Robotic Part Manipulation for 3D-Print Post-Processing

Joyce Lim Xin Yan

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

18 Aug. 2023
. . . .

Date


. . . .

Joyce Lim Xin Yan

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

18 Aug. 2023

....

Date

....

Assoc Prof Pham Quang Cuong

Authorship Attribution Statement

This thesis contains material from 4 papers submitted / published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as H. Nguyen, N. Adrian, **J. X.-Y. Lim**, M. Salfity; W. Allen, Q.-C. Pham, "Development of a robotic system for automated decaking of 3D-printed parts," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 31 May to 31 Aug 2020. DOI: 10.1109/ICRA40945.2020.9197110

The contributions of the co-authors are as follows:

- A/Prof Q.-C. Pham provided the initial project direction.
- H. Nguyen prepared the manuscript and overall implementation.
- N. Adrian prepared the physical set up and supporting algorithm.
- I prepared the network training and data collection.
- H. Nguygen, N. Adrian and I were involved in all discussions throughout the study.
- M. Salfity and W. Allen provided suggestions for the manuscript.

Chapter 4 is published as **J. X.-Y. Lim** and Q.-C. Pham, "Automated post-processing of 3D-printed parts: Artificial powdering for deep classification and localization," in *Virtual and Physical Prototyping*, vol. 16, pp. 333-346, 2021. DOI: 10.1080/17452759.2021.1927762

The contributions of the co-authors are as follows:

- A/Prof Q.-C. Pham provided directions during the stufy and revised the manuscript.
- I proposed the initial idea and prepared the manuscript.
- I designed the simulations, implemented the algorithm, performed all data collection, network training and experimental analysis.

Chapter 5 is submitted as **J. X.-Y. Lim** and Q.-C. Pham, "Fingerpad customization with set boolean operators for precise and versatile grasping," in *IEEE Robotics and Automation Letters*, 2023.

The contributions of the co-authors are as follows:

- A/Prof Q.-C. Pham provided initial direction, additional suggestions during the study and revised the manuscript.
- I prepared the manuscript.
- I proposed the method, implemented the algorithm, designed the experiments, executed all physical demonstrations and performed experimental analysis.

Chapter 6 is submitted as **J. X.-Y. Lim** and Q.-C. Pham, "Grasping, part identification and pose refinement in one shot with a tactile gripper," in *IEEE Robotics and Automation Letters*, 2023.

The contributions of the co-authors are as follows:

- A/Prof Q.-C. Pham provided directions during the study and revised the manuscript.
- I proposed the initial idea and prepared the manuscript.
- I proposed the method, implemented the algorithm, designed the experiments, executed all physical demonstrations and performed experimental analysis.

18 Aug. 2023
....

Date

....

Joyce Lim Xin Yan

Acknowledgements

I wish to express my greatest gratitude to my spouse, family, and friends who have given me their utmost encouragement and support throughout this journey. My heartfelt appreciation also goes to my NTU supervisor and both HP managers for their guidance. I am also very grateful to my fellow colleagues, my thesis advisory committee, and my examiners for their directions.

Abstract

The rise in Additive Manufacturing (AM) comes with unique opportunities and challenges. 3D-Print (3DP) allows rapid changes to part design, mass production, and massive part customization in manufacturing, which meets industrial manufacturing needs for customized parts such as dental moulds, shoe insoles, or engine vanes in turbo-machinery. However, a major drawback to 3DP that prevents its wide application stems from the bottleneck in post-production processes. Post-production tasks rely heavily on manual labor, which is tedious, repetitive and exposes the operators to hazardous substances. Some examples of post-processing are part cleaning, painting, sorting, and packing. Therefore, it is desirable to introduce robotics and automation in 3DP post-processing. However, current automated post-processing solutions are scarce and limited to specific materials or tasks. The opportunity for massive part customization also comes with unique challenges for the existing production paradigm of robotics applications. Two main challenges are the unique environment due to the presence of powder in AM, and the production of non-standard, geometrically complex parts due to customization. Hence, there is a need to develop generalized robotics solutions for implementation in end-to-end 3DP post-processing. As there are many AM technologies, this dissertation focuses only on powder-based AM processes.

First, to demonstrate the feasibility of robotics in automated part cleaning, we developed a fully functional robotic prototype for the automated removal of residue powder on 3DP parts that mimics part cleaning by a human with a brush. Second, to support robot perception in 3DP post-processing, we proposed a fully automated vision pipeline for deep classification and localization of parts covered in powder, which is the first method that artificially simulates unique 3DP powder accumulation on the objects. Third, to support robot grasping and manipulation for batch-produced customized 3DP parts, we present an automated gripper customization method that designs versatile gripper fingers to grasp and manipulate a batch of objects resting at various positions with high precision. Fourth, to support part identification, grasping, and manipulation of unique parts with similar

features, we introduce a method of pattern augmentation on 3DP parts, which has never been considered before, to perform grasping, part identification, and pose refinement in one-shot with a tactile gripper. With these contributions, we also sketched some possible directions for advancing the implementation of robotics and automation for 3DP post-processing.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Acronyms

AM	Additive Manufacturing
3D	3-Dimensional
3DP	3D Print
HP	HP Inc.
AI	Artificial Intelligence
F/T	Force-Torque
STL	Stereolithography
CAD	Computer-aided design
AP	Average Precision
mAP	mean Average Precision
IoU	Intersection over Union

Chapter 1

Introduction

1.1 Background

AM is a process of joining materials to make objects from 3D model data, usually layer by layer rather than subtractive manufacturing technologies [5]. The initial commercial use of AM emerged in 1987 with the concept of STL [6], where users were able to generate a physical object from digital data [7]. Since then, AM has been increasingly popular due to its ability to manufacture unique parts. AM technologies have evolved rapidly and they form 3 main categories: solid-based, powder-based, and liquid-based systems [7]. Liquid-based systems include technologies such as Stereolithography Apparatus (SLA) while solid-based systems include the Fused Deposition Modelling (FDM) technology [7]. In powder-based 3DP systems, metal or plastic powder can be used. These methods include Selective Laser Melting (SLM) [7, 8], Selective Laser Sintering (SLS) [7, 8] and HP Multi Jet Fusion (MJF) [7, 9]. AM technologies are highly competitive as large volumes of parts, customized objects, and rapid design changes are made achievable with AM, a major drawback stems from the need for post-production processes, such as part cleaning, painting, sorting, and packing.

Currently, most post-production treatment relies heavily on manual labor, which is tedious, repetitive, and exposes operators to hazardous substances. Manual labor also creates a bottleneck on the utilization rate of 3D printers, as a leading manufacturer could only utilize its 3D printers up to 60% due to manual post-processing [10]. Therefore, it is desirable to introduce robotics and automation in 3DP post-processing.

Automated post-processing solutions are scarce and currently limited to specific materials or tasks. Part cleaning solutions include using a tumbler to induce vibrations and release trapped powder, which was also seen in commercialized products [11]. This tumbler method could also be extended to surface finishing [12]. However, an automotive company mentioned that its post-processing equipment does not have the capacity to match that of 3D printers running over multiple days [10]. In addition, according to a relatively recent report in 2018 [13], 27% of the total cost to produce a model can be attributed to costs related to post-processing, which include part breakage cost. Hence, although the benefits of AM are apparent, post-processing is limiting, costly, and tedious. These bottlenecks make it desirable to introduce robotics and automation for 3DP post-processing to meet the rising need for automation.

Robotics and automation for 3DP post-processing tasks also require basic techniques such as perception, recognition and pose estimation, grasping, and manipulation. Although these techniques have been widely researched, the nature of the 3DP is unique which challenges the existing paradigm of robotics application. The current robotics paradigm does not consider the unique environment resulting from AM, where the main challenges stem from the powder during printing and unique objects resulting from massive part customization. Hence, there is a need for new robotics paradigms that can support robotics and automation in 3DP post-processing.

1.2 Research objectives and contributions

We aim to develop generalized techniques on robot perception, grasping, and manipulation that can aid the implementation of robotics and automation in 3DP post-processing. We focus only on AM powder-based processes with parts printed in PA11/12 which are Nylon powders. Figure 1.1 illustrates the area of contributions that correspond to our research objectives.

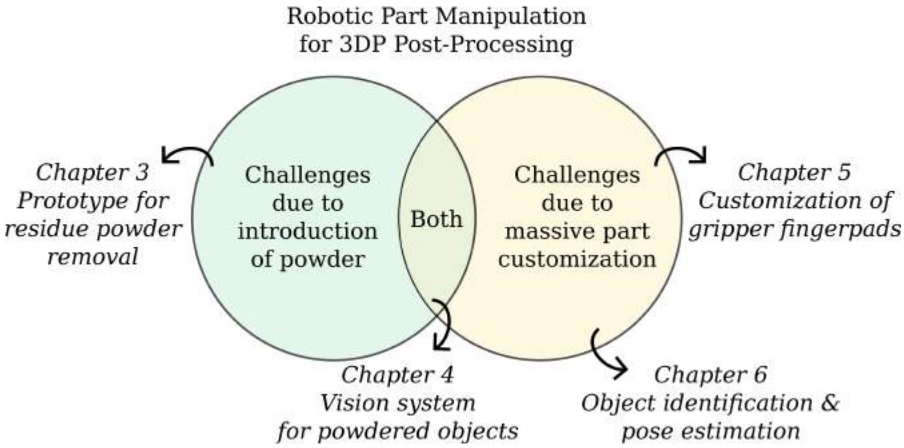


FIGURE 1.1: Area of contributions.

Our research objectives and contributions are:

1. **Demonstrate the feasibility of robotics in 3DP post-processing:** Development of a novel robotic system prototype for automated cleaning of residue powder, known as decaking, of 3DP parts in a fast and efficient manner, which is the first robotic prototype capable of performing 3DP part cleaning (Chapter 3).

2. **Introduce perception for 3DP parts covered in powder:** Development of a fully automated vision pipeline for deep classification and localization of 3DP parts covered in powder, which is the first approach to artificially simulate the powder accumulation unique to 3DP, on the objects (Chapter 4).

3. **Support robot grasping and manipulation of *batch-produced* customized parts:** Automated gripper customization method that designs versatile gripper fingers to grasp and manipulate a batch of objects resting at various positions with high precision. We also introduced a novel geometric grasp quality measure (Chapter 5).

4. **Support identification and manipulation of *unique parts with similar features*:** Implemented the novel idea of pattern augmentation on 3DP parts, which has never been considered before, to conduct grasping, part identification, and pose refinement in one shot with a tactile gripper, where parts with similar features could be distinguished (Chapter 6). Our method achieved high pose estimation accuracy and success rate in tasks that mimic automated sorting and packing.

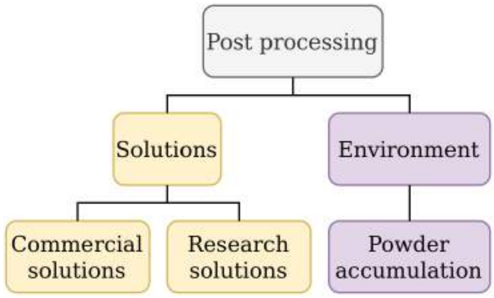
1.3 Organization of thesis

Chapter 2 reviews previous related work on various topics. Chapter 3 introduces in detail a robotic system for automated decaking of 3DP parts, including the integration of hardware and software components to obtain a functional prototype. Chapter 4 presents in detail a fully automated pipeline for deep classification and localization of powdered parts, including the simulation of artificial powder on CAD models to improve detection performances. Chapter 5 proposes a method to customize gripper fingerpads that can achieve precise and versatile grasping for 3DP objects with complex geometries, together with a novel geometric grasp quality measure. Chapter 6 introduces a method to augment unique patterns on 3DP objects that can achieve grasping, part identification, and pose refinement in one shot. Chapter 7 concludes and sketches some future directions.

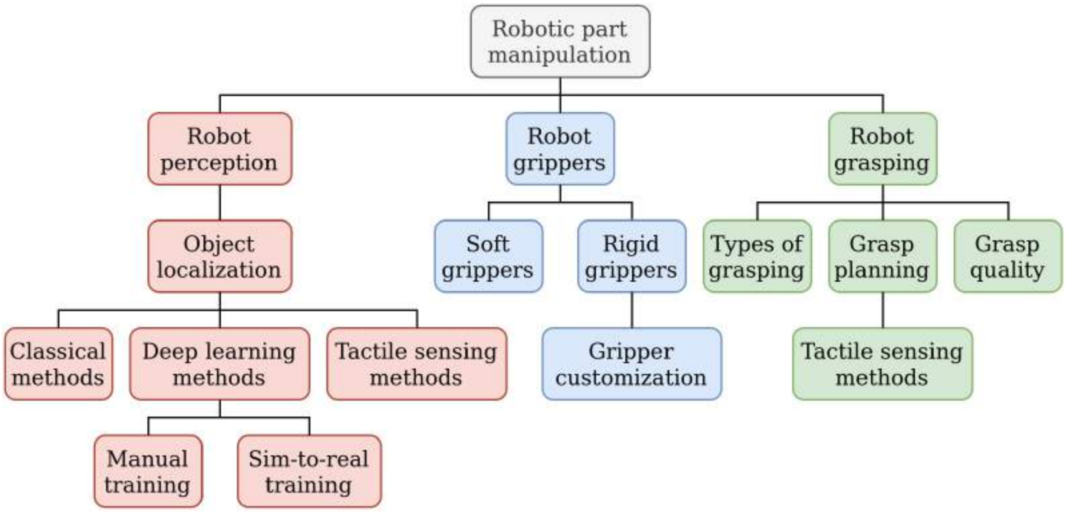
Chapter 2

Literature review

This section reviews relevant literature on several topics. As we aim to propose generalized techniques that can aid the implementation of robotics and automation in 3DP post-processing, we review existing automated solutions for post-processing (Section 2.1) and also study the powder phenomena (Section 2.2) since our focus is on 3D-printing by powder-based processes. Additionally, as the unique environment of 3DP challenges the existing production paradigm of robotics applications, we review previous studies on robot perception (Section 2.3), robot finger design (Section 2.4), robot grasping (Section 2.5) and tactile sensing in robotics (Section 2.6). Figure 2.1 illustrates the organization of relevant topics.



(A)



(B)

FIGURE 2.1: Organization of related topics for (A) 3DP post-processing; (B) Robotics part manipulation.

2.1 Existing automated post-processing

There are commercial automated post-processing stations for cleaning of 3DP parts after printing [11]. In addition, recent research has been done to automate post-processing such as surface finishing [12] or removal of support material [14, 15].

2.1.1 Commercial automated post-processing stations

Bealmer, an online platform that connects engineers and designers to a global network of professional 3D printing services, covered several commercial post-processing equipment and methods in an article [11]. For the solid-based FDM technology, there could be an option to choose soluble support materials depending on the 3D printers. Siemens and Solukon collaborated to produce a post-processing machine (SFM-AT800S) for cleaning residue metal powder, by using a centrifuge rotating the part in three dimensions. This would work for powder-based technologies such as SLS. Two companies, FormLabs and Carbon, have developed machines catered for SLA or other photopolymerization printing techniques to clean the parts after printing.

2.1.2 Research solutions on automated post-processing

A. Ju *et al* proposed the concept of using a tumbler for surface finishing [12]. They mainly focused on nylon parts produced by HP MultiJet Fusion printers and attempted to improve the aesthetics of these parts due to dull and rough surfaces attributed to semi-fused powder on the surfaces. Their vibratory tumbler was capable of finishing processes such as smoothing, coating, and dyeing, by a series of tumbling processes using different media. Figure 2.2 shows the concept of a similar rotating tumbler.

FIGURE 2.2: Automatic bead blasting using a rotating tumbler [1].

S. Nelaturi *et al* proposed a method for automatic support removal [14] using motion planning of machining equipment. They introduced an automatic spatial planning approach to obtain a sequence of support removal operations and an execution path, by explicit sampling of the free space. The goal for the operations was to fracture the contact regions between each support component and the part while including collision avoidance. The paths were executed by multi-axis machining equipment, resulting in parts without support structures, and can be sent for traditional machining to produce the desired design.

S. Raikair *et al* proposed a method for automatic support removal by etching [15]. They introduced a self-terminating etching process for Ti-6Al-4V parts which can remove support and trapped powder while also improving the surface finish for both exterior and interior features. The parts were subjected to a series of heating and chemical processes while preserving the actual parts, to obtain automatically obtain the post-processed parts.

2.1.3 Limitations of current automated post-processing

The issue of post-processing is that it still requires costly and tedious manual labour [16, 17], with the valid concern of exposing hazardous substances to human operators in certain steps of the production line. Although some methods of automated post-processing have been introduced, they require human interventions such as loading, unloading, packing, or sorting. To enable robots to efficiently undertake post-processing tasks in an end-to-end manner, advances in robot technologies would be required.

2.2 Powder phenomena

The presence of powder in a robotic workspace creates additional challenges, especially for robot perception. Caked powder accumulated on the objects would drastically change the original features of these objects, thus feature-based deep-learning classification methods trained on the objects without powder would not be able to perform that well. Hence, a method to simulate the powder accumulation on 3DP parts would be desirable.

We look at some studies that involve powder or the powder bed in AM technologies, such as the study of the spatter distribution on the powder bed [18], characterizing powder materials using computer vision [19], and defect detection in laser-powder bed fusion [20]. However, the simulation of powder distribution on printed parts has yet to be investigated. Thus, this section aims to study the relevance of powder accumulation on the surface of 3DP parts with a similar medium, such as falling snow.

2.2.1 Granular media simulation

Granular media such as snow and 3DP powder visually look very similar, especially for nylon powder which is also white in color. Rendering powder on the surfaces of printed parts can be related to that of snow falling on outdoor objects [21–23]. Other studies have proposed using a general model for falling snow [21], using density distribution of snow based on its weight to model snow [22], and accumulating snow and a user-controllable elastoplastic constitutive model integrated with a hybrid Eulerian and Lagrangian Material Point Method for snow simulation [23].

In [21, 22], physics was used to model snow particles while in [23], solid and fluid simulators were coupled together to simulate a wide variety of snow behaviors. Other simulations of granular media, such as sand, can include computer simulation [24, 25] or analytical modeling simulation [26, 27] on the physics of granular media.

2.2.2 Comparison on powder accumulation with similar granular media

(A) Snow simulation in [22]

(B) 3DP powder accumulation

FIGURE 2.3: Difference between falling snow accumulation and 3DP powder accumulation.

However, we observed that powder accumulation on the surface of 3DP parts results from a different physical process and is visually very distinct from granular media such as falling snow which is a natural process Figure 2.3. In addition, other granular media simulations involve physical force and shear which would not be accurate in the case of powder accumulation on 3DP parts. This is because the manner in which the operators remove the parts from the powder bed or the way that the parts are packed for printing could affect the powder accumulation. These also lead to variations in powder accumulation even for the same type of parts, resulting in a unique distribution for every part. Hence, current methods that simulate granular media may not be suitable for simulating powder accumulation on 3DP parts.

2.3 Robot perception

Robot perception is a crucial aspect in introducing robotics and automation in 3DP post-processing as information on the environment must be made available to perform tasks requiring object manipulation. This section discusses image-based classical and deep-learning methods for robot perception.

2.3.1 Classical methods on object localization

Traditionally, object detection consists of feature extraction and classification methods [28]. Popular feature extraction methods include Scale-Invariant Feature Transform (SIFT) [29], fast binary descriptor (ORB) [30] and Histograms of Oriented Gradients (HOG) [31]. After extracting the features, classification models such as Support Vector Machines (SVM) [32] and AdaBoost [33] could be used to classify the objects. In addition, stereo vision [34] could be used to extract the 3D information of the classified object to compute the localization of the object.

2.3.2 Deep learning methods on object localization

The growth of AI in robotics has been greatly accelerated due to promising results on object detection and classification attributed to the introduction of using Regions of Interest (ROI) in CNNs (R-CNN) [35]. Faster R-CNN was then proposed, by using a Region Proposal Network (RPN) to predict ROI proposals from features in a query image, then predict object classes and bounding-box regression by ROI-Pooling on every proposal. Other methods using CNNs were also introduced, such as SSD [36] and YOLO [37].

Instance segmentation, Mask R-CNN [38], was proposed as an improvement to object classification methods, where masks that enclose different instances of objects were produced on top of the bounding boxes and classification labels. An example of instance segmentation is shown in Figure 2.4. Mask R-CNN was built upon the detector in Faster R-CNN, by performing classification with bounding-box regression and generating segmentation masks for every proposal in parallel.

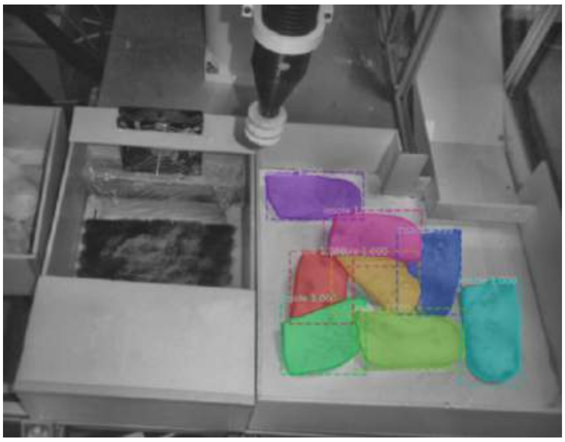


FIGURE 2.4: Instance segmentation of shoe insoles covered in 3DP powder.

Instance segmentation and deep learning have been used to obtain the pose estimation of objects and people. The 6D pose estimation can be computed by RGB images [39–41] or RGB-D images where the point cloud of the scene is also used [42–44]. Methods using RGB-D images as input can also incorporate pose refinement by point cloud registration methods Figure 2.5 such as Iterative Closest Point (ICP) [45], FilterReg [3] and Bayesian coherent point drift [46].

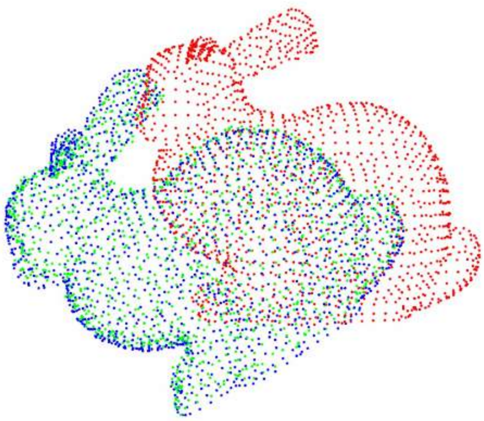


FIGURE 2.5: A library [2] that implemented FilterReg [3]. The initial, target and resulting point clouds are shown in red, green, and blue respectively.

2.3.3 Introduction of synthetic data for deep learning methods

Although deep learning approaches are very good in object detection and classification, a major drawback to deep learning is the significant size of the dataset required to train networks, even with transfer learning to retrain the classifier [47]. These methods are also based on feature extraction thus the networks are object dependent, meaning that a network has to be retrained for different objects. Thus, large datasets made up of objects, animals and people in realistic scenes were developed, such as COCO [48], VGG-Face [49] and Pascal-VOC [50]. However, the customization of 3DP parts results in parts with varying features and geometries, hence these datasets are not applicable and new data on 3DP parts has to be collected to conduct object classification. Due to the large diversity of 3DP parts, it is not practical to collect real-world data for training deep neural networks as it can be extremely tedious and time-consuming. Hence, using synthetic data may be a good option for network training.

Recent approaches in using synthetic training data proved to be rather successful [51–56]. In particular, the creation of synthetic images by domain randomization using CAD models in [53–56] seemed superior compared to composition [51, 52]. An illustration of the methods is shown in Figure 2.6.

(A) Composition [51] render

(B) Domain randomization render

FIGURE 2.6: Comparing synthetic images of piles rendered by composition and domain randomization.

2.3.4 Synthetic rendering by composition

Generation of synthetic images using composition does not require the CAD model of the object. E. Buls *et al* [51] worked on generating a pile of objects using composition by capturing different views of real objects and stitching them together, however, the detection results were not favorable as the accuracy of correctly detected objects ranged from 10% to 29% for different objects.

D. Dwibedi *et al* [52] presented another approach using composition to rapidly synthesize datasets for instance detection, by starting with a set of real images of the objects and background scenes, and the masks of the objects were automatically extracted. The objects were pasted on scenes with different blending to get the final synthetic image. The best performance was 50.8% mAP when trained on a mix of real and synthetic images and tested on an unseen dataset.

2.3.5 Synthetic rendering by domain randomization

Domain randomization is a simple technique for training models purely on simulated images, and these models are able to detect real-world objects with sufficient variability in the simulator [53, 54]. J. Tobin *et al* [53] focused on the task of training neural networks using domain randomization to conduct object localization for robotic grasping in a cluttered environment. They found that for a range of geometric objects, they were able to train a detector that was accurate to 1.5cm in the real world, using only simulated data rendered with simple, generated textures. They also claim that with a sufficiently large texture database to pick from, pre-training the object detector with real images is unnecessary. The authors published another paper [54] on using domain randomization with generative models for robotic grasp planning and achieved 80% success rate on real-world grasp attempts at test time when only trained on random simulated objects.

S. Rajpura *et al* [55] showed that with transfer learning and domain randomization, an effective object detector can be trained almost entirely on a rendered dataset. They applied this strategy to detecting packaged food products clustered in refrigerator scenes and obtained mAP of 24 on a test set with 55 distinct objects of interest and 17 distractor objects. The network was trained on 4000 synthetic images.

M. Danielczuk *et al* [56] proposed a method to automatically generate a synthetic training dataset of 50,000 depth images and 340,000 object masks using simulated heaps of CAD models for a bin-picking task. Domain randomization was applied to the 3D objects, camera poses, and camera intrinsic parameters. The authors modified the Mask R-CNN network to train on both synthetic depth images and real depth images and concluded that the network trained on synthetic depth images outperformed the network trained on real depth images based on the average precision and average recall.

2.3.6 Limitations of using deep learning in 3DP post-processing perception

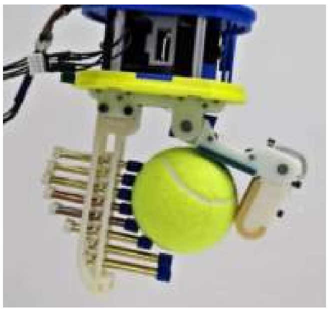
Although a deep learning approach to classify and localize 3DP parts is promising, it cannot be directly applied due to the accumulation of residue powder on the object surfaces, which leads to real powdered parts having shapes and features that are visually different from their CAD models, thus affecting the performance of these feature-based deep learning approaches. As such, there is a need for a specific method that can simulate the distribution of powder on 3D printed parts after printing so that feature-based deep learning classification methods would be able to learn accurate features and provide better detection results.

2.4 Finger design for grasping

Advancements in AM have enabled fast design changes and easy customization of parts, thus creating a large diversity of parts with intricate and complex geometries. To introduce robotics and automation in 3DP post-processing, the robotic gripper design is a fundamental aspect. Yet, current gripper design methods for such customized objects are often manual [57] which can be tedious, and automated methods may not be robust enough for complex and customized objects created by AM. Additionally, manual design methods tend to rely on ad-hoc design intuition rather than rigorous principles that consider grasp quality. Thus, this section reviews recent studies on rigid gripper customization and also non-rigid grippers. Figure 2.7 illustrates several types of grippers.

(A) Rigid customized gripper [58].

(B) Jamming soft gripper [59].



(C) Adaptive gripper [60]

FIGURE 2.7: Illustrations of different types of grippers.

2.4.1 Rigid gripper customization

Pham *et al* [61], appropriate pairs of finger designs are selected from a pre-configured database consisting of simplified geometries. However, complex shapes will result in failed grasping.

Balan *et al* [62], a reconfigurable gripper finger design was proposed, which automatically configures the locations of three cylindrical fingers to obtain a three-point grasp on objects of any shape. However, it is limited to only handling polyhedral shapes.

Velasco and Wyatt [63], a method was proposed to extract the geometry of objects such that fingers generated will enclose the object surfaces, forming a caging grasp. This method was applied in [64, 65] where an end-to-end pipeline to obtain customized grippers was proposed by conducting geometrical analysis, grasp planning, finger design, and experimental verification. However, as the fingers are highly customized to fit the objects at specific poses, it may not be practical in automation since the poses of the objects will often change throughout the automation process, which may lead to failed grasps.

Song *et al* [58], the authors introduced an optimization procedure to cluster various geometries to produce a set of robust finger designs. These designs were used to plan the grasp of several objects by maximizing contact area. However, the grasp planning method requires the center of gravity to be within the parallel fingers, which results in limitations on the size of objects to be grasped as it is dependent on the size of the gripper opening. In addition, unfeasible grasps may be obtained as the grasps were planned after the fingers were designed. This could be observed in cases where an object lies exactly in the position of the planned grasp.

2.4.2 Non-rigid grippers

Soft fingers are versatile as they deform to the local geometry of the object and can better resist external disturbances [66, 67]. For soft grippers, grasps are achieved by three technologies: (1) actuation, (2) controlling gripper stiffness, (3) controlling gripper adhesion [68]. There are also studies on hybrid grippers that combined soft and rigid structures [69–71] to improve fingertip force, actuation speed, friction, or adaptability.

Studies on soft grippers by actuation include conforming pin pads for adaptive grasping [60] and also bio-inspired soft grippers by impactive gripping [72]. Other studies on soft grippers by controlled stiffness include jamming pads that leverage on the natural phenomenon of granular jamming [59, 73] or shape memory materials that have varying stiffness during the transition of phase [74, 75]. Soft grippers with controlled adhesion can achieve large holding forces due to high values of friction forces, by for example, using electro-adhesion to control the electric charges on the interface between the gripper and an object [76], or by suction cups [77].

2.4.3 Limitations of gripper precision and versatility

Although it is possible to manually design grippers for customized parts, it is not recommended as it would be too time-consuming due to the presence of complex geometries. In previous works, customized grippers can achieve precise grasping as the local contours conform well to a specific object. Yet, they lack versatility as the same gripper may not be able to grasp another object, or the same object at a different position. Non-rigid grippers are capable of grasping many different objects as the fingers deform to the local contours of the object, however, they usually lack precision. Thus, precision and versatility are conflicting objectives, yet are highly desirable in robotic grippers for practical applications such as in 3DP post-processing that involves different objects in various tasks.

2.5 Robotic grasping

Robotic grasping is another fundamental aspect in introducing robotics and automation in 3DP post-processing as objects are to be securely grasped to perform object manipulation in tasks.

Secure grasping consists of two primary properties: force closure and form closure grasps which are formed by restraints [78]. The main difference between these two grasps is the reliance on contact friction in force closure grasps, which leads to less number of contact points required to achieve force closure grasps [79]. Force closure grasps rely on the ability of the gripper to squeeze the object tightly so that it creates the capability of resisting external wrenches or disturbances [78]. Form closure grasps are achieved when the robot fingers produce a set of stationary constraints that prevents all motion of the object in grasp [79].

Caging and immobilization grasps are another type of grasping that was proposed because the initial form closure analysis failed to account for the curvature of the grasped object [80, 81]. Thus, immobilization grasps were obtained due to geometrical constraints, where the local motion of the object in grasps would be obstructed by the rigidity of the object and robot fingers [82, 83], which makes these grasps to be insensitive to friction changes [58]. This leads to lesser fingers required than traditional form closure grasps [82]. In addition, immobilization is performed in the configuration space of an object [84], which allows the early definition of form closure to be less restrictive during analysis [82].

2.5.1 Grasp planning methods

The aim of grasp planning is to find potential grasps for an object, by planning initial contact locations. Tools such as Graspit [85] which used shape primitives, or SynGrasp [86] can find contact locations and detect collision for basic flat-fingered or cylindrical grippers. Another proposed method uses decomposition trees to prune a large number of possible grasps into a subspace that is more likely to contain good grasps [87]. In addition, there are also studies on grasp planning in cluttered environments [88, 89]. Learning-based methods were also proposed for grasp planning, such as ambidextrous grasping [90], multi-affordance grasping [91], and cluttered-scene grasping with latent plans [92].

2.5.2 Grasp quality measures

Grasp quality measures are used to determine the optimal grasp when there are multiple possible grasps, which usually stems from grasp planning methods. Classical point-contact quality measures were discussed in [93] for force closure grasps, including analytical methods using grasp wrench space [94] that simplifies force closure grasp analysis but cannot take into account the curvature of object's surface [82].

Surface-contact quality measures were also studied, which can include a surface-contact model that parameterizes the contact area [95], computation of contact profile using solid geometry intersection and barycentric integration [96], or calculation of contact profile using 6D friction wrench or friction cone [67, 97].

2.5.3 Limitations of grasping 3DP parts

Many grasp planners and grasp quality measures are proposed in the direction of the existing robotics paradigms, which are difficult to implement in the unique 3DP environment, where massive part customization is made possible with AM, hence indicating that the objects to be grasped could have complex geometries or could be changed very frequently. For analytical models, it would be difficult to apply to objects with complex geometries as they were usually derived from objects with simple geometries. Although soft fingers may be a good choice due to versatility, they generally lack precision as discussed in Section 2.4.3.

2.6 Robotic tactile sensing

Tactile sensing in robotics is a wide area of research motivated by the "sense of touch" in humans, which can be categorized into (1) perception for action such as grasp control and dexterous manipulation, or (2) action for perception such as exploration and object recognition [98]. Traditionally, sensors that detect force and pressure distributions have been used. Recently, vision-based tactile sensors that measure contact geometry have been introduced. This section discusses the use of both types of tactile sensors for robot perception.

2.6.1 Tactile force sensors

Tactile force sensors can provide force, torque, pressure, and shear measurements that can be used to estimate shape [99–101], surface texture [102, 103] and detect slippage [104, 105] so that grasping, dexterous manipulation, contact point estimation and curvature measurement can be performed [98].

Tactile force sensors for dexterous manipulation can include rotating an object in hand by reinforcement learning with only touch sensors [106], or with additional vision sensors [107]. Other recent studies on grasping and object localization involve proposing a system to use sparse tactile feedback from fingertip touch sensors on a dexterous hand to localize, identify and grasp novel objects without any visual feedback and perform object retrieval from an occluded bin [108], or combining visual and force feedback to create an action-controlled model to conduct packing of items in a box [109].

The use of tactile force sensors is very versatile. There are also studies on using tactile sensors to recognize alphabets such as Morse Code and Braille [110], and also using reinforcement learning for a robot to type on a Braille keyboard [111].

2.6.2 Tactile vision sensors

Vision-based tactile sensors have been widely incorporated in robotics research for object localization, pose estimation, and object shape exploration as they can provide valuable information on the contact geometry of the object for robot tasks.

Some examples of vision-based tactile sensors are the GelSight [112], GelSlim [113], and Digit [114] sensors, which claim to be more informative than traditional tactile sensors that detect force or pressure distributions [98, 115] because local force and shear can be inferred from the high-resolution tactile image of the contact geometry. An illustration of GelSight wedge [4] is shown in Figure 2.8.

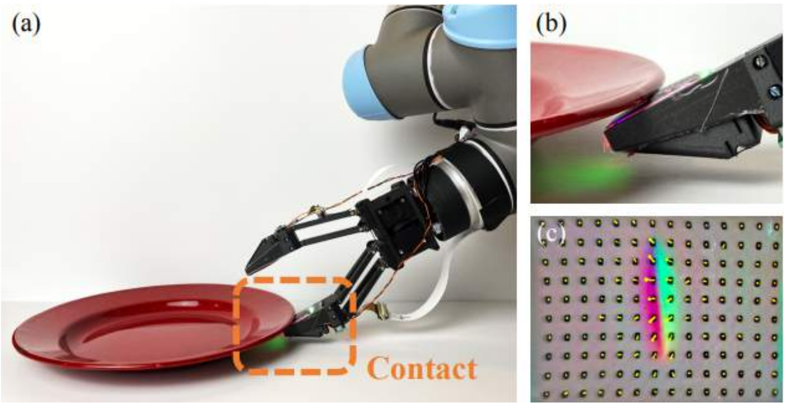


FIGURE 2.8: Illustration of GelSight wedge [4]. (A) A GelSight wedge sensor in contact with a plate; (B) Close-up view of contact; (C) Captured tactile imprint.

Recent works that implemented these sensors with learning-based approaches such as in [116], where the shape of the object was reconstructed from tactile imprints to identify and localize the object for in-hand manipulation, and in [117], where object pose estimates were determined using geometric contact rendering. Other works include using a network trained on simulated contact shapes to obtain the pose distribution [118] and also object recognition by multi-modal associations [119].

Vision-based tactile sensors had also been included in studies for pose estimation such as using an active visuo-tactile point cloud registration for pose estimation between sparse point clouds computed by filter-based methods [120], or combining external cameras and vision-based tactile sensors to conduct visual servoing and localization that could improve the estimation accuracy [121].

Object shape estimation [122, 123] is another implementation of vision-based tactile sensors. In [122], the authors aim to plan grasps by exploration from multiple touches and also claim that an initial grasp attempt based on the initial guess of the overall object shape is capable of providing information of the far side of the object which enables shape estimation, that allows probabilistic approaches to determine the next contact point or grasp location.

Other studies that use vision-based tactile sensors can also include using the images generated from these sensors to analyze and visualize 3D features and infer the stroke sequence of written signatures and toner inking [124].

2.6.3 Limitations of vision-based tactile perception

Vision-based tactile sensors are deemed to be an improvement to traditional force sensors due to the availability of contact geometry and force estimation. However, pose estimation errors in previous works are usually too large to be used for practical tasks when only vision-based tactile sensors are used, as the error ranges from 5mm to 60mm in [117] and the main dimension error was around 5% for reconstructed known objects in [116].

In addition, a survey on robot tactile perception noted that high-accuracy localization might not have been achieved with vision-based tactile sensors [125]. Another survey observed that often easier to provide data from contact-based interactions than to pre-define an accurate analytical model [126] thus many methods tend to be data-driven and object dependent which poses certain challenges during practical implementations, especially with the unique massive part customization capability imposed by 3DP.

Two possible key challenges in using vision-based tactile sensors are: (1) information provided by a sensor is very much limited due to its small area that cannot achieve reasonable feature matching [121, 125], and (2) contact non-uniqueness, where contact is ambiguous due to resemblance to other contacts from another pose of the same or different object, which was illustrated in [118, 119].

Chapter 3

Robotic system for automated decaking of 3DP parts

3.1 Introduction

AM technologies are increasingly competitive and capable of mass manufacturing. However, as reviewed in Section 2.1, limitations on current post-processing technologies and the use of manual labor have become a significant bottleneck in mass manufacturing. In powder-based technologies such as HP MJF, cleaning of residue powder on the parts is required after printing. After the parts are removed from the powder bed, there will be unfused powder stuck on the part surfaces. Parts covered in powder, or with residue powder on their surfaces are defined as *powdered parts*. The removal of residue powder is known as decaking, which is usually done manually (Figure 3.1). Manual decaking is costly and tedious, hence, we introduce, for the first time to our knowledge, a robotic system for automated decaking of 3DP parts (Figure 3.2). In this chapter, our object of interest is the shoe insole[1]. The shoe insole can be seen in the origin container in Figure 3.2, and a close-up view in Figure 3.3.

[1]The author has been granted permission by Footwork Podiatry Laboratory to use their shoe insole design and images in this report.

FIGURE 3.1: An operator removing residue powder from 3DP part.

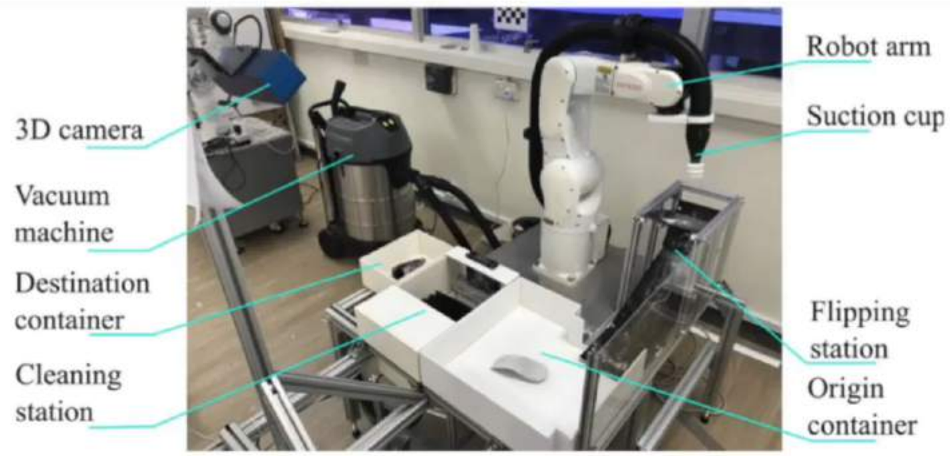
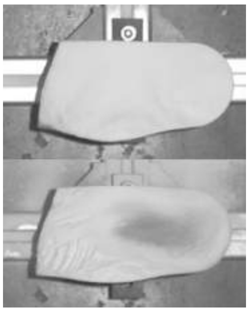


FIGURE 3.2: Our proposed robotic system for automated removal of residue powder. A video of the actual robotic decaking process can be viewed at `https://youtu.be/0QJvNcf2s6s`.

Our system performed the following steps:

1. Localize a part in the origin box.
2. Pick a powdered part from the bin with a suction cup.
3. Remove residue powder on the underside of the powdered part by rubbing it on a brush.
4. Flip the half-clean powdered part by placing it onto the flipping station.
5. Localize the part in the flipping station.
6. Clean the remaining side of the powdered part by rubbing it on a brush.
7. Place the cleaned part in the destination container.



(A) Caked shoe insoles

(B) Cleaned shoe insoles

FIGURE 3.3: Illustrations of caked and cleaned shoe insoles.

Our steps remain general so that they can be applied to other 3DP processes with mostly flat parts that are similar to the shoe insole in our experiment. A video of the actual robotic decaking process can be viewed at `https://youtu.be/0QJvNcf2s6s`. The caked shoe insoles are printed in PA12, which is a white Nylon powder that turns dark grey when printed. Examples of the caked and cleaned shoe insoles are shown in Figure 3.3.

The remainder of the chapter is organized as follows. We present the robotic system pipeline in Section 3.2, and the hardware design and software system in Section 3.3 and Section 3.4 respectively. In Section 3.5, we ran decaking experiments and evaluated the system performances to demonstrate the feasibility of automated decaking. We also sketched some directions for future research in Section 3.6.

3.2 Robotic system pipeline

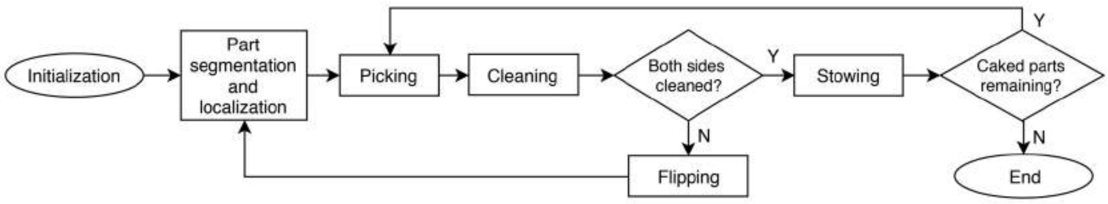


FIGURE 3.4: Pipeline of system with a series of modules for perception and cleaning.

The overview of the system pipeline is shown in Figure 3.4. Each module is customizable and can be fine-tuned to clean a specific object. We have reviewed relevant literature in on robot perception in Section 2.3. However, the task presents unique difficulties:

1. *Perception:* Powdered parts contain unpredictable amounts of residue powder, overlap each other in the bin and are mostly white in color, which may make part detection and localization particularly challenging.

2. *Manipulation:* Residue powder and parts have different physical properties which may affect the manipulation of the powdered part.

3. *Contact tracking:* Part may dislodge from the manipulator when brushing.

Hence, to address these issues, the system integrated hardware and software for 3D perception, manipulation, and force control, which will be elaborated in Section 3.3 and Section 3.4.

3.3 Hardware design overview

We design our system such that it could be scalable and could eventually tackle the problem of automated post-processing of 3DP parts in a real scenario with an advantageous cost. The main components of our system are shown in Figure 3.2 which consists of:

- Denso VS060, a six degrees of freedom industrial manipulator
- ATI Gamma Force-Torque (F/T) sensor
- Ensenso 3D camera (N35-802-16-BL)
- Suction system powdered by a Karcher vacuum machine (NT70/2)
- Cleaning station
- Flipping station

All computations were done on a computer with Intel Xeon E5-2630v3, 64GB RAM.

3.3.1 Suction system

A suction system was used to grasp and manipulate the parts, as it provides versatility and good performances [77]. A suction cup, mounted on the robot end-effector, was connected to a vacuum machine by using a flexible hose. The suction system also provided the advantage of recovering residue powder during cleaning and the collected powder can be recycled for printing. This system was designed to generate both high vacuum and high airflow rates to provide sufficient force to lift parts and to maintain a firm hold of parts during brushing, by using a high-power (2400W Karcher NT70/2) vacuum machine. To remotely control the vacuum machine, we customized a 12V control output from the Denso RC8 controller for binary ON/OFF control.

3.3.2 Camera

To have a robust perception system, we manually optimized the camera location to maximize the view angles of the bins, while avoiding occlusions due to the robot arm during operations such as picking or cleaning. The camera also was placed in a location where collisions could be avoided. The camera used was the Ensenso 3D camera (N35-802-16-BL) as it provides both 2D grayscale and depth images. Our perception system used both 2D images and depth images to estimate the 3D pose of the objects in the bin which will be discussed in Section 3.4.1.

3.3.3 Cleaning station

We developed a cleaning station (Figure 3.5) consisting of a stationary brush rack at the base of the station and a dust management system to collect the residue powder removed during brushing. In Figure 3.5, the fan continuously blows the powder stream into the vacuum outlet, and the vacuum outlet is connected to the same vacuum machine in Section 3.3.1. Hence, when the suction system was activated, the dust management system in the cleaning station would be activated too. The collection of residue powder was so that the powder could be recycled to print other parts.

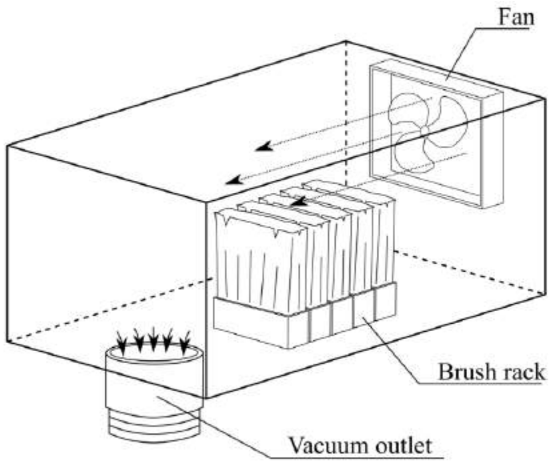


FIGURE 3.5: Cleaning station for brushing of parts.

3.3.4 Flipping station

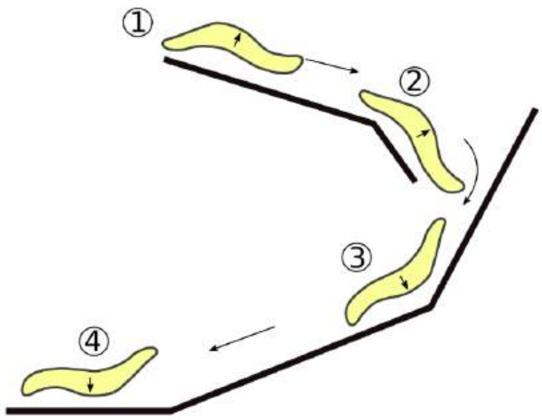


FIGURE 3.6: Passive flipping station using smart mechanical design.

To clean both sides of a flat shoe insole, we implemented a passive flipping station using a smart mechanical design without any use of motors or actuators (Figure 3.6). As illustrated in Figure 3.6,

1. Part is dropped at the entrance of the station.
2. Part slides down the upper level before hitting the mouth of the lower level.
3. Part flips over due to the impact as it enters the lower level.
4. Part continues sliding down the lower level to reach the exit of the station.

Although the design was simple, it would work well with relatively flat parts such as the shoe insole or flat plates. In addition, most residue powder will stick on the large areas of the flat objects. Thus, only the upper side and underside of flat objects were considered for cleaning, and flipping only had to be conducted once.

3.4 Software system overview

Our software system consisted of a series of modules that follows the pipeline in Figure 3.4. This includes perception, motion primitives, motion planning, and control. In addition, a sequence of parts to be cleaned was required as the parts were cluttered or stacked, hence selecting the most feasible part to be cleaned next would be performed based on the results from the part segmentation and localization module.

3.4.1 Perception

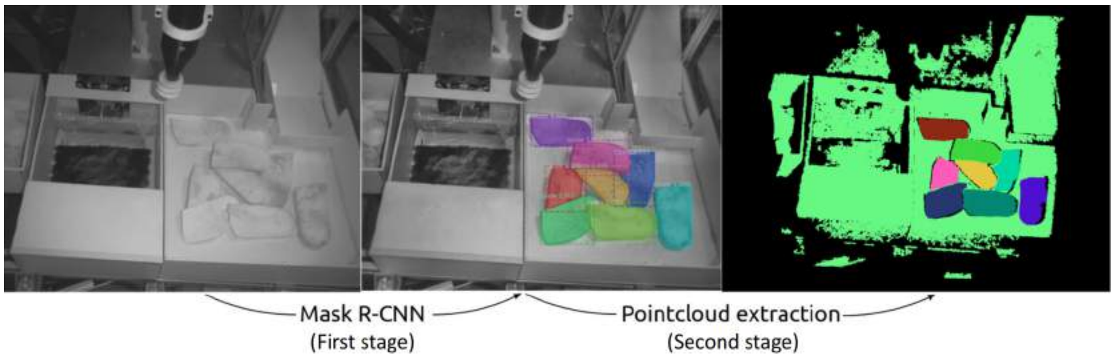


FIGURE 3.7: Sample illustration of instance segmentation using Mask R-CNN and extraction of 3D points of the shoe insoles.

The perception task was to identify and localize visible objects in the working space. However, due to heavy occlusions of the parts and poor contrast between the powdered parts and the bin, we require a robust perception system. Our approach was two-fold (Figure 3.7): (1) object detection and segmentation, (2) 3D pose estimation. In the first stage, we utilized a state-of-the-art deep learning network to perform instance detection and segmentation on 2D images to produce segmentation masks. The second stage extracted the 3D points of each object from depth images, using the segmentation masks obtained in the first stage, to estimate the object pose. This pipeline allowed us to exploit the superior performance of deep object classification and also depth information in estimating object pose to achieve a robust perception solution.

In the first stage, Mask R-CNN [38] was selected due to its superior performance in instance segmentation. The Mask R-CNN model was built on Feature Pyramid Network [127] and ResNet101 [128] backbone, and was open-sourced by Matterport Inc. [129] which was implemented in Tensorflow and Keras. The developers also provided a Mask R-CNN model that was pre-trained on a Microsoft COCO [48], a large image classification dataset. Thus, we applied transfer learning to fine-tune the pre-trained model so that it could classify the shoe insole. The main advantage of transfer learning was that a relatively smaller image dataset could be used to obtain good detection performances. We collected 75-100 real images with 5-10 powdered insoles per image and manually annotated them using VGG Image Annotator [130]. The camera used was discussed in Section 3.3.2 and the viewpoint remained unchanged during data collection and actual perception task. We obtained high detection rates even for occluded objects which is illustrated in Figure 3.7. On average, our network achieved a precision of 0.975 and a recall of 0.967, and it returns the bounding boxes and masks for every predicted object.

In the second stage, pose estimation was conducted by estimating the bounding boxes and computing the centroids of the segmented point clouds. The point cloud of each object was refined by operations such as statistical outlier removal and normal smoothing. After refinement, verification was performed to check if the object can be picked up by the suction cup by verifying that the exposed surface of the object is larger than the suction cup area. This area information also aided the algorithm in generating the cleaning sequence by prioritizing parts with large top surface areas. In addition, to increase the robustness of the pose estimations, certain physical constraints were applied, such as ensuring that the parts lie within the walls of the bin and cannot be floating within the bin.

With the combination of these two stages, we obtained a robust perception solution that addressed the perception challenge as mentioned in Section 3.2.

3.4.2 Motion primitives

Motion primitives include tasks such as picking, cleaning, and stowing the cleaned parts. Picking and cleaning motions require force control and feedback to ensure that the part remained secure on the suction cup. This was also to address the manipulation and contact tracking challenges mentioned in Section 3.2. We proposed two types of motion primitives, picking motion primitives and cleaning motion primitives.

Picking motion primitives: These primitives are useful for parts with flat exposed surfaces. To simplify the task, we adopt a picking motion that picks up the part along the normal of its top surface. The process is as follows:

1. Robot moves the suction cup above the centroid of the powdered part and the centroid is obtained from the perception module discussed in Section 3.4.1.

2. Suction cup is slowly lowered to pick up the powdered part. Compliant force control is enabled to provide feedback for the robot when a force was observed on the F/T sensor. This force indicates that the suction cup has made contact with the part (i.e. part is picked up) and the robot can halt its downward motion.

3. Verify that the powdered part has been successfully picked up by matching the current height that the suction cup is at, with the expected height from the point cloud data while including certain tolerances.

4. The part is lifted up by the suction cup with constant feedback from the F/T sensor to ensure that the part remains in contact while avoiding collisions. A collision can be detected when a sudden large force is observed on the F/T sensor.

As discussed in Section 3.3.1, the suction cup was connected to a high-power vacuum machine. Although the residue powder and the parts have different physical properties, the extra force generated by the vacuum machine in addition to the suction cup enabled sufficient force to lift and manipulate the part.

Cleaning motion primitives: These primitives were developed to remove residual powder on 3DP parts with relatively flat surfaces. After the part has been picked

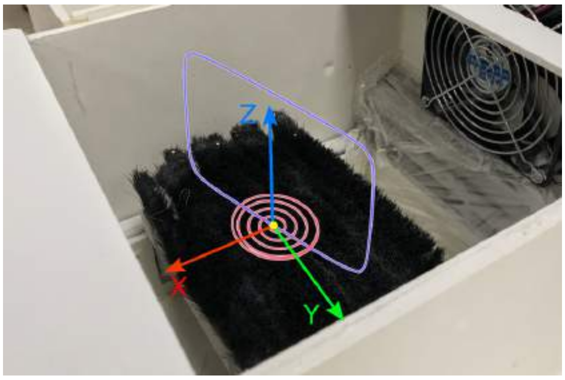


FIGURE 3.8: Illustration of rectangular cleaning motion (blue) and spiral cleaning motion (red) about the centroid (yellow).

up using the picking motion primitives described above, cleaning motion primitives executed are as follows:

1. Robot positions part above the brush rack in the cleaning station that was discussed in Section 3.3.3.

2. Part is slowly lowered to make contact with the brushes. Similar to Step 2 in *Picking motion primitives*, compliant force control is enabled so that the robot halts its motion when contact is observed.

3. Cleaning trajectories are performed. We adopted a combination of spiral and rectangular paths for cleaning (Figure 3.8). For the spiral path, once the user-defined maximum radius (outer ring) has been achieved, the robot will continuously circle about this maximum radius. For the rectangular path, the width, height, and directions along the X-Y plane could be customized. The spiral path was designed to clean nearly flat surfaces while the rectangular path aids the removal of powder in concave areas.

To maintain contact between the brushes and the part during cleaning, a hybrid position-force control scheme was applied. The force was regulated in the direction normal to the brush surface (Z axis) while the position was regulated along the tangential directions (X and Y axes) to ensure that the part remains within the brush area. The coordinate system can be seen in Figure 3.8. The force thresholds were determined through trial-and-error experiments while position thresholds were based on the size of the brush rack.

3.4.3 Motion planning

All planning and trajectories were planned online on OpenRAVE [131], as it is capable of returning collision-free trajectories quickly using the Bi-directional Rapidly-Exploring Random Trees (BiRRT) [132].

3.5 Automated decaking experiments

In this section, the experiments are evaluated and discussed.

3.5.1 Experimental setup

The shoe insoles were printed with PA12 Nylon powder using HP MJF5200 printer. *Ten* freshly printed shoe insoles were unpacked from the powder bed and placed randomly in a bin. The decaking experiment consisted of two runs and each run executed the following steps:

1. Initial localization of all powdered parts in the bin using the perception system described in Section 3.4.1. The view of the camera included the bin, cleaning station, and flipping station (Figure 3.7).
2. Generate a sequence of powdered parts to be cleaned by prioritizing powdered parts with larger exposed surface areas.
3. Pick up the powdered part that was first in the sequence using the Picking motion primitives described in Section 3.4.2.
4. Bring the powdered part to the cleaning station (Section 3.3.3) to remove residue powder on the underside of the part using the Cleaning motion primitives explained in Section 3.4.2. The brushing time was limited to 20s.
5. Bring the partially-cleaned powdered part to the flipping station (Section 3.3.4) to flip the part.
6. Re-localize the parts and the system was made to pay attention to the location of the exit of the flipping station to localize the flipped part.

7. Pick up the flipped part.
8. Bring the flipped part to the cleaning station to remove residue powder on the remaining underside of the part. The brushing time was limited to 20s.
9. Place the cleaned part into a collection bin.
10. Repeat Steps 3-8 for the remaining powdered parts in the sequence.

To provide a comparison, a skilled human operator also performed the decaking task with the same number of insoles with the same brushing limitation of 20s per side, which sums up to a total brushing time of 40s. Although a similar brush was used, the cleaning motions by the human operator were not restricted.

3.5.2 Experimental performance

The evaluation was conducted on two aspects: cleaning quality and running time. The cleaning quality was based on the weight of the insoles before and after the experiment, while the running time reported was the actual cycle time in a realistic setting. The results of the robotic system and human operator are shown in Table 3.1. In Table 3.2, a perfectly-clean insole was one that was fully cleaned by the human operator with no limitations on the brushing time. Figure 3.9 illustrates the timeline representation of all actions by the robotic system.

	Robotic system	Human operator
Mass before (g)	48.6 ± 10.9	48.9 ± 8.0
Mass after (g)	37.6 ± 6.4	29.8 ± 3.2
Cycle time (s)	50.1 ± 2.1	41.2 ± 1.9
Total brushing time (s)	40	40

TABLE 3.1: Performances of our robotic system and skilled human operator.

	Weight (g)	Powder removed (%)
Perfectly-clean shoe insole	22.4 ± 2.0	100
Cleaned parts (Robotic system)	37.6 ± 6.4	42.0 ± 24.4
Cleaned parts (Human operator)	29.8 ± 3.2	72.0 ± 12.1

TABLE 3.2: Comparison on amount of powder removed.

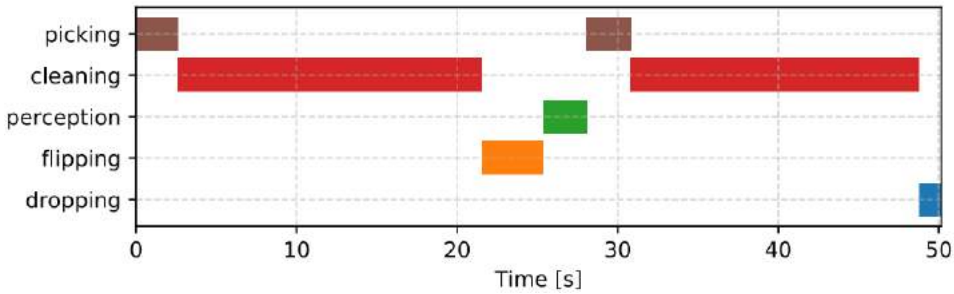


FIGURE 3.9: Average timeline representation of all actions by the robotic system during cleaning.

3.5.3 Discussion on decaking performance

From the results in Table 3.1 and Table 3.2, we make several observations. For the current workcell design, the human operator could remove more residue powder than the robotic system with a given brushing time. However, this might be due to the 3D cleaning motions that a human is capable of, and the robot decaking motions could be only a subset of the actual human 3D decaking motions without optimization. In addition to the dual-arm execution of the human, the human also has visual feedback on the cleanliness of the powdered insoles that allow active adjustment of cleaning motions. This allowed optimization of speed and brushing locations, which enabled a shorter cycle time and cleaning quality. As such, the robot's performance seemed inferior. A plausible improvement would be to redesign the workcell and mount the brushes on the robot end-effector instead to allow 3D cleaning motions.

The robot may have cleaning limitations due to the use of a suction cup as a gripper. However, a suction cup was selected as powder accumulation on the objects would create challenges to securely grasp the object for brushing because the layer of powder could disintegrate at any point in time. On the other hand, the suction cup was connected to the vacuum machine, which enabled the removal of the layer of powder accumulation, thus allowing proper contact between the gripper and the surface of the object. This could be another challenge when redesigning the workcell.

3.6 Conclusion

We presented a working prototype of an automated robotic system for the decaking of 3DP parts. We combined deep learning for 3D perception, smart mechanical design, motion planning, and force control for an industrial robot to develop a system that can clean 3DP parts in a fast and efficient manner. This showed the feasibility of automatizing post-processing tasks and the achievements of this prototype lay the groundwork for several possible extensions such as different types of parts, materials, or other powder-based 3DP technologies. Possible directions could include redesigning the work-cell setup so that the brushes can be mounted on the robot end effector for 3D cleaning motions, and including a cleanliness evaluation module together with a geometry-aware method to optimize cleaning motions.

Chapter 4

Deep classification and localization of powdered objects

4.1 Introduction

The introduction of powder in automation systems poses several challenges, especially for robot perception as the powder accumulation on the objects can drastically change the features of the objects which may lead to perception methods reviewed in Section 2.3 to fail. In addition, for every single part printed by powder-based AM technologies, cleaning would be the very first step for 3DP post-processing. This indicates that there could be frequent changes on the objects to be printed, and many high-performing methods in Section 2.3 tend to be data-driven which are object dependent. Thus, to move towards automated post-processing, we propose a fully automated vision pipeline for powder-based 3DP parts, where only the CAD models of the parts were provided as input. Our method could return a deep neural network capable of classifying and localizing *unseen powdered* parts with high precision and recall. The main motivation is, with a system that can accurately detect powdered parts, more end-to-end automated post-production tasks would be feasible, such as using a robot arm to sort printed parts after extracting parts from the powder bed.

Due to the diversity of 3DP parts from massive customization, it is not practical to collect real-world data for training deep neural networks as it can be extremely tedious and time-consuming. In Chapter 3, the perception system (Section 3.4.1) required a neural network to conduct classification and segmentation. However, the network was trained with manually collected data which took one week to prepare and annotate the data. To increase the accessibility of data, simulation is a viable option to obtain synthetic data. As reviewed in Section 2.3.3, using synthetic training data proved to be rather successful especially when applying domain randomization (Section 2.3.5).

The main aim was to detect *unseen powdered* parts with only the CAD models of the parts as these powdered parts may not resemble their CAD models due to the amount of powder on them, which varies based on their geometry. Also, every powdered part is unique as the amount of residue powder is different even for the same part, due to the manner in which the parts were packed for printing, as discussed in Section 2.2.2, where simulation of granular media such as snow and sand may not be applicable. Thus, there is a need for a method that can simulate artificial powder accumulation on 3DP parts and our proposed approach is as follows:

1. Simulate artificial powder on CAD models (in STL format) to obtain several unique powdered models per part.
2. Feed CAD models and their respective powdered models into a physics simulator and render synthetic images.
3. Automatically generate object masks for every synthetic image to be used for neural network training.
4. Train a Mask-RCNN network with synthetic images and object masks.

Predictions by a network trained based on our approach can be seen in Figure 4.1.

The remainder of the chapter is organized as follows. The automated vision pipeline is presented in Section 4.2, two methods of simulating artificial powder are discussed in Section 4.3, and the generation of synthetic images and object masks are explained in Section 4.4. In Section 4.5, we tested our approach on single-class and multi-class object detection tasks and evaluated the results in Section 4.6.

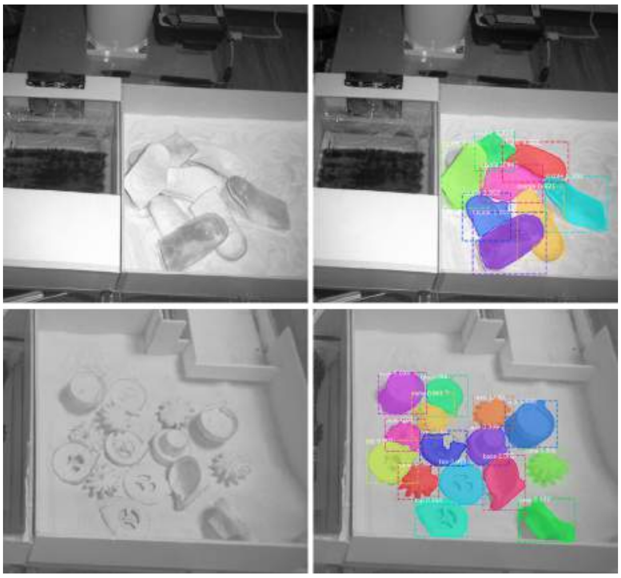


FIGURE 4.1: Mask predictions by a network trained purely on synthetic images on powdered 3D printed parts.

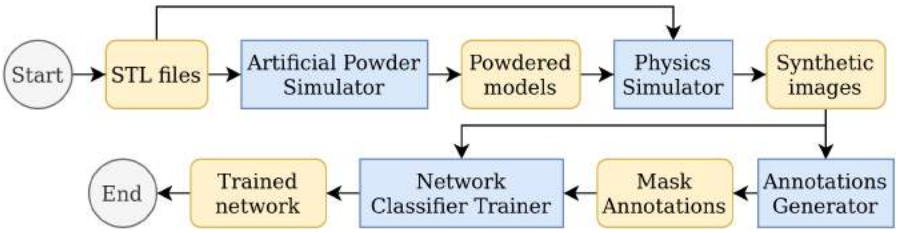


FIGURE 4.2: Overview of automated vision pipeline.

4.2 Automated vision pipeline

The overview of the vision pipeline is illustrated in Figure 4.2. Firstly, the CAD files (in STL format) are fed into the Artificial Powder Simulator (Section 4.3) to simulate artificial powder on them and produce the powdered models. Secondly, these powdered models and their original unpowdered CAD models are used in the Physics Simulator (Section 4.4.1) to generate synthetic images. Thirdly, the synthetic images are fed to the Annotations Generator (Section 4.4.2) to obtain the mask annotations. Finally, the synthetic images and mask annotations are used to train a network classifier to produce the trained network (Section 4.5.2).

4.3 Methodology to simulate object powdering in 3D printing

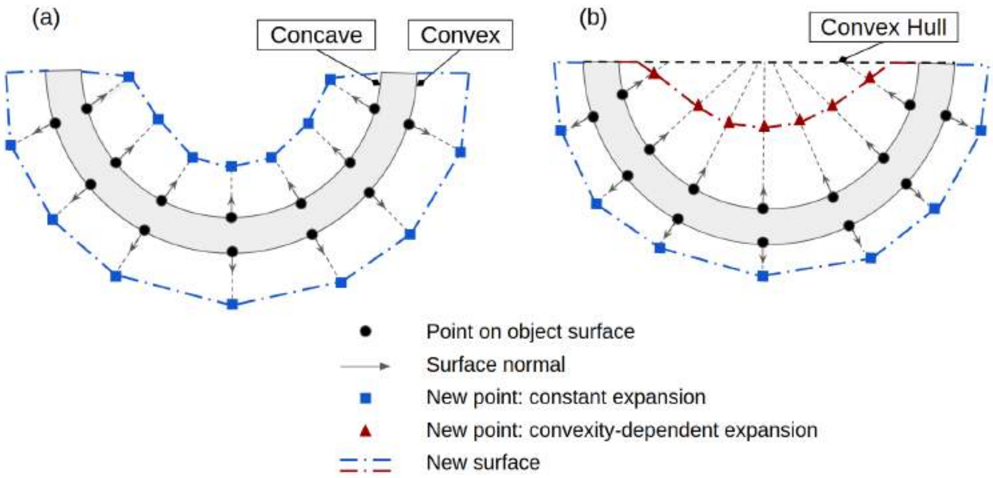


FIGURE 4.3: Illustration of (a) naive powder generation, and (b) enhanced powder generation which takes into account the local convexity of the workpiece.

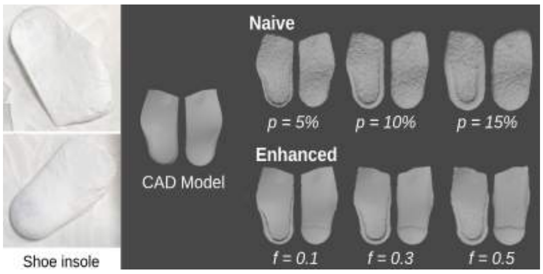
The artificial powder was simulated on the CAD models in an attempt to improve the detection performances of the neural networks. The idea was to obtain a set of powdered models with at least one instance that would look visually similar to real powdered parts. We propose two methods in the generation of artificial powder: the naive method (Section 4.3.1) and the enhanced method (Section 4.3.2), and the illustration is in Figure 4.3. The comparison of real powdered 3DP parts with the artificial models generated from both methods can be seen in Figure 4.4.

4.3.1 Naive powder generation method

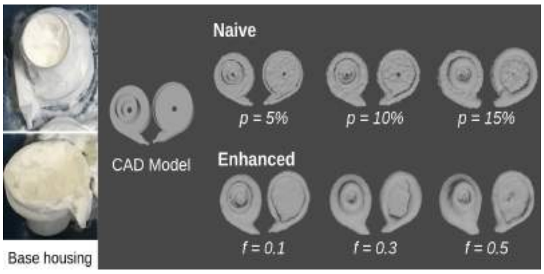
The naive powder generation method was a simple approach that involved the random extension of the vertices of the STL file along their normals (Figure 4.3a). The steps are as follows:

1. Discretize the STL file of the object into a point cloud by using Poisson disk sampling [133]. This was to get a better distribution of points to create artificial powder on these points.

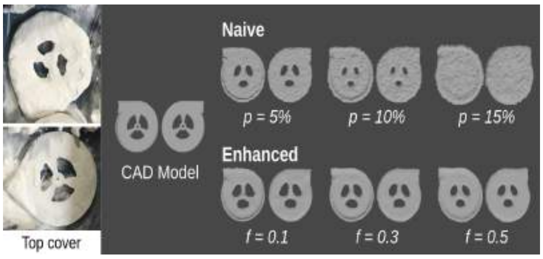
2. Calculate the average dimension (D) by summing the length, height and width of the object bounding box.



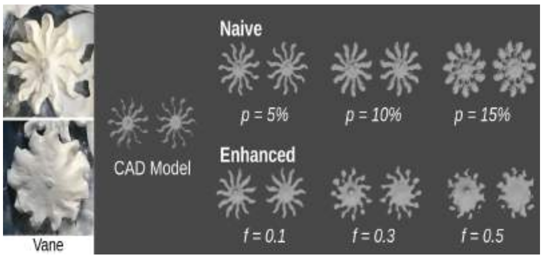
(A) Real insoles and artificial powder models



(B) Real housings and artificial powder models



(C) Real covers and artificial powder models



(D) Real vanes an artificial powder models

FIGURE 4.4: Comparison of real powdered 3DP parts with naive powder generation and enhanced powder generation.

3. Set the dimension range of values to randomly pick from, based on a user-defined powder percentage (p) that had a value from [0-1]. The upper limit was set as $p * D$, while the lower limit was a higher value between 0mm and $(p * D - 5)$ mm. This was to prevent negative values.

4. Pick a random value between the dimension range for every vertex. The new vertex would be extended along the vertex normal by this value.

5. Reconstruct the triangular mesh from the point cloud of new extended points by using the Poisson surface reconstruction method [134] to obtain the powdered model.

The naive powder generation method produced powdered models as shown in Figure 4.4.

4.3.2 Enhanced powder generation method

In Section 4.3.1, the naive powder generation method treats the concave and convex areas equally (freffig:powderillustrationa). Yet, one can empirically observe that powder tends to accumulate more in concave regions. Based on this observation, we propose an enhanced powder generation method that adds powder differentially, depending on the local convexity of the surface, as characterized by the convex hull of the object as shown in Figure 4.3b. The steps are as follows:

1. Discretize the STL file of the object into a point cloud by using Poisson disk sampling [133].
2. Obtain the convex hull of the object. The convex hull provides global information on the geometry of the object, including the concave and convex areas.
3. Find the intersection point (Q) on the convex hull, for every point (P) of the point cloud using the Möller-Trumbore ray-triangle intersection algorithm [135]. The algorithm is a fast method for calculating Q in 3D without having to compute the plane equation of the plane containing the triangle.
4. If $P \neq Q$, calculate the length (PQ) and its direction (n). A larger PQ meant that the point of interest was far from the convex hull, which could indicate that it may be a concave point, and thus more powder may be found at this point. Thus, the new extended point was $E = P + (f + k) * PQ * n$, where f was the enlarging factor, while k was a small random value to ensure that with the same f, the powder model generated would be unique for more variability. When $f = 1$, the model would be equivalent to the convex hull.
5. If $P = Q$, this meant that the point lay on the convex hull and could be a convex point. However, since the real objects also could have some powder on convex surfaces, a small random value was drawn from a single distribution and applied along the normal of P.
6. Reconstruct the triangular mesh from the point cloud of new extended points by using the Poisson surface reconstruction method [134] to obtain the powdered model.

The enhanced powder generation method produced powdered models as shown in Figure 4.4.

4.3.3 Comparison between powder generation methods

The two methods share the same first step and last step, which was the discretization of the STL file into a point cloud and the reconstruction of the point cloud into a mesh, to produce a surface that resembles the real, powdered parts. However, as mentioned in Section 4.3.2, the enhanced powder generation method treats concave and convex regions differentially. Hence, it produced powdered models that were much more visually similar to the real, powdered parts in Figure 4.4, especially in concave regions of the housings (Figure 4.4b) and the vanes (Figure 4.4d), as compared to the naive powder generation method.

4.4 Methodology to generate synthetic training data

The automated generation of training data consisted of two components: the images and the annotations. With the powder models obtained from Section 4.3, we proceed with synthetic data generation.

4.4.1 Creation of synthetic images

We focused on a bin-picking task as post-processing cleaning methods usually involve a bin or a chamber. Hence, we used *pybullet* [136] to simulate objects falling into a bin due to gravity. Domain randomization was applied in the creation of synthetic images. To induce sufficient variability in the simulator, the following parameters were randomly chosen:

- *Bin:* Color, size, and orientation of box.
- *Camera:* Position and orientation of camera.
- *Objects:* Type of object, number of objects per image, and their starting poses above the bin. The type of the object includes the class of the object, its respective CAD model, and powdered models.
- *Background:* Background image to be stitched with the foreground image of the box. The COCO dataset [137] was used as the background dataset.

The synthetic images were rendered by capturing an image of the bin after the fallen objects had settled, and the foreground (bin) was stitched onto a random background image.

4.4.2 Automatic image annotation

The main advantage of using a physics simulator for the creation of synthetic images was that all locations of the objects in the image were known. With this information, we can easily obtain annotations that would be used to train a Mask R-CNN network. The mask annotations required were just the polygon of every object, which was generated using a basic gray-thresholding method. The steps are as follows:

1. Change the color of one object to white and change the color of other objects and the bin to black.
2. Take a snapshot of the bin with the same camera parameters that were used to create the current image.
3. Use gray-thresholding to obtain the contours, or the polygon, of the object which was the foreground.
4. Repeat steps 1-3 for the remaining objects in the bin.
5. Export the contours and class labels to a file that would be used to train the network.

4.5 Experiments

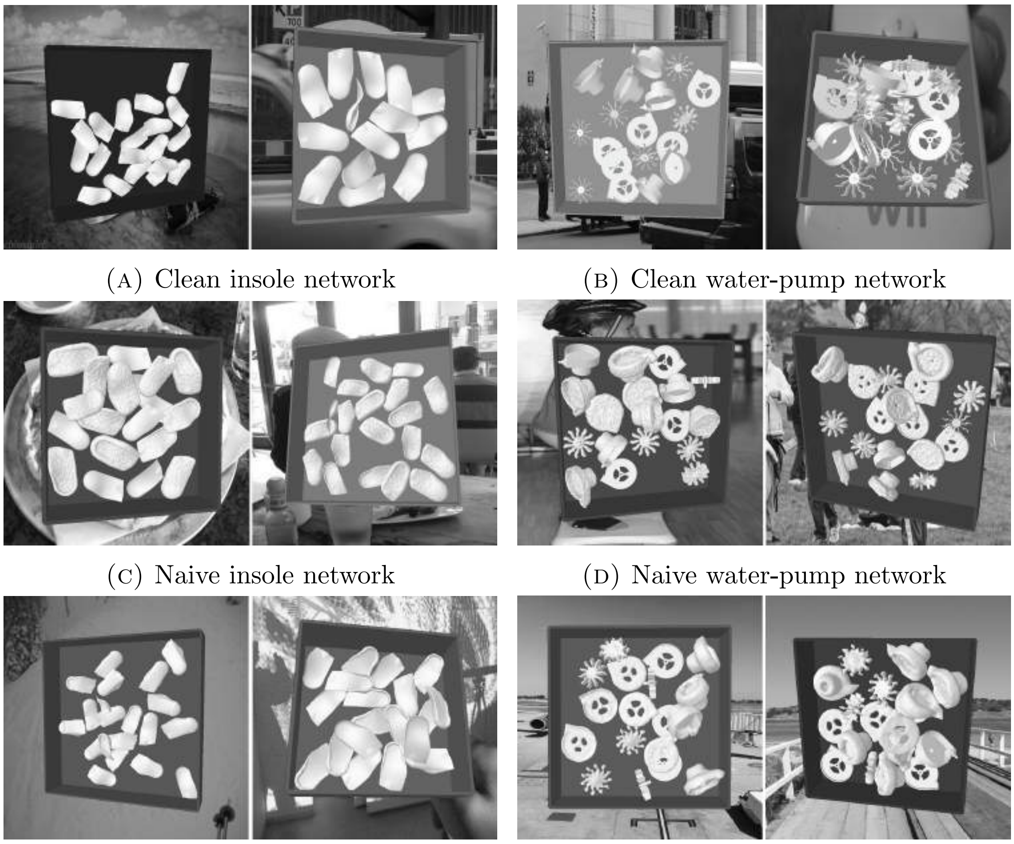
We tested our approach on two tasks: single-class object detection and multi-class object detection. We aim to compare the differences in detection performance between having powder models and just using the CAD models. The single-class object was a shoe insole[1] (Figure 4.4a). The multi-class objects was a water-pump assembly taken from [138] and consisted of 3 classes: base housing (Figure 4.4b), top cover (Figure 4.4c) and vane (Figure 4.4d). It could be seen that there is more powder accumulated on the real parts for the base housing and vane classes, especially for the vane class where the real parts look very much different from its CAD model.

4.5.1 System setup

The synthetic images and annotations obtained from Section 4.4 were fed into a Mask R-CNN network by using transfer learning, where pre-trained COCO weights were used. In this chapter, every network had 1,000 grayscale synthetic images generated and 70% was the training set while 30% was the validation set. A learning rate of 0.001 and ResNet101 backbone were used to train the network over 100 epochs. All of these were executed on a workstation with 11GB NVIDIA GeForce GTX 1080 Ti. The training time for a single network with 1,000 synthetic images was approximately 75 minutes.

[1]The authors have been granted permission by Footwork Podiatry Laboratory to use their shoe insole design and images in this report.

4.5.2 Trained networks



(A) Clean insole network

(B) Clean water-pump network

(C) Naive insole network

(D) Naive water-pump network

(E) Enhanced insole network

(F) Enhanced water-pump network

FIGURE 4.5: Sample grayscale synthetic training images for respective networks with objects: insole (left) and water-pump (right).

For both the shoe insole and water-pump assembly, three networks were trained under the specifications in Section 4.5.1: Clean, Naive, and Enhanced. All CAD and powdered models used in the physics simulator had a plain white color as their texture because the real powdered parts are in white (powder) with some black patches (printed part). To conduct a thorough evaluation of our method, we also used a network trained on *real* images of the shoe insole.

1. *Real:* Network was trained on 340 grayscale images taken from an Ensenso camera N35-802-16-BL where each image had an average of 8 *real, powdered* insoles.

2. *Clean:* Only the CAD models of the objects were fed into the simulator. This was the direct application of previous works in Section 2.3.5, where the

CAD models were directly fed without making changes to them, as shown in Figure 4.5a and Figure 4.5b.

3. *Naive:* The CAD models of the objects and 3 powdered models per class were fed into the physics simulator, as shown in Figure 4.5c and Figure 4.5d. The powdered models were generated by the *naive powder generation* method in Section 4.3.1 using powder percentages of 5%, 10%, 15%, as depicted in Figure 4.4.

4. *Enhanced:* The CAD models of the objects and 3 powdered models per class were fed into the physics simulator, as shown in Figure 4.5e and Figure 4.5f. The powdered models were created by the *enhanced powder generation* method in Section 4.3.2 using enlarging factors of 0.1, 0.3, and 0.5, as depicted in Figure 4.4.

Since three powdered models per class were used in the Naive and Enhanced networks, it meant that for the water-pump assembly, there were 12 models for the simulator to randomly pick from to drop them into the bin.

4.5.3 Test sets and evaluation metrics

The test sets were the sets of grayscale ground truths images that were taken using the Ensenso camera. Each test set consisted of 30 grayscale images, thus the networks were trained on grayscale images as mentioned in Section 4.5.1.

1. *Insole Test Set (ITS):* An average of 8 real, powdered insoles were in each image. The background and powdered insoles used were the same as the setup used to collect real images for the Real network in Section 4.5.2. A total of 240 instances of the insole were in this test set.

2. *Water Pump Test Set (WPTS):* An average of 15 powdered parts, with an average of 5 parts per class, were in each image. A total of 450 instances of the water pump assembly parts were in this test set, with 150 instances each for the 3 classes: base housing, top cover, and vane.

The evaluation metric used was the mean Average Precision (mAP) which is the mean of all average precisions based on the number of images in the test set. Average Precision (AP) is defined as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight, or can also be defined as the area under the precision-recall curve after sorting the class scores in descending order. In our experiments, we calculated the AP using the area under the precision-recall curve, and true positives are predictions that match the ground truths with an Intersection over Union (IoU) score greater than 0.5.

4.6 Experimental results and discussion

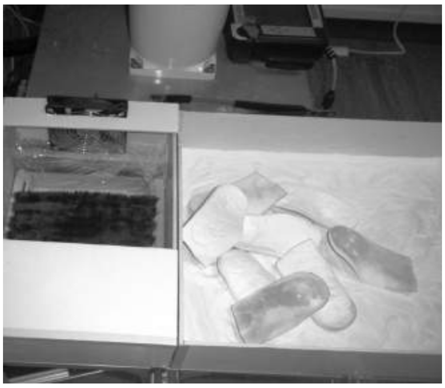
We evaluated the networks trained in Section 4.5.2 on the test sets in Section 4.5.3. The results for the single-class object detection are shown in Table 4.1 and the results for the multi-class object detection are depicted in Table 4.2.

4.6.1 Single-class object segmentation and localization

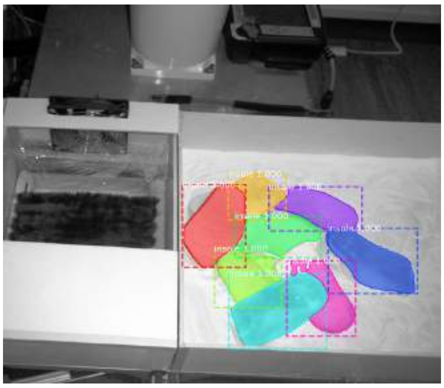
Network	mAP
Real-I	0.929
Clean-I	0.614
Naive-I	0.786
Enhanced-I	0.857

TABLE 4.1: Detection performances on Insole Test Set (ITS).

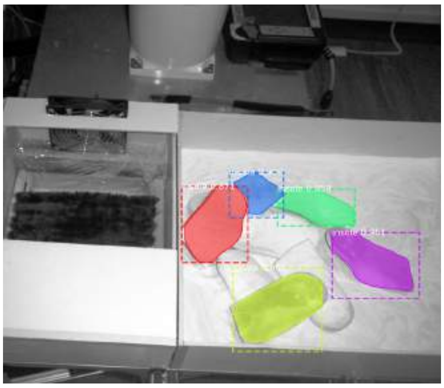
For the single-class object detection where the shoe insole was the object of interest, we made several observations. By comparing the performances on the ITS in Table 4.1 and in Figure 4.6, we noticed that the mAP of the Real network was the highest, which could be attributed to the network being trained on *seen* data as the background and powdered insoles used were the same in the training set and test set. However, the Enhanced-I network was tested on *unseen* data, as it was trained purely on synthetic images. Yet, the mAP of the Enhanced-I network was good and can be comparable to the mAP of the Real network. The Enhanced-I network also performed the best amongst the 3 networks that were trained on synthetic images, while the Clean-I network performed the worst, showing that only using the original, unpowdered CAD model in simulation may lead to limited performances.



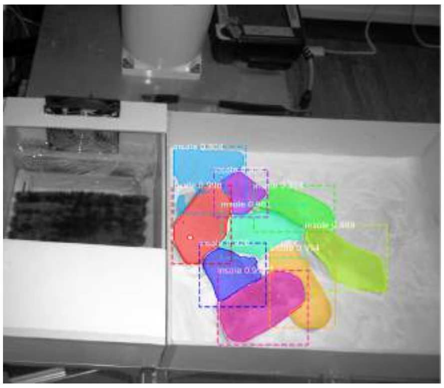
(A) ITS ground truth



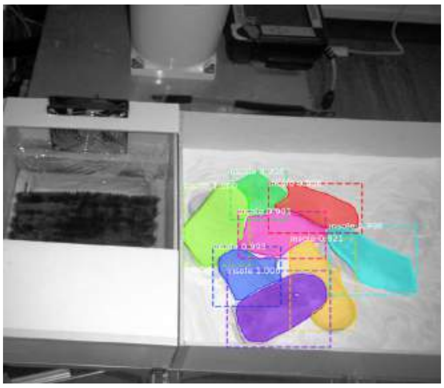
(B) Real-I



(C) Clean-I



(D) Naive-I



(E) Enhanced-I

FIGURE 4.6: Mask predictions of various networks on a ground truth image from Insole Test Set (ITS).

4.6.2 Multi-class object segmentation, classification and localization

Network	Base housing mAP	Top cover mAP	Vane mAP
Clean-WP	0.817	0.751	0.811
Naive-WP	0.868	0.743	0.766
Enhanced-WP	**0.894**	**0.824**	**0.936**

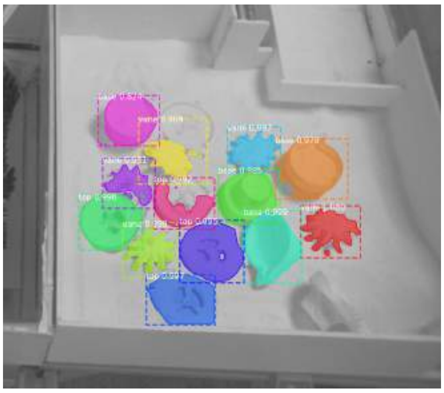
Table 4.2: Detection performances on Water Pump Test Set (WPTS).

For the multi-class object detection where the water-pump assembly parts were the objects of interest, we made several observations based on the results in Table 4.2 and in Figure 4.7. The results for the water-pump classes were competitive across the three networks.

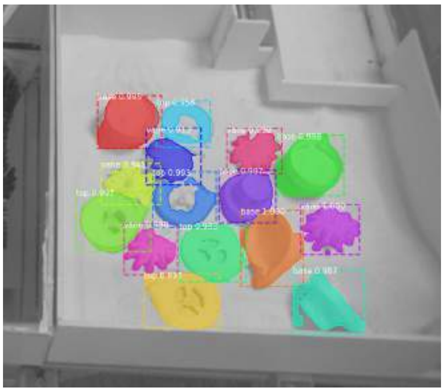
In Enhanced-WP, the implementation of powdered models showed superiority for classes with more powder accumulation on the real parts, such as the base housing and the vane. For the top cover class, where less powder was accumulated on the real parts, the mAP of all three networks was rather comparable. We also observed that the Enhanced-WP network performed the best in this scenario. Although the Naive-WP network did not perform as well as Clean-WP in the top cover class and the vane class, this could be due to too much powder generated on the top class when $p = 15\%$ and irregularities in the internal surface of the vane class during the reconstruction of mesh when at $p = 15\%$ (Figure 4.4).

Surprisingly, the Enhanced-WP network performed extremely well for the vane class, even when the powdered model ($f = 0.5$) did not fully resemble the real parts, as shown in Figure 4.4d. The differences between the detection performances for parts that do not fully resemble the real parts might be caused by the differential treatment of concave and convex regions as explained in Section 4.3.3, which shows that this treatment affects the robustness of the generator. We also noted that the networks for the water-pump assembly performed much better than the insole, and this may be attributed to more intricate shapes in the assembly which allowed the network to easily learn features distinct to the assembly.

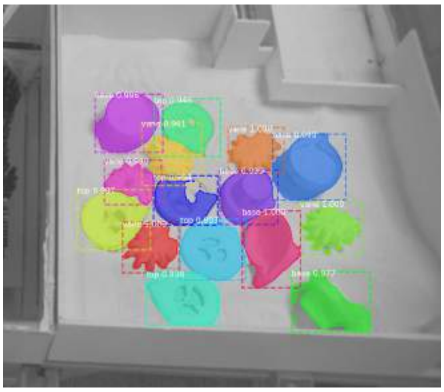
(A) WPTS ground truth



(B) Clean-WP



(C) Naive-WP



(D) Enhanced-WP

Figure 4.7: Mask predictions of various networks on a ground truth image from Water Pump Test Set (WPTS).

4.6.3 Overall discussion

From the observations in Section 4.6.1 and Section 4.6.2, we can state that the enhanced powder generation method is superior compared to the naive powder generation method. This could be attributed to the consideration of concave and convex surfaces in the simulation, as discussed in Section 4.3.3, resulting in more realistic powder models. In addition, the results showed the importance of using powdered models rather than only the CAD model in simulation, especially when the real parts have more accumulated powder, such as for the base housing and vane classes. Also, we showed that replacing real data collection with synthetic data is feasible as the results of using synthetic images were comparable to using real images.

Although the overall mAP may be less than 0.95, note that the Real-I network was trained on 340 real images of the same shoe insoles in the same background, and yet it could only achieve a mAP of 0.929. Hence, a possible improvement for the network performance could be to use a bin of a different colour, as both the caked parts and bin were white in colour. Alternatively, more synthetic images could be used for training the network.

4.7 Conclusion

We have introduced a sim-to-real vision pipeline to detect powdered 3DP parts, which was fully automated, and the detection results from our system proved to be comparable to using real data, as shown in the single-class object detection. We extended our method to a multi-class object detection task and obtained superior results, especially for classes where the real parts have more accumulated powder. This showed the importance of using powdered models in simulation for the detection of powdered 3DP parts. With this vision pipeline, a possible extension would be to relate the brushing motions in Chapter 3 with powder accumulation, so that an empirical model that relates the robot brushing motions, speed, and contact force with powder accumulation could be achieved. For example, a thicker powder deposition would usually be present in concave regions, which requires a larger applied force to allow the brush to penetrate the deeper regions. Also, the brushing actions would have to relate to the geometry of the object, to follow the normal of the surfaces for more effective powder removal.

Chapter 5

Fingerpad customization with set operators for precise and versatile grasping

5.1 Introduction

Precision and versatility are two key objectives of grasping in many robotic applications, such as packing, loading, handling for inspection, and assembly [139]. *Precision*: the grasp should be tight and the relative pose between the object and the gripper should be precisely determined by a priori. *Versatility*: the same gripper should be able to grasp the same or different objects from different initial poses.

Yet, precision and versatility are conflicting objectives. High-precision grasping might be achieved by customizing rigid grippers (Section 2.4.1) that closely conform to the shape of a given object at one grasp point, but these grippers may not be able to grasp other objects or even the same object at another location. Conversely, versatile grippers such as soft grippers (Section 2.4.2), are generally unlikely to achieve high-precision grasping.

Both objectives may be affected by factors including gripper design or the objects in a task. In Section 2.4.3 and Section 2.5.3, the limitations of robotic grippers for 3DP post-processing and grasping 3DP parts were discussed. Thus, there is a need

FIGURE 5.1: **Fingerpad Customization with Set Operators (FCSO):** A single pair of fingerpads is capable of tightly grasping different objects at multiple poses per object. The figure shows a pair of fingerpads that have been designed by FCSO to conform optimally and simultaneously to the geometries of four grasped surfaces (2 objects $\times$ 2 poses per object) to form caging grasps. Physical grasping experiments are available at `https://youtu.be/M68YagfUF1g`

for a robust, principled method that can automatically design precise and versatile grippers for complex objects.

We introduce a fast, end-to-end approach (Figure 5.1) to customize grippers to achieve precise and versatile grasping: Fingerpad Customization with Set Operators (FCSO).

Our approach relies on two key components:

- A method based on set operators (Boolean intersection, union, subtraction), to extract object features and synthesize gripper surfaces that optimally conforms to different local shapes: either at different grasp points on the same object, or on different objects to form caging grasps. Caging grasps are usually formed based on a geometric constraint, as reviewed in Section 2.5;

- A grasp quality evaluation method for synthesized gripper surfaces to select the optimal gripper surface. This might be an extension of existing grasp indicators, as reviewed in Section 2.5.2, to emphasize the geometric quality of the grasp for caging grasps.

We aim to balance two conflicting objectives: precision and versatility. Our grippers are designed to be sufficiently versatile by ensuring that a stable grasp could be achieved for every planned resting position of the objects. In most tasks, there may be specific requirements on the object poses, e.g. for assembly, or prior knowledge of the poses can be obtained by computer vision, hence it would suffice to plan secure grasps for a handful of poses. Precision could be achieved by extracting the local contours of the objects to obtain optimal gripper surfaces that conform to these contours.

The rest of the chapter is as follows: Section 5.2 introduces the pipeline of our algorithm, Section 5.3 details the method for customized fingerpads. Section 5.4 explains the concept for our geometric grasp quality measure and Section 5.5 evaluates our method from two perspectives through experiments.

5.2 Fingerpad customization pipeline

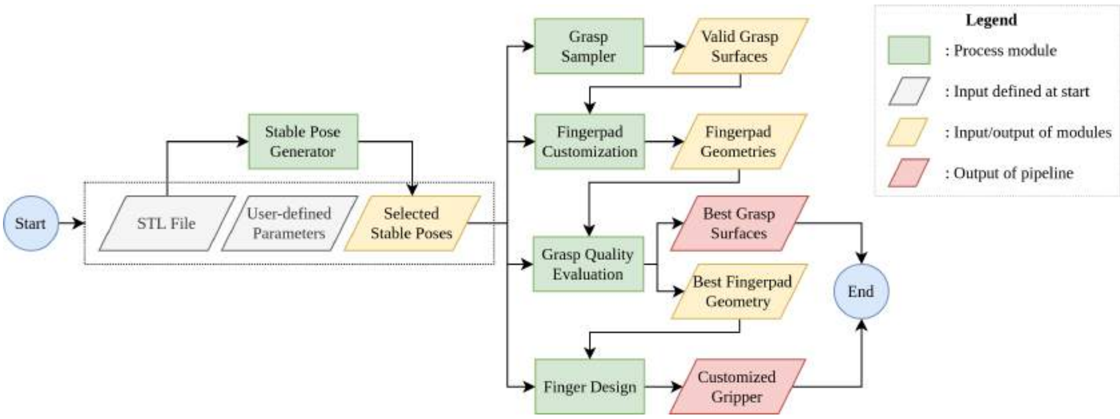


FIGURE 5.2: Proposed pipeline for FCSO. It accepts the STL file of the objects, user-defined parameters from a configuration file, and the flat finger model of a gripper, to return the best grasp surfaces and the best-customized gripper design.

We introduce FCSO in Figure 5.2 which consists of five modules: stable pose generator, grasp sampler, fingerpad customization, grasp quality evaluation, and finger design. The stable pose generator accepts the CAD model of objects and user-defined parameters, e.g. gripper specifications, number of stable poses to plan grasps, and size of fingerpad. A set of stable poses, that rest the objects on a planar surface, is returned and stable poses are automatically selected by random. Alternatively, manual selection could be done if specific poses are desired. At each selected pose, grasps are sampled to obtain valid grasp surfaces and locations. Sampled grasps are used in fingerpad customization to extract object features by set Boolean operators to get fingerpad geometries at each grasp location. The fingerpad geometries at each grasp location are evaluated on their grasp quality to return the best fingerpad geometry and grasp location. The best fingerpad geometry is then fused onto a flat finger to obtain the final customized finger to be mounted on a gripper base.

Detailed explanations of the methodology of each module are as follows. Stable pose generator in Section 5.2.1, grasp sampler in Section 5.2.2, fingerpad customization with set operators in Section 5.3, grasp quality evaluation in Section 5.4 and finger design in Section 5.2.3. Reference coordinate axes in subsequent images follow the convention: X-axis in red, Y-axis in green, and Z-axis in blue.

5.2.1 Stable pose generator

The stable pose generator aims to provide several poses that naturally rest the objects on a planar surface, prior to the grasp approach. The stable orientations are estimated with a quasistatic model [140, 141]. The selection of stable poses is random and automatic, and the number of selected stable poses, N_p, is pre-defined by the user. If there are specific requirements or prior knowledge on the poses of the object, e.g. for assembly tasks, optional manual input or selection can be conducted. The grasp approach direction is defined in the axis of the world where the gripper approaches the object. A top-to-bottom grasp approach (Z-axis) is chosen by default as side approaches are usually difficult for small objects due to possible collision of the gripper base with the table.

5.2.2 Grasp sampler

Many tools can plan initial contact locations for basic grippers, such as Graspit [85] or SynGrasp [86], or learning-based methods for ambidextrous grasping [90] and multi-affordance grasping [91]. For customized grippers, the local contours are key to forming caging or immobilization grasps, such as in [58, 64]. Thus, the grasp sampler is required to fully sample the object geometry. It generates candidate grasps by sliding a pair of rectangular samples (S) along the axes of objects, with a sampling step defined as stride (Figure 5.3). This was motivated by the sliding window in neural networks where receptiveness is improved by adjusting the stride [142]. Similarly, the stride could be applied in grasp sampling to produce more candidates. A smaller stride, or smaller sampling step, returns more grasp candidates. The length (L), width (W) and thickness (T) of S is user-defined. The penetration depth (D) is the amount of penetration of S into the object mesh, and $0 < D < T$.

Feasibility checks are performed on every sampled pair Figure 5.4. A sample pair is valid if a sufficiently large contact area can be established during grasping, the grasp is collision-free, and the object can fit into the gripper opening. The number of valid sample pairs for the m^{th} pose is $N_{s,m}$, where $m = 1, 2, ..., N_p$.

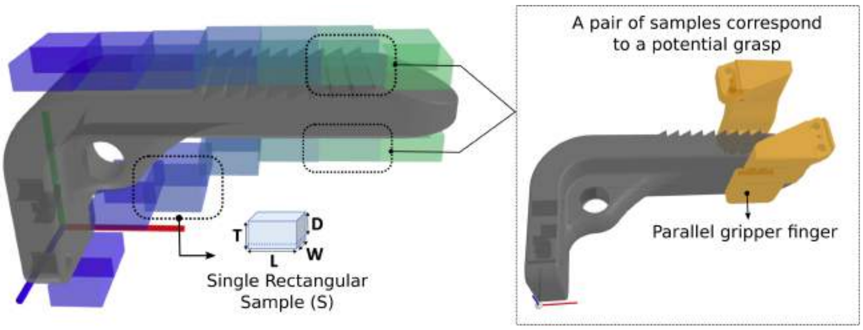
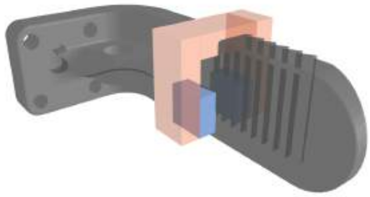
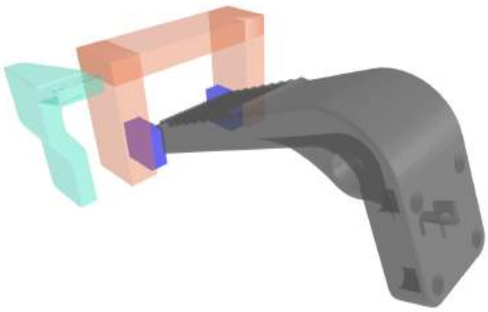


FIGURE 5.3: **Grasp sampling** by a sliding pair of rectangular samples (S) along the lateral axis of an object, with a stride equivalent to L. Each sample pair has the same color code.



(A) Gripper base (pink) collides with object (grey) at sampled fingerpad pair (blue).



(B) A feasible pair of samples (blue) is found that satisfies all three conditions. The original flat fingerpad of a commercial gripper (green) is included for visual comparison.

FIGURE 5.4: Examples of feasibility checks.

5.2.3 Finger design

Commercial grippers are often parallel flat finger grippers with basic flat fingerpads, the CAD model of these basic fingerpads can be retrieved. The optimal gripper geometry obtained in Section 5.4 is fused onto the flat finger to obtain the print-ready CAD model of the customized gripper.

5.3 Fingerpad Customization with Set Operators

Caging grasps and immobilization are essentially performed based on geometrical constraints [82, 83], that has been reviewed in Section 2.5. Perturbations would not affect the pose of a caged object, thus the pose could be precisely determined with prior information on the gripper. Velasco [63] proposed using Boolean intersections to extract simple, local geometries of objects so that grippers that conform to object shapes can be achieved, but manual grouping is required before subtraction. We extended this concept in our method by using a combination of set Boolean operators with a filter, which allows our method to be sufficiently robust to different object geometries thus achieving an automated design process. Set Boolean operators such as intersections, unions, and subtractions are operations that allow easy addition of new objects or poses.

5.3.1 Fingerpad customization without filter

We define the number of geometries to be extracted as N and rectangular fingerpad sample, S. The n^{th} geometry bounded by S and the mesh is G_n, where $n = 1, 2, ..., N$. I_n is the intersection of S with G_n and the union of N intersections is M_N. The customized fingerpad is defined as P. The method to create P without the automatic filter is shown in Figure 5.5. This method would generally work well if the sampled geometries are good. Explanation of good geometries is in Section 5.3.2.

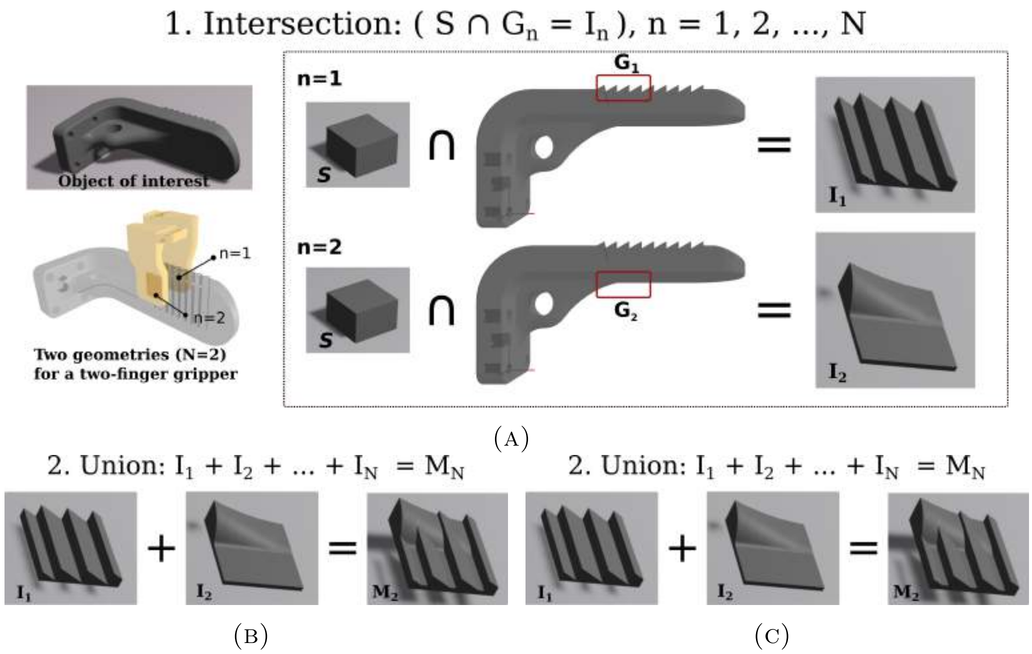
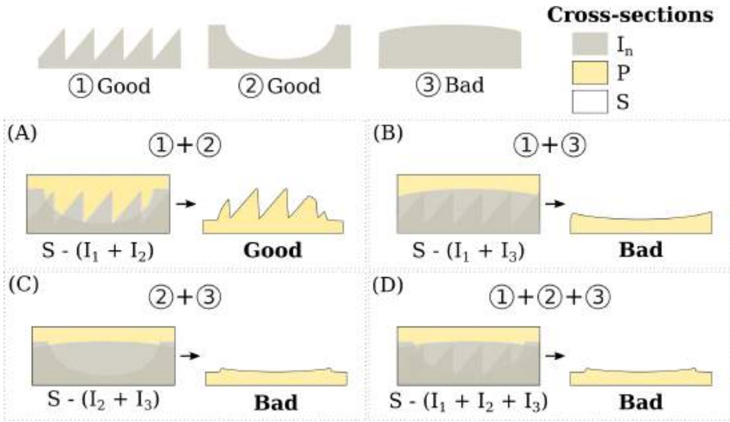


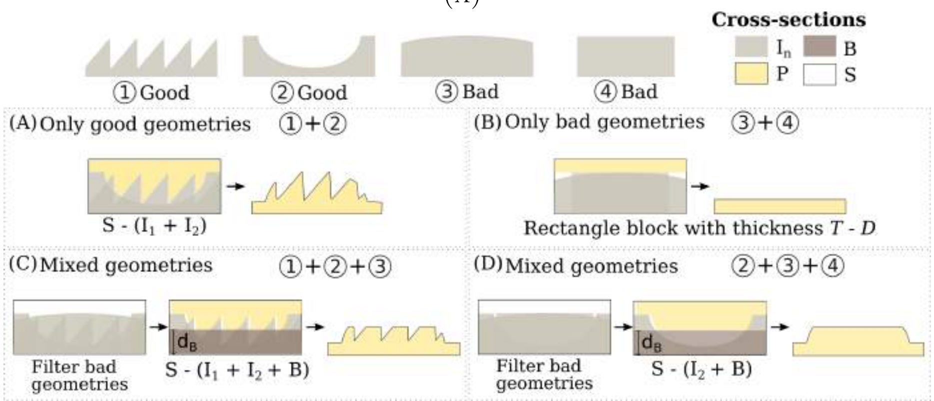
FIGURE 5.5: **Fingerpad customization (without filter)** based on the number of geometries (N), while illustrating a three-step procedure on a pair of fingerpads. (A) Independent Boolean intersections (I_n) resulting from the intersection of every valid rectangular sample (S) and G_n, which is the n^{th} geometry of the mesh bounded by the S. The samples are obtained from the grasp sampler (Section 5.2.2); (B) Boolean union of N intersections (M_N); (C) Boolean subtraction of S and M_N to obtain fingerpad (P) that has a shape which conforms to the mesh at all G_n.

5.3.2 Fingerpad customization with filter

We introduce a volume threshold filter to provide feedback across local geometries. It automatically differentiates 'good' and 'bad' geometries obtained from set intersections, thus improving the robustness of the geometry extraction to achieve grasps formed on geometric constraints. Good geometries are defined as shapes that would create fingerpads that can achieve secure grasps while bad geometries would not achieve such restrictions. The differentiation is crucial as bad geometries such as flat surfaces, are supersets of all geometries, i.e. any geometry G_n can be subtracted from a flat rectangular pad. This also means that any intricate geometries are absorbed by a flat rectangular pad. Thus, if any I_n is flat, M_N would also be flat which results in an undesirable flat fingerpad, P. Figure 5.6a shows the absorption of the good geometries in the presence of a single bad geometry. This was avoided with the filter in Figure 5.6b.



(A)

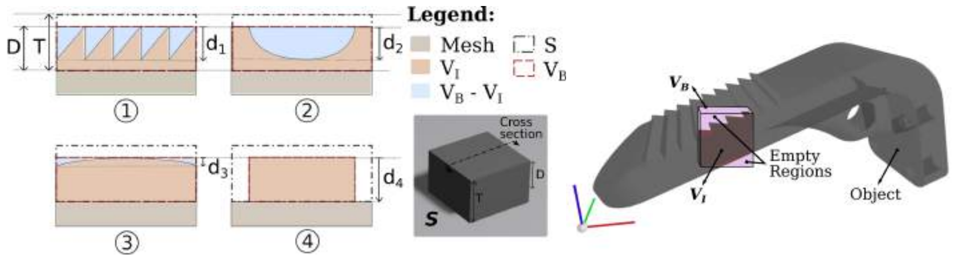


(B)

FIGURE 5.6: Comparing effects of the filter with good and bad geometries. (A) **Without filter**: Undesirable P, in yellow, obtained in the presence of a single bad geometry. This shows the need for a filter to differentiate between geometries; (B) **With filter**: Visibly improved performance. Illustrating three possible cases discussed in Section 5.3.2, with $d_1 > 0, d_2 > 0, d_3 = d_4 = 0$. In Example C, $d_B = min(d_1, d_2) * K$, whereas in Example D, $d_B = d_2 * K$.

The differentiation of geometries uses a volume ratio (R) with a constant threshold (th). The volume ratio, $R = (V_B - V_I)/V_B$, where V_B is the bounding box volume of I_n, and V_I is the volume of mesh I_n. If $R \geq th$, it indicates that the geometry is good, and if $R < th$, it means that the geometry is bad. This simple yet effective method also filters geometries that are relatively flat, such as edges with fillets as $(V_B - V_I) \approx 0$ which results in smaller Rs. We suggest using $th = 0.1$, which was constant in all experiments of this chapter.

A limitation of the volume filter (Figure 5.7) occurs when the mesh edges are at an angle which results in invalid values of R. This is due to excess volume in those empty regions of the bounding box, which increases R. We require the depth of geometry of interest, (d_n), which is the depth from the object surface to the point



(A) (B)

FIGURE 5.7: Volume ratio: (A) Volumes in R and the extracted depth of the geometry of interest (d) in four examples. Note that V_i is a subset of the mesh. Examples 1 and 2 return a large R (good geometries) while Examples 3 and 4 return $R \approx 0$ and $R = 0$ respectively (bad geometries); (B) Limitation of volume filter due to empty regions.

where the bounding box of I_n fully encloses the object, to check the validity of R. For each G_n, if $d_n = D$, any $R \neq 0$ is invalid (Example 4 of Figure 5.7). For geometries that lead to invalid R, we cluster the surface normals of I_n with similar vector angles. Bad geometries would have the largest cluster perpendicular to the surface of S, while good geometries would not. The filtering is complete as every G_n is either labeled as 'good' or 'bad'.

With the addition of the filter, the creation of P has three possible cases depending on the labels of every G_n:

1. *Only good geometries:* The three-step procedure in Figure 5.5 executed, resulting in Example A (Figure 5.6b).

2. *Only bad geometries:* A flat fingerpad with a thickness of $(T - D)$ is obtained in Example B (Figure 5.6b).

3. *Mixture of good and bad geometries:* For P to achieve good geometric constraints, the first two steps in Figure 5.5 are amended. Intersections are only applied for good geometries and a flat rectangle block B is included during the union to cater for the bad geometries (Examples C and D Figure 5.6b).

The depth of the flat rectangular block (d_B) depends on d_n, and $d_n \neq 0$ if and only if the geometries are good. As such, $d_B = min(d_1, d_2, ..., d_n) * K$, where K is a constant that affects the degree of 'flatness' of P. The minimum is considered rather than the maximum so that shallow complex geometries will not be absorbed away by B. We suggest using $K = 1.5$ which was constant in all experiments of this chapter.

The number of possible fingerpad combinations (C) depends on the number of valid sample pairs and the number of stable placements for planning (N_P). If $N_P = 2$ and one pose has three valid pairs of grasp surfaces ($N_{s,1} = 3$) while other pose has four valid pairs of grasp surfaces ($N_{s,2} = 3$), $C = N_{s,1} * N_{s,2} = 3 * 4 = 12$, meaning that there are 12 possible grippers.

5.4 Geometric grasp quality measure

A quantitative measure is needed to evaluate the grasp quality of synthesized gripper surfaces as the caging grasps and immobilization are performed based on a geometrical constraint [82, 83], which makes grasps insensitive to friction changes [58]. Thus, we propose a heuristic grasp quality measure that emphasizes the geometric grasp quality.

5.4.1 Variation of contact normals

In two-finger caging grasps, the concavity of the object is captured to create geometric constraints that immobilize the object [143–145], which may indicate that the contact surface between the gripper and object, e.g. concave surfaces, has sufficient varying contours that resist perturbations. Thus, a logical heuristic to define geometric grasp quality would be the representation of the variation of contact surface normals, where larger variations of surface normals indicate better grasp.

The variation is quantified by mapping every surface contact normal of the contact surface between a pair of fingers and a grasped object to a point on a unit sphere (Figure 5.8), defined as the Radius of the Largest Empty Sphere (RLES). A larger variation of normals would result in a better grasp and denser sphere, which leads to smaller RLES. Thus, a smaller RLES would indicate a better grasp. It is computed using a combination of 3D Voronoi vertices and Delaunay triangulation. Caroli *et al* [146] showed that the convex hull of the input points is equivalent to their Delaunay triangulation on the surface of the sphere. Megan [147] proposed a solution for the largest empty circle in 2D by using Voronoi vertices, as the edges of the Voronoi

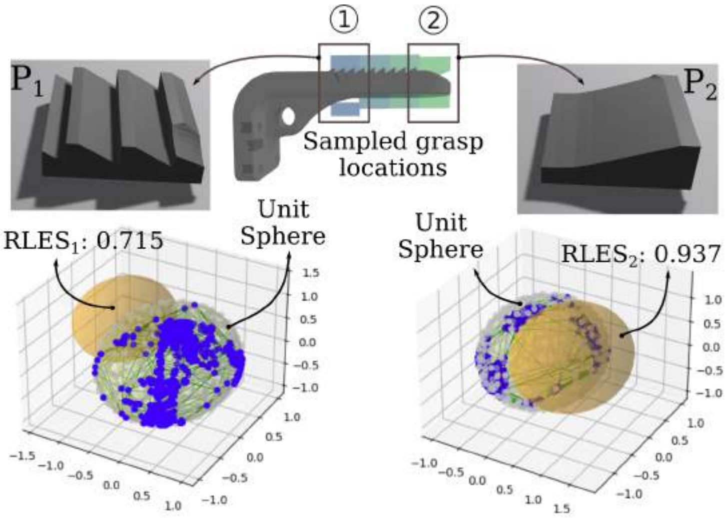


FIGURE 5.8: Quantifying the variation of contact surface normals of fingerpads produced at sampled grasp locations with RLES. Every surface contact normal is mapped to a dot (blue) on a unit sphere. A better grasp would have a larger variation, leading to more dots and smaller RLES.

regions are defined as the circumcenters of the triangles generated by Delaunay. Hence, the spherical Voronoi vertices are possible centers of an empty sphere that intersects any Delaunay triangle at its three ends. A search using KD-trees [148] is conducted to compute the RLES.

5.4.2 Total surface contact area

Although the variation of the contact normals may seem sufficient as a grasp quality measure, the total surface area in contact with the object during grasp (A) should also be considered to achieve full geometric constraint, as small grasping areas may cause unstable grasping even with large variations of surface normals. A is the sum of the areas of the finger pair in contact with the object, which is related to the surface normal variation to a certain extent. A larger contact surface would have larger variations if the object is not flat.

5.4.3 Quantifying geometric quality of grasps

Both the variation of contact normals and total contact area are deemed to be equally important. Thus, the effective area (E), is the geometric quality of the

i_{th} customized fingerpad at the m^{th} stable pose, by multiplying the inverse of RLES with the total surface contact area at m: $E_{i,m} = (1/RLES) * A_m$, where $i = 1, 2, ..., C$ and $m = 1, 2, ..., Np$. A larger E depicts a better quality as it indicates a larger A and better contact normal variation, i.e. smaller RLES.

Each pair of fingerpads are required to grasp object(s) at different pose(s), leading to varying qualities across grasps, i.e. a better grasp may be observed between objects and poses for the same fingerpad pair. Thus, the min-max concept is used, where the quality of the i^{th} fingerpad geometry is the worst possible grasp (minimum E) at the m^{th} stable pose: $Q_i = min(E_{i,1}, E_{i,2}, ..., E_{i,m})$. The geometric quality of the best (maximum Q) fingerpad geometry is then defined as the $Q_{max} = max(Q_1, Q_2, ..., Q_i)$. In simple terms, the grasp quality of each gripper is its worst possible grasp and the best gripper has the highest Q value at its worst grasp across all grippers.

5.5 Experiments

We evaluate our proposed pipeline from two perspectives: (1) Quantitative evaluation of geometric grasp quality measure (Section 5.5.1); (2) Qualitative evaluation of generated customized fingers for a set of objects and tests of actual pick-and-place experiments on objects at multiple poses (Section 5.5.2). Note that most objects used were real samples from HP Labs printed for certain industrial tasks. All objects and fingers are printed by the HP MJF5200 using PA11/PA12.

5.5.1 Evaluation of geometric grasp quality measure

We use the Stanford bunny object [149] to evaluate our geometric grasp quality measure with the following parameters:

- Robotiq Hand-E gripper (linear opening of 50mm) and its default flat fingers.
- Sampling was conducted with a stride $L/5$ and S has dimensions $L = 20, W = 20, T = 5, D = 4$.
- Two stable placements ($N_p = 2$) with T_1 and T_2 as the second and fourth object pose in Figure 5.9a respectively.

FCSO returned $N_{s,1} = 3$ for T_1 and $N_{s,2} = 3$ for T_2 (Figure 5.9b). The number of possible customized grippers would be $C = N_{s,1} * N_{s,2} = 3 * 3 = 9$ (Figure 5.9c) which were evaluated using our geometric grasp quality measure. Each gripper would need to achieve geometric constraints at four surfaces (two surfaces per grasp position as shown in Figure 5.5a). Table 5.1 shows the corresponding RLES value of i^{th} gripper fingerpad. It also depicts the effective area for the i^{th} fingerpad geometry at the m^{th} stable pose, $E_{i,m}$, and the quality for the i^{th} fingerpad: $Q_i = min(E_{i,T1}, E_{i,T2})$. The best gripper obtained was $i = 9$ with the highest Q.

From our experiment, we make two observations: (i) the quality measure requires considering the variation of contact normals and contact surface area to be effective; (ii) the measure is reasonably sufficient in determining the grasp quality as the result coincides with our intuition. The variation of contact normals alone may be insufficient as in Table 5.1, the best finger design would be $i = 1$ after taking the

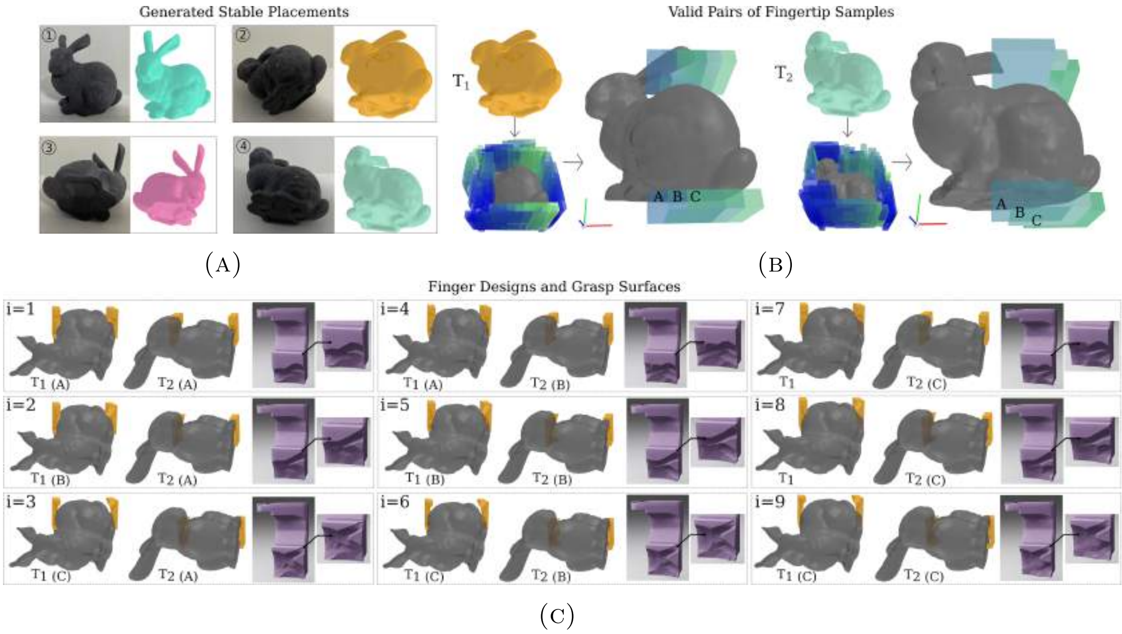


FIGURE 5.9: **Execution of FCSO** A) Stable pose generator (Section 5.2.1) returned four placements of the bunny and the second and fourth poses were randomly selected; (B) Grasp sampler (Section 5.2.2) returns three valid pairs of grasp surfaces (A, B and C) for each pose. T_1 depicts the bunny looking towards the left while T_2 shows the bunny looking upwards; (C) Fingerpad customization (Section 5.3) at these grasp surfaces returned nine possible customized fingerpads that are shown in orange. The gripper fingers (Section 5.2.3) obtained are shown in purple with the corresponding pose and grasp surface combinations.

TABLE 5.1: RLES, contact areas of fingerpads and grasp quality at two object poses.

	RLES		Contact area (A)		Grasp quality		
i	T_1	T_2	T_1	T_2	$E_{i,T1}$	$E_{i,T2}$	Q_i
1	0.4217	0.3878	102	17.9	242.2	46.2	46.2
2	0.5079	0.4668	110	103	216.8	221.1	216.8
3	0.5387	0.4553	126	130	234.3	285.9	234.3
4	0.5333	0.3680	103	25.9	193.2	70.4	70.4
5	0.476	0.6297	124	105	261.4	167.1	167.1
6	0.4543	0.5509	149	120	329.0	217.8	217.8
7	0.6040	0.4529	92.7	42.3	153.5	93.4	93.4
8	0.6515	0.5826	117	94.2	180.2	161.7	161.7
9	0.6233	0.4863	146	114.8	235.7	236.1	**235.7**

max-min of the RLES at every i. By visual inspection, $i = 9$ (Figure 5.9c) would provide the best geometric constraints due to more contouring details throughout the fingerpad, which coincides with the result from our proposed quality measure.

This measure was used to obtain grippers that achieved successful grasps in Section 5.5.2.

5.5.2 Evaluation of customized fingers

We evaluate the grippers from FCSO by conducting actual pick-and-place experiments for three objects: (1) Intricate cube (L30xW30xH30), (2) Stanford bunny (L65xW50xH65), and (3) L-shaped surgical object (L116xW60xH36). The cube and the L-shaped object are *real* samples produced in HP Labs for certain tasks for the industry, while the bunny was also used in [58], which could serve as a good comparison. These objects would be more suitable than datasets with common household items without customization such as YCB. The geometrical complexity of customized objects produced in additive manufacturing for the industry is also evident.

In all experiments, a Universal Robot (UR5e) executed at $15°/s$ joint speed and $10°/s^2$ joint acceleration was used together with a Robotiq Hand-E parallel gripper that has a linear opening of 50mm, specifies a grip force of $60N$ and closing speed of $20mm/s$. Individual pick-and-place experiments for three objects was conducted and snapshots of the experiment are shown in Figure 5.10.

Interestingly, our customized fingerpads contain the most distinct geometries of the object that aids in immobilizing the object. Securely grasping the bunny would seem difficult due to convex geometries and large dimensions compared to the gripper opening. Intuitively, the base of the bunny with small contours along the edges would be the best location to grasp. Our grasp sampler indeed return valid samples along these extrusions and these contours were also present in the gripper. This observation is also evident in both the cube and the L-shaped object, where the internal geometries of the cube and the zig-zag portion of the L-shaped object are present in their respective customized grippers.

A more difficult pick-and-place experiment for different objects and resting poses were also conducted. Objects used were the bunny and the L-shape object resting at two different positions (Figure 5.10). The best gripper returned would intuitively be the combination of the individual-best grippers for both objects and the result matched our intuition, allowing tightly constrained grasps across all objects and

(A) Customized fingers for the cube.

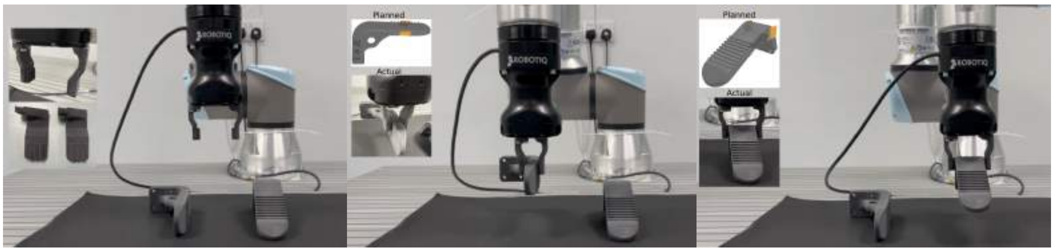
(B) Grasping the cube resting at first position.

(C) Grasping the cube resting at second position.

(D) Customized fingers for the bunny.

(E) Grasping at first pose: Bunny is looking at the camera.

(F) Grasping at second pose: Bunny is looking upwards.



(G) Customized fingers for L-shape object.

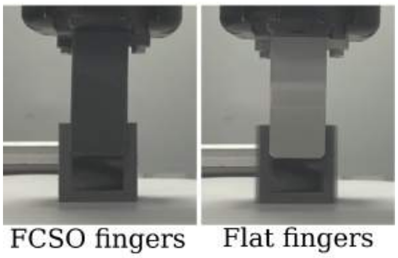
(H) Grasping the L-shape object resting at first position.

(I) Grasping the L-shape object resting at second position.

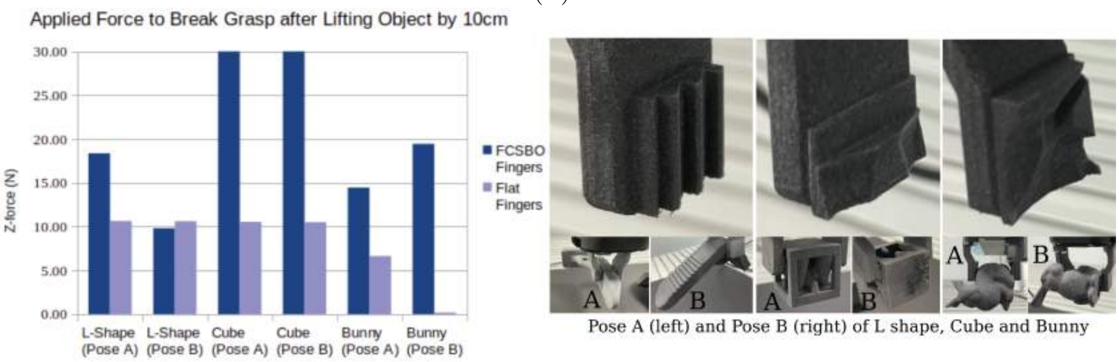
FIGURE 5.10: Snapshots of the pick-and-place experiments for three objects. The video is available at `https://youtu.be/M68YagfUF1g` which demonstrates the robustness of FCSO.

their resting positions, illustrating the ability of FCSO to generalize to different objects and positions.

Objects were manually placed without pose refinement to show that our gripper design may be robust to marginal position errors and uncertainty. Precise positioning can be obtained as the objects slide into contours of the gripper that conform to their geometries during grasps, as evident in Figure 5.11a. As caging grasps are essentially performed based on a geometrical constraint [82, 83], the grasp outcome is highly dependent on the geometry of the gripper rather than friction changes [58].



(A)



(B)

Figure 5.11: **Precision and stability tests.** A) Precision test: The cube was rotated from $-3°$ to $3°$ before executing 10 grasp attempts using FCSO fingers and flat fingers. Superimposed images of the cube after grasping showed that position was constant using FCSO fingers while there were positioning errors (shadows) using flat fingers; (B) Stability test: 3DP flat fingers and FCSO fingers were used to grasp and lift objects upwards for 10cm before applying a downward force (maximum $30N$) on the objects. The chart shows the average measurements after 3 readings. Note that grasps were not broken for both cube poses and the bunny at Pose B slipped out of grasp during the lift.

Thus, friction analysis was omitted. We also evaluated the holding force to show the stability of the grasps against flat fingers that were printed in the same material in Figure 5.11b. Note that both fingers were printed in Nylon powder, but FCSO fingers were in PA12 which is dark grey, and flat fingers were in PA11 which is light grey.

5.6 Conclusion

Customized robot grippers can be obtained by manual means or automation with optimization methods. High-precision customized grippers that closely conform shapes of given objects at planned locations might be inflexible to different locations or objects. Precision and versatility could depend on the gripper design and the objects in a task. With additive manufacturing, objects with diverse geometries could be easily attained to be used in research tasks or industrial production. It would be challenging to manually design grippers to cater to different objects as current automated designs may not be robust to such complex geometries.

We introduce an approach that automatically customizes optimal grippers that could achieve precise yet versatile grasping for complex objects. To evaluate the grasp quality, we emphasize the geometric grasp quality of the contact surfaces based on caging grasps and immobilization. Our geometric grasp quality measure shows to be reasonably sufficient in differentiating good grippers. We also demonstrated that the designed grippers can grasp multiple objects at different resting poses and is robust to marginal position errors as objects slide into conforming contours of the gripper.

A possible limitation could be the number of objects and scenarios that can be considered. Many objects or positions could lead to over-subtracting of geometries, which may result in relatively flat fingerpads. Future work could also involve improving FCSO for tasks that require a high standard of versatility and precision such as assembly tasks.

Chapter 6

Grasping, Part Identification, and Pose Refinement in One Shot with a Tactile Gripper

6.1 Introduction

The rise in AM comes with unique opportunities and challenges. Rapid changes to part design and massive part customization distinctive to 3DP can be easily achieved. Applying robotics in manufacturing industries is also an increasing trend [150], which can include tasks such as sorting and packing.

A key aspect of robotics application is robot perception, where information on the environment is obtained for the robot to plan and execute motions to perform tasks, such as grasping and manipulation. Previous studies were discussed in Section 2.3. Apart from using vision cameras, there are also studies on tactile perception for pose estimation and object classification that was propelled by the introduction of vision-based tactile sensors such as GelSight [112], GelSlim [113] and Digit [114]. Previous studies were discussed in Section 2.6.2.

However, the opportunity for massive customization comes with unique challenges for the existing production paradigm of robotics applications. Customized parts that are unique, yet exhibit similar features such as dental moulds, shoe insoles, or even engine vanes in turbo-machinery could be easily manufactured with 3DP.

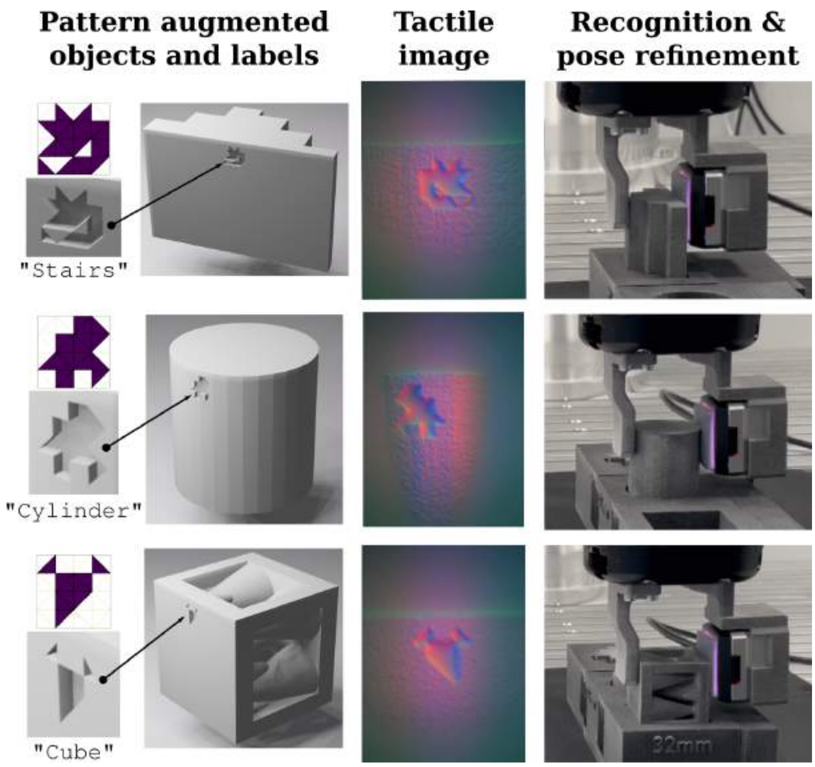


FIGURE 6.1: Pattern augmentation on 3DP parts for object recognition and high accuracy pose refinement to conduct insertion tasks. A video demonstration is available at `https://youtu.be/3e6gvkZUk8c`

Automatic identification of these parts would be difficult because shoe insoles for different people will have similar features, but they are not identical. Hence, it is challenging for deep learning methods to conduct identification because these methods are feature-based. As such, the advantage of part customization in 3DP has become a limitation, and manual imprinting of parts is often used instead. Thus, it is desirable to have automatic identification of these parts to enable end-to-end post-processing automation, such as sorting and packing.

This chapter explores the use of pattern augmentation on 3DP objects to execute grasping, part identification, and pose refinement in one shot with a tactile gripper, which is the first to the best of our knowledge. We want to explore the other capabilities that pattern augmentation can provide apart from identification. Our approach, which leverages the advantage of 3DP since the objects are supposed to be manufactured by 3DP, could correctly classify the objects based on their augmented patterns and also refine the pose to sub-millimeter accuracy for insertion tasks that mimic robotic packing (Figure 6.1), at a high success rate of 95%. Pattern augmentation allows unique patterns to correspond to objects thus enabling differentiation between similar objects. A major advantage of our method is that grasping, part identification, and pose refinement are conducted simultaneously, unlike the current production paradigm of robotics paradigm where the robot has to bring the grasped part to a camera. Additionally, upon extraction of the tactile imprint, part identification and pose refinement were achieved in 0.4s.

The rest of the chapter is as follows: Section 6.2 introduces our method, and Section 6.3 evaluates our approach from two perspectives through actual robotic experiments.

6.2 Methodology

This section discusses the creation of the pattern library and the overall workflow for object recognition and pose refinement of 3DP parts in practical tasks.

6.2.1 Overall pipeline

A graphical pipeline (Figure 6.2) shows the estimation of an initial pose of an object by a depth camera so that grasping can be conducted. After grasping, a vision-based tactile sensor captures the image of the imprint and the point cloud of the indentation. Image segmentation is performed on the imprint image to obtain the pattern mask. An example is Segment Anything Model (SAM) [151], an AI model that can "cut-out" all objects in an image. The pattern mask would be used to conduct object classification and obtain pose refinement. As each pattern in the library corresponds to an object, the original geometrical shapes of the objects would not be necessary for object recognition and pose refinement.

The object class label, L, can be obtained with the IoU loss [152] of the actual imprint I, against all other j^{th} pattern in the pattern library (S) where:

$$L = \min_{P_j \in S}\left(1 - \frac{(I \cap P_j)}{(I \cup P_j)}\right) \tag{6.1}$$

The IoU is an evaluation metric to measure the overlap of two regions, or patterns. A smaller IoU loss value indicates better similarity of I to P_j. The images of the

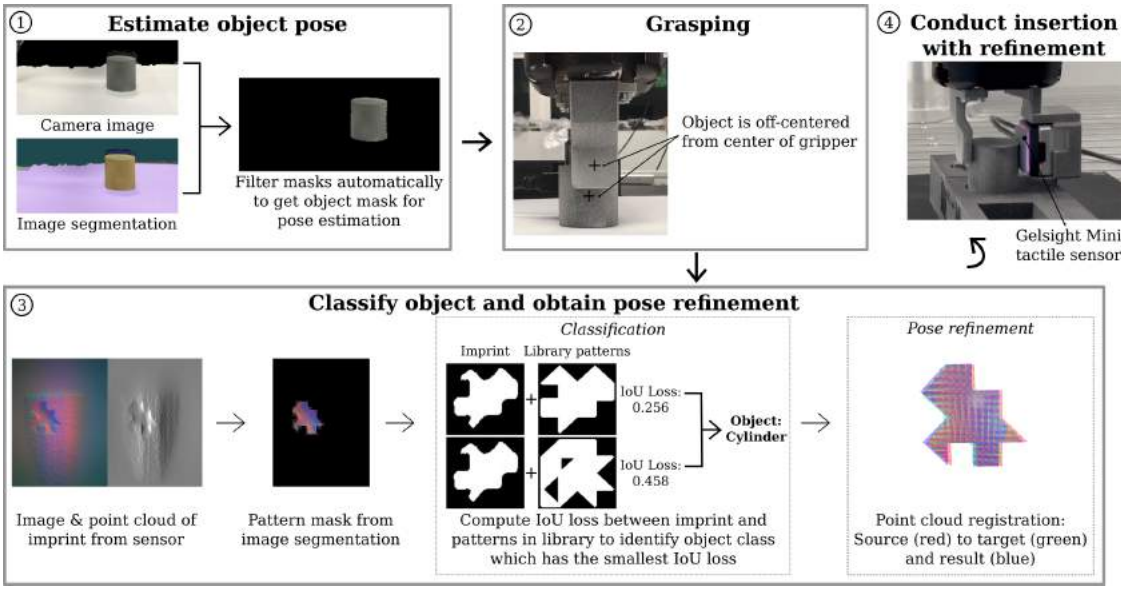


FIGURE 6.2: Graphical pipeline for object classification and pose refinement for pattern augmented 3DP objects.

patterns in the library are also pre-processed with a morphological transformation, namely dilation, by an elliptical structuring element to mimic smooth corners present in the actual imprint.

The actual imprint point cloud is cropped by its image mask and scaled to real-world values. Point cloud registration, such as FilterReg [3], is computed between the imprint point cloud and its corresponding point cloud in the pattern library that was identified during classification. The registration transformation (Figure 6.2), is the transformation of the source (identified pattern point cloud from the library) to the target (imprint point cloud). As such, pose refinement can be conducted as the transformation of the pattern w.r.t to the object can be obtained during the augmentation phase in Section 6.2.2. To improve the accuracy and computation time of the point cloud registration, the source point cloud is also subjected to an initial transformation by translating its centroid to the centroid of a box that bounds the mask of the imprint.

6.2.2 Creation and augmentation of pattern library

Small and unique features would aid feature matching in vision-based tactile sensors as discussed in Section 2.6.3. Thus, we propose the idea of augmenting small and unique patterns on 3DP parts to aid object recognition and pose estimation. In [153, 154], abstract patterns were created by placing triangular elements on a rectangular grid using a simulated annealing stochastic optimization algorithm [155]. We adopted the idea to optimize the triangular placements but with a Delaunay triangulation grid obtained by staggered row sampling [156] instead of a rectangular grid in [153, 154].

Patterns are generated by finding the triangle placement that can meet target connectivity, which is the optimization objective for simulated annealing, used together with a linear multiplicative cooling function. A random number of triangles, N, is selected for every pattern, and the target connectivity is a random number between $[N-2, N]$. Specifically, the connectivity is the number of triangles connected to their neighbors using a graph search. Higher connectivity seems desirable as empty regions between unconnected triangles may cause the formation of sub-patterns that may result in the non-uniqueness of the patterns. To ensure a certain degree of dissimilarity, or dispersion (δ) is present between patterns of the library, we use a distance measure, $d(P_i, P_j)$, based on Hu Moments [157] to conduct shape matching of a new pattern sample P_i against all other P_j patterns in the library, where $H_{m,i}$ and $H_{m,j}$ be the m^{th} log transformed Hu Moment for P_i and P_j. A smaller distance indicates greater similarity.

$$d(P_i, P_j) = \sum_{m=0}^{6} \frac{|H_{m,i} - H_{m,j}|}{|H_{m,i}|} \tag{6.2}$$

Next, for every new P_i, we ensure that the minimum dispersion of the pattern library, S, is greater than a threshold, α.

$$\delta(S) = \min_{P_i, P_j \in S} d(P_i, P_j) > \alpha \tag{6.3}$$

Pattern augmentation can be performed on the objects to printed with the pattern library. The augmentation locations of the patterns are fixed at the center of the plane on the side of the object, and offset by a small and fixed distance from the top edge, e.g. 1mm distance. These objects were properly orientated during the design phase to enable automated augmentation of the patterns. Blender was used to conduct Boolean difference on the objects with the pattern STL files, thus creating

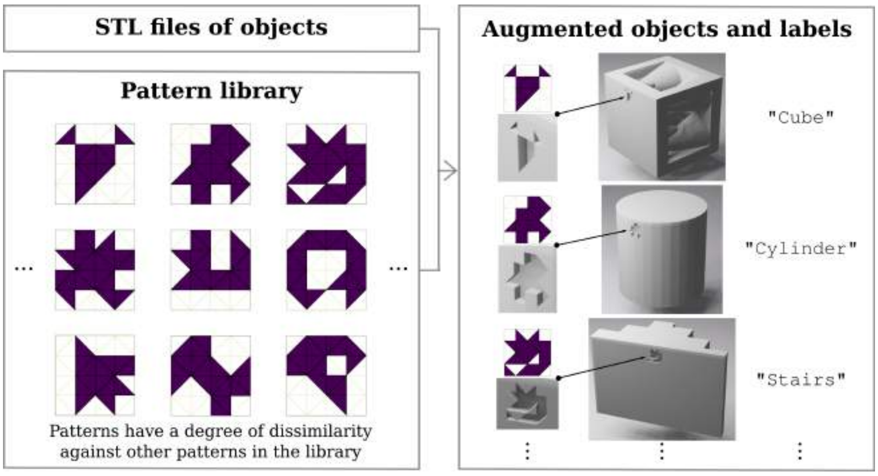


FIGURE 6.3: A unique pattern library is obtained by using simulated annealing to place triangles on a grid. The pattern library and the STL files of the objects are used to create pattern-augmented objects and their corresponding labels. The labels correspond to patterns rather than objects.

imprints of 1mm depth on the objects (Figure 6.3). Labels are automatically created by referencing the pattern number with the name of the object STL file.

Our pattern library of 1095 patterns was created with $N = [10, 20]$ on a 4x4 square Delaunay triangulation grid with $\alpha = 0.1$ using the libraries *Matplotlib* and *Scipy*. Some examples of the patterns obtained are shown in Fig. 6.3. Note that the grid size can be changed and the number of patterns in the library can be increased, as the number of patterns selected for this library is arbitrary. Expansion of a particular library could also be performed by computing $\delta(S)$ for every new P_i. The pattern size can be easily changed by scaling the grid. Our pattern size was scaled to 5mm and used in all experiments. The *trimesh* library was used to obtain the STL files of the patterns and subdivide the meshes to have more vertices, where the value used for subdivision was 0.1. These vertices are translated into voxelized point clouds using *Open3D* library, to be used in point cloud registration in the subsequent steps.

6.3 Experiments

We evaluate the effectiveness of pattern augmentation for 3DP parts in object recognition and pose refinement from three perspectives: (1) Evaluation of robustness of pattern augmentation technique, (2) Evaluation of insertion success rate and pose refinement accuracy, and (3) Evaluation with real insertion tasks to mimic packing parts into shadow boxes.

6.3.1 Specifications

We list some specifications used. In all experiments, a Universal Robot (UR5e) equipped with a Robotiq Hand-E parallel gripper with a flat finger and a GelSight Mini tactile sensor was used. Specifications of the workstation used are Intel Core i7-6700HQ CPU at 2.60GHz $\times$ 8 with NVIDIA Quadro M1000M graphics card. All objects were printed using the HP MJF5200 printer with nylon powder.

6.3.2 Evaluation of pattern augmentation technique

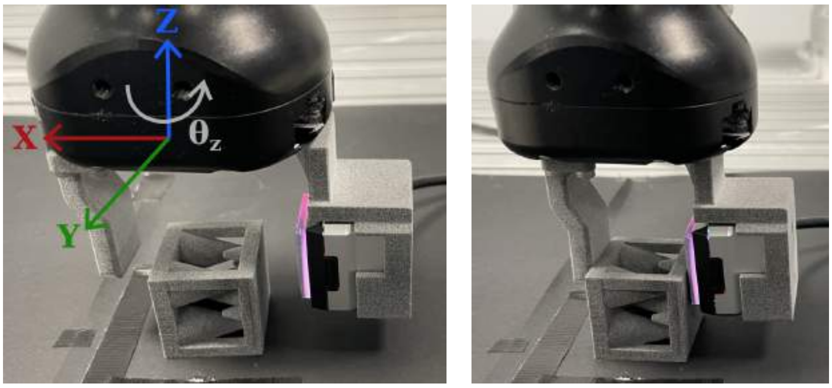
The robustness of the pattern augmentation technique was evaluated by conducting part identification for 30 randomly selected patterns from the library of 1095 patterns. Each pattern was augmented on the same cube as shown in Fig. 6.1, to depict a unique part. The cubes were grasped in the same initial position to capture the imprints and classification was executed with the procedure in Fig. 6.2. All 30 imprints were identified correctly, which illustrates the robustness of our pattern augmentation technique, where a certain degree of dissimilarity between the patterns in the library was ensured.

6.3.3 Evaluation of success rate and accuracy

The evaluation of insertion success rate and pose refinement accuracy was conducted with a physical peg-in-hole insertion task. The objective was to measure the insertion success rate when the robot manipulator was subjected to random perturbations. Specifications of the experiment are listed below:

- Insertion peg was a square cube measuring 30.2mm.
- Dimensions of square holes were 31.6mm and 30.7mm.

- Initial pose of gripper was subjected to random perturbations of (X, Y, θ_z), where X and Y ranges between [-2.5mm, 2.5mm] and θ_z ranges from [-3°, 3°] (Figure 6.4a).



(A) (B)

FIGURE 6.4: Random initial pose of robot manipulator: (A) Illustration of perturbation axes; (B) Cube initial position is unknown after grasping which resulted from the random perturbation of robot manipulator.

The initial position of the cube is unknown after grasping due to the random perturbation of the robot manipulator (Figure 6.4b). However, the position of the cube relative to the gripper can be extracted from the vision-based tactile sensor by point cloud registration between the real pattern imprint and the voxelized point cloud from the pattern dataset, thus allowing pose refinement for successful insertion which was discussed in Section 6.2.1. In a typical insertion task by picking an object from a plane, the pose refinement needed is the translation on the X-axis, Y-axis, and rotation θ_z. During grasping, the gripper fingers push the object to its centroid thus the offset of the object's centroid on the X-axis would be zero. In addition, θ_z could be obtained by extracting the rotation of the gripper. Hence, the only unknown variable needed is translation on the Y-axis, namely the refinement or compensation along the Y-axis (Y_{ref}).

The robot attempted 20 insertions for each hole dimension and the results are in Table 6.1 and Table 6.2, which illustrates a large improvement in success rate with pose refinement, that may be attributed to the unique features of the patterns which are well-captured by the vision-based tactile sensor. From the insertion experiment for the 31.6mm hole in Table 6.1, it can be seen that the refinement magnitude can be rather large at >3mm, while the hole allowance was only 1.4mm which indicates the effectiveness of our pattern augmentation method. Additionally, we were able

TABLE 6.1: Insertion of 30.2mm cube into 31.6mm hole

#	X (mm)	Y (mm)	θ_z (°)	Y_{ref} (mm)	Insert with refinement	Insert w/o refinement
1	-1.713	-1.747	-2.445	-3.396	✓	×
2	-0.046	-0.990	-1.615	-0.691	✓	×
3	-0.514	-0.492	-0.803	-1.756	✓	×
4	-0.567	-0.755	-1.187	-0.892	✓	✓
5	1.195	1.360	-2.487	0.132	×	×
6	2.406	0.817	2.648	0.361	✓	✓
7	1.799	1.242	-0.701	0.557	✓	×
8	-0.625	-2.182	-2.710	-3.235	✓	×
9	-1.999	2.129	-2.813	0.746	✓	×
10	-1.457	1.693	1.274	1.360	✓	×
11	-1.589	1.567	2.026	1.668	✓	✓
12	1.281	-2.031	2.218	-1.923	✓	×
13	1.914	-1.944	-0.511	-2.468	✓	×
14	-0.075	-0.093	0.269	-0.713	✓	✓
15	2.337	-1.786	1.989	-1.925	✓	×
16	2.186	-1.585	-0.208	-0.972	✓	×
17	1.124	-1.970	1.853	-2.117	✓	×
18	1.129	-2.493	-2.586	-2.993	✓	×
19	-1.708	-1.829	2.113	-2.065	✓	×
20	1.041	-0.008	1.2563	-0.429	✓	✓
	Success rate: From 25% to 95% with refinement					

to achieve a high success rate of 95% for a tight hole allowance of 1.4mm. Note that the Y_{ref} does not equate to the random Y perturbation of the manipulator as the actual Y_{ref} needed by the object would be affected due to the rotation of gripper (θ_z) because the gripper fingers will push the object during grasping.

To measure the pose refinement accuracy, we did experiments where known Y-offset values were applied to the manipulator. The target refinement value is the offset and the resulting compensation (Y_{ref}) is shown in Table 6.3, indicating good accuracy due to low percentage errors in sub-millimeter ranges. Thus, our method can conduct pose refinement of sub-millimeter accuracy.

TABLE 6.2: Insertion of 30.2mm cube into 30.7mm hole

#	X (mm)	Y (mm)	θ_z (°)	Y_{ref} (mm)	Insert with refinement	Insert w/o refinement
1	-0.881	-0.739	-2.965	-1.367	✓	×
2	1.671	1.950	-1.601	1.399	✓	×
3	-1.485	1.061	-1.578	1.280	×	×
4	-0.466	-2.488	1.161	-2.695	✓	×
5	-0.625	0.627	2.912	0.740	✓	✓
6	-0.634	1.959	-1.897	1.100	×	×
7	-1.021	-2.229	-0.525	-2.514	✓	×
8	-0.328	2.081	-0.186	1.335	✓	×
9	0.134	-1.210	-2.487	-1.950	×	×
10	-1.669	-2.251	1.749	-2.384	✓	×
11	-0.690	1.869	0.792	0.549	✓	×
12	-2.328	1.537	-0.750	0.009	×	×
13	0.481	1.714	0.642	1.384	✓	×
14	2.222	-1.305	0.660	-1.721	✓	×
15	-0.670	2.310	2.552	2.888	×	×
16	2.067	2.278	-2.948	1.904	×	×
17	-1.004	0.758	-2.152	-0.504	×	×
18	0.752	-2.479	1.140	-2.707	✓	×
19	1.355	1.094	1.606	1.451	✓	×
20	2.149	0.0980	-1.706	-0.067	✓	✓
	Success rate: From 10% to 60% with refinement					

TABLE 6.3: Evaluating pose refinement accuracy.

Y-offset (mm)	Compensation (mm)	Actual Error (mm)
-3.0	-2.964	-0.036
-2.0	-2.060	0.06
-1.0	-0.976	-0.024
1.0	0.846	0.154
2.0	1.715	0.285
3.0	2.605	0.395

6.3.4 Evaluation of the implementation for robotic tasks

The evaluation of the pattern augmentation implementation was conducted by physical insertion tasks that mimic robotic sorting and packing (Figure 6.5). Specifically, three 3DP parts with augmented patterns (Figure 6.3) were placed at a random position on a table and the robot needed to pick and pack them in their respective shadow boxes, or boxes with holes, and their dimensions are below. Note that the stairs and cube were real samples from HP Labs used for certain industrial tasks, and the HP MJF5200 printer has sub-millimeter tolerances.

1. Stairs with L46.2mm by W20.3mm to fit into L48.5mm by W22.3mm hole.
2. Cylinder with a diameter of 30.2mm to fit into a 31.5mm diameter hole.
3. Cube with L30.2mm to fit into L31.6mm hole.

In the experiment, an initial pose estimate of the object was obtained by a L515 Intel RealSense depth camera for the robot to conduct grasping. Upon grasping, the vision-based tactile sensor provides the RGB image and point cloud of the pattern imprint. As discussed in Section 6.2.1, the pattern imprint would be matched with the pattern library to get the correct object class label and the refinement transformation required, which only took 0.4s once the pattern mask was obtained. In addition, although only three objects were used in the experiment, each pattern was matched to a pattern library of 1095 patterns and was still able to quickly identify the correct labels. Note that these patterns used were different from the 30 patterns used in Section 6.3.2. Due to the set-up of the experiment, we would only need to compensate along the Y-axis as mentioned in Section 6.3.3. The robot then moves to the correct shadow box, conducts pose refinement, and successfully inserts all objects into their respective shadow boxes. Thus, this practical example shows that pattern augmentation on 3DP parts is a viable method to achieve grasping, part identification, and pose refinement in one-shot robotic tasks.

We used SAM [151] to obtain the pattern mask. Although SAM is non-specific and claimed to be unachievable in real-time, real-time performance could be achieved with specific models like Mask R-CNN [158] which could return in 0.2s, or using the improved model, Fast SAM [159], that claims to be 50 times faster than SAM. In total, our approach should take less than 0.6s, which is faster than any approach that relies on a middle station for precise vision-based pose estimation.

Although the object was placed such that the pattern faces the tactile sensor, we assume that prior reorientation could be achieved such that the pattern would always be visible on the grasping surface of the tactile sensor. This is a valid assumption as previous works have illustrated that object shape estimation can be achieved using vision-based tactile sensors, such as in [122], where the authors

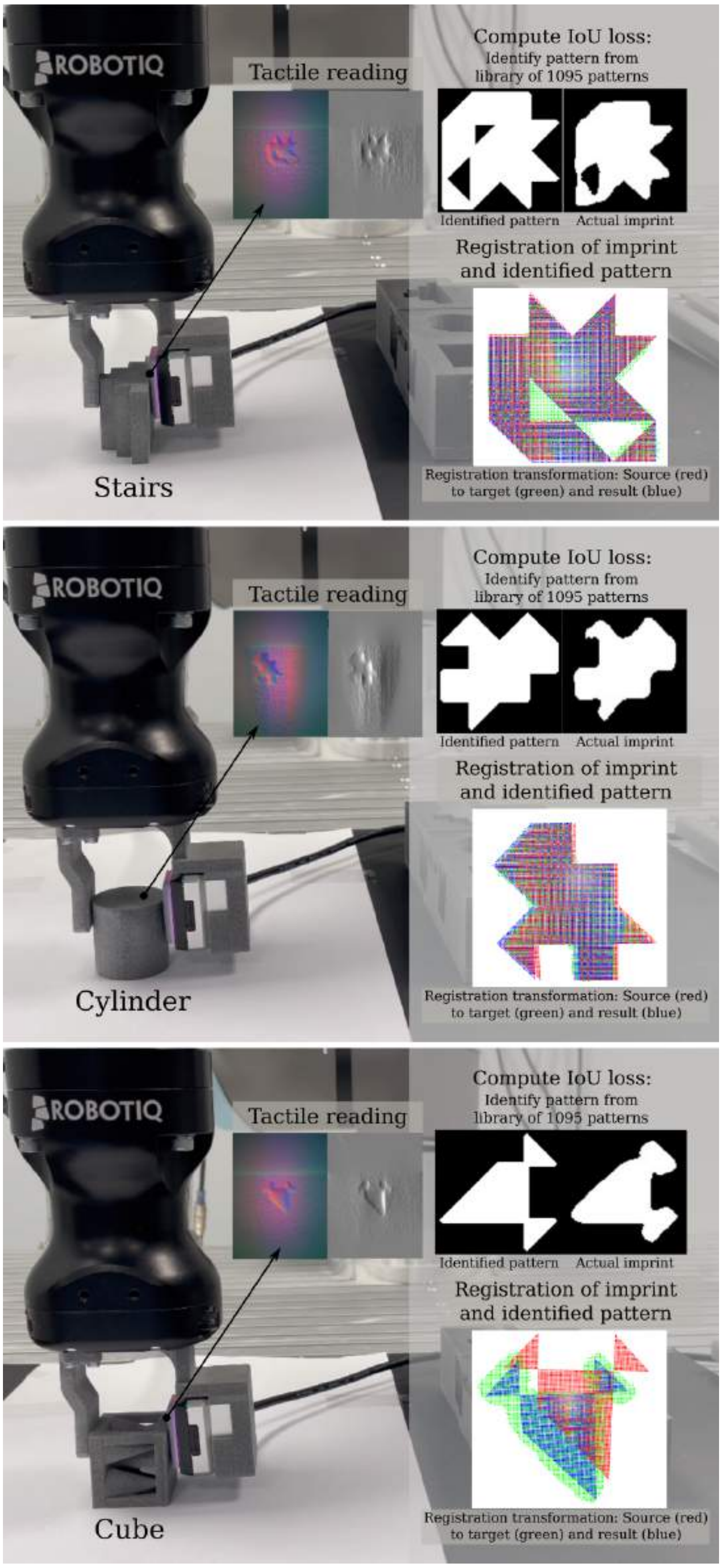


FIGURE 6.5: Experiment snapshots to mimic robotic sorting and packing into shadow boxes for three objects with pattern augmentations. The dimensional allowance between the objects and holes ranges from 1.3mm to 2.3mm. The video is available at `https://youtu.be/3e6gvkZUk8c`

claimed that an initial grasp attempt based on the initial guess of the overall shape is capable of providing information of the far side of the object. Other possible methods to achieve shape estimation can also include using a depth camera [160] or 2D images [161]. With the object shape estimation, we would be able to determine the grasp position that allows the pattern to be visible on the tactile sensor as the transformation of the pattern relative to the object is known, as discussed in Section 6.2.2.

6.4 Conclusion

Competitive additive manufacturing technologies come with a major bottleneck of manual 3DP post-processing. The ability to customize also creates unique challenges for the existing paradigm of robotics applications, thus creating limitations for end-to-end 3DP post-processing automation. Unique customized parts with similar features, such as shoe insoles, dental moulds, and engine vanes can be easily manufactured. However, automatic identification of these parts could be challenging with feature-based methods. Thus, we explore the use of pattern augmentation on 3DP objects to execute grasping, part identification, and pose refinement in one shot with a tactile gripper. This method also leverages the advantage of 3DP since the parts are to be manufactured by 3DP. With pattern augmentation, parts with similar features can be automatically differentiated as each pattern corresponds to one specific part, rather than having to take into account the objects' geometries for identification. We experimentally evaluate our method from three perspectives, including real tasks that mimic robotic sorting and packing, and achieved excellent classification results, a high insertion success rate of 95%, and sub-millimeter pose refinement accuracy. In total, our approach should take less than 0.6s, which is faster than any approach that relies on a middle station for precise vision-based pose estimation.

A current limitation to our work is that planning of grasps to ensure that the patterns would be visible on the sensor has not been considered, which could be an exciting direction for future work especially when parts of varying sizes are involved. In addition, although a certain degree of dissimilarity was imposed on the pattern library, the patterns could be further improved for optimal dispersion.

Chapter 7

Conclusion and future work

7.1 Summary

The rise in AM comes with unique challenges and opportunities. Large volumes of parts, customized objects, and rapid design changes are made achievable with AM. However, a major drawback of AM stems from the need for post-production processes, for example, part cleaning, painting, sorting, and packing. Currently, these processes rely heavily on manual labour which is tedious, repetitive and expose the operators to hazardous substances. Therefore, it is desirable to introduce robotics and automation in 3DP post-processing.

In this thesis, our focus is only on parts printed by powder-based AM technologies. Thus, the two main challenges in implementing robotics and automation in 3DP post-processing are that (1) the introduction of powder into the environment creates challenges, especially for robot perception, and (2) the opportunity for massive part customization poses challenges to the existing production paradigm of robotics applications. These challenges are significant as many previous works focus on objects from databases (YCB [162], etc.), whereas 3DP parts tend to consist of complex geometries due to massive customization. Additionally, many approaches involve the use of learning-based methods that are object-dependent, hence it may be challenging to adopt these methods in an environment where objects at tasks may be changed at will. Our research objectives aim to develop generalized solutions to support the unique challenges posed by 3DP post-processing.

Firstly, in Chapter 3, we develop a fully functional robotic prototype for the automated removal of residue powder of 3DP parts in a fast and efficient manner, by mimicking the brushing action of a human. This is the first robotics prototype that is capable of performing 3DP part cleaning.

Secondly, in Chapter 4, to support robot perception with the presence of powder, we propose a fully automated sim-to-real vision pipeline for deep classification and localization of parts covered in powder, with only the CAD models of the objects as input, and achieved high detection rates for unseen 3DP parts covered in powder.

Thirdly, in Chapter 5, to support robot grasping and manipulation of *batch-produced* customized parts, we present an automated gripper customization method that designs versatile gripper fingers to grasp and manipulate a batch of objects resting at various positions with high precision. We also proposed a novel geometric grasp quality measure based on contact geometry. Our method was able to design grippers to pick and place multiple objects at different resting positions.

Finally, in Chapter 6, to support identification and manipulation of *unique parts with similar features*, we introduce a method of pattern augmentation on 3DP parts to perform grasping, part identification, and pose refinement in one shot with a tactile gripper. This allows distinguishment between parts with similar features, such as shoe insoles, dental molds, and engine vanes. We achieved sub-millimeter pose estimation accuracy with a tactile gripper and high success rates in real insertion tasks that mimic automated sorting and packing.

In conclusion, we illustrate the integration of the different contributions in Figure 7.1. In the clean stage, there are two contributions for decaking: the robotic cleaning system[1], and vision capabilities[2]. In the sorting and packing stage, the two contributions depend on the type of objects printed: (a) Batch-produced parts[3], or (b) Unique customized parts with similar features[4]. To complete the visualization

[1]Contribution 1: Development of a Robotic System for Automated Decaking of 3D Printed Parts (Chapter 3)

[2]Contribution 2: Automated post-processing of 3D-printed parts: Artificial powdering for deep classification and localization (Chapter 4).

[3]Contribution 3: Automatic Fingerpad Customization for Precise and Stable Grasping of 3D-Print Parts (Chapter 5).

[4]Contribution 4: Grasping, Part Identification, and Pose Refinement in One Shot with a Tactile Gripper (Chapter 6).

FIGURE 7.1: Illustration of possible integration of different contributions in three stages of 3DP: (A) Print; (B) Clean; and (C) Sort and Pack. Post-processing tasks would begin after printing.

of the integration, we included work done by another researcher in the HP-NTU Corp Lab for fine cleaning[5].

[5]Research contribution [163] by other members of the research group at HP-NTU Corp Lab.

7.2 Future work

Although we have proposed several methods to aid the introduction of robotics and automation into 3DP post-processing, there are still several challenges before practical implementation could be achieved. We sketch several interesting and possible directions for future research in this section.

In this thesis, we show a proof-of-concept demonstration of the automated cleaning of residue powder of shoe insoles by mimicking the brushing action of a human. However, we have yet to explore true 3D cleaning motions and the efficiency could be improved by including a cleanliness evaluation module to optimize cleaning motions based on feedback received on areas that are not fully cleaned. The use of the suction cup could also be limiting to the cleaning efficiency, thus redesigning the workcell, such that brushes are equipped on the end-effector, could also be another possible direction for future work.

We also proposed several methods revolving around robot perception, grasping, and manipulation. However, it is possible that successful robotic grasping and manipulation may still require further improvements such as specific manipulation techniques based on the size and material of the parts. Parts that are huge in size would also affect the manner in which grasping and manipulation are executed, which may require appropriate grasp planning methods for highly customized parts.

In addition, due to equipment restrictions, the thesis only considered parts printed with nylon powder. Metal part printing is another powder-based AM process that also requires post-processing. However, nylon parts are considerably more sturdy and light compared to metal parts that would be heavy and brittle. Hence, material identification may have to be incorporated into the workflow as well, which may be achieved by using tactile sensors to sense textures.

The combination of brittle and heavy properties in metal parts would pose interesting challenges. Careful grasping and manipulation would be required to prevent part breakage, yet due to the weight of these parts, larger forces might be needed during grasping to ensure secure grasps. Planning of grasps for these parts would also be a challenge, as the location of overhanging portions of an object would need to be accounted for, to prevent breakage due to its own weight. In addition, the design of soft grippers may be suitable for brittle metal parts, such as using jamming grippers, but this would create additional challenges in high accuracy pose estimation or pose refinement as the object would be perturbed during grasping. Thus, the change in printing material might be another huge area for potential research.

Bibliography

[1] LeeringHengelo. *Depowdering of 3D-Printed Products - Normfinish 3D Smart - Blast Cabinet*. LeeringHengelon, 2020. URL https://www.youtube.com/watch?v=BOUIXySq1rA.

[2] Kenta-Tanaka et al. *probreg*, 2019. URL https://probreg.readthedocs.io/en/latest/.

[3] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11095–11104, Long Beach, CA, USA, 16-17 Jun 2019.

[4] Shaoxiong Wang, Yu She, Branden Romero, and Edward Adelson. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6468–6475, Xi'an China, 30 May - 05 Jun 2021.

[5] E. Atzeni, L. Iuliana, P. Minetola, and A. Salmi. Proposal of an innovative benchmark for accuracy evaluation of dental crown manufacturing. *Computers in Biology and Medicine*, 42(5):197–211, 2012.

[6] T. Wohlers and T. Gornet. History of additive manufacturing., 2014. URL http://wohlersassociates.com/history2014.pdf.

[7] C. K. Chua and K. F. Leong. *3D Printing and Additive Manufacturing*. World Scientic Publishing Co Pte Ltd, 5th edition, 2017.

[8] A. Aramian, S. M. J. Razavi, Z. Sadeghian, and F. Berto. A review of additive manufacturing of cements. *Additive Manufacturing*, 33:101130, 2020.

[9] Hewlett-Packard. HP MJF website, 2020. URL https://www8.hp.com/sg/en/printers/3d-printers/products/multi-jet-technology.html. Copyright 2020 HP Development Company, L.P.

[10] T. Grimm. 3d printing: The impact of post-processing, 2019. URL https://www.techbriefs.com/component/content/article/tb/pub/features/articles/33589.

[11] Beamler Additive Manufacturing. Post-processing in 3d printing, 2019. URL https://www.beamler.com/post-processing-3d-printing/.

[12] A. Ju, A. Fitzhugh, J. Jun, and M. Baker. Improving aesthetics through post-processing for 3d printed parts. *Internaltion Symposium on Electronic Imaging*, 6:480-1–480-4, 13-17 Jan 2019.

[13] Wohlers Associates. Wohlers report 2018 shows dramatic rise in metal additive manufacturing and overall industry growth of 21%, 2018. URL `https://wohlersassociates.com/press74.html`.

[14] S. Nelaturi, M. Behandish, A. M. Mirzendehdel, and J. Kleer. Automatic support removal for additive manufacturing post processing. *Computer-Aided Design*, 115:135–146, 2019.

[15] S. Raikar, M. Heilig, A. Mamidanna, and O. J. Hildreth. Self-terminating etching process for automated support removal and surface finishing of additively manufactured Ti-6Al-4V. *Additive Manufacturing*, 2020. In press.

[16] Y. Zhang, Z. Wang, Y. Zhang, S. Gomes, and A. Bernard. Bio-inspired generative design for support structure generation and optimization in Additive Manufacturing (AM). *CIRP Annals*, 69:117–120, 2020.

[17] J. S. Cuellar, G. Smit, D. Plettenburg, and A. Zadpoor. Additive manufacturing of non-assembly mechanisms. *Additive Manufacturing*, 21:150–158, 2018.

[18] A. B. Anwar and Q.C. Pham. Study of the spatter distribution on the powder bed during selective laser melting. *Additive Manufacturing*, 22:86–97, 2018.

[19] B. L. DeCost and E. A. Holm. Characterizing powder materials using keypoint-based computer vision methods. *Computational Materials Science*, pages 438–445, 2017.

[20] H. Baumgart, J. T, R. Buettner, and M. Merkel. A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring. *Progress in Additive Manufacturing*, page 277–285, 2020.

[21] T.B. Moeslund, C.B. Madsen, M. Aagaard, and D. Lerche. Modeling falling and accumulating snow. *Vision, Video and Graphics*, 2005.

[22] T. Nishita, H Iwasaki, Y. Dobashi, and E. Nakamae. A modeling and rendering method for snow by using metaballs. *EUROGRAPHICS*, 16, 4-8 Sep 1997.

[23] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. *ACM Transactions on Graphics*, 32, 2013.

[24] Iván Alduán, Angel Tena, and Miguel A Otaduy. Simulation of high-resolution granular media. In *Spanish Computer Graphics Conference (CEIG)*, pages 11–18, San Sebastián, Spain, 09-11 Sep 2009.

[25] HJ Herrmann and Stefan Luding. Modeling granular media on the computer. *Continuum Mechanics and Thermodynamics*, 10:189–231, 1998.

[26] Colin Thornton. Numerical simulations of deviatoric shear deformation of granular media. *Géotechnique*, 50(1):43–53, 2000.

[27] Dietrich E Wolf. Modelling and computer simulation of granular media. In *Computational Physics: Selected Methods Simple Exercises Serious Applications*, pages 64–95. Springer, 1996.

[28] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[29] G Lowe. Sift-the scale invariant feature transform. *Int. J*, 2(91-110):2, 2004.

[30] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571, Barcelona, Spain, 06-13 Nov 2011.

[31] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition*, volume 1, pages 886–893, San Diego, California, 20-26 June 2005.

[32] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[33] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[34] Nalpantidis Lazaros, Georgios Christou Sirakoulis, and Antonios Gasteratos. Review of stereo vision algorithms: from software to hardware. *International Journal of Optomechatronics*, 2(4):435–462, 2008.

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, Columbus, OH, USA, 23-28 June 2014.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. *European Conference on Computer Vision*, pages 21–37, 11–14 Oct 2016.

[37] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 21-26 Jul 2017.

[38] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 22-29 Oct 2017.

[39] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European conference on computer vision (ECCV)*, pages 269–286, Marseille, France, 12-18 Oct 2018.

[40] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, Munich, Germany, 08-14 Sep 2018.

[41] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3836, Venice, Italy, 22-29 October 2017.

[42] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision*, pages 548–562, Daejeon, Korea, 5-9 Nov 2012.

[43] Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global hypothesis generation for 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 462–471, Honolulu, USA, 21-26 Jul 2017.

[44] Zhikai Dong, Sicheng Liu, Tao Zhou, Hui Cheng, Long Zeng, Xingyao Yu, and Houde Liu. Ppr-net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1773–1780, Macao, Macau, 04-08 Nov 2019.

[45] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606, Boston, MA, United States, 30 Apr 1992. Spie.

[46] Osamu Hirose. A bayesian formulation of coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 43(7):2269–2286, 2020.

[47] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[48] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanam, P. Dollar, and C. L. Zimick. Microsoft COCO: Common objects in context. *European conference on computer vision*, pages 740–755, 06-12 Sep 2014.

[49] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, Swansea, UK, 07-10 Sep 2015.

[50] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[51] E. Buls, R. Kadikis, R. Cacurs, and J. Arents. Generation of synthetic training data for object detection in piles. *Proc.SPIE 11041 Eleventh International Conference on Machine Vision*, 23-25 Apr 2018.

[52] D. Dwibedi, I. Misra, and Hebert M. Cut, paste and learn: Surprisingly easy synthesis for instance detection. *IEEE International Conference on Computer Vision (ICCV)*, 22-29 Oct 2017.

[53] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 24-28 Sep 2017.

[54] J. Tobin and et al. Domain randomization and generative models for robotic grasping. *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 04-08 Nov 2019.

[55] P. S. Rajpura, H. Bojinov, and R. S. Hedge. Object detection using deep CNNs trained on synthetic images. *arXiv preprint, arXiv:1706.06782*, 2017.

[56] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg. Segmenting unknown 3D objects from real depth images using Mask R-CNN. *IEEE International Conference on Robotics and Automation (ICRA)*, 20-24 May 2019.

[57] Gualtiero Fantoni, Marco Santochi, Gino Dini, Kirsten Tracht, Bernd Scholz-Reiter, Juergen Fleischer, Terje Kristoffer Lien, Guenther Seliger, Gunther Reinhart, Joerg Franke, et al. Grasping devices and methods in automated production processes. *CIRP Annals*, 63(2):679–701, 2014.

[58] Haoran Song, Michael Yu Wang, and Kaiyu Hang. Fingertip surface optimization for robust grasping on contact primitives. *IEEE Robotics and Automation Letters*, 3(2):742–749, 2018.

[59] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.

[60] Zak Flintoff, Bruno Johnston, and Minas Liarokapis. Single-grasp, model-free object classification using a hyper-adaptive hand, google soli, and tactile sensors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1943–1950, Madrid, Spain, 01-05 Oct 2018.

[61] Duc Truong Pham, Nasir Salah Gourashi, and Eldaw Elzaki Eldukhri. Automated configuration of gripper systems for assembly tasks. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(11):1643–1649, 2007.

[62] Lucian Balan and Gary M Bone. Automated gripper jaw design and grasp planning for sets of 3d objects. *Journal of Robotic Systems*, 20(3):147–162, 2003.

[63] VB Velasco and Wyatt S Newman. Computer-assisted gripper and fixture customization using rapid-prototyping technology. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 4, pages 3658–3664, Leuven, Belgium, 16-21 May 1998.

[64] Mohammadali Honarpardaz, Johan Ölvander, and Mehdi Tarkian. Fast finger design automation for industrial robots. *Robotics and Autonomous Systems*, 113:120–131, 2019.

[65] Mohammadali Honarpardaz, Mehdi Tarkian, Xiaolong Feng, Daniel Sirkett, and Johan Ölvander. Generic automated finger design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 50169, page V05BT07A071, New York, N.Y., 21-24 Aug 2016.

[66] Vincent Wall, Gabriel Zöller, and Oliver Brock. A method for sensorizing soft actuators and its application to the rbo hand 2. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4965–4970, 29 May-03 Jun 2017. doi: 10.1109/ICRA.2017.7989577.

[67] Jingyi Xu, Tamay Aykut, Daolin Ma, and Eckehard Steinbach. 6dls: Modeling nonplanar frictional surface contacts for grasping using 6-d limit surfaces. *IEEE Transactions on Robotics*, 2021.

[68] Jun Shintake, Vito Cacucciolo, Dario Floreano, and Herbert Shea. Soft robotic grippers. *Advanced materials*, 30(29):1707035, 2018.

[69] Wookeun Park, Seongmin Seo, and Joonbum Bae. A hybrid gripper with soft material and rigid structures. *IEEE Robotics and Automation Letters*, 4(1): 65–72, 2019. doi: 10.1109/LRA.2018.2878972.

[70] Raymond Ma and Aaron Dollar. Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Robotics & Automation Magazine*, 24(1):32–40, 2017.

[71] Vatsal V Patel and Aaron M Dollar. Robot hand based on a spherical parallel mechanism for within-hand rotations about a fixed point. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 709–716, Prague, Czech Republic, 27 Sep - 01 Oct 2021.

[72] Liang Zhou, Lili Ren, You Chen, Shichao Niu, Zhiwu Han, and Luquan Ren. Bio-inspired soft grippers based on impactive gripping. *Advanced Science*, 8 (9):2002017, 2021.

[73] Taogang Hou, Xingbang Yang, Yasumichi Aiyama, Kaiqi Liu, Zeyu Wang, Tianmiao Wang, Jianhong Liang, and Yubo Fan. Design and experiment of a universal two-fingered hand with soft fingertips based on jamming effect. *Mechanism and Machine Theory*, 133:706–719, 2019.

[74] Amir Firouzeh and Jamie Paik. Grasp mode and compliance control of an underactuated origami gripper using adjustable stiffness joints. *Ieee/asme Transactions on Mechatronics*, 22(5):2165–2173, 2017.

[75] W Wang and SH Ahn. Non-intrusive load monitoring system for anomaly detection based on energy disaggregation by cascading semi-supervised learning and deep learning methods. *Soft Robotics*, 3:379, 2017.

[76] J Shintake, H Shea, and D Floreano. Ieee/rsj int. conf. intelligent robots and systems, 09-14 Oct 2016.

[77] N. Correll and et al. Lessons from the amazon picking challenge. *arXiv preprint arXiv:1601.05484*, 2016.

[78] Domenico Prattichizzo and J.C. (Jeff) Trinkle. *Grasping*, pages 671–700. Springer Science and Business Media, 01 2008. doi: 10.1007/978-3-540-30301-5_29.

[79] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, USA, 1st edition, 2017. ISBN 1107156300.

[80] Elon Rimon and Joel W Burdick. Mobility of bodies in contact. i. a 2nd-order mobility index for multiple-finger grasps. *IEEE transactions on Robotics and Automation*, 14(5):696–708, 1998.

[81] Elon Rimon and Joel Burdick. Mobility of bodies in contact. ii. how forces are generated by curvature effects? In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2336–2341, San Diego, CA, USA, May 1994.

[82] Satoshi Makita and Weiwei Wan. A survey of robotic caging and its applications. *Advanced Robotics*, 31(19-20):1071–1085, 2017. doi: 10.1080/01691864.2017.1371075. URL `https://doi.org/10.1080/01691864.2017.1371075`.

[83] Elon Rimon and AFVD Stappen. Immobilizing 2-d serial chains in form-closure grasps. *IEEE Transactions on Robotics*, 28(1):32–43, 2011.

[84] Tomas Lozano-Perez, Joseph L Jones, Patrick A O'Donnell, and Emmanuel Mazer. *Handey: a robot task planner*. Mit Press, 1992.

[85] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[86] M. Malvezzi, G. Gioioso, G. Salvietti, and D. Prattichizzo. Syngrasp: A matlab toolbox for underactuated and compliant hands. *Robotics Automation Magazine, IEEE*, 22(4):52–68, Dec 2015. ISSN 1070-9932. doi: 10.1109/MRA.2015.2408772.

[87] Corey Goldfeder, Peter K Allen, Claire Lackner, and Raphael Pelossof. Grasp planning via decomposition trees. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4679–4684, Roma, Italy, 10-14 Apr 2007.

[88] Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. In *7th IEEE-RAS International Conference on Humanoid Robots*, pages 42–48, Pittsburgh, PA, USA, 29 Nov - 01 Dec 2007.

[89] Mehmet R Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. *Robot.: Sci. Syst. VIII*, page 57, July 2012.

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

[91] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022.

[92] Lirui Wang, Xiangyun Meng, Yu Xiang, and Dieter Fox. Hierarchical policies for cluttered-scene grasping with latent plans. *IEEE Robotics and Automation Letters*, 7(2):2883–2890, 2022.

[93] Máximo A Roa and Raúl Suárez. Grasp quality measures: review and performance. *Autonomous robots*, 38(1):65–88, 2015.

[94] Carlo Ferrari and John F Canny. Planning optimal grasps. *IEEE International Conference on Robotics and Automation (ICRA)*, 3(4):6, 12-14 May 1992.

[95] Jian Ruan, Houde Liu, Anshun Xue, Xueqian Wang, and Bin Liang. Grasp quality evaluation network for surface-to-surface contacts in point clouds. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1467–1472, Hong Kong, 20-21 Aug 2020.

[96] Michael Danielczuk, Jingyi Xu, Jeffrey Mahler, Matthew Matl, Nuttapong Chentanez, and Ken Goldberg. Reach: Reducing false negatives in robot grasp planning with a robust efficient area contact hypothesis model. In *International Symposium on Robotics Research (ISRR)*, Hanoi, Vietnam, 06-10 Oct 2019.

[97] Jingyi Xu, Michael Danielczuk, Eckehard Steinbach, and Ken Goldberg. 6dfc: Efficiently planning soft non-planar area contact grasps using 6d friction cones. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7891–7897. Paris, France, 31 May - 31 Aug 2020.

[98] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26 (1):1–20, 2009.

[99] Ronald S Fearing and Thomas O Binford. Using a cylindrical tactile sensor for determining curvature. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 765–771. IEEE, 1988.

[100] R Andrew Russell and Simon Parkinson. Sensing surface shape by touch. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 423–428, Georgia, USA, May 1993.

[101] M Charlebois, Kamal Gupta, and Shahram Payandeh. On estimating local shape using contact sensing. *Journal of Robotic Systems*, 17(12):643–658, 2000.

[102] Richard Crowder. Toward robots that can sense texture by touch. *Science*, 312(5779):1478–1479, 2006.

[103] Vivek Maheshwari and Ravi F Saraf. High-resolution thin-film device to sense texture by touch. *Science*, 312(5779):1501–1504, 2006.

[104] Robert D Howe and Mark R Cutkosky. Sensing skin acceleration for slip and texture perception. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 145–150, Scottsdale, AZ, 14-19 May 1989.

[105] Marc R Tremblay and Mark R Cutkosky. Estimating friction using incipient slip sensing during a manipulation task. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 429–434, Atlanta, GA, USA, 02-06 May 1993.

[106] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

[107] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[108] Sameer Pai, Tao Chen, Megha Tippur, Edward Adelson, Abhishek Gupta, and Pulkit Agrawal. Tactofind: A tactile only system for object retrieval. *arXiv preprint arXiv:2303.13482*, 2023.

[109] Wenyu Liang, Fen Fang, Cihan Acar, Wei Qi Toh, Ying Sun, Qianli Xu, and Yan Wu. Visuo-tactile feedback-based robot manipulation for object packing. *IEEE Robotics and Automation Letters*, 8(2):1151–1158, 2023.

[110] Yoren Gaffary, Ferran Argelaguet, Maud Marchal, Adrien Girard, Florian Gosselin, Mathieu Emily, and Anatole Lécuyer. Toward haptic communication: Tactile alphabets based on fingertip skin stretch. *IEEE Transactions on Haptics*, 11(4):636–645, 2018. doi: 10.1109/TOH.2018.2855175.

[111] Alex Church, John Lloyd, Raia Hadsell, and Nathan F Lepora. Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard. *IEEE Robotics and Automation Letters*, 5(4):6145–6152, 2020.

[112] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[113] Ian H Taylor, Siyuan Dong, and Alberto Rodriguez. Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787, Philadelphia, USA, 23-27 May 2022.

[114] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.

[115] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors and Actuators A: physical*, 167(2):171–187, 2011.

[116] Maria Bauza, Oleguer Canal, and Alberto Rodriguez. Tactile mapping and localization from high-resolution tactile imprints. *2019 International Conference on Robotics and Automation (ICRA)*, pages 3811–3817, 20-24 May 2019. doi: 10.1109/ICRA.2019.8794298.

[117] Maria Bauza Villalonga, Alberto Rodriguez, Bryan Lim, Eric Valls, and Theo Sechopoulos. Tactile object pose estimation from the first touch with geometric contact rendering. In *Conference on Robot Learning*, pages 1015–1029, London, UK, 08=11 Nov 2021.

[118] Maria Bauza, Antonia Bronars, and Alberto Rodriguez. Tac2pose: Tactile object pose estimation from the first touch. *arXiv preprint arXiv:2204.11701*, 2022.

[119] Justin Lin, Roberto Calandra, and Sergey Levine. Learning to identify object instances by touch: Tactile recognition via multimodal matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3644–3650, Montreal, Canada, 20-24 May 2019.

[120] Prajval Kumar Murali, Michael Gentner, and Mohsen Kaboli. Active visuo-tactile point cloud registration for accurate pose estimation of objects in an unknown workspace. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2838–2844, Prague, Czech Republic, 27 Sep - 01 Oct 2021.

[121] Arkadeep Narayan Chaudhury, Timothy Man, Wenzhen Yuan, and Christopher G Atkeson. Using collocated vision and tactile sensors for visual servoing and localization. *IEEE Robotics and Automation Letters*, 7(2):3427–3434, 2022.

[122] Cristiana De Farias, Naresh Marturi, Rustam Stolkin, and Yasemin Bekiroglu. Simultaneous tactile exploration and grasp refinement for unknown objects. *IEEE Robotics and Automation Letters*, 6(2):3349–3356, 2021.

[123] Takamitsu Matsubara and Kotaro Shibata. Active tactile exploration with uncertainty and travel cost for fast shape estimation of unknown objects. *Robotics and Autonomous Systems*, 91:314–326, 2017.

[124] Justin Gould, Simon Clement, Bradley Crouch, and Roberto SP King. Evaluation of photometric stereo and elastomeric sensor imaging for the non-destructive 3d analysis of questioned documents–a pilot study. *Science & Justice*, 63(4):456–467, 2023.

[125] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[126] Qiang Li, Oliver Kroemer, Zhe Su, Filipe Fernandes Veiga, Mohsen Kaboli, and Helge Joachim Ritter. A review of tactile information: Perception and action through touch. *IEEE Transactions on Robotics*, 36(6):1619–1634, 2020.

[127] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 21-26 Jul 2017.

[128] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 27-30 Jun 2016.

[129] Matterport Inc. Mask R-CNN for object detection and segmentation, 2018. URL `https://github.com/matterport/Mask_RCNN`.

[130] A. Dutta and A. Zisserman. The VIA annotation software for images. audio and video. *27th ACM International Conference on Multimedia*, 21-25 Oct 2019. URL `https://doi.org/10.1145/3343031.3350535`.

[131] R. Diankov. Automated construction of robotic manipulation programs. *Ph.D. dissertation, Carnegie Mellon University*, 2010.

[132] D. Hsu, J.C. Latombe, and R. Motwani. Automated path planning in expansive configuration spaces. *International Conference on Robotics and Automation (ICRA)*, pages 2719–2726, 20-25 Apr 1997.

[133] C. Yuksel. Sample elimination for generating poisson disk sample sets. *EUROGRAPHICS*, 35, 04-08 May 2015.

[134] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *Eurographics Symposium on Geometry Processing*, 26-28 Jun 2006.

[135] T. Möller and B. Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2:21–28, 1997.

[136] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016–2019. URL `http://pybullet.org`.

[137] COCO Common Objects in Context. COCO dataset, 2020. URL `https://cocodataset.org/`.

[138] NikodemBartnik. Water pump, 2017. URL `https://www.thingiverse.com/thing:2467758`.

[139] Rui Li and Hong Qiao. A survey of methods and strategies for high-precision robotic grasping and assembly tasks—some new trends. *IEEE/ASME Transactions on Mechatronics*, 24(6):2718–2732, 2019. doi: 10.1109/TMECH.2019.2945135.

[140] Dawson-Haggerty et al. *trimesh*, 2019. URL `https://trimsh.org/`.

[141] Ken Goldberg, Brian V Mirtich, Yan Zhuang, John Craig, Brian R Carlisle, and John Canny. Part pose statistics: Estimators and experiments. *IEEE Transactions on Robotics and Automation*, 15(5):849–857, 1999.

[142] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0588–0592, Melmaruvathur, Tamilnadu, India, 06-08 Apr 2017.

[143] Elon D. Rimon and Andrew Blake. Caging 2d bodies by 1-parameter two-fingered gripping systems. *Proceedings of IEEE International Conference on Robotics and Automation*, 2:1458–1464, 22-28 Apr 1996.

[144] Elon Rimon and Andrew Blake. Caging planar bodies by one-parameter two-fingered gripping systems. *The International Journal of Robotics Research*, 18(3):299–318, 1999. doi: 10.1177/02783649922066222.

[145] P. Pipattanasomporn and Attawith Sudsang. Two-finger caging of concave polygon. *IEEE International Conference on Robotics and Automation*, 15-19 May 2006:2137 – 2142, 06 2006. doi: 10.1109/ROBOT.2006.1642020.

[146] Manuel Caroli, Pedro MM de Castro, Sébastien Loriot, Olivier Rouiller, Monique Teillaud, and Camille Wormser. Robust and efficient delaunay triangulations of points on or close to a sphere. In *International Symposium on Experimental Algorithms*, pages 462–473, Ischia Island, Naples, Italy, 20-22 May 2010.

[147] Megan Schuster. The largest empty circle problem. In *Proceedings of the Class of 2008 Senior Conference*, pages 28–37, Computer Science Department, Swarthmore College, 2008.

[148] Songrit Maneewongvatana and David M. Mount. Analysis of approximate nearest neighbor searching with clustered point sets. *CoRR*, cs.CG/9901013, 1999. URL `https://arxiv.org/abs/cs/9901013`.

[149] Jordan Miller. *High Resolution Stanford Bunny*. Rice University, 2011. URL `https://www.thingiverse.com/thing:11622`.

[150] Ruchi Goel and Pooja Gupta. Robotics and industry 4.0. *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*, pages 157–169, 2020.

[151] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[152] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, Long Beach, CA, USA, 15-20 Jun 2019.

[153] Andreas Gartus and Helmut Leder. The small step toward asymmetry: Aesthetic judgment of broken symmetries. *i-Perception*, 4(5):361–364, 2013.

[154] Andreas Gartus, Mark Völker, and Helmut Leder. What experts appreciate in patterns: Art expertise modulates preference for asymmetric and face-like patterns. *Symmetry*, 12(5):707, 2020.

[155] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[156] Pierre Legendre and Louis Legendre. Chapter 13 - spatial analysis. In Pierre Legendre and Louis Legendre, editors, *Numerical Ecology*, volume 24 of *Developments in Environmental Modelling*, pages 785–858. Elsevier, 2012. doi: https://doi.org/10.1016/B978-0-444-53868-0.50013-7. URL `https://www.sciencedirect.com/science/article/pii/B9780444538680500137`.

[157] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[158] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[159] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[160] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352, Columbus, OH, USA, 23-28 Jun 2014.

[161] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. Sparse representation for 3d shape estimation: A convex relaxation approach. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1648–1661, 2016.

[162] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, Istanbul, Turkey, 27-31 Jul 2015. doi: 10.1109/ICAR.2015.7251504.

[163] Van-Thach Do and Quang-Cuong Pham. Geometry-aware coverage path planning for depowdering on complex 3d surfaces. *IEEE Robotics and Automation Letters*, 8(9):5552–5559, 2023.

List of Author's Awards, Patents, and Publications

Awards

- **Joyce Lim Xin Yan**, "Poster Awards May 2020: Honorable Mentions", *HP-NTU Digital Manufacturing Lab.*
- **Joyce Lim Xin Yan**, "SWE2023 Patent Recognition Award", *Society of Women Engineers.*

Patents

- Huy Nguyen, Nicholas Adrian, **Joyce Lim Xin Yan**, Pham Quang Cuong, Jonathan M. Salfity and Will Allen, "3D printed object cleaning", *U.S. Patent 17/904,331*, filed on 09 Apr 2020 in U.S., issued on 16 Aug 2022.
- **Joyce Lim Xin Yan** and Pham Quang Cuong, "Simulated powdered model generation for neural networks", *U.S. Patent 18/279,940*, *European Patent 21929379.2*, *Chinese Patent 202180095107.7* filed on 03 Mar 2021 in PCT, issued on 01 Sep 2023.
- **Joyce Lim Xin Yan** and Pham Quang Cuong, "Robot gripper geometries", *U.S. Patent Application*, filed on 12 Apr 2022 in PCT, pending issue.

Defensive Publications

- Nicholas Adrian, Pham Quang Cuong, Do Van Thach, **Joyce Lim Xin Yan**, "Precise robot manipulation system using deep feature-based visual servoing for automation in unstructured environments", published on 26 Jan 2022.
- **Joyce Lim Xin Yan** and Pham Quang Cuong, "A method that augments unique patterns on 3DP parts to aid object recognition and pose refinement for vision-based tactile sensors in robotic tasks", published on 20 Sep 2023.

Journal Articles

- **Joyce Xin-Yan Lim** and Quang-Cuong Pham, "Automated post-processing of 3D-printed parts: Artificial powdering for deep classification and localization", in *Virtual and Physical Prototyping*, vol. 16, pp. 333–346, 2021.

- **Joyce Xin-Yan Lim** and Quang-Cuong Pham, "Automatic Fingerpad Customization for Precise and Stable Grasping of 3D-Print Parts", submitted to *IEEE Transactions on Automation Science and Engineering (T-ASE)* , 2023.

- **Joyce Xin-Yan Lim** and Quang-Cuong Pham, "Grasping, Part Identification and Pose Refinement in One Shot with a Tactile Gripper", submitted to *IEEE Robotics and Automation Letters*, 2023.

Conference Proceedings

- Huy Nguyen, Nicholas Adrian, **Joyce Lim Xin Yan**, Jonathan M. Saltify, William Allen and Pham Quang Cuong "Development of a Robotic System for Automated Decaking of 3D-Printed Parts," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 31 May to 31 Aug 2020.