Ph.D. DISSERTATION

Toward Reliable Vision-Based Robotic Grasping in Transparency and Clutter

투명 물체 및 복잡한 환경에서의 신뢰성 있는 비전 기반 로봇 파지 연구

By

Junho Lee

AUGUST 2025

DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
COLLEGE OF ENGINEERING
Seoul National University

Ph.D. DISSERTATION

Toward Reliable Vision-Based Robotic Grasping in Transparency and Clutter

투명 물체 및 복잡한 환경에서의 신뢰성 있는 비전 기반 로봇 파지 연구

By

Junho Lee

AUGUST 2025

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
Seoul National University

Toward Reliable Vision-Based Robotic Grasping in Transparency and Clutter

투명 물체 및 복잡한 환경에서의 신뢰성 있는 비전 기반 로봇 파지 연구

지도교수 김 영 민

이 논문을 공학박사 학위논문으로 제출함

2025 년 8 월

서울대학교 대학원

전기 정보 공학부

이 준 호

이준호의 공학박사 학위논문을 인준함

2025 년 8 월

위 원 장	오성희
부위원장	김영민
위 원	박재식
위 원	김아영
위 원	이용현

Abstract

Vision-based grasping focuses on determining successful grasp configurations using input from vision sensors. Typically, algorithms take RGB and depth images from modern RGB-D cameras and output where and how to grasp a target object in 3D space. With advances in grasping algorithms and vision sensors, vision-based grasping has become more reliable across a wide range of scenarios. Grasping models, enhanced by sophisticated engineering, can incorporate physically grounded biases—such as smoothness and center of gravity—to generate high-quality grasps based on object geometry. At the same time, vision sensors leveraging technologies like stereo vision, LiDAR, and infrared have become more capable and affordable, allowing accurate geometry capture for most objects. Moreover, the increasing availability of real-world datasets has significantly boosted performance in practical robotic applications.

However, scenes containing transparency and clutter often fail to be correctly recognized by vision sensors, greatly reducing the reliability of vision-based grasping algorithms. First, transparent objects cause complex sensing failures in depth cameras due to its physical properties. Second, clutter, where multiple objects are in contact, results in occlusions and unobservable surfaces. These elements obstruct the acquisition of accurate geometry in a vision-based manner, leading to inaccurate grasps.

In this thesis, I propose a reliable approach for acquiring scene geometry in environments with transparency and clutter. By leveraging information available from general pretrained vision modules, the method focuses on the aspect of generalization to various scenes. It extracts and utilizes mid-level representations such as masks and surface normals in spatially structured ways to achieve stable geometric reconstruction. First, I introduce a data-driven method for reliably obtaining instance masks in cluttered scenes containing transparent objects, along with a corresponding grasping algorithm. Second, I propose a novel 3D representation based on surface normals to effectively capture the geometry of transparent objects. Finally, I present an interactive perception pipeline that actively acquires instance-level 3D geometry in cluttered scenes with occlusions. Furthermore, experiments conducted on a real-world robotic platform demonstrate the potential for practical deployment in real-world scenarios.

To enable robots to effectively replace human labor across diverse scenarios, consistent performance in variable conditions must be ensured. By addressing transparency and clutter—two major challenges for vision-based grasping—and proposing solutions built on general-purpose vision modules, this thesis aims to improve the stability and robustness of robotic manipulation.

Keywords: Vision-based robotic grasping, reliability, vision-based models, transparency, clutter

Student Number: 2020-25228

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Vision-based robotic grasping refers to techniques that enable robots to grasp objects using input from vision sensors. Formally, it can be viewed as a mapping from sensor observations—such as images or point clouds—to grasp configurations that enable successful object manipulation. This approach has attracted significant attention due to its potential for enabling robotic grasping in unstructured and diverse environments [104]. When the size, shape, and position of objects are unknown, vision sensors provide essential information for efficiently computing feasible grasps [98, 93, 50]. Moreover, vision-based grasping plays a key role in supporting more complex tasks such as object rearrangement and bin packing.

In its early stages, vision-based robotic grasping adopted a straightforward classification approach, mapping visual input and grasp configurations to a scalar value representing grasp quality [98, 95]. These methods typically used a single RGB-D image captured by mainstream depth sensors [38], along with grasp candidates constrained to be perpendicular to the image plane. Training data was often generated in simulation using diverse object meshes annotated with successful grasps computed via classical algorithms [114]. Based on this data, CNN-based models were able to learn grasping strategies and be deployed in similar environments. However, these approaches suffered from domain discrepancies—such as noise in real-world depth data—that limited their robustness. Additionally, restricting grasps to be perpendicular to the image plane reduced the degrees of freedom, making it difficult to handle objects with complex geometries effectively.

With the proliferation of 3D scanning hardware [38, 40, 126], many methods began collecting data directly from real-world environments, significantly narrowing the domain gap between training and deployment [84]. At the same time, grasp configurations evolved from 2D plane grasps (3-DoF) to full 3D volume grasps (6-DoF). While 2D planar grasping benefited from a reduced search space, its heavy dependence on image viewpoint limited its generalization to diverse object geometries. By transitioning to 6-DoF grasping, vision-based approaches became applicable to more realistic and unconstrained scenarios. Recent methods have adopted more complex processing pipelines, incorporated improved heuristics, and utilized realistic datasets to enable robust object grasping in varied real-world environments [51]. Building upon traditional CNN-based encoders, newly developed modules were carefully engineered to not only interpret object geometry but also attend to local regions most likely to support successful grasps [131]. In addition, physically grounded heuristics—such as modeling object parts as cylinders and factoring in their center of gravity—further improved grasp quality [51]. The increasing availability of real-world grasping data has also made it feasible to train vision-based models that can generalize and be deployed across arbitrary robotic grasping settings [50, 52].

However, despite significant progress in developing reliable vision-based grasping models, certain artifacts still severely impact grasping accuracy. In scenarios where object geometry is difficult to capture with vision sensors, grasp success rates can drop sharply. A prominent example involves transparent objects, such as glass. Depth sensors—commonly relying on optical cues like stereo vision, LiDAR, or infrared—struggle to accurately reconstruct the surfaces of transparent materials [121]. Moreover, simulating the measurement noise associated with these sensors remains challenging, as it varies significantly depending on factors such as the camera type, background, and properties of the transparent material. Another major obstacle to accurate geometric reconstruction is occlusion, which often occurs in cluttered scenes or at contact points between different objects. In such cases, the unobservable surfaces hinder effective scanning, resulting in incomplete or inaccurate shape estimation [78].

To address the challenges of vision-based grasping in the presence of transparency or clutter, numerous studies have emerged, generally following two main approaches. The first focuses on leveraging the noisy sensor data directly, based on the idea that well-trained grasping models can still succeed despite input noise. Some methods simulate the depth distortions caused by transparent objects using advanced graphics techniques like ray tracing [135], enabling the training of grasping models that are robust to such artifacts [72]. Other approaches aim to grasp effectively using only the visible surfaces in cluttered environments, without relying on complete object geometry [17, 117, 131]. The second approach seeks to improve the quality of geometric reconstruction itself. For transparent objects, some methods employ vision models pretrained on specially generated datasets [121], or use multiview aggregation techniques such as neural radiance fields (NeRF)[102] to produce more accurate 3D representations[66]. In cluttered scenes, other methods adopt primitive-based modeling strategies—such as fitting superquadrics—to infer unobservable surfaces and complete the object geometry [78, 142].

My research aligns with the second line of thought, focusing on acquiring geometry in transparent and cluttered scenes. Distinctively, I place strong emphasis on reliability by leveraging outputs from vision models designed for general-purpose data, avoiding any scene-specific fine-tuning. I treat mid-level vision estimators—such as object masks and surface normals—as noisy sensors, and design pipelines that extract consistent object geometries from these imperfect signals. Unlike prior work [66, 74], I deliberately avoid scene-specific post-processing or hyperparameter tuning, which can compromise generalization. Real-world experiments on our robotic platform demonstrate the robustness and deployability of the proposed approach.

I believe vision-based robotic grasping has the potential to revolutionize a wide range of physical tasks, particularly those involving the manipulation of diverse objects. Many operations that still rely on manual labor—such as picking and placing items in logistics settings—could be automated through robotics. However, to realize this potential, robots must perform reliably across arbitrary real-world environments. Addressing and understanding the current failure modes in vision-based grasping is therefore a crucial direction. To improve reliability, solutions should capitalize on advances in general-purpose vision modules without relying on scene-specific adaptations. With this thesis, I aim to contribute to methods that harness deep learning-based vision systems to make vision-based robotic grasping more dependable.

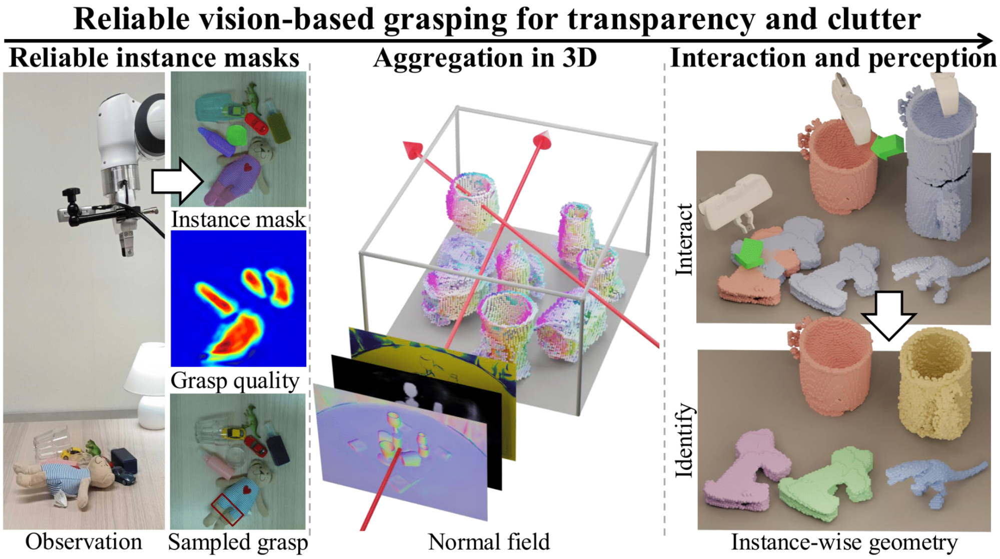


Figure 1.1: Overview of this thesis. To advance the goal of reliable vision-based grasping in the presence of transparency and clutter, this thesis presents three key contributions. First, I propose a data-driven approach for obtaining robust instance masks. Second, I introduce a 3D geometry aggregation method for transparent objects based on a novel normal field representation. Finally, I present an integrated interaction and perception pipeline designed to handle complex, cluttered scenes.

1.2 Research Goals

The ultimate goal of this thesis is to develop reliable methods that leverage information from pretrained vision models to robustly perceive object geometry in challenging scenarios where traditional vision sensors often fail. These challenging cases include, but are not limited to, environments containing transparent objects, occluded scenes, and visually ambiguous arrangements, all of which significantly degrade the performance of conventional depth sensing techniques.

Accurate and generalizable geometric perception in such settings is essential for enabling robust robotic manipulation and interaction in real-world environments. In particular, I focus on two major difficulties—transparency and clutter—where existing methods frequently rely on handcrafted heuristics or scene-specific calibration to compensate for sensing limitations. Instead, my goal is to design methods that generalize across diverse conditions without requiring such manual interventions. To this end, I propose three key ideas that augment and utilize pretrained vision models in novel ways.

Idea 1: *Enhancing instance segmentation modules to handle both transparent and opaque objects.* In Chapter 3, I present a method for reliably obtaining instance masks in cluttered scenes containing a mix of transparent and opaque objects, along with a corresponding mask-based manipulation strategy. To address the limitations of existing datasets, which are largely biased toward opaque objects [89], I augment the data with additional transparent object examples, resulting in more robust instance segmentation. Building on prior grasp estimation methods [104], I further develop a mask-based grasping pipeline that effectively avoids clutter, enabling more stable grasping performance, as shown on the left of Fig. 1.1.

Idea 2: *Estimating and aggregating multiview surface normals to reconstruct 3D volumes of transparent objects.* In Chapter 4, I propose a method that leverages pretrained vision models predicting surface normals [4] from RGB inputs for multiview reconstruction of transparent objects. Unlike existing approaches that directly aggregate RGB views—often suffering from high per-view variance due to transparency and requiring scene-specific thresholding [66]—my method builds a surface normal-based field representation in 3D space. This representation enables both effective and efficient reconstruction of transparent object geometry. The resulting volume is further used for real-world 6DoF grasping, as illustrated in the center of Fig. 1.1.

Idea 3: *Leveraging instance segmentation outputs for interaction-based perception of contacting objects.* When multiple objects are in contact, occlusions often lead to missing or unobservable surfaces, posing challenges for reliable geometric perception. In Chapter 5, I introduce a pipeline that uses pretrained instance segmentation models [21] to infer hidden surfaces, actively induce changes through interaction, and efficiently update the object geometry. To achieve this, I design image-efficient optimization strategies that selectively refine instance-level geometry, enabling accurate reconstruction despite occlusions.

Although presented as separate ideas, the three contributions of this thesis are closely aligned toward the overarching goal of achieving reliable vision-based perception in previously unsolved and challenging scenarios—such as transparent objects, heavy occlusion, or objects in contact—in practical, real-world environments. These scenarios are particularly difficult because they challenge the assumptions underlying most conventional vision-based systems, such as the availability of clean depth maps or unambiguous visual features. To address these limitations, the combined use of robust instance masks, multiview surface normal aggregation, and interaction-driven updates provides a comprehensive solution that operates reliably even under these adverse conditions. Robust instance masks enable the initial decomposition of the scene into meaningful object-level regions, even when objects are partially visible or visually similar. Multiview normal aggregation takes advantage of geometric consistency across viewpoints to overcome local sensing noise and produce coherent surface representations, which are especially important for grasping or manipulation planning. Meanwhile, interaction-driven updates allow the system to actively gather new information by physically perturbing the environment, thereby resolving ambiguities that cannot be disambiguated through passive observation alone. Together, these components form a tightly integrated pipeline that addresses both the sensing and reasoning aspects of the perception problem, enabling a robot to form a more complete and accurate understanding of its surroundings. Crucially, all three methods are designed to work with vision modules trained on general-purpose datasets, such as those used for instance segmentation or normal prediction in natural images. By building on pretrained models rather than training bespoke systems for each new scene, the proposed framework avoids scene-specific tuning and brittle hand-engineered heuristics that often limit generalization. This greatly enhances the scalability and applicability of the system across a wide range of robot platforms and deployment environments. Ultimately, this integrated and generalizable approach represents a meaningful step toward dependable robotic perception for complex, real-world manipulation tasks, bringing vision-based robotic systems closer to the level of robustness and adaptability needed for everyday use.

Chapter 2

Related Works

2.1 Vision-Based Robotic Grasping and Challenges

Robotic grasping guided by visual sensing has long been recognized as a fundamental and challenging problem in robotics, especially in scenarios involving unstructured or dynamic environments where explicit prior knowledge about object shape, location, or identity is scarce or entirely absent. The ability to perceive and understand complex scenes through vision is essential for enabling autonomous robots to interact robustly and effectively with the physical world. Vision-based grasping systems address this challenge by integrating perception and control, aiming to infer reliable grasp configurations based entirely on visual input. These visual cues are typically captured in the form of RGB images, depth maps, or a combination of both modalities (i.e., RGB-D), each offering complementary information about the appearance and geometry of the scene [2, 31, 49, 56]. The general structure of vision-based grasping frameworks often follows a modular pipeline that begins with identifying the object or region of interest, which may involve techniques for object detection, instance segmentation, or salient region localization. Once the target is localized, the system proceeds to estimate a suitable grasp pose, leveraging the visual information to infer the local geometry and determine how the robot's gripper should be oriented and positioned to achieve a successful grasp [48, 10, 11].

The representation of grasp plays a critical role in shaping the algorithm. Early approaches typically adopt a planar grasp formulation—representing a grasp as a 2D point, angle, and gripper width—allowing lightweight regression from top-down visual input. These methods have shown success in relatively constrained environments and are fast enough for real-time deployment. RGB-based models regress grasp rectangles directly from image features [118, 19, 112, 145, 46], while others leverage depth input [94, 104, 121, 58] or fused modalities [36, 111] for better spatial localization. Despite their efficiency, such planar grasping methods often fall short in cluttered or highly variable environments, where a 3D understanding of object geometry is required.

To tackle more general scenarios, researchers have moved toward full 6-DoF grasping frameworks that predict the 3D position and orientation of the gripper with respect to the target object. These approaches typically rely on 3D input representations, such as point clouds [67, 136, 105] or voxelized data derived from RGB-D sensors [128, 107, 59]. Such methods offer greater flexibility in grasping objects at arbitrary poses, including those in stacks or corners, but often demand greater computational resources and higher-fidelity scene reconstructions. More recent approaches utilize both real-world data and human-like heuristics to achieve impressive performance [42]. Some papers use attention mechanisms to allow a learned network to focus on graspable regions, while others leverage cylinder assumptions to simulate the centor of gravity [131, 51].

From these advances, 6-DoF grasping given the geometry of the objects can reliably be solved.

However, certain scenes pose significant challenges for vision-based object geometry acquisition. One major challenge is transparency, which affects the reliability of various depth sensors. Time-of-flight sensors, such as those using infrared light [38] or LiDAR [39], often fail to return accurate depth readings when rays pass through or are scattered by transparent surfaces, resulting in unstable measurements. In addition, occlusions caused by clutter can render parts of objects unobservable, making it difficult for vision-based systems to capture complete and accurate geometry [142].

2.2 Transparency

Perceiving the geometry of transparent objects remains a long-standing challenge due to their unique optical properties, such as refractive distortion and the absence of strong visual or depth cues [1]. Conventional RGB-D cameras, which rely on active infrared sensing or structured light, often fail with transparent materials, as these materials tend to refract, transmit, or scatter the projected signals—leading to missing or erroneous depth measurements [120]. As a result, vision-based robotic systems often struggle in everyday environments such as households or laboratories, where transparent objects like glassware, plastic containers, or water surfaces are commonly encountered.

To address the challenges of transparent object perception, prior research has explored alternative sensing modalities and capture strategies leveraging specialized hardware. For example, polarization cameras [57] and thermal imaging systems [76] have shown the ability to detect transparent surfaces under specific conditions. However, such sensors are not yet widely adopted in robotic platforms due to factors such as high cost, integration complexity, and limited accessibility. Additionally, classical approaches that capture multiple RGB images of transparent objects against structured backgrounds (e.g., checkerboard patterns) can reconstruct high-fidelity geometry [134]. Nevertheless, these methods are computationally intensive and require controlled capture setups involving specific hardware like calibrated backgrounds and motorized turntables.

A more scalable alternative involves developing methods that operate solely on RGB inputs captured from hardware configurations closely aligned with typical robotic grasping setups. In particular, a growing body of work adopts data-driven approaches, where deep learning models trained on large-scale datasets predict object geometry directly from RGB images. For example, ClearGrasp [120] introduces a synthetic dataset of transparent objects to facilitate depth estimation from RGB input. On the dataset front, annotated images of transparent objects are becoming increasingly available at scale [53, 75]. The method introduced in Chapter 3 follows this paradigm by aiming to extract reliable instance segmentation masks for transparent objects in a data-based manner. These masks abstract away material-specific visual properties while preserving object boundaries and spatial relationships, enabling the grasping module to function in a material-agnostic manner [61]. By explicitly reasoning about object identities and their spatial configuration, the proposed approach is better equipped to handle failure cases arising from cluttered scenes or ambiguous boundaries. Furthermore, I propose a training strategy that augments existing RGB image datasets with synthetic segmentation masks and corresponding grasp annotations. This design supports scalable learning across a wide range of object types—both transparent and opaque—without relying on costly depth data collection or specialized simulation infrastructure. By decoupling instance-level perception from grasp prediction, the method improves robustness and enhances transferability across domains and sensor modalities.

On the other hand, the rise of implicit neural representations has led to a shift in how scenes are encoded [87, 124, 143, 130, 24, 91, 132]. Neural Radiance Fields (NeRF) [102] parameterize a scene as a continuous function that maps 3D coordinates and viewing directions to RGB color and volume density. Although capable of synthesizing high-quality images from novel viewpoints, NeRF's reliance on dense sampling and MLP inference makes them computationally expensive for robotics applications. To accelerate inference, several methods have proposed using sparse voxel grids or learned feature volumes to store intermediate representations. Examples include DVGO [127], TensoRF [23], and Instant-NGP [106], all of which significantly reduce training and rendering time.

The capability to aggregate multi-view information into 3D geometry inspired works that perform robotic grasping on NeRFs [133, 85]. Especially for transparent objects, where scanning hardware fail, methods to obtain the geometry and perform grasping have been developed [66, 43, 73]. From the geometry captured via NeRF frameworks, some methods use post processing such as thresholding [66] to obtain depth images. Other use a dataset based approach of directly predicting 3D geometry in the form of SDFs(Signed Distance Functions) [43]. However, in Chapter 4, I propose a fundamental mismatch in the NeRF framework and the RGB images of transparent objects. This leads to NFL, or Normal Field Learning, which successfully captures the geometry of transparent objects without scene-wise post-processing. NFL demonstrates that reliable surface normal fields can be constructed from posed RGB images alone, bypassing the need for accurate depth sensing. This property is particularly valuable in scenes with transparent or specular surfaces. Compared to methods like Dex-NeRF [66] that rely on noisy NeRF-derived depth, or GraspNeRF [43] that train end-to-end from synthetic images, NFL offers a modular and efficient pipeline. It achieves fast scene reconstruction (under 40 seconds) and supports generalization to real-world scenarios without extensive retraining.

2.3 Clutter and Occlusion

In scenes where multiple objects are densely arranged, occlusions hinder the accurate acquisition of object geometry. Moreover, adjacent objects pose additional challenges for perceiving instance-level information [54]. In particular, neighboring surfaces can introduce inherent ambiguity in instance identification—where an instance refers to a set of parts that move together as a single unit. For example, it can be unclear whether the lid and body of a bottle constitute a single instance, depending on their attachment or articulation. These challenges in accurately perceiving both geometry and instance segmentation often result in unreliable or failed grasp attempts [142].

One approach to handling occlusion-prone scenes is to leverage geometric primitives to complete unobservable object surfaces. Among these, superquadrics [12] are commonly employed to approximate and fill in missing regions of partial or noisy geometry [142, 78]. While effective for representing a wide range of convex shapes, superquadric-based methods are inherently limited in expressiveness—they struggle with complex or concave geometries, such as cups or bowls, and typically do not address instance-level ambiguity, such as distinguishing articulated or adjacent parts of objects.

An alternative line of work seeks to simultaneously recover both geometry and instance information in cluttered scenes using implicit neural representations. Volumetric aggregation frameworks such as NeRF [102] can be extended to incorporate rich image features extracted from vision foundation models [125, 139]. Several recent studies closely related to our work leverage segmentation labels [21] to construct feature fields that encode instance-level information [14, 74, 60, 144, 28, 149, 34, 138, 147]. These fields jointly represent object geometry and instance identity in a unified manner. In particular, contrastive learning has been employed to train affinity fields [14], where points belonging to the same object are mapped to similar feature embeddings. To extract discrete object instances from these continuous fields, clustering algorithms such as HDBSCAN [20] are typically applied in 3D space. Despite the expressive power of these methods, most rely on passive observation of static scenes, which often leads to persistent ambiguities in both geometric reconstruction and instance discrimination. To overcome these limitations, our method—introduced in Chapter 5—incorporates active perception, applying controlled scene interactions and using their effects to refine and update the instance-aware field representations.

Active perception [6, 8, 7] refers to a class of methodologies in which a robot leverages its own actions to improve its understanding of the environment. It is a well-established area of research with broad applications [16], including pose estimation [137, 35], object segmentation [22, 122], articulation [109] and dynamics modeling [3], and the learning of manipulation [82] or grasping strategies [113]. Recent works have extended active perception toward high-level scene understanding by incorporating large language models and structured heuristics. These systems can perform complex interactions to discover objects and construct symbolic representations such as scene graphs [68]. However, these approaches often rely heavily on precise prompt engineering and assume access to near-perfect segmentation modules—conditions that may not hold in real-world settings. Among related approaches, the work most aligned with our philosophy is UNCOS [54], which addresses instance ambiguity in cluttered scenes by using robotic pushing guided by an uncertainty-based metric. While

UNCOS focuses on resolving ambiguity among adjacent objects, our proposed *Interact-to-Identify* framework in Chapter 5 fundamentally differs in both objective and formulation. Specifically, UNCOS does not explicitly consider occlusion in its design, whereas *Interact-to-Identify* in Chapter 5 is built to operate in occlusion-prone environments by actively uncovering previously unobservable regions through physical interaction. In addition, UNCOS is developed in the 2D domain and relies on texture-based visual tracking [33] to detect scene changes. In contrast, *Interact-to-Identify* in Chapter 5 operates entirely in 3D and relies solely on geometric cues, making it more robust to texture variations and better suited for complex, cluttered, and partially occluded scenes.

Chapter 3

Enhancing Instance Segmentation Modules for Grasping

One of the elementary tasks for an autonomous agent is to detect and grasp objects for advanced manipulation tasks such as picking, sorting, and placing items. Approaches to vision-based grasping aim to predict an optimal grasp pose for each object in an unstructured environment. With the advance in deep learning, most of the recent successful approaches [94, 98, 92, 95, 96] are based on Convolutional Neural Networks (CNNs) in a supervised learning setup. However, the approaches often fail to generalize to transparent objects because their training datasets [84, 45] often under-represent transparent objects which exhibit very different visual properties and unreliable depth measurement. Nonetheless, transparent objects (*e. g.*, glasses, cups, or plastic containers) are easily found in daily life and especially dominant in laboratories or kitchens, which the autonomous agents should be able to handle.

Constructing a large-scale dataset that encompasses a diverse and balanced mix of both opaque and transparent objects with accurate grasping annotations would be a crucial step toward overcoming the current limitations faced by vision-based grasping systems. Such a dataset would enable the training and evaluation of models that generalize across different material properties, particularly addressing the unique challenges posed by transparent objects, which are often invisible or severely distorted in depth sensing. However, to the best of my knowledge, no publicly available dataset currently satisfies this requirement. Existing large-scale vision datasets, such as MS COCO [89] and ImageNet [44], primarily consist of opaque objects and contain only a negligible number of transparent instances, which are insufficient for training models specifically tailored to handle transparency. On the other hand, datasets that do focus on transparent objects tend to be limited in scope and lack annotations relevant to robotic grasping. For example, TOM-Net [26] and the segmentation datasets by Xie et al.[140, 141] concentrate exclusively on the perception of transparent materials but do not include grasp-related labels or pose information. As a result, they are not directly applicable to robotic manipulation tasks. Some recent approaches have attempted to address the problem by demonstrating robotic grasping capabilities specifically for transparent objects using specialized perception techniques. These include methods that perform depth completion based on RGB-D fusion[121] or leverage multiple light-field images to reconstruct the geometry of transparent surfaces [148]. While effective in controlled settings, these approaches often involve complex, multi-stage pipelines that introduce significant computational overhead. Consequently, their practicality for real-time robotic applications in unstructured environments remains limited, highlighting the need for unified, efficient, and generalizable solutions supported by more comprehensive datasets.

In this chapter, I propose an efficient, real-time robotic grasping approach, **MasKGrasp**, that effectively handles both transparent and opaque objects and

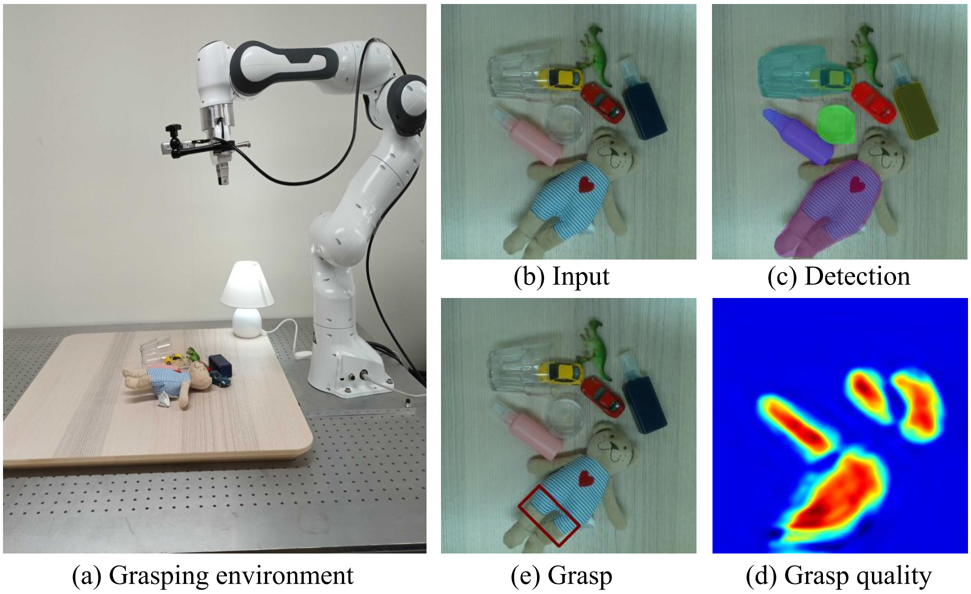


Figure 3.1: In the (a) robotic grasping environment, given an (b) input RGB image, my approach detects (c) instance masks of each object and estimates (d) grasp pose and (e) grasp quality for each instance mask.

generalizes to real-world objects with complex shape variations. My method consists of two simple stages: detection and grasp estimation. As shown in Fig. 3.1, given an input RGB image, the detection module first accurately estimates instance segmentation masks for both transparent and opaque objects (Fig. 3.1 (c)). Then given top **K** instance masks, the grasp estimation module finds an optimal grasp pose (Fig. 3.1 (e)) from the grasp quality (Fig. 3.1 (d)) for each instance. The essence of my approach is the usage of instance masks as the intermediate representation for both types of objects from which the optimal grasp is estimated. Because the instance mask effectively extracts essential geometric layout while factoring out the appearance variation (*i. e.*, opaque or transparent), the grasp estimator can handle both types of objects as long as accurate instance masks are given.

I make the key contributions as follows. *(i)* I propose a mask-based robotic grasping approach that successfully handles both opaque and transparent objects. I show that instance masks contain sufficient geometric context of the scene for grasp estimation even in challenging multiple-object scenarios. *(ii)* I design my grasping algorithm to consider free space between multiple instance masks and predict the best probability grasp that avoids cluttered regions. *(iii)* To accurately estimate instance masks for both types of objects, I propose a large-scale instance segmentation dataset that contains both object types with their instance annotations, by extending an existing large-scale dataset [89] with synthetic transparent objects augmentation [26]. *(iv)* Training a state-of-the-art instance segmentation method [62] on my dataset, I demonstrate that the model generalizes robustly to both types of real-world objects without sacrificing the accuracy on opaque objects.

On a real-world test environment with unseen challenging objects, MasKGrasp outperforms the previous approach [121] to transparent object grasping while achieving real-time performance. I demonstrate that my augmentation scheme, along with my mask-based grasping algorithm, provides a solution for multi-object grasping in the real world with unseen opaque and transparent objects.

3.1 Method

My method targets a general and challenging scenario for robotic grasping, where the scene contains multiple transparent and opaque objects. Similar to previous works [94, 19, 118, 46, 104], I follow the 2D planar-grasp representation. The gripper posits perpendicular to the ground plane, and my method outputs 2D planar grasp pose [104] which is represented with the 2D position (x, y),

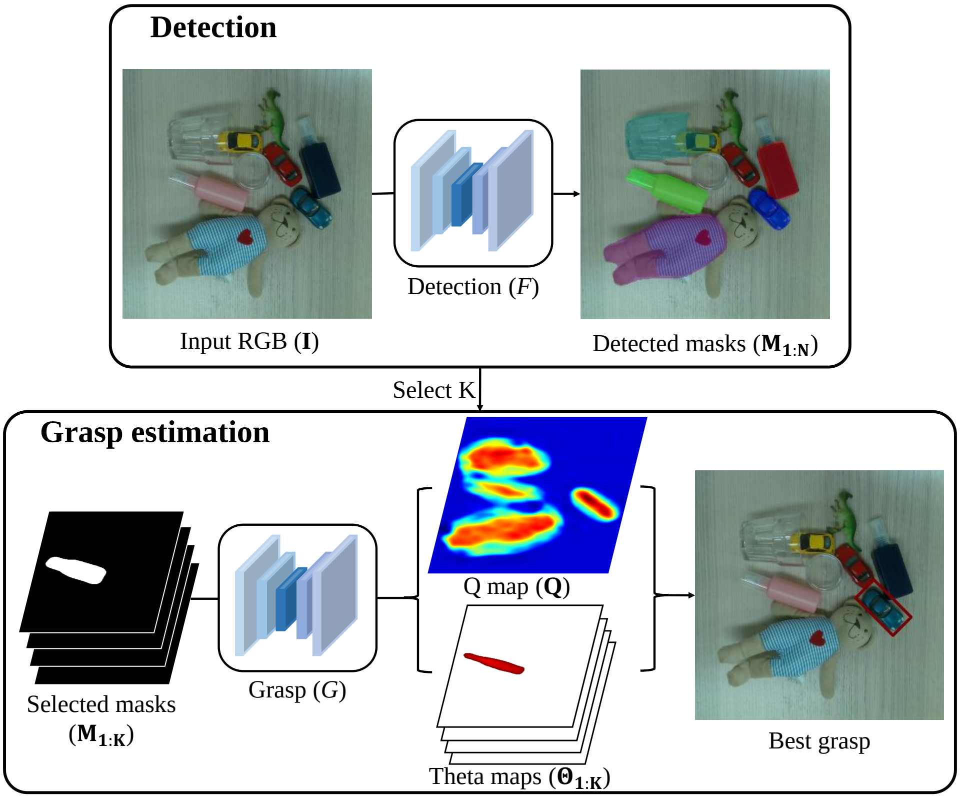

Detection
Input RGB (**I**)
Detection (F)
Detected masks ($\mathbf{M_{1:N}}$)
Select K
Grasp estimation
Selected masks ($\mathbf{M_{1:K}}$)
Grasp (G)
Q map (**Q**)
Theta maps ($\mathbf{\Theta_{1:K}}$)
Best grasp

Figure 3.2: **Overview of my approach**: From an input RGB image (**I**), the detection network (F) first segments both transparent and opaque objects into binary instance masks (**M**) [62]. The grasp estimator (G) takes in the top K confident masks and predicts one global quality map (**Q** map) and K theta maps ($\mathbf{\Theta}_{1:K}$). A single grasp is selected from the quality map and theta maps.

the angle θ, and the width w of the gripper in the image coordinate. Fig. 3.2 visualizes the overview of my approach, consisting of two convolutional neural networks. Given an input RGB image $\mathbf{I}$, the detection network F first detects instance segmentation masks $\mathbf{M}$ of N objects in the image, for both transparent and opaque objects:

$$F : \mathbf{I} \mapsto \mathbf{M}_{1:N}. \tag{3.1}$$

The detection network F is trained on my extended dataset for both opaque and transparent objects, which I discuss details in Sec. 3.1.1. I assume that the masks contain sufficient geometric cues for grasping, which I validate in the experiment.

Then after selecting top K confident instance masks, the subsequent grasp estimator G predicts a grasp quality map $\mathbf{Q}$ and theta maps $\mathbf{\Theta}_i$ for each instance i:

$$G : \mathbf{M}_{1:K} \mapsto \mathbf{Q}, \mathbf{\Theta}_{1:K}. \tag{3.2}$$

The grasp quality map $\mathbf{Q}$ encodes the probability of grasp success when the gripper's center is located at the corresponding pixel, and theta maps $\mathbf{\Theta}_i$ represents the optimal angle of the gripper at the corresponding location to pick ith object. The quality map $\mathbf{Q}$ and a theta map for each instance $\mathbf{\Theta}_i$ are both 1-channel maps. Note that the network outputs a single grasp $\mathbf{Q}$ out of K object instances. The grasp estimator G is trained to jointly consider the crowd of multiple object instances and regress the best successful grasp position and angle while avoiding collision between the gripper and the clutter of objects. I describe further details in Sec. 3.1.2.

3.1.1 Instance Mask Augmentation for Transparent Objects

One of my key ideas is to use instance masks as an intermediate representation of both opaque and transparent objects for grasping because the mask is agnostic of appearance variation. Thus, obtaining accurate instance masks for both object types is crucial to success of my approach.

However, as far as my knowledge, there exists no dataset that represents

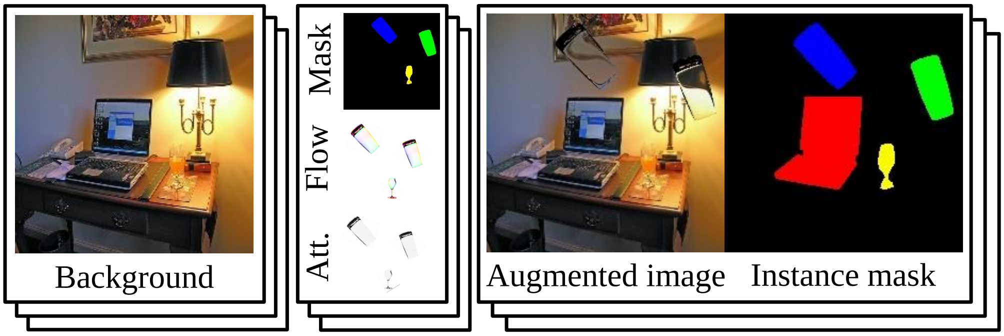


Figure 3.3: **Dataset augmentation for transparent object instance segmentation**: I build my new dataset by synthesizing transparent objects [26] on the images from the MS-COCO dataset via image matting.

both opaque and transparent objects with the same importance, or at least reflects the occurrence in the real world. Public large-scale datasets [89] mostly focus on opaque objects, prohibiting algorithms to generalize to many household transparent objects (*e. g.*, bottles, glasses, or containers). Some datasets for transparent objects only include labels of transparent objects and treat other objects as background [26, 141]. Yet, it is challenging to construct an annotated dataset for both classes on a large scale.

To build such a dataset in an efficient manner, I generate my new dataset by synthesizing transparent objects on top of an existing large-scale real-world dataset. Fig. 3.3 describes the pipeline of my dataset generation. I first sample an image $\mathbf{I}$ from MS-COCO dataset [89] with its ground truth instance segmentation $\mathbf{I}_{\mathrm{GT}}$. I then randomly select a 3D transparent object from TOM-Net [26] and build a 2D image matte: object mask $\mathbf{m}$, attenuation mask ρ, and refractive flow map $\mathbf{R}$. Then, the transparent object is synthesized using the image matting equation:

$$\mathbf{I}' = (1 - \mathbf{m}) \cdot \mathbf{I} + \mathbf{m} \cdot \rho \cdot M(\mathbf{I}, \mathbf{R}), \tag{3.3}$$

where $\mathbf{I}'$ is the synthesized image, and $M(\cdot,\cdot)$ bi-linearly warps the input image $\mathbf{I}$ given the refractive flow $\mathbf{R}$ and hallucinate the refractive effect from the transparent object. I also combine the ground truth instance segmentation mask $\mathbf{I}_{\mathrm{GT}}$ and the transparent object mask $\mathbf{m}$ to produce our new dataset with ground truth instance masks of mixed appearance. This way, I are able to construct a large-scale dataset with diverse real-world objects and abundant transparent objects with their realistic visual refraction effects, without any extra annotation cost.

Following Eq. (3.3) above, I sequentially synthesize a randomly sampled transparent object on the image $\mathbf{I}$ up to n times, where n is randomly chosen between 0 and 3. This sequential procedure allows us to realistically model the occlusion between transparent objects. Furthermore, I make sure that occluded parts in the synthesized image do not appear in the instance-wise ground truth mask. As my robotic grasping method does not require the knowledge of object class, I treat all objects as the same object class in the ground truth instance mask, mainly for simplicity. Based on my newly-built dataset, I train Mask R-CNN [62] to output instance masks for all graspable objects in the image.

3.1.2 Mask-Based Grasping Avoiding Clutter

My grasping network G is based on GG-CNN [104], which originally regresses grasp labels from a depth input. While maintaining the network architecture, I modify the network to estimate grasp labels from instance masks, by replacing an one-channel depth input with a K-channel mask image obtained by stacking top-K masks from the detection network F. I used $K = 4$ in my experiments.

Because my perceptual module distinguishes multiple objects in the scene, I can design my grasp estimator to avoid collision between the robot gripper and nearby objects when grasping a target object. I encourage my grasping

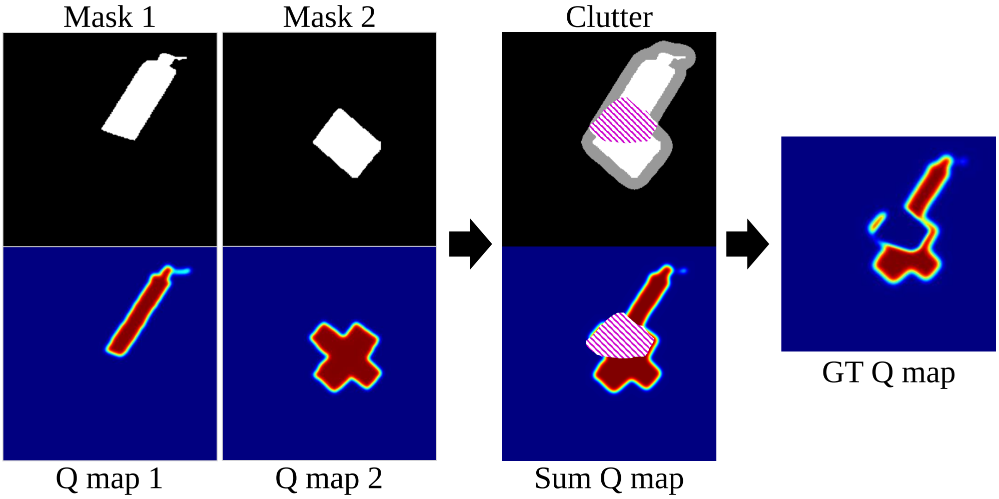


Figure 3.4: **Dataset generation for clutter avoidance**: Example of composing clutter avoidance dataset from objects and labels in the Jacquard dataset [45]. From the original mask (white), I consider clutter in the vicinity (gray) of each object. Q maps are summed in a way that excluding cluttered pixels (hatched pattern), which are replaced with the lowest success rate.

model to learn this behavior by composing the grasp labels for single objects from the Jacquard dataset [45]. I randomly select K masks and accompanied quality maps. After I aggregate the global quality maps, the quality values are attenuated by the distance from the other object masks. As shown in Fig. 3.4, I extract the cluttered pixels by identifying the pixels where the vicinity (shaded gray) of objects or masks (shaded white) overlap. For such pixels, the ground truth grasp quality map (Q map) is set to 0. Using the input K masks and the attenuated global grasp quality values, my grasping network G learns to discourage grasps in cluttered regions while encouraging grasps where other objects will not obstruct. To handle the case with smaller number of objects than K, the number of objects within a single mask-Q map pair is randomly selected between 1 and 4 leaving the remaining masks to be empty when there are less than 4 masks. I make 80 000 pairs to train my grasping module.

I further utilize instance masks for finding the grasp width. Unlike GG-CNN, which uses an extra network, I directly find the grasp width from the instance mask at the regressed grasp position and angle. The optimal grasp point $\{x^*, y^*\}$ is the position of the maximum value at $\mathbf{Q}$. I can obtain the corresponding mask index i at the location and find the optimal angle as $\theta_i^* = \mathbf{\Theta}_i(x_i^*, y_i^*)$. Given the mask of the target object with the grasp position and angle, I find two boundary points of ith mask intersecting with the line extended at the grasp position, calculate the distance between them, and use it as the grasp width. This is possible because instance masks clearly discern the region occupied by the detected objects.

3.2 Experiments

I evaluate my grasping method with a real robot without fine-tuning for novel objects at test time. I further present the instance segmentation accuracy with my proposed augmentation scheme and the grasping accuracy utilizing only mask information compared to that incorporating depth.

Implementation details. I first train the detection network (Mask R-CNN [62]) on my augmented dataset with 114 000 training images as described in Sec. 3.1.1. I use the Adam optimizer [80] and train the model for 20 epochs with the learning rate of 1×10^{-5}. I then train my grasp estimator which takes in the top-K confident masks to predict the global quality map $\mathbf{Q}$ and grasp theta maps $\mathbf{\Theta}_{1:K}$, by using my clutter avoidance scheme in Sec. 3.1.2.

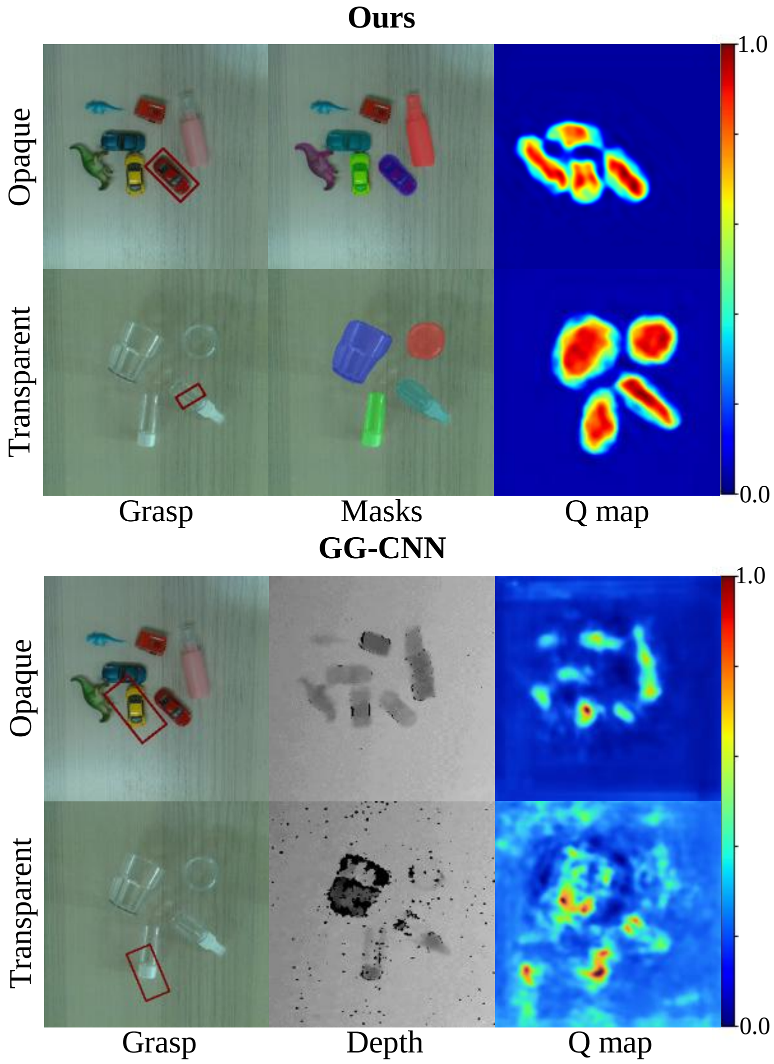
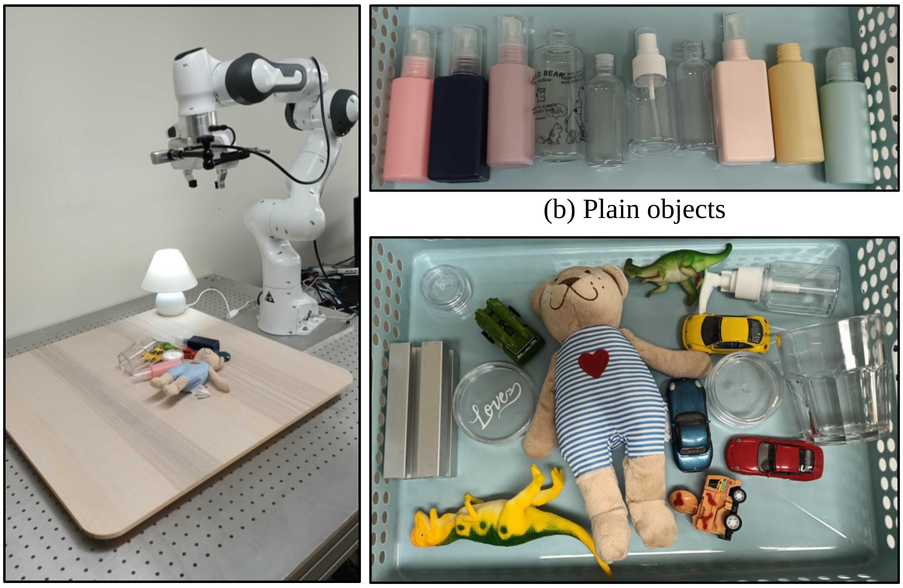


Figure 3.5: **Qualitative comparison of my method with GG-CNN [104]**: For scenes with either only opaque or transparent objects, I visualize the representations and predicted grasp quality maps predicted of my method and GG-CNN. My approach successfully tracks $K = 4$ objects and clearly demonstrates a better grasp quality map (Q map) than GG-CNN.



(a) Grasping environment

(b) Plain objects

(c) Complex objects

Figure 3.6: **Real-world test environment**: Using (a) a grasping robot, I test robotic grasping of both (b) plain objects and (c) complex objects.

3.2.1 Real-World Robotic Grasping

Experiment setup. We conduct a real-world grasping experiment using the Panda Franka robot, as shown in Fig. 3.6 (a). We attach the RealSense 435i camera to the robot gripper with calibration. We take a single RGB image from the camera and run MasKGrasp to obtain instance masks and grasping pose (*i. e.*, optimal grasp position, angle, and width in image coordinates) for each instance. Then, given the calibration parameters, we transform the predicted grasp parameters into the real-world coordinates with a fixed height and then execute the grasp. Given the target grasp pose in real-world coordinates, we first move 20 *cm* above the intended grasping point, then descend to perform the grasp. The same path planning scheme is used for all of my real-world grasping experiments.

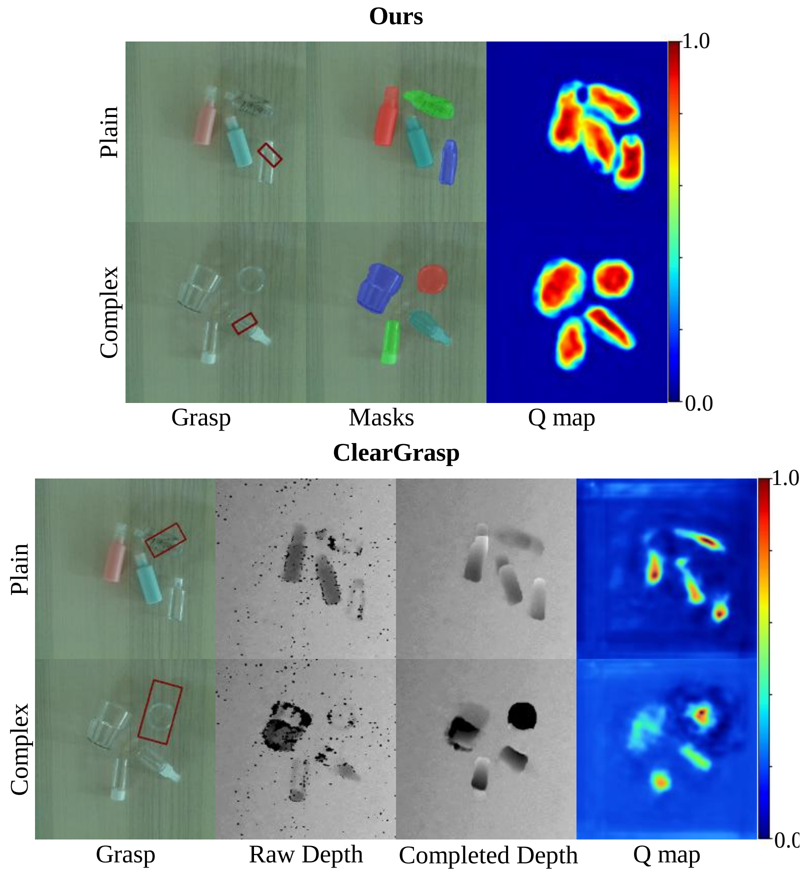


Figure 3.7: **Qualitative comparison of my method with ClearGrasp [121]**: For scenes with plain (cylindrical, simple textured) and complex (difficult geometry) objects, I visualize the representations and quality maps of my method and ClearGrasp. Comparing to ClearGrasp which uses a completed depth map for grasp estimation, my mask-based method outputs better quality map for grasping.

We compare my method with two previous methods, ClearGrasp [121] and GG-CNN [104]. In contrast to mine, both methods require a depth image as input. ClearGrasp completes a noisy input depth map especially for transparent

Configuration	**MasKGrasp (Ours)**	ClearGrasp [121]	GG-CNN [104]
Plain T.	**53.8** %	**53.8** %	38.4 %
Plain O.	53.8 %	53.8 %	**76.7** %
Complex T.	**61.5** %	46.1 %	15.3 %
Complex O.	**69.2** %	**69.2** %	53.8 %

- Plain: simple cylindrical objects with plain texture
- Complex: challenging objects with complex texture
- T: transparent, O: opaque

Table 3.1: Grasp success rates: My method outperforms previous methods especially for transparent or complex objects.

objects and passes it to a grasp estimator. Since ClearGrasp does not specify a grasping module, we use GG-CNN on the completed depth. GG-CNN here is also trained on the Jacquard dataset [45], the same as my grasp estimator.

We use 24 real-world objects for testing, which consists of 10 plain objects (*e. g.*, simple cylindrical objects with plain texture in Fig. 3.6 (b)) and 14 complex objects (*e. g.*, toys of various shapes and sizes, and transparent objects with challenging geometries in Fig. 3.6 (c)). These objects are not included in the training stages of any of the methods. For each trial, we randomly place different combination of objects in the grasping environment (Fig. 3.6 (a)). A trial is considered as a success if the robot successfully picks up any of the objects. After a single grasp attempt, the objects are once again randomly placed to maintain the number of objects within the grasping scene. For each configuration, we report the results of 13 trials.

Experiment result. Table 3.1 shows that MasKGrasp is a general solution that outperforms previous methods for grasping transparent or challenging objects with complex texture and shape. It is the only approach that achieves over the 50% success rate in all tested scenarios.

GG-CNN demonstrates the best accuracy for plain opaque objects, which is the training setup for the method. However it catastrophically fails on transparent objects due to noisy depth inputs. Fig. 3.5 qualitatively compares MasKGrasp and GG-CNN [104] on the scene that contains either opaque or transparent objects exclusively. Note that the depth measurement on transparent objects is not reliable and thus results in very noisy grasp quality map (Q map), compared to the case with opaque objects. On the other hand, my method does not use depth; thus it is free from the noisy depth measurement. My method benefits from the stable instance mask segmentation from which the geometric layout of both types of objects are observed well. As a result, my approach can produce clear Q maps for both object types.

ClearGrasp shows comparable performance for plain objects that are similar to their training examples, but does not generalize to novel transparent objects with exotic shapes and textures. In Fig. 3.7, we compare the output of my method (MasKGrasp) and ClearGrasp [121]. ClearGrasp is able to successfully complete depth (which is used for grasp estimation) for plain cylindrical objects, but it fails to generalize to more complex objects such as partially edged cups and flat jars. The main reason is the domain gap between their training data and the real-world environment with unseen, challenging objects. Because it is hard to obtain real measurements with ground truth depth, their training set consists of synthetic data with simple shapes. Also their training images are taken from a slanted viewpoint, which possibly introduces additional domain discrepancy. For better generalization to novel real-world settings, it is desirable to collect a large-scale diverse dataset; yet, it is a challenging task. On the other hand, my method leverages an existing large-scale real-world dataset with various types

Configuration	Clutter	No clutter
With clutter avoidance	**69.2**%	**92.3**%
Without clutter avoidance	25.3%	84.6%

Table 3.2: **Grasping success rates with and without clutter avoidance**: My clutter avoidance grasping substantially improves the accuracy.

of objects and thus generalize well to unseen, novel objects.

Note that the complex opaque objects in my setup include metallic objects where the raw depth measurement suffers from specular reflection. Although ClearGrasp is not specifically trained to handle such a specular noise, its depth completion module compensates the inaccuracy and leads better performance than GG-CNN with complex objects. MasKGrasp successfully generalizes to the unseen challenging objects, including specular objects and complex transparent objects, without depth information. Also the depth completion in ClearGrasp requires additional computation (roughly 1.19s) compared to MasKGrasp (about 0.001s).

Effect of clutter avoidance. We argue that the clutter avoidance in grasping module (described in Sec. 3.1.2) is crucial for a higher success rate of the algorithm. We provide an ablation study in Table 3.2. As a baseline without clutter avoidance, we train the grasping module only with the most confident mask as input. We compare the grasp success rate on both scenes with and without clutter. In the scenes with clutter, at least one edge of each object makes contact with others, while no edges touch each other in the scenes without clutter. All experiments are performed for 13 times, the same as the main experiment. While both configurations perform well on the scenes without clutter, the model with clutter avoidance substantially outperforms the ablated version

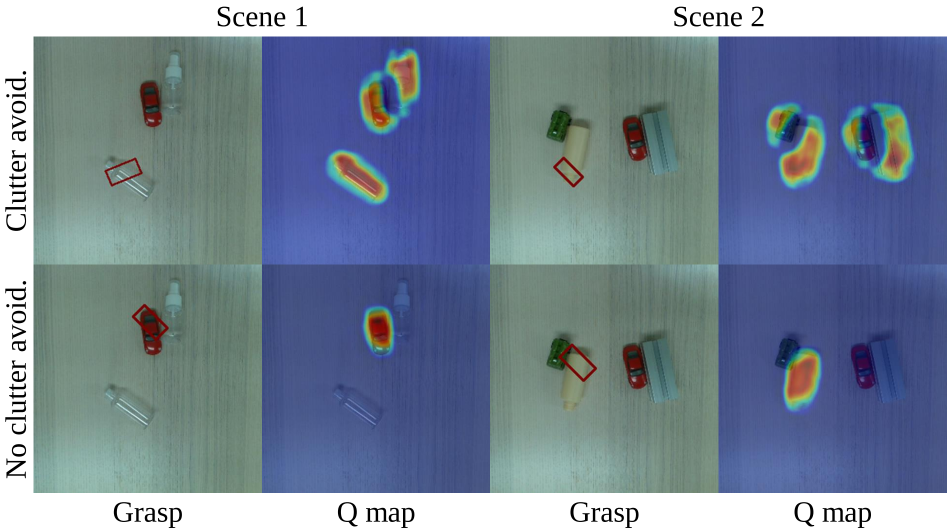


Figure 3.8: **Qualitative comparison with and without clutter avoidance grasping**: With my clutter avoidance grasping, my method outputs grasping poses that can avoid collision with multiple objects.

in the cluttered scenes.

Fig. 3.8 visualizes the effect of my clutter avoidance in the Q map. With the clutter avoidance (*i. e.*, the first row), the output quality map successfully suggests poses that can avoid collision between adjacent objects, and thus can prevent from failures during grasping. As desired, the pixels with tight inter-object distances are suppressed while the regions around isolated objects remain intact.

3.2.2 Instance Segmentation with Augmentation

We verify that my augmented dataset can train the universal instance segmentation algorithm which is applicable for multiple objects with mixed materials. We train Mask R-CNN [62] with the original MS-COCO dataset [89] (*i. e.*, baseline) and my augmented version. Then we evaluate the instance segmentation

Evaluation dataset	Method	AP_{50}	AP_{75}	IoU
MS-COCO(T)	Baseline	51.1	28.3	0.523
	Ours	**57.4**	**36.7**	**0.544**
MS-COCO(O)	Baseline	27.2	14.2	**0.358**
	Ours	**27.7**	**14.8**	0.337

Table 3.3: **Instance segmentation accuracy on multiple datasets**: We train Mask R-CNN on my dataset and MS-COCO dataset (baseline), and test the accuracy of instance segmentation mask on multiple datasets (Higher the better).

accuracy on two datasets based on MS-COCO [89]. MS-COCO(T) contains images from the validation set of MS-COCO with transparent objects, and MS-COCO(O) contains the rest of the images from the MS-COCO validation set. We use the following evaluation metrics:

- AP_{50}: AP for masks with the IoU threshold of 0.5
- AP_{75}: AP for masks with the IoU threshold of 0.75
- IoU: Average IoU of predicted and ground truth (GT) masks

Table 3.3 presents that the network trained with my proposed augmentation outperforms the baseline method on MS-COCO(T) and achieves on-par accuracy on MS-COCO(O) dataset. The result indicates that my augmentation successfully captures the visual evidence to segment transparent objects, while still sustaining accuracy for general objects.

Fig. 3.9 contains visualization of detected segmentation masks with real images of transparent objects available from the ClearGrasp dataset, my real environment, and the MS-COCO(T) dataset. While Mask R-CNN sometimes

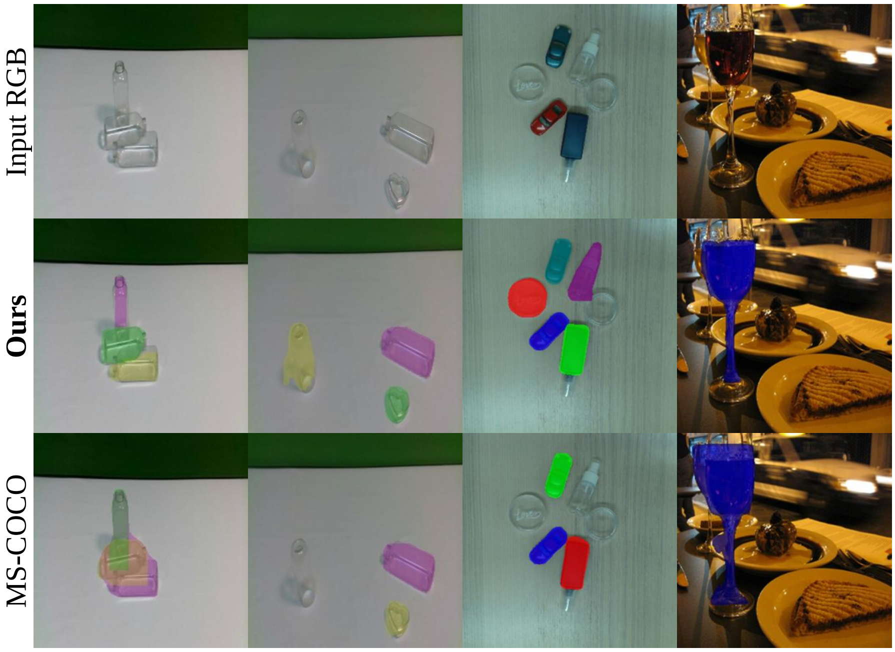


Figure 3.9: **Qualitative comparison with and without transparent augmentation**: First two columns show results from ClearGrasp [121] real testset, the third column from my robot grasping environment, and the last column from MS-COCO [89] validation set.

misses the objects or blends the instances with background, with my augmentation it is trained to successfully estimate more accurate instance masks for transparent objects. The proposed transparent augmentation achieves better results especially for transparent objects outside of the MS-COCO object categories.

3.2.3 Mask- *vs.* Depth-Based Grasping

We further validate that masks contain sufficient information for grasping compared to a conventional depth-based approach. We train two versions of GG-

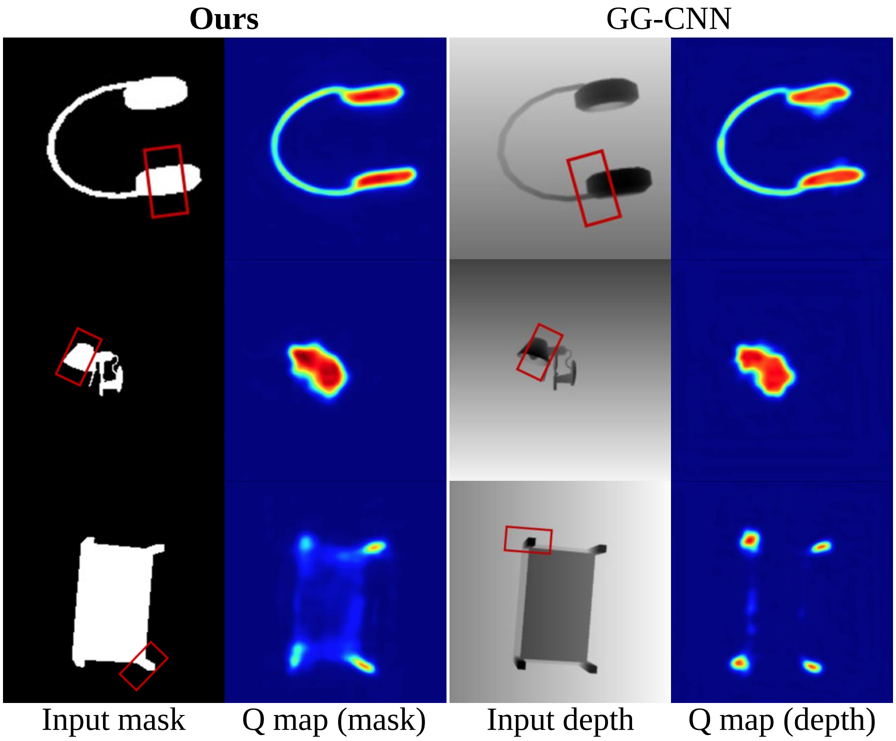


Figure 3.10: **Qualitative comparison of grasp quality maps**: I visualize the grasp quality maps, estimated from the object instance mask and depth. The output quality maps (Q map) from both inputs are visually very similar.

CNN [104] by providing either instance masks or depth maps as input, and evaluate the grasp estimation accuracy using the Jacquard Grasping Dataset [45]. We use the evaluation metric from [84], which is the same metric that GG-CNN [104] uses.

From 30 trials, my mask-based grasping achieves the accuracy of 77.5% on average, while the depth-based grasping achieves 80.6%. The accuracy drop from using the mask is only 3.85%. The qualitative results in Fig. 3.10 show that the grasp quality maps (*i. e.*, Q map) of both networks are very similar, indicating that masks carry as much information as depth in the aspect of vision-based robotic grasping. We therefore conclude that the instance masks contain sufficient information for grasping.

Chapter 4

Aggregating Multi-View Surface Normals for Grasping

Transparent objects exhibit unreliable measurements, making it difficult for robots to grasp reliably. The images of transparent objects often contain minimal visual cues, and depth cameras also miss the transparent surfaces. Recent approaches build a designated module for transparent objects using different modalities, such as thermal cameras or polarized cameras [103, 65, 71, 100], but such approaches require additional hardware. Meanwhile, for images including transparent objects, some data-driven approaches have shown promising results in 2D vision tasks (e.g., segmentation) [27, 101, 139, 88]. On the other hand, existing data-driven methods for 3D recognition for scenes including transparent objects [29] have shown less-than-desirable performance. To this end, I propose a framework for learning 3D volume given multiple RGB images including transparent objects, by exploiting the 2D recognition results of their mid-level representations (e.g., segmentation masks).

Recently, neural field representations have gained significant traction and demonstrated remarkable success across a wide range of computer vision tasks, including 3D reconstruction, novel view synthesis, and scene understanding [150, 115, 146]. These representations model spatially continuous signals using neural networks, enabling compact and high-fidelity encoding of complex scenes. One of the most prominent examples is Neural Radiance Fields (NeRF) [102], which was originally proposed to generate high-quality novel view images of 3D scenes using only a sparse set of 2D input images. While NeRF was primarily designed for view synthesis, it also implicitly encodes rich 3D geometric structure, making it attractive for applications beyond image rendering. In the NeRF framework, a neural network is trained to map continuous 3D coordinates and viewing directions to an output consisting of RGB color values and volume density σ. The volume density serves as an estimate of how much light is absorbed or scattered at a given 3D location and, as such, acts as an implicit representation of object occupancy. This capability enables NeRF to recover surface geometry indirectly, without requiring explicit 3D supervision. Building on this foundational idea, several recent works have extended NeRF to robotics applications, particularly in the context of grasping. For example, Dex-NeRF [66] and Evo-NeRF [72] adapt the NeRF architecture to model scenes containing graspable objects. In these methods, the learned volume density σ is treated as a proxy for object presence, allowing the system to reason about 3D structure and identify feasible grasp points directly from the neural representation. These NeRF-based grasping approaches offer a promising direction for robotic manipulation, especially in scenarios where conventional depth sensors struggle—such as with transparent or reflective surfaces. By learning from 2D images alone, they bypass many of the limitations associated with explicit depth sensing or mesh reconstruction.

However, I observe that raw images of transparent objects cannot directly train an accurate NeRF volume, as the volume density σ in NeRF reflects opacity rather than the actual presence of surfaces. This poses a fundamental limitation for modeling transparent materials, since, by definition, a perfectly transparent object would exhibit near-zero or zero σ values at its surfaces. As a result, the surface geometry of transparent objects becomes underrepresented or entirely absent in the learned volume, leading to failure in downstream tasks such as grasp prediction. This opacity-centered interpretation of σ severely restricts the applicability of existing NeRF-based grasping methods to transparent or semi-transparent objects. To address this limitation, I propose an alternative representation, which I term Normal Field Learning (NFL). Instead of relying on RGB intensity values, I train a neural volume from dense, pixel-wise surface normal estimates, which can be obtained from existing normal estimation models. In this framework, the focus shifts from learning radiance fields to reconstructing the surface normal field defined on the visible regions of object surfaces. To ensure that only object-related geometry is learned, segmentation masks are employed to exclude background and occlusion regions during training. By anchoring the learning process to surface orientation rather than color or opacity, the learned volume density σ becomes more aligned with actual object existence, independent of material transparency. This makes NFL a more robust and generalizable alternative for reconstructing geometry in scenarios involving transparent or complex surface materials.

My framework is designed to compensate for the inevitable inaccuracies present in surface normal and segmentation mask estimations, which are typically obtained from pre-trained networks that may not be optimized for all viewing conditions or object materials. These estimations often contain noise due to factors such as occlusion, lighting variation, and the intrinsic ambiguity of transparent surfaces. To address this, my method leverages a multi-view ag-

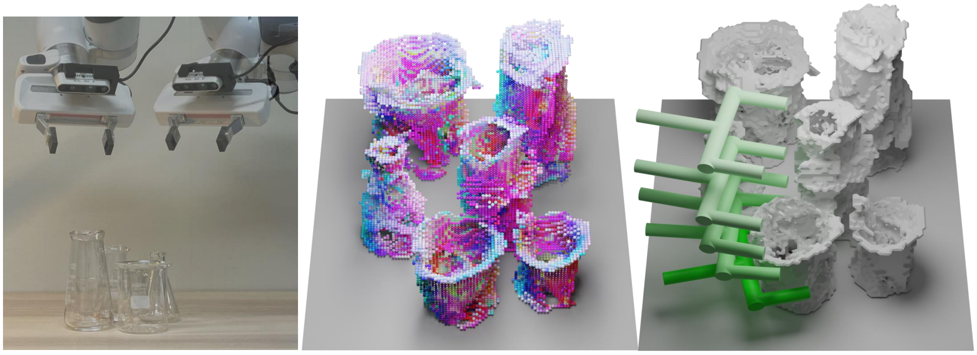

(a) RGB capture (b) Normal field (c) Grasp algorithm

Figure 4.1: Overview of NFL method. My method collects RGB images with a robot arm (a), then represents the scene as a grid-based normal field (b). I search for viable grasps via the reconstructed geometry obtained from the normal field (c).

gregation strategy, where surface normals and masks are estimated from multiple viewpoints and then fused to reinforce consistent information and suppress spurious errors. By integrating evidence across views, the framework is able to denoise and refine the input signals, allowing the training of a coherent and structured volume in a manner analogous to conventional NeRF [102] pipelines. To further enhance the robustness of the learned representation, I explicitly incorporate estimation uncertainty into the training process. Instead of treating all pixel-wise normal and mask predictions equally, I weight the supervision signal based on the confidence of each estimate. This confidence-aware learning ensures that the model gives higher importance to reliable observations while down-weighting noisy or ambiguous predictions. In particular, the uncertainty in surface normal estimation is quantified by measuring the degree of disagreement among network outputs when the input image is subjected to controlled perturbations, such as color-jittering. These variations reveal the sensitivity of the predictions and serve as a proxy for epistemic uncertainty. To model this uncertainty in a mathematically grounded way, I adopt the von Mises–Fisher distribution [55], which is the spherical analogue of the Gaussian distribution on the unit sphere S^2. The von Mises-Fisher distribution enables the probabilistic modeling of directional data, such as surface normals, while capturing both the mean direction and the concentration (confidence) around it. This allows the learning algorithm to represent and propagate directional uncertainty during training. Similarly, the uncertainty associated with segmentation masks is naturally expressed through the soft probability outputs of the segmentation network, which I model as samples from a Bernoulli distribution. These probabilistic outputs provide a continuous measure of confidence for each pixel belonging to the object class, allowing the training loss to adapt dynamically based on the certainty of each segmentation decision. By jointly modeling and leveraging these sources of uncertainty, the framework achieves a more resilient and semantically faithful reconstruction of the normal field across diverse object materials and viewing conditions.

While the vanilla NeRF is widely recognized for its high computational cost and slow performance in both training and rendering, recent advances in neural scene representation have introduced more efficient alternatives. One prominent direction is the use of feature-grid representations, such as the one introduced in DVGO (Direct Voxel Grid Optimization) [127]. These approaches replace the densely sampled continuous MLPs in NeRF with spatially discretized voxel grids, significantly reducing computation by directly optimizing learnable voxel features. Inspired by this advancement, my method adopts the DVGO architecture to accelerate the learning of the normal field, benefiting from the fast convergence and reduced memory overhead of grid-based representations. Furthermore, to tailor this approach specifically for surface normal learning, I introduce an additional optimization by removing the intermediate MLP module typically used to map voxel features to color and density values. In my context, where the goal is to represent only the surface normal field rather than full RGB radiance or volumetric opacity, this MLP becomes redundant. Instead, I directly regress surface normals from the voxel grid, resulting in a leaner architecture with fewer parameters and reduced training time. This streamlined setup not only simplifies the network but also improves stability, as it avoids the non-linearities and overfitting risks associated with deep MLPs. As a result, I am able to train a full 3D normal field for transparent objects using only 30 multi-view images in approximately 40 seconds, as illustrated in Fig. 4.1 (b). This represents a significant speedup compared to traditional NeRF pipelines that typically require hours of optimization even for opaque scenes. More importantly, the learned feature-grid-based normal field is compact, queryable in real-time, and directly usable for downstream robotic tasks. For example, I demonstrate that it can be efficiently integrated into motion planning algorithms to identify collision-free grasping trajectories, as shown in Fig. 4.1 (c). This stands in stark contrast to existing NeRF-based grasping approaches such as Dex-NeRF [66] and Evo-NeRF [72], which rely on computationally expensive volumetric rendering pipelines to synthesize multiple depth maps. These rendered outputs are then used to infer potential grasping points, adding latency and introducing redundancy. In contrast, my method leverages the direct surface normal representation stored in the voxel grid, eliminating the need for rendering altogether. This not only accelerates grasp planning but also makes the system more interpretable and adaptable to real-world robotic manipulation tasks involving transparent or visually ambiguous objects.

In summary, my contributions are as follows:

- I propose to use estimated surface normals and masks, rather than raw

RGB images, to achieve more accurate geometric reconstructions for transparent objects;

- I formulate a probabilistic framework robust to prediction errors, by taking into account the estimation uncertainty of surface normals and masks;
- My method is fast and can be directly used for grasping without the need for rendering depth images, leveraging the feature-grid representation of the volume.

My experiments display the performance and practicality of NFL in terms of reconstruction quality, speed, and the grasping success rate in real-world scenarios under significant domain discrepancies. I additionally evaluate the functionality of my algorithm on a photorealistic scene created using Blender Cycles [37].

4.1 Method

In this section, I present a probabilistic framework that learns the 3D geometric field of a scene that contains multiple transparent objects, from which we can assess reliable grasp poses. I assume that the only available observations are multiple RGB images taken from different angles with known camera poses. I do not utilize any depth image as input to my algorithm. The surfaces of objects are assumed to be smooth almost everywhere so that the surface normals are well-defined for most parts of the objects.

In NFL, the primary step is to learn normal and density fields simultaneously, where the normal field $n : \mathbb{R}^3 \to S^2$ maps a 3D point to a unit vector and the density field $\sigma : \mathbb{R}^3 \to \mathbb{R}$ maps a 3D point to a non-negative scalar. Specifically, for any point $x \in \mathbb{R}^3$ on the surface of an object, $n(x)$ is defined to be the surface normal. For any point $x \in \mathbb{R}^3$ not on the surface of an object,

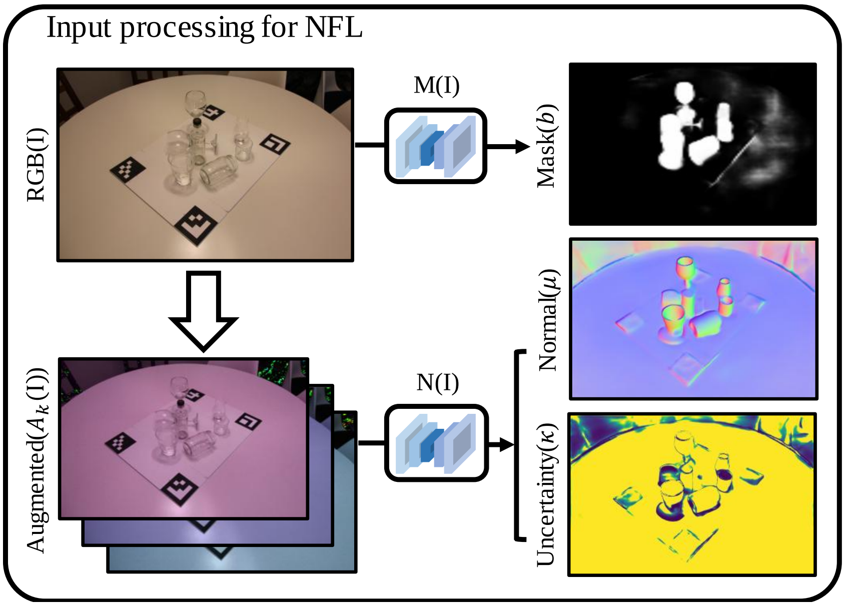


Figure 4.2: Inputs for the NFL model. The inputs for probabilistic normal field learning are the pixel-wise estimation of surface normal modeled as von Mises-Fisher distribution and estimated object mask modeled as a Bernoulli distribution.

$n(x)$ is undefined and I allow it to take any arbitrary value. This arbitrary assignment will not be problematic since my grasp pose generation only uses normal vectors on object surfaces. The density $\sigma(x)$ of a point can serve as an indicator for surface points with valid normal values. Non-zero density values indicate surface points, and zero otherwise.

In Sec. 4.1.1, I propose a probabilistic method that fits the normal and density fields $n(x), \sigma(x)$ from the RGB image set. Sec. 4.1.2 describes the grasp pose generation and motion planning algorithms based on the estimated normal and density fields.

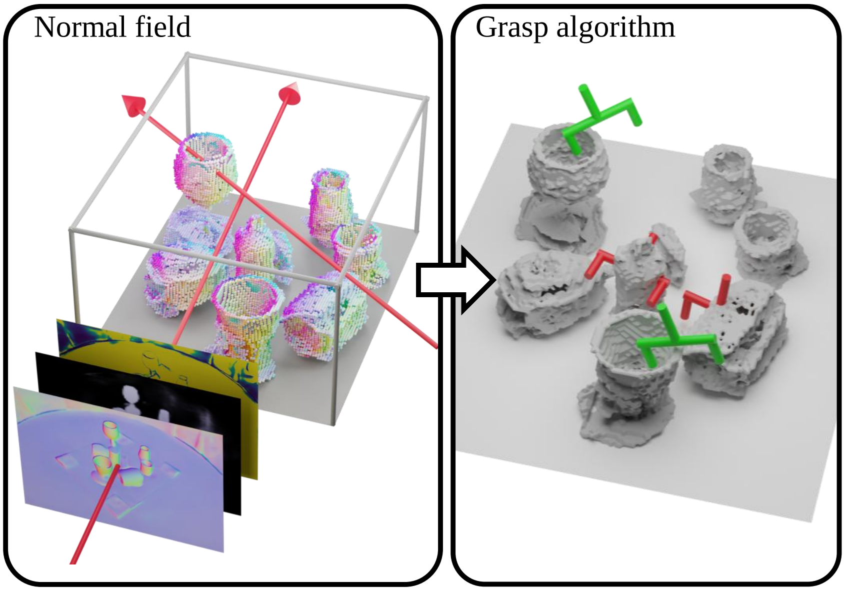


Figure 4.3: The outputs of the NFL model. I obtain a 3D normal field where each point is mapped to a normal vector n and density σ. From the normal field, I sample reliable grasps, among which I select one that can induce trajectory without collision.

4.1.1 Probabilistic Normal Field Learning Framework

My normal field adopts the standard volume rendering technique along the camera rays [70, 102]. However, I propose to learn $n(x)$ and $\sigma(x)$ with 2D mid-level representations, namely normal maps and segmentation masks estimated with pre-trained networks, instead of directly using the RGB input images. Fig. 4.2 and Fig. 4.3 illustrates an overview of my normal field learning framework. In the following sections, I propose stochastic representations of the estimated mid-level representations, and a maximum likelihood training of the normal field, where I take into account estimation uncertainties of both the normal maps and segmentation masks.

Stochastic Normal and Mask from RGB Images

I find the stochastic representation of the estimated normal maps with test-time augmentation [4]. I denote a pre-trained normal estimator by $N : \mathrm{I} \mapsto N(\mathrm{I})$ where I is an input RGB image and $N(\mathrm{I})$ is an estimated normal map. The normal vector at the (i, j)-th pixel is denoted by $N_{ij}(\mathrm{I}) \in S^2$. I consider a class of operators that transform input data $\mathcal{A} : \mathrm{I} \mapsto \mathcal{A}(\mathrm{I})$ that should not alter the outputs if N is a robust estimator. That is, $N(\mathrm{I}) = N(\mathcal{A}(\mathrm{I}))$. For instance, if I emulate subtle changes in lighting with a color-jittering transformation, the estimated shape should remain constant. However, pre-trained models often fail to remain invariant under those transformations; I use the extent of deviations as a measure of estimation uncertainty.

Let $\mathcal{A}_k$ be a normal-preserving transformation operator for $k = 1, \ldots, m$ (including the identity map), as discussed earlier, and consider m normal estimates of an image I, $\{N(\mathcal{A}_k(\mathrm{I}))\}_{k=1}^m$. For each (i, j)-th pixel, there are m estimated normal vectors $\{N_{ij}(A_k(\mathrm{I})) \in S^2\}_{k=1}^m$. By using these estimates, I fit a continuous probability density function for each pixel of the normal map.

I use the von Mises-Fisher distribution [55] as a density model for $N \in S^2$:

$$f(N; \mu, \kappa) := \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} \exp(\kappa \mu^T N), \tag{4.1}$$

where f is a probability density function, $\mu \in S^2$ is the mean direction parameter, and $\kappa \in \mathbb{R}$ is the concentration parameter. The greater the value of κ, the higher the concentration of f around μ, and the lower the uncertainty of N. For each (i, j)-th pixel in RGB image I, excluding the background regions, the Maximum Likelihood Estimates (MLEs) of the parameters can be computed as follows. The MLE of the mean parameter is simply given as $\mu_{ij}(\mathrm{I}) = \bar{N}/\|\bar{N}\|$ where $\bar{N}$ is the arithematic mean $\bar{N} := \frac{1}{m}\sum_{k=1}^m N_{ij}(\mathcal{A}_k(\mathrm{I}))$. On the other hand, the MLE of the concentration parameter has no closed-form expression, yet instead, a simple approximation to $\kappa_{ij}(\mathrm{I})$ is available [9]:

$$\kappa_{ij}(\mathrm{I}) = \frac{\|\bar{N}\|(3 - \|\bar{N}\|^2)}{1 - \|\bar{N}\|^2}. \tag{4.2}$$

In addition, I find the stochastic representation of the estimated segmentation masks using a pretrained mask estimator $M : \mathrm{I} \mapsto M(\mathrm{I})$. Let the estimated segmentation mask at the (i, j)-th pixel be denoted by $M_{ij}(\mathrm{I}) \in [0, 1]$. I then interpret each pixel of the segmentation masks as the Bernoulli distribution with a parameter $M_{ij}(\mathrm{I})$, denoted by $\mathrm{B}(1, M_{ij}(\mathrm{I}))$, since my segmentation network is trained with the cross entropy loss.

Maximum Likelihood Normal Field Learning

Given the stochastic representations of the normal maps and segmentation masks, I can formulate the normal field learning as a variant of maximum likelihood training with the differentiable volume rendering [70].

Let $r_{ij}(\cdot; \mathrm{I})$ be a ray emitted from the camera that passes through (i, j)-th pixel of an image I. I accumulate $n(x)$ and $\sigma(x)$ along a ray $r_{ij}(t; \mathrm{I})$ with near and far bounds t_n and t_f, and define a projected normal map as

$$N_{ij}^{\mathrm{proj}}(\mathrm{I}) := \mathrm{Normalize}\Big(\int_{t_n}^{t_f} T(t)\sigma(r_{ij}(t; \mathrm{I}))n(r_{ij}(t; \mathrm{I}))dt \Big), \tag{4.3}$$

where Normalize$(\cdot)$ maps a vector to a unit vector and $T(t) = \exp(-\int_{t_n}^{t} \sigma(r_{ij}(s; \mathrm{I}))ds)$ is the accumulated transmittance along the ray. The projected normal map has a dependency to $n(x)$ and $\sigma(x)$, so it may be better to write as $N_{ij}^{\mathrm{proj}}(\mathrm{I}; n, \sigma)$, but I omit n and σ for notation convenience.

I then define a per-pixel loss function for an (i, j)-pixel of an input image I as the negative log-likelihood that measures how unlikely the projected normal map $N_{ij}^{\mathrm{proj}}(\mathrm{I})$ is, given the probability density function of the estimated normal map $f(N; \mu_{ij}(\mathrm{I}), \kappa_{ij}(\mathrm{I}))$, as follows:

$$l_{ij}(\mathrm{I}) := -\log f(N_{ij}^{\mathrm{proj}}(\mathrm{I}); \mu_{ij}(\mathrm{I}), \kappa_{ij}(\mathrm{I})). \tag{4.4}$$

Ignoring the normalization constant that does not depend on both n and σ, the per-pixel loss further simplifies to

$$l_{ij}(\mathrm{I}) = -\kappa_{ij}(\mathrm{I})\mu_{ij}(\mathrm{I})^T N_{ij}^{\mathrm{proj}}(\mathrm{I}). \tag{4.5}$$

By minimizing the loss, the projected normal $N_{ij}^{\mathrm{proj}}(\mathrm{I})$ is fitted to $\mu_{ij}(\mathrm{I})$ – since the inner product of two unit vectors is maximal when they are equal – with the weight of $\kappa_{ij}(\mathrm{I})$. Higher weights are assigned to pixels with more certain normal estimations, i.e., those with higher values of $\kappa_{ij}(\mathrm{I})$.

Although it is tempting to sum $l_{ij}(\mathrm{I})$ over all the indices i, j to define the final loss function, it is unnecessary to take into account l_{ij} for the background pixels where no object exists. I use the stochastic representation of the segmentation mask to minimize l_{ij} only when (i, j) pixel is an object pixel. Specifically, I sample $b_{ij}(\mathrm{I})$ from the per-pixel Bernoulli distribution $\mathrm{B}(1, M_{ij}(\mathrm{I}))$, and consider the product $b_{ij}(\mathrm{I})l_{ij}(\mathrm{I})$ as a new loss term. Therefore, when $b_{ij}(\mathrm{I}) = 0$ (i.e., (i, j) belongs to background pixels), the loss will be ignored.

Additionally, it is important to learn accurate σ since I use density values in practice to distinguish between object and non-object regions. Up to this point, my attention has been on the normal field component for object pixels, i.e., when $b_{ij}(\mathrm{I}) = 1$, and the loss is not sufficient to learn the correct σ. I therefore introduce a density penalization term $(1 - 2b_{ij}) \int_{t_n}^{t_f} \sigma(r_{ij}(t; \mathrm{I}))dt$ into the loss function. For an object pixel $b_{i,j}(\mathrm{I}) = 1$, minimizing the loss encourages the accumulated density σ to maintain a positive value. When $b_{i,j}(\mathrm{I}) = 0$, or it is a background pixel, the loss effectively suppresses the density along the ray $r_{ij}(t; \mathrm{I}), t \in [t_n, t_f]$ to be zero.

In summary, the loss function for an image I is:

$$\mathcal{L}(\mathrm{I}; n, \sigma) := \sum_{i,j} b_{ij}(\mathrm{I}) l_{ij}(\mathrm{I}) + (1 - 2b_{ij}(\mathrm{I})) \int_{t_n}^{t_f} \sigma(r_{ij}(t; \mathrm{I})) dt, \tag{4.6}$$

where $b_{ij}(\mathrm{I}) \sim \mathrm{B}(1, M_{ij}(\mathrm{I}))$. And, given a set of images $\{\mathrm{I}^{(l)}\}_{l=1}^{L}$, the final loss function for the normal and density field is the empirical mean of losses for input images:

$$\mathcal{L}(n, \sigma) := \frac{1}{L} \sum_{l=1}^{L} \mathcal{L}(\mathrm{I}^{(l)}; n, \sigma). \tag{4.7}$$

4.1.2 Grasping Algorithm Based on Normal and Density Grids

After I obtain the 3D geometric layout parameterized by normal and density functions, I can regress for 6 DoF grasp positions and collision-free trajectories as shown in Fig. 4.3 (right). Since the process to train normal and density values is similar to conventional NeRF formulation [102], I can accelerate the training by employing discrete voxel grid representations as suggested by recent works [127]. Fast speed is particularly useful where robot concurrently observes the scene and grasps an object. Furthermore, my formulation can avoid volume rendering to find the surface points and their normals, and estimate them directly from individual grid points. Note that the original voxel-grid implementation trained with color images stores feature vectors on the grid and uses an additional shallow MLP to regress for the color values. However, my approach heavily utilizes the density and normal grid, where the grid points contain the raw density and normal values. The direct access of grid representation enables us to quickly find feasible grasping points, and generate collision-free paths.

6-DoF Grasp Candidate Generation

The density and normal grids provide surface point positions and their surface normal vectors, respectively, which can be directly used to find feasible grasping points for a two-finger gripper [97, 64]. The grasp candidate generation algorithm is presented in Algorithm 1 From a set of 3D points on the density grid, I extract a subset of points $\{x_1, \ldots, x_N\}$ that have density values higher than a threshold value δ_{density}, which represent points on the surface. Denote the corresponding normal vectors for those surface points by $\{n_1, \ldots, n_N\}$, obtained from the normal grid. Then, I first find a set of index pairs (i, j) that satisfies two conditions: (i) $|x_i - x_j| < \delta_{\text{dist}}$ with some distance threshold δ_{dist} defined considering the gripper width and (ii) $n_i \cdot (x_i - x_j) \geq 0.99$ to find antipodal points. I denote the set of these index pairs S, which serves as the candidate grasp points.

Algorithm 1 Algorithm for Two-Finger Grasping

Input

- x: point in space,
- $n(x)$: normal field at x

Algorithm

```
$C = \{x_1, x_2, ..., x_N\}$
$S = \{\}$
**for** $i, j \in [1, N]$ **do**
    **if** $|x_i - x_j| \leq dist.thresh$ **then**
        $valid_i = (n(x_i) \cdot (x_i - x_j) \geq 0.99)$
        $valid_j = (n(x_j) \cdot (x_j - x_i) \geq 0.99)$
        **if** $valid_i \wedge valid_j$ **then**
            $S \leftarrow \{i, j\}$
**return** $S$
```

Collision-Free Path Planning

Given the set S of candidate grasp points, I find a robot configuration and path that grasps the object while avoiding collisions with the surrounding environments. The collision against 3D scene layout is approximated by comparing against the set of surface points $\{x_1, \ldots, x_N\}$ which are already extracted. Ideally, looking at all of the candidate grasps in S would lead to better performance. However, in order to expedite the process of selecting grasps, I sort the index pair set S with the density score $\sigma(x_i) + \sigma(x_j)$ for $(i, j) \in S$, and start from the one with the highest density score. I examine the top 100 pairs from grasp candidates in practice. For each candidate grasp pair, I test 8 pitch angles for a gripper and search for the configuration that does not collide with any of surface points. Then I find the joint trajectory of a robot that arrives at the target pose without collision. I use PyBullet planning library [41] for the collision detection and path planning.

4.2 Experiments

In this section, I compare my NFL-based 3D reconstruction method and 6-DoF grasping algorithm with existing RGB image-based 3D reconstruction methods and depth rendering-based grasping methods. In Sec. 4.2.1 I compare geometry reconstruction results, and in Sec. 4.2.2 I compare the grasping perfomance in the real world.

Baselines. I select baselines for comparison that satisfy two conditions. First, baselines should take as input multiview RGB images along with their camera parameters. Second, baselines should be trained solely by real-world data, since my main target is grasping in complex configurations in the real world. The compared baselines are: NeRF [102], DVGO [127], Dex-NeRF [66], Dex-DVGO and GraspNeRF [43]. [1]. All of them train the color and density fields directly from RGB images. NeRF uses neural networks to represent the fields, whereas DVGO employs voxel grid representations followed by a shallow MLP. Dex-NeRF applies a threshold on the density values to better capture transparent objects; I implement the same technique for DVGO and denote it by Dex-DVGO. For DVGO and Dex-DVGO, I retain the MLP in the original implementation of DVGO, since they could not converge without retraining for transparent objects.

Implementation Details. I estimate surface normals and masks given RGB images by neural networks trained with the large-scale real-world dataset [32]. For surface normal estimation, I finetune the work by Bae et al. [4] for 20000 steps. To predict segmentation masks, I train a CNN based model [30] for 20 epochs. For test-time augmentations, I employ color jittering transformations (hue transformations) provided by the Torchvision library. I use one original and nine augmented images to fit von Mises-Fisher distribution on pixel-wise normals.

I train NFL on an RTX 3090 for 5000 steps with a grid resolution of 150^3 for my real scene. The bounding box dimensions are 50cm × 60cm × 40cm and it is positioned to enclose the robot's workspace. It takes around 40 seconds to train a normal field.

4.2.1 3D Scene Reconstruction: Synthetic and Real

First, I provide a quantitative evaluation of geometry reconstruction using synthetic scenes with transparent objects. I create a photo-realistic rendering of a scene with 3 objects of glass textures on top of a wooden table using Blender Cy-

[1] ClearGrasp [120] is not included since ClearGrasp uses only a single image. Although Evo-NeRF [72] satisfies both conditions, I are unable to find the custom grasping dataset that is necessary to implement the algorithm.

cles [37]. I train all models until convergence given 100 input images. I compare the accuracy of rendered depth images on three metrics from ClearGrasp [120]. Specifically, I render test-view depth images from viewpoints equally spaced on a cylinder bounding the objects. Accuracy is defined as the ratio of object pixels where the error is within a threshold. Compared to RMSE, this metric is agnostic of scene scale. The threshold is selected as 5%, 10%, and 25% of the groundtruth depth, as in [120].

Table 4.1: Depth reconstruction results on Blender dataset. Bold represents best results.

Config	Grid-based (DVGO)			Non grid-based (NeRF)	
Method	**NFL**	Dex-DVGO	DVGO	Dex-NeRF	NeRF
$\delta_{0.05}$	**85.35**%	20.48%	19.13%	74.96%	27.17%
$\delta_{0.10}$	**92.64**%	29.25%	38.02%	81.39%	56.25%
$\delta_{0.25}$	**97.49**%	46.50%	82.47%	95.15%	93.81%
Time (min.)	**2**	15	15	720	720

Table 4.2: Depth reconstruction accuracy depending on input modality. On both grid-based and non-grid-based methods

Config	Grid-based (DVGO)				Non grid-based (NeRF)			
Modality	Normal	Mask	N + M	RGB	Normal	Mask	N + M	RGB
$\delta_{0.05}$	11.48%	90.57%	**96.85**%	19.13%	83.01%	89.94%	**93.40**%	27.17%
$\delta_{0.1}$	72.30%	94.18%	**97.52**%	38.02%	92.26%	94.33%	**96.97**%	56.25%
$\delta_{0.25}$	**99.10**%	97.78%	98.17%	82.47%	99.66%	98.72%	**99.93**%	93.81%

- N + M: normal + mask

Table 4.1 contains the quantitative results on the depth accuracy. My model, NFL, marks the best accuracy in all of the depth accuracy metrics while taking less time than others. Dex-NeRF and NeRF are the vanilla representation that utilize a single MLP to represent the entire scene, and take about 12 hours. The grid-based acceleration shortens the training time of Dex-DVGO and DVGO into 15 minutes. NFL further accelerates the time into 120 seconds by removing MLPs, which are required to synthesize novel view images for Dex-DVGO and

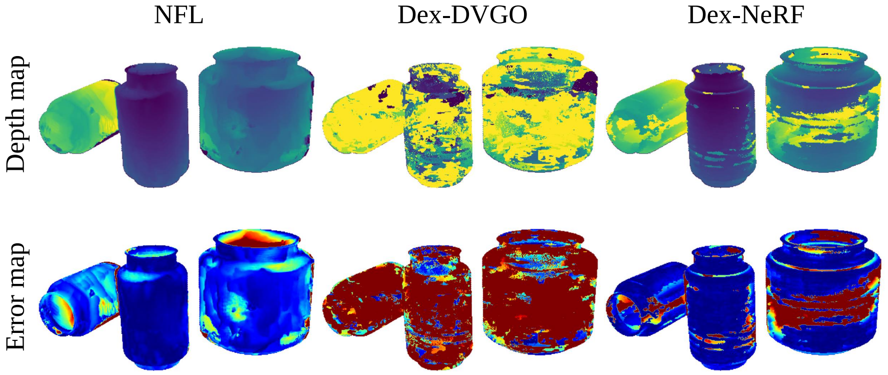


Figure 4.4: Qualitative results on synthetic data. Top row shows rendered depth for object pixels. Bottom row depicts error maps with respect to groundtruth depth (red: high error, blue: low error). My model captures more accurate depth of all objects.

DVGO. I also observed that depth rendering technique of Dex-NeRF improves the accuracy of geometry compared to NeRF.

The qualitative results are presented in Fig. 4.4. The top row shows the depth map for the object pixels, whereas the bottom row shows the error map relative to the groundtruth depth. For the error map, red pixels indicate higher error while blue pixels indicate more accurate depth measurements. My method reconstructs more accurate depth for most of the object pixels while Dex-DVGO fails to capture geometry. Dex-NeRF performs reasonably except the middle part of the large bowl, where the object appears more transparent and lacks visual evidence in RGB images.

Ablation on Input Modality. I verify that normals and masks are more effective to reconstruct the geometry of transparent objects compared to directly using RGB images. Similar to the previous experiment, I test on a scene with 3 objects with 100 images as input and compare the reconstruction results in Table 4.2. Given the same set of images and the camera parameters, the

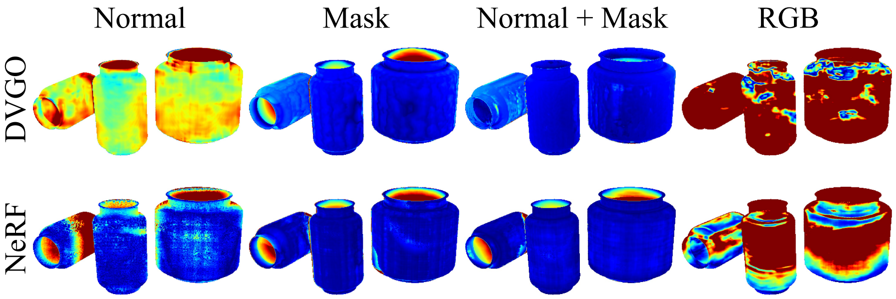


Figure 4.5: Error maps of depth obtained from different input modalities (red: high error, blue: low error). For both grid-based (DVGO) and non-grid-based (NeRF) methods, RGB input cannot accurately reconstruct depth for transparent objects. Using both normal and mask leads to the best results. Grid-based method (DVGO) also struggles to capture geometry when using only normals for an input.

reconstruction is significantly more effective with surface normals and masks than with RGB images. Although all combinations show better reconstruction performance compared to RGB input, using normals and masks together demonstrates the best performance in both grid based (DVGO) and non-grid based (NeRF) approaches. The error maps in Fig. 4.5 show similar results. The normal maps without masks are not sufficient to reconstruct accurate geometry. Masks alone cannot accurately capture the concave parts of the bottle.

Table 4.3: Effects of mask sampling and stochastic normals

Uncertainty	None	Mask Sampling	Stochastic Normal Mask Sampling
$\delta_{0.05}$	84.71%	85.25%	**85.35**%
$\delta_{0.10}$	**92.85**%	91.27%	92.64%
$\delta_{0.25}$	97.24%	96.25%	**97.49**%

Effect of Considering Input Uncertainty. While normals and masks are useful in training the field to obtain geometry of transparent objects, the estimations can be erroneous. NFL employs probabilistic formulation as described in Sec. 4.1.1 to consider the uncertainty in the estimated inputs. Stochastic normal incorporates the distribution of normal estimation from test-time augmentation, and it is ablated by considering all rays equally in Eq. (4.6). Mask sampling can be ablated by using binary values for $b_{ij}(\mathrm{I})$ after thresholding. Table 4.3 shows that using both stochastic normal and mask sampling records the best results.

Table 4.4: Real world grasp success rates for several configurations: single small, single big, and clutter.

Model	**NFL**	GraspNeRF [43]	Dex-NeRF[66]	Dex-DVGO	NeRF [102]	DVGO [127]
S.S.	**71.43**%	14.28%	28.57%	0%	0%	14.28%
S.B.	**57.14**%	14.28%	0%	0%	0%	0%
C.	**85.71**%	28.57%	28.57%	0%	14.28%	0%
Time	40 sec	**90ms**	12 hr	15 min	12 hr	15 min

- S.S.: single small
- S.B.: single big
- C.: clutter

Robustness Across Scenes. My normal field aggregates normal estimates from pre-trained networks and shows robust performance across challenging appearance variations. Fig. 4.6 compares the reconstructed geometry of my model and two baselines in different representations. While NFL builds the 3D normal field, baselines use depth maps rendered from the learned neural volume to obtain grasp points. NFL successfully builds normal fields for all cases using the same setup despite the variation in objects, lighting, camera parameters and more. The input from my real world scene (left) is especially challenging, containing fewer visual cues (edges of transparent objects) compared to other datasets. In contrast, Dex-NeRF and Dex-DVGO are sensitive to the appearance or lighting of the transparent objects and could not render an accurate depth map from our real-world images. The dataset of Dex-NeRF exhibits considerable visual evidence, such as stark edges of objects, although the objects are transparent. GraspNeRF utilizes only 6 images and has a fast prediction

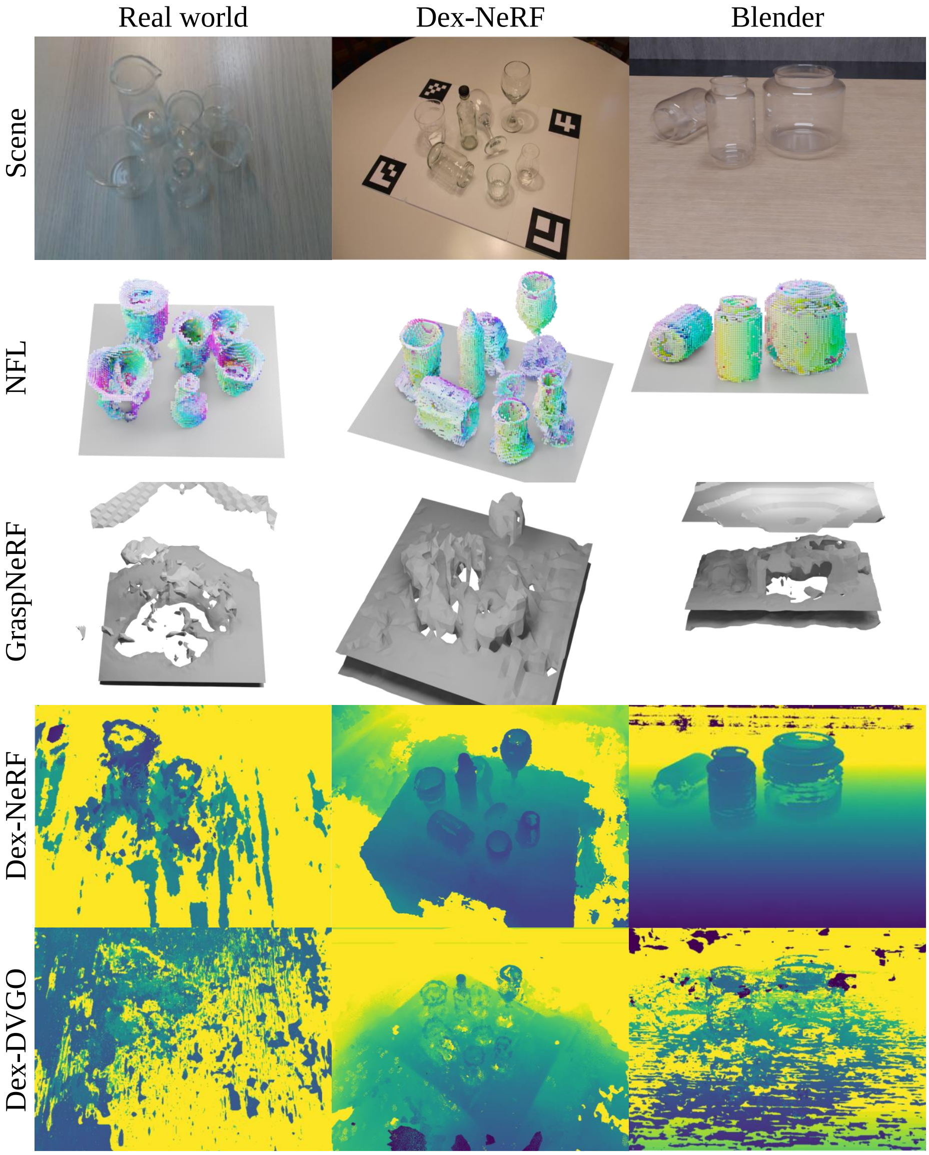


Figure 4.6: Robustness across scenes. I visualize the geometries different methods use for grasping (normal field for NFL, depth image for baselines). My method stably creates normal fields for real world, Dex-NeRF, and blender scenes.

time, but fails to accurately capture the geometry of the objects. In order to successfully run GraspNeRF, I need to find the six viewpoints that are similar to the original implementation. In addition, I re-scale the scene to fit in the 30cm cube originally used in GraspNeRF. After these measures, GraspNeRF captures the ground stably. However, GraspNeRF's reconstruction performance fluctuates depending on the scene. Dex-NeRF and Dex-DVGO directly use color images and only obtain the volume density σ for such opaque appearances out of transparent objects.

4.2.2 Real Robot 6-DoF Grasping

I use a real-world robot to capture input images to acquire geometric layout and perform grasping tasks. I attach a Realsense d435i camera at the end effector of a Panda Franka Emika robot using a 3D printed mount, and use only RGB images for input. I calculate the camera poses using the end effector location and the relative transformation between the mounted camera and the end effector. I utilize 30 images for my model and baselines other than GraspNeRF. Since GraspNeRF is a pretrained network as a whole, adjusting the number of utilized images is not straight-forward. Thus for GraspNeRF, I match the 6 viewpoints utilized in the original GraspNeRF paper. My image capturing system takes up to 1 second to move and capture to each viewpoint. To assist placing the objects in the same configuration for different methods, I built a GUI that overlays object positions from the previous observations. For grasping baselines other than GraspNeRF, I render a depth image from the view looking straight down at the objects as in Dex-NeRF. Then I calculate the best top-down grasp points using the model from Dex-Net [93], which is pretrained for two-fingered grippers. I move the gripper 20 cm above the grasping point then lower it to grasp. For GraspNeRF, I utilize the pretrained model which predicts the neural field and grasping end effector pose. I follow the Gaussian smoothing process, then select the grasp with the highest quality value. Each trial is classified as a success if the robot successfully picks up an object and places it into a bin.

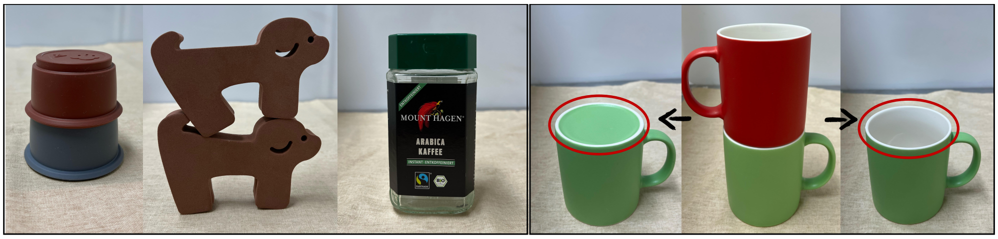
Table 4.4 contains the grasp success rates after seven grasps for three different scene configurations: Single Small, Single Big, and Cluttered. The Big and Small are assessed based on the relative size compared to the gripper width, which reflects the grasping difficulty. For Cluttered scenes, I put six objects within a 30 cm × 30 cm square region. While my model shows good performance on all three configurations, baselines struggle to effectively grasp objects, even with more training time. Specifically, NFL excels in the cluttered scene thanks to a rich set of candidate grasps obtained from accurate holistic reconstruction. All of the baselines struggle on the Single Big scene. In Single Big configuration, the thickness for a candidate grasp is comparable to the width of an open gripper, and therefore I need to find the precise grasp location from accurate geometry. In contrast NFL succeeds in grasping over 50% of single big scenes, indicating the superior reconstruction accuracy of NFL. GraspNeRF and Dex-NeRF show the second best performance. Especially, GraspNeRF succeeds for at least one experiment for all configurations, with the least inference time. Dex-DVGO, NeRF, and DVGO fail in my grasping experiments. GraspNeRF marks the least time to build a neural field. Different to the Blender dataset experiment of Table 4.1, my method takes 40 seconds to train. This is because I use less images as input (100 vs 30), which allows shorter training time.

Chapter 5

Interaction-Based Perception

With recent advances in vision-based algorithms, many robot perception tasks can be accomplished with image input only. Some learning-based methods can predict both object 3D geometry and instances simultaneously [78, 79]; however, these approaches rely on simplified shape representations, such as superquadrics, and are limited to known shape categories, which hinders performance on out-of-distribution data. Recent methods based on Neural Radiance Fields (NeRFs) and their variants [102, 66, 83] can capture the 3D geometry of unknown objects using multi-view RGB images, overcoming the limitations of data-specific learning and generalizability issues. By leveraging 2D segmentation masks from a pre-trained model like Segment Anything Model (SAM) [81], the multi-view information can be aggregated to jointly estimate object geometry and instances [14, 74], even on novel, unknown objects.

However, a single-session visual observation of a scene inevitably introduces ambiguities that challenge accurate instance and geometry identification. In cluttered scenes, distinguishing individual object instances – by an *object in-*



(a) Instance ambiguity

(b) Geometric ambiguity

Figure 5.1: Instance and geometric ambiguity. Distinguishing whether two objects move together or separately also shows ambiguity (a). Unobservable surfaces induce geometric uncertainty (b).

stance, I refer to a physically connected component within a scene that moves as a single unit – becomes particularly difficult when objects are in contact. Moreover, occlusions obscure surfaces, limiting the ability to recognize complete geometry. Fig. 5.1 highlights these ambiguities and a common failure mode in existing NeRF-based methods: without additional information, it is unclear whether attached objects, such as those with similar textures or a cup's lid and body, move together or separately (instance ambiguity). Additionally, contact surfaces remain hidden, preventing accurate geometric reconstruction (geometric ambiguity). A natural solution is to design a loop that enables the robot to interact with objects, then gather more information about the scene. Here, this loop should incorporate features to efficiently reflect change in the scene, since instance and geometric identification from scratch is an expensive operation.

To this end, I propose a novel *Interact-to-Identify* framework that: (i) enables the robot to autonomously incur change in the scene, (ii) after the interaction, quickly identifies the instances, and (iii) refines geometry without requiring extensive additional data capture or computation. As a result, my method can acquire disambiguated information on both instance and geometry, benefiting downstream tasks such as manipulation [108]. My core approach involves constructing a 3D geometric estimate along with an instance candidate tree using three levels of SAM's mask prediction results. This tree representation offers discrete levels of granularity to quickly identify instance ambiguity within the visual observations. Based on each object's ambiguity level, my rule-based algorithm generates waypoints, enabling the robot to interact with the environment and induce changes in the scene. Finally, with only a few sparse additional observations from altered scenes, my method can efficiently identify accurate instances based on the rigid-body assumption for each object and quickly refine object geometry.

My extensive experiments demonstrate the effectiveness of my methods in both simulated and real-world scenarios, validating, for the first time, a framework capable of rapidly identifying object instances and geometry of unknown 3D objects in cluttered scenes through interaction. With these recognition results, the robot can effectively perform diverse manipulation tasks that require instance and 3D geometry information, such as sequential grasping, where instance information allows instant removal of grasped objects from the reconstructed scene as well as efficient geometric finetuning of newly visible surfaces. My contributions can be summarized as follows:

- A framework for simultaneous recognition of object geometry and instances through interaction;

- A formulation to estimate and exploit change given few observations after the change to handle instance ambiguity;

- Instance-wise geometric finetuning utilizing a novel visibility-based uncertainty metric to quantize geometric uncertainty.

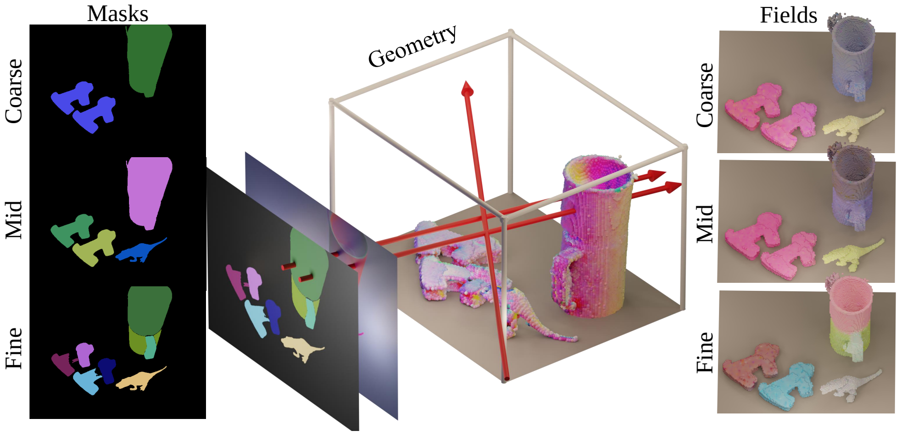


Figure 5.2: Process to learn field. From coarse, mid, fine granularity masks and normal images, I learn three fields each reflecting the granularity on shared geometry.

5.1 Initial Field Training

I employ *normal*, *density*, and *feature field* representations of a 3D scene for simultaneous object instance and geometry identification. Building on prior work [83], I use a surface normal field $n(x)$ and a volume density field $\sigma(x)$ for $x \in \mathbb{R}^3$ to capture object geometry[1]. In this work, inspired by [14], I further incorporate a feature field $F(x)$ – which outputs a multi-dimensional *feature* vector – defined on surface points to encode instance-related information. By clustering in this feature vector space, I can then identify object instances.

While my main contribution is an efficient method for rapidly updating the initial field representations using a few sparse additional observations (discussed in the next section), in this section, I first describe how to *initialize* these representations from multi-view, relatively dense RGB images, following previous works [83, 86, 14]. This section consists of the following three subsec-

[1] The density field $\sigma(x)$ assigns zero values to regions outside the surface, while the normal field $n(x)$ is defined only on object surface points.

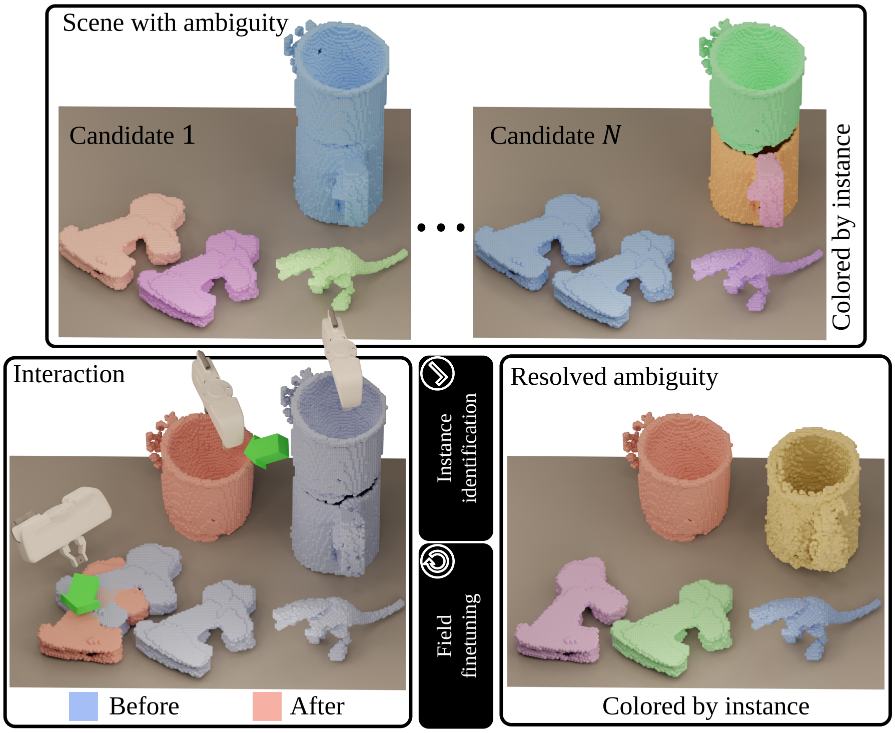


Figure 5.3: Overview of Interact-to-Identify. For a scene with geometric and instance ambiguity, I perform interaction via a robot arm. From few new observations, I resolve both types of ambiguity resulting in simultaneous reconstruction and instance identification.

tions: (i) input preprocessing, (ii) normal and density fitting loss (with my novel contribution on density-normal consistency loss), and (iii) feature learning loss.

5.1.1 Input Preprocessing

Given raw multi-view RGB images with camera poses, I preprocess these images to generate mask and surface normal images using pre-trained models. Specifically, I employ SAM [21] to map each RGB image I to three sets of masks with different granularities: $M_c(I)$, $M_m(I)$, and $M_f(I)$, representing coarse, mid, and fine-level masks, respectively. In this process, I use SAM's point query method, similar to the preprocessing approach in LangSplat [116]. I then use

DSINE [5] to estimate surface normals denoted by $N(I)$.

5.1.2 Normal and Density Fitting Loss

The normal and density fields, $n(x)$ and $\sigma(x)$, are approximated using voxel grid representations and fitted to the predicted normal and segmentation masks, following prior work [83]. While I adopt two key loss terms from the previous work – (i) the normal vector rendering loss and (ii) the density penalization loss for background pixel rays (refer to the previous paper for details) – in this section, I introduce a novel *density-normal consistency regularization loss* to further enhance geometric accuracy.

I empirically observe that for highly concave objects, the normal rendering loss alone tends to cause $n(x)$ to overfit the rendered normal images while significantly violating geometric consistency with $\sigma(x)$, leading to distorted results. To address this issue, I propose a density-normal consistency regularization loss for each ray r, defined as the difference between the normal vector obtained from the gradient of the density field and the predicted normal. Denoting the accumulated transmittance along the ray r as $T(r,t) = \exp(-\int_{t_n}^{t} \sigma(r(s))ds)$, the loss can be expressed:

$$L_{\text{normal}}(r) := \left\| \frac{\int_{t_n}^{t_f} T(r,t)\nabla\sigma(r(t))\sigma(r(t))dt}{\left\|\int_{t_n}^{t_f} T(r,t)\nabla\sigma(r(t))\sigma(r(t))dt\right\|} - N(r) \right\|. \tag{5.1}$$

5.1.3 Feature Learning Loss

Given the density field $\sigma(x)$, I train a hierarchical field $F(x) = (F_c(x), F_m(x), F_f(x))$ for each granularity level c, m, and f. Each field is trained simultaneously but independently using a contrastive loss formulation [14]. In brief, contrastive loss encourages feature vectors along rays passing through pixels of the same mask within an image to be similar (positive pairs), while ensuring that feature vectors along rays passing through pixels of different masks are distinct (negative pairs). At each training step, I sample 256 pairs of rays from a single viewpoint to construct positive and negative pairs for the triplet loss [123].

I can perform clustering to identify object instances using $F_c(x)$, $F_m(x)$, or $F_f(x)$. However, as discussed in the introduction (Fig. 5.1), there is inherent ambiguity in determining which level of the hierarchy corresponds to the correct object instances. Moreover, in many cluttered scenarios, none of these levels may yield the correct segmentation. As illustrated in Fig. 5.2, the medium-level representation fails to distinguish between different dog-shaped objects, while the fine-level representation over-segments objects, such as separating a cup's handle from its main body.

To obtain the correct instance identification, I should leverage an appropriate combination of representations from different levels, which is achieved through my Interact-to-Identify algorithm, introduced in the next section.

5.1.4 Instance Field Representation

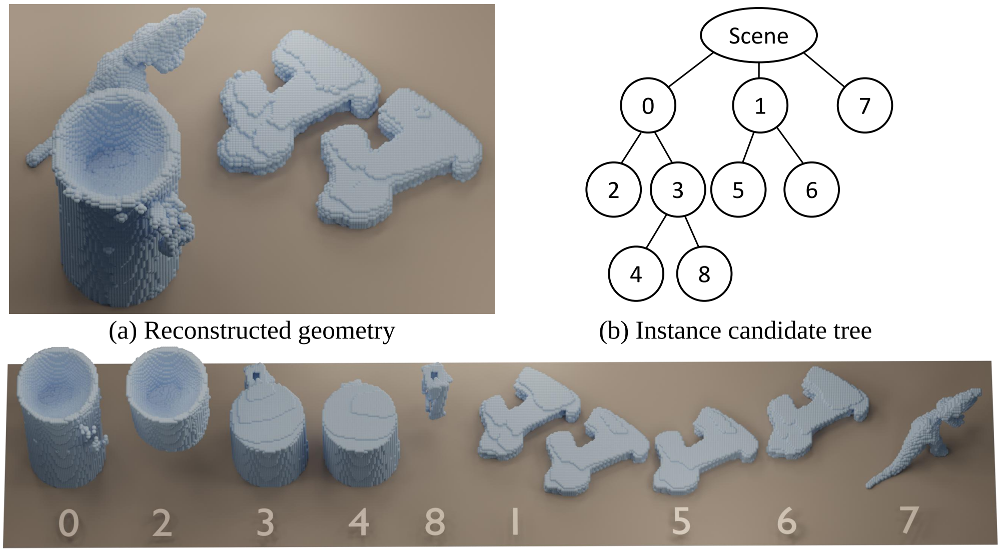
In Section 5.2, I obtain the initial field representations $n(x), \sigma(x)$ and three levels of feature fields: $F_c(x)$, $F_m(x)$, and $F_f(x)$. Leveraging these representations, my *Interact-to-Identify* framework enables robots to autonomously change the scene, gather more information, identify object instances, and fine-tune the geometry.

Specifically, it consists of four steps. I first construct an *instance candidate tree* using the density and feature fields with clustering algorithms. Given the tree representations, I then propose an effective heuristic algorithm to determine which part to interact with – using pick-and-place or pushing actions – to introduce *informative changes* in the scene. With a few additional sparse RGB images, I identify object instances and estimate a transformation matrix for each object based on rigid-body assumptions. Finally, I fine-tune each object's geometry using additional images. The overall process is summarized in Fig. 5.3 and each step is detailed in the subsequent sections.

5.1.5 Instance Candidate Tree Construction

An example of the final output of this section, referred to as the *instance candidate tree*, is visualized in Fig. 5.4. Since I have three levels of hierarchical feature fields, one might reasonably assume that the tree representation can be easily obtained by performing clustering in each feature space and applying appropriate heuristics to determine inclusion relationships. However, I empirically find that clustering results at each feature level are often imperfect, leading to inconsistencies. In some cases, a cluster present in the mid-level features may be missing from the coarse level, while in other cases, a mid-level cluster may be larger than its corresponding coarse-level cluster, resulting in noisy and inconsistent hierarchical structures.

I develop a tree construction algorithm that is robust to noisy and inconsistent hierarchical clustering results. First, recall that I have voxel-grid representations of σ and F. By extracting object point clouds through thresholding σ and clustering them using the corresponding feature vectors F with HDBSCAN [99], I obtain multiple point clouds – each corresponding to a cluster – at each feature level. Second, irrespective of whether a cluster belongs to the coarse, mid, or fine level, I sort all clusters in descending order based on the bounding box size in their respective point clouds. Third, the largest cluster is assigned as the first root node. I then apply an iterative tree generation algorithm that sequentially determines whether each subsequent cluster should be a child of an existing node, a new root node, or an identical duplicate of an



(a) Reconstructed geometry

(b) Instance candidate tree

(c) Cluster per node

Figure 5.4: Instance candidate tree and its nodes. The Instance candidate tree (b) contains information about the reconstructed geometry (a). Each node in (b) corresponds to one cluster as in (c), with the child node included in its parent node.

existing node, expanding the tree accordingly[2].

5.2 Method

5.2.1 Algorithms for Determining Where-to-Interact

To uncover previously unobserved surfaces through robotic interaction, I aim to actively manipulate objects within the scene to achieve configurations that maximize surface visibility. To this end, I present an algorithm that evaluates and selects interactions based on their potential to reveal new geometric information. I first introduce a method to quantify the expected information gain from an interaction by leveraging a visibility-based metric. Subsequently,

[2] Intersection over Union (IoU) between clusters in voxel space – comparing the existing cluster and the one being added – are used to determine whether the cluster should be assigned as a child node, a new root node, or identified as an identical duplicate.

I propose a sampling-based action selection strategy that identifies the optimal interaction by maximizing the anticipated visibility improvement.

First, to calculate visibility in 3D, I define a single visibility scalar value per voxel. The visibility field represents how much each point was observable during the training process. Given the standard definition of transmittance: $T(r,t) = \exp\left(-\int_{t_n}^{t} \sigma(r(s))\,ds\right)$, I define the visibility U for a voxel (i,j,k) as:

$$U(i,j,k) = \max_{r \in R_{ijk}} \left(T(r,t^*)\sigma(r(t^*))\right), \tag{5.2}$$

where R_{ijk} is the set of training rays that pass through the voxel (i,j,k), and t^* is defined for each ray r such that $r(t^*) = (i,j,k)$. If none of the rays passes through the voxel, U is set to zero. A lower U indicates unobserved voxels, hence regions which should be uncovered by the interaction, as in Fig. 5.5(a).

Building upon the visibility metric, I evaluate candidate actions by sampling from a discretized action space and computing the expected utility of each action. Each action a is defined by a target node within the instance candidate tree and a displacement vector on the x-y plane. The utility of an action is quantified by the amount of newly visible surface area it reveals, as measured in the image space of viewpoints V, while adhering to constraints such as collision avoidance. Specifically, for each sampled action, I compute the the value Q, proportional to the cumulative increase in pixel-level visibility and discard actions that result in collisions with other objects or the environment as in the following optimization objective:

$$Q(a) = \sum_{v \in V} proj_v(act(U,a)) + C(a). \tag{5.3}$$

Here, $proj_v$ refers to projection on the viewpoint v, and $act(U,a)$ is the visibility field after applying the action a. $C(a)$ encapsules constraints such as collision and workspace. For each connected component in the instance candidate tree, I select actions that maximize the total value: $\arg\max_a Q(a)$. To ensure computational efficiency, I discretize the action space into 12 uniformly spaced directions and 20 distance magnitudes ranging from 1 cm to 40 cm.

5.2.2 Instance Identification under Rigid-Body Assumption

Through interaction, if different instance objects – initially closely attached and difficult to distinguish – move apart and become more separated, new observations (i.e., a few additional sparse RGB images) can provide valuable information to resolve initial ambiguities. With this information, the goal of instance identification is to determine which combination of nodes at each level corresponds to the correct object instances. For example, in Fig. 5.4(b), I must decide whether (0,1,7), (2,3,1,7), or another combination is the correct set of object instances.

My key idea is based on the *rigid body assumption* for each object. The high-level intuition of our method is as follows: given a candidate instance pair, e.g., (0,1,7), I quantify how *well* the selected 3D clusters can be transformed through rotation and translation such that the projections of the transformed clusters onto 2D images align with the masks of the new observations. Starting from the root node combination, I progressively refine the node partitioning, searching for an optimal combination until the alignment error is sufficiently low.

To elaborate, I first introduce some notations. Let $v = 1, \ldots, V$ be the index for input images, each captured from a different camera pose, given V new observations. Let m_v be the mask image corresponding to the v-th view observation. Let X_i denote a 3D cluster or point cloud in the tree, where $i = 0, \ldots, N-1$, and N is the total number of nodes in the tree. Let L be the number of leaf nodes, and let l be an array of leaf node indices. For example, in Fig. 5.4, $L = 6$ and $l = [2, 4, 8, 5, 6, 7]$. Let T_i be the rotation and translation parameters for X_i, and when X_i is transformed by T_i and projected onto the image plane in the v-th direction, it forms a 2D point cloud, which I denote by $\mathrm{proj}_v(T_i X_i)$.

My algorithm consists of two steps. First, I determine the view-dependent mask assignment matrices $S_v \in \{0,1\}^{M \times L}$ with binary values, where M is the number of masks in m_v for $v = 1, \ldots, V$. The matrix S_v maps each leaf node to one of the masks[3], ideally to a mask that contains it, and is surjective (noting that $L \geq M$). Specifically, to find S_v for each m_v with M masks, I use LightGlue [90], which provides pixel-wise correspondences (outputting one-to-one matches for a subset of pixels). I define $W_v \in \mathbb{R}^{M \times L}$ such that $W_v(i,j)$ represents the number of matches where the two pixels belong to the i-th mask and the j-th cluster. With a convex relaxation allowing S_v to take values within $[0,1]$, I then formulate the following linear programming for each v independently:

$$\begin{aligned} &\min_{S_v} -\mathrm{Tr}(W_v^T S_v) \\ &\text{s.t.}\ \ 0 \leq S_v(i,j) \leq 1,\ \ \sum_{j=1}^{L} S_v(i,j) = 1,\ \ \sum_{i=1}^{M} S_v(i,j) \geq 1, \end{aligned} \tag{5.4}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, L$. This linear programming can be efficiently solved, e.g., using [47]. Then I project S_v to take either 0 or 1.

Second, as discussed above, I progressively adjust the node combination to find the optimal selection. During this process, I optimize the rotation and translation parameters for each cluster to minimize the alignment error, where S_v is explicitly used to define the error. Specifically, let I be an initial guess (the root node combination) for the instances (e.g., $I = \{0, 1, 7\}$ for the example in Fig. 5.4). Let $\mathrm{leaf}(i)$ be the set of leaf node indices of the i-th node, and let m_v^j be the set of 2D pixel coordinates in the j-th mask of m_v. Lastly, let $\mathrm{supp}(v)$

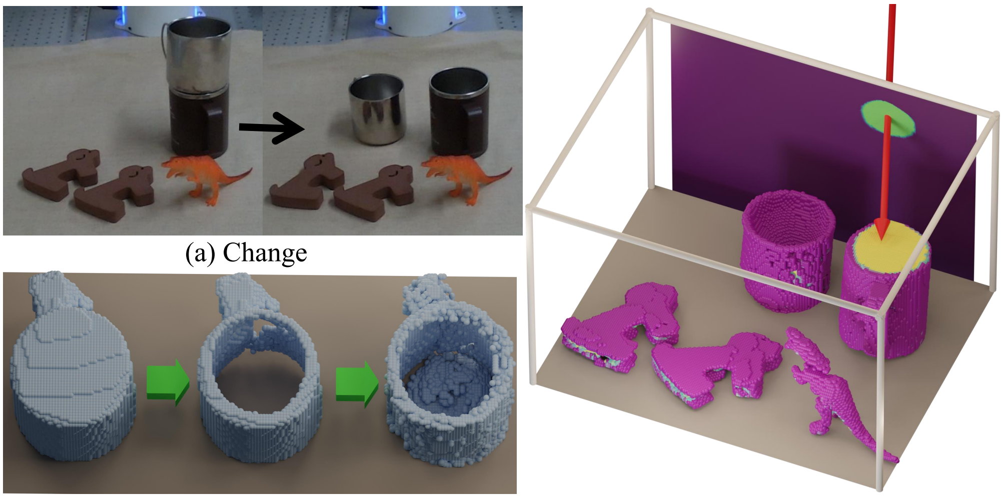
[3] To see why, think of the nodes and masks as one hot vectors.

denote the set of indices where the vector v has nonzero entries. I then solve the following optimization:

$$\min_{\{T_i : i \in I\}} \sum_{v=1}^{V} \sum_{i \in I} d\big(\mathrm{proj}_v(T_i X_i), \bigcup_{k \in \mathrm{leaf}(i)} \bigcup_{j \in \mathrm{supp}(S_v e_k)} m_v^j\big), \tag{5.5}$$

where $d(\cdot, \cdot)$ is the Chamfer distance metric between two 2D point clouds and e_k is k-th standard basis in $\mathbb{R}^L$. If the objective function is minimized below a predefined threshold, I stop and return the instances I. Otherwise, I split one of the nodes and redefine I, e.g., $I : \{0, 1, 7\} \mapsto \{2, 3, 1, 7\}$, then solve the optimization again. I repeat this process until the alignment error is sufficiently low.

When the number of instances in I is large, simultaneously optimizing all transformation parameters can be computationally expensive. To develop an efficient algorithm, I propose a strategy to obtain a proxy solution more quickly. I decouple the optimization in Eq. (5.5) for each i and replace the distance metric d with a one-way Chamfer distance metric – specifically I utilize the robust loss from [13] in order to mitigate the effects of noise –, which is zero if the first argument is contained within the second. If the objective for the i-th node is sufficiently minimized, I accept it. Otherwise, I consider its children. If any of the children is a leaf node, I accept it; otherwise, I repeat the process. As such, independently optimizing multiple nodes enables parallel computation and reduces the problem dimension, significantly improving speed. Once this process is complete, I obtain a set of selected nodes I and transformation parameters $\{T_i\}$ for each $i \in I$. Finally, I simultaneously fine-tune all parameters $\{T_i\}$ using Eq. (5.5).


(a) Change

(c) Instance-wise update process

(b) Render visibility

Figure 5.5: Selective geometric finetuning process. After a scene change (a), I calculate uncertain surfaces (yellow in (b)) that represents newly visible parts. For instances with uncertain parts, I perform geometric finetuning (c) to alleviate geometric ambiguity.

5.2.3 Selective Geometric Finetuning

Once $\{T_i\}$ is obtained, I transform each cluster as $X_i \mapsto T_i X_i$, forming an updated 3D scene. Scene updates may disclose previously occluded surfaces, which may have uncertain (not observed) geometry. To solve this problem, I design a geometric finetuning algorithm to efficiently reuse the new observations captured in Sec. 5.2.2, while utilizing the visiblity in Sec. 5.2.1. I perform selective finetuning on newly discovered volume only, leading to a more cost efficient algorithm.

First, I filter instances with newly visible surfaces from the new observations for efficiency. I use a volume rendering technique to obtain visibility values from each instance as in Fig. 5.5(b). I consider an instance as a target of geometric finetuning if it contains low-visibility surfaces from all new observation viewpoints. Next, I perform geometric finetuning by leveraging existing geometry

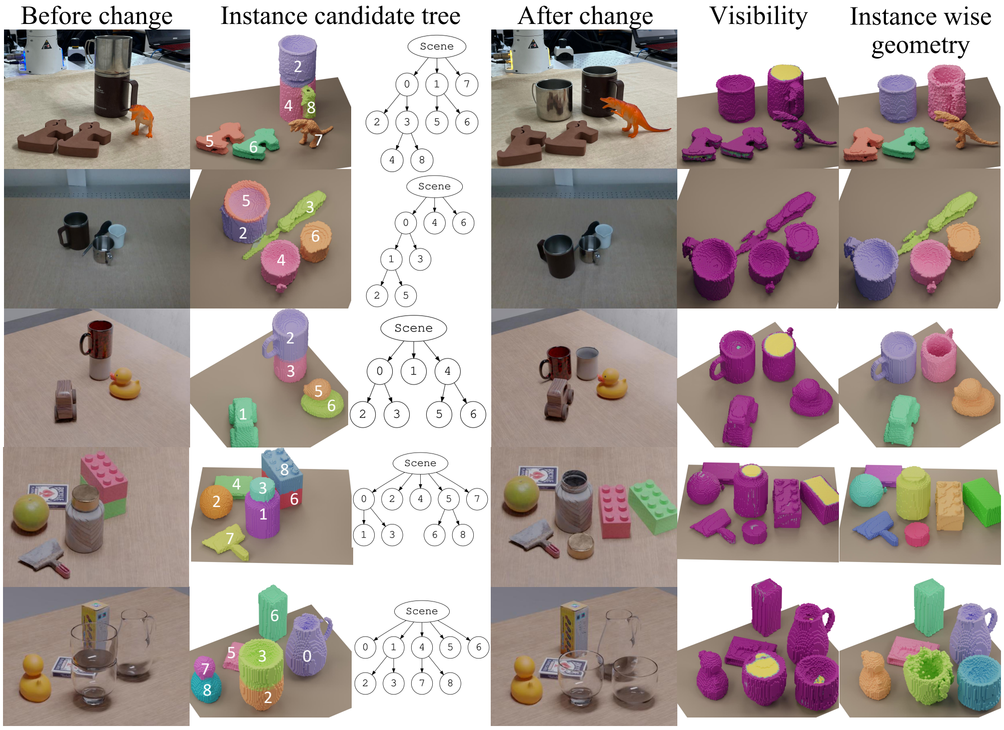


Figure 5.6: Instance candidate fields, uncertainty, and instance wise geometry for various scenes. Different colors represent different instances in instance candidate tree and instance wise geometry. In instance candidate tree, I only depict the leaf nodes. For uncertainty visualization, yellow represents high uncertainty (newly visible surface due to change). *Interact-to-Identify* successfully generates instance candidate trees and recovers instance wise geometry given change.

$T_i X_i$, the visibility U, and images from the new observations. In order to preserve the certain (well-observed) geometry in $T_i X_i$, I consult U to retain parts with high visibility. Then, with the new observations, I reuse the loss of Sec. 5.1.2 to finetune the geometry as in Fig. 5.5(c).

5.3 Experiments

In this section, I provide both qualitative and quantitative analysis on the performance of *Interact-to-Identify* on identifying individual object instances and their 3D geometries by understanding and utilizing change in the scene.

Implementation-wise, I first predict surface normals and instance masks each using [5, 21]. To reconstruct the objects solely, I retain predicted masks within the predefined workspace of the robot. The workspace is defined to be a 50 cm x 60 cm x 30 cm box positioned at the center of the franka panda workspace. All of my experiments are conducted on an RTX 3090, with an i7-8700 CPU. In addition to my real-world setup including a Realsense d435i with Franka Panda, I create photo-realistic scenes using Blender Cycles [15] by populating a tabletop with household objects, for groundtruth annotations.

First, I report the qualitative results in Fig. 5.6. The first column represents images before change, and the second and third columns visualize the instance candidate tree. For the instance candidate tree, different colors notate different instance candidates, and only the leaf nodes of the tree are visualized. The fourth column reports the change inflicted upon the scene, while the fifth and last columns each represent the uncertainty in Sec. 5.2.3 and the final instance-wise geometry. The first and second rows are examples from my real-world setup, while the last row originates from my photo-realistic Blender dataset. For all cases *Interact-to-Identify* successfully captures instance candidates in a tree formulation, estimates, and exploits change to reflect and finetune changes in the geometry.

Second, I provide quantitative results regarding the ability of my model to obtain geometry and instance information. I conduct evaluation on my Blender dataset for accessibility to groundtruth instance and geometry. I simulate change within the scene by changing the position and orientation of each object. I render RGB images to 50 viewpoints uniformly sampled from a hemisphere and predict surface normals and instances using [5, 21].

Table 5.1 compares the geometric accuracy of variations of my model with

Table 5.1: Geometric update performance of various methods. Metrics include visual surface discrepancy (VSD) [63] and intersection over union (IoU) of depth for 150 novel viewpoints. Bold represents best results while underline refers to second best.

	VSD (↓)				IoU (↑)			
Models	2 img	4 img	8 img	16 img	2 img	4 img	8 img	16 img
Ours (U+F)	**0.0368**	**0.0361**	**0.0323**	**0.0325**	**0.876**	0.885	0.903	0.903
Ours (U)	0.0418	0.0384	0.0367	0.0372	0.844	0.862	0.884	0.885
Ours (S)	0.0871	0.0421	0.0380	0.0352	0.668	**0.897**	**0.942**	**0.953**
Dex-NeRF (U)	0.1348	0.0666	0.0549	0.0503	0.797	0.808	0.895	0.919
Dex-NeRF (S)	1.0429	0.9748	0.0547	0.0556	0.050	0.072	0.861	0.918
NeRF (U)	0.1222	0.0712	0.0614	0.0611	0.797	0.808	0.895	0.919
NeRF (S)	0.8468	1.0522	0.1050	0.0652	0.050	0.072	0.861	0.918

- U: update
- S: Scratch
- F: Finetune

other field-based methods such as Dex-NeRF [66] and NeRF [102]. For my method, the update scheme represents instance-wise rigid body transform obtained from Sec. 5.2.2, while finetuning refers to geometric finetuning of Sec. 5.2.3. For the update scheme for Dex-NeRF [66] and NeRF [102], I follow the method of Evo-NeRF [72] by loading the trained weights and resuming training. I compare in such a manner since my update method cannot be directly applied on continuous field representations such as Dex-NeRF [66] and NeRF [102].

I evaluate geometry with 2 metrics: Visual Surface Discrepancy (VSD) [63] and Intersection over Union (IoU). VSD refers to the average error of the rendered object depth with respect to the groundtruth depth. IoU measures the match between rendered object masks and groundtruth masks. I render object depth and mask to 150 novel viewpoints uniformly sampled from a hemisphere surrounding the objects. My experiments show that my method is both accurate and image efficient in capturing the geometry for a scene with change. With only 2 images, my method can successfully transform objects in 3D space (update) while finetuning newly visible parts.

Table 5.2 reports the instance identification performance of my model, the ablated versions of my model, and Garfield [74]. The ablated versions (coarse, mid, fine) are given only fixed types of masks from SAM [21], which cannot be updated based on observed changes. Instead, they represent the 3D instances I can obtain from passive visual observation. For Garfield [74], I sweep the scale hyperparameter from 0 to 0.40 in 0.05 steps. Garfield [74] output only a single cluster for scales over 0.15. For a fair comparison, I manually crop the background geometry captured by Garfield and leave only the objects in the clustering phase.

For the metric, I calculate the precision and recall of the 3D bounding boxes of the predicted instances with respect to the groundtruth bounding box. Precision refers to IoUs in the 3D bounding box averaged over predicted instances while recall refers to IoUs averaged over groundtruth instances. My experiments report that my method can identify instances more accurately than other methods such as Garfield [74]. In addition, I find that determining instances in a change-based manner outperforms utilizing any level of mask predicted via texture from [21].

Table 5.2: Precision and recall of 3D bounding boxes. Precision averages IoU over predicted instances, while recall averages over GT instances. Bold represents best results.

Models	Precision (↑)	Recall (↑)
Ours	**0.776**	**0.776**
Ours (coarse)	0.547	0.612
Ours (mid)	0.650	0.674
Ours (fine)	0.611	0.505
Garfield [74] (0.00)	0.628	0.431
Garfield [74] (0.05)	0.596	0.417
Garfield [74] (0.10)	0.494	0.472
Garfield [74] (0.15)	0.247	0.245

Chapter 6

Conclusion

6.1 Summary

Vision-based grasping aims to identify effective grasp configurations by utilizing data from visual sensors. In most cases, algorithms process RGB and depth inputs from contemporary RGB-D cameras to determine the precise position and orientation of the gripper for grasping an object in 3D space. This field has been widely explored due to its promise in facilitating robotic manipulation within unstructured and dynamic settings. When robots encounter unfamiliar objects in diverse environments, vision-based grasping plays a fundamental role in enabling practical tasks such as organizing items or performing bin packing operations.

As grasping algorithms and vision sensing technologies continue to evolve, vision-based grasping has grown increasingly dependable across a broad spectrum of applications. Contemporary grasping models, refined through advanced engineering, are able to incorporate physical principles—such as surface smoothness and object center of mass—to produce precise and effective grasps grounded in object geometry. Simultaneously, advancements in sensing technologies like stereo cameras, LiDAR, and infrared have made accurate geometric perception more accessible and cost-effective. In addition, the growing availability of real-world datasets has played a crucial role in enhancing the performance of vision-based grasping in practical, real-world scenarios.

Despite these advancements, scenes involving transparency and clutter remain difficult for vision sensors to interpret accurately, which significantly undermines the reliability of vision-based grasping systems. Transparent objects, in particular, present major challenges for depth sensing due to their inherent optical properties, often resulting in distorted or missing depth data. Similarly, cluttered environments—where multiple objects are in close contact—lead to occlusions and hidden surfaces that prevent complete geometric understanding. These factors hinder the accurate reconstruction of object geometry from visual input, ultimately leading to unreliable grasp predictions.

This thesis introduced a robust approach for capturing scene geometry in challenging environments that include transparency and clutter. The proposed method made use of general-purpose pretrained vision models, eliminating the need for environment-specific adjustments or fine-tuning. It leveraged mid-level visual cues—such as instance masks and surface normals—and proposed methods to utilize them in a spatially coherent manner to enable reliable geometric reconstruction.

In Chapter 3, I proposed MasKGrasp, a simple CNN-based robotics grasping method trainable on augmented real images, that processes both transparent and opaque objects and generalizes to real-world objects. I demonstrate that the instance mask is an effective yet light-weight intermediate representation for stable grasp pose estimation for individual objects, regardless of their significant appearance variations. Also, I propose a pipeline and dataset that augments large-scale instance segmentation datasets such as [89] with more transparent objects, leading to more reliable instance masks. In addition, instance masks provide a clue to avoid crowded regions, which results in better grasping performance on scenes cluttered with multiple objects. MasKGrasp outperforms previous approaches on a real-world test environment with unseen objects. As more real-world annotated datasets for instance segmentation become available [21], mask-based robotic grasping is expected to become increasingly reliable.

However, MasKGrasp is a 2D grasping algorithm that assumes a fixed grasping height perpendicular to the image plane, and has difficulty grasping objects incompatible with such a fixed grasping height. In addition, when an opaque object is placed inside a transparent object, the masks are often combined into one, leading to inaccurate grasps. Even with flow-based augmentations, it seems that understanding complex real-world refractions requires more sophisticated datasets.

In Chapter 4, I proposed NFL, a robust and practical solution to perform 6-DoF grasping of transparent objects. Since scanning hardware fails to obtain the correct geometry for transparent objects, it is a natural thought to change the input modality to multi-view images. In contrast to other methods, which aggregate multiview information based on RGB images, I propose that using surface normals has substantial benefits in terms of multi-view consistency. NFL models predicted surface normals and masks as a probabilistic distribution and learns a normal field of a real scene in 40 seconds. The normal field includes accurate, holistic 3D geometry from which I can quickly infer grasp positions. Experiments on various datasets show the robustness of NFL to many real-world scenes and superior grasping performance. I also conduct ablation studies to support the choice of using surface normals and segmentation masks rather than RGB to form neural fields for transparent objects. Moreover, with recent advancements in both surface normal estimation [5] and instance segmentation [21], in terms of datasets and models, the applicability of the proposed approach is expected to further improve in the near future.

Although NFL exhibits stable performance for a variety of scenes, it still has room for enhancement. First, expediting the algorithm will definitely improve the practicality of grasping. Even building on the grid-based DVGO [127] algorithm, NFL is still slower than GraspNeRF [43]. With faster speed, one can deploy the algorithm to quickly refresh the geometry in sequential grasping. Additionally, NFL finds the 3D geometry of the scene from input images surrounding the bounded volume of known workspace. Although the required setting is not difficult for a conventional manipulation setting, it may hinder generalizing to all images in the wild. For example, I could not evaluate NFL on the HAMMER dataset [69] which has images captured from only one side of the objects.

In Chapter 5, I proposed *Interact-to-Identify*, a method to actively resolve ambiguities of object instance and geometric information for cluttered scenes. My method can (i) interact with objects via a robot arm, (ii) estimate and exploit change to clearly determine instances, and (iii) reflect change in the scene while also finetuning the geometry without extensive data capture. Centered on the concept of visibility, *Interact-to-Identify* generates actions that not only maximize the exposure of object surfaces but also enable efficient geometric updates for newly observed regions. Experiments on both real and synthetic settings show that *Interact-to-Identify* can stably acquire object instances and geometry, which can later benefit complicated tasks such as rearrangement or packing.

Although *Interact-to-Identify* exhibits consistent performance in a variety of scenes, it is still far from perfect. First, *Interact-to-Identify* requires that the true instances should be included in the nodes initially formulated in the instance candidate tree, which depends on the detection performance of general vision modules such as SAM [21]. While the generalization capability of current vision models was sufficient for the conducted experiments, introducing mechanisms for further splitting or merging nodes could enhance the algorithm's flexibility in handling challenging or adversarial cases. In addition, the scope of change *Interact-to-Identify* asserts is confined to rigid body transforms. Changes in the scene outside of such scope, for example, the addition of new objects, cannot be modeled by the current pipeline. Further adding modes would also improve the flexibility of the algorithm.

Although stated separately, the three elements agree on the final goal of reliable vision-based robotic grasping. For reliability, the three methods commonly utilize the outputs of general-purpose vision models such as instance segmentation or surface normal estimation. They combine the outputs in a way that avoids scene-specific post-processing for reliable acquisition of the objects' geometry.

6.2 Open Questions and Future Directions

Sensor Modality and Reliability In the pursuit of more reliable vision-based grasping, an intriguing direction involves the integration of emerging vision sensors, such as event cameras. While not yet as commonly adopted as traditional RGB-D sensors, event cameras offer a set of unique characteristics that are particularly well-suited to robotic applications—including high temporal resolution, low power consumption, and the ability to function effectively in low-light conditions. These properties make them promising candidates for enhancing the robustness of perception in dynamic or challenging environments. Incorporating such advanced sensing technologies into vision-based grasping systems could open up new possibilities for achieving greater reliability and responsiveness in real-world robotic manipulation tasks.

End-to-end Models and Reliability Many recent works in robotic manipulation adopt a data-driven approach that directly predicts robot actions from sensor inputs [18, 151, 110, 77, 129]. These end-to-end models—often referred to as vision-language-action (VLA) models—learn the relationships among visual observations, language instructions, and action tokens using large-scale transformer architectures. Despite their impressive capabilities, such models often struggle to generalize to unseen environments, primarily due to the limited availability of web-scale training data [119]. While increasing the size and diversity of the dataset is a straightforward solution, the ideas in this thesis can propose a complementary direction. Specifically, incorporating geometric cues derived from mid-level visual representations, such as surface normals and instance masks, may enhance generalization across diverse scenes [25].

Bibliography

[1] Sven Albrecht and Stephen Marsland. Seeing the unseen: Simple reconstruction of transparent objects from point cloud data. In *Robotics: Science and Systems*, volume 3, pages 1–6, 2013.

[2] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[3] Christopher G Atkeson, Chae H An, and John M Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.

[4] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021.

[5] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024.

[6] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[7] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Active perception: Past, present, and future. *arXiv preprint arXiv:1603.02729*, 2016.

[8] Ruzena Bajcsy and Larry S. Hutchinson. Active vision. *Proceedings of the IEEE*, 76(8):996–1005, 1990.

[9] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

[10] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5965–5974, 2016.

[11] Igor Barros Barbosa, Marco Cristani, Barbara Caputo, Aleksander Rognhaugen, and Theoharis Theoharis. Looking beyond appearances: Synthetic training data for deep cnns in re-identification. *Computer Vision and Image Understanding*, 167:50–62, 2018.

[12] Alan H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[13] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4331–4339, 2019.

[14] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] Blender Foundation. Blender - cycles renderer, 2023. Version 3.1,.

[16] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[17] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on Robot Learning*, pages 1602–1611. PMLR, 2021.

[18] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[19] Junhao Cai, Hui Cheng, Zhanpeng Zhang, and Jingcheng Su. MetaGrasp: Data efficient grasping by affordance interpreter network. In *ICRA*, pages 4960–4966, 2019.

[20] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[21] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36:25971–25990, 2023.

[22] Lillian Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In *2012 IEEE International Conference on Robotics and Automation*, pages 3875–3882. IEEE, 2012.

[23] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 2022.

[24] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[25] Bryan Chen, Alexander Sax, Gene Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. *arXiv preprint arXiv:2011.06698*, 2020.

[26] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. TOM-Net: Learning transparent object matting from a single image. In *CVPR*, pages 9233–9241, 2018.

[27] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Tom-net: Learning transparent object matting from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9233–9241, 2018.

[28] Haoran Chen, Kenneth Blomqvist, Francesco Milano, and Roland Siegwart. Panoptic vision-language feature fields. *IEEE Robotics and Automation Letters*, 2024.

[29] Kai Chen, Stephen James, Congying Sui, Yun-Hui Liu, Pieter Abbeel, and Qi Dou. Stereopose: Category-level 6d transparent object pose estimation from stereo images via back-view nocs. In *IEEE International Conference on Robotics and Automation*, pages 2855–2861, 2023.

[30] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[31] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.

[32] Xiaotong Chen, Huijie Zhang, Zeren Yu, Anthony Opipari, and Odest Chadwicke Jenkins. Clearpose: Large-scale transparent object dataset and benchmark. In *European Conference on Computer Vision*, pages 381–396. Springer, 2022.

[33] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

[34] Xinhua Cheng, Yanmin Wu, Mengxi Jia, Qian Wang, and Jian Zhang. Panoptic compositional feature field for editable scene rendering with network-inferred labels via metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4957, 2023.

[35] Alan D Christiansen, Matthew T Mason, and Tom M Mitchell. Learning reliable manipulation strategies without initial physical models. *Robotics and Autonomous Systems*, 8(1-2):7–18, 1991.

[36] Fu-Jen Chu, Ruinian Xu, and Patricio A. Vela. Real-world multiobject, multigrasp detection. *RA-L*, 3(4):3355–3362, 2018.

[37] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[38] Intel Corporation. Intel realsense depth camera d435. `https://www.intelrealsense.com/depth-camera-d435/`. Accessed: 2025-05-27.

[39] Intel Corporation. Intel RealSense LiDAR Camera L515. `https://www.intelrealsense.com/lidar-camera-l515/`. Accessed: 2025-07-15.

[40] Microsoft Corporation. Kinect for windows v2. `https://developer.microsoft.com/en-us/windows/kinect/`, 2014. Hardware device.

[41] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

[42] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[43] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In *IEEE International Conference on Robotics and Automation*, 2023.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[45] Amaury Depierre, Emmanuel Dellandréa, and Liming Chen. Jacquard: A large scale dataset for robotic grasp detection. In *IROS*, pages 3511–3516, 2018.

[46] Amaury Depierre, Emmanuel Dellandréa, and Liming Chen. Scoring graspability based on grasp regression for better grasp prediction. *arXiv:2002.00872v3 [cs.RO]*, 2021.

[47] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[48] 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. *Artificial Intelligence Review*, 54, 2021.

[49] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[50] Hao-Shu Fang, Minghao Gou, Chenxi Wang, and Cewu Lu. Robust grasping across diverse sensor qualities: The graspnet-1billion dataset. *The International Journal of Robotics Research*, 2023.

[51] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

[52] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[53] Hongjie Fang, Hao-Shu Fang, Sheng Xu, and Cewu Lu. Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline. *IEEE Robotics and Automation Letters*, 7(3):7383–7390, 2022.

[54] Xiaolin Fang, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Embodied uncertainty-aware object segmentation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2639–2646. IEEE, 2024.

[55] Nicholas I Fisher, Toby Lewis, and Brian JJ Embleton. *Statistical analysis of spherical data*. Cambridge university press, 1993.

[56] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[57] Daoyi Gao, Yitong Li, Patrick Ruhkamp, Iuliia Skobleva, Magdalena Wysocki, HyunJun Jung, Pengyuan Wang, Arturo Guridi, and Benjamin Busam. Polarimetric pose prediction. In *European Conference on Computer Vision*, pages 735–752. Springer, 2022.

[58] Alexandre Gariépy, Jean-Christophe Ruel, Brahim Chaib-Draa, and Philippe Giguère. GQ-STN: Optimizing one-shot grasp detection based on robustness classifier. In *IROS*, pages 3996–4003, 2019.

[59] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. RGB Matters: Learning 7-dof grasp poses on monocular rgb images. In *ICRA*, 2021.

[60] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. In *European Conference on Computer Vision*, pages 382–400. Springer, 2025.

[61] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *ICLR*, 2021.

[62] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017.

[63] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[64] Haojie Huang, Dian Wang, Xupeng Zhu, Robin Walters, and Robert Platt. Edge grasp network: A graph-based se (3)-invariant approach to grasp detection. In *IEEE International Conference on Robotics and Automation*, pages 3882–3888, 2023.

[65] Dong Huo, Jian Wang, Yiming Qian, and Yee-Hong Yang. Glass segmentation with rgb-thermal image pairs. *IEEE Transactions on Image Processing*, 32:1911–1926, 2023.

[66] Jeffrey Ichnowski*, Yahav Avigal*, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning*, 2021.

[67] Kuang-Yu Jeng, Yueh-Cheng Liu, Zhe Yu Liu, Jen-Wei Wang, Ya-Liang Chang, Hung-Ting Su, and Winston H. Hsu. GDN: A coarse-to-fine (c2f) representation for end-to-end 6-dof grasp detection. In *CoRL*, 2020.

[68] Hanxiao Jiang, Binghao Huang, Ruihai Wu, Zhuoran Li, Shubham Garg, Hooshang Nayyeri, Shenlong Wang, and Yunzhu Li. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation. *arXiv preprint arXiv:2402.15487*, 2024.

[69] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, et al. On the importance of accurate geometry data for dense 3d vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023.

[70] James Kajiya and Brian Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 18:165–174, 1984.

[71] Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8602–8611, 2020.

[72] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. Evo-nerf: Evolv-

ing nerf for sequential robot grasping of transparent objects. In *6th Annual Conference on Robot Learning*, 2022.

[73] Ninad Khargonkar, Neil Song, Zesheng Xu, Balakrishnan Prabhakaran, and Yu Xiang. Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands. In *Conference on Robot Learning*, pages 516–526. PMLR, 2023.

[74] Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *arXiv*, 2024.

[75] Jeongyun Kim, Myung-Hwan Jeon, Sangwoo Jung, Wooseong Yang, Minwoo Jung, Jaeho Shin, and Ayoung Kim. Transpose: Large-scale multispectral dataset for transparent object. *International Journal of Robotics Research*, 2024. Accepted. To appear.

[76] Jeongyun Kim, Myung-Hwan Jeon, Sangwoo Jung, Wooseong Yang, Minwoo Jung, Jaeho Skin, and Ayoung Kim. Transpose: Large-scale multispectral dataset for transparent object. *arXiv preprint arXiv:2307.05016*, 2023.

[77] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[78] Seungyeon Kim, Taegyun Ahn, Yonghyeon Lee, Jihwan Kim, Michael Yu Wang, and Frank C Park. Dsqnet: a deformable model-based supervised learning algorithm for grasping unknown occluded objects. *IEEE Transactions on Automation Science and Engineering*, 20(3):1721–1734, 2022.

[79] Young Hun Kim, Seungyeon Kim, Yonghyeon Lee, and Frank C Park. T2sqnet: A recognition model for manipulating partially observed transparent tableware objects. In *8th Annual Conference on Robot Learning*.

[80] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[81] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[82] Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 177–184. IEEE, 2015.

[83] Junho Lee, Sang Min Kim, Yonghyeon Lee, and Young Min Kim. Nfl: Normal field learning for 6-dof grasping of transparent objects. *IEEE Robotics and Automation Letters*, 2023.

[84] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *IJRR*, 34(4-5):705–724, 2015.

[85] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022.

[86] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity

neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[87] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[88] Jiaying Lin, Zebang He, and Rynson WH Lau. Rich context aggregation with reflection prior for glass surface detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13415–13424, 2021.

[89] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[90] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

[91] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022.

[92] Jeffrey Mahler and Ken Goldberg. Learning deep policies for robot bin picking by simulating robust grasping sequences. In *CoRL*, pages 515–524, 2017.

[93] 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. *arXiv preprint arXiv:1703.09312*, 2017.

[94] 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. *RSS*, 2017.

[95] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *ICRA*, pages 5620–5627, 2018.

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

[97] 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 *IEEE International Conference on Robotics and Automation*, pages 4919–4926, 2015.

[98] 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 *ICRA*, pages 1957–1964, 2016.

[99] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[100] Haiyang Mei, Bo Dong, Wen Dong, Jiaxi Yang, Seung-Hwan Baek, Felix Heide, Pieter Peers, Xiaopeng Wei, and Xin Yang. Glass segmentation using intensity and spectral polarization cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12622–12631, 2022.

[101] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson WH Lau. Don't hit me! glass detection in real-world scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3687–3696, 2020.

[102] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[103] Daisuke Miyazaki, Megumi Saito, Yoichi Sato, and Katsushi Ikeuchi. Determining surface orientations of transparent objects based on polarization degrees in visible and infrared wavelengths. *JOSA A*, 19(4):687–694, 2002.

[104] Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *RSS*, pages 1–10, 2018.

[105] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *ICCV*, pages 2901–2910, 2019.

[106] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[107] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-DOF grasping for target-driven object manipulation in clutter. In *ICRA*, 2020.

[108] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6232–6238. IEEE, 2020.

[109] Stefan Otte, Johannes Kulick, Marc Toussaint, and Oliver Brock. Entropy-based strategies for physical exploration of the environment's degrees of freedom. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 615–622. IEEE, 2014.

[110] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024*

IEEE International Conference on Robotics and Automation (ICRA), pages 6892–6903. IEEE, 2024.

[111] Dongwon Park and Se Young Chun. Classification based grasp detection using spatial transformer network. *arXiv:1803.01356 [cs.CV]*, 2018.

[112] Dongwon Park, Yonghyeok Seo, Dongju Shin, Jaesik Choi, and Se Young Chun. A single multi-task deep neural network with post-processing for object detection with reasoning and robotic grasp detection. In *ICRA*, pages 7300–7306, 2020.

[113] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Efficient planning in non-gaussian belief spaces and its application to robot grasping. In *Robotics Research: The 15th International Symposium ISRR*, pages 253–269. Springer, 2016.

[114] Jean Ponce, Darrell Stam, and Bernard Faverjon. On computing two-finger force-closure grasps of curved 2d objects. *The International Journal of Robotics Research*, 12(3):263–273, 1993.

[115] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[116] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[117] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes. In *Conference on robot learning*, pages 53–65. PMLR, 2020.

[118] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *ICRA*, pages 1316–1322, 2015.

[119] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[120] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *IEEE International Conference on Robotics and Automation*, pages 3634–3642, 2020.

[121] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. ClearGrasp: 3D shape estimation of transparent objects for manipulation. In *ICRA*, pages 3634–3642, 2020.

[122] David Schiebener, Jun Morimoto, Tamim Asfour, and Aleš Ude. Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior*, 21(5):328–345, 2013.

[123] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[124] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[125] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv: Arxiv-2210.05663*, 2022.

[126] Stereolabs. Zed stereo camera. `https://www.stereolabs.com/zed/`, 2015. Hardware device.

[127] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[128] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-GraspNet: Efficient 6-dof grasp generation in cluttered scenes. In *ICRA*, 2021.

[129] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[130] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021.

[131] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp

detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021.

[132] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021.

[133] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021.

[134] Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. Refractive shape from light field distortion. In *2011 International Conference on Computer Vision*, pages 1180–1186. IEEE, 2011.

[135] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.

[136] Chaozheng Wu, Jian Chen, Qiaoyu Cao, Jianchi Zhang, Yunxin Tai, Lin Sun, and Kui Jia. Grasp Proposal Networks: An end-to-end solution for visual learning of robotic grasps. In *NeurIPS*, 2020.

[137] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

[138] Tianhao Wu, Chuanxia Zheng, Qianyi Wu, and Tat-Jen Cham. Clusteringsdf: Self-organized neural implicit surfaces for 3d decomposition. In *European Conference on Computer Vision*, pages 255–272. Springer, 2025.

[139] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *European Conference on Computer Vision*, pages 696–711. Springer, 2020.

[140] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *ECCV*, 2020.

[141] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer. In *IJCAI*, 2021.

[142] Yifeng Xu, Fan Zhu, Ye Li, Sebastian Ren, Xiaonan Huang, and Yuhao Chen. Rgbsqgrasp: Inferring local superquadric primitives from single rgb image for graspability-aware bin picking. *arXiv preprint arXiv:2503.02387*, 2025.

[143] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

[144] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024.

[145] Hanbo Zhang, Xuguang Lan, Site Bai, Xinwen Zhou, Zhiqiang Tian, and Nanning Zheng. ROI-based robotic grasp detection for object overlapping scenes. In *IROS*, pages 4768–4775, 2019.

[146] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022.

[147] Yuqi Zhang, Guanying Chen, Jiaxing Chen, and Shuguang Cui. Aerial lifting: Neural urban semantic and building instance lifting from aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21092–21103, 2024.

[148] Zheming Zhou, Tianyang Pan, Shiyu Wu, Haonan Chang, and Odest Chadwicke Jenkins. GlassLoc: Plenoptic grasp pose detection in transparent clutter. In *IROS*, pages 4776–4783, 2019.

[149] Runsong Zhu, Shi Qiu, Qianyi Wu, Ka-Hei Hui, Pheng-Ann Heng, and Chi-Wing Fu. Pcf-lift: Panoptic lifting by probabilistic contrastive fusion. In *European Conference on Computer Vision*, pages 92–108. Springer, 2025.

[150] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.

[151] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2:

Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

초록

비전 기반 파지는 로봇이 시각 센서로부터 얻은 정보를 바탕으로 성공적인 파지 구성을 결정하는 데 초점을 둔다. 일반적으로 이러한 알고리즘은 최신 RGB-D 카메라로부터 획득한 RGB 및 깊이 영상을 입력으로 받아, 3차원 공간에서 물체를 어떻게, 어디서 파지할지를 출력한다. 이 방식은 비정형 환경에서의 로봇 조작을 가능하게 하는 잠재력 덕분에 활발히 연구되어 왔다. 특히, 임의의 장면에서 처음 마주하는 물체를 다뤄야 할 때, 비전 기반 파지는 물체 정렬, 빈 정리 등과 같은 유의미한 작업 수행을 위한 핵심 구성 요소로 작용한다.

최근 파지 알고리즘과 시각 센서의 발전으로, 비전 기반 파지는 다양한 상황에서도 점점 더 신뢰성 있는 성능을 보이고 있다. 최신 파지 모델은 정교한 공학적 설계를 바탕으로, 표면의 매끄러움이나 무게중심과 같은 물리 기반의 편향을 반영해 물체의 기하 정보를 활용한 고품질 파지를 생성할 수 있다. 동시에 스테레오 비전, LiDAR, 적외선 기반 기술을 활용한 시각 센서 역시 정밀도는 물론 가격 면에서도 개선되어 대부분의 물체에 대해 정확한 기하 추정을 가능하게 하고 있다. 또한, 실세계 기반 데이터셋의 확산은 실제 로봇 응용 분야에서의 성능 향상에 크게 기여하고 있다.

그러나 투명 물체나 복잡하게 얽힌 장면(clutter)에서는 여전히 시각 센서가 올바른 인식을 하지 못해, 비전 기반 파지 알고리즘의 신뢰도를 크게 떨어뜨리는 문제가 발생한다. 첫째, 투명 물체는 물리적 특성상 깊이 카메라에서 복잡한 센싱 오류를 유발한다. 둘째, 여러 물체가 접촉해 있는 클러터 환경에서는 가려진 표면이나 관찰이 불가능한 영역이 많아져 정확한 기하 정보 획득이 어렵다. 이러한 요소들은 시각 기반의 정확한 기하 추정을 방해하며, 결과적으로 파지 실패로 이어질 수 있다.

본 논문에서는 투명 물체 및 클러터 환경에서도 신뢰성 있는 기하 정보를 획득

할 수 있는 방법을 제안한다. 일반 데이터로 학습된 사전 학습 비전 모듈의 출력을 활용함으로써, 특정 장면에 의존한 튜닝 없이도 안정적인 기하 재구성이 가능하도록 한다. 구체적으로, 마스크와 표면 법선과 같은 중간 수준의 표현을 공간적으로 구조화하여 활용함으로써 견고한 기하 정보를 추출한다. 또한, 실제 로봇 시스템을 활용한 실험을 통해 제안된 방법의 실용성과 현장 적용 가능성을 입증하였다.

다양한 환경에서 로봇이 인간의 노동을 효과적으로 대체하기 위해서는, 환경 변화에 따른 성능 안정성이 반드시 보장되어야 한다. 본 논문은 비전 기반 파지의 대표적 난제인 투명성과 클러터 문제를 해결하고, 일반 비전 모듈을 기반으로 한 해법을 제안함으로써 로봇 조작의 안정성과 견고함 향상에 기여하고자 한다.

주요어: 시각 기반 물체 파지, 안정성, 비전 기반 모델, 투명함, 혼잡함

학번: 2020-25228

Chapter 7

Acknowledgements

졸업이 가까워진 시점에서 박사 생활을 돌이켜 보았을 때, 정말 많은 사람들에게 도움을 받았다는 것을 다시 한번 깨달았습니다. 이 글을 통해 그 분들에게 감사의 마음을 전하고 싶습니다.

우선 가장 먼저 저를 지도해주신 김영민 교수님께 정말 많은 감사의 말씀을 드리고 싶습니다. 학부 졸업 프로젝트부터 박사논문까지 정말 많은 부분을 교수님 덕분에 무사히 끝낼 수 있었습니다. 학부 때 교수님의 신호 및 시스템 수업을 듣고 졸업프로젝트도 신청을 하게 되었는데 돌이켜 보았을 때 너무 잘한 선택이었다고 생각합니다. 앞으로 인생을 살아 가면서도 대학원 생활을 통해 연구 내적으로도 외적으로도 교수님께 배운 것들을 잊어버리지 않겠습니다.

다음으로 이 논문을 심사해주신 심사위원 분들께 감사의 말씀을 드리고 싶습니다. 우선 심사위원장을 맡아주신 오성회 교수님께서 제 연구에 대해 긍정적인 의견을 가져 주시어 저 또한 제 연구에 대해 더 많은 자신감을 가질 수 있게 되었습니다. 또한 박재식 교수님께서 연구의 접근법에 대한 근본적인 코멘트들을 해주시어 연구의 필요성을 더 논리있게 정리할 수 있었습니다. 김아영 교수님께서 주신 각 실험결과와 그 의미에 대한 코멘트들을 통해 더욱 설득력 있도록 논문을

수정할 수 있었습니다. 그리고 마지막으로 이용현 박사님께서 주신 코멘트들을 통해 논문의 각각의 요소들에 대해 더욱 연결성 있게 서술할 수 있었습니다.

다음으로 박사과정을 진행하는 동안 매일같이 보았던 연구실 동료들에게 감사를 드리고 싶습니다. 연구실에 처음 들어왔을 때 코딩에 대하여 많은 것을 가르쳐준 장동수 형과 졸업 프로젝트때 부터 저에게 로봇의 길을 보여준 민철희 형, 연구실 안과 밖에서 여러가지 활동을 같이 한 김준호 형, 그리고 연구와 야구와 운동에 대해서 많은 이야기를 한 최창운 형에게 먼저 감사드립니다. 또한 학부 동기이자 연구실 동료로서 오랜 시간을 함께한 김상민과 장호준, 연구실의 초창기에 많은 역할을 담당했었던 손현태 형과 김주현 형, 같이 전문연구요원을 진행중인 이민기 형과 김민관, 연구실의 분위기를 밝게 해주는 허형준 형, 저의 첫 연구를 함께 해 준 황인우 형, 미팅을 같이 하며 연구주제를 같이 고민하던 배광탁 형과 배진성 형과 이은선 누나에게도 고맙다고 말씀드리고 싶습니다. 또한 앞으로 연구실을 이끌어 나갈 임동근 형, 정승빈, 강승구, 문성빈, 이영환에게도 감사를 드리며 앞으로도 잘 부탁드립니다.

연구실 밖에서도 저의 박사과정을 함께한 모든 사람들에게도 감사의 말을 전하고 싶습니다. 저의 첫번째 연구부터 박사 졸업, 그 사이에 훈련소까지 같이한 김나현에게 감사합니다. 앞으로 더욱 많은 일들을 같이하게 될 텐데, 그 모든 순간에서도 지금까지처럼 옆에서 지켜보고 지지해 줬으면 좋겠습니다. 마지막으로 긴 대학원 생활, 그리고 평생토록 저를 지지하고 지탱해준 가족들에게 다시 한번 감사의 말씀을 전하고 싶습니다. 연구가 잘 될때나 잘 안될 때나 저를 응원해 주시고, 같이 미래에 대해 고민해 주셔서 감사합니다. 앞으로도 저를 계속 응원해 주시리라 믿어 의심치 않고, 그러한 응원에 보답할 수 있도록 더욱 열심히 살겠습니다.