Coupling between Perception and Manipulation: Learning to Grasp Objects in Highly Cluttered Environments

1st Sarandis Doulgeris
Rijksuniversiteit Groningen
S4928628
Serres, Greece
s.doulgeris@student.rug.nl

2nd Pedro Rodriguez de Ledesma Jimenez
Rijksuniversiteit Groningen
S4745779)
Madrid, Spain
p.rodriguez.de.ledesma.jimenez@student.rug.nl

Abstract—**Object grasping and manipulation knowledge is essential for service robots which goal is to assist humans with their daily tasks. In order for a robot to interact with the environment and the user a perception system (a system that helps the robot translate the environment) should provide useful information that the robot processes and then relies on them to plan its actions. A bridge between perception and manipulation of an object enables the robot. Consider a clear table task, where a robot needs to remove all objects from a table and put them into a basket. Such tasks consist of two phases: the first phase is dedicated to the perception of the object, and the second phase is about the planning and execution of the manipulation task. In this work, we mainly focus on deep visual object grasping and manipulation.**

I. Introduction

In unstructured settings, such as human-centric environments, object grasping is a challenging task due to the high demand for real-time and accurate responses for a vast number of objects with a wide variety of shapes and sizes under various clutter and occlusion conditions. As increasing research is being done to make the robots more intelligent, there exists a demand for a generalized technique to infer fast and robust grasps for any kind of object that the robot encounters. Traditional object grasping approaches explicitly model how to grasp different objects by considering prior knowledge about object shape, and pose. This a laborious task and also time-consuming. More recent approaches deal with the hand-engineering of features by formulating object grasping as an object-agnostic problem. Real time learning of the object features without prior object-specific information helps to generalize new grasping policies of unknown objects. In this vein, much attention has been given to object grasping approaches based on Convolutional Neural Network (CNN).

In this work, we are evaluating two different approaches by running experiments of grasping and manipulating an unknown object (robot agnostical approach). In these experiments we are using two different deep learning networks (Gr-ConvNet and GG-CNN) for the purpose of evaluating the grasp and the manipulation executing in a virtual environment. Gr-Convnet get a 4-D RGB-D image as an input while GG-CNN recieves a 2.5-D depth image. Both network generates as output three images from which we can obtain grasp rectangles.

The advantages of GG-CNN over other state-of-the-art grasp synthesis CNNs are twofold

The are two main advantages by using these networks over other state-of-the-art grasp synthesis CNNs. First, they generate grasp poses directly for a single or multiple objects, instead of comparing each grasp candidate with a list of possible grasps, and the grasp with the highest probability is selected.Secondly, both of these networks have fewer parameters than other networks, in particular the GG-CNN allowing our grasp detection pipeline to execute in about $19-20[ms]$.

The main goal of this lab assignment is to provide a link between perception and manipulation using eye-to-hand camera coordination [1]. We evaluate the performance of our system in different scenarios by performing grasping trials, with static and cluttered objects on a specific dataset.

II. Generative Residual Convolutional Neural Network

Generative Residual Convolutional Neural Network (GR-ConvNet) [2], is a deep learning neural network which generates three images. From those we can extract a grasp probability or classify possible grasp points to get the best out of them.

It is separated into two main modules:

- Inference module: Takes as input the n-channel image. After suitable pre-processing, the image is fed to the GR-ConvNet where quality, angle and width images are generated. These are used to decide the best grasp pose.
- Control module: Makes the optimal plan to execute the robot's task using the generated grasp pose from Inference module.

This system uses an eye-to-hand visual feedback to gain information about the surroundings. This means that the system cannot act immediately when a grasping pose is picked. The control system has to transfer the grasping problem for the camera position to the robot's effector (antipodal hand).

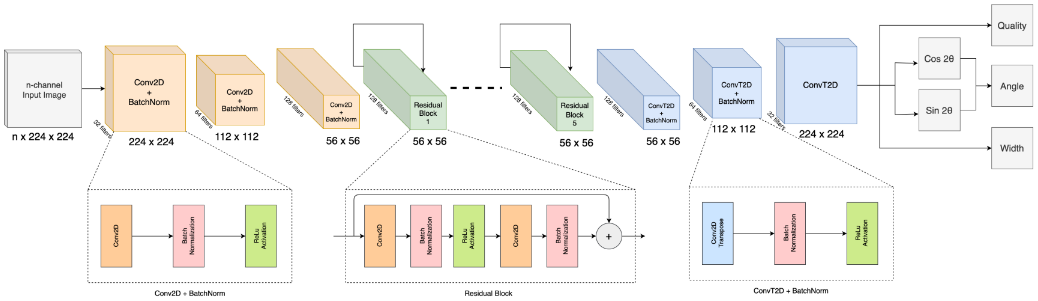

Fig. 1. GR-ConvNet layer structure

A. Inference module

RGB-D multi-modal data input images are cropped, resized, and normalized to final reach 224×244 size. This image is led to the GR-ConvNet. It has to pass through all the convolutional networks and residuals layers as pictured in Fig1, necessary for feature extraction. Throughout the process, the input image changes in dimensions either increasing in channels or decreasing in width and height. This is due to the application of different filters which provide diversity feature-wise and the down-sampling of the image. With the network depth increasing, accuracy gets saturated. This is not caused by *overfitting* but from the extra addition of layers which leads to a higher training error [3] That is why the network has 5 residual blocks, which help in the learning process of the identity functions. This is a technique which is similar to the structure of neuron in the cerebral cortex in a human brain [4].

By the end the final image has $1/4$ of its input size (width, height). In order to retain the the spatial features for easier interpretation, we need to do the opposite operation called convolution transpose. At the output layer four output images are generated that different pixel-wise information. The first one contains quality score of the grasp, the second and third one have each partial information about the required angle of the grasp in the form of $\cos 2\theta$ and $\sin 2\theta$, because the grasp is uniform around $\pm\frac{\pi}{2}$. A trigonometric combination of these extract the θ angle. The last image has information about the robot's effector width for each pixel grasp.

Different loss functions were tested for(in or for) the network. The training of the network was done by minimizing the negative log-likelihood of the mapping function. By finding the minimum of the negative log-likehood of the detected grasp given an input image, the model basically learns where the entropy has minimized which translated to the minimum disorder of the system.

B. Control module

This module mainly focuses and performing a certain task such as pick-and-place calibration. After receiving the grasp pose with the highest quality score, a trajectory is planned using inverse kinematics through a ROS interface. This trajectory is then used by the robot to execute it's task. As we mentioned the system has a hand-to-eye servoing system. That means that the image has information in camera coordinate system. In order to perform the task, the control module has to convert these camera coordinates to the robot coordinates. This conversion goes as:

- First the highest quality grasp is detected.
- This grasp is transformed from image space into camera's 3D space using the built-in parameters of the camera.
- Finally a transformation is done to transfer the coordinates into the robot space now using the camera pose calibration value.

III. Generative Grasp Convolutional Neural Network

Another approach to predict possible object grasps and qualify them as optimal or not is by using the Generative Grasp Synthesis approach. This approach uses a fully convolutional neural network that that being trained with the L2 loss function.

A. Architecture of the network

The Generative Grasp Convolutional Neural Network [5] is construct with 6 different convolutional layers, 3 standard and 3 transpose for image reconstruction (Fig2). The Generative Grasp Convolution Neural Network (GG-CNN) receives as input a 300×300 pixel depth image which passes through all the layers. The output layer has the same (philosophy) format as (with) the previous network where heatmap images of three pixel-wise information about the grasp (are) extracted (Grasp Quality image, an Grasp Angle image and a Grasp Width image).

The main features of this CNN are the following:

- Fewer computational time required to predict the qualify and pose of grasps in every pixel.
- Acts on unknown object and scenery by generating grasp quality pose in each pixel of the image.
- Close-loop and open-loop grasp execution. While many networks can not have close-loop approach's due to the high computational time, the GG-CNN can due to his lower number of parameters. The experiments shown in this work are ran using the open-loop approach.

The network has been trained with the Cornell Grasping Dataset that contains 885 RGB-D images of real objects, with 5110 human-labelled grasp as positive and 2909 labelled as negative graps.

B. Closed-loop grasping system

This type of grasping system gives the robot constant feedback allowing it to adapt to dynamic environments where object are moving. This is referred to as visual servoing. An depth image is generated in a rate of $30[Hz]$ which are used to track movement of objects. This low rate indicates that the object move at a low speed. Because multiple similarly-ranked good quality grasps in an image, we should avoid rapidly switching between them, which would confuse the controller. Three grasps from the highest local maxima are computed, and the one which is closest (in image coordinates) to the grasp used on the previous iteration is selected.

C. Opened-loop grasping system

In order to perform the open-loop grasp, the experiment is configured with a virtual RGB-D camera situated over the robot workspace in the virtual environment. From this, only the depth pixel-image is kept, which is the solely input of the GG-CNN network. There is no refresh rate of the scenery, and the grasp is categorized from this static position of the camera. Once the grasp is selected, the robot starts the process by moving to a pre-grasp position with the gripper lined up with the grasp chosen, moves straight down until the grasp pose is met or a collision is detected. No feedback of the process is given to the robot controller.

One of the disadvantages of using open-loop grasping methods is that the approach is more tend to noise errors while close-loop counteracts with the continuous feedback to the robot controller. The grasp success rate of the robot using open-loop grasping methods can drop about 40%.

IV. Experiments

A virtual system composed of a camera and a robot is needed for testing both deep neural network architectures, not only for their prediction of the best grasp for an object, but also for the execution of it by the robot based on that selected grasp. Such environment is developed and provided to us to simulate each scenario.

The environment is implemented in *PyBullet* (Fig.2) where a Universal Robot with a two-fingered gripper perceives the environment through a virtual RGB-D camera that is placed above the central point of the robot's operating area.

Different objects are placed within the area defined with the green rectangle. The simulation are performed in three separate scenarios: isolated, picked and packed.

Results such as the grasp and manipulation success rate are obtained for each scenario. Furthermore, different presentation of the captured image by the camera are provided (RGB, Depth and Segmentation Mask). The RGB-D image from the virtual camera is provided to the network and as result a grasp map is given. The system define as a successfully experiment those which the robot has picked the object and placed inside the bucket.

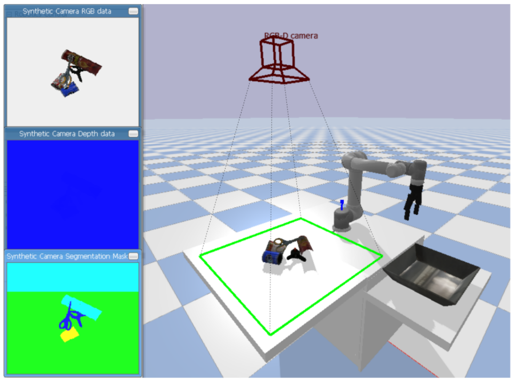
Synthetic Camera RGB data
Synthetic Camera Depth data
Synthetic Camera Segmentation Mask
RGB-D camera

Fig. 2. Virtual Environment in PyBullet

A. Metrics

As a performance metric, for a grasp, a rectangle metric [6] was used. In order to get the grasp prediction rectangle from the image-based grasp representation, the value corresponding to each pixel in the output image is mapped to its equivalent rectangle representation. A rectangle metric is used to measure the performance of the grasp. A optimal grasp is one that satisfies two conditions:

- The intersection over union (IoU) score between the ground truth grasp rectangle and the predicted grasp rectangle is more than 25%.
- The offset between the grasp orientation of the predicted grasp rectangle and the ground truth rectangle is less than $30°$.

Each network approach and their three different object placement topologies are evaluated with the following metrics:

- The grasp success rate: this parameter indicates the average of tries that it takes to grasp an object, considering that a grasp is detected.
- The manipulation success rate: this parameters indicates the number of attempts that it took to the robot to place an object grasped in the designated basket.
- Percentage of objects removed from workspace: this parameter indicates the total number of items removed from the workspace without taking tries into consideration.

The model was tested with two standard datasets which contains samples from household object, objects that had complex geometry and objects in clutter.

B. Results

In Table I the results from our experiments are presented. In our experiments, both network models were able to generate grasps for every object in the dataset and also create multiple grasps for multiple objects in clutter.

Both networks achieved better when the objects where isolated and performed the worst when the objects were in a pile. The task of cleaning the table was almost carried away in each experiment with the exception of the GG-CNN approach in the ”Pile” scenario. Despite GR-ConvNet's better performance in predicting the optimal grasp from all other

Network	Scenario	Manipulation success rate	Grasp success rate	% of object removed from workspace
GR-CNN	Packed	61.3	62.5	98
	Pile	51	55.2	98
	Isolated	86.2	89.8	-
GG-CNN	Packed	62.5	62.5	100
	Pile	47.1	52.9	82
	Isolated	84.8	90.2	-

TABLE I
PERFORMANCE PARAMETERS OF THE GR-CNN.

CNN in the literature, the performance for both networks is similar when they are tested with in a virtual simulation. This can be ascribed to the number of rounds per experiment which is very low, thus we cannot obtain reliable data for the network performance. To get a better overview of the experiment, the number of grasp attempts should have been at least 100. Also one key reason for the low grasp success rates, is that both networks do not pick the optimal grasp based on the quality which results in a choosing a ”bad” grasp. Furthermore, some of the object geometry confuse the networks into coming to the conclusion about the quality grasp map of the image. For example in Fig.4 we can see that because the GG-CNN network based its results only with a depth image, object with a small depth such as scissors and clamp cannot be treated the same as a mustard bottle object. Another observation we made while running the experiments, was that the robot hand would sometimes do some ”awkward moving” or collide with the table resulting in the object falling from the gripper.

In the following figures (Fig 3 and Fig 4) results in the form of success rate of of grasping and manipulation for all the used object are provided.

For the GR-ConvNet we observe a similar success grasping and manipulation rate throughout the objects expect the for CrackerBox. The network finds it difficult to generate a good quality grasp pose for that object. This is due to the shape of the object his and the placement in the robot's workspace. Because the hand approaches for above, if the cracker is lying on the workspace (big area face down) the gripper cannot grasp the object.

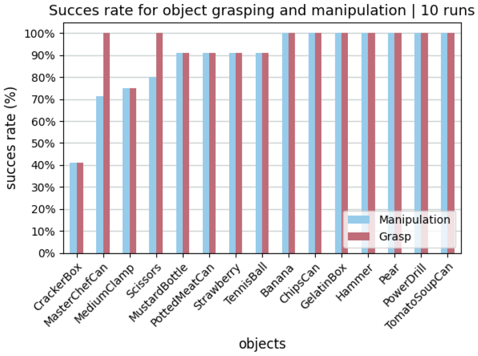

Fig. 3. Performance of the GR-CNN with isolated objects.

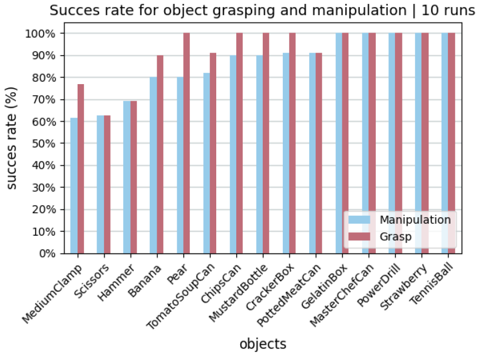

Fig. 4. Performance of the GG-CNN with isolated objects.

In the case of the GG-CNN approach we can see that the GR-CNN had problems to identify a optimal grasp for the scissors, the hammer and the medium clamp. The reason of this could be the geometry of these object, in particular the geometry of the scissor that does not have much depth and therefore, is harder for the GG-CNN identify the contour of the object (the input of the network is a depth image).

In both approaches the objects with lower manipulation success rate are objects that have dissimilar geometries (curvy) that makes the manipulation of the object by the robot difficult.

V. Conclusion

Two different networks are studied and tested in a virtual environment for their performance in identifying the best grasp pose, finding the optimal trajectory plan and also in their ability to complete a specific task(clear table) in isolated scenarios but also in cluttered environments. For better evaluation of these we need to not only make more attempts for an experiment but also add more objects in the object dataset. That will provide us with a more reliable rasult data. Also some objects force the network and the control module to some faulty calculations regardless the image input (RGB-D or just depth image) and maybe some extra dimension about specific features detected objects geometry which is going to be generated in the pre-processing stage of the image is necessary.

References

[1] https://github.com/SeyedHamidreza/cognitive_robotics_manipulation/assignment_description.pdf

[2] Kumra, Sulabh, Shirin Joshi and Ferat Sahin. “Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network.” 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2020): 9626-9633

[3] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks., arXiv:1505.00387v2, 2015.

[4] Thomson Alex, Neocortical layer 6, A review, Frontiers in Neuroanatomy, Volume: 4, 2010, pp 13, doi: 10.3389/fnana.2010.00013, ISSN : 1662-5129

[5] Douglas Morrison, Peter Corke, and Jürgen Leitner. ”Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach”.

[6] Y. Jiang, S. Moseson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 3304–3311