



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Physically-based Sim2Real Transfer for Robotic Perception and Manipulation

Dissertation

with the aim of achieving a doctoral degree at the
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Universität Hamburg

Kaixin Bai
Hamburg, 2026

Day of oral defense: 03.09.2026

The following evaluators recommend the admission of the dissertation:

Supervisor:

Prof. Dr. Jianwei Zhang

Department of Informatics,

Universität Hamburg, Germany

Reviewer:

Prof. Dr. Patrick Vandewalle

Department of Electrical Engineering (ESAT),

Katholieke Universiteit Leuven, Belgium

Chair:

Prof. Dr. Frank Steinicke

Department of Informatics,

Universität Hamburg, Germany

Abstract

In real-world environments, robotic systems often face the challenge of perceiving and manipulating complex objects such as deformable cables, fragile fruits, and transparent materials. These objects exhibit weak textures, non-rigid or intricate geometry, and are susceptible to visual disturbances such as lighting variations, reflection, and refraction. Moreover, collecting large-scale annotated real-world datasets is time-consuming and costly, and label quality is often compromised in cluttered or ambiguous scenes. These limitations significantly hinder the generalization ability of deep learning models in practice.

This dissertation focuses on simulation-driven methods for robotic perception and manipulation, proposing a framework that spans synthetic data generation, structure-aware visual modeling, and robust manipulation strategies. The goal is to enable reliable Sim2Real transfer in scenarios involving complex and diverse objects.

To address the domain gap between simulated and real-world depth data from industrial structured-light sensors, we develop a physically-consistent structured-light simulator. By accurately modeling gray-code projection, sensor response, and surface interactions in a physically-based photorealistic renderer, this system produces synthetic depth maps that closely match those captured by real devices. It enables the training of depth-based perception models with improved Sim2Real performance, reducing reliance on expensive real-world annotations.

For stereo-based depth estimation, especially on transparent and overlapping objects, we propose two complementary models. One enhances stereo matching robustness in structure-deficient regions using transformer-based feature fusion and geometry priors. The other enables zero-shot stereo reconstruction by freezing a pre-trained encoder and learning a lightweight matching head, which achieves strong generalization without fine-tuning.

On the manipulation side, we introduce a collision-aware grasping network for deformable cables. Trained on physically simulated data with labeled contact and collision outcomes, the model predicts grasp feasibility and safety from depth images by implicitly learning contact and collision cues. For damage-free manipulation of soft fruits, we design a vision-guided soft gripper system with multimodal feedback control, demonstrating stable and safe performance in real robotic fruit grasping scenarios.

This work presents substantial advancements in photorealistic simulation, perception model design, and flexible object manipulation. Together, these components establish a practical foundation for enabling robust Sim2Real deployment of robotic systems in complex, real-world tasks.

Zusammenfassung

In realen Umgebungen stehen Robotersysteme häufig vor der Herausforderung, komplexe Objekte wie verformbare Kabel, empfindliche Früchte und transparente Materialien zuverlässig zu erkennen und zu manipulieren. Solche Objekte weisen schwache Texturen, nicht starre oder komplexe Geometrien auf und sind anfällig für visuelle Störungen wie Lichtvariationen, Reflexionen und Brechung. Darüber hinaus ist die Erfassung großer, annotierter Datensätze aus der realen Welt äußerst zeit- und kostenaufwendig, wobei die Labelqualität in unübersichtlichen oder mehrdeutigen Szenen oft unzureichend ist. Diese Einschränkungen behindern die Generalisierungsfähigkeit von Deep-Learning-Modellen erheblich.

Diese Dissertation befasst sich mit simulationsbasierten Methoden für robotische Wahrnehmung und Manipulation und schlägt ein Framework vor, das von der synthetischen Datengenerierung über strukturbewusste visuelle Modellierung bis hin zu robusten Manipulationsstrategien reicht. Ziel ist es, eine zuverlässige Sim2Real-Übertragung in Szenarien mit komplexen und vielfältigen Objekten zu ermöglichen.

Zur Reduzierung der Domänenlücke zwischen simulierten und realen Tiefendaten industrieller Streifenlichtsensoren wurde ein physikalisch konsistenter Streifenlichtsimulator entwickelt. Durch die präzise Modellierung von Gray-Code-Projektion, Sensoreigenschaften und Oberflächeninteraktionen in einem fotorealistischen Renderer erzeugt das System synthetische Tiefenkarten, die den realen Messdaten sehr nahekommen. Damit wird das Training von tiefenbildbasierten Wahrnehmungsmodellen mit verbesserter Sim2Real-Leistung ermöglicht, wodurch der Bedarf an aufwendiger manueller Annotation realer Daten erheblich reduziert wird.

Für die Tiefenschätzung auf Basis von Stereo-Bildpaaren, insbesondere bei transparenten und überlappenden Objekten, werden zwei komplementäre Modelle vorgeschlagen. Das erste verbessert die Robustheit der Stereo-Korrespondenz in strukturalarmen Bildregionen mithilfe von Transformer-basierten Merkmalfusionen und geometrischen Priors. Das zweite Modell erlaubt eine Zero-Shot-Stereorekonstruktion, indem ein vortrainierter Encoder eingefroren und ein leichtgewichtiges Matching-Modul trainiert wird, um auch ohne Feinabstimmung eine starke Generalisierungsfähigkeit zu erreichen.

Im Bereich der Manipulation wird ein kollisionsbewusstes Greifnetzwerk für verformbare Kabel vorgestellt. Dieses wurde auf physikalisch simulierten Daten mit annotierten Kontakt- und Kollisionsinformationen trainiert und sagt die Greifbarkeit und Sicherheit anhand von Tiefenbildern vorher, wobei das Modell Kontakt- und Kollisionshinweise implizit erlernt. Für die beschädigungsfreie Handhabung empfindlicher Früchte wurde ein vision-guided Softgripper-System mit multimodaler Rückkopplungssteuerung entwickelt, das in realitätsnahen Anwendungsszenarien eine stabile und sichere Leistung zeigt.

Die vorliegende Arbeit stellt wesentliche Fortschritte in der fotorealistischen Simulation, der strukturbasierten visuellen Modellierung und der flexiblen Objektmanipulation vor. Insgesamt wird eine praxisorientierte Grundlage geschaffen, um robuste Sim2Real-Anwendungen von Robotersystemen in komplexen realen Aufgabenstellungen zu ermöglichen.

Contents

Abstract	V
Zusammenfassung	VII
1 Introduction	1
1.1 Motivation	1
1.2 Research Questions	3
1.3 Aim of this Thesis	4
1.4 Novelty and Contribution of this Thesis	5
1.5 Thesis Structure	6
2 Preliminaries and Related Work	9
2.1 Sim-to-Real Transfer in Robotic Perception	10
2.1.1 Domain Randomization	10
2.1.2 Domain Adaptation and Alignment	13
2.2 Generalizable Perception for Transparent Objects	15
2.2.1 Sensors and Simulation Platforms	16
2.2.2 Transparent Object Segmentation	19
2.2.3 Transparent Object Depth Completion and Reconstruction	21
2.3 Physically Realistic Depth Sensor Simulation	25
2.3.1 Structured Light Depth Sensor Simulation	25
2.3.2 Active Stereo Depth Sensor Simulation	26
2.3.3 Time-of-Flight Depth Sensor Simulation	26
3 Close the Sim2Real Gap via Physically-based Camera Simulation	28
3.1 Research Motivation and Challenges for Structured Light Depth Sensing	28
3.1.1 Industrial Challenges in Robotic Perception	28
3.1.2 The Importance of Physically-based Camera Simulation	29
3.1.3 Contributions and Innovations	29
3.2 Challenges and Design Considerations in Synthetic Depth Simulation .	30
3.2.1 Synthetic Dataset Generation and Sim2Real Gap	30
3.2.2 Taxonomy of Depth Sensor Simulation Methods	31
3.2.3 Role of Physically-Based Rendering in Accuracy	31
3.3 Physically-Based Simulation Framework	31
3.3.1 Rendering Pipeline Using Ray-Tracing	33

3.3.2	Structured Light Pattern Projection Methodology	34
3.4	Synthetic and Real-World Dataset Construction for Sim2Real Evaluation	37
3.4.1	Object Database	37
3.4.2	Synthetic and Real-World Dataset Creation	38
3.4.3	Annotation Formats and Ground-Truth Generation	38
3.5	Experimental Evaluation	39
3.5.1	Task Setup and Evaluation Metrics	39
3.5.2	Object Detection and Instance Segmentation	39
3.5.3	Domain Adaptation vs. Domain Randomization	41
3.6	Robotic Grasping Experiments	42
3.6.1	Model-based and Model-free Grasping Setup	42
3.6.2	Grasping Performance and Deployment Efficiency	43
3.7	Discussion	43
4	Robust Stereo Perception for Transparent Objects	45
4.1	Research Motivation and Challenges in Transparent Object Perception .	45
4.1.1	The Role of Depth Accuracy in Robotic Manipulation	45
4.1.2	Dataset scarcity and simulation gap	46
4.1.3	Limitations of existing methods	47
4.1.4	Summary of challenges	48
4.2	System Architecture for Transparent Object Stereo Perception	48
4.2.1	Cascaded Vision Transformer Backbone	50
4.2.2	Structural Feature Post-Fusion	51
4.2.3	Synthetic Dataset Generation	52
4.3	Evaluation	54
4.3.1	Stereo Matching Evaluation Metrics	55
4.3.2	Ablation Study of Feature Post-Fusion Module	56
4.3.3	Evaluation of Transparent Object Dataset	57
4.4	Robotic Experiments	59
4.5	Discussion and Future Work	63
5	Zero-Shot Stereo Perception for Robotic Manipulation	64
5.1	Research Motivation and Challenges for Zero-Shot Stereo Perception .	64
5.1.1	Limitations of Task-Specific Adaptation in Zero-Shot Stereo Perception	65
5.1.2	Generalization Gap and Catastrophic Forgetting	65
5.1.3	Affordance-Guided Multi-Finger Grasp Strategies	65
5.1.4	Contributions and Paper Organization	66
5.2	Foundation Models and Zero-Shot Adaptation	66
5.3	Method Overview of the Zero-Shot Stereo Grasping System	67
5.3.1	Zero-Shot Stereo Reconstruction Module (SRM)	68
5.3.2	Point Cloud Grasp Generation Module (PCGG)	68
5.3.3	Five-Finger Grasping Affordance Detection Module (FFGAD) .	68
5.4	Implementation Details	69
5.4.1	Extending Foundation Models with LoRA	69

5.4.2	Synthetic Transparent Object Dataset Generation	71
5.4.3	Affordance Detection and FFH Grasp Sample Generation	72
5.5	Experimental Setup and Evaluation Protocol	73
5.5.1	Dataset and Training Configuration	73
5.5.2	Robotic Grasping Evaluation Setup	73
5.6	Results and Analysis	74
5.6.1	Comparison with SOTA Stereo Models	74
5.6.2	Baseline: Depth Anything V2 vs. StereoAnything	74
5.6.3	Foundation-Model Encoder Extension Study	76
5.6.4	Ablation Studies on Training Strategies and Iterative Inference Trade-offs	77
5.7	Conclusion and Discussion	78
6	Collision-aware Sim2Real Robotic Grasping of Fragile Objects	79
6.1	Research Motivation for Collision-aware Robotic Grasping	79
6.1.1	Challenges in Grasping Fragile Fruit in Stacked Scenes	79
6.1.2	Limitations of Existing Robotic Grasping Methods	80
6.2	Sim2Real Grasping Framework Design	80
6.2.1	Grasp Pose Estimation Network: PG-CNN	81
6.2.2	Grasp Execution Pipeline	82
6.3	Sim2Real Dataset Construction	84
6.3.1	Synthetic Stereo Dataset: PeachStereo	84
6.3.2	Synthetic Grasping Dataset: PeachGrasp	84
6.3.3	Transfer Learning Pipeline	85
6.4	Experimental Evaluation	85
6.4.1	Collision-aware Grasp Pose Prediction in Simulation	85
6.4.2	Grasp Execution Results in Real-world Scenarios	87
6.4.3	Failure Case Analysis	87
6.5	Summary and Future Directions	88
6.5.1	Strengths and Contributions	88
6.5.2	Discussion and Future Directions	89
7	Conclusions and Outlook	90
7.1	Conclusion	90
7.2	Discussion and Open Challenges	91
7.3	Future Research Directions	92
A	List of Abbreviations	95
B	Publications	99
C	Acknowledgements	101
	Bibliography	103

List of Figures

1.1	Examples of challenging real-world objects for robotic manipulation. . .	2
1.2	The structure of this thesis.	6
2.1	Illustration of domain randomization and domain adaptation. Source Domain: synthetic images used for model training. Target Domain: real-world images encountered during deployment.	10
2.2	Examples of domain randomization across different simulation factors. .	11
2.3	Simulated modalities and domain adaptation across depth sensors. . . .	15
3.1	Sim2Real perception and grasping framework.	32
3.2	Object Database. a) Metal Workpiece 1. b) Metal Circle Workpiece 2. c) YellowSaltCube for KIT object model database. d) HygieneSpray from KIT object model database. e) Toothpaste from KIT object model database. f) The corresponding real objects.	37
3.3	Physically-based structured-light simulation and reconstruction pipeline.	38
3.4	Qualitative visualization results in the real-world dataset using models trained on our synthetic datasets: (a) gray image inputs; (b) proposed depth image inputs.	40
3.5	Examples from the domain-randomized dataset generated using the Isaac Sim platform.	41
3.6	Experimental setups for robotic grasping: (left) model-based approach; (right) half-model-based approach.	42
4.1	Overview of the proposed transparent object stereo reconstruction and robotic grasping system.	49
4.2	Our stereo depth recovery network for transparent objects.	50
4.3	Synthetic dataset generation pipeline with physically-based rendering and learning-based acceleration.	53
4.4	Synthetic and real transparent object sets used in SynClearDepth.	54
4.5	Sample images from our SynClearDepth dataset.	55
4.6	Visualization results of our method compared with other SOTA stereo depth estimation methods.	57
4.7	Comparison with ClearGrasp [94], TransCG [31], and ASGrasp [96]. . .	59
4.8	Comparison with Dex-NeRF [40].	60
4.9	Performance under different lighting conditions.	60

4.10	Experimental setup for transparent object grasping. The setup includes a Diana 7 robot with a Robotiq 85 gripper, a ZED2 camera for robotic experiments, and additional cameras (ZED2, RealSense D435) for visual experiments.	61
4.11	Qualitative comparison of transparent object grasping using depth reconstruction from ZED2, TransCG [31], and our ClearDepth. Grasp candidates are generated using GraspNet-Baseline [30].	62
4.12	Representative examples of grasping failures, including (left) noisy grasp areas caused by transparent perception, (middle) failed depth recovery, and (right) grasping in invisible regions due to estimation errors.	62
4.13	Error distribution of baseline method, TransCG [31] and our ClearDepth. We compare the proportions of total failures represented by different failure types.	63
5.1	Overall architecture of StereoAnything.	67
5.2	Pipeline of the Five-Finger Grasping Affordance Detection Module (FF-GAD).	69
5.3	LoRA placement in encoder (transformer blocks 18–24).	70
5.4	Examples of synthetic transparent object training data used in this work.	72
5.5	Robotic configurations for grasping evaluation.	74
5.6	Qualitative comparison with SOTA stereo models. [5]	75
5.7	Comparison between monocular depth predictions from Depth Anything V2 and stereo depth outputs from StereoAnything on transparent object scenes.	75
6.1	Overview of the proposed Sim2Real grasping framework for fragile fruit manipulation, including simulation-based training and real-world deployment.	81
6.2	Overall architecture of the proposed PG-CNN.	82
6.3	Stereo image examples from the PeachStereo dataset.	84
6.4	Visualization of sampled grasp candidates and PG-CNN predictions in simulation and real-world-like setups.	86
6.5	Visualization of predicted grasp poses overlaid on real depth images.	87

List of Tables

3.1	Quantitative evaluation results of object detection and instance segmentation.	40
4.1	Comparison of disparity map quality metrics for different methods. . . .	56
4.2	Comparison with fine-tuned stereo SOTA methods on the transparent object dataset.	57
4.3	Quantitative comparison on the transparent object validation set (no fine-tuning for baselines).	58
4.4	Real-world robotic grasping comparison experiments with SOTA methods for transparent objects.	61
5.1	Quantitative comparison with state-of-the-art algorithms across different scenarios	76
5.2	Comparison with other foundation model image encoders on transparent object scenes.	76
5.3	Validation results of ablation study for training methods.	77
6.1	Grasping success rate in the physical world with different number of peaches.	86

Chapter 1

Introduction

1.1 Motivation

In the long-term vision of bringing artificial intelligence systems out of the laboratory and into real-world applications, robots are increasingly expected to possess complex capabilities, not only to perceive and understand their surroundings, but also to interact stably, efficiently, and intelligently with a wide variety of physical objects. Across application domains such as manufacturing, service robotics, and agriculture, these capabilities are fundamentally embodied in the robot’s manipulation skills. From basic pick-and-place tasks to complex assembly and the manipulation of deformable or fragile objects, successful execution relies on accurate modeling and perception of object geometry, physical properties, and interaction behaviors.

However, the real world is far more complex than controlled lab environments. Robots often operate in unstructured, dynamic, and high-interference settings, where objects may have ambiguous boundaries, reflective surfaces, or deformable and fragile characteristics. Representative examples of such challenges are illustrated in Fig. 1.1, from left to right: a bin of transparent objects, a bin of reflective metal parts, a bin of delicate fruits, and a bin of flexible cables. These object types represent key challenges in perception and manipulation: transparency, specular reflectivity, fragility, and deformability, all of which are common in real-world industrial and service contexts, yet pose significant challenges to current state-of-the-art perception and robotic manipulation systems.

Although deep learning has achieved impressive results on certain benchmark tasks, most models are highly dependent on large amounts of high-quality annotated data, especially in 3D perception tasks, where precise geometry, pose, or interaction labels are required. For the aforementioned complex objects, real-world data collection is extremely expensive, and in industrial scenarios, heavy parts can lead to safety concerns during data acquisition. Furthermore, in dense and occluded environments such as industrial sorting, multi-instance labeling becomes nearly infeasible. As a result, these datasets are often incomplete or unrepresentative, limiting the generalization ability of learned models. Moreover, many industrial and household objects inherently interfere with sensor measurements: transparent or reflective materials may not be captured prop-

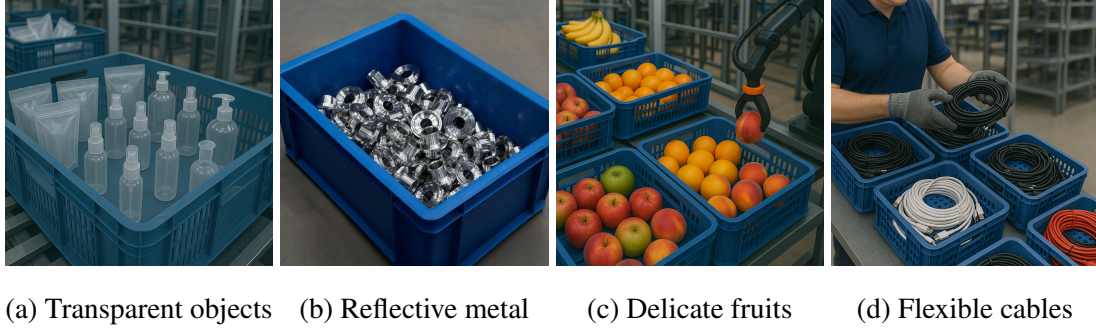


Figure 1.1: Examples of challenging real-world objects for robotic manipulation.

erly; deformable objects are often heavily occluded; and cluttered scenes with overlapping objects degrade stereo matching and 3D reconstruction, ultimately impacting downstream manipulation and grasping tasks.

Even when large-scale datasets are generated in simulation, simulation-to-reality (Sim2Real) transfer remains a critical bottleneck for deployment. Most mainstream simulators fail to replicate the key sensor characteristics and physical interactions observed in the real world. For example, in industrial settings where gray-code structured light 3D cameras are commonly used, factors such as depth noise, non-ideal lighting conditions, and edge artifacts at depth discontinuities are often oversimplified or ignored in the synthetic ground truth depth maps. Consequently, models trained on such data typically underperform in real-world environments. Similar challenges also arise in Sim2Real transfer for robotic grasping of industrial parts. In current physics-based Sim2Real pipelines, the simulation process is often simplified, resulting in insufficient modeling of physical interactions and contact dynamics between the gripper, the target object, and the surrounding environment. As a consequence, models trained in simulation frequently exhibit collision-prone behaviors and limited robustness when deployed in the real world.

In addition, most existing stereo vision models lack explicit structural and semantic awareness, relying predominantly on pixel-level appearance matching to infer depth. While such correspondence-based formulations are effective under favorable imaging conditions, they break down in scenarios involving transparent materials, textureless surfaces, or ambiguous object boundaries, where reliable visual cues are inherently sparse or misleading. In contrast, human depth perception is not solely driven by binocular disparity. Humans are able to robustly estimate depth across a wide range of materials and visual conditions by jointly leveraging binocular cues and strong prior knowledge about object structure, geometry, and physical plausibility. This integration of sensory input with learned priors allows the visual system to resolve ambiguities that remain fundamentally challenging for purely appearance-driven stereo algorithms. Most contemporary stereo learning approaches, however, do not explicitly incorporate such priors, and consequently exhibit limited generalization when faced with distribution shifts. This limitation becomes particularly pronounced under zero-shot or few-shot settings, where models trained on narrowly scoped datasets struggle to transfer to novel object categories, materials, or imaging conditions. As a result, the deployability of current

stereo vision systems in open, real-world environments remains severely constrained.

Taken together, these challenges reveal that current robotic perception and manipulation systems are still far from being robustly deployable in real-world environments. Key limitations include unreliable depth perception, lack of sufficient labeled data, and significant performance degradation due to domain shift. These limitations directly impact performance across the entire manipulation pipeline, including scene understanding, object perception, and grasp or manipulation execution.

To bridge the gap between research and real-world deployment, a systematic transformation of existing robotic perception frameworks is urgently needed:

- By leveraging physically-based ray tracing rendering engines, we construct physically-realistic, high-fidelity simulators that generate RGB and depth training data more closely aligned with real sensor outputs.
- Incorporating structure-aware mechanisms into deep learning models, together with large-scale, high-quality datasets tailored to complex object perception, enables the design of imaging networks that approximate human-relevant geometry understanding. These models are critical for improving depth estimation and perception performance in challenging areas such as transparent occlusions or fuzzy boundaries, and directly support service robots operating in household scenarios or industrial robots performing multi-category product recognition.
- Integrating simulation-aided manipulation policy learning within physical engines, we efficiently generate task experience for deformable and non-rigid objects, enhancing generalization while avoiding the risk of damaging real-world targets during data collection.

This dissertation is motivated by these pressing challenges. It focuses on improving robotic perception and manipulation in unstructured and complex environments, and centers around three core directions: physically-realistic synthetic data generation, structure-aware imaging model development, and Sim2Real policy learning for object manipulation. These approaches are designed to complement one another and jointly contribute to enabling robust, real-world deployment of intelligent robotic systems.

1.2 Research Questions

As outlined in Section 1.1, this dissertation investigates the challenges of robotic perception and manipulation in scenarios involving complex materials and non-rigid objects. The focus is on enhancing the generalization and stability of robotic systems under conditions of limited training data, weak visual cues, and significant Sim2Real gaps. To this end, the research is organized around the following four core questions:

Q1: How can the simulation of structured-light depth sensors overcome the Sim2Real gap?

Most existing simulation platforms fail to accurately reproduce the projection mechanisms, material reflection behaviors, and depth measurement noise characteristics of

real industrial structured-light sensors. This discrepancy severely limits the transferability of depth-based models to real-world deployments. This research seeks to answer: how can a physically-consistent structured-light simulation system be constructed to generate RGB-D data that faithfully match real sensor outputs and support depth-based perception tasks in practical environments?

Q2: How can stereo vision models be designed to incorporate structural awareness for scenes with transparency, occlusion, and weak texture?

Traditional stereo matching models often rely heavily on texture and perform poorly in regions with discontinuous or ambiguous structure, such as transparent or overlapping objects. This research asks: how can geometric priors and architectural guidance be introduced into stereo vision models to enable explicit structure modeling and improve depth estimation performance in visually challenging conditions?

Q3: How can zero-shot generalization be achieved in stereo vision models without access to target domain data?

Most state-of-the-art depth estimation approaches rely on large-scale annotated data and require fine-tuning in the target domain, which limits their adaptability. This dissertation explores: how can a zero-shot stereo vision framework be constructed using a frozen pre-trained visual encoder and a lightweight matching module, enabling robust depth estimation in unseen environments without the need for additional training?

Q4: How can transferable grasping strategies be learned in simulation and applied to real-world manipulation of delicate objects?

Grasping tasks involving delicate or deformable objects, such as cables and fruits, are prone to failure due to deformation, collision risks, and perception errors. Moreover, current research lacks systematic simulation-to-real learning pipelines for such tasks. This research investigates: how can grasp attempts and collision feedback be generated in simulation, and how can a model be trained to leverage them to evaluate grasp stability and predict safe actions, ultimately achieving transferable and low-damage manipulation strategies in real-world scenarios?

1.3 Aim of this Thesis

The aim of this thesis is to advance robotic perception and manipulation by addressing the challenges of transparent object perception, deformable object manipulation, and sim-to-real transfer. To this end, the dissertation develops simulation frameworks, learning-based vision systems, and grasping strategies that improve robot robustness in complex, real-world scenarios. The scientific problems this thesis seeks to solve are:

- **Physically-based simulation for synthetic data generation.** Develop a simulation framework capable of generating physically realistic RGB-D data, including structured-light and ray-tracing pipelines, in order to close the gap between synthetic and real-world sensing.
- **Transparent object perception.** Design stereo-based and learning-driven perception systems that can robustly reconstruct and localize transparent or reflective objects, where conventional depth sensors and geometric methods fail.

- **Zero-shot stereo and generalization.** Leverage foundation models and structure-guided architectures to achieve zero-shot stereo matching and robust 3D reconstruction in novel environments without task-specific fine-tuning.
- **Grasping in complex environments.** Develop grasp planning and execution pipelines that integrate perception with reasoning to enable reliable grasping of fragile or cluttered objects, including collision-aware strategies.

Together, these contributions aim to bridge the gap between simulation and reality, enabling service and industrial robots to perceive and manipulate objects more effectively in everyday unstructured environments.

1.4 Novelty and Contribution of this Thesis

The main contributions of this thesis are summarized as follows:

- **Physically-based Simulation for Synthetic Data Generation.** We propose a physically-based simulation framework for generating large-scale synthetic datasets tailored for robotic perception. The framework integrates ray tracing and structured-light projection to reproduce optical phenomena such as refraction, reflection, and scattering, which are critical for transparent-object perception. In addition, domain randomization and domain adaptation are incorporated to bridge the sim-to-real gap. To the best of our knowledge, this is the first work that systematically applies physically-based structured-light simulation for transparent-object data generation in robotics.
- **Transparent object Perception with Stereo Vision.** We design learning-based stereo vision models that explicitly address the challenges of transparent and reflective surfaces, where conventional RGB-D sensors fail. A cascaded Transformer backbone and structural-guided fusion module are introduced to enhance disparity estimation in complex scenes. Comprehensive experiments demonstrate that the proposed system outperforms prior stereo networks on synthetic and real-world benchmarks for transparent-object reconstruction.
- **Zero-shot Stereo with Foundation Models.** Building on recent advances in vision foundation models, we introduce a zero-shot stereo perception framework (StereoAnything) that enables robust 3D reconstruction without task-specific fine-tuning. The method combines a pre-trained dual-path backbone with lightweight adapters and a geometry-aware head, achieving superior generalization to unseen transparent objects and environments. To the best of our knowledge, this represents the first attempt to extend large-scale stereo pretraining to transparent-object perception in robotics.
- **Sim-to-Real Grasping with Transparent and Fragile Objects.** We propose an end-to-end sim-to-real grasping pipeline that integrates perception, reasoning, and execution. A collision-aware grasping strategy (PG-CNN + reasoning module)

is developed for fragile fruit grasping, supported by two novel datasets (Peach-Stereo and PeachGrasp) for perception and manipulation transfer. The framework demonstrates reliable performance in both simulation and real-world experiments, achieving robustness against occlusions, object shape variations, and collisions. This is the first work to systematically address transparent-object perception and fragile-object grasping within a unified sim-to-real framework.

Together, these contributions establish a comprehensive pipeline that combines physically-based simulation, stereo-based transparent-object perception, zero-shot generalization, and collision-aware grasping. They advance the state of the art in closing the reality gap for robotic perception and manipulation.

1.5 Thesis Structure

The thesis structure is illustrated in Fig. 1.2. This thesis centers on simulation-to-reality (Sim2Real) transfer for robotic perception and manipulation, with emphasis on physically based sensor simulation, stereo depth estimation for transparent objects, zero-shot stereo for grasping, and collision-aware grasping of fragile objects. The overall structure is illustrated in Fig. 1.2. A brief summary of each chapter follows.

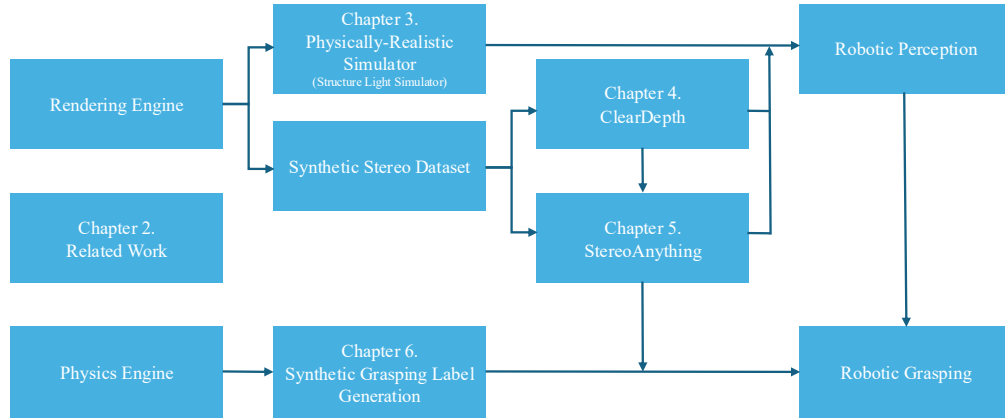


Figure 1.2: The structure of this thesis.

Chapter 2: Preliminaries and Related Work.

This chapter provides a comprehensive survey of prior research on simulation and perception in robotics. We first review physically realistic depth sensor simulation and structured-light imaging, followed by synthetic data generation methods for visual tasks and robotic grasping. We then analyze the development of data-driven Sim2Real stereo networks, with a particular focus on their limitations when dealing with transparent or reflective materials. Finally, we summarize progress in learning-based grasping from visual inputs, highlighting the strengths and shortcomings of existing simulation-based pipelines. This review establishes the technical and conceptual background for the subsequent chapters.

Chapter 3: Close the Sim2Real Gap via Physically-based Camera Simulation.

This chapter introduces a physically-based rendering and simulation framework designed to narrow the gap between synthetic and real-world data. We present the rendering pipeline, which integrates ray tracing and structured-light projection, and describe the construction and annotation of synthetic datasets. A series of experimental studies are conducted to assess the effectiveness of domain randomization and domain adaptation strategies in visual perception tasks such as object detection and instance segmentation. The chapter further reports robotic grasping experiments that validate the practical utility of the generated data, and concludes with a discussion of strengths, limitations, and lessons learned from this approach.

Chapter 4: Robust Stereo Perception of Challenging Transparent Objects.

This chapter proposes *ClearDepth*, a stereo vision system specifically designed for transparent-object perception. We detail the cascaded Vision Transformer backbone and the structural feature post-fusion mechanism that enables improved depth estimation in challenging transparent and reflective scenes. A synthetic dataset tailored to transparent objects is presented, followed by a comprehensive evaluation that includes stereo matching metrics, ablation studies on the fusion module, and qualitative and quantitative analyses of depth prediction. We further assess dataset generalization and demonstrate the system’s applicability in robotic experiments, before concluding with a discussion of open challenges and future research directions.

Chapter 5: Zero-Shot Stereo Imaging for Robotic Manipulation.

Building on *ClearDepth*, this chapter introduces *StereoAnything*, a zero-shot stereo framework for robotic grasping of transparent objects. The architecture integrates three main components: a zero-shot stereo reconstruction module (SRM), a point-cloud grasp generation module (PCGG), and a five-finger grasp affordance detection module (FF-GAD). We describe the adaptation of foundation models (e.g., through low-rank adaptation) for stereo tasks, the design of the integrated pipeline, and the generation of synthetic transparent-object data for training. Experimental evaluations compare the proposed system against state-of-the-art stereo models and baselines, including Depth Anything V2, highlighting both its strengths in generalization and its remaining limitations.

Chapter 6: Collision-aware Sim2Real Robotic Grasping of Fragile Objects.

This chapter presents a complete Sim2Real framework for grasping fragile fruits, which represents a particularly challenging domain due to the requirements of precision and delicacy. The framework consists of a perception module for grasp pose estimation (PG-CNN), a grasp execution pipeline, and newly constructed datasets (PeachStereo and PeachGrasp) that enable transfer learning from simulation to reality. We report systematic evaluations in simulation and real-world environments, including collision-aware grasp prediction, execution success rates, and failure analyses covering slip, perception errors, and grasp planning mistakes. The framework is further tested for cross-species generalization and robustness across variations in fruit size and shape. The chapter concludes by reflecting on the strengths and contributions of the framework, while acknowledging its current limitations.

Chapter 7: Conclusions and Outlook.

The final chapter summarizes the main contributions of the dissertation across simulation, perception, and robotic execution. It reflects on the lessons learned regard-

ing physically-based data generation, transparent-object perception, and collision-aware grasping. Limitations of the current approaches are critically discussed, and directions for future research are proposed, including robust multimodal learning, large-scale synthetic data generation, and broader generalization of Sim2Real techniques to complex real-world manipulation scenarios.

Chapter 2

Preliminaries and Related Work

As discussed in Chapter 1, achieving robust robotic perception and manipulation in real-world environments remains a significant challenge. This dissertation focuses on two key issues: effective visual modeling in scenes involving complex materials (such as transparent objects) and fragile objects (such as fruits), and robust grasping strategy learning based on simulation and perception. These challenges span multiple research areas, including synthetic data generation, sensor simulation, visual modeling, stereo matching networks, grasp strategy optimization, and Sim2Real transfer methods.

Unlike conventional robotic tasks that primarily involve rigid and well-structured objects, the targets addressed in this work often exhibit complex material properties, deformability, and ambiguous geometric boundaries. Such characteristics make it difficult to directly apply classical vision and control methods for modeling and manipulation. Additionally, data acquisition for these objects in the real world is challenging and often lacks high-quality annotations, which further limits the generalization ability of data-driven models. As a result, recent research has increasingly turned to high-fidelity simulation platforms for synthetic data generation, along with multimodal modeling techniques to enhance system robustness and applicability.

This chapter systematically reviews the research background and recent progress closely related to this dissertation. Section 2.1 reviews key strategies to bridge the reality gap in robotic perception, focusing on two complementary approaches: *Domain Randomization*, which introduces variability in simulation to encourage robustness, and *Domain Adaptation and Alignment*, which explicitly reduces discrepancies between synthetic and real data. Together, they illustrate how diversity and alignment enable effective sim-to-real transfer.

In summary, this chapter reviews the current state of the art in simulation modeling, visual perception, and grasp control for complex object scenarios. These studies provide the theoretical foundation and contextual background for the system-level methods and experiments proposed in the following chapters.

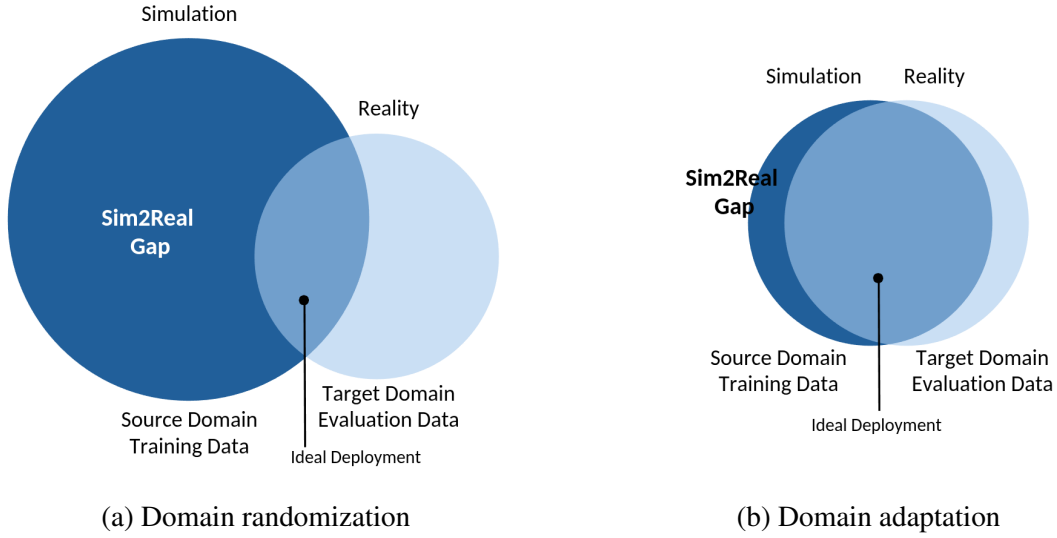


Figure 2.1: Illustration of domain randomization and domain adaptation. Source Domain: synthetic images used for model training. Target Domain: real-world images encountered during deployment.

2.1 Sim-to-Real Transfer in Robotic Perception

2.1.1 Domain Randomization

Domain randomization (DR) addresses the *reality gap* by training perception or control models in simulators where key scene, rendering, and physics parameters are deliberately randomized. Sufficient variability forces the model to learn invariances (e.g., to textures, lighting, and sensor noise), enabling direct sim-to-real transfer without (or with minimal) real-world fine-tuning. As illustrated in Fig. 2.1(a), the training process is conducted entirely in the source domain, which is synthetically generated with diverse randomized conditions. The overlap with the target domain represents the portion of real-world variations that are sufficiently covered by the randomized simulation. By exposing the model to such broad variability, DR encourages the learning of features that generalize beyond simulator-specific biases. Early works [103, 104] demonstrated that models trained solely on highly randomized synthetic images can localize objects and detect categories in the real world with competitive accuracy.

Where to Randomize. Domain randomization can be applied at multiple levels:

- **Scene construction level:** randomizing the number, geometry, and spatial arrangement of target and distractor objects, as well as background elements, occluders, and camera viewpoints.
- **Rendering and sensor level:** randomizing material appearance and textures, lighting conditions, camera parameters (e.g., intrinsics), and imaging effects such as exposure variation, motion blur, and explicit sensor noise models (e.g., read noise and quantization artifacts).

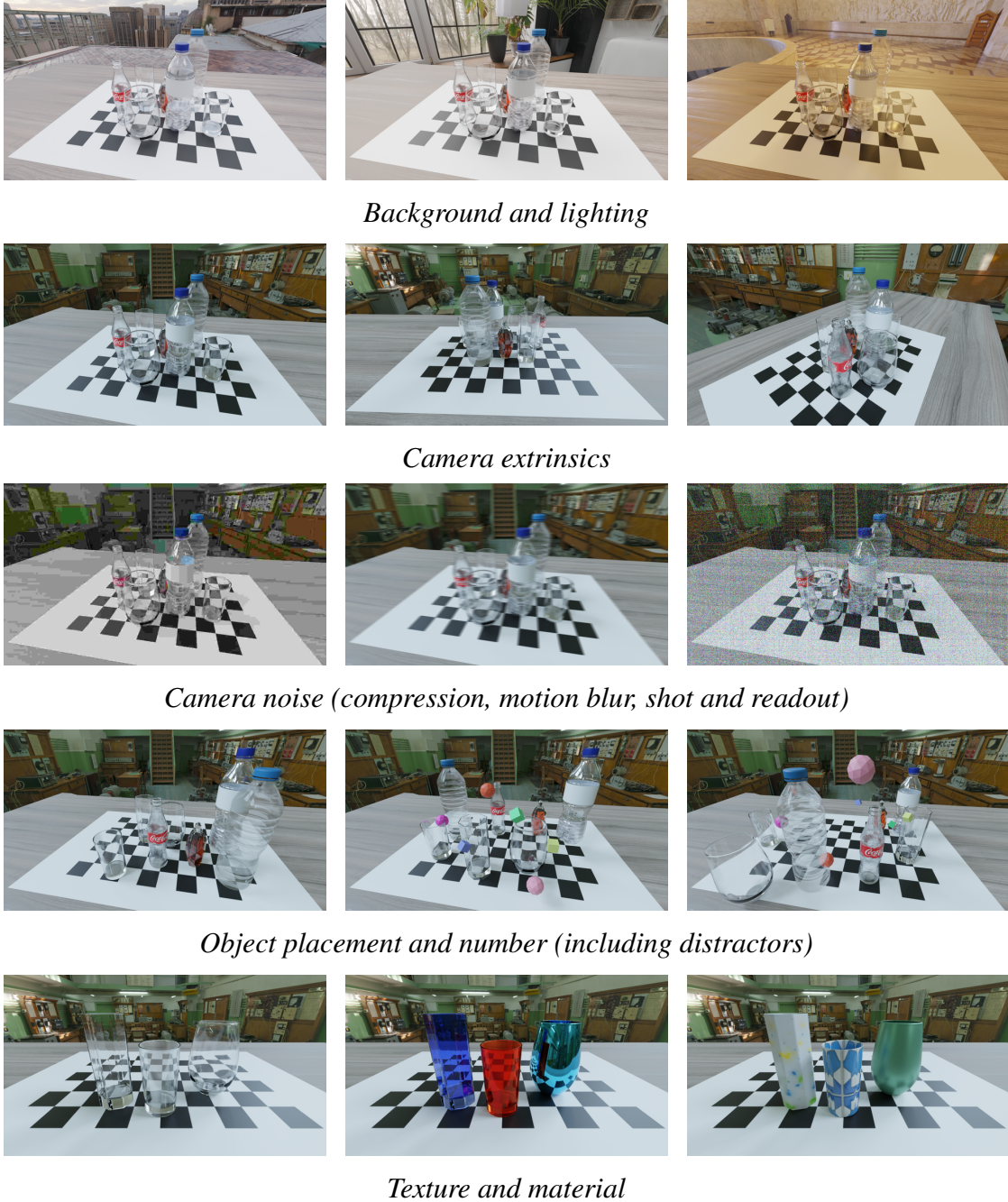


Figure 2.2: Examples of domain randomization across different simulation factors.

As illustrated in Fig. 2.2, we provide concrete examples of domain randomization (DR) across simulation factors that span the data generation pipeline. At the *scene and construction level*, we randomize the number of target and distractor objects as well as their placements, and we vary camera viewpoints to expose the model to diverse spatial configurations. At the *rendering and sensor level*, we diversify background appearance, lighting conditions, and material appearance and textures to prevent overfitting to specific visual cues. We further perturb the rendered observations with imaging artifacts

and sensor noise, including motion blur, compression-related artifacts, and physically motivated shot and read noise. Together, these examples demonstrate how domain randomization operates across both the *scene construction* and *rendering and sensor* levels, complementing the taxonomy summarized above.

Variants of Domain Randomization. Classic DR favored *diversity over realism* [103]. Later, *structured DR* constrained randomization within physically plausible ranges while preserving scene consistency [10]. More recently, *learned DR* approaches optimize the sampling distributions to maximize sim-to-real transfer performance [25, 108]. In addition, DR has been combined with physically-based rendering [141] and neural rendering [138] to achieve both coverage and realism.

Applications in Robotic Vision. Domain randomization has been successfully applied to a range of robotic vision tasks, including object localization [103], detection [104], and grasping [102]. The key finding is that heavy randomization of textures, lighting, and camera viewpoints encourages models to rely on geometric and structural cues rather than fragile appearance details. For example, Sadeghi and Levine [93] demonstrated that quadrotor navigation policies trained purely with randomized synthetic images could transfer directly to real-world flights. Subsequent work refined the method by constraining randomization to physically plausible ranges or by optimizing the sampling distributions. Together, these developments demonstrate how diversity in synthetic training data can play a crucial role in improving sim-to-real transfer.

Summary. Domain randomization is a scalable strategy for Sim2Real transfer that intentionally introduces controlled variability across simulated environments, with the goal of covering a wide range of real-world appearance and dynamics. Rather than striving for strict photorealism, DR exposes a learning system to diverse visual and physical cues during training so that the resulting models focus on task-relevant, invariant features rather than simulator-specific artifacts. This paradigm is particularly attractive when real-world data are scarce, expensive, or difficult to acquire, and in many cases enables effective Sim2Real transfer with little to no real data during training, in contrast to domain adaptation approaches that rely on sufficient target-domain samples.

In practice, DR is typically realized by randomizing multiple factors in the data generation pipeline, including object layouts and geometries, textures and materials, lighting conditions, camera parameters, sensor and imaging effects, and, in some contexts, dynamics or physics-related parameters. By sampling from a broad randomized configuration space, DR aims to approximate the variability encountered in real environments, thereby reducing the reality gap and improving generalization to previously unseen real-world scenarios.

Nevertheless, domain randomization is not a panacea. Its effectiveness critically depends on how well the chosen randomization space captures the underlying structure of real-world variability. Randomization alone cannot compensate for inaccurate or oversimplified physical, optical, or sensing models, and excessive or poorly targeted randomization in high-dimensional parameter spaces may lead to inefficient learning and slow

convergence. Moreover, while DR enhances robustness to appearance and environmental variations, it does not inherently enforce geometric consistency, physical plausibility, or contact-aware reasoning, which are often essential for reliable perception and interaction in complex real-world settings. These limitations highlight the need to complement domain randomization with more faithful physical modeling and structured priors, especially in perception tasks involving challenging materials and sensing modalities.

Specifically, our work addresses the limitation of domain randomization in capturing the structured physical image formation process of structured light sensing, which cannot be resolved by stochastic variability alone.

2.1.2 Domain Adaptation and Alignment

In the process of transferring from simulation to the real world, there usually exists a significant discrepancy between the source domain with synthetic data and the target domain with real-world data. This issue is commonly referred to as *domain shift* or the *reality gap*. The goal of Domain Adaptation (DA) is to leverage the abundant annotations in the source domain to improve performance in the target domain, particularly when labeled real-world data is scarce or unavailable [107, 32]. As illustrated in Fig. 2.1(b), DA attempts to reduce the discrepancy between the two domains by explicitly aligning feature distributions. The increasing overlap between the source and target domains highlights the effectiveness of adaptation in narrowing the reality gap. Depending on the availability of supervision in the target domain, DA methods can be broadly categorized into supervised, weakly supervised, and unsupervised. Among them, Unsupervised Domain Adaptation (UDA) has become the dominant research direction, since large-scale labels in the real world are often difficult or expensive to obtain [117].

However, in robotic perception and control tasks, DA is not only about feature alignment at the algorithmic level, but also closely tied to reducing discrepancies at the simulator and renderer level. This perspective emphasizes aligning the data generation process itself with the target domain, for instance, through high-fidelity physically based rendering to reproduce realistic lighting, materials, and sensor noise [105], or through system identification to tune dynamics parameters toward real robot behavior [13]. Thus, simulator and renderer design is complementary to algorithmic DA methods, together constituting the core strategies for tackling the sim-to-real challenge.

Existing DA methods can be divided into several levels:

- **Pixel-level approaches:** reduce appearance gaps via image-to-image translation. For example, SimGAN [97] employs adversarial training to refine synthetic images to look more realistic, while CycleGAN and CyCADA [83] enforce cycle consistency to preserve structural content, enabling more robust transfer in vision tasks.
- **Feature-level approaches:** align distributions directly in the representation space. DANN [32] introduces a gradient reversal layer to learn domain-invariant features, CORAL [98] performs statistical matching, and DSN [8] separates shared and private features to enhance generalization.

- **Model- and training-level approaches:** emphasize architectural transferability, such as Progressive Nets [92], which enable progressive transfer across tasks or domains, or the method of Cully et al. [24], which leverages simulation knowledge as a prior to improve real-world exploration efficiency.
- **Pseudo-labeling and self-training:** exploit source-trained models to generate pseudo-labels in the target domain and iteratively refine performance [146]. Test-time and source-free adaptation methods adjust model parameters directly at inference time, for example by updating batch normalization statistics or minimizing prediction entropy [65, 111].
- **Simulator-level approaches:** emphasize aligning with the real distribution at the data generation stage, e.g., through high-fidelity rendering of lighting and materials or system identification of assets and dynamics [105, 88]. Although algorithmic post-processing can partially reduce the gap between synthetic and real data, its effectiveness is often difficult to quantify and evaluate in advance, and it typically incurs additional computational and manual overhead. In contrast, directly optimizing parameters at the simulator level to align the distribution of synthetic data with that of real data is often more practical and valuable [1].

In robotics, DA exhibits specific characteristics. Synthetic data is widely used for perception tasks such as object detection and segmentation [112], depth estimation [133], and pose estimation [20], where DA enables effective transfer with only a small amount of real-world data. For control tasks, adaptation can either be applied to the perception module alone or to the entire perception-control pipeline in an end-to-end manner. For example, James et al. [42] demonstrated data-efficient transfer for robotic grasping by combining simulation and real-world data.

Moreover, modality choice plays a crucial role in sim-to-real transfer. RGB images often exhibit large domain gaps in textures, lighting, and material properties, and manual adjustment of these factors is both time-consuming and costly. In contrast, depth-based modalities provide more stable geometric cues and are less sensitive to appearance variations. As illustrated in Fig. 2.3, simulation can generate not only RGB images but also structured light projection patterns and corresponding depth maps. The figure shows two representative setups: Intel RealSense (top row) and Photoneo (bottom row). By explicitly simulating the sensing pipeline of each device, including structured light projection and depth reconstruction, these modality-specific simulations significantly reduce the model mismatch between synthetic and real data. Rather than adapting from generic RGB or depth representations, the learning process operates on sensor-aligned modalities whose statistical and geometric properties more closely reflect real measurements. This alignment is particularly beneficial for Sim2Real transfer, as it shifts the primary adaptation challenge from low-level appearance discrepancies to higher-level task-relevant variations. For each sensor, the simulated outputs include RGB renderings, the projected pattern that the sensor would cast onto the scene, and the reconstructed depth map. These simulated modalities closely mirror the sensing pipeline of the actual devices, thereby providing a realistic proxy for training. Reproducing sensor-specific depth characteristics in simulation helps mitigate the domain gap. Depth sensing has

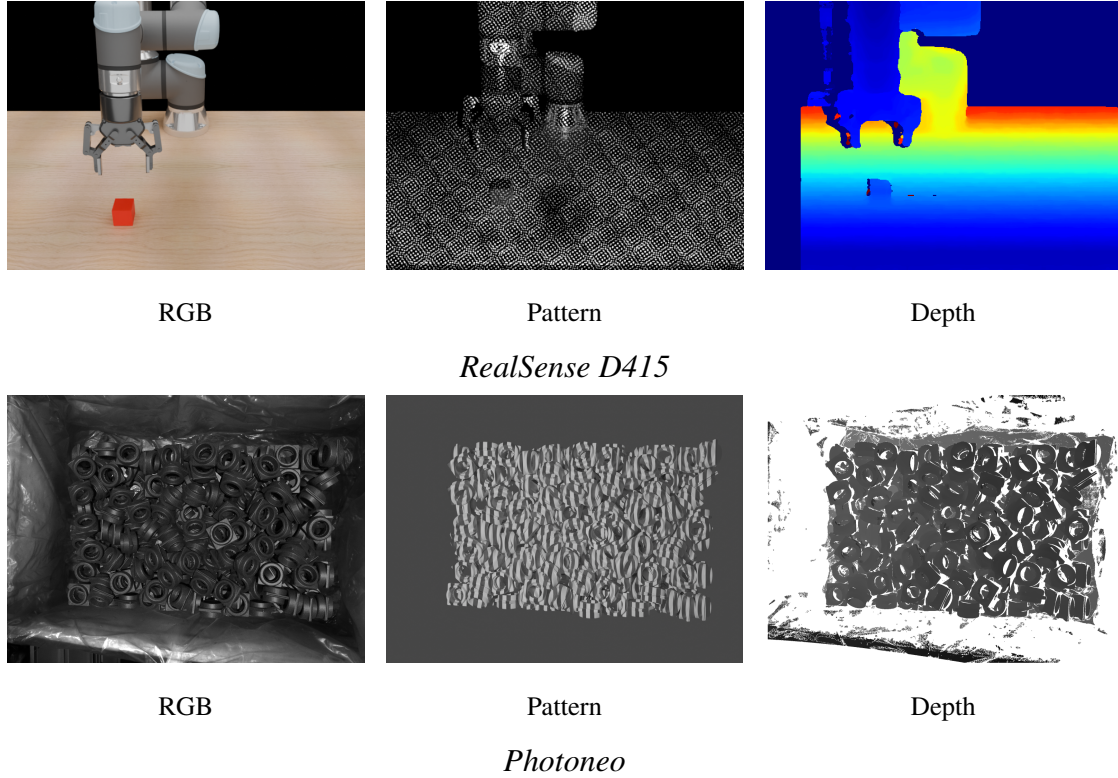


Figure 2.3: Simulated modalities and domain adaptation across depth sensors.

shown particular advantages in challenging scenarios involving transparent or reflective objects, where RGB-based adaptation becomes especially difficult [141, 4].

Despite the broad progress of DA, several challenges remain. Common failure modes include negative transfer due to misalignment, class distribution shift between domains, semantic drift introduced by pixel-level translation, pseudo-label noise accumulation in self-training, and the high cost of adjusting lighting and material properties for RGB-domain adaptation. These issues limit the applicability of DA in complex robotic environments. Overall, This observation highlights a broader limitation of existing Sim2Real techniques and motivates sensor-aware, physically grounded solutions for robust depth perception in challenging real-world scenarios. DA provides an essential strategy for sim-to-real transfer, complementing system identification, domain randomization, and physically based rendering, and together forming the major approaches to bridging the reality gap.

2.2 Generalizable Perception for Transparent Objects

Transparent objects are materials that allow most incident light to pass through while only reflecting or scattering a small portion, commonly seen in daily life (e.g., glass windows, plastic bottles, laboratory glassware). In robotics applications, transparent items like building glass walls in autonomous driving, glass flasks in automated labs, or clear containers in service robots are frequently encountered. However, due to their unique

optical properties, perceiving such objects is a formidable challenge for robots. On one hand, transparent objects lack distinctive visual features like color or texture; their appearance largely shows the background behind them, so traditional feature detection or texture-based recognition struggle. On the other hand, transparent materials violate the Lambertian assumption that underlies many standard vision sensors: instead of diffusing light uniformly, transparent surfaces refract and transmit light. Consequently, depth sensors like RGB-D cameras or LiDAR often produce incorrect or missing depth readings on transparents: either measuring the background behind as if it were the object’s depth, or failing entirely due to specular highlights. In summary, two core perception problems arise for transparent objects: detecting/localizing them (“where is it?”) and recovering their shape or depth (“how big/far is it?”). In recent years, fueled by advances in AI, a multitude of new datasets and methods have emerged to tackle these issues, and publications have surged accordingly [45].

2.2.1 Sensors and Simulation Platforms

Progress in transparent object perception has been closely tied to specialized hardware sensors and data generation techniques. This section reviews both physical sensing modalities and simulation platforms that have been leveraged to perceive transparent objects, highlighting their respective strengths and limitations.

Physical Sensors

A variety of physical sensor modalities have been explored for transparent object perception, each addressing different aspects of the challenges posed by transparency, refraction, and specular reflection.

RGB-D Cameras. RGB-D cameras such as Microsoft Kinect and Intel RealSense provide aligned color images and per-pixel depth measurements and are widely used in general 3D perception. However, their active infrared-based depth sensing often fails on transparent objects due to violations of the Lambertian surface assumption. For instance, when an Intel RealSense D415 observes a glass object, projected infrared patterns may pass through the glass and reflect off background surfaces, causing the sensor to report background depth instead of the glass surface. In other cases, strong specular reflections can confuse the stereo matching algorithms inside the sensor, leading to missing or erroneous depth measurements.

Stereo Cameras. Stereo vision systems infer depth by matching features between two cameras with a known baseline, as exemplified by platforms such as the Stereolabs ZED. In scenes with sufficient background texture, stereo methods can sometimes recover the outline of transparent objects through refractive distortions of the background. Nevertheless, because transparent regions lack consistent appearance for correspondence, standard stereo matching typically fails within the object interior. Several works have proposed specialized matching strategies or regularization techniques to better handle transparent and reflective surfaces [119].

Light Field Cameras. Light field cameras capture not only light intensity but also the directional distribution of incoming rays. This additional angular information enables post-capture refocusing and ray rebinning, which can expose refractive effects introduced by transparent objects. Early studies demonstrated that light field data could be used to detect refractive distortions of a known background pattern, thereby localizing clear glass objects [143]. Despite their advantages, light field cameras suffer from lower spatial resolution and heavy computational requirements. Nonetheless, light field cues have been successfully applied to transparent object recognition [69], segmentation [127], and pose estimation.

Polarization Cameras. Polarization cameras are equipped with pixel-level polarizing filters that measure the polarization state of light. Because transparent dielectrics reflect and refract light in a polarization-dependent manner, polarimetric images can reveal material boundaries and suppress glare [48]. By deriving Degree of Linear Polarization (DoLP) and Angle of Polarization (AoLP) representations, the contrast of glass objects against background regions can be significantly enhanced [48]. Empirical studies have reported notable accuracy improvements when incorporating polarization cues for glass segmentation and detection [48, 75]. The primary drawback is that polarization cameras remain relatively expensive and typically offer lower spatial resolution compared to standard RGB sensors.

RGB-Thermal Cameras. RGB-thermal sensors combine a visible-light camera with a long-wave infrared (LWIR) thermal sensor, producing spatially aligned outputs. Many common transparent materials, including glass, are opaque in the LWIR spectrum. As a result, objects that are visually transparent often appear as solid regions in thermal images. This property has been exploited to improve glass segmentation performance [39]. However, thermal cameras generally provide lower resolution imagery and incur higher hardware costs.

Tactile Sensors. Tactile sensing modalities, such as high-resolution touch sensors (e.g., GelSight) or force-torque sensors, provide complementary information when visual perception is unreliable. In manipulation tasks involving transparent objects, tactile feedback can verify object presence, infer local surface geometry, and assess grasp stability upon contact. For example, tactile signals can help confirm the pose of a transparent cup during robotic grasping [61]. Since tactile sensing is inherently local and contact-based, it is typically used alongside vision for planning and execution rather than as a standalone perception modality.

Overall, each physical sensing modality exhibits distinct trade-offs. Specialized sensors such as polarization or thermal cameras offer rich cues for transparency at higher cost, whereas conventional RGB or stereo systems are more affordable but require algorithmic compensation to address their inherent limitations. Practical systems often combine multiple modalities or select sensors based on task requirements and environmental constraints.

Simulation Platforms

When real-world data of transparent objects are scarce, expensive to acquire, or difficult to annotate, simulation platforms provide an effective alternative for generating synthetic training data. Several tools support rendering transparent materials with varying degrees of realism and efficiency.

Blender. Blender is an open-source 3D modeling and rendering suite that includes physically based renderers such as Cycles and LuxCoreRender. It allows precise control over optical material properties and can simulate complex effects including refraction and caustics. Prior work has used Blender extensively for data generation; for example, ClearGrasp [94] rendered approximately 50k images for depth completion, while TransProteus [29] generated large-scale synthetic datasets for transparent object segmentation. The main limitation of Blender-based pipelines is the high computational cost associated with path-traced rendering. Such path-tracing-based renderers explicitly simulate multiple light bounces and wavelength-independent refraction, making them particularly suitable for modeling complex transparency effects, albeit at high computational cost.

Unreal Engine. Unreal Engine (UE) is a commercial real-time rendering platform widely adopted for interactive simulation. Recent versions of UE support advanced lighting models and hardware-accelerated ray tracing, enabling visually realistic scenes at interactive frame rates. Transparent object simulation has been explored in UE and similar engines such as Unity; for instance, the SuperCaustics framework [78] employs real-time ray tracing in Unreal Engine 4 to simulate caustics, dispersion, and refraction effects. Although real-time engines generally cannot match the ultimate realism of offline renderers, they offer a favorable balance between visual fidelity and rendering efficiency. In contrast, real-time engines rely on approximations or hybrid ray tracing pipelines, which trade strict physical accuracy for efficiency and scalability.

NVIDIA Isaac Sim. NVIDIA Isaac Sim is a recent simulation and collaboration platform that emphasizes photorealistic real-time graphics and physically accurate simulation for robotic applications. It integrates the PhysX physics engine and supports multiple rendering modes, including RTX real-time, RTX path tracing, and Iray. Isaac Sim facilitates the generation of datasets with realistic transparency effects while also providing ground-truth geometry and pose information. For example, the Isaac Sim Object Dataset [145] was generated using this platform and includes diverse scenes with physically simulated transparent objects. However, the high hardware requirements of Isaac Sim, particularly for GPU memory and real-time ray tracing performance, impose practical limitations on its deployment, especially for large-scale data generation or resource-constrained research environments.

In summary, simulation platforms offer flexible means for data generation in transparent object perception. High-fidelity renderers are well suited for offline dataset creation where realism is paramount, whereas real-time engines enable interactive training

and large-scale data generation. While synthetic data have become indispensable in this field, they also introduce the challenge of domain adaptation, namely ensuring that models trained in simulation transfer effectively to real-world scenarios.

2.2.2 Transparent Object Segmentation

Transparent object segmentation aims to distinguish transparent material regions in images, typically producing a pixel-wise mask. It is a fundamental perception task. For example, an autonomous robot must detect glass doors or windows to avoid collisions, and a manipulator may first segment a transparent target object to facilitate subsequent localization and grasping. In recent years, several datasets and algorithms have been introduced for this problem.

Datasets. A variety of public datasets have driven progress in transparent object segmentation. Early on, TransCut [127] was captured with a light field camera and contains a small number of images of transparent containers against different backgrounds, providing one of the first multi-view transparency datasets. Tom-Net [14] combined 178k synthetic images rendered with POV-Ray using random refractive indices and 876 real photos, leveraging diverse backgrounds and materials to improve model generalization to transparent objects. The Glass Detection Dataset [76] followed, offering 3,916 pairs of glass images and binary masks across indoor and outdoor scenes, which became a widely used benchmark for glass detection and segmentation.

A novel direction using polarization cameras was introduced with the Polarized dataset [75], which contains 1,600 images captured by a polarimetric camera and incorporates polarization information for enhanced glass segmentation. Scaling up further, Trans10k [123] provided over 10,000 images with manual annotations, while its refined version Trans10k-v2 [124] offered finer labels and more detailed annotations, laying a strong data foundation for general transparent object segmentation. The Glass Surface Dataset [66] targeted more complex-shaped glass surfaces and offered more challenging examples.

In terms of synthetic data, SuperCaustics [78] was generated in game engines such as Unreal or Unity and includes 9,000 images with physically accurate caustics effects to test models' ability to handle complex light patterns. One of the most challenging datasets to date is TransProteus [29], which consists of 50k synthetic images generated in Blender and 104 real images, encompassing 13k distinct objects and 500 backgrounds, significantly increasing scene complexity and object variety. Other notable datasets include TransTouch [7], which improves transparent-object depth perception by fine-tuning stereo networks with sparse depth supervision automatically collected through tactile probing, and provides a real-world dataset containing both diffuse and transparent objects for evaluation. RGBP-Glass [75], which offers a large-scale polarization-based dataset with 4,511 pairs of RGB intensity and polarization images, and RGB-Thermal [39], which contains 5,551 aligned RGB and thermal image pairs upsampled to a resolution of 640×480 .

Altogether, these datasets expand the coverage of transparent object types, envi-

ronments, and sensing modalities, including daily objects, building glass, laboratory benchtops, outdoor scenes, and RGB, polarization, and thermal imaging. While real-world datasets are costly to annotate and limited in scale, synthetic data plays a crucial role in advancing this field, albeit introducing a simulation-to-reality domain gap.

Methods. Approaches to transparent object segmentation have evolved from traditional hand-crafted feature techniques to modern data-driven deep learning methods. Early hand-crafted approaches exploited observable cues of transparent objects, such as object boundaries, specular highlights, or distortions. Mei et al. [76] leveraged learned contextual and boundary cues to locate glass edges in real scenes, while [82] located transparent objects by the shape of specular highlights. However, these methods typically assumed simple or known backgrounds and did not generalize well to complex scenes. Some improved approaches incorporated multi-modal cues, for example, Guo-Hua et al. [33] retrieve transparent candidates from depth deficits and then use RGB/IR cues for recognition and localization, or using light field images to compute refraction linearity features for fine-grained segmentation [127].

With the rise of deep learning, learning-based methods became dominant and can be broadly categorized into single-modal and multi-modal approaches. Among RGB-only methods, Tom-Net [14] formulates transparent object matting as refractive-flow estimation and predicts a transparent matte consisting of an object mask, an attenuation map, and a refractive flow field from a single image, while object detection frameworks such as Mask R-CNN [36] were adapted for instance-level glass segmentation [21]. To enhance sensitivity to transparent regions, GDNNet [76] introduced spatial attention mechanisms. Transformer-based models were later adopted to capture global context more effectively, including Trans2Seg [124] and the dual-headed Trans4Trans [139] for wearable navigation scenarios. Boundary-aware approaches such as TransLab [123] and EBLNet [35] incorporated explicit edge supervision to improve boundary precision.

Multi-modal segmentation methods further improved performance by incorporating sensor data beyond RGB. Combining infrared images with RGB leverages the fact that many transparent objects are opaque in IR. Yan et al. [128] propose NRGlassNet, which takes paired RGB and near-infrared images and strengthens glass surface detection via cross-modal contrast and difference modeling modules. Polarization imaging has also proven effective: Kalra et al. [48] proposed Polarized Mask R-CNN [36], which fuses polarization cues into Mask R-CNN [36] via a polarization attention module. Similarly, PGSNet [75] dynamically fused polarization and trichromatic color information to balance fine detail preservation and global context. Thermal-RGB fusion represents another direction. Huo et al. [39] developed an RGB-thermal network that uses attention modules in a Transformer encoder to combine thermal and visible features, outperforming RGB-only models on a custom RGB-thermal dataset.

Challenges. Despite these advances, transparent object segmentation still faces several challenges. Constructing large, high-quality, pixel-annotated datasets remains labor-intensive, and polarization or thermal datasets are relatively small, limiting further gains in deep models and generalization. Synthetic data augmentation using tools such as

Blender [94, 29] helps alleviate this issue but introduces a sim-to-real domain gap. In addition, real-world deployment demands robustness under varying illumination, dynamic backgrounds, and real-time constraints, while many state-of-the-art models remain computationally heavy. Finally, most existing methods adapt architectures designed for opaque objects, whereas the unique physical properties of transparency may call for task-specific representations, optical modeling, or generative approaches such as GANs or diffusion models.

2.2.3 Transparent Object Depth Completion and Reconstruction

Standard RGB-D sensors struggle to capture the true geometry of transparent objects, so researchers have proposed various methods to complete the depth of transparent regions or directly reconstruct their 3D shape. Solving this not only gives robots a more complete 3D understanding of the environment, but also enables algorithms originally designed for opaque objects (e.g., grasp planners) to be applied to transparent objects. Below, we first introduce recent datasets for this task, then discuss different methods, and finally summarize challenges.

Datasets. Transparent object reconstruction requires ground-truth depth or shape data for training and evaluation, which is hard to obtain. It often involves either special procedures to measure true depth on transparent objects or creating highly realistic simulations. Several important datasets released after 2020 include: ClearGrasp [94]: the first large-scale transparent object depth completion dataset, containing 50k synthetic images from Blender and 286 real photographs. In the real subset, transparent objects were sprayed with a diffusing coating so that an RGB-D camera could capture approximate true depth (once the object became opaque when sprayed). OOD (Omniverse Object Dataset) [145]: a synthetic dataset rendered with NVIDIA Omniverse at 60k+ images. Ground truth is produced via a two-pass rendering strategy: first render geometry-related labels (e.g., depth) with all objects set opaque, then render the RGB image with selected objects set to glass using path tracing, aligning transparent appearance with the provided supervision. TODD (Transparent Object Depth Dataset) [126]: a real dataset captured with an Intel RealSense camera and an AprilTag rig for automatic ground truth, comprising 14,659 RGB-D images of 6 transparent container objects. By placing AprilTag markers inside and capturing from multiple angles, fairly accurate depth maps were obtained for evaluation. TRANS-AFF [46]: contains 1,346 pairs of RGB and depth images for 8 graspable transparent containers; besides depth, it labels which areas of the container afford stable grasps. This dataset is designed for joint study of depth completion and physical affordances. Dex-NeRF data [40]: includes 8 synthetic scenes and 8 real scenes, each with multi-view RGB images, intended for using NeRF to reconstruct transparent objects and plan grasps (useful for multi-view reconstruction benchmarks). Evo-NeRF data [52]: 8,667 rendered sequences of transparent object scenes using Blender, to test dynamic reconstruction models in sequential grasping scenarios. TransCG [31]: a large-scale real-world dataset with 57,715 RGB-D images of transparent objects in clutter; it utilized precise optical tracking and multi-view fusion to obtain accurate depth

ground truth for transparent items. DREDS and STD [26]: DREDS is a 130k-image synthetic dataset that simulates an active-stereo depth camera (RealSense-D415-like) and produces depths with realistic noise patterns; STD is a real dataset capturing 30 cluttered scenes with 50 objects across specular/transparent/diffuse materials, used for evaluating depth restoration. Generally, synthetic datasets are larger and easier to annotate automatically (DREDS, OOD), but their realism and diversity are bounded by simulation capabilities. Real datasets like TODD or TransCG achieve a certain scale through clever labeling strategies and are truer to life, but often cover fewer object types or simpler scenes. The ideal dataset would span many object categories, complex scenes, and precise 3D ground truth – currently a tall order, so many methods still rely on synthetic data to supplement training.

Methods. Broadly, methods for transparent object 3D reconstruction/depth prediction can be categorized by the number of viewpoints used, into multi-view methods versus single-view methods.

These assume the object can be observed from multiple angles and leverage multi-frame information to recover 3D structure. In classic computer vision, one approach is the visual hull [118]: silhouettes of the object are extracted from each viewpoint and intersected in 3D to approximate the object’s volume. This works for convex shapes but tends to over-fill concavities or interior voids, and with transparent objects it is challenging to obtain clean silhouettes at all, since their contours may be faint or camouflaged by the background. Some refinements instead leverage additional cues beyond silhouettes. For example, RFTrans [99] uses refractive flow as an intermediate representation to estimate surface normals from a single RGB-D image, and combines the predictions with a global optimization module to refine depth and geometry for manipulation, or exploiting object symmetries [84]. However, these often require strong priors, such as a known background or assumed shape types. Another direction is learning-based two-view (stereo) reconstruction using active-stereo sensors. For instance, ASGrasp [96] leverages RGB and IR images from an RGB-D active stereo camera and a learning-based stereo pipeline to reconstruct geometry of transparent objects for grasping.

Major recent advances come from Neural Radiance Fields (NeRF) and related techniques. NeRF represents a scene as a continuous function that, given a spatial coordinate and viewing direction, outputs color and density, enabling photorealistic novel view synthesis. Adapting NeRF to transparency requires handling refraction and reflection, but progress has been made. Dex-NeRF [40] applies NeRF to a transparent object in a robotic context: by capturing multiple RGB images around a transparent object, it trained a NeRF to capture the mixed appearance of object and background, then derived the object’s geometry from the NeRF for grasp planning. Dex-NeRF showed NeRF can implicitly learn a transparent object’s shape, but training took several hours, so it was limited to offline use. Subsequently, Evo-NeRF [52] introduced faster training strategies (leveraging Instant-NGP [79] and other acceleration) making NeRF construction quicker for transparent scenes, and explored sequential updating where a robot continuously refines the NeRF after successive grasps. The advantage of NeRF-based methods is their ability to, in principle, model very complex optical phenomena (including mul-

multiple refractions), but drawbacks include high computational cost, non-real-time inference, and the requirement of multiple views (often with a controlled setup such as static background).

Since capturing multiple views is not always practical (a robot might only see an object from one perspective, or only one image of an object might be available), many works try to infer transparent object depth or shape from just a single frame.

Early approaches took an analysis-and-optimization route: first extract intermediate information from the image, then infer depth via optimization. For example, ClearGrasp [94] uses a two-step pipeline: a CNN first predicts intermediate outputs from the RGB input – namely a surface normal map, a segmentation mask of transparent regions, and edge contours. In the second step, these predictions, along with the raw sensor depth (with holes/noise), feed into a global optimization module (utilizing a cost-volume formulation) to fill in the missing depth for transparent regions. ClearGrasp [94] worked well on simple tabletop scenes, but it relies heavily on correct normal and mask predictions; its performance degrades if lighting causes the normal prediction to fail or if segmentation misses parts of the object. TRANS-AFF/A4T [46] proposes an affordance-guided, multi-step depth reconstruction pipeline for transparent containers, where depth is progressively reconstructed region-by-region using predictions (e.g., normals/boundaries) together with global optimization and RANSAC-based plane fitting.

With more powerful learning techniques, fully end-to-end single-view methods have emerged. Broadly, these approaches can be categorized into methods that primarily leverage point clouds or depth as input and those that rely mainly on RGB images. For the former category, Seeing Glass [126] proposed a joint point-cloud and depth completion framework for transparent objects. The method starts from the incomplete point cloud produced by depth sensors and learns to recover missing geometry caused by transparency using a 3D convolutional network. The recovered geometric information is then projected back to the depth domain and combined with RGB features to produce a dense depth map. By explicitly incorporating three-dimensional structure into the learning process, this hybrid 3D–2D formulation preserves object geometry more effectively than purely image-based approaches. The method in OOD [145] introduced local implicit scene representations for transparent depth completion. Instead of predicting depth explicitly on a grid, it learns a local implicit function that models depth in transparent regions and iteratively refines predictions through self-correction. Trained primarily on synthetic data, this approach captures common depth patterns of transparent objects and demonstrates a degree of generalization to unseen instances. Other approaches avoid explicit point cloud representations and directly process RGB-D inputs. For example, TranspareNet [126] employs dual encoders for RGB images and initial depth maps, fuses their features, and decodes a completed depth map. This joint processing of color and depth has been shown to be effective for both transparent and specular objects. Transformer-based models have also been applied to this problem. As noted earlier, TODE-Trans [16] uses a Transformer encoder to capture global context and a decoder to recover fine-grained details, together with a multi-scale feature fusion module, achieving substantially lower error on transparent depth estimation compared to prior CNN-based methods. Similarly, SwinDRNet [26] employs two Swin Transformer encoders for RGB and depth, respectively, and fuses them via cross-attention. Notably,

it additionally predicts a confidence map indicating the reliability of its depth estimates, which is used to blend the network output with the original sensor depth where appropriate. This design proves particularly effective in synthetic-to-real transfer and achieves state-of-the-art performance on transparent depth completion benchmarks. Finally, some methods prioritize computational efficiency. FDCT [62] adopts a lightweight network that takes RGB images and raw depth with missing values as input and outputs completed depth maps at real-time speed. Although its accuracy is lower than that of more complex models, its efficiency makes it well suited for robotics applications such as grasping, where low latency is critical.

Summary and challenges. In summary, multi-view methods tend to achieve higher reconstruction accuracy and completeness, at the cost of increased data requirements and computational overhead. Single-view methods offer greater deployment flexibility, but their performance degrades on objects with complex geometry or appearances that lie outside the training distribution.

Several open challenges remain in transparent object depth reconstruction. A fundamental difficulty lies in obtaining reliable ground truth depth. Unlike segmentation tasks, where object boundaries can be annotated manually, providing accurate pixel-wise depth for transparent objects is nearly impossible. Existing datasets rely on different acquisition strategies, each with significant limitations. Surface spraying alters the appearance of the object, fiducial markers only apply to specific shapes, and structured light scanners are difficult to deploy in cluttered environments. As a result, many methods rely heavily on synthetic data, and their performance in real-world scenes remains uncertain. Unsupervised or self-supervised learning offers a promising alternative by allowing models to learn from rendering consistency or physical constraints without explicit depth supervision. However, for transparent objects, such approaches are still far from practical.

Another major challenge concerns generalization and computational efficiency. NeRF-based methods struggle severely with transparent objects because standard volumetric rendering formulations are ill-suited to refractive and transmissive materials. In addition, these methods are computationally expensive and impractical for real-time use. While ongoing research on accelerating NeRF may reduce runtime, it does not fundamentally resolve the modeling difficulties posed by transparency. In contrast, single-view CNN and Transformer-based approaches can achieve good performance on data similar to their training distribution, but often fail when confronted with unseen materials, refractive properties, or object geometries. Improving robustness across diverse transparent materials and developing models that generalize reliably across scenarios remain open problems.

Multi-modal sensing presents both opportunities and challenges. Modalities such as polarization, thermal imaging, or tactile feedback can provide additional constraints for transparent object perception. However, designing architectures that fuse these signals effectively, without introducing ambiguity or conflicting cues, remains difficult. Finally, handling complex scenes containing multiple transparent objects is extremely challenging. Most existing methods assume a single dominant transparent object or relatively

simple configurations. When multiple transparent objects overlap or occlude one another, current techniques struggle to disentangle and reconstruct their individual depth. Accurately reconstructing a scene with many stacked or interleaved transparent objects remains beyond the capabilities of current systems.

Addressing these challenges will require a deeper understanding of complex scenes, potentially through the integration of scene-level priors, semantic context, or hierarchical reconstruction strategies, in order to move closer to human-level perception of transparent objects in unconstrained environments.

2.3 Physically Realistic Depth Sensor Simulation

2.3.1 Structured Light Depth Sensor Simulation

Early studies on structured light depth sensors primarily focused on characterizing and modeling real sensor behavior, rather than generating learning-oriented synthetic data. Geometric and statistical analyses of Kinect-class structured light devices quantified depth accuracy, bias, and noise characteristics as functions of distance, incidence angle, and image location [77, 81, 22]. These works provided empirical error models that later simulation pipelines rely on for parameter calibration and realism, but they did not explicitly aim at data generation for deep learning.

Moving beyond pure error modeling, Landau et al. [58] proposed a physics-inspired simulator that predicts Kinect-style infrared and depth images by modeling the interaction between the infrared emitter, the projected pattern, and the receiver, enabling the synthesis of depth images that resemble real Kinect outputs [58]. While this work introduced physically motivated image formation, its primary goal remained sensor behavior analysis rather than large-scale dataset generation for learning.

Subsequent work shifted toward synthetic data generation for deep learning, with increasing emphasis on physical realism. DepthSynth [87] generates realistic depth images from CAD models by modeling sensor noise, surface geometry, and material reflectance, and explicitly evaluates the effect of such synthetic depth on training deep recognition models [87]. Koch et al. [55] further advanced structured light simulation fidelity by co-designing scanning hardware and a calibrated simulation pipeline, achieving near pixel-level agreement between real and simulated images on controlled scenes and providing benchmark datasets for 3D reconstruction evaluation [55].

A major conceptual advance was introduced by Planche and Singh [86], who proposed a physics-based differentiable depth sensor simulation that explicitly models infrared projection, occlusion-aware image formation, and structured light correspondence, while enabling gradient-based optimization of simulation parameters to better match real sensor observations. Building upon this line of work, Bai et al. [4] presented a fully physics-based structured light simulation pipeline capable of generating large-scale RGB-D datasets for industrial robotic perception, and showed that such synthetic data can substantially reduce the Sim2Real gap in detection, segmentation, and grasping tasks, even when real depth measurements are not used for training supervision.

2.3.2 Active Stereo Depth Sensor Simulation

Active stereo depth sensors, exemplified by the Intel RealSense family, fundamentally differ from structured light systems in that depth is recovered through stereo correspondence on infrared images augmented by projected texture, rather than explicit pattern decoding. As a result, depth errors in active stereo are strongly dependent on material properties, infrared image quality, and stereo matching failure, making heuristic depth noise models insufficient.

Recent work has therefore focused on physically grounded simulation of the full active stereo pipeline. Dai et al. [26] introduced a domain-randomization-enhanced simulation framework that models active stereo depth sensing using physically based rendering, enabling the generation of large-scale RGB-D datasets with realistic missing and corrupted depth on specular and transparent objects. The synthetic data were used to train depth restoration networks that generalize effectively to real RealSense data and improve downstream robotic perception tasks.

Zhang et al. [141] further proposed a physics-grounded active stereo sensor simulation that explicitly models material-dependent infrared reflectance, ray-traced infrared image formation, sensor noise, and stereo correspondence, with the goal of closing the optical sensing domain gap for RealSense-style cameras. Their results demonstrate that simulating intermediate infrared observations and the stereo reconstruction process is critical for learning-based methods to transfer from simulation to real active stereo sensors.

Complementary to these research-focused simulators, the SAPIEN simulator integrates a real-time active stereo depth sensor model (“Realistic Depth”) designed as a practical drop-in replacement for ground-truth depth in robotics simulation, further highlighting the importance of sensor-realistic depth for Sim2Real learning with RealSense-class cameras [120].

2.3.3 Time-of-Flight Depth Sensor Simulation

Physically realistic simulation of time-of-flight (ToF) depth sensors has a longer history, originally motivated by sensor design and performance analysis rather than data generation for deep learning. Early work by Keller and Kolb [51] proposed a real-time physics-based simulation of PMD-style ToF sensors, modeling effects such as flying pixels, motion blur, and systematic depth deviations. Lambers et al. [57] extended this line of work with a physically parameterized AMCW ToF simulation framework aimed at evaluating chip layout variants and sensor design choices.

Later studies emphasized the role of multipath interference (MPI) and material-dependent infrared reflectance in ToF sensing. Bulczak et al. [11] introduced a quantified, interactive AMCW ToF simulation that incorporates multipath effects and models sensor behavior down to pixel-level charge accumulation, enabling controlled analysis of ToF artifacts. Yan et al. [129] further improved physical fidelity by adopting path-tracking-based simulation with measured infrared BRDFs, reporting quantitative agreement with real experiments in terms of depth error metrics.

More recently, ToF simulation has been increasingly connected to deep learning.

Instead of fully modeling the physical pipeline, Müller et al. [80] proposed a learning-based artifact generation approach that maps high-quality laser scans to realistic ToF depth images with characteristic artifacts, providing a scalable alternative for generating training data when full physical simulation is computationally expensive.

Chapter 3

Close the Sim2Real Gap via Physically-based Camera Simulation

3.1 Research Motivation and Challenges for Structured Light Depth Sensing

With the rapid development of deep learning, the capabilities of robotic perception have improved significantly in areas such as object recognition, pose estimation, and semantic understanding. However, the application of deep learning models in industrial robotic systems remains constrained due to the inherent complexities of the industrial environment and the challenges associated with data acquisition. This chapter introduces the motivation and background of this research, highlighting the necessity of physically-based camera simulation in bridging the gap between synthetic and real data for robot vision tasks. We conclude the chapter by outlining the key contributions and novel methods proposed in this thesis.

3.1.1 Industrial Challenges in Robotic Perception

In industrial contexts such as smart manufacturing, logistics, and automated warehousing, robots are expected to operate in highly dynamic and cluttered environments. These robots must identify, locate, and manipulate a wide variety of objects that differ in shape, size, texture, and material. Visual perception, particularly object detection and 6D pose estimation, plays a foundational role in ensuring that robots can interact with their surroundings in a safe and efficient manner. However, collecting high-quality training data for these tasks remains a significant bottleneck. Factories are often subject to strict safety regulations and confidentiality agreements, making large-scale data collection and annotation not only logistically difficult but also legally sensitive. Furthermore, manual annotation of industrial datasets, especially those involving depth data and pose information, is labor-intensive and time-consuming.

These challenges are further compounded by the complexity of the environment itself. Industrial tasks often involve heavy occlusion, variable lighting, and high object density, which introduces substantial ambiguity in visual inputs. Additionally, many in-

dustrial parts are relatively heavy and must be physically moved and repositioned during data collection. Repeated handling of such components can cause physical strain to human operators, particularly when performed over extended periods or in awkward postures. This introduces health and safety concerns, further increasing the cost and difficulty of collecting real-world datasets in industrial environments. In such conditions, relying solely on real data for training deep learning models is not only inefficient but also impractical in many scenarios. These limitations have prompted researchers to turn to synthetic data and simulation-based training pipelines. Nevertheless, traditional simulation approaches still suffer from a notable Sim2Real gap, especially in depth perception and fine surface details—both of which are critical for robotic manipulation.

3.1.2 The Importance of Physically-based Camera Simulation

Bridging the Sim2Real gap in robotic vision tasks requires not only the simulation of visually realistic RGB images but also the accurate reproduction of depth sensing processes. However, most existing simulation pipelines concentrate primarily on RGB image generation and ignore the depth modality or treat it as a secondary product derived from geometric information. As a result, these simulations often fail to capture the noise characteristics, occlusion behavior, and structural distortions present in real-world depth sensors. This limitation is especially critical for multimodal perception tasks, such as RGB-D object detection, instance segmentation, and 6D pose estimation, where the accuracy and consistency of depth data are as important as the RGB modality. The absence of physically-informed depth simulation severely limits the applicability and generalization of learned models when transferred from synthetic to real-world domains.

Structured light cameras are widely used in industrial environments due to their high precision, robustness to texture variation, and relatively low cost. Among them, gray-code based systems are particularly prevalent. These sensors acquire depth by projecting a sequence of structured patterns onto the scene and decoding the reflected light. Crucially, the noise characteristics and reconstruction artifacts in the captured depth data are inherently tied to this imaging principle and cannot be realistically replicated through post-processing alone. Therefore, a meaningful simulation of such sensors must explicitly model the projection, capture, and decoding processes. In this thesis, we propose a physically-based simulation framework for gray-code structured light cameras, implemented in Blender’s Cycles engine. This system reproduces depth sensor behavior with high fidelity, accounting for shadow-induced data loss and sensor-specific noise. The simulation outputs aligned RGB-D images along with ground truth labels for tasks such as object detection, segmentation, and pose estimation, offering a robust foundation for multimodal Sim2Real learning.

3.1.3 Contributions and Innovations

This work addresses the practical challenges of applying deep learning to industrial robotic perception by introducing a robust and physically grounded data generation framework. A central contribution lies in the development of a physically-based simulation system for gray-code structured light cameras, which enables the generation of

photorealistic RGB images and physically realistic depth maps. By accurately modeling the imaging process—including projection, reflection, and decoding—the system produces RGB-D data that captures sensor-specific noise and optical artifacts, providing a high-fidelity foundation for training vision models with better generalization to real-world conditions.

In addition, this chapter presents a complete RGB-D dataset designed for industrial robotic grasping scenarios. The dataset includes a large-scale synthetic training set and real-world evaluation set, making it possible to quantitatively benchmark Sim2Real performance across multiple perception tasks, such as object detection and instance segmentation. Finally, the method is validated through its application in a real robotic platform, demonstrating that simulation-driven training using physically accurate RGB-D data can significantly improve the effectiveness and robustness of robotic visual manipulation in complex industrial environments.

3.2 Challenges and Design Considerations in Synthetic Depth Simulation

As synthetic data becomes a prominent tool for training deep learning models in robotic vision, the accuracy and realism of depth image simulation are increasingly important. This section reviews the development of synthetic dataset generation methods, analyzes the Sim2Real gap in depth-based perception, and classifies sensor simulation strategies. Finally, it highlights the role of physically-based rendering in producing high-fidelity RGB-D datasets.

3.2.1 Synthetic Dataset Generation and Sim2Real Gap

The use of synthetic datasets to train robotic perception models and transfer them to real-world environments has gained widespread attention. Tools such as Omniverse [91], Unreal Engine [53, 89, 101], Blender [2, 37, 73, 41], and PyBullet offer diverse functionalities for rendering 3D scenes and producing annotated RGB-D images. These platforms differ in rendering quality, support for physical modeling, and simulation speed. Game engines, while offering high frame rates, often lack accurate ray tracing and light transport simulation, which limits realism in reflective and transparent surfaces.

A persistent challenge in this domain is the Sim2Real gap, which refers to the performance degradation observed when models trained on synthetic data are deployed in real-world environments. This issue is particularly pronounced in RGB-D perception, where inaccuracies in depth maps caused by unrealistic geometry, textures, or sensor noise can significantly impair downstream tasks such as grasping and segmentation. To mitigate this gap, domain randomization introduces variations in appearance, lighting, and material properties, while domain adaptation methods such as GANs aim to align the distributions of synthetic and real data. However, these strategies are largely developed for RGB imagery and often fail to capture the underlying physical characteristics of real depth sensors.

3.2.2 Taxonomy of Depth Sensor Simulation Methods

Depth sensors used in robotic and industrial contexts fall into four primary categories: passive stereo, active stereo, time-of-flight (ToF), and structured light. Each type has distinct principles of depth acquisition and unique noise characteristics. Simulation methods aiming to reproduce their output must therefore take these properties into account.

Low-fidelity simulations typically generate depth maps directly from 3D geometry using idealized rendering techniques. While efficient, these methods produce unrealistic results that fail to replicate sensor-specific behavior, such as shadow-based occlusion or pattern deformation. More advanced approaches simulate stereo disparity or ToF depth using virtual sensors, but often still miss critical details like depth noise and missing values caused by material reflectivity or geometry. For structured light sensors in particular, depth errors arise directly from projection and decoding processes, making it essential to explicitly simulate the entire imaging pipeline rather than relying on post-processing corrections.

3.2.3 Role of Physically-Based Rendering in Accuracy

Physically-based rendering (PBR) plays a critical role in improving the realism and accuracy of synthetic depth images. Unlike rasterization-based rendering, PBR models the actual light transport and material interaction processes, allowing for more accurate representation of surface reflectance, shadowing, and interreflections—all of which impact the quality of both RGB and depth outputs.

In the context of structured light simulation, PBR becomes especially important. Gray-code structured light sensors reconstruct depth based on tiny changes in the reflected projection pattern. These distortions are sensitive to local surface properties and occlusion effects. If the simulation does not model the exact optical projection and capture process, the resulting depth map will be too clean or geometrically idealized to be useful in real-world applications. By using PBR engines like Blender Cycles with ray tracing and denoising, it becomes possible to produce synthetic images that not only look realistic but also behave statistically similar to actual sensor outputs—thereby providing a reliable basis for Sim2Real learning.

3.3 Physically-Based Simulation Framework

To generate realistic synthetic RGB-D data suitable for industrial perception tasks, we build a physically-based structured light simulation framework as in Fig. 3.1. The proposed framework consists of two main components: scene simulation and real-world execution.

In the simulation stage, a physics engine is used to generate random object poses in a bin-picking scenario, and a rendering engine produces projected images. Gray code structured-light reconstruction is then applied to generate physically realistic depth images along with corresponding ground-truth annotations. These datasets are used for sim-only perception model training and subsequent Sim2Real transfer learning. In the

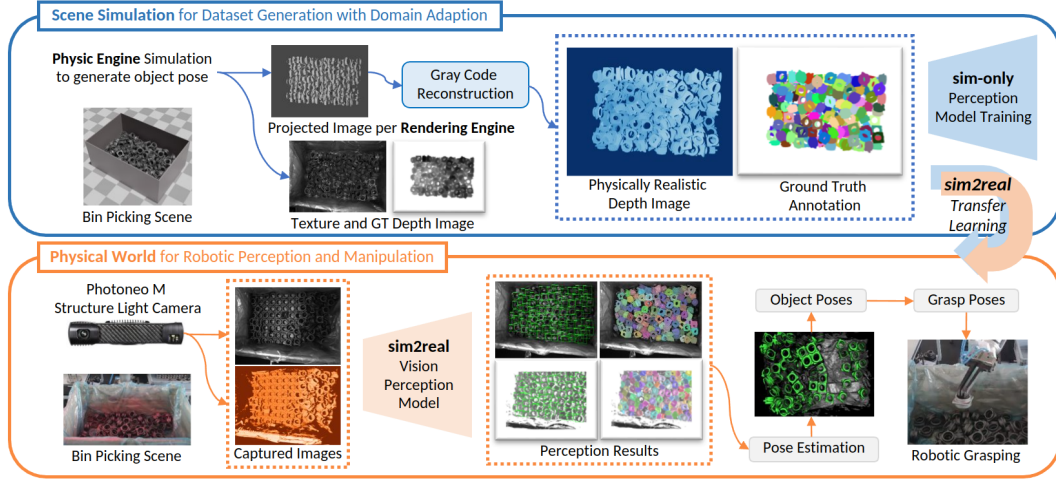


Figure 3.1: Sim2Real perception and grasping framework.

real-world stage, a structured-light camera captures point clouds and depth images of the bin-picking scene. The captured data are fed into the sim-to-real perception model for object pose estimation, which is then converted into grasp poses. Finally, the robot executes the grasping tasks, achieving seamless integration from simulation to reality.

This pipeline integrates geometry sampling, material modeling, structured light encoding and decoding, and physically-accurate rendering via ray tracing. Compared with traditional rasterization or domain randomization approaches, our framework better preserves the physical interaction between projected patterns and object surfaces, producing depth artifacts, occlusions, and shading effects closer to real-world observations.

The simulation consists of the following modules:

- **Scene Construction:** Objects are randomly sampled from a library and placed under gravity to form realistic cluttered layouts.
- **Material Assignment:** Physically-defined surface properties, including roughness, specular, and index of refraction, are assigned to each object based on real-world measurements.
- **Structured Light Projection:** Gray-coded patterns are projected onto the scene using a simulated projector with precise calibration parameters.
- **Ray-Traced Rendering:** A physically-based renderer (Blender Cycles with NVIDIA OptiX backend) is used to trace the light paths from projector to surface to camera, capturing accurate shadows, indirect lighting, and inter-reflections.
- **Depth Recovery:** A decoding algorithm reconstructs depth maps from the rendered Gray code sequence, incorporating realistic decoding failures and depth discontinuities.
- **Annotation Generation:** Ground-truth annotations such as segmentation masks, object bounding boxes, instance IDs, and depth maps are automatically extracted and aligned.

This integrated system enables the generation of synthetic data that more accurately reflect the optical phenomena observed in industrial 3D sensors, especially under complex lighting, clutter, and occlusion.

3.3.1 Rendering Pipeline Using Ray-Tracing

Accurate simulation of structured light imaging requires faithful modeling of light transport in complex 3D environments, especially under challenging conditions such as inter-reflections, sharp occlusions, transparent surfaces, and specular materials. In this context, we adopt a ray tracing–based rendering pipeline as the core of our synthetic data generation framework.

Rasterization, while computationally efficient, fundamentally lacks the capacity to trace light interactions beyond direct visibility. In typical rendering pipelines using rasterization (e.g., OpenGL/DirectX), lighting effects such as indirect illumination, light bouncing, or secondary occlusions are either approximated or entirely ignored. More critically for our case, rasterization cannot simulate how structured light patterns project, scatter, and reflect across complex surfaces.

In structured light systems, the geometric relationship between the projector, scene, and camera forms a critical optical triangle. Accurate simulation requires that light rays emitted by the projector be traced as they interact with the scene—potentially reflecting multiple times—before being captured by the camera. Raster methods cannot support this multi-bounce transport, nor can they model soft shadows, specular highlight distortion, or depth ambiguities caused by occlusions.

Ray tracing explicitly models the full path of light rays through the scene, enabling precise rendering of:

- Projection distortions caused by surface curvature and occlusion edges
- Soft shadows and partial visibility at object boundaries
- Inter-reflections on glossy and metallic materials
- Structured pattern corruption due to indirect lighting or reflective interference
- Accurate depth ambiguity zones, especially near thin structures or depth discontinuities

We employ Blender’s Cycles engine with the OptiX backend to ensure physically-correct rendering. The projector is modeled as an active light emitter with user-defined Gray code patterns, while the camera is a calibrated pinhole sensor capturing the reflected patterns. This configuration allows the synthetic data to match real-world noise distributions, including surface highlights, decoding ambiguity, and missing depth zones—features that are critical for training robust depth-based perception models.

3.3.2 Structured Light Pattern Projection Methodology

Structured light imaging is an active 3D reconstruction technique. Its core principle is to project known encoded patterns onto the surface of an object using a projector, and capture the resulting deformations of these patterns with a camera. By analyzing the deformations together with the calibrated parameters of the camera–projector system, the 3D geometry of the object surface can be accurately reconstructed. The structured light pipeline typically consists of three main stages: (1) Pattern Generation, (2) Pattern Decoding, and (3) Geometric Reconstruction.

Pattern Generation. To uniquely encode projector pixels along the horizontal axis, we employ an n -bit Gray code sequence. Gray code is a binary encoding where consecutive codewords differ by only one bit. This property ensures that errors at stripe boundaries do not propagate, which makes Gray code highly suitable for robust pattern decoding in structured light systems. The recursive definition of an n -bit Gray code is given by:

$$\begin{aligned} G(1) &= [0, 1], \\ G(n) &= \left[\underbrace{0 \parallel G(n-1)}_{\text{first half}}, \underbrace{1 \parallel \text{reverse}(G(n-1))}_{\text{second half}} \right]. \end{aligned} \quad (3.1)$$

Given n encoding bits, the total number of horizontal stripes is 2^n . Each bit corresponds to one binary pattern image, such that stacking all n bit-planes yields a unique binary code for each column of the projector. Formally, the i -th pattern image is defined as:

$$P_i(x, y) = 255 \times b_i \left(\left\lfloor \frac{x}{2^i} \right\rfloor \right), \quad (3.2)$$

where $b_i(k) \in \{0, 1\}$ denotes the value of the i -th bit of the k -th Gray code. In addition to the binary stripe patterns, two auxiliary frames, P_{black} (all black) and P_{white} (all white), are projected to provide reference measurements for reflectance normalization.

Camera–Projector Calibration. Accurate 3D reconstruction requires calibration of both intrinsic and extrinsic parameters of the camera–projector pair. For the camera, modeled as a pinhole system, the focal length in pixels can be derived from the field of view (FOV) and image resolution (W, H) as:

$$f_x = \frac{W/2}{\tan(\text{FOV}/2)}, \quad f_y = f_x, \quad (3.3)$$

with the principal point located at the image center:

$$c_x = W/2, \quad c_y = H/2. \quad (3.4)$$

Thus, the camera intrinsic matrix is

$$K_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.5)$$

The projector is modeled as an inverse camera, where the number of encoding stripes determines its effective focal length. Assuming an encoding width of 2^n and a scaling factor s , we have

$$f_p = 2^n s, \quad c_p = 2^{n-1}.$$

Accordingly, the projector intrinsic matrix is

$$K_p = \begin{bmatrix} f_p & 0 & c_p \\ 0 & f_p & c_p \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.6)$$

The relative pose between camera and projector is described by extrinsic parameters. The rotation matrix R is constructed from Euler angles (α, β, γ) with ZYX convention:

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha). \quad (3.7)$$

The translation vector is defined as

$$t = [t_x, t_y, t_z]^\top.$$

The full projection matrix is then

$$M = K[R \mid t]. \quad (3.8)$$

Image Acquisition and Normalization. During measurement, the projector sequentially displays n Gray code patterns along with one black and one white frame. The camera captures the scene under each projection, producing a sequence of images:

$$I_i(u, v), \quad i = 0, 1, \dots, n+1,$$

where (u, v) denotes pixel coordinates. Since surface reflectance and ambient illumination may distort intensity values, normalization is applied using black/white references:

$$I_{\min}(u, v) = \min_i I_i(u, v), \quad I_{\max}(u, v) = \max_i I_i(u, v), \quad (3.9)$$

$$\tilde{I}_i(u, v) = \frac{I_i(u, v) - I_{\min}(u, v)}{I_{\max}(u, v) - I_{\min}(u, v)}. \quad (3.10)$$

Gray Code Decoding. Each normalized frame is binarized with a fixed threshold T (commonly 0.5) to recover bit values:

$$b_i(u, v) = \begin{cases} 1, & \tilde{I}_i(u, v) > T, \\ 0, & \text{otherwise.} \end{cases} \quad (3.11)$$

The bits are combined into an integer Gray code value:

$$G(u, v) = \sum_{i=0}^{n-1} b_i(u, v) \cdot 2^i. \quad (3.12)$$

A lookup table is then used to map Gray codes to natural binary indices:

$$k = \text{GrayToIndex}(G(u, v)), \quad u_p = k, \quad v_p = v, \quad (3.13)$$

where (u_p, v_p) denotes the corresponding projector pixel.

Geometric Reconstruction. With projector–camera correspondences established, 3D reconstruction reduces to stereo triangulation. Each pixel pair (u_c, v_c) and (u_p, v_p) must satisfy:

$$s_c \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = M_c \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad s_p \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = M_p \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.14)$$

where M_c, M_p are projection matrices, and s_c, s_p are scale factors. By eliminating these unknowns, we obtain a linear system:

$$A \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{b}, \quad (3.15)$$

where

$$A = \begin{bmatrix} m_{11} - u_c m_{31} & m_{12} - u_c m_{32} & m_{13} - u_c m_{33} \\ m_{21} - v_c m_{31} & m_{22} - v_c m_{32} & m_{23} - v_c m_{33} \\ p_{11} - u_p p_{31} & p_{12} - u_p p_{32} & p_{13} - u_p p_{33} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} u_c m_{34} - m_{14} \\ v_c m_{34} - m_{24} \\ u_p p_{34} - p_{14} \end{bmatrix}.$$

Solving this yields the 3D point (X, Y, Z) . Depths outside a valid range are discarded as outliers.

Discussion. The Gray-code structured-light pipeline presented here mirrors the operating principles of industrial 3D cameras. Starting from Gray code design, reflectance normalization, bit-plane decoding, and projector–camera triangulation, the system produces dense depth maps and point clouds compatible with real-world devices. Its robustness is enhanced by single-bit Gray transitions, normalization with reference frames, and

masking invalid pixels. Reconstruction accuracy depends on code depth, baseline length, and calibration quality. The modular architecture also enables fair benchmarking, hardware replacement, ablation studies, and Sim-to-Real transfer, while leaving room for advanced add-ons such as sub-pixel phase-shifting refinement or spatio-temporal filtering.

3.4 Synthetic and Real-World Dataset Construction for Sim2Real Evaluation

To support downstream perception tasks such as instance segmentation, object detection, and robotic grasping, we construct an RGB-D dataset composed of photorealistic synthetic scenes and corresponding real-world scenes. This dataset is derived from our proposed structured light simulation pipeline and serves as both a training and evaluation benchmark for Sim2Real performance validation.

3.4.1 Object Database



Figure 3.2: Object Database. a) Metal Workpiece 1. b) Metal Circle Workpiece 2. c) YellowSaltCube for KIT object model database. d) HygieneSpray from KIT object model database. e) Toothpaste from KIT object model database. f) The corresponding real objects.

As shown in Fig. 3.2, the object database used for both synthetic and real-world data generation includes two dark gray metallic industrial parts sourced from real factories, and three common household objects selected from the public KIT object models database [49]. These objects are representative of components frequently encountered in pick-and-place tasks, including metallic items and everyday supermarket merchandise.

The selection ensures that both industrial and household contexts are reflected in the dataset. The physical objects are used not only for 3D mesh-based simulation but also for real-world capture and testing.

3.4.2 Synthetic and Real-World Dataset Creation

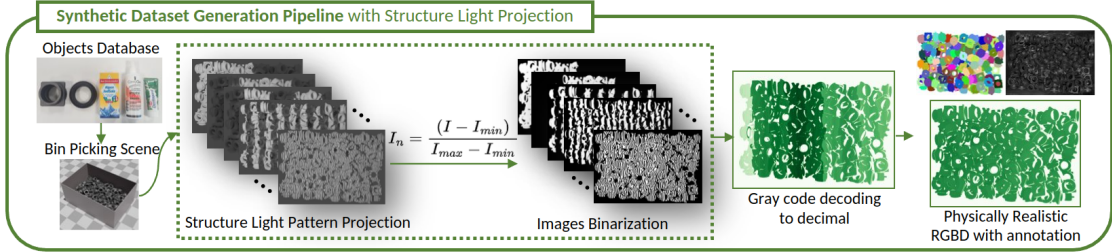


Figure 3.3: Physically-based structured-light simulation and reconstruction pipeline.

To support domain adaptation and Sim2Real evaluations, we construct both a large-scale synthetic dataset and a real-world dataset using the same set of five objects. The synthetic dataset is generated via our proposed physically-based structured light simulation pipeline, as illustrated in Fig. 3.3. In this pipeline, a physically-based rendering engine first generates bin-picking scenes, where each scene contains a single object category with multiple instances arranged in a bin-like environment. For industrial metal parts, we simulate up to 255 instances per scene to capture high-density stacking conditions, whereas for household objects from the KIT dataset, up to 10 instances are placed in each bin to reflect lower-density scenarios.

During rendering, various gray code patterns are projected into the scene, simulating the operation of a structured-light camera system consisting of one projector and one camera. These patterns serve as binary-coded images for subsequent 3D reconstruction. The rendering process employs ray tracing to incorporate realistic lighting, shadows, and material properties. The output of each scene includes RGB images, depth maps reconstructed via the simulated structured-light process, ground-truth depth maps, object poses, and instance segmentation masks. Our method also injects realistic structured-light noise into the synthetic depth images, enhancing domain fidelity and enabling robust Sim2Real transfer.

In contrast, the real-world dataset is collected using an industrial structured light camera under conditions closely matched to the synthetic scenes. For each object, we record 100 real samples, with manually annotated masks and aligned camera parameters. This pairing of synthetic and real data allows us to study Sim2Real transfer under controlled settings. Additionally, for comparison, we generate a domain-randomized dataset using Isaac Sim, varying object number, pose, lighting, and background textures.

3.4.3 Annotation Formats and Ground-Truth Generation

In our synthetic dataset, all annotations are automatically generated during the simulation process, covering object poses, instance masks, 2D bounding boxes, and both ideal

and reconstructed depth maps. The instance masks are stored using run-length encoding (RLE), which not only ensures storage efficiency and compatibility with standard detection and segmentation benchmarks, but also naturally supports complex shapes with holes in the segmentation masks. Object poses are expressed in camera coordinates and precisely aligned with the intrinsic and extrinsic parameters. The depth maps include both the ground-truth geometric depth and the decoded structured light depth, enabling direct performance comparison between different depth sources.

For the real-world dataset, instance masks are manually annotated and stored in polygon format to facilitate editing and integration with common toolchains. To ensure consistency in training and evaluation, all camera parameters and object IDs are strictly aligned with the synthetic dataset, allowing direct assessment of perception model generalization across domains.

3.5 Experimental Evaluation

To evaluate the effectiveness of our proposed physically-based structured light simulation pipeline, we conduct a series of experiments on object detection and instance segmentation tasks. All models are trained using synthetic data generated by our pipeline and tested on real-world datasets captured under consistent object and sensor configurations. The evaluation emphasizes Sim2Real transfer performance, robustness to different input modalities, and the impact of rendering realism on downstream perception accuracy. We also benchmark our approach against traditional domain randomization strategies to verify its practical advantages in industrial applications.

3.5.1 Task Setup and Evaluation Metrics

We evaluate two vision tasks: object detection and instance segmentation. Models are trained on synthetic data generated by our structured light simulation framework and evaluated on real-world datasets for the same object set. For object detection, we adopt YOLOv3 [90] and YOLOv7 [110]; for instance segmentation, SOLOv2 [114] is employed. These models offer a good balance of accuracy, deployment efficiency, and reproducibility.

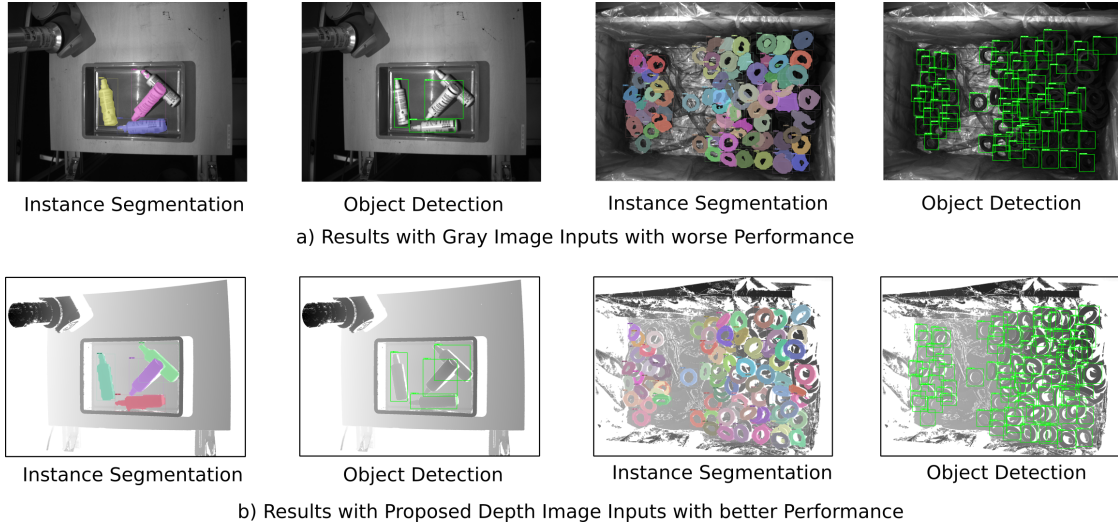
The evaluation primarily uses mean Average Precision (mAP) at IoU = 0.5, and where applicable, we report IoU and mask accuracy. To quantify Sim2Real transferability, we analyze performance differences between synthetic validation results and real-world testing outcomes. Performance is reported per object and per modality.

3.5.2 Object Detection and Instance Segmentation

We evaluate object detection and instance segmentation using YOLOv3 [90] and SOLOv2 [114] trained exclusively on our synthetic structured-light dataset, without any fine-tuning on real-world data. Table 3.1 summarizes the quantitative results on both synthetic and real test sets, along with the Sim2Real gap. Average precision (AP) is computed at Intersection over Union (IoU) as 0.50.

Table 3.1: Quantitative evaluation results of object detection and instance segmentation.

Object	Input Type	Object Detection (YOLOv3 [90])			Instance Segmentation (SOLOv2 [114])		
		Synthetic	Real	Gap	Synthetic	Real	Gap
Metal Workpiece 1.	RGB	0.624	0.547	0.077	0.711	0.620	0.091
	Ground Truth Depth	0.613	0.527	0.086	0.692	0.557	0.135
	Proposed Synthetic Depth	0.634	0.617	0.017	0.698	0.646	0.052
Metal Circle Workpiece 2.	RGB	0.673	0.614	0.059	0.646	0.565	0.081
	Ground Truth Depth	0.662	0.633	0.029	0.636	0.544	0.092
	Proposed Synthetic Depth	0.674	0.628	0.046	0.634	0.627	0.007
Toothpaste	RGB	1.000	0.885	0.115	1.000	0.953	0.047
	Ground Truth Depth	0.999	0.950	0.049	1.000	0.970	0.030
	Proposed Synthetic Depth	1.000	0.980	0.020	1.000	0.979	0.021
YellowSaltCube	RGB	1.000	0.826	0.174	1.000	0.621	0.379
	Ground Truth Depth	0.996	0.678	0.318	0.998	0.802	0.196
	Proposed Synthetic Depth	1.000	0.827	0.173	1.000	0.863	0.137
HygieneSpray	RGB	1.000	0.947	0.053	1.000	0.963	0.037
	Ground Truth Depth	0.999	0.797	0.202	1.000	0.875	0.125
	Proposed Synthetic Depth	0.980	0.953	0.027	1.000	0.966	0.034

**Figure 3.4:** Qualitative visualization results in the real-world dataset using models trained on our synthetic datasets: (a) gray image inputs; (b) proposed depth image inputs.

Overall, models trained with our proposed synthetic depth input outperform those trained with RGB or ground-truth depth in most cases. For example, YOLOv3 [90] with proposed synthetic depth achieves an IoU of 0.617 on *Metal Workpiece 1*, compared to 0.547 for RGB and 0.527 for ground-truth depth. Similarly, SOLOv2 [114] with our depth reaches 0.646 on the same object, reducing the Sim2Real gap from 0.091 (RGB) to 0.052.

In high-clutter scenes such as *YellowSaltCube*, our method reduces the Sim2Real gap from 0.379 (RGB) to 0.037, while maintaining perfect synthetic performance. These results demonstrate that physically-based synthetic depth can provide both higher real-world accuracy and smaller domain gaps compared to conventional inputs.

Fig. 3.4 illustrates the qualitative differences between models trained with gray image inputs and those trained with our proposed depth image inputs. When tested on

cluttered real-world scenes, the RGB-trained models exhibit segmentation leakage and frequent missed detections, particularly for partially occluded or overlapping objects. In contrast, our depth-trained models generate tighter bounding boxes and more accurate instance masks, even in high-occlusion conditions. These observations align with the quantitative results reported in Table 3.1, further confirming the robustness and Sim2Real generalization capability of depth-guided models.

3.5.3 Domain Adaptation vs. Domain Randomization



Figure 3.5: Examples from the domain-randomized dataset generated using the Isaac Sim platform.

To evaluate different data generation strategies, we compare our physically-based simulation approach (*domain adaptation*) with a *domain randomization* approach implemented in Isaac Sim. As illustrated in Fig. 3.5, both datasets use the same object set. However, the randomized dataset introduces variations in background textures, object counts, poses, and lighting conditions (in both intensity and color) without physically grounded rendering, resulting in greater visual diversity but reduced physical consistency.

Fig. 3.5 shows examples from the domain randomization dataset, where each parameter is randomized to increase variability. While this approach can expand the coverage of visual appearances, it does not accurately model the physical imaging process. In contrast, our domain adaptation method leverages physically based rendering and structured light simulation, ensuring geometric and photometric consistency with the real world.

We further evaluated both strategies using the YOLOv7 [110] model trained on RGB and depth datasets derived from each approach. On the real-world test set for the *Metal*

Workpiece 1 object, the domain adaptation-based RGB dataset achieved an IoU of 0.628, while our proposed depth image dataset achieved an IoU of 0.648. Both results exceeded the IoU of 0.584 obtained from the domain randomization-based RGB dataset. These findings indicate that, in industrial grasping scenarios, typically static with consistent lighting, the performance gains provided by domain adaptation surpass those achieved through domain randomization.

In summary, utilizing depth images as input for deep learning perception tasks enhances model robustness, efficiency, and performance by providing a stable representation of 3D structure and facilitating shape and texture separation, while reducing sensitivity to variations in lighting and surface appearance.

3.6 Robotic Grasping Experiments

To evaluate the real-world applicability of our structured-light-based synthetic perception pipeline, we integrate the trained models into robotic grasping systems. Two experimental setups are considered: one using a full model-based grasping pipeline, and another following a half-model-based approach. These experiments validate the end-to-end Sim2Real performance of our system, from perception to execution.

3.6.1 Model-based and Model-free Grasping Setup



Figure 3.6: Experimental setups for robotic grasping: (left) model-based approach; (right) half-model-based approach.

As shown in Fig. 3.6, the model-based setup involves a Diana7 robot equipped with a Photoneo L structured light camera, operating in a bin-picking scenario with densely stacked metallic workpieces. The 3D CAD model of the target object is used to generate grasp candidates through pose estimation and model-matching.

On the other hand, the model-free setup utilizes a UR5e robot and a Photoneo M camera, where only partial geometry or no CAD model is available. In this case, grasping relies on depth-based perception models trained on our synthetic data, combined with learning-based grasp prediction methods such as Dex-Net 3.0 [70].

3.6.2 Grasping Performance and Deployment Efficiency

By incorporating our Sim2Real visual model into the model-free grasping pipeline, we observe a notable improvement in real-world performance. Using Dex-Net 3.0 as the grasping backend, the success rate increases from 95.6% to 98.0%, showing that more accurate instance segmentation and object localization reduce grasping ambiguity.

For the model-based setup, which traditionally requires time-consuming manual annotation for object detection training, our method shortens deployment time. The visual model trained on synthetic depth data enables accurate detection without real-world labeling, reducing development time from two weeks to two days. Additionally, the precision in detection boxes helps avoid oversized or undersized bounding boxes that could affect pose estimation speed or grasp stability.

The experiments demonstrate that our structured light simulation pipeline provides not only robust perception under domain shift, but also practical benefits in robotic task success and engineering efficiency.

3.7 Discussion

Our results demonstrate that physically-based simulation of structured light imaging is a viable and effective strategy to address the data scarcity and domain gap problems that plague many real-world industrial vision applications. When synthetic depth data is rendered with accurate light-material interaction and realistic sensor noise modeling, it can serve as a strong alternative to manually annotated real-world data. This enables the development of robust and scalable perception models while avoiding the costs of data labeling or the complexity of extensive domain randomization.

One of the core insights of this work is that depth input contributes more significantly to cross-domain generalization than RGB input, particularly in industrial settings. This observation is consistent with practical challenges in robotic perception. In bin-picking and part inspection tasks, metal components often exhibit specular or textureless surfaces, which makes it difficult for RGB-based detectors to extract stable features under varying lighting conditions. In warehouse and retail product sorting scenarios, RGB-based models are similarly affected by frequent appearance changes caused by promotional campaigns or packaging updates, even though the physical shape of the products remains unchanged. In both cases, reliance on RGB inputs introduces uncertainty and brittleness into perception pipelines. By contrast, structured-light depth images emphasize geometric structure and spatial consistency, leading to improved stability and better transferability across domains.

Our proposed rendering pipeline explicitly simulates light-material interactions and structured light projection with shadowing and occlusion modeling, which contributes to better downstream performance than random scene generation approaches. Compared to domain randomization, which focuses on creating diverse but physically inconsistent appearances, our physics-based rendering shows smaller Sim2Real performance gaps and more consistent behavior across object types. This supports the conclusion that fidelity in simulation is more impactful than variability when bridging the reality gap in

industrial settings.

Despite its advantages, the proposed framework still has limitations. The ray-tracing-based rendering process is computationally more expensive than rasterization-based pipelines, which may hinder scalability in very large-scale dataset generation. Additionally, our pipeline assumes known camera intrinsics and lacks integration with closed-loop robot control or online calibration, which limits real-time adaptability.

Nevertheless, this work establishes a principled and modular foundation for synthetic depth data generation in robotic perception. Future research will explore integration with feedback-driven systems, support for other sensor types such as ToF or event cameras, and simulation of more diverse object materials and lighting interactions. These efforts will help further close the Sim2Real gap and promote scalable deployment of deep learning systems in industrial automation.

Chapter 4

Robust Stereo Perception for Transparent Objects

4.1 Research Motivation and Challenges in Transparent Object Perception

Transparent objects, such as glass bottles, cups, and cosmetic containers, are common in both domestic and logistics-oriented robotic applications. Their high transparency and refractive surfaces severely impair the performance of conventional depth sensors (e.g., RGB-D cameras), resulting in ambiguous or missing depth information. As noted in earlier research [67], standard imaging systems struggle to accurately capture the position and surface geometry of such objects when light is unpredictably refracted through them. For robots that depend on precise 3D perception, whether for household assistance or warehouse sorting, this visual ambiguity caused by transparency can significantly degrade task performance.

The core challenges arise from two fundamental optical phenomena:

1. **Refraction-induced disparity errors:** Unlike opaque surfaces, transparent materials refract light from the background rather than reflecting incident rays, leading to misleading stereo cues and conventional matching failures.
2. **Weak structural and texture signals:** Because the observed image combines both object and background, there is a scarcity of reliable feature correspondence, with no consistent edges, textures, or contrast, making traditional stereo or depth completion models unreliable.

4.1.1 The Role of Depth Accuracy in Robotic Manipulation

In robotic manipulation, particularly in tasks involving dexterous grasping or tight-clearance interactions, the accuracy of perceived depth is not merely a matter of visual plausibility but the foundation for ensuring physical feasibility. Transparent objects pose a particularly critical challenge in this regard. Owing to their refractive and often specular surfaces, even slight deviations in estimated geometry can propagate through the

control pipeline and lead to execution failures. A depth error of only a few millimeters, though imperceptible to humans, may result in unstable grasps, misaligned trajectories, or even physical collisions, especially when the robot interacts with slender glassware or nested containers.

The same precision requirements apply in structured industrial environments. In logistics or automated assembly lines, robots are expected to perform consistent and efficient pick-and-place operations. In such contexts, perception errors translate directly into reduced throughput or mechanical faults. For transparent objects such as blister packs, acrylic trays, or shrink-wrapped goods, conventional depth sensors often produce noisy or incomplete depth data, which can lead to misdetections or localization drift. In multi-object scenes, these errors can cause significant downstream disruptions, as grasp planning algorithms may select infeasible candidates and placement trajectories may intersect with other objects.

Accurate perception of transparent objects therefore requires more than generating visually plausible depth maps. It demands structurally faithful reconstruction that is both metrically accurate and physically consistent. This involves capturing subtle cues such as object boundaries, surface curvature, thickness gradients, and spatial continuity, even in the absence of strong visual textures. Achieving such fidelity is particularly critical in real-time robotic systems, where perception and control must operate in tightly coupled loops without human intervention.

4.1.2 Dataset scarcity and simulation gap

High-quality depth datasets for transparent objects remain extremely limited and constitute a major bottleneck for the development of perception models in transparent scenes. Unlike opaque objects, transparent surfaces significantly interfere with structured light, active infrared sensing, and stereo correspondence mechanisms, making real-world data collection with RGB-D sensors unreliable. The resulting depth measurements often suffer from severe noise, missing regions, and systematic biases, which prevent them from serving as stable and accurate supervision for model training.

To circumvent the ambiguity of real-world depth annotations, many existing approaches rely on synthetic data generation. However, a large portion of current synthetic transparent object datasets adopt simplified scene configurations and rendering assumptions, such as uniform or unrealistic backgrounds, restricted material variations, or the use of explicit object segmentation as a prerequisite or post-processing step. While these simplifications facilitate data generation, they introduce strong priors that limit scalability and reduce the ability of trained models to generalize to complex real-world environments.

At the rendering level, different simulation strategies involve a clear trade-off between physical realism and computational efficiency. Physically based ray tracing can accurately model refraction, reflection, and multi-path light transport, producing high-fidelity renderings of transparent objects, but it incurs substantial computational cost that hinders large-scale data generation. In contrast, rasterization-based rendering pipelines are significantly faster and easier to deploy at scale, yet their limited ability to capture complex optical effects often results in unrealistic depth cues for transparent surfaces.

Consequently, for robotic perception systems intended for real-world deployment, constructing synthetic datasets requires carefully balancing physical accuracy, generation efficiency, and dataset scale. If the resulting simulation gap is not properly addressed, models trained on synthetic data may exhibit degraded stability and reliability when transferred to real-world scenarios.

4.1.3 Limitations of existing methods

Recent years have witnessed growing interest in transparent object perception, leading to a wide array of approaches ranging from hardware-based sensing enhancements to learning-based depth reconstruction. Despite their respective technical merits, these methods face fundamental limitations when considered for real-time deployment in robotic manipulation systems

Hardware-centric solutions, such as those involving polarimetric cameras [48] or active lighting setups, have demonstrated effectiveness in suppressing specular reflections and enhancing surface reconstruction under controlled conditions. However, their dependency on specialized sensors and calibrated lighting environments poses significant barriers to scalability. The complexity, cost, and fragility of such setups render them impractical for many real-world applications where flexibility, robustness, and rapid adaptation are es

In the domain of learning-based methods, RGB-D depth completion networks, such as [126] and the RGB-D Local Implicit Function framework [145], have been widely explored as a way to estimate complete geometry from partially missing depth maps. These approaches typically take as input the RGB and partial depth images captured by consumer-grade RGB-D cameras, most notably Intel RealSense. The popularity of RealSense in academia and prototyping environments has led to its extensive use as a data source for training and evaluating depth completion models. Consequently, many of these networks are tightly coupled with the structure, resolution, and noise characteristics of RealSense-generated data.

While RealSense offers convenience and affordability, it is fundamentally ill-suited for industrial use. Its depth images tend to be sparse and noisy, particularly in the presence of transparent or reflective materials where the structured infrared light often passes through the surface instead of returning reliable measurements. The performance of RealSense sensors is also sensitive to temperature variation, and their calibration may drift over time. Moreover, RealSense uses non-industrial communication interfaces (e.g., USB), lacks robust physical housing, and does not comply with industrial standards regarding durability, EMI shielding, or real-time I/O. As a result, depth completion pipelines built on RealSense data are inherently incompatible with the deployment needs of industrial robotic systems operating in safety-critical or high-throughput environments.

Beyond RGB-D methods, stereo and multi-view approaches, such as stereo-based pose estimation [15], MVTrans [115], and NeRF-style volumetric reconstruction [64, 27, 63], explicitly model view-dependent effects and scene geometry. These methods can deliver high-fidelity results, but they often require dense camera arrays, precise calibration, or expensive optimization procedures. NeRF and neural rendering-based

methods, in particular, involve iterative volumetric sampling that is computationally intensive and unsuitable for real-time robotic inference. Their latency, memory demands, and offline nature make them difficult to integrate into reactive robotic control loops.

Deep stereo matching frameworks, including RAFT-stereo and cost-volume CNNs, offer fast inference and have shown competitive performance in textured, opaque scenes. Yet their reliance on photometric consistency for matching makes them fragile in transparent regions where refraction violates brightness constancy. Without explicit modeling of material properties or geometric priors, these networks struggle to infer meaningful depth in cases of visual ambiguity.

Altogether, existing methods suffer from a range of interconnected limitations. Some rely on controlled hardware environments that hinder scalability, while others assume opaque, texture-rich scenes not representative of transparent object scenarios. Several approaches depend on non-industrial sensors like RealSense, which limits their field applicability. Others lack structural reasoning and cannot cope with the physical ambiguity caused by refraction. In addition, many solutions are designed for offline processing and disregard the real-time constraints imposed by robotic systems. Addressing these limitations calls for a structurally aware, sensor-agnostic, lightweight stereo model capable of generalization from synthetic to real-world data. ClearDepth is proposed precisely to bridge this gap.

4.1.4 Summary of challenges

In summary, the perception of transparent objects presents a distinct combination of challenges that are rarely encountered in conventional robotic vision tasks.

1. **Optical ambiguity in depth cues:** Refraction and the blending of background information distort geometric measurements, leading to unreliable or missing depth estimates.
2. **Absence of consistent structural features:** The scarcity of reliable edges, textures, and contrast severely limits correspondence estimation, particularly for stereo and multi-view reconstruction.
3. **Limited data availability and sim-to-real transferability:** The lack of large-scale, high-fidelity datasets, coupled with the domain gap between synthetic and real-world data, constrains model generalization and robustness.

Addressing these challenges requires perception methods that jointly account for optical physics, structural reasoning, and domain adaptation, while remaining suitable for real-time robotic deployment.

4.2 System Architecture for Transparent Object Stereo Perception

Transparent object scenes exhibit unique visual characteristics, such as light refraction, reflection, and background penetration, which make it particularly challenging to accu-

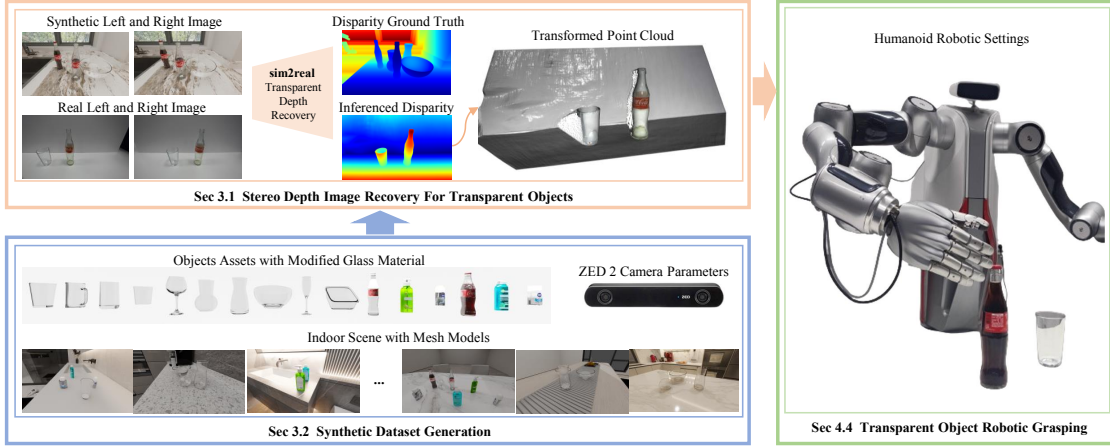


Figure 4.1: Overview of the proposed transparent object stereo reconstruction and robotic grasping system.

rately recognize transparent objects based on texture or local features. Since transparent objects often lack well-defined surface textures and their appearance is heavily dependent on background and environmental lighting conditions, their recognition relies more on the integration of global shape cues and contextual information.

Compared to conventional Convolutional Neural Networks (CNNs), Vision Transformers (ViTs) offer distinct advantages in processing transparent objects. CNNs operate with local receptive fields and strong local inductive biases, making them prone to relying on local features such as texture. This tendency proves ineffective for transparent objects, where texture features are typically weak or even imperceptible [6]. ViTs, on the other hand, are based on a global attention mechanism that enables them to capture long-range dependencies and aggregate visual information across the entire image. This allows ViTs to infer the contours and spatial structures of transparent objects more effectively [106]. Furthermore, due to their weaker inductive biases, ViTs are capable of learning more flexible and generalized feature representations during training, beyond local patterns. This capability is especially crucial for recognizing objects that depend on global structures or non-local visual cues. Consequently, when dealing with the complex optical properties and inherent visual ambiguities of transparent objects, ViTs tend to exhibit more human-like perceptual behavior and achieve more accurate and stable recognition results [56].

Conventional ViTs divide an input image into fixed-size patches and maintain a consistent number and size of tokens across all layers, which restricts the model to learning single-scale features. Pyramid ViTs, by contrast, progressively reduce the number of tokens and increase the channel dimensions as the network deepens, forming a feature pyramid structure akin to that of CNN-based Feature Pyramid Networks (FPN). This hierarchical structure is better suited for tasks such as object detection, semantic segmentation, and pose estimation, where recognizing objects at multiple scales is crucial. The pyramid structure also enables output features to retain spatial hierarchy, facilitating integration with downstream modules such as FPN or Mask Head. In Pyramid ViTs, early layers retain a larger number of tokens to preserve fine details, while later layers

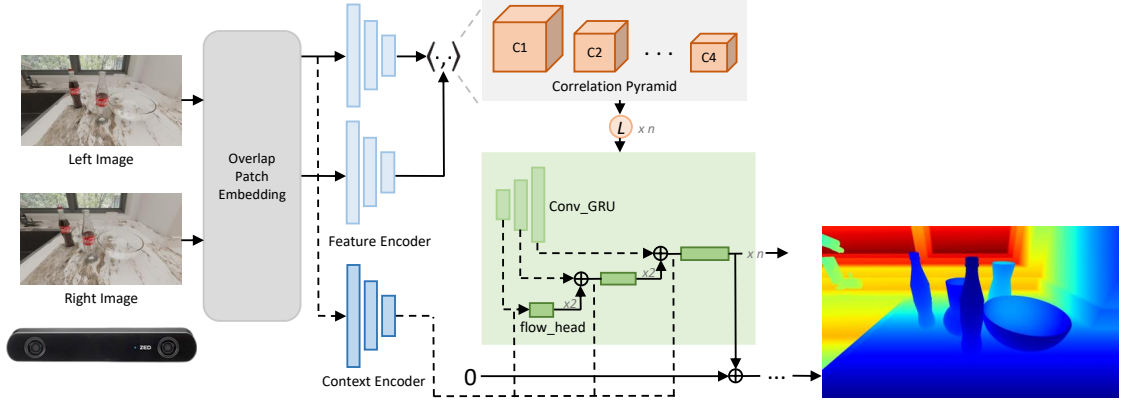


Figure 4.2: Our stereo depth recovery network for transparent objects.

reduce the spatial resolution through pooling or strided attention, significantly lowering computational overhead. Moreover, this design enables processing of higher-resolution images, which is particularly advantageous for tasks like stereo reconstruction that require high-resolution stereo imagery [122].

Our network architecture consists of two main components, namely a vision task encoder and a subsequent feature fusion module. To facilitate deployment in real-world engineering scenarios and to align with recent trends in foundation model design, we adopt a heavier encoder together with a lightweight decoder. In addition, this section describes the process used to generate the synthetic dataset for model training. The overall network architecture is illustrated in Fig. 4.2. Beyond the network itself, the complete system pipeline, including synthetic data generation, transparent depth recovery, and downstream robotic grasping experiments, is shown in Fig. 4.1. This figure provides a system-level view of how the proposed perception model integrates with subsequent manipulation tasks.

4.2.1 Cascaded Vision Transformer Backbone

The backbone network employed in our framework is based on the B5 configuration of SegFormer [122], and is initiated with an overlap patch embedding strategy. This design effectively preserves local structural information during the initial tokenization of the input image. The embedded tokens are subsequently processed by four hierarchical Transformer blocks, producing multi-scale feature maps at spatial resolutions of $1/4$, $1/8$, $1/16$, and $1/32$, respectively.

To improve computational efficiency, we incorporate an efficient self-attention mechanism that reduces the original quadratic complexity $O(N^2)$ to $O(\frac{N^2}{R})$ by applying a reduction ratio R . Specifically, the input sequence is reshaped from $N \cdot C$ to $\frac{N}{R} \times (C \cdot R)$ using 2D convolutional layers with strides of 8, 4, 2, and 1 across the different Transformer stages, as shown in Equation 4.1. The sequence is then projected back to the original channel dimension C via a linear layer (Equation 4.2):

$$\hat{K} = \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K) \quad (4.1)$$

$$K = \text{Linear}(C \cdot R, C)(\hat{K}) \quad (4.2)$$

To further enhance robustness to varying input resolutions and avoid performance degradation typically caused by interpolated positional embeddings in standard ViT architectures, we adopt the Mix-FFN module. This module replaces absolute positional encodings with learnable depth-wise convolutions, as described in Equation 4.3:

$$\mathbf{x}_{\text{out}} = \text{MLP}(\text{GELU}(\text{Conv}_{3 \times 3}(\text{MLP}(\mathbf{x}_{\text{in}})))) + \mathbf{x}_{\text{in}} \quad (4.3)$$

Finally, feature maps generated at different scales are upsampled to a uniform resolution of $1/4$ and concatenated. This aggregated feature map is then refined using a 1×1 convolutional layer to normalize channel dimensions, ensuring compatibility with downstream fusion modules.

4.2.2 Structural Feature Post-Fusion

In this study, we design a modified GRU-based architecture tailored for coarse-to-fine disparity refinement, with a particular focus on transparent object scenes. To address the inherent challenges posed by transparent materials—such as ambiguous texture cues and background interference caused by refraction—we introduce a structural post-fusion mechanism into the recurrent update process. During empirical evaluation, we observed that transparent object reconstruction demands greater reliance on precise structural information, whereas traditional dot-product-based similarity features, commonly effective for opaque objects, are often misleading in the presence of refractive distortions.

To remedy this, our model incorporates structural features extracted from the reference image at multiple levels—denoted as c_k , c_r and c_h —into the GRU recurrence loop as residuals. These features help preserve structural consistency across iterations and resolutions. The GRU update is governed by the following equations:

$$x_k = [\mathbf{C}_k, \mathbf{d}_k, \mathbf{c}_k, \mathbf{c}_r, \mathbf{c}_h] \quad (4.4)$$

$$z_k = \sigma(\text{Conv}([h_{k-1}, x_k], W_z) + c_k), \quad (4.5)$$

$$r_k = \sigma(\text{Conv}([h_{k-1}, x_k], W_r) + c_r), \quad (4.6)$$

$$\tilde{h}_k = \tanh(\text{Conv}([r_k \odot h_{k-1}, x_k], W_h) + c_h), \quad (4.7)$$

$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k, \quad (4.8)$$

Here, x_k concatenates the correlation volume, disparity estimate, and structural cues. These cues are vital for guiding the network in regions where appearance-based matching fails due to transparency.

To achieve high-resolution output in a computationally efficient manner, we adopt a coarse-to-fine refinement strategy. GRU features are decoded at three scales to generate incremental disparity updates:

$$\Delta \mathbf{d}_{k, \frac{1}{32}} = \text{Decoder}(h_{k, \frac{1}{32}}), \quad (4.9)$$

$$\Delta \mathbf{d}_{k, \frac{1}{16}} = \text{Decoder}(h_{k, \frac{1}{16}} + \text{Interp}(\Delta \mathbf{d}_{k, \frac{1}{32}})), \quad (4.10)$$

$$\Delta \mathbf{d}_{k, \frac{1}{8}} = \text{Decoder}(h_{k, \frac{1}{8}} + \text{Interp}(\Delta \mathbf{d}_{k, \frac{1}{16}})), \quad (4.11)$$

where Decoder consists of two convolutional layers and Interp denotes bilinear up-sampling. The final disparity is updated as:

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \Delta \mathbf{d}_k \quad (4.12)$$

In summary, our model leverages a structural-aware GRU framework and a lightweight encoder-heavy decoder design. This enables robust performance on transparent object stereo reconstruction while meeting the efficiency requirements of practical deployment in robotic systems. Visual comparisons in the experimental section confirm the effectiveness of our method in handling transparent scenes.

4.2.3 Synthetic Dataset Generation

Recent advances in transparent object perception and 3D reconstruction have been significantly supported by the availability of public datasets. While several real-world datasets dedicated to transparent objects are already available in the community [31], they often fall short in terms of data quality and practical utility. In Depth Anything v1 [130], both real and synthetic datasets were used for training to enhance generalization. However, in Depth Anything v2 [131], the authors shifted entirely to synthetic data. This transition was motivated by the realization that even a small proportion of real-world data, only 10%, can substantially impair performance due to sensor-induced label noise, especially in depth estimation tasks. Furthermore, accurate annotation of real scenes containing transparent objects is both labor-intensive and error-prone due to frequent occlusions and complex interactions with the background. As a result, our work prioritizes synthetic data as the core training resource.

To reduce the domain gap between synthetic and real-world data, we adopt physically-based ray tracing to generate photorealistic images, with a particular focus on accurately modeling light refraction in transparent materials. Although such physically accurate rendering pipelines are computationally expensive, our approach is inspired by recent acceleration techniques developed in the real-time graphics and gaming industry. Specifically, we integrate NVIDIA OptiX’s AI-based denoising and deep-learning super-resolution modules [23] into the rendering pipeline, allowing early termination of Monte Carlo ray tracing while reconstructing high-quality images (Fig. 4.3).

Unlike game-oriented rendering systems that primarily optimize for perceptual realism, our design explicitly prioritizes physically realistic data distributions that are critical for learning-based depth estimation. We therefore carefully adapt these learning-based acceleration techniques to preserve physically meaningful image statistics, such

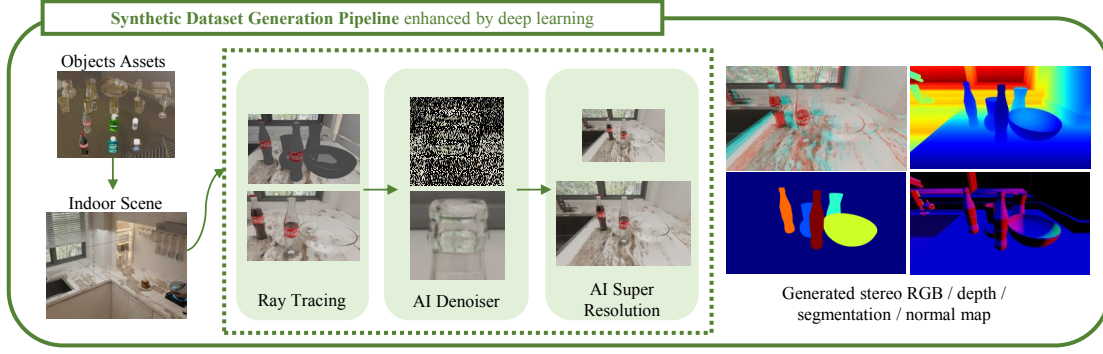


Figure 4.3: Synthetic dataset generation pipeline with physically-based rendering and learning-based acceleration.

as refraction, reflection, and shading consistency, rather than merely improving visual appearance. This deliberate trade-off enables a substantial reduction in rendering cost, decreasing the average generation time per sample from 12.77 seconds to 4.40 seconds, while maintaining the physical plausibility required for robust generalization to real-world transparent object scenarios.

Although existing synthetic datasets (e.g., [94, 145, 126, 40, 96, 26]) have made strides in transparent object modeling, they are mostly restricted to simplified desktop scenes. These settings fail to represent the complexity of real-world environments where service or humanoid robots typically operate, such as kitchens, bedrooms, and offices. Furthermore, many of these datasets require extensive pre- or post-processing, such as foreground segmentation or background reconstruction, which limits their applicability in end-to-end learning scenarios for embodied AI.

Datasets that rely on HDRI panoramic backgrounds (e.g., [115]) offer visual variety but suffer from the lack of depth annotations in the background, introducing ambiguity during training and leading to poor generalization, especially in zero-shot inference. Another category of datasets emulates RealSense camera imaging (e.g., [26, 96]). While these datasets can achieve a high degree of realism, they rely on time-consuming pipelines that involve rendering RGB images, simulating infrared (IR) projections, and generating noisy depth maps, making them inefficient for high-throughput data production. Moreover, in real-world deployments, RealSense cameras are often operated with scene-dependent settings of the IR emitter intensity and depth-related parameters, which induce noticeable domain shifts in IR observations and depth noise characteristics, thereby degrading the inference performance of models trained under fixed simulation configurations. In addition, different RealSense camera models employ fundamentally different IR projector designs and projection patterns, causing such simulation-based approaches to be effective only for a specific camera model under carefully tuned parameters. This strong dependency on both camera configuration and hardware significantly limits the generalization, scalability, and practical applicability of deep learning models trained using these datasets.

To overcome these limitations, we introduce SynClearDepth, a purpose-built synthetic dataset for accurate depth learning in transparent object scenarios. The dataset



Figure 4.4: Synthetic and real transparent object sets used in SynClearDepth.

includes 16 carefully selected transparent objects comprising 10 everyday containers (e.g., cups, spice jars) and 6 glass cosmetic products. These items are geometrically similar, though not identical, to real objects used in evaluation, enhancing the diversity and realism of experimental conditions.

In addition to the rendering pipeline design, we carefully construct a dedicated transparent object database to support physically realistic data generation and fair evaluation. As illustrated in Fig. 4.4, the synthetic object set used for data generation consists of high-quality transparent assets modeled to capture diverse geometric profiles and material properties commonly observed in daily-life containers and cosmetic products.

For evaluation, we intentionally select real-world objects that are geometrically and functionally similar, but not identical, to the synthetic ones. This design choice avoids trivial shape memorization and enables a more rigorous assessment of generalization performance, ensuring that improvements stem from physically grounded learning rather than overfitting to specific object instances.

For environment modeling, we curated 20 indoor scenes representative of real service robot settings, including 6 bathrooms, 3 dining rooms, 5 kitchens, and 6 living rooms. Each sample includes stereo RGB images at a resolution of 1280×720 , ground-truth depth maps, instance segmentation masks, and precise pose annotations for both objects and cameras. To further improve diversity and robustness, we employed domain randomization techniques, introducing variations in object types, quantities, spatial layout, poses, lighting conditions, and camera viewpoints to ensure strong generalization capability across varied indoor environments. The generated images in our dataset are shown in Fig. 4.5.

4.3 Evaluation

We begin by pretraining our model on the CREStereo dataset [60] and the Scene Flow dataset [74], both of which provide large-scale disparity supervision for opaque objects. This stage helps establish a robust disparity estimation backbone. To adapt the network to transparent object scenarios, we then conduct a fine-tuning phase using our proposed SynClearDepth dataset, which focuses specifically on transparent object stereo recon-



Figure 4.5: Sample images from our SynClearDepth dataset.

struction. All experiments are conducted on a single NVIDIA RTX A6000 GPU with a batch size of 8. The training process consists of 300,000 iterations. We employ the AdamW optimizer with an initial learning rate of 0.0002. A warm-up strategy is applied at the beginning of training, followed by a one-cycle learning rate schedule that gradually reduces the learning rate to 0.0001 by the end of training. The input resolution is standardized to 360×720 pixels. Importantly, the fine-tuning phase for transparent data uses the same set of hyperparameters as the initial training on opaque data to ensure consistency and controlled evaluation.

4.3.1 Stereo Matching Evaluation Metrics

To comprehensively evaluate the quality of disparity maps in stereo reconstruction, we adopt a suite of widely used quantitative metrics, each capturing different aspects of performance.

AvgErr (Average Error) This metric computes the mean absolute disparity error across all valid pixels:

$$\text{AvgErr} = \frac{1}{N} \sum_{i=1}^N |d_i - d_i^{\text{gt}}|, \quad (4.13)$$

where d_i is the predicted disparity and d_i^{gt} is the ground truth. Lower values indicate overall high accuracy and that the model closely matches the true disparities. Higher values suggest systematic errors in the predicted disparity map, possibly due to poor texture, occlusion, or transparency.

RMS (Root Mean Square Error) RMS calculates the square root of the mean squared disparity error:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - d_i^{\text{gt}})^2}, \quad (4.14)$$

Lower RMS signifies smaller deviations and fewer large errors, indicating both accuracy and stability of predictions. Higher RMS may indicate the presence of large outliers or noise in the disparity map.

Bad- x (%) These metrics report the percentage of pixels for which the disparity error exceeds a fixed threshold, with commonly used values being 0.5, 1.0, 2.0, and 4.0 pixels. Lower Bad- x values (especially Bad-1.0 and Bad-2.0) indicate that the model maintains high precision across the image, with few large errors. Higher Bad- x percentages mean a greater proportion of pixels with substantial disparity error, which is especially problematic in regions with fine structure, occlusion, or transparency. For transparent object scenes, high Bad-0.5 or Bad-1.0 often reflects the failure of local matching due to missing or distorted texture cues. Together, these metrics provide a comprehensive assessment of stereo matching performance, balancing both overall accuracy and the frequency of large errors.

4.3.2 Ablation Study of Feature Post-Fusion Module

Table 4.1: Comparison of disparity map quality metrics for different methods.

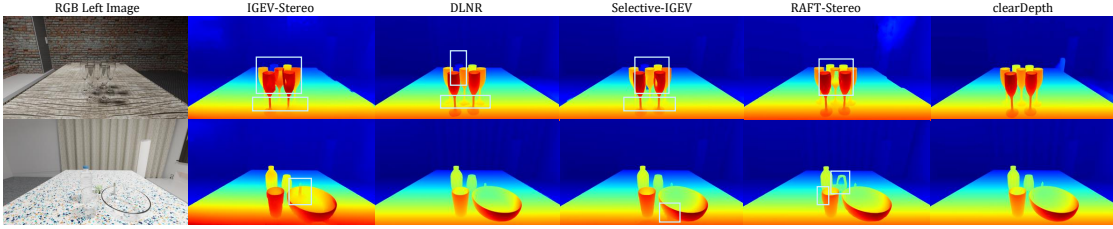
Methods	AvgErr	RMS	Bad 0.5 (%)	Bad 1.0 (%)	Bad 2.0 (%)	Bad 4.0 (%)
w/o Fusion	6.90	15.48	43.34	29.63	21.52	16.62
Feature Fusion	2.64	8.59	27.23	16.87	11.28	7.72

To rigorously assess the contribution of our structural post-fusion module, we conducted a series of targeted ablation experiments. These experiments were designed to compare two variants of the model, one equipped with the proposed module and one without, under identical training conditions using our transparent object dataset, Syn-ClearDepth. Each configuration was trained for 100,000 steps to ensure sufficient convergence.

The results, summarized in Tab. 4.1, clearly indicate that integrating the post-fusion module leads to a notable improvement in performance, particularly in scenes involving complex transparency and strong refraction effects. The module proves highly effective at preserving structural coherence and enhancing the fusion of critical contextual cues, thereby enabling more accurate depth estimation and object delineation in challenging transparent environments. These findings underscore the module’s significance in advancing the overall capability of stereo networks tailored for transparent materials.

Table 4.2: Comparison with fine-tuned stereo SOTA methods on the transparent object dataset.

Methods	AvgErr	RMS	bad 0.5 (%)	bad 1.0 (%)	bad 2.0 (%)	bad 4.0 (%)
IGEV-Stereo [125]	2.0766	5.5301	58.9743	36.1270	19.9668	10.7771
DLNR [142]	3.0997	8.4269	28.1088	21.8442	16.4810	11.9046
Selective-IGEV [113]	1.2730	4.3365	34.8229	17.6288	9.5610	5.8707
RAFT-Stereo [67]	2.2450	8.8016	29.7356	17.4521	10.8350	6.2107
ClearDepth (ours)	2.1380	8.7282	24.7329	16.3178	9.8459	5.7600

**Figure 4.6:** Visualization results of our method compared with other SOTA stereo depth estimation methods.

4.3.3 Evaluation of Transparent Object Dataset

To comprehensively evaluate the effectiveness of our proposed method in the task of stereo depth reconstruction for transparent objects, we fine-tuned our pre-trained model on the custom-built SynClearDepth dataset, using the same training parameters as in the pretraining phase. Additionally, we fine-tuned the baseline method RAFT-Stereo [67], as well as several other advanced models that perform well on the Middlebury benchmark [95], under the same experimental conditions for fair comparison.

It is worth highlighting that many of the compared models feature significantly more complex architectures and larger parameter counts than our approach. Some of them rely on computationally expensive components such as cross-attention mechanisms or multi-model ensembles, making them less feasible for real-time deployment in robotic systems. In contrast, our model is designed with a lightweight and task-specific architecture, achieving comparable or superior performance while maintaining much lower computational cost. Thanks to innovations in the image encoder and overall system optimization, our method is highly suitable for robotics. Furthermore, our TensorRT deployment achieves real-time inference at 50 FPS on a standard consumer-grade NVIDIA RTX A6000 GPU, which further underscores the practical deployability of our network.

As demonstrated by the quantitative results (Table 4.2) and visual comparisons (Fig. 4.6), our method delivers sharper object boundaries, fewer depth estimation errors, and more consistent structural reconstruction in transparent regions, highlighting both its robustness and adaptability for stereo perception of transparent objects.

To evaluate the performance of our method on transparent object stereo depth estimation, we conduct a comprehensive comparison against several state-of-the-art open-source approaches on the transparent object validation set. In particular, Foundation-Stereo [116], DEFOM-Stereo [44] claim to be capable of zero-shot generalization to

Table 4.3: Quantitative comparison on the transparent object validation set (no fine-tuning for baselines).

Methods	AvgErr	RMS	Bad 0.5 (%)	Bad 1.0 (%)	Bad 2.0 (%)	Bad 4.0 (%)
IGEV-Stereo [125]	26.8047	39.0820	0.968369	0.939858	0.890477	0.783732
DLNR [142]	26.5240	38.0410	0.962301	0.927577	0.865248	0.758555
Selective-IGEV [113]	23.8168	36.2975	0.959417	0.921393	0.854298	0.731494
RAFT-Stereo [67]	29.2919	39.2978	0.971589	0.946811	0.901663	0.821213
DEFOM-Stereo [44]	16.1635	25.8188	0.890889	0.792165	0.680660	0.550519
FoundationStereo [116]	8.8985	16.0874	0.891110	0.799528	0.668938	0.498191
ClearDepth (ours)	3.1084	6.9570	0.806759	0.631477	0.375651	0.153726

arbitrary scenes, including challenging transparent object scenarios, without additional training. For fairness, we directly use their released models and evaluate them under identical settings. All quantitative results are computed only on the foreground regions of the scenes, i.e., the object areas, since the primary goal of our algorithm is to support robotic grasping. As shown in Table 4.3, our method significantly outperforms all competitors across all metrics, achieving notably lower average error and RMS, as well as the lowest bad-pixel rates under different thresholds. These results demonstrate that our approach not only generalizes well to unseen transparent object scenes but also delivers substantial accuracy gains over existing zero-shot stereo methods that claim to be universally applicable across diverse scenes.

We conducted three sets of qualitative experiments targeting representative challenges: comparisons against existing state of the art methods, evaluation against a NeRF based depth estimation approach [40], and analysis of model robustness under varying lighting conditions. As shown in Fig. 4.7, we compare ClearDepth with three leading approaches: ClearGrasp [94], TransCG [31], and ASGrasp [96], across scenes involving both single transparent objects and complex multi object clutter. The input data is collected using both ZED and RealSense stereo cameras. On the ZED inputs, ClearGrasp [94] produces relatively smooth backgrounds but fails to preserve sharp boundaries at object edges, often leading to depth discontinuities around transparent surfaces. TransCG [31] suffers from pronounced geometric inconsistency, with fragmented object contours and depth distortions that are particularly severe under occlusions. ASGrasp [96] demonstrates partial success in simpler scenes but fails to reconstruct accurate geometry in cluttered setups, where depth collapse and indistinct object boundaries are frequent. In contrast, ClearDepth consistently generates depth maps that are structurally coherent and visually complete.

When test on RealSense images, where active infrared projection interferes with transparent surface imaging, ClearDepth continues to produce stable depth estimations, outperforming other methods that degrade under IR related noise. In Fig. 4.8, we further compare ClearDepth with Dex-NeRF [40], a neural radiance field based method. In a real world tabletop scene populated with various transparent materials and reference patterns, Dex-NeRF [40] suffers from blurry contours, ambiguous depth layering, and background noise, resulting in plausible but inaccurate geometry. In contrast, ClearDepth captures sharp object boundaries and reconstructs clean background sur-

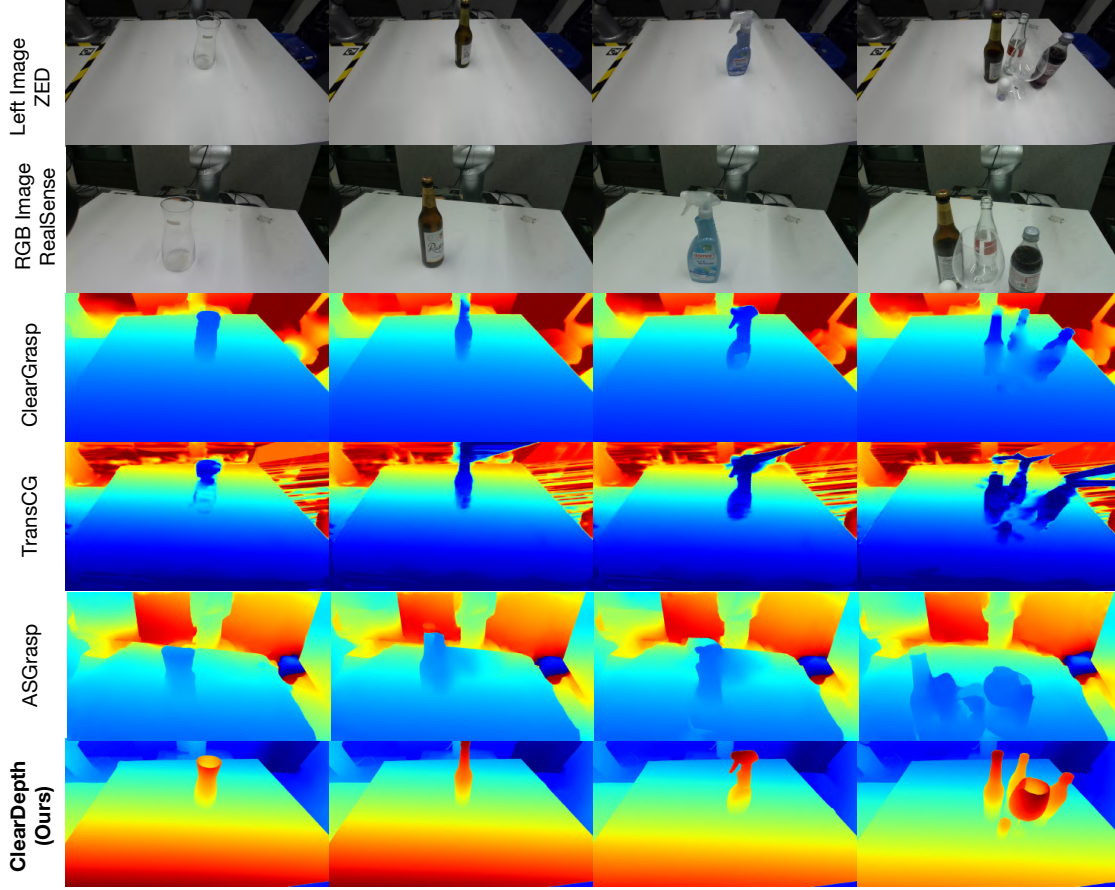


Figure 4.7: Comparison with ClearGrasp [94], TransCG [31], and ASGrasp [96].

faces, highlighting the limitations of NeRF based volumetric rendering for transparent materials.

To evaluate lighting robustness, we designed an experiment under three illumination settings: low light, normal indoor light, and strong light with glare. As shown in Fig. 4.9, ClearDepth maintains object recognizability and depth consistency across all conditions, even when specular highlights or low contrast degrade most methods. These results validate ClearDepth’s robustness and generalization under environmental uncertainties.

4.4 Robotic Experiments

To evaluate the performance of the proposed ClearDepth-based transparent object grasping pipeline, we conducted extensive real-world experiments and compared the results with state-of-the-art (SOTA) methods [31]. The experimental platform is shown in Fig. 4.10, where a Diana 7 robotic arm equipped with a Robotiq 85 two-finger gripper was used for grasp execution. A ZED2 camera was mounted for depth sensing in the robotic experiments, while additional cameras (RealSense D435 and ZED2) were employed for visual comparisons. A diverse set of transparent and opaque objects was used to evaluate both perception and grasping performance.

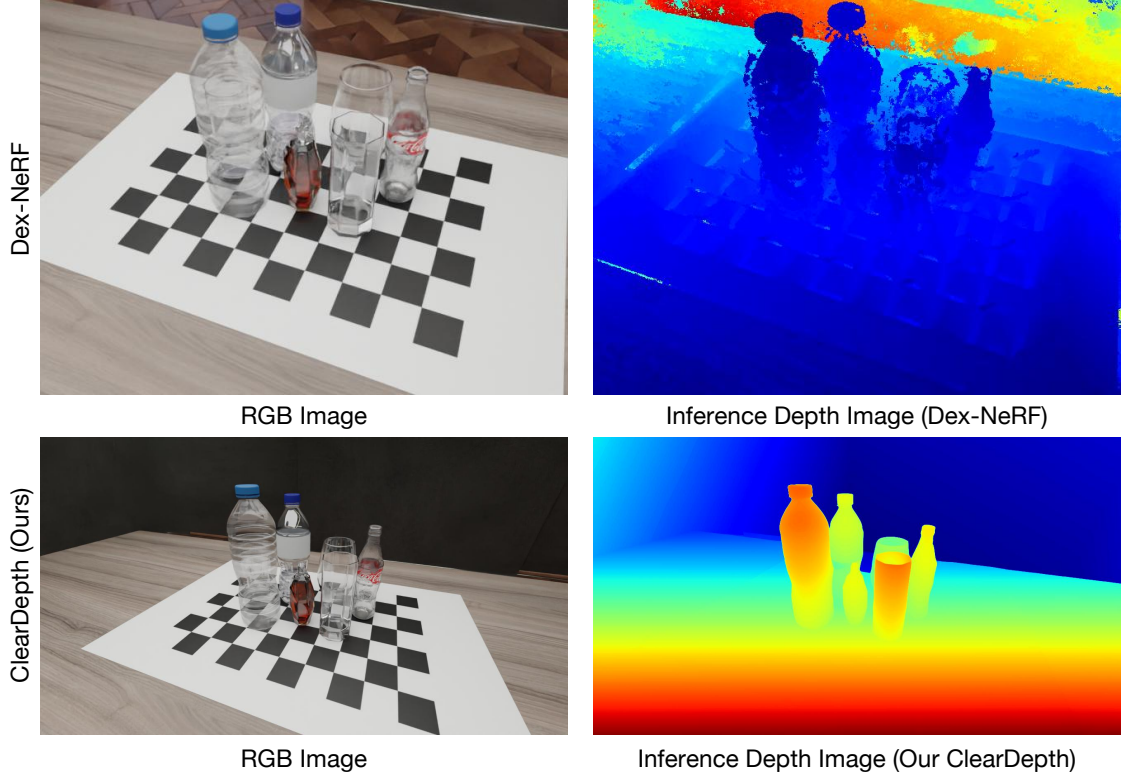


Figure 4.8: Comparison with Dex-NeRF [40].

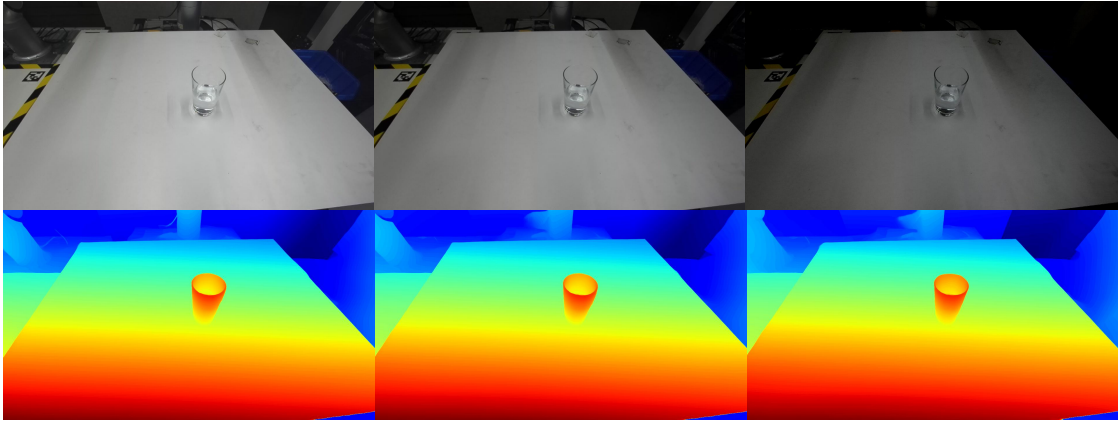


Figure 4.9: Performance under different lighting conditions.

As a baseline, we employed depth data obtained from the ZED2 camera for grasp generation. The evaluation covered both single-object grasping and cluttered environments. Specifically, Level-1 (L1) scenes contained a mixture of transparent and opaque objects, while Level-2 (L2) scenes consisted exclusively of fully transparent objects. For each setting, we performed 150 grasping trials. The grasp success rate was computed as the ratio of successful grasps to the total number of attempts. The quantitative results, summarized in Table 4.4, show that ClearDepth consistently outperforms all baseline methods, achieving the highest grasp success rates across different levels of scene

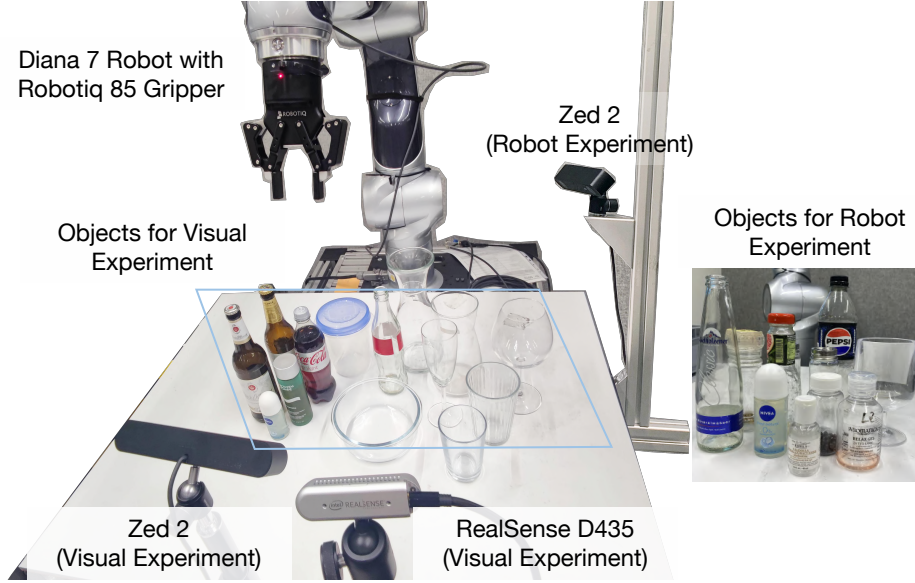


Figure 4.10: Experimental setup for transparent object grasping. The setup includes a Diana 7 robot with a Robotiq 85 gripper, a ZED2 camera for robotic experiments, and additional cameras (ZED2, RealSense D435) for visual experiments.

Table 4.4: Real-world robotic grasping comparison experiments with SOTA methods for transparent objects.

Grasp SR	single (L1)	cluttered (L1)	single (L2)	cluttered (L2)
ZED2 Camera	78%	63%	62%	58%
TransCG [31]	80%	70%	78%	67%
ClearDepth (ours)	98%	92%	98%	90%

and object complexity. Qualitative reconstruction and grasping results are presented in Fig. 4.11, where ClearDepth demonstrates accurate depth recovery, reliable point cloud reconstruction, and stable grasp candidate prediction, while baseline methods frequently fail in transparent regions.

To better understand the limitations, we conducted a detailed failure case analysis. Representative examples are illustrated in Fig. 4.12. The main cause of unsuccessful grasps was inaccurate depth reconstruction, which directly impacted grasp planning and execution. Two typical issues were identified: (1) missing reconstruction of transparent regions, leading to unexpected collisions between the gripper and the object; and (2) noisy depth points inside transparent regions, causing unreliable grasp candidate predictions. Additionally, occasional errors in grasp estimation could lead to finger-object collisions or unstable grasps, resulting in object drops. The distribution of failure causes across different methods (Fig. 4.13) clearly shows that ClearDepth substantially reduces failures caused by depth reconstruction errors, thereby improving overall grasp robustness.

To further validate the generality of our approach, we extended the experiments to

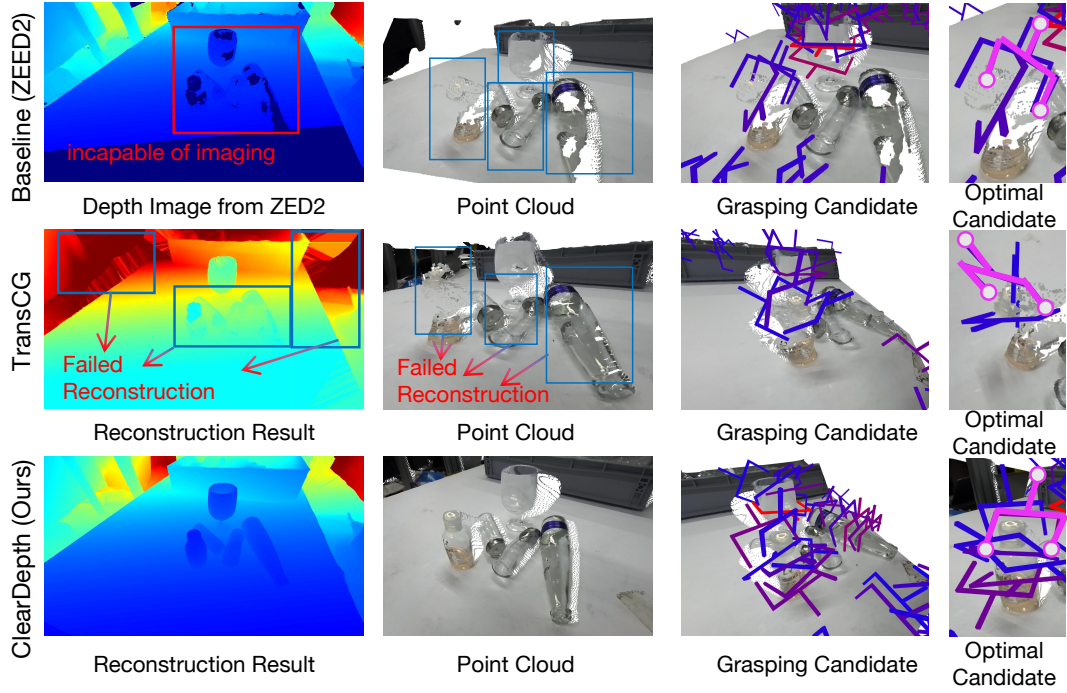


Figure 4.11: Qualitative comparison of transparent object grasping using depth reconstruction from ZED2, TransCG [31], and our ClearDepth. Grasp candidates are generated using GraspNet-Baseline [30].

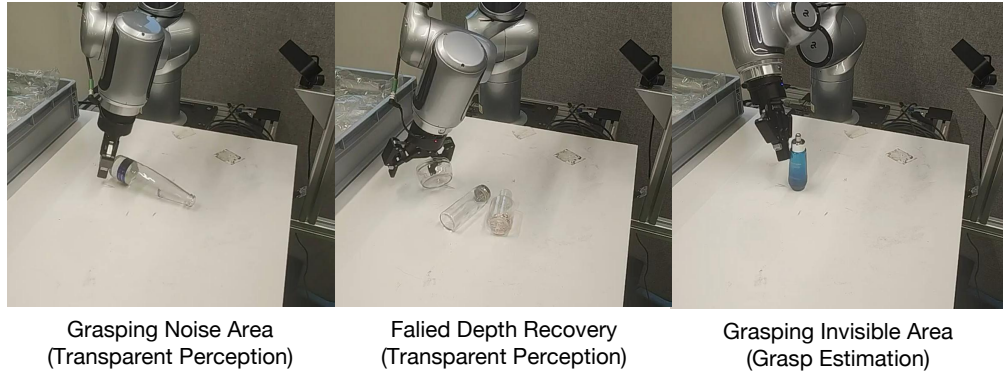


Figure 4.12: Representative examples of grasping failures, including (left) noisy grasp areas caused by transparent perception, (middle) failed depth recovery, and (right) grasping in invisible regions due to estimation errors.

multi-fingered robotic grasping. A robotic arm equipped with a dexterous multi-fingered robotic hand executed grasps planned by ClearDepth, following the grasp generation pipeline of ContactDexNet [9]. In these experiments, ClearDepth achieved an average grasp success rate of 86.2%, demonstrating that our method is not only effective for parallel-jaw grasping but also generalizes well to dexterous manipulation tasks.

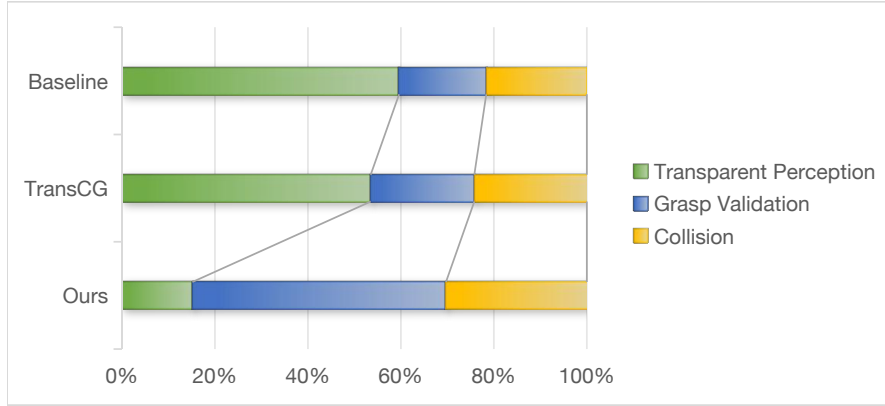


Figure 4.13: Error distribution of baseline method, TransCG [31] and our ClearDepth. We compare the proportions of total failures represented by different failure types.

4.5 Discussion and Future Work

Although this work demonstrates promising results in stereo reconstruction for transparent objects, several limitations remain. First, the diversity and coverage of publicly available stereo datasets in the open-source community are still insufficient, particularly for specialized scenarios such as industrial bin-picking, reflective materials, and transparent objects with mutual occlusion. This lack of representative data hinders the generalization of stereo imaging models, especially in zero-shot settings. Therefore, continuously expanding high-quality and task-specific datasets tailored to industrial and robotic applications will be of great long-term value.

Second, current high-fidelity simulation data generation still relies heavily on traditional offline renderers, which are often slow and computationally expensive. Future work should explore high-performance real-time rendering techniques, especially those leveraging AI-based acceleration to boost rendering efficiency and visual realism.

Moreover, robotic systems often operate under strict real-time constraints, which limits the practicality of large and computationally heavy models. Reducing model size and maintaining inference speed are therefore crucial. Techniques such as model distillation, architectural compression, and engineering optimization will be important to ensure deployability without sacrificing performance.

Lastly, real-world robotic applications inevitably expose models to unforeseen corner cases, making rapid model adaptation essential. Building a lightweight and resource-efficient training pipeline that enables fast updates and low-cost retraining is key to ensuring long-term robustness and adaptability in dynamic deployment environments.

Chapter 5

Zero-Shot Stereo Perception for Robotic Manipulation

5.1 Research Motivation and Challenges for Zero-Shot Stereo Perception

Zero-shot stereo imaging aims to generalize depth perception models to previously unseen scenes, object categories, and environmental conditions without additional fine-tuning or task-specific annotations. While recent vision foundation models have demonstrated strong zero-shot capabilities in tasks such as semantic segmentation and monocular depth estimation, extending these properties to stereo perception remains an open and non-trivial challenge. Compared to monocular setups, stereo systems inherently require precise geometric reasoning, reliable correspondence matching, and accurate metric scale recovery. These requirements cannot be fully satisfied by monocular priors alone and are particularly sensitive to domain shifts in appearance, lighting, and scene composition. For robotic manipulation, stereo perception further serves as a critical upstream component, where depth errors directly propagate to grasp planning and execution. Consequently, a zero-shot stereo perception system must maintain stable performance across a wide range of object materials, including both transparent and opaque surfaces, as well as under diverse lighting conditions and cluttered environments. To address these challenges, this work develops a zero-shot stereo perception framework that explicitly prioritizes generalization preservation. Instead of relying on extensive task-specific fine-tuning, we introduce minimal, low-rank adaptations to pre-trained foundation models, enabling the system to adapt to stereo geometry while retaining broad domain coverage. This design philosophy balances adaptability and stability, making the proposed framework particularly suitable for deployment in real-world robotic manipulation scenarios.

5.1.1 Limitations of Task-Specific Adaptation in Zero-Shot Stereo Perception

Stereo depth estimation relies on the assumption of photometric consistency between corresponding pixels across views. Reflective and refractive surfaces violate this assumption by introducing view-dependent appearance changes, background distortion, and light bending effects, which often lead to incorrect correspondences and unstable depth estimates. Although our previous work, ClearDepth, demonstrates robust stereo reconstruction for transparent objects by explicitly modeling such effects, it requires fine-tuning on a dedicated transparent object dataset. As a result, its performance is optimized for a narrow subset of scenes and may degrade when deployed in environments that differ from the training distribution. This specialization limits its applicability in open-world robotic scenarios where object materials and configurations are highly diverse. A truly zero-shot stereo perception system must therefore address these optical failure modes without relying on dataset-specific tuning. Instead, it should exhibit uniform depth recovery across reflective, refractive, and diffuse surfaces, while maintaining consistent performance on standard opaque objects. This requirement motivates a representation-preserving adaptation strategy that enhances robustness to optical distortions without sacrificing generalization.

5.1.2 Generalization Gap and Catastrophic Forgetting

Fine-tuning foundation models for downstream tasks often introduces a significant generalization gap. Post-training datasets are typically orders of magnitude smaller and less diverse than the data used during large-scale pre-training, leading to biased adaptations that favor narrow distributions. This imbalance frequently results in catastrophic forgetting, where task-specific optimization degrades performance on previously learned domains. In robotic manipulation, this issue is particularly severe, as real-world environments contain numerous corner cases, such as rare object geometries, unconventional lighting, and partial occlusions, which are underrepresented in public datasets. Overfitting to limited task data can therefore reduce robustness in precisely those scenarios where reliability is most critical. Moreover, continually collecting new corner-case data and re-training large models is computationally expensive and impractical for deployed robotic systems. These constraints motivate the need for adaptation strategies that minimize parameter updates, preserve pre-trained knowledge, and enable efficient model evolution without repeated full-scale retraining.

5.1.3 Affordance-Guided Multi-Finger Grasp Strategies

Human grasping behavior is inherently goal-oriented, with contact regions selected based on intended interaction rather than geometry alone. Such intent-driven reasoning enables flexible adaptation to novel objects and tasks without explicit prior knowledge of object categories. Inspired by this principle, our framework integrates affordance detection into the grasp planning pipeline to enhance zero-shot adaptability. By

predicting pixel-wise affordance maps for a five-finger dexterous hand, the system associates regions of the perceived scene with functional interaction semantics. These affordance cues are then used to filter and prioritize candidate grasps that align with the intended task, such as handle-oriented grasps for pouring or surface grasps for cleaning. This affordance-guided mechanism enables diverse and context-aware multi-finger grasp strategies without requiring object-specific or task-specific training. By operating at an intent-level abstraction, the approach naturally supports generalization to unseen objects and contributes to robust zero-shot manipulation performance.

5.1.4 Contributions and Paper Organization

We present StereoAnything, a zero-shot stereo perception framework for multi-finger robotic manipulation that preserves pre-trained generalization by constraining task adaptation to a low-rank parameter space, thereby mitigating catastrophic forgetting during post-pretraining adaptation. The main contributions of this work are summarized as follows:

1. A zero-shot stereo depth network built via LoRA-based adaptation of a monocular depth estimation foundation model, retaining broad domain coverage with minimal additional resources;
2. An affordance-guided grasp planner that dynamically selects five-finger strategies aligned with human intent;
3. A large-scale synthetic stereo dataset of transparent objects for training and evaluation;
4. Extensive quantitative and qualitative validation on public benchmarks and real robots, demonstrating consistent zero-shot performance across diverse scenes.

5.2 Foundation Models and Zero-Shot Adaptation

Foundation models have revolutionized computer vision by pre-training massive image encoders on diverse, large-scale datasets and then applying them zero-shot to downstream tasks. For instance, the Segment Anything Model (SAM) [54] was trained on the SA-1B dataset (11 million images, 1 billion masks) to deliver prompt-driven segmentation across unseen domains, while Depth Anything v2 [131] uses a DINOv2-Giant backbone trained on 62 million images to generate pseudo-depth labels for monocular depth estimation. However, both models incur prohibitive costs in terms of GPU-hours, storage, and financial resources for pre-training and fine-tuning.

Several adaptation techniques have emerged to reduce this overhead. Feature-freezing approaches keep the backbone fixed and train only lightweight heads or decoders [137, 136, 18, 47, 34], which lowers GPU memory needs but often fails to incorporate task-specific geometric priors. Delta-tuning methods, such as LoRA [38], insert low-rank trainable adapters into transformer layers, enabling task shifts with only 1–5%

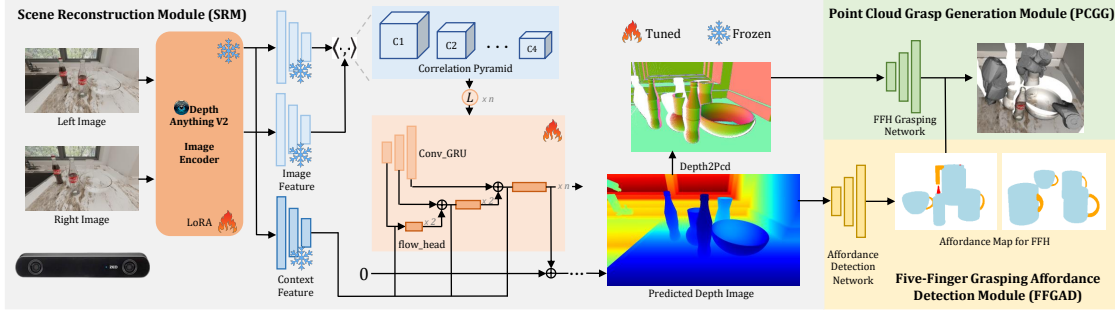


Figure 5.1: Overall architecture of StereoAnything.

of the original parameter count. Parallel strategies add small CNN branches alongside the frozen encoder [134], [28, 19, 50, 109, 12], or employ token-level adapters within the vision transformer [121]. For example, Q-Former [144] couples a frozen SAM encoder with a few learnable query tokens for few-shot segmentation.

Despite these advances, all existing adaptations focus on monocular tasks such as classification, segmentation, or single-view depth, and none address zero-shot stereo depth estimation, which requires precise geometric reasoning and multi-view fusion. Moreover, naïve fine-tuning of any large foundation component on limited stereo data leads to catastrophic forgetting, erasing the model’s original zero-shot capabilities on other domains. Our work extends delta-tuning into the stereo domain: by freezing the DINOv2-Large encoder and inserting LoRA adapters only in high-level attention layers, we achieve zero-shot stereo depth recovery with minimal extra parameters, while preserving the backbone’s broad pre-trained generalization and avoiding expensive re-training of the full model.

5.3 Method Overview of the Zero-Shot Stereo Grasping System

Fig. 5.1 illustrates the overall architecture of **StereoAnything**, which unifies three functional modules to enable zero-shot stereo-based multi-finger grasp detection.

(1) Stereo Reconstruction Module (SRM). This module processes synchronized left–right RGB pairs using a frozen foundation image encoder augmented with LoRA adapters. A GRU-based refinement loop with structural feature fusion produces a dense disparity map, which is converted to metric depth without requiring transparent-object-specific fine-tuning.

(2) Point Cloud Grasp Generation Module (PCGG). The reconstructed depth is transformed into a scene point cloud, which is then consumed by the grasp generation network. This module samples and ranks collision-free grasp candidates for a five-finger dexterous hand.

(3) Five-Finger Grasping Affordance Detection Module (FFGAD). To align grasp planning with human intent and task context, this module predicts pixel-wise affordance maps. The affordance cues are used to filter and prioritize the grasp proposals

generated by PCGG.

By integrating these three components in a zero-shot manner, StereoAnything generalizes effectively across opaque, transparent, and reflective scenes without additional task-specific training.

5.3.1 Zero-Shot Stereo Reconstruction Module (SRM)

During training, we freeze the pre-trained image encoder of a monocular foundation model and inject trainable LoRA adapters into its attention layers. Instead of fine-tuning the entire encoder, we jointly train only these adapters using both public stereo benchmarks and our synthetic transparent object dataset. At inference, the full encoder (including LoRA layers) processes synchronized RGB pairs to produce multi-scale feature maps. A correlation pyramid is built via dot-product similarities, and a lightweight GRU iteratively updates a disparity map, fusing low-level structural features in each loop. Finally, we convert the disparity to metric depth using known camera intrinsics and baseline. This design retains broad generalization from pre-training while adapting to transparent and mixed-material scenes with minimal additional parameters.

5.3.2 Point Cloud Grasp Generation Module (PCGG)

The PCGG module takes the reconstructed point cloud as input and first predicts a semantic contact map via our CoSe-CVAE network, identifying potential fingertip contact regions on object surfaces. Grasp candidates are then generated by optimizing fingertip-to-contact distances under hand kinematic constraints. A PointNet++-based evaluator scores each candidate for collision-free, high-quality contact. Finally, we perform SE(3) interpolation-based trajectory planning for the five-finger hand. This pipeline produces a diverse set of feasible grasps without task-specific training.

5.3.3 Five-Finger Grasping Affordance Detection Module (FFGAD)

FFGAD learns to predict pixel-wise affordance labels (e.g., *Handle-Grasp*, *Support*, *Wrap-Grasp*) directly from depth images by training a SegFormer B3 model on synthetic depth-affordance pairs rendered from the objects with annotated affordances in the AffordPose dataset [43].

As shown in Fig. 5.2, the pipeline begins with the AffordPose object database, where each 3D model is assigned functional grasping affordances according to the FFH taxonomy. From these annotated models, synthetic depth maps are generated together with ground-truth affordance masks covering eight semantic regions: background, no-grasp, handle-grasp, lever, press, pull, support, twist, and wrap-grasp. The right side of the figure illustrates how FFGAD predicts affordance maps from depth inputs, effectively highlighting task-relevant regions such as handles, supporting surfaces, or wrapping areas.

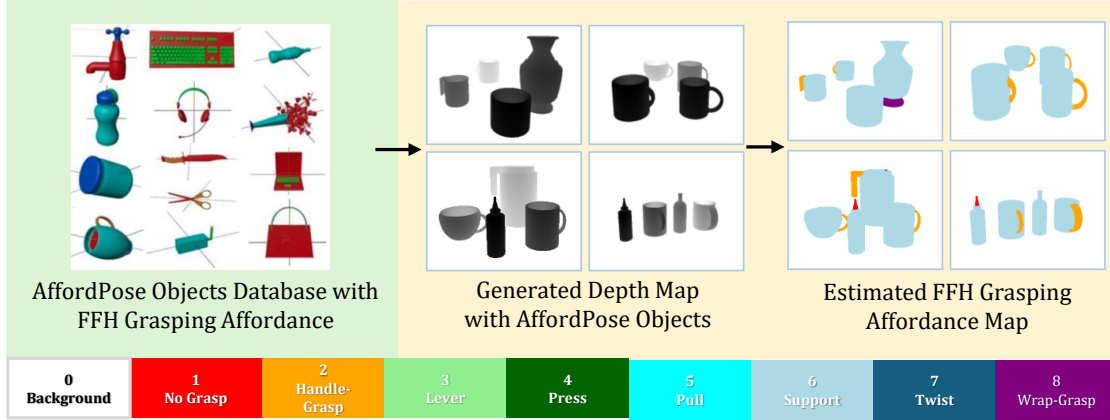


Figure 5.2: Pipeline of the Five-Finger Grasping Affordance Detection Module (FFGAD).

During training, each scene is composed of 2–4 randomly arranged objects; for every scene, we render the depth map and its corresponding affordance mask to form paired supervision. The SegFormer B3 backbone learns to associate structural depth cues with affordance semantics across these diverse object configurations. At inference, FFGAD takes as input the depth map produced by the Stereo Reconstruction Module (SRM), and predicts a dense affordance map in which each pixel is assigned to one of the predefined semantic regions.

Finally, we integrate this affordance prediction into grasp candidate selection. Specifically, the grasp poses generated by PCGG are revisited and any candidate whose fingertip contacts fall outside affordance regions consistent with the current task intent is discarded (e.g., only handle pixels are considered valid for pouring tasks). This affordance-based filtering introduces task-level semantics into multi-finger grasping, enabling the system to adapt to diverse manipulation goals without requiring per-object or per-task retraining.

5.4 Implementation Details

5.4.1 Extending Foundation Models with LoRA

To adapt large-scale vision foundation models to stereo depth estimation while preserving their zero-shot generalization capability, we adopt a parameter-efficient adaptation strategy based on Low-Rank Adaptation (LoRA). Rather than re-learning visual representations, the core objective of this design is to introduce minimal task-specific flexibility under a frozen pre-trained encoder. This avoids representation drift and catastrophic forgetting that often accompanies full fine-tuning, while improving robustness to challenging real-world scene distributions.

The image encoder from Depth Anything V2 [131] consists of a stack of Transformer blocks. For a given block, the self-attention operation is defined as

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \quad (5.1)$$

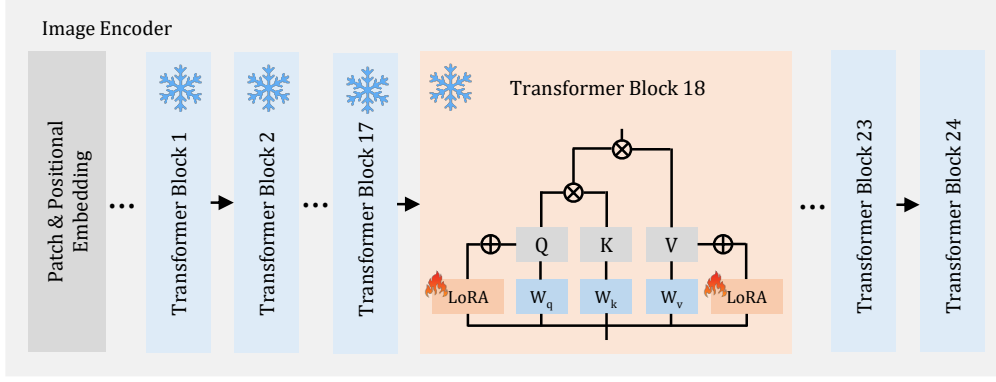


Figure 5.3: LoRA placement in encoder (transformer blocks 18–24).

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value matrices, and d denotes the feature dimension.

The query, key, and value matrices are obtained through linear projections of the input token features $\mathbf{X}$:

$$\begin{aligned}\mathbf{Q} &= \mathbf{X}\mathbf{W}_Q, \\ \mathbf{K} &= \mathbf{X}\mathbf{W}_K, \\ \mathbf{V} &= \mathbf{X}\mathbf{W}_V,\end{aligned}\tag{5.2}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ are the pre-trained full-rank projection matrices.

Instead of updating these full-rank matrices, LoRA parameterizes task-specific updates in a low-rank subspace. For a given projection matrix $\mathbf{W}$, the adapted weight is defined as

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W},\tag{5.3}$$

with

$$\Delta\mathbf{W} = \mathbf{B}\mathbf{A},\tag{5.4}$$

where $\mathbf{A} \in \mathbb{R}^{r \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times r}$ are trainable low-rank matrices and $r \ll d$ denotes the adaptation rank. During training, the original weight matrix $\mathbf{W}$ remains frozen, and only $\mathbf{A}$ and $\mathbf{B}$ are optimized.

Under this formulation, the number of trainable parameters introduced by LoRA scales as $\mathcal{O}(2rd)$ per projection matrix, compared to $\mathcal{O}(d^2)$ for full fine-tuning. This enables efficient adaptation while largely preserving the semantic and geometric priors learned during large-scale pre-training, thereby maintaining strong zero-shot generalization.

As illustrated in Fig. 5.3, we integrate LoRA modules into the self-attention layers of Transformer blocks 18 through 24. This design follows a hierarchical representation perspective. Deeper layers primarily encode high-level semantic context and long-range geometric relationships that are directly relevant to stereo correspondence and depth reasoning, making them suitable locations for limited task adaptation. In contrast, earlier layers focus on low-level visual primitives such as edges and textures, which are typically domain-invariant and therefore remain fully frozen to preserve broad generalization.

Let $\mathcal{L}_{\text{stereo}}$ denote the stereo depth supervision loss computed from ground-truth disparity or depth. The overall training objective optimizes only the LoRA parameters:

$$\min_{\{\mathbf{A}, \mathbf{B}\}} \mathcal{L}_{\text{stereo}}, \quad (5.5)$$

while all original encoder parameters remain fixed. This constrained optimization encourages the model to adapt its high-level representations for stereo depth estimation without disrupting the general visual knowledge inherited from foundation model pre-training.

By restricting adaptation to a small subset of late-stage attention layers and parameterizing updates in a low-rank form, the proposed strategy achieves a balance between task adaptability and representational stability. This allows the model to improve performance on challenging real-world scenes while retaining the encoder’s zero-shot generalization ability.

5.4.2 Synthetic Transparent Object Dataset Generation

Although Depth Anything V2 demonstrates strong zero-shot performance in general monocular depth estimation, we observe that its predictions deteriorate significantly in scenes involving transparent objects, particularly in robotic manipulation settings. This limitation arises not from the model architecture or encoder itself but from the scarcity of transparent objects in existing monocular depth training datasets. To retain the foundation model’s zero-shot generalization in stereo vision while improving its performance on this challenging corner case, we build upon the synthetic generation techniques used in ClearDepth and construct a more diverse and comprehensive dataset focused on transparent scenes. This dataset notably expands the variety of object materials, background complexity, and the number of transparent objects per scene, thus better covering the scenarios frequently encountered in real-world robot perception.

To address the scarcity of labeled stereo datasets for transparent object depth estimation, we generate a synthetic dataset using a physically accurate rendering pipeline. Transparent objects with varying materials (glass, plastic, frosted surfaces) are placed in procedurally generated scenes using domain randomization. We render left–right stereo pairs with perfect depth ground truth using ray-tracing, simulating real-world camera intrinsics and baseline setups. The dataset includes over 100,000 stereo pairs, covering a wide range of lighting, backgrounds, and object geometries, facilitating supervised learning under controlled yet diverse conditions.

This dataset is designed as an extension and augmentation of the ClearDepth dataset, incorporating significantly richer object diversity, more complex backgrounds, and a higher proportion of multi-object scenes, which are critical for robust model generalization and real-world applicability. In particular, it is constructed to explicitly complement scenarios where ClearDepth exhibits suboptimal performance, such as densely cluttered scenes and complex transparent object interactions. Moreover, the dataset generation process is guided by representative failure cases and annotation limitations observed in existing open-source datasets, enabling the synthesis of high-quality, densely labeled stereo data that better aligns with the demands of real-world robotic perception. Fig. 5.4

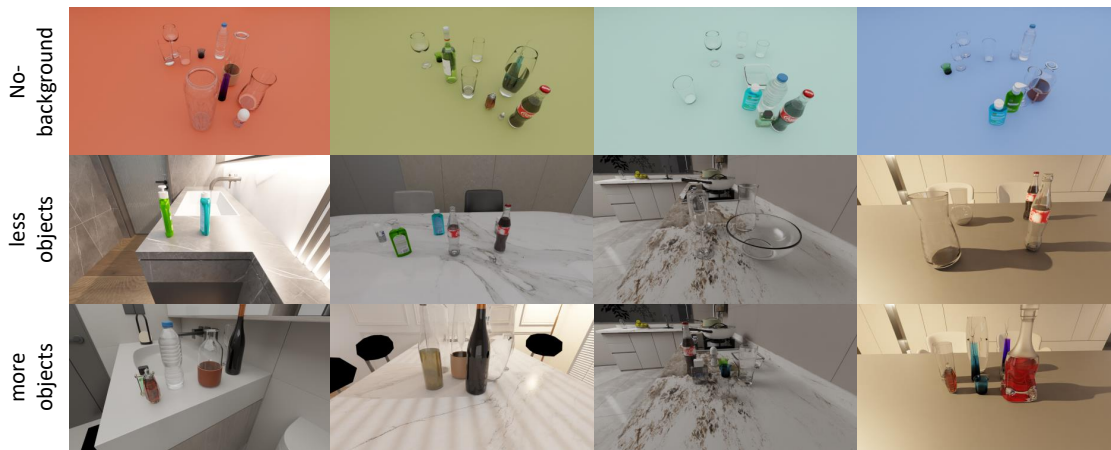


Figure 5.4: Examples of synthetic transparent object training data used in this work.

illustrates representative examples of the synthetic transparent object training data used in this work. The dataset is constructed to cover multiple challenging transparent object configurations, including scenes where multiple transparent objects are densely arranged on a tabletop, scenes with a small number of transparent objects randomly placed in indoor environments, and more complex indoor scenes containing multiple transparent objects with severe occlusion or overlap. These diverse scene types are deliberately included to complement the limitations observed in existing datasets such as ClearDepth and to better reflect the visual complexity encountered in real-world robotic perception tasks.

Furthermore, we introduce this transparent-object-focused dataset deliberately during training because the frozen Depth Anything V2 encoder exhibits notably poor performance in transparent object scenes. Since the pretraining of such foundation models lacks sufficient representation of transparency and refraction effects, our additional dataset helps compensate for this domain gap and improves depth prediction accuracy under these challenging conditions.

5.4.3 Affordance Detection and FFH Grasp Sample Generation

To validate the downstream applicability of our stereo depth prediction system in robotic manipulation, we introduce an affordance-guided grasp filtering pipeline for a five-finger dexterous hand (FFH). A SegFormer-based affordance segmentation network is employed to predict pixel-wise interaction categories, including Support, Handle-Grasp, and Wrap-Grasp, from the estimated depth map. These affordance predictions are further projected onto the three-dimensional point cloud reconstructed from stereo depth, enabling the association of semantic interaction information with object surfaces in 3D space.

For each detected object, the grasp generation module produces a diverse set of candidate grasps that cover multiple grasp types and finger configurations. Each candidate grasp corresponds to a specific FFH pose and induces a unique set of finger-to-object contact points in three-dimensional space. During inference, a filtering process is ap-

plied to evaluate whether the predicted contact points of each grasp lie within the corresponding affordance regions on the object surface and whether the intended grasp type is consistent with the predicted interaction semantics. Only grasp candidates that are both geometrically feasible and semantically compatible with the affordance predictions are retained.

This affordance-aware filtering strategy improves the relevance and success rate of downstream grasp execution by discarding grasps that are physically valid but semantically inappropriate for the intended interaction. The remaining grasp candidates are subsequently evaluated in physics-based simulation to verify reachability, contact stability, and overall task success.

5.5 Experimental Setup and Evaluation Protocol

5.5.1 Dataset and Training Configuration

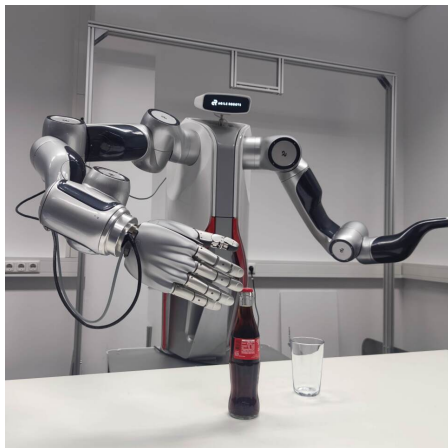
Our stereo depth estimation network is trained end-to-end using a combination of two public benchmarks (CREStereo and Scene Flow) and our synthetic transparent object dataset (no-background tabletop, small-object indoor, and complex occluded indoor scenes). CREStereo supplies over 40,000 ShapeNet-derived models rendered with diverse textures and real-image backgrounds; Scene Flow contributes 35,454 training pairs (FlyingThings3D, Monkaa, Driving final/clean) plus 4,370 test pairs with dense ground-truth disparity. For transparent scenarios, we created three validation splits focused on transparent objects, reflective materials, and general indoor scenes, and additionally collected a small real-world transparent/reflection test set.

Training is implemented in PyTorch on a single NVIDIA A6000 GPU. We freeze the DINOv2-Tiny encoder from Depth Anything V2 distilled student model (augmented with LoRA adapters in the last attention blocks) and optimize only the disparity decoder and adapter parameters. Optimization uses AdamW with gradient clipping to $[-1, 1]$, a one-cycle learning rate schedule peaking at 2×10^{-4} (100 warm-up steps), and random 320×720 crops with standard stereo augmentations. The model is trained for 400,000 iterations at batch size 8, with 22 GRU refinement steps per training pass and 32 during validation. In the comparative experiments, the fully fine-tuned version of the model was trained on two H100 GPUs due to its high computational cost.

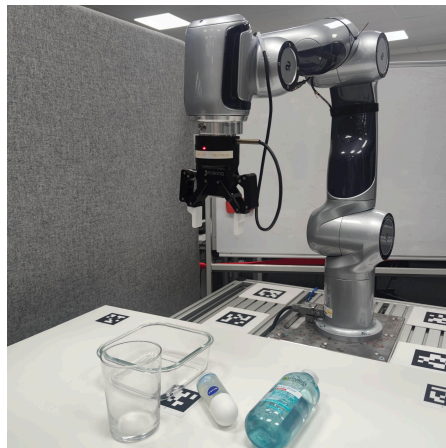
For affordance detection, we render 50 object meshes (5 categories $\times$ 10 each) from AffordPose in scenes of 2–4 randomly placed items, producing depth–mask pairs. A SegFormer B3 is trained with AdamW ($LR = 1.5 \times 10^{-4}$, batch 10, polynomial decay), cross-entropy loss, and resize/flip/distort augmentations.

5.5.2 Robotic Grasping Evaluation Setup

As shown in Fig. 5.5, we assess grasping performance on two robotic configurations: (1) an Agile Arnie-dual humanoid equipped with a DLR-HIT five-finger hand for service tasks, and (2) a logistic sorting station with a Robotiq 85 antipodal gripper for object handling. In the first scenario, ContactDexNet [140] generates multi-finger can-



(a) Humanoid with DLR-HIT hand



(b) Robotiq 85 antipodal gripper

Figure 5.5: Robotic configurations for grasping evaluation.

didate poses from the reconstructed point cloud, filters them via the affordance map, and executes trajectories via $SE(3)$ interpolation. In the second, GQCNN 4.0 [71] plans antipodal grasps on transparent items.

For both setups, we design 20 distinct tabletop scenes, each containing 1–5 randomly placed objects. Each scene undergoes 150 grasp trials, and we report success rate as the ratio of successful lifts to total attempts. Failure cases are analyzed for collision, fingertip slippage, or lack of force-closure.

5.6 Results and Analysis

5.6.1 Comparison with SOTA Stereo Models

We evaluated StereoAnything against SOTA stereo matching methods, including Selective-IGEV [113], DLNR [142], IGEV-Stereo [125], and ClearDepth [3], across three validation sets: transparent, reflective, and indoor scenes. The quantitative results are summarized in Table 5.1. Although ClearDepth, which is fine-tuned on transparent objects, achieved the lowest disparity errors on the transparent set, StereoAnything achieved comparable performance without any object-specific fine-tuning. On reflective and indoor scenes, StereoAnything outperformed all baselines in both the bad-1.0 and RMS metrics, demonstrating superior zero-shot generalization across diverse materials and environments. As shown in Fig. 5.6, our method produces sharper disparity maps in occluded regions and significantly reduces mismatches near specular highlights.

5.6.2 Baseline: Depth Anything V2 vs. StereoAnything

To evaluate the limitations of Depth Anything V2 and the effectiveness of our StereoAnything model in handling transparent object scenes, we conducted experiments on real-world data involving transparent materials. As shown in Fig. 5.7, we compared

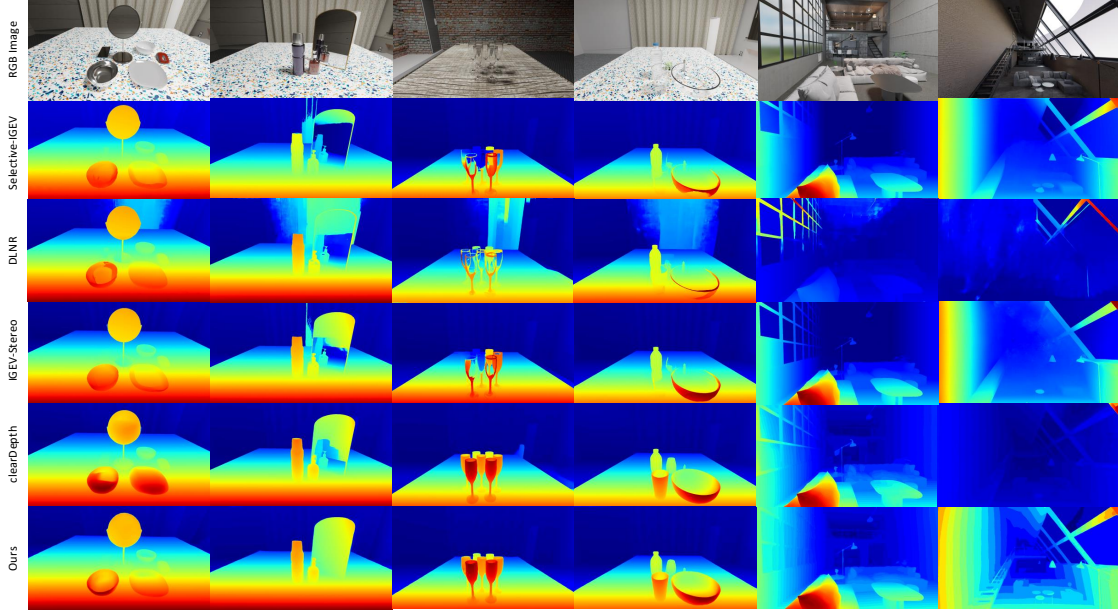


Figure 5.6: Qualitative comparison with SOTA stereo models. [5]

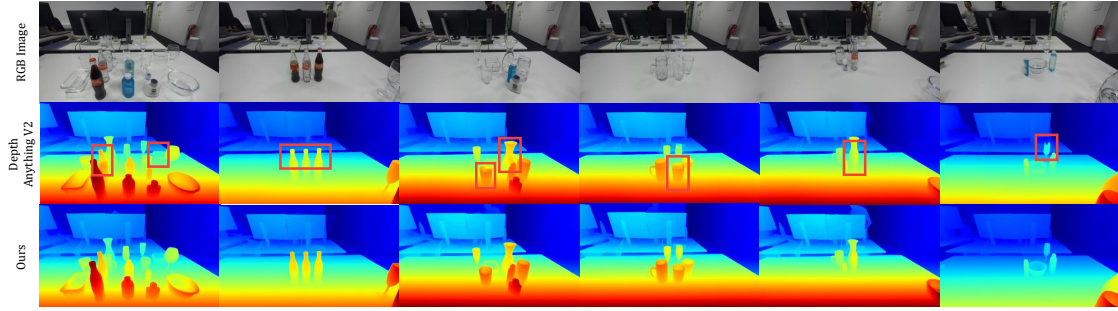


Figure 5.7: Comparison between monocular depth predictions from Depth Anything V2 and stereo depth outputs from StereoAnything on transparent object scenes.

the monocular depth predictions from Depth Anything V2 with the stereo depth outputs from StereoAnything. In scenes containing transparent objects, Depth Anything V2 often produced incorrect depth estimates behind refractive surfaces, for example predicting background depth as if there were no foreground obstruction. In contrast, StereoAnything generated more accurate and continuous depth maps, even in cases of severe occlusion and complex refraction.

These results demonstrate the improvements brought by our model over monocular depth estimation baselines in challenging scenes. First, we inject geometric priors into the stereo network via LoRA adapters, which enhances the model’s ability to handle the depth ambiguities caused by transparency. Second, we introduce a large-scale synthetic dataset focused on transparent objects, which significantly improves supervision in difficult regions. Although acquiring real-world stereo data with ground truth depth is challenging, our results show that by combining a monocular foundation model with geometry-aware adaptation and targeted data augmentation, it is possible to achieve

Table 5.1: Quantitative comparison with state-of-the-art algorithms across different scenarios

transparent object scenes						
Methods	AvgErr	RMS	bad 0.5	bad 1.0	bad 2.0	bad 4.0
Selective-IGEV [72]	2.2475	9.5405	47.3043	22.5444	9.8186	6.3994
DLNR [73]	3.8759	12.0645	62.8922	29.0394	15.4108	10.8462
IGEV-Stereo [74]	2.3890	10.1936	46.6069	18.5314	9.2266	6.5963
ClearDepth [20]	2.1380	8.7282	24.7329	16.3178	9.8459	5.7600
Ours	0.9945	3.5853	28.1808	16.5258	10.5957	5.5717
reflective object scenes						
Methods	AvgErr	RMS	bad 0.5	bad 1.0	bad 2.0	bad 4.0
Selective-IGEV [72]	4.1978	14.7176	55.8128	30.4737	17.0952	10.9580
DLNR [73]	6.4431	20.6239	69.8387	33.9513	17.7701	15.3820
IGEV-Stereo [74]	3.8076	13.9845	54.8323	24.7586	11.6816	8.8984
ClearDepth [20]	4.3041	16.2131	18.4664	14.8962	11.7792	9.4394
Ours	1.3571	5.5539	30.5030	16.9322	9.9297	5.3029
indoor scenes						
Methods	AvgErr	RMS	bad 0.5	bad 1.0	bad 2.0	bad 4.0
Selective-IGEV [72]	1.2936	2.6167	37.1819	28.9914	21.2650	9.7053
DLNR [73]	17.8353	33.0150	72.7614	67.7402	63.2795	57.0967
IGEV-Stereo [74]	1.4266	2.8212	41.2824	32.5755	23.0272	9.5211
ClearDepth [20]	1.3186	4.3911	27.6281	19.0965	11.8810	5.6306
Ours	1.0364	2.3879	23.5261	13.6087	10.7121	8.3703

Table 5.2: Comparison with other foundation model image encoders on transparent object scenes.

Encoder	AvgErr	RMS	bad 0.5	bad 1.0	bad 2.0	bad 4.0
TinyViT (SAM)	18.3955	26.4897	96.5526	93.1536	86.4317	74.2145
RepViT (SAM)	3.6071	13.7799	51.6111	34.5033	19.8542	10.6834
DINOv2-ViT-L (Depth Anything V2)	0.9945	3.5853	28.1808	16.5258	10.5957	5.5717

robust zero-shot stereo depth estimation for transparent scenes, surpassing the performance of purely monocular methods.

5.6.3 Foundation-Model Encoder Extension Study

To investigate how vision foundation models trained for different purposes transfer to stereo depth estimation, we compared three frozen image encoders: DINOv2-Large from Depth Anything V2 (pretrained for monocular depth estimation), SAM-TinyViT, and SAM-RepViT (both pretrained for segmentation). During training, only the stereo decoder and GRU modules were updated. The quantitative results are summarized in Table 5.2. Under this frozen setting, TinyViT produced noisy depth maps with frequent

Table 5.3: Validation results of ablation study for training methods.

Method	Scenes	EPE	RMSE	bad 0.5	bad 1.0	bad 2.0	bad 4.0
Full-Tuning	Transparent	5.0106	10.8112	90.7635	81.7816	63.8780	35.1437
Frozen Encoder w/o LoRA	Transparent	1.0290	4.1215	30.5056	16.6968	10.6174	5.5933
Ours (Frozen Encoder + LoRA)	Transparent	0.9945	3.5853	28.1808	16.5258	10.5957	5.5717
Full-Tuning	Reflective	5.0446	10.3309	86.7491	74.8941	55.2792	33.2216
Frozen Encoder w/o LoRA	Reflective	1.4594	6.2511	34.1759	17.6190	9.9581	5.8059
Ours (Frozen Encoder + LoRA)	Reflective	1.3571	5.5539	30.5053	16.9322	9.9297	5.3029

mismatches, while RepViT, despite its cascaded design, failed to mitigate these errors and exhibited artifacts similar to those of TinyViT.

It is important to emphasize that these shortcomings mainly stem from a mismatch between the freezing strategy and the pretraining objective. TinyViT and RepViT are typically used in segmentation frameworks where the encoder is frozen or nearly frozen, making their features highly sensitive to semantic prompts and boundaries but less adaptable to geometric and disparity cues. When kept completely frozen in our stereo setting, they could not adapt to the requirements of depth reasoning. By contrast, DINOv2-Large, which had been extensively pretrained on monocular depth data and thus already encodes strong geometric priors, aligns much better with the stereo task even in a frozen state. This leads to significantly more accurate and artifact-free reconstructions, especially in transparent and mixed-material scenes.

In summary, under a frozen-encoder configuration, the performance differences are primarily determined by the compatibility between the pretraining objective and the freezing strategy. The weaker results of TinyViT and RepViT arise from their segmentation-oriented pretraining combined with the frozen transfer setting, rather than inherent architectural limitations. If allowed moderate adaptation during fine-tuning, segmentation backbones are expected to yield improved stereo depth performance.

5.6.4 Ablation Studies on Training Strategies and Iterative Inference Trade-offs

Our ablation study compared three training variants using the distilled Tiny encoder from Depth Anything V2. As shown in Table 5.3, full fine-tuning of this encoder in an end-to-end manner resulted in poor performance and clear catastrophic forgetting. This is likely due to the limited size and diversity of our stereo datasets as well as the reduced capacity of the encoder. Moreover, this variant required two H100 GPUs, making it impractical.

Freezing the encoder without LoRA preserved its pre-trained capabilities for opaque-object depth estimation and achieved only partial success on transparent objects, leaving overall performance slightly below our proposed approach. By contrast, freezing the encoder while injecting lightweight LoRA adapters enabled targeted updates of high-level features. This configuration maintained strong accuracy on opaque scenes, delivered significant improvements on transparent objects, and avoided catas-

trophic forgetting, all with minimal additional parameters.

We further examined the number of GRU refinement iterations. Increasing the iterations steadily improved disparity accuracy, but gains beyond twelve loops were marginal while inference time continued to rise. For robotic applications that demand real-time response, twelve GRU iterations offer the best compromise between accuracy and processing speed.

5.7 Conclusion and Discussion

This work introduced StereoAnything, a depth-aware perception framework that bridges foundation models with transparent object manipulation in robotic systems. By inserting LoRA-based geometric adapters into a frozen monocular encoder and training with a large-scale, physically accurate synthetic dataset, our approach enables robust zero-shot stereo depth estimation across opaque, transparent, and mixed-material scenes. The resulting disparity maps not only achieve competitive benchmark performance but also yield substantial improvements in downstream grasping tasks, including five-finger and antipodal strategies. These findings demonstrate that combining foundation model priors with geometry-aware adaptation is an effective pathway toward generalizable robotic perception.

Looking forward, several challenges remain. Current open-source stereo datasets are limited in scale and diversity, restricting progress toward complex industrial and service-robot scenarios. Sustainable model updates are another critical issue, as full fine-tuning of large vision encoders is computationally prohibitive; incremental and online learning strategies may provide a more efficient alternative. Finally, deploying such models on embedded and edge platforms requires balancing compactness and versatility. Distilling “micro-foundation” models that retain broad zero-shot capabilities while meeting real-time latency and power constraints will be key to long-term robotic deployment in the wild.

Chapter 6

Collision-aware Sim2Real Robotic Grasping of Fragile Objects

6.1 Research Motivation for Collision-aware Robotic Grasping

6.1.1 Challenges in Grasping Fragile Fruit in Stacked Scenes

Grasping fragile fruits in stacked scenes presents significant challenges arising from a combination of environmental complexity, object material properties, and stringent industrial requirements. In real-world packing and sorting facilities, fruits are typically densely arranged in bins or containers, leading to severe occlusions, frequent inter-object contacts, and limited free space for safe approach trajectories [132, 17]. Such cluttered configurations substantially increase the difficulty of reliable perception, collision-free motion planning, and stable grasp execution.

Beyond environmental factors, fragile fruits exhibit physical characteristics that further complicate robotic manipulation. Their irregular shapes, soft internal structures, and smooth or slippery skins make them highly sensitive to contact forces [135, 100]. Even minor deviations in grasp pose or contact location can cause bruising, surface scuffing, or internal damage, which may not be immediately visible but can accelerate decay and lead to significant product loss during storage and transportation. In commercial fruit handling, bruise-induced degradation is one of the primary contributors to economic loss, placing strict constraints on allowable contact forces and grasp stability.

Accurate perception plays a critical role in mitigating these risks. In stacked scenes, depth estimation uncertainty, reflective or semi-transparent packaging, and dynamic interactions between neighboring fruits often result in incomplete or noisy geometric observations [68]. These perception errors propagate directly into grasp planning and execution, increasing the likelihood of unintended collisions, unstable contacts, or failed picks. Moreover, the dynamic nature of fruit piles means that grasping one item can induce motion in surrounding objects, further complicating collision avoidance and force control.

As a result, grasping fragile fruits in stacked environments requires more than geo-

metrically feasible grasp synthesis. It demands collision-aware perception and planning strategies that explicitly reason about inter-object constraints, material fragility, and contact safety. Such systems must jointly account for cluttered scene geometry, uncertainty in visual sensing, and the delicate physical properties of the objects to achieve reliable, damage-free manipulation in real-world industrial settings.

6.1.2 Limitations of Existing Robotic Grasping Methods

Despite substantial progress in robotic grasping, existing methods remain insufficient for handling fragile fruits in stacked and cluttered environments. A key limitation lies in the mismatch between common grasping assumptions and the realities of agricultural scenes, where heavy occlusion, tight inter-object spacing, and sensing uncertainty are pervasive.

Early analytical and model-based grasping approaches rely on accurate object geometry and relatively clean observations, which are rarely available in stacked fruit scenarios. Occlusions and partial views often lead to unreliable depth estimates and incomplete object representations, causing grasp synthesis to fail when geometric assumptions are violated [59]. Moreover, such methods typically focus on the target object in isolation and struggle to reason about surrounding objects, frequently producing grasps that are locally feasible but unsafe in dense stacks.

Learning-based grasping methods improve robustness by directly predicting grasp poses from RGB-D images or point clouds, supported by large-scale datasets and benchmarks. However, most existing datasets and training protocols primarily emphasize grasp success on rigid objects and relatively uncluttered scenes [71, 72, 85, 30]. When deployed in stacked fruit environments, these models often lack sufficient exposure to the severe occlusion patterns and spatial constraints that characterize real agricultural settings, limiting their reliability under heavy clutter. In these existing grasping pipelines, collision-related constraints arising from neighboring objects are not sufficiently incorporated during grasp generation. Grasp candidates are commonly produced based on the local geometry of the target object, with limited consideration of how surrounding fruits restrict approach directions and execution feasibility. In densely stacked scenes, even minor unintended contacts during the approach phase can disturb nearby objects, leading to cascading failures or unstable outcomes. As a result, grasps that appear valid in isolation often prove brittle when executed in real stacked environments.

Overall, these limitations indicate that existing robotic grasping techniques do not adequately address the combined challenges of cluttered perception, inter-object constraints, and Sim2Real deployment in agricultural applications. This gap motivates collision-aware grasping frameworks that explicitly reason about stacked scene structure and robust perception under real-world conditions.

6.2 Sim2Real Grasping Framework Design

The proposed robotic grasping framework, illustrated in Fig. 6.1, is designed for robust manipulation of fragile fruits in cluttered and stacked environments, where collision

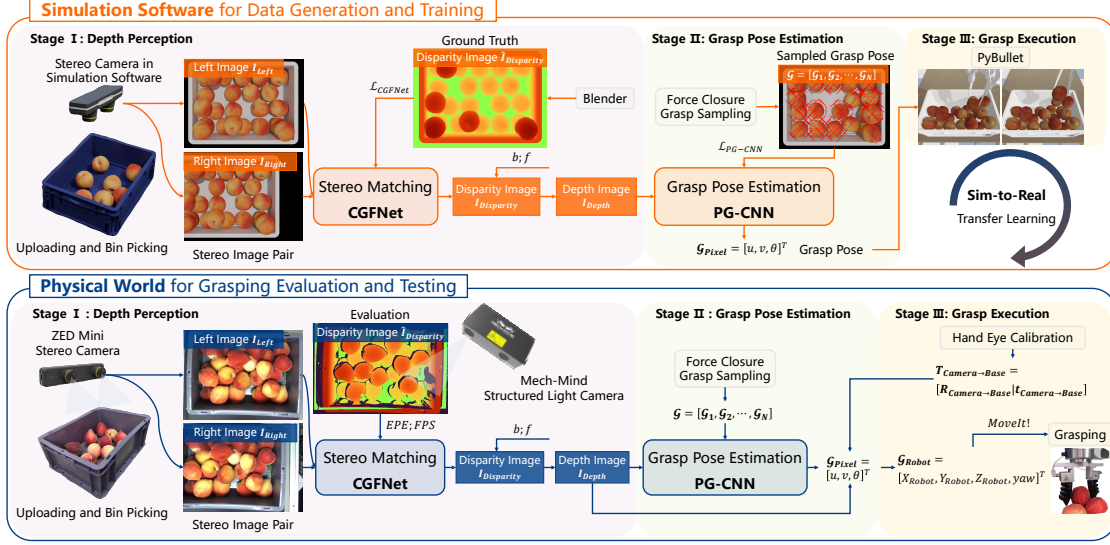


Figure 6.1: Overview of the proposed Sim2Real grasping framework for fragile fruit manipulation, including simulation-based training and real-world deployment.

avoidance is as critical as grasp success. To address these challenges while maintaining Sim2Real transferability, the framework adopts a modular three-stage pipeline that decouples perception, grasp reasoning, and execution.

In the first stage, stereo image pairs are captured and processed by CGFNet to reconstruct a dense depth map of the scene. Stereo-based depth perception is chosen to provide reliable geometric cues under varying surface appearances without relying on object segmentation. Based on local geometric features such as depth gradients and edges, a set of force-closure-based grasp candidates is generated to ensure physical feasibility.

In the second stage, the grasp candidates are evaluated by a neural grasp quality estimator (PG-CNN), which predicts a quality score for each candidate. During simulation-based training, each grasp is labeled not only by execution success but also by whether it results in collisions with neighboring fruits. This implicit supervision enables the network to learn collision-aware grasp strategies directly from data, without requiring explicit object models or collision checking at inference time.

In the final stage, the highest-scoring grasp is transformed from image coordinates to the robot frame using hand-eye calibration and executed through motion planning. By training the perception and grasp evaluation modules entirely in simulation and deploying them unchanged in the physical system, the proposed framework achieves effective Sim2Real transfer while ensuring safe and reliable grasp execution in complex real-world scenarios.

6.2.1 Grasp Pose Estimation Network: PG-CNN

To enable collision-aware grasp evaluation, this study proposes the PG-CNN (Peach Grasping Convolutional Neural Network). The network takes cropped regions from the

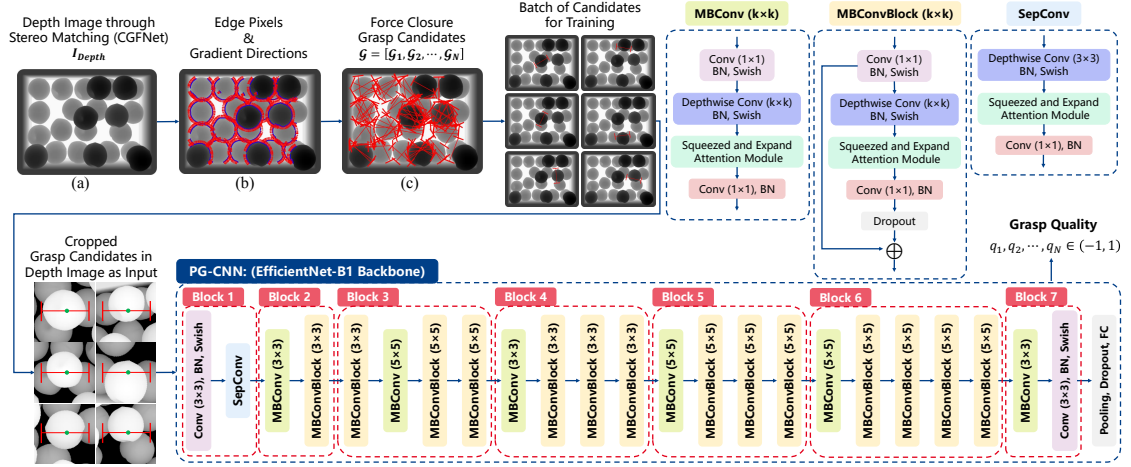


Figure 6.2: Overall architecture of the proposed PG-CNN.

depth map as input and outputs a grasp quality score $q_i \in (-1, 1)$ for each candidate pose to assist in final selection.

As shown in Fig. 6.2, the backbone of PG-CNN is based on EfficientNet-B1, consisting of 7 MBConvBlock modules with kernel sizes of 3×3 or 5×5 . These blocks are enhanced with Squeeze-and-Excitation attention and Swish activation to capture multi-scale features. The output is generated through pooling and fully connected layers.

Each grasp candidate $G_i = [u, v, \theta]^T$ is used to crop a fixed-size region (e.g., 200×200) from the depth map I_{Depth} . The region includes both the grasp point and its surrounding context to help the model perceive potential collisions.

Training data is generated in PyBullet, with each grasp labeled $\hat{q}_i \in (-1, 1)$. A successful grasp without collision is labeled $\hat{q}_i = 1$, while any failure or collision is labeled $\hat{q}_i = -1$. This allows the model to implicitly learn the relationship between grasp actions and environmental constraints.

To address class imbalance between positive and negative samples, we adopt a weighted cross-entropy loss function:

$$\mathcal{L} = - \sum_{i=1}^N \phi_m (\hat{q}_i \log q_i + (1 - \hat{q}_i) \log(1 - q_i)),$$

where N denotes the number of grasp candidates, and ϕ_m is a weight factor dynamically adjusted based on sample distribution.

During deployment, PG-CNN evaluates all grasp candidates and selects the one with the highest score $G_{Pixel} = [u, v, \theta]^T$ for execution. Due to its lightweight architecture, inference is completed within milliseconds, ensuring real-time performance.

6.2.2 Grasp Execution Pipeline

To avoid collisions in cluttered environments, we adopt a collision-aware grasp pose estimation network named PG-CNN. Grasp candidates $\mathcal{G} = \{G_1, G_2, \dots, G_N\}$ are sampled from the depth image I_{Depth} following the force closure principle.

First, I_{Depth} is preprocessed with normalization and filtering. Edge pixels p_i and their normals $\vec{n}_i$ are extracted to generate antipodal grasp pairs. The grasp angle α is calculated as:

$$\alpha = \arccos(\vec{n}_i \cdot \vec{n}_j), \quad (6.1)$$

where $\vec{n}_i$ and $\vec{n}_j$ denote the surface normals of the edge pixels. Pairs that satisfy the antipodal condition are retained as valid grasp candidates. Each grasp is defined as

$$G_i = [u, v, \theta]^\top, \quad (6.2)$$

where (u, v) is the image center of the grasp and θ the in-plane angle.

Each candidate region is cropped from the depth map and evaluated by PG-CNN, which assigns a grasp quality score $q_i \in (-1, 1)$. The network is trained using an adaptive weighted cross-entropy loss:

$$\mathcal{L} = - \sum_{i=1}^N \phi_m(\hat{q}_i \log q_i + (1 - \hat{q}_i) \log(1 - q_i)), \quad (6.3)$$

where $\hat{q}_i \in \{-1, 1\}$ is the ground-truth label obtained from PyBullet simulation, and ϕ_m is a dynamically adjusted weight to address label imbalance. During inference, the grasp with the highest score is selected as

$$G_{Pixel} = [u, v, \theta]^\top. \quad (6.4)$$

The selected grasp in pixel coordinates G_{Pixel} is transformed into the robot base frame pose G_{Robot} for execution. The hand-eye calibration yields a 4×4 transformation:

$$T_{\text{Camera} \rightarrow \text{Base}} = \begin{bmatrix} R_{\text{Camera} \rightarrow \text{Base}} & \mathbf{t}_{\text{Camera} \rightarrow \text{Base}} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad R \in SO(3), \quad \mathbf{t} \in \mathbb{R}^3. \quad (6.5)$$

The pixel (u, v) is back-projected to camera coordinates with depth $I_{Depth}(u, v)$:

$$\begin{bmatrix} X_{\text{Camera}} \\ Y_{\text{Camera}} \\ Z_{\text{Camera}} \end{bmatrix} = I_{Depth}(u, v) K_{\text{Camera}}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (6.6)$$

where K_{Camera} is the intrinsic matrix of the left stereo camera.

The 3D grasp point in camera frame G_{Camera} is mapped to the robot base frame:

$$\begin{bmatrix} X_{\text{Robot}} \\ Y_{\text{Robot}} \\ Z_{\text{Robot}} \\ 1 \end{bmatrix} = T_{\text{Camera} \rightarrow \text{Base}} \begin{bmatrix} X_{\text{Camera}} \\ Y_{\text{Camera}} \\ Z_{\text{Camera}} \\ 1 \end{bmatrix}, \quad (6.7)$$

and the in-plane orientation is assigned as

$$\text{yaw} = \theta. \quad (6.8)$$

Thus, the final grasp pose in the robot base frame is:

$$G_{\text{Robot}} = [X_{\text{Robot}}, Y_{\text{Robot}}, Z_{\text{Robot}}, \text{yaw}]^\top, \quad (6.9)$$

which is sent to MoveIt! for motion planning and trajectory execution with the Franka Panda arm.



(a) Synthetic stereo pair

(b) Real stereo pair

Figure 6.3: Stereo image examples from the PeachStereo dataset.

6.3 Sim2Real Dataset Construction

6.3.1 Synthetic Stereo Dataset: PeachStereo

To train the CGFNet depth estimation model, we construct the PeachStereo dataset using the Blender simulator, which is specifically designed to support robust Sim2Real transfer for stereo-based depth perception in cluttered fruit grasping scenarios. The dataset emulates real-world sensor configurations by rendering synchronized stereo image pairs along with their corresponding dense depth maps using two virtual cameras. All images are generated at a resolution of 960×540 , consistent with the real stereo camera setup used in the physical system. Fig. 6.3(a) shows a representative synthetic stereo image pair from the PeachStereo dataset, while Fig. 6.3(b) presents a corresponding real stereo image pair captured by the physical system. Both examples consist solely of left and right images.

To improve scene diversity and reduce overfitting to specific geometric arrangements, extensive randomization is applied during data generation. This includes variations in the number of fruits, spatial layouts, stacking patterns, degrees of occlusion, and illumination conditions. Such randomized scene configurations encourage the network to learn depth cues that are invariant to appearance changes and clutter complexity, which is critical for reliable deployment in real-world environments.

Each synthetic depth map I_{Depth} is precisely aligned with its corresponding stereo image pair (I_L, I_R) at the pixel level, forming supervised training triplets $(I_L, I_R, I_{\text{Depth}})$. The availability of accurate dense depth ground truth enables effective supervision of the CGFNet model and facilitates stable convergence during training.

6.3.2 Synthetic Grasping Dataset: PeachGrasp

PeachGrasp is a synthetic grasp annotation dataset constructed for training the PG-CNN network. Generated in the PyBullet simulated environment, it begins with sampling grasp candidates $G_i = [u, v, \theta]^T$ on each depth map I_{Depth} using edge-based heuristics and force closure filtering. Each candidate is then executed in PyBullet to determine grasp success and collision status.

Importantly, even if a grasp is mechanically successful, it is labeled as a negative sample if any collision occurs. This design encourages the model to learn implicit collision avoidance. The dataset samples are structured as $(I_{\text{Depth}}, G_i, \hat{q}_i)$, where $\hat{q}_i \in \{-1, 1\}$ indicates grasp feasibility and safety.

6.3.3 Transfer Learning Pipeline

To bridge the domain gap between simulation and reality, we adopt a pretraining–fine-tuning transfer learning strategy. The CGFNet is first pretrained on the Scene Flow dataset to capture generic stereo matching capabilities and then fine-tuned on PeachStereo to adapt to the structural characteristics of cluttered fruit scenes. The PG-CNN is trained from scratch using PeachGrasp, enabling it to learn to distinguish grasp safety and feasibility in complex environments.

At deployment, both models are used with frozen weights, eliminating the need for further fine-tuning on real data. This design ensures robust generalization even in the absence of large-scale real-world annotations.

It is worth noting that acquiring high-quality stereo depth data in real-world settings is extremely challenging. Obtaining ground-truth depth for every pixel with consumer-grade depth cameras in real scenes is practically infeasible. In contrast, the PeachStereo dataset provides densely aligned, error-controlled depth supervision at scale, which greatly alleviates this issue.

For grasp policy training, the cost of collecting negative samples in real environments is even more prohibitive. Most failed grasps, particularly those involving unintended collisions, can physically damage the fruit, leading to direct economic losses. It is therefore impractical to deliberately acquire such failure cases from the real world. Our method circumvents this limitation by simulating grasp executions and recording collision outcomes in a physics-based environment, enabling the collection of rich negative samples without any physical harm.

In summary, our approach, which combines synthetic datasets with transfer learning, not only addresses the difficulty of real data acquisition but also improves safety and generalization in real-world deployments, all without incurring additional annotation or material costs.

6.4 Experimental Evaluation

6.4.1 Collision-aware Grasp Pose Prediction in Simulation

To evaluate the performance of PG-CNN in simulation, we conducted both qualitative visualization and quantitative benchmarking. As shown in Fig. 6.4, PG-CNN was tested in both simulation and real-world-like setups. In Figs. 6.4(c) and 6.4(g), all sampled grasp candidates conformed to the force closure principle, indicating the physical feasibility of the sampling strategy.

The cropped depth images helped reduce background noise and allowed the model to focus on critical regions around the grasp target. In Figs. 6.4(d) and 6.4(h), the grasp pose with the highest predicted quality score was marked with a green check mark. These recommended poses showed effective collision avoidance in stacked fruit configurations. In contrast, red cross marks indicated grasp poses with lower scores, which were more likely to cause collision between the gripper and adjacent fruits. This qualitative analysis supports the model’s collision-awareness and validates the feasibility of our Sim-to-Real transfer strategy.

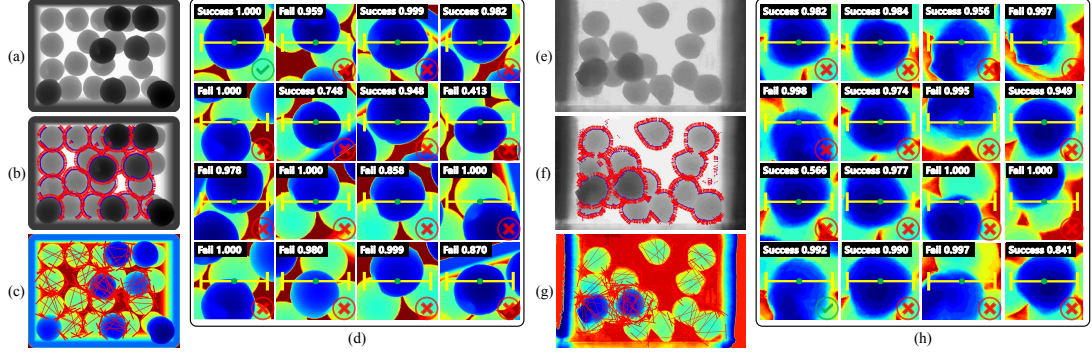


Figure 6.4: Visualization of sampled grasp candidates and PG-CNN predictions in simulation and real-world-like setups.

Table 6.1: Grasping success rate in the physical world with different number of peaches.

Number of Peaches	Attempts	Cause of Failure			Success	Rate
		Perception	Grasping	Slip		
1–5	70	0	1	3	66	94.29%
6–10	70	2	3	3	62	88.57%
11–15	70	3	2	5	60	85.71%
16–20	70	2	4	3	61	87.14%
Our Method	280	7	10	14	249	88.93%
Dex-Net 4.0 [71]	280	16	17	15	232	82.86%

For quantitative evaluation, we compared PG-CNN with three baselines: random sampling (RS), Dex-Net 4.0 (DN), and GraspNet (GN). Table III reports the grasp success rates under different quantities of fruit (ranging from 1 to 10 objects), where all methods were applied to the same grasp candidate pool.

PG-CNN consistently achieved the best or second-best performance across all test cases, with an average success rate of 93.45%. In comparison, Dex-Net 4.0 [71] and GraspNet [30] achieved 87.15% and 89.65%, while random sampling yielded only 35.05%. Notably, when the size of the PeachGrasp training dataset was halved, PG-CNN still maintained 93.15% success, highlighting its data efficiency and robustness.

As the number of objects increased, the task complexity rose due to tighter graspable spaces and higher collision risk. Dex-Net and GraspNet tended to suggest grasps that led to more frequent collisions, particularly in densely stacked scenes. In contrast, PG-CNN remained stable, suggesting that it had learned implicit representations linking grasp feasibility to environmental constraints.

In summary, PG-CNN achieves high prediction accuracy in simulation while demonstrating strong collision-awareness and adaptability, validating its suitability for real-world deployment.

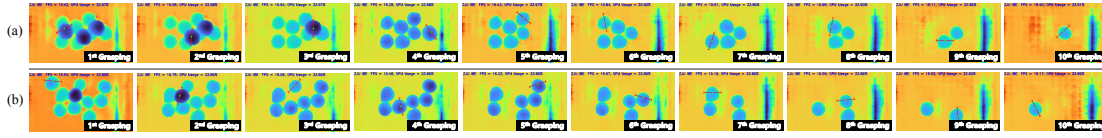


Figure 6.5: Visualization of predicted grasp poses overlaid on real depth images.

6.4.2 Grasp Execution Results in Real-world Scenarios

To evaluate the real-world performance of the proposed grasping framework, we conducted 280 grasp attempts under various object quantities and stacked configurations, ranging from 1 to 20 peaches. The execution pipeline consists of perception, grasp pose prediction, motion planning, and actuation. Fig. 6.5 visualizes the predicted grasp poses overlaid on real depth images, where the inferred grasp configurations are directly aligned with the perceived depth information to guide subsequent trajectory planning and execution.

In general, the system achieved an average success rate of 89.29% (249/280). As shown in Table 6.1, the grasp success rate decreased slightly as the number of objects increased. In simpler scenes (1–5 peaches), the system reached 94.29% success. However, in more cluttered scenes (16–20 peaches), the success rate dropped to 87.14%, reflecting the increased difficulty due to tighter collision-free space.

To analyze failure causes, we categorized them into three types: perception errors, where PG-CNN predicted poses that led to potential collisions; execution errors, which were due to positioning deviation during path execution; and slip, caused by insufficient gripping force or fruit geometry during pick-and-place. Among the 31 failures, slip was the most common (14 cases), followed by grasping (10) and perception (7). The primary factor behind slippage was the limited holding stability of the two-finger pneumatic gripper, especially when grasping fruit tips or stems with small diameters.

We further assessed robustness to fruit orientation. When the fruit tips pointed upward, the success rate was 86.43%; when stems were lateral, the rate increased to 89.64%. This indicates the system maintained reliable grasp performance under moderate object variation.

Compared with Dex-Net 4.0, our PG-CNN-based framework significantly reduced failure due to collision. Specifically, perception-related failures dropped by 9 cases and grasping-related failures by 7, resulting in an overall success rate of 89.29%, which outperformed Dex-Net’s 82.86%. Although real-world performance was slightly lower than simulation (93.45%) due to domain shift, gripper constraints, and sensor errors, the results demonstrate strong generalization and deployment readiness of the system.

6.4.3 Failure Case Analysis

During real-world grasp execution, a total of 31 failure cases were recorded and categorized into three major types: slip, perception error, and grasping error. According to Table 6.1, slip was the most frequent cause (14 cases), followed by grasping errors (10) and perception errors (7). A detailed analysis of each failure type is presented below.

Slip Failures. Slip occurred during both grasping and releasing phases, particularly when the fruit tip pointed upward or when the stem region was thin. The pneumatic-driven two-finger gripper used in this work has limited holding stability due to its dependence on air pressure, contact geometry, and surface friction. Moreover, pneumatic systems are inherently nonlinear, multivariable, and time-delayed, making precise force control difficult. This hardware limitation explains the relatively high slip rate and highlights the need for improved gripper design.

Perception Errors. Perception errors refer to grasp poses predicted by PG-CNN that led to collisions in the physical world. These errors typically occurred in scenes with occlusions, overlapping fruits, or discontinuous depth edges. Although PG-CNN demonstrated strong performance in simulation, the Sim-to-Real domain gap—caused by sensor noise, specular reflections, or missing depth values—still impacted reliability. Incorporating multimodal perception (e.g., RGB-Depth-Tactile) could further improve robustness.

Grasping Errors. Grasping errors were mainly caused by positional deviations during trajectory execution, often due to hand-eye calibration inaccuracies or planning overshoot. Despite using Tsai-Lenz calibration and the MoveIt! planner, accumulated errors and redundancy in the kinematic chain could result in offset grasps. Future work may explore visual servoing or online correction strategies to enhance execution fidelity.

6.5 Summary and Future Directions

This chapter has presented the implementation and evaluation of the proposed “Imaging–Perception–Execution” grasping framework for fragile fruits in cluttered scenes. By constructing two simulation datasets (PeachStereo and PeachGrasp) and leveraging transfer learning for Sim-to-Real deployment, the system demonstrated strong robustness and generalization. Extensive experiments validated the effectiveness of PG-CNN in predicting safe grasp poses under varying densities, object types, and physical configurations.

6.5.1 Strengths and Contributions

The major strengths and contributions of this work are summarized as follows:

1. **Integrated grasping system for fragile object manipulation:** The proposed framework combines stereo-based perception, grasp quality prediction, and motion execution into a unified pipeline suitable for real-world deployment.
2. **Collision-aware PG-CNN model:** Trained with simulation data labeled for collision outcomes, the model implicitly learns spatial constraints without requiring explicit modeling of surrounding objects.

3. **High-quality synthetic datasets:** The PeachStereo and PeachGrasp datasets address the data scarcity problem in real-world scenes by providing annotated stereo images and grasp samples with collision information.
4. **Sim-to-Real generalization without real-data fine-tuning:** Both CGFNet and PG-CNN perform robustly on real-world scenes using frozen weights trained solely on simulation data.
5. **Cross-species generalization:** The system maintains stable performance when manipulating various previously unseen fruits with different shapes and sizes, demonstrating practical applicability.

6.5.2 Discussion and Future Directions

While the proposed approach demonstrates encouraging performance, several aspects provide opportunities for further extension and refinement in future studies.

1. **Gripper design considerations:** The present system employs a pneumatic-driven rigid gripper, which is well-suited for the studied scenarios but offers limited force modulation capabilities. Future work may investigate soft or hybrid gripper designs with enveloping structures to further enhance grasp robustness and object safety.
2. **Control strategy extension:** In this work, grasp execution is realized under an open-loop control setting, which simplifies system integration and evaluation. Incorporating tactile feedback and closed-loop force control is a promising direction to improve adaptability in more dynamic or uncertain environments.
3. **Perception modality expansion:** The perception module primarily leverages depth information, which provides efficient geometric cues for grasp reasoning. Extending the framework to integrate complementary modalities such as RGB images, tactile signals, and proprioceptive feedback could further improve robustness under challenging sensing conditions.
4. **Grasp reasoning beyond geometry:** The current grasp strategy emphasizes geometry-driven reasoning, which proves effective for collision-aware grasp generation. Future extensions may incorporate semantic and task-level information, such as object categories or functional context, enabling more informed and task-aware grasp decisions.

Overall, this work establishes a solid foundation for collision-aware grasp learning and dataset construction, and offers a scalable framework that can be progressively extended toward more complex and intelligent agricultural manipulation tasks.

Chapter 7

Conclusions and Outlook

7.1 Conclusion

This dissertation presented a comprehensive study of Sim2Real transfer in robotic perception and manipulation, addressing four fundamental challenges: realistic data generation, transparent object reconstruction, fragile object grasping, and zero-shot generalization. These representative works span physics-based rendering, scene understanding, and robotic grasp prediction, and together form a cohesive Sim2Real pipeline. The research demonstrates the feasibility of transferring from simulation-driven modeling to robust real-world perception and manipulation in robotics.

- A physically grounded framework for synthetic data generation in structured light imaging was proposed. By explicitly modeling light–material interactions, structured shadows, and sensor noise, the pipeline produces depth maps with both high fidelity and strong task relevance. These synthetic datasets significantly enhanced Sim2Real generalization for detection and segmentation tasks, providing a scalable and reproducible alternative to costly manual annotation in industrial vision applications.
- The challenge of transparent object reconstruction was addressed through a stereo-vision method guided by physics-based priors. By incorporating principles of light refraction and reflection into stereo matching, the method enabled physically plausible recovery of transparent geometries in both synthetic and real-world scenes. This approach overcame the inherent limitations of conventional RGB-D sensors, which often fail when confronted with transparent or reflective surfaces.
- Zero-shot 3D reconstruction was explored by combining a frozen vision backbone with a lightweight LoRA-based branch and task-specific decoder. This design retained the generalization capability of foundation models while adapting efficiently to novel object categories. The approach demonstrated a data-efficient paradigm for robotic 3D perception, showing that scalable adaptation can be achieved without large-scale retraining.

- A unified Sim2Real framework for fragile object grasping was developed, which integrates 3D reconstruction, visual reasoning, and collision-aware grasp planning. The system was trained entirely in simulation under physical constraints and then successfully deployed on real robotic hardware. Experimental results verified its safety and reliability in handling delicate objects, where contact-rich interactions demand precise perception and careful control.

Collectively, these contributions demonstrate that closing the Sim2Real gap requires progress on multiple fronts, including data synthesis strategies, supervision design, training objectives, and evaluation methodologies. An important observation across the studies is that depth-based modalities consistently exhibit stronger performance than purely RGB-based approaches in industrial domains. This advantage is particularly evident in scenarios where texture information is weak, such as metallic parts, or unstable, as in the case of packaging materials. Geometry, therefore, emerges as a reliable perceptual anchor for robotic perception and manipulation, offering robustness against visual ambiguity and environmental variability.

7.2 Discussion and Open Challenges

While this dissertation has made systematic progress in simulation-based data generation and Sim-to-Real transfer for robotic perception and manipulation, several open challenges remain that point to promising directions for future research. These challenges relate to dataset coverage, computational efficiency, generalization strategies, and the effective integration of synthetic and real data.

1. **Dataset coverage for industrial-scale scenarios.** In recent years, both synthetic and real-world datasets for robotic perception have expanded rapidly and enabled significant advances. However, many publicly available datasets primarily target household objects or generic tabletop settings. Large-scale datasets tailored to industrial scenarios—such as bin-picking, manipulation of reflective or transparent materials, and perception in densely cluttered warehouse environments—remain comparatively underexplored. As a result, models trained on open-source benchmarks may exhibit reduced robustness when applied to specialized industrial conditions, for example when handling transparent packaging, specular metallic surfaces, or severe object occlusions. Expanding dataset coverage toward such challenging cases remains an important step toward broader industrial deployment.
2. **Computational demands of physically grounded rendering.** Physically realistic rendering plays a critical role in generating high-quality synthetic depth data by modeling light-material interactions, sensor noise, and illumination effects. However, such fidelity typically comes at the cost of high computational complexity. Offline rendering engines, such as PBRT or Blender Cycles, can produce highly accurate results but require substantial time and computational resources, which limits scalability for large-scale dataset generation. This challenge motivates future exploration of more efficient rendering strategies, including real-time

approximations or learning-based acceleration methods, to balance physical accuracy with practical data generation efficiency.

3. **Sensitivity of domain randomization strategies.** Domain randomization has proven to be an effective and flexible approach for improving Sim-to-Real transfer by introducing variability during training. Nevertheless, its success depends strongly on how well the randomized simulation space captures the variations present in the real world. In practice, the design of randomization ranges for textures, lighting, and geometry is often heuristic and task-dependent, and theoretical guarantees remain limited. When critical real-world factors are insufficiently represented, generalization performance may degrade. Developing more principled or adaptive domain randomization strategies therefore remains an open research direction.
4. **Understanding the role of limited real data integration.** Empirical results in this dissertation, as well as in related work, indicate that incorporating a small amount of real data during training can yield substantial performance improvements. This observation suggests a promising path toward data-efficient Sim-to-Real learning. However, the underlying mechanisms are not yet fully understood. It remains an open question whether such gains arise from mitigating covariate shift, anchoring learned representations to real-world statistics, or providing regularization against simulation-specific artifacts. Clarifying when, where, and how small-scale real data should be integrated into training pipelines will be essential for developing more principled and reliable hybrid learning frameworks.

7.3 Future Research Directions

Building on the limitations identified in this dissertation, several promising directions for future research are outlined below. These directions aim to further advance the reliability, efficiency, and generalization capability of Sim-to-Real transfer in robotic perception and manipulation.

1. **Continuous dataset generation for industrial applications.** A key bottleneck remains the insufficient availability of large-scale and task-specific datasets for industrial robotics. Future work should focus on establishing pipelines for the continuous generation of simulation data that reflect real-world industrial scenarios, including bin-picking under clutter, reflective materials, transparent packaging, and objects with complex geometries or occlusions. Instead of static datasets, a dynamic and expandable data generation framework would enable open-source repositories to evolve in tandem with emerging application needs. This would ensure that foundation models trained on these datasets remain relevant, adaptable, and deployable in real-world industrial contexts.
2. **Acceleration and optimization of simulation pipelines.** The high computational cost of physically realistic rendering limits the scalability of dataset generation.

Future work should therefore explore faster rendering strategies that balance physical fidelity with computational efficiency. This includes the development of AI-accelerated differentiable rendering, neural radiance field (NeRF) based scene synthesis, and real-time GPU ray tracing tailored to depth imaging. Such innovations would make it possible to generate millions of physically realistic samples at a fraction of the current cost, thereby enabling broader exploration of data-driven methods without prohibitive time and resource constraints.

3. **Physically realistic modeling for multiple 3D sensors.** Current simulation pipelines are often tailored to a single depth sensing modality. However, real-world robotics increasingly relies on a diverse set of 3D sensors, such as Intel RealSense structured-light cameras, ZED stereo cameras, and time-of-flight sensors. Future extensions should incorporate accurate sensor-specific models into simulation frameworks, accounting for their unique noise profiles, optical distortions, and measurement principles. This would support not only single-sensor training but also cross-device generalization, where models trained on simulated data for one sensor can robustly transfer to another without significant performance degradation.
4. **Multi-modal adaptation and closed-loop integration.** Robust robotic perception will require leveraging complementary sensory modalities beyond RGB and depth. Future research should investigate domain adaptation techniques that jointly exploit multi-modal inputs, such as surface normals, tactile feedback, and event-based vision. Moreover, embedding simulation into closed-loop perception-and-control pipelines will allow perception models to be optimized jointly with action policies, ensuring that representations learned from synthetic data are directly aligned with downstream control performance. This closed-loop integration will be crucial for deploying robots that operate reliably in dynamic and unstructured environments.
5. **Standardized synthetic dataset construction.** In the long term, advancing general-purpose robot learning requires large-scale, standardized synthetic datasets that are both physically grounded and richly annotated. Such datasets should include multi-task labels for perception and manipulation, covering tasks such as grasping, placing, tool use, deformable object handling, and long-horizon manipulation. By establishing standardized protocols for dataset construction, annotation, and evaluation, the community can move towards reproducible benchmarks that accelerate progress and foster collaboration across industrial and academic domains.

In conclusion, this dissertation establishes a principled framework for Sim-to-Real transfer in robotic perception and manipulation. The proposed works demonstrate that advances in simulation modeling, dataset design, and robust learning algorithms can significantly narrow the gap between synthetic training and real-world deployment. Looking forward, deeper synergy across data generation, multi-modal adaptation, and control

integration will be essential for building robotic systems that are reliable, data-efficient, and generalizable in real-world industrial and service scenarios.

Appendix A

List of Abbreviations

Sim2Real	Simulation to Reality
DR	Domain Randomization
DA	Domain Adaptation
DoLP	Degree of Linear Polarimetric
AoLP	Angle of Polarization
LWIR	Long-wave Infrared
POV-Ray	Persistence of Vision Raytracer
PMD-style	Photonic Mixer Device-based
AMCW	Amplitude-Modulated Continuous-Wave
MPI	Multipath Interference
UDA	Unsupervised Domain Adaptation
R2R	Room-to-Room
VLN	Vision-and-Language Navigation
SR	Success Rate
3D	Three-Dimensional
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue + Depth
ViT	Vision Transformer
LoRA	Low-Rank Adaptation

SOTA State of the Art
PBR Physically Based Rendering
GAN Generative Adversarial Network
LiDAR Light Detection and Ranging
AI Artificial Intelligence
GPU Graphics Processing Unit
CNN Convolutional Neural Network
GRU Gated Recurrent Unit
DoF Degrees of Freedom
6D Six Degrees of Freedom
ToF Time of Flight
CAD Computer-Aided Design
IoU Intersection over Union
mAP mean Average Precision
AP Average Precision
RLE Run-Length Encoding
DLP Digital Light Processing
FOV Field of View
FPN Feature Pyramid Network
FFN Feed-Forward Network
MLP Multi-Layer Perceptron
GELU Gaussian Error Linear Unit
IR Infrared
HDRI High Dynamic Range Image
FPS Frames Per Second
RMS Root Mean Square (Error)
AvgErr Average Error

PCGG Point Cloud Grasp Generation

FFGAD Five-Finger Grasping Affordance Detection

FFH Five-Finger Hand

SRM Stereo Reconstruction Module

EPE End-Point Error

RMSE Root Mean Square Error

SE(3) Special Euclidean group in 3D

MBConv Mobile Inverted Bottleneck Convolution

SO(3) Special Orthogonal Group in 3D

Appendix B

Publications

Below are all the publications published during my PhD study, sorted by publication date.

- [1] **Kaixin Bai**, Lei Zhang, Yiwen Liu, Zhaopeng Chen, and Jianwei Zhang. StereoAnything: Advanced zero-shot stereo imaging for multi-finger grasp detection with transparent objects. *IEEE Transactions on Cybernetics (TCYB)*, 2025. (Under Review).
- [2] Qingyu Wang*, **Kaixin Bai***, Lei Zhang, Qiang Li, Alois Knoll, Jianwei Zhang, Yibin Ying, and Mingchuan Zhou. Towards collision-aware robotic fragile fruit grasping: A sim-to-real framework for perception, reasoning, and execution. *IEEE/ASME Transactions on Mechatronics (TMECH)*, 2025.
- [3] Qingyu Wang, **Kaixin Bai**, Lei Zhang, Zhizhong Sun, Tianze Jia, Dong Hu, Qiang Li, Jianwei Zhang, Alois Knoll, Huanyu Jiang, and Mingchuan Zhou. Towards reliable and damage-less robotic fragile fruit grasping: An enveloping gripper with multimodal strategy inspired by Asian elephant trunk. *Computers and Electronics in Agriculture*, 2025.
- [4] Lei Zhang, Soumya Mondal, Zhenshan Bing, **Kaixin Bai**, Diwen Zheng, Zhaopeng Chen, Alois Christian Knoll, and Jianwei Zhang. DORA: Object Affordance-Guided Reinforcement Learning for Dexterous Robotic Manipulation. *Under Review*, 2025.
- [5] Minheng Ni, Lei Zhang, Zihan Chen, **Kaixin Bai**, Zhaopeng Chen, Jianwei Zhang, Lei Zhang, and Wangmeng Zuo. Don't Let Your Robot be Harmful: Responsible Robotic Manipulation via Safety-as-Policy. *Under Review*, 2025.
- [6] Lik Hang Kenny Wong, Xueyang Kang, **Kaixin Bai**, Jianwei Zhang. A survey of robotic navigation and manipulation with physics simulators in the era of embodied AI. *ACM Computing Surveys*, 2025. (Under Review).

- [7] **Kaixin Bai**, Huajian Zeng, Lei Zhang, Yiwen Liu, Hongli Xu, Zhaopeng Chen, and Jianwei Zhang. ClearDepth: Enhanced stereo perception of transparent objects for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA)*, 2026. (Under Review).
- [8] Lei Zhang, **Kaixin Bai**, Guowen Huang, Zhenshan Bing, Zhaopeng Chen, Alois Knoll, and Jianwei Zhang. Multi-fingered robotic hand grasping in cluttered environments through hand-object contact semantic mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.
- [9] Yunlong Wang, Lei Zhang, Yuyang Tu, Hui Zhang, **Kaixin Bai**, Zhaopeng Chen, and Jianwei Zhang. ToolEENet: Tool affordance 6D pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [10] **Kaixin Bai**, Lei Zhang, Zhaopeng Chen, Fang Wan, and Jianwei Zhang. Close the sim2real gap via physically-based structured light synthetic data simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [11] Lei Zhang, **Kaixin Bai**, Qiang Li, Zhaopeng Chen, and Jianwei Zhang. Deep learning-based collision-aware cable grasping in cluttered environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [12] Lei Zhang, Jiacheng Pei, **Kaixin Bai**, Zhaopeng Chen, and Jianwei Zhang. A Closed-Loop Multi-perspective Visual Servoing Approach with Reinforcement Learning. In *IEEE International Conference on Robotics and Biomimetics (RO-BIO)*, 2023.
- [13] **Kaixin Bai***, Lei Zhang*, Zhaopeng Chen, and Jianwei Zhang. Learning of 6D object poses with multi-task point-wise regression deep networks. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2022.
- [14] Lei Zhang*, **Kaixin Bai***, Zhaopeng Chen, Yunlei Shi, and Jianwei Zhang. Towards precise model-free robotic grasping with sim-to-real transfer learning. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2022.

Appendix C

Acknowledgements

As I bring my doctoral journey to a close, I am filled with gratitude. These four years have been not only a pursuit of academic knowledge but also an important chapter of personal growth. Here, I would like to express my sincere thanks to all those who have supported me throughout this journey.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Jianwei Zhang. His rigorous academic attitude, broad vision, and insightful guidance have profoundly shaped my research and personal development. During my doctoral study, Prof. Zhang not only provided invaluable academic supervision but also gave me encouragement and support in many aspects of life. It has been a great honor and privilege to complete my PhD under his guidance.

I am also grateful to my colleagues and friends in the lab. Their collaboration, discussions, and companionship have made the research journey more rewarding and enjoyable. Their encouragement has helped me overcome challenges and celebrate achievements together.

Finally, I would like to express my heartfelt appreciation to my family, whose unconditional love and constant support have been my greatest source of strength throughout these years.

Bibliography

- [1] Fatemeh Mohammadi Amin, Darwin G Caldwell, and Hans Wernher van de Venn. Enhancing human-robot collaboration: A sim2real domain adaptation algorithm for point cloud segmentation in industrial environments. *arXiv preprint arXiv:2506.09552*, 2025.
- [2] Janis Arents, Bernd Lesser, Andis Bizuns, Roberts Kadikis, Elvijs Buls, and Modris Greitans. Synthetic data of randomly piled, similar objects for deep learning-based object detection. In *International Conference on Image Analysis and Processing*, pages 706–717. Springer, 2022.
- [3] Kaixin Bai, Huajian Zeng, Lei Zhang, Yiwen Liu, Hongli Xu, Zhaopeng Chen, and Jianwei Zhang. Cleardepth: enhanced stereo perception of transparent objects for robotic manipulation. *arXiv preprint arXiv:2409.08926*, 2024.
- [4] Kaixin Bai, Lei Zhang, Zhaopeng Chen, Fang Wan, and Jianwei Zhang. Close the sim2real gap via physically-based structured light synthetic data simulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17035–17041. IEEE, 2024.
- [5] Kaixin Bai, Lei Zhang, Yiwen Liu, Zhaopeng Chen, and Jianwei Zhang. Stereoanything: Advanced zero-shot stereo imaging for multi-finger grasp detection with transparent objects. *Authorea Preprints*.
- [6] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018.
- [7] Liuyu Bian, Pengyang Shi, Weihang Chen, Jing Xu, Li Yi, and Rui Chen. Transtouch: Learning transparent objects depth sensing through sparse touches. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9566–9573. IEEE, 2023.
- [8] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.
- [9] Samarth Brahmabhatt, Ankur Handa, James Hays, and Dieter Fox. ContactGrasp: Functional multi-finger grasp synthesis from contact. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393, 2019. doi:10.1109/IROS40897.2019.8967960.
- [10] Rafal Broda, Alexander Schnerring, Dominik Schnaus, Michael Nieslony, Julian J Krauth, Marc Röger, Sonja Kallio, Rudolph Triebel, and Robert Pitz-Paál. Bridging the sim2real gap: Training deep neural networks for heliostat detection with purely synthetic data. *Solar Energy*, 300:113728, 2025.

- [11] David Bulczak, Martin Lambers, and Andreas Kolb. Quantified, interactive simulation of amcw tof camera including multipath effects. *Sensors*, 18(1):13, 2017.
- [12] Shurong Chai, Rahul Kumar Jain, Shiyu Teng, Jiaqing Liu, Yinhao Li, Tomoko Tateyama, and Yen-wei Chen. Ladder fine-tuning approach for sam integrating complementary network. *Procedia Computer Science*, 246:4951–4958, 2024.
- [13] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.
- [14] 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.
- [15] 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. *arXiv preprint arXiv:2211.01644*, 2022.
- [16] Kang Chen, Shaochen Wang, Beihao Xia, Dongxu Li, Zhen Kan, and Bin Li. Tode-trans: Transparent object depth estimation with transformer. *arXiv preprint arXiv:2209.08455*, 2022.
- [17] Mingyou Chen, Zengxing Chen, Lufeng Luo, Yunchao Tang, Jiabing Cheng, Huiling Wei, and Jinhai Wang. Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. *Computers and electronics in agriculture*, 219:108774, 2024.
- [18] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, Xinge Zhu, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. *Advances in Neural Information Processing Systems*, 36:75896–75910, 2023.
- [19] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023.
- [20] Yugan Chen, Lin Zhao, Yalong Xu, Honglei Zu, Xiaoqi An, and Guangyu Li. Domain adaptive pose estimation via multi-level alignment. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.
- [21] Anoop Cherian, Siddarth Jain, and Tim K Marks. Few-shot transparent instance segmentation for bin picking. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5009–5016. IEEE, 2024.

- [22] Benjamin Choo, Michael Landau, Michael DeVore, and Peter A Beling. Statistical analysis-based error models for the microsoft kinect™ depth sensor. *Sensors*, 14(9):17430–17450, 2014.
- [23] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. In *International Conference on Pattern Recognition, ICPR*, 2020.
- [24] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.
- [25] Aidan Curtis, Eric Li, Michael Noseworthy, Nishad Gothoskar, Sachin Chitta, Hui Li, Leslie Pack Kaelbling, and Nicole E Carey. Flow-based domain randomization for learning and sequencing robotic skills. In *Forty-second International Conference on Machine Learning*, 2025.
- [26] Qiyu Dai, Jiyao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European Conference on Computer Vision*, pages 374–391. Springer, 2022.
- [27] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhaoh Zhang, and He Wang. Graspernerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. *arXiv preprint arXiv:2210.06575*, 2022.
- [28] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.
- [29] Sagi Eppel, Haoping Xu, Yi Ru Wang, and Alan Aspuru-Guzik. Predicting 3d shapes, masks, and properties of materials inside transparent containers, using the transproteus cgi dataset. *Digital Discovery*, 1(1):45–60, 2022.
- [30] 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.
- [31] 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.
- [32] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by back-propagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

- [33] Chen Guo-Hua, Wang Jun-Yi, and Zhang Ai-Jun. Transparent object detection and location based on rgb-d camera. In *Journal of Physics: Conference Series*, volume 1183, page 012011. IOP Publishing, 2019.
- [34] Jing Hao, Moyun Liu, and Kuo Feng Hung. Gem: Boost simple network for glass surface segmentation via segment anything model and data synthesis. *arXiv preprint arXiv:2401.15282*, 2024.
- [35] Hao He, Xiangtai Li, Guangliang Cheng, Jianping Shi, Yunhai Tong, Gaofeng Meng, Véronique Prinet, and LuBin Weng. Enhanced boundary learning for glass-like object segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15859–15868, 2021.
- [36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [37] Christoph Heindl, Lukas Brunner, Sebastian Zambal, and Josef Scharinger. Blendtorch: A real-time, adaptive domain randomization library. In *International Conference on Pattern Recognition*, pages 538–551. Springer, 2021.
- [38] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [39] 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.
- [40] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021.
- [41] Sunghoon Im, Hyowon Ha, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Deep depth from uncalibrated small motion clip. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1225–1238, 2019.
- [42] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12627–12637, 2019.
- [43] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023.

-
- [44] Hualie Jiang, Zhiqiang Lou, Laiyan Ding, Rui Xu, Minglang Tan, Wenjie Jiang, and Rui Huang. Defom-stereo: Depth foundation model based stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21857–21867, 2025.
 - [45] Jiaqi Jiang, Guanqun Cao, Jiankang Deng, Thanh-Toan Do, and Shan Luo. Robotic perception of transparent objects: A review. *IEEE Transactions on Artificial Intelligence*, 5(6):2547–2567, 2023.
 - [46] Jiaqi Jiang, Guanqun Cao, Thanh-Toan Do, and Shan Luo. A4t: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation. *IEEE Robotics and Automation Letters*, 7(4):9826–9833, 2022.
 - [47] Chuhao Jin, Wenhui Tan, Jiange Yang, Bei Liu, Ruihua Song, Limin Wang, and Jianlong Fu. Alphablock: Embodied finetuning for vision-language reasoning in robot manipulation. *arXiv preprint arXiv:2305.18898*, 2023.
 - [48] 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.
 - [49] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.
 - [50] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36:29914–29934, 2023.
 - [51] Maik Keller and Andreas Kolb. Real-time simulation of time-of-flight sensors. *Simulation Modelling Practice and Theory*, 17(5):967–978, 2009.
 - [52] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects. In *6th annual conference on robot learning*, 2022.
 - [53] Samin Khan, Buu Phan, Rick Salay, and Krzysztof Czarnecki. Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks. In *CVPR workshops*, volume 3, page 4, 2019.
 - [54] 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.

- [55] Sebastian Koch and Yurii Piadyk. Hardware design and accurate simulation for benchmarking of 3d reconstruction algorithms. *Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2022.
- [56] Sarah C Kucker, Larissa K Samuelson, Lynn K Perry, Hanako Yoshida, Eliana Colunga, Megan G Lorenz, and Linda B Smith. Reproducibility and a unifying explanation: Lessons from the shape bias. *Infant Behavior and Development*, 54:156–165, 2019.
- [57] Martin Lambers, Stefan Hoberg, and Andreas Kolb. Simulation of time-of-flight sensors for evaluation of chip layout variants. *IEEE Sensors Journal*, 15(7):4019–4026, 2015.
- [58] Michael J Landau, Benjamin Y Choo, and Peter A Beling. Simulating kinect infrared and depth images. *IEEE transactions on cybernetics*, 46(12):3018–3031, 2015.
- [59] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.
- [60] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022.
- [61] Shoujie Li, Haixin Yu, Wenbo Ding, Houde Liu, Linqi Ye, Chongkun Xia, Xueqian Wang, and Xiao-Ping Zhang. Visual–tactile fusion for transparent object grasping in complex backgrounds. *IEEE Transactions on Robotics*, 39(5):3838–3856, 2023.
- [62] Tianan Li, Zhehan Chen, Huan Liu, and Chen Wang. Fdct: Fast depth completion for transparent objects. *IEEE Robotics and Automation Letters*, 8(9):5823–5830, 2023.
- [63] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2020.
- [64] Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. Neto: neural reconstruction of transparent objects with self-occlusion aware refraction-tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18547–18557, 2023.

- [65] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020.
- [66] 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.
- [67] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021.
- [68] Wei Liu, Shijie Tian, Qingyu Wang, and Huanyu Jiang. Key technologies of plug tray seedling transplanters in protected agriculture: A review. *Agriculture*, 13(8):1488, 2023.
- [69] Kazuki Maeno, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Light field distortion feature for transparent object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2786–2793, 2013.
- [70] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5620–5627, 2018. doi:10.1109/ICRA.2018.8460887.
- [71] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.
- [72] Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1957–1964. IEEE, 2016.
- [73] Cristina Mata, Nick Locascio, Mohammed Azeem Sheikh, Kenny Kihara, and Dan Fischetti. Standardsim: A synthetic dataset for retail environments. In *International Conference on Image Analysis and Processing*, pages 65–76. Springer, 2022.
- [74] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. URL: <http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16>.

- [75] 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.
- [76] 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.
- [77] Fabio Menna, Fabio Remondino, Roberto Battisti, and Erica Nocerino. Geometric investigation of a gaming active device. In *Videometrics, Range Imaging, and Applications XI*, volume 8085, pages 173–187. SPIE, 2011.
- [78] Mehdi Mousavi and Rolando Estrada. Supercaustics: Real-time, open-source simulation of transparent objects for deep learning applications. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 649–655. IEEE, 2021.
- [79] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi:10.1145/3528223.3530127.
- [80] Tobias Müller, Tobias Schmähling, Stefan Elser, and Jörg Eberhardt. Deep learning for generating time-of-flight camera artifacts. *Journal of Imaging*, 10(10):246, 2024.
- [81] Chuong V Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *2012 second international conference on 3D imaging, modeling, processing, visualization & transmission*, pages 524–530. IEEE, 2012.
- [82] Margarita Osadchy, David Jacobs, Ravi Ramamoorthi, and David Tucker. Using specularities in comparing 3d models and 2d images. *Computer Vision and Image Understanding*, 111(3):275–294, 2008.
- [83] Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. *arXiv preprint arXiv:1806.09755*, 2018.
- [84] Cody J Phillips, Matthieu Lecce, and Kostas Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Robotics: Science and Systems*, volume 3, page 3. Michigan, USA, 2016.
- [85] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

- [86] Benjamin Planche and Rajat Vikram Singh. Physics-based differentiable depth sensor simulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14387–14397, 2021.
- [87] Benjamin Planche, Ziyang Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Oliver Lehmann, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, et al. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. In *2017 International conference on 3d vision (3DV)*, pages 1–10. IEEE, 2017.
- [88] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019.
- [89] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1221–1224, 2017.
- [90] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [91] Maximiliano Rojas, Gabriel Hermosilla, Daniel Yunge, and Gonzalo Farias. An easy to use deep reinforcement learning library for ai mobile robots in isaac sim. *Applied Sciences*, 12(17):8429, 2022.
- [92] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [93] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.
- [94] Shreyak 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 *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3634–3642. IEEE, 2020.
- [95] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.
- [96] Jun Shi, A Yong, Yixiang Jin, Dingzhe Li, Haoyu Niu, Zhezhu Jin, and He Wang. Asgrasp: Generalizable transparent object reconstruction and 6-dof grasp detection from rgb-d active stereo camera. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 5441–5447. IEEE, 2024.

- [97] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.
- [98] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.
- [99] Tutian Tang, Jiyu Liu, Jieyi Zhang, Haoyuan Fu, Wenqiang Xu, and Cewu Lu. Rftrans: Leveraging refractive flow of transparent objects for surface normal estimation and manipulation. *IEEE Robotics and Automation Letters*, 9(4):3735–3742, 2024.
- [100] Yunchao Tang, Shaojun Qi, Lixue Zhu, Xianrong Zhuo, Yunqi Zhang, and Fan Meng. Obstacle avoidance motion in mobile robotics. *Journal of System Simulation*, 36(1):1–26, 2024.
- [101] Thang To, Jonathan Tremblay, Duncan McKay, Yukie Yamaguchi, Kirby Leung, Adrian Balanon, Jia Cheng, William Hodge, and Stan Birchfield. Ndds: Nvidia deep learning dataset synthesizer. In *Proceedings of the CVPR 2018 Workshop on Real World Challenges and New Benchmarks for Deep Learning in Robotic Vision, Salt Lake City, UT, USA*, pages 18–22, 2018.
- [102] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489. IEEE, 2018.
- [103] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017. doi:10.1109/IROS.2017.8202133.
- [104] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018. doi:10.1109/CVPRW.2018.00143.
- [105] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

- [106] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021.
- [107] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.
- [108] Quan Vuong, Sharad Vikram, Hao Su, Sicun Gao, and Henrik I Christensen. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies? *arXiv preprint arXiv:1903.11774*, 2019.
- [109] An Wang, Mobarakol Islam, Mengya Xu, Yang Zhang, and Hongliang Ren. Sam meets robotic surgery: an empirical study on generalization, robustness and adaptation. In *International conference on medical image computing and computer-assisted intervention*, pages 234–244. Springer, 2023.
- [110] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023.
- [111] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.
- [112] Han Wang, Xingyu Chen, and Xuguang Lan. An exploration of domain adaptation applying to grasp detection algorithm. In *2020 Chinese Automation Congress (CAC)*, pages 5332–5337. IEEE, 2020.
- [113] Xianqi Wang, Gangwei Xu, Hao Jia, and Xin Yang. Selective-stereo: Adaptive frequency information selection for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19701–19710, 2024.
- [114] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, 33:17721–17732, 2020.
- [115] Yi Ru Wang, Yuchi Zhao, Haoping Xu, Saggi Eppel, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Mvtrans: Multi-view perception of transparent objects. *arXiv preprint arXiv:2302.11683*, 2023.
- [116] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5249–5260, 2025.

- [117] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.
- [118] B Wu and H Huang. Survey on 3d reconstruction of transparent objects. *Journal of Computer-Aided Design & Computer Graphics*, 32(2):173–180, 2020.
- [119] Zhiyuan Wu, Shuai Su, Qijun Chen, and Rui Fan. Transparent objects: A corner case in stereo matching. In *ICRA*, pages 12353–12359, 2023.
- [120] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.
- [121] Aoran Xiao, Weihao Xuan, Heli Qi, Yun Xing, Ruijie Ren, Xiaoqin Zhang, and Shijian Lu. Cat-sam: Conditional tuning network for few-shot adaptation of segmentation anything model. arxiv 2024. *arXiv preprint arXiv:2402.03631*, 2024.
- [122] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.
- [123] 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.
- [124] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer. *arXiv preprint arXiv:2101.08461*, 2021.
- [125] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023.
- [126] Haoping Xu, Yi Ru Wang, Sagi Eppel, Alan Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Seeing glass: joint point cloud and depth completion for transparent objects. *arXiv preprint arXiv:2110.00087*, 2021.
- [127] Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3442–3450, 2015.
- [128] Tao Yan, Shufan Xu, Hao Huang, Helong Li, Lu Tan, Xiaojun Chang, and Rynson WH Lau. Nrglassnet: Glass surface detection from visible and near-infrared image pairs. *Knowledge-Based Systems*, 294:111722, 2024.

-
- [129] Zhiqiang Yan, Hongyuan Wang, Zebin Wang, Xiang Liu, and Qianhao Ning. Imaging simulation of the amcw tof camera based on path tracking. *Applied Optics*, 61(18):5474–5482, 2022.
- [130] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024.
- [131] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.
- [132] Lei Ye, Fengyun Wu, Xiangjun Zou, and Jin Li. Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bi-directional rrt approach. *Computers and Electronics in Agriculture*, 215:108453, 2023.
- [133] Yu-Ting Yen, Chia-Ni Lu, Wei-Chen Chiu, and Yi-Hsuan Tsai. 3d-pl: Domain adaptive depth estimation with 3d-aware pseudo-labeling. In *European Conference on Computer Vision*, pages 710–728. Springer, 2022.
- [134] Dongshuo Yin, Leiyi Hu, Bin Li, and Youqun Zhang. Adapter is all you need for tuning visual tasks. *arXiv preprint arXiv:2311.15010*, 2023.
- [135] Sheng Yu, Di-Hua Zhai, and Yuanqing Xia. Cgnet: Robotic grasp detection in heavily cluttered scenes. *IEEE/ASME Transactions on Mechatronics*, 28(2):884–894, 2022.
- [136] Wenxi Yue, Jing Zhang, Kun Hu, Qiuxia Wu, Zongyuan Ge, Yong Xia, Jiebo Luo, and Zhiyong Wang. Part to whole: Collaborative prompting for surgical instrument segmentation. *CoRR*, 2023.
- [137] Wenxi Yue, Jing Zhang, Kun Hu, Yong Xia, Jiebo Luo, and Zhiyong Wang. Surgicalsam: Efficient class promptable surgical instrument segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6890–6898, 2024.
- [138] Sergey Zakharov, Rareş Ambruş, Vitor Guizilini, Wadim Kehl, and Adrien Gaidon. Photo-realistic neural domain randomization. In *European Conference on Computer Vision*, pages 310–327. Springer, 2022.
- [139] Jiaming Zhang, Kailun Yang, Angela Constantinescu, Kunyu Peng, Karin Müller, and Rainer Stiefelhagen. Trans4trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1760–1770, 2021.

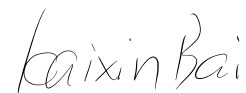
- [140] Lei Zhang, Kaixin Bai, Guowen Huang, Zhenshan Bing, Zhaopeng Chen, Alois Knoll, and Jianwei Zhang. Multi-fingered robotic hand grasping in cluttered environments through hand-object contact semantic mapping. *arXiv preprint arXiv:2404.08844*, 2024.
- [141] Xiaoshuai Zhang, Rui Chen, Ang Li, Fanbo Xiang, Yuzhe Qin, Jiayuan Gu, Zhan Ling, Minghua Liu, Peiyu Zeng, Songfang Han, et al. Close the optical sensing domain gap by physics-grounded active stereo sensor simulation. *IEEE transactions on robotics*, 39(3):2429–2447, 2023.
- [142] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1327–1336, 2023.
- [143] Zheming Zhou, Xiaotong Chen, and Odest Chadwicke Jenkins. Lit: Light-field inference of transparency for refractive object localization. *IEEE Robotics and Automation Letters*, 5(3):4548–4555, 2020.
- [144] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large language models meet few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3065–3075, 2024.
- [145] Luyang Zhu, Arsalan Mousavian, Yu Xiang, Hammad Mazhar, Jozef van Eenbergen, Shoubhik Debnath, and Dieter Fox. Rgb-d local implicit function for depth completion of transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4649–4658, 2021.
- [146] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

Erklärung der Urheberschaft

Hiermit versichere ich an Eides statt, die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt zu haben. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Hamburg, 26. 01, 2026
Ort, Datum

Kaixin Bai
Unterschrift

A handwritten signature in black ink that reads "Kaixin Bai". The script is cursive and fluid, with the first letters of the first and last names being capitalized and prominent.

Erklärung zur Veröffentlichung

Ich erkläre hiermit mein Einverständnis zur Einstellung dieser Dissertation in den Bestand der Bibliothek.

Hamburg, 26. 01, 2026
Ort, Datum

Kaixin Bai
Unterschrift

A handwritten signature in black ink that reads "Kaixin Bai". The script is cursive and fluid, with the first letters of each word being capitalized and slightly larger than the rest of the letters.

