FCN-Based 6D Robotic Grasping for Arbitrary Placed Objects

Hitoshi Kusano¹, Ayaka Kume², Eiichi Matsumoto² and Jethro Tan²

Abstract—We propose a supervised end-to-end learning method that leverages on our grasp configuration network to predict 6 dimensional grasp configurations. Furthermore, we demonstrate a novel way of data collection using a generic teaching tool to obtain high-dimensional annotations for objects in 3D space. We have demonstrated more than 10,000 grasps for 7 types of objects and through our experiments, we show that our method is able to grasp these objects and propose a larger variety of configurations than other state-of-the-art methods.

I. INTRODUCTION

The goal in robotic grasping is to derive a series of configurations of a robot and its end-effector when given data acquired from some sensors in such a way that its end-effector ‘grasps’ a user-defined target. Being able to grasp arbitrary placed objects would not only enable automation of tasks found in factories and logistics, but also progress the development towards a general purpose service robot. Traditional approaches to grasp an object often rely on object pose estimation [1], [2] or pattern recognition techniques to detect object-specific features in point cloud data [3], [4].

However, such approaches are ill-suited for use in objects placed in unstructured environments, or for use with complex end-effectors with high degrees-of-freedom (DOF), because they provide information about the state of the object, but not the state in which the robot or its end-effector should be in order to grasp the object. Therefore, these approaches make use of heuristics to select a configuration (i.e. joint angles, position, orientation, etc.) for the robot and its end-effector. Furthermore, detection of object-specific features may fail or provide false positives on either deformable objects or objects unknown to the system.

The recent advances and popularity of deep learning have given birth to new methods for grasping that are able to directly generate the configurations of the robot as well as its end-effector. The idea of these new methods is to evaluate the success rate in the configuration space of the end-effector and select the configuration with the highest score.

A. Problem Statement

These new methods, however, have a disadvantage when applied on higher DOF end-effectors or when there is need for increasingly complex grasp strategies as it becomes more and more difficult or even impossible to annotate the data. Therefore, such methods rely on low dimensional annotations making them unable to output 6D grasp configurations. In this work, we therefore pose the following question: How to enable end-to-end learning to grasp arbitrary placed objects using high dimensional annotations?

B. Contribution

While current state-of-the-art CNN-based grasping methods rely on annotation on images, to the best of our knowledge there has been no work that utilizes annotations in 3D space for grasping. Therefore, our contributions are as follows:

Contribution 1: A supervised learning method using an end-to-end convolutional neural networks (CNN) model that we call the grasp configuration network (GCN), capable of predicting 6D grasp configurations for a three-finger gripper as well as probability of the configurations to successfully grasp a target object.

Contribution 2: A novel data collection strategy to obtain 6D grasp configurations to use in the training of GCN using a generic teach tool.

The remainder of this paper is organized as follows. Related work is described in Section II, while Section III outlines the GCN. Experimental results using GCN and evaluation are presented in Section IV. Future work is discussed in Section V, while the paper is concluded in Section VI.

II. RELATED WORK

A. Traditional Methods

Methods not using machine learning include CAD model matching [5] or template matching. As these methods require careful feature engineering and parameter tuning, data-driven methods have been actively studied to overcome these disadvantages. For surveys on data-driven approaches for grasp synthesis, we refer the reader to Bohg et al. [6] and to Li et al. [7] for a method based on object pose estimation.

B. Machine Learning Methods

In the past few years, CNN have become the state-of-the-art for performing general object recognition tasks [8]. In our method, we leverage a fully convolutional network (FCN), which is a type of CNN meant for semantic segmentation [9], [10]. FCN has shown to be compatible with RGB-D input images, which can be exploited for to predict pickability [9]. Nguyen et al. [11] uses a CNN model to predict affordances for an object. Unlike our method, however, their CNN model does not predict configurations for the grasp itself. In [12], Lenz et al. deploy a deep neural network model with two stages to propose and score grasp configurations. Similarly, Araki et al. [13] proposed the use of a single-stage model which was able to simultaneously predict the grasp
probability and the configuration of the end-effector. Guo et al. [14] make use of a model inspired by Faster R-CNN [15] and were able to output 4 DOF grasp candidates at multiple locations in the image with a single forward computation of the neural network.

Pinto et al. [16] and Levine et al. [17] have proposed self-supervising methods by automating the data collection process. An alternative approach to collect data for grasping is from learning by demonstration. Gupta et al. [18] have trained a multi-fingered hand to manipulate by demonstrating the desired manipulation motions for objects with their own hands. The robot then tries to imitate these demonstrations using reinforcement learning.

III. THE GRASP CONFIGURATION NETWORK

A. Data Collection

We propose a method for manually annotated data collection. As teach tool, we created a gripper with three augmented reality (AR) markers, see Figure 1. This teach tool allows us to demonstrate grasps by capturing a 400×400 image from our RGB-D camera, after which the RGB and depth channels are registered and resized to 200×200 for training and inference of GCN. Although the shape of the gripper used in our system is not represented by our teach tool, the teach tool allows capturing of 6-dimensional configurations, containing the position (XYZ), as well as orientation (RPY) in the camera coordinate system, which can in turn be used as grasp configurations. When multiple configurations in one location are demonstrated for an object, all of them are utilized to train GCN.

Because we want to record the actual position of the demonstrated grasp instead of the AR markers, the offset between the AR markers and the center between the pinch tips are added to determine the true position of the demonstrated grasp. Additionally, to increase the accuracy of the observed rotation, two orthogonal vectors of the gripper (in the longitudinal and lateral direction) are used to calculate the rotation.

B. Model Overview

To predict end-effector configurations for grasping an object, we introduce an extension for fully convolutional networks (FCN) that we call the grasp configuration network (GCN), as shown in Figure 2. As input, we provide an RGB-D image obtained from an Intel Realsense SR300 camera to our model, which in turn outputs two maps with scores: (1) a predicted location map (PLM) for graspability per pixel, and (2) a predicted configuration map (PCM) providing end-effector configurations per pixel.

C. Data Discretization

Despite the multimodal nature of grasping, prior works have tried to use regression to model grasp configurations [14]. However, since the optimal value in a regression model is calculated by minimizing a mean of distances between outputted and training data, an assumption is needed that a configuration between multiple grasp configurations must also be valid.

If we let \( g_1 \) and \( g_2 \) be valid grasp configurations for a target object \( o \) in location \((x, y)\), we argue that the mean of \( g_1 \) and \( g_2 \) might neither be an optimal nor valid grasp. We therefore discretize grasp configurations and use a classification model instead, as proposed in [21]. A naive solution to achieve discretization is to take all possible combination of quantized configurations along each degree of freedom. However, the amount of combinations would increase exponentially as the dimension of a configuration increases. To alleviate this problem, we categorize valid grasp configurations to 300 classes using k-means clustering.

D. Training

1) Pre-training: Similar to FCN, first initialize the weights in the first layer corresponding to RGB of the convolution network using VGG-16 pre-trained on ILSVRC dataset [8] followed by fine-tuning on the NYUDv2 pixel-classification task [22]. We then use these weights as initial weights of GCN.

2) Settings: To train GCN, we employ Adam as optimizer with learning rate \( \alpha = 0.0004 \) and batch size of 80. To further reduce overfitting during training, we perform data augmentation by applying label-preserving transformations.

Because of the imbalance ratio between positive and negative data (less than 1 : 10000), we define \( L_{\text{PLM}} \), the loss function for PLM to magnify the gradient for valid grasp.

\[
L_{\text{PLM}} = \alpha \cdot L_{\text{PLM}}(\text{PC}) + (1-\alpha) \cdot L_{\text{PLM}}(\text{NC})
\]

where \( \alpha \) denotes the magnification ratio, \( n \) the number of training data, \( W, H \) the width and the height of the image, \( t \) the target PLM, and \( y \) the output. We use a value of 200 for magnification ratio \( \alpha \).

For PCM, we calculate the loss for only valid locations. We define \( L_{\text{PCM}} \) as follows.
Fig. 2. Network architecture of the grasp configuration network (GCN). The stride of each layer is denoted with S1 (stride 1) or S2 (stride 2). S1 layers use a kernel size of $3 \times 3$, and $4 \times 4$ for S2 layers. All layers have their padding set to 1. After each layer except for the last layer we apply batch-normalization [19] and use rectified linear units [20] as activation functions. For the last layer, we apply sigmoid function. For training, GCN receives a $64 \times 64$ RGB-D image, and a $200 \times 200$ RGB-D image as input for inference. Outputs are a predicted location map (PLM) and a predicted configuration map (PCM).

(a) Known objects

(b) Unknown objects

Fig. 3. Objects used in our experiments (from): (A) laundry detergent, (B) plush bear, (C) energy drink, (D) white board eraser, (E) vacuum hose piece, (F) sprinkler, (G) wool, (H) cap, (I) glucose bottle, (J) duct tape, (K) bottle of tea, and (L) sippy cup. Items (A)–(G) were used in the training process, while (H)–(L) were left as ‘unknown’ objects for experiments.

\[
L_{PCM} = -\frac{1}{nC} \sum_{k=1}^{n} \sum_{c=1}^{C} \sum_{i=1}^{W} \sum_{j=1}^{H} (t^{i,j,k} - u^{i,j,k}) \log y^{i,j,k} + (1 - y^{i,j,k}) \log (1 - u^{i,j,k}),
\]

where $C$ denotes the number of classes, $S_k$ the total number of pixels at which $t^{i,j,k} = 1$, $u$ the target PCM, and $y$ the output. We use $L = L_{PLM} + \lambda L_{PCM}$ as full objective and we set $\lambda = 200$.

IV. EXPERIMENT RESULTS

A. Setup

To evaluate the performance of GCN on grasping, we have conducted experiments on 12 objects with different rigidity and shapes, see Figure 3. Experiments were done in an office environment using the same setup shown in Figure 1. In this environment, the camera provides a view on a $70 \times 50$ cm plane in which objects can be placed by us and grasped by the robot. The camera itself is mounted 75 cm above this plane.

Data collection was performed for only objects (A)–(G). For each of these seven objects, we created 12 cases for arbitrary placement with at least 100 demonstrated grasps per case, totaling to 11320 demonstrated grasps before data augmentation, and 35865 after data augmentation. Data collection per object took about 90 minutes. The remaining objects ((H)–(L)) were used to evaluate the performance of GCN for unknown objects. Training of GCN was performed
TABLE I
RESULTS FOR OBJECT
GRASPING EXPERIMENT

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>70</td>
</tr>
<tr>
<td>(B)</td>
<td>50</td>
</tr>
<tr>
<td>(C)</td>
<td>60</td>
</tr>
<tr>
<td>(D)</td>
<td>40</td>
</tr>
<tr>
<td>(E)</td>
<td>20</td>
</tr>
<tr>
<td>(F)</td>
<td>40</td>
</tr>
<tr>
<td>(G)</td>
<td>60</td>
</tr>
</tbody>
</table>

1 Each object has been trialed 10 times.

TABLE II
RESULTS FOR UNKNOWN
OBJECT GRASPING
EXPERIMENT

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H)</td>
<td>60</td>
</tr>
<tr>
<td>(I)</td>
<td>20</td>
</tr>
<tr>
<td>(J)</td>
<td>20</td>
</tr>
<tr>
<td>(K)</td>
<td>40</td>
</tr>
<tr>
<td>(L)</td>
<td>30</td>
</tr>
</tbody>
</table>

1 Each object has been trialed 10 times.

VI. Conclusion

In this paper, we proposed a fully convolutional network model, called the grasp configuration network (GCN) that is able to output 6D grasp configurations when given an RGB-D image as input. Furthermore, we introduced a data collection method for use with GCN that leverages on demonstrations of the grasps rather than on-image annotations to enable training on high-dimensional annotated data. Through the use of GCN, a classification-based model, we demonstrated robotic grasps on arbitrary placed objects with different rigidity and shapes.

REFERENCES