

Proposal for the Bachelor thesis

Student: Anna Fedorchenko
Supervisor: M.Sc. Atanas Tanev
Advisor: Prof. Dr.-Ing. Rüdiger Dillmann

December 11, 2019

Current title of the thesis

Data-driven robotic grasp synthesis in a simulated environment.
(Datengesteuerte Robotergreifenssynthese in einer Simulationsumgebung.)

Motivation

Grasping is a beginning of many diverse robotic manipulations in research and industry (like ...?). It is crucial for the grasping part to succeed in order to be able to complete the proceeding steps of the manipulation.

There are some variations to this problem such as: the type of the object to be grasped (rigid/non-rigid, transparent, having certain shape, color,...), the number of objects (cluttered environment, a couple of objects, a single one), whether the object is known, familiar or completely unknown, the type of the gripper (two-finger, five-finger,...), the type of the camera and so on.

Different approaches have been created to solve the grasping problem. A lot of them that were developed in the last couple of years are based on Machine Learning. Machine learning helps to incorporate the training examples in order to learn from them. —Übergang zum human labeling—

Human labeling is time and effort consuming. Although there exist software for easier labeling (LabelFusion) and already pre-labeled datasets (Cornell Dataset) in many cases depending on the use case still new objects have to be labeled. The reinforcement learning approach helps to avoid the labeling part by making system to acquire experience from its own successes and failures. Such self-supervised system have showed good results at solving the grasping problem.

Reinforcement learning paradigm (?) is similar to how humans learn. One of the easy everyday tasks that every human learns is catching a ball. Either we watch somebody else doing it or try it out ourselves. We automatically in the course of split seconds estimate the size and mass of the ball, what our position should be in order to catch it. The result of catching - positive or negative - influences our behavior for the next time when we have to perform the same task.

Another important aspect of solving the grasping problem is being able to adapt to new surrounding environments and new objects.

— One of the drawbacks of using machine learning is the need to use a lot of

training data to avoid under- and overfitting.

The state of the art approach Information??? Pipeline —state of the art in FZI—

Problem statement

The focus of the thesis is to develop an approach for the robot to learn how to grasp rigid objects using images from RGB-D camera and two-finger parallel gripper. The learning process should be self-supervised: any human labeling should be avoided. This will be achieved with the help of deep and reinforcement learning. The training and testing will be conducted in a simulated environment. In case of successful performance of the developed algorithm in the simulation, the next step will be to prepare the transition to the real-world environment with the help of domain randomization.

One of the advantages of such algorithm is the ability to generalize to new objects. Previous research shows that data-driven algorithms that are based on machine learning have successful rates of grasping objects that were never used in training. Another advantage is the fact that no human labeling will be required. This spares time and effort.

Training the robot in real world is time consuming and expensive, simulation makes it possible to avoid these costs. However, there is a so-called "reality gap" problem, when the algorithm trained in simulation does not perform well in the real world scenario. One of the possible solutions to this problem is using domain randomization and/or fine-tuning with training in reality. The sim-to-real transfer is an additional optional part of the thesis, that will take place only if the algorithm performs well in simulation.

Related work

Throughout the years there has been a lot of research on the topic of robotic grasping. Sahbani et al. [1] divided different approaches in analytical and empirical. Analytical approaches investigate the physics, kinematics and geometry of the object and the robot in order to determine contact points and gripper position [2], [3]. However, as Bohg et al. [4] stated, they require exact 3D models of objects and are not robust to noise.

Empirical or also called data-driven approaches are based on sampling grasps for objects and then evaluating them [4]. In some methods robots learn to grasp by comparing its grasp attempts to the database of human labeled grasps [13]. Some empirical approaches try to avoid the effort of human labeling to create databases for training, instead they attempt grasps on objects and learn from these attempts. Berscheid et al. [16] trained the neural network by executing grasps and adapting the network according to success or failure of the attempt. Empirical approaches can generalize better to new objects and are more resistant to noise. However they usually require big resources of data [16].

In data-driven approaches training data is required to learn for a successful grasp. Levine et al. [5] used 14 robotic arms that sampled 800 000 grasp at-

tempts. Pinto and al. [6] used one robot manipulator to conduct 50 000 grasp attempts in course of 700 hours. However, it is expensive to collect such amount of data and it is very time consuming, this is where the arguments for learning in simulation come to a point.

Goldfeld et al. [7] created a grasp planner containing database of grasps for 3D models of different objects, that were generated using GraspIt! [8] simulation engine. Kasper et al. [9] introduced the system for digitizing objects. In year 2010 the OpenGRASP [10] simulation toolkit for grasping and dexterous manipulation was created.

James et al. [11] developed an approach that used only a small amount of real-world training data in addition to simulation, which helped to reduce the real-world data by more than 99%. Bousmalis et al. [12] implemented a grasping method that helps to significantly reduce the amount of additional real-world grasp samples. In their experiment 50 times less real-world samples were required to achieve the same level of performance compared to their previous system.

Another advantage of using simulation is the ability to pretrain the network. Redmon et al. [13] stated that "pretraining large convolutional neural networks greatly improves training time and helps avoid overfitting".

Approach

The data-driven approach introduced in the thesis consists of predicting good grasp parameters for the object, executing the predicted grasp and adapting the network according to the result of the grasp. The scheme of the selected approach is represented in Figure 1.

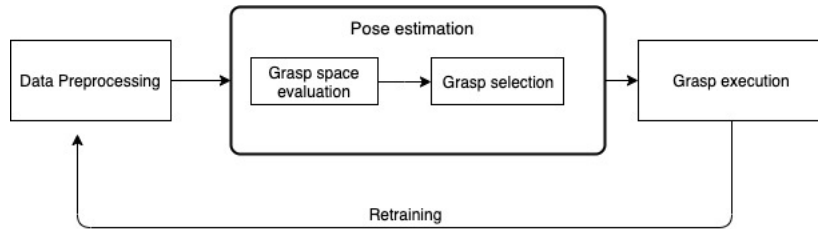


Figure 1: Steps of the algorithm. After preprocessing the input image from the camera the parameters of location of the gripper will be evaluated in course of a two step approach - evaluation of the grasp space and grasp selection. Then the gripper will execute a planar grasp with parameters that were calculated in the last step. Depending on the result of the grasp, the system will be trained. After that the new loop cycle will start for the next object.

- Data Preprocessing

Although most part of the training will take place in a simulation, some of the training samples will be taken from the real world. The depth camera can take an image of the scene, the algorithm will then simulate the scene and try to locate and grasp the object. This can help to bring the simulation closer to the real-world problem. In case of using the 3D camera

it might be helpful to additionally use data augmentation for creating more diversified training data. Redmon et al [13] used random translating and rotating the input image during training, generating 3000 training examples per image.

- Pose estimation

In this part the correct gripper location to execute a planar grasp of the object will be calculated. The gripper location is defined as (x,y,q) where x, y are coordinates of the gripper along according axes (the height of the gripper will be predefined, the depth camera will help to calculate how low does the gripper have to execute a grasp), q is the angle of the rotation of the gripper around the z - axis. All the grasps are restricted to planar grasps in order to reduce the possible grasp space, so gripper rotations around x - and y -axes are emitted. (See Figure 2)

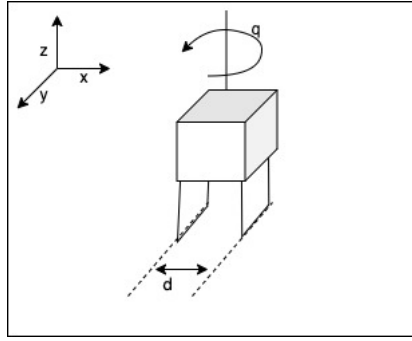


Figure 2: The gripper position will be defined by its x and y coordinates and angle q around the z axis. The jaw width will be set to a possible maximum before executing the grasp in order to cut down the number of gripper parameters that have to be estimated for the grasp.

- Grasp space evaluation

Convolutional neural networks are better at classification rather than regression [6], which is why they are going to be used not to directly define the (x, y, q) parameters but to compute the heatmap of probabilities for all possible combinations of x, y, q .

Heatmap is one of the representations of probabilities for possible grasps. Tobin et al. [14] divided the bounding box of the object in buckets and sampled the grasp for every bucket. Pinto et al. [6] extracted patches from the input image and calculated a 18-dimensional binary likelihood vector for each patch, each component of the vector estimates whether the object can successfully be grasped at the center of the patch with respectively $0^\circ, 10^\circ, \dots, 170^\circ$ of rotation of the gripper. In [15] Zeng et al. used neural networks to calculate 16 probability maps, every value in each map represents whether the object is graspable at the to the value corresponding input pixel. There are 16 maps for 16 rotation angles (with $22,5^\circ$ step).

- Grasp selection

Using the heatmap that was calculated in previous step one or several grasp parameters will be selected. It will not be necessary the grasp with the biggest probability, other metrics can be applied, e.g. random in order to explore the grasp space. Berscheid et al. [16] used four different selection functions to explore the grasp space and decrease error rates.

- Weighted retraining

Retraining of the network will be conducted using the Reinforcement learning approach. The weights in the neural network will be changed according to achieved result in the previous step. The formula for the calculation of the new weights is to be defined. It will include the reward for the successful or failed grasp.

All the steps will be continued for new experiments. The duration of the training has to be long enough to achieve a good success rate but also not too long to prevent the network from overfitting. The focus of the thesis is to conduct all the training in a simulation environment. The existing database of 3D objects is to be used.

References

- [1] Anis Sahbani, Sahar El-Khoury, and Philippe Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.
- [2] Van-Duc Nguyen. Constructing force-closure grasps. *The International Journal of Robotics Research*, 7(3):3–16, 1988.
- [3] Richard M Murray. *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [4] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2013.
- [5] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.
- [6] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.
- [7] Corey Goldfeder and Peter K Allen. Data-driven grasping. *Autonomous Robots*, 31(1):1–20, 2011.
- [8] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. 2004.

- [9] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.
- [10] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moio, Jeannette Bohg, James Kuffner, et al. Opengrasp: a toolkit for robot grasping simulation. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 109–120. Springer, 2010.
- [11] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019.
- [12] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250. IEEE, 2018.
- [13] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.
- [14] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489. IEEE, 2018.
- [15] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. 2019.
- [16] Lars Berscheid, Thomas Rühr, and Torsten Kröger. Improving data efficiency of self-supervised learning for robotic grasping. *arXiv preprint arXiv:1903.00228*, 2019.