ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

RSL Robotic Systems Lab
Prof. Dr. Marco Hutter

Semester Thesis

Zero-Shot Language-Conditioned Mobile Manipulation via VLMs

Spring Term 2024

Supervised by:
Kaixian Qu
Carmen Scheidemann

Author:
Luca Franceschetti

Declaration of Originality

I hereby submit the written work entitled:

Your Project Title

- ■ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it[1]. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[2].
- □ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it[1]. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies[2].
- □ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it[1]. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies[2]. In consultation with the supervisor, I did not cite them.

Author(s)

Luca	Franceschetti

Student supervisor(s)

Kaixian	Qu
Carmen	Scheidemann

Supervising lecturer

Marco	Hutter

With the signature, I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on The citation conventions usual to the discipline in question here have been respected. I have documented all methods, data, and processes truthfully and fully. Additionally, I have mentioned all persons who were significant facilitators of the work.

The above-written work may be tested electronically for plagiarism.

Zurich 11.09.2024

Place and date

Signature

[1]Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

[2]E.g. ChatGPT, DALL-E 2, Google Bard

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zurich 11.09.2024

Place and date

Signature

Contents

Preface

After completing my Bachelor's degree in physics, I wanted to pursue robotics as it allows to apply my mathematical knowledge in a more practical context. This is why is changed my field by switching to a master's program in Computational Engineering and Science (CSE) with a specialization in robotics. During my studies, I gained a understanding of the theoretical aspects of robotics. However, this project marked my first important practical experience of heading a large practical project in this field and it has been a great value for me. I was able to get an understanding of major frameworks such as ROS and common practices employed in this field.

I would like to express my gratitude to Prof. Dr. Marco Hutter for giving me the opportunity to work on this project at the Robot Systems Lab. I am also very grateful for the assistance of Kaixian Qu and Carmen Scheidemann during the entire project. Their expertise and assistance were crucial in helping me navigate this new area of work.

Zurich, August 2024
Luca Franceschetti

Abstract

Including semantic information and applying zero-shot planning allows robots to adapt to new environments dynamically. Previous approaches for generating grasp positions from an image and a task description need the desired object to be specified in the task description, and distinct parts must be identifiable. To address this shortcoming, the proposed pipeline utilizes the multimodal foundation model Chat-GPT4o to choose between a set of grasps checked for stability and feasibility beforehand. This way, the whole reasoning capability of the Vision-Language Model (VLM) can be taken into account. The system shows a reasonable performance with an overall success rate of 69.4% on 13 scenes with 17 task descriptions designed to test for open vocabulary, task-oriented grasping, and scene understanding. While promising, the pipeline shows inconsistency due to a geometric bias and the VLM occasionally misinterpreting the scene.

Symbols

Symbols

$\hat{n}$	Surface normal vector
T_{sc}	Transformation matrix camera-to-scene
t_{sc}	The translation vector camera-to scene
T_{ws}	Transformation matrix world-to-scene
T_{wc}	Transformation matrix world-to-camera
z_{scene}	z-axis in the scene coordinate frame

Acronyms and Abbreviations

FPS	Farthest Point Sampling
SR	Success rate
LLM	Large Language Model
VLM	Vision-Language Model

Chapter 1

Introduction

A typical household contains a wide range of object categories and even more possible instances of one object's appearance. Arranging multiple objects gives a countless number of different possible scenes. Furthermore, how these objects need to be picked up might vary depending on the task at hand. Therefore, having an effective robot that can interact with such a dynamic environment needs an enormous training effort.

This challenge is addressed by zero-shot grasping. A robot should be able to determine how to grasp an object without any prior training on that specific object or task. Traditionally, grasping systems focused only on the geometric properties of objects to ensure grasp stability. Understanding more about the scene, such as the purpose of the objects, its properties and the context of the task can improve the quality of the grasp. This information must be incorporated somehow into the grasp generation to allow for semantic grasping.

Recent VLMs like OWL-ViT [1] and CLIP [2] have significantly contributed to this field by providing the ability to associate visual and textual inputs. This allows the integration of semantic knowledge into the scene. Furthermore, the size of their training data allows for zero-shot object recognition and task reasoning. Building on these advanced VLMs, recent models like ChatGPT-4o incorporate the information from images directly into LLMs. This gives the model the ability to reason about the visual input and put it in conjunction with the textual description.

This project aims to leverage these capabilities to develop a system for zero-shot grasping in an open-vocabulary manner. The robot should be able to choose not only between different objects in a scene but also to identify the best grasping location on a single object based on a task description. The approach starts with analyzing the geometric properties of objects to find stable grasps. Then, the VLM uses semantic reasoning to refine these options, selecting the optimal grasp for the task. This combination of geometric analysis and semantic reasoning results in grasp proposals that take both the entire visual information of the scene as well as the given task description into account.

Chapter 2

Related Work

The recent advances in LLMs and VLMs have made it possible to go beyond grasping based on geometric considerations alone. Now grasping algorithms can incorporate additional context by understanding both the object and the task.

Model-based approaches integrate visual and textual input into a network to determine the quality of grasping. Tang et al. [3] use two CLIP-based encoders in their system to predict grasping positions from an RGB image and a task description. The method allows for generalization when the scene is modified with the same objects, other object instances are chosen, or unknown category-task combinations are tested. Similarly, in GraspGPT [4] they use an LLM to create a description of the object and task, which is then encoded for input to a network. This approach also generalizes well, as it assumes that objects with similar descriptions have a similar optimal grasp position. With FoundationGrasp Tang et al. [5] build on both methods and use both a VLM and an LLM for similarity comparison. In this way, they ultimately outperform GraspGPT.

Zero-shot grasping algorithms often use a part-based approach. Li et al. [6] build on the idea of letting an LLM reason about the geometric structure of the object. In ShapeGrasp, the LLM names the different parts and then directly makes a decision about which grasp is best suited for the task. This is possible by first breaking the object down into simple geometric shapes. In this way, the LLM can use visual information by drawing conclusions based on this decomposition.

The usage of VLMs allows the integration of visual information without first abstracting it. Rashid et al. [7] use CLIP to embed semantic information into neural radiation fields. This makes it possible to localize specific objects or parts that need to be explicitly named in the image. The optimal grasp is then determined by combining this with the result of a traditional grasp planner. Similarly, Mirjalili et al. [8] utilize OWL-ViT to find the part in the image and generates grasps based on the identified part. The main difference is that this proposed method, LAN-Grasp, uses an LLM beforehand to select the relevant part. Van Oort et al. [9] use open vocabulary detectors to segment and localize the object and part for grasping.

In all of these approaches either the object or even object and specific part, have to be given and named in the task description. Furthermore, the objects need to be destructable into distinct parts. The approach taken in this project addresses these shortcomings.

Chapter 3

Methods

In this chapter, the pipeline used to generate a grasp proposal based on the visual input of the scene and a given task description is outlined.

3.1 System Overview

The approach taken is to enhance the image with visual cues to ground the VLM's scene understanding, in order to arrive at a specific grasp selection. The overall grasp proposal pipeline is illustrated in Figure 3.1. During this process the point cloud extracted from the depth image is preprocessed in a way that it can be utilized within the system. From this, a number of potential grasp proposals are sampled. Each candidate grasp is evaluated based on two key factors: stability and feasibility. Among those grasps, only the ones satisfying these conditions are displayed on the RGB image as gripper wireframe showing their position and orientation towards the object.
This enhanced image, combined with the task description, is then passed to the VLM. The VLM interprets the scene, reasons about the task, and selects the best grasp proposal to achieve the specified goal. This process ensures that geometric, semantic and task-specific information are taken into account when determining the final grasp.

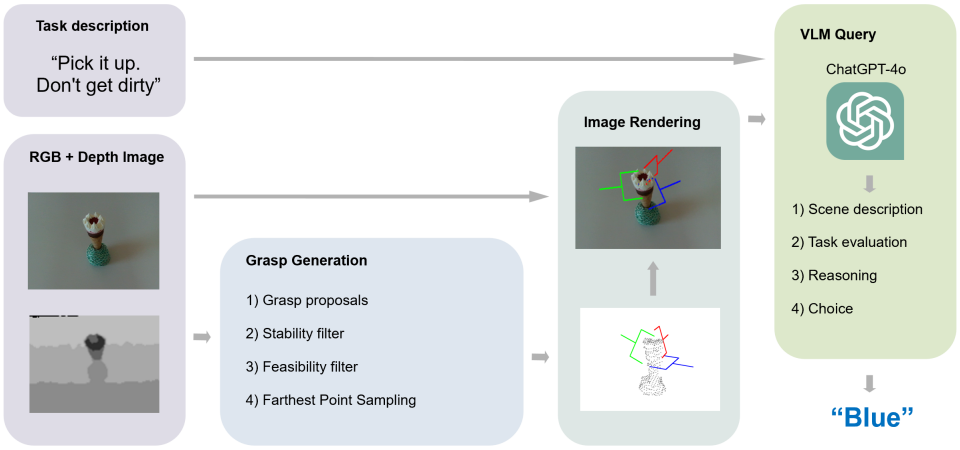


Figure 3.1: Pipeline overview for a generating grasp proposal based on visual input and task description

3.2 Grasp Generation

3.2.1 Pointcloud Preprocessing and Grasp Sampling

The point cloud transformation process begins by filtering the input point cloud based on distance. This way, points beyond a specified threshold are excluded to focus on relevant areas within the robot's workspace. Following this, RANSAC plane segmentation is applied to distinguish the dominant plane from the objects in the scene. This way the objects can be isolated from the table for the grasp generation procedure. Additionally, the normal vector of this plane is computed, which provides essential orientation data for subsequent tasks. Once the plane is removed, DBSCAN clustering is used to identify and segment distinct object clusters from the remaining points. Afterward, the point cloud is downsampled using voxel grid filtering to reduce the computational load. In the end, statistical outlier removal is performed to improve the object structure.

Grasp proposals are derived by the process of estimating surface normal from the obtained object point cloud. Farthest Point Sampling (FPS) is utilized to select the grasping points on the object, ensuring an even distribution of grasp candidates. Possible approach vectors are computed and filtered based on angle, distance, and depth conditions. Grasps approaching from the bottom near the table are also filtered out at this stage to avoid impossible grasps. This is the first low-level assurance to have collision-free candidates. Comparing the approach vector with surface normals keeps only grasps that align properly with the object's surface.

3.2.2 Stability Filter

The first evaluation that the grasp candidates are subjected to is the edge grasp model to identify their stability. This pretrained neural network assesses stability of the grasp by analyzing the geometric structure of the point cloud. In the edge grasp model [10] each of the grasps is represented by an approach point and a contact point. These edge features are processed through several layers of the network, to predict the probability of a successful grasp.

Edge grasp is particularly effective because it has been trained on grasp attempts from single-viewpoint scenes. This makes it especially suitable to be used in real world and scenarios where the robot has a very limited vision of the environment, which is the case of this pipeline. The model consistently achieves high grasp success rates, outperforming other methods like VGN and GIGA. It is, however, relevant to point out that other grasp evaluators could also be used on this step.

A predefined stability threshold is applied to filter out grasps that have a low score. This ensures only grasps continue in the pipeline that could effectively pick up the object.

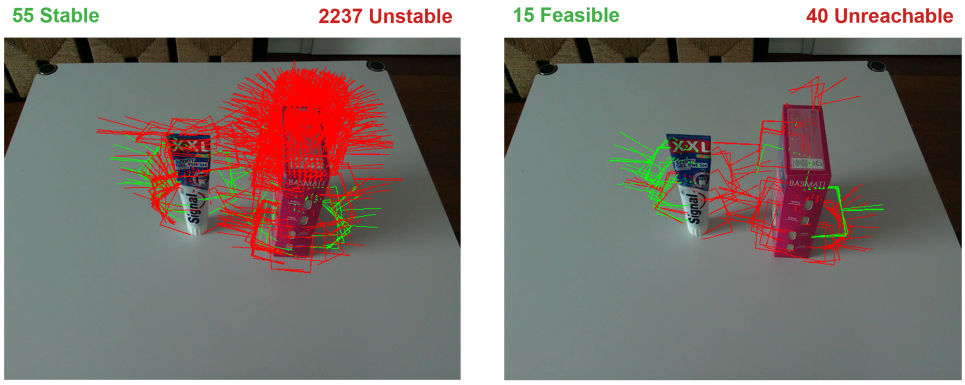


Figure 3.2: Visualization of the grasp proposal filtering process for a single scene. Grasps displayed in green are accepted, while those in red are discarded. On the left, the stability filter eliminates grasps with an edge grasp score below a threshold of 0.9. On the right, the feasibility filter removes unreachable grasps, either blocked by objects or too close to the table.

3.2.3 Feasibility Filter and Final Choice

The next step in the grasping process is a feasibility test, conducted by simulating the robot in Gazebo and representing the scene as an octomap. The MoveIt path planner is used to evaluate whether the grasp candidates are within the robot's reach.
Since the camera image lacks a predefined camera-to-world transformation matrix T_{wc}, we must define one to properly align the camera frame with the scene and ensure the robot can effectively interact with the objects in the simulation.
By determining the camera-to-scene transformation matrix T_{sc} from the image, we can position the scene anywhere in the simulation by defining T_{ws} (world-to-scene transformation) and computing:

$$T_{wc} = T_{ws} \cdot T_{sc} \tag{3.1}$$

The rotation R in the transformation matrix is calculated by aligning the surface normal vector $\hat{n}$ from the camera's viewpoint with the scene's z-axis z_{scene}. The translation vector t_{sc} is derived from the center of the point cloud. The complete camera-to-scene transformation matrix is:

$$T_{sc} = \begin{bmatrix} R & t_{sc} \\ 0 & 1 \end{bmatrix} \tag{3.2}$$

When the scene is in the form of an octomap in the simulation, the MoveIt path planner applies a Rapidly-exploring Random Tree algorithm, as it is proven to be an effective and efficient way of finding valid paths. Every grasp candidate is brought to a test by simulating its plan towards the corresponding pose. Grasps that cannot be reached are discarded.

Rendering all remaining grasps into the image could lead to clutter and confusion for the VLM. To prevent this, the number of grasps is reduced. FPS is used for select grasps which are spread over the various parts of the object so as to get proper coverage and variety.

3.3 Image Rendering

The grasp image renderer is responsible for overlaying grasp proposals onto the RGB-camera images. This step creates a clear visualization of potential grasp locations as later input for the VLM.

3.3.1 Wireframe Creation

First, a digital model of the real Franka Panda gripper is created using precise 3D meshes of the base and fingers, closely mirroring the real gripper. Key geometric locations, including the fingertips and base, are identified as control points that define the gripper's structure. A wireframe model of the gripper is then created by connecting these control points by lines. Along these lines, equally spaced points are generated. So, the wireframe is represented as a point cloud. In the end, the point cloud is uniformly colored to distinguish each grasp proposal from the others visually.

The transformation matrices for each grasp are computed based on the gripper's approach, surface normal vectors, and position in space. These matrices are then applied to properly position and orient the wireframe model within the 3D environment. This way the final wireframes are accurately reflecting how the gripper would interact with the object during grasping.

3.3.2 Mask Creation and Backprojection

By projecting the object's 3D points onto the 2D image plane using the camera's intrinsic matrix, a mask is generated to highlight the object's surface.This is done with an aim of handling occlusions between the gripper and the object in a manner that the VLM will accurately infer the position and orientation of the grasp in the future. Occlusions from other parts of the scene, like the table, are disregarded to ensure that enough of the wireframe is visible to the VLM.

In the next step, each point from the grasp wireframe is backprojected onto the RGB image by calculating the 2D pixel coordinates for each 3D point using the camera's intrinsic matrix. The depth value for each point is compared to the corresponding pixel in the depth image. The grasp point should only be shown in the image if it either closer to the camera or outside the object mask. If this condition is met, both the depth and RGB images are updated. The depth image stores the new depth value to manage overlapping grasps. The RGB image is updated with the assigned grasp color. By also adjusting neighboring pixels the thickness of the wireframe lines in the image can be defined.

3.4 VLM Query

The final step in the grasp planning pipeline is to use the rendered RGB image and the given task description to query the VLM. It should reason about the grasp proposals and choose the most suitable one for the task. In this case, GPT-4o is used to analyze both the visual and textual inputs. Any other VLM capable of processing images and interpreting complex task instructions would also be suitable for this step.

First, the VLM is given some context, letting it know that it should argue from the perspective of a robot arm. The rest of the prompt is structured into four steps to guide the model's reasoning process.

1. **Scene description:**
   First, the VLM is tasked with observing the scene. It should describe the objects present and their spatial arrangement relative to the gripper. This step adds the semantic information to the pipeline, since here all the objects and parts are named. The highlighted grasp candidates are not of importance here.

2. **Task evaluation:**
   Second, the model evaluates the task instructions and assesses how the given goal would be generally approached. This phase gives reasons for which object should be grasped and which part of that object is most suitable for fulfilling the task. It should take the object's geometry and orientation into consideration without focusing on how the specific wireframes are arranged.

3. **Reasoning:**
   Third, the model analyzes each grasp candidate step by step and reasons about its suitability. Each grasp color should be evaluated on factors such as the orientation of the grasp, the shape of the object, and general scene and task understanding. This step is important so that each potential grasp is reviewed based on context in order that the VLM can compare the options.

4. **Choice:**
   Finally, the model makes a decision on which grasp is best suited to complete the task. In case the VLM encounters few grasps that seem almost equally feasible, the one with a small edge in terms of functionality is chosen. If none of these options are applicable the model can suggest that no grasp should be applied.

Chapter 4

Results and Discussion

This chapter presents the outcomes of the proposed grasp proposal pipeline. Both a quantitative and a qualitative analysis are used to determine the system's performance. A series of different scenes are designed to test its ability to generate stable, task-oriented grasps for different instructions. Additionally, going into detail on a couple of experimental runs allows for showcasing different capabilities and limitations of the approach.

4.1 Experimental Setup

The experiments were conducted using an Intel Lidar 515 camera. This sensor captures color and depth information with a resolution of 640x480. The camera was configured with the depth alignment setting enabled. The accurate spatial correspondence between the RGB and depth data is needed to ensure that the correct pixels are colored in the color image during the image rendering process. Additionally, the camera generates a point cloud directly and publishes it as a ROS topic. This real-time access to 3D spatial information makes the grasp generation simpler. The camera setup was positioned one to two meters from the table and tilted down at approximately 45 degrees. This setting simulates the viewpoint of a Franka Panda robot.

13 scenes were recorded for the experiments and there are 6 of which that contain two different objects in the same scene. Out of these dual-object scenes, the objects were interacting or encircling each other in 2 cases, and were out of contact with each other in the next 4 cases. As for the mechanics, all scenes were performed on a simple background, on a clean tabletop with no other items in the scene than the objects themselves. For some of them, more than one task descriptions were used in order to mimic different grasping conditions, thus, in total, 17 distinct task descriptions were given.

4.2 Quantitative Results

For each of the task descriptions, 20 runs of the pipeline were performed with the same scene. This repetition was necessary because the pipeline introduces variability even when working with identical inputs. The reasons are that, first of all, the sampling algorithms generate varying grasp proposals with each run. Second, the VLM inherently produces different responses with each query, even when the same visual and task inputs are presented. Therefore, the approach of having multiple


Figure 4.1: Examples of two different scenes captured by the Lidar 515. All the experimental runs were designed to evaluate specific capabilities, such as omitting the object's name in the task description (left) and performing task-oriented grasping (right).

Objects	**Task Description**	**SR**
Carrot, Cucumber	Hand me the green veggie.	100%
Carrot, Cucumber	I need more vitamin A, help me.	90%
Toothpaste, Rice Box	I want to brush my teeth.	100%
Milk Chocolate, Dark Chocolate	Hand me the milk chocolate.	95%
Milk Chocolate, Dark Chocolate	Hand me the healthier option.	80%
Orange Juice, Sponge	I am thirsty, hand me a drink.	95%
Orange Juice, Sponge	I spilled orange juice, help me clean it.	80%
Drill	Drill a hole.	85%
Paint Brush	Paint something.	80%
Brush	Clean the table.	75%
Papers	Pick up all the papers, they are loose, keep them from scattering.	70%
Mug, Paint Brush	I want to paint.	60%
Mug, Paint Brush	Hand me the mug.	50%
Chicken Drumstick	Pick it up, so I can take a bite.	55%
Ice Cream	Pick up the ice cream.	40%
Cutting Board	Pick up the cutting board in a way that minimizes the torque.	25%
Hot Dog	Pick it up, don't get dirty.	0%
Total		**69.4%**

Table 4.1: Task descriptions, corresponding objects, and success rates for each task in percentages over 20 runs.

runs on the same scene provides higher reliability of the assessment of consistency and performance of the system in successive tries.

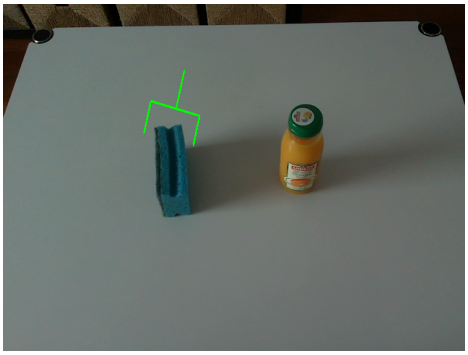
The success rate (SR) of the system is calculated as the ratio of runs where a human observer perceived that the grasp chosen by the robot was suitable for the completion of the tasks described. As shown in Table 4.1, the success rates vary significantly across different scenes, with some achieving 100% success while others perform much lower. This variability will be further explained in the following sections to understand the factors influencing the performance.

A general trend can be observed the system is quite accurate in scenes that contain two separable objects. This implies that the identification and selection of the right object according to the capacity of the model is quite effective. Altogether, The system obtains an average success rate of 69.4%, demonstrating a reasonably good performance across a variety of tasks and object configurations.

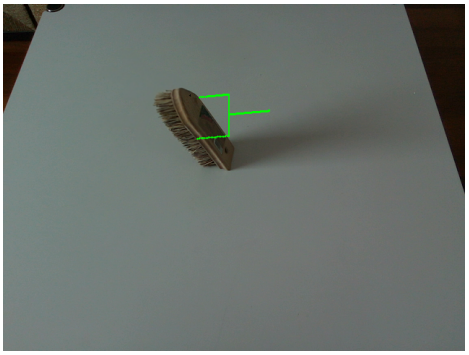
4.3 Capabilities

4.3.1 Open-vocabulary reasoning

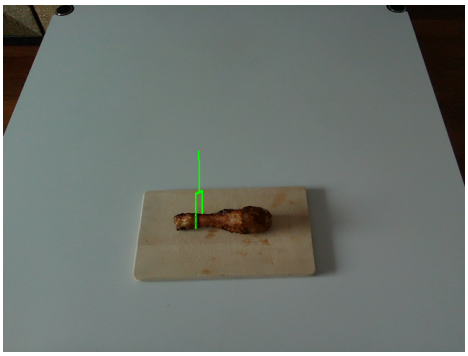
One of the main strengths of the proposed pipeline is its ability to generate effective grasp proposals without requiring the task description to name the objects or parts to be grasped explicitly. The VLM interprets the visual input and autonomously decides which object and grasp are appropriate for the task. This enables the system to handle task descriptions with minimal or abstract information, such as "Pick it up." Also, descriptions based on object properties like "Hand me the healthier option" are successfully interpreted. In the case demonstrated in Figure 4.2(a), the VLM even goes beyond the literal interpretation of the task description. Despite "orange juice" being explicitly named and visible in the scene, the VLM selects the sponge as the object to grasp. It understands that "The blue sponge is the most suitable object for cleaning."



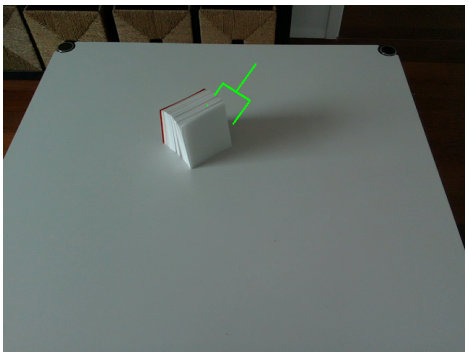
(a) "I spilled orange juice, help me clean it."



(b) "Clean the table."



(c) "Pick it up, so I can take a bite."



(d) "Pick up all the papers, they are loose, keep them from scattering.

Figure 4.2: Four different scenes, where the final output of the pipeline is rendered back into the image with a green wireframe.

4.3.2 Task-oriented grasping

Task-oriented grasping goes beyond just selecting the suitable object. The aim here is to choose the appropriate part of the object for the correct manipulation. To test this capability, one of the scenes used depicted a brush. In 75% of cases, the pipeline proposed a grasp that approached the brush from the correct side, securing a stable grip, as shown in Figure 4.2(b). The VLM reasoned that this grasp would allow "a secure grip without interfering with the bristles." Performance was even better in a scene involving a drill, where the task was to drill a hole. In nearly every case, the pipeline proposed a grasp on the handle from the correct side, ensuring the tool could be used properly.

4.3.3 Objects without distinct parts

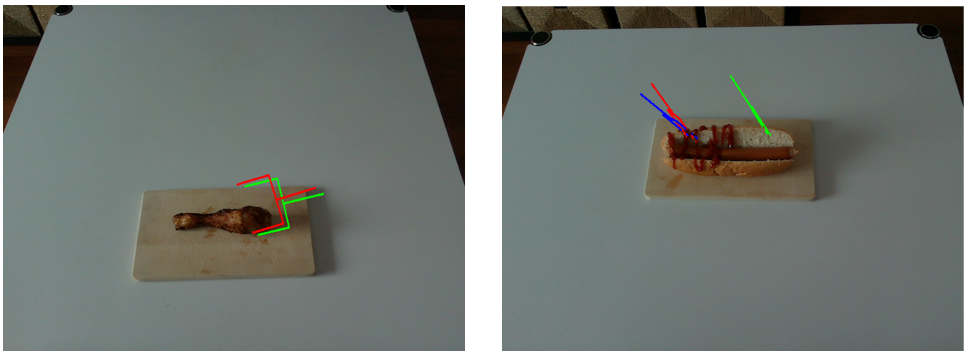
Some objects, however, do not have clear, distinct parts like bristles or handles. One example tested was a chicken drumstick. Through experience, humans know that picking up a drumstick by the bone leaves the meaty part accessible for eating. The VLM showed similar reasoning, often identifying the correct location to grasp, as shown in Figure 4.2(c). However, the correct grasp location was not always proposed. The reasons for the success rate of just 55% will be further discussed in section 4.4.

4.3.4 Understanding the scene

In some situations, it is not enough to identify the objects and their parts to determine the optimal grasp position. A deeper understanding of how objects behave based on their physical properties is required. This is tested in a scene where several loose papers are stacked together. To prevent scattering, the VLM proposes a side approach, stating, "The side approach should allow for a secure grip on the bulk of the stack, minimizing the risk of pages scattering." This kind of reasoning was often given by the VLM, leading to a success rate of 70% for this scene. However, the VLM performed poorly in another experiment where the goal was to pick up a cutting board in a way that minimized torque. The reasoning was correctly given in the successful trials, but often incorrect reasoning led to failed attempts. This challenge will be further addressed in section 4.4.
Nevertheless, the analysis shows that the pipeline can suggest grasps based on a deeper understanding of the scene beyond geometric data and semantic labeling. The quality, however, highly depends on the complexity of the understanding needed and the specific scene.

4.4 Limitations

In this section, we discuss two main reasons why the pipeline might produce grasps that do not align with how a human would pick up an object to fulfill a task. First, the filtering process can remove too many viable grasp options. As a result, there may be no correct grasp left for the VLM to reason about. Second, the VLM's reasoning itself can be flawed, leading to incorrect interpretations of the visual information.



(a) (b)

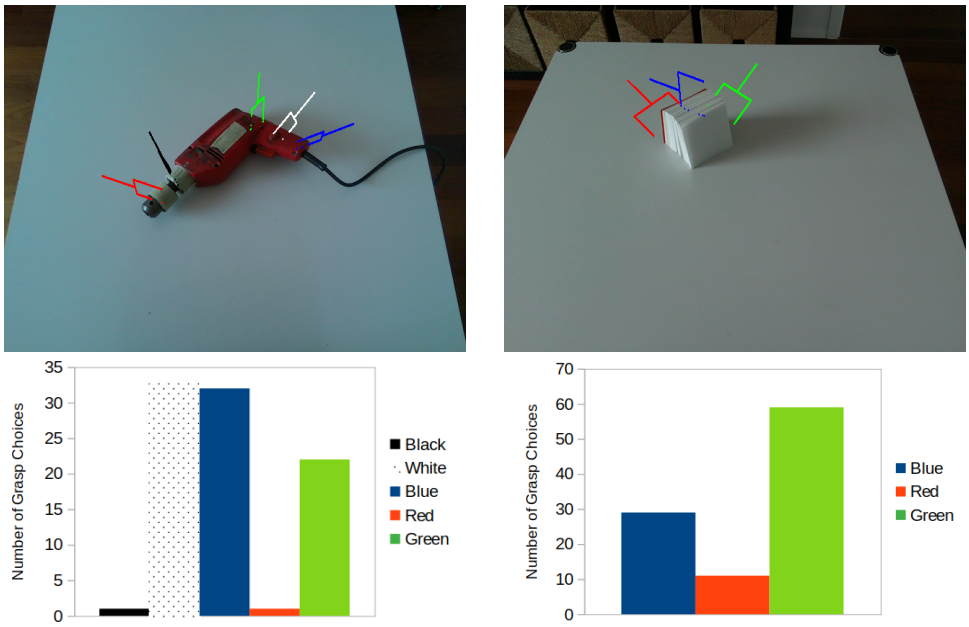
Figure 4.3: Two examples where the grasp generation and filtering process does not produce a viable set of grasps, making it impossible for the VLM to choose a correct grasp location

4.4.1 Geometric Bias in Grasp Filtering

The first part of the pipeline, up until the image rendering, focuses entirely on generating stable and reachable grasp options for the VLM based purely on geometric considerations. No semantic reasoning is involved in this stage. While this ensures that the final grasp selected by the VLM is likely practical, it can also lead to issues. Edge Grasp tends to favor certain local geometric features, which can result in many high-stability grasps clustering in one area of the scene. In scenarios with two objects, this can cause the final rendered image to display grasp wireframes only on one object. Even with a single object, as shown in Figure 4.3(a), the grasps might not cover the area we are actually interested in. Additionally, Figure 4.3(b) highlights another issue: although the grasps are well distributed across the object, the lack of semantic reasoning means the pipeline overlooks important details. For instance, while the grasps would be stable if the hotdog were a solid object, the soft bread and loose sausage would cause it to fall apart if lifted this way.

4.4.2 Faulty VLM Reasoning

The pipeline can still fail due to the VLM's reasoning process even when stable, well-distributed grasp proposals are present. Common issues with LLMs, such as hallucinating information, misunderstanding subtle details, or struggling with ambiguous situations, are present here as well. The consistency of its choices was tested in different scenes to find the factors that influence the quality of the VLM results. For each scene, the VLM gave 100 responses for the exact same rendered image. This way, the uncertainty of the VLM can be analyzed. While this also largely depends on the temperature setting of the VLM, the qualitative analysis of these experiments allows to find general aspects leading to more inconsistent results of the VLM. The complexity of the scene and task description and the closeness of the different grasps were identified as the main factors for the inconsistency of the VLM. Inconsistency in its answers does not necessarily mean that the result will be wrong (e.g., multiple answers could be correct, as in Figure 4.4a). However, recognizing that proximity of grasps leads to inconsistent answers is important, since it can result in suboptimal selections for precise tasks.



(a) (b)

Figure 4.4: Rendered images and the respective histogram of choice selection with the task (a) "Drill a hole" (b) "Pick up all the papers, they are loose, keep them from scattering." The VLM is queried 100 times on each scene with a temperature of 1.

4.5 Runtime Analysis

The runtime of the entire pipeline is deduced by taking the average of the runtimes of one run for each of the 17 task descriptions. Intermediate timestamps are taken at different points in the pipeline to show how long each of the steps takes. The runtime analysis was performed on an NVIDIA GeForce RTX 3050 Ti Laptop GPU. The total time to produce a final grasp position is 13.05 seconds. As shown in Table 4.2, two steps are used most of the time. First of all, the path planning takes 8.56 seconds. Even though this is a crucial step to ensure the grasps are reachable, another approach to test this (e.g., based on geometric considerations of the scene) could reduce the time needed for this step. Second of all, the VLM Query takes 6.96 seconds. A better network service quality could potentially improve this. However, the response time also largely depends on the provider of the VLM (in this case, OpenAI). Therefore, it can only be influenced to a certain degree or by choosing a different VLM in the future.

Processing Step	**Time [s]**
Pointcloud processing	0.80
Stability Filter: Edge Grasp	0.64
Feasibility Filter: MoveIt Path Planning	8.56
Image Rendering	0.31
ChatGPT-4o Query	6.96
Total	**13.05**

Table 4.2: Processing times for different steps

Chapter 5

Conclusion and Future Work

This projec presented a novel pipeline for generating semantic grasps. First, stable and feasible grasps are generated based on the 3D point cloud. Only in a second step are these grasps rendered back into the image and given to a VLM for reasoning about which grasp to pick. The system successfully demonstrates the ability to propose stable grasps that align with human intuition in many scenarios. It performs particularly well when distinguishing between objects. It uses visual information as well as additional knowledge about these objects to propose a grasp on the correct one. The quantitative results across multiple scenes and tasks indicate that the pipeline performs well. An overall success rate of 69.4% was achieved.

Additionally, the qualitative analysis of the resulting grasps shows that the system can reason beyond geometric data. It is able incorporate more abstract and task-specific requirements into its decision-making. One of the key strengths of the pipeline is that the VLM does not necessarily require specific object names or parts. Hence, there is a lot of flexibility when giving different commands or objects to the grasping robot. Furthermore, the resulting grasp position can manipulate objects in a way that aligns with the intended action. This enables tools such as a brush or a drill to be handled correctly in a task-oriented f1ashion.

However, it is possible to single out two limitations of the research. First of all, the edge grasp algorithm shown in Section 3.2 prefers specific local shapes. This means that, at times, the grasp proposals are summarized in a single area of an object or scene. Future work could improve on this by taking images from multiple points of view. More understanding of the 3D object may be beneficial for improving the grasp proposals' quality and diversity. Second of all, the VLM's reasoning is not always reliable. This is particularly the case with more complex input or when grasps are closely spaced. Here, an approach with multiple VLM queries or a different choice of grasp visualization could produce more consistent and correct results.

Despite these challenges, the proposed pipeline shows that utilizing VLMs to directly reason on possible grasp positions is a promising approach to achieve zero-shot semantic grasping. With implemented improvements and tests in real-world scenarios, it could serve as a powerful tool for autonomous robots interacting with dynamic environments.

Bibliography

[1] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple open-vocabulary object detection with vision transformers."

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision."

[3] C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, "Task-oriented grasp prediction with visual-language inputs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4881–4888, ISSN: 2153-0866.

[4] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "GraspGPT: Leveraging semantic knowledge from a large language model for task-oriented grasping."

[5] C. Tang, D. Huang, W. Dong, R. Xu, and H. Zhang, "FoundationGrasp: Generalizable task-oriented grasping with foundation models."

[6] S. Li, S. Bhagat, J. Campbell, Y. Xie, W. Kim, K. Sycara, and S. Stepputtis, "ShapeGrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition."

[7] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping."

[8] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, "LAN-grasp: Using large language models for semantic object grasping."

[9] T. van Oort, D. Miller, W. N. Browne, N. Marticorena, J. Haviland, and N. Suenderhauf, "Open-vocabulary part-based grasping."

[10] H. Huang, D. Wang, X. Zhu, R. Walters, and R. Platt, "Edge grasp network: A graph-based SE(3)-invariant approach to grasp detection."