Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection

Örebro Studies in Technology 71

Rasoul Mojtahedzadeh

Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection

Title: Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection

Publisher: Örebro University, 2016
www.publications.oru.se

Printer: Örebro University, Repro 08/2016

ISSN 1650-8580
ISBN 978-91-7529-152-9

Abstract

Rasoul Mojtahedzadeh (2016): Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection. Örebro Studies in Technology 71.

This thesis is concerned with the task of autonomous selection of objects to remove (unload) them from a pile in robotic manipulation systems. Applications such as the automation of logistics processes and service robots require an ability to autonomously manipulate objects in the environment. A collapse of a pile of objects due to an inappropriate choice of the object to be removed from the pile cannot be afforded for an autonomous robotic manipulation system. This dissertation presents an in-depth analysis of the problem and proposes methods and algorithms to empower robotic manipulation systems to select a safe object from a pile elaborately and autonomously.

The contributions presented in this thesis are three-fold. First, a set of algorithms is proposed for extracting a minimal set of high level symbolic relations, namely, gravitational act and support relations, of physical interactions between objects composing a pile. The symbolic relations, extracted by a geometrical reasoning method and a static equilibrium analysis can be readily used by AI paradigms to analyze the stability of a pile and reason about the safest set of objects to be removed. Considering the problem of undetected objects and the uncertainty in the estimated poses as they exist in realistic perception systems, a probabilistic approach is proposed to extract the support relations and to make a probabilistic decision about the set of safest objects using notions from machine learning and decision theory. Second, an efficient search based algorithm is proposed in an internal representation to automatically resolve the inter-penetrations between the shapes of objects due to errors in the poses estimated by an existing object detection module. Refining the poses by resolving the inter-penetrations results in a geometrically consistent model of the environment, and was found to reduce the overall pose error of the objects. This dissertation presents the concept of minimum translation search for object pose refinement and discusses a discrete search paradigm based on the concept of depth of penetration between two polyhedrons. Third, an application centric evaluation of ranging sensors for selecting a set of appropriate sensors for the task of object detection in the design process of a real-world robotics manipulation system is presented. The performance of the proposed algorithms are tested on data sets generated in simulation and from real-world scenarios.

Keywords: Object Selection; Object Pose Refinement; Gravitational Support Relation; Inter-penetration Resolving; 3D Ranging Sensor Evaluation.

Rasoul Mojtahedzadeh, School of Science and Technology
Örebro University, SE-701 82 Örebro, Sweden

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Achim J. Lilienthal for the continuous support of my Ph.D study and research, whose expertise, understanding and patience added considerably to my graduate experience. His guidance always helped me not only in all the time of research and writing this thesis, but also during the most difficult times of my Ph.D life. Thanks to him I had the opportunity to learn from and work in an excellent atmosphere for doing research and developing my ideas.

Many thanks to my co-supervisors, Dr. Todor Stoyanov, Dr. Abdelbaki Bouguerra and Dr. Erik Schaffernicht for many fruitful discussions and valuable feedback. I will certainly miss all the technical discussion we have had during these years. I appreciate all the moments you patiently listened to my next research idea following with long but constructive discussions led me to write and publish articles. Next, I would like to extend my thanks to all my co-workers and friends at the university who made this place a friendly research environment.

Finally, I want to express my appreciation for my family, my parents and siblings, and my nephew and nice, Arian and Helia for their love and support where visiting them far back home always gave me extra energy to continue. Without their emotional support and encouragement over these years I couldn't make it. Last but not least, I would like to thank Mehdi Bidokhti, one of my best friends, for being present and encouraging me during writing this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1
Introduction

In everyday life, working at home or in an industrial environment, people move things from one place to another. Depending on the complexity of task, an appropriate selection of objects for manipulation is an essential decision that people make beforehand and often without explicit thinking. For example, the preliminary stages of arranging and organizing bookshelves, cupboards and cabinets require that the stacked objects are unloaded safely. In other words, people normally choose and remove an object from a shelf such that other objects stay motionless, and this is to prevent the other objects from falling down or toppling over. In industry, logistic processes often deal with piles of objects which may come in random configurations (see Figure 1.1 for a few real-world examples). Looking at a pile of objects with an arbitrary configuration, people are usually able to employ their experience and knowledge to select a set of safe-to-remove candidates from the pile such that removing the selected objects preserves the stability of the pile. In this thesis, safety is reflected through selecting an object from a pile such that removing it leads to as little motion as possible of the other objects in the pile. The ability to select a safe object for removing from a pile minimizes the risk of a collapse and thus prevents damage to the objects and the environment.

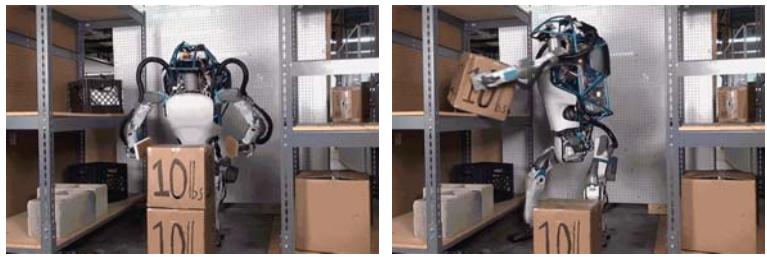
Introducing robots increased the demand for replacing humans with machines for performing drudgeries and complex jobs. In order to employ robots in jobs involving moving objects, apart from appropriate design of a mechanical body and corresponding controllers, an autonomous robot must also be able to perceive the surrounding objects, analyze the structure of the environment and make proper decisions to reduce accidental damage due to manipulation of objects. An example of a recent demonstration of an advanced humanoid robot with walking and manual skills is the Atlas robot made by Boston Dynamics (see Figure 1.2). A key requirement for such advanced robots for safe use in real-world manipulation of objects is the ability to select safe objects for manipulation. For example, a robot such as Atlas needs to be able to autonomously reason and safely unload the piles of carton boxes shown in Figure 1.1. The

(a) (b)

Figure 1.1: A few snapshots of configurations of objects inside shipping containers at unloading sites.

ability of a robot to answer the questions such as: "how many other objects would fall down if I remove this object?", "which object does not support any other object in the pile?" and "which object is the safest candidate to remove from a pile?" enables the robot to automatically reason about the safest sequence of unloading objects from a pile. Robots without the ability to analyze the complexity of the task and make proper decisions cannot be utilized for accomplishing tasks autonomously and need to be supervised by humans continuously.

This thesis work presents a set of contributions towards the development of autonomous robotic manipulation systems for real-world applications of unloading objects from piles. The main focus is on the ability of a robot to use efficiently the available description of the objects extracted from perception to select a safe object to be unloaded from a pile. This ability, which will be discussed further in Chapter 5, is essential for the autonomy as well as the performance of the robotic manipulation system. When the description of poses of objects is inaccurate, the geometrical shapes may inter-penetrate into each other representing a model of the environment which is inconsistent with a rigid body assumption. Inter-penetrations between the shapes of objects have to be resolved for making proper decisions about the safe-to-remove objects. This thesis presents algorithms to resolve the inter-penetrations and refine the poses of object in order to obtain a model of the environment in which there is no inter-penetration between pairs of objects. This will be discussed in detail in Chapter 4. One reason for inaccuracy in the poses of objects is the use of inappropriate visual sensors for the underlying application. In order to study the effects of appropriate selection of visual sensors on the task of pose estimation of objects, this thesis presents an application-centric evaluation of range sensors, which will be further discussed in Chapter 3.



(a) (b)

Figure 1.2: A demonstration of the humanoid robot Atlas made by Boston Dynamics. (a) The robot grasps a carton box and (b) lifts and places the box on a shelf. The research question of this thesis is how to empower a robot such as Atlas with the ability to autonomously reason and safely unload the carton boxes from stacks or complex piles such as shown in Figure 1.1?

1.1 Motivation

Globalization has increased the volume of transported goods, and as a result there is a huge demand for the fast and reliable logistics processes. Every day, thousands of cargo containers are shipped between continents, where a variety of goods composing a cluttered environment, and often stacked chaotically need to be unloaded in short time. Although manually unloading goods by human workers is a tedious, strenuous job imposing serious health risks, it is common practice to use human workers for removing goods from cargo containers. Lifting and handling heavy objects is a prohibitively exhausting job which may result in permanent injuries to the workers and costly damage to goods through unexpectedly falling objects. Apart from the need of fast unloading of containers, the lack of manpower to work under unhealthy working conditions combined with strict labor union regulations make human labor a high cost factor and increase the demand for autonomous unloading machines. The European Union project titled “Cognitive Robot for Automation of Logistics Processes”, in which the work of this dissertation was carried out (Chapter 2 describes the project in more detail), aims at developing an autonomous robotic solution for the task of unloading goods from cargo containers.

Among a number of engineering and scientific difficulties, a major problem in automating the task of unloading goods is the autonomy in making a safe decision about the sequence of objects to unload under noisy data and the uncertainty in the execution of the unloading actions. Even with advanced capabilities of grasping and moving an object from a pile to another place, robotic manipulation systems cannot be afforded to unload goods if the removed object cause a collapse of the pile. In order to deploy robots for use in real-world

environments an autonomous understanding of the structure of the underlying task is crucial for making proper decisions. A robotic unloading system, in particular, needs to make a safe decision about the sequence of objects to unload from a pile.

The idea that goods are usually stacked neatly into cargo container at loading stations may suggest the possibility of using a preprogrammed unloading plan. However, the long-distance freight transport requires shipping cargo containers between ship, rail and truck resulting the configuration of goods, to some degree, random, as some real-world examples can be seen in Figure 1.1. Even if the pile of goods inside cargo containers preserves the neat initial configuration when reaches to the unloading sites, making decisions based on a preprogrammed plan for unloading goods may fail due to uncertainty both in perception of objects and in executing unloading actions. Uncertainty in the perception may cause errors in grasp planning, obstacle avoidance and path planning of the robotic manipulator, and failures to execute an unloading action may change the arrangement of the objects. Consequently, when the arrangement of the objects change the preprogrammed unloading plan is no longer valid. In addition to the possibility of a change in the arrangement of objects that are neatly stacked, the piles of objects for which the arrangement is not known in advance and may have been chaotically stacked need appropriate algorithms to analyze the stability of the pile and make safe decision about the next object to unload.

Domestic and service robots [1, 2] can also benefit from the algorithms predicting the effects of manipulating a selected object on the stability of the environment to make safer decisions. For example, when asking a service robot such as a Willow Garage Personal Robot [3] to bring an elderly person a food box located inside a refrigerator of possibly filled with other objects it will not be accepted to cause any object to topple over or fall down. If the robot is able to predict the consequences of manipulating objects on the stability of the surrounding objects, then it can plan for a safer sequence of actions to perform a desired task.

1.2 Problem Statement

The specific research problem addressed in this dissertation can be generally stated as below,

Problem. *Given the geometrical shapes and an estimation of the poses of a set of objects that are part of a pile, determine a sequence of unloading actions such that removing an object maintains the stability of the pile.*

Considering the shapes of commonly-used objects in logistics processes, it is assumed that the shape of an object can be well approximated with a convex polyhedron. Nevertheless, most of the algorithms presented in this thesis can be extended to deal with concave shaped objects by decomposing a concave shape

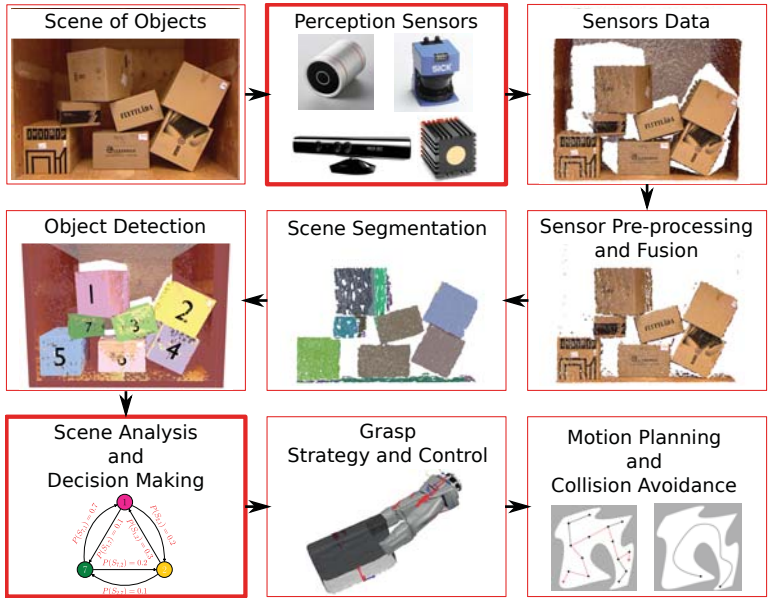


Figure 1.3: The pipeline of the robotic unloading system starting from perception and moving on to making decision about the safest object to unload.

into a set of connected convex polyhedrons. It is explicitly assumed that the pile is in static equilibrium, that is, the objects composing the pile are motionless. An estimation of the poses is assumed to be obtained by an existing object detection and pose estimation module, where, to some extent, there exists uncertainty in the estimated poses.

It is important to further clarify one condition of the problem statement, that is, there is no guarantee that the description of all the objects composing a pile are available for the analysis of the stability of the pile and determining a safe sequence of unloading actions. The undetected objects could be the result of occlusion or a failure in the detection process of the existing object detection module.

1.3 Challenges

A robotic manipulation system to operate autonomously and unload objects safely has to deal with a number of scientific and engineering challenges. Starting from low-level perception and moving on to complex analysis of the possibly noisy and incomplete data to extract a high-level meaningful interpretation of the environment represents a multitude of challenges to address. A variety

of the type of objects, which may come in different size, shape and material imposes not only difficulties in grasp planning and execution, but also it makes reliable identification of objects and making safe decisions challenging.

Figure 1.3 shows a conceptual pipeline for an autonomous robotic unloading system starting from perception and moving on to the execution of the unloading action. The process starts by sensing the environment (i.e., the scene of objects) using perception sensors. The sensor fusion and pre-processing are then performed to reduce the noise and prepare the data for high level processing. Scene segmentation further prepares the input data for object detection and pose estimation algorithms. In the next step, the configuration of the set of detected objects is analyzed for making decision about the safest object to remove from the scene. Motion and grasping plans are then computed for the selected object to move it to the desired place without colliding with the other objects in the environment.

The description of objects such as shapes and poses is normally not available and has to be extracted from the sensory data, which is inherently uncertain. For a reliable analysis of the stability of a pile in order to identify safe-to-remove objects, a fundamental component, similar to the visual system of human being, is the quality of the detection of objects. Object detection and pose estimation algorithms, however, represent errors in the estimated description of objects due to occlusion, noisy data and internal failures in algorithms. Some objects may be inherently invisible due to occlusion or not being in the field of view of perception sensors. Conversely, false positive objects, which are non-existing objects that are detected as some type of objects by the object detection algorithm are another challenging issue in the analysis of the stability of a pile. The uncertainty about the estimated poses of the location of objects further complicates reasoning about the safe object to unload. A misclassification of the type of objects, as an internal failure in object detection algorithms, represents an incorrect hypotheses about the corresponding geometrical shapes of the objects. The aforementioned difficulties highlight the importance of an evaluation of the perception sensors in order to minimize the negative effects of an inappropriate sensor selection on the task of object detection.

The problem of undetected objects of a pile could play a dominant role in the stability analysis of the pile. The objects that are located behind other objects are inherently occluded, thus they cannot be perceived and detected. Even objects that are not occluded may not be detected by an existing object detection algorithm. The problem of having access to only a subset of objects of a pile represents the lack of information in making decision about the safest object. When facing the lack of information, human beings usually use a heuristic solution, e.g., not being able to see the objects behind the front layer of a pile, people choose an object which is most probable to be safe according to their own justification. An algorithmic decision making about the set of safe-to-remove objects from a pile, in turn, needs to be able to deal with the lack of information.

The errors in the estimated poses even for a complete and correct detection of objects may result in a set of inter-penetrations between the shapes of adjacent objects, which represent a geometrically inconsistent model of the environment. The geometrical consistency of the estimated poses is required due to a rigid body assumption of the real-world, where solid objects are assumed not to deform or penetrate into each other. A model of the environment that is inconsistent with a rigid body assumption may cause failures in geometrical reasoning on the stability of a pile. A partial or complete inter-penetration of shapes thus needs to be resolved as a preliminary stage of a geometrical reasoning.

1.4 Outline

The rest of this thesis is organized as follows.

Chapter 2 presents an overview of the previous related works on the problems addressed in this thesis, and introduces the EU-funded project titled "Cognitive Robot for Automation of Logistic Processes" (RobLog) in which the presented work was carried out.

Chapter 3 is focused on the problem of 3D range sensor evaluation and selection in the design process of a complex robotic system with a specific attention to the challenging scenarios in the RobLog project. An application centric 3D range sensor evaluation is presented and discussed.

Chapter 4 proposes a framework to refine the noisy estimated poses of a set of objects in order to obtain a geometrically consistent model of the environment. In this chapter, the depth of penetration between two polytopes is utilized to define a reduced search space for resolving the inter-penetrations between objects due to errors in the initially estimated poses.

Chapter 5 discusses the problem of determining a set of safe-to-remove objects from a pile under complete and incomplete information. Depending on the availability of a description of the objects, two major approaches are proposed. This chapter introduces algorithms to represent and extract gravitational *act* and *support* relations based on notions from geometry and static equilibrium in classical mechanics. Machine learning techniques and probabilistic decision making approaches are employed to address the problem of undetected objects of a pile and the uncertainty in the input data.

Chapter 6 concludes this thesis with final remarks and suggested directions for future research work.

1.5 Contributions

The contributions presented in this thesis work can be summarized as follows:

- **An application-centric method** for comparative evaluation and selection of a set of appropriate 3D range sensors in the context of automatic unloading goods from cargo containers.
- **An object pose refinement framework** based on the concept of depth of penetration between two overlapping polytopes and search algorithms to obtain geometrical consistent models of the environment.
- **Development of** a methodology to identify and select a set of safe-to-remove objects from a pile for integration into fully autonomous robotic manipulation systems. The method is not tied to any specific robotic manipulator and neither to a particular object detection algorithm. Thus, the proposed method can be readily adopted for different designs of robotic manipulation setups.
- **A method** for extracting gravitational support relations by automatically analyzing the stability of a pile of objects with an arbitrary configuration and possibly under uncertainty and lack of information about the complete set of the objects. Machine learning techniques employed to estimate the probability of the support relations and notions from decision theory are used to select the set of safe-to-remove objects.
- **An open-source C++ library** implementing aforementioned object pose refinement framework under Robot Operating System (ROS).
- **Comprehensive, quantitative evaluation** of the proposed methods on data sets generated in simulation and from real-world scenarios.

1.6 Publications

The contributions of this thesis work have been presented in different peer reviewed journal articles or conference papers. The major results from this dissertation were published in the following articles:

- *R. Mojtahedzadeh, T. Stoyanov, A. Lilienthal.* **Application based 3D sensor evaluation: A case study in 3D object pose estimation for automated unloading of containers**. In Proc. of 6th European Conference on Mobile Robots (ECMR), Barcelona, Spain, 2013, pp 313-318.
  Part of Chapter 3
- *T. Stoyanov, R. Mojtahedzadeh, H. Andreasson, A. Lilienthal.* **Comparative evaluation of range sensor accuracy for indoor mobile robotics and**

automated logistics applications. Robotics and Autonomous Systems (RAS), 2012, ISSN 0921-8890, Vol. 61, pp 1094-1105.
Part of Chapter 3

- *R. Mojtahedzadeh, A. Lilienthal.* **A Principle of Minimum Translation Search Approach for Object Pose Refinement.** In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp 2897-2903.
Part of Chapter 4

- *R. Mojtahedzadeh, A. Bouguerra, A. Lilienthal.* **Automatic Relational Scene Representation For Safe Robotic Manipulation Tasks.** In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp 1335-1340.
Part of Chapter 5

- *R. Mojtahedzadeh, A. Bouguerra, E. Schaffernicht, A. Lilienthal.* **Probabilistic Relational Scene Representation and Decision Making Under Incomplete Information for Robotic Manipulation Tasks.** In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2014, pp 5685-5690.
Part of Chapter 5

- *R. Mojtahedzadeh, A. Bouguerra, E. Schaffernicht, A. Lilienthal.* **Support Relation Analysis and Decision Making for Safe Robotic Manipulation Tasks.** Robotics and Autonomous Systems (RAS), 2015, ISSN 0921-8890, Vol. 71, pp 99 - 117.
Part of Chapter 5

The following publication is not in the core contribution of this dissertation, however, it describes the results of the RobLog project and represents the work I performed during this thesis to autonomously identify safe-to-remove objects.

- *T. Stoyanov, N. Vaskevicius, C. Muller, T. Fromm, R. Krug, V. Tincani, R. Mojtahedzadeh, S. Kunaschk, R. Mortensen Ernits, D. Canelhas, M. Bonilla, S. Schwertfeger, M. Bonini, H. Halfar, K. Pathak, M. Rohde, G. Fantoni, A. Bicchi, A. Birk, A. Lilienthal, W. Echelmeyer.* **No more heavy lifting: Robotic solutions to the container unloading problem.** IEEE Robotics and Automation Magazine, to appear.

Chapter 2
Background

In many practical applications of robotic manipulation where the objects are stacked in a pile, it is of great importance to prevent other objects from moving and possibly falling down by an inappropriate selection of an object to remove from the pile. The process of automating the task of unloading goods inside cargo containers is such a real-world application that requires the ability to autonomously select safe-to-remove objects. This chapter reviews literature about the object selection problem and highlights the need for a principled treatment of the task of identifying safe-to-remove candidates from realistic configurations of objects.

The problem of algorithmic object selection for robotic manipulation is mainly investigated in the research for designing "bin-picking" robots. A robotic bin-picking system requires scene analysis, object detection and pose estimation, grasp planning, and path planning. The parts to be assembled in a production line are the main focus of industrial robotic bin-picking systems. In related work about bin-picking systems it is common to assume configurations of objects sitting on top of a table or being stacked in a bin. In such cases the problem of object selection typically addressed with a heuristic to pick up the topmost objects of a bin.

An appropriate selection of an object from an arbitrary configuration of goods which are stacked inside a cargo container requires a more complex analysis than the simple heuristic of always selecting the topmost objects. For a bin-picking scenario in which the bin is filled with a number of identical assembly parts it is a plausible strategy to identify and select the topmost object. In such scenarios it makes no difference which part is chosen to be picked up, and also motions of other parts due to the pick-up action do not matter. A cargo container filled with possibly fragile goods it is crucial to predict the effects of unloading a selected object on the stability of the pile of objects.

The key motivation of the problem addressed in this thesis can be seen in the scenario of the EU-funded project RobLog, which is summarized below.

(a) Real-world Coffee Sacks Scenario (b) Scientifically Challenging Scenario

Figure 2.1: Two sub-senarios of the RobLog project are illustrated: (a) the real-world industrial task of unloading coffee sacks neatly stacked inside shipping containetrs; (b) the scientifically more challenging scenario of a cluttered pile of objects that could come in random configurations without known models.

2.1 RobLog Project

The work presented in this thesis is motivated by and was carried out in the context of an European Union funded project titled "RobLog - Cognitive Robot for Automation of Logistics Processes" [4]. The central objective of the RobLog project was the development of a robotic manipulation system for the task of unloading goods from cargo containers. A large portion of trading goods are packaged and shipped in standardized containers. While some of the tasks along the logistic chain can be performed by machines, manually unloading goods from containers is a strenuous and costly job presenting a key bottleneck in the process. Therefore, safe and reliable automated container unloading machines constitute a commercially and socially important research area. The ultimate goal of the RobLog project was to develop solutions for the scientific challenges on the road to automated container unloading. With the contribution of this thesis the project successfully demonstrated prototypes of fully autonomous robotic unloading systems [4].

In order to be economically feasible, the robotic manipulation system must be very robust, efficient and safe in comparison to manually unloading goods by human workers. The lack of automation in unloading of containers is mainly due to the complexity of the task, which must be accomplished under restricted time demands and requires a high level of software and hardware capabilities. A further challenge is the high variability of the objects shipped in the containers in terms of shapes, size, texture and material. The unstructured arrangement of objects loaded into the cargo containers requires the robot to be able to deal with unknown configurations of objects. It is not unusual to observe that several goods might topple over when picking up one object from a cargo con-

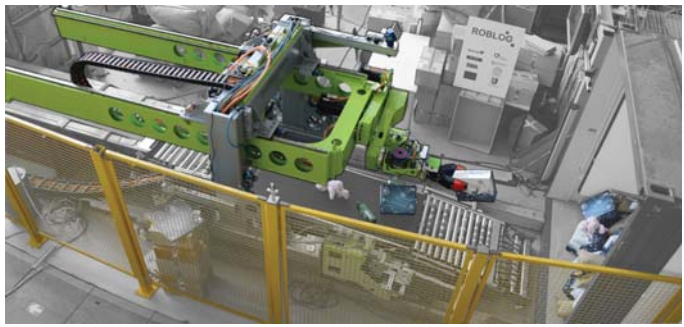


Figure 2.2: Industrial Robotic Platform of the RobLog Project

tainer. A proper choice of the object to unload reduces the risk of accidental damage to other objects due to toppling over or falling down.

The RobLog scenario requires reliable capabilities to address the problem of selecting objects to pick from varying, possibly heterogeneous, and potentially chaotically arranged goods inside cargo containers. The project addressed two different sub-scenarios, one motivated by a real-world industrial task of unloading 70 kg coffee sacks stacked inside cargo containers (see Figure 2.1a), and another scenario aims at the scientifically more challenging domain of unloading containers (see Figure 2.1b). The latter scenario contributes to research on autonomous manipulation in unstructured environments that piles of objects may have random configurations, and that there exist objects without known models. Figure 2.2 depicts the industrial robotic platform developed for the scientific scenario in the design process of the RobLog project.

In the scenario of unloading coffee sacks, a heuristic approach of always selecting the topmost sack from the front layer could efficiently be employed as long as coffee sacks are neatly stacked in layers on top of each other. The assumption that the objects are neatly stacked considerably simplifies reasoning on the geometry of the pile and reduces the complexity of the scene analysis. However, such a simplified strategy of always selecting the topmost object fails in more complicated configurations where there are complex gravitational support relations between the objects, and where the problem of undetected objects is more severe. The work of this thesis is dedicated to develop algorithms and present methodologies for the problem of selecting safe-to-remove objects in the more challenging scenario of the RobLog project.

2.2 Related Work

Despite the importance of making a safe decision about the sequence of objects to remove from a pile for autonomous robotic manipulation, only a few

(a) Bley et al. [5]

(b) Jang et al. [6]

(c) Klingbeil et al. [7]

(d) Kenney et al. [8]

(e) Agrawal et al. [9]

(f) Real-world Containers

Figure 2.3: (a)-(e) depict typical configurations of objects in related works to bin-picking research. (f) depicts two real-world configurations of carton boxes inside shipping containers at unloading sites.

papers address this problem. The problem of object selection is occasionally briefly mentioned within bin-picking literature about object localization. This section therefore, first, reviews bin-picking literature with a focus on the target scenarios and the object selection task. Then, the few available related works that specifically attempt to identify gravitational support relations between objects of a pile are reviewed, and their limitations are highlighted. Table 2.1 categorizes the related work reviewed in this chapter based on three items, the scenario, the properties of the objects and the type of analysis to represent an overview of the differences between this work and the related work.

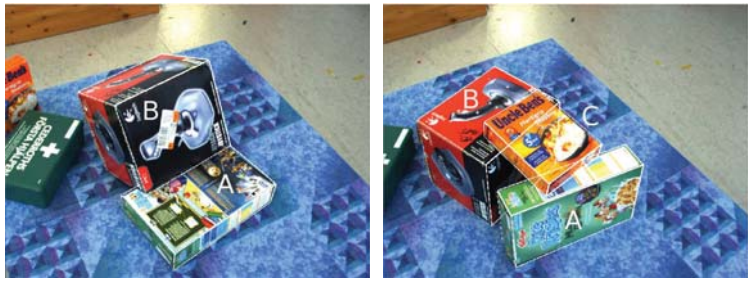
2.2.1 Bin-Picking

In an early work by Ikeuchi et al. [10] in 1983, a bin-picking system was introduced based on an analysis of the surface normals extracted from a stereo vision sensor. The main focus of their paper is to address the problem of how to isolate an object from the background, and how to determine the relative pose of the object with respect to the camera. One year later in 1984, Horn and Ikeuchi [11] published their study about manipulation of randomly oriented parts where they present an object template matching method to autonomously determine the orientation of parts in a pile.

Dessimoz et al. [12] propose a fast filtering approach to detect potential holdsites – a location on an object at which to grasp the part – in images to decrease the burden of scene analysis on low computational powered computers made available in 1980's. Yang and Kak [13] describe strategies for analyzing structured-light range maps for determining the identity and pose of the topmost object in a pile. Al-Hujazi and Sood [14] propose a range image segmentation method based on region growing technique to determine the best holdsite position and orientation of objects for bin-picking. Rahardja and Kosaka [15] present a vision-based bin-picking technique to identify and estimate the pose of assembly parts by stereo vision data, where in particular the objects are alternator covers. Berger et al. [16] propose a three steps methodology for bin-picking where in the first step the robot picks the topmost object from a bin with a vacuum gripper to drop it in an empty workplace, then the CAD model of the object is fit to a structured light image of the workplace to determine the pose, and finally the correct mounting of the part is being ensured. Agrawal et al. [9] present a bin-picking system with model based 3D pose estimation and with the ability of picking singulated 3D objects. They evaluate the performance of the system on experimental setups with few objects sitting on a flat ground and being clearly separated. In a work by Kenney et al. [8], an interactive segmentation of cluttered scene is presented, where objects are sitting on a tabletop without being completely nor partly supported by each other. Tabletop scenarios in which objects are either clearly separated or being in a simple interaction are widely used in literature, to name but a few, in a grasp planning based on generic object knowledge by Bley et al. [5], a real-time motion planning for manipulation of objects by Jang et al. [6], an assistive mobile manipulator implementation for helping people with motor impairments by Jain and Kemp [17], a grasp selection algorithm by Klingbeil [7], a framework for push-grasping [18] and a physics-based grasp planning [19] by Dogar et al. Figure 2.3 shows a few sample configurations of objects used in tabletop scenario-based research.

Chaotically stacked objects are also considered in the literature. An approach to interactive singularization of a pile of objects presented by Chang et al. [21] which in essence gathers information about a cluttered scene by iteratively moving hypothetical objects and observing the outcome of taking such actions. A similar interactive approach for LEGO bricks sorting is presented by Gupta and Sukhatme [22]. The key problem with the interactive approach when dealing with real-world goods stacked inside shipping containers is the fact that it cannot be afforded to risk the possibility of letting objects (e.g., carton boxes of electronic appliances) fall down in order to identify the objects.

In all the studies related to bin-picking, the main research focus is on localization and manipulation of objects, and the essential hypothesis is that the topmost object is the best candidate to be selected; a multitude of experiments are conducted with the objects sitting on a tabletop scenario and clearly separated for easy detection and manipulation.



(a) Sjöö et al. [20] (b) Sjöö et al. [20]

Figure 2.4: Two real-work experimental scenarios of cuboid shaped objects used in the work by Sjoo et al. [20]. (a) An object labeled as B is leaning on another object labeled A with a probability of $\mathrm{P}(\mathtt{ON}(A, \mathrm{B})) = 0.25$, and (b) the objects A and B support another object labeled C with probabilities $\mathrm{P}(\mathtt{ON}(\mathrm{C}, A)) = 0.28$ and $\mathrm{P}(\mathtt{ON}(\mathrm{C}, A)) = 0.30$. A major drawback of this approach is that it is not clear how to choose a threshold to infer logical values of the ON relations.

2.2.2 Support Relation Analysis

In a closely related work to this thesis by Sjöö et al. [20], gravitational support of a cuboid shaped object by another is represented as symbolic ON relation between the objects and modeled to be a function of the minimum of an exponential distance factor and a sigmoid-shape contact factor. A conditional probability distribution over poses of the supporting object is then computed and thresholded to imply the logical value of the ON relation. Figure 2.4 shows two real-world experimental scenarios composed of a few cuboid shaped objects. The most complex real-world scenario investigated in [20] consists of three boxes, A, B and C where C is supported by two others (see Figure 2.4b), and the extracted probabilities for ON(C,A) and ON(C,B) are reported to be both less than 0.3, while both A and B clearly support C. One major drawback of this approach is that in a cluttered pile of objects (see real-world examples in Figure 2.3f) where objects are in complex contact with each other, and consequently there would be a set of ON relations with small probabilities, it is not clear how to choose a threshold to imply the logical truth of the ON relations.

Figure 2.5 depicts one type of configurations of objects in which the topmost object is not the best candidate to remove from the pile. In the shown configuration, object A is on top of other objects but it actually supports object B which is located under object A. If we take the approach proposed in [20], the probability that object B is on object A will be close to zero, while it can be clearly seen that there is a high probability that object A supports object B.

Figure 2.5: A class of configurations of objects in which the heuristic of always unloading the topmost object does not result in a safe choice. Object A is on top of other objects, but if we choose and remove object A from the configuration, object B will fall down due to fact that object A supports object B.

As it will be discussed further in Section 5.2.3, selecting the topmost objects to remove from a pile is not a reliable and safe strategy to unload goods from shipping containers.

In addition to the approach described above, methods to learn support relations have been investigated. Kopicki et al. [23] study the problem of predicting the behavior of rigid objects in the domain of robotic push manipulation, which is, as discussed above, not applicable for a static configuration of goods inside shipping containers. Rosman and Ramamoorthy [24] present a method for learning spatial relationships between objects from the segmented point clouds. In their work a potential lack of information about the complete set of objects and physical interactions between objects are not considered. Sjöö and Jensfelt [25] present a method to learn models for functional spatial relations from experience where they use physics simulation to learn about configuration of objects. In their experiment a simulated solid square surface is used as a tabletop on which other simulated objects are stacked on top of each other. Panda et al. [26] attempt to learn the "object-object interaction" only for three simple interactions of stacked objects on tabletop scenarios, namely, *support from below*, *support from side* and *containment*. In the target scenario of this dissertation, however, objects inside shipping containers could be configured in a totally random manner, and the configurations are unknown beforehand. Moreover, it is a requirement to deal with the case of having access to only a subset of the objects in the configuration.

2.3 Discussion

This chapter presented an overview of the problem of algorithmic object selection for robotic manipulation systems. A review of state-of-the-art literature

related to the problem of object selection highlighted the type of objects and scenarios considered in the related work. The European Union funded project, RobLog, aiming at automating the task of unloading goods from cargo containers was introduced, where the challenge of algorithmic selection of safe-to-remove objects for the RobLog scenario is one of the motivations of this thesis.

The task of object selection is mainly considered within "bin-picking" research and is normally addressed with a simple heuristic of always selecting the topmost object from a bin. While such a heuristic is plausible for a bin filled with identical assembly parts, it may not result in a safe choice when dealing with piles of objects. Configurations of goods stacked inside cargo containers are one of real-world examples of piles that the unloading strategy of always selecting the topmost objects may cause the pile to collapse. Real-world piles such as the RobLog scenario represent a cluttered scene of objects that may come in random configurations where the objects cannot afford and sustain tumbling and falling over.

The few available research papers that specifically study and propose probabilistic and learning based methods for identifying the spatial and gravitational support relations between objects have been reviewed. The single probabilistic method attempts to estimate a probability of an on-relation between two box-shaped objects. It is not clear however, how to select a threshold to infer logical on-relation between two objects. The other methods reported on learning push manipulation, spatial relationships of a segmented point cloud, functional spatial relations and object-object interaction are not capable of dealing with uncertainty and the lack of information about the objects composing a pile.

		[10, 11]	[12]	[13]	[14]	[15, 16]	[9]	[8]	[5, 17]	[6]	[7]	[18]	[19]	[21]	[22]	[25]	[24]	[23]	[26]	[20]	this work
Scenario	Containers	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
	Cluttered Scene	✓	✓	✓	-	✓	-	-	-	-	✓	-	-	✓	✓	✓	✓	-	✓	✓	✓
	Tabletops	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	-
	Separated Objects	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	-	-
	Bins	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
Objects	Cargo goods	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
	Identical parts	✓	✓	-	✓	✓	✓	-	-	-	-	✓	-	✓	✓	✓	-	-	✓	✓	✓
	Primitive shapes	✓	-	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Everyday things	-	-	-	-	-	-	✓	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-
Analysis	Support relation	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
	On relation	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	✓	✓	-
	Topmost Selection	✓	✓	✓	-	✓	-	-	-	-	✓	-	-	-	✓	✓	-	-	-	-	-
	Push manipulation	-	-	-	-	-	-	✓	-	-	-	✓	✓	✓	✓	-	-	✓	-	-	-
	Spatial relationship	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	-	-	-
	Segmentation	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	-	-	✓	-	✓	-	✓	-	-

Table 2.1: Related work comparison table. The features are divided into three categories based on target scenarios, type of objects and the analysis.

Chapter 3
3D Range Sensor Selection

The recent developments in range sensing devices introduced relatively low-cost solutions for dense 3D range measurements. Among different technologies, the long distance measurement and accuracy of 2D laser range finders (LRFs) outperforms other competitor devices [27, 28]. Commercially available compact designs of 3D laser range finders (e.g., Velodyne LiDAR) are prohibitively costly. A popular alternative and cost efficient solution widely used in the robotics community is to mount a 2D laser range finder on a tilting actuator — known as an actuated LRF (aLRF). Nevertheless, the systematic errors, low refresh rates and the required mechanical parts for the actuation are the major limitations of using the aLRFs in robotic systems.

In order to overcome the shortcomings of actuated LRFs, a number of commercially available competing technologies have been recently developed. Popular and widely used among the robotics groups are time-of-flight (TOF) and structured light cameras. An inexpensive technology of TOF cameras exploits the relation between the phase shift of the reflections of a modulated light and the distance of the surface of the reflections (e.g., SwissRanger SR-4000 and Fotonic B70). Structured light cameras, on the other hand, estimate distances similar to stereo vision systems by measuring the disparity of a projected light pattern on a CCD camera (e.g., the Kinect sensor).

This chapter concerns an application centric evaluation of 3D range sensors used for selecting appropriate 3D perception technology in the development of the RobLog project (see Chapter 2). The performance of four carefully selected 3D range sensors, an actuated SICK LMS200 laser range finder, two TOF cameras SwissRanger SR-4000 and Fotonic B70, and a Microsoft Kinect sensor is evaluated for the task of object detection and pose estimation. A number of configurations of three commonly-used objects inside shipping containers, namely, carton boxes, sacks and tires is created for data generation. Two representative state of the art object detection approaches are selected as performance indicators. It will be demonstrated that sensor characteristics other than the traditionally evaluated distance accuracy can influence the performance of

the target application. Therefore, this chapter makes a case for an application-based evaluation of 3D range sensors — the device with the best performance with respect to the object detection task is selected for use in the final automated system.

3.1 An Overview of Range Sensor Evaluation

The current literature on 3D range sensor evaluation abounds with examples of the characterization of the intrinsic parameters and sensor calibration. Ye and Borenstein [29] present a characterization study of the SICK LMS200 laser scanner. They investigate the effect of a number of parameters, such as operation time, data transfer rate, target surface properties, as well as the incidence angle on the device sensing performance. Luo and Zhang [30] report the characterization of the laser range finder AccuRange 4000 by Acuity Research. They study the performance of the ranging device under various operating conditions including lighting, temperature, and surface color, and orientation. A group of researchers reported their study on the calibration of the available TOF cameras in literature [31, 32, 33]. The utility of TOF cameras in robotics problems such as pose estimation [34], 3D mapping [35], 3D shape scanning [36] and collision avoidance [37] has been also evaluated.

Introducing a low-cost structured light camera, the Kinect sensor by Microsoft motivated researchers to study the properties and the utility of the sensor in robotics domain. Khoshelham and Elberink [38] study depth accuracy and resolution, and point density of the Kinect sensor and report a calibration parameters for the infrared and color cameras of the sensor. Chin et al. [39] present an investigation of the quality of depth data obtained by the Kinect sensor. DiFilippo and Jouaneh [40] report the accuracy, repeatability, and resolution of the different Kinect models in determining the distance to a planar target.

Having single-sensor characterization and parameter evaluation, selecting a set of range sensors for a complex robotic system solely based on a comparison between the intrinsic properties and in isolation of the target task may result in an inappropriate choice. Wong et al. [28] evaluated the utility of ten 3D range sensors in a holistic manner for a real-world industrial application — underground void modeling. They define a set of representative metrics of the target application (mapping a tunnel) and evaluate the range sensors based on the obtained metrics. From their experimental results in situ mapping evaluation, while a class of sensors perform better in obtaining some metrics, they represent a weaker ability for other metrics. As they concluded, the selection for the appropriate sensor considers a right balance of performance, mass, features and cost. In the article [27] that the author is involved we develop a holistic method for the measurements accuracy evaluation of a set of 3D range sensors — namely, the Swiss Ranger SR-4000, Fotonic B70 and Microsoft Kinect using an actuated laser range finder as reference. Observing the results in [27],

it is not immediately clear which sensor would represent a better performance for a complex robotic system such as the RobLog project. In the author's later work [41], which this chapter is based on, we evaluated the same set of 3D range sensors for the target application of the RobLog scenario. As the discussion at the end of this chapter concludes, evaluating 3D range sensors based on an application centric performance reveals the underlying capabilities of different sensors in dealing with diverse configurations of the target application.

3.2 Application Centric 3D Range Sensor Evaluation

In order to compare two given 3D range sensors, S_i and S_j with sets of properties $\mathbf{p}_i^n = \{p_{i,1}, \ldots, p_{i,n}\}$ and $\mathbf{p}_j^m = \{p_{j,1}, \ldots, p_{j,m}\}$ respectively, we define $\mathbf{p}_c = \mathbf{p}_i^n \cap \mathbf{p}_j^m$ and call the elements in $\mathbf{p}_c$ *comparable properties*. For 3D range sensors, the properties such as the distance accuracy, the level and type of noise, field of view, the point cloud density, and the lens distortion can be considered. Some of the properties (e.g., the distance accuracy) may be found in both sensors (which are the elements in $\mathbf{p}_c$) while some other properties (e.g., the lens distortion) may be specific for one of the sensors. Comparing the sensors based on the effects of the properties that are not comparable is not trivial. On the other hand, let's assume that for the target application (e.g., object pose estimation) it is known that a subset $\mathbf{q}_c$ of $\mathbf{p}_c$ contains all the properties that have a direct effect on the performance of the target application. Preferring a sensor solely based on comparing the $\mathbf{q}_c$ properties and in isolation from the target application is made difficult because, although different properties represent different aspects of the sensor, there can be correlations between the effects of the sensor properties on the performance of the target application. Having this said, selecting a set of 3D range sensors in a holistic manner — when designing autonomous systems with specific target applications, is suggested.

3.2.1 Performance Indicators

The target application is the detection and pose estimation of the most popular categories of goods, carton boxes and tires [42], that shipping containers are typically filled with. As performance indicators, two different approaches to estimating object poses from 3D sampled points (e.g., point clouds) are used. The first approach is based on extracting the local features FPFH (Fast Point Feature Histogram) [43] while the second approach, proposed by Detry and Piater [44], is based on a probabilistic framework that can achieve object detection by avoiding explicit model-to-scene correspondences.

For the first indicator, FPFH features are initially computed from the identified interest points of the object templates and the scene. Then the Sample Consensus Initial Alignment algorithm (SAC-IA, see section IV in [43]) runs to roughly align the object template to the scene. The final step is to perform a local optimization using Levenberg-Marquardt (LM) algorithm to minimize

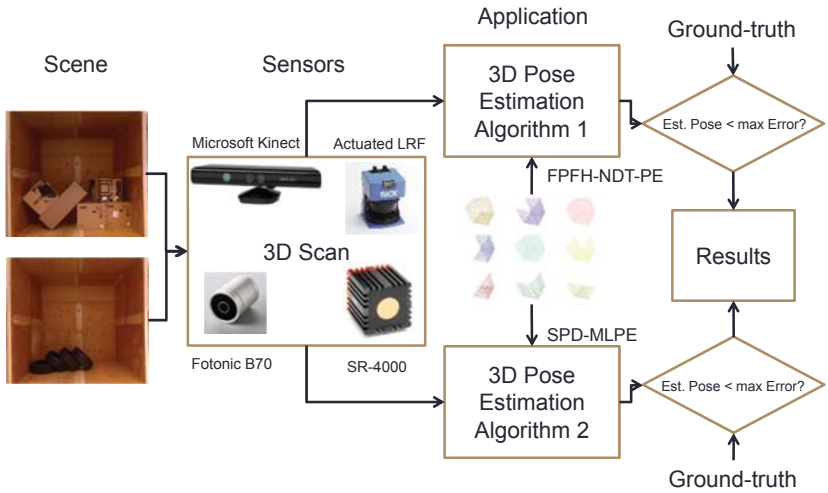


Figure 3.1: Application Centric 3D Range Sensors Evaluation Block Diagram

the distance between the object template and the scene points. The experimental results showed that the final step often fails to produce fine-aligned results, although SAC-IA is able to roughly align the object templates to the scene point cloud. As an alternative to the final fine-alignment step, a 3D-NDT based registration [45] was examined that turned out to be more successful than LM optimization. This pose estimation approach, which is the first performance indicator, is referred to as *FPFH-NDT-PE*.

For the second indicator, a local surface normal at each point of the object template is computed using k-nearest neighbors [46]. Sampling points from an object's surface constructs the spatial configuration consisting of the point coordinates and their local orientations — *a surface-point distribution*, which has the highest values around object surfaces. Probabilistic pose inference is obtained by convolving surface-point distributions of the object template and the scene resulting in a measure of *object pose likelihood* over the entire scene. Pose estimation, is then performed by searching for the maximum likelihood. The method is capable of learning an initial model from only one view-point of the object template, i.e., it can also work with partial models. It is demonstrated that the performance of this probabilistic approach is competitive to the other state-of-the-art algorithms on public datasets (see Evaluation Section in [47]). Moreover, this approach is intended for detection and localization of objects within cluttered scenes such as the objects filled in shipping containers.

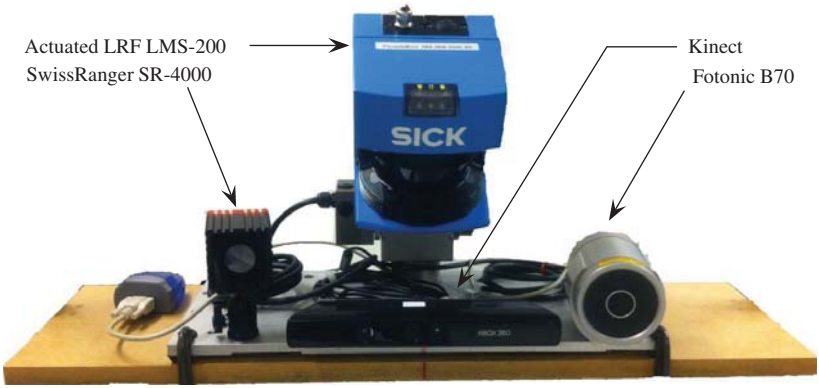


Figure 3.2: The picture shows the setup of 3D range sensors for data collection.

3.2.2 Evaluation Methodology

The evaluation process starts with capturing a 3D scan of the target scene by a set of 3D range sensors. The captured data is then fed to the performance indicators where the object detection and pose estimation algorithms attempt to find the best match of the given objects templates to the captured data and estimate the poses. The estimated poses of the target objects are then compared to the ground truth poses of the object instances in the scene. The error in the estimated translation is defined to be the Euclidean distance between the ground truth reference point of the template in the scene and its estimated translation. For the orientation error, the angle between the ground truth reference frame in the scene and its estimated rotation is measured. If the translation and orientation errors are both less than user defined thresholds the returned pose is accepted as a successful estimation. The performance criterion is the success rate which refers to the number of successful estimations of the target object divided by the total trials. Figure 3.1 shows the evaluation procedure in block diagram format.

3.2.3 Data Collection

For collecting data, a set of different arrangements of two selected objects (i.e., carton boxes and tires) inside a mock-up container was used to generate several data sets (see Figure 3.3). The dimensions and type of the selected carton box and tire to be detected are $0.59 \times 0.57 \times 0.55$ meters and P205/55R1691V respectively, which are popular packaging dimension and tire size shipped over European countries. The algorithms of the performance indicators require templates of the objects. A cuboid and a cylinder approximate the geometric shape of the templates for the carton box and tire respectively (see Figure 3.4).

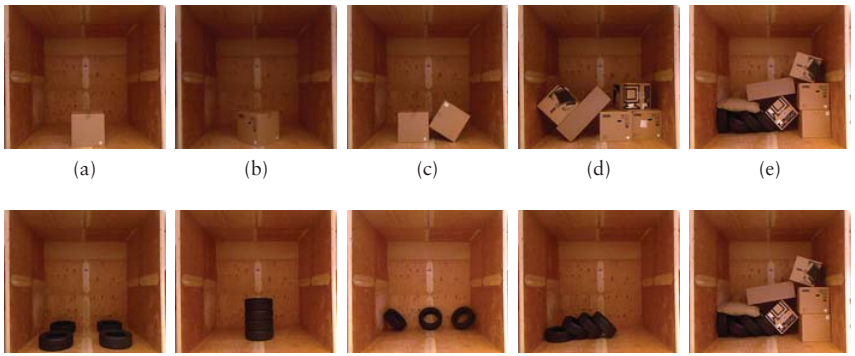


Figure 3.3: Different arrangements of carton boxes and tires inside a mock-up container used for data collection by four 3D range sensors: an actuated SICK LMS-200 laser range finder, two time-of-flight cameras: Fotonic B70 and SissRanger SR-4000, and a Microsoft Kinect structured light camera.

In the experimental setup, an actuated SICK LMS-200 laser range finder, two time-of-flight cameras: Fotonic B70 and SissRanger SR-4000, and a Microsoft Kinect structured light camera were selected for the evaluation and mounted on a rigid portable stand (see Figure 3.2). The height of the sensor board was set to be approximately equal to the middle height of the mock-up container. For all the sensors their factory pre-calibrations were used in the experiment. Table 3.1 represents the comparable properties ($\mathbf{p}_c$) of the selected 3D range sensors.

For each arrangement, 10 complete scans were captured by the sensors at six equally spaced distances (0.5 meters) away from the front edge of the mock-up container starting at 0.5 meters. The mixed measurements in the aLRF data are filtered out using the method explained in the article [27]. Since the mock-up container itself is not of interest, i.e., it is assumed that the size and pose of the container are known, the floor, ceiling and walls of the container in the captured data were filtered out in a pre-processing step.

For each target object in the arrangements the ground truth pose was extracted by manual registration of the object's template to the scene point cloud using the aLRF data. Each performance indicator sequentially searches for the instances of the input target object in the scene point cloud and returns a list of the estimated poses. The estimated poses of the target object are then compared to the ground truth poses of the instances in the corresponding arrangement.

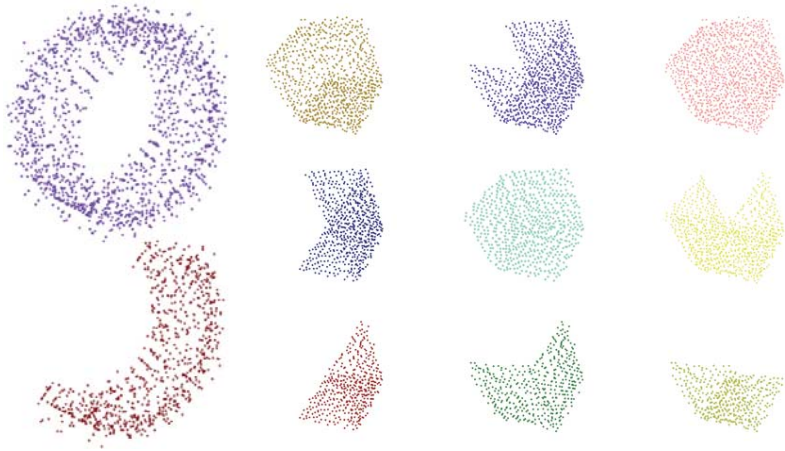
Actuated LRF LMS-200
- FOV (h×v): 180° × 45°
- Resolution: 181 × 850 (150k average points per scan)
- Maximum Range: 8m
- Frame Rate: 0.1Hz
The Kinect sensor
- FOV (h×v): 57° × 43°
- Resolution: 640 × 480 (220k average points per scan)
- Maximum Range: 3.5m
- Frame Rate: 30Hz
SwissRanger SR-4000
- FOV (h×v): 43° × 34°
- Resolution: 176 × 144 (25k average points per scan)
- Maximum Range: 5m
- Frame Rate: 35Hz
Fotonic B70
- FOV (h×v): 70° × 50°
- Resolution: 160 × 120 (19k average points per scan)
- Maximum Range: 7m
- Frame Rate: 25Hz

Table 3.1: Set of comparable properties ($\mathbf{p}_c$) of the sensors.

3.2.4 Results

For each combination of the sensors, target objects (the box and the tire) and the performance indicators, the results are presented in bar graphs of the overall success rates (in percentage) with respect to the distance of the sensors to the entrance of the container as explained in the previous sections (see Figure 3.5). The graphs with total null performance, which occurred in some combinations consisting of the tire object, are not shown. Observing the performance of the indicators for detecting the box from the data captured by the sensors, it can be seen that the indicator SPD-MLPE outperforms FPFH-NDT-PE (Figures 3.5a, 3.5b, 3.5g and 3.5h in comparison with Figures 3.5d, 3.5e, 3.5j and 3.5k respectively). However, when the target object is the tire, the indicator FPFH-NDT-PE shows a better and more stable performance than SPD-MLPE (Figures 3.5f and 3.5c in comparison with Figures 3.5l and 3.5i). The success rates for detecting the tire are considerably lower than that of detecting the box, though.

Comparing the sensors based on the performance of the indicators, the TOF camera SwissRanger SR-4000 shows a more consistent performance than other sensors in detecting the target box regardless of the indicator algorithm (see Fig-



(a) Tire templates (b) Cuboid templates

Figure 3.4: (a) Two templates extracted from a cylinder shape; (b) Nine templates extracted from a cubiod shape representing the selected carton box and tire.

ures 3.5g and 3.5j), although its capability to detect the target tire is limited (see Figure 3.5i). The other TOF camera, Fotonic B70, shows a null performance in detecting the target tire while it is capable of being used for detecting the target box with an overall low and dependent performance on the selected indicator (see Figures 3.5h and 3.5k). The structured light camera Kinect shows a dependent performance on both the selected indicator and target object type. While a combination of the Kinect sensor and the indicator SPD-MLPE detects the target box with a high success rate (see Figure 3.5b), the same combination shows a null performance in detecting the target tire. The actuated laser range finder is the only 3D range sensor in this experiment that its data can be used for detecting both target objects using the selected indicators, although its performance drops dramatically in detecting the tire object.

The analysis of the results highlights the fact that the selection of 3D range sensors highly depends on the target application — the object types and the object detection and pose estimation algorithms in this experiment.

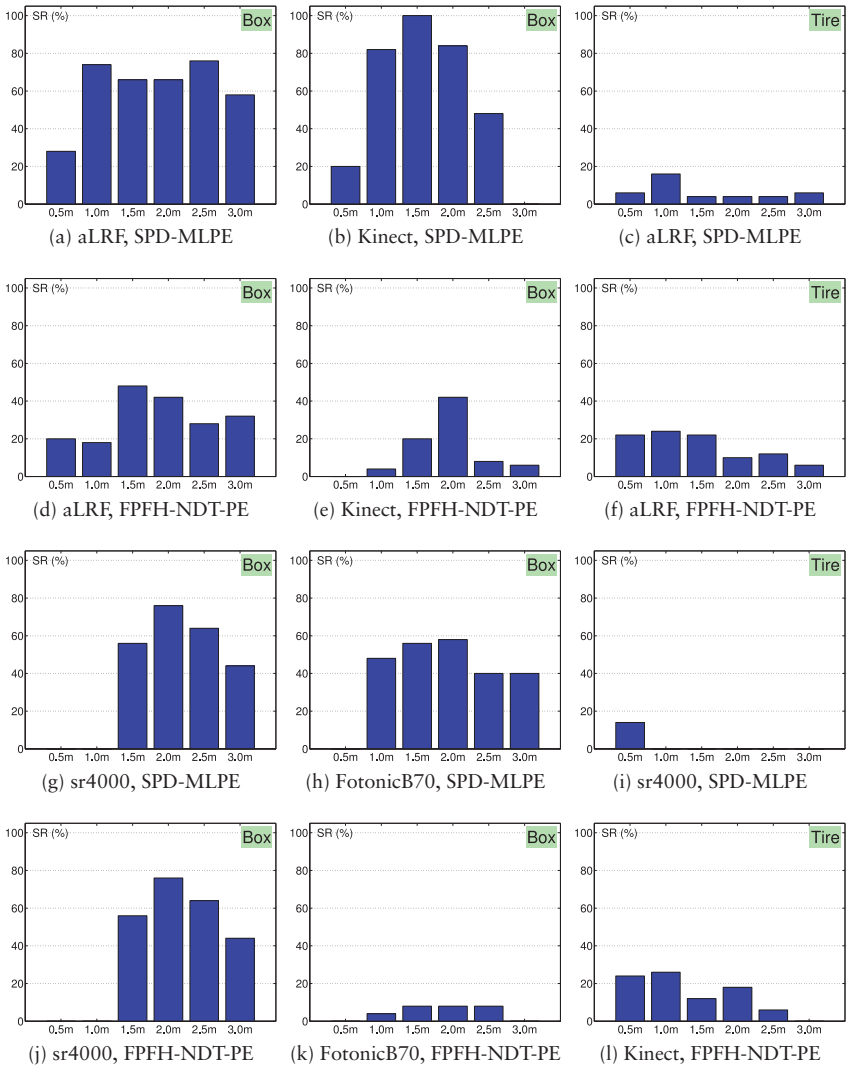
3.3 Discussion

This chapter proposes to evaluate the utility of a set of 3D range sensors based on their performance in the target application to select the most applicable 3D range sensors in the design process of a complex robotic system. It is argued

that the selection of 3D range sensors solely based on the characteristics of the sensors and in isolation of the target application may result in an inappropriate selection. For example, in a study of the characteristic of the laser range finder SICK LMS200 by Ye and Borenstein [29] they examine the effect of target surface properties by three groups of materials — namely, shiny colors, matted colors and gray levels (see Section 4.3 in [29]). From their experiment of evaluating the range measurement distribution (see Figure 5c in [29]) from white to black surfaces we can observe slightly more than 0.6% mean error. However, such characteristic is not adequately informative for us to predict, for instance, how well the laser range finder would perform for detecting and pose estimating of the tires in comparison with carton boxes stacked inside shipping containers. In the results section of this chapter, on the other hand, it can be observed that the laser range finder performance considerably drops when dealing with tires in comparison with carton boxes.

In order to evaluate the performance of the 3D range sensors in the target application, the object detection and pose estimation task in the scenario of RobLog project was used as performance indicator. The results show that the dark surfaces with tread patterns, as they can be found on the surface of tires, significantly absorb infrared light of the TOF camera SwissRanger SR4000. Such dark surfaces, although not to the same extent, also substantially reduce the performance of the laser range finder SICK LMS200 and the structured light camera Kinect sensor. In conclusion, we observe that TOF cameras are not appropriate choice for detecting objects like tires, Kinect-type sensors do not perform better, and even laser range finders have difficulties with such objects.

The experiments presented in this chapter also suggest that the performance of the different 3D range sensing technologies varies greatly over different object and surface types. The best overall combined detection rates (in comparison with aLRF as reference) were obtained by the most dense range sensor — namely, the structured light camera Kinect sensor.



(a) aLRF, SPD-MLPE (b) Kinect, SPD-MLPE (c) aLRF, SPD-MLPE

(d) aLRF, FPFH-NDT-PE (e) Kinect, FPFH-NDT-PE (f) aLRF, FPFH-NDT-PE

(g) sr4000, SPD-MLPE (h) FotonicB70, SPD-MLPE (i) sr4000, SPD-MLPE

(j) sr4000, FPFH-NDT-PE (k) FotonicB70, FPFH-NDT-PE (l) Kinect, FPFH-NDT-PE

Figure 3.5: Success rate bar graphs for each combination of sensor model, object type and performance indicator. Horizontal axis is the distance of the corresponding sensor to the container, and vertical axis is the average success rate of all scenarios at each distance step in percentage.

Chapter 4
Object Pose Refinement for Geometrical Consistency

A complete and accurate estimation of the poses of the objects is of great importance especially for high level reasoning (as it is the main topic of the next chapter) and motion planning for manipulation of the objects. State-of-the-art object pose estimation methods (e.g., [48, 49]) represent the uncertainty in their estimations, which may result in a geometrically inconsistent model of the environment due to inter-penetrations between pairs of adjacent objects. For example, a carton box that is partly (or completely) overlapping with the floor or a wall of a container is not consistent with a rigid body assumption.

This chapter concerns the problem of refining the initially estimated poses of a set of objects in order to obtain a geometrically consistent (i.e., an inter-penetration free) model of the environment. A search based methodology is presented in this chapter to resolve such inter-penetrations between rigid objects leading to a refinement of the poses. It should be noted that the type of search presented in this chapter differs from the search for the initial poses that an object pose estimation algorithm performs. In other words, the ultimate goal is to refine and not to estimate the initial poses.

A number of approaches have been proposed to estimate the poses of objects from 2D images [50, 51, 52] and 3D sampled points [53, 54, 48]. The main focus of the proposed approaches for object pose estimation is to obtain accurate object poses, while geometrical consistency of the estimated poses has received less attention. For example, Lim et al. [55] describe a fine pose estimation method to fit 3D models of IKEA furniture to the images. They use a database of the 3D models and define a multi-criteria score function to find the best fit for the models in an image. As their results show, the error in the pose of the fit 3D model may result in an inter-penetration with the environment. For instance, the fit 3D model of an IKEA bookcase considerably intersects with the ground floor due to the error in the estimated pose of the bookcase. Such

geometrically inconsistent situations can be resolved, for example, by a collision detection algorithm (that is expected to push the bookcase up) resulting in a higher accuracy of the estimated pose.

The presence of other objects – either fixed (e.g., a wall) or movable – nearby the target object corresponds to additional geometrical constraints which requires extra analysis. Grundmann et al. [56] present a probabilistic approach, called Rule Set Joint State Update, to estimate the poses of a set of objects simultaneously using an approximation for the full joint posterior. They assume independence between prior belief, measurement and prediction models to approximate the full state. The results of the proposed method, however, are presented on tabletop scenarios with only one object. Aldoma et. al [57] describe an approach for verifying 3D models of objects (hypothesis verifying) in cluttered scenes according to a global optimization paradigm by minimizing a cost function which encompasses geometrical cues. The ultimate goal of their method is to select the best set of models and poses from a given pool of hypotheses subjective to maximize the number of correct recognitions while minimizing the number of wrong recognitions. Hypothesis verifying may improve the quality of object recognition and pose estimation. However, there is no guarantee that the verified hypothesis represent an inter-penetration free configuration of objects. Wong et. al [58] propose collision-free state estimation where they attempt to solve a constrained optimization problem in order to find a feasible collision-free configuration. They assume that all the objects are resting stably on a 2D surface (i.e., no object is on top of another object). In their method, the projections of the objects onto the 2D surface create a set of boundaries, and the inter-penetrations between the boundaries are resolved through optimization. However, the method is not applicable for the problems where goods are usually stacked on top of each other (e.g., shipping containers) with arbitrary configurations.

Another approach that one may consider is to utilize the collision resolvers of Physics Engines (e.g. see [59]) to tackle the problem of inter-penetrations between a set of static objects. However, the collision resolvers of Physics Engines are based on dynamic collision detection where impulse forces are used to simulate the trajectories of two objects after their dynamic impact. Such impulse force based algorithms when initialized with a static configuration of overlapping objects result in a spread of objects far from their initially estimated poses.

The approach presented in this chapter, on the other hand, attempts to resolve all the initial inter-penetrations between objects with minimum change in their initially estimated poses and independently of the corresponding object recognition and pose estimation algorithm.

In what follows, the computation of the *depth of penetration* is sketched, including a review of the existing methodologies for convex and concave shaped objects. Then, the algorithm to compute the depth of penetration of convex polytopes based on the Separating Axis theorem is described. This will be followed by formal definition of a graph search problem to resolve the inter-

penetrations between objects. Two selected discrete search algorithms are then described to be applied to the graph search problem. Next, the results of applying the approach of this chapter to the data generated in simulation and from real-world setups are presented. The chapter concludes with a discussion of the methodology employed to achieve a geometrically consistent model of the environment.

4.1 Depth of Penetration Computation

The inter-penetration between two overlapping polytopes can be represented by another polytope that contains the overlapping space. This representation is a precise description of the inter-penetration space while the computation of the overlapping polytope, especially in 3-dimensional space is expensive [60]. Although the volume of the overlapping space can be used as a measure for the amount of inter-penetration between two polytopes, the overlapping space provides no clue about how to separate two overlapping polytopes.

Another representation of the overlapping space between two polytopes is an inter-penetration vector such that translating one of polytopes by the vector will resolve the inter-penetration with the minimum possible translation; the length of this vector is referred to as *depth of penetration* (DOP). Zhang et. al [61] study the generalized depth of penetration where both translation and rotation are considered. The generalized depth of penetration is the minimum length of a trajectory along which moving a polytope will disjoint two overlapping polytopes. They prove [61, Theorem 1] that for convex polytopes the general depth of penetration is equal to the translational depth of penetration. A number of algorithms have been proposed for computing the depth of penetration. A category of algorithms is based on the relation between the Minkowski sum and DOP of two polytopes [62, 63, 64]. Another approach to compute DOP is based on the separating axis theorem (SAT) [60], which is widely used in computer graphics and physics simulations for collision detection. SAT is a corollary of the separating hyperplane theorem [65], which is an essential theory in convex set analysis. While both approaches, based on Minkowski sum and SAT can be used for computing the depth of penetration, SAT is a faster algorithm for polyhedrons that have less features (faces and edges) [66] such as cuboids and cylinders, where we notice that they are good geometrical representations for carton boxes and barrels shipped in cargo containers. This dissertation describes and implements 3-dimensional SAT algorithm to compute the minimum translation vector between two overlapping convex polyhedrons.

4.1.1 SAT Algorithm

The separating hyperplane theorem states that for two convex sets A and B, either the two sets are overlapping or there exists at least one separating hyper-

Algorithm 4.1: Computation of MTV and DOP

Data: Vertices and `LSet` of two convex polytopes, A and B
Result: MTV and DOP of A and B

```
DOP ← inf;
MTV ← 0⃗;
for each axis L in LSet do
    project vertices of A and B on L;
    compute each projection interval on L;
    if two intervals intersect then
        d ← the length of intersection;
        if d < DOP then
            DOP ← d;
            MTV ← DOP · l̂;
        end
    else
        DOP ← 0;
        return;
    end
end
```

plane P such that A is on one side of P and B is on the other side. The normal of a separating hyperplane is called a separating axis for the two convex sets.

For two non-overlapping convex polytopes, A and B, if L is a separating axis along the unit vector $\hat{l}$, then the orthogonal projections of A and B on L result in two non-overlapping intervals (see Fig. 4.1a). In other words, if there exists at least one axis on which the orthogonal projections of two convex polytopes have non-overlapping intervals, then the two polytopes are separated.

On the other hand, if A and B are two overlapping convex polytopes, in order to separate them with minimum translation, it is adequate to compute the orthogonal projections of A and B on all their *fundamental axes*, `LSet` and select the axis on which the overlapping interval (DOP) is minimum; the vector along this axis with DOP length is called *minimum translation vector* (MTV). In 3-dimensional space, for each pair of convex polyhedrons, A and B, the set of fundamental axes, `LSet` contains all the normals of the faces as well as all possible cross products between the edges of A and the edges of B [60]. In Fig. 4.1b, the set of fundamental axes for computing overlapping intervals of two polytopes, A and B is depicted for the 2D case. The procedure of computing MTV and DOP for two convex polyhedrons is presented in Algorithm 4.1.

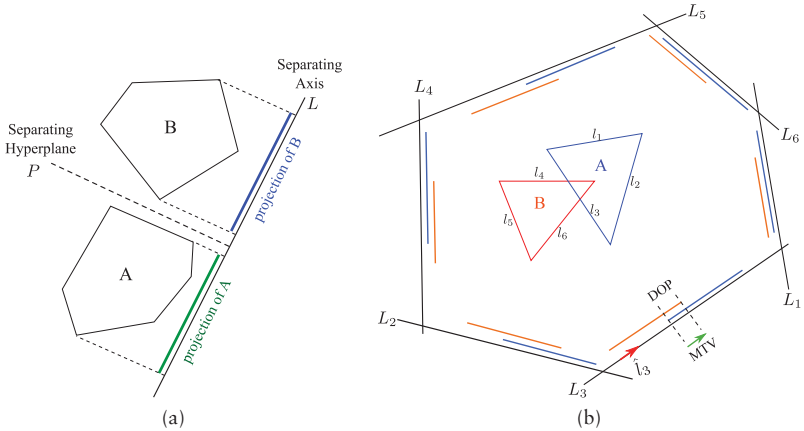


Figure 4.1: (a) A Separating axis and a separating hyperplane of two non-overlapping polytopes. (b) A and B are two overlapping convex polytopes. The set of fundamental separating axes, $\texttt{LSet} = \{L_1, \ldots, L_6\}$, which are along the normals (e.g., $\hat{l}_3$) of edges of A and B are drawn. The minimum overlapping intervals of the orthogonal projections of A and B on axes in LSet is found on L_3. The minimum translation vector (MTV) and depth of penetration (DOP) are identified by the overlapping projections on L_3. Translating A by MTV (or B by negative MTV) resolves the inter-penetrations between A and B with minimum translation.

4.2 Pose Refinement Search

This section describes a discrete search approach in the state space of minimum translations vectors to obtain an inter-penetration free configuration of a set of convex polytopes. A search in the state space of poses to obtain an inter-penetrations free configuration of more than two objects is necessary. For the sake of illustration of the problem, Fig. 4.2 depicts a toy configuration of three movable polytopes, A, B and C, and one fixed convex polytope, *W*, in which A and B are overlapping. It can be seen that translating A by the MTV introduces a new inter-penetration between A and *W*, and translating B by the negative MTV introduces a new inter-penetration between B and C. Hence, a search into the state space of poses is required to reach an inter-penetration free configuration, while minimizing the sum of changes in the initial poses of the polytopes. It is worth mentioning that depending on the initial configuration of the objects and the structure of the environment, there might be no solution resulting in an inter-penetration free configuration.

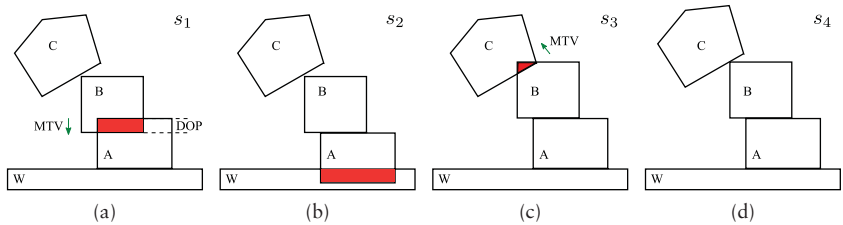


Figure 4.2: A configuration of three movable objects, A, B and C, and a fixed object, *W*. (a) Initial state, s_1 with an inter-penetration between A and B which generates two possible actions, $\mathbf{a}(s_1) = \{a^1_A, a^1_B\}$. (b) Taking action a^1_A translates A by MTV resulting in a new inter-penetration between A and *W*. Since *W* is fixed, there is only one possible action in s_2, $\mathbf{a}(s_2) = \{a^2_A\}$, where taking a^2_A goes back to s_1, and hence this path in the search will not be expanded further. (c) Taking action a^1_B translates B by negative MTV which results in a new inter-penetration between B and C. Since C is a movable object, there are two possible actions in s_3, $\mathbf{a}(s_3) = \{a^3_B, a^3_C\}$. (d) Taking a^3_C results in s_4 which is an inter-penetration-free configuration, i.e., a goal state.

4.2.1 Minimum Translations Search Problem

In this section the problem of searching for an inter-penetration free configuration of a set of polytopes is formally defined.

Definition 4.1. *A state,* s *is a configuration of polytopes with a set of poses denoted by* $\mathcal{P}(s)$*. The initial state,* s_0 *is the given configuration of polytopes that search progress starts with.*

Definition 4.2. *A set of possible actions in each state* s *denoted by* $\mathcal{A}(s)$ *is defined as below. For each pair of overlapping objects, i.e.,* $\forall \mathrm{DOP}_{ij} \in s$ *such that* $\mathrm{DOP}_{ij} \neq 0$*, two possible actions,* $a_i, a_j \in \mathcal{A}(s)$ *is defined such that*

- a_i *translates* i*-th object by the* MTV_{ij}*;*
- a_j *translates* j*-th object by the negative* MTV_{ij}*.*

If a static object (e.g., a wall) overlaps with a movable object, only the action that translates the movable object is considered in the search (see Fig. 4.2*).*

The structure of the search space is a graph. This implies from the fact that a state s' may have been reachable from multiple predecessors $(s_1, s_2, \ldots, s_n)$ by taking different sequence of actions; there may be more than one goal state (i.e., an inter-penetration free configuration) in the graph.

It is worth mentioning that in the actual implementation of generating actions, in order to prevent the search from visiting redundant states it is required

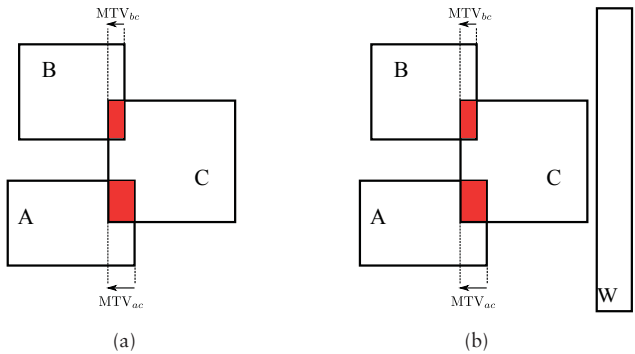


Figure 4.3: (a) A sample configuration illustrating that Eq. 4.2 is not admissible in general. (b) The same configuration of objects in (a) with a more realistic assumption of existing a static object where Eq. 4.2 estimates the cost of reaching a goal state exactly.

to preserve the visited states in memory. For an illustration, see Fig. 4.2 where back translation of A in s_2 results in a redundant state same as s_1 which is already visited.

Although the search for a sequence of the aforementioned actions resulting in an inter-penetration free configuration is not tied to specific search algorithms, this dissertation employs A-star and Depth Limited search methods [67] to demonstrate the utility of the proposed method for object pose refinement.

4.2.2 A-star Search

A-star is a search algorithm guided by the cost function $f(s) = g(s) + h(s)$, where $g(s)$ is the actual cost to reach the state s from the initial state, s_0, $h(s)$ is a heuristic function that estimates the cost to reach a goal state, s_g, from s, and $h(s_g) = 0$. In order to find a path from the initial state to a goal state with the minimum cost, A-star algorithm first evaluates the state with minimum $f(s)$ value. As it is shown in [67], if the heuristic function is *admissible*, i.e., the value of $h(s)$ is always lower than or equal to the actual cost of reaching a goal state from s, A-star algorithm guarantees to find one of possible shortest path (i.e., the optimum path) from s_0 to s_g. If for every state s, $h(s)$ is able to exactly compute the actual cost of reaching a goal state from s, the A-star algorithm will follow one of the shortest path to a goal state without evaluating other possibilities, resulting in a very fast search. On the other hand, as the value of a heuristic function becomes lower than the exact cost, the more possibilities have to be evaluated, which is making the search slower.

Algorithm 4.2: Object Pose Refinement Using A* Search

Data:
The Set of Convex Polytope Models of Objects, $\mathcal{M} = \{m_1, \ldots, m_n\}$
The Set of Initially Estimated Poses of Objects, $\mathcal{P}_0 = \{p_1^0, \ldots, p_n^0\}$
Result: A Sequence of Translation Actions, $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$;
SolutionMap $\leftarrow \emptyset$;
OpenSet $\leftarrow \mathcal{P}_0$;
$g(\mathcal{P}_0) \leftarrow 0$;
$f(\mathcal{P}_0) \leftarrow$ `Total_DOP`$(\mathcal{P}_0, \mathcal{M})$;
while OpenSet *is not empty* **do**
  $\mathcal{P}_c \leftarrow \underset{\mathcal{P} \in \text{OpenSet}}{\text{argmin}}\, f(\mathcal{P})$;
  if `Total_DOP`$(\mathcal{P}_c, \mathcal{M}) = 0$ **then**
    $\mathcal{S} \leftarrow$ `Construct_Solution(SolutionMap)`;
    return $\mathcal{S}$; `// a solution found`
  end
  ActionsSet $\leftarrow$ `Generate_All_Actions`$(\mathcal{P}_c, \mathcal{M})$;
  for *each action* a *in* ActionsSet **do**
    $\mathcal{P}_a \leftarrow$ `ExecuteAction`(a);
    if $g(\mathcal{P}_a)$ *is not defined* **then** $g(\mathcal{P}_a) \leftarrow \infty$;
    $g' \leftarrow g(\mathcal{P}_c) +$ `DOP`(a);
    if $\mathcal{P}_a$ *is not in* OpenSet **then**
      OpenSet.Add$(\mathcal{P}_a)$;
    else if $g' \geqslant g(\mathcal{P}_a)$ **then**
      continue;
    end
    SolutionMap.Add$(\langle \mathcal{P}_c, a, \mathcal{P}_a \rangle)$;
    $g(\mathcal{P}_a) \leftarrow g'$;
    $f(\mathcal{P}_a) \leftarrow g(\mathcal{P}_a) +$ `Total_DOP`$(\mathcal{P}_a, \mathcal{M})$;
  end
  OpenSet.Remove$(\mathcal{P}_c)$;
end
return *failure*;

In the case of graph search, where a state may be reachable from multiple predecessors, the optimality of A-star algorithm additionally requires that the heuristic function is consistent. If $h(s)$ is the heuristic function that estimates the cost to reach the goal state from s and $c(s, a', s')$ is the actual cost of tak-

ing action $a' \in \mathcal{A}(s)$ to go from s to the successor state s', then the heuristic function $h(.)$ is said to be consistent if,

$$h(s) \leqslant c(s, a', s') + h(s'). \tag{4.1}$$

It is noted that if a heuristic function is consistent, it is also admissible [67]. However, an inadmissible heuristic function cannot be consistent as the inequality 4.1 will not hold if $h(.)$ overestimates the cost to reach a goal state (i.e., $h(s)$ may be greater than $c(s, a', s') + h(s')$).

For the graph search problem defined in Section 4.2.1 finding a heuristic function that for each state exactly computes the cost of reaching a goal state is not trivial. One difficulty is due to the fact that although a translation of a polytope along the corresponding MTV resolves one inter-penetration but it may introduce one or more inter-penetrations with other polytopes, which cannot be seen before translating a polytope. On the other hand, using a heuristic function that estimates a lower bound for the cost results in a very slow search.

It should be noticed that what really matters in our search problem is to find an inter-penetration free configuration (i.e., a goal state) with a minimum total pose distance between the initial state, s_0 and the goal state, s_g. Having this said, in order to accelerate the A-star search in a large state space, a heuristic function that for some states is able to estimate the exact cost is selected as below,

$$h(s) = \sum_{i,j} \mathrm{DOP}_{ij}(s) \ , \ i, j = 1, \ldots, N \ , \ i \neq j. \tag{4.2}$$

The motivation is the fact that a goal state is a configuration of objects in which all the inter-penetrations have been resolved, hence, from any state it is more likely to reach a goal state by translating polytopes along the corresponding MTVs. However, as mentioned above, Eq. 4.2 is not an admissible heuristic function since it may overestimate the cost of reaching a goal state. In Fig. 4.3a if we move C along negative MTV_{ac} we reach a goal state, while the sum of depth of penetrations in this state (which is $|\mathrm{MTV}_{bc}| + |\mathrm{MTV}_{ac}|$ as Eq. 4.2 computes) overestimates the cost to reach a goal state. Nonetheless, from the experimental results (see Section 4.3) it is observed that in many cases when the minimum required cost to reach a goal state is equal to all the translation actions that must be taken to resolve the inter-penetrations, the heuristic function in Eq. 4.2 can be used with good results. This can be especially seen where there exists static objects (e.g., walls) that limit the space of moving objects. Fig. 4.3b shows how existing an additional static object nearby the dynamic objects in Fig. 4.3a limits their movability where Eq. 4.2 estimates the exact cost of reaching a goal state.

The pseudo code shown in Algorithm 4.2 presents an implementation of the A-star search for object pose refinement. The function `Total_DOP`$(\mathcal{P}, \mathcal{M})$ computes the sum of depth of penetrations in a configuration identified by the set of shapes, $\mathcal{M}$ and the corresponding poses, $\mathcal{P}$ according to Eq. 4.2, while the

function DOP($\mathfrak{a}$) returns the depth of penetration corresponding to the action $\mathfrak{a}$, and the function Generate_All_Actions($\mathcal{P}, \mathcal{M}$) generates all actions according to Definition 4.2; the function ExecuteAction($\mathfrak{a}$) returns the new set of poses after the execution of the action $\mathfrak{a}$. If a solution is found, the sequence of actions that the execution of them results in an inter-penetration free configuration, i.e., a set of poses, $\mathcal{P}_{\text{goal}}$ that satisfy the condition Total_DOP($\mathcal{P}_{\text{goal}}, \mathcal{M}$) $= 0$, is returned by the function Construct_Solution().

The solution that A-star search finds (if there exists any) is a sequence of actions such that their execution results in a transition from start to goal state with a minimum total cost of taken actions. Since a goal state is an inter-penetration free configuration, and the total cost of reaching the goal state in many cases is minimal, a solution returned by the A-star search algorithm satisfies the two criteria: maximizing geometrical consistency and (sub-optimally) minimizing the sum of translations required to reach a goal state.

4.2.3 Depth Limited Search

The state space of MTVs can grow exponentially as the successors of states are expanded. This may limit the number of objects and the inter-penetrations that the proposed A-star search is capable to deal with in a reasonable time. In order to accelerate the search in a large state space, Depth Limited Search (DLS) algorithm is selected.

DLS only explores a branch of the state space and finds suboptimal but geometrically consistent solutions. A suboptimal solution is a sequence of translation actions that results in an inter-penetration free configuration while the total cost of taking the actions is not necessarily minimum. On the other hand, as it is mentioned earlier in Section 4.2, there could be configurations of overlapping objects for which no goal state exists (i.e., there exists no inter-penetration free configuration). In such cases an unlimited search algorithm may generate infinitely many intermediate states. DLS overcomes this issue by limiting the depth of search into the state space.

With similar sub-functions explained for Algorithm 4.2, the pseudo code shown in Algorithm 4.3 presents a recursive implementation of the depth limited search for object pose refinement. The user selected input, *limit* in the algorithm limits the depth of search, where a *cutoff* message is propagated into recursive calling of the function RecursiveDLS() in order to execute the next possible action and explore another branch of search space.

4.2.4 Concave Shaped Objects

The extension of the search process to cover the objects with concave shapes can be easily conducted by slightly modifying the graph search problem defined in Section 4.2. The first method is based on the decomposition of a concave shaped object into a set of connected convex shapes [68], where the idea is to

Algorithm 4.3: Object Pose Refinement Using Depth Limited Search

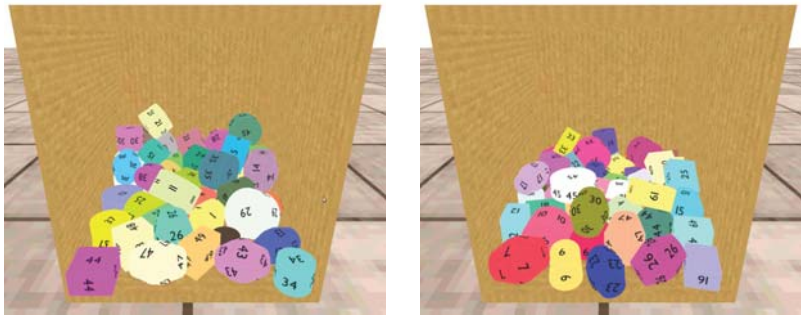
Data:
The Set of Convex Polytope Models of Objects, $\mathcal{M} = \{\mathfrak{m}_1, \ldots, \mathfrak{m}_n\}$
The Set of Initially Estimated Poses of Objects, $\mathcal{P}_0 = \{\mathfrak{p}_1^0, \ldots, \mathfrak{p}_n^0\}$
The Maximum Depth of Search, *limit*
Result: A Sequence of Translation Actions, $\mathcal{S}$

```
1  S ← ∅;
2  SolutionMap ← ∅;
3  return RecursiveDLS(P_0, M, SolutionMap, limit);
4  Function RecursiveDLS(P_c, M, SolutionMap, limit) is
5      if Total_DOP(P_c, M) = 0 then
6          S ← Construct_Solution(SolutionMap);
7          return S; // a solution found
8      else if limit = 0 then
9          return cutoff;
10     else
11         cutoff_status ← false;
12         ActionsSet ← Generate_All_Actions(P_c, M);
13         for each action a in ActionsSet do
14             P_a ← ExecuteAction(a);
15             SolutionMap.Add(⟨P_c, a, P_a⟩);
16             result ← RecursiveDLS(P_a, M, SolutionMap, limit − 1);
17             if result = cutoff then cutoff_status ← true;
18             else if result ≠ failure then return result;
19         end
20         if cutoff_status = true then return cutoff else return failure;
21     end
22 end
```

translate the whole shape of a concave object if a translation action applies to one of convex parts of the concave object. The second method is based on the computation of the shortest trajectory (i.e., a combination of translations and rotations) along which transforming one of overlapping objects will resolve the inter-penetration [61]. Replacing the minimum translation vectors with the shortest trajectories as the definition of actions will extend the search process to cover concave shapes.

4.3 Results

This section presents results showing the performance of the object pose refinement approach on both simulated and real-world data. Using scenarios generated in simulation enables us to create a large data set of different con-



(a) (b)

Figure 4.4: A few samples of simulated configurations with random arrangement of objects inside a shipping container.

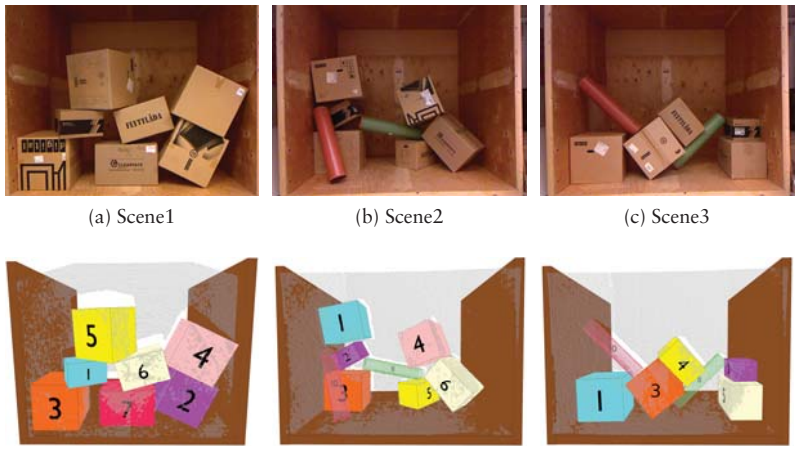
figurations of objects with their ground truth poses to capture the statistical properties of the approach. The real-world configurations are used to verify the approach on real data.

In order to fairly compare the search algorithms, the total number of visited nodes for both DLS and A-star search algorithms are limited to 5000, and the maximum depth for DLS is limited to 1000.

4.3.1 Simulated Configurations

Three categories of shapes of objects commonly used in shipping containers, i.e., box, cylinder and barrel are selected to generate random configurations in simulation. A physics engine is used to create a 20′ standard shipping container with randomly arranging $N_G \in \{10, 20, 30, 40, 50\}$ objects inside. For each number of objects, N_G, 40 samples of configurations are generated, and the number of shapes in each configuration are equally likely drawn from the three categories with uniform random dimensions. In addition to N_G objects, there exist 6 static objects: left, right, back wall, floor and ceiling of the container as well as the ground plane that supports all other objects (see Fig. 4.4 for a few examples).

For each configuration, a Gaussian noise, $\mathcal{N}(0, \sigma^2)$ is added to each component of the translation vectors and the Euler angles of the objects poses to generate a set of noisy poses. The noisy poses simulate the error in the estimated poses by an existing object detection algorithm. In this experiment, a standard deviations of 0.05 meters and 5 degrees are selected for the translation and the rotation components respectively.



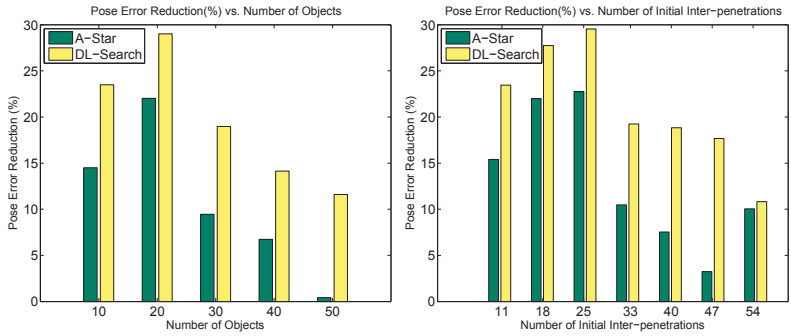
(a) Scene1 (b) Scene2 (c) Scene3

(d) Scene1 Ground Truth (e) Scene2 Ground Truth (f) Scene3 Ground Truth

Figure 4.5: (a-c) Three real-world configurations of objects inside a mock-up shipping container. (d-f) The ground truth of the objects 3D models and poses.

4.3.2 Real-World Configurations

A set of real-world configurations of objects inside a mock-up shipping container used for evaluating the approach on real data (see upper row of Fig. 4.5) A Microsoft Kinect sensor looking at the entrance of the mock-up shipping container captures a point cloud of the scene. The set of 3D models of the objects are then registered to the scene point cloud, and the poses are manually refined for obtaining the ground truth of the poses (see lower row of Fig. 4.5). In order to examine the approach independent of and not tied-to any particular object pose estimation algorithm, a set of noisy poses are sampled from the ground truth of the poses. This means that we can expect the same results if the noise that comes from the sensing and estimation process is distributed in the same way. Similar to the generated configurations in simulation, a Gaussian noise, $\mathcal{N}(0, \sigma^2)$ is added to each component of the translation vectors and the Euler angles of the poses, which may result in a configuration of overlapping adjacent objects (see Fig. 4.8a for an example). A total of 1000 samples per each real-world configuration (see upper row of Fig. 4.5) are generated with a standard deviation of 0.05 meters for the translation noise, and a standard deviation of 5 degrees for the rotation noise.



(a) (b)

Figure 4.6: Results for the simulated configurations. The average of PER with respect to (a) the number of objects; (b) the number of initial inter-penetrations between pairs of objects.

4.3.3 Evaluation

The result of a search by A-star and DLS is considered as a *successful search* if a valid solution (i.e., an inter-penetration-free configuration) can be found with the specified limitation of the number of node visits and search depth. The *success rate* is the percentage of successful searches.

In order to evaluate pose accuracy, let us define *pose error reduction* (PER) as the difference between the initial pose error (IPE) and the refined pose errors (RPE)

$$\mathrm{PER} = \frac{\mathrm{IPE} - \mathrm{RPE}}{\mathrm{IPE}} \times 100\% \tag{4.3}$$

where,

$$\mathrm{IPE} = \sum_{i=1}^{N_G} \|\mathbf{t}_i^d - \mathbf{t}_i^g\| \tag{4.4}$$

$$\mathrm{RPE} = \sum_{i=1}^{N_G} \|\mathbf{t}_i^r - \mathbf{t}_i^g\| \tag{4.5}$$

and $\mathbf{t}_i^g$, $\mathbf{t}_i^d$ and $\mathbf{t}_i^r$ are the translation vectors of the i-th object's ground truth, detected (i.e., noisy) and refined (i.e., a goal state) poses respectively. A positive value of PER indicates a reduction in the refined poses with respect to the initially detected poses.

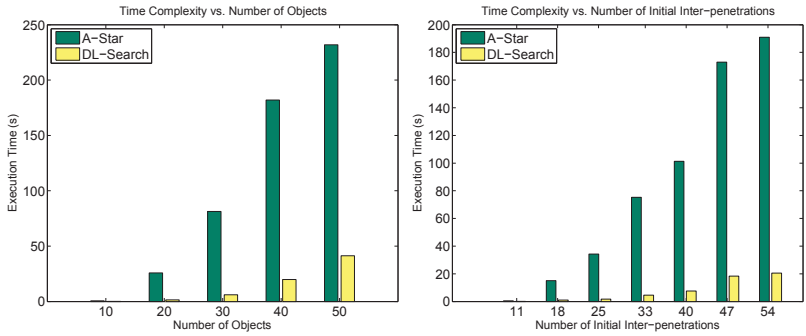


Figure 4.7: Results for the simulated configurations. The average of execution time with respect to (a) the number of objects; (b) the number of initial inter-penetrations between pairs of objects.

No. Objects	10	20	30	40	50
A* Success Rate	80%	72.5%	35%	25%	2.5%
DLS Success Rate	100%	100%	100%	100%	100%

Table 4.1: The success rate (see Section 4.3.3) of the proposed search algorithms with respect to the number of objects.

Simulated Configurations

In Fig. 4.6a and Fig. 4.6b, the average of PER for each search method is depicted with respect to the number of objects and initial inter-penetrations between objects respectively. Fig. 4.7a and Fig. 4.7b show the average execution time for the search methods with respect to the number of objects and the initial inter-penetrations respectively. It can be seen that both search methods are approximately equally fast for scenarios with 10 objects. However, as the complexity of the scenarios increases with an increasing number of objects, the execution time for A-star rapidly increases, while the depth-limited search algorithm is able to resolve the inter-penetrations between objects in highly cluttered scenarios still in a reasonable time (less than 50 seconds on average for scenarios with 50 objects).

Table 4.1 depicts the success rate of the search algorithms with respect to the number of objects. While depth-limited search manages to find a goal state for all the simulated test scenarios, A-star with the proposed approximate heuristic shows a decreasing performance as the number of objects increases.

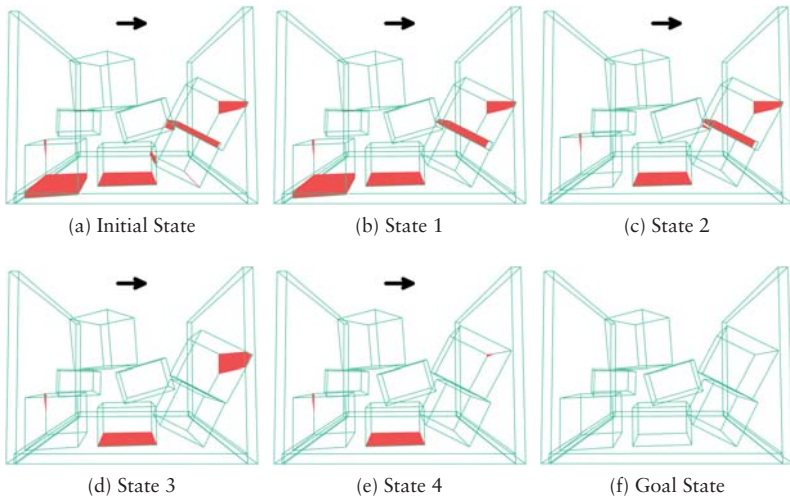


Figure 4.8: (a) An example of the result of a noisy estimated poses of a pile of objects inside a shipping container where the inter-penetrations between adjacent objects are highlighted. (b) A goal state, that is, an inter-penetration free configuration is shown. (c),(e),(f) and (d) depict in order four intermediate states in the search space.

Real-World Configurations

Fig. 4.8 visualizes a typical path found through a search for an inter-penetration free configuration of the real-world setup shown in Fig. 4.5a. The initial state, i.e., a noisy pose estimation of the objects introduces inter-penetrations between pairs of objects (see Fig. 4.8b). Executing the corresponding translation actions along the found path results in a goal state (see Fig. 4.8f), where the inter-penetrations are resolved. A few intermediate states are shown in Fig. 4.8b through Fig. 4.8e in the order in which they will be reached from the previous states by the execution of the corresponding action. In Table 4.2 the results of applying the proposed search methods to the real-world configurations are summarized. The first observation is that both search methods reduce the average pose error and result in configurations of objects which are geometrically consistent. It can be also seen that the proposed approach is computationally inexpensive (less than 100 mili-seconds) for real-world configurations where the number of visible objects to the perception module is less than 10. The success rate of A-Star search, as in simulation, is less than with DLS, which manages to successfully find inter-penetration-free configurations in all the trials. We also

	Scene 1		Scene 2		Scene 3	
	A*	DLS	A*	DLS	A*	DLS
Avg. ExT (ms)	9.53	0.69	45.31	2.86	72.60	4.47
Succ. Rate(%)	96.1	100	95.0	100	91.7	100
Avg. PER (%)	8.6	11.3	8.9	8.0	19.6	18.0
Avg. RPE (m)	0.511	0.496	0.580	0.586	0.500	0.510
Avg. IPE (m)	0.559		0.637		0.622	
Avg. IOL (#N)	6		5		6	

Table 4.2: The results of applying the proposed search methods on real-world configurations (see Fig. 4.5). From top to bottom row, the values of average execution time in milliseconds (Avg. ExT), success rate (Succ. Rate), average pose error reduction (Avg. PER, see Eq. 4.3), average refined pose error (Avg. RPE, see Eq. 4.5), average initial pose error (Avg. IPE, see Eq. 4.4), and initial average number of overlaps (Avg. IOL).

observe that the results obtained for the real-world data is consistent with that of simulated data.

4.4 Discussion

In this chapter an algorithm was proposed to resolve the inter-penetrations between a set of convex shaped objects due to errors in the initially estimated poses. Resolving the inter-penetrations results in a geometrically consistent model of the environment that a robotic system works in. The target application of the framework presented in this chapter is to refine the poses of the detected objects inside shipping containers in the process of automating the task of unloading goods. However, the framework can be easily adopted for other applications such as domestic robotics where robots are dealing with everyday objects.

The approach is based on the computation of the minimum translation vectors between pairs of overlapping convex objects, where the separating axis theorem is used for this purpose. A discrete search paradigm in the state space of the minimum translation vectors is defined to find an inter-penetration free configuration of objects. The utility of two search methods, A-star and depth limited search examined for exploring a solution in the state space of minimum translation vectors. Nevertheless, the extension of the approach to cover concave shaped objects based on either the decomposition into a set of convex shapes or the direct computation of the shortest resolving penetration trajectory is discussed.

The fact that a solution to the problem is solely based on a high level reasoning and not tied to any object detection and pose estimation algorithm may suggest that resolving the inter-penetrations results in less accurate poses. However, the experimental results show that resolving the inter-penetrations not only represents a geometrically consistent model of the environments, but also reduces the total pose error on average.

The approach was tested and verified on data sets generated from real-world and simulated configurations. From the results we can observe that using the depth-limited search technique significantly prunes the state space to find a geometrically consistent solution. The results also suggest that a trade-off analysis between computational resources and the amount of resolved inter-penetrations with respect to the number of objects is necessary to select a proper search paradigm.

Chapter 5
Support Relation Analysis and Decision Making

Considering the real-world task of unloading goods autonomously, two previous chapters of this dissertation discussed the problems of selecting appropriate 3D range sensors for object pose estimation and refining the estimated poses to obtain geometrically consistent models. This chapter analyzes the problem of identifying safe-to-remove objects from a pile and presents algorithms to reason about the stability of the pile with respect to the configurations of the objects. In the context of unloading a pile of objects, a candidate object is safe to be unloaded if the pile remains static by removing the candidate. Expressly, the ultimate goal is to avoid causing the other objects to move (e.g., fall down) by removing an object from a pile.

For human beings, using the knowledge acquired through experience and the senses, it may be trivial to immediately identify which objects are safe to remove from a pile. But how can we algorithmically implement such cognitive ability into robots? This chapter is an attempt to answer the preceding question from deterministic to probabilistic manner. In order to autonomously select safe-to-remove objects, a robotic manipulation system needs two main abilities. First, it needs to be able to create models to reflect how objects in the configuration are physically interacting with each other, i.e., to identify which objects are supporting other objects. Second, it should be able to use the created models to make an optimal decision regarding which object is the safest to remove.

The approach of this chapter incrementally relaxes a set of assumptions on the input data to address more complicated, real-world scenarios. It is assumed that an existing object detection algorithm provides the input data for further analysis. In addition to the uncertainty in the estimated poses, object detection algorithms may produce false negatives, i.e., a failure in the detection of some existing objects in the scene. The lack of information about a pile and errors in

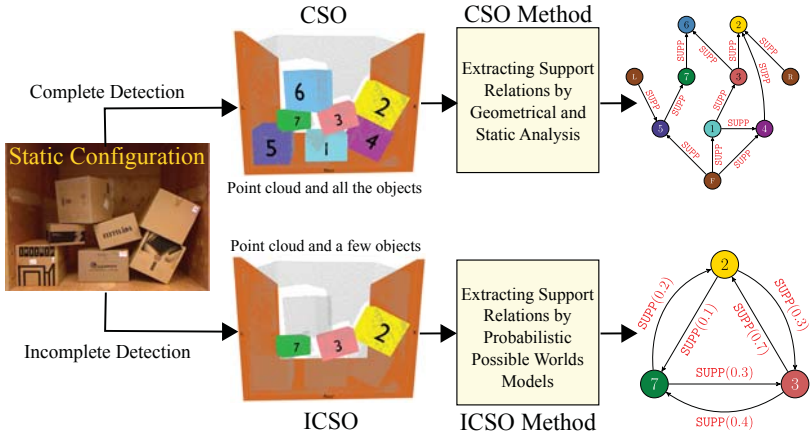


Figure 5.1: A static configuration of objects is detected by an existing 3D visual perception module. (top) All the objects in the configuration are detected (CSO case), and support relations are extracted by geometrical and static equilibrium analysis. (bottom) Only a few objects in the configuration are detected (ICSO case), and support relations between the detected objects are extracted by probabilistic possible world models.

the detected poses are two important sources of uncertainty that unavoidably complicate the analysis and decision making about the safe-to-remove objects. Depending on the available description of the objects, the problem of identifying safe-to-remove objects is divided into two major branches. In the first approach to answer the question of the previous paragraph, it is assumed that the shapes and poses of all the objects are known — this is referred to as the Complete Set of Objects (CSO) case. In the CSO case, a geometrical reasoning followed by an static equilibrium analysis identify the gravitational support relations between objects. The second approach relaxes the assumptions in the CSO case and introduces a representative probabilistic framework to address real-world issue of uncertainty in the data. In case a number of objects composing a pile are not detected, it is referred to as the Incomplete Set of Objects (ICSO) case. In the ICSO case, machine learning techniques are employed to estimate the probability of support relations, and the concept of possible world models is the basis for making an optimum decision about the safe-to-remove objects. Figure 5.1 illustrates two assumptions on the input data and the corresponding approaches in a block diagram.

This chapter is organized as follows. First, terminology and notation used throughout this chapter is described in Section 5.1. Section 5.2 explains the process of extracting gravitational support relations in the CSO case, where a geometrical reasoning to identify *act* relations and a static equilibrium anal-

ysis to extract support relations are discussed. The probabilistic approach to the ICSO case is described in Section 5.3, where this section details the procedure of learning support relations and explains the concept of the possible world models employed for a probabilistic representation of the environment. Section 5.4 explains a probabilistic decision making approach to identifying the most probable safe-to-remove objects using the representation discussed in two previous sections. Section 5.5 presents the results of the two approach on data generated in simulation and from real-world configuration of objects, and Section 5.6 concludes this chapter.

5.1 Terminology and Notation

This section defines terminology and the corresponding assumptions together with Table 5.1 showing the notations consistently used throughout this chapter. Whenever an assumption additionally made or relaxed it is mentioned inline with the text.

Definition 5.1. *An **object** is a rigid physical entity with a convex polyhedron shape.*

Definition 5.2. *A **flat ground** is a fixed object with a large cuboid shape on which other objects can sit, and the gravity force is perpendicular to the flat ground.*

In practice, a flat ground can be, for example, the floor of a shipping container, the ledge of a shelf, or a tabletop.

Definition 5.3. *The **reference frame** is a fixed three-dimensional Cartesian coordinate system with* xz*-plane representing the side of the flat ground facing up, where the gravity force direction is opposite to that of* y*-axis.*

Definition 5.4. *The **geometrical attributes** of an object are the geometry of the shape and the pose of the object with respect to the reference frame.*

Definition 5.5. *A **configuration** is an environment in which there exists one flat ground and a set of static objects with an arbitrary arrangement sitting on top of the flat ground, where the only acting force is gravity.*

The term *static configuration* is interchangeably used instead of configuration in the text whenever it is required to emphasize that objects are motionless.

Definition 5.6. *For two objects* X *and* Y *in a static configuration, if removing* X *from the configuration causes* Y *to lose its motionless state, the symbolic **support relation** between* X *and* Y *is defined and denoted by* $\texttt{SUPP}(\mathrm{X},\mathrm{Y})$*; it is read as* X *supports* Y.

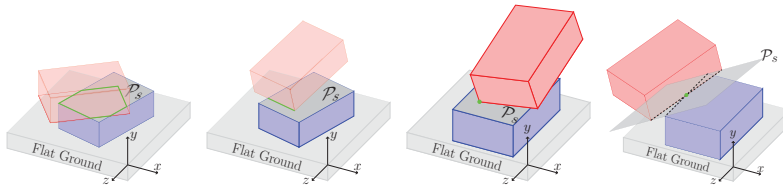
Table 5.1: Table of notations, commonly used throughout this chapter.

	General Notation
$\mathbb{R}$	the set of real numbers
$\mathbb{N}$	the set of natural numbers
$\vec{\mathbf{a}} \dots \vec{\mathbf{z}}$	column vectors in $\mathbb{R}^3$
$\vec{\mathbf{a}} \cdot \vec{\mathbf{b}}$	scalar product of $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$
$\vec{\mathbf{a}} \times \vec{\mathbf{b}}$	vector product of $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$
$\mathcal{O}$	a set of objects
$\mathcal{C}$	a configuration of objects
	Geometrical and Mechanical Notation
CPS	the contact point-set between two objects
$\mathcal{P}_s$	the separating plane between two objects
$\vec{\mathsf{F}}$	a mechanical force vector in $\mathbb{R}^3$
$\vec{\tau}$	a mechanical torque vector in $\mathbb{R}^3$
	Symbolic Relations
`ACT`	symbolic gravitational act relation
`SUPP`	symbolic gravitational support relation

A support relation is a directional symbolic relationship that can hold whether two objects are in *direct* or *indirect* contact with each other. And it should be noted that it is possible to have configurations in which both `SUPP(X,Y)` and `SUPP(Y,X)` hold, i.e., there can be a maximum of two possible support relations between two objects.

5.2 Extracting Support Relations - CSO case

In the CSO case, where all the objects of a pile are assumed to be detected, a geometrical analysis can identify which object acts on another due to gravity force. Extracting act relations between objects can explain the stability of simple configurations that objects stacked on top of each other, but it fails to identify which object supports another in more complex configurations. It may be intuitive to take notions from classical mechanics, especially statics to analyze the stability of a pile. However, in the absence of sensing masses and their distribution over geometrical shapes of objects, and the lack of information about friction coefficients between the materials of the objects it is not applicable to directly use the techniques of statics. This section presents a qualitative usage of the static equilibrium concept to extract symbolic support relations between objects with an assumption on the masses and their distributions.



(a) Face-On-Face (b) Edge-On-Face (c) Vertex-On-Face (d) Edge-Cross-Edge

Figure 5.2: Illustration of the types of contact point-sets and the corresponding separating planes, $\mathcal{P}_s$, between two convex polyhedrons in contact. (a) Face-On-Face; contact is a polygon. (b) Edge-On-Face; contact is a line-segment. (c) Vertex-On-Face; is a single point. (d) Edge-On-Edge; contact is a single point.

In order to identify act and support relations, the first step is to compute the possible contact points between each pair of the objects. We note that the object pose refinement discussed in the previous chapter can be employed to obtain an inter-penetration free configuration of objects, thus each pair of objects are either in contact or completely separated. The rest of this section describes the possible contact points between two objects with convex polyhedron shapes and how to extract the corresponding act and support relations.

5.2.1 Contact Point-Set Network

Identifying the contact points between objects forms the basis for the further geometrical and static equilibrium analysis. In a static configuration of objects where the gravity is the only acting force, the points of action of the weights, and consequently the corresponding torques between objects are determined by the contact points and the mass distribution of the objects. Since each object could be in contact with more than one other object, a network of contact points represents the topology of contacts between objects.

The contact points are computed based on the available geometrical information (shape and pose) of the objects. The geometrical consistency of configurations, as discussed in Chapter 4 suggests that the shapes of two adjacent objects cannot penetrate into each other. Among six possibilities, four types of geometrically possible contacts between two adjacent objects are considered and computed in the following order:

1. Face-On-Face. This type of contact arises when a face of one object and a face of another object partly or completely coincide. The result is a polygonal area with at least 3 vertices (see Figure 5.2a).

Algorithm 5.1: Contact Point-Set of Two Objects

```
   Data: Geometrical description of two polyhedra X and Y
   Result: CPS(X, Y)
 1 CPS(X, Y) ← GeoSetsIntersection(Faces(X), Faces(Y));
 2 if CPS(X, Y) ≠ ∅ then
 3 |   return CPS(X, Y);
 4 end
 5 CPS(X, Y) ← GeoSetsIntersection(Faces(X), Edges(Y));
 6 if CPS(X, Y) ≠ ∅ then
 7 |   return CPS(X, Y);
 8 end
 9 CPS(X, Y) ← GeoSetsIntersection(Faces(X), Vertices(Y));
10 if CPS(X, Y) ≠ ∅ then
11 |   return CPS(X, Y);
12 end
13 CPS(X, Y) ← GeoSetsIntersection(Edges(X), Edges(Y));
14 return CPS(X, Y);
15 Function GeoSetsIntersection(SetX, SetY) is
16 |   for each geometrical entity m_X in SetX do
17 |   |   for each geometrical entity m_Y in SetY do
18 |   |   |   if m_Y and m_X are in the same plane then
19 |   |   |   |   return the intersection points of m_Y and m_X;
20 |   |   |   end
21 |   |   end
22 |   end
23 |   return ∅;
24 end
```

2. Edge-On-Face. This arises when an edge of one object partly or completely touches a face of another object. The result is a line segment (see Figure 5.2b).

3. Vertex-On-Face. This arises when a vertex of one object touches a face of another object. The result is a single point (see Figure 5.2c).

4. Edge-Cross-Edge. This happens when an edge of one object intersects with, but is not parallel to an edge of another object (see Figure 5.2d).

The unstable contacts such as "a vertex of one object touches a vertex of another object" are excluded from further analysis due to the assumption that the objects are static at the perception time.

The steps to compute the contact point-set (CPS) between two convex polyhedron shaped objects are shown in Algorithm 5.1, where four types of con-

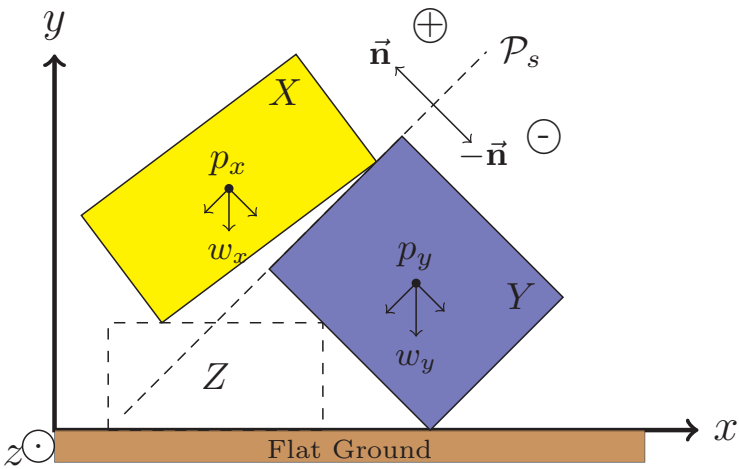


Figure 5.3: Extracting the possible ACT relation between two objects X and Y in contact (Proposition 1).

tact points in order mentioned above are evaluated. The abstract function GeoSetsIntersection(SetX, SetY) computes and returns the intersection of two sets of geometrical entities, such as faces, edges and vertices of polyhedrons. The computed contact points are used to build a network, CPSN $= (\mathcal{O}, \Omega)$ where the set of nodes, $\mathcal{O} = \{o_1, \ldots, o_N\}$ represents the objects, and the set of links, $\Omega = \{CPS(o_i, o_j) : o_i, o_j \in \mathcal{O}, i \neq j\}$ represents the set of points at contact between each pair of objects; such graph is called *contact point-set network* (CPSN) in this dissertation. We notice that $CPS(o_i, o_j)$ is an empty set if o_i and o_j are not in contact, i.e., there is no link between two objects in the network if the objects have no contact point.

5.2.2 Geometrical Reasoning

Having CPSN computed, for each pair of objects in contact, the object acting on another is labeled according to the gravity direction. From Newton's third law of motion, we know for two objects X and Y in contact, that if object X exerts a force on Y, then Y exerts a force, which is equal in magnitude but opposite in direction on X; calling X "*acting object*" and Y "*reacting object*".

The geometrical reasoning to label acting objects is based on extracting the separating plane, $\mathcal{P}_s$ between two objects. Since shapes are convex sets, according to *hyperplane separation* and *supporting hyperplane* theorems [69], for each pair of objects in contact there exists a separating plane which divides 3D space into two half-spaces such that each half-space contains only one of

the objects. The separating plane is identified by the contact point-sets such that $\mathrm{CPS}(X, Y) \in \mathcal{P}_s$ of X and Y. Figure 5.2 shows separating planes for the discussed contact types in the previous section.

The separated half-spaces are labeled as *positive and negative sides* of the separating plane. Considering the definition of reference frame in Section 5.1, a half-space is labeled as positive, respectively negative, side, if the y-component of the separating plane's normal vector at that side is strictly positive, respectively negative. In the case of a perpendicular separating plane to the flat ground (i.e., the y-component of the normal vector is zero), the half-spaces are not labeled.

In order to identify whether object X acts on another object, Y, the first step is to ignore all the other objects in contact with X and Y. The acting object is then determined according to the following proposition.

Proposition 5.1. *For two objects* X *and* Y *in contact, if their separating plane is not perpendicular to the flat ground, then the positive side of the separating plane contains the acting object, and the negative side contains the reacting object. Such a symbolic relation is presented as* `ACT(X,Y)` *which is read as "*X *acts on* Y*".*

Proof. Without loss of generality, let us assume that the positive side of the separating plane contains X and the negative side contains Y (see Figure 5.3). If $\vec{\mathbf{n}}(n_x, n_y, n_z)$ is the normal vector of the separating plane in the positive side (i.e., n_y is strictly positive) and $\vec{\mathbf{w}}(0, -w, 0)$, $w > 0$ is the weight of an arbitrary small piece of Y, it can be shown that none of such pieces of Y can exert force on X due to their weights. To do this, let us compute the projection of the weight, $\vec{\mathbf{w}}$, on the normal vector $\vec{\mathbf{n}}$,

$$\mathrm{Proj}(\vec{\mathbf{w}}, \vec{\mathbf{n}}) = (\vec{\mathbf{w}} \cdot \vec{\mathbf{n}})\vec{\mathbf{n}} = -(w n_y)\vec{\mathbf{n}} \tag{5.1}$$

since $w > 0$ and $n_y > 0$, $\vec{\mathbf{w}}$ has no contribution towards the positive side, and hence no force exertion on X. Similarly, it can be shown that for all weights of arbitrary small pieces of X, there exists a non-zero force contribution towards the negative side, i.e., `ACT(X,Y)`. □

5.2.3 Static Equilibrium Analysis

Having the `ACT` relations identified as described in the previous section, one might suggest that the support relations between objects, and consequently the safest candidates, can be identified through an analysis of the `ACT` relations. The hypothesis behind the analysis of the `ACT` relations is that for each act relation, `ACT(A,B)`, there must be one support relation such that `SUPP(B,A)`, i.e, if `A` acts on `B`, then `B` supports `A`. This can be translated to the intuitive heuristic rule that states removing the objects with the highest height is safe. However, there are many situations where this reasoning fails. In Figure 5.4, through examples, it

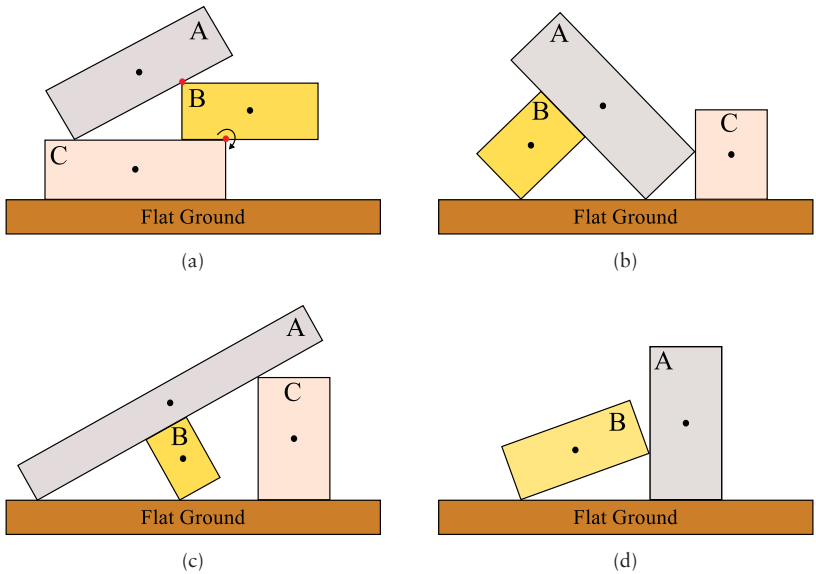


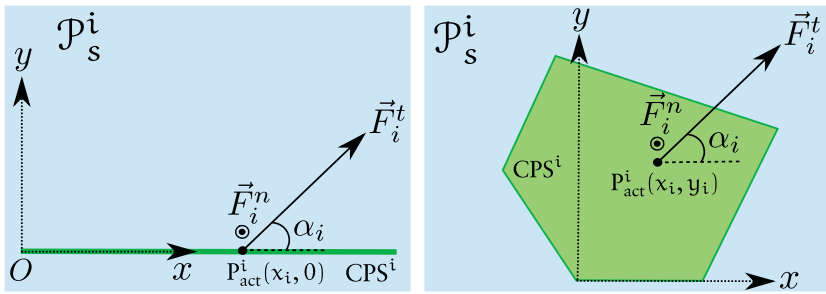
Figure 5.4: Four example configurations where reasoning about the support relations between objects solely based on the extracted ACT relations fail to predict the safest object to remove first.

is illustrated that for a set of configurations reasoning solely based on ACT relations or the height of objects fails to predict the true support relations between objects. In all the configurations shown in Figure 5.4, object A is the highest object in the configuration. However, A is not the safest object to be removed first (A supports B in all the configurations). Figure 5.4a and Figure 5.4b show two cases in which the highest object, A, supports another object, B, while B does not act on A (i.e., $\neg$ACT(B,A)). In Figure 5.4c, A acts on B (i.e., ACT(A,B)), but B does not support A (i.e., $\neg$SUPP(B,A)). In Figure 5.4d, the ACT relation between A and B cannot be identified (the separating plane is perpendicular to the flat ground); it can be clearly seen, however, that A supports B. To summarize, through the examples in Figure 5.4, three classes of configurations are illustrated that ACT relations analysis cannot be employed to reason about the safest object to remove: 1) configurations with bidirectional SUPP relations between two objects (see Figure 5.4a and Figure 5.4b); 2) configurations in which there is no bidirectional SUPP relation but the highest object is not the safest object to remove (see Figure 5.4c), and 3) configurations in which ACT relations between two objects cannot be identified (see Figure 5.4d).

At first glance, we may visually ascertain that it is impossible to find any safe sequence to remove objects from the configurations illustrated in Figure 5.4a and Figure 5.4b. However, the existence of a sequence of safe-to-remove objects depends on the number of end effectors that the robotic manipulator is equipped with. For a manipulation system with two end effectors, for example, a possible plan to solve the dead-lock in the configurations shown in Figure 5.4a and Figure 5.4b is to grasp and hold B with one of the end-effectors and remove A with the second end-effector.

An intuitive alternative possibility to the analysis suggested in this section would be the use of an existing physics simulator to determine the gravitational support relations. A physics simulator is a computer algorithm for solving dynamic equations of classical mechanics to predict the future motion states of a group of objects for a small interval of time; it performs the computations based on discretization of continuous real world quantities [70, 71]. The idea behind utilizing a physics simulator is to remove objects of a static configuration one at a time in simulation and then check whether what remains maintains a stable configuration. A discussion is given below about the reasons that limit the applicability of the physics simulation for identifying the gravitational support relations in real-world problems. First we notice that in addition to the geometrical attributes (shape and pose), a physics simulator needs the physical quantities (e.g., masses, friction factors, etc.) of all the objects in the scene to be precisely known, i.e., the accurate input description of the scene is necessary for physics simulation [72]. In typical robotic systems that the visual perception is the major source of gathering information about a scene, the physical quantities such as the friction coefficients and masses of objects cannot be measured, although the boundary (i.e., minimum and maximum) values for such quantities might be known. In case of drawing a set of values for the physical quantities, the result of the simulation is valid only for those specific values. Having this said, in order to use a physics simulator for identifying the support relations, one may propose to create a grid of possibilities for the uncertain values of the physical quantities and then perform physics simulation for each possible set of values. Such grid based approach has the following issues. First, it could be extremely time consuming due to the large search space of the grid; second, an effective sampling of the values is not trivial, and third, it is unclear how to deduce the existence of a support relation between two objects from all the outcomes of the simulations.

As a more promising solution this dissertation presents a method to perform static analysis of a configuration of objects to determine symbolic support relations under uncertain values of physical quantities such as friction coefficients. The method employs static equilibrium conditions to anticipate the effect of removing an object from a configuration. The problem statement and solution are formally presented as follows.



(a) P_{act}^i is at $(x_i, 0)$ (b) P_{act}^i is at (x_i, y_i)

Figure 5.5: Illustration of the point of action, P_{act}^i on the i-th separating plane, $\mathcal{P}_s^i$ in case of (a) a line-segment or (b) a polygon is the type of i-th contact between X and Y_i.

Problem 5.1. *Given the geometrical attributes of a target object* X *and a set of objects* $\mathcal{O} = \{Y_1, \ldots, Y_N, Z\}$ *in contact with* X*, determine whether* X *remains in static equilibrium by removing* Z *from* $\mathcal{O}$.

Solution. The solution is based on an analysis of static equilibrium conditions of X after Z is removed from $\mathcal{O}$ under uncertainty about the mass of the objects and their distribution, and unknown values of friction coefficients. From classical mechanics [73], an object is in static equilibrium if and only if the vector sum of all external forces is zero, the vector sum of all torques (due to the external forces) about any pivot point is zero, and the linear momentum of the object is zero. Since X and $\mathcal{O}$ are static, the values of their linear momentum are zero by definition. Formally, X is in static equilibrium if and only if,

$$\begin{aligned} \vec{F}_{total} &= \sum_{i=0}^{N} \vec{F}_i = 0 \\ \vec{\tau}_{total} &= \sum_{i=0}^{N} \vec{r}_i \times \vec{F}_i = 0 \end{aligned} \tag{5.2}$$

where, $\vec{F}_{total}$ is the vector sum of all external forces acting on X, $\vec{\tau}_{total}$ is the vector sum of all torques (due to the external forces) applied on X about the selected pivot point at centroid of X, N is the number of forces, $\vec{F}_i$ is the i-th force due to contact point between X and Y_i, $\vec{r}_i$ is the moment arm from the centroid of X to the point of action of $\vec{F}_i$, $i = 0$ refers to the weight of X.

In order to solve Eq. 5.2, it is a requirement that the mass of the objects and their distribution are known. However, the only source of gathering information about the environment, which is the visual perception cannot measure

the values of the physical quantities such as the mass of the objects. Moreover, in three-dimensional space where the objects are not mathematically idealized points, configurations of objects often represent a *statically indeterminate mechanical system* [74], i.e., a system of equations in which the number of unknowns (e.g. forces) is greater than the number of independent equations. The static equilibrium conditions in Eq. 5.2 are often insufficient to determine the unknown forces, even if the values of the physical quantities are known.

This section alternatively poses the static equilibrium analysis of a target object X as a problem of solving a system of nonlinear equality and inequality equations to overcome the issues mentioned above. The knowledge about the boundary values of the physical quantities are implemented into inequality constraints. For example, according to the Coulomb friction model [75] there is a non-linear inequality relation between the friction force, normal force and the friction coefficient. On the other hand, since we are interested in abstracting the symbolic support relations between objects, and the fact that the exact numeric computation of unknown forces is irrelevant for identifying support relations, it is adequate to find a set of consistent values of unknown variables to satisfy the system of equations. In other words, the goal is to find a feasible solution that satisfies a set of predefined constraints even if the configuration under study is statically indeterminate.

In order to construct the system of equations, we need to identify the unknown variables in the corresponding equations. Since the certain values of the mass of the objects as well as the friction coefficients between pairs of objects are not given, the boundary values add constraints to the system of equations,

$$\begin{aligned} &m_{O,\min} \leqslant m_O \leqslant m_{O,\max}\,, O \in \mathcal{O} \cup \{X\} \\ &0 < \mu_i \leqslant \mu_{\max} < \infty\,, i = 1, \ldots, N \end{aligned} \tag{5.3}$$

where, m_O is an unknown variable referring to the mass of the object O, $m_{O,\min}$ and $m_{O,\max}$ are the given boundary values of the mass of objects O, μ_i is an unknown variable referring to the friction coefficient between X and Y_i, and $\mu_{\max}$ is the given maximum value of the friction coefficients. The minimum value for a friction coefficient is zero by definition, while the maximum value can be set based on the maximum measured friction coefficient of the commonly-used materials in the target environment. In the real-world configurations of objects, where there is no glue, for example, between two objects in contact, the friction coefficients have to be bounded ($\mu_i < \infty$).

In the presence of friction, according to Coulomb friction model, resolving a force vector $\vec{F}_i$ into two components, $\vec{F}_i^n$ and $\vec{F}_i^t$ representing respectively normal and tangential (friction) forces, the following inequality must hold,

$$\|\vec{F}_i^t\| \leqslant \mu_i \|\vec{F}_i^n\|\,, i = 1, \ldots, N. \tag{5.4}$$

Computing CPS(X, Y_i) and the separating planes $\mathcal{P}_s^i$ between X and Y_i, ($\forall\, 1 \leqslant i \leqslant N$) using Algorithm 5.1 discussed in Section 5.2.1, there exist three possi-

bilities of contact point-sets, namely, a single-point, a line-segment or a polygon. If a line-segment, $\mathcal{L}$ (see Figure 5.5a), or a polygon, $\mathcal{G}$ (see Figure 5.5b) is $\mathrm{CPS}(X, Y_i)$, according to superposition theorem a point in $\mathcal{L}$, respectively in $\mathcal{G}$, can summarize the effect of all the other points in $\mathcal{L}$, respectively in $\mathcal{G}$, which is called point of action, P^i_{act} (see Figure 5.5). Using the superposition theorem, P^i_{act} can be identified by searching in the corresponding contact point-set. In case $\mathrm{CPS}(X, Y_i)$ is a line-segment, one unknown variable, x_i, and in case $\mathrm{CPS}(X, Y_i)$ is a polygon, two unknown variables, x_i and y_i representing the position of P^i_{act} are added to the system of equations. The point of action is required to identify the momentum arm of the corresponding torque.

The direction of the normal and tangential forces acting at the point of action need to be also modeled. An angle, α_i ($0 \leqslant \alpha_i < 2\pi$) with respect to a chosen fixed axes in $\mathcal{P}^i_s$ (see Figure 5.5) is an unknown variable of the system that parameterizes the direction of the friction force $\vec{F}^t_i$. The magnitude of $\vec{F}^t_i$ is another unknown variable of the system, together with α_i identify the i-th friction force. The direction of the normal force, $\vec{F}^n_i$ is along one of two possible directions of the normal vector of $\mathcal{P}^i_s$, and is determined by the `ACT` relation between X and Y_i as follows. For each $Y_i \in \mathcal{O}$, depending on the `ACT` relation three possibilities can be considered,

1. X acts on Y_i. X exerts force on Y_i due to gravity, thus the direction of the reaction force from Y_i, that is, the normal vector of $\mathcal{P}^i_s$ with positive y-component is the direction of $\vec{F}^n_i$. In this case four unknown variables $\|\vec{F}^n_i\|$, $\|\vec{F}^t_i\|$, α_i, and μ_i are defined.

2. Y_i acts on X. Y_i exerts force on X due to gravity, thus the normal vector of $\mathcal{P}^i_s$ with negative y-component is the direction of $\vec{F}^n_i$. In this case four unknown variables $\|\vec{F}^n_i\|$, $\|\vec{F}^t_i\|$, α_i, and μ_i are defined.

3. The `ACT` relation between X and Y_i is not identified. In this case the friction coefficient between X and Y_i is ignored since we assume that there must be a third object, $Y_j (j \neq i)$, in contact with X and Y_i to cancel their weights for some certain friction coefficient. Thus, only the magnitude of the normal force $\|\vec{F}^n_i\|$ that Y_i may exert on X is defined as an unknown variable.

At this point we can construct the system of equations based on the discussion in the preceding paragraphs and Eq. 5.2, Eq. 5.3 and Eq. 5.4,

$$\begin{cases} F_x = 0, F_y = 0, F_z = 0 \\ \tau_x = 0, \tau_y = 0, \tau_z = 0 \\ P^i_{act} \in \mathrm{CPS}(X, Y_i) \\ 0 \leqslant \alpha_i \leqslant 2\pi \\ 0 < \mu_i \leqslant \mu_{max} \\ \|\vec{F}^t_i\| \leqslant \mu_i \|\vec{F}^n_i\| \end{cases} \tag{5.5}$$

Algorithm 5.2: The Extraction of the SUPP Relation

Data: Geometrical attributes of $\mathcal{O} = \{X = Y_0, Y_1, \ldots, Y_N, Z\}$
Result: Truth of $\mathrm{SUPP}(Z, X)$

VarSet $\leftarrow \emptyset$ /* the set of unknown variables */;
VarSet.Add(m_0) /* the mass of X */;
ICSet $\leftarrow \emptyset$ /* the set of inequality constraints */;
ICSet.Add($m_{0,\min} \leqslant m_0 \leqslant m_{0,\max}$);
$\vec{F}_{total} \leftarrow -m_0 g \hat{y}$ /* g: gravity constant, $\hat{y}$: unit vector of y-axis */;
$\vec{\tau}_{total} \leftarrow \vec{0}$;
for $i = 1, \ldots, N$ **do**
    /* Compute CPS(X, Y_i) and $\mathcal{P}_s^i$ using Algorithm 5.1 */
    /* Identify ACT relation between X and Y_i using Proposition 5.1 */
    /* Identify P_{act}^i based on CPS(X, Y_i), see Figures 5.5 */
    if ACT(X, Y_i) *or* ACT(Y_i, X) *holds* **then**
        VarSet.Add($\|\vec{F}_i^n\|, \|\vec{F}_i^t\|, \alpha_i, \mu_i, m_i$);
        ICSet.Add($0 \leqslant \alpha_i \leqslant 2\pi$);
        ICSet.Add($0 \leqslant \mu_i \leqslant \mu_{max}$);
        ICSet.Add($\|\vec{F}_i^t\| \leqslant \mu_i \|\vec{F}_i^n\|$);
        ICSet.Add($m_{i,\min} \leqslant m_i \leqslant m_{i,\max}$);
        if ACT(X, Y_i) *holds* **then**
            $\vec{F}_i = \|\vec{F}_i^n\|$NormalVectorIn($P_s^i$, X) $+ \vec{F}_i^t$;
        else
            $\vec{F}_i = \|\vec{F}_i^n\|$NormalVectorIn($P_s^i$, Y_i) $+ \vec{F}_i^t$;
        end
    else
        VarSet.Add($\|\vec{F}_i^n\|, m_i$);
        ICSet.Add($m_{i,\min} \leqslant m_i \leqslant m_{i,\max}$);
        $\vec{F}_i = \|\vec{F}_i^n\|$NormalVectorIn($P_s^i$, X);
    end
    $\vec{F}_{total} = \vec{F}_{total} + \vec{F}_i$;
    $\vec{\tau}_{total} = \vec{\tau}_{total} + (P_{act}^i - C_X) \times \vec{F}_i$ /* C_X is the center of mass of X */;
end
/* Solve the system of equations in Eq. 5.5 */
if *there is no solution* **then return** *true* **else return** *false*;

where F_x, F_y and F_z are x, y, and z components of $\vec{F}_{total}$, τ_x, τ_y and τ_z are x, y, and z components of $\vec{\tau}_{total}$ respectively. Solving the non-linear system of equalities and inequalities in Eq. 5.5 is the basis of concluding whether the static equilibrium conditions of object X are met after Z is removed from $\mathcal{O}$.

Depending on the existence of a solution, that satisfies all the equations, two possibilities are discussed. First, if there is no solution for the problem, then it is impossible for X to preserve its static equilibrium state, which means that the removed object Z supports object X, i.e., we can state that $\texttt{SUPP}(Z, X)$ holds. Second, if the exist at least one solution, it implies that there exists one possible set of forces and friction coefficients that can satisfy the static equilibrium conditions of the target object X under predefined constraints. The implication, however, is valid as long as the assumptions on the mass distributions are close to reality. In this case, we assume that the removed object, Z, does not support X, i.e., $\neg\texttt{SUPP}(Z, X)$ holds. Algorithm 5.2 shows the procedure of identifying the truth of $\texttt{SUPP}(Z, X)$. It should be noted that depending on the type of $\text{CPS}(X, Y_i)$, as discussed above, maximum two unknown variables will be added to the system to represent the point of action at line 25 of Algorithm 5.2. □

5.3 Extracting Support Relations - ICSO case

In the absence of a complete object detection, where some of the objects composing a pile are not detected, the approach described in Section 5.2 is not applicable to extract support relations between objects. The effects of the undetected objects on the statics of the pile cannot be neglected. For example, in the ICSO case shown in Figure 5.1, due to the undetected objects (i.e., 1,4,5 and 6) identifying all the contact points and $\texttt{ACT}$ relations is not feasible. Thus, the geometrical reasoning and static equilibrium analysis described in the previous section cannot be used to deduce the support relations even if the detected objects are in contact with each other.

This section alternatively presents a probabilistic approach to extracting the support relations in order to deal with the lack of information and uncertainty. The available data in the ICSO case are the geometrical attributes of the detected objects plus a point cloud of the scene. It should be noted that the set of detected objects can be either in contact or far from each other. The possibility that a detected object, through some undetected objects supports another object implies that the support relation may indirectly exist between a pair of separated objects (see an example of incomplete detection of objects in Figure 5.1).

In the presence of the lack of information, it is of great importance to provide an uncertainty measure of the deduced support relations. Since the underlying probability distribution of the support relations between objects in an arbitrary configuration is not known, and it cannot be approximated by a standard distribution in advance, different machine learning techniques are employed to approximate the probabilities. A set of classifiers are trained to estimate the probability of one object X supporting another object Y given features extracted from the scene point cloud and the relative position between pairs of the detected objects.

In this dissertation the performance of three carefully selected learning paradigms, namely, Support Vector Machines (SVM) [76], Artificial Neural Networks (ANN) [77] and Random Forests (RFT) [78] to estimate support relation probabilities are examined. Since ordinary versions of these classifiers are employed, for details regarding these standard machine learning techniques please see the given references. The only extension being presented in this section is how the probabilities of the class labels (i.e., support relations) are computed. The probabilities of support relations are then used to create possible worlds models based on which a probabilistic decision making reasons about the set of safe-to-remove objects.

5.3.1 Class Probability Estimation

Support Vector Machine SVM in its original formulation can predict only class labels, $l \in \{-1, +1\}$, given the input features, $\mathcal{F}$, and a trained model, $f(.)$; class label probabilities, $P(l|\mathcal{F})$, are not directly computed. A training set, $\mathcal{T}$, includes instances of features and their known class labels, which we call them features-labels. We use a sigmoid function to estimate the posterior probability of the predicted class labels proposed by Platt [79],

$$P(l = 1|\mathcal{F}) = \frac{1}{1 + \exp(Af(\mathcal{F}) + B)} \tag{5.6}$$

where A and B are estimated by minimizing the following negative log likelihood,

$$\begin{aligned} \underset{A,B}{\text{minimize}} \quad & -\sum_{i=1}^{u} (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \\ \text{where} \quad & p_i = P(l_i = 1|\mathcal{F}_i) \\ & t_i = \begin{cases} \frac{N^+ + 1}{N^+ + 2}, & \text{if } l_i = +1 \\ \frac{1}{N^- + 2}, & \text{if } l_i = -1 \end{cases} \\ & N^+ \text{ is the number of } l_i = +1 \text{ instances in } \mathcal{T} \\ & N^- \text{ is the number of } l_i = -1 \text{ instances in } \mathcal{T} \end{aligned}$$

Artificial Neural Networks In order to predict the class labels we use a multilayer perceptron [77] with two real valued outputs (for binary classification), a hyperbolic tangent sigmoid transfer function for hidden-layers and a logarithmic sigmoid transfer function, *logsig*, for the output layer. To estimate the probability of a predicted class label, l, given the corresponding feature vector, $\mathcal{F}$, we first normalize the two output values of *logsig*, y_o^+ and y_o^-, to sum to 1,

$$\beta(y_o^+ + y_o^-) = 1 \Rightarrow \beta = \frac{1}{y_o^+ + y_o^-} \tag{5.7}$$

and finally compute the class label probability as,

$$P(l = 1|\mathcal{F}) = 1 - \beta y_o^- \tag{5.8}$$

Random Forests For a random forest with N decision trees [80], the probability estimation of the predicted class labels, l, given the input features, $\mathcal{F}$, by i-th decision tree, $P_i(l|\mathcal{F})$, is computed as the fraction of training data sets of the class label in the corresponding tree leaf. The posterior probability of a predicted class label by a random forest is then computed as the average of $P_i(l|\mathcal{F})$'s,

$$P(l|\mathcal{F}) = \frac{1}{N}\sum_{i=1}^{N} P_i(l|\mathcal{F}) \tag{5.9}$$

5.3.2 Features Extraction

This section explains the elements of the features vector and the methods to extract each feature element from the point cloud, $\mathbf{P} = \{p_1, \ldots, p_N\}, p_i \in \mathbb{R}^3$, of the given configuration, $\mathcal{C}$, and the geometrical attributes of the detected objects in $\mathcal{C}$. The first step in feature selection is to include all features that could possibly carry information about the target class labels, i.e., the support relations. The point cloud features are included to capture the distribution of the sampled points around an object which may carry information about the existence of other undetected objects around it. A set of geometrical features (e.g., volumes, bounding boxes, heights, Euclidean distances, intersection of areas projected on surfaces, differences of heights, etc.) as well as the point cloud features (e.g., distances of the points in $\mathbf{P}$ to the centroid and vertices of the objects) are extracted to have a pool of features. The features described below are the result of applying mutual information analysis for feature selection [81] that eliminated redundant and irrelevant features.

Point Cloud Features

In the absence of having access to the complete set of objects in $\mathcal{C}$, the hypothesis is to use the set of sampled points of $\mathcal{C}$ to extract features that may improve the probability estimation of the support relations. The mutual information analysis revealed that to some extent the distribution of $\mathbf{P}$ around an object, X, carries information about the undetected objects nearby X. A possible type of feature that captures the distribution of $\mathbf{P}$ with respect to a target object is *distance-based activation function* (DBAF). DBAF is defined as the normalized sum of Gaussian functions of squared Euclidean distances between points in $\mathbf{P}$, and a point of interest, cp in $\mathbb{R}^3$,

$$f(cp) = \frac{1}{N}\sum_{k=1}^{N} \frac{1}{\sqrt{(2\pi\sigma)^3}} \exp\left(-\frac{\|cp - p_k\|^2}{2\sigma^2}\right),\ p_k \in \mathbf{P} \tag{5.10}$$

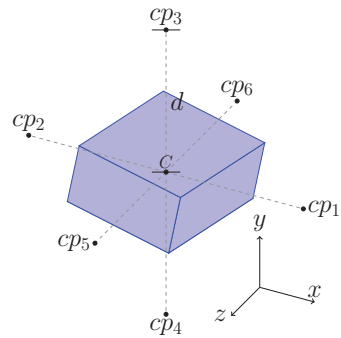


Figure 5.6: Interest points $cp_i, i = 1, \ldots, 6$ for a cuboid object with centroid at C.

where, $f(cp)$ is the DBAF of the point of interest, cp, and σ is a parameter to weight the significance of closer points in $\mathbf{P}$ to cp. The DBAF function balances the contribution of farther points (which represent larger distances) and of closer points (which represent smaller distances).

For each detected object $X \in \mathcal{C}$, the centroid of X is used to define six distinct points of interest by translating the components of the centroid, (x_c, y_c, z_c), $\pm d$ units along each axis of the reference frame (see Figure 5.6),

$$CP = \begin{array}{c} \begin{matrix} cp_1 & cp_2 & cp_3 & cp_4 & cp_5 & cp_6 \end{matrix} \\ \begin{bmatrix} x_c - d & x_c + d & x_c & x_c & x_c & x_c \\ y_c & y_c & y_c - d & y_c + d & y_c & y_c \\ z_c & z_c & z_c & z_c & z_c - d & z_c + d \end{bmatrix} \end{array} \tag{5.11}$$

where, each column of CP is a point of interest for object X. A complete DBAF feature vector for X is formally expressed as below,

$$F_{DBAF}(X) = [f(cp_1), \ldots, f(cp_6)]^T \tag{5.12}$$

Pairs of Objects Features

In order to capture the relative configuration between two objects, X and Y, the difference between their axis-aligned bounding boxes as well as the smallest distance, $d_s(.)$, of the centroid of X and Y to the flat ground are extracted. The axis-aligned bounding box of an object, X, is denoted as,

$$BB_X = [x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}]^T \tag{5.13}$$

where, *min* and *max* subscripts respectively denote the minimum and maximum of x, y and z components of points in X. For the support relation $\texttt{SUPP}(X, Y)$, the corresponding feature vector is defined to be the following difference,

$$F_{BB}(X, Y) = BB_X - BB_Y \tag{5.14}$$

And for the smallest distances, the following feature vector is defined,

$$F_H(X, Y) = [d_s(X), d_s(Y)]^T \tag{5.15}$$

Complete Features Vector

For each pair of detected objects, X and Y in $\mathcal{C}$, two feature vectors are extracted by combining F_H, F_{BB} and F_{DBAF}. The features vector from X's point of view (whether X supports Y) is,

$$\mathcal{F}(X, Y) = [F_H(X, Y), F_{BB}(X, Y), F_{DBAF}(X), F_{DBAF}(Y)]^T \tag{5.16}$$

while from Y's point of view is,

$$\mathcal{F}(Y, X) = [F_H(Y, X), F_{BB}(Y, X), F_{DBAF}(Y), F_{DBAF}(X)]^T. \tag{5.17}$$

A machine learning paradigm uses the features vector $\mathcal{F}(X, Y)$ to output the probability of $\texttt{SUPP}(X, Y)$ being true.

5.3.3 Possible Worlds of Support Relations

A representative model for the probabilistic hypotheses about the support relations is especially important in the ICSO case, where there exists a set of possibilities to infer which object supports another. In order to encode hypotheses about objects supporting each other, the concept of *possible worlds* from modal logic is employed.

We define a possible world to be one realization of support relations between all pairs of detected objects. Formally, let each support relation $\texttt{SUPP}(X, Y)$ between two different objects, X and Y be modeled by a binary random variable, S_k such that $S_k = 1$ if $\texttt{SUPP}(X, Y)$ is true, and $S_k = 0$ if $\texttt{SUPP}(X, Y)$ is false. Let $\Omega = [S_1, S_2, \ldots, S_\eta]$ be a random vector composed of all the binary random variables. For N detected objects, the number of support relations (i.e., the number of the binary random variables), η is,

$$\eta = 2\binom{N}{2} = N(N-1) \tag{5.18}$$

where $\binom{N}{2}$ is a 2-combination of N objects. A possible world is one possible assignment $\omega = [s_1, s_2, \ldots, s_\eta]$ to Ω, where $s_k \in \{0, 1\}, k = 1, \ldots, \eta$. In other words, one possible world is equivalent to one hypothesis about the ground

N	$q = 2^{N(N-1)}$	q'	q'/q (%)
3	64	29	45.3
4	4096	355	8.67
5	1048576	6942	0.66
6	1073741824	209527	0.02

Table 5.2: A comparison of the number of consistent worlds, q', and the number of all possible worlds, q, for different numbers of objects $n = 3, 4, 5, 6$.

truth of how objects are supporting each other. The number of all possible assignments of variables s_k to Ω is $q = 2^\eta$.

The possible worlds of the support relations are illustrated as a graph with nodes and directed links representing the objects and the support relations respectively. In the graph, a directed link from node X to Y denotes $\texttt{SUPP}(X, Y)$ with the corresponding binary random variable S_k. For example, Figure 5.7a shows the graph of possible worlds for $N = 4$ objects, where a total number of $\eta = 12$ directed links (i.e., support relations) represent the set of the binary random variables in this case.

It should be noted that the probability of a support relation between two objects is estimated independently of another pair of objects, however, such probability may not be independent given the support relation of another pair of objects. Since the underlying conditional probability of support relations is not known, the joint probability distribution of the random vector Ω is approximated as,

$$\begin{aligned} P(\Omega = \omega_i) &= P(S_1 = s_{i,1}, \ldots, S_\eta = s_{i,\eta}) \\ &= \prod_{k=1}^{\eta} P(S_k = s_{i,k}) \end{aligned} \tag{5.19}$$

where, $\omega_i = [s_{i,1}, \ldots, s_{i,\eta}]$ is the i-th possible assignment to Ω, and $P(S_k)$ is the estimated probability of S_k by a machine learning paradigm.

Consistent Possible Worlds

Since the support relation is transitive, i.e.,

$$\forall X, Y \in \mathcal{C},\ \texttt{SUPP}(X, Y) \wedge \texttt{SUPP}(Y, Z) \Rightarrow \texttt{SUPP}(X, Z) \tag{5.20}$$

it is required to make sure that a realization of a possible world, ω, with an assignment of variables is consistent with the transitivity property. For example, Fig 5.7b and Fig 5.7c depict graph illustrations of one consistent and one inconsistent possible worlds for four objects respectively. In Fig 5.7c, the in-

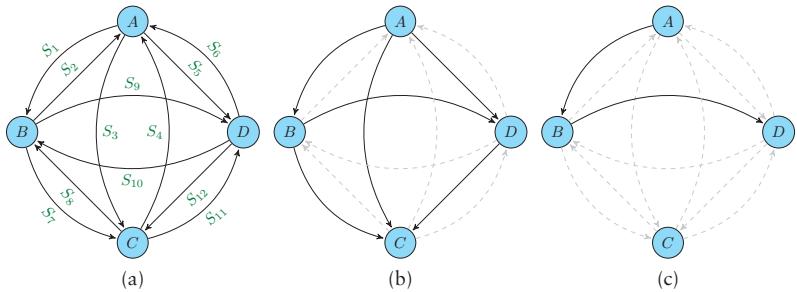


Figure 5.7: Graph illustration of the random vector Ω for four objects, where each edge k is labeled with a binary random variable S_k. Solid and dashed edges denote $S_k = 1$ and $S_k = 0$ respectively. (a) A possible world where all $S_k = 1$. (b) A possible world where some $S_k = 0$. This is a consistent world according to the transitivity constraint. (c) An inconsistent possible world according to the transitivity constraint.

consistency is due to the fact that SUPP(A, B) and SUPP(B, D) are both true, but SUPP(A, D) is false.

In order to eliminate such inconsistent worlds, one solution is to employ the Path Consistency Algorithm [82]. Table 5.2 shows the number of consistent worlds, q', in comparison with the number of all possibilities, q, for different numbers of objects, $N = 3, 4, 5, 6$. It can be observed that discarding inconsistent worlds significantly reduces the size of the representation.

Elimination of the inconsistent worlds implies that the sum of joint probabilities of the consistent worlds in Eq. 5.19 becomes less than one. Thus, to represent a true probability distribution over consistent possible worlds, the corresponding probabilities must be normalized. To distinguish is from all possibilities, ω^c denotes an assignment to a consistent possible world, and $i(.)$ maps index set of the consistent possible worlds into the original set of possible worlds. Introducing a constant normalizing factor β, such that the probability of the j-th consistent world, $P(w_j^c)$, where $w_j^c = [s_{i(j),1}, \ldots, s_{i(j),\eta}]$ becomes

$$P(w_j^c) = \beta P(s_{i(j),1}, \ldots, s_{i(j),\eta}) \tag{5.21}$$

and, the sum of probabilities becomes one,

$$\sum_{j=1}^{q'} P(w_j^c) = 1. \tag{5.22}$$

We note that if p_1, p_2 and p_3 are the estimated probabilities of SUPP(X,Y), SUPP(Y,Z) and SUPP(X,Z) respectively, then it is not necessary to have $p_3 =$

	$P(w_j^c)$	A_1	...	A_n
w_1^c	$P(w_1^c)$	c_{11}	...	c_{1n}
⋮	⋮	⋮	⋱	⋮
$w_{q'}^c$	$P(w_{q'}^c)$	$c_{q'1}$	...	$c_{q'n}$

Table 5.3: Payoff matrix with actions, A_k, consistent possible worlds, w_j^c, their joint probabilities, $P(w_j^c)$, and the costs of taking actions, c_{jk}.

$p_1 p_2$. In fact, the underlying structure of the joint probabilities is unknown, and a machine learning paradigm computes the probability of the support relation between each pair of objects independently of the other objects.

5.4 Decision Making

This section describes a probabilistic decision making to reason about the set of safe-to-remove objects as candidates to be unloaded from a pile given the corresponding representation of the extracted support relations. The decision making approach selects a safe-to-remove object based on minimizing the risk of a change in the pose of the other objects in the pile. The approach can be applied to the both cases – ICSO and CSO, where the CSO case is considered as a special case in which all support relations are known.

The probabilistic decision making approach employs the expected utility principle [83] from decision theory where the *minimization of expected cost* is adopted in order to make an optimal decision. To do this, a *payoff matrix*, with elements as the costs of taking possible actions (i.e., unloading an object) in each consistent world is created. Table 5.3 shows the payoff matrix structure. The first and second columns contain the possible assignments and corresponding joint probabilities of each consistent world respectively. In Section 5.3, the different steps to build a probabilistic world model of support relations between pairs of detected objects in a given configuration $\mathcal{C}$ are outlined. The elements of the other columns represent the cost, c_{jk} of taking actions A_k (i.e. selecting an object in $\mathcal{C}$) in j-th consistent possible world. In other words, an element c_{jk} is the cost of removing the k-th object from $\mathcal{C}$ given the j-th consistent possible world.

Removing the k-th object, $X_k \in \mathcal{C}$, from the j-th consistent possible world is penalized by counting the number of the objects that X_k supports, i.e., the cost c_{jk} is computed as,

$$c_{jk} = \sum_{S_k = \texttt{SUPP}(X_k, Y)} S_k \quad , \quad Y \in \mathcal{C} - \{X_k\} \tag{5.23}$$

For example, in Figure 5.7b, the costs of removing A, B, C and D are 3, 2, 0 and 1 respectively.

The optimal action, A^* (i.e., safest object to remove from $\mathcal{C}$) is the one with the minimum expected cost (EC),

$$A^* = \underset{k}{\operatorname{argmin}}\ \mathrm{EC}(A_k) \tag{5.24}$$

where $\mathrm{EC}(A_k)$ is defined as,

$$\mathrm{EC}(A_k) = \sum_{j=1}^{q'} P(\omega_j^c) c_{jk} \tag{5.25}$$

In the CSO case it is assumed that there exists only one consistent possible world (with a joint probability of 1) represented by the extracted support relations between objects as described in Section 5.2.

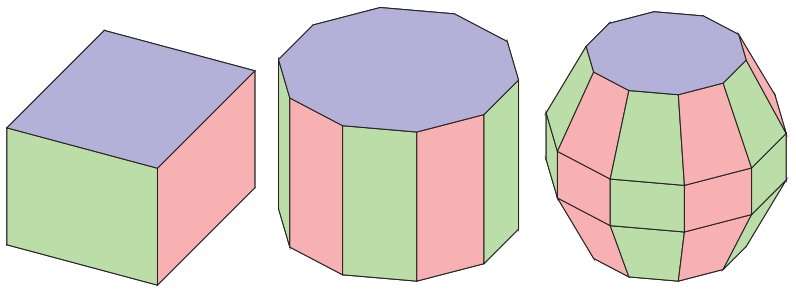
It should be mentioned that scaling is a problem of representations based on possible worlds concept, where the number of possibilities grows exponentially as the number of objects increases. If there are too many objects detected, and N is the maximum number of objects that computationally can be handled in practice, the following heuristic solution can be applied. First, we consider only N objects which have the highest height with respect to the flat ground, and then we create the consistent possible worlds for those N objects. The probabilistic decision maker is then employed to find the best candidate objects to unload first. The candidate will be unloaded from the configuration, and we repeat this procedure until all the objects are unloaded.

5.5 Results

This section presents the experimental results of applying the methodology described for the CSO and ICSO cases in the preceding sections. The experiments were carried out on data generated in simulated and from real-world configurations. Simulation facilitates generating a large number of random configurations with direct access to the ground truth data of the geometrical attributes of the objects. A large number of random configurations is important in analyzing the statistical behavior of the corresponding approach, while the ground truth data is required for the learning of the probability distribution of support relations. The performance of the corresponding approach was then evaluated on data generated on real-world configurations. A mock-up container was used to create piles of objects aiming at validating the constructed representation and examining the probabilistic decision making approach.

5.5.1 Simulated Configurations

For the simulated configurations, a scene generator based on physics simulation was developed. The simulator generates random configurations of polyhe-



(a) A Cuboid (b) A Cylinder (c) A Barrel

Figure 5.8: Polyhedron shapes representing (a) a carton box as a cuboid, (b) a cylinder, and (c) a barrel.

dron shaped objects inside a simulated container. The boundary values of the physical quantities such as the mass of the objects and friction coefficients, in addition to the collision shape descriptions of the objects are set as minimum-maximum intervals. The geometrical attributes and physical quantities of the generated objects are uniformly sampled from the given intervals. A simulated 3D range sensor scans the entrance of the container and produces a set of sampled points $\mathbf{P}$ of the scene.

Test configurations generated in simulation contained three types of objects, namely, carton boxes (CBX), cylinders (CYL) and barrels (BRL). A circle in a shape is approximated by a convex polygon with 36 equal length edges. A cuboid represents the shape of a carton box. A cylinder with the approximated circles represents a cylinder object. Two semi-cones with the approximated circles construct the shape of a barrel (see Figure 5.8).

The generated configurations in simulation are divided into four categories. Configurations made of only carton boxes, $\mathcal{C}^{\text{CBX}}$ (see Figure 5.9a), of only cylinders, $\mathcal{C}^{\text{CYL}}$ (see Figure 5.9b), of only barrels, $\mathcal{C}^{\text{BRL}}$ (see Figure 5.9c), and of a mix of the three objects, $\mathcal{C}^{\text{MIX}}$ in which the frequency of the three object types are equally likely distributed (see Figure 5.9d). For each object category, $\mathcal{U} = \{\text{CBX}, \text{CYL}, \text{BRL}, \text{MIX}\}$, a total of 40 configurations consisting of $n \in \mathcal{N}$ objects were generated, where

$$\mathcal{N} = \{n : n = 10r, 1 \leqslant r \leqslant 10, r \in \mathbb{N}\} \tag{5.26}$$

The total number of configurations generated in simulation is 1600, which is the result of multiplying 4 categories of objects, 40 configurations per category, and 10 different sets of values uniformly drawn from the dimensions of the shapes of the objects per configuration. The notion $\mathcal{C}^{u}_{u,i}$ indicates the i-th configuration consisting of $n \in \mathcal{N}$ objects of type $u \in \mathcal{U}$, where $i = 1, \ldots, 40$. The

(a) Cuboid Shaped Objects (b) Cylinder Shaped Objects

(c) Barrel Shaped Objects (d) Mixed objects

Figure 5.9: A few samples of static configurations generated in simulation: (a) cuboid shapes; (b) cylindric shapes; (c) barrel shapes, and (d) a mix of cuboid, cylindric and barrel shape objects.

set of configurations defined above were used as the input data for the method of extracting support relations in the CSO case (see Section 5.2).

In the ICSO case, the geometrical attributes of only a subset of $n \in \mathcal{N}$ objects in $\mathcal{C}_n^u$ are available. In order to simulate incomplete object detection, in which only a subset of objects are detected, we drew randomly a set of objects from the visible layer of a simulated configuration. A subset of $m \in \mathbb{N}$ objects of a simulated configuration, $\mathcal{C}_n^u$, is an incomplete set of objects denoted by $\mathcal{I}_m^u$, where $m \leqslant n, n \in \mathcal{N}$ and $u \in \mathcal{U}$.

To generate training sets in the ICSO case, for each two objects X and Y in $\mathcal{I}_m^u$, both features vectors $\mathcal{F}(X,Y)$ and $\mathcal{F}(Y,X)$ were generated (see Section 5.3.2) with their true support relations (i.e., the ground truth of the class labels) automatically extracted from the corresponding $\mathcal{C}_n^u$ by the CSO method explained in Section 5.2. The experiment was conducted on a total of 34706, 48422, 93312 and 56886 feature-labels (training sets) extracted from $\mathcal{C}_n^{CBX}$, $\mathcal{C}_n^{CYL}$, $\mathcal{C}_n^{BRL}$ and $\mathcal{C}_n^{MIX}$ respectively.

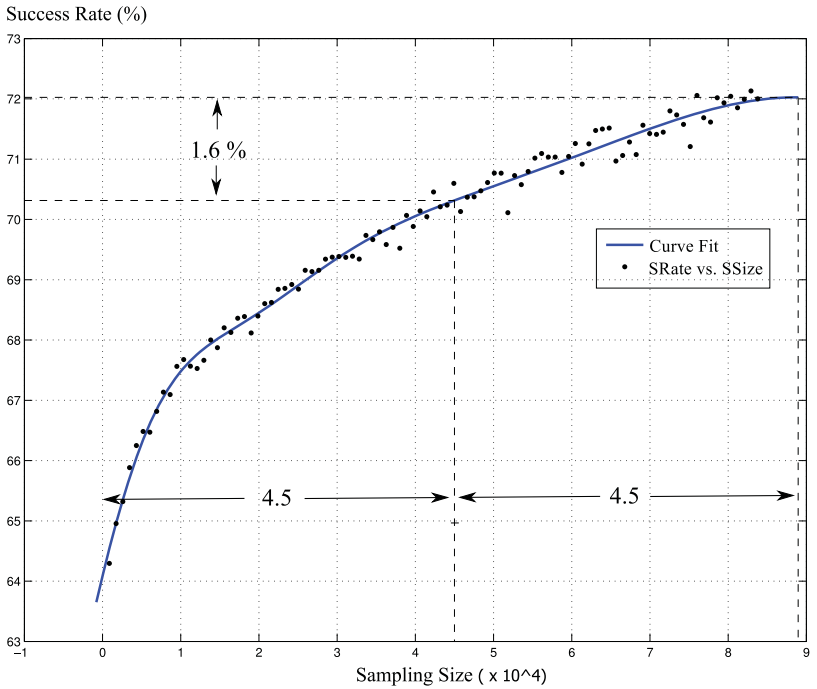


Figure 5.10: The vertical axis is the success rate (SRate) of the random forest classifier, and the horizontal axis is the sampling size (SSize) of pairs of objects used to train the classifier. The black points are the 5-fold classification success rate at each sample size. In order to interpolate values, a curve (blue line) is fit to the points. It can be seen that the success rate is converging as the size of the samples increases.

In order to justify the amount of samples used in the training sets, it will be empirically shown that if the number of samples exceeds some threshold, then a significant increase in the number of the samples (e.g., doubling the size) has a minor contribution to the performance of the classification. Figure 5.10 depicts the success rate of the random forest classifier in terms of the 5-fold classification success rate with respect to the number of samples (i.e., feature-labels extracted for pairs of objects) from $\mathcal{C}^{\text{MIX}}$ configurations. The success rate for a classifier is defined as the percentage of the correctly predicted support relations between the detected objects. As it can be seen from Figure 5.10, by doubling the number of samples from 45000 to 90000, the success rate only increases about 1.6%. And a very similar curve to Figure 5.10 for SVM and ANN classifiers is observed. This behavior is due to the fact that the extracted

features from an incomplete set of detected objects can only contain partial information about the target classes (i.e., binary support relations); thus, it is to be expected that the classification success rate is bounded at a percentage level below 100% even for a very large size of the training set. On the other hand, overfitting can occur for very large numbers of samples. Since there is no additional source of information about a configuration of objects in the ICSO case, a success rate of 70% is considered to be a good performance for this classification problem. It is worth mentioning that in the proposed probabilistic approach the predicted output labels are not directly used, the probability of the predicted labels are actually employed (see Section 5.3.1).

To generate test sets in the ICSO case, a separate set of $M = 500$ simulated configurations was generated similar to that of the training sets. We created incomplete sets of $m = 5$ objects, which are supposed to be detected, from the separately generated configurations. For each $\mathcal{I}^{u}_{5,j}$, where $j = 1, \ldots, M$ all possible pairs of feature-labels, $\eta = 20$ (see Eq. 5.18 in Section 5.3.3), were extracted as the corresponding test set (TSet), which is denoted by T^{u}_{j}.

5.5.2 Real World Configurations

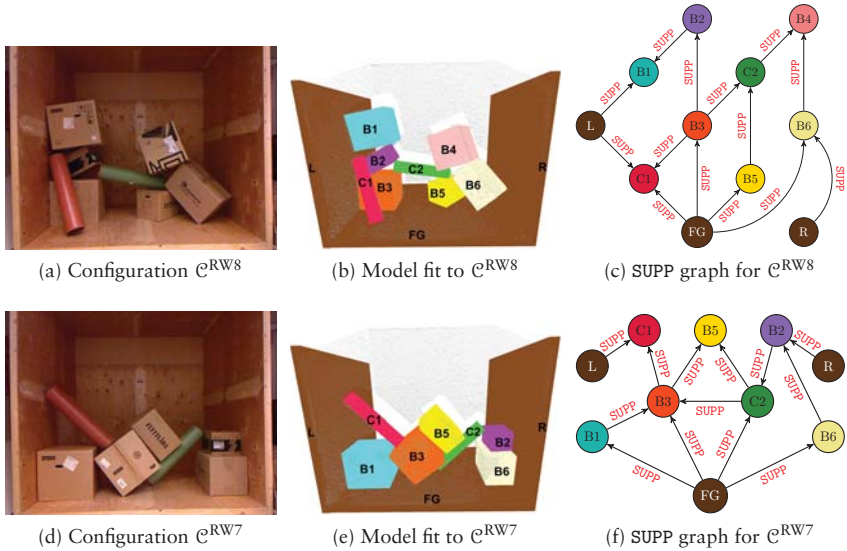
In order to validate the process of identifying the set of safe-to-remove objects on real-world data, we used a Microsoft Kinect sensor to scan two setups of real-world configurations of objects stacked inside a mock-up container, which are denoted $\mathcal{C}^{RW7}$ and $\mathcal{C}^{RW8}$ (see Figure 5.11a and Figure 5.11d). The Kinect sensor placed in front of the middle entrance of the mock-up container was used to capture a point cloud of a configuration. The complete set of detected objects (i.e., the CSO case) was then created by registering the 3D models of the objects to the point clouds, and then the poses of the objects were refined manually (see Figure 5.11b and Figure 5.11e).

For generating the test sets in the ICSO case, a number of subsets of objects, $\mathcal{I}^{RW7}_{m,j}$ and $\mathcal{I}^{RW8}_{m,j}$ were drawn randomly from the real-world configurations, $\mathcal{C}^{RW7}$ and $\mathcal{C}^{RW8}$, where, $m = 3, 4, 5, 6$ is the number of objects in the incomplete set and j is the index of the set. The number of all possible ways of choosing m objects from n objects is

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \tag{5.27}$$

thus, the index set for $\mathcal{I}^{RWn}_{m,i}$ is $i = \{1, \ldots, \binom{n}{m}\}$, where, $n = 7, 8$ is the number of all the objects in real-world configurations, $\mathcal{C}^{RW7}$ and $\mathcal{C}^{RW8}$ respectively.

In order to train the selected machine learning paradigms for estimating the probability of support relations, a training set with similar objects to that of simulated configurations is required to be generated. Physics simulation was employed to generate a set of random configurations, $\mathcal{C}_{RAND}$, consisting of objects with similar shapes of the objects used in the real-world configurations,



(a) Configuration $\mathcal{C}^{\mathrm{RW8}}$ (b) Model fit to $\mathcal{C}^{\mathrm{RW8}}$ (c) SUPP graph for $\mathcal{C}^{\mathrm{RW8}}$

(d) Configuration $\mathcal{C}^{\mathrm{RW7}}$ (e) Model fit to $\mathcal{C}^{\mathrm{RW7}}$ (f) SUPP graph for $\mathcal{C}^{\mathrm{RW7}}$

Figure 5.11: Real world configurations. In the left column two configurations made of carton boxes and cylinders inside a mock-up container are shown. In the middle column the convex polyhedron models of the objects fit to the point clouds of the scenes captured by the Kinect sensor are shown. In the right column the extracted support relations between the objects in the configurations are depicted.

$\mathcal{C}^{\mathrm{RW7}}$ and $\mathcal{C}^{\mathrm{RW8}}$. The true support relations (i.e., class labels) between pairs of objects in $\mathcal{C}_{\mathrm{RAND}}$ were then extracted by using the CSO method; the features were extracted by the method described in Section 5.3.2. The result is a training set of feature-labels for two real-world configurations. The three machine learning paradigms (i.e., SVM, ANN and random forest) were trained on the generated training set, and then used to estimate the probability of class labels, i.e., the support relations given the unseen feature-labels of the testing sets extracted from $\mathcal{T}_{\mathrm{m,j}}^{\mathrm{RW7}}$ and $\mathcal{T}_{\mathrm{m,j}}^{\mathrm{RW8}}$.

5.5.3 Results for the CSO Case

This section presents the results of applying the geometrical reasoning, static equilibrium analysis and decision making on the data generated in simulation and from real-world configurations explained in the preceding sub-sections. As a measure of the complexity of the generated configurations, the number and type of contact point-sets and the number of ACT and SUPP relations between

objects are reported. Moreover, the corresponding execution times were also recorded.

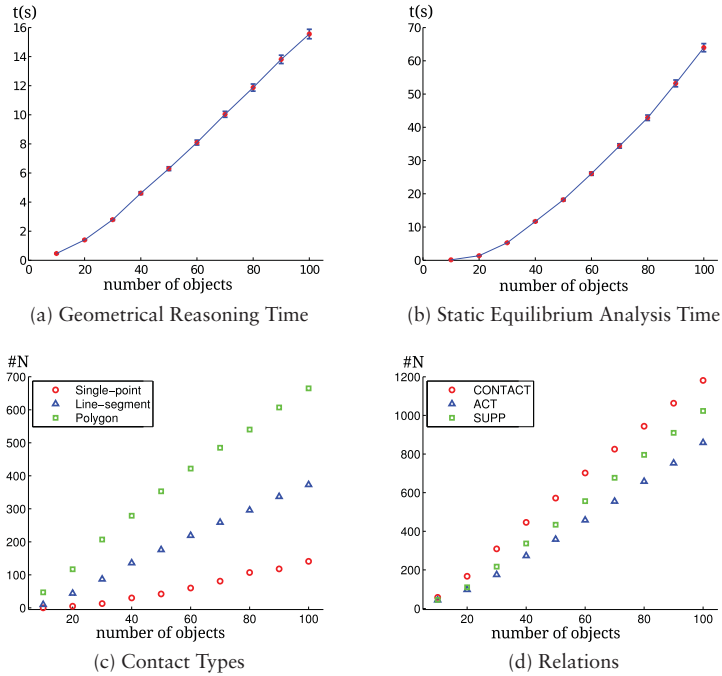
Figure 5.12a and Figure 5.12b respectively show the average time taken by the geometrical reasoning and static equilibrium analysis for the configurations $\mathcal{C}_n^{\text{MIX}}$. We can notice that the geometrical reasoning takes very short time and increases linearly with the number of objects, while the static equilibrium analysis requires considerably more time, and increases polynomially with respect to the number of objects. The longer execution time taken by the static equilibrium analysis is due to the non-linear solver of the system of equations (see Section 5.2.3), which has to be called for each object in a configuration. Nevertheless, for realistic scenarios, we expect the number of objects extracted by an object detection algorithm to be small, and thus, the execution time of the static equilibrium analysis would still be reasonably fast. On the other hand, since for each object the static equilibrium analysis is performed independently of the other objects, it is possible to speed up the process by parallelizing the computation of the support relations.

Figure 5.12c depicts the average number of contact types with respect to the number of objects. As expected, the number of single-point contacts which are the result of less stable configurations (vertex-on-face and edge-cross-edge) is noticeably lower than the number of line-segments and polygons, which are the result of edge-on-face and face-on-face contact types.

The average number of extracted `ACT` and `SUPP` relations between objects is shown in Figure 5.12d. The number of support relations increases linearly. The linear growth is due to the fact that adding one object implies a bounded number of contacts that can arise in the contact point-set network. For $N < 20$, the number of `ACT` and `SUPP` relations are close to each other, that is, for roughly each `ACT(X,Y)` relation, a corresponding `SUPP(Y,X)` relation was found. However, as the number of objects increases, the number of corresponding support relations diverges from that of `ACT` relations. This is due to the fact that an increase in the number of objects stacked in a fixed volume of a container increases the physical interaction between objects resulting in a more gravitational support dependencies.

The behavior of the execution time, the number of contact types and `ACT` and `SUPP` relations of the other three categories, is similar to those shown in Fig. 5.12 for the MIX category, but with different minimum and maximum values summarized in Table 5.4. From Table 5.4, we can observe that the minimum and maximum values for barrel shaped objects are greater than those of the other two objects categories. One reason is the higher number of faces in the shape of a barrel object, which is approximated by a polyhedron. The other reason is the shape of the barrel, which is less stable in a horizontal position. A similar behavior can be observed for cylinder shaped objects with respect to cuboid shaped objects.

The proposed geometrical and static equilibrium analysis (i.e., the method employed for the CSO case) was applied to the created 3D models (see Fig-



(a) Geometrical Reasoning Time (b) Static Equilibrium Analysis Time

(c) Contact Types (d) Relations

Figure 5.12: Complexity analysis of the proposed relational scene representation for simulated configurations. The horizontal axes are the number of objects in the configurations. In (a) and (b) the vertical axes depict the execution time of geometrical reasoning and static equilibrium analysis respectively. In (c) and (d) the vertical axes are the average of the number of contact types and relations respectively.

ure 5.11b and Figure 5.11e) of the two real-world configurations. The set of extracted support relations is represented as a graph in which each node indicates an object, and each directed link between two nodes indicates the existence of the support relation between the two objects. Figure 5.11c and Figure 5.11f show the result of extracting the support relations between objects in the two real-world configurations, $\mathcal{C}^{\mathrm{RW8}}$ and $\mathcal{C}^{\mathrm{RW7}}$ respectively. Looking at the two real-world configurations and the corresponding graphs of the support relations between objects, one can intuitively confirm the correctness of the extracted support relations. However, a further test was carried out in order to verify that the hypothesis of the support relations between objects representing by each graph is correct. Given a graph, each candidate object was unloaded manually from the real-world configurations to find out whether the unloaded

	CBX		CYL		BRL	
	min	max	min	max	min	max
SUP(#N)	10	223	21	474	35	760
ACT(#N)	10	183	20	392	33	642
CON(#N)	12	287	10	580	44	881
GET(Sec)	.001	0.04	0.05	1.59	0.41	12.89
SET(Sec)	0.01	7.74	0.05	23.67	0.12	48.27
SGL(#N)	0	76	0	93	0	103
LSG(#N)	2	113	5	199	8	282
PLY(#N)	10	97	22	287	36	494

Table 5.4: The results for three categories of objects, {CBX, CYL, BRL}, given as minimum and maximum values for the same number of objects in Figure 5.12. SUP(#N), ACT(#N) and CON(#N) stand for the number of support, act and contact relations respectively (see Fig.5.12d). GET(Sec) and SET(Sec) stand for geometrical reasoning and static analysis execution time, respectively (see Figure 5.12a and Figure 5.12b). SGL(#N), LSG(#N) and PLY(#N) stand for the number of single-point, line-segment and polygon contact types respectively (see Figure 5.12c).

objects have any effect on the motion state of the other objects. In order to select the candidate objects, we can look at the corresponding graph of support relations and selected objects that do not support any other object. The result of performing the procedure of manual unloading of objects on two real-world configurations confirmed the correctness of the extracted support relations. For example, box `B5` and cylinder `C1` in $\mathcal{C}^{RW7}$ (`B5` and `C1` are the candidates because there is no directed link of `SUPP` relation from their nodes to any other nodes) were unloaded manually, and it was observed that the rest of the objects in the configuration $\mathcal{C}^{RW7}$ preserved their motionless state.

Using the graph of support relations for each configuration (Figure 5.11c and Figure 5.11f), the costs of removing the objects in each configuration are summarized in Table 5.5. The best first choices (i.e., actions with the minimum possible costs) are $\{B_5, C_1\}$ and $\{B_1, B_4, C_1\}$ for $\mathcal{C}^{RW7}$ and $\mathcal{C}^{RW8}$ configurations, respectively.

5.5.4 Results for ICSO Case

The results of creating a probabilistic representation of support relations and the subsequent decision making process on data generated in simulation and

	Objects Set	
Cost	$\mathcal{C}^{RW8}$	$\mathcal{C}^{RW7}$
0	$\{B_1, B_4, C_1\}$	$\{B_5, C_1\}$
1	$\{B_2, B_5, B_6, C_2\}$	$\{B_1, B_2, B_6\}$
2	$\emptyset$	$\{B_3, C_2\}$
3	$\{B_3\}$	$\emptyset$

Table 5.5: The result of applying the proposed decision making to real-world configurations $\mathcal{C}^{RW7}$ and $\mathcal{C}^{RW8}$. The objects with equal computed cost are collected in the same sets for each configuration.

from real-world configurations are presented in this section. In order to compare the performance of the decision making to other possibilities, two heuristic decision makers are introduced that are supposed to provide a baseline decision performance. First, a *random decision maker* (Random DM) that uniformly selects an object, and the second is a *heuristic decision maker* (Heuristic DM) that selects the object with the highest center of mass to be removed in the corresponding test set.

The following describes the implementation of the approach for the ICSO case. The LibSVM [84] with radial basis function kernel was employed for Support Vector Machines, and a Matlab implementation was used for both Artificial Neural Networks and Random Forests (TreeBagger). For SVM and ANN, 70% of the training set was used for training and 30% for validation. For each category of objects, a 5-fold cross validation was performed to obtain the best values for the SVM parameters [76]. Measuring the classification success rate of ANN, a network with three hidden-layers of 15 neurons was empirically selected, and a Random Forest with 200 decision trees for each category of objects was trained. Figure 5.13 depicts the classification error rate of the three trained classifiers for four objects categories, $\mathcal{U} = \{\text{CBX}, \text{CYL}, \text{BRL}, \text{MIX}\}$ in a Receiver Operating Characteristic (ROC) space [85]. The ideal classification on ROC space is located at the point $(0, 1)$ – zero false positive rate (FPR) and 100% true positive rate (TPR). It can be seen that while SVM and ANN show a similar classification performance, the RFT classifier performs best for all categories of objects.

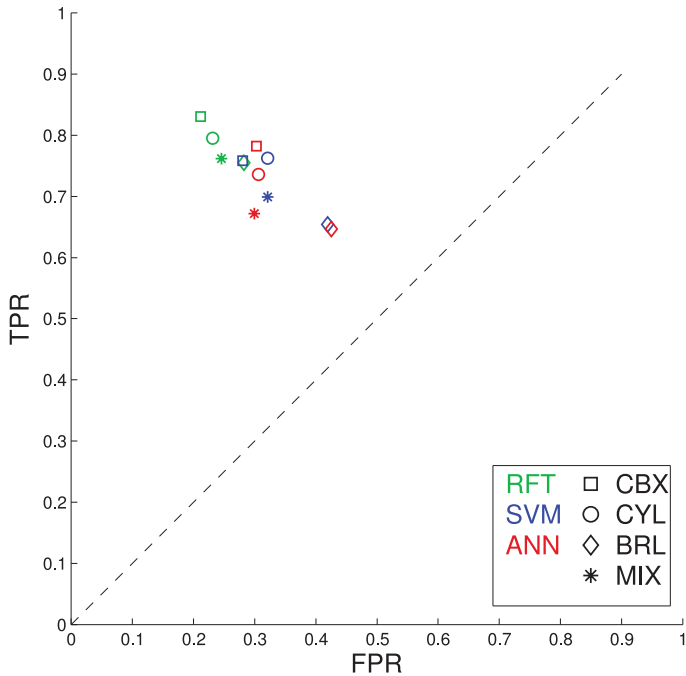


Figure 5.13: An illustration of the classification performance of the three trained classifiers, RFT, SVM and ANN for four categories of objects, {CBX, CYL, BRL, MIX} in ROC space. The horizontal and vertical axis represent the false positive rate (FPR) and the true positive rate respectively. The closest point to coordinate (0, 1) is the RFT classifier for configurations of carton boxes.

Evaluation Criteria

The performance of the decision makers was measured by computing the mean squared error (MSE) of the cost of removing the object selected by the corresponding decision maker for all test configurations, T_j^u,

$$\text{MSE(cost)} = \frac{1}{N}\sum_{j=1}^{N}(DMC_j - MPC_j)^2 \tag{5.28}$$

where, MPC_j is the minimum possible cost of selecting the optimum action for $\mathcal{J}_{m,j}^u$, DMC_j is the cost of the action selected by a decision maker, and N is the total number of $\mathcal{J}_{m,j}^u$.

The performance is represented with respect to three criteria. The first criterion is the average entropy of the estimated probabilities of the predicted

support relations by the corresponding machine learning paradigm for a given $\mathcal{J}^u_{m,j}$ with η support relations,

$$AE_{\mathcal{J}_j} = \frac{1}{\eta}\sum_{k=1}^{\eta}(-p_k \log(p_k) - p'_k \log(p'_k)) \tag{5.29}$$

where $p_k = P(S_k = 1)$ and $p'_k = P(S_k = 0)$. The idea behind using this criterion is to show how the performance varies as the uncertainty in the classifications changes. It is expected that with higher average entropies, we observe a decrease in the performance of the decision makers. The second criterion is the *balanced error rate* (BER) defined as,

$$BER = \frac{1}{2}\left(\frac{N_W(L1)}{N(L1)} + \frac{N_W(L2)}{N(L2)}\right) \tag{5.30}$$

where, $N_W(L1)$ and $N_W(L2)$ are the number of L1 and L2 class instances predicted wrongly, and $N(L1)$ and $N(L1)$ are the number of total L1 and L2 class instances. The third criterion is the *success rate* of a machine learning paradigm, which is the percentage of the class instances predicted correctly.

Configurations Generated in Simulation

For each category of objects, Figure 5.17, Figure 5.18 and Figure 5.19 show the performance of the probabilistic decision maker (e.g., SVM DM) comparing to Random DM and Heuristic DM with respect to each classifier's success rate, balanced error rate, and average entropy. The histogram of the percentage of test configurations $\mathcal{J}^u_{m,j}$ that fall into each bin of a criterion is depicted at the bottom of each graph.

There are two fundamental observations from the results. First, the performance of the probabilistic decision makers outperform both Random DM and Heuristic DM, and second, the Random Forest DM was found to be clearly better than ANN DM and SVM DM. It can be seen that the performance of the probabilistic decision makers improves (i.e., MSE of the cost decreases) as the success rate of the classifier increases (see the third columns of Figure 5.17, Figure 5.18 and Figure 5.19). As expected, a similar behavior can be seen with the balanced error rate (see middle columns of Figure 5.17, Figure 5.18 and Figure 5.19).

In Figure 5.17, a majority of test configurations have the average entropy between 0.4 and 0.7, and for these we observe an approximately constant performance. When the average entropy increases from 0.7 upwards the performance of the decision maker decreases, as higher average entropies reflect the difficulty of classifying support relations in the corresponding configurations. We can see that the average entropies of the estimated class probabilities by ANN are very close to 1 (see the first column of Figure 5.19) which explains

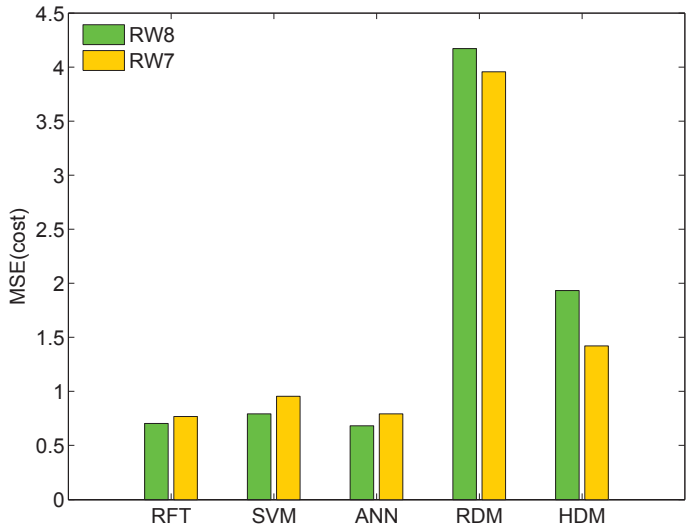


Figure 5.14: An illustration of the proposed decision making performance for the two real world configurations, $\mathcal{C}^{\mathrm{RW7}}$ and $\mathcal{C}^{\mathrm{RW8}}$, compared to the performance of both the random and the heuristic decision maker. The vertical axes represent the mean squared error of cost defined in Eq. 5.28. The three bins on the left side of the horizontal axes represent decision makers based on RFT, SVM and ANN classifiers while the right side bin represents the random decision maker.

the poor performance of the proposed decision maker when using the output probabilities of the ANN. As the average entropy approaches the maximum of entropy, 1, the probabilities of possible worlds are closer to each other, i.e., different possible worlds become more equally likely, and therefore taking a decision is less and less informed about the true support relations.

Real World Configurations

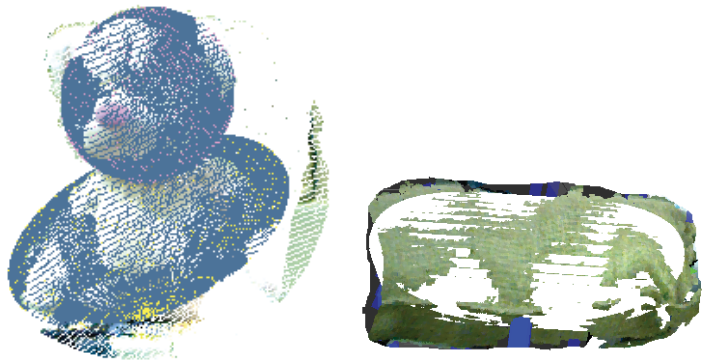
The performance of the probabilistic decision makers on the real-world data of $\mathcal{C}^{\mathrm{RW7}}$ and $\mathcal{C}^{\mathrm{RW8}}$ configurations is presented in Figure 5.14 showing the computed MSE(cost) (see Eq. 5.28) of the probabilistic, random and heuristic decision makers. Similar to the configurations generated in simulation, it can be seen that the decision making based on the probability estimation of the support relations outperforms both the random and heuristic decision makers. We also notice that the performance of decision making based on the ANN classifier is comparable to that of RFT classifier, unlike the behavior we observed in the simulated configurations.

Figure 5.15: The sequence of the selected objects by the probabilistic decision maker is shown for a real-world scenario of the RobLog project. Starting from top left and following the arrows, the selected object at each step is highlighted by a bold boundary line around the object.

RobLog Scenarios

The probabilistic decision making about the safest object to remove from a pile was employed for the unloading scenarios of the RobLog project successfully. Figure 5.15 shows the sequence of the selected objects for a sample configuration of objects in the RobLog scenario (see Section 2.1 for an explanation of the scenarios of the RobLog project). In the scenario there exist deformable objects such as teddy bear dolls and sacks, where the corresponding object detection module estimates the shape of such deformable objects as rigid superquadrics (see Figure 5.16) imposing more uncertainty in the input data to the decision making algorithm. For example, the experiments show that a teddy bear doll is usually detected as two objects, where two superquadrics are fit to the head and body of a teddy bear representing a more ambiguous scenario. However, observing the sequence of the selected objects in the real-world test configurations created for the RobLog project shows a reliable performance of the probabilistic decision making presented in this chapter. For example, we can observe that unloading the sequence of the selected objects in the order shown in Figure 5.15 preserves the stability of the configuration, and we can intuitively verify that the sequence of the selected objects is safe.

The analysis of the estimated probabilities of the support relations in the scenario shown in Figure 5.15, verifies the procedure of the probabilistic decision making approach. For example, observing the second selection, the reason that the decision maker selected the teddy bear, `Bear`, standing behind the wash-



(a) Teddy Bear Doll (b) Sack

Figure 5.16: Superquadrics shape estimation for deformable objects. (a) The shape of a teddy bear doll is estimated with two superquadrics representing the head and the body. (b) The shape of a sack object is estimated by superquadrics.

ing liquid bottle, `Bottle`, is the probability of `SUPP(Bottle,Bear)` (`Bear` leans on `Bottle`) is higher than `SUPP(Bear,Bottle)` (`Bottle` leans on `Bear`).

5.6 Discussion

This chapter presented a novel approach to analyze and represent static configurations of piles of objects under two conditions, having access to complete and incomplete set of objects in the configurations. The proposed approach is mainly aimed to be employed in the process of automating the task of unloading goods from shipping containers, however, the methodology is applicable to a wide range of similar applications (e.g., safely picking up an object from a shelf by a domestic robot).

In case of having access to the complete set of objects, a method to automatically extracting a symbolic relational representation that uses a minimal set of relations to capture possible physical interactions between objects is described. Such a relational representation can be readily used by high-level AI reasoning paradigms to predict the effects of removing objects in contact with each other.

When some objects in a configuration are possibly not detected, a probabilistic world model of the support relations was introduced based on machine learning techniques. The performance of three type of classifiers, Random Forests (of decision trees), Support Vector Machines (SVM) and Artificial Neural Networks (ANN) in the estimation of the probability of the support relations were examined. The probabilistic world models are then used to make an optimal decision on the safest object to be removed from a configuration based on minimizing the cost of taking unloading actions.

The presented methods were evaluated with the data generated in simulation and from real-world configurations of objects. The results show that the performance of the proposed probabilistic decision maker in combination with the output of the classifiers outperforms randomly selecting an object to remove from a pile, and it also shows better results than using a heuristic rule of always removing the topmost object. It is also observed that using the output of Random Forests classifier improves the performance of the probabilistic decision maker most. The abundance of diverse training data available through the simulator leads to the conjecture that an ensemble learning system is more reliably able to exploit the structure in the data (see, for example, the discussion of the advantages of ensemble methods in [86]). Considering the presented results, the probabilistic method for making decisions about the safest object to be removed from a pile in the ICSO case is well motivated. The results also constitute a step forward in terms of bringing cognitive reasoning abilities to the area of robotic manipulation for autonomous object selection.

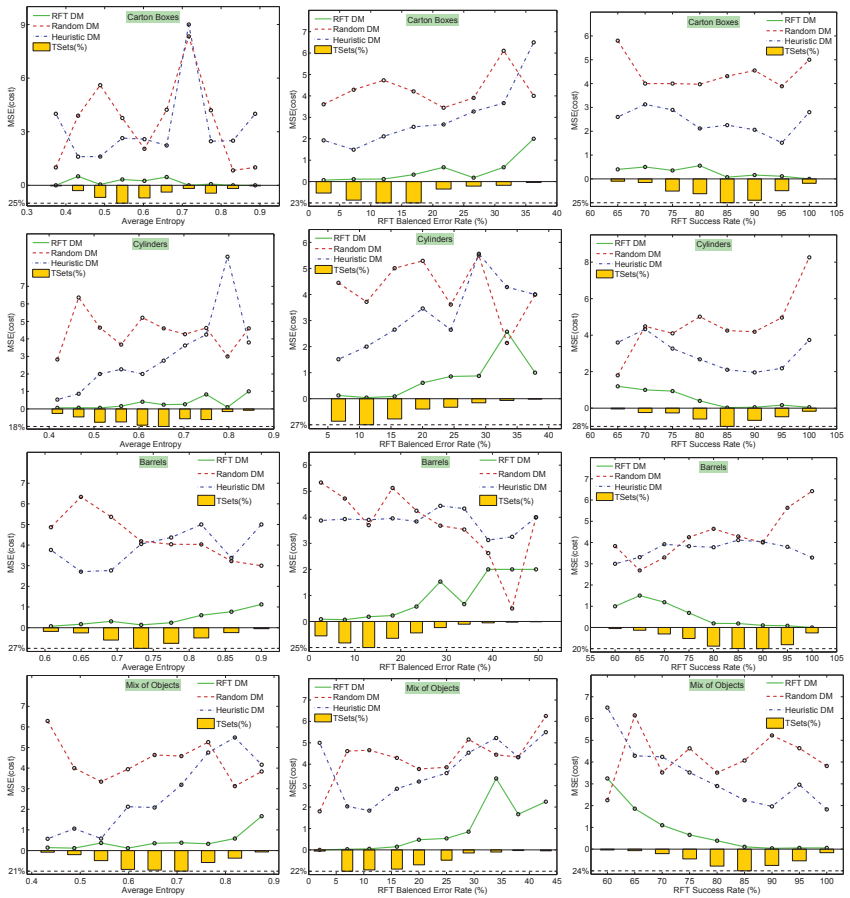


Figure 5.17: The performance of the proposed RFT decision maker (RFT DM), the random decision maker (Random DM) and the heuristic decision maker (Heuristic DM) are depicted versus RFT classifier's average entropy, balanced error rate and success rate from the left to right columns respectively. Vertical axes are MSE(cost) as defined in Section 5.5.4. The category of objects from top to bottom row are: (first row) carton boxes; (second row) cylinders; (third row) barrels; (fourth row) mix of objects. The histograms at the bottom of each graph shows the percentage of test sets (TSets) in the corresponding bin. The lower MSE(cost) especially at higher bins shows better performance.

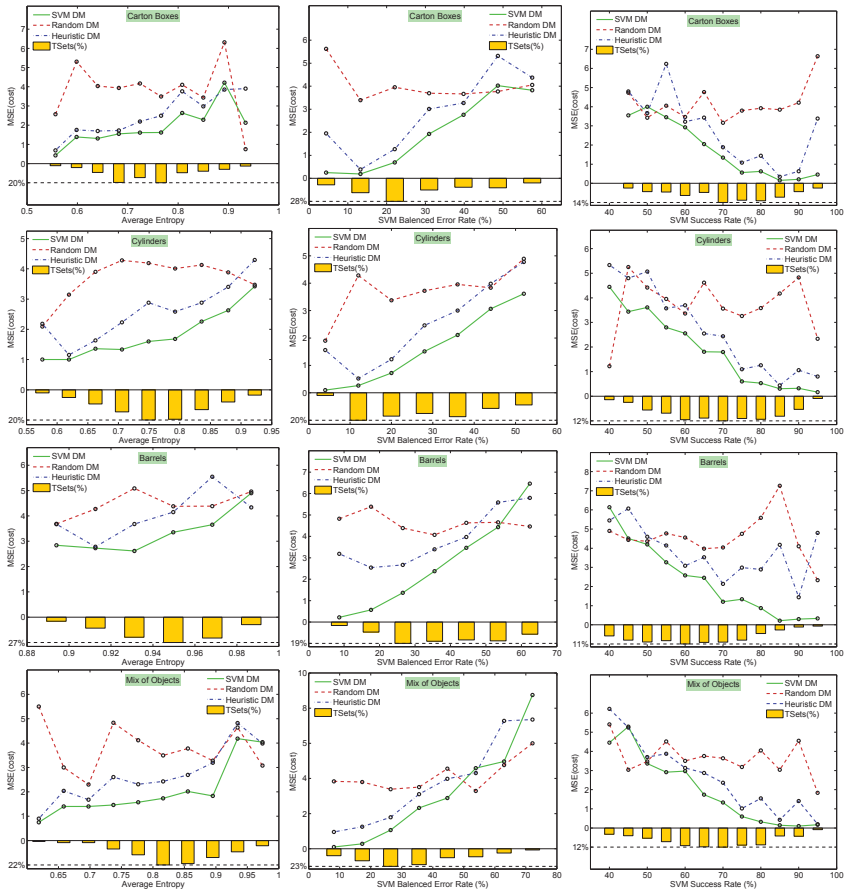


Figure 5.18: The performance of the proposed SVM decision maker (SVM DM), the random decision maker (Random DM) and the heuristic decision maker (Heuristic DM) are depicted versus SVM classifier's average entropy, balanced error rate and success rate from the left to right columns respectively. Vertical axes are MSE(cost) as defined in Section 5.5.4. The category of objects from top to bottom row are: (first row) carton boxes; (second row) cylinders; (third row) barrels; (fourth row) mix of objects. The histograms at the bottom of each graph shows the percentage of test sets (TSets) in the corresponding bin. The lower MSE(cost) especially at higher bins shows better performance.

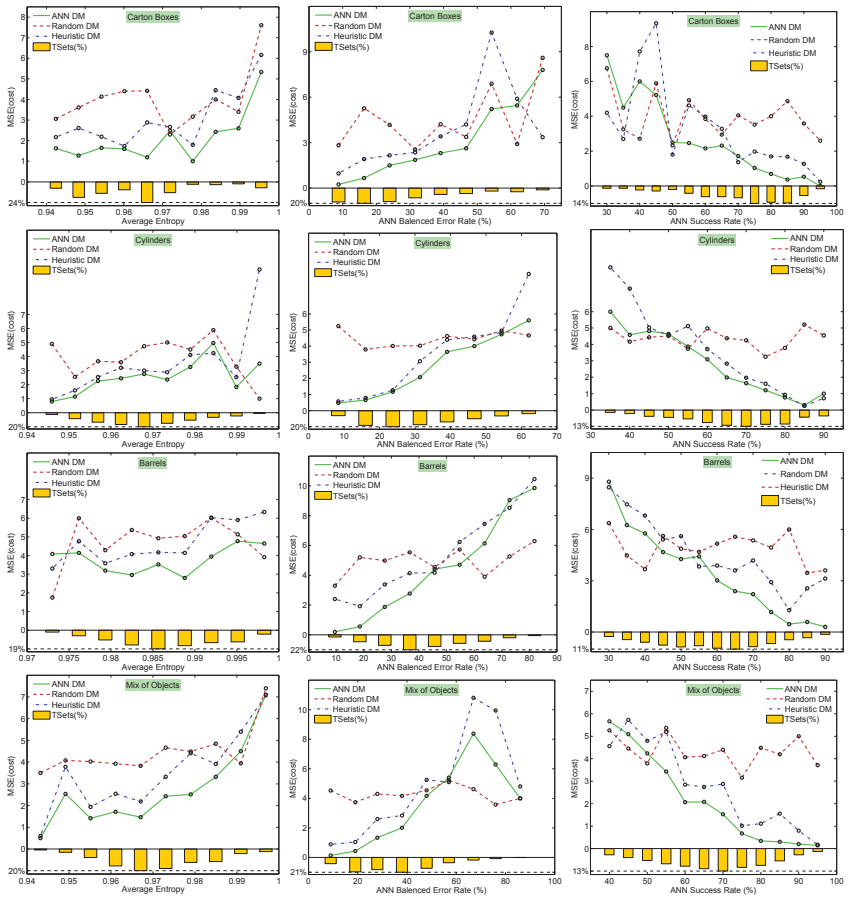


Figure 5.19: The performance of the proposed ANN decision maker (ANN DM), the random decision maker (Random DM) and the heuristic decision maker (Heuristic DM) are depicted versus ANN classifier's average entropy, balanced error rate and success rate from the left to right columns respectively. Vertical axes are MSE(cost) as defined in Section 5.5.4. The category of objects from top to bottom row are: (first row) carton boxes; (second row) cylinders; (third row) barrels; (fourth row) mix of objects. The histograms at the bottom of each graph shows the percentage of test sets (TSets) in the corresponding bin. The lower MSE(cost) especially at higher bins shows better performance.

Chapter 6
Conclusion and Future Work

This dissertation focused on the essential task of object selection by autonomous robotic manipulation systems to reduce the probability of damage to the objects stacked in a pile. Starting from 3D perception and the evaluation of the sensors technologies, reaching geometrical consistency in the detected poses of objects, this thesis attempted to analyze the stability of a pile under an incomplete detection of objects and uncertainty in the data. The contribution presented in this thesis work were developed in the scope of a EU-FP7 project, which successfully demonstrated a robotic manipulation system for automating the logistics process of unloading goods from shipping containers. This chapter presents a summary of the main contributions of the thesis and an analysis of their significance. Open questions are then discussed together with directions for future work.

6.1 Major Contributions

This section highlights the three most important achievements of this work with a description of the corresponding challenges.

The first contribution in this dissertation is an in-depth analysis of the problem of autonomous safe selection of objects from a pile in order to either remove a single object or unload all the objects from the pile. Depending on the available data, two cases were considered: having access to a complete set of objects (CSO) and to an incomplete set of objects (ICSO). In the CSO case it is assumed that all the objects composing a pile are detectable while in the ISCO case only a subset of objects are assumed to be detected.

For the case that shapes and poses of the objects are available, geometrical reasoning following by a static equilibrium analysis were introduced to extract a minimal set of symbolic relations, namely, ACT and SUPPORT relations representing how the objects in a pile are in physical interactions. Such symbolic ACT and SUPPORT relations can be readily used by a high-level AI reason-

ing module to analyze the stability of a pile and reason about the safest set of objects to unload from the pile.

An alternative probabilistic approach to extracting support relations was discussed to tackle the problem of undetected objects of a pile due to occlusion or a failure in the corresponding object detection algorithms, in addition to the problem of uncertainty in the estimated poses. The probabilistic approach estimates the probability of the support relations between pairs of the detected objects using machine learning techniques by extracting features from the relative position of the two objects and the point cloud of the pile. An extensive experimental evaluation of the presented approach to identify the set of safe-to-remove objects was conducted on data generated in simulation and from real-world configurations of objects. It was also demonstrated that the object selection algorithms presented in this dissertation can be employed in practical applications such as the RobLog project successfully.

The second major contribution is an efficient search based algorithm that refines the poses of a set of objects detected by an existing object detection module. The algorithm resolves the inter-penetrations between the shapes due to errors in the estimated poses using high-level reasoning in order to obtain a geometrically consistent model of the environment. In this work, the concept of minimum translation search for object pose refinement was introduced. A discrete search paradigm based on the concept of depth of penetration between two polyhedrons was explored to overcome the practical problem of an exhaustive search in the full state space of the poses to find a geometrically consistent solution. The performance of the object pose refinement algorithm was examined on data sets generated in simulation and from real-world configurations of objects, empirically showing that the presented algorithm not only resolves the inter-penetrations but it also reduces the overall pose error on average. Also provided is an open-source C++ implementation of the introduced algorithm.

Last but not least, an application based evaluation of 3D range sensors is presented in this thesis in order to select a set of appropriate sensors considering the task of object detection in the design process of the RobLog project. Is was demonstrated that selecting 3D range sensors solely based on comparing their intrinsic properties and in isolation of the target application may result in an inappropriate choice. As performance indicators, two state-of-the-art object detection and pose estimation algorithms with two major categories of objects commonly used in shipping containers, namely, carton boxes and tires were selected for experimental trials. With the proposed evaluation approach it was shown that in the design process of a robotic system that is required to autonomously detect objects, the applicability of 3D range sensors, regardless of their intrinsic parameters, significantly depends on the types of objects and the object detection algorithms. Based on this evaluation the Kinect sensor for short range scanning and an actuated laser range finder (SICK LMS-200) for scanning longer distances deep inside cargo containers were selected for the RobLog project.

6.2 Limitations

This section discusses the limitations in some of the developed methods of this dissertation. It is important to take into account the limitations discussed in this section in the integration process of the presented methods into complex robotic systems.

In Chapter 5, depending on the available data of a complete set of objects (CSO) or an incomplete set of objects (ICSO), two different approaches to extract gravitational support relations between a set of objects of a pile were discussed. The presented algorithms have the following limitations to be considered. First, the shapes of objects are assumed to be convex polyhedrons, and the experimental results are presented under the convexity assumption for the shapes. The probabilistic approach, however, is not restricted to convex shaped objects, since the probability distributions of the support relations between objects of concave shapes can be learned through machine learning techniques. Second, for the both cases, CSO and ICSO, it is assumed that the objects are rigid and not deformable. However, the presented results of identifying safe-to-remove objects for the practical setups of the RobLog project shows the applicability of the probabilistic approach to deal with the deformable objects when the shapes are represented by superquadrics models. The third limitation to note is the assumption that the geometrical attributes of all the objects are available in the CSO case. Such an assumption limits the applicability of the approach to the configurations consisting of few detectable objects where the errors in the detected poses are In addition, using the approach presented for the CSO case enables us to automatically label the true support relations for a large training dataset, which is generated in simulation, to be used in the probabilistic approach. As the last limitation we notice that in order to obtain a reliable machine learning model of the support relations of a target configuration, we need to create a training dataset of the same configurations of objects as the target application. The experiments, however, showed that the usage of a physics simulation to generate random configurations of the target environment is an appropriate solution together with employing the approach to automatic labeling of the support relations discussed for the CSO case.

The object pose refinement algorithm, proposed in Chapter 4, has also several limitations, that should be taken into account. Using different methods of searching the state space of depth of penetrations, the proposed algorithm attempts to resolve all the inter-penetrations between the shapes of objects in order to obtain a geometrically consistent model of the environment. While the ultimate goal of the presented algorithm is to obtain an inter-penetration free configuration of the objects, for the reasons discussed in Section 4.1, the rotation part of the poses was not considered in the search process, which is a direction for further evaluation as future work. Applicable to the discussion in the preceding paragraph, the shapes of objects are assumed to be known,

for example, having a descriptive database of the shapes. Another limitation of the pose refinement algorithm is the assumption that the shapes of the detected objects are classified correctly by the corresponding object detection module. It should be noticed that for the experimental results of the proposed object pose refinement algorithm, it is assumed that all the objects of a configuration are detected, but such assumption is not necessary considering the underlying objective which is to resolve the inter-penetrations due to the errors in the initially estimated poses, and not to re-estimate the poses.

6.3 Future Research Directions

Considering the limitations discussed in the previous section, several improvements and further investigations of the algorithms proposed in this dissertation are readily identifiable. Relaxing the rigid body assumption made on the objects is an interesting and important direction for further study of the possible extension to the proposed algorithms for both safe object selection and object pose refinement. An appropriate model development for the deformable objects to be used with the aforementioned algorithms implicitly relaxes the convexity assumption on the shapes of the objects. One possible approach for working with deformable objects would be to employ a polygon mesh representation of the surface of the deformable object and model it as a skeleton of a soft object. Here, the idea would be to divide the skeleton of a soft object into groups of connected polygons labeled based on whether a group is in contact with another object. An extension to the static equilibrium analysis for the labeled regions could identify the stability of the soft object. Assuming that the idea of the soft object based analysis has been already developed, a comparative evaluation is required to contrast the performance of that idea with estimating the deformable objects with superquadrics models (e.g., the approach employed for the RobLog scenario).

Another important direction for future work is the investigation of the possibility to integrate visual clues such as a point cloud of the scene into the search based algorithms for the object pose refinement presented in Chapter 4. Here, an idea is to define a metric function such as the sum of distances of the closest points to the surfaces of the shapes as a measure for the quality of a solution found as an inter-penetration free configuration of objects. Using such metric function, the rotation component of the poses could be part of an optimization problem in an attempt to fine tune the poses while maintaining the geometrical consistency condition.

Finally, considering the search based object pose refinement approach, given a database of the shapes it would be interesting to cast the problem of misclassification of the shapes as, accordingly, *object shape hypothesis refinement* assuming that the estimated poses are nearly correct. Here, the idea comes from the fact that if the poses are correct, then inter-penetrations between pairs of objects are due to misclassification of the estimated shapes. Similar to the pre-

sented object pose refinement approach, a discrete search into the state space of the shapes would resolve the possible inter-penetrations resulting in a geometrically consistent model of the environment, and has the potential to reduce the misclassification error of the estimated shapes. Much like mentioned in above paragraph, investigating the possibility of using the visual clues to improve the number of correctly classified of the shapes is another interesting direction of research.

References

[1] JaYoung Sung, H.I. Christensen, and R.E. Grinter. Sketching the future: Assessing user needs for domestic robots. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 153–158, Sept 2009. (Cited on page 4.)

[2] H. Moradi, K. Kawamura, E. Prassler, G. Muscato, P. Fiorini, T. Sato, and R. Rusu. Service robotics (the rise and bloom of service robots) [tc spotlight]. *Robotics Automation Magazine, IEEE*, 20(3):22–24, Sept 2013. (Cited on page 4.)

[3] K.A. Wyrobek, E.H. Berger, H.F.M. Van der Loos, and J.K. Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2165–2170, May 2008. (Cited on page 4.)

[4] T. Stoyanov, N. Vaskeviciusz, C. A. Mueller, T. Fromm, R. Krug, V. Tincani, R. Mojtahedzadeh, S. Kunaschk, R. M. Ernits, D. R. Canelhas, M. Bonilla, S. Schwertfeger, M. Bonini, H. Halfar, K. Pathak, M. Rohde, G Fantoni, A. Bicchi, A. Birk, A. Lilienthal, and W. Echelmeyer. No more heavy lifting: Robotic solutions to the container unloading problem. *IEEE Robotics and Automation Magazine*, In Press. (Cited on page 12.)

[5] F. Bley, V. Schmirgel, and K. F Kraiss. Mobile manipulation based on generic object knowledge. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 411–416, Sept 2006. (Cited on pages 14, 15, and 19.)

[6] Han-Young Jang, Hadi Moradi, Phuoc Le Minh, Sukhan Lee, and JungHyun Han. Visibility-based spatial reasoning for object manipulation in cluttered environments. *Computer-Aided Design*, 40(4):422 – 438, 2008. (Cited on pages 14, 15, and 19.)

[7] E Klingbeil, D Rao, B Carpenter, V Ganapathi, A Y Ng, and O Khatib. Grasping with application to an autonomous checkout robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2837–2844, Shanghai, China, 2011. (Cited on pages 14, 15, and 19.)

[8] Jacqueline Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1377–1382, May 2009. (Cited on pages 14, 15, and 19.)

[9] Amit Agrawal, Yu Sun, John Barnwell, and Ramesh Raskar. Vision-guided robot system for picking objects by casting shadows. *Int. J. Rob. Res.*, 29(2-3):155–173, February 2010. (Cited on pages 14, 15, and 19.)

[10] Katsushi Ikeuchi, Berthhold K.P. Horn, Shigemi Nagata, Tom, Tom Callahan, and Oded Feingold. Picking up an object from a pile of objects. In *Proceedings of the First International Symposium on Robotics Research*, pages 139–166. MIT Press, 1983. (Cited on pages 14 and 19.)

[11] B.K.P. Horn and Katsushi Ikeuchi. The mechanical manipulation of randomly oriented parts. *Scientific American*, 251(2):100 – 109, August 1984. (Cited on pages 14 and 19.)

[12] Jean-Daniel Dessimoz, John R. Birk, Robert B. Kelley, H.A.S. Martins, and Chi Lin. Matched filters for bin picking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):686–697, Nov 1984. (Cited on pages 15 and 19.)

[13] H S Yang and A C Kak. Determination of the identity, position and orientation of the topmost object in a pile. *Comput. Vision Graph. Image Process.*, 36(2-3):229–255, November 1986. (Cited on pages 15 and 19.)

[14] E. Al-Hujazi and A. Sood. Range image segmentation combining edge-detection and region-growing techniques with applications sto robot bin-picking using vacuum gripper. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(6):1313–1325, Nov 1990. (Cited on pages 15 and 19.)

[15] K Rahardja and A Kosaka. Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues. In *1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1448–57. IEEE Press, 1996. (Cited on pages 15 and 19.)

[16] Martin Berger, Gernot Bachler, and Stefan Scherer. Vision guided bin picking and mounting in a flexible assembly cell. In Rasiah Logananthara, GÃ¼nther Palm, and Moonis Ali, editors, *Intelligent Problem Solving. Methodologies and Approaches*, volume 1821 of *Lecture Notes in Computer Science*, pages 109–117. Springer Berlin Heidelberg, 2000. (Cited on pages 15 and 19.)

[17] Advait Jain and CharlesC. Kemp. El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28(1):45–64, 2010. (Cited on pages 15 and 19.)

[18] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011. (Cited on pages 15 and 19.)

[19] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems VIII*, July 2012. (Cited on pages 15 and 19.)

[20] K. Sjoo, A. Aydemir, T. Morwald, Kai Zhou, and P. Jensfelt. Mechanical support as a spatial abstraction for mobile robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4894–4900, Oct 2010. (Cited on pages 16 and 19.)

[21] L. Chang, J.R. Smith, and D. Fox. Interactive singulation of objects from a pile. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3875–3882, May 2012. (Cited on pages 15 and 19.)

[22] M Gupta and G S Sukhatme. Using manipulation primitives for brick sorting in clutter. In *In Proc. IEEE Int. Conf. on Robotics and Automation , 2012*, pages 3883–3889. IEEE Press, 2012. (Cited on pages 15 and 19.)

[23] M. Kopicki, S. Zurek, R. Stolkin, T. Morwald, and J. Wyatt. Learning to predict how rigid objects behave under simple manipulation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5722–5729, May 2011. (Cited on pages 17 and 19.)

[24] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *International Journal of Robotics Research*, 30(11):1328–1342, 2011. (Cited on pages 17 and 19.)

[25] Kristoffer SjÃ¶Ã¶ and Patric Jensfelt. Learning spatial relations from functional simulation. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1519. IEEE, 2011. (Cited on pages 17 and 19.)

[26] S. Panda, A.H.A. Hafez, and C.V. Jawahar. Learning support order for manipulation in clutter. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 809–815, Nov 2013. (Cited on pages 17 and 19.)

[27] Todor Stoyanov, Rasoul Mojtahedzadeh, Henrik Andreasson, and Achim J. Lilienthal. Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications. *Robotics and Autonomous Systems*, 61(10):1094 – 1105, 2013. Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011). (Cited on pages 21, 22, and 26.)

[28] Uland Wong, Aaron Morris, Colin Lea, James Lee, Chuck Whittaker, Ben Garney, and Red Whittaker. Comparative evaluation of range sensing technologies for underground void modeling. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3816–3823, Sept 2011. (Cited on pages 21 and 22.)

[29] Cang Ye and J. Borenstein. Characterization of a 2d laser scanner for mobile robot obstacle negotiation. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 2512–2518, 2002. (Cited on pages 22 and 29.)

[30] Xiujuan Luo and Hong Zhang. Characterization of acuity laser range finder. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 3, pages 2100–2104 Vol. 3, Dec 2004. (Cited on page 22.)

[31] James Christian Charles Mure-Dubois and Heinz HÃ¼gli. Real-time scattering compensation for time-of-flight camera. In *Proceedings of the International Conference Computer Vision Systems (ICVS)*, 2007. (Cited on page 22.)

[32] Stefan Fuchs and G. Hirzinger. Extrinsic and depth calibration of tof-cameras. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6, June 2008. (Cited on page 22.)

[33] Filiberto Chiabrando, Roberto Chiabrando, Dario Piatti, and Fulvio Rinaudo. Sensors for 3d imaging: Metric evaluation and calibration of a ccd/cmos time-of-flight camera. *Sensors*, 9(12):10080, 2009. (Cited on page 22.)

[34] A. Prusak, O. Melnychuk, H. Roth, I. Schiller, and R. Koch. Pose estimation and map building with a time-of-flight-camera for robot navigation. *Int. J. Intell. Syst. Technol. Appl.*, 5(3/4):355–364, November 2008. (Cited on page 22.)

[35] S. May, D. Droeschel, Stefan Fuchs, D. Holz, and A. Nuchter. Robust 3d-mapping with time-of-flight cameras. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1673–1678, Oct 2009. (Cited on page 22.)

[36] Yan Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1173–1180, June 2010. (Cited on page 22.)

[37] D. Droeschel, D. Holz, J. Stuckler, and S. Behnke. Using time-of-flight cameras with active gaze control for 3d collision avoidance. In *Robotics*

and Automation (ICRA), 2010 IEEE International Conference on, pages 4035–4040, May 2010. (Cited on page 22.)

[38] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437, 2012. (Cited on page 22.)

[39] Lim Chee Chin, S.N. Basah, S. Yaacob, M.Y. Din, and Y.E. Juan. Accuracy and reliability of optimum distance for high performance kinect sensor. In *Biomedical Engineering (ICoBE), 2015 2nd International Conference on*, pages 1–7, March 2015. (Cited on page 22.)

[40] N.M. DiFilippo and M.K. Jouaneh. Characterization of different microsoft kinect sensor models. *Sensors Journal, IEEE*, 15(8):4554–4564, Aug 2015. (Cited on page 22.)

[41] R. Mojtahedzadeh, T. Stoyanov, and A.J. Lilienthal. Application based 3d sensor evaluation: A case study in 3d object pose estimation for automated unloading of containers. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 313–318, Sept 2013. (Cited on page 23.)

[42] W Echelmeyer, A Kirchheim, A J Lilienthal, H Akbiyik, and M Bonini. Performance indicators for robotics systems in logistics applications. In *IROS Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (MMARTLOG)*, 2011. (Cited on page 23.)

[43] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, pages 1848–1853, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 23.)

[44] Renaud Detry and Justus Piater. Continuous surface-point distributions for 3D object pose estimation and recognition. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Asian Conference on Computer Vision*, volume 6494 of *LNCS*, pages 572–585, Heidelberg, 2010. Springer. (Cited on page 23.)

[45] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, pages 803–827, 2007. (Cited on page 24.)

[46] Ping Liang and John S. Todhunter. Representation and recognition of surface shapes in range images: a differential geometry approach. *Comput. Vision Graph. Image Process.*, 52(1):78–109, August 1990. (Cited on page 24.)

[47] Renaud Detry and Justus Piater. Continuous surface-point distributions for 3D object pose estimation and recognition. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Asian Conference on Computer Vision*, volume 6494 of *LNCS*, pages 572–585, Heidelberg, 2010. Springer. (Cited on page 24.)

[48] C. Choi and H.I Christensen. 3D Pose Estimation of Daily Objects Using an RGB-D Camera. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3342–3349, Oct 2012. (Cited on page 31.)

[49] N. Vaskevicius, K. Pathak, A Ichim, and A Birk. The Jacobs Robotics Approach to Object Recognition and Localization in the Context of the ICRA'11 Solutions in Perception Challenge. In *Proc. of the IEEE/RSJ Int. Conf. on Robotics and Automation*, pages 3475–3481, May 2012. (Cited on page 31.)

[50] D. Rother and R. Vidal. A Hypothesize-and-Bound Algorithm for Simultaneous Object Classification, Pose Estimation and 3D Reconstruction from a Single 2D Image. In *IEEE Int. Conf. on Computer Vision Workshops*, pages 553–560, Nov 2011. (Cited on page 31.)

[51] R. Sandhu, S. Dambreville, A Yezzi, and A Tannenbaum. Non-Rigid 2D-3D Pose Estimation and 2D Image Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 786–793, June 2009. (Cited on page 31.)

[52] A Collet, D. Berenson, S.S. Srinivasa, and Dave Ferguson. Object Recognition and Full Pose Registration From a Single Image For Robotic Manipulation. In *Proc. of the IEEE/RSJ Int. Conf. on Robotics and Automation*, pages 48–55, May 2009. (Cited on page 31.)

[53] Wei Wang, Lili Chen, Dongming Chen, Shile Li, and K. Kuhnlenz. Fast Object Recognition and 6D Pose Estimation Using Viewpoint Oriented Color-Shape Histogram. In *IEEE Int. Conf. on Multimedia and Expo*, pages 1–6, July 2013. (Cited on page 31.)

[54] A Aldoma, F. Tombari, J. Prankl, A Richtsfeld, L. Di Stefano, and M. Vincze. Multimodal Cue Integration Through Hypotheses Verification for RGB-D Object Recognition and 6DOF Pose Estimation. In *Proc. of the IEEE/RSJ Int. Conf. on Robotics and Automation*, pages 2104–2111, May 2013. (Cited on page 31.)

[55] J.J. Lim, H. Pirsiavash, and A Torralba. Parsing IKEA Objects: Fine Pose Estimation. In *IEEE Int. Conf. on Computer Vision*, pages 2992–2999, Dec 2013. (Cited on page 31.)

[56] T. Grundmann, M. Fiegert, and W. Burgard. Probabilistic Rule Set Joint State Update as Approximation to the Full Joint State Estimation Applied to Multi Object Scene Analysis. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2047–2052, Oct 2010. (Cited on page 32.)

[57] Aitor Aldoma, Federico Tombari, Luigi Di Stefano, and Markus Vincze. A global hypotheses verification method for 3d object recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 511–524, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on page 32.)

[58] L.L.S. Wong, L.P. Kaelbling, and T. Lozano-Perez. Collision-free State Estimation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 223–228, May 2012. (Cited on page 32.)

[59] Brian Mirtich and John Canny. Impulse-based Simulation of Rigid Bodies. In *Proc. of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, pages 181–ff., New York, NY, USA, 1995. ACM. (Cited on page 32.)

[60] Christer Ericson. *Real-time Collision Detection*. Elsevier, Amsterdam Boston, 2005. isbn: 978-1558607323. (Cited on pages 33 and 34.)

[61] Liangjun Zhang, Young J. Kim, Gokul Varadhan, and Dinesh Manocha. Generalized penetration depth computation. *Comput. Aided Des.*, 39(8):625–638, August 2007. (Cited on pages 33 and 41.)

[62] S. Cameron and R. Culley. Determining the minimum translational distance between two convex polyhedra. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 591–596, Apr 1986. (Cited on page 33.)

[63] Stephen Cameron. Enhancing gjk: Computing minimum and penetration distances between convex polyhedra. In *Proceedings of International Conference on Robotics and Automation*, pages 3112–3117, 1997. (Cited on page 33.)

[64] Y.J. Kim, M.C. Lin, and D. Manocha. Deep: dual-space expansion for estimating penetration depth between convex polytopes. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 921–926 vol.1, 2002. (Cited on page 33.)

[65] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. isbn: 978-0521833783. (Cited on page 33.)

[66] Gino Van Den Bergen. Proximity queries and penetration depth computation on 3d game objects. In *In Game Developers Conference*, 2001. (Cited on page 33.)

[67] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. 2010. isbn: 978-0132071482. (Cited on pages 37 and 39.)

[68] Jyh-Ming Lien and Nancy Amato. Approximate Convex Decomposition of Polyhedra and Its Applications. *Computer Aided Geometric Design*, October 2008. (Cited on page 40.)

[69] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. isbn: 9780521833783. (Cited on page 55.)

[70] Dave H. Eberly. *Game Physics*. Elsevier Science Inc., New York, NY, USA, 2003. (Cited on page 58.)

[71] Adrian Boeing and Thomas Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 281–288, New York, NY, USA, 2007. ACM. (Cited on page 58.)

[72] Ian Millington. *Game physics engine development*. The Morgan Kaufmann series in interactive 3D technology. Morgan Kaufmann Publishers, San Francisco, CA, 2007. Cd-rom d'accompagnement contenant le code source. (Cited on page 58.)

[73] D. Halliday, R. Resnick, and J. Walker. *Fundamentals of Physics 9th Edition Volume 2 Chapters 18-37 for So Methodist Univ*. John Wiley & Sons, 2011. isbn: 9781118115626. (Cited on page 59.)

[74] A Carpinteri. *Structural mechanics : a unified approach*. E & FN Spon, London New York, 1997. (Cited on page 60.)

[75] Anh. *Dynamics of Mechanical Systems with Coulomb Friction*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. (Cited on page 60.)

[76] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. (Cited on pages 64 and 80.)

[77] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, 1998. (Cited on page 64.)

[78] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. (Cited on page 64.)

[79] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999. (Cited on page 64.)

[80] L. Rokach. *Data Mining with Decision Trees: Theory and Applications*. Series in machine perception and artificial intelligence. World Scientific Publishing Company, Incorporated, 2008. (Cited on page 65.)

[81] H. Peng, Fulmi Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, Aug 2005. (Cited on page 65.)

[82] S.J. Russell and P. Norving. *Artificial Intelligence: A Modern Approach*. Series in Artificial Intelligence. Printice Hall, 2010. ch. 6, p208-210. (Cited on page 69.)

[83] G Parmigiani and L Inoue. *Decision Theory: Principles and Approaches*. Wiley Series in Probability and Statistics. Wiley, 2009. (Cited on page 70.)

[84] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. (Cited on page 80.)

[85] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, 2004. (Cited on page 80.)

[86] T.G. Dietterich. Ensemble methods in machine learning. In *Int. Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000. (Cited on page 86.)

Publications *in the series*
Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.
2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.
3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.
4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.
5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.
6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.
7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.
8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.
9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.
10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.
11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.
12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.
13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.
14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.
15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.
16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production*. Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing*. Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals"*. Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots*. Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems*. Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod*. Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics*. Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber.* Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment*. Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision*. Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects*. Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding*. Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner. – Motion control, data capturing and path planning.* Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle*. Doctoral Dissertation.

31. Andreasson, Henrik (2008) *Local Visual Feature based Localisation and Mapping by Mobile Robots*. Doctoral Dissertation.

32. Bouguerra, Abdelbaki (2008) *Robust Execution of Robot Task-Plans: A Knowledge-based Approach*. Doctoral Dissertation.

33. Lundh, Robert (2009) *Robots that Help Each Other: Self-Configuration of Distributed Robot Systems*. Doctoral Dissertation.

34. Skoglund, Alexander (2009) *Programming by Demonstration of Robot Manipulators*. Doctoral Dissertation.

35. Ranjbar, Parivash (2009) *Sensing the Environment: Development of Monitoring Aids for Persons with Profound Deafness or Deafblindness*. Doctoral Dissertation.

36. Magnusson, Martin (2009) *The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. Doctoral Dissertation.

37. Rahayem, Mohamed (2010) *Segmentation and fitting for Geometric Reverse Engineering. Processing data captured by a laser profile scanner mounted on an industrial robot*. Doctoral Dissertation.

38. Karlsson, Alexander (2010) *Evaluating Credal Set Theory as a Belief Framework in High-Level Information Fusion for Automated Decision-Making*. Doctoral Dissertation.

39. LeBlanc, Kevin (2010) *Cooperative Anchoring – Sharing Information About Objects in Multi-Robot Systems*. Doctoral Dissertation.

40. Johansson, Fredrik (2010) *Evaluating the Performance of TEWA Systems*. Doctoral Dissertation.

41. Trincavelli, Marco (2010) *Gas Discrimination for Mobile Robots*. Doctoral Dissertation.

42. Cirillo, Marcello (2010) *Planning in Inhabited Environments: Human-Aware Task Planning and Activity Recognition*. Doctoral Dissertation.

43. Nilsson, Maria (2010) *Capturing Semi-Automated Decision Making: The Methodology of CASADEMA*. Doctoral Dissertation.

44. Dahlbom, Anders (2011) *Petri nets for Situation Recognition*. Doctoral Dissertation.

45. Ahmed, Muhammad Rehan (2011) *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction*. Doctoral Dissertation.

46. Riveiro, Maria (2011) *Visual Analytics for Maritime Anomaly Detection*. Doctoral Dissertation.

47. Rashid, Md. Jayedur (2011) *Extending a Networked Robot System to Include Humans, Tiny Devices, and Everyday Objects*. Doctoral Dissertation.

48. Zain-ul-Abdin (2011) *Programming of Coarse-Grained Reconfigurable Architectures*. Doctoral Dissertation.

49. Wang, Yan (2011) *A Domain-Specific Language for Protocol Stack Implementation in Embedded Systems*. Doctoral Dissertation.

50. Brax, Christoffer (2011) *Anomaly Detection in the Surveillance Domain*. Doctoral Dissertation.

51. Larsson, Johan (2011) *Unmanned Operation of Load-Haul-Dump Vehicles in Mining Environments*. Doctoral Dissertation.

52. Lidström, Kristoffer (2012) *Situation-Aware Vehicles: Supporting the Next Generation of Cooperative Traffic Systems*. Doctoral Dissertation.

53. Johansson, Daniel (2012) *Convergence in Mixed Reality-Virtuality Environments. Facilitating Natural User Behavior.* Doctoral Dissertation.

54. Stoyanov, Todor Dimitrov (2012) *Reliable Autonomous Navigation in Semi-Structured Environments using the Three-Dimensional Normal Distributions Transform (3D-NDT).* Doctoral Dissertation.

55. Daoutis, Marios (2013) *Knowledge Based Perceptual Anchoring: Grounding percepts to concepts in cognitive robots*. Doctoral Dissertation.

56. Kristoffersson, Annica (2013) *Measuring the Quality of Interaction in Mobile Robotic Telepresence Systems using Presence, Spatial Formations and Sociometry.* Doctoral Dissertation.

57. Memedi, Mevludin (2014) *Mobile systems for monitoring Parkinson's disease*. Doctoral Dissertation.

58. König, Rikard (2014) *Enhancing Genetic Programming for Predictive Modeling*. Doctoral Dissertation.

59. Erlandsson, Tina (2014) *A Combat Survivability Model for Evaluating Air Mission Routes in Future Decision Support Systems*. Doctoral Dissertation.

60. Helldin, Tove (2014) *Transparency for Future Semi-Automated Systems. Effects of transparency on operator performance, workload and trust.* Doctoral Dissertation.

61. Krug, Robert (2014) *Optimization-based Robot Grasp Synthesis and Motion Control*. Doctoral Dissertation.

62. Reggente, Matteo (2014) *Statistical Gas Distribution Modelling for Mobile Robot Applications*. Doctoral Dissertation.

63. Längkvist, Martin (2014) *Modeling Time-Series with Deep Networks*. Doctoral Dissertation.

64. Hernández Bennetts, Víctor Manuel (2015) *Mobile Robots with In-Situ and Remote Sensors for Real World Gas Distribution Modelling*. Doctoral Dissertation.

65. Alirezaie, Marjan (2015) *Bridging the Semantic Gap between Sensor Data and Ontological Knowledge*. Doctoral Dissertation.

66. Pashami, Sepideh (2015) *Change Detection in Metal Oxide Gas Sensor Signals for Open Sampling Systems*. Doctoral Dissertation.

67. Lagriffoul, Fabien (2016) *Combining Task and Motion Planning*. Doctoral Dissertation.

68. Mosberger, Rafael (2016) *Vision-based Human Detection from Mobile Machinery in Industrial Environments*.

69. Mansouri, Masoumeh (2016) *A Constraint-Based Approach for Hybrid Reasoning in Robotics*.

70. Albitar, Houssam (2016) *Enabling a Robot for Underwater Surface Cleaning*.

71. Mojtahedzadeh, Rasoul (2016) *Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection*.