Learning-based Object Manipulation with Collapse Prediction in Clutter

TOMOHIRO MOTODA

MARCH 2023

Learning-based Object Manipulation with Collapse Prediction in Clutter

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

TOMOHIRO MOTODA

MARCH 2023

Learning-based Object Manipulation with Collapse Prediction in Clutter

Tomohiro Motoda

Osaka University 2023

Abstract

This dissertation focuses on the leaning-based manipulation of retrieving an object from a cluttered environment. Robotics are crucial for the automation of various tasks, such as transportation, assembly, and pick-and-place. However, it is difficult for robots to accurately locate and manipulate a specific object within a box containing multiple objects in various orientations and configurations. Therefore, it is necessary for the robot to predict how objects will behave and plan actions accordingly to safely extract objects from a pile without having them fall or scatter. The proposed methods in this dissertation allow for a general-purpose manipulator equipped with a standard gripper to carefully and successfully manipulate objects in a physical scene.

This dissertation presents the following contributions. First, we studied the problem of picking an object from clutter and proposed an action planning model based on a convolutional neural network, focusing on playing the game "Yamakuzush" as a case study. The objective is to select the correct piece from a random pile of Shogi pieces. Second, we proposed a bimanual manipulation approach based on collapse predictions for shelf picking and insertion tasks. Our proposed learning-based predictor (the collapse predictor) can detect the risk of collapsing shelves when the robot extracts the target object. Furthermore, we presented a method for shelf replenishment for bimanual manipulation based

on object arrangements and the collapse predictor. Finally, we assumed support relations among objects in a cluttered environment to analyze cluttered scenes. We designed a multi-stage motion planner based on the support relations to enable the robot to tackle the picking task without causing objects to collapse. Therefore, I believe that empirically acquired intuitive analysis is essential for human behavior in everyday life and aim to construct an algorithm for selecting appropriate robot actions by closely observing the target objects.

Table of Contents

Chapter 1

Introduction

1.1 Background and Motivation

Humans unconsciously grasp and use tools. For example, in everyday life, we might pull a cookbook from a bookshelf, pick food from a refrigerator, or take a plate from a cupboard. In manufacturing, workers might take a certain part from a container filled with parts and sort them for supply. In a warehouse, workers might pick goods for delivery from a warehouse shelf. Conversely, manipulating an object in a cluttered environment, such as a container or shelf, is a fundamental task. In recent years, automation through machinery has been considered a method to replace or assist human labor. Robotic manipulation has the potential to perform pick-and-place tasks [1,2].

Robot manipulators have several advantages over human workers. They can repeat a sequential series of tasks with high speed and precision and can handle monotonous tasks for long periods of time. Robot manipulation has been actively discussed in recent decades as a method to address issues, such as labor shortages, cost reduction, and productivity improvement. Furthermore, in modern manipulation, robots are used not only for mechanical and repetitive tasks but also for flexible tasks that require them to work autonomously in various situations [3,4,5,6]. One advantage of using automated manipulation in a cluttered environment is that it eliminates the need for a workforce and specialized machinery for rearranging work pieces, significantly improving productivity.

However, it is more difficult to handle objects that are randomly placed in boxes or on shelves than those that are aligned. It causes them to be entangled and collide among objects if the aim is only to extract the desired product, resulting in objects falling from the shelves or scattering the workspace. In such cases, the robot must pick up the fallen items and return them to their original positions, which may result in damage and other losses of tools, products, and household commodities, which should be avoided in practical use. Therefore, to pick any object correctly in such environments, a robotic system should solve several challenges, such as scene analyses that include physical phenomena, object identification from a dense container, and motion planning to avoid damaging the object.

This dissertation focuses on learning object manipulation, particularly robotic manipulation, in general environments. Machine learning has changed the world in the last few years with its breakthrough innovations. In robotics, recent robotic systems have addressed learning-based methods to complete a task, developing integrated systems in various fields [7] (e.g., grasp planning, garment folding tasks, and automation based on probabilistic action planning) [8]. However, it is difficult for robots and other machinery in data-driven learning models to flexibly interpret various physical scenes with the same accuracy as humans without any preconditions attached and to determine the next action. Therefore, it is crucial to develop robotic action plans that can analyze the workspace and determine appropriate manipulations based on the situation to solve practical problems in cluttered environments.

This dissertation presents robotic action planning to extract the desired objects from a pile in the correct manner or order to avoid scattering the objects. Therefore, the main research topics are twofold. First, deep neural networks improve observation and extraction in cluttered environments by focusing on the object arrangement or physical phenomena. Second, robotic action planning is proposed to safely extract an object (to avoid collapsing/falling) based on scene analyses.

1.2 Objectives

The objectives of this dissertation are as follows:

1. To develop a learning-based method for extracting and manipulating objects from cluttered environments. We focus specifically on extracting a desired object without causing stacked objects to scatter or fall. The aim is to create an algorithm that can be used to safely manipulate an object in a cluttered environment using a robotic manipulator and to apply it in real-world scenarios.
2. To integrate scene analysis and action planning to correctly manipulate an object within clutter. We focus on the states of objects within cluttered environments, such as detecting object poses, arrangements, and physical relationships between objects. Furthermore, we predict the outcome of an action to avoid the risk of potential failure based on a captured image and to enable the selection of the appropriate action.

1.3 Dissertation Outline

The outline of this dissertation is as follows:

Chapter 3 focuses on the Yamakuzushi board game, in which you must extract a Shogi piece (called a Japanese chess piece) with only one finger while keeping the pile from collapsing. Furthermore, we emphasized that human operators can choose an object from a pile of pieces. We also used a convolutional neural network (ConvNet) to detect the appropriate objects, which can be extracted without collisions, from a stack of Shogi pieces, enabling the robot to safely extract a piece. We obtained a basic insight into the effectiveness of learning-based models in solving the problem of manipulation in cluttered environments.

In Chapter 4, we proposed a learning-based collapse prediction that can safely extract a single object while supporting other objects. The proposed method is designed to enable the bimanual manipulator to tackle object-picking tasks from a cluttered shelf. Finally, a bimanual manipulator is used to evaluate the robustness of the proposed method for safe object retrieval in real-world experiments.

Chapter 5 discusses a bimanual manipulation planner to address the problem of placing a product on a shelf, such as shelf replenishment. We proposed an action plan to determine the best next action from corrections or replenishment based on the object arrangements. Furthermore, we used a collapse predictor to predict the object collapse with ConvNet while avoiding the object collapse and manipulating an object. Finally, we discussed the proposed method to place an object on a real-world shelf full of boxes.

In Chapter 6, a multi-step object extraction strategy is proposed to safely remove the stacked objects. Furthermore, the support relations of the objects in the clutter were expressed graphically. In this chapter, the inference of support relations and the object extraction order were evaluated based on the accuracy of the scene analysis. The proposed method is shown to enable the manipulator to pick objects more safely compared to the method in previous chapters.

Chapter 7 summarizes the achievements of the methods proposed in this dissertation.

Chapter 2

Literature Review

Robotic manipulation in cluttered environments has recently received increased attention owing to the use of various robots in the logistics and retail domains. The review of the related works particularly focuses on three aspects: manipulation in cluttered environments, deep learning for grasping and picking, and physical relations between objects.

2.1 Manipulation in Cluttered Environments

A picking strategy must be created for a robot to pick objects. The robot's picking strategy depends on the (1) hardware of the robot's hand, (2) shape, (3) application, and (4) placement of the objects to be picked by the robot. However, it is difficult to apply existing methods because target objects have various shapes and sizes and are placed randomly in a container. For example, the Amazon Picking Challenge (later known as the Amazon Robotic Challenge) is a robotics competition that solves a domain challenge in real-world scenes and encourages autonomous robotic manipulations in cluttered environments [9,10,11,12]. Several researchers address robotic object manipulation [13, 14, 15, 16] and visual recognition [17, 18] in cluttered environments. In this chapter, we mainly discuss analytical manipulation methods without the deep learning discussed below.

Many proposals have performed bin-piking [19,20,21,22], shelf/container picking [23, 24, 25, 26], and toy games related to object manipulation in cluttered environments [27, 28, 29, 30, 31]. Temtsin et al. [32] ranked each object using a measure based on the geometrical relationships of objects and extracted an object with a high rank. Mojtahedzadeh et al. [24] and Wu et al. [33] proposed methods to learn the motion of robots in stacked or scattered environments. Few researchers have achieved clutter manipulation using a partially observable Markov decision process (POMDP) [15, 16, 31, 34, 35, 36]. Furthermore, Eidenberger et al. proposed a POMDP-based method for determining the optimal sensor location [15, 16]. Kim et al. proposed a method for the effective grasping of posture selection from a known database while using POMDPs to plan the observations [35]. Hsiao et al. [36] used this framework to search from the state of contact with the external environment and work on grasp action planning. Furthermore, Nagata et al. [37] defined the grasp patterns as being linked to the surfaces of target objects, with a focus on the object shapes/gripper, and proposed a dexterous strategy for sliding a top object or tilting an aligned object from a complex environment to extract the target object. Domae et al. computed the grasp configurations of robotic grippers based only on 2D images to prevent a gripper from colliding with obstacles [21]. Other approaches assumed that objects were placed side by side on a shelf and relocated to pick the target object. Dogar et al. [38] pushed obstacles to reach a target in cluttered environments. Lee et al. [25] and Nam et al. [26] relocated obstacles to retrieve a target object from clutter. Huang et al. planned a sequence of pick-and-place actions to search for the occluded target [39]. However, if objects are piled on a shelf, an object cannot easily be pushed and slid.

In Chapter 3, we use POMDPs to enhance the efficiency of picking from randomly stacked objects for both observation and manipulation action planning. Furthermore, the human manipulation characteristics are learned to select which object to manipulate based on ConvNets. The improving observation and manipulation actions are selected around the manipulated object to increase the success rate of object picking.

2.2 Deep Learning for Grasping and Picking

The deep learning-based method has been extensively researched, and recent research has also contributed to the development of high-precision bin picking [40, 41, 42, 43, 44]. These methods use deep neural networks to detect the best grasp pose or point from visual information and perform high-level picking tasks in various scenarios. Levine et al. proposed an end-to-end approach by using a deep neural network trained by real-world experiments [45]. Mahler et al. built a large-scale dataset, Dex-Net, which is simulated in various 3D models, and predicted the grasp poses based on a convolutional neural network (ConvNet) [42]. Zeng et al. developed a system for specific target retrieval by pick-and-place with multiple ConvNets to address a wide range of object categories in cluttered environments [18]. Matsumura et al. adapted ConvNet to predict object entanglement [46]. The picking systems adopted in [18, 47] used a learning-based grasp detection and action decision model to handle the difficulty involved in picking a specific target from a complex scene. Harada et al. constructed a discriminator to determine the success or failure of picking objects from the pile using a random forest [48, 49]. Recent studies involving shelf replenishment tasks include refs. [50, 51]. The first proposed method for planning manipulation tasks is executed using reactive control. The second proposed knowledge-based autonomous object manipulation method uses implicit failure recovery. These approaches have improved dexterity but do not solve the problem of manipulating objects while avoiding neighboring objects from collapsing.

Humans can predict the outcome of an action and manipulate objects to prevent task failures or product damage. Similarly, previous approaches in robotics have evaluated each desired action with scene understanding to ensure safe and reliable results [13,32]. However, recent studies have been performed to assume future object states using learning models [52]. Janner et al. [53] presented a framework for learning object-oriented representations for physical scene understanding from image observations to predict the object state transition per time lapse. Magassouba et al. [54] predicted the risk of collision from an RGB-D image before the placement of an object. By contrast, Chapter 6 considers the support relationships among stacked objects and presents a multi-step extraction plan to extract the target object. Our method provides a safe extraction process that prevents the fall of neighboring objects.

2.3 Physical Relations Among Objects

Object detection is an integral component of shelf manipulation. Learning-based object detection has been widely investigated in robotics [55], and its accuracy tends to be related to successful robotic manipulations. Goldman et al. [56] provided a network architecture for identifying each object in a dense display. Asaoka et al. [57] proposed a method that groups organized objects in an image and identifies the arrangement pattern of each group. However, these methods assume that the objects are properly stored on a shelf. In Chapter 5, this study categorizes them into disorganized and organized objects, considering that objects on the shelf are cluttered.

Although the grasp detection algorithms for robotic grasping have achieved significant progress, some of these methods assume the grasp of a single-isolated object. Furthermore, a robot should sufficiently understand the cluster to interpret various properties included in an image in a cluttered environment. These properties include the geometrical, spatial [58,59,60,61], and linguistic [62] relations among objects. Zhang et al. proposed a visual manipulation relationship network to address the grasping order of vertically stacked objects [63, 64] and considered the visual relation of object overlapping. Recently, datasets such as the visual manipulation relationship dataset [63] and the relational grasp dataset [65] have been proposed to build inference models for such relationships. However, these relationships only show the geometrical relationships among objects and cannot be extended to a more general situation of clutter.

The support relations among objects have been obtained by analyzing geometric and spatial relationships [24, 66, 67, 68, 69]. Panda et al. extracted the geometrical properties of objects from images and assumed support relations [66, 67]. Mojtahedzadeh et al. [24] estimated the physical interactions between objects using 3D visual perception and machine learning. Kartmann et al. extracted physically plausible support relations using primitive shapes [68]. Grotz et al. [69] extracted physically plausible support relations between objects from point clouds to predict the action effects of picking approaches that consider support relations. Paus et al. [70] assumed the relations in probabilistic representation, including uncertainty in shapes and poses. In support of relation detection, the related works use the simple primitive, which requires preprocessing and pose estimations to approximately predict the scene and restricts real-world use. Conversely, Chapter 6 only uses depth images to predict the object collapse and to assume the support relations among objects without depending on the shapes and numbers of objects.

Chapter 3

Robotic Action/Observation Planning for Playing Yamakuzushi based on Human Motion

3.1 Introduction

In our daily lives, we quickly pick an item from a pile. For example, we look for a book and take it from a shelf, and when shopping, we pick fresh vegetables from a container. However, a robot must detect overlapping objects and manipulate them with the correct strategy to perform similar actions. Therefore, picking is essential to solving the problem of manipulating objects in a pile as humans select the appropriate object.

Manipulating objects in cluttered environments makes it difficult to estimate the object's pose and state because of the occlusion. A human can generally select and retrieve an object while avoiding collisions with other objects. However, a robot must select the appropriate action under 3D recognition from multiple viewpoints. In our study, we address the task of manipulating objects in a cluttered environment by a robotic manipulator for a picking game known as "Yamakuzushi," (Figure 3.1). In this game, the player extracts a Shogi piece from a pile by sliding the pieces with a single finger. Yamakuzushi is suitable as a benchmark for manipulating objects in a stack because it includes many tasks, such as accurate object recognition and manipulating objects without objects collapsing.

In this chapter, we propose a two-step method for selecting objects based on human operations and planned observation. We implement a robot system in-

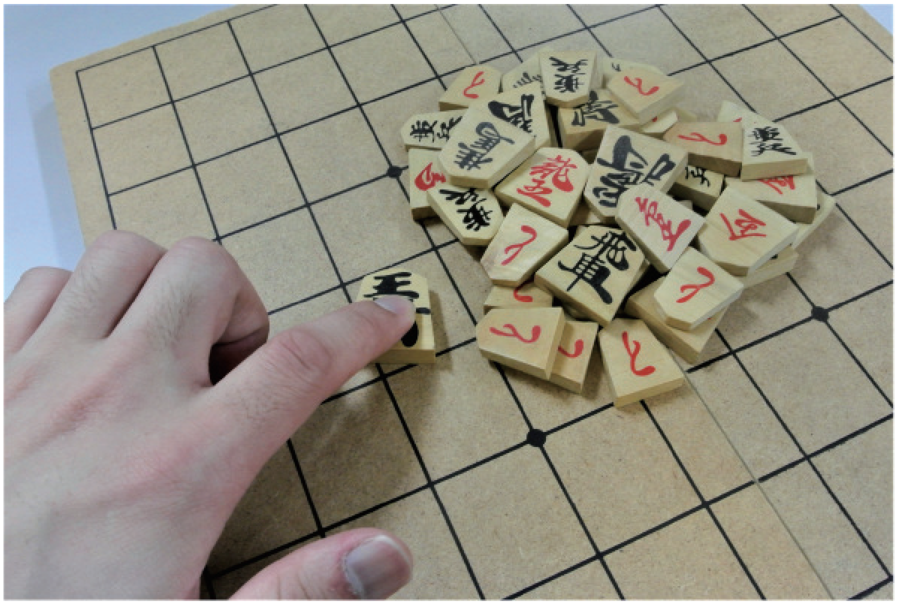


Figure 3.1: Yamakuzshi game. Sliding out a piece of shogi (Japanese chess) from a board with only one finger.

tegrated with our proposed method for robotic experiments. First, a convolutional neural network (ConvNet) is used to select target objects from depth images to determine the objects that should be safely removed from a pile. Furthermore, the environment of the selected object is observed, and actions for manipulation are planned. The problem is formulated as a partially observable Markov decision process (POMDP) to plan actions, ensure sufficient observations, and retrieve the target object.

3.2 Overview

There are two major steps in the proposed system. Figure 3.2 illustrates the experimental setup. One arm of the robot is equipped with a 3D depth sensor [71] for observation, and the other arm is equipped with a tool to retrieve a Shogi

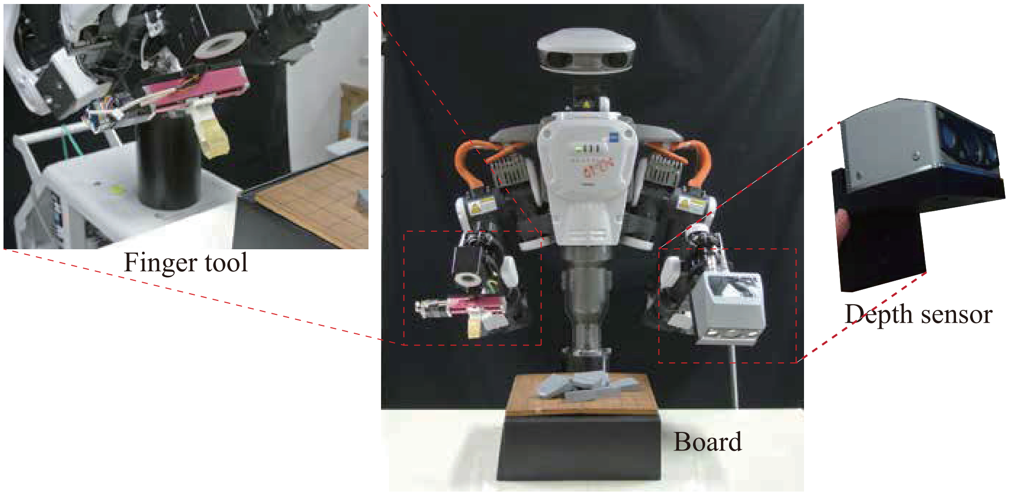


Figure 3.2: Experimental setup. We use a robotic bimanual manipulator, Nextage [72]. One arm of the robot is equipped with a 3D depth sensor [71] for observation, and the other arm is equipped with a tool.

piece (Figure 3.6) from a pile. The overall system framework is shown in Figure 3.3.

We focus on the fact that humans intuitively judge which objects they can easily retrieve from a pile based on their experience. Therefore, the robot first selects an object to manipulate from a pile using ConvNet that predicts an object that can be safely retrieved from randomly stacked objects. Furthermore, we formulate robotic action planning as POMDP. Additionally, the viewpoint is selected to minimize occlusions, and a robot can correctly estimate the poses of an object from iterative observations. When the robot accurately estimates an object's pose, it retrieves the object by sliding it out of the board. The following sections describe the details of two-step motion planning.

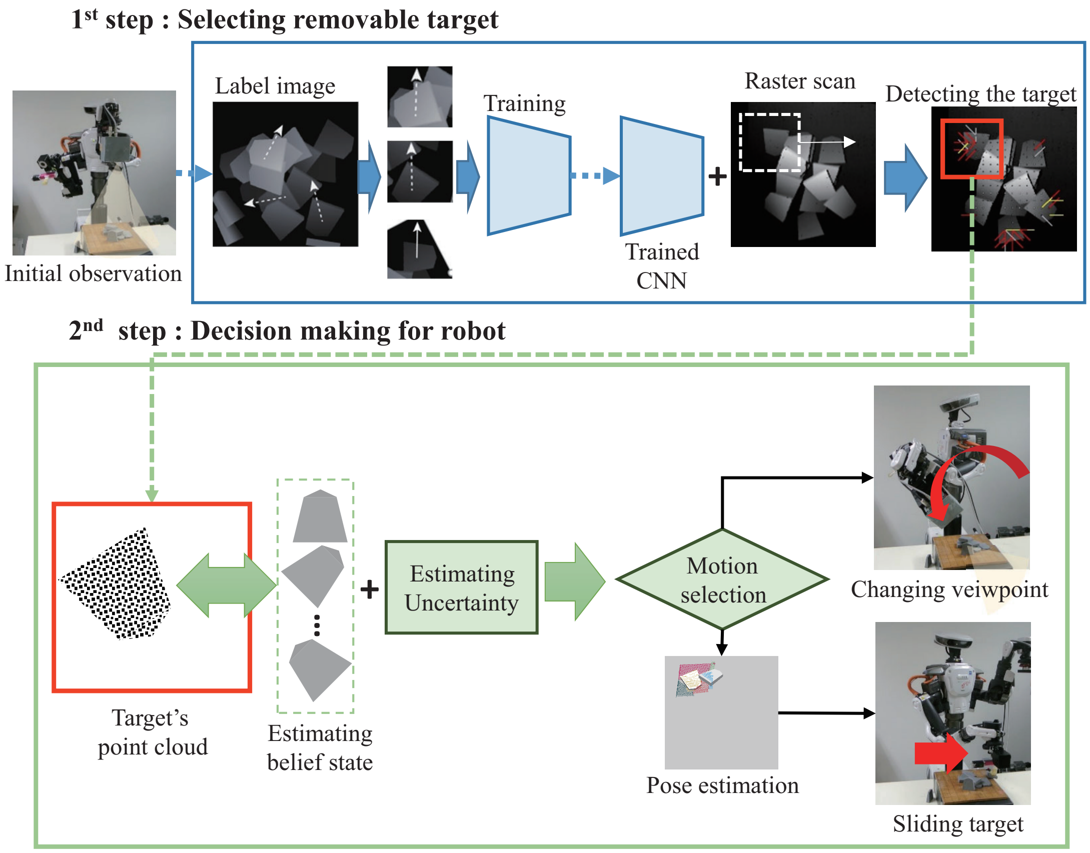


Figure 3.3: Our proposed system for the Yamakuzushi game.

3.3 Target Object Detection

A method for selecting objects by ConvNet that finds the appropriate target object from a depth image is described. Furthermore, we first describe the dataset used for training and then explain the network architecture.

3.3.1 Network architecture

This study uses learning-based predictions to select the object to be manipulated for this action. To predict whether or not we are able to retrieve an object from a pile, the architecture of ConvNet is shown in Figure 3.4, which consists of four convolutional layers (Conv. 1-4) and two fully connected layers with 1024 nodes (Fully conn. 5 and Fully conn. 6). Each layer uses the rectified linear unit (ReLU) function to avoid the loss of gradient points.

$$f(x) = \max(x, 0) \tag{3.1}$$

The softmax function is used to convert the outputs to probabilities. For classification, p_{y_1} (success) and $p_{y_2} = 1 - p_{y_1}$ (failure) are described by the following softmax functions.

$$p_{y_i} = \frac{exp(y_i)}{\sum_{j=1}^{2} \exp(y_j)} \tag{3.2}$$

The input is a 227 × 227 depth image. The output is a binary vector indicating success or failure. In this study, the input depth image is created by cropping and rotating the scene image. Our architecture is built under the condition that the sliding is always upwards. Note that we collect datasets based on human manipulation characteristics, as described later in Section 3.3.2. Further, this study is aimed toward a model that can predict the probability of success in sliding a target object without collisions with surrounding objects.

To detect the optimum target object, we use a raster scan with a fixed size and orientation of a rectangular window. In raster scanning, 100 rectangular boxes are allocated equally to 10 locations in the image, horizontally and vertically, and eight candidates for sliding orientations; for example, we have 800 ($8 \times 10 \times 10$) candidates in the depth images. Among these candidate actions, the one with the highest probability in our ConvNet is selected. The rectangular area that includes this candidate is the target area for the search and verification of operations, as described in later sections.

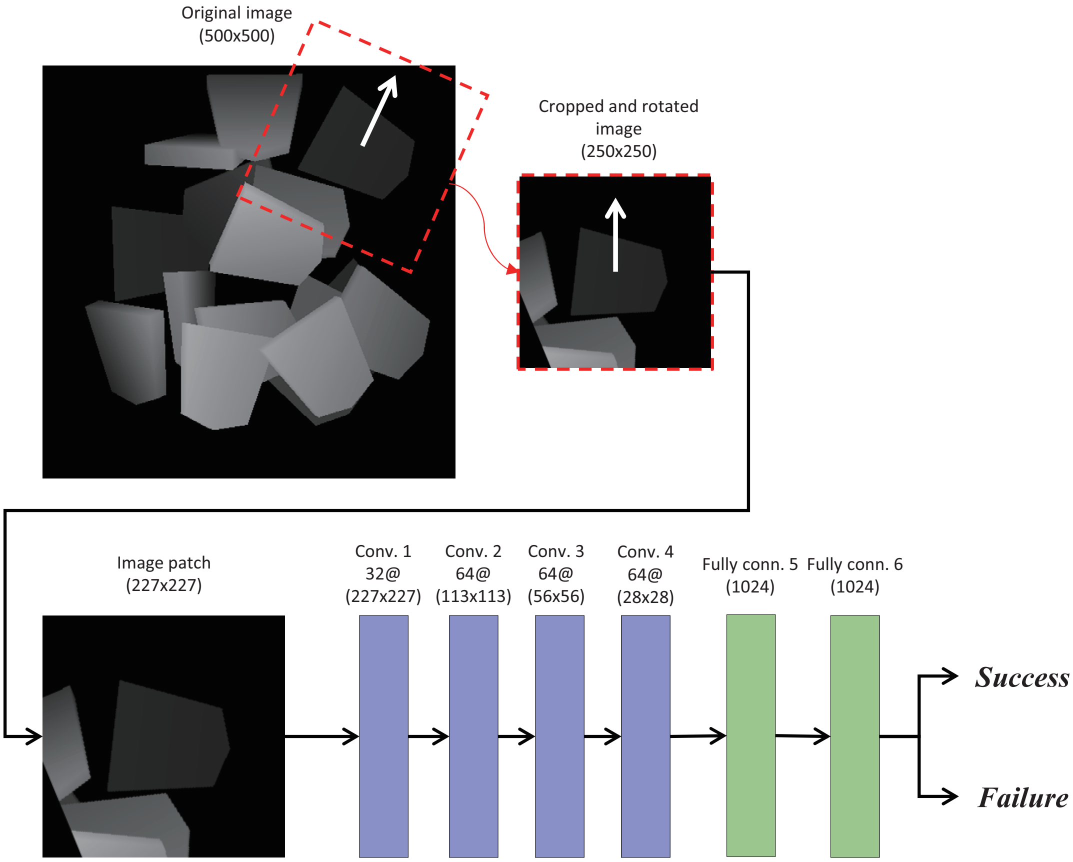
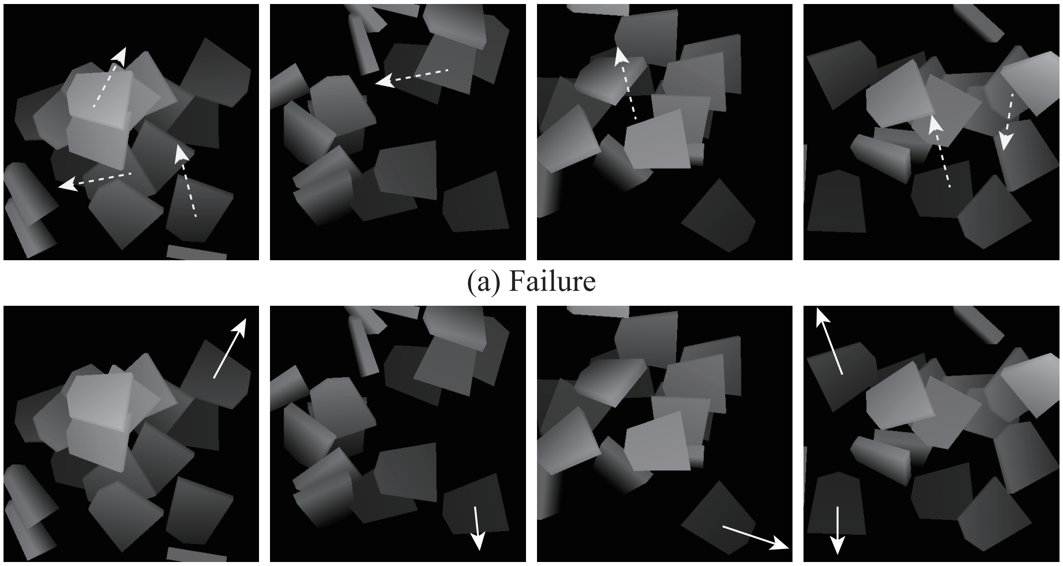


Figure 3.4: The architecture of our convolutional neural network.

3.3.2 Dataset

The dataset used for training is a stacked object depth image generated by the physical simulator environment [46], which comprises a CAD model of a Shogi piece. We manually annotate the possible operations on the in-depth images. To avoid bias in the training data, we generate the failures of operations by selecting objects and sliding orientations that humans would not select. If an object is adjacent to another object, we expect it to fail because adjacent objects tend to fall over. We set success to 1 and failure to 0. An example of annotating a depth image is shown in Figure 3.5.


(a) Failure

(b) Success

Figure 3.5: The depth images labeled as success or failure sliding motion.

3.3.3 Target Detection

3.4 Action Selection

Action planning for observation and manipulation in the vicinity of the manipulation target selected in the previous section is described. It is difficult to plan the subsequent actions accurately because of disturbances such as occlusions caused by overlapping objects and noise. Therefore, we use the POMDP framework [34, 73] to plan actions under uncertainty. This framework represents the state of the environment as a belief distribution for uncertain observations and selects actions that maximize rewards. POMDP is formulated in state s, action a, and reward r, and actions are planned probabilistically to maximize future reward. Kim et al. [35] and Eindenberger et al. [15] defined the observation model from the database. In this study, we define a state s as a pose of the target object and an observation model as the accuracy of pose estimations. We set an evaluation function V with the estimated state s and reward r, which is an index for the effectiveness of the manipulation and observation. These evaluation functions allow for planning the observation and operation within a single framework. We select the best action as follows:

$$a = \arg\max_{a \in A} V(a) \tag{3.3}$$

Here, A denotes the entire action, including manipulation and observation. In the next section, w explain the method of state estimation based on POMDP.

3.4.1 State Formulation

In this study, the object poses are discretely defined as states to select the next action under uncertainty. We assume four basic states based on the shape of a Shogi piece (Figure 3.6), as shown in Figures 3.7 (a)-(d). The basic states (a, b, c) assume that a Shogi piece is in contact with a plane. The other basic state (d) is the one wherein the piece is leaning against another piece with an inclination of 45°, which represents an intermediate state. We further add states by a rotation of 45° around the axis perpendicular to the center of gravity of the basic state (Figure 3.8). Specifically, 8 states are generated for each basic posture, and 32 states are defined as the entire state space S in this experiment. In this paper, the states are defined as $S := \{s_1, s_2, ..., s_{32}\}$.

In POMDP, the probability distribution of a state after it has been observed is called a belief b. The belief b is given as a probability distribution over the state space S, where $b(s)$ denotes the probability that $s \in S$ exists. The initial state is set uniformly for all $b(s_i) = \frac{1}{32}(i = 1, 2, ..., 32)$. The belief state is updated when

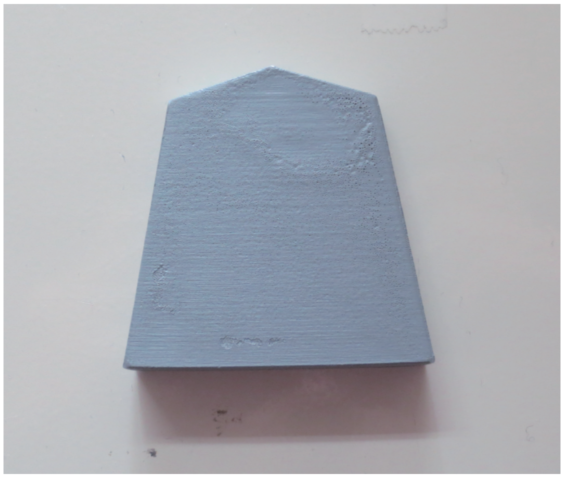


Figure 3.6: Object used in the experiment.

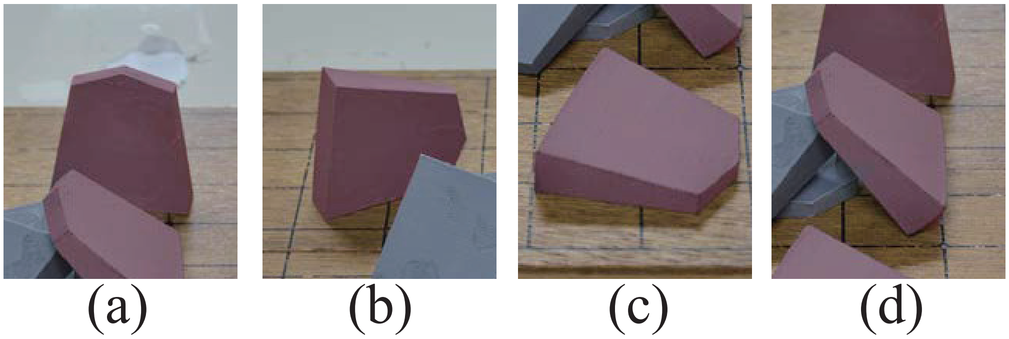


Figure 3.7: Basic object states

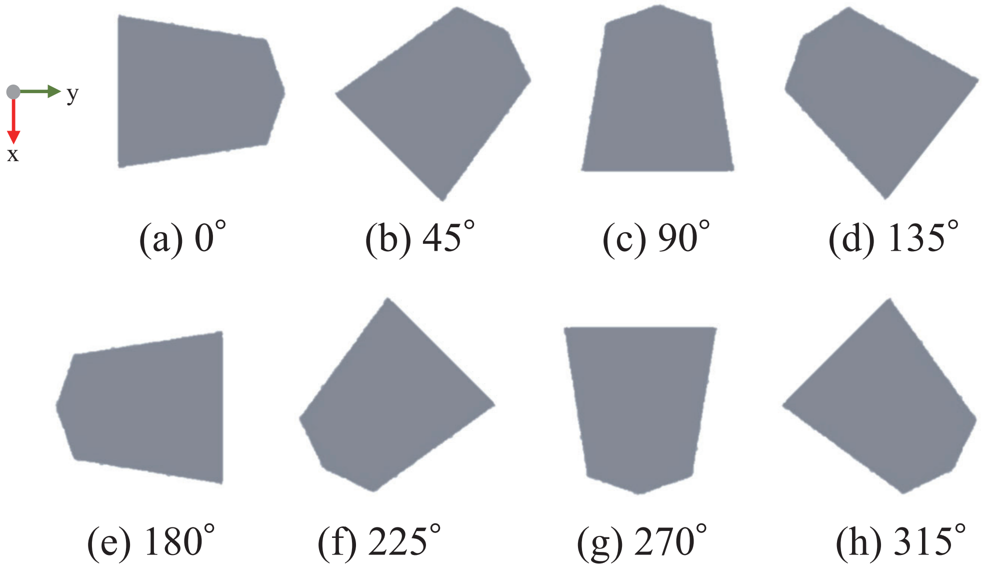
a new observation o_a is obtained by shifting the viewpoint a. The update is performed using the weight coefficient $\omega(s)$. The belief update from $b(s)$ to $b'(s)$ is expressed as follows:

$$b'(s) = \omega(o_s, o_a)b(s) \tag{3.4}$$

The weighting factor $\omega(o_s, o_a)$, which represents the goodness of fit with state s, is defined as

$$\omega_a(s) = 1 - \min\{\frac{e(o_s, o_a)}{e_{th}}, 1\} \tag{3.5}$$

Here, o_a is the point cloud data obtained by the action a, and o_s is the point cloud data when the operation target is assumed to be in the state s. Further, $e(o_s, o_a)$ is



(a) 0° (b) 45° (c) 90° (d) 135°

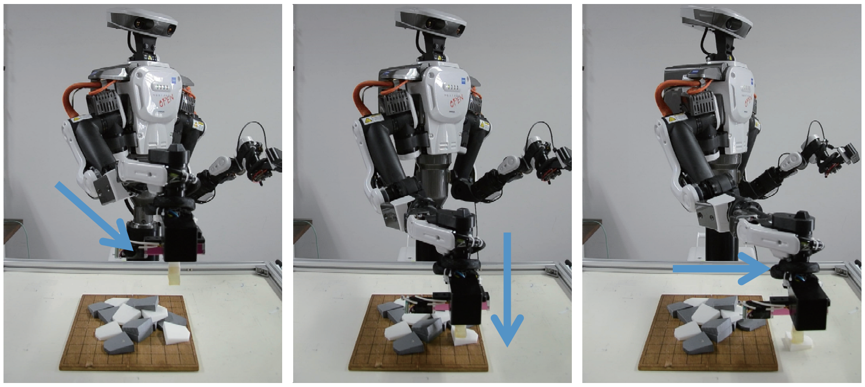
(e) 180° (f) 225° (g) 270° (h) 315°

Figure 3.8: Object poses for one basic state

an error function that indicates the average Euclidean distance between the corresponding point clouds of o_a and o_s, wherein we calculate the correspondence between the point clouds with the iterative closest point (ICP) algorithm [74,75]. e_{th} is a threshold for omitting beliefs that do not match because of low point cloud agreement. Furthermore POMDP generally has the problem of high computational cost. Therefore, rigorous pose estimation is performed only during robot motion generation. We represent the state of the Shogi pieces discretely in the action planning framework.

3.4.2 Action selection

In this study, we set an evaluation of observation based on the sliding operation as the index of the manipulation. As shown in Figure 3.9, we assume that a sliding operation continues until a Shogi piece is removed from the board. The



(a) Move above the target (b) Press the target down (c) Slide the target out the board

Figure 3.9: Scene of sliding the target. (a) The robot's arm moves above the target, (b) presses the target with the finger tool, and (c) slides the target out the board.

reward $R(s, a)$ for a sliding operation a for a state s is set as follows:

$$R(s', a) = \begin{cases} r & (s' \in S_{slide}) \\ 0 & (otherwise) \end{cases} \tag{3.6}$$

where $S_{slide} \subset S$ is the set of withdrawable states and $r > 0$ is the arbitrarily given reward. The evaluation function for the sliding operation is determined by the expected value of the reward function $R(s', a)$ as follows:

$$V(a') = \sum_{s' \in S} b(s') R(s', a') \tag{3.7}$$

Here, action a represents the manipulation of the robot. To generate actions, we estimate object poses (see Section 3.4.4) based on the point clouds of both the target object and surrounding objects. To avoid contact with surrounding objects, we select an orientation from $0°, 45°, 90°, ..., 270°, 315°$. To plan the path of sliding the target object, we use the single-query bi-directional lazy-collision checking (SBL) probabilistic roadmap planner [76] to avoid interference with other objects.

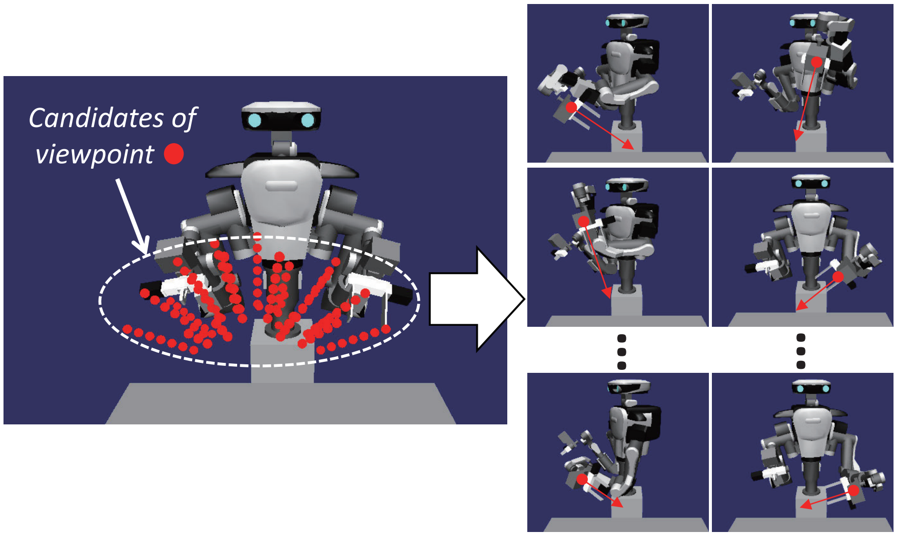


Figure 3.10: Candidates of viewpoint and view pose. We set the depth sensor towards the center of the board.

3.4.3 Estimate Uncertainty

The observation is to select the best viewpoint among the candidates to reduce uncertainty. We first define a set of sensor pose candidates using [49]. Let us assume an n-faced regular polyhedron whose geometrical center is located at the center of the box's bottom surface (Figure 3.10). We set the candidates to define the center of any regular polyhedron, orienting the viewpoints placed in the searching area. We also assume that a line passes through the geometrical center and orthogonally intersects the face of the polyhedron, with a set of points along the line at regular intervals. We assume that the sensor faces the geometrical center at each point. The following conditions are imposed on the sensor pose candidates. (1) The sensor is located in the searching area. (2) The path to the point is reachable. (3) No collision occurs with the links of the robot.

The amount of uncertainty in the point clouds is defined by the probability mass of the belief b, as follows:

$$M(o_a) = \sum_{s' \in S} b(s') \tag{3.8}$$

Here, o_a denotes the point clouds obtained after the observation a. The beliefs that do not match the point cloud are reduced by the weight calculation. If the sum of the beliefs, $M(o_a)$, is small, the state representation is not considered uncertain. The robot can reduce uncertainty when we select the best viewpoint from several candidates. The action a' to a viewpoint can update the amount of uncertainty as follows:

$$M(o_{a'}) = \sum_{s' \in S} \omega(o_{s'}, o_{a'}) b(s') \tag{3.9}$$

We define the effect of the observed action a' using a decreased amount of uncertainty as follows:

$$V_1(a') = M(o_a) - M(o_{a'}) \tag{3.10}$$

The equation represents the decreased amount of uncertainty when executing the next best observation $o_{a'}$, which is used as the recognition accuracy for the observed action a' toward the target object. Furthermore, it is crucial to observe the surrounding objects without occlusion. We set an evaluation for visibility to reduce the occlusion in the workspace. First, to evaluate the occlusion, the workspace is represented by a grid and divided into multiple cells with reference to [49, 77]. We first mark "Occupied" to the cells using point clouds obtained from the previous observation, including the point cloud. We also mark "Occluded" to the visible grid cells which are not marked as "Occupied." For each observed action a', the number of cells labeled "Occluded" is n_{occ}, and the visibility is evaluated by the following index:

$$V_2(a') = \frac{n_{occ}}{N_{grid}} \tag{3.11}$$

where N_{grid} is the total number of cells. The formula allows us to select the viewpoint that provides the largest area of point clouds.

The above indicators are integrated for each action a', and each viewpoint was evaluated as follows:

$$V(a') = \alpha V_1(a') + \beta V_2(a') \tag{3.12}$$

The action a' is for observation, and the parameter α and β are the weights, respectively, for the uncertainty and the visibility evaluation. In this study, $\alpha = 0.9$ and $\beta = 0.1$ are set to give priority to improving uncertainty.

3.4.4 Pose Estimation

We use the point cloud obtained by a depth sensor to recognize the object pose. After the point clouds are obtained, the object pose is estimated by using a method by using a known CAD model. First, the point clouds are segmented into each object by using the locally convex connected patches (LCCP) algorithm [78]. Second, the pose of an object is estimated based on a clustered viewpoint feature histogram (CVFH) and camera roll histogram (CRH) features [79]. The CVFH is the feature of point clouds from a certain viewpoint. The pose is finally estimated using the CRH features because the camera roll angle is not uniquely determined. After performing a rough alignment with the CVFH and CRH feature values, it repeatedly executes ICP [74, 75] and improves the estimated pose. Each of these recognition methods were implemented using the point cloud library [80].

3.5 Experiments

In this section, we execute experiments on a real robot to verify the effectiveness of the proposed method. The action planning flowchart of the proposed method is shown in Figure 3.11. The experimental conditions are set as follows.

- At the beginning of the game, nine Shogi pieces are placed in a pile, where all the pieces are randomly stacked each time.
- Only one finger, which is the tool at the end of the right arm, is used to remove a Shogi piece from a pile. The operation is limited to pushing and sliding the target object.
- If the piece is removed without other pieces collapsing the other pieces, the action is considered successful; otherwise, it is considered a failure.

We use a Shogi piece made of ABS resin, weighing approximately 9 g, with a shape of approximately 52.0 $mm \times$ 52.5 $mm \times$ 15.5 mm, which is created by a 3D printer (Figure 3.6).

Our experiment aims to remove all pieces, and as many pieces as possible, considering the performance under the rules of Yamakuzushi. First, we verify the estimation of the success rate of the starting point and orientation of sliding an object by ConvNet. Thereafter, we perform robotic experiments to verify the effectiveness of the proposed method in the real-world environment, including target selection based on the learning results and action planning using POMDP.

3.5.1 Evaluating the Detection Model

ConvNet was trained using depth images generated by a simulator environment and a dataset based on human annotations. The depth images are disjointed states generated by a simulation environment [46] built using a CAD model of a Shogi piece. The depth images are grayscale and indicate the height (Figure 3.5). The human visually annotates these images with the position and orientation of the piece retrieval (Figure 3.5).

The dataset consists of 12, 000 depth images, of which 6, 000 are images for successful object retrievals and 6, 000 are images for failed object retrievals. Specifically, 90% of the images (10, 800) are used to train ConvNet, and the remaining 10% is used to validate the trained ConvNet.

The results of the sliding object, for example, starting point and orientation, for the 10 depth images of the validation data using the orientation of sliding operations constructed by the trained ConvNet are shown in Figure 3.12. Considering that the outputs are similar to the human intuition, we consider the following three scenes as a success: (1) when the piece is not placed on top of other pieces, (2) when other pieces do not cover the piece, and (3) when the piece is not in Figure 3.7 (a), (b), or (d), but its wide side contacts the board as shown in Figure 3.7 (c).

3.5.2 Robotic Experiments

Figure 3.13 shows an example of the robot playing Yamakuzushi. It shows 10 frames extracted from the video, and each motion is selected by the proposed method. The robot removed three pieces in the example shown in Figure 3.13. In Figure 3.13 (a), all the objects are observed from the board, and our ConvNet detects the target object. In (b), a sliding operation is performed by the finger tool. In (c), the other target object is detected and again observed. In (d), our system is observed from another viewpoint based on our action plan, and (e) shows the sliding operation based on the second observation. Further, (f) shows that the target object is detected again by observing from other viewpoints; (g)-(i) shows iterative observations by selected viewpoints; and (j) shows that the action is finally selected and executed.

Figure 3.14 shows the point cloud obtained for each observation. Each observation shows the results of object pose estimation using features based on the CAD model of the Shogi piece. The frame shown in the figure is the region that includes the target and its surroundings. First, the sliding operation was selected by the target object selected in Figure 3.14 (a) and the point cloud at that time (Figure 3.14 (b)). In this case, the target object has no contact with the surrounding objects. Furthermore, in the case of the target object selected in Figure 3.14 (c) and the point cloud at this time (Figure 3.14 (d)), the result of the observation includes a large occlusion. Additionally, the state is insufficient because other pieces hide the target object. Therefore, the sliding operation was not executed, and instead, the observation was performed, as shown in Figure 3.14 (e). Furthermore, the third target object is detected in Figure 3.14 (f). However, the point cloud at that time (Figure 3.14 (g)) did not select the sliding operation. Because the reward r for the manipulation is small, the sliding operation is not selected if the observation is not sufficient.

To evaluate the object retrieval, we executed Yamakuzushi with 10 randomly placed Shogi pieces and counted the total number of trials and the number of pieces retrieved. The results are shown in Figure 3.15. The number of pieces removed ranged from 0 to 3 at each scene. In total, one or more pieces were successfully removed in 12 of 15 trials. Three pieces were removed in the experiment, and the average number of pieces retrieved in each trial was 1.47. The success rate of retrieval in each sliding operation was 59.5%, indicating that the robot selected an appropriate target object among multiple pieces.

Conversely, the unsuccessful retrievals were caused by the occlusion of multiple overlapping objects. Furthermore, the object pose estimation needs active searching action, e.g., pushing other surrounding objects, to improve the success rate of Yamakuzushi. In these tasks, the robot failed to retrieve the target object because of the initial placement of the object. Therefore, the robot needs the strategy of retrieving an object, not only sliding operations.

We also compared the success rate with and without ConvNet. The results are shown in Figure 3.15. In the case without ConvNet, we randomly selected the targeted piece with the smallest occlusion to verify the effect of using ConvNet, for example, considering human intuition. Furthermore, we expect the action plans for observation and sliding operation to be similar. In the case of object selection based on random observations, eight of the 15 trials failed to retrieve the object. This is a low success rate compared to our method, which successfully extracted one or more in 12 of 15 trials. The reason is that the target object was inappropriate, and the robot could not find the target object during iterative observations. These results suggest that object selection using the proposed ConvNet based on human behavior effectively improves the success rate for robotic manipulations.

3.6 Summary

In this chapter, I propose a method for observing and manipulating objects in a Yamakuzushi game as a typical example of human manipulation of objects in a pile-up. First, the manipulation target was determined using a ConvNet learned from human manipulation characteristics and depth images of the object. Thereafter, I proposed an action planning method for selecting appropriate actions for both observation and manipulation by setting the state of the manipulated object with respect to its posture and uncertainty. Furthermore, I conducted the Yamakuzushi game using a robot to verify the effectiveness of the proposed method.

For future work, I plan to adapt the robot to several manipulations to utilize it in various human tasks, in addition to the extraction as in the Yamakuzushi game. In addition, it was not possible to consider the effects of, for example, other objects that were completely undetectable because other objects covered them, and also because this study relied on the results of prior posture estimation for object contact. I can consider discriminating success or failure for images acquired from multiple angles to accurately determine from learning how to remove objects so that the pile can be maintained while focusing on the Yamakuzushi game. Therefore, improving the success rate of manipulation by the robot, including the above factors, is a future challenge. In addition, I used human behavior as a norm in this study and focused on objects that correspond to human intuition to conduct strict observation and manipulation action planning. However, quantitative analysis and verification of whether the characteristics of human behavior and intuitive judgments are acquired through learning and whether sufficiently valid choices are obtained is a future issue.

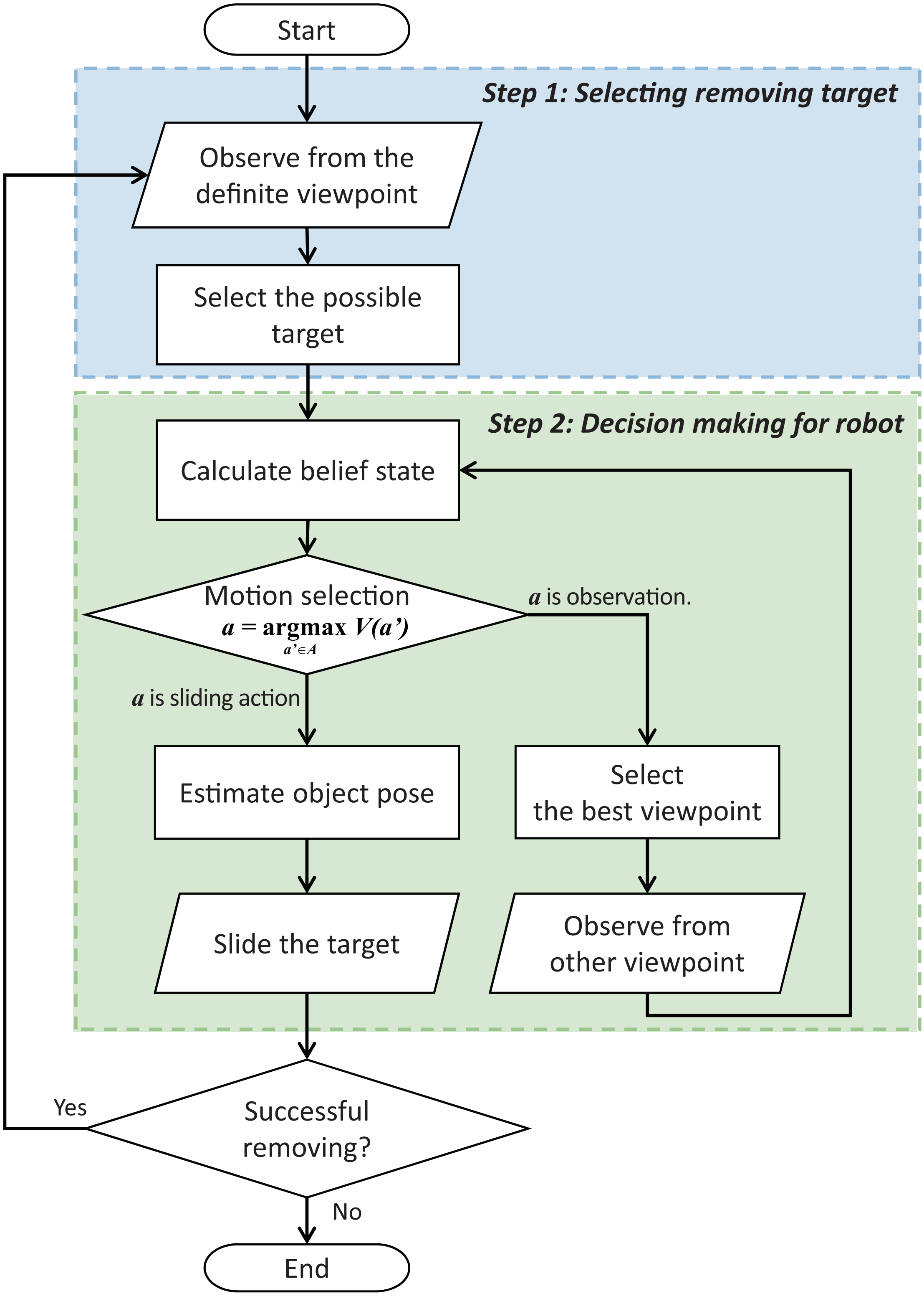

Start
Step 1: Selecting removing target
Observe from the definite viewpoint
Select the possible target
Step 2: Decision making for robot
Calculate belief state
Motion selection
a = argmax V(a')
a'∈A
a is observation.
a is sliding action
Estimate object pose
Select the best viewpoint
Slide the target
Observe from other viewpoint
Successful removing?
Yes
No
End

Figure 3.11: Flow chart of our proposed system.

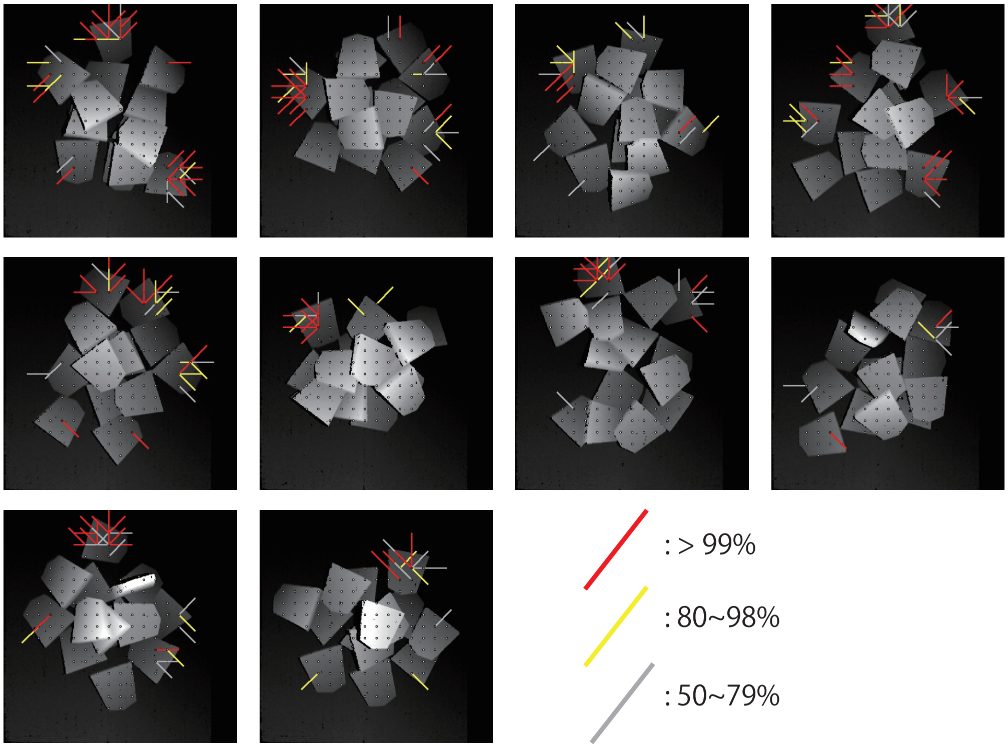
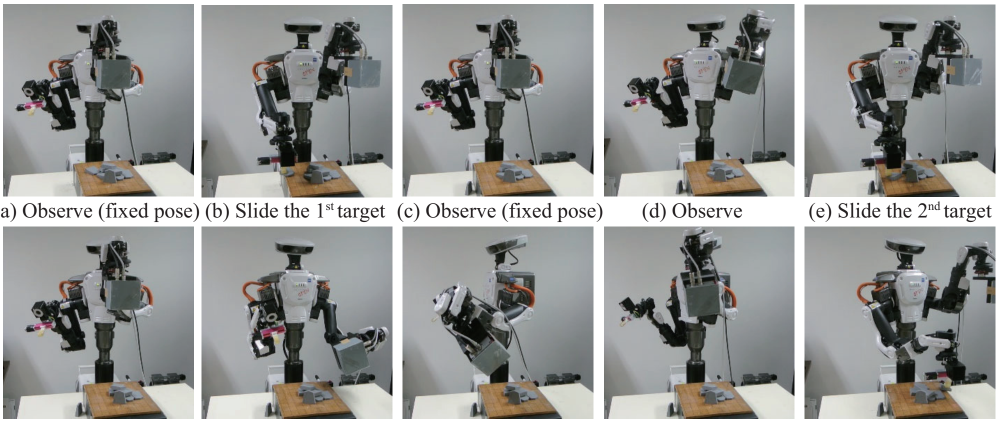
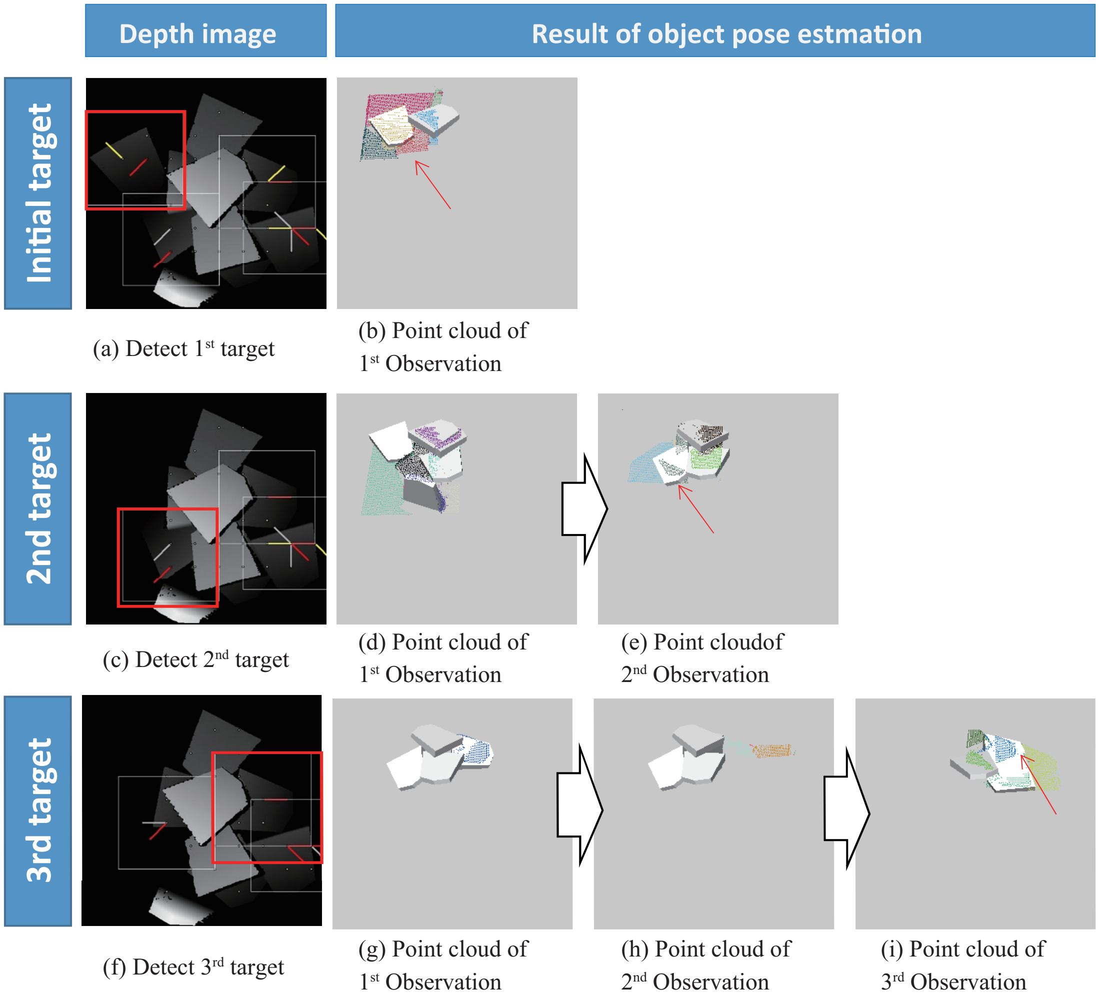


Figure 3.12: Detection results of the sliding motion in the proposed method. Some lines are for the robot's sliding motion. The red lines indicate a success rate of more than 99%, the yellow lines indicate a success rate of 80%-98%, and the gray lines indicate a success rate of 50%-79%. Some dots are the candidates of the starting point for sliding.


(a) Observe (fixed pose) (b) Slide the 1^{st} target (c) Observe (fixed pose) (d) Observe (e) Slide the 2^{nd} target

(f) Observe (fixed pose) (g) Observe (h) Observe (i) Observe (j) Slide the 3^{rd} target

Figure 3.13: Experimental scene for the Yamakuzushi game.



(a) Detect 1st target

(b) Point cloud of 1st Observation

(c) Detect 2nd target

(d) Point cloud of 1st Observation

(e) Point cloudof 2nd Observation

(f) Detect 3rd target

(g) Point cloud of 1st Observation

(h) Point cloud of 2nd Observation

(i) Point cloud of 3rd Observation

Figure 3.14: Detection and recognition results for sequential measurements of each target.

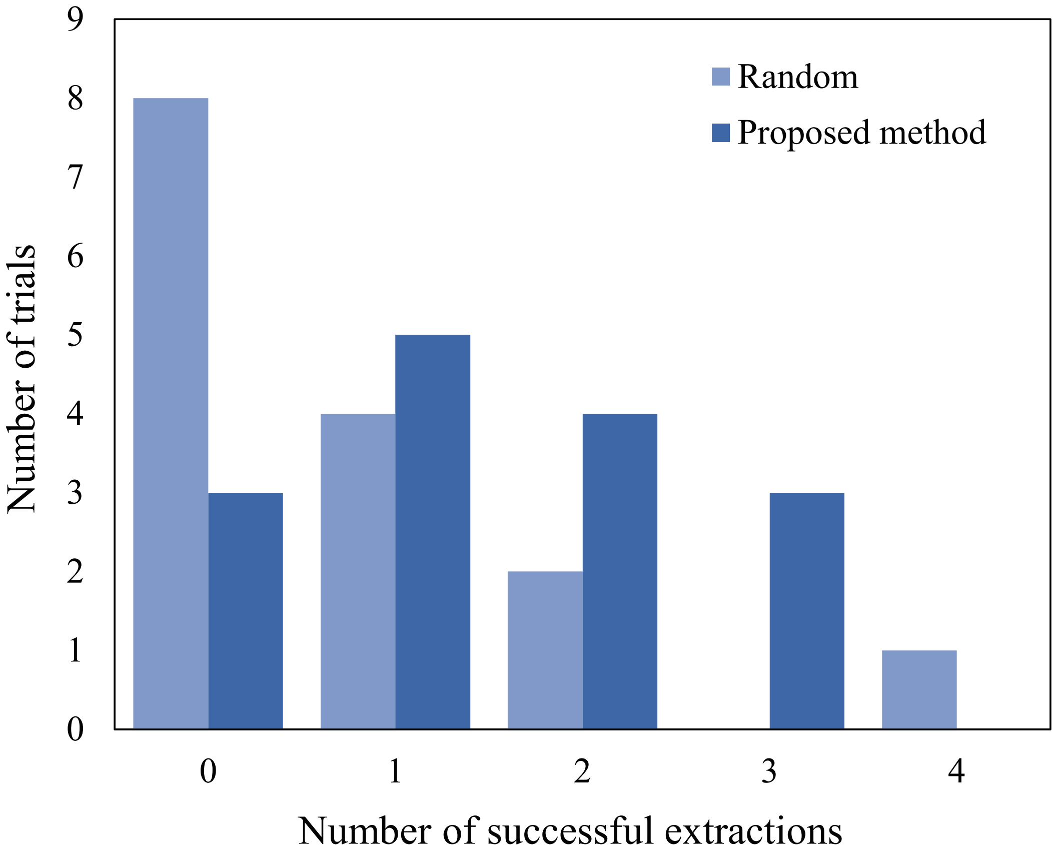


Figure 3.15: The number of successful extractions for each of the 15 trials.

Chapter 4

Bimanual Shelf Picking Planner Based on Collapse Prediction

4.1 Introduction

In logistics warehouses, we often have to extract a single object that is wedged between other objects on a shelf, which is potentially dangerous for heavy objects to fall and injure human workers. In this case, when a robot tries to extract one of the objects, it has to consider the positional relationship of overlapping objects and manipulate them accordingly. So far, various approaches have been proposed to extract an object from a shelf. In [23, 25, 26], different methods are proposed but require a series of rearrangement operations. In other cases, extraction and support relations are analyzed between pairs of objects from 3D visual perception [24]. However, in all previous approaches, a robot extracts the target object after rearranging its neighboring objects.

Humans, however, extract an object from a shelf while supporting other neighboring objects as shown in Figure 4.1 (a). Based on this observation, we propose a bimanual manipulation planner to extract a target object from a shelf while supporting the other object as shown in Figure 4.1 (b). To extract an object from a pile without any collapse, we need to determine which of the target's neighboring object the robot has to support. We propose a learning-based approach to extracting the target object from the pile while supporting the other objects. A network model based on a Fully Convolutional Network (FCN) [81] has been designed to predict the pile state while extracting the target object with a pixel-wise collapse probability map. The inputs of the network are a depth image of the shelf content and two binary masks corresponding to the two objects

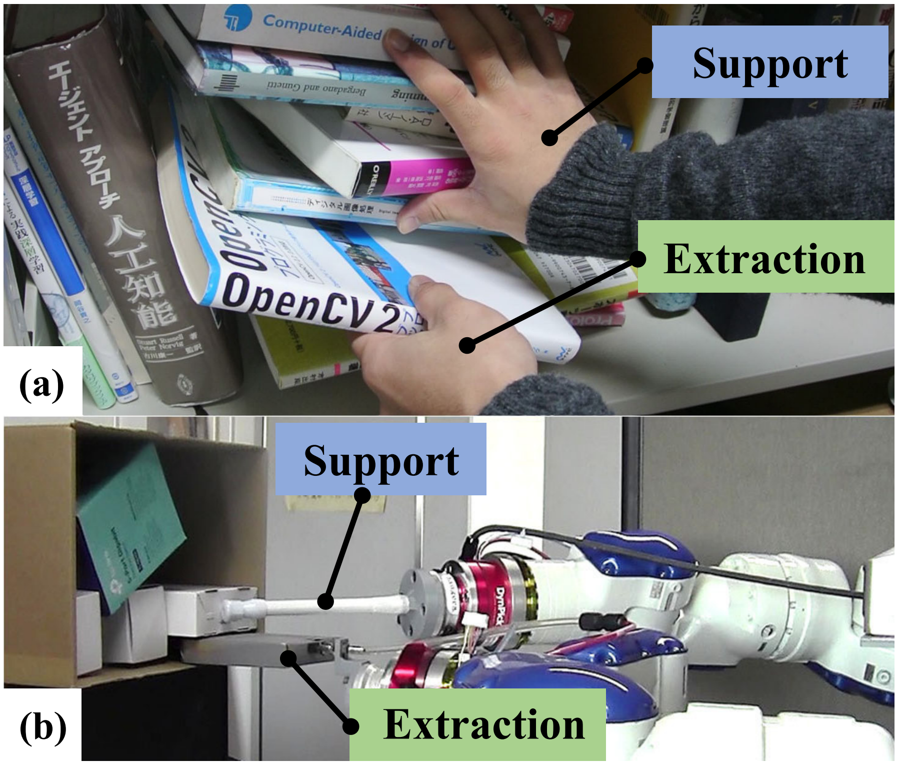


Figure 4.1: Extracting the target object while supporting others: (a) a human is extracting a book from the shelf while supporting the neighboring books, (b) robotic bimanual manipulation for safely extracting an object from the shelf.

selected for extraction and support. The output of the network model is a labeled image predicting the collapsing region while the target object is extracted. Given this output, the robot can select the proper object to support by defining the ratio of the predicted collapsing region as the safety index to the shelf picking. In addition, to generate a large number of training data of depth images, related binary masks, and label images, we use a physics simulation of the piled objects and of the extraction/support action. We experimentally verify the effectiveness of our proposed method by using a real dual-arm manipulator. We show that the robot can safely extract the target object from a shelf with a success rate larger than 80%. By using our proposed method, we do not need to rearrange the objects placed on a shelf to extract the target object and so we

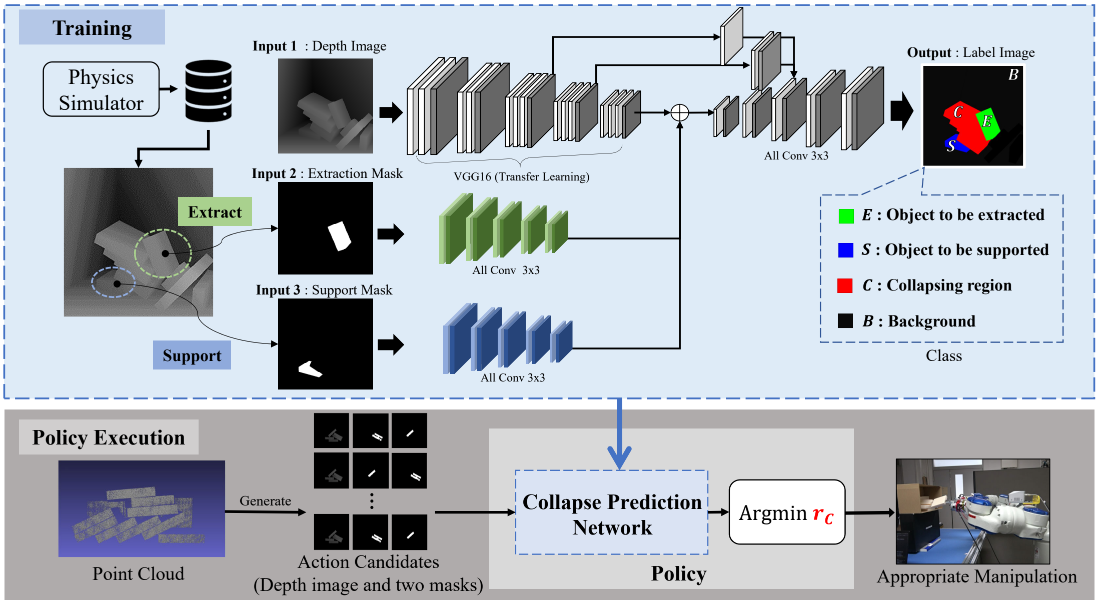


Figure 4.2: Schematic overview of the proposed manipulation policy based on Fully Convolutional Network (FCN). The inputs are a depth image and two binary masks. The model encodes the depth image with the VGG-16 network and two masks with a five-layer network, and concatenates these three networks. We focus on the collapsing region, C (highlighted in red), in the output of the network. The method uses the argmin of r_C (ratio of C in the image), to return the appropriate action. The size of the depth image and its related two masks are 256×256.

increase the picking efficiency.

Our main contributions are:

- A Fully Convolutional Network to infer the pixel-wise probability map of the collapsing region while extracting a selected object from a shelf (Subsection 4.2.2).
- A physics simulation that generates the necessary training data for the FCN (Subsection 4.2.1).
- A robotic system able to extract a target object from a pile, on a shelf, without rearranging its surrounding objects.

4.2 Shelf Picking Method Implementation

We propose a bimanual manipulation method to extract a target object from a pile while supporting the other object. In order to first verify the effectiveness of our new approach, we assume that the robot achieves the task by pulling a box-shaped object out horizontally. Assuming a situation in which the insertion of fingers between objects is difficult for the robot, one arm is mounted with a suction gripper to extract the target object. The other arm has a rod-shaped end-effector to support other objects as seen in Figure 4.1. We use a depth sensor to provide a 3D point cloud captured from the robot point of view in front of the shelf containing the pile of objects.

Figure 4.2 illustrates the flow of our overall architecture. The user selects the object to extract, and then a FCN is used to predict which objects will be affected during the extraction (I.e., collapsing region).

In the following subsections, the different steps are explained in detail.

4.2.1 Physics Simulator for Data Generation

In this subsection, we describe the setup of the physics simulation system used for data generation.

Scene Generation

We generate a randomly stacked state of objects in the simulator. In this study, we use PhysX [82], a physics simulator, to configure and simulate the environ-

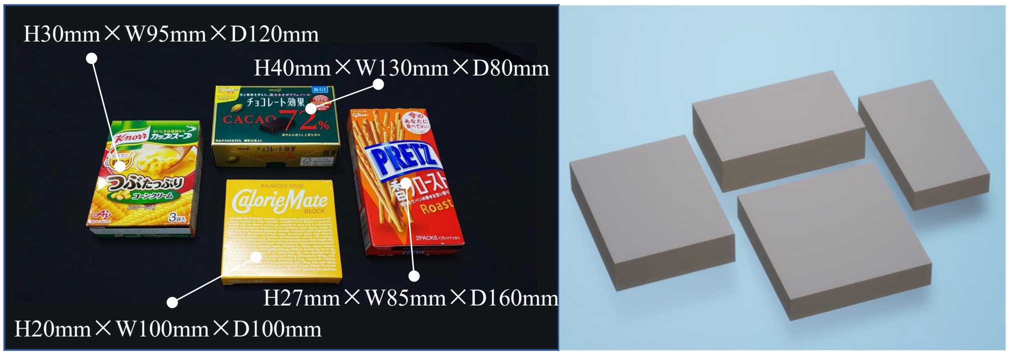


Figure 4.3: Types of objects used in the simulations. The left actual boxes. The right side shows the 3D models used in the simulator.

ment.

Our simulator is designed with the following settings. We consider a situation where many product boxes are on a shelf, thus we set the simulation parameter referred to their actual movements. For both, objects and the shelf in our environment, we empirically set the coefficient of static to be 0.9, dynamic friction to be 0.8, the coefficient of restitution to 0.1, and the density to 1.0 kg/m^3, respectively. We perform the shelf picking simulation by placing six objects from a set of objects. As the number of objects on a shelf increases, the extraction generally becomes more difficult. In our study, we fix the number of the objects to be six, which can generate the successful cases empirically in about 50% even if the target object for extraction/support is randomly selected. Moreover, for the after-mentioned verification, we prepared two sets of objects: One type of object (H 20 mm× W 100 mm× D 100 mm), and four objects of various sizes (H 27–40 mm× W 85–130 mm× D 80–160 mm), illustrated in Figure 4.3 for the detail. We generate the dataset with either of the sets according to the conditions.

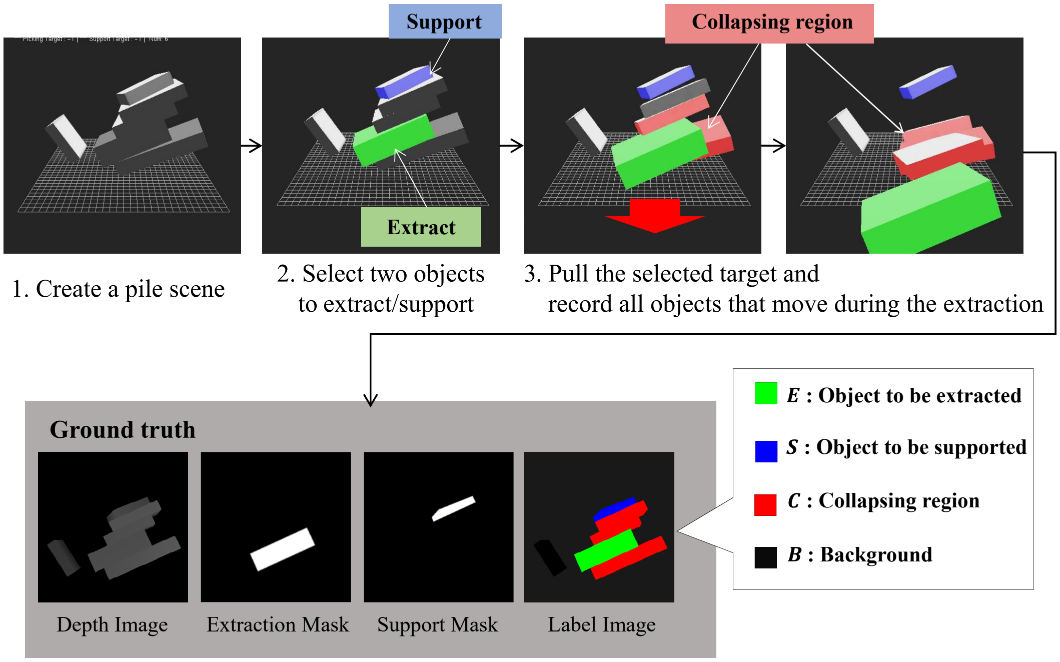

Support
Collapsing region
Extract
1. Create a pile scene
2. Select two objects to extract/support
3. Pull the selected target and record all objects that move during the extraction
Ground truth
Depth Image
Extraction Mask
Support Mask
Label Image
🟩 *E* : Object to be extracted
🟦 *S* : Object to be supported
🟥 *C* : Collapsing region
⬛ *B* : Background

Figure 4.4: Dataset generation procedure with physics simulator. The upper row shows a scene of a simulation while the lower row shows the images of ground truth generated from the simulated scene.

Data generation and Simulation Procedure

Figure 4.4 shows the simulation process. First, a pile of objects is created in the simulated environment. Second, the pair of target/support objects are selected randomly, and the corresponding extraction/support masks are generated. The target object is extracted horizontally toward the virtual observer (robot). The supported object remains fixed in the environment and is not affected by interference or gravity; its pose does not change. Finally, in case there are some changes in the other objects' pose (other than the target object), we label these objects as the collapsing region. In one simulation, we obtain the tuple consisting of three images as input data (depth, extraction mask, support mask) and one labeled image as output data, as shown in Figure 4.4.

4.2.2 Collapse Prediction Network

This subsection describes the neural network that predicts the objects affected by the target object extraction and so most likely to fall or collapse. Similar to the fully convolutional network (FCN) used in [81], our model classifies each pixel belonging to the collapsing region.

Ground Truth

The input data consist of a depth image (256×256) and two binary masks (256×256). One mask is an object to extract, and the other mask for the object to support. The output data is a labeled image that represents the classifications of each pixel of the image (256×256). We define four classes: Object to be extracted E, Object to be supported S, Collapsing region S, and Background region B, as shown in Figure 4.4. The collapsing region C expresses the region of objects which move or fall from the shelf while extracting the target object. These input/output data are automatically generated from the simulator.

Network Architecture

Our network model consists of an encoder for extracting the feature value of the input and a decoder for producing the segmented image at its original resolution. Figure 4.2 illustrates the network architecture. First, the encoder part consists of three networks. The model generates the feature maps for a depth image with the VGG-16 network [83] pre-trained by ImageNet [84] and for two masks, each with five convolution layer network. These three outputs are then concatenated into one feature map. Next, the decoder part with five convolution layers up-sample the feature maps to the original resolution with deconvolution. Moreover, our network uses the skip architecture by referring to prior examples [81, 85], which combines the feature maps of the lower layers with those of the upper layers to recover the general location information while preserving the local information.

4.2.3 Manipulation Planning

This section describes the manipulation procedure to perform the extraction by applying the trained network. The robot acquires point clouds from a depth sensor installed in front of the shelf and generates three input images from this observation. One is a depth image converted from the point clouds to the depth map. The other two images are mask images representing the object to be extracted and the object to be supported. The mask image, M_c, is a binary image from each cluster, c_i $(i = 0, 1, 2, ..., N - 1)$ of point clouds, which is classified by object segmentation based on the region growing method [86] and the binarization. Figure 4.5 shows the process. In the actual experiments, the robot end-effectors approach each object toward the center of gravity in these masks.

We set action candidates A according to two situations: (1) One situation is that the robot chooses the safest pair of extraction/support objects (for example, to empty a shelf). In this case, we define action candidates A by preparing all the combinations of two different targets of the extraction/support action. (2) The other is that we need to extract a predetermined target object. In this case, we define action candidates A by choosing each object to support the specific target.

Next, we define the safety index to select the best action from all the candidates. The output shows the region, R_E, R_S, R_C, and R_B, that indicates the regions of four different classes (E, S, C, and B). If R_C is large, it will increase the risk of collapse. Based on this assumption, we can define the following risk index:

$$r_c\,(a) = \frac{area(R_C^a)}{area(R_E^a \cup R_S^a \cup R_C^a \cup R_B^a)} \tag{4.1}$$

R_E^a, R_S^a, R_C^a, and R_B^a denote the regions in the output of an action candidate a for two selected objects. $area(\cdot)$ indicates the area of the region. Our algorithm selects the input data that is the smallest for index r_c based on Eq. (4.1) and determines the best action, a, of all the action candidates to be manipulated by the robot.

$$a = \underset{a' \in A}{\arg\min}\, r_c\,(a') \tag{4.2}$$

If the robot's motion is out of the control range, we eliminate it from the candidates and select the next best move.

4.3 Experiments

4.3.1 Training Settings

In our experiments, we acquired 15,000 pairs of input and output images from the simulator. From these datasets, we used 90% as training data and 10% as validation data. We augmented our training dataset through left-right inversion and utilized the network using 30,000 pairs. We set the initial learning rate to 0.0001 up to 30 epochs and 0.00001 from 30 epochs onward. The batch size was

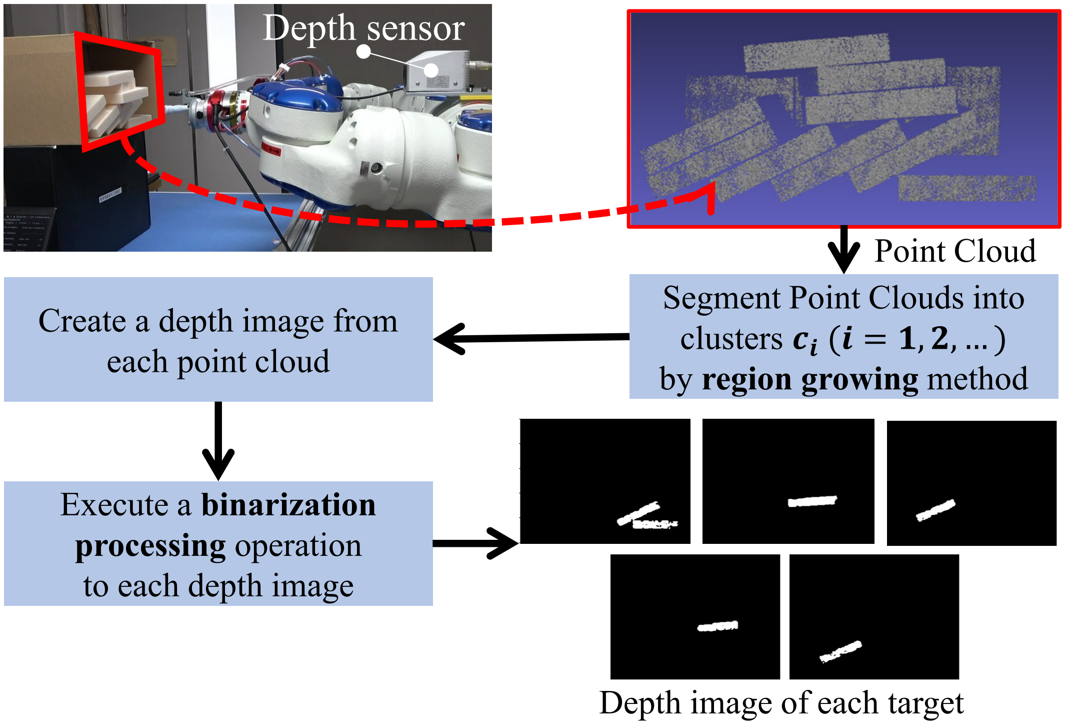
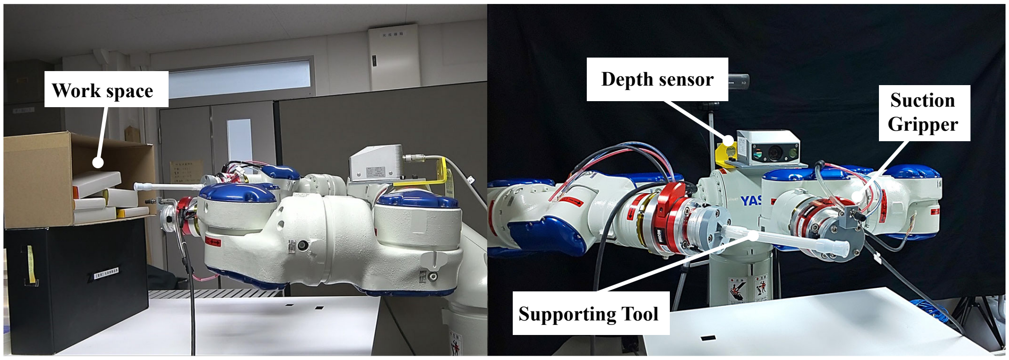

Depth sensor
Point Cloud
Segment Point Clouds into clusters c_i $(i = 1, 2, \ldots)$ by **region growing** method
Create a depth image from each point cloud
Execute a **binarization processing** operation to each depth image
Depth image of each target

Figure 4.5: Process of generating candidates with the segmentation method.

1, and we use Adam method [87] as the optimizer. The number of epochs during training was 50, and each epoch required 27,000 iterations. In our training, we used the NVIDIA RTX 2060 super (8 GB VRAM).

4.3.2 Experimental Setup

To verify the effectiveness of the proposed method, we conducted experiments using an actual robot under several conditions. Figure 4.6 shows the experimental environment used in the verification. We use the MOTOMAN-SDA5F (Yaskawa Electric Corp.) [88], a bimanual robot with 7 degrees-of-freedom robot arms, which has a suction gripper and a plastic rod-shaped end effector (the bar's length is 20 cm) at the tip of each arm of the robot. The YCAM3D-10L



(a) Side view (b) Front view

Figure 4.6: Experimental setup.

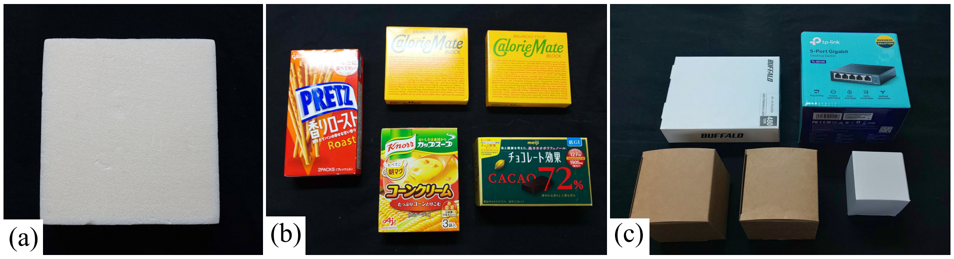


Figure 4.7: Objects used to evaluate the generalization of the proposed method: (a) experiments for objects of the same size, and experiments for objects of various sizes, and (c) experiments for new objects (not used in simulations).

(YOODS Co., Ltd.) [89], a 3D depth sensor, is installed on the bimanual robot, facing the shelf.

4.3.3 Results

To evaluate the performance by using the actual robot, we consider two scenarios.

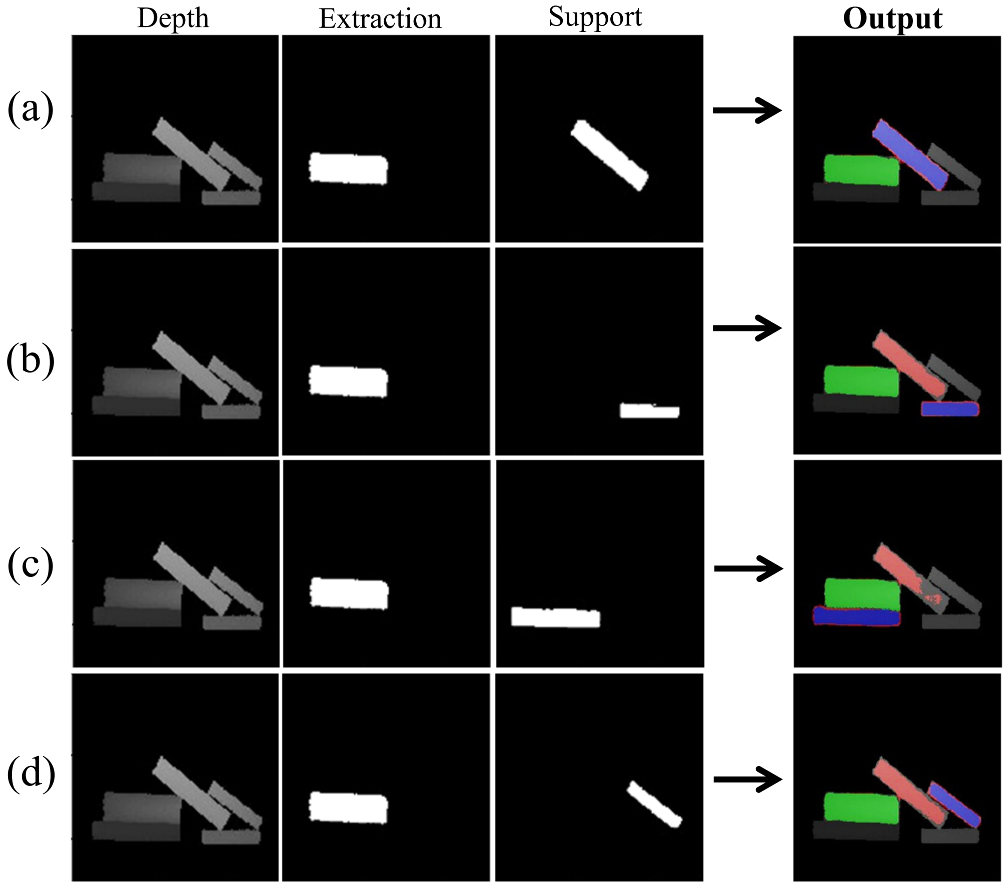


Figure 4.8: Visualization results from the proposed network model: (a)-(d) Outputs of extraction for a target object while supporting different objects. The red area on the images shows the predicted collapse region (R_C). The green region shows the object to be extracted correspond to R_E, and the blue region shows the object to be supported correspond to R_S.

Choosing the safest pair of extraction/support object

We verify the performance of our prediction network through experiments that the robot always chooses the safest action. In our real-world experiments, we used the target objects as shown in Figure 4.7 (a), (b), which are the same size as the models used in our simulations (Figure 4.3). Moreover, in order to evaluate the generalization capability, we separately prepared new objects (Figure 4.7 (c)). We used the following conditions in experiments:

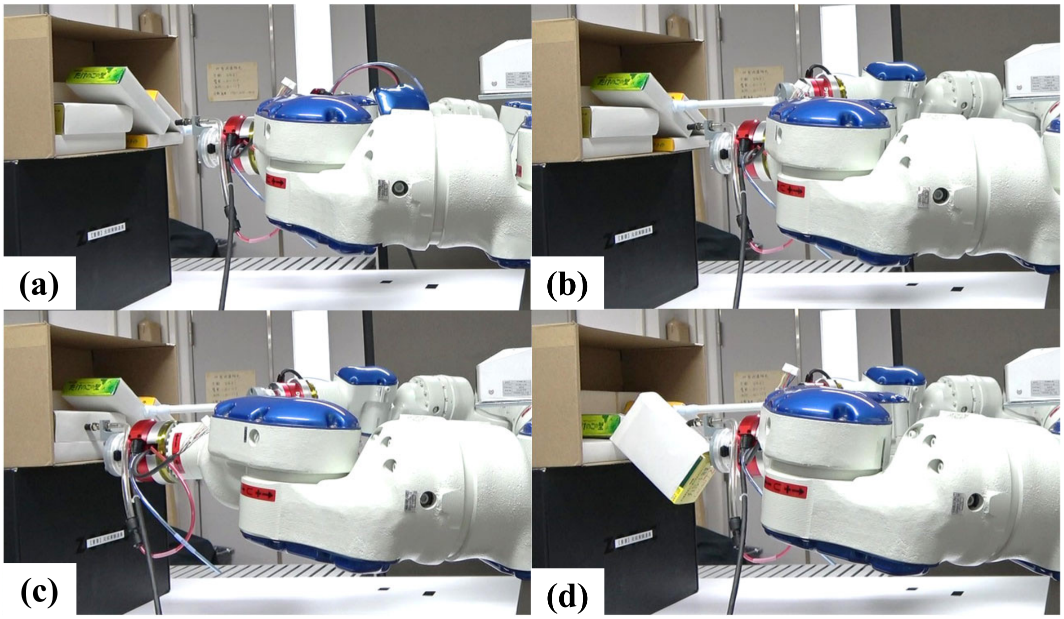


Figure 4.9: Description of target manipulation with the proposed method: (a)-(d) a series of scenes in one task.

Table 4.1: Experimental result

	Objects used in experiments				
Proposed model	**Same-size** (5 objects)	**Various-size** (5 objects)	**New various-size** (5 objects)	**Same-size** (10 objects)	**Total**
Trained with same-size objects	18/20	16/20	15/20	16/20	**65/80 [81.3%]**
Trained with various-size objects	17/20	17/20	17/20	13/20	**64/80 [80.0%]**

- 5 objects of the same size, as shown in Figure 4.7 (a).
- 5 objects of various sizes, as shown in Figure 4.7 (b).
- 5 new objects of various sizes (not used in the simulator), as shown in Figure 4.7 (c).
- 10 objects of the same size, as shown in Figure 4.7 (a).

Furthermore, we prepared two network models trained with different datasets generated from the simulations with objects of the same size or various sizes. With each trained model, we conducted 20 trials in every four patterns by changing the size or number of objects. If the robot removes only one object from the shelf, we regard it as a success; otherwise, we consider it a failure.

As shown in Table 4.1, we conducted robotic experiments under the above-mentioned conditions. The robot achieved a high success rate across all conditions, and the overall extraction success rate was 81.3% (65/80) for the model trained with objects of the same size and 80% (64/80) for the model trained with the dataset of various sizes.

Extracting a predetermined target object

We assume that a specific target object is needed in a practical situation. The user chooses one object to extract from a shelf in advance, and in that case, our policy determines which object to support correctly.

In this case, the output of our network is shown in Figure 4.8 (a)–(d). The robot selects the best action from the output; i.e. where region R_C (highlighted in red) is small shown as Figure 4.8 (a). Figure 4.9 shows the experimental set-up. When the correct action is selected, the robot first presses the support object with the stick from its right-hand and then pulls out the target object with its left-hand suction gripper. Based on the results, we confirmed that the robot selected combinations of objects are less likely to collapse and so execute the safest manipulation.

Moreover, we conducted 20 trials in that case. At each trial, the object to be extracted is not changed. It should be noted that we trained the network with the dataset generated in the simulations using various-size objects (Figure 4.3) in this verification. In 20 trials of the experiment under this condition, the robot can extract a single target object without collapse with a success rate of 85% (17/20). These results confirm that our network works well for those conditions.

4.4 Evaluation

In this section, the performance concerning two points is evaluated. (1) We set a benchmark of the prediction performance based on segmentation metrics and compare our proposed network under different conditions, (2) we acquire the success rate, representing the percentage of the completion when extracting a single target object without collapse by using the real robot.

4.4.1 Prediction Performance

We confirm that the network can correctly predict the collapsing regions with ground-truth data, as shown in Table 4.2. To evaluate the performance of the collapse prediction, we focus only on the collapsing region C in this study. Our metrics include $precision$, $recall$, and IoU calculated in pixels between the predicted and ground-truth data. We calculate the average values on metrics with a hundred ground-truth data and compare two networks trained with different training datasets. Moreover, to verify a generalization of the performance, we prepare the ground truth in two different patterns: target objects of the same size or various sizes. We empirically set the threshold of classification for each pixel to 0.4.

Table 4.2: Trained Model Evaluation.

		Network performance		
Proposed Model	**Ground-truth**	Avg. IoU	Avg. Recall	Avg. Precision
Trained with same-size objects	Same-size target	0.339	0.511	0.437
	Various-size target	0.438	0.641	0.530
Trained with various-size objects	Same-size target	0.359	0.438	0.576
	Various-size target	0.452	0.511	0.697

As shown in Table 4.2, even when we use networks with different training datasets, there is no significant difference on each metric to the same ground-truth. This result indicates that the size of the object has little effect on learning. In contrast, when we use the network trained with the objects of various sizes, IoU and *precision* increase in both ground-truth data. By using our method, the collapsing region tends to become a shape similar to the object model. It is assumed that the network trained with objects of the same size is relatively sensitive to shape differences. Therefore, training with objects of various sizes works well for correctly predicting the region.

4.4.2 Real-world Manipulation

As shown in Table 4.1, the robot extracted successfully up to 81.3% (65/80) for the same object dataset and 80% (64/80) for the dataset of objects of different sizes. The success rate of each object is not significantly affected in different datasets. Similarly, there is no difference in the success rate when the objects are the same (Figure 4.7 (a)) and when the size of the objects is randomized (Figure 4.7 (b)). The success rates of 75% (15/20) and 85% (17/20) were confirmed in the experiments with objects of new various sizes (Figure 4.7 (c)), indicating that there was no overfitting of our learning results.

In extracting a predetermined target object, our method achieved a high success rate of 85%, indicating that our method can work well in logistics warehouse conditions. The success rate is almost equal to other experimental results.

In failed cases, the robot executed incorrect actions, such as supporting an object unrelated to extracting a target. Therefore, it is necessary to reconstruct the dataset or evaluate each action on each successful trial so that the robot avoids selecting an uncertain action. Besides, our method cannot extract the specific target when more than two objects directly overlap the target by one-shot manipulation. As shown in Table 4.1, the success rate decreased with ten objects of the same size. For example, if an object is not simply put on another object, the robot needs to support more than two objects. In our method, however, the robot can only support one object, causing a low success rate. In our future work, we will address this issue.

4.4.3 Discussion

We proposed a learning-based approach that predicts and minimizes the risk of collapse while extracting a target object and supporting another. The conventional learning-based approaches [63,64] predict the support relationship as Section 2 mentioned. However, considering a complex pile, it becomes more difficult to determine the support object by its geometry shape and/or physical interaction. In contrast, our proposed method can directly predict whether the selected action is proper or not without checking the complex scene structure. However, in order to realize the new method, we focused only on box-shaped objects for the sake of prototyping. Our future work will be extended to more complex-shaped objects which are used in daily life.

4.5 Summary

This chapter described a shelf picking method for safely extracting a single object from a shelf while supporting another object. By using the proposed network model that predicts the objects that would collapse, a bimanual robot was able to extract the object without objects falling.

In the future, I plan to make improvements to support actions and the simulator. In particular, I will analyze the trial result of each simulation by adding actions to support and extract in appropriate way different types of objects.

Chapter 5

Shelf Replenishment Based on Object Arrangement Detection and Collapse Prediction for Bimanual Manipulation

5.1 Introduction

Shelf replenishment in warehouses and retail stores is a particularly challenging example of dexterous robotic tasks. Recently, the use of robots in retail has rapidly increased. However, presently, most practical situations require humans to handle shelf-related tasks, owing to their flexibility and reliability, despite the recent progress in vision processing, manipulations, and the development of functional grippers [1, 5, 23, 42, 63].

The replenishment process is divided into two cases. In the first case, a space is found in which the object to be inserted fits, and the object is placed there. In the second case, no space is available, and objects already on the shelf must be moved to create space to place the new object. In the latter case, the manipulation of the objects on the shelf to create an insertion space must be performed carefully to avoid tipping over or damaging the objects already on the shelf. Appropriate manipulation strategies are required in both cases.

In our previous work [90], we proposed the learning-based evaluator to predict the risk of collapse of a shelf, based on both the desired object extraction and object evaluation supporting the successful extraction. The neural network explicitly learns the relationship between objects (extract/support) and evaluates whether a collapse would occur. The extracting action with the minimum risk of collapse was selected; however, manipulations necessary for shelf replenishment were not suggested. The present study automates replenishment by improving our previous collapse prediction network and proposes a new action plan while minimizing changes to the state of objects on the shelf. Moreover, our method supports the use of bimanual arms to create an insertion space while considering the safety of the shelf content.

In this study, a novel approach for automating the replenishment of disorganized shelves with a bimanual robot is presented (Figure 5.1). First, we classified the objects in organized/disorganized displays using a general object detection method. This allowed us to treat these categorized objects explicitly. Our deep neural network infers the neighboring object's behavior from a depth image when removing a specified manipulation target; that is, the network can predict which objects fall from a shelf. The deep neural network was trained on a dataset generated using a simulator. The proposed inference-based strategy provides an appropriate decision and course of action on whether to create an insertion space while considering the safety of the shelf content. Compared to our previous work, we improved our collapse prediction estimator to be applied to a shelf replenishment task, allowing the robot to estimate the risk of single-arm manipulation without supporting the other objects. We considered the replenishment task through single-arm/bimanual manipulation to cover various practical cases.

The main contributions of this study can be summarized as follows.

- We classify objects in organized/disorganized displays to understand the shelf display as a whole, which reduces the complexity of inter-object relationship analysis and allows the manipulation of a group of objects as a unit instead of single objects.

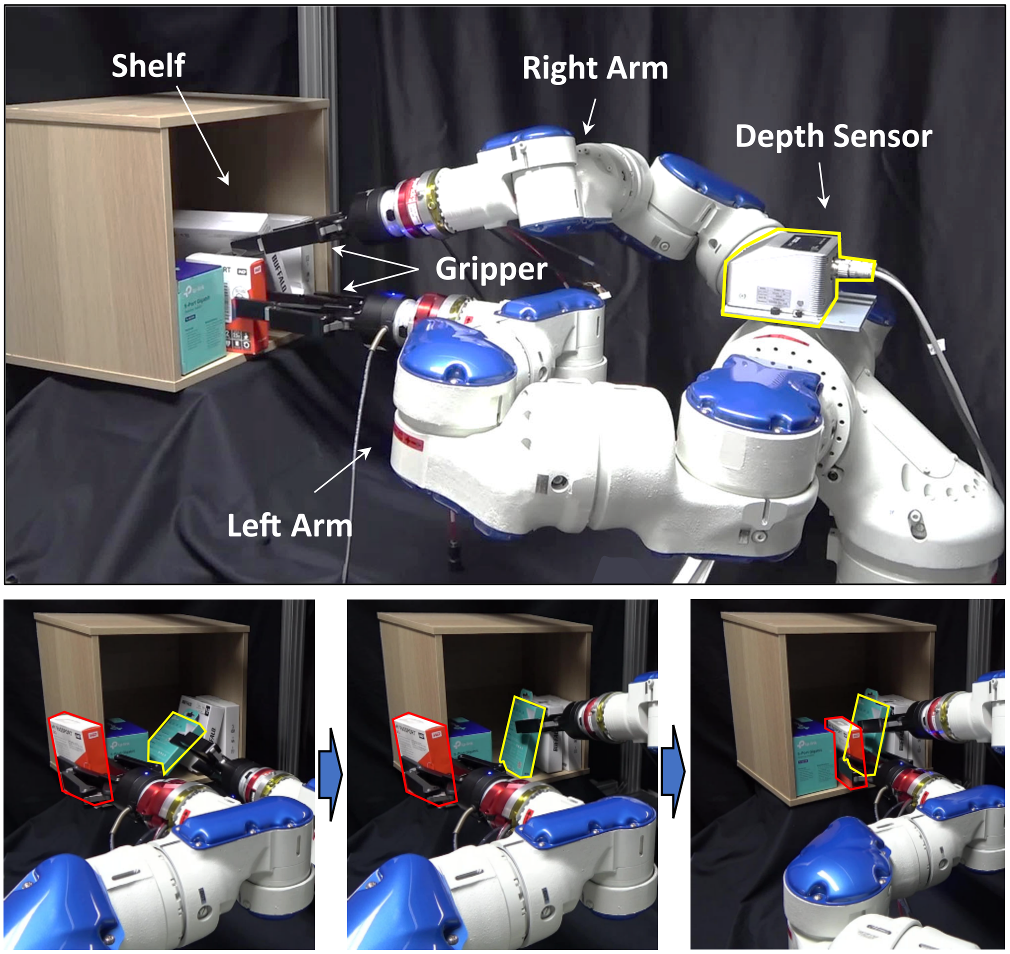


Figure 5.1: Bimanual robotic shelf replenishment. We present a robotic shelf replenisher with bimanual manipulation, which fills the shelf with an object. Our method allows for the slight rearrangement of the shelf to create space for replenishment without damaging the shelf or the other objects. Given the state of the shelf, bimanual or single-arm operation is appropriately selected to plan the action.

- Our method enables novel action planning with a bimanual robot for shelf replenishment by predicting the occurrence of an object collapsing via a neural network. In particular, our method can consider any state of the shelf, and select the best action for each state, including single-arm or bimanual manipulation.

The remainder of this chapter is organized as follows. Section 5.2 explains the proposed shelf-replenishment algorithm. Section 5.3 describes the experiments and network benchmark used to evaluate our architecture. Section 5.4 provides a discussion. Finally, Section 4.5 concludes the paper.

5.2 Materials and Methods

Figure 5.2 illustrates the flow of our architecture. We present an approach for automating replenishment using vision-based detection and bimanual manipulation. First, the scene is analyzed to classify objects into object arrangement patterns, i.e., stacked, shelved, and disorganized. Second, a collapse prediction network is used to predict the safety of different actions. Third, the proposed strategy selects a bimanual action plan from a list of potential safe actions to organize the shelf, if necessary, and place the object on the shelf.

5.2.1 Objects Arrangement Classification

The first step of our framework regards classifying the object arrangement. We used YOLOv3 (You Only Look Once, version 3) [91], a real-time object detection algorithm that identifies specific objects in a picture, to classify clusters of objects into object arrangement patterns.

As shown in Figure 5.3, the arrangements of objects are defined as one of the following four classes: stacked C^h, shelved C^v, disorganized right C^r, and disorganized left C^l. C^h and C^v denote horizontally and vertically arranged patterns, respectively, and C^r and C^l are disorganized patterns that lean to the right and left, respectively. In the case of a single object, it will be classified as C^v.

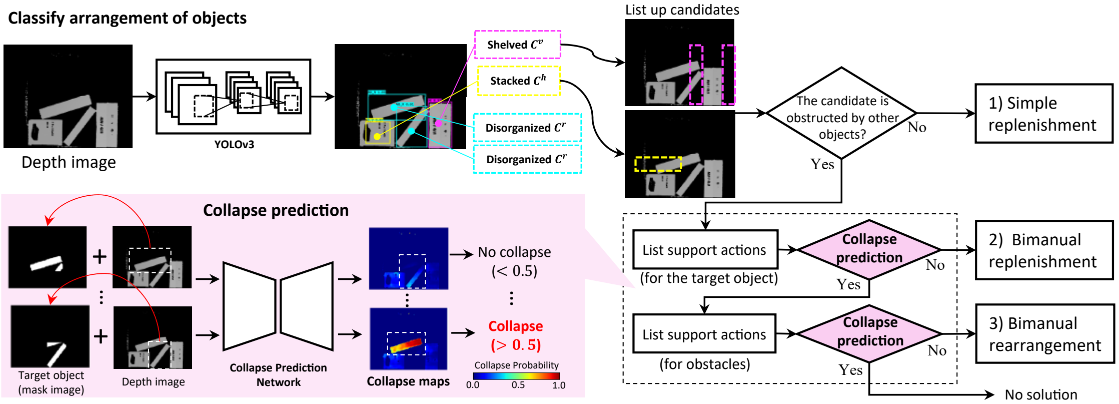


Figure 5.2: Overview of our bimanual robotic replenishment pipeline, which consists of (1) shelf scene classification into organized/disorganized arrangement using YOLOv3 [91], and (2) action planning based on a collapse prediction network that predicts the probabilities of collapse from a shelf in the form of a heatmap. The depth image is captured from the 3D vision sensor and then fed to YOLOv3 to classify the shelf scene into organized/disorganized arrangements (top left). The flowchart on the right shows the action planning to replenish an object based on the classification results. Each action is evaluated with the collapse prediction network (bottom-left) to avoid the objects from collapsing during bimanual manipulation.

YOLOv3 also generates a bounding box of the cluster. We define bounding box B_i ($i = 1, \ldots, N$) as follows (N is the number of the generated B_i):

$$B_i = (x_i, y_i, w_i, h_i) \tag{5.1}$$

where (x_i, y_i) denotes a center position, and w_i, h_i denotes width and height, respectively. To apply YOLOv3 for our object arrangement pattern classification, we used the weighted model pretrained on ImageNet [84] and pretrained the model with real depth images. A depth sensor acquired the depth image with 256-step grayscale, which showed 5–10 rectangular objects on a shelf. Here, we do not use RGB images but depth images with the assumption that the object's textures are unnecessary for classifying the object arrangement patterns. To distinguish the disorganized pattern as either C^r or C^l, our training process does not use data augmentation by randomly flipping the images. We used 500

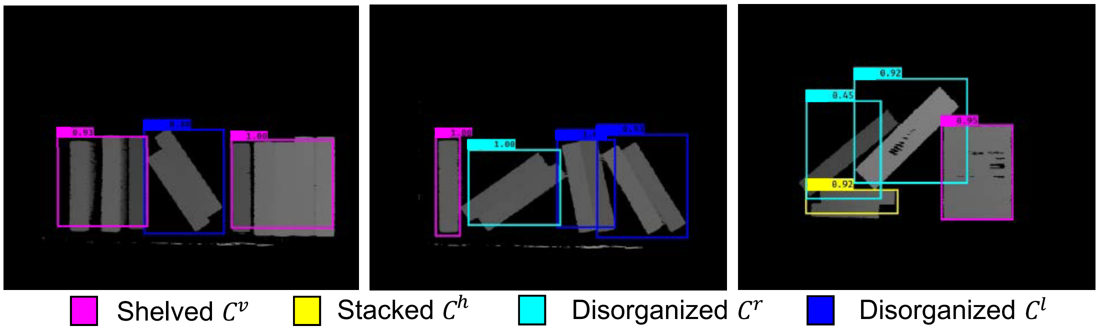


Figure 5.3: Detection results of object arrangement category using YOLOv3 classification. As shown in these images, each object arrangement is categorized into four classes: stacked C^h, shelved C^v, disorganized right C^r, and disorganized left C^l.

images to train the model, and annotations were performed manually. The confidence score was empirically set to 0.30, and the threshold of the intersection over union (IoU), which is the accuracy of the individual identification of the bounding box, was set to 0.45. The training at 100 epochs took 1 h on a system running Ubuntu 16.04 with an Intel Core i7-9700F CPU clocked at 3.00 GHz and a single NVIDIA GeForce RTX 2060 SUPER graphics card with CUDA 10. The results are presented in Figure 5.3.

5.2.2 Collapse Map

We propose a collapse prediction network that can manipulate an object without the collapse of neighboring objects. The network outputs a heatmap that shows the pixel-wise collapse probabilities, that is, the collapse map. In this section, we first describe the architecture of our network model and then introduce the data collection and training settings applied to generate our model.

Network Architecture

In our previous study [90], we proposed an approach for shelf picking to assess whether extracting an object is possible based on a collapse prediction network. However, the use of the network is limited to a specific bimanual action to extract a target object while supporting an adjacent object; thus, we can only hold the adjacent object so as not to move based on the result of the collapse prediction. In the present study, we improve the collapse prediction network to directly determine the potential of an object falling from a shelf when removing the specified object with a single arm. This enables us to plan a sequential approach for replenishment based on the collapse probability.

The network is comprised of an encoder and decoder. The input data were a depth image of the shelf scene and a target mask image (binary image) of the specified object, in which the region representing the target object was set to 1 and the other regions were set to 0. The encoder network has two pipelines, as shown in Figure 5.4. One network extracts features from a depth image based on the convolutional layer of VGG-16 [83]. The other has five convolutional layers to compress the binary mask image. The outputs of the two pipelines are concatenated and fed into the decoder network. Finally, the computed collapse map is upsampled to match the size of the input depth image. The first branch has a skip architecture to improve the semantic segmentation performance. The input image was 256×256 grayscale and normalized in advance. Similarly, the mask image size was 256×256.

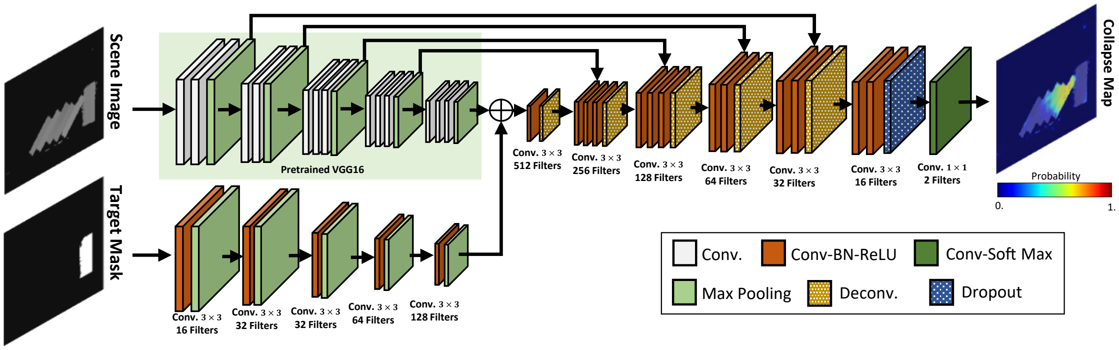


Figure 5.4: Network architecture. The collapse map network receives both a depth scene and a binary mask of a target object as the input. The output of the collapse map network is a heatmap, which shows the probability of position changes, e.g., an object turning over or falling down.

Dataset

For dataset generation, we used a maximum of 10 rectangular objects of various sizes in PhysX [82]. As shown in Figure 5.5, five to nine objects were first randomly sampled, which were initially positioned in an organized arrangement pattern as either C^h and C^v. Half of these objects were then assigned random poses to generate a disorganized arrangement. Subsequently, a target object was randomly selected and removed from the shelf. We then checked the positions of all objects, except for the target object, after the target object was removed from the shelf and the other objects reached a stable state. The objects that move during this operation constitute a collapse mask (binary image), in which the regions representing those objects were set to 1 and the other regions were set to 0 as shown in Figure 5.5. If the change of the objects' center position exceeds the threshold, we judge the objects to be moved. Note that we empirically set the threshold to 6.4 mm. To train our network on a pixel basis, we collected a depth image, target mask, and collapse mask, where the images were rendered from the recorded results. The depth image shows the initial arrangement before the selected target object is removed. The target mask shows the selected target ob-

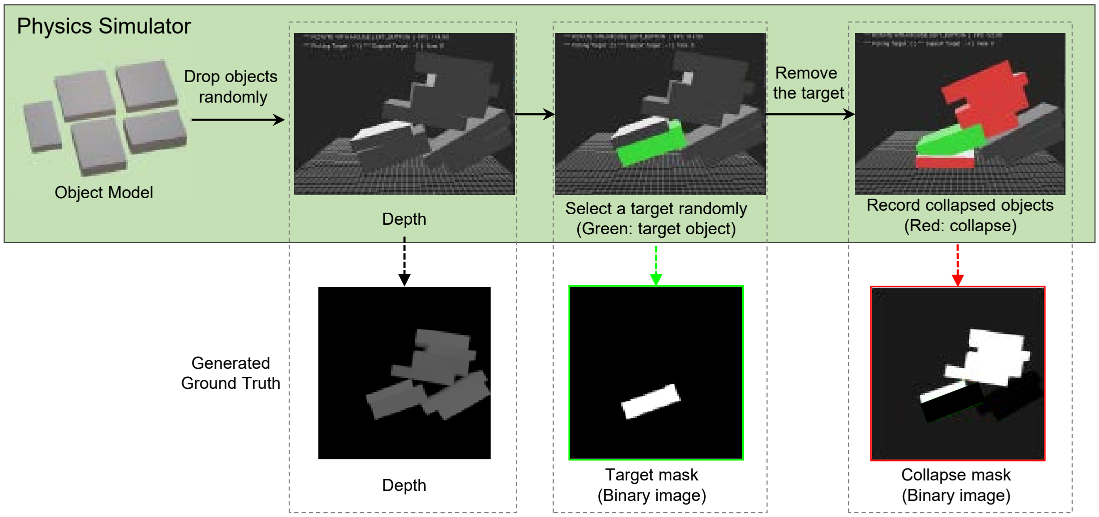


Figure 5.5: Dataset generation procedure. In our method, nine novel objects are used from five types of objects. First, the initial objects are stacked randomly in vertical or horizontal status. Next, the target object is removed, and the simulator monitors the movement of the other objects. Finally, the moving objects are marked as the collapse region, and one dataset is generated through a single simulation (bottom of the figure: depth scene image, target mask, and collapse mask).

ject, and the collapse mask shows the objects that moved after the selected target object was removed. Finally, we empirically set the simulation parameters according to their actual movements as follows: we set the coefficient of static to 0.9, dynamic friction to 0.8, the coefficient of restitution to 0.1, and the density to 1.0 $\mathrm{kg/m^3}$.

Implementation details

We built a dataset of 22,400 training images generated from the simulator described in Section 5.2.2 and trained the collapse map network. To eliminate the discrepancies between the real and synthetic depth images, noise was randomly added to the generated depth images.

We used a batch size of 32 (700 iterations) and the Adam optimizer [87] with a learning rate of 1.0×10^{-4}. The other Adam hyperparameters were set as the default values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The training at 50 epochs took 8 h on a system running Ubuntu 16.04 with an Intel Core i7-9700F CPU clocked at 3.00 GHz, and a single NVIDIA GeForce RTX 2060 SUPER graphics card with CUDA 10. The network achieved a processing time of 0.02 s or less to generate one collapse map.

5.2.3 Shelf Replenishment

Shelf replenishment requires action planning to place an object along the arrangement. However, moving an object in a densely stacked scene, i.e., a shelf, involves the risk of dropping the other object because safe manipulation on a shelf is complicated, especially considering the dynamics. To solve the problem, we formulate the robotic action as the manipulation within the limit of the bounding box based on the arrangement classification. Here, the collapse map can detect the risk of handling the object inside the bounding box. If there is no risk, we can provide the replenishment strategy without considering the strict dynamics.

As shown in Figure 5.3, the shelf scene is represented by bounding boxes B_i and classes. To find sufficient space to replenish an object, we define three placement candidates as rectangles on the top, left, and right of each target bounding box B_i in case the class is a stacked C^h or shelved C^v. Here, let B_i^{top}, B_i^{left}, and B_i^{right} denote the bounding box of the placement candidates to be placed on the shelf. B_i^{top} denotes the area where an object can be stacked on C^h, which we describe as

$$B_i^{top} = (x_i - \frac{w_i}{2} + \frac{w'}{2}, y^i - \frac{h_i}{2} - \frac{h'}{2} - g, w' + 2g, h') \tag{5.2}$$

B_i^{right} and $B_i{}^{left}$ denote the areas where an object can be placed on the right and left sides of C^v, respectively, which we describe as

$$B_i^{right} = (x_i - \frac{w_i}{2} - \frac{w'}{2} - g, y^i + \frac{h_i}{2} - \frac{h'}{2}, w' + 2g, h') \tag{5.3}$$

$$B_i^{left} = (x_i + \frac{w_i}{2} + \frac{w'}{2} + g, y^i + \frac{h_i}{2} - \frac{h'}{2}, w' + 2g, h') \tag{5.4}$$

where w' and h' are the height and width of the area to secure space, respectively, and g is the thickness of the fingers of the gripper. Each arrangement has the potential to place an object on B_i^{top}, B_i^{left}, and B_i^{right} unless the bounding box is outside the shelf. Note that the object is known, which fits in the secure area (the size of $w' \times h'$), and each candidate is excluded when it exceeds the limit of the working space. In the present study, the size of the inside of the working space (the shelf) is W330 × D280 × H330 mm.

As shown in Figure 5.2, based on the predicted collapse map for a target object and the prediction for each action, we assume three manipulations for replenishment.

Simple Replenishment

Firstly, we check that there is sufficient space in the candidate area (B_i^{top}, B_i^{left}, or B_i^{right}) to place an object on the shelf. Note that the size of the object placed on the shelf is known. When this condition is satisfied (i.e., there is no object inside the candidate area), the robot places the object at the center position of the candidate area.

Bimanual Replenishment

Secondly, when objects occupy all candidates, the objects must first be removed from the candidates. In the present study, we define this action as the supporting action in this study. Let B_s denote the arrangement overlapping the candidate (B_i^{top}, B_i^{left}, or B_i^{right}). On the collapse map targeted at the bounding box B_s, an area with a probability equal to or higher than the predefined probability threshold is defined as the collapse region R_s. If $\mathrm{IoU}(B_j, R_s) < th \quad (j \neq s, i)$, the objects in B_j should be stable after moving B_s; that is, B_s is movable. $\mathrm{IoU}(\cdot)$ denotes a function that outputs the IoU, and th is a threshold value. The supporting action is defined as moving B_s horizontally to create space in a cluttered scene. The starting point p^{start} and goal point p^{goal} are defined as follows:

$$p^{start} = (x_s, y_s) \tag{5.5}$$

where (x_s, y_s) is the center position, and x_s and y_s denote the coordinates. Then, we then define the goal point subject to the target position as follows:

$$p^{goal} = \begin{cases} (x_s + \frac{w'}{2} + \frac{w_s}{2} + g, y_s) & \text{if } B_s \text{ on } B_i^{right} \\ (x_s - \frac{w'}{2} - \frac{w_s}{2} - g, y_s) & \text{if } B_s \text{ on } B_i^{left} \\ (x_s, y_s - \frac{h'}{2} - \frac{h_s}{2} - g) & \text{if } B_s \text{ on } B_i^{top} \end{cases} \tag{5.6}$$

where g denotes the margin of the gripper fingers in the experiment. Figure 5.6 shows the bimanual replenishment process. When the class of B_s is either C^r or C^l, the objects in B_s are rotated, aligned with the organized arrangement, and moved to the goal point with one robotic arm. While holding them for safety, the object is then placed in the placement candidate with the other robotic arm.

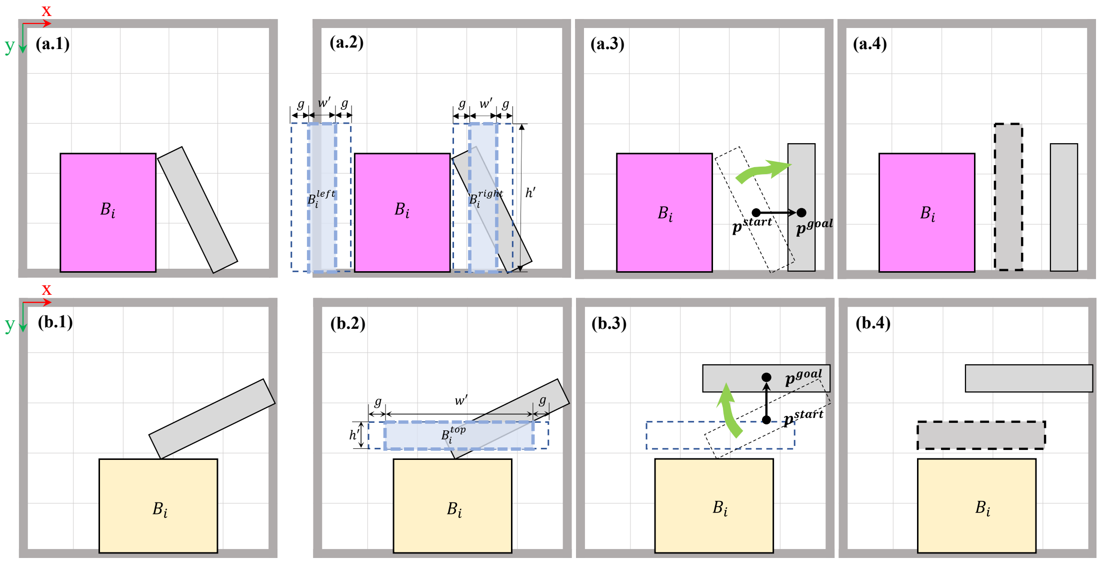


Figure 5.6: Example executions of bimanual replenishment: (a.1–a.4) Replenishment for a shelved scene. The robot moves the disorganized object to the right or left to place an object aligned with the shelved objects. (b.1–b.4) Replenishment for a stacked scene. The robot lifts the disorganized object to place an object on the stacked objects.

Bimanual Rearrangement

Finally, in case there is no candidate that satisfies the requirements, we repeatedly consider the rearrangement. If $\mathrm{IoU}(B_i, R_s) \geq th \quad (i \neq s)$ for all candidates, then supporting the objects in B_i increases the collapse risk. We select the arrangement $B_i\{i = 1, \ldots, N\})$ that has the highest overlapping rate to the collapse region R_s.

$$k = \mathop{\arg\min}_{i \in 1,\ldots,N} |\mathrm{IoU}(B_i, R_s)| \tag{5.7}$$

Here, we generate the collapse map targeted at the bounding box B_k and calculate the collapse region R_k. As mentioned above, If $\mathrm{IoU}(B_j, R_k) < th \quad (j \neq s, i, k)$, B_k is movable. The supporting action is defined as moving B_k horizontally to avoid object collapse. The starting point $p^{start,\dagger}$ and goal point $p^{goal,\dagger}$ are defined

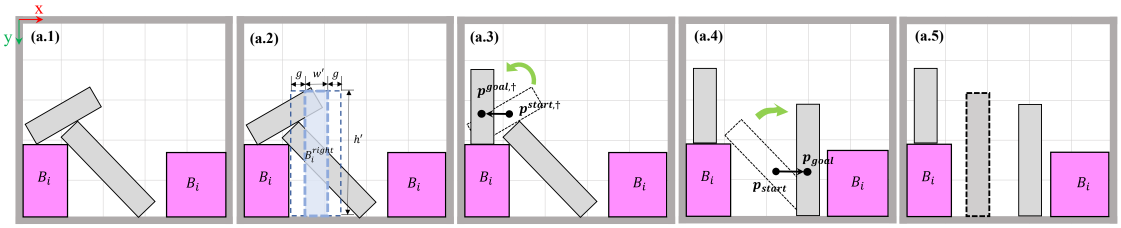


Figure 5.7: Bimanual arrangement example: (a.1–a.5) When more than two objects obstruct the replenishment, we need to move them with multi-step actions. The robot moves the objects one by one to make space to place an object.

as follows:

$$p^{start,\dagger} = (x_k, y_k) \tag{5.8}$$

$$p^{goal,\dagger} = \begin{cases} (x_s - \frac{w_s}{2} - \frac{w_k}{2} - g, y_k) & \text{if } x_k < x_s \\ (x_s + \frac{w_s}{2} + \frac{w_k}{2} + g, y_k) & \text{if } x_k \geq x_s \end{cases} \tag{5.9}$$

When the obstacle for the supporting action is moved, it is possible to safely move B_s (Figure 5.7). If B_k is not movable, supporting it is also required. However, the bimanual robot cannot place the object while holding two or more objects. In the present study, we excluded such cases from consideration.

5.3 Experiments and Results

In this section, we report the implementation details, experimental results, and the benchmark of the collapse network for evaluating the performance of our proposed method.

5.3.1 Predicting the Collapse Map

Figure 5.8 shows the collapse maps results with the validation data. We report the pixel accuracy to quantify the classifications and calculate these metrics as follows:

$$\text{Pixel Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{5.10}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{5.11}$$

where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively, counted on a pixel basis. In our evaluations, we assume that a pixel is classified as a collapse region when the probability is higher than 0.5.

We validated our prediction model using 1000 simulated images created from the simulator. Compared to our previous baseline model (based on FCN-8s [81]), we achieved a pixel accuracy and IoU score of 0.982 and 0.668, respectively, as shown in Table 5.1. A comparison between the other parameters and these results shows that the batch size parameter was chosen appropriately (Table 5.1). Based on this result, we set the batch size to 32 and used the transfer learning of VGG-16, which was pretrained with ImageNet. We further note that our model infers that the object moves under physical dynamics; however, it achieves similar or better IoU scores than those of related studies [92, 93].

5.3.2 Robotic Experiments

The efficiency of the proposed method was validated using real robotic experiments. We used MOTOMAN-SDA5F from Yaskawa Electric Corp. for our ex-

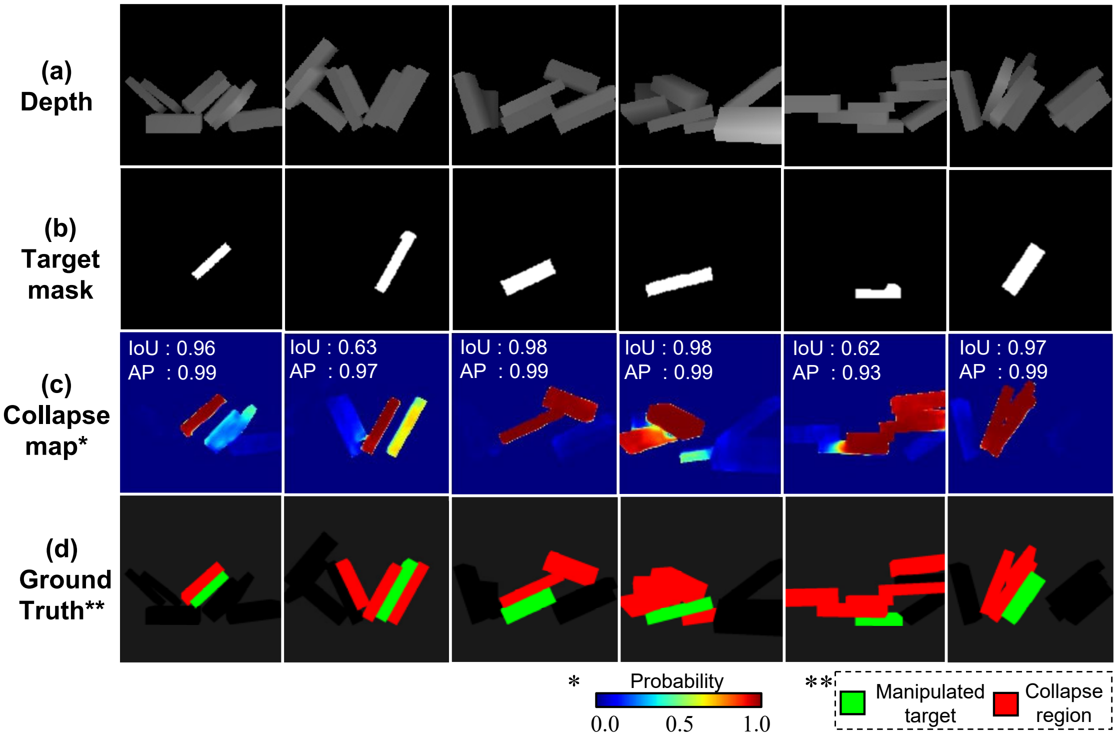


Figure 5.8: Collapse maps results: (a) Depth image showing the depth from a viewpoint in grayscale. (b) Target mask that shows a targeted object (white). (c) Collapse map generated from the collapse prediction network. (d) Ground truth, which consists of collapse regions (red) and target object (green).

Table 5.1: Performance comparison between collapse predictions for each setting.

Method	PA *	IoU **
FCN-8s-based	0.941	0.461
Ours (Batch size = 32)	**0.982**	**0.668**
Ours (Batch size = 16)	0.981	0.662
Ours (fine-tuned, Batch size = 16)	0.980	0.640
Ours (fine-tuned, Batch size = 32)	0.957	0.545

* Pixel Accuracy. ** Intersection Over Union.

periments [88]. The SDA5F has 15 degrees of freedom (DoFs): 7 DoFs per arm and one DoF for the waist. The robot was programmed using Choreonoid [94] and graspPlugin [95]. Two Robotiq gripper 2F-140 adaptive grippers [96] were used, which were installed at the arms of the SDA5F. The 2F-140 adaptive gripper is an underactuated parallel gripper. We used a YCAM3D-10L from YOODS Co. Ltd., Yamaguchi, Japan [89], which is a depth camera based on the phase shift method. We obtained a depth image from YCAM3D-10L. We used a median filter to smooth the image for noise removal from the real data. Each original image was resized to 256 × 256 pixels. We used 4–6 different rectangular objects. The objects were presented to the robot on the shelf, and a similar scene was maintained in the simulator.

We report the results of the experiments with a real robot in three typical scenarios: shelved, stacked, and random. The objects were randomly placed in each scenario. Success was defined as the case in which the replenishment of an object was completed. In a sequence of 100 experiments, 68 trials succeeded in obtaining the entire result (68.0%). From the viewpoint of each arrangement, the success rates were 57.5%, 84.0%, and 30.0% in the stacked, shelved, and random scenes, respectively. Moreover, we evaluated the performance of our collapse prediction. Our method without the collapse prediction showed comparatively lower success rates. In particular, it performed poorly on rearrangements, which required moving objects inside the shelf, compared to the case when using the collapse prediction. In a sequence of 25 experiments, only 11 trials achieved the entire result (44.0%), and the success rates were 50.0%, 60.0%, and 0.0% in the stacked, shelved, and random scenes, respectively. Table 5.2 presents the corresponding statistics.

Figure 5.9 shows snapshots of the experiment, where the object was initially placed vertically on the shelf. Figure 5.9 (a.1, b.1) show two scenes within the experiment. The depth images shown in Figure 5.9 (a.2, b.2) were classified by our fine-tuned YOLOv3. The steps of these experiments are depicted in Fig-

Table 5.2: Robotic experiment results.

	Stacked	**Shelved**	**Random**	**Total**
Success w/ Collapse Prediction	23/40 (57.5%)	42/50 (84.0%)	3/10 (30.0%)	68/100 (68.0%)
Success w/o Collapse Prediction	5/10 (50.0%)	6/10 (60.0%)	0/5 (0.0%)	11/25 (44.0%)

ure 5.9 (a.3–a.8, b.3–b.8), where the candidate placement can be derived by placing the objects on the left or right according to the display identified as shelved. An object leaning to the left (Figure 5.9 (a)) or to the right (Figure 5.9 (b)) is located at the planned placement point. The object was grasped by the right-hand gripper, rotated to align it, and moved to the right. We assumed the diagonal direction of the bounding box to be the angle of inclination of the object under the prior positional information (C^l/C^r). Additionally, snapshots of the experiment in which the object was placed horizontally on the shelf are shown in Figure 5.9 (c,d). Figure 5.9 (c) shows how the obstructing object was grasped with one hand, lifted, and placed on top of the other hand. In Figure 5.9 (d), the object was placed on top of the objects on the shelf without the need to use the right hand, as no other object was detected.

If the obstacle cannot be moved off the shelf with one hand, we can select the multi-step motions to organize the objects with dual arms, as shown in Figure 5.10 (a.1–a.8, b.1–b.8). Using the collapse maps for each object, we selected the supported and moved objects that could be securely manipulated and well organized.

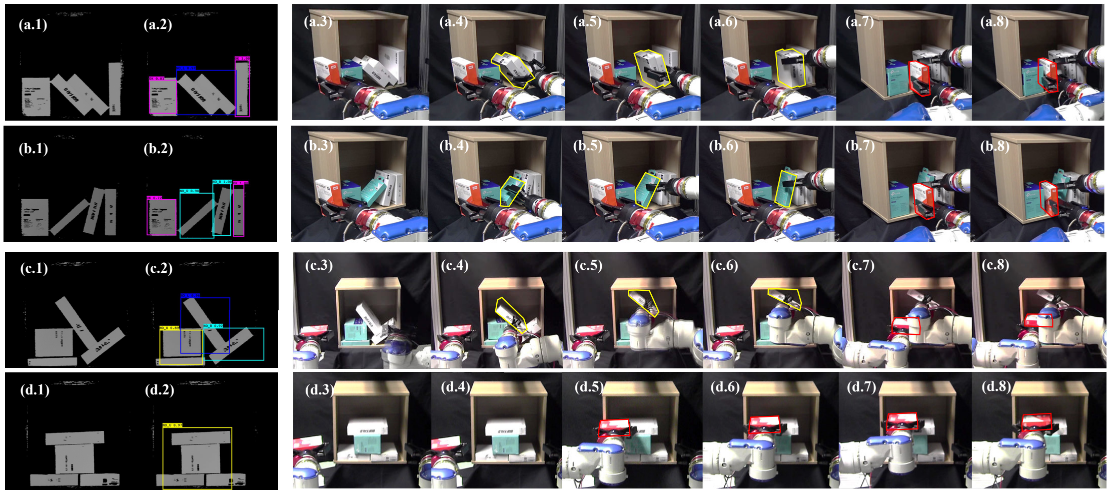


Figure 5.9: Snapshots of the experiments: To align an object with the vertical arrangement in (a.1, a.2, b.1, b.2), the robot horizontally moves the other objects that occupy the space of the shelf, using the other arm, see (a.3–a.8, b.3–b.8). To align an object with the horizontal arrangement in (c.1, c.2, d.1, d.2), the robot lifts the other objects that occupy the space on the shelf using the other arm, see (c.3–c.8, d.3–d.8)

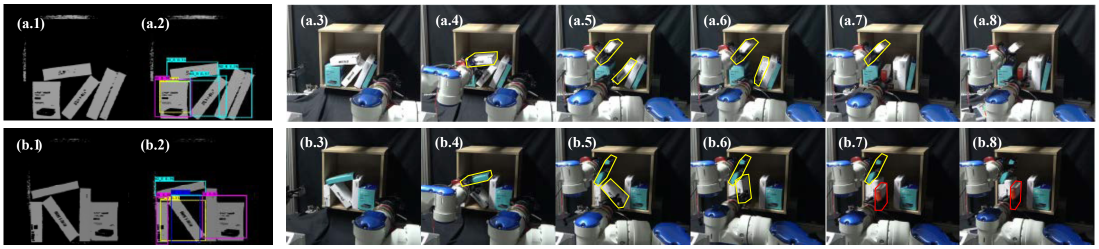


Figure 5.10: Snapshots of the specific scenarios in (a.1, a.2, b.1, b.2), which require multi-step actions to replace an object and to make space for replenishment in (a.3–a.8, b.3–b.8).

5.4 Discussion

This study analyzed an unknown shelf display to predict the risk of collapse during a replenishment operation. This enables the robot to replenish a shelf with an object by selecting a strategy based on the situation. Experiments on two typical and complex arrangements confirmed that the bimanual action plan replenished the object while dealing with disorganized arrangements.

Our proposed method has practical relevance when considering the difficulty of maintaining organized arrangements in retail and warehouses. When objects were not organized, Lee et al. [25] and Nam et al. [26] conducted rearrangement actions similar to ours and approached the target object, assuming that all objects were on the same plane. By contrast, our proposed method can move objects without collapse, even if the objects overlap with each other. In terms of safety, Zhang et al. [63] and Panda et al. [66] acquired knowledge about the geometrical structure of a scene to individually detect the support relation. However, they refer only to the safety of operations on an object with no support, that is, an object placed on the top. In contrast, our method quantitatively assesses all objects on the shelf.

It should be noted that the success rate of our method for random scenes is low (approximately 30%), as shown in Table 5.2. However, we conducted the experiments under strict conditions without the object collapsing, as opposed to [10, 37, 50, 57]. The previous studies, in fact, required the system on a case-by-case basis for the object collapse. Compared with motion planning without collapse predictions, our proposed method can perform successfully on a complex scene using a bimanual robot. Despite its many advantages, there are some limitations associated with the present study. First, we only evaluated the risk of collapse in an instantaneous and static scene to determine the sequential action for replenishment. Therefore, because it cannot handle the collisions and the dynamics that may occur during object movement, the present study assumes that the target object for manipulation is limited to within the bounding box in order to avoid contact between the objects. In other words, this study had minimum space requirements, which makes it difficult to achieve the necessary conditions in narrow and dense shelf environments. In future, handling items requiring high dexterity will need the integration of reactive grasping control and motion planning to perform such tasks, even with grippers with limited dexterity, as shown in [50]. Second, our planner assumes that the robot has two arms and that when one arm moves an object, the other arm supports an obstacle. However, if there are too many disorganized objects, the support action with only one arm is insufficient, and collapse cannot be avoided. Our framework was limited to using only one support action. Accordingly, the mutual support relations among the objects should be analyzed, and an action planner developed based on a search algorithm to deal with many objects. Third, it is difficult to avoid interference between arms in a confined environment. Both arms tend to be close to each other, which makes the computation of inverse kinematics difficult. Particularly, in this method, we do not consider the dynamics and physical contact when considering the stability of learning the network to predict the collapse. Therefore, to increase the success rate, we should use a simulation to consider the robot arm and train the collapse prediction network by considering external interference and self-interference.

We assume that replenishment is to place an object on a shelf so that it is aligned with the typical arrangement in the warehouse or retail store. However, the display becomes disorganized as the objects move in or out of the shelf. The collapse prediction network makes it possible to evaluate the risk of any manipulation to replenish an object without collapse. Additionally, we solved the difficulty of organizing the shelf using a simple algorithm based on collapse predictions. However, because we must handle various objects in different environments, further verification of our proposed method is necessary. We should also examine whether it can be applied to other research fields. Thus, we intend to develop a sequential prediction network that considers the dynamical transition of objects in order to apply our approach to other tasks with different objects, for example, a policy to consider objects in an unstable pose or entangled objects. Similarly, considering the other damage source of items is also necessary for safe shelf manipulation, such as breaking an item with the robot hand's clamping force. Therefore, another interesting future study would be to use the property of the gripper to learn the collapse and the graspability of objects for further adaptability in realistic scenes [68, 97, 98].

5.5 Summary

This chapter presented a shelf replenishment system that selects the safest action based on a collapse prediction estimator. The collapse prediction network generates a probabilistic map from scene images and actions, making safe manipulation possible. In addition, the proposed method plans the best action based on single-arm or bimanual manipulation, making it possible to deal with complicated arrangements. In experiments using a real robot, I demonstrated the efficiency of The method for shelf replenishment.

In future work, I plan to extend the implementation of both the network and the data collection processes: (1) to further deal with any object shape and more disorganized arrangements, such as in retail stores and kitchens; (2) to use the robot properties in the simulations to estimate the physical contacts; and (3) to develop a prediction network to help analyze the state of stacked objects.

Chapter 6

Multi-step Object Extraction Planning from Clutter based on Support Relations

6.1 Introduction

In a logistic warehouse, human workers usually pick and place products from a shelf into a box for service delivery. To replace this logistic operation with a robot, the robot must be able to search for the target product and safely extract it from a shelf wherein many products are randomly placed. Thus far, although several learning-based methods [1, 5] have designed the motions of robots for picking objects from clutters, extracting the target object from a shelf imposes a new challenge. When extracting the target object from the clutter on a shelf, a robot needs to safely extract the object while preventing the fall of neighboring objects. To address this issue, we proposed a single-step motion planning framework for selecting and extracting target objects without collapsing the pile [90].

Figure 6.1 shows a scenario where our multi-step motion planner is effective. Here, the robot extracts the target object, box 0, marked in white from the pile. Boxes 1 and 2, however, are stacked on the target. The robot is expected to remove these boxes and subsequently extract the target object. To this end, we need information regarding where box 2 is supported by box 1 and box 1 by box 0.

In this study, the support relations of the objects in the clutter are expressed graphically. For example, the support relations of the boxes shown in Figure 6.1 can be expressed by a hierarchically structured graph. To extract box 0 from

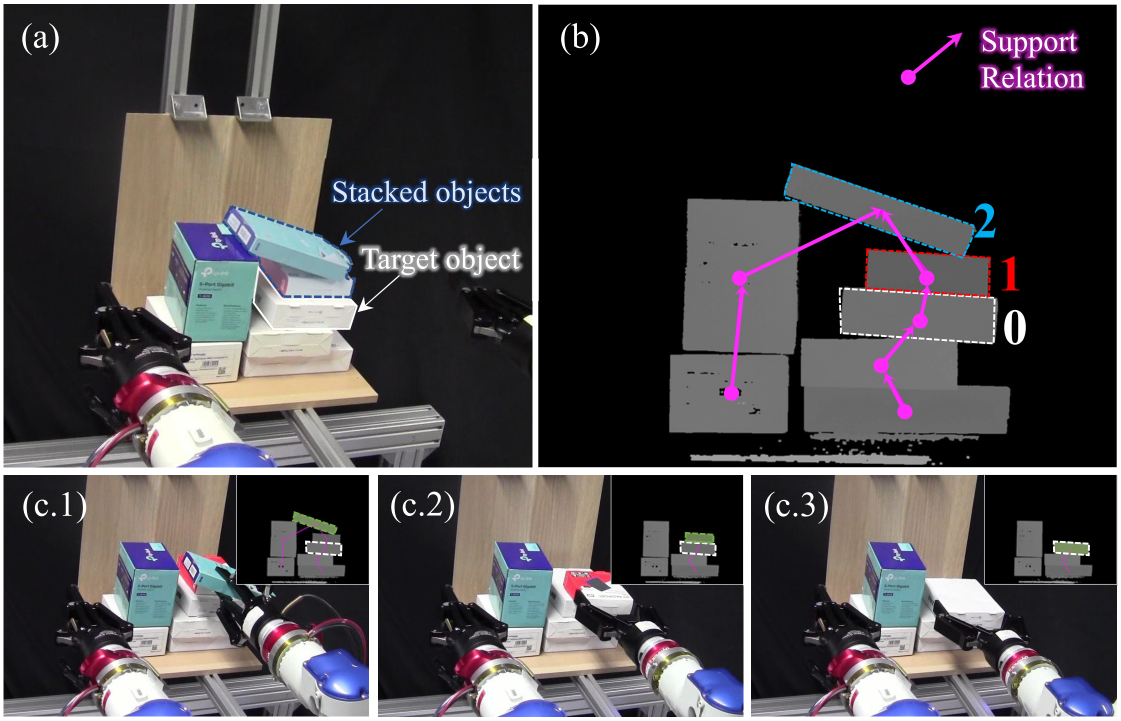


Figure 6.1: Safe object extraction based on support relations. The support relations in the right upper are visualized in a collapse prediction graph. To extract the target object marked in yellow, the robot extracts the object in a safe extraction order.

the clutter, the graph indicates that boxes 2, 1, and 0 should be extracted in this order. This study proposes a novel multi-step object extraction planning from clutters using graphs obtained by estimating the support relations of objects in the clutter.

The proposed multi-step object extraction planning contains three major components: (1) a collapse predictor (CP) that predicts the support relations between two objects from the clutter by using depth images, (2) a collapse prediction graph (CPG) that consists of the support relations to ensure safe extraction, and (3) a multi-step extraction planner based on the CPG. We infer support relations using a CP that is based on a deep neural network proposed in [99]. The predictor can predict the movement of objects when extracting an object

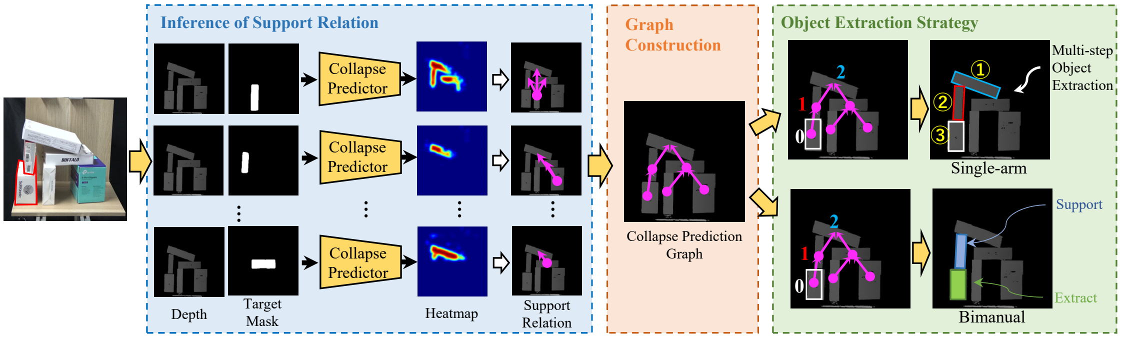


Figure 6.2: Proposed system overview. The collapse predictor (CP) outputs the probability that other objects might fall. Support relations are estimated from this result and graphically represented. Based on the collapse prediction graph (CPG), the robot picks objects successively from the pile. In bimanual manipulations, the robot directly extracts the target object by holding the supported object if the target object is supporting only a single supported object.

and identify supported objects for different targeted objects using only depth images. The CPG consists of inferred support relations and provides the best extraction planning by searching for the target object via a recursive traversal search. Additionally, to efficiently extract stacked objects, we propose a novel bimanual extraction planning based on the CPG and validate typical scenes.

The rest of this chapter is organized as follows. Section 6.2 describes the proposed method. In Section 6.3, we evaluate robotic experiments. In Section 6.4, we discuss the limitations and possible future extension. Finally, we present our conclusions and future work in Section 6.5.

6.2 Methodology

An overview of the multi-step extraction planning is illustrated in Figure 6.2. The proposed framework consists of a CP, the inference of support relations, and a safe extraction strategy. First, we begin with the details of the CP proposed in our previous study [99] (Section 6.2.1). Then, we infer support relations, which represent the physical relationship between two objects with the CP, given a depth image captured from a shelf scene (Section 6.2.2). By concatenating all the support relations, we create a CPG to determine the object that can be extracted from the pile. Herein, we generate a multi-step plan to extract the target object (Section 6.2.3). Furthermore, we propose bimanual manipulation based on the CPG for efficiently extracting the target object. The proposed method is described in the following section.

6.2.1 Collapse Predictor

CP is a deep neural network based on the model proposed in [99] and is further customized to infer support relations in cluttered environments. This section describes the network architecture, data collection process, and training details. Our method needs sufficient accurate predictions to infer physical relations among objects. Therefore, we extend the dataset and adjust the network parameters to improve the accuracy compared with that of the previous studies. The details are as follows.

Network Architecture

The neural network architecture includes two encoders and a decoder. The scene encoder compresses the input of depth images (256 × 256) with a grayscale using the VGG-16 [83] (until the last convolutional block), pre-trained with ImageNet [84]. The first ten convolutional layers are fixed in training to transfer feature extraction. The target encoder converts target masks (256 × 256) into

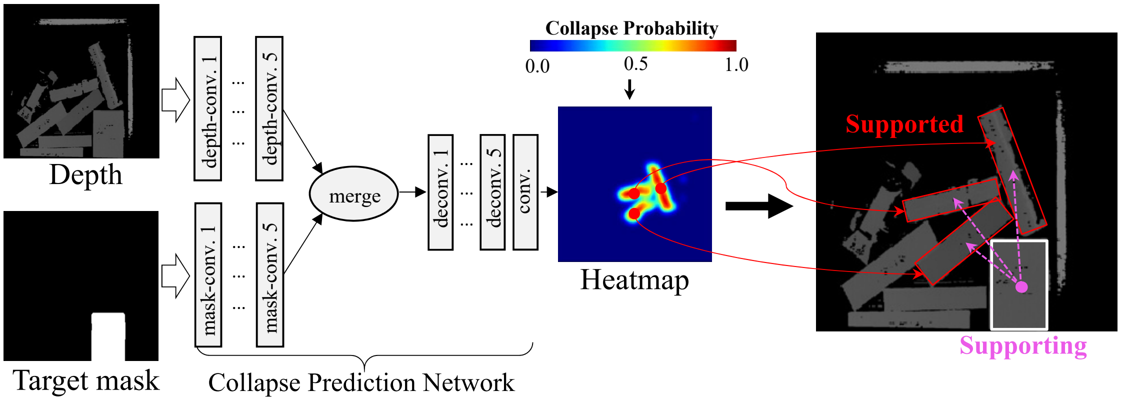
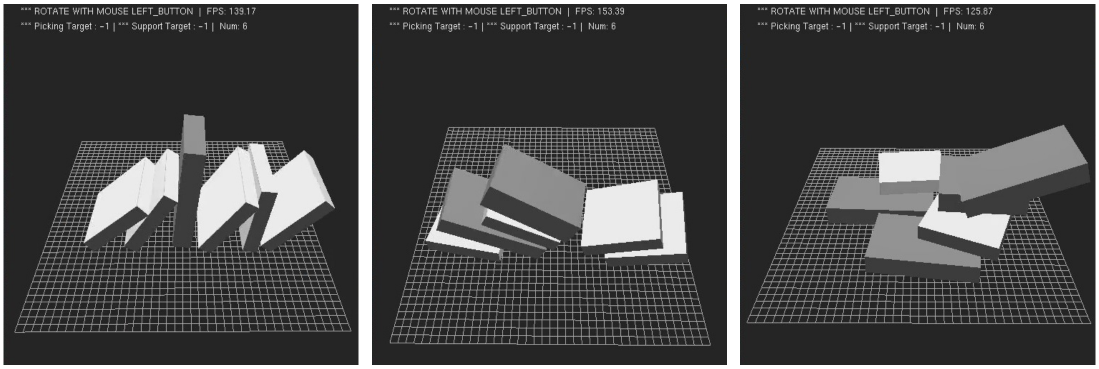


Figure 6.3: Architecture of the CP consists of two encoders that compress the depth image (256 × 256) and target mask (256 × 256). These outputs are concatenated, and a decoder network generates a heatmap (256×256), showing the probability of an object falling. Finally, the support relation is inferred based on the heatmap.

feature maps using five convolutional (Conv) layers, each followed by batch normalization and rectified linear unit activation layers, respectively. The convolution layers comprised 16, 32, 32, 32, and 64 layers. The Conv layers output latent codes (8 × 8 × 64). The decoder upsamples the latent code concatenated with both outputs, the head of VGG-16 (8×8×512) and target encoder (8×8×64), using five Conv layers and one Conv layer. The networks output a heatmap (256 × 256), which shows the probability of falling objects in pixels. The architecture is shown in Figure 6.3.

Generating Training Dataset

In this section, we introduce the process for our collapse dataset generation. A PhysX physics simulator [82] simulates object removal. First, we place the objects in any of the following scenes: (a) shelved, (b) stacked, and (c) random (see Figure 6.4). In the shelved scene, we arrange objects vertically at random


*** ROTATE WITH MOUSE LEFT_BUTTON | FPS: 139.17
*** Picking Target : -1 | *** Support Target : -1 | Num: 6
*** ROTATE WITH MOUSE LEFT_BUTTON | FPS: 153.39
*** Picking Target : -1 | *** Support Target : -1 | Num: 6
*** ROTATE WITH MOUSE LEFT_BUTTON | FPS: 125.87
*** Picking Target : -1 | *** Support Target : -1 | Num: 6

(a) Shelved (b) Stacked (c) Random

Figure 6.4: Simulating the extraction of a box from a clutter. Each scene is generated by adjusting the object poses/positions (a, b) and random pose (c) and dropping on the random points (c) on top of others (b) and horizontally (a).

intervals, i.e., bookshelves. In the stacked, we randomly place objects on each object. In the random, we drop some objects at random poses and heights. At each simulation, we use 5-10 objects in five types of rectangular objects. Then, a target object is randomly selected and removed from the shelf. During data generation, if the change in the object's center position exceeds a threshold, the objects are moved. We empirically set the threshold to $5.0mm$, coefficient of static friction to 0.9, coefficient of dynamic friction to 0.8, coefficient of restitution to 0.1, and density to $1.0kg/m^3$. Notably, the viewpoint is set to face the shelves, implicitly assuming that the direction of gravity is downward.

We collect a depth image, target mask, and collapse-labeled image. The depth image is a 256×256 grayscale height map showing an initial scene wherein objects are placed. The target mask is a 256×256 binary image wherein all the pixels are black, except that of a target object. The collapse-labeled image is also a 256×256 binary image annotated in other objects after the target is removed.

For data collection, we executed all the simulations in 10,000-shelved, $10,000$-

Table 6.1: Comparison with our previous work.

Model	Pixel Acc.	IoU	Prec.
Previous work	0.984	0.559	0.734
Our method	**0.985**	**0.578**	**0.740**

stacked, and $30,000$ random scenes. The dataset of $50,000$ simulations is split into training (90%) and validation (10%). As a test set, we prepared 1,000 simulated data in random scenes.

Training Details

The batch size is 24, learning rate is 0.001, and total epoch is 100 with an early-stopping with loss monitored. In this study, the training process stopped at 58 epochs. Moreover, the background occupies the heatmap within a wide range, and the network estimates the risk of collapse that is lower than the real. Herein, we used the focal loss from RetinaNet [100] as follows:

$$L(y) = -\alpha_y(1 - y)^\gamma \log(y), \tag{6.1}$$

where y is the probability that the predicted labels are equal to the ground truth $\in \{1, 0\}$. $\alpha_y \in [0.0, 1.0]$ is the focusing parameter for y. Intuitively, this scaling factor decreases the contribution of easy examples, i.e., a black background. In our training, α_1 and α_0 are set to 0.25 and 0.75, respectively, and γ is set to 2.0. Table 6.1 compares the proposed model with that of the previous work. The improved model achieves high pixel accuracy, Intersection over Union (IoU), and precision values by using the focal loss. Therefore, we use a weighted model to predict object collapse in later sections. Figure 6.5 illustrates the outputs of the trained network.

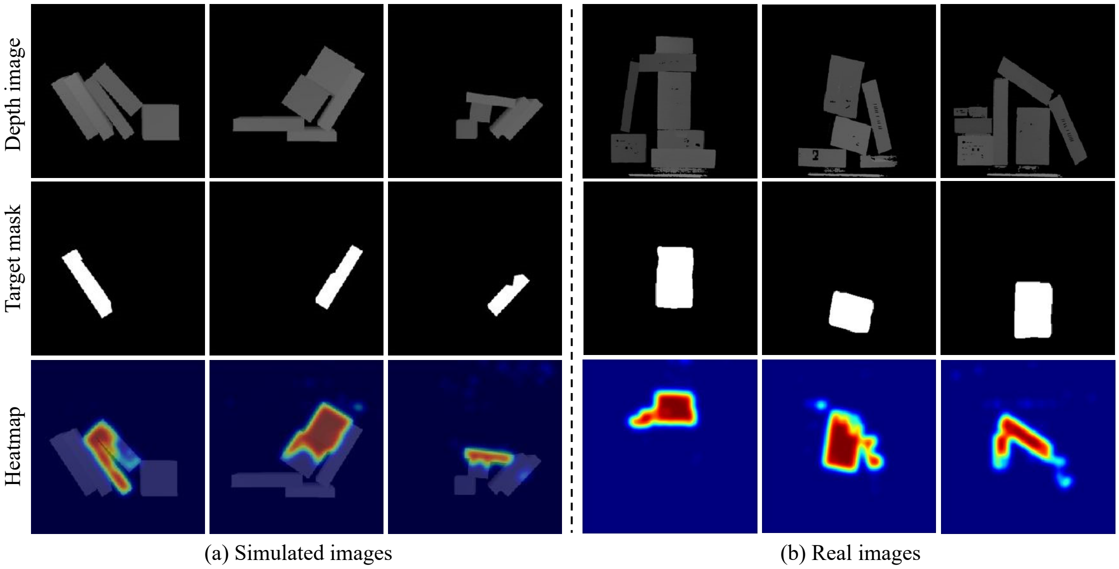


Figure 6.5: Outputs of the CP. From a set of both the depth images (top row) and target masks (middle row), the proposed network outputs the heatmaps (bottom row), which are the probabilistic color-scale $\in [0.0, 1.0]$. (a) The three images on the left are simulated and (b) the three on the right are real scenes.

6.2.2 Inference of Support Relation

In this section, we infer support relations based on the CP. Support relations have been defined in [24, 68, 69]. Summarily, given two objects X and Y, X supports Y is denoted as SUPP(X, Y). X is the supporting object, and Y is the supported object, i.e., if we remove X from the relation, Y falls. Herein, we focus on the fact that the CP detects objects that fall after removing a target object. Based on this definition, the CP is considered appropriate for detecting the relations between supporting and supported objects.

The flow of inference is as follows. First, we divide point clouds captured with a depth sensor into each object, which provides its target mask and object area R_O. Then, the CP outputs a probability map, a dense pixel-wise heatmap with values ranging from 0.0 to 1.0, as aforementioned. We calculate the area in the heatmap above a threshold value as the collapse area R_C. If an object is in the collapse area, we consider it a supported object for a target, i.e., a supporting object. To detect supported objects, we use the IoU between the collapse area and the area of each object: $(R_O \cap R_C)/(R_O \cup R_C)$. If the IoU exceeds a certain value, the two objects have a support relation. In a cluttered environment, removing an object may cause several objects that are not in direct contact with the object to fall. When using the CP, such indirect relations between objects should be excluded. Each object is detected as a bounding box (BB), and we evaluate adjacency scores with IoU based on object BBs that are larger than the original ones. If an adjacency score exceeds a threshold, it is considered a pseudo-direct contact.

Algorithm 1 Multi-step Object Extraction Planning

```
Input: all objects in clutter $O$ and selected target $o_t$
$Im$ ← Take depth image;
$G$ ← Create Collapse Prediction Graph with $Im$ and $O$;
while selected target $o_t$ is not extracted do
    $o$ ← Select the extractable object from $G$;
    $g$ ← Generate grasp pose for $o$;
    Grasp object $o$ in $g$;
    // Detect the collapse during the object extraction
    while true do
        $Im$ ← Take depth image;
        $cp$ ← Compute collapse score with $Im$ and $o$
        if $cp > cp_{max}$ then
            Release object $o$;
            Exit the loop;
        end if
        Pull object $o$ forward;
        if object $o$ has be extracted to a certain place then
            Exit the loop;
        end if
    end while
    $Im$ ← Take depth image;
    $G$ ← Renew Collapse Prediction Graph with $Im$;
end while=0
```

6.2.3 Multi-step Object Extraction

We construct a CPG to determine the next best target that can be safely extracted from the clutter. Given all the support relations, a tree is built with the target object as the root. As shown in Figure 6.6 (a, b), we connect the support relations and remove them except for those between adjacent targets, as abovementioned.

Our strategy exploits the CPG and safely removes other objects iteratively until the target is extracted. The procedure is shown in Algorithm 1. Safe extraction requires selecting a child node for a parent node to minimize the risk of collapse. We explore the CPG by reverse level order traversal with reference to [66,67]. If the objects are supported hierarchically, the leaf node, which is not supported by any other object, can be safely extracted in the CPG. Therefore, leaf nodes are extracted first. In a special scenario wherein the parent node has multiple child nodes, we retain a relation between the child and parent nodes at the lowest layer and ignore the other relations (see Figure 6.6 (c)). This is because if part of the supported objects is ignored when picking an object at a lower node, a collapse will occur.

Because this research does not consider dynamics during manipulations, an object may fall because of unexpected contact or friction. Therefore, we divide an action into several steps and ensure safety by predicting a collapse score cp before each step. The score is calculated using the collapse area R_C and manipulated object area R_O as follows:

$$cp = \frac{area(R_C \cap R_O)}{area(R_O)} \tag{6.2}$$

$area(\cdot)$ indicates the area of R. If cp exceeds a threshold, we can re-determine

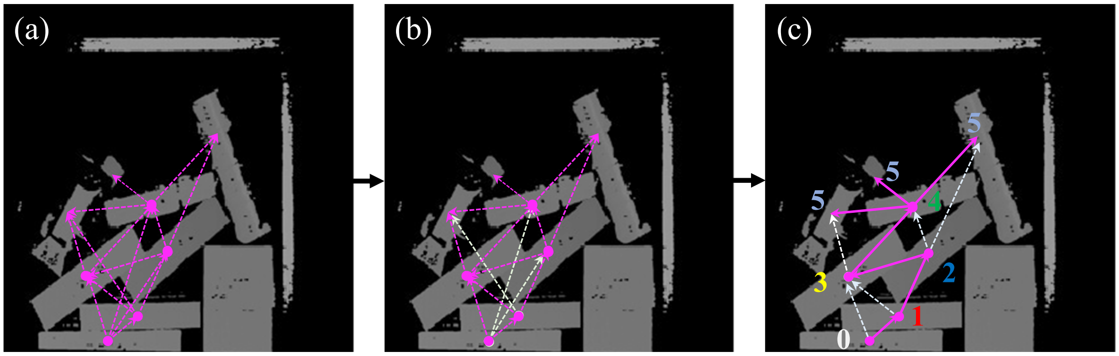


Figure 6.6: Creating a CPG. (a) We connect support relations, create a CPG G on a given object, and (b) remove relations except those between adjacent targets. (c) In these scenarios, the relations connecting to parent nodes at higher child nodes (white edges) are pruned to maintain crucial relations. The numbers indicate the hierarchy based on the target object.

the extraction order to select the other removable object (see lines 8-13 in Algorithm 1).

If support relations are detected on SUPP(X, Y) and SUPP(Y, X), i.e., supporting each other, we select only the support relationship with the higher collapse score and ignore the other. Then, we determine the extraction order. Notably, when removing these supporting objects with a single arm, bimanual arms should be used.

Bimanual manipulation is relevant for both efficient and safe extractions of clutters. In this study, we consider a bimanual manipulation for picking objects while supporting other objects. This technique can reduce action steps and pick objects efficiently. First, we ensure that a robot can retrieve a target object while ensuring sufficient support with the other arm.

Figure 6.7 illustrates the bimanual manipulation based on support relations. A robot can perform a bimanual action when only one supported object is related to the target (Figure 6.7 (d)). One arm grasps the object to prevent it from falling,

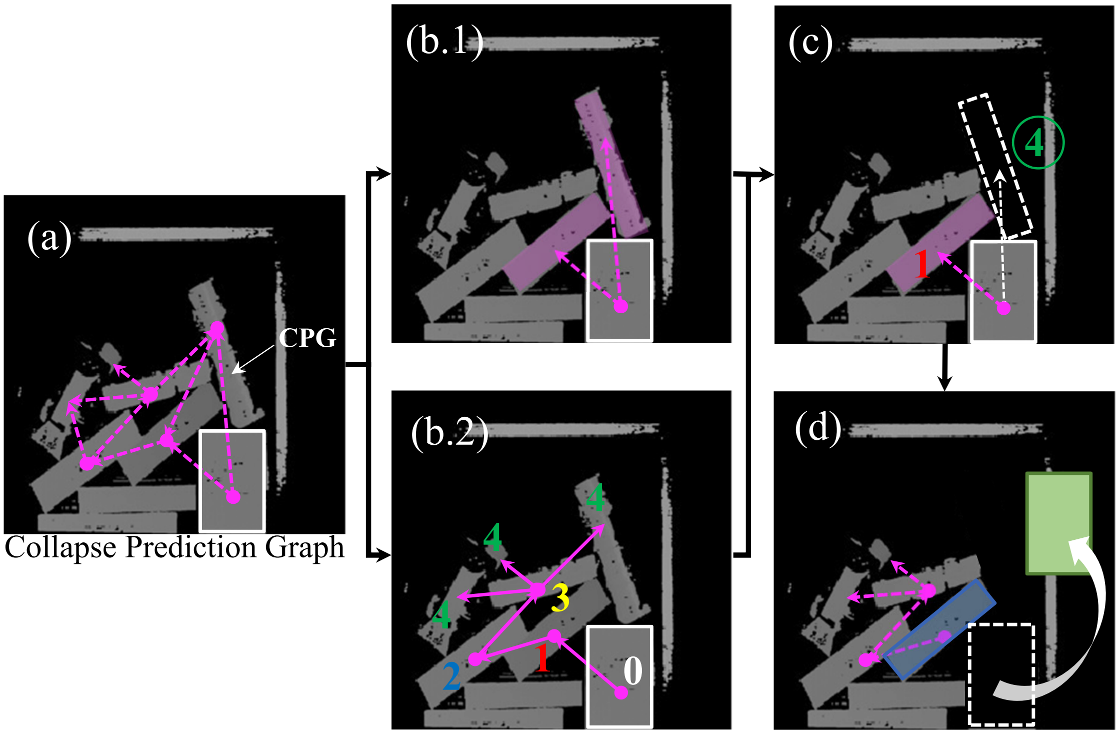


Figure 6.7: Bimanual manipulations based on support relations. Given a CPG (a), we can estimate support relations in contact with the target object marked in white (b.2) and generate the extraction order (b.2). (c) We iteratively remove objects using the extraction order until the target object supports only a single object. (d) The target object marked in green can be safely extracted by fixing the supported object marked in blue.

and the other extracts the target object. If two or more supported objects are present (Figure 6.7 (b.1)), before retrieving the target object, the robot extracts the supported objects to possibly satisfy the condition. The CPG for each supported object is constructed, as shown in (Figure 6.7 (b.2)). We select and extract the object with the lowest leaf node from all the CPGs (Figure 6.7 (c)) to satisfy the condition of the bimanual manipulation in a minimum step. For example, in Figure 6.7 (a), at least six objects should be removed based on the CPG. In contrast, when using bimanual manipulation, a robot can extract a target object after removing only one object.

6.3 Experiments

In this section, we evaluate the scene analysis from the estimation of support relations and test robotic experiments in a real-world environment.

6.3.1 Extraction of Support Relations

We evaluate the estimation of the support relations with reference to [68]. The depth images for several real scenes are captured with a YOODS YCAM3D-10L in front of a shelf. We construct the CPG $G_{HYP} = (O_{HYP}, E_{HYP})$ using the proposed methods. O denotes objects in the scene, and E denotes a support relation. $G_{GT} = (O_{GT}, E_{GT})$ is generated as the ground truth and manually annotated for the test. In this study, we focus only on the accuracy of the detections of the support relations but ignore the case where O_{HYP} does not correspond to O_{GT}. Herein, we evaluate our results in terms of precision and recall as follows:

$$Prec = \frac{|E_{HYP} \bigcap E_{GT}|}{|E_{HYP}|} \tag{6.3}$$

$$Rec = \frac{|E_{HYP} \bigcap E_{GT}|}{|E_{GT}|} \tag{6.4}$$

Table 6.2 shows precision and recall for 15 scenes. The results of the precision and recall are similar in accuracy to those of the related work [68].

6.3.2 Real-world Robot Experiments

In all the experiments, we used Yaskawa Electric MOTOMAN-SDA5F (a bimanual robot with 15 degrees of freedom (DOFs): 7 DOFs in each arm and 1 DOF in

Table 6.2: Precision and Recall of estimating support relations for all the tested scene images

Scene	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Mean**
Prec	0.833	1.000	1.000	0.571	0.750	0.750	0.857	1.000	0.750	0.800	1.000	1.000	1.000	0.700	1.000	**0.867**
Rec	1.000	0.667	1.000	1.000	0.750	0.750	1.000	1.000	0.750	1.000	1.000	1.000	1.000	1.000	1.000	**0.928**

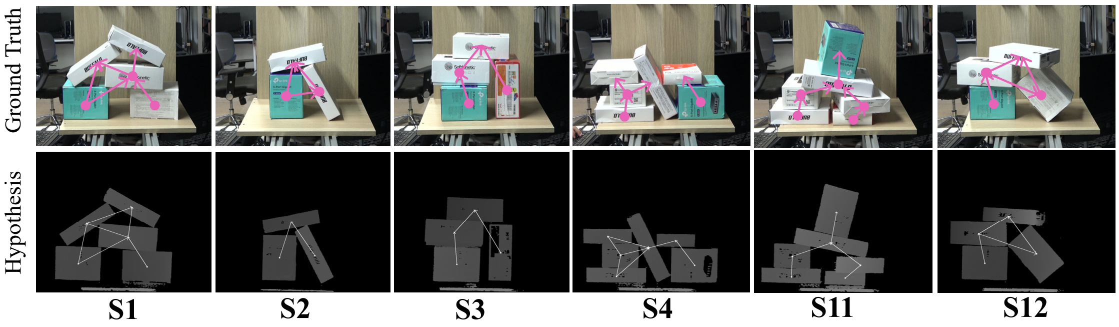


Figure 6.8: Selected evaluation scenes. We estimate the support relations (white edges), except for those between adjacent objects, from the depth images on the bottom row. We sample six cluttered scenes at the top row, where we manually annotated the support relations (pink edges) as the ground truth.

the waist) [88]. The method is programmed using Choreonoid [94] and graspPlugin [95]. The gripper is an adaptive gripper 2F-140 [96] from Robotiq and installed in the arms of MOTOMAN-SDA5F. The YOODS YCAM3D-10L is used in front of the shelf and can observe the inside [89]. The experimental environment is illustrated in Figure 6.9 (a). The system uses a Core i7-8550U CPU @ 1.80 GHz with 16 G RAM and Python 2.7. The OS is Ubuntu 16.04.

We used 3-10 rectangular objects (Figure 6.9 (b)) and randomly stacked them on the shelf. The robot detects objects by segmenting point clouds with the region growing [86] and creates a grasp pose from the detected object area.

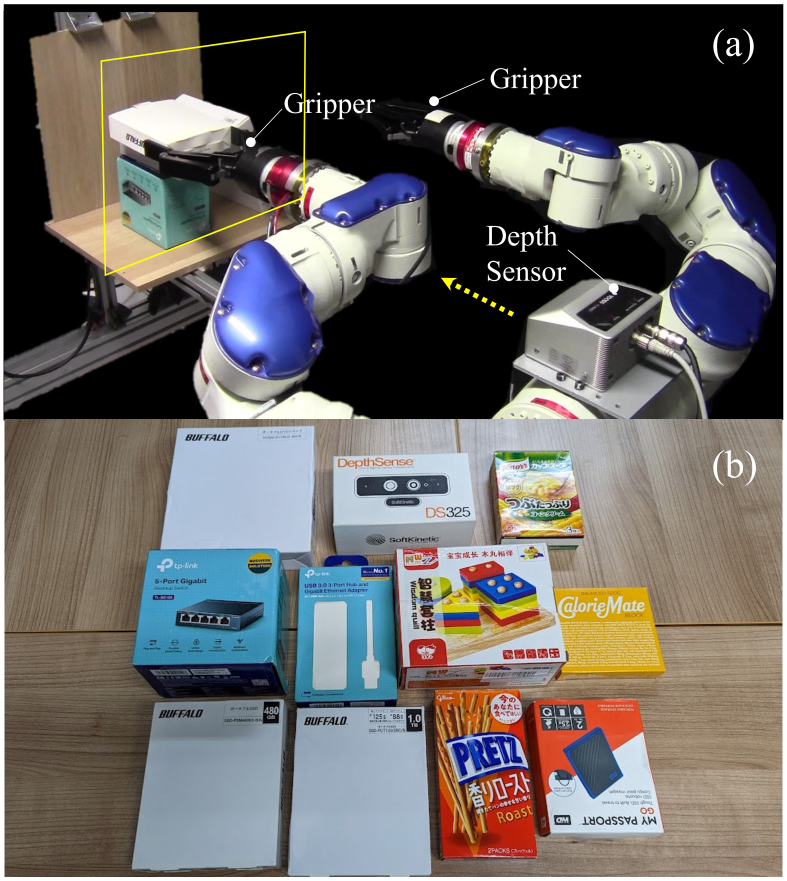


Figure 6.9: (a) Experiment setup, including a MOTOMAN-SDA5F robot, Robotiq 2F-140 grippers, and a YCAM3D-10L depth sensor. (b) Objects used for real-world extraction experiments.

Experiments on a Single Arm

These experiments test object picking from a viewpoint whereby support relations are correctly detected. Figure 6.10 shows snapshots of the experiments using a real robot; the upper images result from estimating the CPG and extraction order. We conducted 25 experiments using only one-handed manipulation in the proposed algorithm (Algorithm 1). This algorithm performs well at picking a selected target object with a success rate of 80.0% (20/25), and the mean of the steps is 2.3.

Table 6.3: Real-world extraction performance of different approaches and conditions

Method	Number of objects	Success rate
Single-step Extraction [90]	3-5	85.0% (51/60)
Single-step Extraction [90]	10	65.0% (13/20)
Multi-step Extraction	**3-10**	**91.2 % (52/57)**

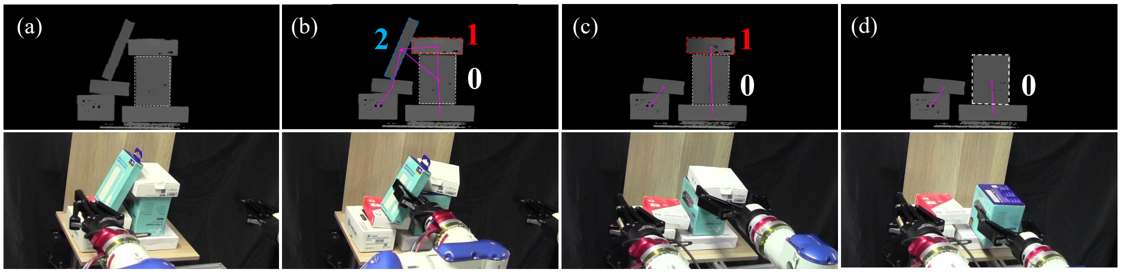


Figure 6.10: Real-world experiment using single arm. (Top) The proposed algorithm estimates support relations from a depth image. A CPG (lined in pink) is generated from these relations. (Bottom) The robot selects and extracts an object from the extraction order at (a)-(d).

We compare the proposed method to a single-step method [90]. The single-step method directly extracts the target object based only on initial collapse predictions. The robot attempts to extract an object from 3–5 or 10 objects randomly using the single-step method and an object from 3-10 objects using the proposed method. The result is shown in Table 6.3. The success rate at each step is used as the evaluation metric. The proposed method performed better than that in our previous work in terms of the success rate and achieved better performance regardless of the number of objects.

Experiments on Bimanual Arms

To validate bimanual manipulation, we conducted experiments with the bimanual arms of the MOTOMAN-SDA5F. Under the aforementioned condition in the

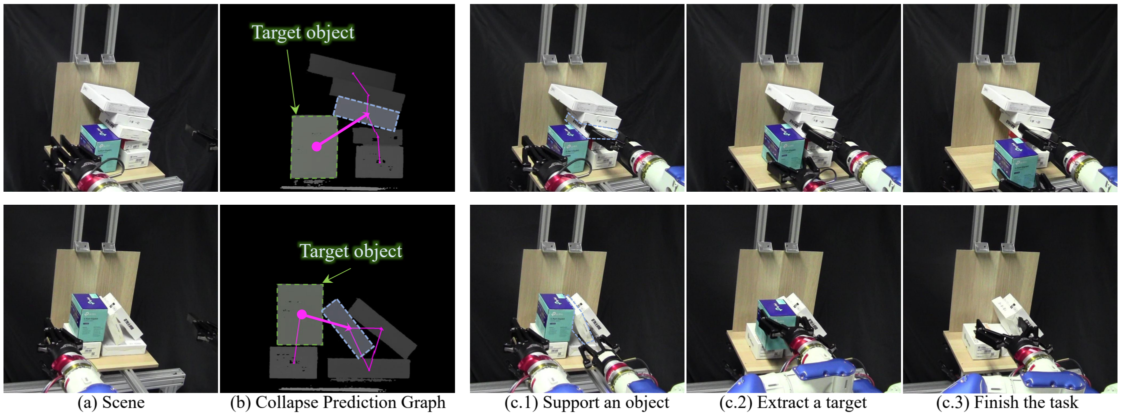


Figure 6.11: Real-world experiments using bimanual manipulations. (a, b) When a target object supports only a single object from the estimated CPG, (c.1) the robot holds the supported object and (c.2,c.3) extracts the target object. (Top) In stacked objects, the robot grasps an object on the target object. (Bottom) When arranging objects horizontally, the robot grasps any object that leans on the target object.

first experiments, we determine an effective option using a supporting and extracting action simultaneously. Figure 6.11 shows snapshots of the experiments to trigger the safe extraction order based on the proposed CPG. First, the robot captures the scene and detects support relations. If the target supports only a single supported object, the robot directly extracts the target object while supporting the supported object. If other support relations are identified on the target object, the robot removes an object following the extraction order using the single-arm method.

In Figure 6.11, the upper part is a stacked scene experiment and the bottom part is a shelved scene experiment. In the upper part, three objects are supported by the target object, and in the bottom, two objects are supported by the target object. In each scene, the robot immediately extracts the target object without removing all the supported objects.

6.4 Limitations and Possible Future Extension

Our study proposes a robotic manipulation system that can safely extract objects from a pile. The experimental results illustrate the importance of identifying support relations and adaptability to safe extraction in the real world. Notably, conventional methods, such as those proposed by [68, 69], developed the inference of support relations from the contact of approximate models and used heuristics with human understanding to predict uncertain information. These studies focused on scene analysis because their applications, which detect a complex scene and real-world manipulation, were problematic. In our study, we proposed a novel multi-step extraction plan and applied it to real-world robotic experiments. Our method achieved more than a 90-% success rate in retrieving selected objects by verifying the appropriate extraction order.

Our limitations are observed through physical experiments. First, learning accuracy has a significant implication for safe manipulation. Missing important support relations can cause damage to the object. One of the causes is that internal unobserved parameters such as friction, mass, density, and shape yield unexpected results. To improve detection accuracy, we adjust the trained model on known object shapes [101], a specific grasp condition [102]. Moreover, we need inference based on higher-dimensional observation information, such as point clouds, to accurately obtain support relations. Recently, a learning-based model for point clouds has been investigated to extract shape features. Danielczuk et al. [103] proposed a model architecture that examines the collision between point clouds. Chen et al. [104] designed an implicit estimation network to extract a 3D affordance heatmap for each potential task. By using these models, we can accurately detect contact between objects based on observations. In the future, we will extend the inference model to 3D to develop a more robust detection model of support relations.

Second, the arrangement between two arms is challenging. However, detection using the CPG shows that the operation can be performed efficiently with appropriate two-handed manipulation. However, both objects are assumed to be in contact and extremely mutually close. Therefore, the left and right arms can mutually interfere during robot manipulation, and the robot must plan the best motion sequences considering the pose and placement of the two grippers. Recent studies have focused on bimanual manipulation for various tasks. Chen et al. [105, 106] constructed an assembly sequence evaluated with the graspablity, safety, and assemblability of two manipulators. Avigal et al. [98] proposed the BiMaMa-Net architecture for bimanual manipulation, which predicts two corresponding gripper poses without any spatial constraint, to improve the bimanual folding for garments. In the future, we consider using a motion planning method for bimanual manipulation to improve usability.

Thus, identifying support relations from the CP is essential for adaptability to safely pick objects. Our CPG can guarantee a high level of safety in object picking by robots. In the future, we will incorporate lifting and repositioning objects based on action-based physical reasoning. To this end, we will integrate available information, such as the shapes, textures, and masses of objects, to improve the inference model.

6.5 Summary

In this chapter, I proposed an approach for safe object extraction based on the estimation of support relations between objects. The experimental results showed that a robot could estimate support relations based on collapse predictions with high accuracy equivalent to those in conventional works and perform safe extraction in real environments. I primarily considered the issue of safe extraction, which should be removed to secure each object from the graph structure by predicting the support relations between supported and supporting objects. This enabled the robot to choose the best next action from the limited observations. Further, a novel bimanual manipulation to extract the selected target object directly and efficiently was proposed.

In future studies, I will learn object movement from time-series data using updated simulation and integrate information on objects' external properties to predict the action's outcome. Moreover, I will consider different sensor viewpoints, which can be applied both on a shelf and in cluttered scenes, e.g., on a table.

Chapter 7

Conclusions

Manipulation in cluttered environments is difficult because products in warehouses and retail stores must not be damaged by a collision or falling. In this dissertation, we propose learning-based robotic manipulation for cluttered environments. We used a bimanual manipulator to address the problem. The proposed method enables a manipulator in common environments, consisting of a common robot gripper and a vision camera, and pick-and-place for the desired target object from the stacked objects on the table or high-piled shelf. The learning-based model enables the robot to safely manipulate the desired object by predicting the collapsing object beforehand to prevent the failure of picking and placing one target object, including an unexpected fall, clutter, and collision between objects. Furthermore, I have proposed robot action plans with single or bimanual arm operations to apply real-world tasks, from a toy problem to logistic automation.

This dissertation consists of seven chapters. Chapter 1 is the introduction. Chapter 2 analyzes the research topics related to our methods. Asides from Chapter 7, the following are discussions from the main contents of four chapters (Chapters 3, 4, 5, 6).

Chapter 3 examined the approach toward a cluttered environment through a toy problem. I focused on "Yamakuzushi," a Japanese board game that involves selecting and sliding a Shogi piece from a randomly stacked pile, and proposed a robotic action or observation planning method. The proposed method contributes to action planning considering the uncertainty of pose estimations and ConvNet that predicts the target object to enable the robot to slide a piece out of the board safely. Consequently, the robot can select the best subsequent behavior based on the previous action.

Many objects are randomly stacked on shelves in a logistics warehouse. A robot needs to safely extract one of the objects without other objects falling from the shelf. The proposed framework enables safe object extraction in a real-world environment. Furthermore, automated machinery is required to improve the efficiency of logistics automation while avoiding heavy objects from falling and injuring human workers. Additionally, Chater 4, introduces a novel bimanual manipulation based on deep learning. I presented a strategy for extracting a single object while supporting other objects, as well as a collapse prediction that determined the safe object extraction.

In Chapter 5, I presented an approach for detecting and analyzing a shelf display to safely manipulate and organize its content with a robot, wherein the bi-manual, dexterous manipulation capabilities of the robot are exploited to allow the task to be resolved without requiring to reorganize the entire shelf. The study made a significant contribution because shelf replenishment is a challenge that requires precise and careful handling of densely piled objects. Furthermore, I proposed a novel approach for automating the replenishment of disorganized shelves using a bimanual robot, building on the learning-based evaluator that predicted the risk of collapse of a shelf without selecting an extraction action according to the minimum risk of collapse. A safe replenishment process is extracted from a single depth image provided as an input network, where arrangement patterns are identified and the likelihood of collapsing objects is predicted. I demonstrated how the proposed algorithm, based on the improved collapse prediction method, could arrange the shelf and place an object horizontally and vertically, aligned with the shelf arrangement. The performance was demonstrated through experiments that involve randomized situations on a shelf with a real bimanual robot.

In Chapter 6, I proposed a multi-step motion planner to stably extract an item using the support relations of each object included in a clutter to address the challenge of automating operations in a logistic warehouse. A robot must extract an item from the clutter on a shelf without causing the clutter to collapse. I constructed a collapse prediction graph to obtain the appropriate order of object extraction by estimating the support relation, finally leading to the targeted item being extracted without causing collapse. Furthermore, I demonstrated that a robot's efficiency could be improved if it uses one arm to extract the target object while the other arm supports a neighboring object. This study made a significant contribution because the experimental results indicate that the robot could estimate support relations based on collapse predictions and perform safe extraction in real environments. Additionally, this study primarily focused on the issue of safe extraction, which allows the robot to choose the best next action based on the limited observations.

Acknowledgments

This thesis would not have been possible without the support of my supervisor, educators, peers, and family. I would like to particularly extend my sincerest gratitude to my advisor, Prof. Kensuke Harada, for his immense patience and support throughout this endeavor. I am extremely grateful that he mentored me as a student and continued to have faith in me for 6 years. His guidance and counsel have been invaluable in facilitating my research efforts. I would also like to express my gratitude to the members of my thesis committee, Profs. Koh Hosoda and Kosuke Sato, for their insightful comments and suggestions, which have been greatly beneficial to me.

I am deeply grateful for the invaluable assistance of Dr. Damien Petit, who generously devoted his time to guide me through the entire research process. I would also like to extend my gratitude to Prof. Weiwei Wan, Dr. Takao Nishi, and Dr. Kazuyuki Nagata for their valuable contributions through insightful discussions and collaborations in the early stages of the preparation of my first research paper.

Thank you to all the staff members in Harada Lab for their kind support during my Ph.D. study. I am grateful to Dr. Ixchel Ramirez-Alpizar, Prof. Keisuke Koyama, Prof. Takuya Kiyokawa, and Prof. Zhengtao Hu for their invaluable assistance in refining my presentations and for their generous support throughout my laboratory experience. I also thank my lab team members, Dr. Cristian Beltran, Ms. Xinyi Zhang, Mr. Masato Tsuru, and especially Mr. Kazuki Higashi, for all of the kind feedback, moral support, and sharing of academic ideas. I thank Dr. Tomu Tominaga, Mr. Ryo Matsumura, and Mr. Yuya Sato for teaching me the basics of research in the lab. Thank you to Ms. Hitomi Yoshioka and Ms. Makiyo Ueno, secretaries at Harada Lab, for supporting me in my research activities. I also extend my gratitude to all my colleagues from the Harada Lab at Osaka University for their time, advice, and moral support.

Finally, I would be remiss in not mentioning my family, especially my parents, my younger brother, and my younger sisters. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank Hayang, my lovely cat, for all the entertainment and emotional support.

Bibliography

[1] M. Fujita, Y. Domae, A. Noda, G. A. Garcia Ricardez, T. Nagatani, A. Zeng, S. Song, A. Rodriguez, A. Causo, I. M. Chen, and T. Ogasawara. What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics. *Advanced Robotics*, 34(7-8):560–574, 2020.

[2] Artur Cordeiro, Luís F. Rocha, Carlos Costa, Pedro Costa, and Manuel F. Silva. Bin picking approaches based on deep learning techniques: A state-of-the-art survey. In *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 110–117, 2022.

[3] Elena Garcia, Maria Antonia Jimenez, Pablo Gonzalez De Santos, and Manuel Armada. The evolution of robotics research. *IEEE Robotics Automation Magazine*, 14(1):90–103, 2007.

[4] Jur van den Berg, Stephen Miller, Ken Goldberg, and Pieter Abbeel. Gravity-based robotic cloth folding. In David Hsu, Volkan Isler, Jean-Claude Latombe, and Ming C. Lin, editors, *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*, pages 409–424. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[5] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

[6] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous Systems*, 60(10):1340–1353, 2012.

[7] Matthew T. Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):1–28, 2018.

[8] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22(1), jul 2022.

[9] Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Arne Sieverling Roberto Martín-Martín, Vincent Wall, and Oliver Brock. Four aspects of building robotic systems: lessons from the amazon picking challenge 2015. *Autonomous Robots*, 42(7):1459–1475, 2018.

[10] Haifei Zhu, Yuan Yik Kok, Albert Causo, Keai Jiang Chee, Yuhua Zou, Sayyed Omar Kamal Al-Jufry, Conghui Liang, I-Ming Chen, Chien Chern Cheah, and Kin Huat Low. Strategy-based robotic item picking from shelves. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2263–2270, 2016.

[11] Max Schwarz, Anton Milan, Arul Selvam Periyasamy, and Sven Behnke. RGB-D object detection and semantic segmentation for autonomous manipulation in clutter. *The International Journal of Robotics Research*, 37(4-5):437–451, 2018.

[12] Yung-Shan Su, Shao-Huang Lu, Po-Sheng Ser, Wei-Ting Hsu, Wei-Cheng Lai, Biao Xie, Hong-Ming Huang, Teng-Yok Lee, Hung-Wen Chen, Lap-Fai Yu, and Hsueh-Cheng Wang. Pose-aware placement of objects with semantic labels - brandname-based affordance prediction and cooperative dual-arm active manipulation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4760–4767, 2019.

[13] Oni Ornan and Amir Degani. Toward autonomous disassembling of randomly piled objects with minimal perturbation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4983–4989, 2013.

[14] Megha Gupta, Jörg Müller, and Gaurav S. Sukhatme. Using Manipulation Primitives for Object Sorting in Cluttered Environments. *IEEE Transactions on Automation Science and Engineering*, 2015.

[15] Robert Eidenberger, Thilo Grundmann, and Raoul Zoellner. Probabilistic action planning for active scene modeling in continuous high-dimensional domains. In *2009 IEEE International Conference on Robotics and Automation*, pages 2412–2417, 2009.

[16] Robert Eidenberger and Josef Scharinger. Active perception and scene modeling by planning with probabilistic 6d object poses. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1036–1043, 2010.

[17] Shengyong Chen, Youfu Li, and Ngai Kwok. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research*, 30:1343–1377, 10 2011.

[18] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald

Romo, Nima Fazeli, Ferran Alet, Nikhil Chavan Dafle, Rachel Holladay, Isabella Morena, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3750–3757, 2018.

[19] Donna C. Dupuis, Simon Léonard, Matthew A. Baumann, Elizabeth A. Croft, and James J. Little. Two-fingered grasp planning for randomized bin-picking. In *the Robotics: Science and Systems 2008 Manipulation Workshop-Intelligence in Human Environments*, 2008.

[20] Dirk Buchholz, Marcus Futterlieb, Simon Winkelbach, and Friedrich M. Wahl. Efficient bin-picking and grasp planning based on depth data. In *2013 IEEE International Conference on Robotics and Automation*, pages 3245–3250, 2013.

[21] Yukiyasu Domae, Haruhisa Okuda, Yuichi Taguchi, Kazuhiko Sumi, and Takashi Hirai. Fast graspability evaluation on single depth maps for bin picking with general grippers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1997–2004, 2014.

[22] Yuya Sato, Kensuke Harada, Nobuchika Sakata, Weiwei Wan, and Ixchel G. Ramirez-Alpizar. Two-stage robotic picking for randomly stacked objects with recognition difficulty. *Transactions of the Japan Society of Mechanical Engineers (JSME)*, 84(861):1–14, 2018. (in Japanese).

[23] Jue Kun Li, David Hsu, and Wee Sun Lee. Act to see and see to act: Pomdp planning for objects search in clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5701–5707, 2016.

[24] Rasoul Mojtahedzadeh, Abdelbaki Bouguerra, Erik Schaffernicht, and Achim J. Lilienthal. Support relation analysis and decision making for safe robotic manipulation tasks. *Robotics and Autonomous Systems*, 71:99–117, 2015. Emerging Spatial Competences: From Machine Perception to Sensorimotor Intelligence.

[25] Jinhwi Lee, Younggil Cho, Changjoo Nam, Jonghyeon Park, and Changhwan Kim. Efficient obstacle rearrangement for object manipulation tasks in cluttered environments. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 183–189, 2019.

[26] Changjoo Nam, Jinhwi Lee, Sang Hun Cheong, Brian Y. Cho, and

ChangHwan Kim. Fast and resilient manipulation planning for target retrieval in clutter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3777–3783, 2020.

[27] Jiuguang Wang, Philip Rogers, Lonnie Parker, Douglas Brooks, and Mike Stilman. Robot jenga: Autonomous and strategic block extraction. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5248–5253, 2009.

[28] Shinya Kimura, Tsutomu Watanabe, and Yasumichi Aiyama. Force based manipulation of Jenga blocks. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 4287–4292, 2010.

[29] Tsuneo Yoshikawa, Hirohito Shinoda, Seiji Sugiyama, and Masanao Koeda. Jenga game by a manipulator with multiarticulated fingers. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 960–965, 2011.

[30] Neil Dantam and Mike Stilman. The motion grammar: Linguistic perception, planning, and control. In *Robotics: Science and Systems*, 2011.

[31] Tomohiro Motoda, Weiwei Wan, and Kensuke Harada. Probabilistic action/observation planning for playing yamakuzushi. In *2020 IEEE/SICE International Symposium on System Integration (SII)*, pages 150–155, 2020.

[32] Sharon Temtsin and Amir Degani. Decision-making algorithms for safe robotic disassembling of randomly piled objects. *Advanced Robotics*, 31(23-24):1281–1295, 2017.

[33] Huadong Wu, Zhanpeng Zhang, Hui Cheng, Kai Yang, Jiaming Liu, and Ziying Guo. Learning affordance space in physical world for vision-based robotic object manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4652–4658, 2020.

[34] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998.

[35] Sung-Kyun Kim and Maxim Likhachev. Planning for grasp selection of partially occluded objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3971–3978, 2016.

[36] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Grasping pomdps. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4685–4692, 2007.

[37] Kazuyuki Nagata and Takao Nishi. Modeling object arrangement patterns and picking arranged objects. *Advanced Robotics*, 35(16):981–994, 2021.

[38] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems*, 07 2012.

[39] Huang Huang, Michael Danielczuk, Chung Min Kim, Letian Fu, Zachary Tam, Jeffrey Ichnowski, Anelia Angelova, Brian Ichter, and Ken Goldberg. Mechanical search on shelves using a novel “bluction” tool. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6158–6164, 2022.

[40] Ovidiu Ghita and Paul F. Whelan. A bin picking system based on depth from defocus. *Machine Vision and Applications*, 13(4):234–244, 2003.

[41] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *Int. J. Rob. Res.*, 34(4–5):705–724, apr 2015.

[42] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[43] Ryo Matsumura, Kensuke Harada, Yukiyasu Domae, and Weiwei Wan. Learning based industrial bin-picking trained with approximate physics simulator. In Marcus Strand, Rüdiger Dillmann, Emanuele Menegatti, and Stefano Ghidoni, editors, *Intelligent Autonomous Systems 15*, pages 786–798, Cham, 2019. Springer International Publishing.

[44] Yuhong Deng, Xiaofeng Guo, Yixuan Wei, Kai Lu, Bin Fang, Di Guo, Huaping Liu, and Fuchun Sun. Deep reinforcement learning for robotic pushing and picking in cluttered environment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 619–626, 2019.

[45] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[46] Ryo Matsumura, Yukiyasu Domae, Weiwei Wan, and Kensuke Harada. Learning based robotic bin-picking for potentially tangled objects. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7990–7997, 2019.

[47] Max Schwarz, Christian Lenz, Germán Martín García, Seongyong Koo, Arul Selvam Periyasamy, Michael Schreiber, and Sven Behnke. Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3347–3354, 2018.

[48] Kensuke Harada, Weiwei Wan, Tokuo Tsuji, Kohei Kikuchi, Kazuyuki Nagata, and Hiromu Onda. Initial experiments on learning-based randomized bin-picking allowing finger contact with neighboring objects. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1196–1202, 2016.

[49] Kensuke Harada, Weiwei Wan, Tokuo Tsuji, Kohei Kikuchi, Kazuyuki Nagata, and Hiromu Onda. Experiments on learning-based industrial bin-picking with iterative visual recognition. *Industrial Robot: the international journal of robotics research and application*, 45(4):446–457, 2018.

[50] Marco Costanzo, Simon Stelter, Ciro Natale, Salvatore Pirozzi, Georg Bartels, Alexis Maldonado, and Michael Beetz. Manipulation planning and control for shelf replenishment. *IEEE Robotics and Automation Letters*, 5(2):1595–1601, 2020.

[51] Jan Winkler, Ferenc Balint-Benczedi, Thiemo Wiedemeyer, Michael Beetz, Narunas Vaskevicius, Christian A. Mueller, Tobias Fromm, and Andreas Birk. Knowledge-enabled robotic agents for shelf replenishment in cluttered retail environments: (extended abstract). In *2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’16, page 1421–1422, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[52] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3d dynamic scene representations for robot manipulation. In *4th Conference on Robot Learning (CoRL)*, 2020.

[53] Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-centric models. In *International Conference on Learning Representations*, 2019.

[54] Aly Magassouba, Komei Sugiura, Angelica Nakayama, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, and Hisashi Kawai. Predicting and attending to damaging collisions for placing everyday objects in photo-realistic simulations. *Advanced Robotics*, 35(12):787–799, 2021.

[55] Kristen Grauman and Bastian Leibe. *Visual object recognition, Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan Claypool Publishers, 1995.

[56] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5222–5231, 2019.

[57] Tadashi Asaoka, Kazuyuki Nagata, Takao Nishi, and Ikuo Mizuuchi. Detection of object arrangement patterns using images for robot picking. *The International Journal of Robotics Research*, 5(23):1–18, 09 2018.

[58] Vladimir Tchuiev, Yakov Miron, and Dotan Di-Castro. Duqim-net: Probabilistic object hierarchy representation for multi-view manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10470–10477, 07 2022.

[59] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80:300–316, 2008.

[60] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[61] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *The International Journal of Robotics Research*, 30:1328–1342, 10 2011.

[62] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 852–869, Cham, 2016. Springer International Publishing.

[63] Hanbo Zhang, Xuguang Lan, Xinwen Zhou, Zhiqiang Tian, Yang Zhang, and Nanning Zheng. Visual manipulation relationship network for autonomous robotics. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 118–125, 2018.

[64] Hanbo Zhang, Xuguang Lan, Site Bai, Lipeng Wan, Chenjie Yang, and Nanning Zheng. A multi-task convolutional neural network for autonomous robotic grasping in object stacking scenes. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6435–6442, 2019.

[65] Hanbo Zhang, Deyu Yang, Han Wang, Binglei Zhao, Xuguang Lan, Jishiyu Ding, and Nanning Zheng. Regrad: A large-scale relational grasp dataset for safe and object-specific robotic grasping in clutter. *IEEE Robotics and Automation Letters*, 7(2):2929–2936, 2022.

[66] Swagatika Panda, A.H. Abdul Hafez, and C.V. Jawahar. Learning support order for manipulation in clutter. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 809–815, 2013.

[67] Swagatika Panda, A. H. Abdul Hafez, and C. V. Jawahar. Single and multiple view support order prediction in clutter for manipulation. *Journal of Intelligent & Robotic Systems*, 83:179–203, 2016.

[68] Rainer Kartmann, Fabian Paus, Markus Grotz, and Tamim Asfour. Extraction of physically plausible support relations to predict and validate manipulation action effects. *IEEE Robotics and Automation Letters*, 3(4):3991–3998, 2018.

[69] Markus Grotz, David Sippel, and Tamim Asfour. Active vision for extraction of physically plausible support relations. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 439–445, 2019.

[70] Fabian Paus and Tamim Asfour. Probabilistic representation of objects and their support relations. In Bruno Siciliano, Cecilia Laschi, and Oussama Khatib, editors, *Experimental Robotics*, pages 510–519, Cham, 2021. Springer International Publishing.

[71] Orbbec 3d. https://orbbec3d.com/, 2022. Accessed on 5 December 2022.

[72] Kawada robotics corporation. https://nextage.kawadarobot.co.jp/, 2022. Accessed on 5 December 2022.

[73] Sebastian Thrun, olfram Burgard, and Dieter Fox. *Probabilistic robotics*. The MIT Press, 2005.

[74] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[75] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[76] Gildardo Sanchez-Ante and Jean-Claude Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. *International Journal of Robotic Research (IJRR)*, pages 403–417, 01 2001.

[77] Kazuyuki Nagata, Takashi Miyasaka, Dragomir N. Nenchev, Natsuki Yamanobe, Kenichi Maruyama, Satoshi Kawabata, and Yoshihiro Kawai. Picking up an indicated object in a complex environment. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2109–2116, 2010.

[78] Simon Christoph Stein, Markus Schoeler, Jeremie Papon, and Florentin Wörgötter. Object partitioning using local convexity. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311, 2014.

[79] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. Our-cvfh – oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In Axel Pinz, Thomas Pock, Horst Bischof, and Franz Leberl, editors, *Pattern Recognition*, pages 113–122, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[80] Point cloud library, 2022. Accessed on 21 December 2022.

[81] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional

networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[82] NVIDIA DEVELOPER. https://developer.nvidia.com/physx-sdk, 2022. Accessed on 26 October 2022.

[83] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[84] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[85] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[86] T Rabbani, F.A. Heuvel, and George Vosselman. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, 01 2006.

[87] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[88] Industrial Robots & Robot Automation Tech — Yaskawa Motoman. https://www.motoman.com/en-us/products/robots/industrial/assembly-handling/sda-series/sda5f, 2022. Accessed on 7 November 2022.

[89] YOODS Co. Ltd. . https://www.yoods.co.jp/products/ycam.html, 2022. Accessed on 7 November 2022.

[90] Tomohiro Motoda, Damien Petit, Weiwei Wan, and Kensuke Harada. Bimanual shelf picking planner based on collapse prediction. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 510–515, 2021.

[91] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[92] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[93] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[94] Choreonoid Official Site. `https://choreonoid.org/en/`, 2022. Accessed on 7 November 2022.

[95] graspPlugin for Choreonoid. `http://www.hlab.sys.es.osaka-u.ac.jp/grasp/en/`, 2022. Accessed on 7 November 2022.

[96] Robotiq: Start Production Faster. `https://robotiq.com`, 2022. Accessed on 7 November 2022.

[97] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[98] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2022.

[99] Tomohiro Motoda, Damien Petit, Takao Nishi, Kazuyuki Nagata, Weiwei Wan, and Kensuke Harada. Shelf replenishment based on object arrangement detection and collapse prediction for bimanual manipulation. *Robotics*, 11(5), 2022.

[100] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

[101] Hiroki Tachikake and Wataru Watanabe. A learning-based robotic bin-

picking with flexibly customizable grasping conditions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9040–9047, 2020.

[102] Shumpei Wakabayashi, Shingo Kitagawa, Kento Kawaharazuka, Takayuki Murooka, Kei Okada, and Masayuki Inaba. Grasp pose selection under region constraints for dirty dish grasps based on inference of grasp success probability through self-supervised learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8312–8318, 2022.

[103] Michael Danielczuk, Arsalan Mousavian, Clemens Eppner, and Dieter Fox. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6010–6017, 2021.

[104] Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Learning 6-dof task-oriented grasp detection via implicit estimation and visual affordance. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–7, 2022.

[105] Hao Chen, Weiwei Wan, and Kensuke Harada. Planning to build soma blocks using a dual-arm robot. In *2021 IEEE International Conference on Development and Learning (ICDL)*, pages 1–7, 2021.

[106] Hao Chen, Weiwei Wan, Keisuke Koyama, and Kensuke Harada. Planning to build block structures with unstable intermediate states using two manipulators. *IEEE Transactions on Automation Science and Engineering*, 19(4):3777–3793, 2022.

List of publications

Journal Articles

- 元田智大, 万偉偉, 原田研介, ヒトの行動を規範とした山崩しゲームに対するロボットの観測・操り行動計画, 日本機械学会論文集, 86(892), 2020.
- Tomohiro Motoda, Damien Petit, Takao Nishi, Kazuyuki Nagata, Weiwei Wan, Kensuke Harada. Shelf Replenishment Based on Object Arrangement Detection and Collapse Prediction for Bimanual Manipulation. *Robotics*, 11, 104, 2022.
- Tomohiro Motoda, Damien Petit, Takao Nishi, Kazuyuki Nagata, Weiwei Wan, Kensuke Harada. Multi-step Object Extraction Planning from Clutter based on Support Relations. *IEEE Access*, 2022. (Submitted)

International Conferences (peer-reviewed)

- Tomohiro Motoda, Weiwei Wan and Kensuke Harada. Probabilistic Action/Observation Planning for Playing Yamakuzushi. In *Proceeding of IEEE/SICE International Symposium on System Integrations (SII 2020)*, p. 150–155, 2020.
- Masato Tsuru, Pierre Gergondet, Tomohiro Motoda, Adrien Escande, Eiichi Yoshida, Ixchel G. Ramirez-Alpizar and Kensuke Harada. POMDP Action Planning for 6 DoF Object Recognition on Humanoid. In *Proceeding of 2020 International Conference on Artificial Life and Robotics (ICAROB)*, 2020.

- Tomohiro Motoda, Damien Petit, Weiwei Wan and Kensuke Harada. Bimanual Shelf Picking Planner Based on Collapse Prediction. In *Proceeding of IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 510–515, 2021.
- Shusei Nagato, Tomohiro Motoda, Keisuke Koyama, Weiwei Wan and Kensuke Harada. Motion Planning to Retrieve an Object from Random Pile. In *Proceeding of 2022 International Conference on Artificial Life and Robotics (ICAROB)*, 2022.
- Kazuki Higashi, Tomohiro Motoda, Akiyoshi Hara and Kensuke Harada. Acquisition of Synergy for Low-dimensional Control of Multi-fingered Hands by Reinforcement Learning. *In Proceeding of 2023 International Conference on Artificial Life and Robotics (ICAROB)*, 2023.

Local Conferences (non-peer-reviewed)

- 元田智大, 原田研介. 確率モデルによるバラ積み物体の認識と操りのための行動計画. 第18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, 仙台, 3A6-02, 2017.
- 元田智大, 万偉偉, 原田研介. ばら積み物体の認識と操りのための確率的行動計画の検証. 第36回日本ロボット学会学術講演会, 春日井, 1G2-05, 2018.
- 元田智大，万偉偉，原田研介. ばら積み物体の認識と操りのための確率的行動計画の検証. 第63回システム制御情報学会研究発表講演会, 大阪, GSf04-3, 2019.
- 都留将人, Pierre Gergondet, 元田智大, Ixchel Ramirez, Adrien Escande, 万偉偉, 吉田英一, 原田研介. ヒューマノイドロボットによる物体探索のた

めの認識行動計画. ロボティクス・メカトロニクス講演会, 広島, K07-1, 2019.

- 元田智大, 万偉偉, 原田研介. ロボットによる将棋崩しの実現に向けた操り・観測の確率的計画手法の検証. 第20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, 高松, 3E2-09, 2019.
- 元田智大, 万偉偉, Damien Petit, 原田研介. 双腕ロボットによるバラ積みされた対象物の引き出し動作計画. 第21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, 3D3-07, 2020.
- 東和樹, 元田智大, 西村優佑, 原彰良, 濱本孝仁, 原田研介強化学習による多指ハンドの低次元制御を実現するシナジーの獲得. ロボティクス・メカトロニクス講演会, 大阪, 2021.
- 長門秀征, 元田智大, 小山佳祐, 万偉偉, 原田研介. 複雑な環境下における目標物体の取り出し動作計. ロボティクス・メカトロニクス講演会, 大阪, 2021.
- 元田智大, Petit Damien, 西卓郎, 永田和之, Wan Weiwei, 原田研介. 陳列の乱れの識別に基づく双腕による充填作業計画. 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, 1D1-05, 2021.
- 長門秀征, 元田智大, 西卓郎, 清川拓哉, 万偉偉, 原田研介. 物体の状態遷移予測を用いた山積みからの双腕引き出し計画. 第23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, 3P2-H05, 2022.

Articles

- Masato Tsuru, Pierre Gergondet, Tomohiro Motoda, Adrien Escande, Ixchel G. Ramirez-Alpizar, Weiwei Wan, Eiichi Yoshida, Kensuke Harada.

How a Humanoid Robot Searches for an Object in Our Daily Environment, Open Access Government, April Issue, pp. 200–202, 2021.

- 東和樹, 元田智大, 都留将人. 学生編集委員会取材企画：“顔パスで支払いまで”未来の小さなロボット店員を開発PLEN Robotics株式会社代表取締役社長赤澤夏郎氏, 日本ロボット学会誌, 40(6), 2022.

Awards

- 優秀講演賞: 元田智大, 原田研介. 確率モデルによるバラ積み物体の認識と操りのための行動計画. 第18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, 2017.
- SCI学生発表賞: 元田智大. ばら積み物体の認識と操りのための確率的行動計画の検証. 第63回システム制御情報学会研究発表講演会, 2019.
- Young Award (IROS, CASE2021): Tomohiro Motoda. *IEEE Robotics and Automation Society Japan Joint Chapter, 2021.*
- 優秀講演賞*:* 元田智大*, Petit Damien,* 西卓郎*,* 永田和之*, Wan Weiwei,* 原田研介. 陳列の乱れの識別に基づく双腕による充填作業計画. 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*, 2021.*