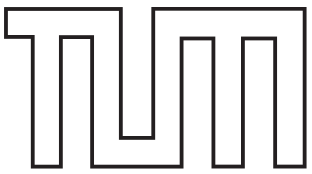


Technische Universität München
Chair of Media Technology
Prof. Dr.-Ing. Eckehard Steinbach

Master Thesis

Analytical 6-DOF Robotic Grasp Estimation Based on Surface Normals

Author:	Abdullah Cem Özbay
Matriculation Number:	03681672
Address:	Breisacherstraße 5 81667 München
Advisor:	Hasan Furkan Kaynar
Begin:	04.01.2023
End:	04.07.2023

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, July 4, 2023	
Place, Date	Signature

München, July 4, 2023	
Place, Date	Signature

Kurzfassung

Diese Masterarbeit befasst sich mit der anspruchsvollen Aufgabe, verschiedene Objekte zu erfassen, die als Punktwolken beobachtet werden, ohne dass Objektmodelle, Trainingsdaten oder Vorwissen erforderlich sind. Um dieses Problem zu lösen, wird eine neue Greifmetrik eingeführt. Diese Metrik ermöglicht die Schätzung möglicher Greifhaltungen mit 6 Freiheitsgraden (DOF), indem sie sowohl die Greiffinger als auch die Oberflächennormalen des Objekts zur Bestimmung der lokalen geometrischen Ähnlichkeit verwendet. Einer der Hauptvorteile dieser Methode ist ihre Fähigkeit, die Einschränkungen zu überwinden, die sowohl bei analytischen Griffschätzern als auch bei datengesteuerten Ansätzen bestehen. Im Gegensatz zu neueren lernbasierten Ansätzen macht die vorgeschlagene Methode umfangreiche Trainingsdaten überflüssig. Sie unterscheidet sich auch von den Ansätzen zum Kraftschluss, da keine physikalischen Parameter benötigt werden. Die Effektivität der vorgeschlagenen Methode wird durch das Finden praktikabler Greifposen auf Objektpunktwolken demonstriert, die von einer RGB-Tiefenkamera erfasst wurden. Diese Posen werden dann in realen Roboterexperimenten ausgeführt, bei denen verschiedene Objekte gegriffen werden. Diese Experimente haben gezeigt, dass die Methode alle Objekte erfolgreich greifen kann, wenn sie mit den rekonstruierten Punktwolken der Objekte präsentiert wird. Die Ergebnisse dieser Dissertation zeigen das Potenzial des vorgestellten Ansatzes für ein robustes und effizientes Greifen von verschiedenen Objekten in realen Szenarien. Die vorgeschlagene Methode eröffnet Möglichkeiten für flexiblere und adaptive robotische Manipulationsaufgaben, indem sie die Abhängigkeit von umfangreichem Vorwissen oder Trainingsdaten eliminiert.

Abstract

This thesis addresses the challenging task of grasping various objects observed as point clouds without the need for object models, training data, or any prior knowledge. A new grasp metric is introduced to address this problem. This metric enables the estimation of feasible 6 degrees of freedom (DOF) grasp poses by utilizing both the gripper fingers and the surface normals of the object to determine local geometry similarity. One of the key advantages of this method is its ability to overcome the limitations found in both analytical grasp estimators and data-driven approaches. Unlike recent learning-based approaches, the proposed method eliminates the need for extensive training data. It also differs from force closure approaches by eliminating the need for physical parameters. The effectiveness of the proposed method is demonstrated by finding feasible grasp poses on object point clouds captured by an RGB depth camera. These poses are then executed by real robot experiments involving grasping various objects. These experiments proved that the method could successfully grasp all objects when presented with the reconstructed point clouds of objects. The results of this thesis research highlight the potential of the introduced approach to enable robust and efficient grasping of various objects in real-world scenarios. The proposed method opens up possibilities for more flexible and adaptive robotic manipulation tasks by eliminating the dependency on extensive prior knowledge or training data.

Contents

Chapter 1

Introduction

Robots have become a part of our daily routines, and their role in our daily lives is increasing. Thanks to rapid robotic research in recent years, the capabilities and application areas of robots have expanded significantly. From household robots to industrial robots, the use of robots in our daily lives makes our lives easier at home and work. We have seen kitchen robots helping to cook or floor robots sweeping the whole house. Beyond these, using robots goes much further. It means significant progress in essential fields such as medicine, education, physics, or chemistry. Robots have routinely and consistently grasped various things in factory settings for decades. This achievement is based on straightforward programmed actions on predefined objects in contexts with much structure. Research is still needed to solve the problem of grasping arbitrarily shaped objects observed as partial point clouds in unstructured situations without requiring models of objects, physics parameters, or training data.

Alongside these advances, robots are currently used in a limited way due to their lack of reliability. Some tasks that are simple for us humans can be pretty complex for robots. Many objects that are just familiar, simple objects to us can be perceived as complex structures to a robot because, in fact, objects vary significantly in their physical properties and functions. This makes perception and successful grasping even more complicated, so fundamental tasks such as grasping and manipulating objects are still challenging problems in robotics.

Determining the ideal 6 degrees of freedom (DOF) pose for a robot's gripper to grasp an object efficiently is crucial in robotic manipulation. While data-driven techniques such as Convolutional Neural Networks (CNNs) have shown exceptional effectiveness in grasp estimation [KK17, RA15], there is significant interest in developing analytical techniques based on geometry [JMS11, TPP18, FV12] and surface normals. In this paper, it is explored the idea of 6-DOF robotic grasp pose estimation using analytical methods based on surface normals and identifies the advantages of such an approach over data-driven methods.

Traditional physics-based grasping techniques, such as the GraspIt! framework [MA04], rely on the robot having precise knowledge about the shape of the grasped object. However, the research discussed here presents a revolutionary strategy that eliminates the need to learn or analyze training data using classical physics. Instead of relying on explicit knowledge of the physical properties of the object, the proposed method utilizes analytical techniques that do not require a deep understanding of the object. Using geometry and surface normals, the robot can determine the ideal grasping posture without troublesome physics calculations or significant training. This method takes advantage of geometric features built into objects to infer grasp stability and contact forces. The algorithm can determine the optimal grasp pose configuration by analyzing surface normals at various points on the object's surface. Unlike traditional methods that rely on specific object parameters, this technique offers a more generalizable solution that can adapt to different objects without prior knowledge [KRC+11, BLE08].

Creating mathematical models and algorithms that exploit the geometric properties of an object is an essential component of analytical methods for grasp pose estimation. These techniques estimate ideal grasp pose using explicit equations and assumptions rather than learning from evidence. A frequently used strategy is creating guidelines or heuristics based on geometric properties such as curvature, symmetry, or contact points [BLAL16]. By examining surface normals at various positions on the surface of the object, the method can detect surfaces that resemble the surface of the gripper's fingers. It can thus determine the ideal grasp pose that maximizes stability and contact forces. The ability of analytical methods to generalize to new objects or object variants is excellent. Because they focus on the geometric features captured by the surface normals, they can make grasp estimations for objects of various sizes, forms, or textures without further training. This flexibility is highly beneficial in scenarios where the robot encounters new objects in real-time, enabling faster adaptation and improved performance. By examining the direction and orientation of the surface normals, the shape and structure of the object can be learned, which is essential for grasp planning. The advantage of surface normals is that they can capture object features without needing much training data. Analytical techniques based on surface normals can generalize well to various objects without the need for significant training, unlike CNNs that depend on large datasets for training. Compared to data-driven approaches that require complex neural network topologies, analytic methods based on surface normals often involve simple computations. As a result, real-time 6-DOF grasp poses estimation is possible using analytical techniques, allowing robots to make fast decisions and react to dynamic environments more effectively. They rely on mathematical models and geometric attributes that can be obtained from a small sample size or even from pre-built object models. As a result, the process of grasp estimation is made more effective without requiring significant amounts of data collection and explanation.

While analytical methods based on surface normals offer promising advantages, they have some challenges to overcome. One of the limitations is the noise and imperfections in erroneous point clouds captured by RGB depth cameras. Today, it is known that many RGB depth cameras do not obtain a completely accurate point cloud, which can lead

to inaccuracies in the surface normal estimation process [MN03]. Robust and accurate methods for surface normal calculation are crucial to ensure reliable grasp estimation. Further research is also needed to develop analytical models that can handle complex objects with irregular or deformable shapes.

6-DOF robotic grasp estimation based on surface normals and analytical techniques exploits the geometric properties of objects, enabling better generalization and computational efficiency with less reliance on data. Despite some obstacles to overcome, surface normals and analytical methods hold significant promise for improving robotic manipulation capabilities by enabling robots to grasp objects in various situations with higher accuracy and adaptability. Consequently, the research presented here offers a new path for robotic grasping that does not rely on data-driven methodologies or classical physics studies. The proposed method offers a versatile and adaptive approach to determine the best 6-DOF grasping pose configurations based on surface normals. The goal is to enable robots to manipulate objects more effectively without needing in-depth training.

The thesis will proceed as follows: Chapter 2 presents a background and related work on data-driven and analytical methods for robotic grasping. Chapter 3, Methodology, details the candidate to pose sampling algorithm for the ideal grasp pose search and the design of the grasp score metric assigned to the found poses, followed by a comprehensive experimental evaluation in Chapter 4. Finally, Chapter 5 summarizes the thesis with a conclusion, discusses the implications of the findings, and provides valuable insights for future work directions in robotic grasp estimation.

Chapter 2

Background and Related Work

The following section gives background information about the topic robotic grasp estimation.

2.1 Robotic Grasp Pose Estimation Overview

In order for robots' grippers to be able to grasp an object safely and stabily, the optimal grasp poses need to be estimated. This is done by first analyzing the object's shape, size and other characteristics and then calculating the best grasp configuration for the robot gripper.

Here is the general pipeline of robotic grasp pose estimation:

1. **Object Representation:** The objects must be represented in a suitable format for the robotic grasp estimation process. These object representations are usually captured as 3D point cloud data using RGB depth cameras.
2. **Feature Analysis:** In order to analyze and observe the shape and geometry of an object, various features are extracted from the object's point cloud. These features are surface normals, curvature, edges, and symmetry.
3. **Generating Grasp Poses:** Once the object's features are extracted, candidate grasp poses are generated in various configurations. These candidate poses represent the different possible ways in which the robot can grasp the object.
4. **Grasp Score:** Each candidate grasp pose is evaluated using metrics that measure its stability and effectiveness. Factors such as force closure[MA04, LUD+10], contact points and stability under external forces are considered when evaluating grasp quality.
5. **Grasp Selection:** The methods select the best grasp pose from the generated candidates based on the metrics used to assess grasp quality. The selected grasp is the

one that maximizes the chances of a successful grasp.

6. **Execution of the Robot:** Once a grasp is selected, the grasp configurations are given to the robot, which performs the grasp using its gripper. Feedback from the robot's motion planning is used to make the necessary adjustments to ensure the object is grasped safely.

7. **Learning and Improvement:** Learning algorithms can enhance robotic grasp estimation techniques. Data-driven approaches, such as deep learning, involve training models on extensive datasets of successful grasps, which improves the system's grasp estimation capabilities over time.

Robotic grasp estimation presents challenges due to the wide variety of object shapes, sizes, and materials encountered. Researchers and engineers are continuously developing new algorithms and techniques to enhance the accuracy and reliability of robotic grasp estimation systems.

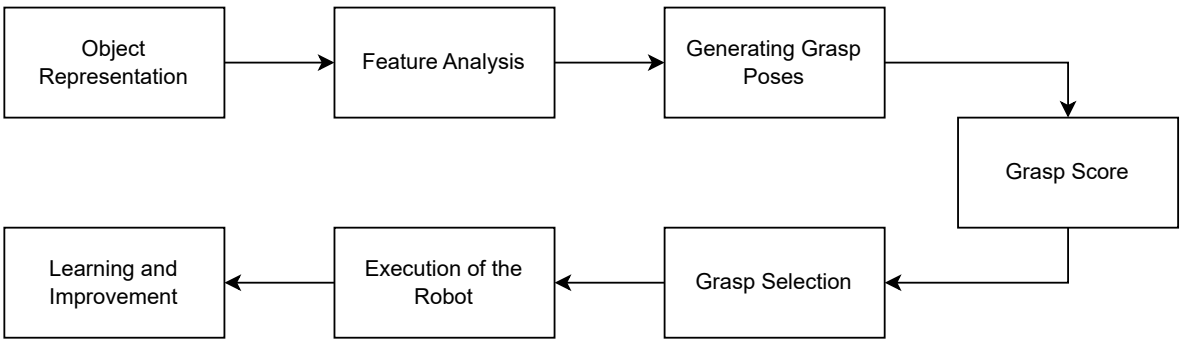


Figure 2.1: General pipeline of the robotic grasp estimation.

2.2 Prior Knowledge in Grasp Pose Estimation

The exploration of effective grasping techniques has seen significant progress, mainly through the utilization of 3D simulations [BK00, PMAJ04]. These simulation-based approaches harness the power of virtual environments but require access to a comprehensive 3D model and additional physical information to determine appropriate grasps. In work [SHKK10], a Bayesian network has been employed to establish connections between objects, actions, target attributes, and tasks. This probabilistic framework amalgamates prior knowledge with observed data, enabling inference about target attributes even in the presence of partial observations.

However, relying on prior knowledge about object shapes and predefined affordance or semantic constraints limits the applicability of these techniques in realistic environments characterized by perception and motion uncertainties. In situations where complete object models are not available in advance, general-purpose robots may need to grasp unfamiliar objects without pre-rendering complex 3D models. In such cases, grasp detection techniques based solely on RGB depth data [PMAJ04, KCF11, SHKK10] have been developed, and machine learning has been used [KK17, WYPS22, SDT+22] to distinguish signs of successful grasps from the available data. Remarkably, grasp models exhibit strong generalization capabilities for novel objects, eliminating the need for a complete physical model [SWN08]. Only a single image of an object's face is typically sufficient.

In modern robotics, integrating RGB depth sensors and data has become ordinary and highly beneficial for a wide range of tasks. These tasks include object recognition[UVDSGS13], detection, mapping[HKH+14], and various other interactions with 3D environments. So a notable advantage of RGB depth sensors is the additional depth information they provide alongside the traditional RGB color data. This depth of information allows robotic systems to perceive and understand the three-dimensional aspects of the environment more accurately.

Generally, grasp pose estimation can be divided into analytical methods [FV12, DPM17, JMS11], and data-driven methods [KK17, KCF11, SHKK10, MLM+21, FZG+20, KKB20, WYPS22].

However, it is essential to acknowledge that both analytical and data-driven methods have their own limitations. Analytical approaches, which generate grasp poses based on criteria like force closure [MA04, LUD+10], rely heavily on precise geometric information and complete 3D models of objects. In real-world scenarios, obtaining such detailed models may not always be feasible, thus restricting the applicability of these methods.

On the other hand, data-driven methods leverage experiences collected during grasp execution to estimate grasp poses. While these approaches do not rely on explicit geometric models, they come with their own challenges. The process of collecting extensive grasp pose annotations is extremely time-consuming and labor-intensive. Additionally, there is a concern regarding the generalizability of the acquired experience to new and unseen objects.

2.3 Data-Driven Methods for Grasp Pose Estimation

Robotic grasp estimation plays a vital role in enabling robots to manipulate objects effectively and independently. The main objective of grasp estimation is to determine the best possible grasp configuration for a robot's hand or gripper, ensuring a secure and stable grasp on the object. There are two approaches to grasp estimation: data-driven and analytical techniques. When applying data-driven approaches in robotic grasp estimation, deep learning techniques have significantly transformed the field. This approach involves sev-

eral steps. First, extensive datasets, e.g., ImageNet 2012, [HZRS15, MPH+16, MLN+17] comprising numerous successful grasps are collected. These datasets contain thousands of object instances and their corresponding grasp configurations, which can be obtained through robot simulators or real-world grasp scenarios.

Data-driven methods are then employed to analyze the collected data and extract relevant features from the objects. These features, including shape, size, surface normals, edges, and curvature, provide critical information for grasp estimation. Deep learning algorithms, such as convolutional neural networks [KK17, RA15] or recurrent neural networks (RNNs), are trained using the collected datasets [HZRS15]. The objective is to learn the relationship between the object features and the optimal grasp configurations. The models are trained to estimate grasp quality or directly output proper grasp poses. Using the trained models, multiple grasp hypotheses are generated based on the learned grasp configurations. These hypotheses represent different potential grasps for a given object. The model evaluates each grasp hypothesis using grasp quality metrics learned during training. The grasp with the highest estimated quality is selected as the optimal grasp for the robot to execute. Once the grasp is selected, the robot executes it using its hand or gripper. Feedback from tactile or force sensors assesses the grasp's stability and effectiveness. This feedback can be used to refine the grasp or adjust the hand or gripper's position if necessary. The models can continuously improve grasp estimation through iterative learning. Real-world grasp data and feedback are utilized to update and refine the models, enhancing their accuracy and adaptability over time. This iterative learning process enables the data-driven models to continually improve their grasp estimation capabilities, leading to more successful and efficient grasp by robots. The integration has revolutionized robotic grasp estimation. It allows robots to analyze and learn from large-scale datasets, extract relevant object features, generate grasp hypotheses, select optimal grasps, and continuously improve their grasp estimation capabilities through feedback and iterative learning.

A significant challenge with data-driven methods is their need for a large volume of training data. However, large datasets with manually labeled images are unavailable for most robotics applications. In computer vision, transfer learning techniques are used to pre-train deep convolutional neural networks on some large datasets, e.g., ImageNet, which contains 1.2 million images with 1000 categories [HZRS15], before the network is trained on the target dataset. These pre-trained models are either used as an initialization or as a fixed feature extractor for the task of interest.

Three different models developed for robotic grasp estimation are examined in the following.

2.3.1 Neural Descriptor Fields

The work presented in [SDT+22] introduces a new approach called Neural Descriptor Fields (NDFs). NDFs serve as object representations that encode both points and relative poses between an object and a robot gripper using category-level descriptors.

The primary focus of this research is object manipulation and targets explicitly the ability to perform the same task on a new instance of an object of the same category. To achieve this goal, an optimized search process is proposed to find the pose whose descriptor matches the observed representation. A notable aspect of NDFs is the training methodology, which is performed in a self-supervised manner. This approach eliminates needing expert-labeled key points using a 3D auto-coding task. By training NDFs in this way, NDFs can obtain meaningful representations without relying on human annotations. Furthermore, NDFs exhibit SE(3)-equivalence, ensuring their performance generalizes well across various 3D object transitions and rotations. This property increases the robustness and adaptability of the NDF-based system in handling different object configurations and orientations.

The introduction of Neural Descriptor Fields (NDFs) as an object representation offers a promising way for object manipulation tasks, enabling the repetition of tasks on new examples of the same category. The self-monitored training approach and the SE(3)-equivariant nature of NDFs contribute to their effectiveness and generalization ability.

2.3.2 PointNet

PointNet [QSMG17], the deep learning architecture presented in the paper, revolutionizes the field of 3D data analysis by enabling the direct processing of raw point clouds. Unlike traditional approaches that rely on intermediate representations like voxel grids or meshes, PointNet operates on unordered point sets, capturing intricate geometric details and scaling effectively to large-scale scenes. By leveraging shared multi-layer perceptrons and symmetric functions, PointNet effectively learns permutation-invariant functions, making it invariant to the ordering of points. This fundamental property allows PointNet to extract meaningful features from unstructured point cloud data, leading to superior performance in object classification and semantic segmentation tasks. The paper's comprehensive evaluations demonstrate PointNet's robustness, generalizability, and potential for real-world applications, solidifying its position as a groundbreaking method in 3D classification and segmentation.

2.3.3 Convolutional Neural Networks

Convolutional neural networks are a powerful model for learning feature extractors and visual models [KK17, RA15, DWLZ21, MCL18]. The paper [KK17] aims to estimate the optimal grasp pose for a parallel-plate robotic gripper when it encounters new objects using an RGB depth image of the scene as input. The proposed model utilizes a deep convolutional neural network to extract meaningful features from the given scene. These features capture relevant information about objects and their spatial arrangements. A separate convolutional neural network is then used to estimate the grasp configuration of the object based on the extracted features. Convolutional Neural Networks (CNNs) have emerged as a powerful tool for robotic grasp estimation and have revolutionized the field of robotic manipulation. These neural networks have demonstrated remarkable success in

capturing intricate patterns and extracting meaningful features from visual data, enabling accurate and efficient grasp estimation for robotic systems.

One of the critical advantages of CNNs is their ability to learn hierarchical representations from raw input data automatically. In the context of robotic grasp estimation, this means that CNNs can analyze images or depth maps of objects and learn to identify relevant visual cues indicative of graspable regions or optimal grasp poses. By leveraging large-scale datasets containing diverse object instances and their corresponding grasp annotations, CNNs can be trained to generalize across different objects, shapes, and orientations. The architecture of CNNs plays a crucial role in their grasp estimation performance. Typically, CNNs consist of multiple layers of convolutional and pooling operations, which allow them to capture local features and their spatial relationships effectively. These layers are followed by fully connected layers that enable higher-level feature representation and grasp estimation. Moreover, the use of techniques such as dropout regularization and batch normalization aids in preventing overfitting and improving generalization.

To train CNNs for robotic grasp estimation, a substantial amount of labeled data is required. This data is often collected through real-world robotic experiments or simulated environments. Researchers employ techniques such as data augmentation, where training samples are artificially augmented with various transformations, to enhance the diversity and richness of the dataset. This augmentation helps CNNs generalize and perform reliably in different real-world scenarios. Once trained, CNNs can estimate grasp poses for novel objects by analyzing their visual representations. This enables robotic systems to autonomously plan and execute grasping actions with improved accuracy and efficiency. Combining CNN-based grasp estimation algorithms with robotic platforms enhances their capability to handle complex objects, adapt to different shapes, and cope with various environmental conditions.

Despite their effectiveness, CNNs for robotic grasp estimation face some challenges. One of the main challenges is the need for large annotated datasets, which can be time-consuming and expensive to acquire. Additionally, the generalization of CNNs to novel objects or unseen scenarios remains an active area of research, as it is crucial for practical robotic applications.

In conclusion, Convolutional Neural Networks have revolutionized robotic grasp estimation by leveraging their ability to learn from visual data. Their hierarchical representation learning and large-scale training datasets enable accurate and efficient grasp estimation for diverse objects. By integrating CNN-based grasp estimation algorithms into robotic systems, we can enhance their capabilities and enable them to interact with the physical world more effectively and autonomously. As research in this field progresses, we can expect further advancements in CNN-based grasp estimation techniques, paving the way for increasingly sophisticated and capable robotic manipulation systems. Although CNNs have brought significant advances to robotic grasp estimation, several drawbacks exist. The demand for extensive labeled training data, difficulties in generalizing to novel objects and scenarios, limitations in capturing non-visual factors, and computational requirements

are barriers to the widespread adoption and practical application of CNN-based grasp estimation methods. Addressing these limitations through research and development efforts will be crucial to realize the full potential of CNNs in robotic manipulation tasks and enabling their successful integration into real-world robotic systems.

Dex-Net 2.0

In [MLN+17], a research is presented on training using a synthetic dataset, with the aim to shorten the duration of the deep learning process to develop robust robotic grasp poses. This dataset consists of a large collection of 6.7 million point clouds, grasping metrics generated from thousands of 3D models obtained from Dex-Net 1.0 [MPH+16]. These models were placed in random poses on a table to allow for a variety of training scenarios.

The main goal of this approach is to overcome the need to collect extensive amounts of data, especially considering the large number of possible objects. However, the relationship between point clouds, grasps and metrics is non-linear, which poses a challenge for traditional linear or kernel models. In response, a new approach called the Grasp Quality Convolutional Neural Network (GQ-CNN) model has been developed. This model is specifically designed to classify robust grasp poses in depth images. To train the GQ-CNN, we use a large amount of data from Dex-Net 2.0 to enable the model to learn and understand the complex relationships between input depth images, grasp poses and grasp quality metrics.

This approach demonstrates the potential of using synthetic data and convolutional neural networks to improve the efficiency and effectiveness of grasp planning in real-world scenarios.

2.4 The Challenge of Data-Driven-Methods for Grasp Pose Estimation

Finding suitable candidate grasp poses for robotic grasp estimation has always posed a significant challenge [BMAK13]. Various deep-learning models have already been designed to find these candidate poses. These models have been trained on datasets and have achieved remarkable success in identifying optimal candidate poses for task-oriented grasping. However, the most crucial challenge here is that model training in deep learning is extremely costly.

The training process in deep learning is demanding and requires significant resources and considerable computational power. It involves countless iterations as models learn from much data and adjust their parameters accordingly. The magnitude of this computational burden acts as a significant deterrent to wider adoption of these models. To alleviate the onerous computational cost of training deep learning models from scratch, Transfer learning techniques are gaining traction. Transfer learning enables models to leverage

knowledge from pre-existing networks or related tasks, quickly adapting to new tasks with fewer training iterations. This approach not only alleviates the computational burden, but also facilitates the transfer of acquired knowledge and features, leading to improved performance on specific grasp estimation tasks. While deep learning models have proven effective in identifying suitable candidate grasper poses for task-oriented grasping, the prohibitive cost of training these models remains a significant barrier. However, while promising to overcome this challenge, ongoing research efforts involving transfer learning and optimized data collection methodologies are far from sufficient.

A major challenge with deep learning is that it needs a large volume of training data. However, large datasets with manually labeled images are unavailable for most robotics applications. In computer vision, transfer learning techniques are used to pre-train deep convolutional neural networks on some large datasets, e.g., ImageNet, which contains 1.2 million images with 1000 categories [19], before the network is trained on the target dataset [20]. These pre-trained models are either used as an initialization or as a fixed feature extractor for the task of interest.

Convolutional Neural Networks (CNNs) have offered significant advantages in accuracy in robotic grasp estimation. These neural networks are good at learning hierarchical representations from visual data, enabling them to extract meaningful features for robotic systems and estimate optimal grasp positions. However, despite their benefits, there are also some disadvantages associated with the use of CNNs for robotic grasp estimation.

One of the main disadvantages is the need for a large amount of labeled training data. CNNs thrive on large datasets to learn complex relationships between visual inputs and grasp positions effectively. Obtaining such datasets can be time-consuming, costly, and labor-intensive. The process often involves extensive manual annotation of grasp poses, which may not always be feasible or readily available for a wide variety of objects and scenarios. This limitation hinders the scalability and applicability of CNN-based grasp estimation methods in real-world environments.

Another challenge lies in the generalization capability of CNNs. While CNNs can learn to make grasp estimations for objects in the training set, their performance may degrade when presented with new objects or scenarios not encountered during training. This lack of generalization is particularly problematic in dynamic environments where the appearance and geometry of objects can vary significantly. Adapting CNNs to deal with such variations and generalize effectively across different objects and environments remains an ongoing research challenge. Furthermore, CNN-based grasp estimation methods rely heavily on visual inputs such as RGB or depth images. While visual information provides valuable cues for grasp estimation, it may not capture all relevant factors that influence successful grasping. Essential aspects such as object weight, material properties, or haptic feedback are not directly included in the CNN-based grasp estimation process. Neglecting these aspects can limit the robustness and reliability of grasp planning and execution, especially in scenarios where precise force control or delicate manipulation is required. Another limitation is the computational complexity associated with training and deploying CNNs. CNNs

typically require significant computational resources both during the training phase and during inference. Training CNNs on large-scale datasets with complex architectures can be computationally intensive and time-consuming. Deploying CNN-based grasp estimation models on resource-constrained robotic platforms can lead to challenges due to memory and processing constraints.

In conclusion, while data-driven methods have brought significant advances to robotic grasp estimation, there are several drawbacks to consider. The demand for extensive labeled training data, difficulties in generalizing to novel objects and scenarios, limitations in capturing non-visual factors, and computational requirements are all barriers to the widespread adoption and practical application of CNN-based grasp estimation methods. Addressing these limitations through research and development efforts will be crucial to realize the full potential of CNNs in robotic manipulation tasks and ensuring their successful integration into real-world robotic systems.

2.5 Analytical Methods for Grasp Pose Estimation

Before the proliferation of machine learning methodologies, robotic grasp estimation relied on traditional techniques that involved studying object geometries and adhering to predetermined methods. Traditional methods approach the grasping problem from a purely geometric perspective. They optimize grasp quality metrics based on analytical models of constraints derived from geometry and physics.

Here is how the concept estimation algorithms work, drawing on object geometry and mathematics.

Initially, objects were examined, looking at their shapes using techniques like point clouds or object models [TPGSP17]. Important geometric features such as curves[Tau91], edges, and symmetry[CRT04] were identified either by manual inspection or by performing calculations. Then, based on this geometric analysis, constraints are defined for ideal positions for a robot's gripper relative to the object's features. These rules were established to ensure a successful and secure grasp. Metrics are determined to obtain a grasp score to evaluate potential grasps. These metrics considered various factors, such as the number and location of contact points, ensuring that the grasp would be stable and resistant to external disturbances. Once the metrics and constraints were established, the system would select the best grasp configuration that met the desired criteria. This process relied on predefined constraints or thresholds for different grasp aspects. After selecting a grasp, the robot would execute it, and real-time feedback from sensors, such as force or tactile sensors [BLE08], would be used to make adjustments if necessary. By observing the performance, the system would gradually refine the grasp parameters, such as the position or orientation of the gripper, to improve the overall success rate of the grasp. This iterative method is used in many techniques as in [TPGSP17, TPP18, CRT04, AMO+18].

Furthermore, many grasp detection algorithms exhaustively generate grasp pose candidates

using sliding window algorithms [SWN08, JMS11, FV12] to generate grasp candidates. The oriented rectangle representation for object detection to search for optimal grasp configurations in [JMS11] is similar to our idea of using bounding boxes to represent the grasp area of the gripper finger.

Another effective method for identifying suitable candidates for the robotic grasp involves exploiting the convex body geometry of an object. The convex hull represented the object's outer boundary, encompassing the minimum convex shape covering all points on its surface. By studying the convex hull, valuable information about the object's shape and structure could be obtained, enabling the identification of potential grasp points. An innovative approach presented in [MPK+15] introduced the utilization of Gaussian process implicit surfaces (GPISs) to represent shape uncertainty based on RGB-D point cloud observations of objects. The research presented the GP-GPIS-OPT algorithm, which computed grasp configurations for parallel grippers using 2D Gaussian process implicit surfaces object representations. Sequential Convex Programming (SCP) [SHL+13] was employed to approximate the probability of force closure probability, considering antipodal constraints applied to parallel fingers. The ultimate objective was to determine the probability of force closure, indicating the gripper's ability to maintain a stable grasp on an object by exerting inward forces at the contact points, thereby preventing slippage or separation caused by external forces.

Addressing the challenge of grasping arbitrarily shaped objects observed as partial point clouds without relying on object models, physics parameters, training data, or prior knowledge, the paper [AMO+18] proposed a grasp metric based on Local Contact Moment (LoCoMo). LoCoMo incorporated zero-moment shift features from both hand and object surface patches to assess local similarity, and this metric was then utilized to search for feasible grasp poses and associated grasp likelihoods. The main principle underlying this approach involved identifying points that maximized the contact surface and leveraging areas of the object that matched the surface curvature of the gripper's fingers for a successful grasp.

Overall, the traditional non-AI approach to robotic grasp estimation heavily relied on human expertise and manual analysis. Experts would define constraints and metrics based on their understanding of object shapes, and through an iterative process, the grasp estimation system would continuously refine its parameters to achieve improved results.

The research aims to exclusively utilize object geometry for robotic grasp estimation, aiming to determine optimal seven-dimensional gripper configurations, including grasp point position, three-dimensional orientation, and grasp width (the distance between the gripper's two fingers). Additionally, our approach should accommodate physical constraints within the learning algorithm, such as the maximum opening width of the gripper or collision detection between the gripper's body and objects in the point cloud. The primary focus is to identify the most suitable surface matching between objects and the gripper, achieved by comparing the surface normals of the objects with those of the gripper to determine the best matching surfaces.

2.6 Surface Normal Estimation

Surface normal estimation is a fundamental task in point cloud processing and computer vision. It involves estimating the normal vector for each point in a point cloud, which represents the local orientation of the surface at that point. The surface normals provide important geometric information and are useful for various applications such as 3D reconstruction, object recognition, and segmentation.

There are several methods for surface normal estimation in point clouds, and can be divided in two common approaches: local methods and global methods.

- **Local Methods:** Local methods calculate surface normals based on the local geometry surrounding each point. Local neighborhood analysis using k-nearest neighbors (k-NN) is a much preferred local technique. First, for each point in the point cloud, find its k nearest neighbors., then computes the covariance matrix of the local neighborhood points and performs eigenvalue decomposition on this covariance matrix to obtain the eigenvectors and eigenvalues. At the end, the eigenvector corresponding to the smallest eigenvalue represents the estimated surface normal at that point.

  Another local method is the "oriented neighborhood averaging" (ONA) method, which computes the average normal vector of the neighboring points with respect to the query point.

- **Global Methods:** Global approaches for estimating surface normals consider the entire point cloud. The principal component analysis (PCA) based method is a very popular approach. First, to center the point cloud around the origin, it calculates the center of the point cloud and extracts the center from each point. Then the covariance matrix of the centered point cloud is calculated. Similar to local methods, by eigenvalue decomposition of the covariance matrix the eigenvectors and eigenvalues are calculated. Then, again as in local methods, the eigenvector with the smallest eigenvalue represents the surface normal of the point.

Local and global approaches have both advantages and disadvantages[MN03]. Local techniques are generally better at capturing small-scale surface details but can be sensitive to noise or inaccurate sampling. On the other hand, global approaches offer a more comprehensive assessment of surface normals but may exclude finer features.

Numerous improvements are made to these basic techniques, such as data-driven approaches, analytical methods and robust estimation techniques. The chosen method will depend on the specific requirements of the application and the characteristics of the point cloud.

Open3D [Zho21] is an open-source library and is used for 3D data processing and visualization. It offers a wide range of operations, such as rendering point clouds, modifying meshes, and saving geometry. Open3D's ability to estimate surface normals is one of its main features. To estimate surface normals, the k nearest neighbors (kNN) method is used,

where each vertex in the kNN graph is connected to each of its k nearest neighbors. The "nearness" is determined by the Euclidean distance metric [KAWB09].

$$\rho(v_i, v_j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2 \tag{2.1}$$

The kNN method of normal estimation with Open3D, the maximum number of neighbors (max nn), the number of n neighbors (k) and the neighbor search radius are the three inputs of the normal estimation methods in Open3D.

The ideal values for the max nn, k and radius parameters in Open3D for normal estimation may vary depending on the specific characteristics of point cloud data. However, some general rules of thumb can help to choose reasonable values.

- **max nn:** This parameter determines the maximum number of nearest neighbors that should be considered when calculating the surface normal of a point. In general, increasing max nn increases the accuracy of the normal estimate while reducing computation time and memory usage. Usually, a starting point for max nn between 10 and 20 is appropriate.
- **k:** This parameter determines how many nearest neighbors the consistent tangent plane routing algorithm uses. The accuracy of normal routing is often improved by increasing the value of k, but this comes at the cost of more computation time and memory usage. Similar to the maximum value of nn, a reasonable starting value for k is usually between 10 and 20.
- **radius:** This parameter controls the radius of the search sphere when determining nearest neighbors for normal estimation. The number of neighbors considered for normal estimation generally increases as the radius value increases, but this can also increase the likelihood of including distant points that are not on the surface of the target point. Typically, 0.1-0.2 times the average distance between points in your point cloud serves as a suitable starting point for the radius.

If the normal estimation got worse after setting the parameter values, it is possible that the selected parameter values are not suitable for the specific point cloud. In this case, some other options are available

As a starting point, different combinations of radius, maximum nn and k values are tried to find the parameters that best match the characteristics of the point cloud [HRD+12]. This continuous exploration allows fine-tuning and obtaining the best results. Furthermore, preprocessing the point cloud before normal estimation can be useful. If the point cloud contains noise or outliers, applying filters or outlier removal techniques can help improve the quality of the data. [HRD+12] provides a variety of methods to effectively address these issues.

If none of the options mentioned above are successful in obtaining the required results, it may be worth looking at other libraries or standard estimation procedures such as PCL

(Point Cloud Library)[RC11] or CGAL (Computational Geometry procedures Library). These libraries can provide extra features and methods to overcome specific problems.

In [KAWB09] introduced PlanePCA, which instead of minimizing the fit error, can minimize variance by subtracting the empirical mean from the $\mathbf{Q}_i^+$ data matrix and then performing an SVD on the modified data matrix $min_{n_i}\|[\mathbf{Q}_i^+ - \mathbf{I}_{k-1}\mathbf{q}_i^-] \cdot \mathbf{n}_i\|_2$.

Surface normal estimation can be quite complex, requiring extensive parameter tuning and data pre-processing for each point cloud. Through iterative experiments and modifications, it is possible to obtain satisfactory results for a given point cloud dataset and improve the accuracy and reliability of the normal estimation. Due to time and complexity constraints, instead of complex surface estimation methods such as PCA, SVD, PCL, or CGAL, the kNN method from the Open3D library is applied in this research. Of course, parameter fine-tuning is performed on the kNN method.

Chapter 3

Methodology

This section examines the implementation and methods of analytical 6-DOF robotic grasp estimation based on surface normals.

3.1 Overview

Finding secure and stable candidate grasping poses for the robot to grasp particular objects in real-life scenarios safely has always been a challenge. Several data-driven methods have already been developed to locate these candidate grasp poses. These models have been trained on datasets, and they have demonstrated a remarkable ability to locate the best potential grasp poses. The biggest problem is that deep learning model training is computationally very costly. Training models are a procedure that needs significant resources and strong computing capacity. Numerous iterations are necessary as models learn from the abundant data and modify their parameters accordingly. The magnitude of this calculation burden is a significant disincentive for the broader adoption of these models. However, before the proliferation of machine learning methodologies, robotic grasp estimation relied on traditional techniques that involved studying object geometries and adhering to predetermined methods. Traditional methods approach the grasping problem from a purely geometric perspective. They optimize grasp quality metrics based on analytical models of constraints derived from geometry and physics.

Due to the high training computational costs of the aforementioned data-driven models, instead of using data-driven methods to find safe candidate grasp poses, we mainly focus on studying the underlying local geometry of objects.

The aim of this research is to find the optimal 6-DOF grasp pose configurations for the robot using only the local shape geometry of the objects. These configurations consist of the 3D position of the grasp point, the 3D orientation of the gripper, and the width of the grasp in centimeters (the separation between the fingers of the gripper). In addition, it is defined as a fundamental physical constraint to obtain executable grasp poses: The body of the

gripper and the gripper fingers should collide neither with the object nor the environment. When examining local surface geometry similarities between objects and gripper fingers, The primary focus is determining a grasp score metric based on the optimal surface match between objects and gripper. This is achieved by comparing the surface normals of the objects with those of the gripper to identify the best matching surfaces. For this purpose, an RGB depth image is taken as input from a depth camera.

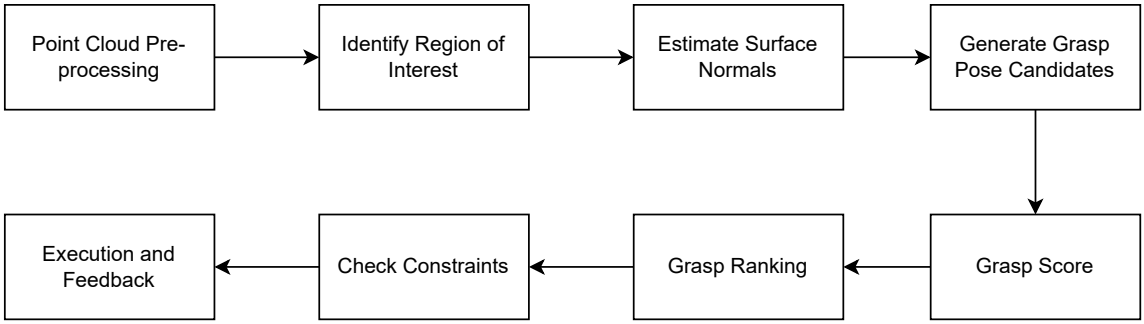


Figure 3.1: The pipeline overview of robotic grasp pose estimation method.

3.2 Point Cloud Creation

The scenes were shot in front of a table with a "Microsoft Kinect V2, Franka Emika Panda 7-DOF manipulator" camera. Objects were placed on the table and positioned not to block the camera's view.

The camera setup is given in Figure 3.3 below. The depth camera was used to create a 3D partial point cloud, and the resulting point cloud is given to the method as an input. Hand-eye calibration was performed beforehand to convert the point cloud data obtained with the camera into the robot's coordinate system and simplify the calculations. Since the scene was captured using a single camera from a single viewpoint, only a partial point cloud be obtained. There are shifts in the surface points of objects in the 3D point cloud in the direction of the camera's viewpoint. For example, missing depth readings can occur due to imperfect detection and camera noise. These shifts in the surface points are a weaker side of the presented method since it damages the estimation of applicable real-life grasp pose configurations.

Figure 3.2: Scene capturing and RGB depth camera set up.

3.3 Searching for Optimal Gripper Poses for Grasping

In this research, an algorithm that estimates 6-DOF candidate grasp poses for a safe and stable grasp is created. The steps of the algorithm 1 are summarized and explained below.

1. **Point Cloud Pre-processing:** The RGB depth image (point cloud) is taken as input. The only difference between the RGB depth image and the point cloud is that in the point cloud, the coordinates reflect the actual values in the real world[Shi18]. Hand-eye calibration was performed to convert the point cloud data obtained with the camera into the robot's coordinate system. The method of obtaining a point cloud is described above 3.2.

2. **Identifying Region of Interest:** A region of interest (ROI), $\mathcal{R}$, where the grasp is likely to occur is identified. Some parts that do not contain objects or are too irrelevant for the grasp estimation task are cropped from the point cloud.

3. **Estimating Surface Normals of Point-Cloud:** In the next step, the surface normals at each point of point cloud, $x \in \mathcal{X}$, are computed. The method of estimating surface normals of point cloud is examined in detail below 3.3.1.

4. **Generating Grasp Pose Candidates:** The goal of generating grasp candidates in algorithm 1 is to find a large set of grasp candidates (i.e., 6-DOF grasp poses) where a secure grasp might be located. These grasp candidates are distributed over the graspable parts of the object surface as evenly as possible. Several thousand candidate grasps (the size of candidate grasps are depend on the size of the point

cloud) are sampled at homogeneous down-sampled points inside the region of interest $\mathcal{R}$. Each candidate in the region of interest has a 6-DOF grasp pose, three dimensions showing the grasp pose's center point (the point in $\mathcal{R}(x)$), a random 3D orientation, and one dimension indicating the width of the grasp in centimeters (the distance between the gripper's fingers). Gripper is aligned to fit as closely as possible to the surface of the object after sampling the 6-DOF grasp pose candidates. Following this alignment, the grasp poses are rotated 30 degrees in the x-axis at a time until they return to their original location. This rotation allows the sampling of 12 gripper poses that are centered at each down-sampled point cloud. Then, the candidate grasp poses where the gripper body and finger body collide with the point cloud are eliminated.

5. **Grasp Pose Score:** The grasp score metric is defined to observe the surface geometry similarities between the object and the gripper finger. For this purpose, an error value is obtained by summing the surface normals of the object and the gripper finger. Since the normals are in opposite directions, the sum of the normals represents the error of matching the surfaces. This error is inserted into the defined grasp score formula to assign a grasp score to the candidate grasp pose with certain 6-DOF configurations. The error and the grasp score are calculated for each 6-DOF gripper pose configuration that does not collide with the point cloud. The grasp quality metric will be described in detail below 3.3.4.

6. **Grasp Pose Ranking:** Depending on the grasp score, each candidate is assigned a color on the "RGB" scale indicating the candidate's suitability for the grasp. Red represents the lowest grasp score, while green represents the highest grasp score. Grasp poses in green are suitable poses for the grasp. The grasp score representations on the gripper can be observed from the Figure 3.7 below.

7. **Check for the Constraints:** The algorithm has two constraints: The first and most obvious one is the maximum grasp width, which is basically the maximum possible span of the gripper's fingers. This constraint is given at the beginning of the algorithm as the maximum gripper finger span, defined as 10 centimeters. The second constraint is that the body of the gripper is not in collision with the point cloud: $\mathcal{X} \cap \mathcal{B} = \emptyset$

8. **Execution and Feedback:** After the grasp score is evaluated in the grasp selection step, the next step considers not only the grasp score but also other factors that influence the feasibility and success of the grasp in a real-life scenarios. This step aims to select the feasible grasp pose for the robot to execute robust and effective grasp pose execution through motion planning.

Algorithm 1 Grasp pose sampling algorithm

Require: Point Cloud $\mathcal{X}$, Grippers' 3D model
Ensure: Non-colliding grasp poses
Identify a region of interest $\mathcal{R}$ in point cloud
Remove the ground surface plane
Identify a main region of interest $\mathcal{R}_m$ to eliminate grasp poses on the edges of the $\mathcal{X}$
Compute the surface normal at each point $x \in \mathcal{X}$
for $\forall x \in \mathcal{R}_m$ **do**
    for grasp width=2,3,...,10cm **do**
        Create gripper at point x with random 3D orientation
        Compute desired normal
        Align gripper normal to the object surface depending on desired
        for degree= 0,30,...,360 **do**
            Rotate gripper around x-axis at grasp point x
            Check constraint $\mathcal{X} \cap \mathcal{B} = \emptyset$
            Compute grasp score
            Color candidate grasps depending on grasp score
            Store grasp configuration
        end for
    end for
end for
Return the non-colliding grasp poses

Many objects, including those without flat surfaces, can be held with a parallel jaw gripper with flat fingers. However, the grasp would be more secure and stable on objects with parallel flat surfaces. In this research, the 6-DOF grasp pose estimation algorithm is planned to be applicable to these flat surfaces. The goal of this phase is to find relatively flat surface regions that can be grasped is the goal of this phase.

In the following sections, the steps of Algorithm 1 are discussed in detail.

3.3.1 Surface Normal Estimation

The estimation of surface normals is of fundamental importance in the application of analytical methods. It can be used to process point clouds. For each point in a point cloud, the normal vector representing the local orientation of the surface at that point is estimated. The resulting surface normals provide important geometric information and are extremely useful. This research uses surface normals to determine a grasp quality metric. The grasp score is determined from the correspondence between the estimated surface normals of the objects and the normals of the gripper's fingers. For this, it is crucial to estimate the object surface normals correctly. In order to determine the candidate grasp poses correctly, it is aimed to estimate the object surface normals as accurately as possible by setting appropriate parameters.

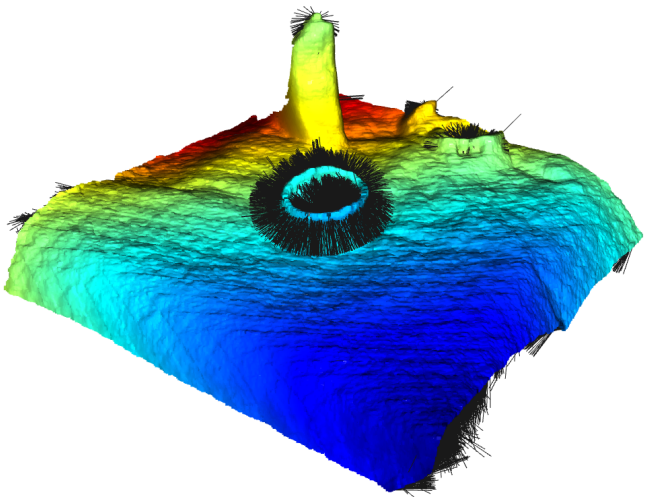


Figure 3.3: Surface normal estimation of the point cloud: black vectors represent the surface normals.

There are various methods for estimating surface normals in point clouds, and the two most common approaches are local methods and global methods, which are explained in chapter 2. In this research, local methods are used. Local methods estimate surface normals based on the local geometry around each point. This research applies the hybrid KDTree search algorithm from the Open3D library.

The method first finds k nearest neighbors for each point in the point cloud and constructs a KDTree from the point cloud. Thus, it optimizes the nearest neighbor searches. This search process aims to find neighboring points within a given radius or the maximum number of nearest neighbors given as search parameters. This process is repeated for all points in the input point cloud. The KDTree search algorithm efficiently estimates surface normals for each point in the input point cloud, taking into account the specified search parameters. Good specification of these parameters is critical for accurate estimation of surface normals.

The KDTree method described above can have some problems. It can examine local surface details. However, the partial point cloud is also very important for the accuracy of the KDTree search. Noisy or irregular sampled partial point clouds can lead to miscalculations.

During the estimation of surface normals from the input partial point clouds in this research, it was observed that the orientations of some normals deviated due to miscalculations. As the presented grasp pose estimation algorithm 1 heavily relies on accurate surface normals, these deviations cause a significant challenge. To solve this issue, various approaches were implemented. Improving the accuracy of the estimated surface normals ensures the reliability and efficiency of the algorithm.

Initially, parameter fine-tuning was performed by a trial-and-error. For the number of nearest neighbors (k), values ranging from 10 to 20 were tried, ultimately determining that 12 yielded optimal results. Then, the values between 0.1 and 1 were tested for the radius parameter.

After trial and error, two different methods were attempted. Firstly, the radius was determined using the formula given in Equation 3.1. However, it was noted that this approach resulted in weaker surface normal estimation.

$$\text{radius} = 3 - 4 \times \text{Average1stNNdistance} \tag{3.1}$$

Another approach was adopted to search more accurately for the radius parameter. This method calculated the average distance between the points, and values of 0.1 and 0.2 times the average distance were initially selected as the radius. However, it was observed that this approach resulted in a deteriorated normal orientation. A second approach was tried to search for a more accurate radius parameter. In this approach, the average distance of the point cloud points was calculated, and the average distance was multiplied by 0.1 and 0.2 to select the radius value. However, the radius parameter value obtained in this approach also made the normal estimation worse.

The values obtained by both methods were no better than the radius obtained by trial and error. Furthermore, applying these approaches to dense point clouds such as the one used in this research slowed the candidate grasp pose estimation algorithm by almost 5-6 minutes. Therefore, in the end, the radius was set to 0.1 on average for general point cloud samples.

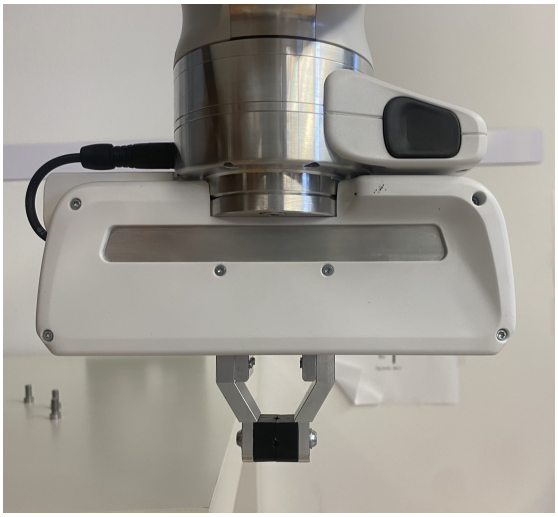
In this research, the surface normal estimation task caused a significant challenge and required detailed parameter tuning and pre-processing for each point cloud. Through the experimentation as mentioned above and fine-tuning processes, satisfactory results were finally obtained, proving the effectiveness of the proposed approach. By that reliability of the proposed grasp pose estimation method is secured.

3.3.2 Define Gripper

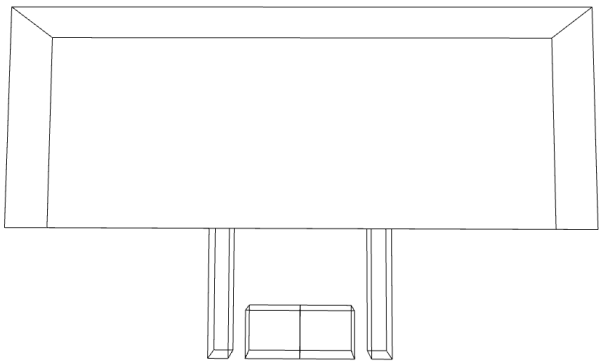
A parallel gripper can be modeled as three parts: two fingers that move against each other along a one-dimensional space, two finger bodies connecting the fingers to the gripper body, and a gripper body. A basic example of a two-fingered hand is a standard parallel jaw gripper. In this research, the typical parallel jaw gripper is used. Since only two-fingered parallel grippers are used, it is sufficient to examine the similarity of the surface of these two fingers to the object's surface to calculate the pose that will allow the gripper to grasp the object securely.

In this research, five oriented bounding boxes were defined to represent the gripper body, two gripper fingers and the bodies of the gripper fingers. The size of the bounding boxes corresponds to the size of the original gripper: 1.75cm by 1.75cm for the gripper finger, 0.75cm x 1.75cm x 4.75cm for the gripper finger bodies and 21cm x 4cm x 8cm for the gripper body.

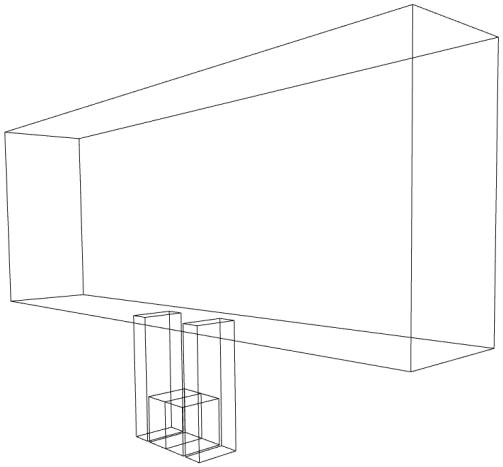
Various analyzes can be made from the points inside these bounding boxes. For instance, collision detection becomes possible by analyzing the presence of points within the body and finger-body bounding boxes, allowing the algorithm to identify potential collision areas. Furthermore, the surface normals of the points inside the bounding boxes could be compared to the normals of the gripper fingers to examine the similarity of the surfaces. This comparison allows for a deeper examination of the surface similarity between the gripper fingers and the object, providing valuable information about the feasibility and appropriateness of specific grasp poses. By leveraging these oriented bounding boxes, the algorithm improves the ability to proactively identify potential collisions and assess the compatibility of grasp configurations, resulting in safer and more robust estimation poses. By representing the gripper finger bodies with oriented bounding boxes, it is possible to precisely measure whether the gripper's fingers will collide at points near the object.



(a) Gripper of the robot.



(b) Gripper view from front.



(c) Gripper view from side.

Figure 3.4: Gripper Representation.

3.3.3 Grasp Pose Sampling

To initiate the gripper pose sampling process, the first step involves obtaining a homogeneous down-sampled alternative point cloud through a farthest point down-sampling technique from the Open3D library. The aim is to reduce the unnecessarily high computational cost and thus create a low-cost system. Selecting the farthest points allows for a better and more efficient representation of the point cloud. The candidate's grasp poses are sampled at homogeneous down-sampled points of the region of interest. Here, each candidate has a 6-DOF grasp pose, three dimensions for showing the center point of the gripper fingers (the point in the ROI), a random 3D orientation for the gripper, and the width of the grasp in centimeters (the separation between the fingers of the gripper). After sampling these 6-DOF grasp pose candidates, the gripper is aligned to fit the object's surface optimally.

The gripper is initially placed at points in the down-sampled point cloud with a random 3D orientation. However, the optimal grasp pose is searched for the gripper to make a successful grasp. Therefore, the gripper must be aligned to fit the object's surface best. To align the gripper pose, the average of the surface normals of the point cloud points inside the fingers of the randomly oriented candidate pose of the gripper is calculated. The desired normal of the gripper pose candidate is obtained by multiplying this average normal by minus one and then inverting it. In order to achieve the closest normal to this desired normal, the angle between the current normal of the gripper poses candidate and the desired normal is calculated. The candidate gripper pose is rotated and aligned with this angle. These steps generate grasp pose candidates that are appropriately aligned to the surface of the objects across all points of the efficiently homogeneous down-sampled point cloud.

Once the gripper is aligned, the grasp poses are sampled by rotating them around the x-axis every 30 degrees until a full round is completed. The right-hand rule is used for the rotation, which codifies alternating signs. The right-hand rule only works when multiplying $R_x(\theta) \cdot \vec{x}$ [LC14]. The rotation matrix for rotating around the x-axis is given below.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix}$$

After aligning and rotating the grasp pose candidate, the collision constraint is checked, which is $\mathcal{X} \cap \mathcal{B} = \emptyset$, where $\mathcal{B}$ represents the body of the gripper and $\mathcal{X}$ represents the point set of a point cloud. If no points are detected in the gripper body bounding box or the gripper finger body bounding box, it indicates that the gripper will not collide with the object or the environment. However, in the parts where there are shifts in the points in this point cloud, it may cause some suitable grasp poses to be eliminated due to the specified constraint. In order to avoid this situation, a little bias value is set. In this case,

up to 3 points are allowed in the bounding boxes. This 3-point is a convenient bias value considering that there are a total of 200000 points on average in the point cloud. That eliminates grasp poses where the gripper collides with the point cloud. It ensures that the constraint is met. The remaining samples of 6-DOF grasp poses that do not collide with the object environment are stored.

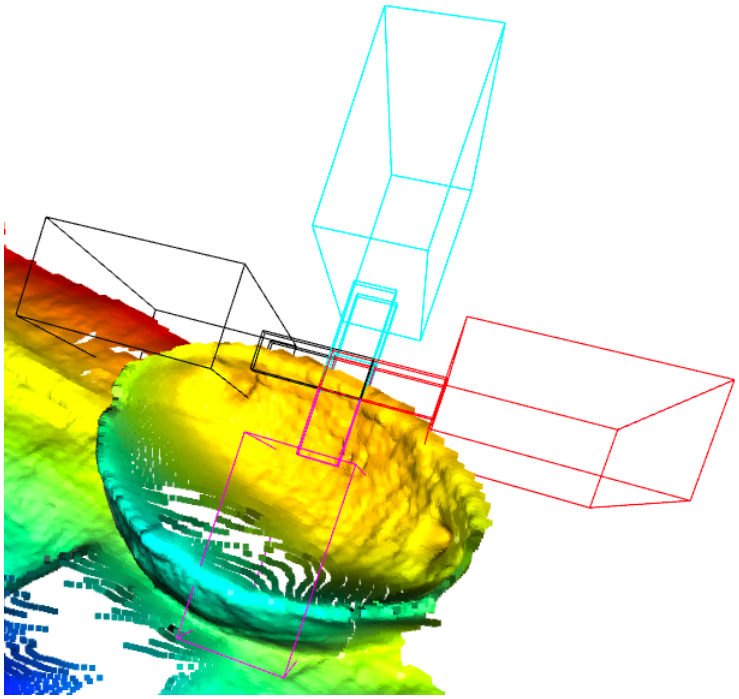


Figure 3.5: X-axis rotation.

In summary, the process starts with obtaining a down-sampled alternative point cloud, then a large number of candidate grasps are sampled at points in the down-sampled alternative point cloud and aligned to the surface of the object, taking into account surface normals. At each point, one candidate pose is extracted for every 30 degrees, which corresponds to 12 poses per point. In total, 230400 candidate pose samples are extracted for 19200 points. 19200 points are the number of points in the down-sampled point cloud. Finally, candidate grasp poses that collide with the point cloud are eliminated. This leaves candidate grasp poses that are compatible with the surface geometry of the object and suitable for grasping.

3.3.4 Grasp Score

In this research, it is aimed to use the surface normals of objects to grasp objects of various shapes. The focus of using surface normals is to identify similar surfaces between the object to be grasped and the surfaces of parallel gripper's fingers. To this end, a grasp score, which is actually a surface similarity metric, is defined. When calculating this grasp score, the set of points containing the objects is assumed to be pre-processed and filtered from outliers.

The aim is to obtain a grasp score to observe the suitability of a grasp pose candidate. First, the candidate for the grasp is placed at a point on a pose point cloud. Then the individual grasp score of each point is determined and this score is calculated for each point within the gripper fingers. The matching error between the surface normals of the object point and the gripper's finger normal is calculated. The resulting error is added to the similarity function that gives the grasp score for the point. In this way, the grasp score is calculated for only one point. However, this grasp score needs to be integrated over the entire surface of the object. This process is repeated on each point between the fingers of the candidate grasp pose. Then the average of the grasp scores of each point is calculated. This gives a grasp score for the candidate grasp pose.

The comparison of two local surfaces can be reduced to the comparison of the function S in 3.2 defined to compute the surface similarity between two surfaces, where $\phi(X, \mu, \Sigma)$ represents a gaussian density function.

$$S = \frac{\phi(\epsilon; \vec{0}, \Sigma)}{\max(x, \phi(x, \vec{0}, \Sigma))} \tag{3.2}$$

Gaussian density function is given below in 3.3, which is used from the paper [AMO+18], where $X, \mu \in R^n, \Sigma$ is the covariance matrix, n the space dimension and $\vec{0}$ is the null vector of R^3.

$$\phi(X, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(X - \mu)\Sigma^{-1}(X - \mu)\right) \tag{3.3}$$

ϵ, represents the difference between the matching of surfaces. The error should be low if the surface normals of the gripper fingers and the object match precisely, otherwise it should be high.

The error 3.4 is calculated by adding the normals of each point inside the gripper finger bounding box and the normals of the gripper's finger to each other. The reason for adding surface normals is that the normals are opposite to each other. This process is applied for both fingers. After calculating the error, it is inserted in 3.3 as $\phi(\epsilon; \mu; \Sigma)$, where ϵ is the calculated error, μ is zero vector $\vec{0}$ and $\Sigma \in R^3$ is a unit matrix.

$$\epsilon = n_{pcd} + n_g \tag{3.4}$$

n_{pcd} and n_g are expressed in the same reference frame. $max(x, \phi(x, \vec{0}, \Sigma))$ is the maximum value of the function $\phi(x, \vec{0}, \Sigma)$ for all $x \in R^3$. When error is minimum, then highest value for similarity function S is expected. Thus, maximum value for S is equals to 3.5

$$\phi(\vec{0}; \vec{0}; \Sigma) = \frac{1}{\sqrt{(2\pi)^n \, |\Sigma|}} \tag{3.5}$$

The surface normal of the gripper finger is calculated by subtracting the centers of the bounding boxes of the gripper fingers from each other. The normals of the fingers have the same magnitude and direction but are opposite to each other. It is shown in the figure below 3.6. The red vector is the normal of the gripper finger on the left side.

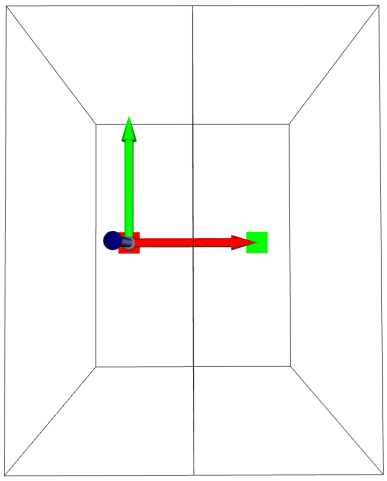
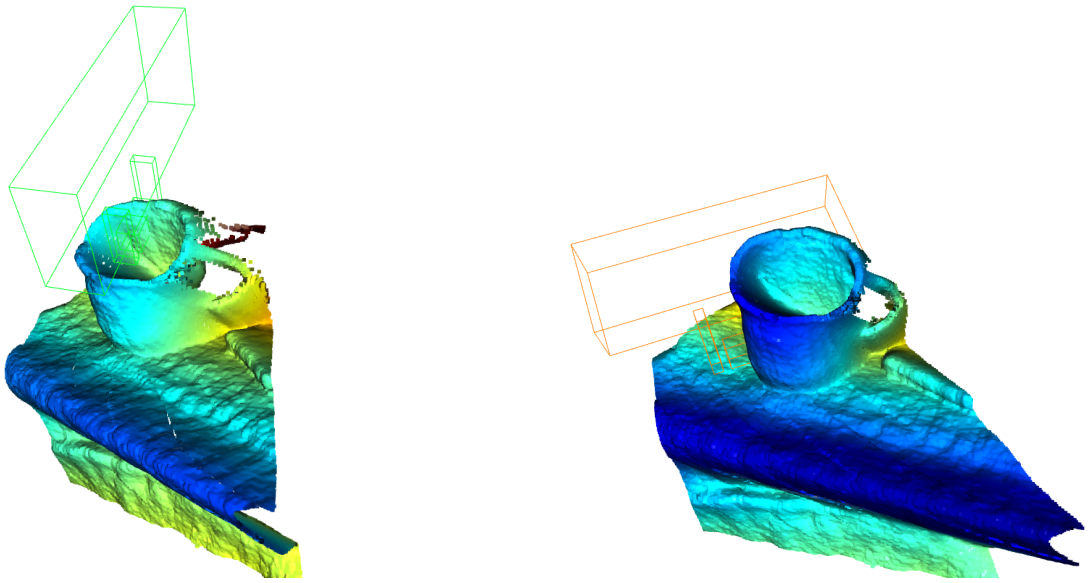


Figure 3.6: Gripper finger normal.

This metric, based on the properties of surface normals, is extremely useful for grasping as it provides a clear indication of the local similarity between the surfaces of a gripper and an object. By comparing the surface normals of the gripper and the object at a given point of contact, one can assess how well they are aligned and determine their compatibility for successful gripping. This information allows the suitability of a gripper configuration to be evaluated and informed decisions to be made during robotic manipulation tasks.

The grasp score is an important factor for the robot to manipulate objects safely and accurately. It allows the quality and feasibility of a grasp to be measured and evaluated. The score considers factors such as how well the gripper is aligned to the object surfaces, the stability of the contact and how well the grasp pose configuration matches the shape of the object. By choosing a high-scoring grasp poses ensures that the selected grasp pose is



(a) High grasp score. (b) Low grasp score.

Figure 3.7: Grasp score representation.

more likely to succeed and reduces the risk of the object slipping or the grasp failing. These grasp scores are assigned to the 6-DOF candidate poses sampled. The highest-scoring pose configurations are given to the robot and then the robot can perform safe and effective object manipulation.

3.3.5 Refinement of Grasp Pose Sampling

Several improvements have been made to increase the accuracy and reliability of the grasp estimation algorithm, including the inclusion of an additional step to eliminate unsuitable grasp poses. This improvement aims to facilitate the selection of feasible and successful grasp configurations while eliminating poses that are unsuitable or unlikely to result in a stable grasp.

The inappropriate grasp pose elimination considers various factors such as collision detection with the object or environment, kinematic constraints of the robotic system, and grasp stability and feasibility considerations. By implementing this mechanism, the algorithm becomes more robust. It can generate grasp poses that not only align well with the surface of the object but also adhere to practical constraints, thus increasing the likelihood of successful grasping operations in real-world scenarios.

Eliminate grasp poses on the edges of point cloud

In Figure 3.8, it is clear that a significant number of inappropriate grasp exposures are located at the edges of the object's point cloud. However, these edge grasps are redundant and can potentially lead to errors or inaccuracies in the grasp estimation process. Furthermore, they significantly reduce the overall performance and reliability of the algorithm, hindering more accurate and feasible grasp pose estimation for successful robotic manipulation tasks. Recognizing the impact of these edge grasps, additional measures have been incorporated into the algorithm to effectively identify and eliminate such poses during the grasp selection phase.

Moreover, the existence of perfectly flat surfaces along these edges causes a particular challenge for the algorithm, as it tends to assign high scores to grasps in these regions of the point cloud. However, this can damage the algorithm's performance, as the actual high-scoring grasp poses may not be counted among the best grasp poses. Consequently, during the execution phase, when the generated 6-DOF grasp configurations are tested with the robot, it may prioritize grasping the edges of the experimental surface rather than focusing on more reliable grasp locations. This highlights the importance of refining the algorithm to handle such edge cases better and ensuring that the selected grasp poses align with the true grasp quality, leading to improved grasp success rates and more efficient robotic manipulation.

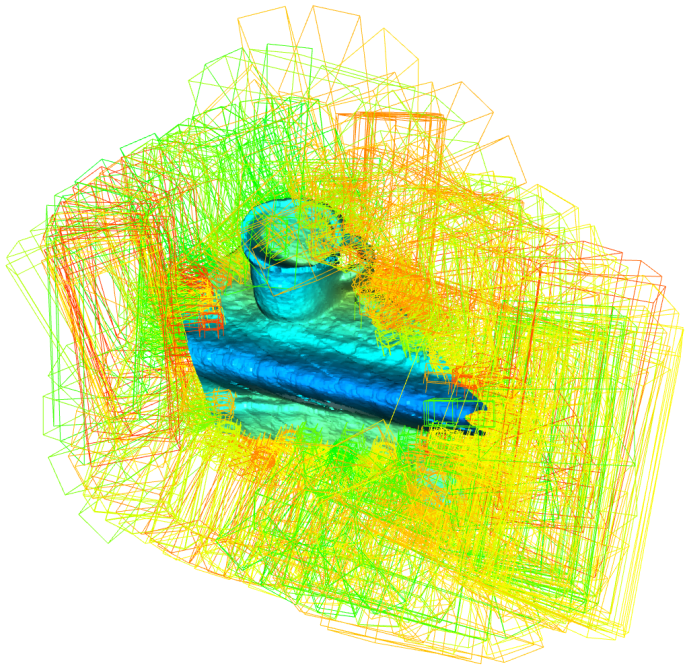
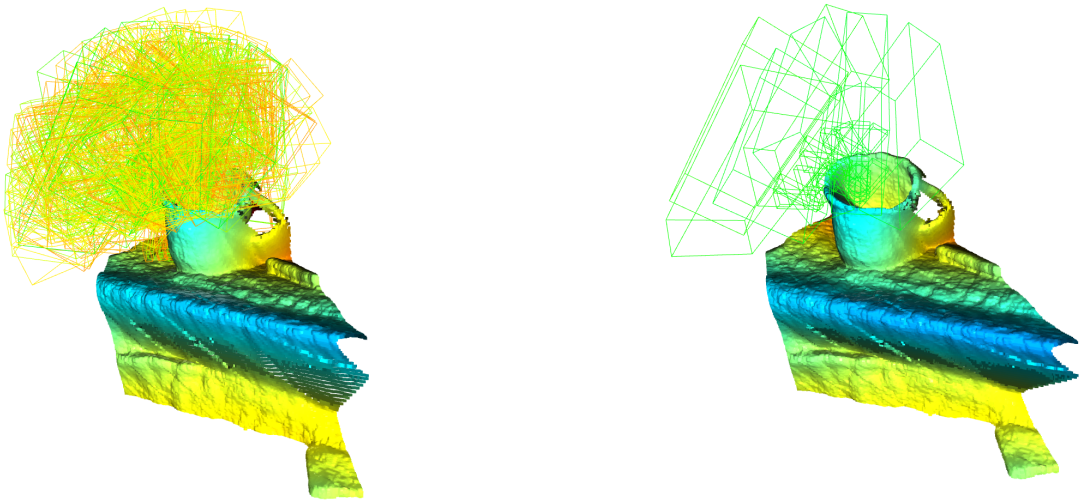


Figure 3.8: Candidate grasp poses before eliminating grasp poses on the edges.



(a) Candidate grasp poses after eliminating grasp poses on the edges.

(b) Best grasp poses after eliminating grasp poses on the edges.

Figure 3.9: Candidate grasp poses after eliminating grasp poses on the edges.

Another advantageous aspect of eliminating grasp poses that are on the edges, the algorithm's efficiency is improved as the computational cost and processing time are reduced. By having fewer grasp configurations to evaluate and score, the overall performance of the algorithm is optimized. Furthermore, the elimination of edge grasp poses results in a more refined and accurate set of candidate grasps, focusing only on the most appropriate and feasible grasp poses. This improvement not only increases the reliability and success rate of subsequent robot grasping attempts, but also contributes to an overall more efficient and effective grasp process.

To eliminate gras poses on the edges, a bounding box is defined around the objects. This area inside the bounding box is defined as the main region of interest. Grasp poses are only sampled at points inside this bounding box $\mathcal{R}_m$; thus there are no candidate grasp poses on the edges. Those on the edges of the newly defined bounding box are automatically eliminated as they collide with the point cloud $\mathcal{X}$, which means with the environment.

Edge isolation has been applied to optimize the grasp pose estimation performed on the partial point cloud. It reduces the computational cost by eliminating parts that should not be grasped, such as the experimental surface (in this case table), and provides a safe environment for the robot not to collide with the environment. Without removing the grasp poses on the edges of the point cloud, the processing time is much more as the algorithm finds many more grasp poses. For example, if the number of candidate pose configurations is usually 150, this increases to 1500-2000 when the edges are included.

In addition, the fact that the surface is flat (the surface of the table is flat) and perfectly matches the flat surface of the gripper's finger results in high-scoring grasp poses, proving that the algorithm works perfectly in detecting the similarity of the normals of the object surface and the surface of the gripper finger. However, eliminating this high-scoring grasp at the surface/ground edges pushes candidate grasp poses that are feasible and even desirable in real life to the top of the ranking. Thus, the 10 highest-scoring candidate grasp poses can be selected from those suitable and desired poses for the robot's execution in real scenarios. This 6-DOF candidate grasp pose configuration is given as input to the robot's motion planning, which is given to execute the grasp pose in real life.

Remove the ground surface from the point cloud

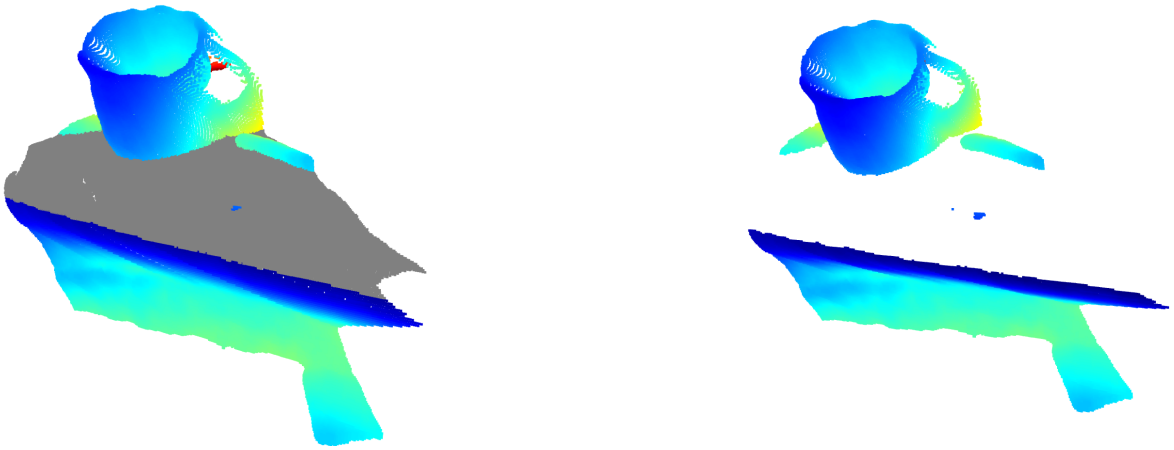
Sampling candidate grasp poses on the ground surface of the point cloud will not add any valuable insights or results to the method. This is primarily because the ground surface is often not a suitable location for finding viable grasp poses. Its flat and horizontal nature usually suggests it has the necessary object properties for a successful grasp. This will cause the method to detect non-feasible poses, but these poses will be eliminated anyway by collision detection. Therefore, devoting computational resources to studying the ground surface will lead to an unnecessary increase in the overall processing time of the method.

To address this inefficiency and simplify the computational workflow, a optimization strategy is implemented. Removing the ground surface from the point cloud analysis significantly improves the computational performance of the method. This allows the algorithm to focus only on region of interest where suitable grasp poses are more likely to be found. This results in a more efficient and targeted grasp estimation process. This optimization not only speeds up the overall execution time, but also improves the method's ability to handle larger datasets or real-time grasp applications where computational efficiency is extremely important.

A Random Sample Consensus (RANSAC) [Der10] algorithm is implemented to segment and removes a planar surface from a point cloud. Plane segmentation is performed using RANSAC. To segment the plane the distance threshold, the minimum number of points required to define a plane, and the maximum number of iterations for RANSAC are specified. By that, a plane equation is defined in the form of $ax + by + cz + d = 0$, where a, b, c, and d are the plane equation coefficients. Two lines are then identified: The first one, the "inlier cloud" , selects inlier points, which lie on the detected plane. The second one, "outlier cloud", reverses the selection and selects outlier points, which are points that do not belong to the detected plane. Thus, the "outlier cloud" is obtained, representing the point cloud with the plane removed.

As can be seen from the results of removing the ground surface in Figures 3.10 below, this refinement has been implemented very accurately.

The overall improvement of the grasp pose estimation algorithm has significantly enhanced its performance and reliability in several aspects. First, the inclusion of unsuitable grasp



(a) Detect the ground surface of the point cloud.

(b) The ground surface of the point cloud is removed.

Figure 3.10: Remove the ground surface of the point cloud.

pose elimination has contributed to the generation of more suitable and reliable grasp configurations by eliminating poses that are unsuitable or unlikely to result in stable grasps. This improvement ensures that the grasp poses selected for collision detection, taking into account kinematic constraints, are compatible with practical constraints and increase the overall success rate of grasping operations.

Furthermore, the algorithm has been improved to eliminate inappropriate grasp poses located at the edges of the object's point cloud. By recognizing the negative impact of these edge grasps and taking measures to eliminate them, the performance of the algorithm has been improved, resulting in more accurate and feasible grasp pose estimation. This improvement is particularly important as it prevents the algorithm from assigning high scores to grasps on flat surfaces along edges, which can mislead the selection process and lead to sub-optimal grasp selection during robot execution.

In conclusion, improving the grasp pose estimation algorithm has improved grasp quality and the success rate of grasping operations. Besides, these refinements ensures that the selected grasp poses are compatible with the actual grasp quality and practical constraints. These improvements contribute to the algorithm's overall reliability, adaptability and applicability in real-world robotic manipulation tasks, enabling robots to perform more efficient and successful grasping actions on various objects and scenarios.

Chapter 4

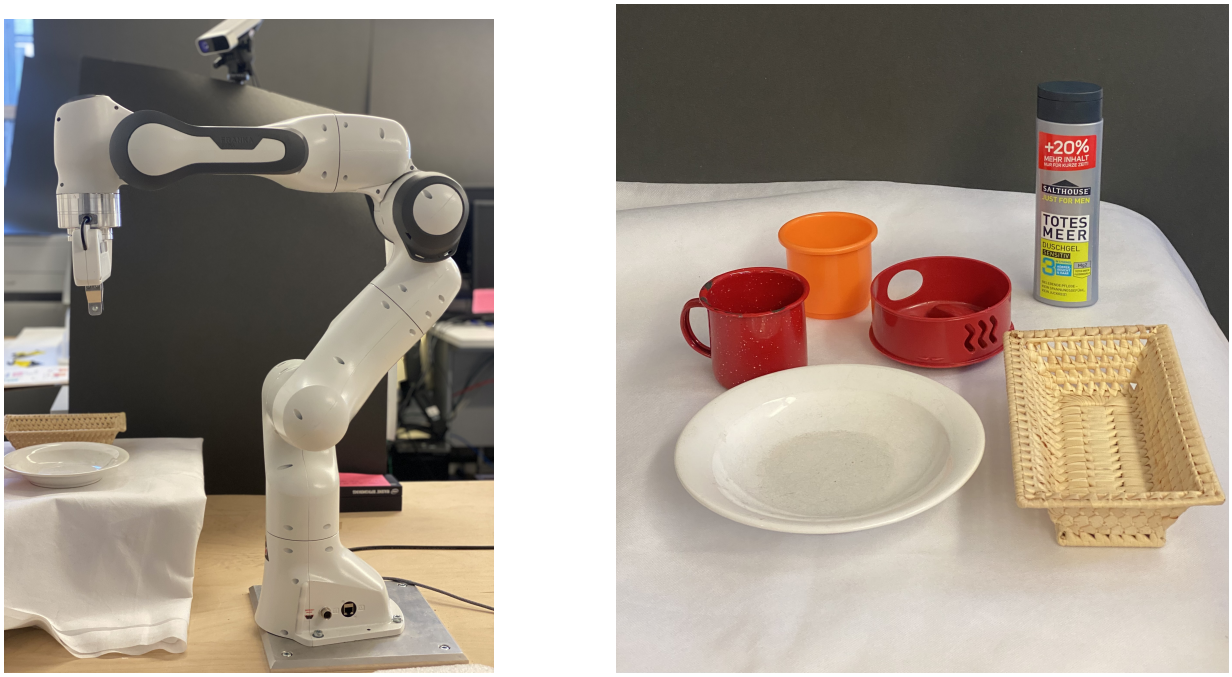
Results

This section describes the findings and contributions made by some experimental results. The experiment is conducted in a real environment using a "Microsoft Kinect V2, Franka Emika Panda 7-DOF manipulator" robot to evaluate the grasp score metric. These experiments aim to determine the surface similarities of the object and gripper fingers calculated by the method defined in this research and to evaluate the validity of the grasp scores given these similarities using various objects. The objects selected for the tests are a wicker basket, a plastic mug, a metal mug, a plate, a shampoo bottle, and a jug lid. After calculating the grasp scores for each object, obtained grasp scores are sorted in decreasing order. The top ten grasp pose configurations with the highest grasp scores were selected. These poses represented the most suitable 6-DOF grasp configurations for the objects in the context of the experiment. The robot then manipulates the objects appropriately by applying these poses. The developed method neither used any prior knowledge of the scene nor used any object models.

The validity and applicability of the concept relevance metric were thoroughly tested by performing these experiments on real objects in a real environment.

4.1 Experimental Setup

The experimental setup (shown in Figure 4.1a) consists of a "Microsoft Kinect V2, Franka Emika Panda 7-DOF manipulator" with 6 degrees of freedom mounted with a parallel jaw gripper with a flat finger. Hand-eye calibration has been performed beforehand to transform the camera-acquired point cloud data to the robot's coordinate system as well as to simplify the computations. After calibrating the camera, the robot is manually mobilized to the target area. In this way, it is aimed that the method is to minimize the impact of the robot's joint motion constraints on execution when applying grasp poses. There are two main reasons for computing the robot's motion plan in joint space instead of Cartesian space. First, joint space planning plans the robot's movements by controlling



(a) Robot.

(b) 6 objects used for the experiments.

Figure 4.1: 6 objects used for the experiments: wicker basket, plastic mug, metal mug, plate, shampoo bottle and jug lid.

joint angles. This approach allows the robot to follow the kinematic chain strictly and offers more precise control. Second, joint space planning can also better adapt to the requirements of moving in areas surrounded by obstacles. While Cartesian space planning requires extra complexity to account for obstacles, joint space planning can optimize the robot's joint angles to move around obstacles.

The proposed grasp method has been tested on 6 objects, as shown in Figure 4.1b, comprising a wide variety of shapes, masses, and materials. The objects are selected such that they are small enough to be physically graspable by the used gripper.

Two sets of experiments were conducted. Firstly, it is tested the robot's ability to grasp and lift individual objects from the surface of a table. The second set of tests was performed to analyze the robot's ability on reconstructed point clouds. By that, it is expected to be a method to detect more realistic grasp points in real life. During trials, the method took 90 seconds (on average) to generate 350 grasp hypotheses for a point cloud with 243292 data points.

The processing time is not the primary concern since the presented algorithm is not developed for real-time applications. The time required to find successful candidate poses is sufficient for a method that is not intended to run in real-time. In order to achieve this processing time, some operations were still applied. First, the outliers of the point cloud

are removed. As mentioned in the chapter 3, the region of interest for the candidate grasp search is defined and the rest is cropped from the original partial point cloud. Second, the point cloud points of the experimental surface (in this case, the table) are extracted. Next, the aim is to reduce further the number of points visited, for which the farthest point sampling algorithm is applied to the cropped point cloud.

A color scheme was used in the experiment to represent the grasp scores for the different candidate grasp visually poses, thus aiming for a clear visualization of the quality and appropriateness of each grasp pose. The colors assigned to the candidate grasp poses correspond to the grasp scores and are intended to facilitate a straightforward interpretation of the results. The candidate grasp poses with the highest scores was assigned a vibrant green color in the color scheme used. This color signified that these poses had achieved the highest level of grasp quality according to the algorithm's scoring criteria. For grasp poses with mediocre scores, a range of colors from yellow to orange was used. Finally, the grasp poses with the lowest scores are shown using the color red. This helped to identify poses that were considered less appropriate or potentially problematic according to the algorithm's scoring methodology. Using this color representation, the quality of each grasp pose candidate can be quickly observed. Visualizing the grasp scores helped to analyze their distribution across the parts of the objects and to evaluate the overall performance of the algorithm in grasp scenarios. Grasp scores are normalized between 1.0 and 0. Grasp poses with a grasp score above 0.75 represented in bright green and are considered excellent and executable grasp poses.

Figure 4.2: Color scheme.

Colors	**Grasp Score**
Green	> 0.75
Light Green	$0.75 > 0.60$
Yellow	$0.60 > 0.45$
Orange	$0.45 > 0.35$
Red	$0.35 >$

Table 4.1: Grasp score scale in RGB.

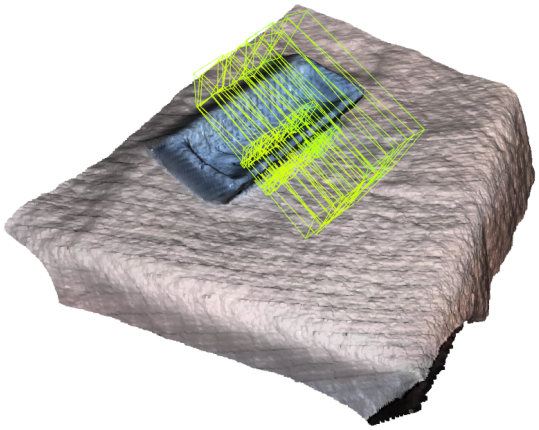
4.2 Grasping Individual Objects

The first experiment evaluates the robot's ability to grasp and lift individual objects on a flat table surface. 6 objects were used, with 10 grasping trials performed on each object. Each object is placed randomly on the table with different orientations and positions for each of the ten trials. After capturing and registering partial point clouds, points belonging to the table surface are filtered out, and the resulting object point cloud is then used to generate grasp hypotheses, as described in Algorithm 1. The original goal of the experiment was to examine a wider variety of items to fully assess how well the algorithm performed. However, as the experiment progressed, it became clear that some objects created challenges for the algorithm while others worked just fine. In order to focus on learning the advantages and disadvantages of the algorithm, it was decided not to add more objects in this particular evaluation phase.

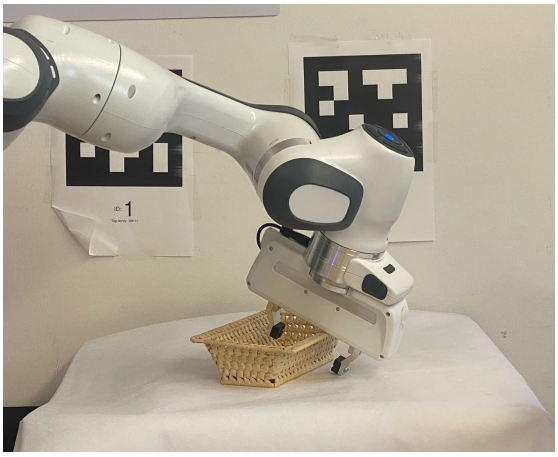
The behavior of the algorithm was carefully examined to find potential areas for improvement. By limiting the range of objects to those initially selected (wicker basket, plastic and metal mug, plate, shampoo bottle and jug lid), the experiment aimed to give a clear picture of the algorithm's capabilities and limitations in context. The areas where the algorithm struggled and performed best were identified. The algorithm can be further developed and improved using this knowledge, so that its errors can be corrected and its performance improved. The experiment did not cover a wide range of objects, but this was done to examine the algorithm's performance more accurately. This method allows the algorithm to gain in-depth knowledge about its performance on selected items, allowing for further refinement and optimization.

The challenges and limitations mentioned here will be addressed later when discussing detailed examples.

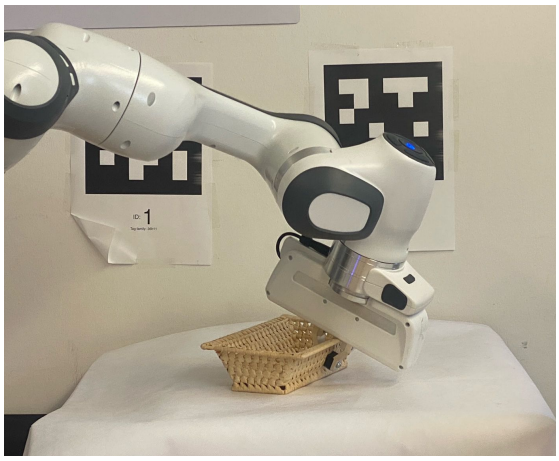
A wicker basket was chosen as the first object to evaluate how well the algorithm can grasp and control objects. The wicker basket was chosen as the starting point of the evaluation process due to its smooth shape and flexible structure. After placing the basket in various orientations on the table, the camera captures the scene, and a partial point cloud is obtained. This single view partial point cloud is given to the algorithm 1 as input. Then grasp scores are given to multiple grasp hypotheses as described in methodology chapter 3. Afterward, the grasp scores are ranked depending on their grasp scores, and the top 10 grasp poses are saved in a file. The grasp poses with the highest grasp score are executed. In the end, grasp is recorded as successful if the robot manages to grasp and lift the object to a post-grasp position 5 cm above the table and hold the object for more than 5 seconds without dropping it. This process can be observed from Figure 4.3. As a result, we observed a perfect execution for all types of placed baskets and a smooth execution of the given grasp poses from the algorithm.



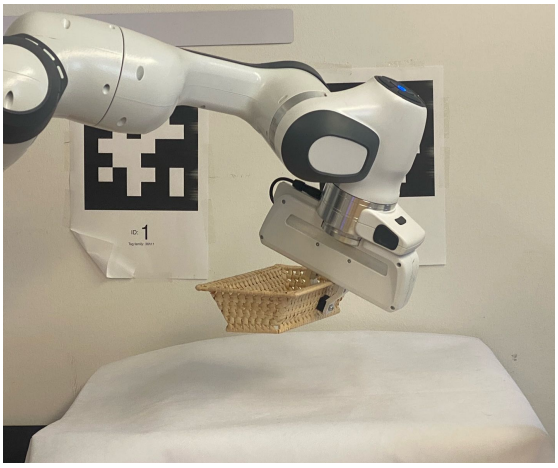
(a) Point cloud of the basket.



(b) Pre-grasp pose.



(c) Grasp pose.



(d) Post-grasp pose.

Figure 4.3: Grasp pose estimation of the basket.

Analysis of the experimental findings led to an interesting finding about the holding motions of the wicker basket. It is noticeable from the Figure above 4.3a that all the high-scoring grasp poses are on the front of the basket from the camera's perspective. This spatial distribution of highly rated grasp poses can be attributed to the characteristics of the wicker basket and the camera's perspective. The points on right side of the basket are shifted relative to the perspective of the camera. Due to this perspective shift, the points on right side of the basket appears to form a triangular prism-like shape instead of the basket's actual flat shape, as shown in Figure 4.4. Due to the discrepancy between the perceived shape of the point cloud and the actual flat shape of the basket, the algorithm incorrectly considers the grasp positions behind the basket unsuitable.

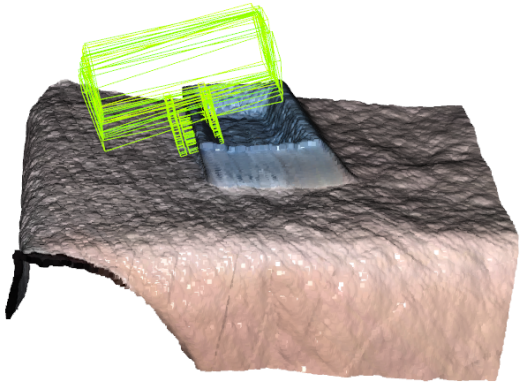
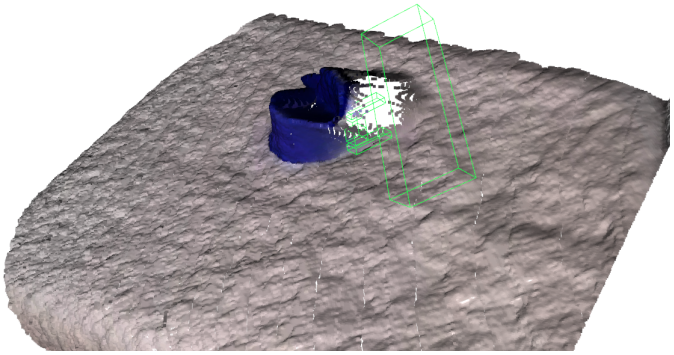
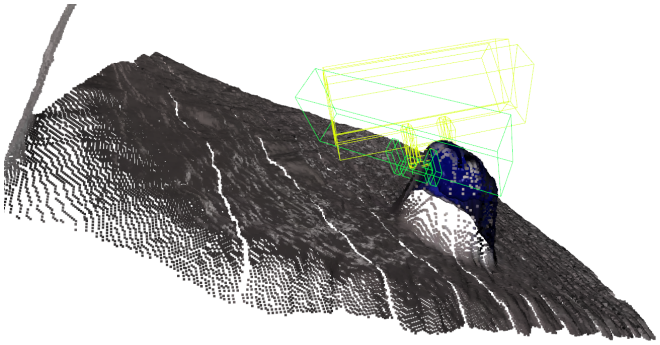


Figure 4.4: Candidate grasp poses on partial point cloud of the basket.

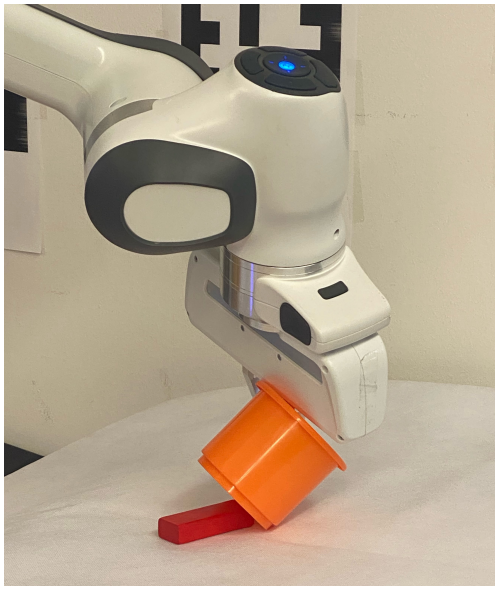
In the next part of the experiment, the performance of the algorithm was evaluated on a plastic and a metal mug. Both types of mugs are included because of the varying ability of RGB depth cameras to detect these different materials. RGB depth cameras are more accurate in detecting and capturing depth information of metal objects than plastic objects, so a more accurate grasp estimation is expected. The selection of both plastic and metal cups enabled the observation of the effect of material properties on the grasp accuracy of the grasp poses given by the algorithm. In the Figures below, best grasp scores for metal 4.7 and plastic mug 4.5. 5 experiments were planned to run for the plastic mug, but due to the partial point cloud not being complete, incorrect grasp points were detected and less trial are attempted since it is understood that the partial point cloud of the plastic mug is not suitable for searching optimal grasp poses. For example, in 4.5b, a pose was estimated where the gripper is located on the left side of the back of the point cloud of the plastic mug. This is because the points on the back of the point cloud of the plastic mugs find a flat surface that matches the surface in the partial point cloud of the algorithm. However, in real life, the plastic mug is actually a complete object and the estimated grasp pose could not be applied to the robot. This happened with various placements of the plastic mug, which caused the algorithm to estimate a failed grasp pose. In Figure 4.5a, it can be observed that the partial point cloud is not good enough to estimate realistic grasp poses. The object appears buried in the ground, making grasp pose estimation very difficult.



(a) Side of the plastic cup from the camera perspective.



(b) Back of the plastic cup from the camera perspective.



(c) Post grasp using robot in real environment.

Figure 4.5: Partial point cloud of plastic cup and candidate grasp pose.

The third object for the experimental evaluation of the grasp pose estimation algorithm is a metal cup. Since metal objects reflect light better, a metal mug was used to obtain a better and more accurate partial point cloud of the object. In Figure 4.7, it is clear that the partial point cloud obtained is much better compared to the plastic mug's point cloud. In this case, the best grasp poses are found on the handle of the cup. This is actually not related to task-oriented grasp estimation.

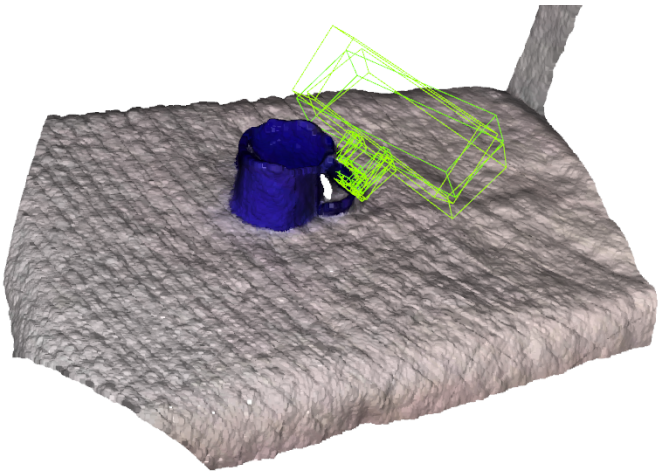
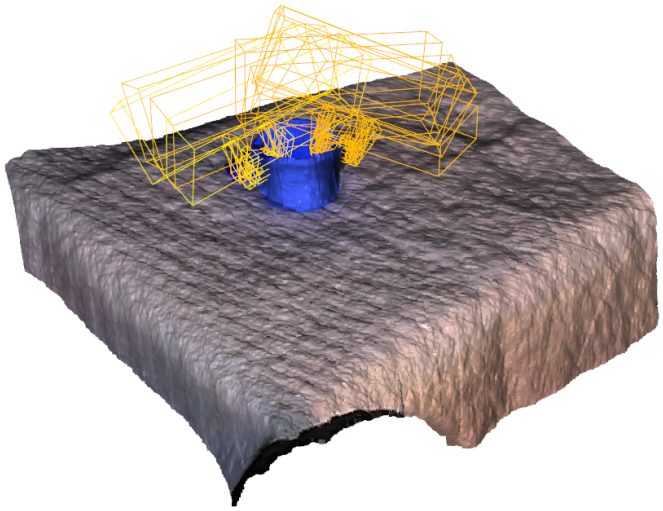


Figure 4.6: Candidate grasp pose on point cloud of metal mug.

In further experiments on a metal mug, one particular example (as depicted in Figure 4.7a) demonstrated a grasp score that fell below the desired level of quality. When these mediocre-scoring grasp poses were tried to be executed with the robot in a real environment, a successful grasp could not be achieved. This proves that mediocre scores are not sufficient for a successful grasp. This important observation highlighted the importance of prioritizing and selecting grasp poses with significantly higher scores to maximize the likelihood of successful application of grasp in real-world scenarios.

In Figure 4.7, the candidate grasp poses found by the algorithm can be observed. The result of applying the highest-scoring grasp pose is also shown in Figure 4.7. Since the robot tries to grasp the object at an incorrect angle, it is not considered sufficient for a successful grasp, even if it manages to lift the object. Because it is, in real-life scenarios, not an appropriate grasp angle for the metal mug, this observation can be taken as proof of how consistent and accurate the grasp scores are. When a pose with a low grasp score was tried to be applied, a successful robotic grasp could not be achieved. Therefore, the importance of preferring the grasp poses, represented by the green color, was demonstrated.



(a) Point cloud of the metal mug.

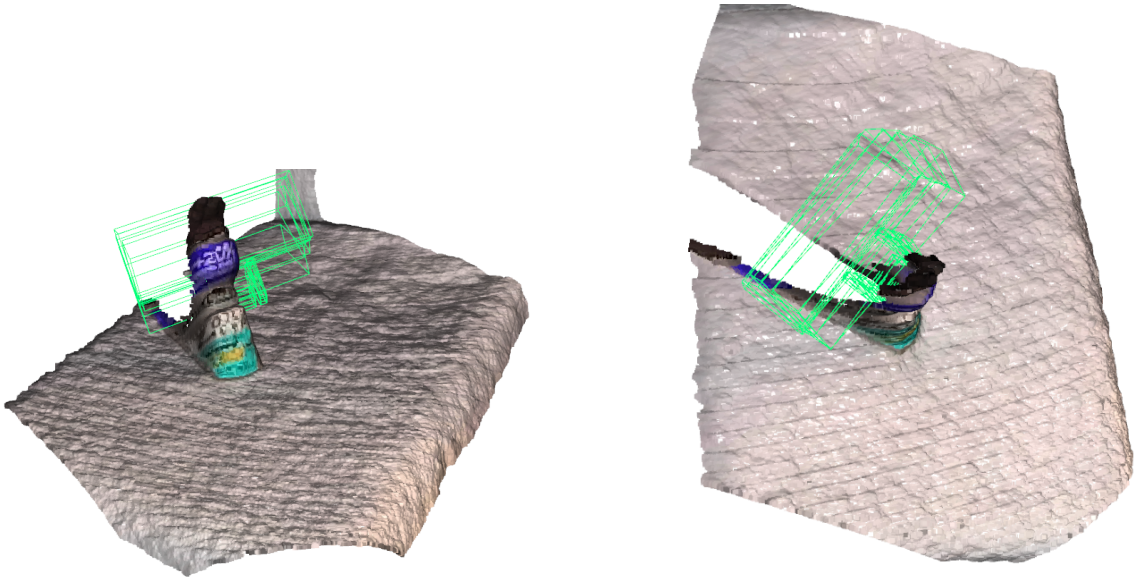
(b) Pre-grasp pose.

(c) Grasp pose.

(d) Post-grasp pose.

Figure 4.7: Execution of the pose configuration with mediocre grasp score on the metal mug.

In the continuation of the experimental evaluation on the successful grasp of objects by applying candidate grasp poses to the robot in a real environment, a shampoo bottle, jug lid, and plate were used. As with the plastic mug, the shampoo bottle could not be grasped successfully because the shampoo's partial point cloud was incomplete. The algorithm found unrealistic grasp poses with a high score. This is because, from the cameras' point of view, the back of the shampoo is not in the point cloud, and so the flat surface of the back left side (as shown in Figure below 4.8)of the shampoo's point cloud is identified as a suitable grasp surface. Since this surface of the shampoo's partial point cloud mathematically matches the gripper's flat fingers, this region was identified as a candidate grasp pose. Of course, the robot could not execute this in real life.

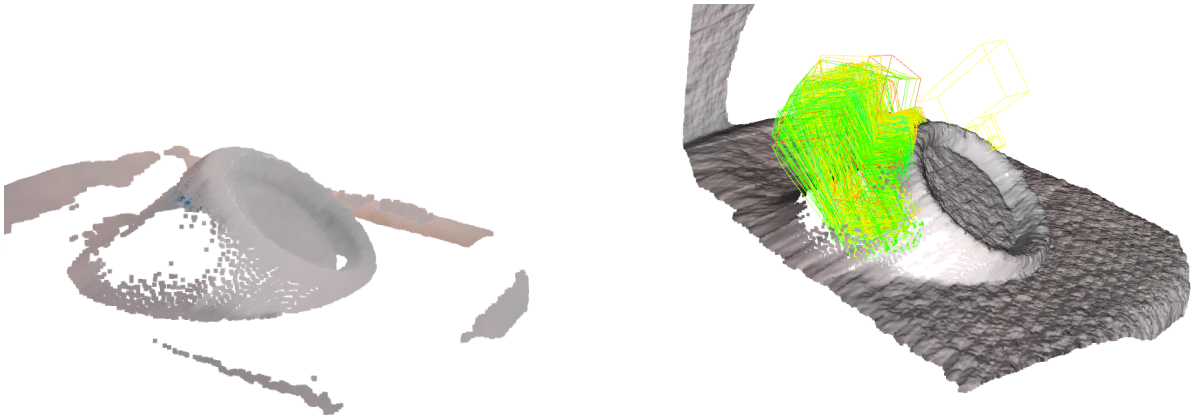


(a) Front view of the shampoo from the camera perspective.

(b) Top view of the shampoo from the camera perspective.

Figure 4.8: Candidate grasp pose on partial point cloud of shampoo.

Various candidate grasp poses are found for the experiment with the plate shown in Figure 4.9. These potential grasp configurations showed promise for the successful manipulation of the plate. However, applying these grasp poses to robots in real life was impossible. Because the grasp poses do not provide the joint constraints of the robots. To perform the grasp, the robot needs to go under the plate, but this would cause the robot to collide with the table (the experimental surface). To perform the grasp motion, the robot had to position itself under the plate. However, this maneuver posed a challenge as it risked colliding with the experimental surface, the table. Due to joint limitations in the robot's working range, it was impossible to perform the desired grasp motion without jeopardizing the robot's physical integrity, and the robot was unable to perform these poses. Here it is emphasized that joint constraints and collision avoidance measures should be carefully considered to ensure safe and successful grasp operations in real-life robotic systems. In



(a) Partial point cloud of plate.

(b) Candidate grasp poses on partial point cloud of plate.

Figure 4.9: Candidate grasp poses on partial point cloud of plate.

this experiment, the robot's limitations and joint constraints were also shown to influence the execution process. While initially, the first goal was to identify candidate grasp poses for the plate, it was later realized that the execution of these grasps in a real-world scenario is not only dependent on the properties of the object but is also greatly influenced and important by the robot's own capabilities and constraints. In particular, joint constraints and physical limitations made it difficult for the robot to move its gripper under the plate. These constraints prevented the robot from reaching the desired configuration on the plate sample without the risk of collision with the experimental surface (table).

Object	**Grasps**	**Success Rate**
Basket	10/10	100%
Plastic Mug	0/4	0%
Metal Mug	2/5	40%
Shampoo	0/2	0%
Plate	0/2	0%
Jug Lid	1/3	33%
Success Rate	**(13/26)**	**50%**

Table 4.2: Set of objects used for the experiment.

Table 4.2 shows the algorithm's results when grasping objects individually placed on a table. Figure 4.1b shows images of objects. The overall success rate for all trials on all 6 objects is 50% (13 successful grasps out of 26). In a few cases, the algorithm found feasible grasp configurations, but grasp poses could not be executed due to the joint constraints of the robot. These situations are called failed cases. These failed cases are not related to the algorithm but rather to the robot's limitations. For example, suitable grasp poses were found for the plate, but the plate was too close to the experimental surface (in this

case, the table). Since the plate was too close to the surface, motion planning for the gripper could not be found due to the robot's joint constraints or collision constraints, so the 6-DOF grasp poses could not be applied to the robot.

4.3 Experimental Evaluation on Reconstructed Point Cloud

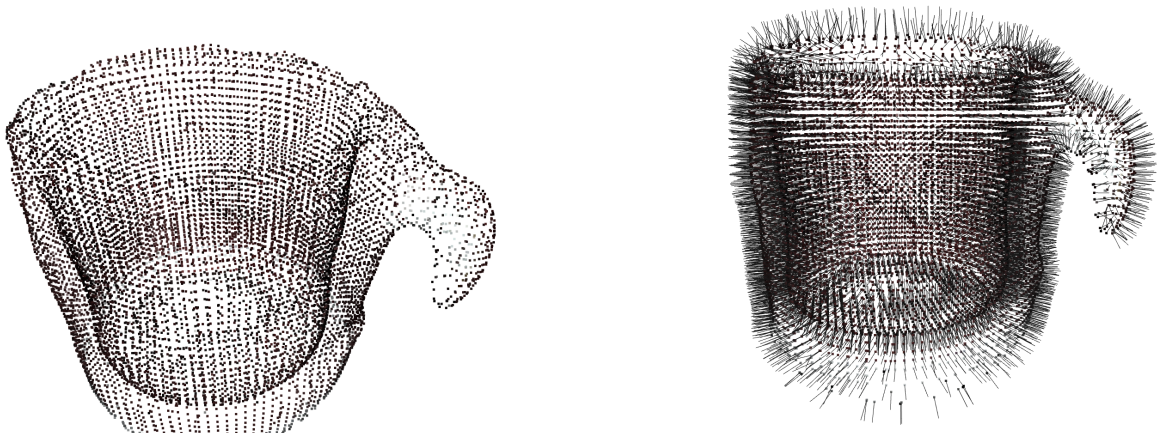
In the second phase of the experiment, the method designed in this research was evaluated on the reconstructed point clouds. The goal was to prove that, as claimed, the developed method gives much more accurate results when the point cloud is complete and precise. To achieve this goal, the developed method is evaluated on reconstructed point clouds with various object shapes, sizes, and complexities.

Reconstructed point clouds are obtained from the YCB Object and Model Set[Wan18] to evaluate the method further. This set of everyday objects of different shapes, sizes, textures, weights, and stiffnesses, and some commonly used manipulation tests, served as the basis for these rigorous evaluations. The physical version of the metal mug in this set was also used in the first part of the experiment 4.7.

In the first phase of the experiment, the effectiveness of the developed method was evaluated by performing tests on a reconstructed point cloud of the metal mug. Figure 4.14a presents a visual representation of the reconstructed partial point cloud. The reconstructed point cloud exhibits a high level of detail and completeness, allowing a comprehensive evaluation of the method's performance.

The accuracy of the surface normal estimation of the reconstructed point cloud is presented in the Figure below 4.14b. It is clear from the Figure that the surface normal orientations are extremely well estimated. This accuracy is achieved by having all surface points in the reconstructed point cloud.

On this reconstructed point cloud more grasp poses were sampled using the developed method. The experiment resulted in higher grasp scores for candidate grasp poses. The grasp scores obtained in the partial point cloud of the metal mug were mediocre as it can be seen from the Figure 4.7. However, in the reconstructed point cloud of the exact same object, the grasp scores were above 0.9 out of 1.



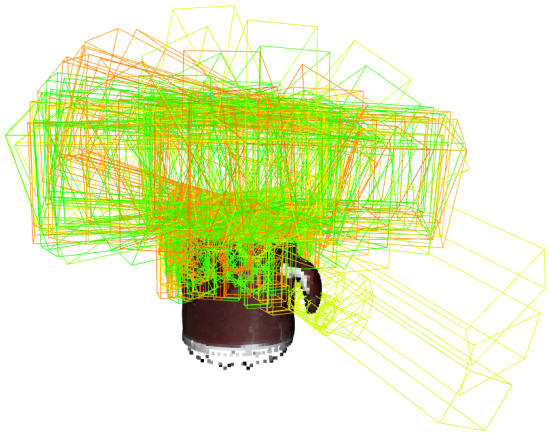
(a) Reconstructed point cloud of metal mug

(b) Normals of reconstructed point cloud of metal mug

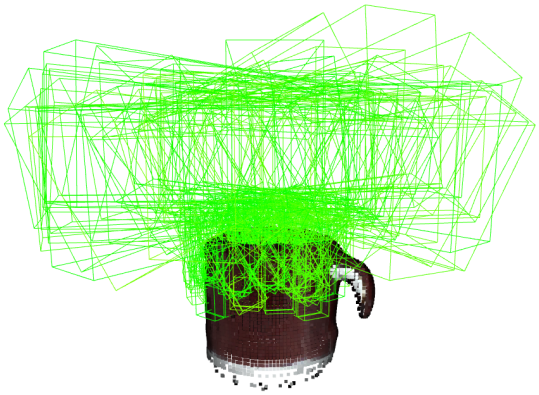
Figure 4.10: Reconstructed point cloud of metal mug

In addition, it can be seen from Figure 4.14 that all the estimated grasp poses are realistic, e.g., no poses are colliding with the body of the mug, nor are there any poses on the underside of the mug as expected. When the best pose found on the partial point cloud was given as input to the robot's motion planning, the robot could sometimes not successfully perform the grasp 4.7d. The pose found here is much higher scoring and can be successfully executed by the robot. When comparing the best pose found on the partial point cloud (Figure 4.7d) with the unsuccessful execution by the robot, it is evident that the higher-scoring pose is essential for achieving a reliable grasp. This highlights the importance of using reconstructed point clouds instead of fragmented point clouds. Furthermore, for successful grasping tasks, both the quality of the grasp and the robot's motion-planning capabilities should be considered.

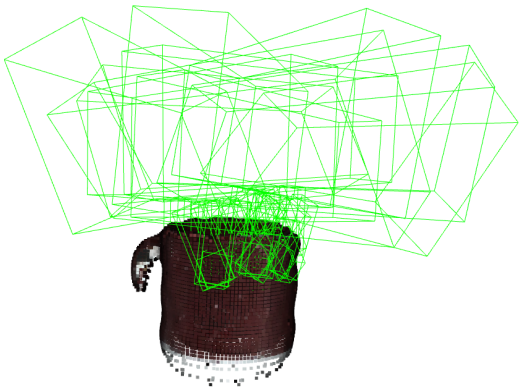
It is clear that using reconstructed point clouds instead of partial point clouds offers several advantages in the context of grasp estimation. The reconstructed point cloud of the metal mug provides a complete representation of the mug's geometry and surface, enabling more accurate grasp planning. The algorithm gains a more comprehensive understanding of the mug's shape by utilizing reconstructed point clouds, leading to improved grasp success rates. Additionally, the reconstructed point cloud enables better collision avoidance by providing a holistic view of the mug, allowing the robot to avoid potential obstacles or undesirable contact regions. Therefore, leveraging the reconstructed point cloud enhanced the overall grasp estimation process, leading to more reliable and efficient robotic grasping capabilities.



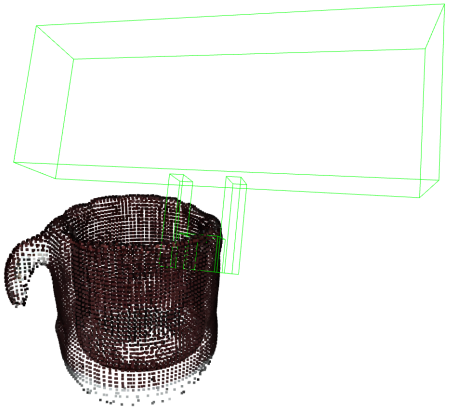
(a) All grasp poses.



(b) Top-k grasp poses.



(c) Top 10 grasp poses.



(d) Top grasp pose.

Figure 4.11: Top grasp poses on reconstructed point cloud of the metal mug.

Further experimental evaluation on the reconstructed point cloud was performed with a banana object. Since the banana has a very different shape compared to the mug, it is an excellent object to observe the versatility of the grasp pose sampling method. The Figure below 4.12 shows the estimated feasible grasp pose on the reconstructed point cloud of the banana.

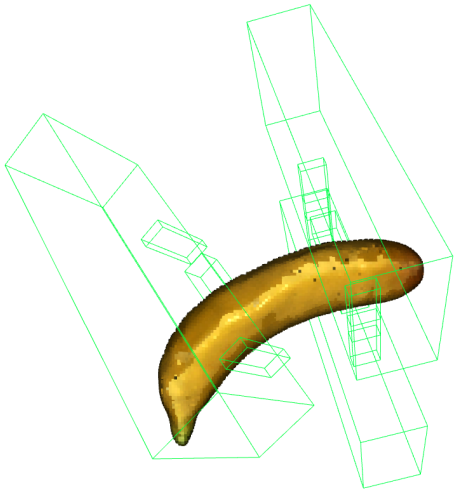


Figure 4.12: Top grasp poses on reconstructed point cloud of the banana.

Another experiment was done for a cracker box, which was larger than a banana and needed to be grasped with wider gripper fingers. The presented method again found feasible grasp poses for the robot. Since there is no surface (floor) in this experiment, the method found the pose under the cracker box. Thus, the method demonstrated its adaptability to different object configurations.

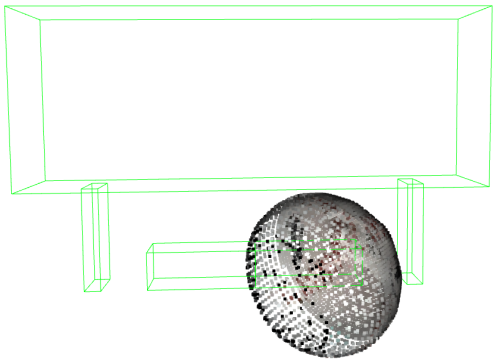
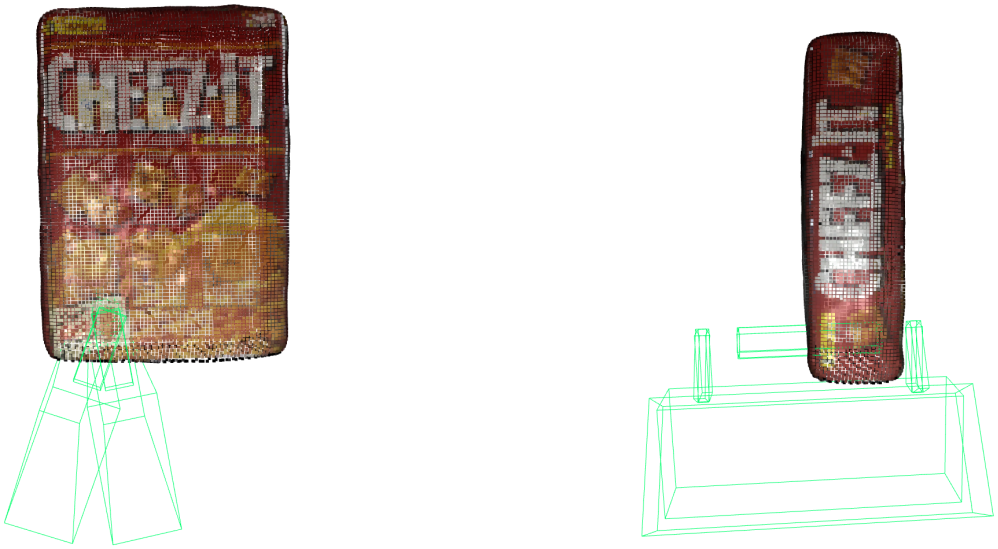


Figure 4.13: Reconstructed point cloud of baseball and estimated top grasp pose.



(a) Top 2 grasp poses.

(b) Top 2 grasp pose from the side view.

Figure 4.14: Top grasp poses on reconstructed point cloud of the craker box.

The ability of the method to find realistic and feasible grasp poses for objects with different shapes, such as baseballs, cracker boxes, and bananas, is remarkable. A common feature observed in these grasp poses is that they have a wider grasp compared to the previous ones. The algorithm 1 positions the center point of the candidate grasp pose at the intersection of the bounding boxes of the grasping fingers on the point clouds. The gripper must therefore spread his fingers wide to perform the grasp pose and grasp large objects. For example, in the baseball example (Figure 4.13), the center of the candidate grasp pose is located on the middle left of the ball. As a result, by increasing the width of the grasp, the right side of the grasp covers the entire ball. However, this situation is not a problem for the method since it is still executable for the gripper. This demonstrates the effectiveness of the proposed method in generating feasible grasp solutions for challenging objects like baseball.

The results obtained from the experiment were highly promising and showed the remarkable capabilities of the developed method. The method showed a remarkable level of accuracy in estimating grasp poses with the completeness of the point cloud.

Moreover, the experimental findings demonstrated the robustness and adaptability of the developed method, highlighting its ability to handle various object shapes and sizes with precision and reliability. The method's superior performance in scenarios where the point cloud is complete and accurate underlines its effectiveness in exploiting the full potential of rich and accurate geometric information for grasp estimation.

4.4 Discussion

Overall, the results suggest that the presented algorithm is very promising. Although a success rate of 50% was obtained over different trials on 6 different objects, featuring an extensive variety of shapes and appearances for lifting individual objects, a success rate of 100% for the wicker basket was obtained. The wicker basket had the best and most accurate partial point cloud. This show that the algorithm works perfectly fine if the point cloud is sufficiently accurate. This result is remarkable, considering the system had no model or prior knowledge of the objects. Additionally, no training data was required, and no learning was involved in obtaining these results. The results were insufficient for other grasp trials, like plastic mugs or shampoo. This is mainly because of the lack of completeness of the point clouds captured by the RGB depth camera. The algorithm finds the highest-ranked grasp poses depending on the grasp score in the empty parts of the objects. This is because empty parts seem like flat surfaces, perfectly matching the gripper's fingers. These situations are observable in Figures 4.5 and 4.8. However, in real-life scenarios, those poses were not executable since those parts were actually not appropriate for grasping. Another reason for the unsuccessful grasp pose estimation was the high noise in partial point clouds. An example is shifted points in an object's partial point clouds. Shifted points cause the algorithm not to detect grasp poses that are suitable for a stable and secure grasp. These cases are observed in experiments with metal mug. The shifted points disrupt the flat surface to be observable in a partial point cloud, and the algorithm could not find appropriate grasp poses.

On the other hand, these shifted points, and the results obtained on incomplete partial point clouds serve as valuable indicators of the capabilities and limitations of the represented grasp pose estimation algorithm. In the challenges faced, the algorithm demonstrated the ability to detect and identify flat surfaces matching those of parallel-jaw gripper fingers, highlighting its potential to estimate grasp poses for objects with appropriate surface features accurately. Ultimately, the algorithm can successfully examine the object surface and gripper finger's similarities using surface normals. By that, the algorithm is expected to work perfectly on complete point clouds.

It would have been efficient to obtain a segmented point cloud to eliminate the problem of missing points on the point clouds in the experiments, but this was not feasible due to the time constraints of the thesis. The robot workspace containing the test objects could have been observed by moving a wrist-mounted depth camera to different positions for each grasp trial, thus obtaining a segmented point cloud. After segmenting the ground plane, the resulting point could be used to test the proposed method for generating grasp hypotheses more accurately.

Another noteworthy approach is to use the reconstructed point clouds as input for the grasp pose estimation algorithm. As previously described in Section 4.3, the algorithm is able to find a more significant number of high-scoring grasps when applied to reconstructed point clouds. This observation further strengthens the algorithm's effectiveness in handling

complete point cloud data. It proves its robustness and proficiency in accurately identifying suitable grasp poses.

However, it can improve robustness in various ways. It has been observed that the set of ten highest-ranked grasps sometimes contains a grasp pose that performs better than the highest-ranked grasp. This is because the algorithm only selects grasps based on the geometry of the surfaces. Combining the algorithm's robust selection of graspable geometric features with other types of information, such as joint constraints of the robot, could lead to more robust performance.

The experiment result showed that an effective grasp planning algorithm should not only evaluate the grasp quality and feasibility based on the object's characteristics but also consider the robot's constraints and limitations. Those environmental factors can significantly affect the accuracy and reliability of grasp pose estimation. This was most notably observed in the jug lid experiment. Accumulated errors such as shifts in the calibration of the camera, the quality of the camera's depth image acquisition and the robot's execution cause the grasp poses found by the algorithm to fail to execute in real-life scenarios. Despite efforts to calibrate the robot and minimize errors, it is essential to recognize that inherent imperfections in the robot implementation, camera measurement variations, and environmental uncertainties can lead to accumulated errors throughout the grasping process.

The second part of the experimental evaluation revealed a possible substantial advancement over the proposed techniques. The experiment's findings were quite encouraging and demonstrated the created method's outstanding qualities. The method demonstrated a surprising level of accuracy in determining grasp positions by making use of the completeness and accuracy of the point cloud.

The experimental results also showed the robustness and adaptability of the created approach, emphasizing its capacity to reliably and precisely manage a range of object forms and sizes. The method is more effective at utilizing the full potential of rich and accurate geometric information for grasp estimate, as seen by its higher performance in situations where the point cloud is complete and accurate.

In summary, this experiment provided a valuable reminder that the success of grasping poses depends not only on the object's properties but also on the limitations and joint constraints of the robot. Recognizing and addressing these constraints in grasp planning algorithms is crucial to enable robots to effectively and safely perform grasping actions in real-world environments. The algorithm's methodology can be further developed to address this issue by recognizing and understanding these limitations. Future iterations of the algorithm could include reconstructing the point clouds to avoid point shifts due to the camera's perspective. That method can accurately assess the appropriateness of grasp positions on objects with complex shapes and perceptual variations.

The algorithm has proven its ability to detect surface similarities between objects and gripper finger surfaces. It is also proven that by detecting surface similarities, given grasp scores

are accurate. This means that in an ideal environmental setup with reconstructed point clouds, the presented algorithm will provide and estimate feasible candidate grasp poses for stable and secure grasps. This underscores the algorithm's effectiveness in leveraging surface information to generate practical and reliable grasping solutions.

Chapter 5

Conclusion

In this research, an innovative grasp estimation metric based on surface normals was developed. The goal was to address the challenge of accurately estimating and evaluating the 6 degrees of freedom (DOF) grasp poses. To achieve this, a metric was defined that evaluates the similarities between the shape of finger surfaces and the local shape of an object observed through a partial point cloud. By comparing surface normals, which contain valuable information about the local surface of both objects and gripper fingers, a robust grasp score metric was defined, which serves as a quantitative measure of the appropriateness and effectiveness of a given grasp pose. The approach stands out in particular because it is independent of the physical parameters of objects, requires no prior knowledge or training data, and relies solely on comparing local surface normals.

A partial point cloud was captured from an RGB depth camera to estimate candidate grasp poses. The generalizability and effectiveness of the introduced method have been rigorously assessed through extensive experimental evaluations involving a real robot and various objects. Provided the point cloud data is accurate enough, the results demonstrate the method's remarkable ability to generalize and fit well to a variety of object shapes. Experimental analyses underline the significance of accurate point cloud acquisition for robust grasp pose estimation, emphasizing that the accuracy of the point cloud data plays a vital role in determining secure and stable grasp poses.

A notable advantage of the algorithm is its ability to identify parts of the point cloud that should not be grasped, such as flat table surfaces. This additional functionality adds to the overall versatility and reliability of the grasp estimation process, increasing its real-world applicability. Additionally, the algorithm's ability to distinguish non-graspable areas from graspable ones contributes to the overall safety of the grasping application by preventing possible damage or accidents during robotic manipulation tasks.

The research also highlights the critical importance of considering the robot's inherent constraints and limitations when developing an adequate grasp pose estimation method. It revealed that accumulated errors, including deviations in the camera's calibration, the

quality of depth image acquisition, and the robot's implementation, can undermine the successful implementation of algorithm-determined grasp poses in practical scenarios. Consequently, a comprehensive grasp planning system should account for and mitigate these potential sources of error and ensure accurate and reliable execution.

In addition to its superiority over traditional approaches, presented grasp pose estimation method offers better computational efficiency than data-driven methods. Unlike data-driven approaches that often require significant computational resources to train complex models on large datasets, the presented method eliminates the need for extensive training. By relying on surface normals and exploiting similarities between object and finger surfaces, the approach provides a computationally efficient solution for grasp planning, emphasizing the importance of considering the point cloud's completeness. This research represents an essential step in developing robust and adaptive grasp pose estimation for real-world applications.

Chapter 6

Future Work

As a future work, two main improvements can be made for a more robust and more useful grasp pose estimation method based on surface normals: Working on reconstructed point clouds instead of partial point clouds and task-oriented grasping.

The RGB depth camera captures partial point clouds, which can then be processed and reconstructed to represent the whole shape of objects as a point cloud. The experimental evaluation in the second part of the research proved that the introduced method performs excellent when applied to reconstructed point clouds. The evaluation findings demonstrated the method's remarkable efficiency and accuracy in processing reconstructed point clouds. By exploiting the completeness and accuracy of the reconstructed data, the method produced precise and reliable candidate grasp poses. This shows that the presented method is well suited for applications with reconstructed point clouds, as it can effectively utilize the rich geometric information in these representations.

Another possible way to further develop the method could be adding task-oriented grasping to the introduced method. Task-oriented grasping involves considering not only the geometric properties of the object but also the specific task or goal associated with the grasp. By considering the goal of grasping, such as picking up an object to place it in a particular place or manipulating it in a certain way, the method could generate grasp poses optimized for the given task. By incorporating task-oriented grasping, the method can improve its applicability and utility in real-world scenarios where specific tasks or objectives need to be accomplished through robotic grasping. This would enable the method to provide more detailed and task-specific grasp solutions, improving robotic manipulation tasks' overall performance and efficiency.

List of Figures

List of Tables

List of Algorithms

Bibliography

[AMO+18] Maxime Adjigble, Naresh Marturi, Valerio Ortenzi, Vijaykumar Rajasekaran, Peter Corke, and Rustam Stolkin. Model-free and learning-free grasping by local contact moment matching. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2933–2940. IEEE, 2018.

[BK00] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 1, pages 348–353. IEEE, 2000.

[BLAL16] Joao Bimbo, Shan Luo, Kaspar Althoefer, and Hongbin Liu. In-hand object pose estimation using covariance-based tactile to geometry matching. volume 1, pages 570–577. IEEE, 2016.

[BLE08] Gary M Bone, Andrew Lambert, and Mark Edwards. Automated modeling and robotic grasping of unknown three-dimensional objects. In *2008 IEEE International Conference on Robotics and Automation*, pages 292–298. IEEE, 2008.

[BMAK13] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. volume 30, pages 289–309. IEEE, 2013.

[CRT04] Ulrich Clarenz, Martin Rumpf, and Alexandru Telea. Robust feature detection and local classification for surfaces based on moment analysis. volume 10, pages 516–524. IEEE, 2004.

[Der10] Konstantinos G Derpanis. Overview of the ransac algorithm. volume 4, pages 2–3, 2010.

[DPM17] Renaud Detry, Jeremie Papon, and Larry Matthies. Task-oriented grasping with semantic and geometric scene understanding. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3266–3273. IEEE, 2017.

[DWLZ21] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. volume 54, pages 1677–1734. Springer, 2021.

[FV12] David Fischinger and Markus Vincze. Empty the basket-a shape based learning approach for grasping piles of unknown objects. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2051–2057. IEEE, 2012.

[FZG+20] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. volume 39, pages 202–216. SAGE Publications Sage UK: London, England, 2020.

[HKH+14] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Experimental robotics: The 12th international symposium on experimental robotics*, pages 477–491. Springer, 2014.

[HRD+12] Stefan Holzer, Radu Bogdan Rusu, Michael Dixon, Suat Gedikli, and Nassir Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2684–2689. IEEE, 2012.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[JMS11] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International conference on robotics and automation*, pages 3304–3311. IEEE, 2011.

[KAWB09] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *2009 IEEE International Conference on Robotics and Automation*, pages 3206–3211, 2009.

[KCF11] Michael Krainin, Brian Curless, and Dieter Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *2011 IEEE international conference on robotics and automation*, pages 5031–5037. IEEE, 2011.

[KK17] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776. IEEE, 2017.

[KKB20] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets. volume 5, pages 3352–3359. IEEE, 2020.

[KRC+11] Ellen Klingbeil, Deepak Rao, Blake Carpenter, Varun Ganapathi, Andrew Y Ng, and Oussama Khatib. Grasping with application to an autonomous checkout robot. In *2011 IEEE international conference on robotics and automation*, pages 2837–2844. IEEE, 2011.

[LC14] Feng Lu and Ziqiang Chen. A general homogeneous matrix formulation to 3d rotation geometric transformations. 2014.

[LUD+10] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, et al. Opengrasp: a toolkit for robot grasping simulation. In *Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference, SIMPAR 2010, Darmstadt, Germany, November 15-18, 2010. Proceedings 2*, pages 109–120. Springer, 2010.

[MA04] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. volume 11, pages 110–122, 2004.

[MCL18] Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. 2018.

[MLM+21] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on Robot Learning*, pages 1540–1557. PMLR, 2021.

[MLN+17] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. 2017.

[MN03] Niloy J Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328, 2003.

[MPH+16] Jeffrey Mahler, Florian T. Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In

2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1957–1964, 2016.

[MPK+15] Jeffrey Mahler, Sachin Patil, Ben Kehoe, Jur Van Den Berg, Matei Ciocarlie, Pieter Abbeel, and Ken Goldberg. Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4919–4926. IEEE, 2015.

[PMAJ04] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3512–3518 Vol.4, 2004.

[QSMG17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[RA15] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[RC11] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[SDT+22] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[Shi18] Jian Shi. An introduction towards 3D Computer Vision. https://medium.com/@jianshi_94445/an-introduction-towards-3d-computer-vision-71be8ce11956, 2018.

[SHKK10] Dan Song, Kai Huebner, Ville Kyrki, and Danica Kragic. Learning task constraints for robot grasping using graphical models. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1579–1585. IEEE, 2010.

[SHL+13] John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: science and systems*, volume 9, pages 1–10. Berlin, Germany, 2013.

[SWN08] Ashutosh Saxena, Lawson LS Wong, and Andrew Y Ng. Learning grasp strategies with partial shape information. In *AAAI*, volume 3, pages 1491–1494, 2008.

[Tau91] Gabriel Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. volume 13, pages 1115–1138. Citeseer, 1991.

[TPGSP17] Andreas Ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. volume 36, pages 1455–1473. SAGE Publications Sage UK: London, England, 2017.

[TPP18] Andreas Ten Pas and Robert Platt. Using geometry to detect grasp poses in 3d point clouds. pages 307–324. Springer, 2018.

[UVDSGS13] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. volume 104, pages 154–171. Springer, 2013.

[Wan18] Chenxi Wang. ycb-scripts. https://github.com/chenxi-wang/ycb-scripts, 2018.

[WYPS22] Hongtao Wen, Jianhang Yan, Wanli Peng, and Yi Sun. Transgrasp: Grasp pose estimation of a category of objects by transferring grasps from only one labeled instance. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 445–461. Springer, 2022.

[Zho21] Zhou, Qian-Yi and Miller, Henry P. and Shi, Jianchao and Zhang, Chenlong and Li, Zhuwen and Zhang, Yihan and Chen, Hui and Chao, Yutong and Huang, Qixing and Zhang, Shurui and Xu, Youquan and Xu, Kui. Open3D: A modern library for 3d data processing. http://www.open3d.org, 2021. Accessed on June 20, 2023.