network. The whole system of the multi-stage learning is described in Fig.2, and this learning-based system adapts the network to various objects and requires fewer grasping trials with the real robot. In the experiments, we demonstrated our system worked well in warehouse picking task.

Contact: Shingo Kitagawa, Graduate School of Information Science and Technology, University of Tokyo, 113-8656, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan, 03-5841-7416, s-kitagawa@jsk.imi.i.u-tokyo.ac.jp

∗1 http://www.rethinkrobotics.com/baxter/

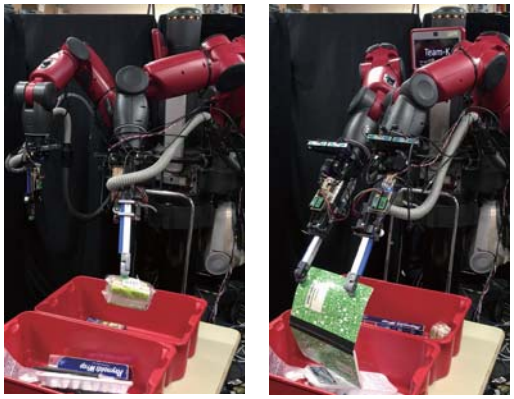


Figure 1: The robot learns how to grasp and whether single-arm or dual-arm grasping is optimal to execute corresponding to the environment. Images are from [Kitagawa 18]

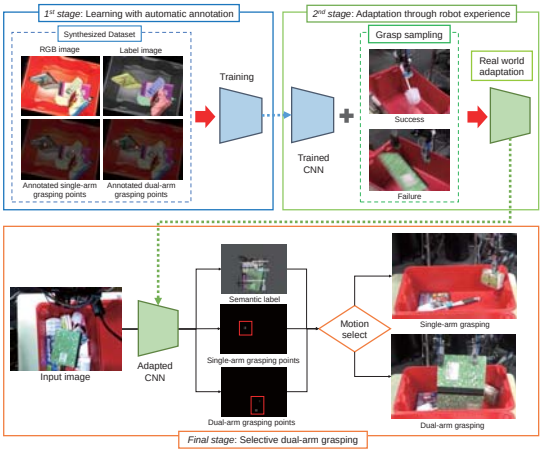


Figure 2: We propose a multi-stage learning system of dual-arm selective grasping. The CNN first learns how to grasp with automatic grasping point annotation, and the robot samples grasping with the trained network and learns with the collected data. Finally, the robot selectively executes dual-arm grasping with the adapted network.

2. Related Work

Learning-based Grasping. Recent approaches use CNN to predict the success probabilities of grasping pose [Pinto 16]

[Mahler 17], and semantic segmentation network is also used for the prediction of grasping pose [Kusano 17]. In this paper, we focus on the grasping ability of a vacuum gripper and predict grasping point with CNN. In order to predict grasping point (x, y) on RGB image, our approach segments whole region into two region; graspable and ungraspable and predicts the success probabilities of for both single-arm and dual-arm grasping for each pixel.

Self-Supervised Learning of Grasping. One main stream of learning-based grasping is self-supervised approach, and the previous study shows its effectiveness with high success rate of grasping [Pinto 16]. This study uses Mixture of Gaussians (MOG) background subtraction algorithm as a primitive algorithm, but it is not efficient enough for sampling, so that it requires more than 300 trials for each object on average, which is time consuming and high workload for robot. In this paper, we give a primitive grasping algorithm beforehand, and a robot samples grasping efficiently and learns grasping in short time.

3. Learning with Automatic Annotation

3.1 Automatic Grasping Point Annotation

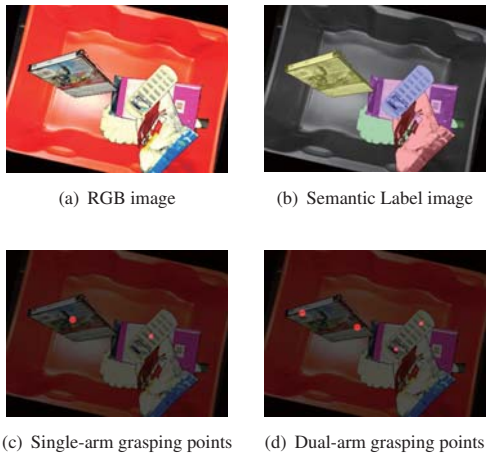
For the automatic annotation, we design a primitive grasping algorithm. We prepare several RGB images for each object and calculate the grasping points for both single-arm and dual-arm grasping from geometric conditions of the object. First, we do background subtraction on RGB image to get the object region, and the single-arm grasping point then is calculated as a center point of the region as shown in Fig.3(a). For dual-arm grasping points, we assume that the points are lined in the first principal component of the object region, so that we do PCA on x and y axes of the object region and annotate them as shown in Fig.3(a). Therefore, the background subtraction and geometric analysis such as PCA on the object region is implemented as the primitive algorithm, and the generated images are used for the dataset synthesis.

(a) Single-arm grasping point (b) Dual-arm grasping points

Figure 3: The object region is inside the green boundary, and the annotated grasping points are drawn as red crosses. We first do the background subtraction on an original RGB image to get the object region. A single-arm grasping point (a) and dual-arm grasping points (b) are calculated from the primitive grasping algorithm from the region.

3.2 Grasping Dataset Synthesis*2

Using the generated data with the automatic annotation, we synthesize a dataset for semantic and grasping point segmentation. For the dataset synthesis, we do image stacking [Dwibedi 17] and paste subtracted object images randomly on a background image in the real experimental environment. With stacking several RGB images of object images on one background image, the synthesized image reproduces a clutter such as warehouse picking environment, where objects occlude with each other. The pixelwise semantic labels are simultaneously annotated as the RGB image synthesis, and the grasping points are annotated as pixelwise graspable region. The synthesized images of pixelwise semantic labels and graspable label for both single-arm and dual-arm grasping are shown in Fig.4(b), Fig.4(c), and Fig.4(d). If 10% of an object region is hidden by other objects because of overlaps, we regard the overlapped object as ungraspable because of its physical occlusion and do not annotate its grasping points. Therefore, an object 90% of whose region is visible is treated as placed on the top of a clutter and graspable. With the synthesized dataset, we train a CNN to predict and segment the semantic and graspable regions.



(a) RGB image (b) Semantic Label image

(c) Single-arm grasping points (d) Dual-arm grasping points

Figure 4: We stack the subtracted RGB object images and pixelwise semantic labels are simultaneously annotated as (a). The grasping points for both single-arm (b) and dual-arm grasping (c) are also annotated as pixelwise graspable labels. Images are from [Kitagawa 18].

3.3 Pixelwise Graspable and Semantic Segmentation

3.3.1 Network Design

We propose an FCN-based network to predict pixelwise semantic and graspable label simultaneously. The whole structure is based on FCN32s [Long 15] and described in Fig.5. The network has two parts of convolution layers: the former has the same structure as convolution layers of VGG16 [Simonyan 14] and extracts features from RGB image, and the latter splits into each task and separately predicts pixelwise semantic labels and graspable labels of single-arm and dual-arm grasping. Therefore, the network does three segmentation task and outputs three probability images from one RGB image. Semantic segmentation is formulated as same as FCN [Long 15], and it predicts the pixelwise probabilities of semantic labels including the background. For grasping point prediction, we treat it as pixelwise graspable segmentation, and the network predicts the pixelwise probabilities of graspable labels. For the input and the output of the network, we resize RGB image and corresponding label images in 640x480.

3.3.2 Network Training

The loss function of semantic segmentation L_{seg} is calculated with softmax cross entropy SCE for each pixel For graspable segmentation, we define the same loss function for both single-arm and dual-arm grasping. As the graspable regions are much smaller than ungraspable regions, we set a weight w^c_{grasp} for label c based

*2 **The subsections are from [Kitagawa 18]**

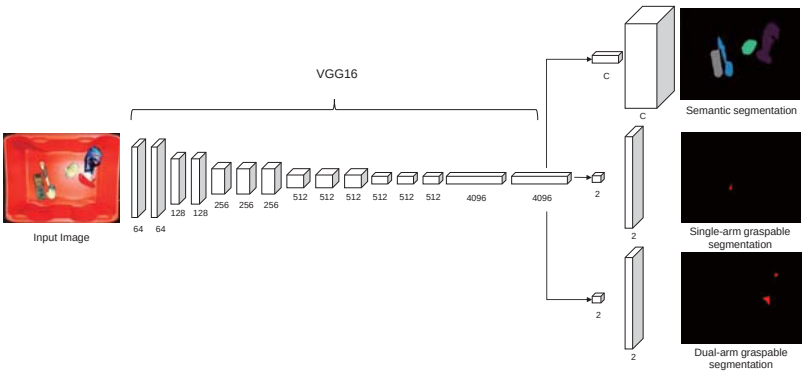


Figure 5: Our network predicts the pixelwise probabilities of semantic and graspable labels simultaneously. C is the number of semantic labels including the background.

on frequency balancing [Badrinarayanan 15], which increases loss value of smaller regions:

$$w^c_{grasp} = \begin{cases} \frac{N_{foreground}}{\alpha_c N_c} & (N_c \neq 0) \\ 0 & (N_c = 0) \end{cases}$$

where $grasp \in \{single, dual\}$ is a grasping way, $N_{foreground}$ is the number of non-background pixels in ground truth image of semantic segmentation, N_c is the number of labels c pixels in ground truth image of grasping segmentation and α_c is a constant parameter for class c. With w^c_{grasp}, the loss of graspable segmentation L_{grasp} for both single-arm and dual-arm grasping is calculated with softmax cross entropy SCE as follows:

$$L_{grasp} = -\sum_{c}^{C_{grasp}} \sum_{x,y}^{W,H} w^c_{grasp} SCE(l^c_{grasp}(x,y), h^c_g(x,y)) \quad (1)$$

where C_{grasp} is the number of grasp labels, l^c_{grasp} is one hot vector of class c at a pixel (x, y) and $h^c_{grasp}(x, y)$ is network output of class c at a pixel (x, y). Since graspable segmentation divides the whole region into two labels: graspable and ungraspable, the number of grasp labels C_{grasp} is fixed as 2. For the whole network, the total loss L_{total} is calculated as the summation of all losses, L_{seg}, L_{single} and L_{dual}, and back-propagated through the network except the deconvolution layer.

For the optimization, learning rate is initially set as $1.0\mathrm{e}{-5}$. We train the network with $\alpha_{graspable} = 20.0$ and $\alpha_{ungraspable} = 1.0$, and the training is done in 200000 iteration with around 20000 synthesized data pairs. In order to make the segmentation robust, we do data augmentation, and random flip, rotation and RGB modification are added in every iteration.

4. Adaptation through the Robot Experience

4.1 Grasp Sampling with Trained Network

For the real world adaptation of the trained network, a robot executes grasping trials in the real world with the network and train and adapt the network with the collected data. In the sampling, we put an object in front of the robot one by one and try to grasp it with both single-arm and dual-arm grasping. The robot samples grasping points by weighted random sampling using graspable segmentation output of the trained network, and we use the pixelwise graspable probabilities inside the object region as the weight of the sampling, The robot records the grasping result with two labels success and failure by air pressure sensors, and the failure consists of grasping failure and self-collision failure. The grasping failures may happen when a robot does not sample appropriate grasping points and occurs in both single-arm and dual-arm grasping cases. On the other hand, the self-collision failure only happens in the case of dual-arm grasping, and two arms of the robot are too close or crossing in this case.

4.2 Adaptation with the Sampled Data

After sampling and collecting grasping data in the real world, we adapt the network by the data. First, we generate an object mask image with semantic segmentation output of the network. For graspable segmentation, we annotate successful grasping points with graspable label and ungraspable label on the rest of the object region. With the annotated data, we synthesize a dataset with the same algorithm in Subsection 3.2, and we do the adaptation of the network with the synthesized dataset.

The learning rate is initially set as $1.0\mathrm{e}{-6}$, and the training is done in 12000 iterations, but all the other parameters for training is same as the training with the automatic annotation. We also do the same data augmentation in this phase.

5. Selective Dual-arm Grasping

5.1 Extracting Object Graspable Region

With the adapted network, a robot selects to execute proper grasping motion for successful grasp, and we design selective dual-arm grasping system shown in Fig.6. In order to evaluate the two grasping motion, we first do pixelwise multiplication of the probabilities of semantic label and graspable label image. With the result of the multiplication, we select object label and grasping way from single-arm or dual-arm grasping with $Argmax$ function for every pixel. Therefore, the robot determines which object to grasp and which grasping way should be executed with $Argmax$ function. After the motion select, we get the selected object region from $softmax$ output of the semantic probabilities. The graspable region is also extracted from the selected graspable probabilities P_{grasp} with the threshold $max(P_{grasp}) - 0.05$. The final graspable region of the selected object is generated by merging the two regions with $PixelwiseAnd$ function.

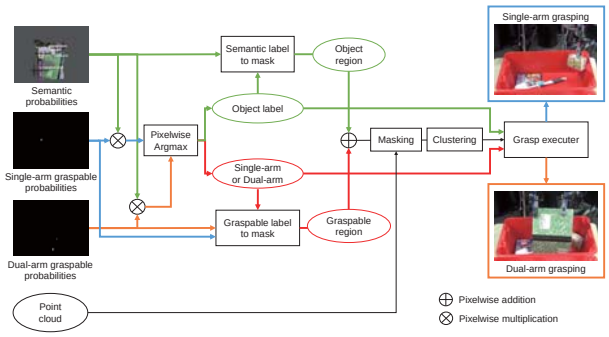


Figure 6: We propose a selective dual-arm grasping system using semantic and graspable segmentation. The robot selects the object and the grasping way from the network outputs and determines grasping points using the region and point clouds.

5.2 Selecting Grasping Points with Point Clouds

From the graspable region, the robot tries to grasp the center of the region. In order to grasp the center of the region, we first extract point clouds in the region by masking. After the masking, we do Euclidean clustering on the extracted point clouds and calculate the each center of each clusters. We use the center of biggest cluster as the grasping point of single-arm grasping, and for dual-arm grasping, we use the centers of two biggest clusters as grasping points. In the end, the robot executes grasping with the calculated grasping points and selected grasping way.

6. Experiments

6.1 Experimental Configuration*2

We chose 9 objects from the target objects in Amazon Robotics Challenge 2017 [Morrison 17], and all the experiments in this paper are conducted in [Kitagawa 18]. For the multi-stage learning, we did the self-supervised adaptation stage twice after learning with the automatic annotation. The robot collected 94 data pairs in the first sampling time and 121 pairs in the second time. Therefore, the network was trained with a dataset synthesized from 94 sampled data and original object data in the first adaptation stage, and it was again trained only with 215 sampled data in the second adaptation stage.

6.2 Warehouse Picking Experiments*2

For the final experiments, we applied our method to warehouse picking task. We put all the 9 objects in one tote, and the robot grasped one object and moved it to the other tote. The objects in tote were cluttered and overlapped with each other, and the robot is required to recognize both semantic and graspable region correctly. We did the picking experiments twice in different settings, and the robot successfully grasped and moved 6 objects in the first configuration and 8 objects in the second one. In the experiments, the robot mostly grasps objects with one arm (Fig.7(a)), but also executed dual-arm grasping once in both trials (Fig.7(b)).

7. Conclusion

In this paper, we propose a multi-stage learning method of dual-arm grasping for warehouse picking and implement it to the real robot. For efficient grasp learning, we propose a multi-stage learning method with the automatic annotation and grasping trials in the real world, and our method achieved high grasping success rate with few trail times. In the experiment, dual-arm grasping was only executed for 10 times because it often failed in the sampling. In order to make the robot choose dual-arm grasping more, we need to normalize the distribution of the sampled data.

(a) (b)

Figure 7: The robot successfully grasped Aluminum foil with one arm (a) and Pink table cloth with two arms (b). Images are from [Kitagawa 18].

References

[Badrinarayanan 15] Badrinarayanan, V., Kendall, A., and Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *CoRR*, Vol. abs/1511.00561, (2015)

[Dwibedi 17] Dwibedi, D., Misra, I., and Hebert, M.: Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection, *CoRR*, Vol. abs/1708.01642, (2017)

[Edsinger 07] Edsinger, A. and Kemp, C. C.: Two arms are better than one: A behavior based control system for assistive bimanual manipulation, in *Recent progress in robotics: Viable robotic service to human*, pp. 345–355, Springer (2007)

[Harada 12] Harada, K., Foissotte, T., Tsuji, T., Nagata, K., Yamanobe, N., Nakamura, A., and Kawai, Y.: Pick and place planning for dual-arm manipulators, in *International Conference on Robotics and Automation (ICRA)*, pp. 2281–2286IEEE (2012)

[Kitagawa 18] Kitagawa, S., Wada, K., Okada, K., and Inaba, M.: Multi-stage Learning of Selective Dual-arm Grasping Based on Obtaining and Pruning Grasping Points Through the Robot Experience in the Real World, in *International Conference on Intelligent Robots and Systems (IROS)*IEEE/RSJ (submitted in 2018)

[Kusano 17] Kusano, H., Kume, A., Matsumoto, E., and Tan, J.: FCN-based 6D Robotic Grasping for Arbitrary Placed Objects, in *International Conference on Robotics and Automation (ICRA): Warehouse Picking Automation Workshop 2017: Solutions, Experience, Learnings and Outlook of the Amazon Picking Challenge*IEEE (2017)

[Levine 16] Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D.: Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection, *CoRR*, Vol. abs/1603.02199, (2016)

[Long 15] Long, J., Shelhamer, E., and Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440 (2015)

[Mahler 17] Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., and Goldberg, K.: Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, *Robotics: Science and Systems (RSS)* (2017)

[Morrison 17] Morrison, D., Tow, A. W., McTaggart, M., Smith, R., Kelly-Boxall, N., Wade-McCue, S., Erskine, J., Grinover, R., Gurman, A., Hunn, T., Lee, D., Milan, A., Pham, T., Rallos, G., Razjigaev, A., Rowntree, T., Vijay, K., Zhuang, Z., Lehnert, C. F., Reid, I. D., Corke, P., and Leitner, J.: Cartman: The low-cost Cartesian Manipulator that won the Amazon Robotics Challenge, *CoRR*, Vol. abs/1709.06283, (2017)

[Pinto 16] Pinto, L. and Gupta, A.: Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours, in *International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413IEEE (2016)

[Simonyan 14] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *CoRR*, Vol. abs/1409.1556, (2014)

[Viereck 17] Viereck, U., Pas, ten A., Saenko, K., and Jr., R. P.: Learning a visuomotor controller for real world robotic grasping using easily simulated depth images, *CoRR*, Vol. abs/1706.04652, (2017)