Generation of Synthetic AI Training Data for Robotic Grasp-Candidate Identification and Evaluation in Intralogistics Bin-Picking Scenarios

Dirk Holst(✉), Daniel Schoepflin, and Thorsten Schüppstuhl

Hamburg University of Technology (TUHH), Am Schwarzenberg-Campus 1, 21073 Hamburg, Germany
dirk.holst@tuhh.de

Abstract. Robotic bin picking remains a main challenge for the wide enablement of industrial robotic tasks. While AI-enabled picking approaches are encouraging they repeatedly face the problem of data availability. The scope of this paper is to present a method that combines analytical grasp research with the field of synthetic data creation to generate individual training data for use-cases in intralogistics transportation scenarios. Special attention is given to systematic grasp finding for new objects and unknown geometries in transportation bins and to match the generated data to a real two-finger parallel gripper. The presented approach includes a grasping simulation in Pybullet to investigate the general tangibility of objects under uncertainty and combines these findings with a previously reported virtual scene generator in Blender, which generates AI-images of fully packed transport boxes, including depth maps and necessary annotations. This paper, therefore, contributes a synthesizing and cross-topic approach that combines different facets of bin-picking research such as geometric analysis, determination of tangibility of objects, grasping under uncertainty, finding grasps in dynamic and restricted bin-environments, and automation of synthetic data generation. The approach is utilized to generate synthetic grasp training data and to train a grasp-generating convolutional neural network (GG-CNN) and demonstrated on real-world objects.

Keywords: Synthetic data · Grasp Simulation · Tangibility · Bin picking · Automation

1 Introduction

Grasping under uncertainties and in highly dynamic environments remains an unsolved problem and has repeatedly been a central target of state-of-the-art research. Current approaches use empirical solution strategies and try to derive rules for robotic grasping by applying machine learning. To identify a valid grasp for a given object different strategies can be applied such as the 6D pose estimation of objects in RGB images [1] or a graspability analysis based on point clouds [2] or depth images [3]. The availability of individualized training data is a constant problem in this research area and prevents

K.-Y. Kim et al. (Eds.): FAIM 2022, LNME, pp. 284–292, 2023.
https://doi.org/10.1007/978-3-031-18326-3_28

a wide adoption of developed technologies. The manual generation of real-world data is considered time- and cost-intensive and is prone to errors [4], especially concerning the ground truth which is used for supervised machine learning algorithms. On the other hand, synthetic data generation is a promising approach for generating large and high-quality data sets for training neural networks. It overcomes many of the problems of real data generation, such as scalability of data synthesis, inaccurate annotations, or the introduction of unwanted effects such as dataset bias. Existing solutions for synthetic data generation often cover only a limited fraction of possible annotations, which means that the generated datasets can only be used for a subset of possible grasping research [1, 3, 4]. To support this branch of research and to ensure, that results are more comparable, it is important to cover as many different solution strategies as possible with one data set. Therefore, this work presents a holistic toolchain that creates synthetic data and annotations for object identification and segmentation, object pose localization, grasp candidate detection, and grasp candidate quality evaluation.

Following previous work [5], an existing synthetic data generator is used for the creation of training images, to identify objects on load carriers in intralogistics settings. This tool will be extended with functionalities to annotate grasps for a two-finger parallel gripper. Of particular importance is the systematic graspability analysis of single objects under uncertainties in the physics simulation software Pybullet, to integrate insights from analytical graspability research into synthetic data generation. For the determination of suitable grasp candidates, the underlying geometry of the object model will be analyzed and simplified to an object skeleton. The calculated bones are used to approach individual points with a two-finger parallel gripper and then be tested for force-fit under varying simulation parameters. Afterward, valid grips are transferred under a strict set of rules to fully loaded carriers in the 3D rendering software Blender. This method is intended to ensure that context-based intralogistics information is preserved and not altered by the tangibility analysis.

2 Related Works

Research in the field of robotic grasping can mostly be separated into analytical and empirical methods [6]. The former tries to calculate valid grasps for different types of grippers, with varying amounts of contact points [7, 8], by solving analytical problems. The complexity of this approach is strongly based on the number of equations and geometric properties, to be calculated for determining valid grasp candidates. The latter describes a set of data-driven methods, to derive rules for robotic grasping based upon examples [9]. They represent a broad topic in machine learning and the field of grasp planning has been strongly influenced by the progress of Convolutional Neuronal Networks (CNN) [3, 10]. It is possible to solve a variety of problems like object detection, image segmentation, 6D pose estimation [1], and grasp candidate generation [10] or evaluation [3]. All of them have in common, that they need training data to adapt their underlying algorithm to a given task. Since training on real robotic systems can be time-consuming, expensive, and sensitive to changes to the physical setup [11] and creating annotations for real image datasets is error-prone, another approach seems feasible: Synthetic Data. This approach is using 3D-Rendering software to create image

databases and is able to create perfect ground truth to aid supervised machine learning algorithms.

To incorporate knowledge from analytical research into empirical solutions and synthetic data generation, the use of physics simulations has become popular [4, 12, 13]. They offer the possibility to rapidly recreate grasping situations and check whether a grasp candidate results in a successful grasp or fails to hold an object. Two main challenges arise from this approach: The first one is the free choice of simulation parameters. Since there is often no unique solution to the parameter space, it is necessary to determine intervals from which to choose values and evaluate simulation results under a given uncertainty [3, 14]. The second problem is based upon geometric analyses and the varying complexity of shapes. Since a grasp planning algorithm should be able to evaluate a huge variety of objects, techniques have been developed to simplify mesh structures into a couple of geometric primitives [15] or to collapse a volumetric body into lines and segments to build a skeleton out of its structure. This reduced model can then be used to perform path planning of a simulated gripper and to evaluate high-quality grasp candidates [16].

Recent work shows the use of synthetic data generation in the space of robotic grasping and bin picking. Depending on a given task, these generators create a certain amount of annotations, like 6D object pose information [4] or grasp candidates with a quality value to differentiate between several grasp candidates and their rate of success [3]. There is no one method that outperforms all others in grasping situations and creating several datasets for various applications with separate pipelines is laborious. Therefore, this paper presents a method that includes different types of annotations and rendering methods and combines techniques from different research areas such as Geometric Analysis, Physical Simulation, and Synthetic Data Generation.

3 Methodology - Synthetic Grasp Data Generation

This chapter presents the systematic generation of synthetic grasp data. As shown in Fig. 1, the process consists of six sub-steps across two programs and two separate databases.

As a foundation for the general graspability analyses in Pybullet, the pipeline extends an existing Graspit! [12] clone [17] with features such as geometric analyses, partial domain randomization of simulation parameters, and the ability to store information in a database. For the final data generation and the transfer of found grasps, the 3D rendering software Blender is. The sub-steps are described in the following:

1) Deriving **Object Skeletons**, to enable Grasp Simulation: Graspability analyses start with an understanding of the object to be grasped. To determine suitable grasp candidates, the mesh structure of an object is repeatedly collapsed until only a skeleton, consisting of individual interconnected points, emerges [16]. The simplified structure is located either on or within the mesh model and can consist of several hundred points. To further reduce the amount of data points and to identify meaningful grasp centers, the individual points of the determined skeleton are clustered, based on their

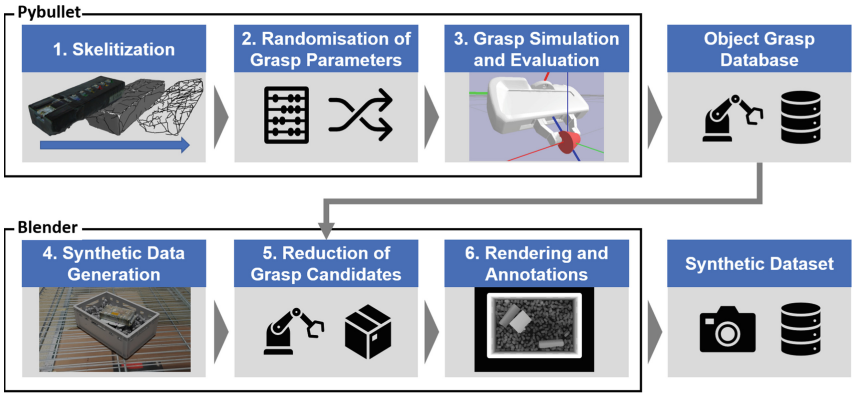

Fig. 1. Pipeline for synthetic grasp data generation, consisting of geometric and graspability analysis in Pybullet and synthetic image data generation, including annotations, in Blender. Results of the respective simulations are stored in databases and are available for analyses and machine learning algorithms.

spatial coordinates, using k-means clustering. The determining centers of the individual clusters then serve as an approaching point for the simulated gripper, which is moved to the object from different directions using spherical coordinates.

2) **Randomisation** of **Grasp Parameters** and rigid body **Simulation**: A physical simulation offers the possibility to adjust a large number of parameters and thus significantly influence the outcome. Since it is usually not possible to determine exact and correct values a priori, intervals for parameters such as mass, grasp force, and friction coefficient are determined based on physical constraints. These parameters can directly be derived from real objects and the robotic grasping systems used later on. From this value space, randomized values are chosen for each grasp candidate and are then checked for force-fit under consideration of the weight force. This process is repeated at least ten times for each grasp found so that it can be analyzed which parameter combinations are leading to successful grasps. If it turns out that an object can only be gripped under difficult conditions, the simulation should be repeated more frequently in order to generate meaningful results.
3) **Evaluation** of Grasps: A grasp is considered successful if the object to be grasped has remained in the gripper under the action of its weight. Due to the randomly chosen simulation parameters and in order to create the possibility to compare grasps with each other and to enable statistical analyses, like distributions of succesfull grasp parameters, the contact points of a these grasp including their simulation parameters are saved.
4) Transfer **to Synthetic Data Generation** Toolbox [5]: A structured set of rules is used to generate packed transport boxes, ensuring that logical consistencies are maintained within the training data domain. These include lighting conditions, camera positions, positioning of objects, and packaging material used. The aim is to generate scene compositions that are as realistic as possible and to identify intervals for partial randomization for the parameters mentioned. The program developed for Blender generates random variations of packed transport boxes and simulates the insertion

of objects to be gripped through rigid-body simulation. The tightly integrated set of rules ensures for each generated scene that visual material matches the generated annotations and does not contain any unwanted inconsistencies.

5) **Reduction of Grasp-Candidates** with respect to the boxing scenario and reachable grasps: One problem in grasp planning is to identify suitable candidates in dynamic and unpredictable environments. Here, the gripping system has to be able to reach the identified grasp and subsequently execute it. For this purpose, ray tracing is used to examine the reachability of the previously identified grasp candidates. It is evaluated whether the individual fingers of the gripping system can be placed and the object to be gripped is not covered by another object or the grip is not too low and prevented through collisions.
6) **Derivation of Grasp Annotations** and **Rendering** of the Images: The goal of the developed pipeline is to cover a wide range of possible machine learning algorithms and to enable further research on different methods, with only one data generation tool. Core tasks in grasp planning are image classification, object identification, object localization, image segmentation, 6D pose determination, depth images processing as well as the possibility to evaluate the quality of grasps. The solution presented here consists of a database, which archives all annotations from simulations parameters, in addition to a render pipeline, which creates the corresponding depth maps and segmentation masks from the RGB images generated in Blender, as shown in Fig. 2. Furthermore, information about the generated scene compositions, such as camera and light positions or packaging material used, is also archived. This offers the possibility to statistically evaluate the bias of a dataset and to generate new data if necessary to increase diversity.

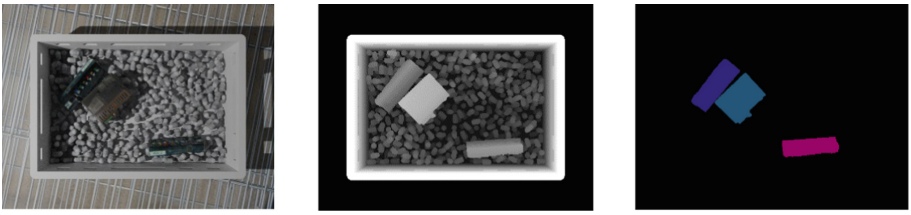

Fig. 2. Showcase of the rendering pipeline, (left) realistic rendering with the Cycles engine, (middle) derived depth image, (right) corresponding segmentation mask

Besides the 6D pose determination of objects or the generation of grasp candidates on depth images, the field of grasp evaluation procedures is still available for research. For example, DEX-Net [3] uses the method of identifying grips analytically on depth images and then having them evaluated by a neural network. For learning such a method, grips have to be distinguishable in terms of their quality. This can be done using the stored simulation parameters of the graspability analysis (Step 3). An object to be grasped can be grasped and held with different ease at different points, which is why grasping data with unusual parameter combinations can be statistically identified and evaluated with a lower quality. To enable further research in this branch, each grip in the final training data

is uniquely identifiable and can be subsequently modified. This provides the opportunity to observe the outcome for various metrics of a given grasp quality index.

4 Demonstration of the Developed Toolbox and Training Results

With the presented toolbox, a test data set for different geometric objects such as cubes, pyramids, and rings are created. This includes the sex step approach from general graspability analysis, the archiving of found grips, as well as the final transfer of possible grasps into packed transport boxes, as they are commonly found in intralogistics settings (Fig. 3).

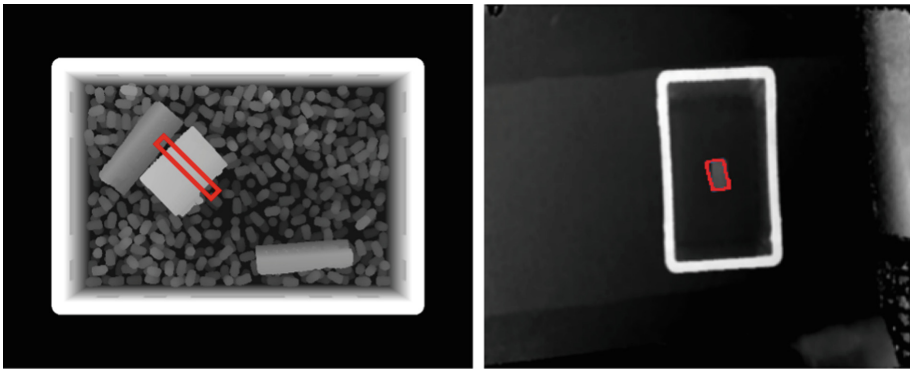

Fig. 3. (left) Synthetic depth image of three objects in a transport box with packaging flips, including annotated grasp in red. (right) Real depth image, with a found grasp for a test cube by the GG-CNN Architecture, trained only on synthetic data.

The parameters for the gripping simulation from step 2 were adapted to the technical specifications of the two-finger parallel gripper of the Panda Franka Emika robot and performed with a virtual replica of the system. The randomized grasping simulation produced results to the expected degree, such as objects with lower mass being easier to hold, or that increased closing force is associated with increased grasping success rate.

To demonstrate that purely synthetic data can be used to determine grasp candidates in real images, a non-pretrained Gras Quality Evaluation network (GQ-CNN) [10] is used. This architecture was chosen for its state-of-the-art performance, but the dataset created is not limited to it, as a wide range of annotation is generated. The training dataset consists of 525 depth images, with slightly varying bird's eye camera angles, and a total of 176,000 grasp candidates for 9 different objects. Annotations for grasps are given in the format: Grip center and finger spread in pixels and rotation angle in radians. The data is split into 80% training data and 20% validation data.

To demonstrate that the domain gap between synthetic- and real-world data is successfully overcome, a RealSenseL515 lidar camera is used to capture depth images of heavily cluttered test scenes (s. Figure 4). The scaling of the grayscale was normalized to the minimum and maximum height values of the recorded scene. Test series with real

depth images have shown that the used architecture is able to identify grasp candidates for the examined objects, by giving them the highest grasp probability. The network was able to distinguish between box edges, unknown objects, and objects to be grasped and did not output unwanted grasps.

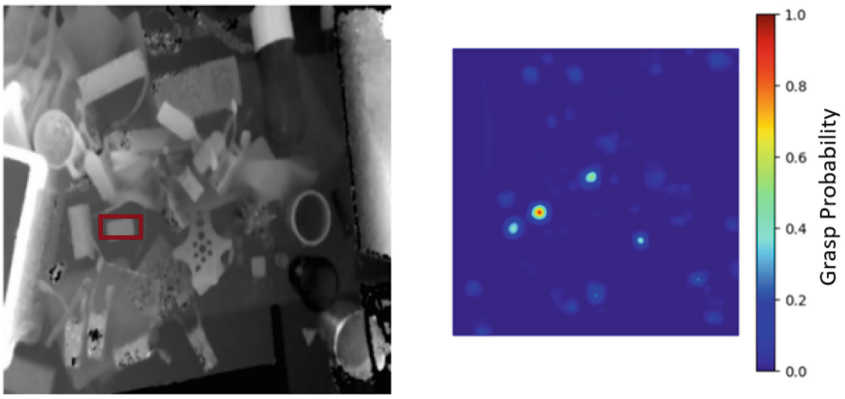

Fig. 4. (left) Successfully plotted grasp in red for a specific test object, using a wide variety of interfering sources, including objects similar to the test object. (right) The generated output of the GG-CNN [10] shows the highest probability for a possible grasp, including location and gripper width in pixel coordinates and angle of rotation.

Additionally, grasping tendencies towards objects that have parallel faces were detected, which can be attributed to the simulated two-finger parallel gripper and a significantly higher number of annotated grasps for this form of objects. This tendency can be minimized in further work by evenly distributing candidate grasps for different objects. Even in test environments with a large number of interfering sources, as shown in Fig. 4, grasps for the target object could be successfully identified.

5 Conclusion

In this work, a cross-disciplinary method for the generation of synthetic grasp data in highly dynamic environments was presented. Methods for geometric object analysis of mesh models, detailed graspability analysis under uncertainty, and techniques for transferring possible grasps into context-based scenes of packed transport boxes from intralogistics are used. The developed toolchain enables further research in grasp planning and execution by a wide range of annotations and offers the possibility to produce targeted data sets for individual problems.

Test series have shown that it is possible to identify grasp candidates on real depth images by using only synthetic data in combination with the GG-CNN architecture. The results were robust to sources of interference and offer a promising approach to enable further research in robotic grasping in a highly dynamic environment.

In further research, the developed pipeline can be used to generate grasp data for a wide variety of objects and geometric shapes, to test the dataset with different types of machine learning algorithms.

Acknowledgments. Research was founded by the German Federal Ministry for Econoic Affairs and Climate Action under the program LuFo VI-1 WLIBoro.

References

1. Kehl, W., et al.: SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1530–1538 (2017)
2. Kleeberger, K., et al.: Automatic Grasp Pose Generation for Parallel Jaw Grippers (2021)
3. Mahler, J., et al.: Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics (2017)
4. Periyasamy, A.S., Schwarz, M., Behnke, S.: SynPick: a dataset for dynamic bin picking scene understanding. In: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pp. 488–493 (2021)
5. Schoepflin, D., et al.: Synthetic training data generation for visual object identification on load carriers. Procedia CIRP **104**, 1257–1262 (2021)
6. Sahbani, A., El-Khoury, S.: Bidaud P An overview of 3D object grasp synthesis algorithms. Robot. Auton. Syst. **60**, 326–336 (2012)
7. Li, J.-W., Liu, H., Cai, H.-G.: On computing three-finger force-closure grasps of 2-D and 3-D objects. IEEE Trans. Robot. Autom. **19**, 155–161 (2003)
8. Ponce, J., et al.: On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In: 1993 Proceedings IEEE International Conference on Robotics and Automation, vol. 2, pp. 821–827 (1993)
9. Kleeberger, K., Bormann, R., Kraus, W., Huber, M.F.: A survey on learning-based robotic grasping. Current Robot. Reports **1**(4), 239–249 (2020). https://doi.org/10.1007/s43154-020-00021-6
10. Morrison, D., Corke, P., Leitner, J.: Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach (2018)
11. Pinto, L., Gupta, A.: Supersizing self-supervision: learning to grasp from 50 K tries and 700 robot hours. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3406–3413 (2016)
12. Miller, A.T., Allen, P.K.: Graspit! A versatile simulator for robotic grasping. IEEE Robot. Autom. Mag. **11**, 110–122 (2004)
13. Kleeberger, K., Landgraf, C., Huber, M.F.: Large-scale 6D Object Pose Estimation Dataset for Industrial Bin-Picking (2019)
14. Mahler, J., et al.: Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a Multi-Armed Bandit model with correlated rewards. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1957–1964 (2016)
15. Miller, A.T., et al.: Automatic grasp planning using shape primitives. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 2, 1824–1829 (2003)
16. Vahrenkamp, N., et al.: Planning high-quality grasps using mean curvature object skeletons. IEEE Robot. Autom. Lett. **3**, 911–918 (2018)
17. Carlyn Dougherty Robot Graspit! Project (2019). https://github.com/carcamdou/cr_grasper. Accessed 24 Jan 2022