Practical course robotics project proposal

Marc Tuscher, Ralf Gulde

May 13, 2019

1 Introduction

Recent advancements in robotics and automated systems have led to the expansion of autonomous capabilities and more intelligent machines being utilised in ever more varied applications [LS05, JYL+14]. While manipulating objects is relatively easy for humans, reliable grasping arbitrary objects remains an open challange for general-purpose robots. Resolving it would advance the application of robotics to industrial use cases, such as part assembly, binning, and sorting. The capability of adapting to changing environments is a necessary skill for task generalised robots [KKGB12, KP10]. A grasp describes how a robotic end-effector can be arranged to securely grab an object and successfully lift. Grasp planning relates to the path planning process that is required to securely grasp the object and maintain the closed gripper contacts to hold and lift the object from its resting surface [BK00]. Initially, deriving the grasp-points of a perceived object is an essential aspects. Consequently, accurate and diverse detection of robotic grasp candidates for target objects should lead to a better grasp path planning and improve the overall performance of grasp-based manipulation tasks.

In this work we propose a deep learning strategy to detect grasp points from raw sensor data and plan collision free robot paths to manipulate arbitrary objects. In the past decade, deep learning has achieved major success on detection, classification, and regression tasks [BRF13, KSH12]. Its key strength is the ability to leverage large quantities of labelled and unlabelled data to learn powerful representations without hand-engineering the feature space. To conclude this work we experimentally validate our approach using a humanoid robotic system and a structured light RGB-D sensor.

This project proposal is organized as follows: In section 2 the goal of the work is described. Further, in section 3 a formulation and the proposed methods are described. Section 4 thematizes the milestones of implementing the approach in a robotic demonstator system to evaluate the proposed approach.

2 Goal (Ralf)

Goals:

* detect parallel jaw grasps a priori unknown objects
* plan collision free paths
* test
* drop the grasped object in a bin

Restrictions:

* dynamic changing environments are not considered
* objecte must in principle be manipulable by a two-finger parallel gripper.

optional

* suction cup gripper

Robot grasping can be subdivided into two related problems: perception and planning. A typical example of this approach is the ROS grasp pipeline [CJCH12]. Herein, a CAD model is mapped to a point cloud of the object. Optimal grasp poses are then derived from the CAD model. This works well in ideal scenarios, however, in most realistic applications the scenario is far from ideal. Our main goal is to set up a framework which extracts robust grasp poses from RGBD-Images and grasps an object lying on a table.

3 Problems and methods

This section gives an overview over the general problems of the objective and the methods we will use to overcome these difficulties.

- Problems:
  - Localization of objects.
  - Localization of robust grasp points.
- Methods:
  - Robot control using *rai* framework
  - Sampling of grasp candidates using a grasping policy (e.g. *CrossEntropyRobustGraspingPolicy*[1])
  - evaluation of grasp candidates with pretrained *GQCNN*[2]
  - localization of objects using bounding boxes (if necessary)

3.1 Perception

Robotic systems increasingly leverage RGB-D sensors and data for tasks like object recognition [LBRF11], detection [LBRF12], and mapping [DHR+11]. RGB-D sensors like the Kinect are inexpensive, and the extra depth information is invaluable for robots that interact with a 3-D environment. Recent work on grasp detection focusses on the problem of finding grasps solely from RGB-D data [SDN08]. Due to the recent successful results of deep learning methods in computer vision, many robotics researchers have applicated the insights to the research field of robotic grasp detection. These techniques rely on machine learning to find the features of a good grasp from data. Visual models of grasps generalize well to novel objects and only require a single view of the object, not a full physical model [ESTA14]. A deep CNN is built with multiple layers to extract information representations. Goodfellow et al. [GBC16] has stated that the representation of the learning process of a deep neural network has similar attributes to the method through which information is processed by the human brain. Literature suggests that lower level features are identified using the convolutional layer while application specific features are extracted by the fully connected portion of the network. Grasp detection methods can be categorized as follows:

- Structured Output for grasp candidate parameters ($e.g.: x, y, z, w, ix, jy, kz$)
- Grasp Robustness (selecting a grasp candidate based on the robustness factor)

3.1.1 Structured Output

The most popular method for structured grasp detection was the sliding window approach proposed by Lenz et al. [ESTA14]. In their approach, a classifier is used to predict if a small patch of the image contains a potential grasp. This method yielded a detection accuracy of 75% and a processing time of 13.5 s per image. As an alternative, Redmon et al. [RA15] proposed the one-shot detection method. In most one-shot detection methods, a direct regression approach for predicting a structured grasp output is used. In the first one-shot detection approach, the authors argued that a faster and more accurate method was necessary and proposed to use transfer learning techniques to predict grasp representation from images [RA15]. They reported a detection accuracy of 84.4% in 76 milliseconds per image.

[1] `https://github.com/BerkeleyAutomation/gqcnn/blob/a0930e9d2fef3c930c41dd91cde902d261348fbe/gqcnn/grasping/policy/policy.py#L627`

[2] `https://github.com/BerkeleyAutomation/gqcnn`

3.1.2 Grasp Robustness

Learning a grasp robustness function also had been the central idea of many studies in deep grasp detection. The researchers used this function to identify the grasp pose candidate with the highest score as the output. Grasp robustness described the grasp probability of a certain location or an area of an image. A method to learn an optimal grasp robustness function was proposed by Mahler et al. [MLN+17]. They considered the robustness as a scalar probability in the range of [0, 1]. The authors compiled a dataset known as Dex-Net 2.0 with 6.7 million point clouds and analytic grasp quality metrics with parallel-plate grippers planned using robust on a dataset of 1500 3D object mesh models *Dex-Net 2.0 Dataset*[3]. They further trained a grasp quality convolutional network (GQ-CNN) that was used to learn a robustness metric for grasp candidates. They tested their CNN with their dataset which achieved an accuracy of 98.1% for grasp detection.

3.2 Motion Planning

Sampling grasp points from RGBD-Images is a major problem and has been studied by a wide variety of approaches. A core problem in robot motion planning and robotics in general is defined as: given a cartesian position $y \in \mathbb{R}^d$, find a corresponding joint configuration $q \in \mathbb{R}^n$ such that

$$y = \phi(q), \phi : \mathbb{R}^n \rightarrow \mathbb{R}^d \tag{1}$$

In robotics this problem is referred to as the inverse kinematics problem. We formulate this problem as a constrained optimization problem

$$\min_x f(x) \text{ s.t. } g(x) \leq 0,\ h(x) = 0 \tag{2}$$

where different objectives can be defined. Such objectives can either be equality constraints h, e.g. the distance between to frames, inequality constraints g or costs terms f. The generality of this formulation allows to directly incorporate constraints for avoiding collisions. To solve this constrained optimization problem we rely on the KOMO library which formulates the constrained optimization problem as an unconstrained optimization problem, which is then solved using Newtons method [Tou17]. As a first step, we only use inverse kinematics with some kind of collision avoidance for motion planning. For further improvements we test the path planning feature of the KOMO library which allows to calculate complete trajectories for path planning.

[3] https://berkeley.app.box.com/s/6mnb2bzi5zfa7qpwyn7uq5atb7vbztng

4 Milestones

The following section presents a timeline which serves as a workplan for the project.

	Marc	**Ralf**
16.05.19	get *RAI* running on machines with nvidia gpu	port *GQCNN* to python3, load pretrained model and basic setup
23.05.19	set up bounding box detector for object localization	transform the output pose of *GQCNN* into cartesian coordinates
06.06.19	Tune control, test KOMO path planning	optimize prediction speed, check online capabilities
13.06.19	Tune camera intrinsics with tensorflow-graphics[4]	Experiment on combining opencv pipeline and *GQCNN*
rest of time	**cleanup super messy code**	

[4] `https://github.com/tensorflow/graphics`

References

[BK00] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. 1:348–353 vol.1, April 2000.

[BRF13] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. pages 387–402, 2013.

[CJCH12] Sachin Chitta, E. Gil Jones, Matei Ciocarlie, and Kaijen Hsiao. Perception, planning, and execution for mobile manipulation in unstructured environments. *IEEE Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 19(2):58–71, 2012.

[DHR+11] H Du, P Henry, X Ren, M Cheng, DB Goldman, SM Seitz, and D Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. 2011.

[ESTA14] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. pages 2147–2154, 2014.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[JYL+14] Zhangfeng Ju, Chenguang Yang, Zhijun Li, Long Cheng, and Hongbin Ma. Teleoperation of humanoid baxter robot using haptic feedback. pages 1–6, 2014.

[KKGB12] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.

[KP10] J. Kober and J. Peters. Imitation and reinforcement learning. *IEEE Robotics Automation Magazine*, 17(2):55–62, June 2010.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LBRF11] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. pages 1817–1824, 2011.

[LBRF12] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Detection-based object labeling in 3d scenes. pages 1330–1337, 2012.

[LS05] M. Lopes and J. Santos-Victor. Visual learning by imitation with motor representations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(3):438–449, June 2005.

[MLN+17] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[RA15] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. pages 1316–1322, 2015.

[SDN08] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[Tou17] Marc Toussaint. A Tutorial on Newton Methods for Constrained Trajectory Optimization and Relations to SLAM, Gaussian Process Smoothing, Optimal Control, and Probabilistic Inference. In Jean-Paul Laumond, Nicolas Mansard, and Jean-Bernard Lasserre, editors, *Geometric and Numerical Foundations of Movements*, volume 117, pages 361–392. Springer International Publishing, Cham, 2017.