Vision-based Robotic Grasping Adapted to Diverse Physical Properties

KOSHI MAKIHARA

MARCH, 2025

Vision-based Robotic Grasping Adapted to Diverse Physical Properties

A dissertation submitted to

THE GRADUATE SCHOOL OF ENGINEERING SCIENCE

OSAKA UNIVERSITY

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN ENGINEERING

BY

KOSHI MAKIHARA

MARCH, 2025

Abstract

The grasping of general objects by robots plays an important role in many manipulations and is used in a wide range of applications, including industrial settings, logistics warehouses, and service robots. Grasping is necessary as a preliminary step to enable the appropriate execution of downstream processes such as transport tasks, assembly, and tool manipulation; it is essential to grasp objects while satisfying the object's posture and operating conditions. Some research has been conducted to date, focusing on stable grasping methods, including analyses considering both static and dynamic mechanics, as well as data-driven approaches based on these. Analysis-based methods can guarantee stable grasping through rigorous calculations that consider the physical properties of objects, such as shape, hardness, and friction. However, adapting to a wider range of objects and achieving appropriate grasping in complex scenes, such as random picking, is challenging due to factors such as the cost of calculations and sensing limitations. In addition to using sensing to solve these problems, it is necessary to have an approach that can adapt quickly.

A data-driven approach is used to tackle this problem, and it is possible to adapt to various situations using data such as the success or failure of grasping without the need for rigorous analysis. However, there are cases where grasping fails due to insufficient consideration of the physical properties of the object, and there are instances where the object is damaged or an inappropriate grasping strategy is used. There is a possibility of solving this problem by securing data with many variations, but the cost of this is enormous. Therefore, it is necessary to generate data and learn while reducing the costs of using data and explicitly considering the physical properties to which you want to adapt, as well as being able to adapt to this diversity.

The main contributions of this thesis are as follows: First, to account for the softness of objects, a framework for estimating their characteristics from visual information is applied to object grasping. Second, to account for the shape of objects, a 3D model with a complex shape that can efficiently learn grasping is generated and applied to the grasping model. The proposed method enables fast application to unknown objects

while reducing the cost of data generation through a virtual-emotion learning method. Finally, by designing a learning model that simultaneously considers shape and softness characteristics while using a framework that addresses both factors, it is possible to achieve effective grasping strategies for many types of objects and complex scene settings such as random picking. The effectiveness of these methods was verified in both simulation and real-world environments for vision-based object grasping.

Contents

List of Figures

List of Tables

Abbreviations

GWS Grasp Wrench Space

TWS Task Wrench Space

FEM Finite Element Method

6DoF Six Degrees of Freedom

ToF Time-of-Flight

GAN Generative Adversarial Network

CNN convolutional neural network

DoF degrees of freedom

FDSL Formula-driven supervised learning

RMSE root mean square error

SSIM structural similarity index measure

IFS Iterated Function System

RPN region proposal network

FPN feature pyramid network

FCN Fully Convolutional Network

Chapter 1

Introduction

1.1 Background and motivation

In the last 20 years, as a result of the development of various services in line with major changes in consumer needs and population growth, there has been an acceleration in demand. This has led to a labor shortage [1]. In addition, as competition intensifies, there is a need for lower costs and greater efficiency at work while also taking into account the impact on the environment and safety. There is ongoing debate about the need for automation using robots and other technologies [2].

In industrial settings, such as the automotive industry, some dangerous or physically demanding tasks, like welding and transporting large objects, are partially automated and often introduced into the workplace through system integration that repeats a set of tasks. On the other hand, automation that is not limited to a specific situation but can also cope with environments or the state of the object being handled changing is a difficult task. Nevertheless, there have been developments, such as the introduction of systems that use robot manipulators, hands, and sensors to perform recognition and manipulation. This includes automating the picking process for single parts [3]. However, particularly in logistics, there is a demand for systems that can handle a wide variety of objects and situations, such as product picking. There is also a great need for

object manipulation technology that works in conjunction with a set of sensors. Many picking methods that integrate image recognition [4], particularly in competitions such as the Amazon Picking Challenge [5], have been proposed. However, the issue of stable grasping remained. There is some research focused on methods that emphasize stable grasping, which includes data-driven approaches and analyses that consider static and dynamic mechanics [6, 7]. Analysis-based methods can guarantee stable grasping through rigorous calculations that consider the physical properties of objects, such as shape, hardness, and friction. Based on this, stable grasping can be achieved when the object is limited, with certain shape and physical properties given. However, it is difficult to achieve proper grasping in complex scenes, like random stacking, due to computation cost and sensing limitations. Besides using sensing to solve these problems, an approach that can be quickly adapted is also necessary.

Data-driven approaches are being used to tackle this problem, and it is possible to adapt to various situations using data such as the success or failure of grasping, without the need for rigorous analysis. Some methods are proposed for collecting a large amount of robot experience in real environments [8, 9], and there are also for achieving high-precision picking while reducing costs, such as automatically generating grasping data for a wide variety of products using simulations [10, 11]. While it is possible to solve this problem by collecting a large amount of data, there are cases where the physical properties of the object cannot be fully taken into account due to a lack of information for grasping, and the object may be damaged or an inappropriate grasping strategy may be used. There is also the possibility of solving this problem by securing data with many variations, but the cost of this is enormous. Therefore, it is necessary to generate data and learn in a way that can adapt to the diversity of the physical properties you want to accommodate while keeping the cost of using the data down.

The work presented in this dissertation addresses the following challenges: generating a dataset that can adapt to the diversity of object shapes and deformability. By utilizing this data, the method can achieve visual grasping in complex scenes such as random piles and single-object scenes.

1.2 Objectives

The general objectives of this dissertation are as follows;

1. By generating a database of object deformability and shape, it is possible to achieve safe grasping of objects with various physical properties. Using visual information and a virtual experience-based learning method, this can be applied to unknown objects in the real environment at high speed while reducing the cost of data generation.

2. By designing a learning model that takes into account both shape and deformability characteristics, it is possible to reduce damage to objects and increase the success rate of picking up items with unknown physical characteristics that were not present during training. This is achieved by using effective grasping force in complex scene settings, such as picking up various types of objects or retrieving items from a random pile.

1.3 Dissertation outline

This dissertation is organized as follows.

In Chapter 3, a grasp pose detection method for unknown deformable objects is presented, based on visual information. The model generates a stiffness map that indicates the object's stiffness for each pixel in an image using generative adversarial networks (GAN) for pix2stiffness estimation and grasp pose detection, which adapts the stiffness map to minimize the object's deformation and avoid any potential damage. The framework can plan how to grasp an object using a few 3D models of objects.

In Chapter 4, using 3D mesh models generated from a single formula based on fractal geometry, we propose a database for robotic picking. We construct a database of paired images and grasp performance values generated from the 3D models based on Dex-net's generation rules and train a GQ-CNN. In single-object pick-up experiments,

we compared the performance with the case of using digitized 3D models. The time required to generate 3D models was also computed.

Chapter 5 is a method for estimating simultaneous deformability and instance segmentation from depth images, as well as enhancing grasp pose detection for deformable objects under specific conditions using the outcomes of our deformability assessments. The efficacy is verified in a variety of challenging settings involving cluttered scenes with various deformable objects.

Finally, in Chapter 6, the achievements and limitations of the proposed methods presented in this dissertation are discussed, as well as open challenges and ideas for future work.

Chapter 2

Literature Review

2.1 Physics-based grasp quality evaluation

Grasp quality is often defined by considering some properties of rigid objects, such as disturbance resistance and stability, etc. [6]. These metrics usually adopt grasping force and torque when analyzing grasp candidates, such as grasp wrench space (GWS) [7], and the expanded method for task completion, known as task wrench space (TWS) [7]. Using these measures for non-rigid objects is difficult because they must consider the deformation generated by the grasping wrench. Moreover, various methods have been proposed to evaluate grasp quality by examining the object's deformation [12, 13].

In addition, the grasp quality evaluation is successful for grasping, deformability, and preventing damage. Xu et al. [14] proposed quality metrics that consider task completion of deformable objects, including liquids. Using an elastic 3D model, the grasp quality is defined as the minimal grasping wrench, which reduces resistance according to Hooke's law. Also, there are some metrics considering contact dynamics [15] based on the Finite Element Method (FEM) that can simulate grasping in a more realistic way than static analytical methods. However, these methods are difficult to apply to unknown objects (re-calculation is needed) and to objects with material properties like nonuniform stiffness (FEM assumes homogeneity).

2.2 Grasp pose detection from vision

Several approaches have introduced 6DoF pose estimation methods based on 3D models of objects for applications in assembly and parts-supply processes in manufacturing [16, 17]. Deep learning-based methods are highly generalizable to multi-species objects and are suitable for applications in domains where various items are handled, such as logistics warehouses.

In addition, owing to the evolution of object recognition using deep learning, many research have presented methods for efficiently grasping several types of objects [9,10,18, 19]. For instance, Levine et al. [9], established a relationship between robot commands (motion vectors) and the grasp success rate observed before and after picking an object using a deep learning model. This model was trained using $800,000$ grasps performed by real robots and could generalize to grasp several tbjects. This method sequentially determines robot actions in a closed-loop system using a reinforcement learning framework [20] and includes approaches that involve learning from human demonstrations [21]. Alternatively, an approach that generates the grasp pose without prior scene information has been proposed. Although learning a robot's behavior in complex scenarios, such as environments containing piles of objects, is technically feasible, the associated costs are prohibitive. Consequently, an approach that focuses exclusively on grasp pose generation has proven to be more effective.

In simulation-based approaches, several datasets that utilize a diverse array of three-dimensional models have been proposed. These models are employed to analytically determine the posture of a robotic hand capable of achieving a stable grasp, thereby facilitating the automatic generation of various scenes [10, 22]. Mahler et al. [10] computed suitable grasps for the three-dimensional models of specific objects in advance, evaluated the similarity of the object (in appearance) to those previously calculated, and determined the grasp pose using one of the most similar objects. Furthermore, adopting simulation methods extending to six degrees of freedom (DoF) has facilitated object picking in complex scenes [23]. These methods do not require manual annotation, allowing the use of large datasets of clean and accurately generated data based on numerical

evaluations of the grasping performance. However, bridging the gap between real and simulated environments in terms of noise and physical discrepancies remains challenging. Approaches such as configuring the real environment to mirror the simulated conditions and employing domain randomization [24] are necessary to address these issues.

For applications involving real-world data, the use of manually or automatically annotated data for selecting corresponding to a specific scene has been proposed. Benchmarks that derive grasp positions for a single object from RGB images [8] demonstrate that accurate picking is achievable [25]. For complex scenarios, such as piled objects, datasets with a large number of candidate grasp points have been suggested to be derived from point clouds [26,27]. One approach processes related tasks in parallel, such as instance segmentation and collision detection [28], while the other approach is tailored for small-scale environments and objects characterized by limited data availability [29]. These methods were designed to enhance the adaptability and efficiency of handling intricate tasks simultaneously. Domae et al. [18] convoluted a 2D model of a robotic hand with the depth image of an object to estimate a grasp pose that does not result in a collision with other objects and is close to the center of gravity of the object. Several small parts were successfully selected.

All of these methods can be used to grasp various objects, regardless of their type or shape. However, as mentioned previously, in the case of deformable objects, grasping can fail, and the object can be squashed and/or damaged due to its inherent deformability. However, these methods do not provide intrinsic solutions for deformable objects.

In recent approaches to grasp pose detection, the focus has shifted to objects that had not been previously handled. Matsumura et al. [30] proposed a deep learning model to establish the relationship between a specific grasp pose and the likelihood of creating a tangle with potentially complex-shaped objects in a bin. Sajjan et al. [31] proposed a deep learning model for estimating the depth maps of transparent objects. Transparent objects can be selected by combining this model with a grasp-pose detection method. Objects that are difficult to recognize and manipulate optically and/or physically continue to pose an important problem.

2.3 Physical property estimation

Several approaches have focused on estimating the stiffness and other physical properties of objects. For example, Lu et al. [32] proposed a method for estimating the damage conditions of fruits based on the amount of water in hyperspectral images. Fujiwara et al. [33] proposed a method to estimate the stiffness of an object based on the deformation generated on its surface by applying ultrasonic waves. In both of these research, a relatively specialized sensor was necessary; hence, these methods were difficult to implement. In addition, Tanaka et al. [34] estimated the material composition of an object by measuring different distortions on its surface generated by applying light of different cycles using a time-of-flight sensor (ToF). Meka et al. [35] estimated the material composition of an object by extracting its diffuse and specular reflection components using a deep learning model.

As mentioned previously, the necessary information for estimating the physical properties of materials related to stiffness is included in the images of the object and/or range imagery. Under certain conditions, it is possible to estimate the stiffness. The stiffness map was proposed by Xu et al. [14], who mapped the surface of a 3D model with a stiffness score measured based on Hooke's law to analyze the quality of the sampled grasp poses while considering the stiffness of the object. However, this method can only be adopted for objects with fully known 3D models and not for unknown objects. Adapting to unknown objects requires visual sensors to determine their texture and shape. However, values based on physical deformation measurements, such as the stiffness of a material, are difficult to estimate from images alone; therefore, only approximate values of deformability are used.

2.4 Grasp datasets

There have been many proposals for datasets using deep learning to input images and achieve grasping, and methods have been developed using real environments or simulations to provide correct labels for grasping positions and grasping performance to input

into models specialized for grasping. In real-world datasets, labels are often provided manually or analytically [8,9], and while they are accurate, the cost of constructing them is high. On the other hand, a major advantage of methods using simulations is that they can automatically generate a large number of variations [11,23,36]. It is difficult to generate labels and images for grasping that match real-world phenomena, but the large number of data variations makes it possible to effectively achieve learning for grasping.

When using simulations, 3D models are required, and there are many proposed scanned 3D models, such as the YCB dataset [37], ShapeNet [38], and Objverse [39], and the variations are quite large. On the other hand, the time and effort required for 3D scanning must be considered. In response to this, there are also several initiatives to automatically generate 3D mesh models, such as Procedual [40], which generates data by randomly combining patterns from primitive shapes, and EGAD! [41], which enables automatic generation based on the evolution of biological shape patterns. However, Procedual uses some information from ShapeNet, and EGAD! has many parameters that need to be set in advance, and the dataset is specialized for hand shapes, so its range of application is limited.

2.5 Deep Learning models

Deep learning can achieve high performance in estimating targeted tasks by generating and learning from training data. It is also applicable to a variety of tasks beyond object recognition, including image generation. A generative adversarial network (GAN) [42] can be considered a type of image generation model, so methods based on a GAN or other simpler image-generating models can be utilized. The pix2pix [43] improved image generation accuracy by learning the relationship between a pair of images (original and conditional). The pix2pixHD [44] has also improved pix2pix to adapt to high-resolution images. Additional external methodologies include ASAPNet [45], which accelerates the conversion of high-resolution images, and CoCosNet-v2 [46], which enhances accuracy through a multistage conversion process at various resolutions. pix2stiffness [47] proposed an image translation from a depth image to a deformability map. Training data

were obtained as paired images from simulations.

Moreover, in bin-picking scenes where several objects are mixed, each object must be segmented. He et al. [48] proposed a Mask R-CNN, which can perform instance segmentation. In addition, this method can multitask, allowing for complementary improvements in task recognition. In the segmentation models, a Cascade Mask R-CNN was proposed by modifying the model architecture, and the vision transformer architecture (ViT) [49] was used as the backbone by scaling the data and model size. For example, Mask2Former [50], Mask DINO [51], and EVA [52] have successfully obtained strong features for segmentation tasks by learning the visual representation focused on the mask of the object. InternImage [53] adapts an existing CNN for training large-scale data.

2.6 Formula-driven supervised learning

Formula-driven supervised learning (FDSL) is one of the most promising concepts that can solve the problem of dataset construction costs, and it has been applied to various tasks, including not only image recognition [54] but also video recognition [55], multi-view recognition [56], 3D object detection [57], and multi-modal recognition [58]. In addition to fractals, the FDSL framework is also used to design circular harmonics [59] that change by focusing on the contours of objects, and it is also used in datasets consisting of primitive shapes for medical image segmentation [60].

FDSL reduces the time required to construct datasets by automatically generating data and teacher labels based on mathematical formulas, eliminating the need for human intervention. The datasets proposed in the FDSL framework have been used for pre-training in each recognition task and have achieved performance equivalent to that of pre-training on real image datasets despite not using real images.

2.7 Foundation model in Robotics

Increasing the amount of data, computational power, and model size can improve various tasks in natural language processing and image recognition [61]. Consequently, methods that leverage large-scale data and resources—often referred to as base models have emerged primarily in the field of language, leading to the proposal of numerous large language models (LLM). In addition to closed-source models such as GPT [62] and Gemini [63], various open-source models like LLaMA [64] and models specialized for specific languages have been developed for different use cases.

In the field of image processing, there are models that handle images exclusively (e.g. ViT-22B [65]), as well as numerous Vision-and-Language Models (VLM), including GPT-4V [62], Gemma [66], and LLaVA [67], that accept both text and images as inputs and can solve many language processing and image recognition tasks with high performance. For robotics applications, knowledge of LLM and VLMs is increasingly being utilized in robotic manipulation [68]. Efforts are also being made to collect and fine-tune new types of knowledge, such as physical knowledge [69, 70]. However, their application remains largely limited to high-level planning, like scenario planning, while essential data for grasping and manipulation still needs to be supplemented [71]. In addition, End-to-End action models such as RT-1 [72] and RT-2 [73], which are based on imitation learning, have been proposed and are referred to as Vision-Language-Action models (VLA). VLA are input language instructions and images from the robot's perspective as input and output for the subsequent action. Although these datasets are smaller than those for language or image tasks, it has been reported that performance on robotic tasks improves when pre-trained on millions of episodes (RT-X [74]).

RT-X [74] and RH20T [75] are proposed as large-scale datasets, but training with all of them simultaneously does not necessarily improve performance in robot tasks. A key issue is that, although large amounts of data are collected in specific environments, these data often lack diversity. Hence, methods to ensure diversity—such as carefully sampling the data used for pre-training are required [76, 77].

There are also reports that collecting approximately 10,000 hours of behavioral data improves performance in numerous tasks [78], yet such large-scale data collection incurs substantial costs. Because models relying on language, image, and behavioral information require vast datasets, it is critical to develop efficient methods for data collection and utilization, explicitly considering the physical properties of the environment and target objects. Moreover, minimizing the cost of real-world data collection and usage by leveraging data that are difficult to obtain in actual environments remains an important objective.

Chapter 3

Grasp pose detection for deformable daily items by pix2stiffness estimation

This thesis chapter originally appeared in the literature as

Makihara, K., Domae, Y., Ramirez-Alpizar, I. G., Ueshiba, T., and Harada, K. (2022). Grasp pose detection for deformable daily items by pix2stiffness estimation. Advanced Robotics, 36(12), 600–610. [1]

3.1 Introduction

Recently, robots are expected to work in household environments, where there are several objects with different shapes, materials, mass, and other properties. In particular, robots have to grasp several types of deformable objects, such as paper boxes containing snacks. Although these objects are easy for a human to grasp, they are very difficult for robots that need to search for a grasp pose and control each finger's force. In most robotic grasping research, the main focus has been placed on rigid objects, where the problem can be simplified by assuming a point contact model [79]. These methods exhibit optimal performance for several types of shapes, sizes, and other complexities. However, it

[1] https://doi.org/10.1080/01691864.2022.2078669.

remains difficult to successfully grasp deformable objects because these rigid-body-based approaches do not consider the object's deformation.

Recent studies have proposed grasping methods for deformable objects, such as analyzing grasp quality, considering the surface deformation by a contact wrench [14, 80], and effectively controlling a grasp wrench by sensing the contact wrench [6]. Additionally, some simulation-based methods that compute deformation using physics engines and then apply the grasp in the real world [15, 81, 82] have been proposed. In these cases, they assumed that the grasping force is controllable by an electric unit and can grasp various deformable objects using some force and/or tactile sensor. However, in most cases of industrial applications, a constant force is used to grasp objects (e.g., pneumatic gripper). Also, sometimes it is impossible to control the grasp of slippery objects, even if the slip is detected. Therefore, there is a need to consider stiffness to avoid damaging the object without force control, which can be accomplished by considering pre-grasping motion before any contact occurs. In addition, many objects have inhomogeneous stiffness, such as daily items, or are unknown, making it difficult, even though it might be possible, to apply force control. We propose a grasp pose detection method for unknown deformable objects using an image as input. This method comprises two parts: (1) stiffness estimation, which generates a "stiffness map" that indicates the object's stiffness for each pixel in an image using generative adversarial networks (GAN) [42] as an image translation method, and (2) grasp pose detection, which generates a grasp pose, thereby avoiding damage to the object from the robot's gripper, using the stiffness map and executing the grasping motion. The overview of the proposed method is shown in Figure 3.1. Our contributions are as follows:

1. Our proposed pix2stiffness method can convert the image of objects to a map of the stiffness score for each pixel by adapting the pix2pix [43]. The image translation can be performed by training semi-automatically generated images using a physics simulator.

2. By combining the obtained stiffness map with the grasp pose detection method, we can detect a grasp pose that can prevent damages to unknown (same category

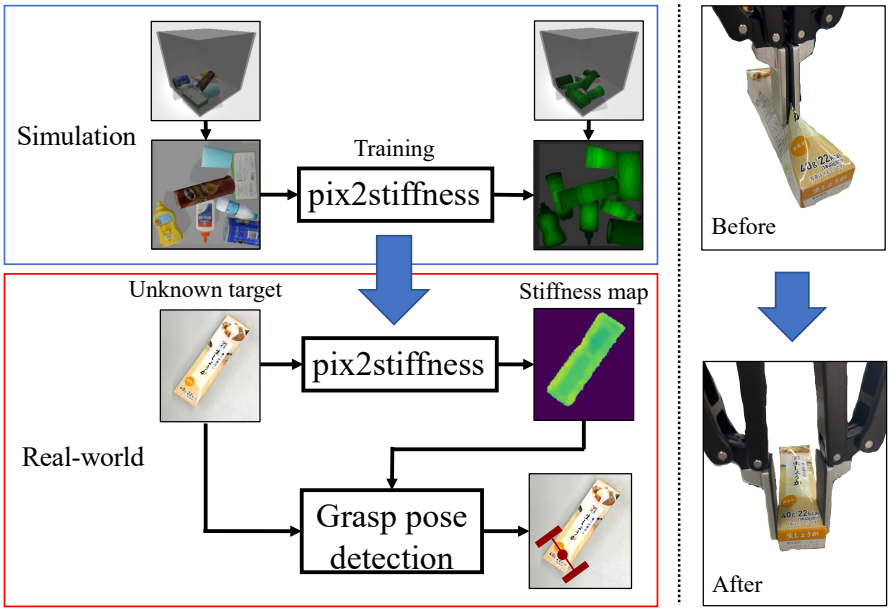


Figure 3.1: Overview of the proposed grasp pose detection method via stiffness estimation: we adopt an image as the input, and utilize it for image translation by pix2stiffness. After image translation, a stiffness map that indicates the object's stiffness score for each pixel is generated. Finally, grasp pose detection is executed using the map for the case of a 2-finger gripper (the red lines represent the grasp candidate).

of a bottle or a box, but has different shape and size) deformable objects with fewer 3D object models used adopted in training the GAN.

This paper is organized as follows. First, we review related works in section 2. Next, we comprehensively present an overview of the proposed method (pix2stiffness and grasp pose detection) in section 3. In section 4, we evaluate these method in simulation. In section 5, we describe the experimental setup and compare the real-world results with the simulation results obtained in section 4. Finally, we conclude this study in section 6.

3.2 Proposed method

In this section, we introduce the proposed methods for stiffness estimation and grasp pose detection. To estimate the object's stiffness, a stiffness map is constructed, which indicates a score of the stiffness for each pixel in an image using image translation with GAN. Using this map, a grasp point is detected for the object to be grasped.

3.2.1 Stiffness map generation (pix2stiffness)

In this study, we employ pix2pix [43] network architecture to generate a stiffness map. We solely adopt one image as input; however, the representation of an object's stiffness depends not only on its texture, but also on its shape, material, and other physical properties. Therefore, other information is also required as input. If we consider them as conditions when using cGAN, then there is no need to train a new network, as we would simply need to adjust the inputs. For executing pix2stiffness, we employ the pix2pix architecture because the pair of images required for translation (for the tasks considered in this dissertation) can be generated via simulations.

Data Collection

To train the pix2pix network, we need pairs of before and after translation images. To prepare the stiffness map, the annotation of a stiffness score for each pixel is required, and the cost of doing it manually is high. In addition, because stiffness can change in some parts of the object, it is necessary to prepare stiffness maps for various poses of each object, which further increases the cost of doing it manually. To address these problems, we propose a method that semi-automatically generates synthetic data via simulations. We adopt the 3D object model with its texture attached, as well as the Blender physics engine [83]; accordingly, data is prepared as follows:

I. Coloured stiffness map:

   For each 3D model, we decide to divide a green color gradation into 10 tones, as

a guideline for understanding the effect of damage triggered by grasping, and for representing the stiffness score of each object's surface. Accordingly, this approach generates a 3D stiffness map as a texture attached to the object's surface (Figure 3.2a). The stiffness scores are based on manual measurements by a hardness meter.

II. Execute simulation:
After preparing a white bin (Figure 3.2b), a simulation that involves dropping objects with random positions and postures from above the bin is computed by Blender.

III. Capture images:
We took images from the top of the bin when using the original texture of the object (Figure 3.2c) and also when using the stiffness map texture (Figure 3.2d). Accordingly, we obtained a pair of images. In the stiffness map, because the element value of green indicates the stiffness score for each position, we convert the map from 3-channel to 1-channel (green). And we use this map for training.

We semi-automatically generated the training data by repeating the above steps. Because we generated a clutter scene, the synthetic data exhibited various scenes with randomized object poses. The green tone solely represents the stiffness score of the created stiffness map.

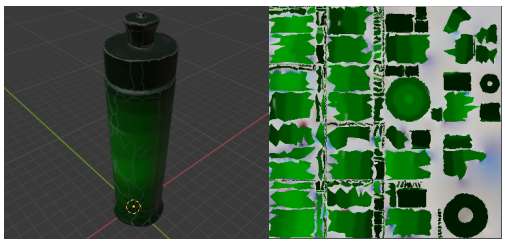
GAN Training

We employ the pix2pix architecture for pix2stiffness translation. The objective adversarial loss is defined by pix2pix [43]:

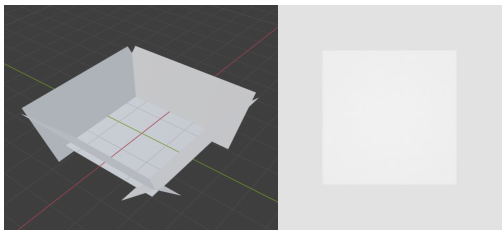
$$L_{GAN} = \mathbb{E}_s[log(D(x,s))] + \mathbb{E}_x[log(1 - D(x, G(x)))] \,, \tag{3.1}$$

where G is trained to minimize this objective and D is trained vice versa. x and s indicate the input (RGB or Depth) and stiffness map images, respectively. In addition, the loss function is also based on the $L1$ distance to obtain a generated image $G(x)$ closer to the ground truth s:

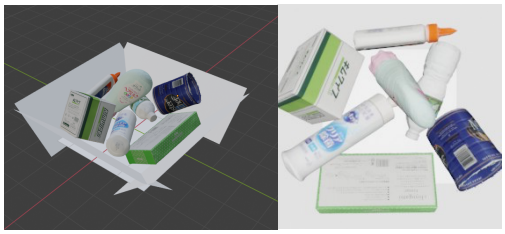
$$L_{L1} = \mathbb{E}_{x,s}[||s - G(x)||_1] \,, \tag{3.2}$$



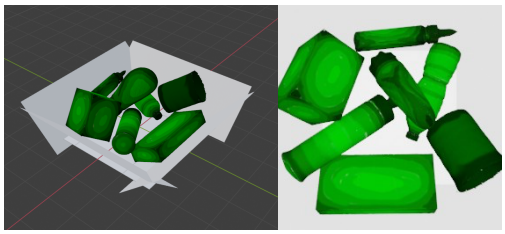
(a) Coloured stiffness map and map image of 3D object model



(b) Initial state and background image



(c) Dropped state and original image



(d) Dropped state and stiffness map image (same state as Figure 3.2c)

Figure 3.2: Data collection using a physics simulator

Our problem is

$$G^* = \arg\min_G \max_D L_{GAN} + \lambda L_{L1} \ . \tag{3.3}$$

The image input and stiffness map output have a size of 256×256, the generator has a U-Nets structure [84], which has a seven-layer encoder and a seven-layer decoder with skip-connection and dropout for all layers. The discriminator value is calculated using PatchGAN, which judges True/False for each small region of an image (Figure 3.3). To train the network, we can use an arbitrary number of pair of image-stiffness map images (described in the previous section).

3.2.2 Grasp pose detection using stiffness map

In this section, we describe the grasp pose detection method using the stiffness map generated by pix2stiffness. This stiffness map can be easily applied to a method that indicates a grasp score for each pixel in an image. In this study, we propose a 4-DoF

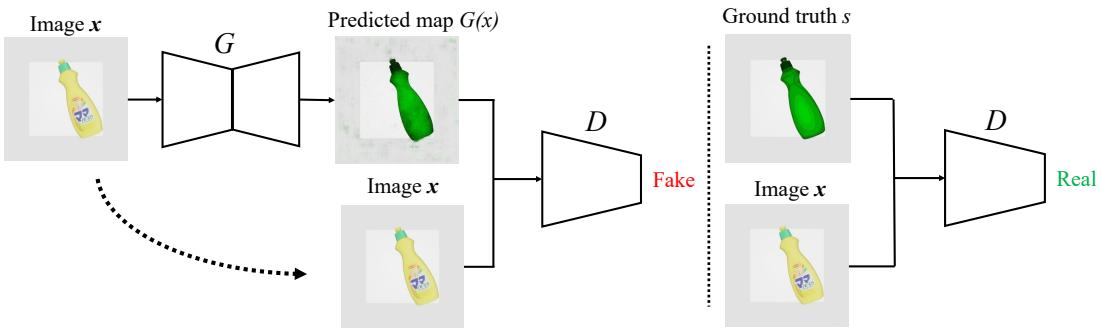


Figure 3.3: Image translation network architecture of pix2stiffness referenced by pix2pix

grasp pose detection using a stiffness map and a depth image constrained to a grasping pose vertical to a plane located in the target objects. The proposed method adopts the stiffness score as the grasp quality score for each grasp candidate in an image.

Overview of the FGE [18]

The FGE is a method that detects a 4-DoF grasping pose using a single depth image. Using these depth and template images, FGE calculates contact and collision regions of the hand and a target object, then it computes a non-collision region that represents grasp pose candidates. Subsequently, a graspability map that indicates the points that are closer to the object's center of mass is generated by convoluting a Gaussian filter with the non-collision region. The optimal grasping pose is detected as the position with the highest graspability value. The contact $\mathbf{T}_t$ and collision $\mathbf{T}_c$ templates are predefined, while the contact $\mathbf{I}_t$ and collision $\mathbf{I}_c$ images are obtained from a single depth image.

Then, the contact region $\mathbf{A}_t$ can be calculated by convoluting $\mathbf{T}_t$ with $\mathbf{I}_t$:

$$\mathbf{A}_t = \mathbf{T}_t \otimes \mathbf{I}_t \ , \tag{3.4}$$

$\otimes$ denotes the convolution. The collision region $\mathbf{A}_c$ can be calculated by convoluting $\mathbf{T}_c$ with $\mathbf{I}_c$:

$$\mathbf{A}_c = \mathbf{T}_c \otimes \mathbf{I}_c \ . \tag{3.5}$$

To compute the non-collision region and the graspability score of each pixel, the graspability map is calculated as the region where the gripper does not interfere with the

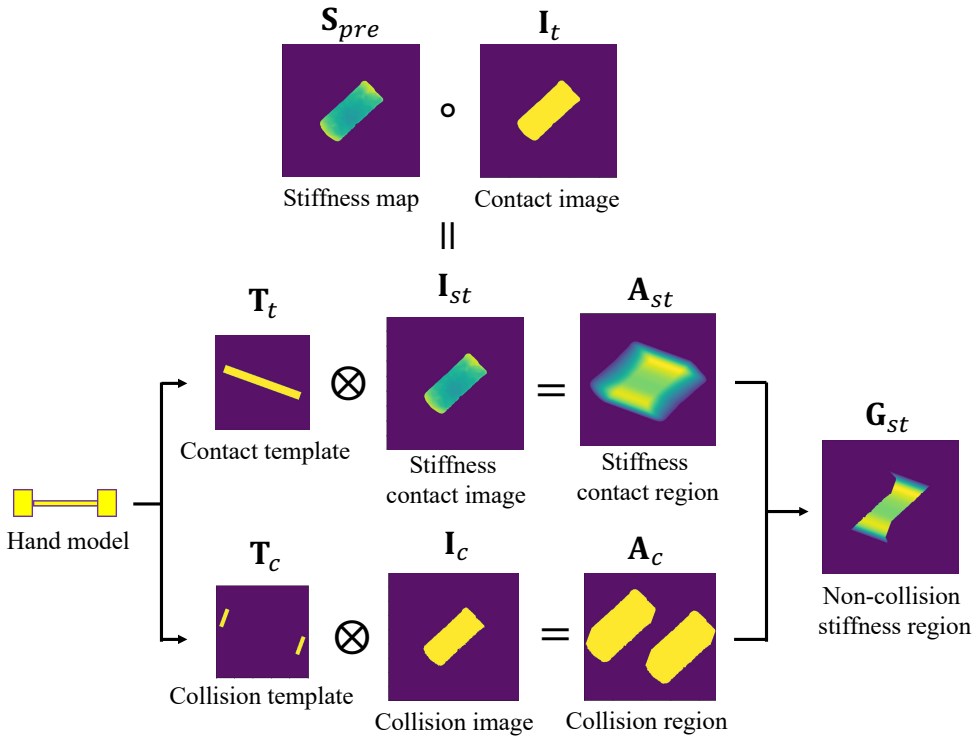


Figure 3.4: Processing pipeline of the grasp pose detection method

surrounding area near the center of gravity of the object region (each image can be seen in Figure 3.4).

Finally, the graspability map is calculated for each angle of the hand model, and the optimal grasp pose is the point with the highest graspability score.

Grasp pose detection using stiffness map

FGE can select the grasp pose nearest to the position of the object's center of mass; however, for deformable objects, it may cause a large deformation that triggers permanent damages. By using a stiffness map, we can detect a grasp pose that addresses these problems.

At first, a stiffness map $\mathbf{S}_{pre}$ that indicates $G(x)$ for each pixel is generated by pix2stiffness, normalization and some pre-processing are used. In $\mathbf{S}_{pre}$, a larger value

denotes that it is more difficult to deform. When a grasp pose with a high score is detected, this implies that it is possible to grasp and prevent damage simultaneously. By using this stiffness map, we proposed a modified FGE that fits the objectives of this study.

Secondly, the stiffness contact image $\mathbf{I}_{st}$ is calculated by multiplying the generated stiffness map $\mathbf{S}_{pre}$ and the contact image $\mathbf{I}_t$:

$$\mathbf{I}_{st} = \mathbf{S}_{pre} \circ \mathbf{I}_t \ , \tag{3.6}$$

$\circ$ denotes the Hadamard product. Then, it is convoluted with the contact template ($\mathbf{I}_t$ is replaced with $\mathbf{I}_{st}$). Subsequently, the stiffness contact region $\mathbf{A}_{st}$ is generated;

$$\mathbf{A}_{st} = \mathbf{T}_t \otimes \mathbf{I}_{st} \ , \tag{3.7}$$

$\mathbf{A}_{st}$ represents the average stiffness score of each pixel in the rectangular region surrounded by the 2-finger gripper (in the contact template $\mathbf{T}_t$). Via this convolution, this stiffness score differs from the original one, and the score of the locations near the object's silhouette is slightly lower than those closer to its center. Additionally, the contact region $\mathbf{A}_t$ and the collision region $\mathbf{A}_c$ are also generated in the same way as FGE.

The grasp candidates which are collision free, are obtained using a logical AND operation between $\mathbf{A}_t$ and $\overline{\mathbf{A}_c}$, thus, the non-collision stiffness region $\mathbf{G}_{st}$ (similar to the graspability map) is generated as:

$$\mathbf{G}_{st} = \mathbf{A}_{st} \circ (\mathbf{A}_t \cap \overline{\mathbf{A}_c}) \ , \tag{3.8}$$

where $G_{st}(h, w)$ denotes the element value of the position (h, w) in $\mathbf{G}_{st}$. The objective function is defined as:

$$f(h, w, \theta) = \begin{cases} G_{st}(h, w) & if \ \ A_c(h, w) = 0 \\ 0 & otherwise \end{cases} \ , \tag{3.9}$$

where θ denotes the rotation angle of the detected grasp candidate, and $A_c(h, w)$ denotes the element value of the position (h, w) in $\mathbf{A}_c$. The calculated coordinate index

is expressed as:

$$[H, W, \Theta] = \arg\max_{h,w,\theta} f(h, w, \theta) \ . \tag{3.10}$$

Here, we only utilize a two-finger gripper's hand template; hence, we can apply Eq. (6) as the objective function (described in Figure 3.4).

3.3 Simulation results

In this section, we evaluate the accuracy of pix2stiffness estimation and the effectiveness of our grasp pose detection method in simulation scenes. For training, we prepared fifteen models of 3D objects in Figure 3.5a and stiffness maps annotated as explained in section 3.2.1. For validation, seven unknown (we define some categories such as bottle and box, then we target objects in the same category but with different shapes) models (Figure 3.5b) are prepared.

(a) Training (b) Validation

Figure 3.5: Dataset of 3D object models used in simulation.

3.3.1 Image quality evaluation

Using the training data presented above, we evaluate the results obtained with different input data types (RGB and Depth). The quantitative evaluation metrics of the adopted pix2stiffness estimation are: 1) root mean square error ($RMSE$) and 2) structural similarity index measure ($SSIM$) [85] between the ground truth stiffness map $\mathbf{S}$

and the predicted map $\hat{\mathbf{S}}$ using pix2stiffness. Especially, $SSIM$ considers changes in brightness, contrast and the entire structure. These metrics are usually used for depth estimation [86]. $RMSE$ is calculated as:

$$RMSE(\mathbf{S}, \hat{\mathbf{S}}) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(s_i - \hat{s}_i)^2}\ , \tag{3.11}$$

where M denotes the number of $\mathbf{S}$ pixels (same as $\hat{\mathbf{S}}$). s_i and $\hat{s}_i$ denote each element i-th value of $\mathbf{S}$ and $\hat{\mathbf{S}}$, respectively. The closer $RMSE$ is to zero, the lower the pixel-wise error is. $SSIM$ is a metric based on appearance, which is computed for each of the evenly divided small areas. This metric can analyze spatial similarity. We adopt the mean of $SSIM$ ($MSSIM$) to evaluate the entire image quality:

$$MSSIM(\mathbf{S}, \hat{\mathbf{S}}) = \frac{1}{N}\sum_{j=1}^{N} SSIM(\mathbf{S}_j, \hat{\mathbf{S}}_j)\ . \tag{3.12}$$

$SSIM(\mathbf{S}_j, \hat{\mathbf{S}}_j)$ is calculated between $\mathbf{S}_j$ and $\hat{\mathbf{S}}_j$ (N is the number of areas, and we use $N = 100$) for each region j. The closer $MSSIM$ is to one, the better the similarity of the entire image. For evaluation, the predicted stiffness map $\hat{\mathbf{S}}$ is preprocessed map $\mathbf{S}_{pre}$. The obtained results are summarized in Table 3.1. It can be observed that higher accuracy is obtained with depth as input.

Table 3.1: Estimation results for different training dataset types

Dataset type	$RMSE$	$MSSIM$
RGB	47.68	0.7250
Depth	**32.91**	**0.8552**

3.3.2 Effectiveness of grasp pose detection

Furthermore, we also evaluate the influence of pix2stiffness on the grasp pose detection by calculating the mean of stiffness score in the rectangular space surrounded by a two finger gripper (grasp region) when using the ground truth stiffness map s. In this evaluation, we adopt the segmentation image from simulation as the contact and collision

images, assuming a picking scene with an object placed on the table. After cropping the grasp region from $\mathbf{I}_{st}$, The mean of stiffness score (the higher the better) is calculated as:

$$R = \sum_{k=1}^{L} \begin{cases} 1 & \text{if } \mathbf{s}_k > 0 \\ 0 & \text{otherwise} \end{cases} , \tag{3.13}$$

$$Mean\ of\ stiffness = \frac{1}{R} \sum_{k=1}^{L} \mathbf{s}_k \ . \tag{3.14}$$

where L denotes the number of pixels in the grasp region, and R is calculated as the size of the object's region. The proposed grasp pose detection method described in section 3.2.2 is adopted in this evaluation, to demonstrate the validity of the proposed method for grasping deformable objects while preventing damage. Using the model of pix2stiffness with the depth image as input, Figure 3.6 presents examples of detected grasp poses in simulations, where the red line represents the detected pose for a two-finger gripper (the blue one is detected by FGE). Table 3.2 presents the evaluation of seven images for each single object when using the proposed method and FGE. The result of each object and the mean of stiffness are relatively higher than FGE's result. In the result of "Bottle 3", the score of FGE is close to the proposed method. The reason is that the lowest stiffness of this object is 0.7 which has overall a high stiffness. In the result in Figure 3.6, the grasp poses were detected far from the center of gravity, therefore it has some possibility of failure to grasp. Because we only verified the possibility to prevent damages by considering grasping to the hard part, more evaluation that the grasp pose can be executed successfully in the real-world is needed.

3.4 Real-world experiments

In this section, we evaluate the predicted stiffness map and detected grasp pose using the map for real images (adopt depth image as input, same as in section 3.3.2). Because real depth images have some noise, missing values, and errors, the estimation networks are trained with only simulation images, and it is inadequate to adopt raw images as the input for pix2stiffness. To address this problem, raw images are preprocessed via fast digital inpainting [87] and contrast emphasis. By using these simple pre-processing

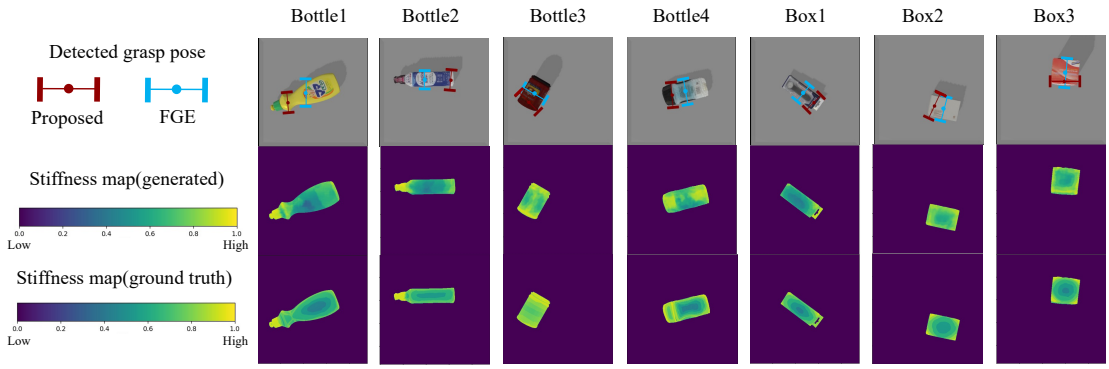


Figure 3.6: Detected grasp pose using the proposed method for seven objects in simulation: in the top row, the red line represents the detected pose for a two-finger gripper. In the middle row, the images represented the stiffness maps generated by pix2stiffness. In the bottom row, the images represent the ground truth stiffness maps generated via simulations.

methods, the stiffness map can be generated more clearly. The target objects in grasping experiments are presented in Figure 3.7. The hardware used in the experiments are a UR5, a Robotiq 2-finger gripper (140mm stroke), and a Realsense SR305 attached to the gripper.

Figure 3.7: Target objects in real experiments that are not included in training data

Table 3.2: Mean of stiffness ($Mean\ of\ stiffness$) for single object in simulations. Each object's name is as described in Figure 3.6

Name	$Mean\ of\ stiffness$	
	FGE [18]	Proposed method
Bottle 1	0.6141	0.7149
Bottle 2	0.6293	0.6896
Bottle 3	0.7710	0.8138
Bottle 4	0.6043	0.7977
Box 1	0.5446	0.7207
Box 2	0.5897	0.7129
Box 3	0.6011	0.7125
Mean	0.6220	**0.7374**

3.4.1 Grasp experiments for single object scene

In this section, we evaluate the effectiveness of the proposed method on real-world images via grasping experiments. The experimental scene is assumed to be a single object placed on a table (same as in simulation). In addition, we manually set the height of the gripper from the table in the proposed 4-DoF grasping pose detection, where the silhouette of the object's region can be almost obtained in the contact/collision image.

Figure 3.8 presents the grasp pose detection results, generated stiffness map, and grasping behavior for each object. It can be observed that the proposed method can grasp the hard part of each of the objects. However, for "Object 1" and "Object 6", the poses of these objects changed during the lifting motion. This result indicates that the proposed method can fail to stably grasp the object, as the grasp pose for most of the objects is detected far from their center of gravity. However, because we adopt the strategy of grasping the hard part of the object, it can be considered to be successful because few deformations are generated by a posture change of the object.

As a quantitative evaluation, we analyze the deformation the object sustains by grasping. To evaluate the mean of stiffness in the simulation results, we adopt the stiffness map of the ground truth; however, it is difficult to prepare the same map for real-world experiments. Therefore, we use the grasping width after performing grasping for the evaluation. First, we manually measure the grasping width at the moment of contact with the object (called l_c), then, we measure the grasping width after grasping with a certain grasping force (called l_g). The grasping force can be determined by using Robotiq's gripper function (10-125 [N]) provided by URCaps [88]. In this experiment, we set the constant grasping force to 62.5 [N] assuming the case of no-force control, this value is the smallest force that can grasp the heaviest object (Object 1 in Figure 3.8) in our experiments. After measuring the two grasping widths (l_c and l_g), we define the deformation rate using the following equation.

$$Deformation\ rate = \frac{l_c - l_g}{l_c} * 100\ [\%]\ . \tag{3.15}$$

Table 3.3 presents the deformation rate by grasping in Figure 3.8 for each object. For most of the results, the deformation rate is lower than FGE, which indicates that the proposed method can prevent the deformation of the object. However, in the result for "Object 2", the deformation rate is higher than FGE. The reason is that the grasping force is applied to a narrower surface than FGE's result because the finger surface was slightly inclined to the object's surface. By addressing this problem, it is expected that the object's deformation can be prevented.

3.4.2 Grasp experiments for clutter scene

Similarly to section 3.4.1, we evaluate the deformation rate and success rate in a cluttered scene, and compare it with FGE. The robot repeats the trial until all eight objects are grasped successfully in the given cluttered scene. Although it is necessary to determine the grasping height in the proposed method, it is not trivial in a cluttered scene. In this experiment, the candidate grasping height is set at regular intervals (each of 10 [mm]), and the grasp pose is searched from the height where there is a certain amount of one object area in the contact image to a height five steps lower. This method is also

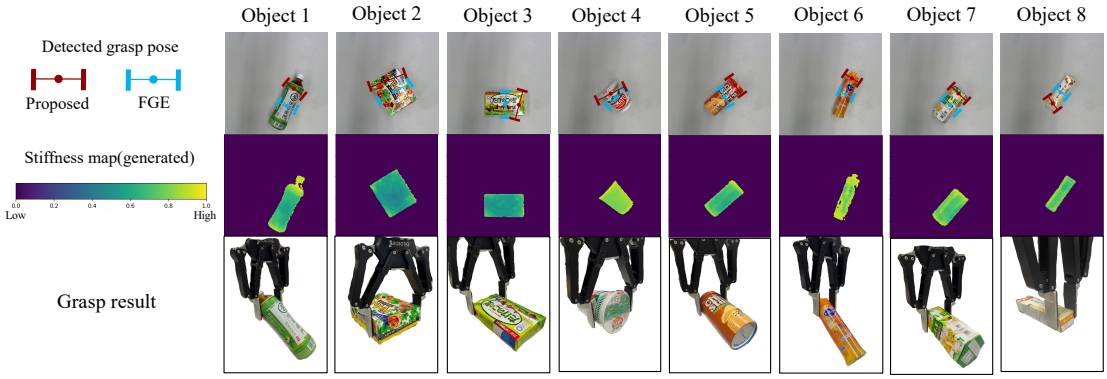


Figure 3.8: Detected grasp pose using the proposed method for eight objects in real-world: in the top row images, the red line represents the detected pose for a two-finger gripper. In the middle row, the stiffness maps generated by pix2stiffness are presented. In the bottom row, results obtained for grasping and lifting a single object placed on a table are presented.

Table 3.3: Deformation rate results for single objects in real-world

Name	*Deformation rate*	
	FGE [18]	Proposed method
Object 1	25.13	12.44
Object 2	7.913	9.428
Object 3	14.25	9.757
Object 4	13.74	7.610
Object 5	13.69	8.885
Object 6	18.18	10.91
Object 7	18.33	14.47
Object 8	66.14	21.64
Mean	22.17	**11.89**

applied to the FGE, and the grasp pose with the highest graspability score is selected. As explained in section 3.2.2, the score map $\mathbf{G}_{st}$ in our proposed method does not represent the original stiffness score; hence, it is difficult to select a relatively high score in all grasp

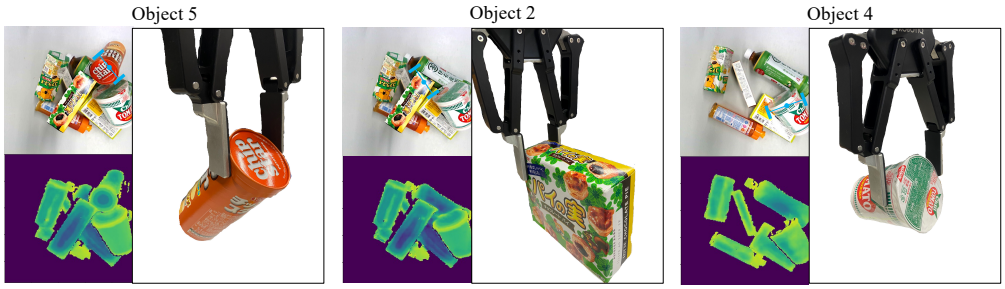


Figure 3.9: Some examples of successful grasping results in clutter scene. There are three successful cases of grasping each target object while preventing deformation and avoiding collision with other objects during grasping.

candidates. Instead of the score, we calculate a new score, same as the mean of stiffness (described in Eq. (3.15)), and delete the grasp pose candidate whose size difference between the contact image and collision image (cropped grasping area) is higher than a threshold value (eliminating failure cases owing to the slippage of the hand and the object).

Figure 3.9 shows some of the successful grasping results. The deformation rate evaluation is presented in Table 3.4. Here, it can be observed that the deformation of the object is suppressed in several cases. For "Object 2", the grasping result is not optimal because the grasping direction differs from the object's surface; hence, the grasping force is applied to a narrower surface than the FGE's result. Regarding the success rate of grasping, FGE succeeded in all attempts, and the proposed method failed in three attempts. It is necessary to improve the method specifically introduced for the cluttered scene, as well as expand the method for 3D because the stiffness of the contact point with the gripper cannot be properly measured using only 2D images.

3.4.3 Discussion

In the two experiments in section 3.4.1 and 3.4.2, when the deformation rate is more than 20%, the damage was caused by large deformation in the FGE case (Figure 3.10), which suggests that the proposed method can reduce damage. The reason for the three failures in the experiments of section 3.4.2 is that the grasping pose selected was often close to

Table 3.4: Deformation rate results for each cluttered object in real-world

Name	*Deformation rate*	
	FGE [18]	Proposed method
Object 1	12.19	8.373
Object 2	14.80	15.39
Object 3	24.35	12.42
Object 4	13.09	6.144
Object 5	16.59	9.129
Object 6	17.45	9.191
Object 7	19.10	16.14
Object 8	64.58	29.69
Mean	22.77	**13.31**

Figure 3.10: Some examples of grasping with significant deformation in the FGE case. Each deformation rate was more than 20%.

the edge of the object. This makes the contact surface smaller, and the possibility of failure was increased by a small disturbance or error in the grasp pose control. Therefore, we need to accurately determine the grasping depth in the clutter scene, and determine the grasping pose based on the grasp stability.

3.5 Conclusion

In this study, we proposed a pix2stiffness estimation method, which generates a stiffness map that indicates the object's stiffness for each pixel on an image using the pix2pix architecture. We demonstrated that the stiffness estimation has a higher accuracy when using depth images as input data than when adapting RGB. Furthermore, we introduced a grasp pose detection method using a stiffness map based on FGE. This method can robustly detect grasp poses in clutter scenes in the real-world. However, more experiments are required for various objects, and generating the stiffness map (data collection in section 3.2.1) is time consuming and cumbersome because it is manually done. In the future, we would like to automatically generate the annotated stiffness map using contact force (e.g. grasped distance for each object [14]). Also, we would like to introduce a force-adjustable method that can grasp with the smallest deformation by considering contact dynamics.

Chapter 4

Formula-based grasping datasets without digitized 3D assets

4.1 Introduction

Research on image-based detection of suitable grasp pose has spawned various approaches, ranging from methods targeting known and unknown objects [16, 18] to deep learning-based techniques [8, 36, 89]. With the rapid advancement of deep learning technology, approaches employing Convolutional Neural Networks (CNNs) and depth images have received increasing attention. However, these methods generally rely on large-scale 3D datasets tailored for object grasping tasks, necessitating a wide range of geometric shapes and complexities to ensure robust performance across diverse scenarios.

As with image datasets, there is a growing trend toward large-scale open-source repositories of 3D datasets, which are increasingly being utilized for applications such as 3D object generation and spatial understanding [39]. However, compiling diverse 3D models with complex geometries demands substantial manual effort. In addition, copyright-related challenges persist—similar to those in the image domain—and consequently, a robust system for managing large-scale 3D data is also essential.

Moreover, it remains unclear whether one can consistently prepare sufficient quantities of training data in precisely the shapes required for grasping a target object. Current practices typically rely on selecting numerous 3D models at random for training. Therefore, leveraging generative 3D models capable of shape manipulation emerges as an effective approach to address this need.

One such approach is EGAD! [41], which facilitates automatic dataset generation for grasping by leveraging an evolutionary model of biological morphology. However, it relies solely on a single global shape complexity metric and does not address the specific geometric requirements essential for robust grasping. Consequently, a dataset generation method capable of capturing a broader spectrum of shape complexities is needed.

In this dissertation, we employ fractal geometry to generate 3D models from a single mathematical formula and thereby construct training data for robotic grasping. By leveraging the intrinsic complexity of fractals, which exhibit both locally and globally intricate structures, we enhance deep learning-based grasp pose detection for improved accuracy and robustness.

The contributions of this dissertation are as follows.

1. Building on fractal geometry, we generate 3D models from a single formula and create a dataset for robotic grasping that spans a wide range of shape complexities. This dataset satisfies the variation in object geometry required for effective grasp learning.

2. In a grasping performance evaluation using GQ-CNN, our method demonstrates accuracy on par with approaches that utilize scanned 3D models for single-object grasping.

3. 3D model generation methods and techniques that can rapidly produce models with a minimal set of parameters, while enabling shape complexity to be controlled through a single parameter.

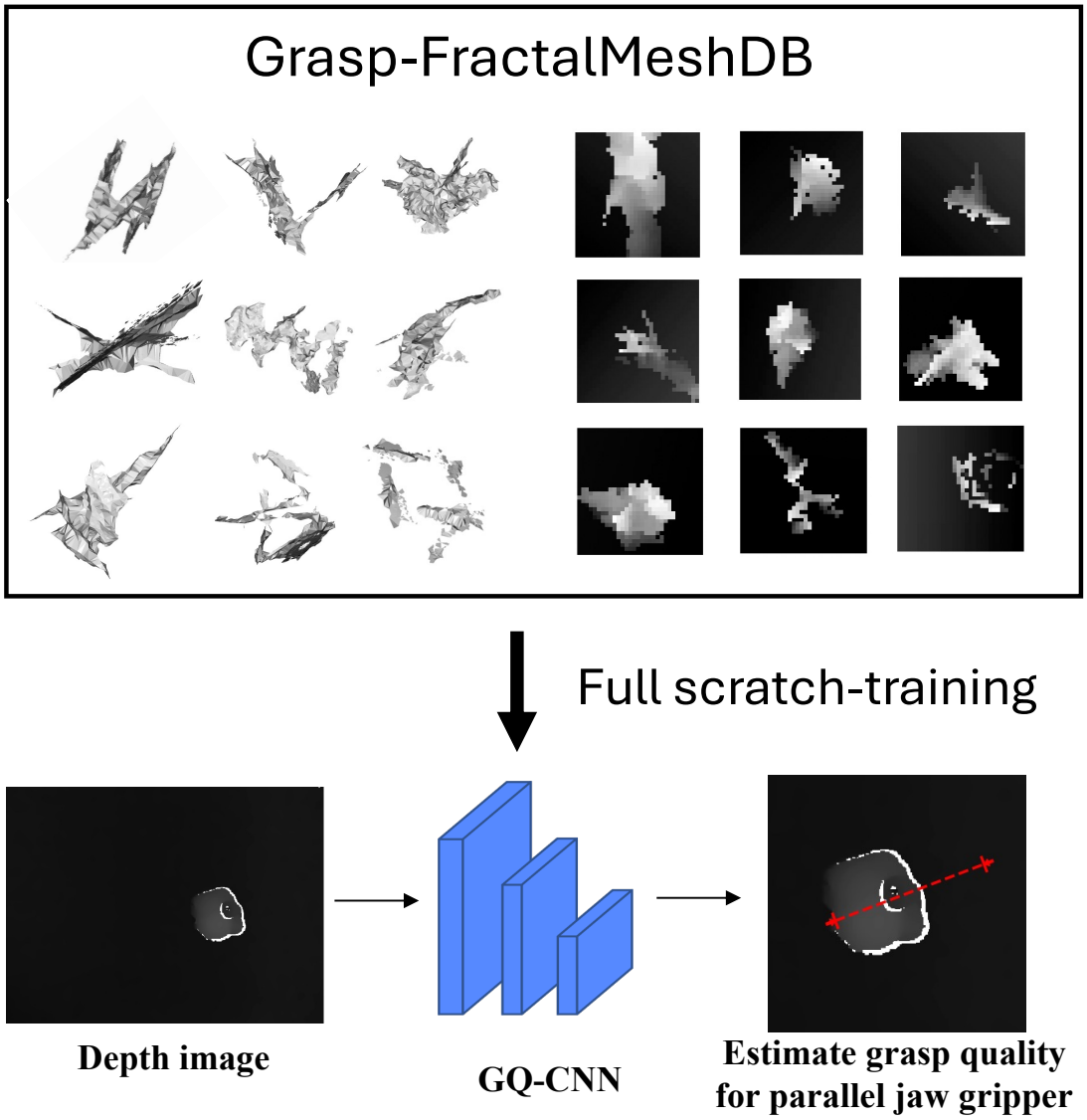


Figure 4.1: Overview of Grasp-MeshFractalDB. A single fractal-based formula is used to generate 3D models, each defined by a mesh surface. From these models, an image database is created to encode grasp quality based on the pose applied during simulation. Once trained on this dataset, the system enables single-object grasping in a real environment.

4.2 Proposed method

In this chapter, we detail the process of constructing a fractal geometry-based 3D mesh model database for robotic grasping. First, we automatically generate 3D fractal models using a 3D Iterated Function System (3D-IFS), a mathematical formulation of fractal geometry proposed in [56, 57]. We then verify the diversity of the resulting point clouds via variance checks and apply alpha-shape surface reconstruction to produce 3D mesh models. Finally, using these mesh models, we build 3D scenes and create the associated image dataset by following the Dex-Net 2.0 generation pipeline [36].

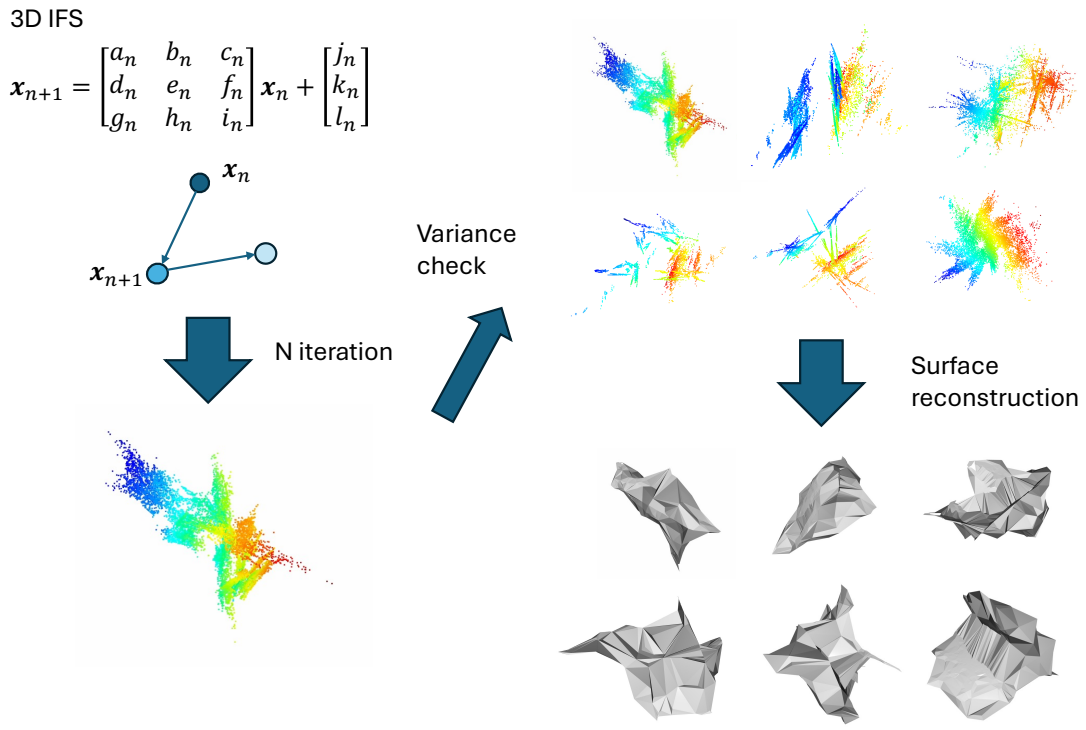


Figure 4.2: Example of a 3D Mesh Model Generated from a Single Fractal Geometry Formula. By applying parameterized 3D-IFS, we first generate a fractal-shaped point cloud. We then compute a dispersion metric for each fractal to verify its suitability as a model. Finally, we configure surface reconstruction parameters and convert the validated point cloud into a 3D mesh.

4.2.1 Point cloud generation

As shown in Equation (1), a 3D-IFS is a mathematical representation of a fractal shape, comprising two key components: an affine transformation function and a corresponding selection probability.

$$\mathit{3D\text{-}IFS} = \{(w_i, p_i)\}_{i=1}^{N} \tag{4.1}$$

A 3D fractal model is generated by repeatedly applying the affine transformation function indicated by its corresponding selection probability, as specified in Equation (2).

$$\mathbf{x}_j = w_i * \mathbf{x}_{j-1} \tag{4.2}$$

Here, $\mathbf{x}_j \in \mathbb{R}^3$ denotes a three-dimensional coordinate, and the initial coordinate $\mathbf{x}_0$ is set to the origin. The affine transformation function is given in Equation (3).

$$w_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ g_i & h_i & i_i \end{bmatrix} + \begin{bmatrix} j_i \\ k_i \\ l_i \end{bmatrix} \tag{4.3}$$

The parameters of the affine transformation function, $\{a_i, \cdots, l_i\}$, are randomly sampled from the range [-1.0, 1.0] . The selection probability is is computed using a 3x3 matrix derived from the randomly chosen parameters $\{a_i, \cdots, i_i\}$, as shown in Equation (4). Here, the 3x3 matrix formed by these parameters is defined as T_i.

$$p_i = \frac{|\det T_i|}{\sum_{i=0}^{N} |\det T_i|} \tag{4.4}$$

Because a uniform selection probability p_i would not inherently yield shape diversity, computing it from random values in this manner enables the generation of a single 3D model with distinct fractal scales and orientations for each component.

4.2.2 Variance check

Because the 3D models are generated from random parameters, similar shapes may appear even when different parameter values are used. To ensure diversity among the

generated shapes, we perform a quality check based on the variance of the point cloud. Specifically, we compute the variance of the points in each model; if the variance exceeds a predefined threshold, the model is retained. This variance is calculated according to the following equation.

$$min(var(X), var(Y), var(Z)) > \sigma \tag{4.5}$$

Here, var computes the variance for each dimension, and the smallest of these values is adopted as the model's overall variance. We then set a variance threshold σ at discrete levels ranging from 0.0 to 0.2, in increments of 0.05.

4.2.3 Surface reconstruction

A closed mesh model is constructed via the Alpha-shape algorithm applied to the point cloud generated by the aforementioned fractal-based method. Alpha-shape generalizes the concept of a convex hull, allowing a single parameter to control the level of detail in the resulting mesh. Specifically, the algorithm first computes a Delaunay triangulation of the point set and then selectively filters the resulting primitives (e.g., triangles in 2D, tetrahedra in 3D) based on the chosen value. As with fractals, the parameters uniquely determine the final shape. The complete data generation pipeline is illustrated in Figure 4.2, and Figure 4.3 shows an example of Alpha-shape applied to a fractal point cloud.

The value of alpha can be set from 0 to infinity. When alpha is set to infinity, a perfectly convex hull is produced, whereas lower alpha values preserve more local features of the original point cloud. This parameter must be determined empirically, as the optimal value depends on the specific characteristics of the underlying data. In generating a 3D mesh model, alpha is chosen such that the resulting mesh is closed and contains a single connected object. As an example, Figure 4.3 illustrates how adjusting alpha affects the mesh. Around alpha = 0.1, the mesh accurately reflects local structure, while at alpha = 0.7 to 1.0, it approaches a near-convex hull representation.

Finally, we construct a dataset for robotic grasping using these mesh models. Various data generation methods can be employed to build this dataset, including Dex-Net,

which relies on 3D models.

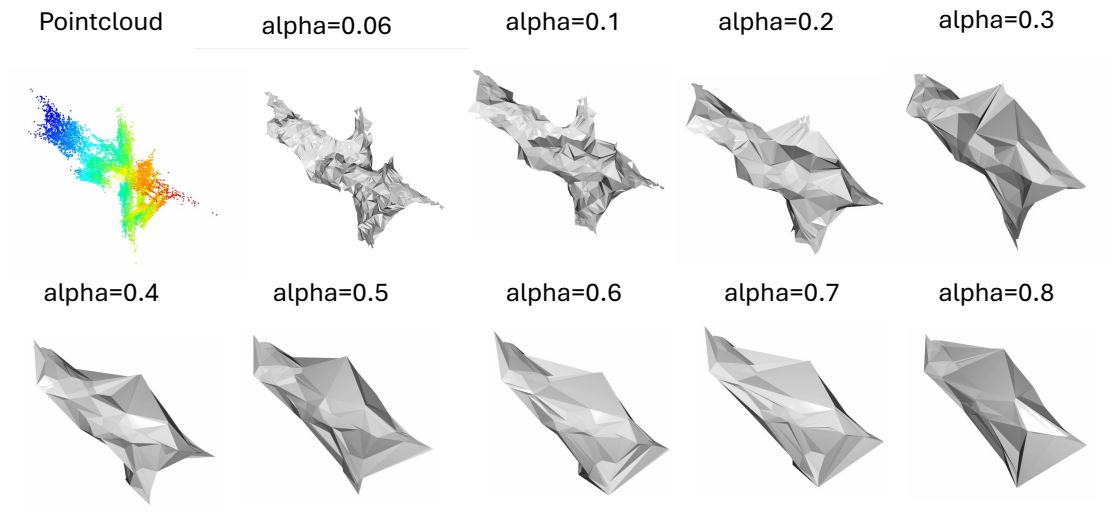


Figure 4.3: An example of a 3D mesh model generated from a single fractal geometry formula, illustrating how the alpha-shape parameter influences the resulting shape when varied from 0.06 to 0.8.

4.3 Experiments

In this section, we evaluate how the method introduced in the previous chapter affects the performance of an image-based model for grasp pose detection.

4.3.1 Implementation settings

In this experiment, we adopt the GQ-CNN model from Dex-Net 2.0 [36]. Following the dataset generation rules outlined in Dex-Net 2.0, we start with a 3D mesh (comprising the object model, gripper, and desk plane) and automatically collect sampled grasp poses and corresponding images within a designated region around each grasp center. We also record the 3D coordinates of the grasp candidate as well as its grasp-performance metric. The 3D model is adjusted so that it can be grasped with the specified gripper width (50 mm in this dissertation).

Figure 4.4 illustrates examples of the sampled grasp poses, their performance values, and the resulting rendered scene. The dataset generation process follows the Dex-Net 2.0 pipeline and proceeds as follows.

1. Sampling grasp candidates and analyzing their grasp stability
2. Determining stable object poses on a flat surface
3. Performing collision detection among the desk, gripper, and object for 4-DoF grasp candidates
4. Rendering images from a randomly specified camera pose
5. Pairing each rendered image with the performance results obtained in step 1)

As comparison baselines, we adopt Dex-Net 2.0 and EGAD!, along with datasets constructed by varying the value of alpha for the alpha-shape parameter. Each dataset comprises approximately 150 models, yielding between 15k and 20k images. We select six object types for grasping trials and conduct experiments in the picking environment depicted in Figure [reference], grasping each object five times in various orientations. Figure 4.5 shows representative examples from these datasets.

While Dex-Net features a diverse set of object shapes, EGAD! enables the creation of geometrically complex objects while retaining basic outlines, such as spheres and rectangles. By contrast, our proposed database includes a wide range of shape variations, resulting in greater appearance diversity.

4.3.2 Simulation experiments

We utilize the dataset from the previous section to evaluate the effectiveness of our learning-based grasping approach. Six target objects are tested—two everyday objects, two from the YCB dataset, and two adversarial objects from Dex-Net shown in Figure 4.6. Each model is evaluated using images corresponding to the sampled grasp poses generated by the same pipeline as in the previous section. Ground truth labels are

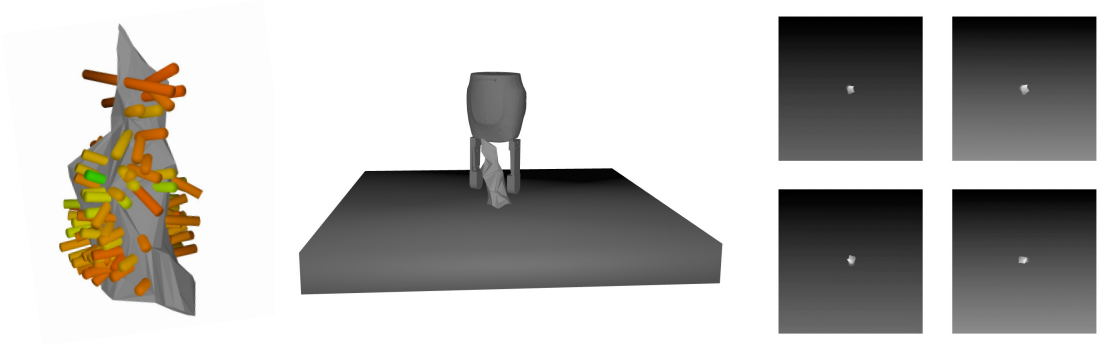


Figure 4.4: Illustration of the 3D annotations and scene setup in Dex-Net 2.0's dataset generation process. Left: Sampled grasp poses for a parallel gripper on the object model. The red line represents low grasp quality, while the green line indicates high grasp quality. Middle and Right: Scenes where the object is successfully grasped without collisions, followed by rendering from a camera posed above the desk.

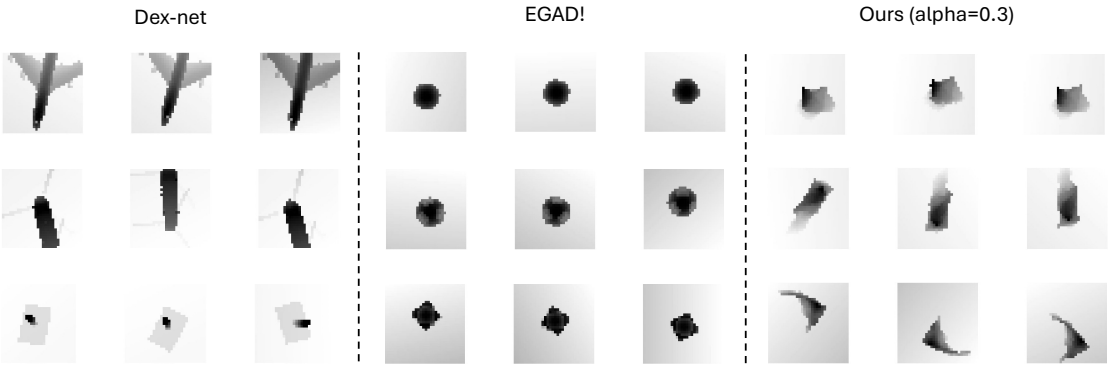


Figure 4.5: Image datasets generated from Fractal mesh models

assigned based on whether the analytical grasp performance value exceeds 0.5 ("graspable") or falls below 0.5 ("ungraspable"). We vary the alpha parameter at three levels (0.06, 0.3, 0.7), chosen empirically to ensure visible shape differences.

As summarized in Table 4.1, our proposed method achieves the highest classification accuracy, highlighting the benefits of learning from diverse shape contours—even those not found in reality. Figure 4.7 provides examples of successful and failed classifications for the proposed method when alpha is 0.3. While it robustly estimates stable grasps for everyday objects and detects unstable grasps in complex shapes (e.g., Dex-Net

adversarial objects), it occasionally misclassifies objects with intricate details, such as brushes. One reason is that higher alpha values reduce shape fidelity, making it harder to capture fine concavities and protrusions. Furthermore, training with objects at multiple alpha values simultaneously yields lower accuracy than using only 0.3, indicating that the model may not adequately generalize across a broad spectrum of object types.

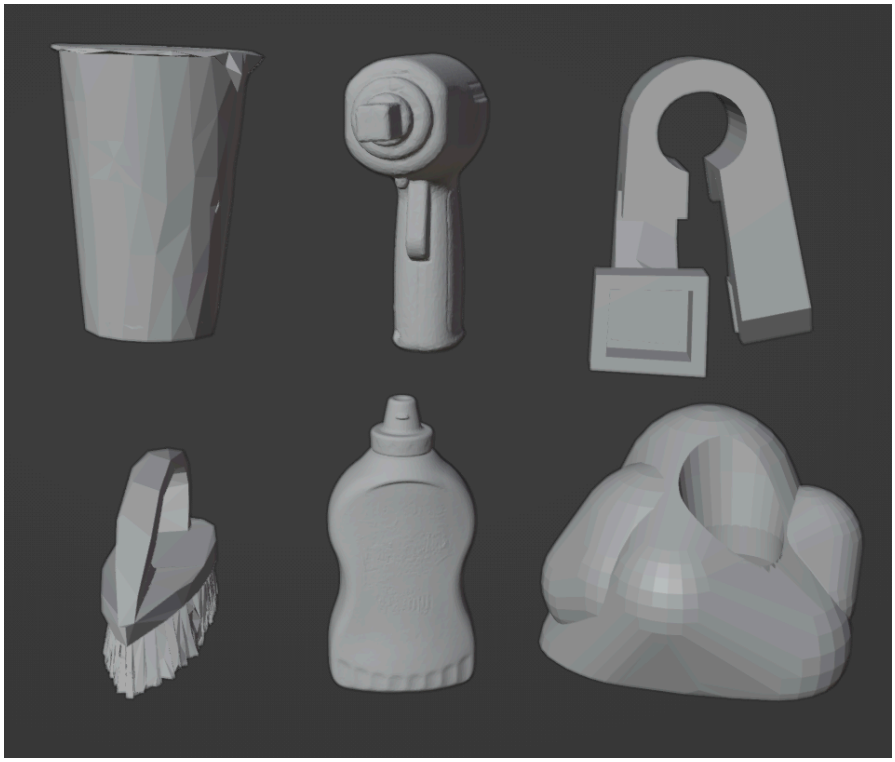


Figure 4.6: Target objects from daily items, YCB dataset, and Dex-net Adversarial objects for evaluating image datasets

4.3.3 Real-world experiments

System Configuration

We evaluate the effectiveness of pre-training on Grasp-FractalDB by performing grasp experiments on a UR5e robot equipped with a Robotiq two-finger gripper (140 mm) featuring rubber tips. A RealSense SR305 depth sensor is utilized, and the overall setup parallels the Dex-Net 2.0 configuration (left side of Figure 4.8). The GQ-CNN model,

Table 4.1: Grasping results for each object and dataset

Dataset	Accuracy [%]
Dex-net 2.0 [36]	63.42
EGAD! [41]	61.10
Fractal (alpha=0.06)	51.16
Fractal (alpha=0.3)	**66.70**
Fractal (alpha=0.7)	65.93
Fractal (alpha=0.06, 0.3, 0.7)	60.23

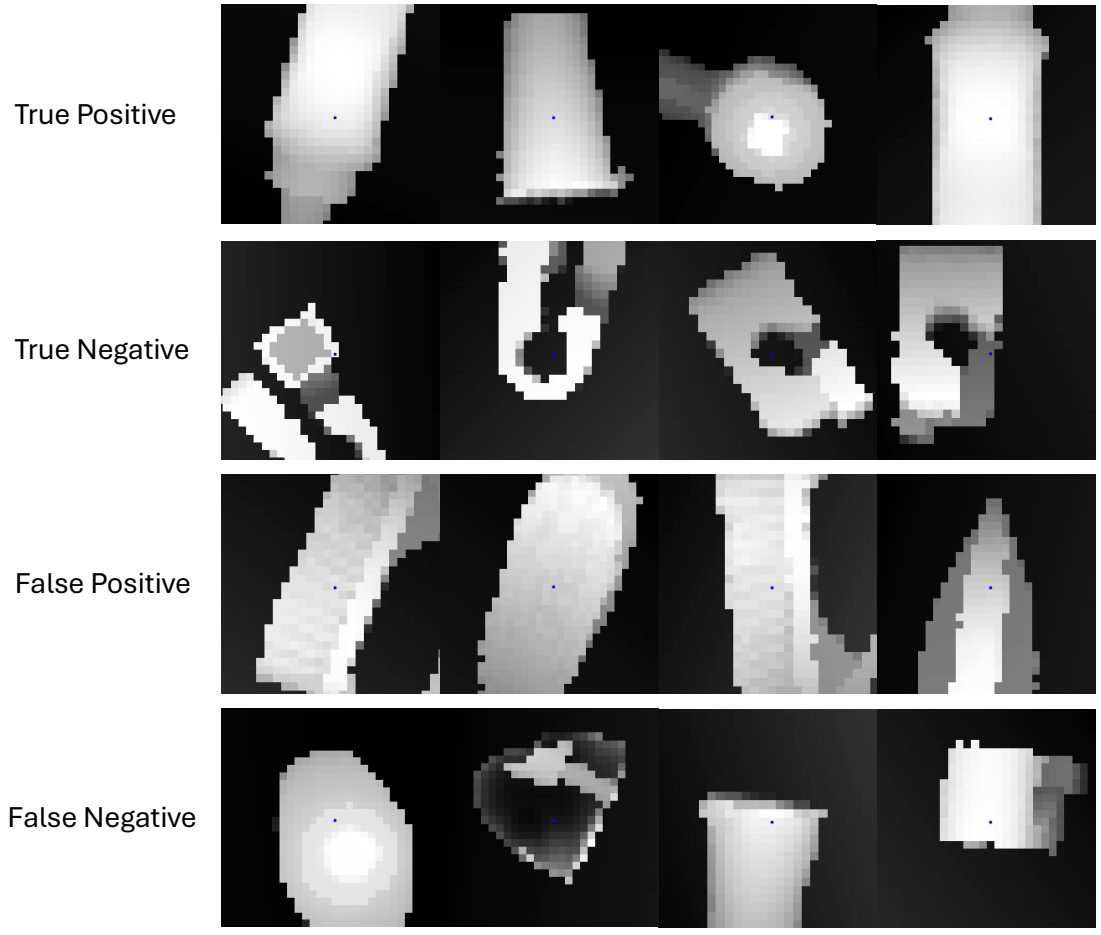


Figure 4.7: The examples of classification results.

also used in the simulation experiments, is employed here to determine a robust grasping strategy through the cross-entropy method. GQ-CNN samples grasp poses and crops images based on a specified gripper width, which must be adjusted depending on the target object. To automatically determine the appropriate gripper width for each object in the scene, we apply the following pipeline:

1. Inpaint missing regions in the depth image.
2. Extract a binary mask based on a specified height threshold (distance from the desk).
3. Retain only objects whose minimum average depth value falls within the segmented region.
4. Perform ellipse fitting to compute the minor axis and convert its length to millimeters.
5. Adjust the image crop size to match the gripper width.

The standard crop size is 96×96 pixels, corresponding to a 50 [mm] gripper. We scale this crop proportionally to the gripper size determined in the above pipeline.

Results

The experimental results indicated that the proposed method performed as well as or better than both Dex-Net 2.0 and EGAD!. Adjusting the parameter revealed that a value of 0.06 yielded the best results, while a value of underperformed relative to the benchmark methods. These observations suggest that alpha is a crucial parameter influencing grasping performance. Interestingly, even with a mix of three alpha values, the performance remained high. Therefore, it appears that if the dataset encompasses a specific range of alpha parameters, a high success rate in picking tasks can be achieved without extensive parameter tuning.

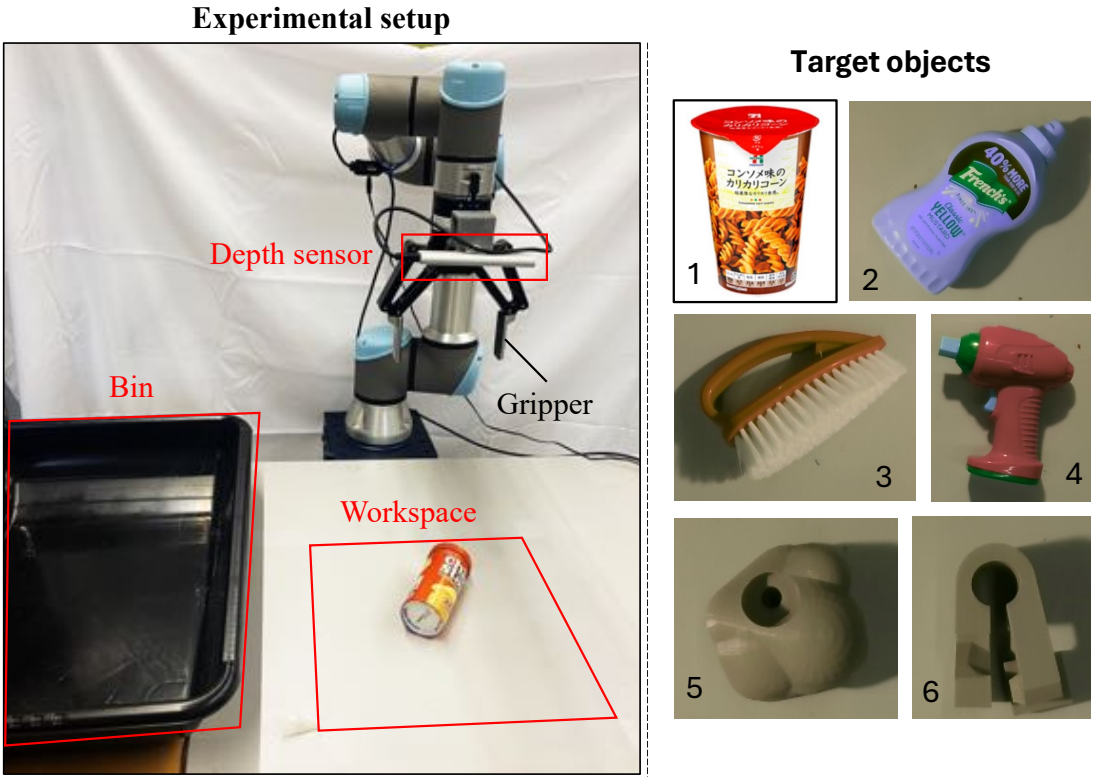


Figure 4.8: (Left) Experimental setup featuring a UR5e robot and a Robotiq two-finger gripper (140 [mm]). A depth image is captured from a sensor mounted on the table beneath the workspace. The GQ-CNN then detects the grasp pose. A grasp is considered successful if the object is dropped into a black bin adjacent to the workspace. (Right) Sample test objects include everyday items commonly found in Japanese convenience stores; the lower-left image shows items from the YCB dataset

4.3.4 Discussion

We applied the database we proposed to the spatial distribution of the difficulty of grasping and the complexity of the shape of the 3D models defined in EGAD! and analyzed the relationship with the target object Figure 4.9. We applied our proposed database to analyze the spatial distribution of grasping difficulty and the complexity of the 3D model shapes as defined in EGAD!. The analysis explores the relationship with the target object, as depicted in Figure 4.9. The results demonstrate that training on datasets featuring complex liquid shapes, which are inherently difficult to grasp, facilitates the

Table 4.2: Grasping results for each object and dataset

Dataset \ Object	1	2	3	4	5	6	Mean
Dex-net 2.0 [36]	1.0	1.0	0.8	0.8	0.6	1.0	0.867
EGAD! [41]	1.0	1.0	1.0	0.8	0.8	1.0	0.933
Fractal (alpha=0.06)	1.0	1.0	0.8	1.0	1.0	1.0	**0.967**
Fractal (alpha=0.3)	1.0	1.0	0.8	1.0	0.2	0.8	0.800
Fractal (alpha=0.7)	1.0	0.8	1.0	1.0	0.8	1.0	0.933
Fractal (alpha=0.06, 0.3, 0.7)	1.0	1.0	0.8	1.0	0.8	1.0	0.933

successful handling of simpler shapes and more easily graspable objects. However, the spatial distribution analyzed does not encompass all the critical elements required for effective grasping. Therefore, it is necessary to re-evaluate and consider additional axes that have not yet been assessed. While the Dex-Net dataset broadly covers the map, the proposed method includes many data points associated with high grasping difficulty. This trend is attributed to the fractal 3D models having non-smooth surfaces and a limited number of parallel planes conducive to easy grasping, resulting in many sampled grasp poses being challenging. In terms of shape complexity, the distribution shifts towards higher complexity areas as the Alpha parameter is reduced. Although many target objects align with the Dex-Net dataset mapping, the continued high success rate using the Fractal method demonstrates that it is feasible to learn effective grasping strategies for datasets characterized by complex shapes and high grasping difficulty, even without fully covering this distribution.

4.3.5 Compulational efficiency

We quantified the time required to generate the dataset. To establish baseline metrics for database construction times, we used EGAD! for comparison on an Apple M1 Max platform.

The verification results presented in Table 4.3 indicate a significant reduction in the

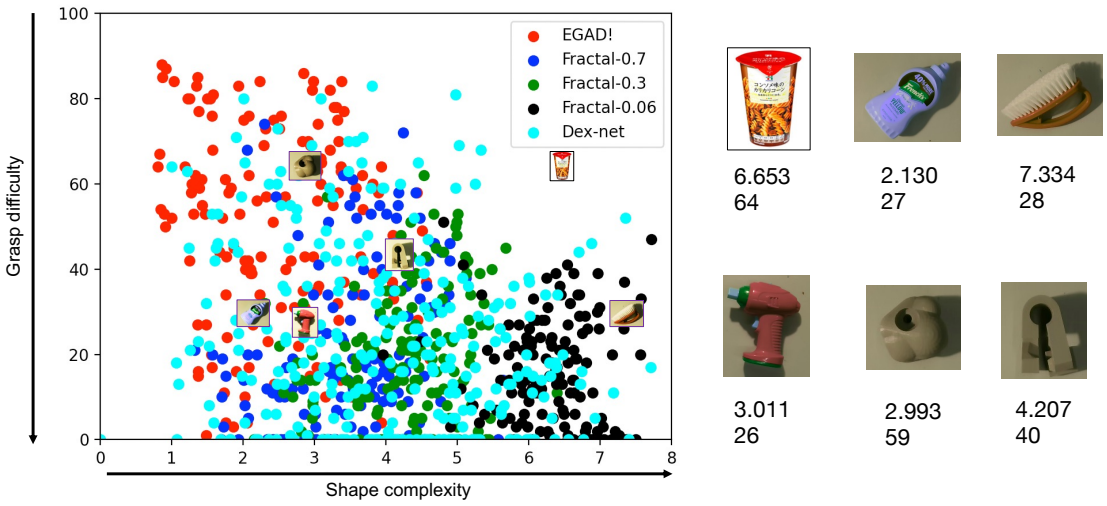


Figure 4.9: 2D feature map for each 3D model datasets and targets

Table 4.3: Cost performance for generating dataset

Dataset	Hyper-parameter	3D model [sec/model]	Image [sec/model]
EGAD! [7]	59+	17.6+	48 (120 images)
Fractal (alpha=0.3)	13	1.29-2.43	47 (111 images)

time required to generate 3D models using our proposed method. Variation in the Alpha parameter influenced the generation times, with more complex shapes (lower value of alpha) necessitating longer calculations. However, these times remained shorter than those required for generating models with EGAD!.

4.4 Conclusion

In this dissertation, we constructed a 3D mesh model using a mathematical formula derived from fractal geometry and applied a database built according to Dex-Net generation rules to a deep learning model for grasping. We concentrated on optimizing the data generation pipeline, explored the implications of varying the number of parameters, and conducted a thorough verification of the calculation times involved.

A limitation of our method is that the shape of the object is determined randomly, making it challenging to generate shapes and scales akin to real objects. Scanned 3D models offer a rich variety of shapes but do not adequately represent composite objects such as articulated items or variations in hand and object scales. Consequently, there is a need for a data generation method that can accurately reflect these complexities.

Chapter 5

Deformability-based grasp pose detection from a visible image

This thesis chapter originally appeared in the literature as

K. Makihara, Y. Domae, R. Hanai, I. G. Ramirez-Alpizar, H. Kataoka and K. Harada, "Deformability-based grasp pose detection from a visible image," in IEEE Access, doi: 10.1109/ACCESS.2024.3511546.

5.1 Introduction

In recent years, several approaches focused on grasp pose detection methods that estimate robotic grasps based on the appearance of an object [9,10,18,19].This allows robots to grasp objects of different colors, shapes, and materials. Although these methods can be used to grasp deformable objects, grasping can fail, and the object can be squashed and/or damaged owing to deformation. To address this problem, the deformability of an object must be considered when estimating the grasp based on the appearance of the object. Some methods for grasp planning that consider object deformability [14,90] and techniques for detecting grasp poses from images [47,82] have been proposed; however, their application remains limited in scenarios involving numerous objects. Achieving

effective grasping in complex environments, such as random-pile scenes with densely arranged objects, presents significant challenges. This dissertation aims to approximately estimate the deformability of objects in cluttered scenes, enabling successful grasping without inducing deformation. Furthermore, our approach ensures robust grasping performance even in situations where deformable obstacles hinder access to target objects.

In this dissertation, we assume that a constant correlation exists between the appearance of an object and its deformability. For example, when manipulating everyday items in a supermarket, humans typically estimate the approximate firmness or softness of a new product based on visual information alone or on past experiences. Therefore, we propose a deep learning model to establish this relationship. As extensive training data are necessary for learning, we built an image database composed of images and deformability map pairs by creating bin-picking scenarios using several objects from a small dataset of 3D models, with their respective deformability maps manually evaluated.

Based on the experimental results, we showed that it is possible to estimate the deformability of unknown objects in the same product category that have different shapes, sizes, and colors. Furthermore, we propose a grasp pose detection method based on the deformability estimated using a trained deep learning model. Finally, we present experimental results to demonstrate that the proposed grasp pose detection method can select a suitable grasp and thereby enable successful grasping of deformable objects that are difficult to grasp using existing methods (Figure 5.1).

The contributions of this dissertation are as follows.

1. We propose a network that can estimate the deformability of an object with instance segmentation by combining a Mask R-CNN [48] and a deformability estimation layer. We verified that we could estimate the deformability of each object from its appearance in scenes with different objects.

2. The grasp pose detection method that considers the estimated object's deformability, making it possible to properly pick up an object even in scenes where existing methods either fail to grasp the object or damage it.

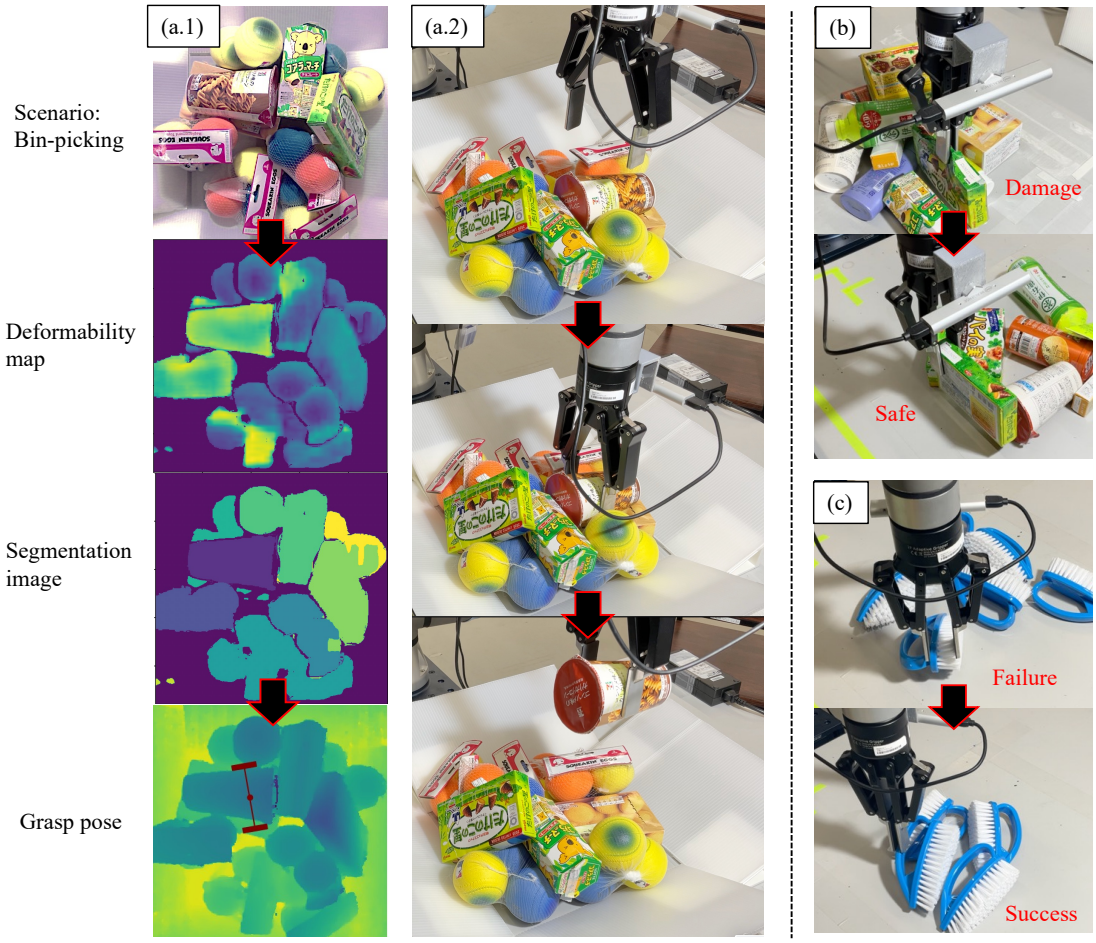


Figure 5.1: Results of the proposed Method: (a.1) From the original image, the deep learning model generates a deformability map and a segmentation image. A suitable grasp pose (highlighted in red) is then selected to prevent potential damage while simultaneously displacing nearby obstacles. (a.2) This approach enables the successful grasp of the target object while pushing away deformable obstacles. (b) Our method effectively mitigates the risk of damage, and (c) achieves successful grasping by directing the gripper towards rigid areas of the target.

The remainder of this paper is organized as follows. Related works are reviewed in Section II. Section III presents an overview of the proposed methods (deformability estimation and grasp pose detection). In Section IV, we evaluate these methods in real-world environments. Finally, we conclude this paper and discuss some of its limitations in Section V.

5.2 Proposed method

In this section, we describe three proposed methods: 1) a method for instance segmentation and deformability map generation based on a Mask R-CNN, 2) a method for data collection and model training, and 3) a grasp pose detection method based on estimated deformability (Figure 5.2).

5.2.1 Definition of the deformability

In existing grasp pose detection methods, objects are typically assumed to be rigid bodies, and grasp poses are identified based on collision-free constraints with surrounding obstacles. However, this assumption can result in significant deformation or even damage when interacting with deformable objects. Additionally, in cluttered environments with numerous obstacles, this approach may fail to detect valid grasp candidates or may select suboptimal candidates prone to grasp failure. Therefore, it is crucial to incorporate the consideration of object deformability in the grasp pose detection stage to enhance both the reliability and safety of grasping in complex environments.

The deformability map indicated the location of the deformability of an object for each pixel in the image. Deformability refers to the ease with which an object can be deformed; a high score signifies that the object is easy to deform, whereas a low score indicates that it is difficult to deform. For grasping deformable objects, the absolute score was determined by measuring both the deformation and grasping forces. In this context, a stiffness map, as proposed in [14], plays a crucial role in controlling the grasping force. Understanding this definition is vital when considering the force applied; however, estimating the score without contact is challenging for objects of unknown shapes and materials. Therefore, the score was categorized into 10 steps and annotated by humans. For example, parts that do not deform even under a significant mechanical force, such as a PET bottle cap, receive the highest score, whereas those that can deform with minimal force, such as cloth, receive the lowest score. The annotation process involves scoring each part of an object within a 3D model.

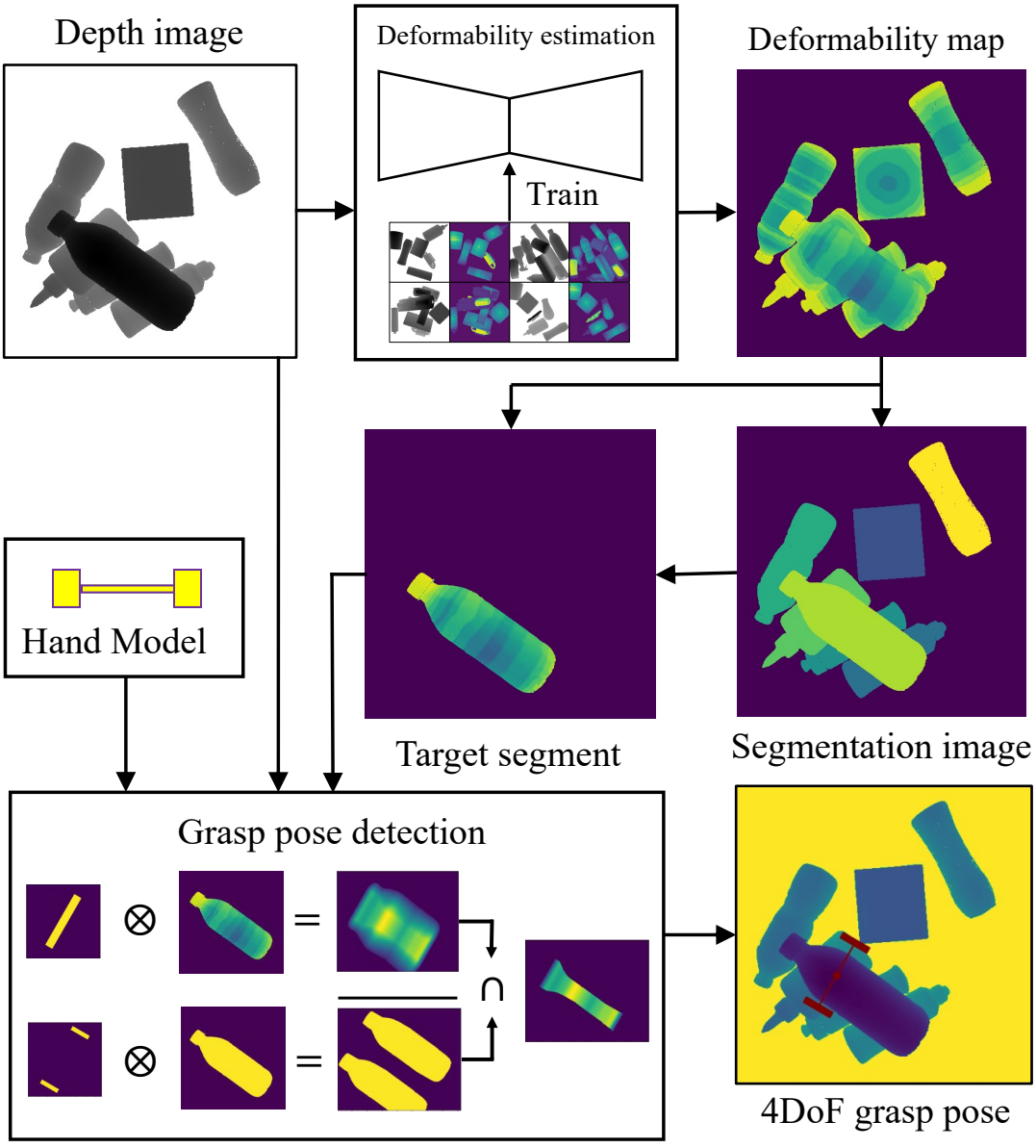


Figure 5.2: Overview of the proposed method: With a single depth image as input, the deformability map and target segment was generated using our deformability estimation method. Using the images, a 4-DoF grasp pose for a two-finger gripper was detected.

5.2.2 Encoder–Decoder model for deformability estimation

We contract the model for estimateing the deformability represented by a 2D input image by assigning a deformability value to each pixel of the object image, thus generating a

deformability map of the same size as the input image. The proposed deformability estimation method is based on a Mask R-CNN [48].

As mentioned in Section II, Mask R-CNN can perform multiple tasks simultaneously, and the multitasking performance is complementarily enhanced. In the proposed method, we add deformability estimation layer to the Mask R-CNN and design the model such that instance segmentation is also possible by devising the structure of the network and generating image data for training. Specifically, the detection and segmentation parts are included in the head of the Mask R-CNN in a Fully Convolutional Network (FCN) [91]. During training, the objective multi-task loss function is $L = L_{cls} + L_{det} + L_{mask} + L_{deform}$. the classification loss L_{cls}, bounding-box loss L_{det}, and segmentation loss L_{mask} are same ones as in [48]. In the classification task, we conduct binary classification to distinguish whether an object is in the foreground or background. In the detection task, we identify the bounding box of the object's candidate region. For the segmentation task, we output the probability of each pixel within the bounding box belonging to the object's region. In the deformability estimation task, we predict the deformability of the object for each pixel within the bounding box. The deformability estimation component is constructed to generate a map similar to the segmentation map. For segmentation loss, it is necessary to determine whether each pixel in the obtained region belongs to a defined class. Therefore, we used binary cross-entropy as a loss function for training. However, for deformability, we also require an output score between 0 and 1. Therefore, a layer was required to generate whole images similar to those in the image generation model. In particular, the estimated deformability map must be similar to the structure in part of the image. Therefore, we use structural similarity [85] loss to perform more robustly for unknown similar objects for the deformability estimation loss L_{deform}. In multi-task learning, each task contributes to the learning of other tasks, resulting in enhanced overall performance. Consequently, the same weight is assigned to each loss function throughout the learning process.

For pretraining, the model learns only the detection and segmentation layers, similar to the Mask R-CNN using the WISDOM [92] datasets. The input image was a depth image, and the depth was extended to three channels as the input to the network to

unify the input channels for feature extraction. To adapt to unknown objects of the same product category but with different colors, sizes, and shapes, it is more likely that feature extraction for deformability estimation is better using depth images than using RGB images such as texture. From the feature map obtained by feature extraction using ResNet50-FPN, which integrates ResNet50 [93] and a feature pyramid network (FPN) [94] into the backbone of the input image, we can obtain the feature map of the input image. In related segmentation techniques, it has been indicated that using a transformer as the backbone and increasing the model size are factors that enhance the accuracy. However, we focused on determining whether deformability estimation is effective in models that solve multiple tasks, such as the Mask R-CNN. Therefore, we used basic models, such as ResNet. A region proposal network (RPN) [95] was used to extract the object candidate regions in the feature map, and the individual object candidate region outputs from the RPN were extracted by the RoIAlign layer as the output. In the output, each object candidate region from the RPN is extracted by the RoIAlign layer and an image showing the object and its deformability (0 to 1) is generated from the FCN structure. In this manner, we realized a network that performs instance segmentation and deformability estimation (Figure 5.3).

Following the integration of the deformability estimation layer into the pre-trained Mask R-CNN, fine-tuning is conducted with a learning rate initialized at 0.00002 for 20 epochs. A StepLR scheduler is applied to decay the learning rate by a factor of 0.1 every three epochs. The optimization process employs the Adam [96] optimizer, with decay parameters β_1 and β_2 set to 0.9 and 0.999, respectively. The training process is executed on a single NVIDIA RTX A6000 GPU with a batch size of 16.

5.2.3 Simulation-based semi-automatic data collection

A large number of object depth images and their corresponding estimated deformability maps are needed to train the deformability estimation model, and generating them in real-world is difficult and time-consuming.Therefore, simulations were used to automate half of the procedure.

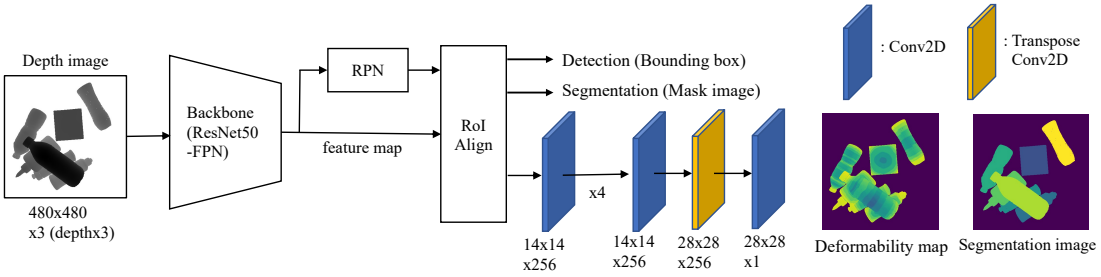


Figure 5.3: Encoder–decoder model for deformability estimation. Using a 3-channel depth image as input, identify object candidate regions from the feature map extracted by Backbone (ResNet50-FPN) and RPN. The object candidate regions are extracted from the RoIAlign layer, and deformability maps are generated for each object candidate region by classifying the object or background, estimating the bounding box, and using the FCN structure. Finally, by combining each map, we can obtain the entire deformability map. In addition, by generating a mask image with a certain threshold from each map, we can also perform instance segmentation and obtain a segmentation image.

We assumed a bin-picking environment for grasping one object and generating a bin-picking scene in the simulation and rendered images. To create the scene, we used 3D models of deformable objects gathered from daily object databases (NEDO ITEM DATABASE[1], APC2017 RGB-D dataset [97] and YCB dataset [37]). Following the data generation pipeline described in [47], the training dataset was constructed using Pybullet simulations, which were used to generate segmentation maps concurrently (Figure 5.4). The dataset includes 15 types of deformable hollow objects, such as bottles and boxes with hollow interiors, and 5 types of partially deformable objects, such as brushes with both deformable and rigid components. In the simulation, between 3 and 8 objects are randomly selected and repeatedly dropped into a bin from above. This process is repeated 10,000 times in total to generate the image dataset, which is subsequently used for training.

[1]http://mprg.cs.chubu.ac.jp/NEDO_DB/

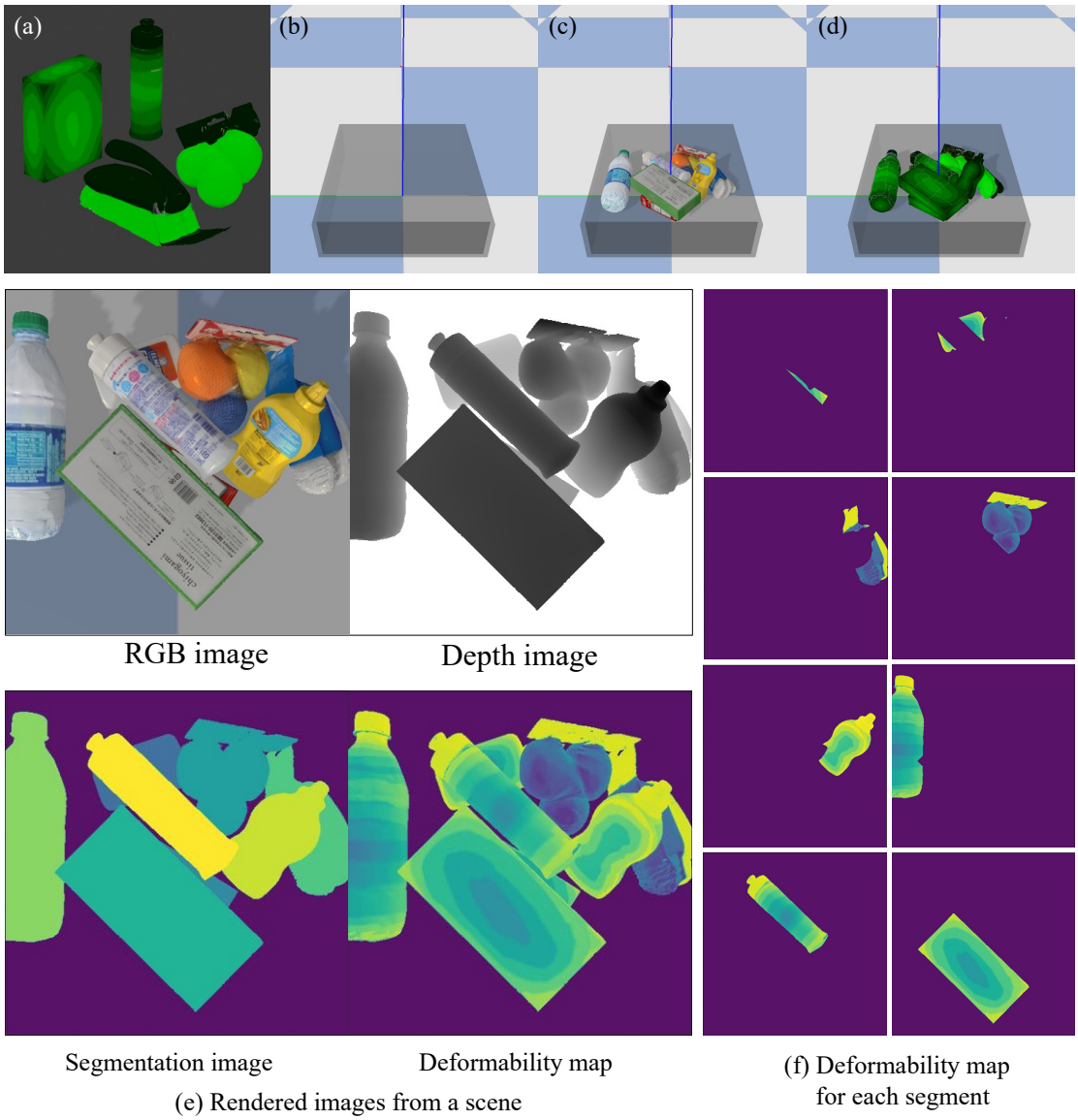


Figure 5.4: Simulator-based data collection: (a) examples of objects textured the deformability as green tone, (b)-(d) the scene generation using a physics simulator. (e) After various images are rendered from the scene, (f) the deformability map as a segment was created for each object.

5.2.4 Grasp pose detection based on deformability map

We performed grasp pose detection using the segmentation image and deformability map obtained as the output of the neural network, as explained in the previous section.

Target object selection from segmentation image

In bin-picking scenarios, it is important to select the object to be grasped correctly. To select an appropriate target object using the segmentation image obtained from the network, we targeted the segment that was judged as an object with a probability of more than 95% in the probability of being an object or background obtained by object detection. Among these segments, the segment with the smallest depth (maximum height of the object from the desk visible in the image) in each object region was designated as the target.

Grasp pose detection with segment and deformability map

The grasp pose detection method is similar to that of pix2stiffness [47]. The contact $\mathbf{T}_t^\theta$ and collision $\mathbf{T}_c^\theta$ templates for each rotation angle θ are predefined, whereas the contact $\mathbf{I}_t$ and collision $\mathbf{I}_c$ images are obtained from a single-depth image $\mathbf{D}$. The deformability contact image $\mathbf{I}_{st}$ was calculated by multiplying the generated deformability map $\mathbf{S}^*$ in the area of the target segment $\mathbf{I}_s$ and $\mathbf{I}_t$ and then convoluted with $\mathbf{T}_t^\theta$. Subsequently, the deformability contact region $\mathbf{A}_{st}^\theta$ was generated, and $\mathbf{A}_c^\theta$ was calculated by convoluting $\mathbf{T}_c^\theta$ with $\mathbf{I}_c$. After applying a logical AND operation between $\mathbf{A}_{st}^\theta$ and $\mathbf{A}_c^\theta$, a noncollision graspability map considering the deformability $\mathbf{G}^\theta$ was generated. The objective function is defined as follows:

$$f(x, y, \theta) = \begin{cases} G^\theta(x, y) & if \ \overline{A_c^\theta}(x, y) = 1 \\ 0 & otherwise \end{cases} , \tag{5.1}$$

where θ is the rotation angle of detected grasp candidates. The calculated coordinate index is expressed as

$$[X, Y, \Theta] = \arg\max_{x,y,\theta} f(x, y, \theta) \ . \tag{5.2}$$

Here, we solely utilize the hand template of the two-finger gripper; therefore, we can apply Eq. (2) as the objective function.

Detection was performed by changing the width of the grasp in the hand template, and the grasp position with the smallest possible width was selected. To select the

grasping height, we start at the height where the segment of the selected object had a certain size K; then, we took cross-sections of L mm, each divided into M steps. L and M were determined by the average size of the target object and the length of the toe of the hand, respectively. The overall structure of the algorithm is as follows. The deformability map for each region of the extracted input depth image is $\mathbf{S}_i (i = 1, ..., N)$. For each of these deformability maps, a binary image was generated at a certain threshold R to simultaneously obtain a segmentation image. Then, $min(\mathbf{I}_g)$ extracts the minimum value in an image $\mathbf{I}_g$, and $binary(\mathbf{I}_g, h)$ generates a binary image with 1 for the threshold h and 0 otherwise. $region(\mathbf{I})$ returns the calculated number of pixels in each labeled region of the binary image $\mathbf{I}$. $GP(\mathbf{I}_t, \mathbf{I}_c, \mathbf{S}^*)$ is a function that detects grasp pose $[X, Y, \Theta]$ from Eq. (2) using contact, collision, and deformability maps.

The grasping score $Q(\mathbf{S}^*, \mathbf{I}_t, X_i, Y_i, \Theta_i)$ calculates the average estimated deformability in the area inside the hand from $\mathbf{I}_t$ to determine the optimal grasp poses at different depths. The detection algorithm is presented as **Algorithm 1**.

The aforementioned process is used to detect the grasp position of the obtained 4-DoF grasp pose.

5.3 Experiments

We verified whether the deformability map could be useful for grasp-pose detection. For this purpose, we used (1) a cluttered scene with several unknown objects that were not included in the training data but belonged to the same category, (2) a cluttered scene of partially deformable objects where the hard and soft parts were clearly separated (e.g., a brush with a handle), and (3) a specific case in which a target object was specified, but some deformable objects were located around the target.

Algorithm 1 Grasp pose detection based on deformability estimation

Input: $K, L, M, R, \mathbf{D}, \mathbf{S}_i (i = 1, ..., N)$

Output: $[X^*, Y^*, Z^*, \Theta^*]$

Target segment selection :

$D^* \leftarrow \infty$

for $i = 1$ to N **do**

  if $D^* > min(\mathbf{D} \bigodot binary(\mathbf{S}_i, R))$ **then**

    $D^* \leftarrow min(\mathbf{D} \bigodot binary(\mathbf{S}_i, R))$

    $\mathbf{I}_s \leftarrow binary(\mathbf{S}_i, R)$

    $\mathbf{S}^* \leftarrow \mathbf{S}_i$

Grasp pose detection :

for $i = 0$ to M **do**

  if $K < region(binary(\mathbf{D}, i * L))$ **then**

    $H^* \leftarrow i$

    break

$Q^* \leftarrow 0$

for $i = 0$ to 4 **do**

  $\mathbf{I}_t \leftarrow binary(\mathbf{D}, (i + H^*) * L)$ * $\mathbf{I}_s$

  $\mathbf{I}_c \leftarrow binary(\mathbf{D}, (i + H^* + 1) * L)$

  $(X_i, Y_i, \Theta_i) = GP(\mathbf{I}_t, \mathbf{I}_c, \mathbf{S}^*)$

  if $Q(\mathbf{S}^*, \mathbf{I}_t, X_i, Y_i, \Theta_i) > Q^*$ **then**

    $Q^* \leftarrow Q(\mathbf{S}^*, \mathbf{I}_t, X_i, Y_i, \Theta_i)$

    $[X^*, Y^*, Z^*, \Theta^*] \leftarrow [X_i, Y_i, (D^* + i * L), \Theta_i]$

5.3.1 Hardware settings

We validated the effectiveness of the proposed method using UR5e and a Robotiq two-finger gripper (140 mm) with rubber tips. We used Realsense SR305 as the depth sensor attached to the end-effector.

The camera height used to capture depth images, set at 550 [mm], matched that used in the simulation, and the camera was oriented such that its optical axis was

perpendicular to the plane of the desk. The grasping force, pre-measured to verify its adequacy for lifting all objects within the scene, was set to 67.5 [N].

5.3.2 Grasping for deformable hollow objects

We evaluated the efficacy of the proposed method using unknown objects that fell within similar categories. Given that the training data encompassed hollow bottles and boxes, we selected 10 unknown objects from categories that varied in texture and scale. In addition to the grasping success rate, we assessed the object deformation as a quantitative evaluation index. To compute this index, we measured the distance between the gripper fingers when grasping with the minimum gripping force w_c (12.5 [N] in this case) and when grasping with a constant grasping force w_g used in the experiments (67.5 [N] in the experiments). The deformation was quantified using the following equation:

$$Deformation = \frac{w_c - w_t}{w_c} \, . \tag{5.3}$$

This equation, similar to the definition of strain, indicates that lower values correspond to lower deformations. We created a bulk stacking scenario with randomly placed objects and repeated the picking task three times until all the objects were cleared. For comparative analysis, we employed FGE [18] and Dex-net 4.0 [10], which do not account for object deformability but have high grasp success rates. Additionally, we utilized the pix2stiffness [47], which is the method most similar to ours. The results of Table 5.1 demonstrate that while Dex-net achieves the highest grasp success rate, our proposed method excels in minimizing deformation, with a grasp success rate close to that of Dex-net. In addition, the detection time per trial is presented, which represents the duration from inputting the depth image to detecting the grasp pose. The proposed method achieves the shortest calculation time overall. While Dex-Net 4.0 exhibits the shortest detection time when a specific gripper's width is pre-defined, it requires additional time in this experiment to identify the optimal gripper's width that maximizes the grasping performance score to ensure a high success rate. The scene images, estimated images, and examples of the grasp position detection in the experiment are presented in Figure

5.5(a).

Table 5.1: Comparison of grasping results for deformable hollow objects

Method	Grasping success [%]	Deformation	Detection time [sec / attempt]
FGE [18]	75.9	0.148	10.9
Dex-net 4.0 [10]	**85.7**	0.127	22.0
pix2stiffness [47]	76.9	0.264	16.1
Ours	83.3	**0.115**	**9.07**

5.3.3 Grasping for partially deformable objects

For scenes composed of partially deformable objects arranged in piles, we generated an appropriate grasp pose and assessed the grasping feasibility. These objects are distinctly segmented into hard and soft parts, enabling targeted detection of the optimal grasp pose on the harder sections. To facilitate this, the grasp pose detection process involves setting the deformability score to -1 for areas in the target segment that represent less than a predefined threshold of the maximum score, effectively creating contact images. The threshold for extracting the region with the highest deformability is determined by using the reciprocal of the number of predefined regions (two in this experiment) with varying hardness levels. In this case, a threshold of 50% is identified as a suitable candidate for grasping. When prior information is unavailable, automatic differentiation between regions can be achieved by analyzing the histogram of the deformability in the object region. The target object of this dissertation, a brush with a handle, presents a more complex geometry than simpler objects such as bottles, complicating the extraction of the object region and the grasping of the hard part of the handle. We generated three scenarios with six brushes each, piled in disarray, and defined the task success rate as the proportion of attempts in which the hard part was successfully grasped. The comparative analysis employed the same methods as previously—FGE, Dex-net, and pix2stiffness. The results of Table 5.2 indicated that the proposed method outperformed pix2stiffness, with a notably higher success rate when object deformability was considered. The scene

images, estimated images, and examples of the grasp position detection in the experiment are presented in Figure 5.5(b).

Table 5.2: Comparison grasping results for partially deformable objects

Method metrics	Task success [%]
FGE [18]	5.56
Dex-net 4.0 [10]	11.1
pix2stiffness [47]	61.2
Ours	**77.8**

5.3.4 Pushing away of deformable obstacles

When a specific target object is identified, if normal grasping is impeded by a nearby deformable object with relatively low deformability, a viable strategy involves grasping while compressing the obstructing deformable object without considering its structural integrity. This approach allows the detection of appropriate grasping positions in scenarios where conventional methods fail. We assume that the interfering object is elastic—like a sponge—which returns to its original shape after compression if its hardness, as determined by deformability estimation, is low. In our detection method, this strategy is implemented in the collision image by classifying areas with an arbitrarily high percentage (in this case, more than 50[%]) of the surrounding deformabilities are considered obstacles, neglecting the remaining ones). In the proposed scenario, the proposed method was compared with the traditional methods, FGE and pix2stiffness, in a task involving the retrieval of all five target objects across three separate piled scenes. The results of Table 5.3 demonstrate that our method achieved the highest success rate, largely because of the effective identification and handling of crushable obstacles. Notably, failures occur when a deformable object significantly blocks the target, crushing both the obstruction and the target object itself. To mitigate such issues, it is advisable to employ a force sensor to measure the exertion force and decide whether to halt the mid-process action. The scene images, estimated images, and examples of the grasp

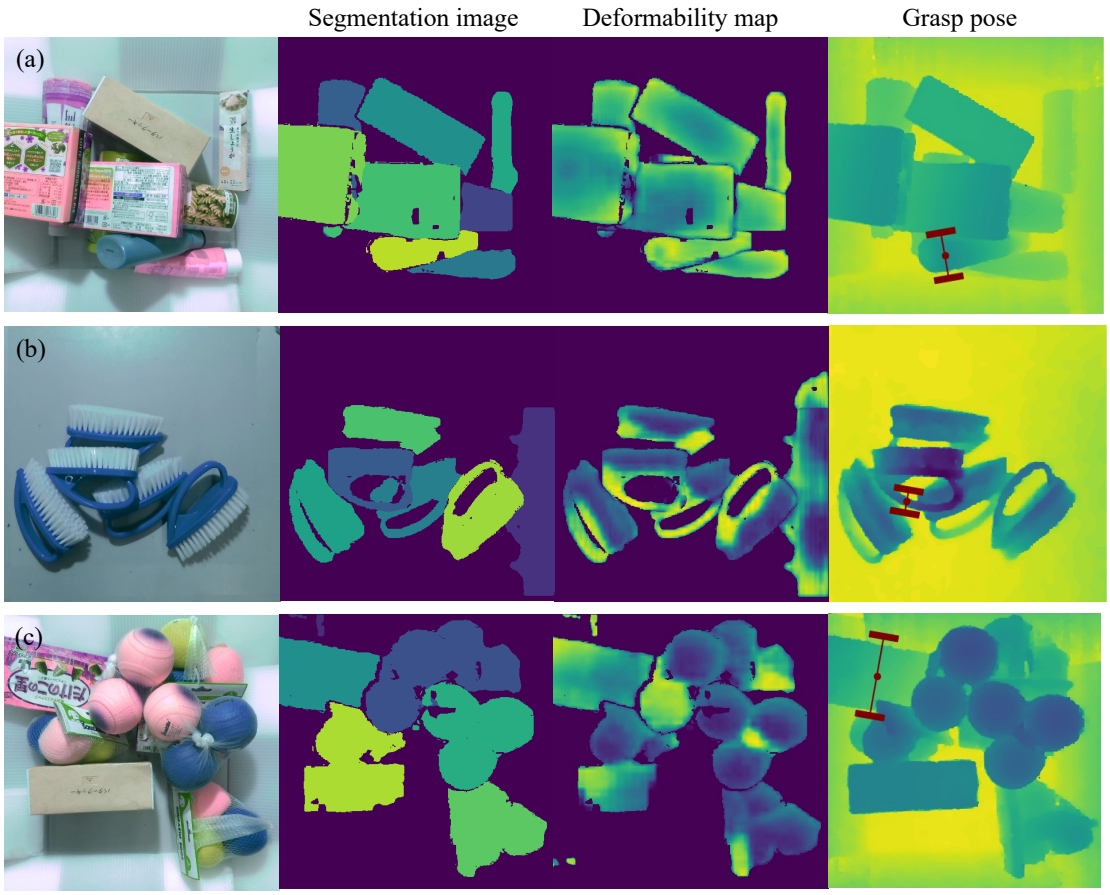


Figure 5.5: Example of deformability estimation and grasping results

position detection in the experiment are presented in Figure 5.5(c).

Table 5.3: Comparison grasping results in the task of pushing away deformable obstacles

Method	Grasping success [%]
FGE [18]	53.6
pix2stiffness [47]	53.6
Ours	**75.0**

5.4 Conclusion

In this dissertation, we propose an encoder-decoder model for deformability estimation, which assumes that the deformability of an object is related to its appearance. Based on the Mask R-CNN, we construct a model for generating a deformability map that indicates the tendency deformability in an object for each pixel, which can also generate a segmentation image. To train this model, a semiautomatic generation of image datasets via simulation was proposed, and a deformability map was created for each segment to adapt to our model. Using the results of deformability estimation with instance segmentation, we propose a grasp pose detection method from a single image that can grasp various deformable objects, thereby preventing the deformation shown in real-world experimental results.

One limitation of the proposed method is that it cannot be adapted to different materials with similar shapes. When the internal structure (filled with a specific material) differs in objects with shapes equivalent to deformable hollow objects, distinguishing between them becomes challenging. Nevertheless, both internal structures can serve as targets for preventing deformation. Moreover, our method does not aim for fully deformable objects such as cloth and does not consider deformation in the training data. Therefore, we require an estimation method that uses other modalities to consider material information.

Chapter 6

Discussion

6.1 Contributions

In this dissertation, a method is proposed for generating data and applying it to grasping, taking into account the diversity of the physical properties of objects. The framework considers softness and shape and applies it to a wide variety of shapes while achieving grasping without damaging the object. In Chapter 3, a spatial hardness map is estimated for objects with different hardness distributions depending on their shape, and grasping is achieved in a way that prevents the deformation of the object.

In the dataset that corresponds to the diversity of shapes, scanned 3D models are randomly selected and used. By collecting a large number of these, it is possible to achieve a high grasping rate for similar shapes and unknown objects. However, because there is no uniformity in the need to retain data, its quality, or the quality of the data as grasping data, it is thought that unnecessary data for learning is also used. In order to solve this problem and adapt to the diversity of shapes, Chapter 4 uses procedurally generated data, which allows achieving the same level of grasping as when using scanned 3D models. By using fractal geometry to quickly and automatically build a database with a wide variety of shapes that follow the formation principles of natural objects, it is possible to avoid the problems associated with using scanned 3D models. The ability to

manipulate shapes using parameters enables the generation of complex shapes, and we have confirmed in real-world picking that this has a significant effect when using small amounts of data.

However, Chapter 3 and Chapter 4 are only applicable to single objects, and it is difficult to handle scenes with a variety of objects, such as those found in logistics warehouses. In the case of complex scenes, it is necessary to simultaneously recognize multiple objects while also taking occlusion into account, and the framework must be realistic and feasible while keeping the computational cost down. In Chapter 5, we address this issue and enable picking in real environments using a framework that considers both softness and shape diversity at the same time. We have achieved a model that processes tasks related to softness and shape simultaneously using a single truth learning model and have accomplished a reduction in computational cost while enhancing mutual performance. We have confirmed that the proposed method has the highest success rate in three grasping scenarios. In particular, in scenes with many obstacles, more stable grasping can be achieved using an operation strategy that considers the physical properties of the object.

6.2 Open Challenges and Future Work

The limitation of this method is that it only relates to the shape and softness of objects. While there are objects where shape and softness are related, it is difficult to apply this method to objects with the same shape but different materials. In this case, it depends on the texture and feel of the object, so it is necessary to generate a new database for these and to use information other than shape, such as RGB images. Also, in the case of a large number of different types of objects, it is difficult to generate a large amount of softness databases. In this dissertation, the softness scores were assigned by humans, and the cost of scaling up is high. Therefore, a framework that enables easy annotation, such as collecting hardness evaluation values using a robot, is necessary. Additionally, LLMs and similar systems acquire physical knowledge in the form of linguistic information [69]. By effectively leveraging this knowledge and continuously updating it with real-world robot motion data, these models can be extended to a broader range of objects.

In addition, in the shape database, the shape manipulation is controlled by only one parameter. To apply it to objects with complex structures, such as articulated objects, where there remains a large discrepancy between the actual object and its appearance, it is necessary to execute surface reconstruction to smooth the object's surface and create a 3D model that is aware of the parts in order to create a large number of grasping points. And the 3D models generated by fractal geometry facilitate successful grasping by partially incorporating the shape of the real 3D object. Because they also contain shapes that cannot be collected in real-world environments, they can be leveraged to compensate for the limited availability of real data (e.g., Data augmentation).

References

[1] N. Yashiro, "Aging of the population in japan and its implications to the other asian countries," Journal of Asian Economics, vol. 8, no. 2, pp. 245–261, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1049007897900191

[2] L. Kugler, "Addressing labor shortages with automation," Commun. ACM, vol. 65, no. 6, p. 21–23, May 2022. [Online]. Available: https://doi.org/10.1145/3530687

[3] Y. Domae, A. Noda, T. Nagatani, and W. Wan, "Robotic general parts feeder: Bin-picking, regrasping, and kitting," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 5004–5010.

[4] H. Fujiyoshi, T. Yamashita, S. Akizuki, M. Hashimoto, Y. Domae, R. Kawanishi, M. Fujita, R. Kojima, and K. Shiratsuchi, "Team c2m: Two cooperative robots for picking and stowing in amazon picking challenge 2016," Advances on Robotic Item Picking, pp. 101–112, 2020.

[5] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 1, pp. 172–188, 2018.

[6] M. Pfanne, M. Chalon, F. Stulp, H. Ritter, and A. Albu-Schäffer, "Object-level impedance control for dexterous in-hand manipulation," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2987–2994, 2020.

[7] Z. Li and S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," IEEE Journal on Robotics and Automation, vol. 4, no. 1, pp. 32–44, 1988.

[8] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," The International Journal of Robotics Research, vol. 34, no. 4-5, pp. 705–724, 2015. [Online]. Available: https://doi.org/10.1177/0278364914549607

[9] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," International Journal of Robotics Research, vol. 37, no. 4-5, pp. 421–436, 2018.

[10] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," Science Robotics, vol. 4, no. 26, p. eaau4984, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aau4984

[11] H.-S. Fang, M. Gou, C. Wang, and C. Lu, "Robust grasping across diverse sensor qualities: The graspnet-1billion dataset," The International Journal of Robotics Research, 2023.

[12] K. opalakrishnan and K. Goldberg, "D-space and deform closure grasps of deformable parts," The International Journal of Robotics Research, vol. 24, no. 11, pp. 899–910, 2005. [Online]. Available: https://doi.org/10.1177/0278364905059055

[13] Y.-B. Jia, F. Guo, and H. Lin, "Grasping deformable planar objects: Squeeze, stick/slip analysis, and energy-based optimalities," The International Journal of Robotics Research, vol. 33, no. 6, pp. 866–897, 2014. [Online]. Available: https://doi.org/10.1177/0278364913512170

[14] J. Xu, M. Danielczuk, J. Ichnowski, J. Mahler, E. Steinbach, and K. Goldberg, "Minimal work: A grasp quality metric for deformable hollow objects," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 1546–1552.

[15] I. Huang, Y. Narang, C. Eppner, B. Sundaralingam, M. Macklin, R. Bajcsy, T. Hermans, and D. Fox, "Defgraspsim: Physics-based simulation of grasp outcomes for

3d deformable objects," IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 6274–6281, 2022.

[16] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 998–1005.

[17] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor," in 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 1724–1731.

[18] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1997–2004.

[19] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," The International Journal of Robotics Research, vol. 27, no. 2, pp. 157–173, 2008.

[20] L. Wang, Y. Xiang, A. Mousavian, and D. Fox, "Goal-auxiliary actor-critic for 6d robotic grasping with point clouds," in 5th Annual Conference on Robot Learning, 2021.

[21] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," Robotics and Automation Letters, 2020.

[22] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A large-scale grasp dataset based on simulation," in 2021 IEEE Int. Conf. on Robotics and Automation, ICRA, 2020.

[23] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 13 438–13 444.

[24] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23–30.

[25] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 13 452–13 458.

[26] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11 441–11 450.

[27] H. Zhang, D. Yang, H. Wang, B. Zhao, X. Lan, J. Ding, and N. Zheng, "Regrad: A large-scale relational grasp dataset for safe and object-specific robotic grasping in clutter," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2929–2936, 2022.

[28] Y. Li, T. Kong, R. Chu, Y. Li, P. Wang, and L. Li, "Simultaneous semantic and collision learning for 6-dof grasp pose estimation," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 3571–3578.

[29] M. Haoxiang and D. Huang, "Towards scale balanced 6-dof grasp detection in cluttered scenes," in Conference on Robot Learninsg (CoRL), 2022.

[30] R. Matsumura, Y. Domae, W. Wan, and K. Harada, "Learning based robotic bin-picking for potentially tangled objects," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 7990–7997.

[31] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3634–3642.

[32] R. Lu, R. Van Beers, W. Saeys, C. Li, and H. Cen, "Measurement of optical properties of fruits and vegetables: A review," Postharvest Biology and Technology, vol. 159, p. 111003, 2020.

[33] M. Fujiwara, K. Nakatsuma, M. Takahashi, and H. Shinoda, "Remote measurement of surface compliance distribution using ultrasound radiation pressure," in 2011 IEEE World Haptics Conference, 2011, pp. 43–47.

[34] K. Tanaka, Y. Mukaigawa, T. Funatomi, H. Kubo, Y. Matsushita, and Y. Yagi, "Material classification using frequency-and depth-dependent time-of-flight distortion," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2740–2749.

[35] A. Meka, M. Maximov, M. Zollhofer, A. Chatterjee, H. Seidel, C. Richardt, and C. Theobalt, "Lime: Live intrinsic material estimation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6315–6324.

[36] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017.

[37] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," IEEE Robotics Automation Magazine, vol. 22, no. 3, pp. 36–52, 2015.

[38] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[39] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsanit, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 13 142–13 153.

[40] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel, "Domain randomization and generative models for robotic grasping," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 3482–3489.

[41] D. Morrison, P. Corke, and J. Leitner, "Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4368–4375, 2020.

[42] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, 2014, p. 2672–2680.

[43] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5967–5976.

[44] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[45] T. Rott Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," in Computer Vision and Pattern Recognition (CVPR), 2021.

[46] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5143–5153.

[47] K. Makihara, Y. Domae, I. G. Ramirez-Alpizar, T. Ueshiba, and K. Harada, "Grasp pose detection for deformable daily items by pix2stiffness estimation," Advanced Robotics, vol. 36, no. 12, pp. 600–610, 2022. [Online]. Available: https://doi.org/10.1080/01691864.2022.2078669

[48] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[50] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[51] F. Li, H. Zhang, H. xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," 2022.

[52] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," arXiv preprint arXiv:2211.07636, 2022.

[53] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al., "Internimage: Exploring large-scale vision foundation models with deformable convolutions," arXiv preprint arXiv:2211.05778, 2022.

[54] H. Kataoka, K. Okayasu, A. Matsumoto, E. Yamagata, R. Yamada, N. Inoue, A. Nakamura, and Y. Satoh, "Pre-training without natural images," International Journal of Computer Vision (IJCV), 2022.

[55] H. Kataoka, E. Yamagata, K. Hara, R. Hayashi, and N. Inoue, "Spatiotemporal initialization for 3d cnns with generated motion patterns," 2022.

[56] R. Yamada, R. Takahashi, R. Suzuki, A. Nakamura, Y. Yoshiyasu, R. Sagawa, and H. Kataoka, "Mv-fractaldb: Formula-driven supervised learning for multi-view image recognition," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 2076–2083.

[57] R. Yamada, H. Kataoka, N. Chiba, Y. Domae, and T. Ogata, "Point cloud pre-training with natural 3d structures," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 21 283–21 293.

[58] R. Yamada, K. Hara, H. Kataoka, K. Makihara, N. Inoue, R. Yokota, and Y. Satoh, "Formula-supervised visual-geometric pre-training," in Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part XXII. Berlin, Heidelberg: Springer-Verlag, 2024, p. 57–74. [Online]. Available: https://doi.org/10.1007/978-3-031-72670-5_4

[59] S. Takashima, R. Hayamizu, N. Inoue, H. Kataoka, and R. Yokota, "Visual atoms: Pre-training vision transformers with sinusoidal waves," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023, pp. 18 579–18 588.

[60] R. Tadokoro, R. Yamada, K. Nakashima, R. Nakamura, and H. Kataoka, "Primitive geometry segment pre-training for 3d medical image segmentation," in 34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023. BMVA, 2023. [Online]. Available: https://papers.bmvc2023.org/0152.pdf

[61] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, C. Hallacy, B. Mann, A. Radford, A. Ramesh, N. Ryder, D. M. Ziegler, J. Schulman, D. Amodei, and S. McCandlish, "Scaling laws for autoregressive generative modeling," 2020. [Online]. Available: https://arxiv.org/abs/2010.14701

[62] OpenAI, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[63] Gemini Team Google, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[64] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave,

and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[65] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme Ruiz, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. V. Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. Collier, A. A. Gritsenko, V. Birodkar, C. N. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetic, D. Tran, T. Kipf, M. Lucic, X. Zhai, D. Keysers, J. J. Harmsen, and N. Houlsby, "Scaling vision transformers to 22 billion parameters," in Proceedings of the 40th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 7480–7512. [Online]. Available: https://proceedings.mlr.press/v202/dehghani23a.html

[66] Gemma Team, "Gemma: Open models based on gemini research and technology," 2024. [Online]. Available: https://arxiv.org/abs/2403.08295

[67] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in NeurIPS, 2023.

[68] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 9493–9500.

[69] Y. Wang, J. Duan, D. Fox, and S. Srinivasa, "NEWTON: Are large language models capable of physical reasoning?" in Findings of the Association for Computational Linguistics: EMNLP 2023, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9743–9758. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.652/

[70] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024.

[71] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in The Twelfth International Conference on Learning Representations, 2024. [Online]. Available: https://openreview.net/forum?id=IEduRUO55F

[72] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," in arXiv preprint arXiv:2212.06817, 2022.

[73] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in Proceedings of The 7th Conference on Robot Learning, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 2165–2183. [Online]. Available: https://proceedings.mlr.press/v229/zitkovich23a.html

[74] Open X-Embodiment Collaboration, "Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0," in 2024 IEEE

International Conference on Robotics and Automation (ICRA), 2024, pp. 6892–6903.

[75] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 653–660.

[76] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in Proceedings of Robotics: Science and Systems, Delft, Netherlands, 2024.

[77] J. Hejna, C. A. Bhateja, Y. Jiang, K. Pertsch, and D. Sadigh, "Remix: Optimizing data mixtures for large scale imitation learning," in 8th Annual Conference on Robot Learning, 2024. [Online]. Available: https://openreview.net/forum?id=fIj88Tn3fc

[78] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," 2024. [Online]. Available: https://arxiv.org/abs/2410.24164

[79] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," Autonomous Robots, vol. 38, pp. 1–12, 2015.

[80] M. Danielczuk, J. Xu, J. Mahler, M. Matl, N. Chentanez, and K. Goldberg, "Reach: Reducing false negatives in robot grasp planning with a robust efficient area contact hypothesis model," Int. S. Robotics Research (ISRR), 2019.

[81] B. Thach, A. Kuntz, and T. Hermans, "Deformernet: A deep learning approach to 3d deformable object manipulation," 2021. [Online]. Available: https://arxiv.org/abs/2107.08067

[82] T. N. Le, J. Lundell, F. J. Abu-Dakka, and V. Kyrki, "Deformation-aware data-driven grasp synthesis," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3038–3045, 2022.

[83] "Blender," https://www.blender.org/.

[84] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[85] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[86] K. G. Lore, K. K. Reddy, M. Giering, and E. A. Bernal, "Generative adversarial networks for spectral super-resolution and bidirectional rgb-to-multispectral mapping," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 926–933.

[87] M. M. de Oliveira Neto, B. Bowen, R. McKenna, and Y.-S. Chang, "Fast digital image inpainting," in IASTED International Conference on Visualization, Imaging and Image Processing, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:1803602

[88] "Universal Robotics," https://www.universal-robots.com/.

[89] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," The International Journal of Robotics Research, vol. 41, no. 7, pp. 690–705, 2022. [Online]. Available: https://doi.org/10.1177/0278364919868017

[90] I. Huang, Y. Narang, R. Bajcsy, F. Ramos, T. Hermans, and D. Fox, "Defgraspnets: Grasp planning on 3d fields with graph neural nets," in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 5894–5901.

[91] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

[92] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 7283–7290.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[94] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944.

[95] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.

[97] R. Araki, T. Yamashita, and H. Fujiyoshi, "ARC2017 RGB-D Dataset for Object Detection and Segmentation," in Late Breaking Results Poster on International Conference on Robotics and Automation, 2018.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Kensuke Harada, for accepting me as a student in the Robotic Manipulation Laboratory and guiding me throughout my four years as a Ph.D. student. His invaluable advice, continuous support, and patience have been instrumental in helping me reach this stage. I am also profoundly grateful to my Ph.D. advisor, Dr. Yukiyasu Domae from the National Institute of Science and Technology (AIST), for his exceptional supervision, support, and mentorship throughout my doctoral studies. Over the past five years, he has guided me in my research, provided invaluable advice on my career, and actively supported me in building my academic achievements, for which I am truly thankful.

Furthermore, I would like to express my appreciation to Dr. Hirokatsu Kataoka, Dr. Ryo Hanai, and Dr. Ixchel Ramirez-Alpizer for their insightful suggestions on my research at AIST, to Mr. Ueshiba for providing valuable technical advice on experimental systems, and to Mr. Ryosuke Yamada at Tsukuba University for the many valuable discussions.

I also sincerely thank Associate Professor Weiwei Wan, Assistant Professor Keisuke Koyama, and Assistant Professor Takuya Kiyokawa for their guidance and suggestions on my research. I am genuinely grateful to all the members of the Harada Laboratory whom I have had the pleasure of working with over the past four years—your support and camaraderie have made this journey all the more meaningful.

I would like to once again express my sincere gratitude to Associate Professor Satoshi Makita from the Fukuoka Institute of Technology for his invaluable supervision over three years, from my fifth year at the National Institute of Technology, Sasebo College, to completing my bachelor's degree. I am also deeply indebted to Professor Yasumichi Aiyama from the University of Tsukuba for his dedicated guidance throughout my two years in the master's program.

Most importantly, none of this would have been possible without the unwavering support of my family. My heartfelt thanks go to my parents, Satoshi and Harumi, for their unconditional love, guidance, and support, without which I would not be where I am today. I am also immensely grateful to my older sister, Misato, and my extended family for their encouragement. To my partner, Mirei, I cannot thank you enough for your patience, support, and unwavering belief in me during these challenging times. I am truly fortunate to have you by my side.

Publications

Journal Papers:

1. K. Makihara, Y. Domae, R. Hanai, I. G. Ramirez-Alpizar, H. Kataoka and K. Harada, ”Deformability-based grasp pose detection from a visible image,” in IEEE Access, 2024.
2. K. Makihara, Y. Domae, I. G. Ramirez-Alpizar, T. Ueshiba and K. Harada(2022). Grasp pose detection for deformable daily items by pix2stiffness estimation. Advanced Robotics, 36(12), 600–610.
3. K. Makihara, R. Yamada, Y. Domae, H. Kataoka and K. Harada, ”Grasp datasets based on Fractal Geometry for robotic grasping,” in submission of IEEE Robotics and Automation Letters, 2024.

International Conference Papers (with peer-review):

1. R. Yamada, K. Hara, H. Kataoka, K. Makihara, N. Inoue, R. Yokota and Y. Satoh, ”Formula-Supervised Visual-Geometric Pre-training”, European Conference on Computer Vision, 2024.

Local Conference papers (without peer-review):

1. 牧原 昂志, 山田 亮佑, 堂前 幸康, 片岡 裕雄, 原田 研介, ”数式ドリブン教師あり学習を用いた把持位置検出,” 第26回 画像の認識・理解シンポジウム, 2023.
2. 牧原 昂志，堂前 幸康，片岡 裕雄，ラミレス イクシェル，原田 研介, ”アピアランスからの物体柔軟性推定に基づく把持位置検出,” 第22回システムインテグレーション部門講演会(SI2021)予稿集, 2021.
3. 牧原 昂志, 堂前 幸康, Ramirez Alpizar Georgina Ixchel, 植芝 俊夫：”pix2stiffnessによる柔軟物体の把持位置検出”, 第27回画像センシングシンポジウム, 2021年6月.

Awards:

1. Best Oral Presentation (優秀講演賞), 第22回計測自動制御学会システムインテグレーション部門講演会 (SI2021).