ME336 Project Milestone

S.Y. Wong, Z.W. Su, P.J. Lin, R.H. Zhao, Z.Y. Liu

I. Introduction

An automated operating system used in logistics sorting will directly impact on the cost, efficiency and quality of the entire logistics center. However, because of the variable shapes of packages, manual sorting is still the usual method in logistics center. Based on the hardness of the automation system, we decide to design a practical robot for sorting goods. Our project aims to reimplement and try to improve a paper named “Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach”, in order to finish the grasping task. In this paper, a real-time, object-independent grasp synthesis method that can be used for closed-loop grasping is proposed. The proposed Generative Grasping Convolutional Neural Network (GG-CNN) predicts the quality and pose of grasps at every pixel, and it improves the computation time while ensuring performance compared with current state-of-the-art techniques. Here, we would like to implement the simulation with pybullet, which is an easy-to-use Python module based on the Bullet Physics SDK.

II. Problem Statement

In recent years, with the rapid development of e-commerce, online shopping has been very popular in China. Logistics speed has become one of the most important factors in letting consumers have a better online shopping experience. In e-commerce logistics warehouses, sorting means taking goods from the shelves or turnover boxes to the order boxes corresponding to the goods. There are thousands of types of goods in a single e-commerce logistics warehouse, so adaptability to various types of goods is a major challenge. (Liang et al., 2020) So far in e-commerce logistics warehouses, sorting is conducted by humans. But for the huge amount of goods especially on some shopping festivals like Double Eleven Day, the burden of the employees will be too great to handle in one day. As a result, we want to implement a closed-loop logistics sorting system to address this challenge. We will create a dataset from the Cornell Grasping Dataset to train the network. The Cornell Grasping Dataset contains 885 RGB-D images of real objects, with 5110 human-labeled positive and 2909 negative grasps. Though the dataset is relatively small, it best suits the pixel-wise grasp representation as multiple labeled grasps are available per image. (Morrison et al., 2018) Despite the condition that the logistics packages are normally in regular shapes, e.g. cuboid and cylinder, there are sometimes alien shapes, so we use more complex objects to test our system. We propose two sets of reproducible benchmark objects to test the grasp rate. The adversarial set is created from the released datasets for Dex-Net 2.0, and the household set is chosen from the standard robotic grasping datasets APB and YCB. For the evaluation, we will produce a similar experiment to the paper we implement. The test is divided into two parts, static grasping, and dynamic grasping. In each part, the two test sets of objects will be grasped separately, and the success rate and execution time will be recorded to evaluate the performance of our approach. We hope that our approach will improve the accuracy and speed of grasping.

III. Literature Review

We intend to replicate the research work presented in Peter Corke’s 2019 publication entitled ”Learning robust, real-time, reactive robotic grasping.” This includes the investigation of grasping unknown objects, closed-loop grasping, and benchmarking for robotic grasping.

A. Grasping unknown objects

Grasp synthesis pertains to the development of a dependable robotic grip suitable for a designated object, an extensively explored area that has yielded numerous techniques. Broadly, these can be classified into analytic methods and empirical methods (Bohg et al., 2014; Sahbani et al., 2012). Analytic methods, which take into account mathematical and physical models of geometry, kinematics, and dynamics, enable the calculation of stable grasps but suffer from inadequate transferability to real-world scenarios due to the arduous task of modeling physical interactions that occur between a manipulator and an object. In contrast, empirical methods focus on using models and experience-based approaches.

A considerable number of previous techniques feature a shared pipeline: the process of categorizing grasp options that have been extracted from either an image or point cloud, then evaluating them singularly through CNNs. Upon determination of the optimal grasp candidate, the robot will execute the grasp in an open-loop manner (without receiving any feedback), which necessitates both accurate calibration between the camera and robot systems, as well as precise control of the robot and a static environment free of motion.

Execution time is the primary reason that grasps are executed open-loop. We replicate the work of Peter Corke, address the issues of execution time and grasp sam- pling by directly generating grasp poses for every pixel simultaneously, using a comparatively small neural network.

B. Closed-loop grasping

Visual servoing, which involves controlling a robot in a closed-loop manner to reach a desired pose using visual feedback, is a well-known technique (Hutchinson et al., 1996). The benefits of visual servoing and other closed-loop strategies are their ability to adjust to dynamic surroundings and their reduced dependence on fully precise camera calibration or position control.

We reproduce the experimental setup from Peter Corke (2019) et al. And the primary advantage of their real-time grasp generation approach is the ability to use it in a reactive, closed-loop fashion. The experiment aims to evaluate the ability of the system to conduct dynamic grasping by grasping objects that are in motion while attempting to be grasped. To demonstrate the superiority of the closed-loop grasping method over open-loop grasping, they conduct grasping operations in the presence of simulated kinematic errors in our robot's control.

C. *Benchmarking for robotic grasping*

Comparing the outcomes of robotic grasping experiments is challenging due to the diverse range of grasp detection methods employed, the absence of standardization between object sets, and the limitations arising from the use of different physical hardware such as robot arms, grippers, or cameras.

We emulate the research of Peter Corke (2019), propose a set of 20 reproducible items for testing, which include 8 3D-printed adversarial objects from Mahler et al. (2017) and 12 items sourced from the APB and YCB object sets. We believe that this combination provides a sufficient breadth of sizes, shapes, and complexities to enable effective comparison with other similar research while ensuring compatibility with commonly used robots, grippers, and cameras.

IV. Technical Approach

The methods used in this project involve using a generative grasping convolutional neural network (GG-CNN) to predict a pixel-wise grasp quality, which can be deployed in closed-loop grasping scenarios. We intend to propose two sets of reproducible benchmark objects for testing the grasp success rate of our approach, and demonstrate that our approach achieves a high grasp success rate on a set of previously unseen objects with adversarial geometry and household items.

Our generative grasping convolutional neural network (GG-CNN) produces a dense prediction of antipodal grasp poses and corresponding quality measures for each pixel in an input depth image. This process is accomplished in real-time, with sufficient speed to facilitate closed-loop control of grasping in dynamic environments.

GG-CNN affords two distinct advantages over other state-of-the-art grasp synthesis convolutional neural networks (CNNs). Firstly, we do not rely on grasp candidate sampling to generate grasp poses; instead, grasp poses are directly produced on a pixelwise basis. This approach is analogous to recent advances in object detection, where fully convolutional networks are commonly used for semantic segmentation at the pixel-level, as opposed to conventional sliding windows or bounding boxes (Long et al., 2015). Secondly, GG-CNN comprises orders of magnitude fewer parameters than other grasp synthesis networks, fast enough for closed-loop grasping.

V. Preliminary Results

A. *Robotic arm and scene simulation*

It involves Pybullet as the simulation software, based on bullet, encapsulated as a module of Python used for simulation and learning. The robot arm and scene object are modelled by the URDF files, so that it makes it easily to use the embedded function provided by pybullet to get the joints information and Configure collision parameters.

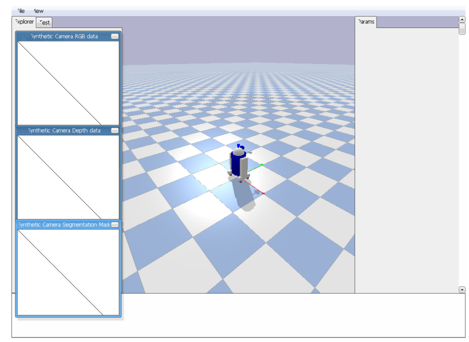

Fig. 1. Pybullet Simulation Environment

B. *Grasping method*

So far, the construction and testing of the pybullet environment has been completed, and the panda robot arm has been used to achieve a simple grasp. The grasping strategy adopts the way of visual servo, and calculates the actual coordinates of the object according to the pixel information of the image and implements the grasping. At present, we have realized a simple demo, which calculates the joint parameters of the manipulator arm through the way of position control, and realizes tracking the object in one plate and picking it up in another plate. At present, the joint control simulation, camera parameter calculation and coordinate transformation of the manipulator have been realized.

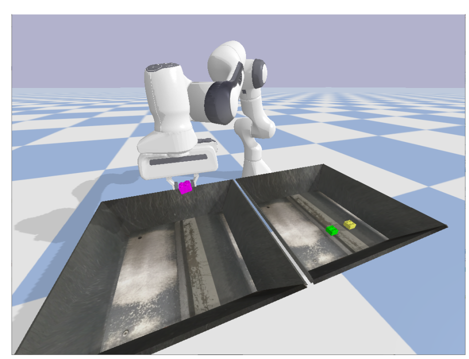

Fig. 2. Grasping Simulation