Robotic Manipulation under Transparency and Translucency from Light-field Sensing

by

Zheming Zhou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in The University of Michigan
2021

Doctoral Committee:

Professor Odest Chadwicke Jenkins, Chair
Professor Peter K. Allen, Columbia University
Associate Professor Dmitry Berenson
Assistant Professor David F. Fouhey
Associate Professor Robert Platt Jr., Northeastern University

Zheming Zhou

zhezhou@umich.edu

ORCID iD: 0000-0002-7549-1778

ACKNOWLEDGMENTS

The path towards this dissertation has been circuitous. Its completion is thanks in large part to my advisor Professor Odest Chadwicke Jenkins, who gave me guidance throughout my graduate studies and supported me to become an independent researcher. I would also like to thank my dissertation committee members, Professor Peter Allen, Professor Robert Platt, Professor Dmitry Berenson, and Professor David Fouhey for their insights and feedback in shaping my final dissertation.

I am fortunate to be a member of Lab4Progress. Many thanks to Zhiqiang Sui, Zhen Zeng, and Karthik Desingh for lending their time and expertise to brainstorm with me and give me help in every aspect of my Ph.D. life. I am so grateful to have the opportunity to work with my collaborators Kevin French, Xiaotong Chen, Tianyang Pan, Shiyu Wu, and Haonan Chang. Without their endeavors, all those works would not be possible. Thanks to Jana Pavlasek, Emily Sheetz, and Zhefan Ye for discussing and giving me feedback. I am also thankful to all staffs in the robotics institute and CSE for being super supportive during my graduate studies.

I would like to thank my significant other, Jiahui Liu, for always being there for me regardless of the physical distance we have. My heart is always with her and I know I wouldn't be where I am today without all of her unfailing love and understanding. In the end, I would like to express my deepest appreciation to my parents for their unreserved love and support in my life. Thank you both for giving me support and encouragement to chase my dream.

TABLE OF CONTENTS

Acknowledgments . . . . **ii**

List of Figures . . . . **v**

List of Tables . . . . **ix**

Abstract . . . . **x**

Chapter

1 Introduction **1**

1.1 Contribution 7

2 Background **9**

2.1 Robot Perception for Manipulation 10
2.1.1 Object 6-DoF Pose Estimation 10
2.1.2 Grasp Pose Detection 12
2.2 Light-field Perception 13
2.2.1 The Plenoptics and the Light-field 13
2.2.2 Light-field Cameras 15
2.2.3 Conventions, Visualization, and Applications 16

3 PMCL: Plenoptic Monte Carlo Object Localization for Robot Grasping under Layered Translucency **19**

3.1 Motivation 19
3.2 Related Work 21
3.2.1 Perception for Manipulation 21
3.2.2 Light Field Photography 22
3.3 Problem Formulation 23
3.4 Depth Likelihood Volumes 24
3.4.1 Formulation 26
3.4.2 Validity 28
3.4.3 Computation 30
3.5 Plenoptic Monte Carlo Object Localization 32
3.6 Results 33
3.6.1 Pose Estimation Results 35
3.6.2 Manipulation Results 36
3.7 Summary 37

4 GlassLoc: Plenoptic Grasp Pose Detection in Transparent Clutter 38
4.1 Motivation 38
4.2 Related Work 40
4.2.1 Grasp Perception in Clutter 40
4.2.2 Light Field Photography 42
4.3 Problem Formulation and Approach 42
4.4 Plenoptic Grasp Pose Detection Methods 43
4.4.1 Multi-view Depth Likelihood Volume 45
4.4.2 Reflection Suppression 47
4.4.3 Grasp Representation and Classification 49
4.4.4 Training Data Generation 50
4.4.5 Grasp Search 50
4.5 Results 52
4.5.1 Experimental Setup 52
4.5.2 Evaluation 54
4.6 Summary 55

5 LIT: Light-field Inference of Transparency for Refractive Object Localization 56
5.1 Motivation 56
5.2 Related Work 59
5.2.1 Pose Estimation for Robot Manipulation 59
5.2.2 Light-field Perception for Transparency 60
5.3 LIT Estimator 61
5.3.1 LIT Pipeline 61
5.3.2 Network Architecture 62
5.3.3 Particle Optimization 65
5.4 ProLIT Light-field Dataset 67
5.5 Experiments 68
5.5.1 Evaluation of Light-field Filters on Image Segmentation 69
5.5.2 Evaluation of Pose Estimation 69
5.5.3 Champagne Tower Demonstration 75
5.6 Summary 76

6 Conclusion and Discussion 78
6.1 Conclusion 78
6.2 Strengths and Limitations 79
6.2.1 Comparison between Different Sensor Modalities 80
6.2.2 Comparison between Proposed Methods 81
6.3 Future Work 83
6.3.1 Grasp Pose Detection under Cluttered Scenes Mixing with Transparent and Opaque Objects 84
6.3.2 Dense Surface Reconstruction for Reflective and Refractive Objects 84
6.4 Summary 85

Bibliography 86

LIST OF FIGURES

1.1 Modern robot in a variety of applications. (Top left) Ford robot performs package delivery task. (Top right) Diligent robot collects equipment for nurse in hospital. (Bottom left) Nino robot arms act as bartender to serve drinks. (Bottom right) Robots assemble parts in a factory. 2

1.2 Robotic manipulation works using RGB and RGB-D cameras. (Top left) Barrett WAM arm from CMU successfully grasps a bottle from RGB percpetion [1]. (Top right) ABB robot from UC Berkeley performs a bin-picking task using a depth camera [2]. (Bottom left) Multi-finger grasp poses are generated for a Barrett Hand from Columbia University with RGB-D input [3]. (Bottom right) Detected grasp poses for Baxter's hand from Northeastern University [4]. 3

1.3 Challenging scenes for current modern robot vision perception. (Top left) Drifting snow on a road can fool even the human vision system, causing problems for autonomous driving systems. (Top right) Unstructured domestic environments are hard for robots to reason and perform manipulation. (Bottom left) Transparent and translucent objects in a domestic environment will violate many of our assumption for robot vision perception. (Bottom right) Reflective industrial parts will cause a problem for robots' to perceiving them. 4

1.4 Robots perform manipulation for transparent and translucent objects. 5

1.5 Three sensor technologies that are already or can be potentially integrated into a robot perception system: (a) RGB-D camera, (b) ultrasonic ranging sensor, (c) light-field camera. 6

2.1 7D plenoptic function. (a) Parameter explanation of full plenoptic function. (b) Ideal eye with $360°$ field of view to capture all the light field impinging on the pupil. 14

2.2 4D parameterizations of light rays. 15

2.3 Three different light-field measuring devices. (a) Camera array. (b) Artificial compound eye. (c) Lytro first generation microlens camera. (d) Lytro Illum microlens camera. 16

2.4 Microlens-array based light-field camera. Light rays (light yellow) emitted from a point on the subject pass through the main lens and focus on the microlens array. The sensor behind each microlens capture the light rays from a specific direction. For example, red pixel records the light ray that travels the space whose path is painted red. 17

2.5 (Left) Example raw light-field image. A close look at the light-field image establish that the raw pixel is hard to interpret. (Right) sub-apertures images decomposed from raw image. Center image labeled with a blue boundary. . . 18

3.1 (Top row) a robot equipped with a wrist-mounted light field camera correctly localizing, grasping, and placing a clear drinking glass from a sink of running water. (Bottom row) this grasp is performed by Plenoptic Monte Carlo Localization on the observed center view image (left), which computes a Depth Likelihood Volume (middle) to localize the object (right) through generative inference. 21

3.2 An overview of our Plenoptic Monte Carlo Localization framework. A light field camera is installed on the end effector of the robot. After taking a single shot light field image of the scene, sub-aperture images are extracted (center view highlighted in red). The depth likelihood volume (DLV) is then computed as a 3D array of depth likelihoods over certain pixels (i, j) for depth d. The DLV is a comparator of color and gradient similarity between the center view and other sub-aperture images. Assuming a known geometry and region of interest, the six-DoF object pose is estimated by Monte Carlo Localization over a constructed DLV. 25

3.3 (Left) a scene with a transparent glass jar containing a ping-pong ball at rest on an opaque table. Along ray ρ_1, two surfaces (incident to the ball and the front surface of the jar) contributes to the pixel value, while along ray ρ_2 only one surface (incident to the table) appears. (Right) a planar top-down view of rays incident to the ball and the jar. The center view image plane, (i_1, j_{ρ_1}) receives a weighted sum of light rays reflected from both the glass surface point G_1 and the ping-pong surface P_1. Three example rays corresponding to ρ_2 (reflection of the surface from the glass jar), ρ_3 (reflection of the ping-pong ball through the glass), and ρ_4 (random ray) received by the image plane with incidence to scene points (G_1, P_2), (G_2, P_1), and (G_2, P_2), respectively. They indicate three depth d_g, d_p, d_i when form stereo pair with ray ρ_1. 26

3.4 Test objects for evaluating PMCL 6D pose estimation include: (to the left) opaque objects behind a partially opaque translucent surface (a stained glass window film), and (to the right) transparent objects. 32

3.5 Two types of scene for localizing object poses. (a-b) the scene with a single transparent object with an opaque but possible reflective background objects. (c-d) the scene with opaque objects behind translucent non-transparent surfaces 35

3.6 The percentage of correctly localized object under different thresholds for the object behind a stained glass panel and a single transparent object. In each plot, the translation error bound is fixed to 1cm (a) and 2cm (b). The x-axis is the decreasing dot product bound indicate the error between ground truth and estimated result. The y-axis is the percentage of correctly localized objects. For each type of scene, these plots consider two types of rotation error ranges: $[0, 1]$ in dot product space indicates for [90,0] in degrees, and the absolute value [-1,1] in dot product space indicates [180, 0] in degrees. 36

3.7 The robot executes pick-and-place action for the bleach bottle floating on the water. The bleach bottle is inside the aquatic tank so it is occluded by the stained glass from the camera view. 37

4.1 (Top) a robot using *GlassLoc* to pick up transparent objects from clutter and place on the tray. The robot is observing the scene using a light field camera. Grasp candidate is sampled in DLV (bottom left) and mapped to the world frame in the visualizer (bottom middle). The robot successfully picks up a transparent cup from the clutter (bottom right). 39

4.2 An overview of *GlassLoc* framework. A light field camera is mounted on the end-effector of the robot. After taking a set of light field observations by moving robot arms, sub-aperture images are extracted (center view is highlighted in red). The Depth Likelihood Volume (DLV) is then computed as a 3D volume of depth likelihoods over transparent clutter. Given gripper configuration, we can sample grasp poses in DLV and extract grasp features for the classifier to label whether the samples are graspable or not. 40

4.3 Example of DLV value calculation of two randomly sampled points (x_1, y_1, z_1) and (x_2, y_2, z_2) through examining the ray consistency in different view points. Each sample point corresponds to different pixel indices with depths in different center view plane I_{v_0} and I_{v_1}. 44

4.4 Example DLV feature image before (middle) and after (right) reflection suppression. The center view of part of raw observation is shown in (left). The intensity of pixel in the gray-scale image (middle and right) indicates the likelihood value. The high likelihood region caused by specular light is suppressed. 47

4.5 Training data generation procedure. (a) The glass cup is wrapped with opaque tape for depth sensor to get point cloud. (b) Grasp candidates are generated based on point-cloud-based method and local-to-world transform. (c) The glass cup is placed at the same pose to take multiple light field observations. (d) Grasp candidates generated from point cloud are mapped to DLV. 48

4.6 Training and testing objects for evaluating our *GlassLoc* algorithm. Two objects are used in training: wine cup and short cup (wrapped object for generating point cloud). Five objects are used in testing: wine cup, toothbrush holder, spoon, short cup, and tall cup. 51

4.7 Eight scenes for evaluating *GlassLoc* pipeline. We randomly choose a number of transparent objects from the test set and put them on the table for the robot to perform manipulation on. 53

4.8 The robot successfully picks and places all transparent objects in scene (g). Each column shows the pick and place action over one object in the scene. . . 53

5.1 Demonstration of our *LIT* pipeline. (Top row) Lytro Illum camera is mounted on the tripod and robot arm to capture the transparent objects in challenging environments. (Bottom row) final estimated poses are overlapped to the center view of the observed light-field image. 57

5.2 An overview of the *LIT* framework with the *ProLIT* dataset. (a) *ProLIT* contains 75,000 synthetic light-field images in training set and 300 real images with 442 object instances in testing set. (b) *LIT* estimator is a two-stage pipeline. The first stage takes light-field images as input and outputs transparent material segmentation and object center point prediction. The segmentation results are passed through a detection network to obtain object labels. In the second stage, for each predicted center point, we predict point depth likelihood by local depth estimation using Depth Likelihood Volume. The particle optimization samples over center points and converge to the pose that best matches the segmentation results. 58

5.3 Illustration of three light-field filters. Angular filter (AF) has dimension $1 \times 1 \times (H_a \times W_a)$ to capture features in angular pixels. sEPI and tEPI filters have sizes of $n \times n \times W_a$ and $n \times n \times H_a$ respectively, here n refers to kernel size. tEPI also has a dilation W_a. All features will be concatenated together after passing filters. 62

5.4 (Left) example synthetic light-field images rendered in three different environments. (Middle) example test images in different backgrounds and different pose configurations. (Right) results visualization by overlaying estimated poses to the original test images. 66

5.5 Comparison of 6D pose estimation results with respect to ADD-S and Accuracy Under Curve metric. 70

5.6 Example of RGB-D image pairs for opaque background. (Top row) color image of the background objects. (Bottom row) corresponding point cloud captured by the depth camera. 72

5.7 Comparison of 6D pose estimation results under opaque background with respect to ADD-S and AUC metric. 73

5.8 Example of RGB-D image pairs for reflective backgrounds. (Top row) color image of the background objects. (Bottom row) corresponding point cloud captured by the depth camera. 74

5.9 Comparison of 6D pose estimation results under reflective background with respect to ADD-S and AUC metric. 75

5.10 The robot is building a champagne tower by successfully picking and placing champagne cups on the table. The first row shows light-field observation (left) and pose estimation result from *LIT* (right). The following five rows show pick and place actions to finish the champagne tower. 77

LIST OF TABLES

4.1 Object-wise grasp performance . . . 52

4.2 Results of manipulation experiments for eight scenes. The first row shows the number of object in the scene. Number of manipulation runs shown in row two refers to the task runs for the scene. The object grasp percentage refers to successful picking ratio over all trials for each scene. . . . 52

5.1 Comparison of *LIT* and baseline methods on transparent material segmentation. The performance is quantified through global accuracy (gAcc), mean of class accuracy (mAcc), mean of Intersection over Union (mIoU), weighted IoU (wIoU), and mean BF (Boundary F1) contour matching score (mBFS). The definitions are detailed in [5]. 'AF only' here refers to the baseline method with only angular filters. . . . 69

5.2 Comparison of *LIT*, DOPE, AAE, PMCL, and ClearGrasp on transparent object pose estimation. The column headings wc, tc, gj, cc, and sb refer to the wine cup, tall cup, glass jar, champagne cup, and starbucks bottle objects, respectively. All columns, except for the last, refers to the area under the curve (AUC) for accuracy-threshold values for the symmetric objects metric (ADD-S), shown in Figure 5.5. . . . 71

5.3 Pose estimation results of *LIT*, DOPE, AAE, PMCL, and ClearGrasp under opaque background. . . . 74

5.4 Pose estimation results of *LIT*, DOPE, AAE, PMCL, and ClearGrasp under reflective background. . . . 76

ABSTRACT

From frosted windows to plastic containers to refractive fluids, transparency and translucency are prevalent in human environments. The material properties of translucent objects challenge many of our assumptions in robotic perception. For example, the most common RGB-D sensors require the sensing of an infrared structured pattern from a Lambertian reflectance of surfaces. As such, transparent and translucent objects often remain invisible to robot perception. Thus, introducing methods that would enable robots to correctly perceive and then interact with the environment would be highly beneficial. Light-field (or plenoptic) cameras, for instance, which carry light direction and intensity, make it possible to perceive visual clues on transparent and translucent objects.

In this dissertation, we explore the inference of transparent and translucent objects from plenoptic observations for robotic perception and manipulation. We propose a novel plenoptic descriptor, Depth Likelihood Volume (DLV), that incorporates plenoptic observations to represent depth of a pixel as a distribution rather than a single value.

Building on the DLV, we present the Plenoptic Monte Carlo Localization algorithm, **PMCL**, as a generative method to infer 6-DoF poses of objects in settings with translucency. PMCL is able to localize both isolated transparent objects and opaque objects behind translucent objects using a DLV computed from a single view plenoptic observation.

The uncertainty induced by transparency and translucency for pose estimation increases greatly as scenes become more cluttered. Under this scenario, we propose **GlassLoc** to localize feasible grasp poses directly from local DLV features. In GlassLoc, a convolutional neural network is introduced to learn DLV features for classifying grasp poses with grasping confidence. GlassLoc also suppresses the reflectance over multi-view plenoptic observations, which leads to more stable DLV representation. We evaluate GlassLoc in the context of a pick-and-place task for transparent tableware in a cluttered tabletop environment.

We further observe that the transparent and translucent objects will generate distinguishable features in the light-field epipolar image plane. With this insight, we propose Light-field Inference of Transparency, **LIT**, as a two-stage generative-discriminative refractive object localization approach. In the discriminative stage, LIT uses convolutional neural networks to learn reflection and distortion features from photorealistic-rendered light-field images. The learned features guide generative object location inference through local depth estimation and particle optimization. We compare LIT with four state-of-the-art pose estimators to show our efficacy in the transparent object localization task. We perform a robot demonstration by building a champagne tower using the LIT pipeline.

CHAPTER 1

Introduction

One hundred years following the creation of the word ”Robot,” we are gradually achieving the promise of the robot with its definition early works of fiction – an artificial agent with actuators that can perform a variety of tasks and effect changes to the real world. To engage in physically interaction with the world, robots must be able to perceive the environment and perform correct manipulation action in the environment. Figure 1.1 shows several representative scenarios where robots perform tasks in the real world. It is not hard to imagine the critical role of vision perception in all these applications (e.g., the robot needs to detect where to place the delivered items; the robot needs to localize which object needs to be picked from the bin). The integration of vision perception into robotic manipulation can date as far back as 1963, when a binary robot vision system was developed to assist the robot to do obstacle avoidance [6]. Nowadays, a robot's vision perception is dominated by the RGB-D camera (shown in Figure 1.5 (a)) – a hybrid vision system with a conventional RGB camera (passive light sensor) and a structured light depth camera (active light sensor). The RGB camera provides textured information of the environment, which has advantages in completing the task of object detection [7, 8, 9, 10, 11] and semantic segmentation [12, 13, 14, 15, 16] but with the prerequisite of reliable and distinguishable textures on the object surface. On the other hand, the depth camera is not relying on the texture information on the object surface but requires the observable structure light patterns (structure-light based) or receivable outgoing measurement light (time-of-flight

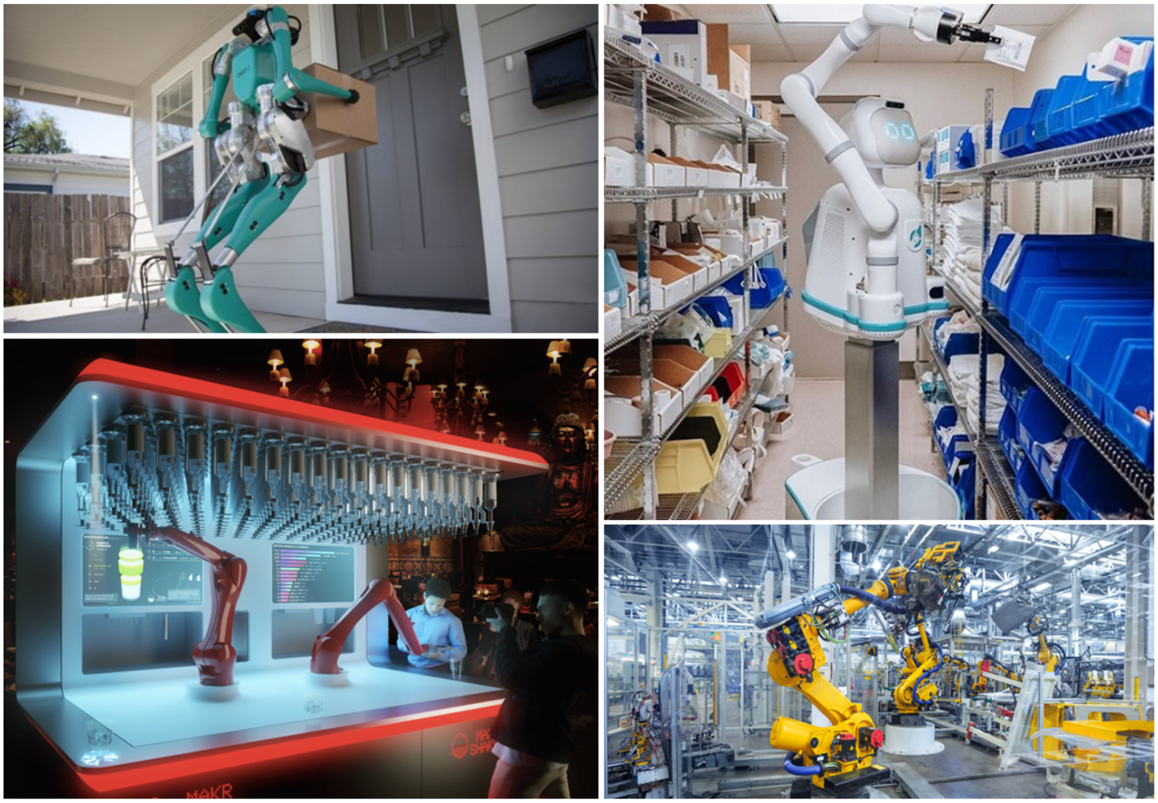


Figure 1.1: Modern robot in a variety of applications. (Top left) Ford robot performs package delivery task. (Top right) Diligent robot collects equipment for nurse in hospital. (Bottom left) Nino robot arms act as bartender to serve drinks. (Bottom right) Robots assemble parts in a factory.

based). In this regard, the depth camera gives an environment 3D structure, which is more suitable for the task of 3D registration [17, 18, 19, 20, 21] and grasp pose detection in 3D space [22, 23, 24, 25, 26], for example. Figure 1.2 has shown several representative robotic manipulation research using RGB and depth sensing. Early research [1, 22, 23] uses a single RGB camera to infer robot grasp poses as geometric primitives such as points, lines, and rectangles. Those representations are relatively straightforward during the inference step but pose restrictions to the robot when using dexterous grippers or performing elaborated tasks (e.g., assembly) that require high localization accuracy. More recent works leverage the advantages from both RGB and the depth camera, enabling robots to generate target grasping for more complicated grippers [3, 27], manipulate different kinds of objects in cluttered environments [2, 4, 28, 29], and perform sophisticated tasks that require long-

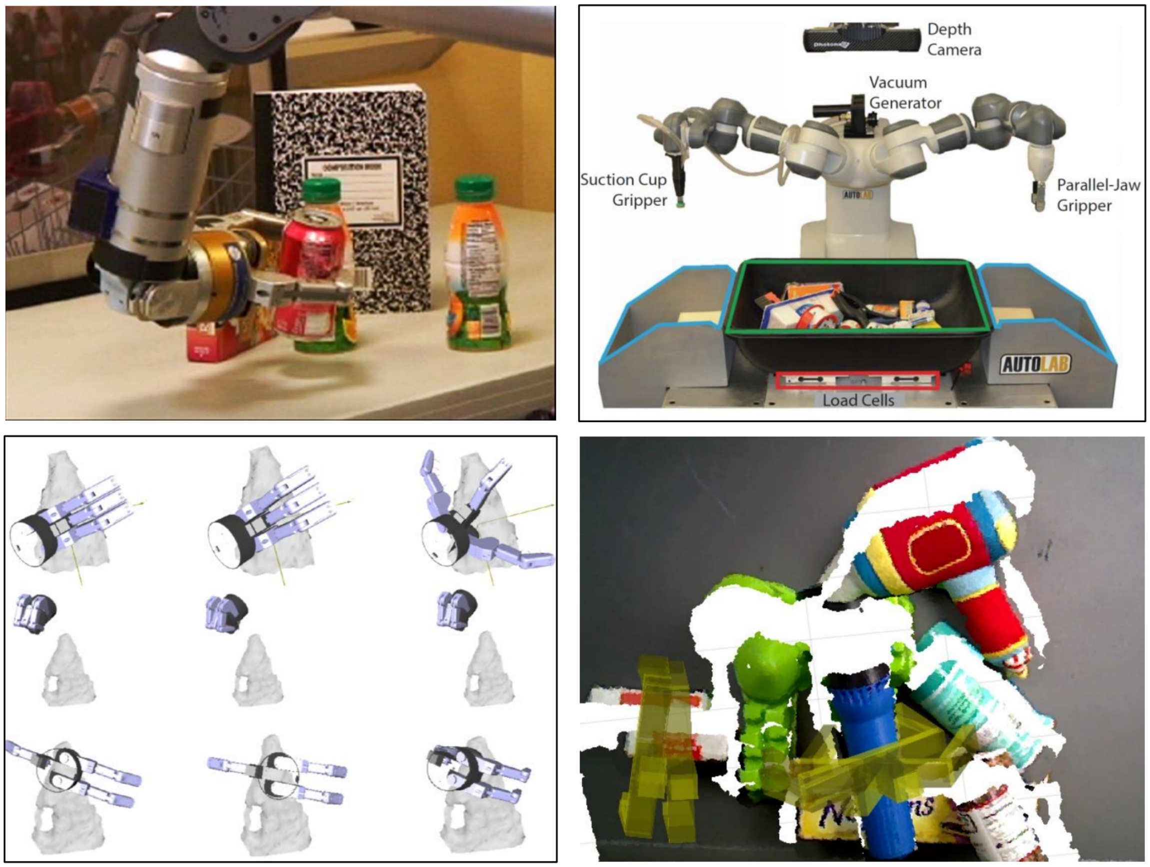


Figure 1.2: Robotic manipulation works using RGB and RGB-D cameras. (Top left) Barrett WAM arm from CMU successfully grasps a bottle from RGB percpetion [1]. (Top right) ABB robot from UC Berkeley performs a bin-picking task using a depth camera [2]. (Bottom left) Multi-finger grasp poses are generated for a Barrett Hand from Columbia University with RGB-D input [3]. (Bottom right) Detected grasp poses for Baxter's hand from Northeastern University [4].

horizon and active perception [30, 31]. Those works have shown that two types of sensor – RGB and depth – provide complementary advantages in supporting a great deal of robot manipulation research and many applications in robotic manipulation.

However, the robot vision perception system is still far from perfect. Figure 1.3 illustrates several challenging scenes for current robot applications. In this dissertation, we are dealing particularly with transparent and translucent objects, which violate many of the underlining assumptions we made for the success of the RGB-D camera. Translucent and transparent objects are very common in domestic environments, making their appearance inevitable in robot manipulation tasks. Our objective is to enable robots to correctly

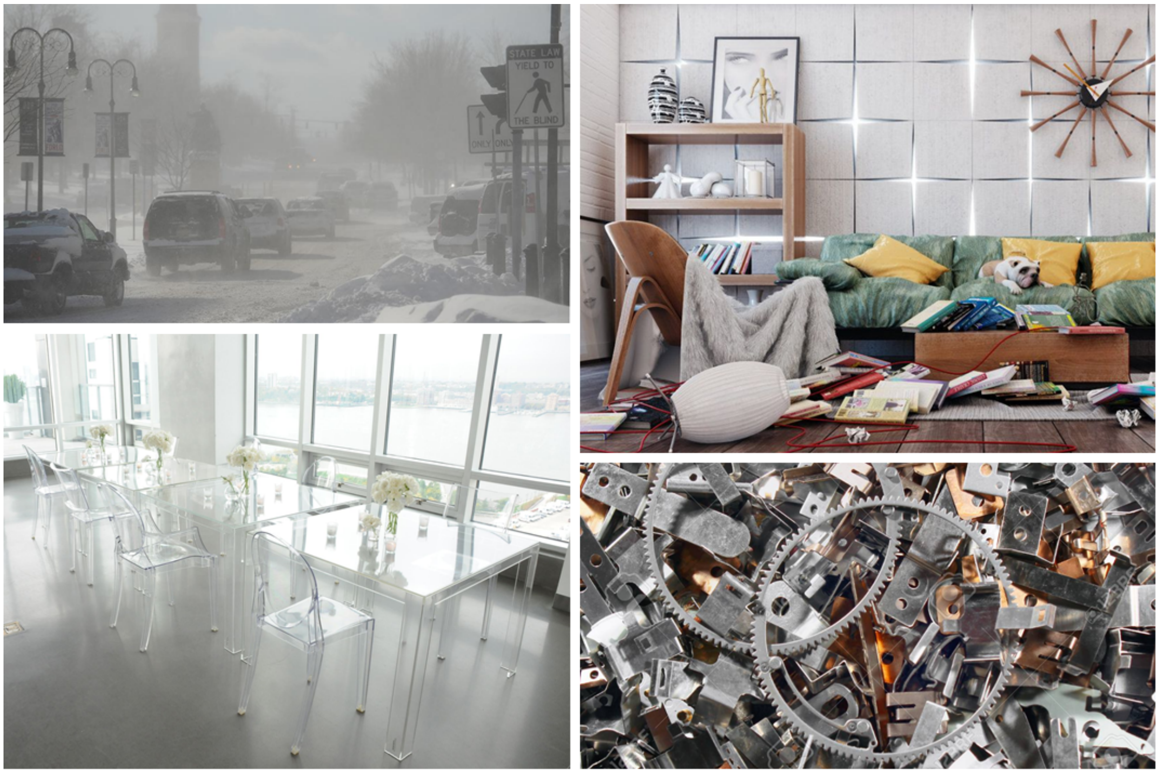


Figure 1.3: Challenging scenes for current modern robot vision perception. (Top left) Drifting snow on a road can fool even the human vision system, causing problems for autonomous driving systems. (Top right) Unstructured domestic environments are hard for robots to reason and perform manipulation. (Bottom left) Transparent and translucent objects in a domestic environment will violate many of our assumption for robot vision perception. (Bottom right) Reflective industrial parts will cause a problem for robots' to perceiving them.

perceive and perform interaction under transparent and translucency (Figure 1.4). The vision perception challenges carried by these objects are mainly two fold. First, for conventional RGB cameras, their texture information is highly environment dependent. The non-Lambertian surface of these objects encodes environment lighting conditions and background appearance. For instance, transparent surfaces will produce specularity and project distorted background texture on their surfaces due to refraction. Second, transparent and translucent objects are almost invisible to the active light sensors. For the structure-light camera, the projected patterns will transmit through those materials which cannot be correctly decoded for depth calculation. For the time-of-light camera, the outgoing measuring lights are very easily refracted by the transparent and translucent material, which can result

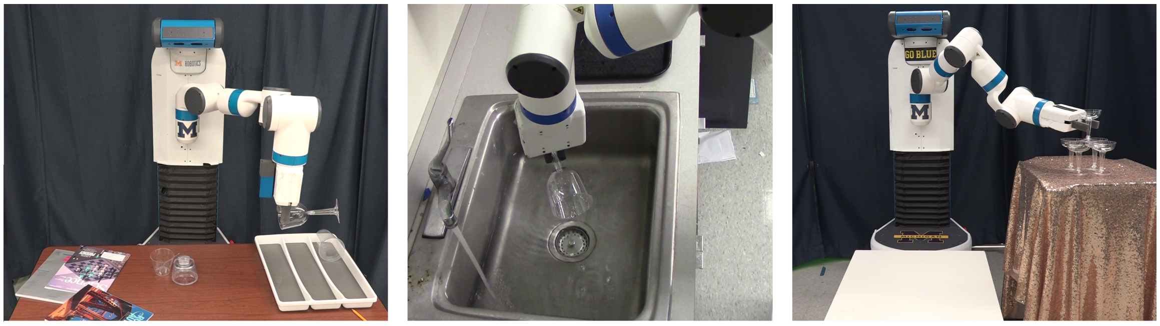


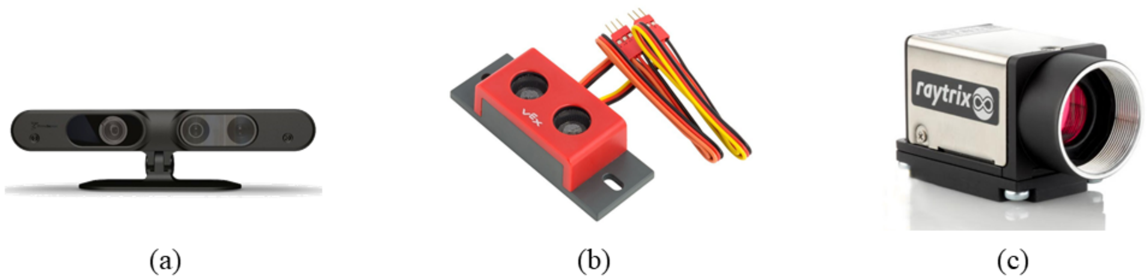
Figure 1.4: Robots perform manipulation for transparent and translucent objects.

in incorrect measurements.

While we assume that transparent and translucent are defined in the light media domain, one could argue that sound-based sensors such as sonar (shown in Figure 1.5 (b)) could tackle the problem. Unfortunately, ultrasonic sensors pose a great many more shortcomings for the purpose of robotic manipulation tasks. First, the poor directionality of the sensor results in large measurement errors (> 10 cm) in the real world [32]. Since most of the manipulation targets in the domestic environment are at the same level of the error, the sensor measurement is not reliable for the manipulation task. Second, reflections and specularities also occur for the ultrasonic data, when the angle between the wave front and the normal of the smooth surface is large [32, 33], further corrupting the reliability of the ranging data.

These observations underline the challenges of robotic perception for manipulation under transparency and translucency. Research in this area is still limited, with most of the existing work [34, 35] taking the invalid reading from the depth camera as the descriptor for the transparent and translucent objects. This binary description easily reaches its bottleneck in a real-world scenario when the background cannot provide reliable depth or the scene becomes cluttered.

To address this problem, we turn to the third type of sensor depicted in Figure 1.5 (c), the light-field camera. This passive light sensor shares many advantages with the conventional RGB camera, but measures a rich, 4D light-field with light intensity and direction that



(a) (b) (c)

Figure 1.5: Three sensor technologies that are already or can be potentially integrated into a robot perception system: (a) RGB-D camera, (b) ultrasonic ranging sensor, (c) light-field camera.

implicitly encodes both the texture and geometry. We argue that the light-field sensor is a well fit for the robotic perception of manipulation problem under transparency and translucency for two main reasons. First, with a 4D light-field of the environment, the specularity and distortion from transparent and translucent surface will establish distinguishable features in the light direction space which is not available for the conventional RGB camera. Second, because the different light directions in a 4D light-field can be treated as different viewpoints looking at the same scene, we can also infer the 3D depth information of the objects. Like the function of the RGB-D camera in the manipulation task for opaque objects, the light-field camera is well-posed to handle perception challenges under transparency and translucency. Some initial works conducted by Goldluecke et al. [36, 37] have already shown promising results in recovering multiple layers of transparency and translucency in a simulated environment. Building on those ideas, we propose to incorporate light-field perception for the robotic manipulation tasks under transparency and translucency.

In this dissertation, we exploit the potentials and performance of using light-field perception to recognize and localize transparent and translucent objects for robot manipulation. Aside from pushing the robotic manipulation boundary to more real-world objects, we expect that light-field sensors would accelerate the next generation of RGB-D perception for the next level of autonomy in a robot platform.

1.1 Contribution

This dissertation introduces light-field perception for robotic manipulation under transparency and translucency. Based on light-field observation, we focus on enabling the robot to perform grasping actions through inferring grasp pose over transparent and translucent objects.

A formal formulation of the problem will be: Given a set of light-field observations Z and corresponding transformation T to the robot coordinate frame O_r, we wish to find a set of feasible grasp pose G in the robot coordinate frame that can be planned and executed by the robot.

In this dissertation, we have proposed three approaches to address this problem under different scenarios with different scene clutterness, task complexity, and computation cost.

Plenoptic Monte Carlo Localization (PMCL) uses generative methods to infer the single object 6-DoF pose under structured environments. The generative inference is highly repeatable but computationally expensive. GlassLoc is proposed as a discriminative approach for finding object grasp pose under cluttered environments. GlassLoc is more computationally efficient but aims for primitive task like pick-and-place action. LIT leverages the power of a generative and discriminative approach which uses a deep neural network to semantically segment the transparent objects and then performs local generative inference. LIT is able to estimate the object 6-DoF pose with fast speed under moderately unstructured environments. More specifically, this dissertation's contribution includes:

1. **PMCL: Plenoptic Monte Carlo Localization** (Chapter 3). In PMCL, we introduce for the first time the light-field descriptor Depth Likelihood Volume (DLV), which can represent layered translucent environments by representing the depth as a distribution rather than a determinate depth value. We also show that the level of transparency of the object surface is proportional to the likelihood of that point. Building on DLV, we introduce a generative optimization method to iteratively estimate the

object 6-DoF pose . The maximum likelihood estimation is then used to transform the local defined grasp pose to the real environment for robot manipulation.

2. **GlassLoc: Plenoptic Grasp Pose Detection** (Chapter 4). To grasp transparent objects in a cluttered environment, instead of estimating each object's pose, we localize feasible graspable poses in the scene. In GlassLoc, we introduce the multi-view DLV construction with reflection suppression for more reliable DLV representation. We further introduce a convolutional neural network as a discriminative grasp pose classifier. The network uses the 2D projection of local DLV features as input to label grasp pose with confidence. For every graspable pose, we verify its feasibility against robot motion planning pipeline and choose the most confident one for grasping.

3. **LIT: Light-field Inference of Transparency** (Chapter 5). Real world manipulation tasks are not limited by grasping and require more fast inference. In LIT, we introduce a generative-discriminative two-stage framework for fast transparent object detection, segmentation, and localization. We introduce an LIT network that regresses the transparent segmentation and object centers. The network is trained on the pure synthetic data generated by customized photorealistic light-field rendering environments. The segmentation and object centers serve as a prior for the second stage generative inference. At this stage, the optimization algorithm uses object centers with local DLV to initialize the sampling, and segmentation serves as the optimization target. The localized objects with corresponding object labels are then sent to the robot for the manipulation task.

CHAPTER 2

Background

This dissertation aims to tackle light-field perception for the pick-and-place manipulation problem specifically under transparency and translucency. As mentioned in the previous chapter, the surface reflection and refraction behaviors of transparent and translucent objects are difficult to characterize using conventional RGB or RGB-D vision sensors. In contrast, a light-field camera, with its ability to capture both the direction and intensity of light, makes it possible to describe reflection and refraction in a 4D light-field space. Early research by Goldluecke et al. [36, 37] explores this possibility by investigating the light-field epipolar line patterns in recovering reflective surfaces and objects behind translucent surfaces. Their results have shown that reflection and refraction patterns in epipolar images can be identified and analyzed accurately with higher order structure tensors. The application of light-field sensing in robotics is still limited, however. Recent work by Oberlin and Tellex [38], for example, uses an RGB camera to mimic the light-field imaging process by time-lapse capturing from multiple viewpoints. This dissertation work builds on those insights while extending it further to more challenging real-world scenarios for robotic manipulation.

In this chapter, we first cover the related works of robotic perception for manipulation in general. We then introduce the background of light-field sensing with its applications in robotics.

2.1 Robot Perception for Manipulation

Beginning with the first general-purpose robot, Shakey-the-robot, the current robot platform is able to reason about its own actions through perception feedback. In particular, in the robotic manipulation field, we have witnessed many successful manipulation pipelines building on the modern robot platform. For example, Ciocarlie et al. [39] propose a robust pick-and-place pipeline for structured environments with a PR2 robot. On an HERB robot platform, Collet et al. [40, 1] create the MOPED perception framework for localizing objects in RGB images. Papazov et al. [41] take a bottom-up approach to enable an LWR-III robot to perform a sequential scene estimation for object manipulation. Similarly, Sui et al. [42, 28] leverage discriminative and generative methods to build the SUM robot manipulation pipeline on a Progress Fetch robot. With a more generalized perception knowledge base, the KnowRob system [43] provides a task-oriented manipulation by leveraging different sources of knowledge for a TUM Rosie robot.

Of all the manipulation tasks, grasping is the most fundamental but also the important. In this section, we will focus on vision perception works that target the problem of robot grasping. The related works fall into two main categories:

- **Object 6-DoF Pose Estimation** aims to recover the 6-DoF pose of the target objects in the scene. Existing methods can be further divided into a matching-based approach and an end-to-end learning-based approach.
- **Grasp Pose Detection (GPD)** aims to directly detect feasible grasp poses for the observations. Different from the pose estimation method, which relies on *matching*, the focus of the GPD is to *search* local features that will fulfill specific requirements.

2.1.1 Object 6-DoF Pose Estimation

The objective of 6-DoF pose estimation is to transform a predefined action in an object local frame to current observations for robot manipulation. This strategy is widely used in robotic simulation systems [44, 45, 46]. The related works for pose estimation fall into two categories: a matching-based approach and an end-to-end learning approach.

For the matching-based approach, the objective is either to match the extracted descriptors or the entire object model. To match the extracted descriptors, the Iterative Closet Point (ICP) [17] is a huge family containing both local features and global features for 3D registration. For example, the Generalized-ICP [18] uses surface normal for plane-to-plane matching; NICP [47] combines normal and surface curvature for dense point cloud registration; CICP [48] clusters points into a united bin and matches them with the surface normal; and GoICP [49] uses branch and bound optimization to change the standard ICP into a global optimization problem. Other features, such as the point-pair feature [50] and a congruent set [20], are also used with a voting scheme or iterative-based methods to achieve the descriptor matching.

Aside from the descriptor, an entire object model can also be directly registered to the observations, which is often referred to as *analysis-by-synthesis*. Match and Refine [51] tries to find the best match between observation 2D image and rendered image. PERCH [52] uses tree search to find the best match between a model and a target. D2P [53] extends the work from PERCH but uses A* search to find the matches for multiple objects in the scene. APF [54] uses Monte Carlo optimization to localize the given object by comparing the matching score between the rendered depth image and the observed depth image.

The matching-based approach is explainable but relies heavily on hand-crafted features or precious object models. Most of these approaches, while repeatable, are also computationally expensive, particularly when the model or descriptor is dense or large. In contrast, deep learning methods have been a viable approach for performing accurate and fast inferences for this problem. Early methods [55, 56] focus on the use of convolutional neural networks to first perform recognition or completion tasks that then follow with pose estimation. Recent work tends to directly regress the 6-DoF pose from the observation. Xiang *et al.* [57] proposed PoseCNN to recognize and estimate objects and their 6D poses by decoupling translation and rotation separately in a neural network structure. Other end-to-end method methods have explored using synthetic data in training [58, 59], pixel-wise voting over keypoints [60, 61], and residual networks to iteratively refine object poses [62, 63].

2.1.2 Grasp Pose Detection

Grasp Pose Detection (GPD) tries to characterize grasp poses based on the local geometry or appearance features directly from observations. As such, grasping representation is at the heart of the GPD approach.

In the early work of robotic grasping, a full knowledge of the 2D or 3D model of the target is often required to investigate the grasp representation such as force-closure [64] or form-closure [65], to determine whether a grasp pose is feasible. But in the real world, partial observation often occurs which means the full object observation is unavailable to the grasping system. To deal with partial observation, some works [66, 67, 68] represent the objects by primitive geometry shapes such as spheres, cones, boxes, edges, and contours. However, these representations still require access to the object models for designing these simplified shapes.

Instead of building a representation on the objects, other research tries to directly represent what is graspable in the scene. Some initial tries [69, 70] represent grasp poses using one point or pair of points. This representation, however, has limitations in modeling different grippers and also introduces strong ambiguities in feature extraction. To tackle that problem, more recent works [71, 22, 23, 72] have represented the grasp poses as oriented rectangles in RGB-D observations. Then, given several manually labelled grasp candidates, the system will learn to predict whether a sampled rectangle is graspable or not. While this representation is popular even in the most recent research, a major constraint is that the approaching direction of the generated grasp candidates must be orthogonal to the RGB-D sensor plane. Fischinger and Vincze [24] tried to lessen this constraint by integrating heightmap-based features. They also designed a heuristic for ranking the grasp candidates in a clutter bin setting. Another approach was taken by ten Pas and Platt [25]. Instead of detecting the grasp pose in 2D, they directly detected grasp poses in $SE(3)$ space by estimating curvatures and extracting handle-like features in local point cloud neighborhoods. Gualtieri et al. [26] proposed more types of local point cloud features for grasp representation and projected those features to 2D image space for classification. While these 3D representations are often more reliable under changes in lighting, they rely on the quality of the point cloud. In this dissertation, we extend these ideas to transparent and translucent objects, which don't have a reliable texture or 3D point cloud.

2.2 Light-field Perception

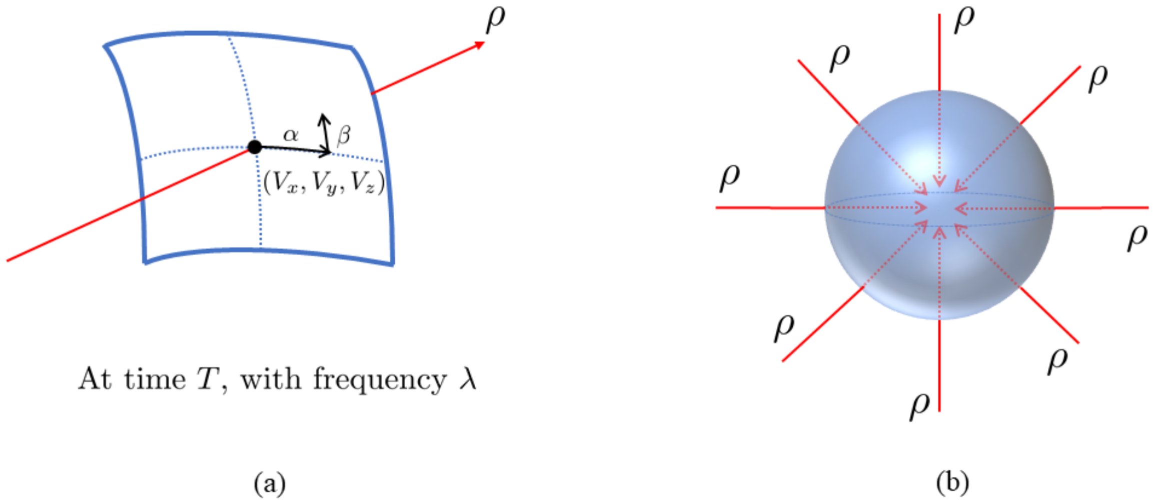
2.2.1 The Plenoptics and the Light-field

It is a common practice to describe light using RGB colors in the context of a 2D image. But in the real world, a light ray is described in a much higher dimensional space using the so-called plenoptic function [73]. *plenoptic* is a combination of two Latin words *plenum*, meaning "full" and *optics*. More specifically, plenoptic function describes a light ray ρ using Equation 2.1 with seven variables representing seven dimensions of the light – one dimension of time (T), one dimension of frequency (λ), two dimensions of direction (α, β), and three dimensions of space (V). The more illustrative explanation of plenoptic function is shown in Figure 2.1(a).

$$\rho = P(T, \lambda, \alpha, \beta, V_x, V_y, V_z) \tag{2.1}$$

Instead of representing our world with textures and geometry, the plenoptic function presents a new way to represent the world as a volume flow with light rays. That means that the action of visual perception is no longer one of reaching out to objects with rays but of measuring all the light rays that flow through the ideal eyes (depicted in Figure 2.1(b)).

Of course, in the real world, it is virtually impossible and pretty much unnecessary to



(a)

(b)

Figure 2.1: 7D plenoptic function. (a) Parameter explanation of full plenoptic function. (b) Ideal eye with $360°$ field of view to capture all the light field impinging on the pupil.

have such a device to capture and save all the light flow in space. In 1996, Levoy and Hanrahan [74] proposed an approximation of the plenoptic function, called the *light-field*, for the purpose of image-rendering by reducing the plenoptic function to a four dimensional space: two dimensions of direction (u, v) and two dimensions of space (s, t). The light-field representation discards time in favor of static scenes, and frequency is simply replaced by the RGB channel, which does not figure into the dimensions of the plenoptic function (*Note*: We characterize an image captured by a conventional camera as 2D rather than 5D). More importantly, the light-field representation reduces the space dimension from 3D to 2D by embedding the light ray 3D space information into the intersection of the light ray with two parallel planes, as illustrated in Figure 2.2. More specifically, the 4D light-field parameterization can be represented as:

$$\rho = L(s, t, u, v) \tag{2.2}$$

In this dissertation, we use plenoptic and light-field interchangeably to refer to the 4D representation of the light with two dimensions in space and two dimensions in direction.

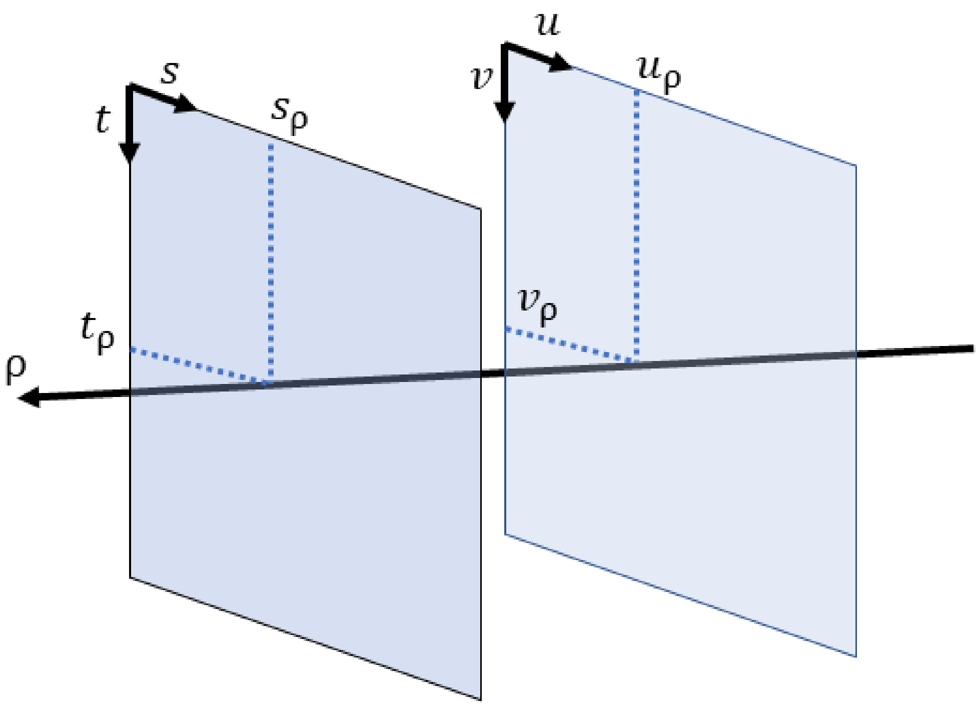


Figure 2.2: 4D parameterizations of light rays.

2.2.2 Light-field Cameras

The different kinds of light-field sensors that are available off the shelf can be categorized as either camera-array-based or microlens-based. Both types of cameras originate from the conventional RGB camera. We can first investigate what the conventional RGB camera tells us about the plenoptic function. For a conventional RGB camera, if we assume that the (u, v) plane in Figure 2.2 is the main lens plane and the (s, t) plane is the sensor plane, a specific sensor will capture all the light rays going through the main lens. That said, a 2D RGB image is then an integral of the plenoptic function in all possible combinations of (u, v) and can be represented as $\int_u \int_v L(s, t, u, v) du\, dv$. Since the (u, v) plane determines the light direction, the conventional RGB camera preserves only the intensity of the light.

Then it is straightforward to think that if we have an array of cameras in the space, with each camera's main lens at a different position (u_c, v_c), then this camera system become a light-field camera, as shown in Figure 2.3 (a). Among all the light-field measuring devices,

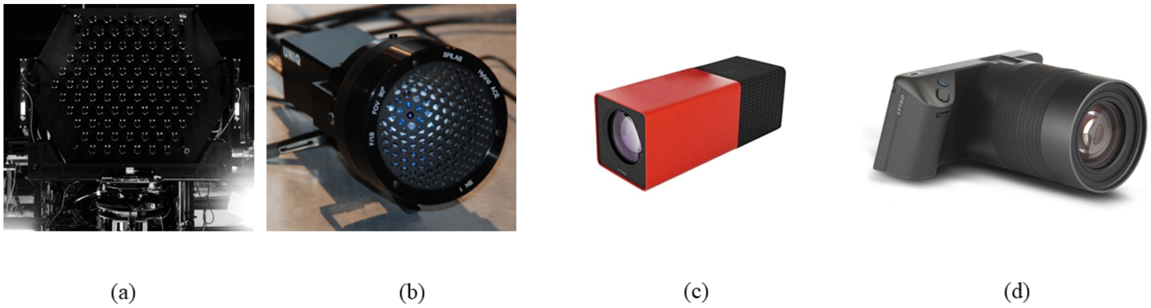

(a) (b) (c) (d)

Figure 2.3: Three different light-field measuring devices. (a) Camera array. (b) Artificial compound eye. (c) Lytro first generation microlens camera. (d) Lytro Illum microlens camera.

the camera array is the easiest to understand. In nature, some insects also have developed a similar structure for their eyes. Figure 2.3 (b) shows an example of an artificial compound eye that mimics the eyes of a fly.

The problem, however, is that the camera-array-based light-field cameras are always too large to use in robotic applications. Inspired by the 4D plenoptic function, Ng [75] proposes a microlens-based camera to capture the light field. The proposed design adds a microlens array between the main lens and photo sensors. Considering the plenoptic function in this situation, the (s, t) plane aligns with the microlens array with each microlens located at (s_c, t_c). Behind each microlens is a group of photo sensors that captures a unique configuration of (s, t, u, v). A more illustrative explanation is depicted in Figure 2.4.

The biggest advantage of the microlens light-field camera is that it can maintain its size at the same level as a single conventional RGB camera while capturing the light-field. But it also brings the high requirements to optical components of the camera. Figure 2.3 (c)(d) show commercial-level microlens array light field cameras. Overall, all the light-field cameras can be categorized into the class of generalized and computational cameras [76].

2.2.3 Conventions, Visualization, and Applications

An important property of a camera is resolution. The resolution of a light-field camera is defined on both the (u, v) and (s, t) plane, labeled angular resolution and spatial resolution

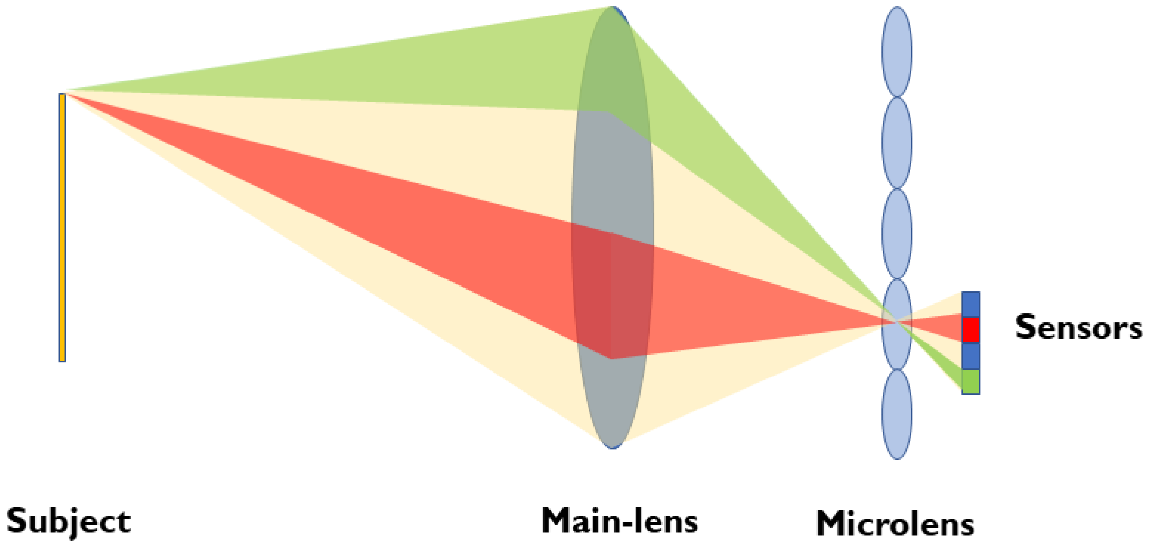


Figure 2.4: Microlens-array based light-field camera. Light rays (light yellow) emitted from a point on the subject pass through the main lens and focus on the microlens array. The sensor behind each microlens capture the light rays from a specific direction. For example, red pixel records the light ray that travels the space whose path is painted red.

respectively. Angular resolution indicates the number of light directions sampled for a single point and is determined by the number of photosensors behind each microlens. Spatial resolution describes the number of points sampled for a real scene, which is determined by the size of the microlens array. There is always a tradeoff between the spatial and angular resolution of the camera, with the physical size being nearly proportional to the total pixel number. For example, a Lytro first generation camera has a total of 11 Megapixels, 10 × 10 angular resolution, and 328 × 328 spatial resolution, while a Lytro Illum camera is approximately four times that of the first generation, with a total of 40 Megapixels, 14 × 14 angular resolution, and 552 × 385 spatial resolution. Because of the data size of one light-field image, most low-end commercial light-field cameras support only a single shot image rather than a light-field video.

A raw light-field image captured by a microlens camera is difficult to interpret (Figure 2.5 (Left)) so it is common practice to decompose it into 2D image slides, which we called sub-aperture images (Figure 2.5 (Right)). Each of the 2D image slides is a query of plenoptic function with a certain index in the u, v plane; more specifically, a sub-aperture

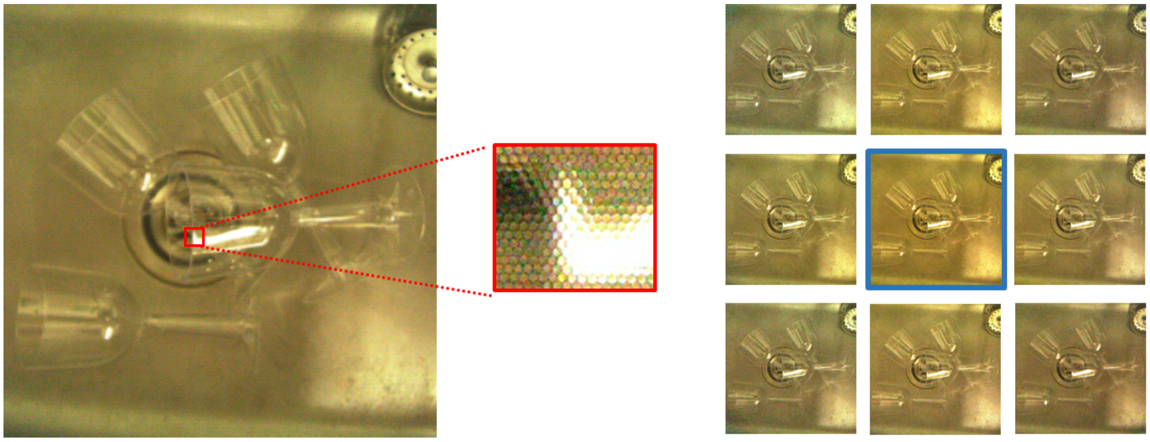


Figure 2.5: (Left) Example raw light-field image. A close look at the light-field image establish that the raw pixel is hard to interpret. (Right) sub-apertures images decomposed from raw image. Center image labeled with a blue boundary.

image can be represented as $L(s, t, u = u_i, v = v_i)$. The sub-aperture image located at the center of the u, v plane is called the center view sub-aperture image.

Research related to the light-field has been steadily increasing in recent years, but the trend seems to be inversely proportional to the level of integration: highest in computational imaging, lower in computer vision, and lowest in robotics [77]. In the robotics field, light-field is deployed primarily for motion planning and odometry [78, 79, 80], a few in underwater image dehazing [81], and some for street sign recognition in extreme weather conditions [82]. In this dissertation, we further explore the potential of light-field sensing for robot manipulation under transparency and translucency.

CHAPTER 3

PMCL: Plenoptic Monte Carlo Object Localization for Robot Grasping under Layered Translucency

3.1 Motivation

From frosted windows to plastic containers to refractive fluids, translucency is prevalent in human environments. Translucent materials are commonplace in our daily lives and households, but remain an open challenge for autonomous mobile manipulators. Various previous methods [83] have enabled robots to navigate autonomously in the presence of glass and transparent surfaces. When handling objects, however, robot perception systems must contend with a wider diversity of objects and materials.

Translucent objects, in particular, break many of our assumptions in robot sensing and perception about opacity and transparency. For example, existing six-DoF pose estimation methods [28] [52] often heavily rely on RGB-D sensors to reconstruct 3D point clouds. Such sensors are typically ill-equipped to handle the uncertainty caused by the reflection and refraction properties of translucent materials. As a result, translucent objects are often invisible to the robots for the purposes of dexterous manipulation.

An important topic related to this problem is multi-layer stereo depth estimation as studied by Borga and Knutsson [84]. These findings establish that even transparent surfaces will emit their own patterns. When the pattern from translucent surfaces mixed with patterns from Lambertian surfaces, the result will be multi-orientation epipolar image lines in multi-view stereo images. These stereo images can record light fields and equip a robot with the ability to identify surfaces with transparent properties.

Light field photography offers considerable potential for robot perception in scenes with translucency. For example, Oberlin and Tellex [38] found that a high-resolution camera on the wrist of a robot manipulator can capture light fields for a static scene. By moving the robot end-effector in a designed trajectory, this time lapse approach to capture light field was demonstrated as capable of manipulating transparent and reflective objects. We now aim to extend similar ideas to the larger class of translucent materials, along with explicit pose estimation for more purposeful object manipulation.

In this chapter, we propose Plenoptic Monte Carlo Localization (PMCL) as a method for six-DoF object pose estimation and manipulation under uncertainty due to translucency. Our PMCL method uses observations from light field imagery collected by a Lytro camera mounted on the wrist of a mobile manipulator. These observations are used to form a new plenoptic descriptor, called Depth Likelihood Volume (DLV). The DLV is introduced to describe a scene with multiple layers of depth due to translucency. The DLV is then used as a likelihood function with a Monte Carlo localization method for our PMCL algorithm to estimate object poses.

We demonstrate the efficacy of PMCL with DLV for manipulation in translucency with an implementation using a Michigan Progress Fetch robot. We present results of object localization and grasping for two situations: transparent objects in transparent media (Figure 3.1) and opaque objects diffusely occluded by translucent media.

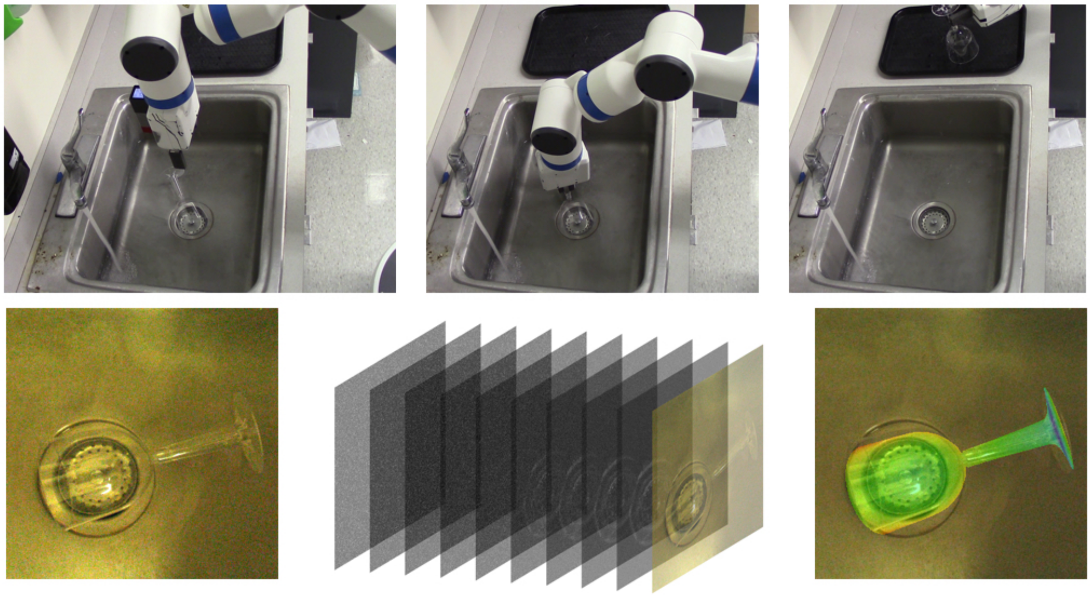


Figure 3.1: (Top row) a robot equipped with a wrist-mounted light field camera correctly localizing, grasping, and placing a clear drinking glass from a sink of running water. (Bottom row) this grasp is performed by Plenoptic Monte Carlo Localization on the observed center view image (left), which computes a Depth Likelihood Volume (middle) to localize the object (right) through generative inference.

3.2 Related Work

3.2.1 Perception for Manipulation

The problem of perception for manipulation remains challenging for robots working in human environments and the natural world. The presented concepts for PMCL build on a substantial body of work in this area, which we summarize briefly.

Ciocarlie et al. [39] proposed a robust pick-and-place pipeline for the Willow Garage PR2 robot. This pipeline segments and clusters points which comprise isolated opaque tabletop objects observed from an RGB-D sensor. For more cluttered environments, Collet et al. [40] proposed the MOPED perception framework for localizing objects by discriminatively clustering multi-view features in color images. Narayanan et al. [85] take a deliberative approach to infer the pose of objects in clutter from RGB-D observations. This work performs A* search over possible scene states using a discriminative algorithm for

3D pose estimation. Similar in its aims, Sui et al. [42, 28] have proposed generative models for scene inference and estimation. Such generative models combine object detection from neural networks with Monte Carlo localization algorithms in the scenario of object sorting on highly cluttered tabletops.

For transparent object perception, McHenry et al. [86, 87] have used reflective features from transparent objects for segmentation in a single RGB image. Lei at al. [88] segment out transparent objects by searching failure detection from laser rangefinding (LIDAR) combined with RGB image features. Methods by Phillips et al. [89] describe detection and estimation of rotationally symmetric transparent objects using edge features. Lysenkov et al. [35] perform six-DoF pose estimation of transparent objects based on a silhoutte model corresponding with invalid RGB-D depth measurements. Partial opacity from translucent materials can be problematic for such methods, where clear edge features become blurred due to diffuse reflection.

3.2.2 Light Field Photography

The contributions of this paper are founded upon models described by Levoy and Hanrahan [74] for understanding light fields and plenoptic functions. Their seminal paper covers the foundations of capturing light fields from digital imagery and using them to synthesize new viewpoints from arbitary camera positions. Building on this work, microlens-based light field photography [90, 75] has witnessed significant advancements in depth estimation, image refocusing, transparent object recognition, and surface reconstruction.

In computer vision, Maneo et al. [91] proposed “light field distortion features” to capture distortions and recognize transparent objects. Sulc et al. [92] separates diffuse color components from 4D light field imagery to suppress non-lambertian surface’s reflection. Wang et al. [93] introduced a light field occlusion model for accurate recovery of the depth information around the edge where occlusion occurs. Jeon et al. [94] overcome the narrow baseline problem of light field cameras based on the sub-pixel shift method. This method generates accurate depth images even when the displacement of two adjacent sub-aperture images is less than 1 pixel. Our presented methods for PMCL build directly upon ideas in recent work by Goldluecke et al. [36, 37] for 3D reconstruction in multi-translucent environments. This work proposes generating multi-orientation features observed in epipolar plane images generated by a light field imagery, with impressive results for 3D reconstruction in high translucency.

In robotics, Oberlin and Tellex [38] introduced a time lapse approach to capture light for pick-and-place localization with a Rethink Baxter robot. This work demonstrated compelling results for localizing grasp and placement points in scenes with transparency and reflection, which has been problematic for current sensors.

Our PMCL method shares similar aims with more general models of translucency in mind. Further, estimation of six-DoF object pose estimation by PMCL will allow for greater flexibility in planning and executing manipulation actions. We posit PMCL to be readily capable of object tracking from plenoptic observations, although such experiments are left for future work.

3.3 Problem Formulation

Given an input light field image observation Z, the purpose of six-DoF pose estimation is to infer the rigid transformation from an object's local coordinate frame $\mathcal{O}$ to the camera's coordinate frame $\mathcal{C}$. We assume as given the geometry of the target object o. Formally, we aim to find the maximum likelihood estimate for the object's pose q given o and a map representation m in 3D world coordinates:

$$\arg\max_{q} P(q|m, o) \tag{3.1}$$

The map m is often computed as a metric representation, such as a 3D reconstruction or point cloud. In the case of common RGB-D cameras, the map representation is a one-to-one mapping from locations in 3D space (x, y, z) into depth value d at pixel index (i, j) of a depth image. Such a one-to-one mapping assumes opacity in that the sensed depth at a particular pixel is due to light from only one object.

We propose the **Depth Likelihood Volume (DLV)** as an alternative one-to-many mapping to consider the likelihood of a pixel over multiple levels of depth. As the case for translucent objects, the DLV representation is advantageous in environments where multiple objects at more than one depth are responsible for the light sensed at a pixel. The DLV representation expresses m as the mapping:

$$m : \mathcal{M}_{\rho}(x, y, z) \rightarrow L(i, j, d) \tag{3.2}$$

where $\mathcal{M}_{\rho}(x, y, z)$ represents a 3D point (x, y, z) along a light ray ρ taken as input. The output $L(i, j, d)$ is the likelihood of light along the ray ρ emitted from depth d being received by pixel (i, j) in the image plane.

For our light field cameras, we assume the image plane is determined by the center view image of the sub-aperture images extracted from light field observation Z. d is discretized possible depths along light ray ρ. An overview of our approach to this problem is shown in Figure 3.2.

3.4 Depth Likelihood Volumes

Before presenting our PMCL method for pose estimation, we first define the Depth Likelihood Volume. We describe the properties of the DLV for distinguishing multiple depths at a given point in an image due to translucency. The construction of the DLV and its use for pose localization is described in the following section.

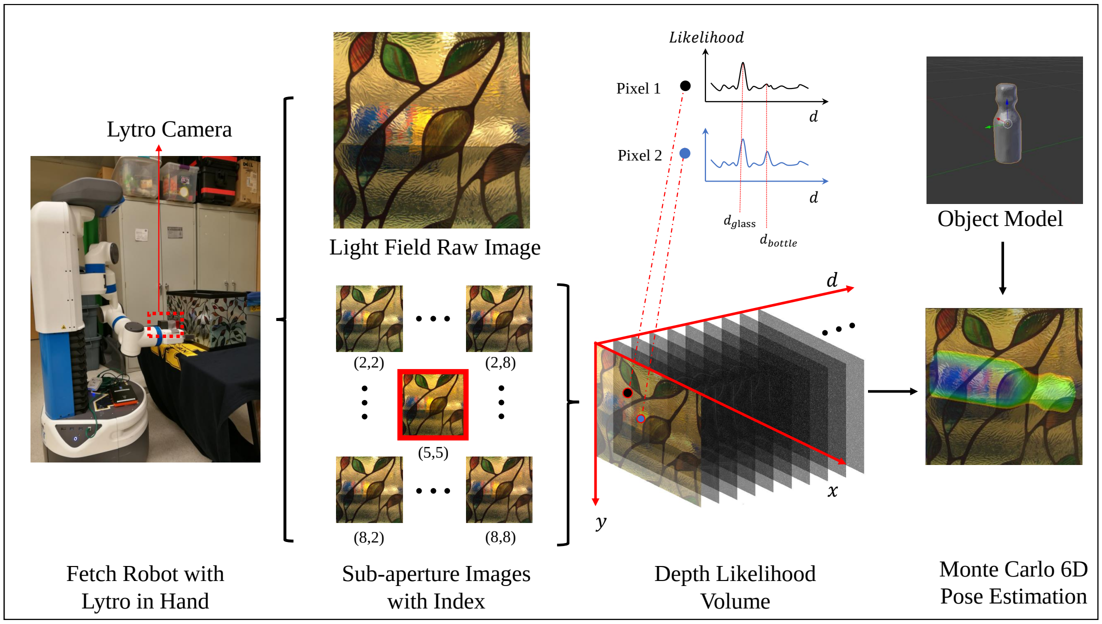


Figure 3.2: An overview of our Plenoptic Monte Carlo Localization framework. A light field camera is installed on the end effector of the robot. After taking a single shot light field image of the scene, sub-aperture images are extracted (center view highlighted in red). The depth likelihood volume (DLV) is then computed as a 3D array of depth likelihoods over certain pixels (i, j) for depth d. The DLV is a comparator of color and gradient similarity between the center view and other sub-aperture images. Assuming a known geometry and region of interest, the six-DoF object pose is estimated by Monte Carlo Localization over a constructed DLV.

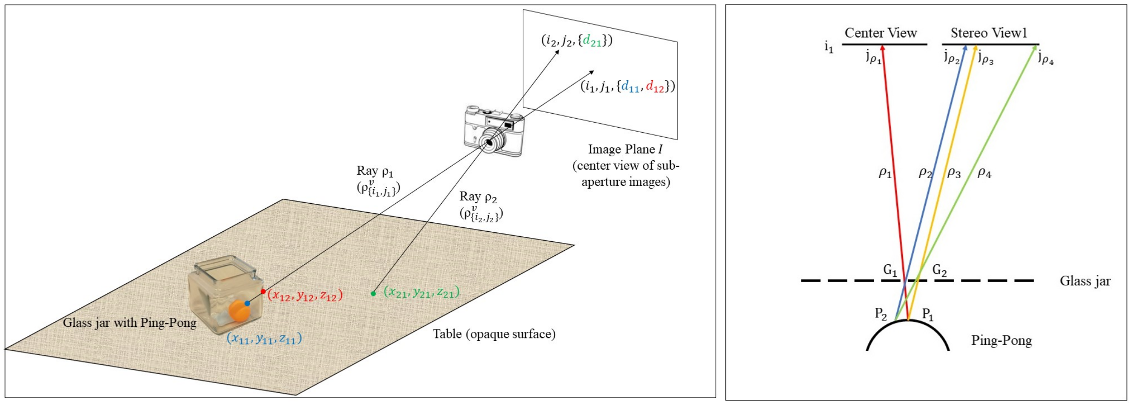


Figure 3.3: (Left) a scene with a transparent glass jar containing a ping-pong ball at rest on an opaque table. Along ray ρ_1, two surfaces (incident to the ball and the front surface of the jar) contributes to the pixel value, while along ray ρ_2 only one surface (incident to the table) appears. (Right) a planar top-down view of rays incident to the ball and the jar. The center view image plane, (i_1, j_{ρ_1}) receives a weighted sum of light rays reflected from both the glass surface point G_1 and the ping-pong surface P_1. Three example rays corresponding to ρ_2 (reflection of the surface from the glass jar), ρ_3 (reflection of the ping-pong ball through the glass), and ρ_4 (random ray) received by the image plane with incidence to scene points (G_1, P_2), (G_2, P_1), and (G_2, P_2), respectively. They indicate three depth d_g, d_p, d_i when form stereo pair with ray ρ_1.

3.4.1 Formulation

Given a known 3D workspace and its corresponding center view sub-aperture image plane I, a Depth Likelihood Volume is defined in Equation 3.2. The DLV makes the following basic assumptions and notations for the scene:

(1) Each surface point emits light rays ρ in each channel as a Gaussian over (r, g, b) with mean (μ_r, μ_g, μ_b) and variance $(\sigma_r^2, \sigma_g^2, \sigma_b^2)$ which means $\rho = \mathcal{N}(\lambda; \mu_c, \sigma_c^2), c \in \{r, g, b\}$, as similarly assumed by Oberlin and Tellex [38]. Under constant lighting condition we assume every point in the scene shares the same variance for the same color channel which means $\sigma_c = \sigma_c', c \in \{r, g, b\}$ for all points in the scene.

(2) An observed bundle of rays located at pixel plane (i, j) is a linear combination of all light rays emitted by surface points along the light rays with the normalization scalers α_i. α_i indicates the percentage of rays emitted by the surface in observed rays

which measures the transparency of the surface, and we have $\sum_i \alpha_i = 1$.

Consider the example in Figure 3.3 (Left) of two light rays $\rho^v_{\{i_1,j_1\}}, \rho^v_{\{i_2,j_2\}}$ imaged by the central view sub-aperture image. The index v indicates center view, and $\{\cdot,\cdot\}$ are pixel coordinates in the center view. These rays are in the 3D space hitting the center view plane I at $(i_1, j_1), (i_2, j_2)$, respectively. Along $\rho^v_{\{i_1,j_1\}}$, there are two surfaces emitting light which are sensed by the central view: one is a ping-pong ball and the other is the glass jar. In contrast, along $\rho^v_{\{i_2,j_2\}}$, only light emitted by the table is sensed in the central view. Then $\rho^v_{\{i_1,j_1\}}, \rho^v_{\{i_2,j_2\}}$ can be expressed respectively as:

$$\begin{aligned} \rho^v_{\{i_1,j_1\}} &= \alpha_g \rho_{\text{glass}} + \alpha_p \rho_{\text{ping-pong}} \\ \rho^v_{\{i_2,j_2\}} &= \alpha_t \rho_{\text{table}} \end{aligned} \tag{3.3}$$

where $\rho_{\text{glass}}, \rho_{\text{ping-pong}}, \rho_{\text{table}}$ represents the light rays emitted by glass, ping-pong, and table surfaces, respectively. According to our second assumption, we also have $\alpha_g + \alpha_p = 1$ and $\alpha_t = 1$.

Then the depth likelihood is defined as:

$$\begin{aligned} &L(i,j,d) = \\ &\sum_n \frac{\max_k ||\rho^v_{\{i,j\}}, \mathcal{T}^n_k(\rho^v_{\{i,j\}})||^2 - ||\rho^v_{\{i,j\}}, \mathcal{T}^n_d(\rho^v_{\{i,j\}})||^2}{\sum_k ||\rho^v_{\{i,j\}}, \mathcal{T}^n_k(\rho^v_{\{i,j\}})||^2} \end{aligned} \tag{3.4}$$

where $\mathcal{T}^n_k(\rho^v_{\{i,j\}})$ is the transformation function finding the light ray corresponding to $\rho^v_{\{i,j\}}$ in stereo pair image index with n that indicates depth k . For light field camera known baseline b and focal length f, the $\mathcal{T}^n_k(\cdot)$ can be expressed as $\frac{bf}{D}$, where D is disparity which is the function of n and k. $||\cdot,\cdot||^2$ is the squared similarity distance between two light rays over $\{r, g, b\}$ color space which is defined as L_2 distance between two Gaussian mixture models according to assumption (1) and (2) and can be expressed as Equation 3.9.

3.4.2 Validity

We claim that for a given (i, j) in DLV the following Lemma holds:

Lemma 1

$$\alpha_1 < \alpha_2 \iff L(i, j, d_1) < L(i, j, d_2)$$

where d_1, d_2 indicates the true surface depth viewed from center view with transparency indicator α_1, α_2. This means, the more transparent a surface, the less likelihood the depth of this surface will be in the DLV.

To show the Lemma 1, we consider the scene as shown in Figure 3.3 (Right). In the center view (where DLV will be built), $\rho^{v}_{\{i,j_{\rho_1}\}}$ (simplify notation as ρ_1) contains rays from the glass surface point G_1 and ping-pong surface point P_1 which has depths d_g, d_p respectively. We then evaluate three possible depths in this scene: d_g, d_p, and a invalid depth d_i. For every surface point, corresponding $\alpha_g, \alpha_p, \alpha_i$ are set as $\alpha_g = \alpha, \alpha_p = (1 - \alpha), \alpha_i = 0$. Notice that $\alpha < 0.5$ since glass is a transparent surface while ping-pong is not. Using function $\mathcal{T}^n_k(\rho_1)$ we can find three rays (ρ_2,ρ_3,ρ_4) in stereo image n corresponding to three depths d_g, d_p, and d_i separately. Then, we can write ray ρ_1 as:

$$\rho_1 = \alpha\mathcal{N}(\lambda; \mu_{G_1c}, \sigma^2_{G_1c}) + (1 - \alpha)\mathcal{N}(\lambda; \mu_{P_1c}, \sigma^2_{P_1c}) \tag{3.5}$$

where $c \in \{r, g, b\}$ represents three color channels. Without loss of generality, we investigate the red channel and write ρ_2, ρ_3, ρ_4 in same fashion:

$$\rho_2 = \alpha\mathcal{N}(\lambda; \mu_{G_1r}, \sigma^2_{G_1r}) + (1 - \alpha)\mathcal{N}(\lambda; \mu_{P_2r}, \sigma^2_{P_2r}) \tag{3.6}$$

$$\rho_3 = \alpha\mathcal{N}(\lambda; \mu_{G_2r}, \sigma^2_{G_2r}) + (1 - \alpha)\mathcal{N}(\lambda; \mu_{P_1r}, \sigma^2_{P_1r}) \tag{3.7}$$

$$\rho_4 = \alpha\mathcal{N}(\lambda; \mu_{G_2r}, \sigma^2_{G_2r}) + (1 - \alpha)\mathcal{N}(\lambda; \mu_{P_2r}, \sigma^2_{P_2r}) \tag{3.8}$$

Here, we assume that transparent surfaces emit an equal amount of light rays between any two stereo images because the disparity range between adjacent sub-aperture views of the Lytro camera is smaller than ± 1 pixel [95] (around 10^{-4} rads in view angle in our experiment setting). The squared similarity ($||\cdot,\cdot||^2$) distance between ρ_1 and any other rays can be expressed as:

$$||\rho_1(\lambda), \rho_n(\lambda)||^2 = \int (\rho_1(\lambda) - \rho_n(\lambda))^2 \, d\lambda \tag{3.9}$$

where $n \in \{2, 3, 4\}$. Given this general expression of distance, we can now provide explicit expressions for the example shown in Figure 3.3 Right:

$$||\rho_1(\lambda), \rho_2(\lambda)||^2 = 2(1-\alpha)^2(A - \mathcal{N}(\mu_{P_1r}; \mu_{P_2r}, 2\sigma_r^2)) \tag{3.10}$$

$$||\rho_1(\lambda), \rho_3(\lambda)||^2 = 2\alpha^2(A - \mathcal{N}(\mu_{G_1r}; \mu_{G_2r}, 2\sigma_r^2)) \tag{3.11}$$

$$\begin{aligned} ||\rho_1(\lambda), \rho_4(\lambda)||^2 = {} & ||\rho_1(\lambda), \rho_2(\lambda)||^2 + ||\rho_1(\lambda), \rho_3(\lambda)||^2 \\ & + 2\alpha(1-\alpha)(\mathcal{N}(\mu_{G_1r}; \mu_{P_1r}, 2\sigma_r^2) \\ & - \mathcal{N}(\mu_{G_1r}; \mu_{P_2r}, 2\sigma_r^2) \\ & + \mathcal{N}(\mu_{G_2r}; \mu_{P_2r}, 2\sigma_r^2) \\ & - \mathcal{N}(\mu_{G_2r}; \mu_{P_1r}, 2\sigma_r^2)) \end{aligned} \tag{3.12}$$

where $A = \frac{1}{\sqrt{4\pi\sigma_r^2}}$ and given the following relation:

$$\int \mathcal{N}(x; \mu, \Sigma)\mathcal{N}(x; \mu', \Sigma')dx = \mathcal{N}(\mu; \mu', \Sigma + \Sigma') \tag{3.13}$$

For the same object, under 10^{-4} rads view difference, we assume the color difference between two surface points have the same scale Δ. This assumption implies, for some small value ϵ, that $\epsilon > \Delta = |\mu_{P1r} - \mu_{P2r}| = |\mu_{G1r} - \mu_{G2r}|$.

Disregarding constant scale 2, Equation 3.10, 3.11, 3.12 can be simplified as Equation 3.14, 3.15, 3.16:

$$||\rho_1(\lambda), \rho_2(\lambda)||^2 = (1-\alpha)^2 A(1 - \exp^{-\frac{\Delta^2}{4\sigma_r^2}}) \tag{3.14}$$

$$||\rho_1(\lambda), \rho_3(\lambda)||^2 = \alpha^2 A(1 - \exp^{-\frac{\Delta^2}{4\sigma_r^2}}) \tag{3.15}$$

$$||\rho_1(\lambda), \rho_4(\lambda)||^2 = ((1-\alpha)^2 + \alpha^2) A(1 - \exp^{-\frac{\Delta^2}{4\sigma_r^2}}) \tag{3.16}$$

Considering an individual stereo pair and applying Equation 3.4, we can now express the DLV values for the possible depths for the surface of ping-pong ball, d_p, the glass surface, d_g, and the invalid depth, d_i, as:

$$L(i,j,d_p) = \frac{(1-\alpha)^2}{(1-\alpha)^2 + \alpha^2} \tag{3.17}$$

$$L(i,j,d_g) = \frac{\alpha^2}{(1-\alpha)^2 + \alpha^2} \tag{3.18}$$

$$L(i,j,d_i) = \qquad 0 \tag{3.19}$$

which implies that the ping-pong surface must return more light than the glass surface:

$$\alpha_g < \alpha_p \iff L(i,j,d_g) < L(i,j,d_p), \alpha_p, \alpha_g \in [0,1] \tag{3.20}$$

Therefore, Lemma 1 holds.

3.4.3 Computation

Our implementation uses the L_2 distance between adjacent pixel colors to approximate the similarity of rays in stereo pairs, as photosensors are unable to capture the distribution over wavelengths of light. Considering this limitation, a cost-volume stereo comparison method based on sub-pixel shift [96, 94] was implemented. Two different cost volumes were implemented: the sum of L_2 distance in color space (C_c) and the sum of gradient differences (C_g). The cost volume C then can be defined as:

$$C(\mathbf{x}_\rho, l) = \beta C_c(\mathbf{x}_\rho, l) + (1 - \beta) C_g(\mathbf{x}_\rho, l) \tag{3.21}$$

where $\mathbf{x}_\rho = (i, j)$ describes the image coordinate of ray ρ, l is depth labels and β is a scalar to weight two parts. The terms C_c and C_g are defined as:

$$\begin{aligned}
&C(\mathbf{x}_\rho, l) = \\
&\quad \sum_{\mathbf{s} \neq \mathbf{s}_c} \sum_{\mathbf{x}_\rho \in R_\mathbf{x}} \min(|I(\mathbf{s}_c, \mathbf{x}_\rho) - I(\mathbf{s}, \mathbf{x}_\rho + \Delta\mathbf{x}(\mathbf{s}, l))|, \tau_1) \\
&C_g(\mathbf{x}_\rho, l) = \\
&\quad \sum_{\mathbf{s} \neq \mathbf{s}_c} \sum_{\mathbf{x}_\rho \in R_\mathbf{x}} \gamma \min(|I_x(\mathbf{s}_c, \mathbf{x}_\rho) - I_x(\mathbf{s}, \mathbf{x}_\rho + \Delta\mathbf{x}(\mathbf{s}, l))|, \tau_2) \\
&\quad + (1 - \gamma) \min(|I_y(\mathbf{s}_c, \mathbf{x}_\rho) - I_y(\mathbf{s}, \mathbf{x}_\rho + \Delta\mathbf{x}(\mathbf{s}, l))|, \tau_2)
\end{aligned} \tag{3.22}$$

where I is the image, I_x, I_y is the image gradient in x, y direction, $R_\mathbf{x}$ is a rectangular region that center at $\mathbf{x}_\rho$; τ_1, τ_2 is a truncation value of a robust function, $\Delta\mathbf{x}(s, l)$ is the sub-pixel displacement, and $\gamma = \frac{|\mathbf{s} - \mathbf{s}_c|}{|\mathbf{s} - \mathbf{s}_c| + |\mathbf{t} - \mathbf{t}_c|}$ weights different sub-aperture's gradient contributions to the center view image. Variables $\mathbf{s}, \mathbf{t}$ represent pixel in sub-aperture image index coordinate and $\mathbf{s}_c, \mathbf{t}_c$ represent pixel in the center view.

For a certain depth label l_i, the depth likelihood can be expressed as below based on Equation 3.4:

$$L(\mathbf{x}_\rho, l_i) = \log\left(\frac{\arg\max_l C(\mathbf{x}_\rho, l) - C(\mathbf{x}_\rho, l_i)}{\sum_{l_i} (C(\mathbf{x}_\rho, l_i))} + 1\right) \tag{3.23}$$

Optionally, to further distinguish possible depths, the DLV can be truncated by finding N_{lm} number of local maximum with its K_{lm} number of neighbors and setting the other depth likelihoods to 0.

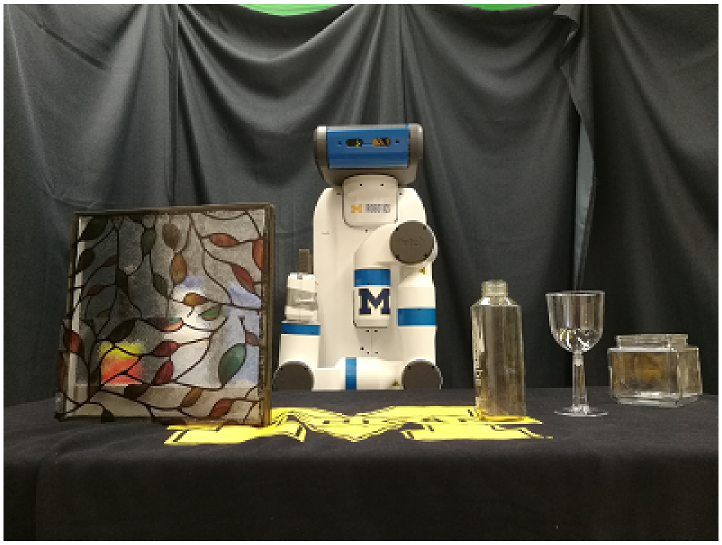


Figure 3.4: Test objects for evaluating PMCL 6D pose estimation include: (to the left) opaque objects behind a partially opaque translucent surface (a stained glass window film), and (to the right) transparent objects.

3.5 Plenoptic Monte Carlo Object Localization

Building on the DLV, we now describe our method of object pose estimation as Plenoptic Monte Carlo Localization. PMCL employs particle filtering to estimate the pose of target objects from the computed DLV. PMCL takes direct inspiration from the work of Dellaert et al. [97] for approximate inference in the form of a sequential Bayesian filter,

$$Bel(q_t) \propto p(z_t|q_t) \sum_j p(q_t^{(j)}|q_{t-1}^{(j)}) Bel(q_{t-1}^{(j)}) \tag{3.24}$$

where a collection of n weighted particles $\{q_t^{(j)}, w_t^{(j)}\}_{j=1}^{n}$ is used to represent the pose belief q_t.

Each particle $q_t^{(j)}$ is a hypothesized six-DoF pose of the object and is associated with the weight $w_t^{(j)}$ indicating how likely the sample is to be close to the actual pose. The initial samples are generated by uniformly sampling the six-DoF pose with identical weight. The weight of each sample is then calculated by using the observation likelihood function described in the next paragraph. With the computed weights, an importance sampling with resampling procedure is performed to concentrate hypothesized particles to more weighted range. For state transition, each particle will be perturbed by a zero-mean Gaussian distribution in the space of six-DoF in the action model. This inference can be naturally extended to the case of tracking with an explicit action model and observations over time. In our implementation, the process will iteratively repeat until the average weight is above a chosen threshold for taking an estimate.

Our likelihood function measures the score of a sample's rendered depth image for a scene DLV. The z-buffer of a 3D graphics engine is used to render each sample into a depth image for comparison with the observation. This rendered depth image, represented as $z^{(j)}$, is mapping back to DLV to find the corresponding depth likelihood interval $[l_n, l_m)$. Here, we use an interval because the rendered depth value for a certain pixel may not exactly match its discretized depth value. After finding the corresponding interval, the depth likelihood is calculated using linear interpolation:

$$L(\mathbf{x}_\rho, l_n) = L(\mathbf{x}_\rho, l_n) + \frac{(l - l_n)(L(\mathbf{x}_\rho, l_m) - L(\mathbf{x}_\rho, l_n))}{l_m - l_n} \tag{3.25}$$

For the rendered image, with every rendered pixel having non-zero (vaild) depth value l_i, the score for this depth image can be expressed as:

$$L(z_t) = \frac{\sum_i L(\mathbf{x}_\rho, l_i)}{N} \tag{3.26}$$

where N is the number of valid depths in the rendered image.

3.6 Results

We now present results for our implementation of PMCL for object localization and grasping in environments with different forms of translucency. We have implemented PMCL using observations from a Lytro light field camera mounted on the wrist of a Michigan

Progress Fetch robot (Figure 3.4). These results consider pick-and-place grasping in two types of scenes with: 1) a single transparent object with an opaque but possible reflective background objects (Figures 3.6, 3.6), and 2) opaque objects behind translucent non-transparent surfaces (Figures 3.6, 3.6).

Our implementation uses the Lytro on-chip wifi to trigger the shutter remotely and receive raw image data. We are currently unable to capture video with this triggering system. Calibration and sub-aperture images are generated using the methods described by Bok et al. [98]. This toolbox generates 9×9 sub-aperture images, where the image at index $(5, 5)$ is deemed the center view image. Each sub-aperture image has resolution 328×328. During DLV construction, we disregard edge sub-aperture images due to strong color distortion and pixel shifting artifacts. Our PMCL algorithm is implemented on CUDA and OpenGL. This implementation ran on a Ubuntu 14.04 operating system with a Titan X graphics card and CUDA 8.0. The light field camera calibration, sub-aperture images extraction, and DLV construction ran in MATLAB. The chosen parameters for building the DLV were $\beta = 0.5$, $\tau_1 = 0.5$, $\tau_2 = 0.5$, $l = 75$, $N_{lm} = 2$, and $K_{lm} = 2$. The Monte Carlo localization process ran on the GPU with 100 particle samples over 500 iterations.

With an assumed object geometry, our implementation renders all the particle hypotheses on the GPU. These renderings can be accessed by the CUDA kernels to compute the corresponding weights. Our implementation additionally assumes a given 3D region of interest on the object pose in workspace.

For robot control, we use our custom manipulation pipeline developed by the Laboratory for Progress. This pipeline uses our implementation of handle grasp localization as proposed by ten Pas and Platt [25]. This grasp localization returns an end-effector pose for grasping from an estimated object pose with a given geometric model. Grasping is then executed for this end-effector pose using TRAC-IK [99] and MoveIt! [100] for inverse kinematics and motion planning.

To evaluate the pose estimation accuracy of our algorithm, we used two methods to

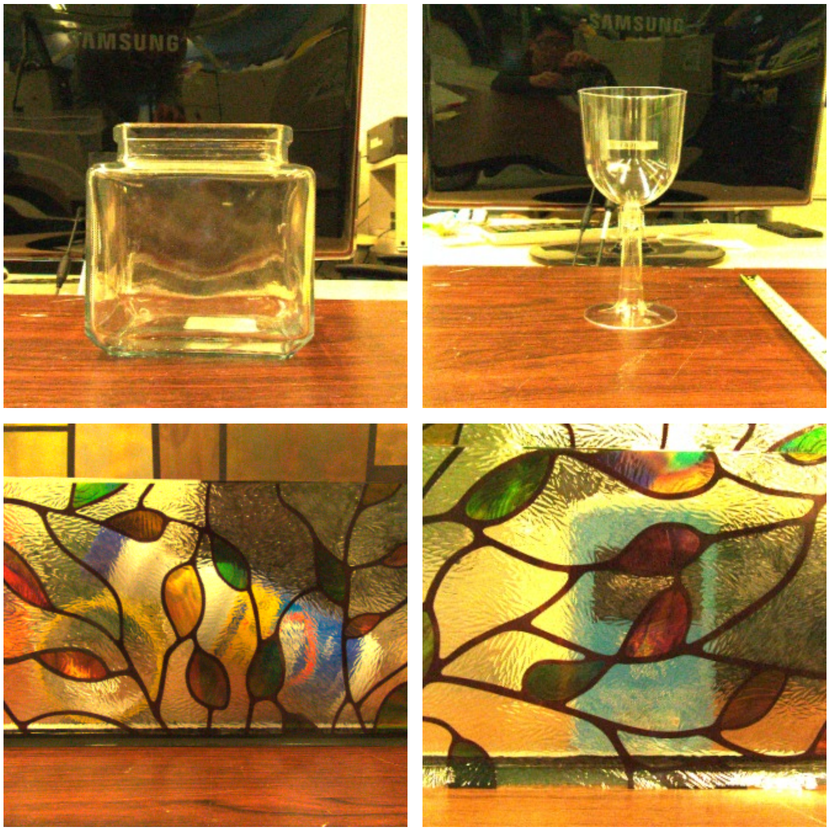


Figure 3.5: Two types of scene for localizing object poses. (a-b) the scene with a single transparent object with an opaque but possible reflective background objects. (c-d) the scene with opaque objects behind translucent non-transparent surfaces

collect ground-truth object poses. For objects behind the window covered by stained glass film, we captured point clouds by removing the glass and using Asus Xtion Pro RGB-D on the robot. Object models were then fit manually to determine ground truth pose values. For transparent objects, their surfaces were covered with opaque tape to generate point clouds for ground truth annotation.

3.6.1 Pose Estimation Results

We evaluate our proposed algorithm on six scenes and run ten trials for each. Two types of error are applied to evaluate our pose estimation accuracy:

- Translation error: defined as the Euclidean distance between estimated object position (x, y, z) and ground truth position (x_{gt}, y_{gt}, z_{gt})

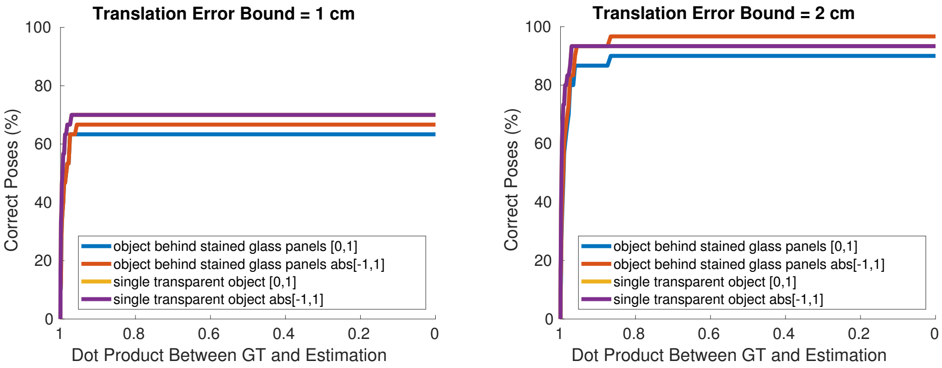


Figure 3.6: The percentage of correctly localized object under different thresholds for the object behind a stained glass panel and a single transparent object. In each plot, the translation error bound is fixed to 1cm (a) and 2cm (b). The x-axis is the decreasing dot product bound indicate the error between ground truth and estimated result. The y-axis is the percentage of correctly localized objects. For each type of scene, these plots consider two types of rotation error ranges: $[0, 1]$ in dot product space indicates for [90,0] in degrees, and the absolute value [-1,1] in dot product space indicates [180, 0] in degrees.

- Rotation error: defined as dot product between ground truth pose z-axis and estimated pose z-axis. We assume the objects are rotational symmetric along z-axis.

We consider an object is correctly localized when both translation and rotation errors fall into a certain threshold. Figure 3.6 establishes our estimation accuracy on two types of the scene.

For the single transparent object, the all rotation error in dot product space laid in [0,1], which leads to the overlapping of yellow and purple lines in both plots. For an object behind stained glass panels, the estimated poses sometimes have 180 degree flipping, a negligible form of error assuming symmetry.

3.6.2 Manipulation Results

We succeed in demonstrating our method in two challenge scenarios for manipulation

1. Pick-and-place glass cup from a sink with running water
2. Pick-and-place bleach bottle from an aquatic tank covered with private window film.

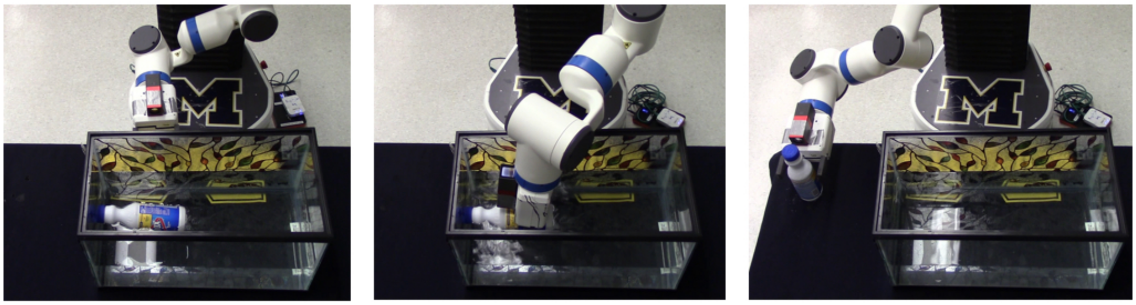


Figure 3.7: The robot executes pick-and-place action for the bleach bottle floating on the water. The bleach bottle is inside the aquatic tank so it is occluded by the stained glass from the camera view.

The scenarios are shown in Figure 3.1 and Figure 3.7. We attach the Lytro camera to the wrist of the robot and add extra link for it. For both scenarios, the robot moves its arm to the appropriate area to capture the light field images, from which the DLV is calculated. Our PMCL then performs estimation to infer the pose of the object and the final estimation is taken to transform the pre-calculated grasp poses in robot base link. With the accurate pose estimation, the robot is able to pick up objects from both aquatic tank and sink and place the objects on the desired location.

3.7 Summary

In this chapter, we present Plenoptic Monte Carlo Localization for localizing object pose in the presence of translucency from plenoptic (light-field) observations. We propose a new depth descriptor, the Depth Likelihood Volume, to address the uncertainties from the translucency by generating possible depth likelihoods for each pixel. We show that by using the Depth Likelihood Volume within a Monte Carlo object localization algorithm our method is able to accurately localize objects with translucent materials and objects occluded by layered translucency and perform manipulation.

CHAPTER 4

GlassLoc: Plenoptic Grasp Pose Detection in Transparent Clutter

4.1 Motivation

Robot grasping in household environments is challenging because of sensor uncertainty, scene complexity and actuation imprecision. Recent results suggest that Grasp Pose Detection (GPD) using point cloud local features [101] and manually labeled grasp confidence [102] can be applied in generating feasible grasp poses over a wide range of objects. However, domestic environments include a great amount of transparent objects, ranging from kitchen utilities (e.g. wine cups and containers) to house decoration (e.g. windows and tables). The reflective and transparent material on those objects will produce invalid readings from depth camera. This problem becomes more significant in the real world where there are piled transparent objects which will lead to unexpected robot manipulation behaviors if the robot was trying to interact with the objects. A correct estimation of transparency is necessary to protect the robot from performing hazardous actions and extend robot applications to more challenging scenarios.

The problem of performing grasping in transparent clutter is complicated by the fact that robots cannot perceive and describe the transparent surfaces correctly. Several previous methods [35, 34] tried to approach this problem by finding invalid values in depth observation, but they were limited to top-down grasping and made assumption that target objects establish distinguishable contour (formed by invalid points) in depth map. Recently, several approaches employed light field camera to observe the transparency and showed promising results. Zhou et al. [103] used single shot light field image to form a new plenoptic descriptor named Depth Likelihood Volume (DLV). They succeeded in estimating the pose of single transparent object or object behind translucent surface by given the corresponding object CAD model. Based on that, we extend the idea to a more general-purpose grasp detection scenario with transparent objects clutter.

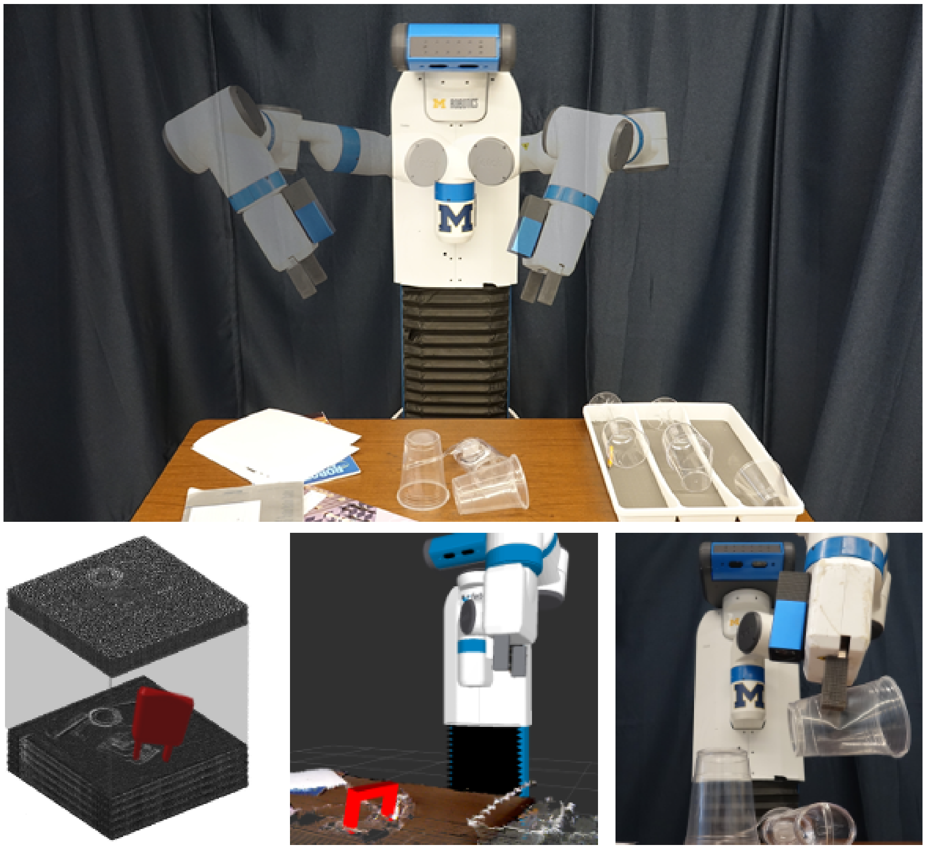


Figure 4.1: (Top) a robot using *GlassLoc* to pick up transparent objects from clutter and place on the tray. The robot is observing the scene using a light field camera. Grasp candidate is sampled in DLV (bottom left) and mapped to the world frame in the visualizer (bottom middle). The robot successfully picks up a transparent cup from the clutter (bottom right).

We make several contributions in this chapter. First, we propose *GlassLoc* algorithm for detecting six-DoF grasp poses of transparent objects in both separated and minor over-

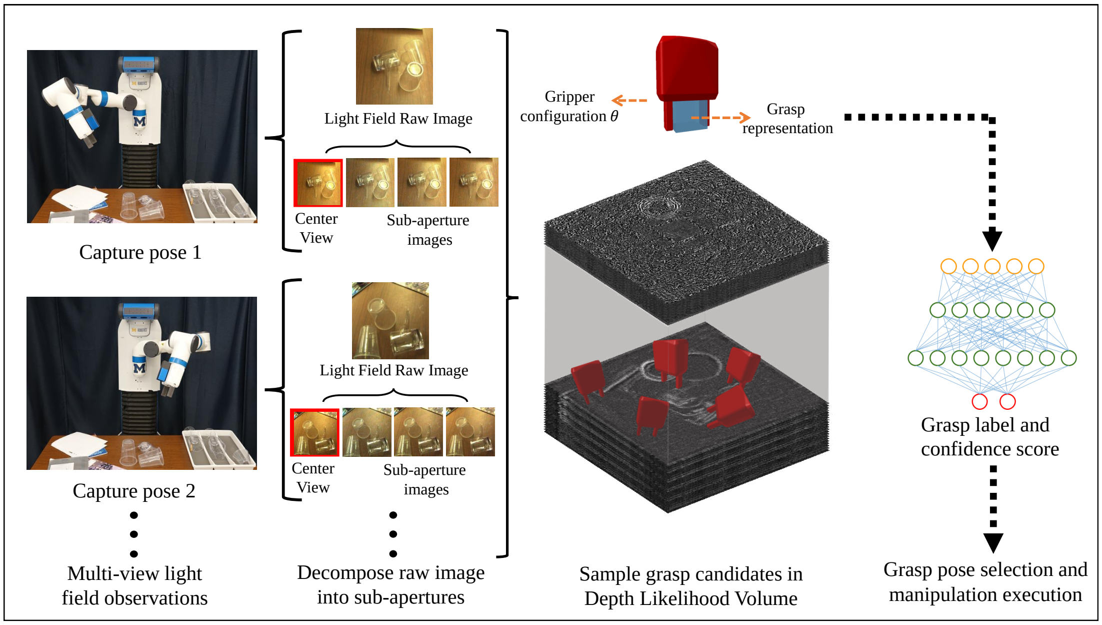

Light Field Raw Image
Center View
Sub-aperture images
Capture pose 1
Light Field Raw Image
Center View
Sub-aperture images
Capture pose 2
Multi-view light field observations
Decompose raw image into sub-apertures
Gripper configuration θ
Grasp representation
Sample grasp candidates in Depth Likelihood Volume
Grasp label and confidence score
Grasp pose selection and manipulation execution

Figure 4.2: An overview of *GlassLoc* framework. A light field camera is mounted on the end-effector of the robot. After taking a set of light field observations by moving robot arms, sub-aperture images are extracted (center view is highlighted in red). The Depth Likelihood Volume (DLV) is then computed as a 3D volume of depth likelihoods over transparent clutter. Given gripper configuration, we can sample grasp poses in DLV and extract grasp features for the classifier to label whether the samples are graspable or not.

lapping cluttered environments. Next, we propose a generalized model for constructing Depth Likelihood Volume from multi-view light field observations with multi-ray fusion and reflection suppression. Finally, we integrate our algorithm with a robot manipulation pipeline to perform tabletop pick and place tasks over eight scenes and five different transparent objects. Our results show that the grasping success rate over all test objects is 81% in 220 grasp trials.

4.2 Related Work

4.2.1 Grasp Perception in Clutter

It remains a challenging task for robots to perform perception and manipulation in cluttered environments considering the complexity of the real world. We consider there are two major categories of methods for robots to perform grasp perception in clutter. The first category is model-based pose estimation methods. By estimating object poses, grasp configurations calculated based on the local model can be further transformed to the robot environments. Collet et al. [40] utilized color information to estimate poses of object in cluttered environments. Their proposed algorithm clusters and then matches the local color patch from object model to robot observations to generate pose hypotheses. Sui et al. [28, 42] constructed generative models to evaluate pose hypotheses against point cloud using object CAD models. The generative models perform object detection followed by particle filtering for robot grasping in the highly cluttered tabletop environments. With a similar idea, Papazov et al. [41] leveraged RANSAC-based bottom-up approach with Iterative Closest Point registration to fit 3D geometries to the observed point cloud.

On the other hand, rather than associating a grasp pose with a certain object model, Grasp Pose Detection (GPD) tries to characterize grasp poses based on the local geometry or appearance features directly from observations. Several early works [22, 23] represented the grasp poses as oriented rectangles in RGB-D observations. Further, given a number of manually-labelled grasp candidates, the system will learn to predict whether a sampled rectangle is graspable or not. One major restriction of those systems is that the approaching directions of generated grasp candidates need to be orthogonal to the RGB-D sensor plane. Fischinger and Vincze [24] tried to lessen the restriction by integrating hightmap-based features. They also designed a heuristic for ranking the grasp candidates in a clutter bin settings. ten Pas and Platt [25] directly detected grasp poses in $SE(3)$ space by estimating curvatures and extracting handle-like features in local point cloud neighborhoods. Gualtieri et al. [26] proposed more types of local point cloud features for grasp representation and projected those features to 2D image space for classification. Our work with *GlassLoc* extends these ideas to transparent clutter with a different grasp representation and a new plenoptic descriptor.

4.2.2 Light Field Photography

The models describing the light field rendering proposed by Levoy and Hanrahan [74] introduced foundations of light field captured from multi-view cameras. Based on this work, [75, 90] succeeded in producing commercial level hand-held light field camera using the microlens array structure. Building on the property that the plenoptic camera can capture both intensity and direction of light rays, light field photography has shown significant advancement in different applications. Wang et al. [93] explicitly modeled the light field image pixel angular consistency to generate accurate depth map for the object with occlusion edges. Jeon et al. [94] performed sub-pixel shifting in image frequency domain in tackling the microlens camera narrow baseline problem for accurate depth estimation. Maeno et al. [91] introduced distortion feature in light field to detect and recognize the transparent object. Johannsen et al. [36] leveraged multi-view light field images to reconstruct multi-layer translucent scenes. Skinner and Johnson-Roberson [104] introduced a light propagation model suited to underwater perception using plenoptic observations.

The use of light field perception in robotics is still relatively new. Oberlin and Tellex [38] proposed a time-lapse light field capturing pipeline for static scenes by mounting a RGB camera on the end-effector of the robot and moving in a designed trajectory. Dorian et al. [105] introduced a algorithm for distinguishing refracted and Lambertian features from light field image. Zhou et al. [103] used a Lytro camera to take a single shot of the scene and construct a plenoptic descriptor over that. Given the target object model, their methods can estimate single object six-DoF pose in layered translucent scenes. Our *GlassLoc* pipeline extends the idea proposed in [103] for more general-purpose manipulation over transparent clutter.

4.3 Problem Formulation and Approach

GlassLoc addresses the problem of grasp pose detection for transparent objects in clutter from plenoptic observations. For a given static scene, we assume there is a latent set of end-effector poses $G \subset SE(3)$ that will produce a successful grasp of an object. A successful grasp is assumed to result in the robot obtaining force closure on an object when it moves gripper and closes its fingers. The plenoptic grasp pose detection problem is then phrased as estimating a representative set of valid sample grasp poses $G_{\text{v}} \subset G$.

Within the grasp pose detection problem, a major challenge is how to classify whether a grasp pose is a member of G, and, thus, will result in a successful manipulation. For grasp pose classification, we assume as given robot end-effector pose $q \in SE(3)$ and a collection of observations Z from a plenoptic sensor. It is assumed that each observation $z_{1:N} \in Z$ captures a raw light field image o_i of a static scene from camera viewpoint $v_i \subset SE(3)$.

The classification result calculated from these inputs is a likelihood $l \in [0, 1]$ that relates the probability of end-effector pose, q, resulting in a successful grasp. Described later, our implementation of *GlassLoc* will perform the classification using a neural network.

Illustrated in Figure 4.3, grasp pose classification within *GlassLoc* is expressed as a function $l = \mathcal{M}(U)$ that maps transparency occupancy likelihood features U to grasp pose confidence l. Transparency occupancy features $U(q, D)$ are computed with respect to the subset of a Depth Likelihood Volume (DLV) D that is within the graspable volume of pose q. The DLV estimates how likely a point $p \in \mathbb{R}^3$ belongs to a transparent surface. To test all sampled grasps, a Depth Likelihood Volume D is computed from observations Z over an entire grasping workspace $P \subset SE(3)$ within the visual hull of $v_{1:N}$. We assume the grasping workspace is discretized into $p_{1:M} \in P$ a set of 3D points, with each element of this set expressed as $p_i = (x_i, y_i, z_i)$.

4.4 Plenoptic Grasp Pose Detection Methods

An outline of the *GlassLoc* algorithm is described in Algorithm 1. *GlassLoc* begins by computing a Depth Likelihood Volume from multi-view light field observations. By in-

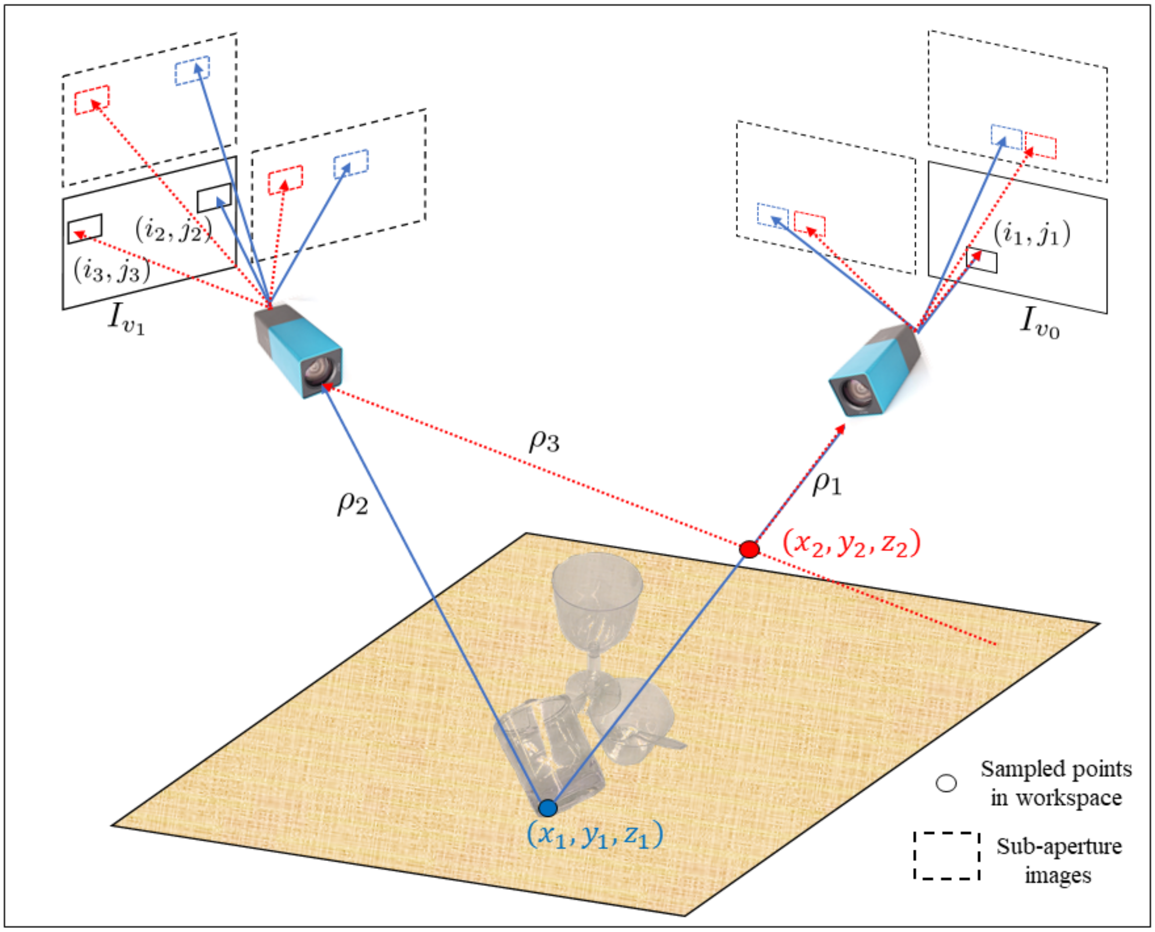


Figure 4.3: Example of DLV value calculation of two randomly sampled points (x_1, y_1, z_1) and (x_2, y_2, z_2) through examining the ray consistency in different view points. Each sample point corresponds to different pixel indices with depths in different center view plane I_{v_0} and I_{v_1}.

tegrating different views, we can further post-process the DLV by suppressing reflection caused by non-Lambertian surfaces. Details of DLV construction are presented in Section 4.4.1 and 4.4.2. In Step 2, we uniformly sample the grasp candidates $C = \{c_j \in P\}$ in workspace P. For each grasp candidate, we extract grasp representations (see Section 4.4.3) and corresponding transparency likelihood features given the robot gripper parameter θ. The generated features will then be classified with a grasp success labels and confidence scores by a neural network. The training data generation strategy for learning this mapping is introduced in Section 4.4.4. Given classified grasp poses, we use a multi-hypothesis particle-based search to find a set of end-effector poses with high confidence

Algorithm 1 *GlassLoc* Plenoptic Grasp Pose Detection

INPUT: a set of light field observations Z, robot gripper parameter θ

OUTPUT: a set of valid sample grasp poses G_v

```
$D$ = Construct_DLV(Z)
$C$ = Sample_Grasp_Candidates$(D, \theta)$
for $i = 1...K$ do
    $G_i$ = Grasp_Classification$(C)$
    $C$ = Resample_Diffuse$(G_i)$
end for
$G_{\text{v}} \leftarrow C$
```

for successful grasp execution (see Section 4.4.5). The finalized set of grasp poses will be ready for the robot to perform grasping.

4.4.1 Multi-view Depth Likelihood Volume

The Depth Likelihood Volume (DLV) is a volume-based plenoptic descriptor which represents the depth of a light field image pixel as a likelihood function rather than a deterministic value. The advantage of this representation is to keep the transparent scene structure by assigning different likelihoods to surfaces with different transparency. In [103], DLV is formulated in a specific camera frame indexed with pixel coordinates and depths. The formulation is restricted to single-view scenarios. In this chapter, we generalize the expression which takes sample points in 3-D space as input and integrates multi-view light field observations.

The DLV is defined as:

$$L(p) = \sum_{i}^{N} f\left(\sum_{a \in A \setminus I_{v_i}} T_{a,d}(\rho_{v_i}(p))\right) \tag{4.1}$$

$$T_{a,d}(\rho) = ||\rho, \mathcal{F}_{a,d}(\rho)|| \tag{4.2}$$

where $L(p)$ is the depth likelihood of sampled points p. A is the set of sub-aperture images. $\rho_{v_i}(p)$ is a light ray that goes through or emitted from point p and is received by view point v_i at (i, j) in center view image plane. N indicates the number of view points in observations. $\mathcal{F}_{a,d}(\rho)$ is the triangulation function finding the light ray corresponding to ρ in sub-aperture images indexed with a that yields depth d. d can be explicitly calculated using camera intrinsic matrix given point and view point. $||\cdot, \cdot||$ is the ray difference which is calculated by color and color gradient differences. Denote $s = \sum_{a \in A \setminus I_{v_i}} T_{a,d}(\rho_{v_i}(p))$, then $f(s)$ is a normalization function mapping color cost to likelihood. There are multiple choices of $f(s)$. In our implementation, we choose:

$$f(s) = \frac{\max_k \sum_{a \in A \setminus I_{v_i}} T_{a,k}(\rho_{v_i}(p)) - s}{\sum_k \sum_{a \in A \setminus I_{v_i}} T_{a,k}(\rho_{v_i}(p))} \tag{4.3}$$

To better explain the formulation presented above, we consider the example shown in Figure 4.3. A cluster of transparent objects are placed on a table with opaque surface. We have two light field observations $z_0 = \{o_0, v_0\}$ and $z_1 = \{o_1, v_1\}$ with center view image plane I_{v_0} and I_{v_1} respectively. There are two points $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ sampled in the space and each of them emits light rays captured by both views. In view I_{v_0}, Ray ρ_1 emitted from both points are received by the same pixel (i_1, j_1), while ρ_2 and ρ_3 are received by (i_2, j_2) and (i_3, j_3) respectively. Then we can express the depth likelihood of point p_2 as:

$$L(p_2) = f\Bigg(\sum_{a \in A \setminus I_{v_0}} T_{a,d_1}(\rho_1)\Bigg) + f\Bigg(\sum_{a \in A \setminus I_{v_1}} T_{a,d_3}(\rho_3)\Bigg) \tag{4.4}$$

Function T calculates the color and the color gradient difference between center view (rectangle with solid line in Figure 4.3) and sub-aperture view (rectangle with dot line in Figure 4.3). The location of red pixel is calculated by function $\mathcal{F}$. For micro-lens based light field camera, the pixel shift between center and sub-aperture images are usually in sub-pixel level. The realization of $\mathcal{F}$ function is based on frequency domain sub-pixel shifting method proposed in [94].

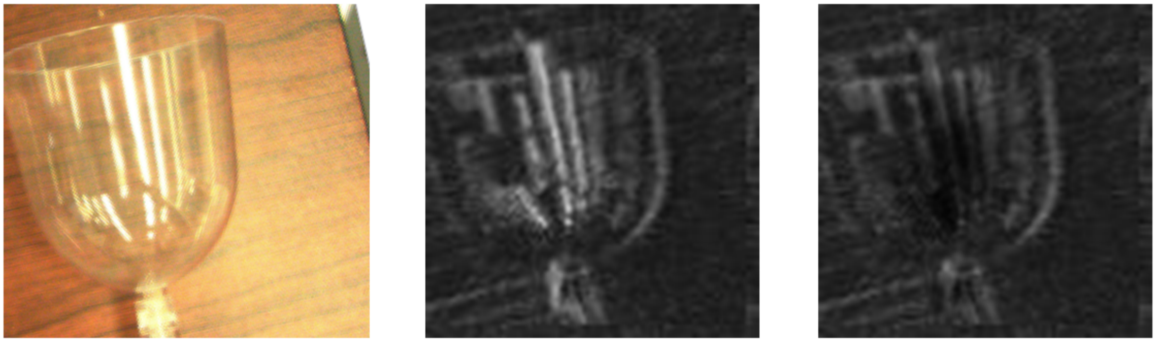


Figure 4.4: Example DLV feature image before (middle) and after (right) reflection suppression. The center view of part of raw observation is shown in (left). The intensity of pixel in the gray-scale image (middle and right) indicates the likelihood value. The high likelihood region caused by specular light is suppressed.

4.4.2 Reflection Suppression

A transparent surface produces non-Lambertian reflectance, which induces specular highlight to light field observations. Those shiny spots tend to produce the saturated color or virtual surface with larger depth than the actual transparent surface. This phenomenon will generate a high likelihood region in DLV that indicates a non-existing surface. To deal with this problem, we calculate the variance of ray differences for DLV points which has saturated color and high likelihood over different view points:

$$var\{\rho_V(p)\} = \sum_{i}^{N} \sum_{a \in A \setminus I_{v_i}} (T_{a,d}(\rho_{v_i}(p)) - E\{\rho_V(p)\})^2 \tag{4.5}$$

where $E\{\rho_V(p)\}$ can be expressed as:

$$E\{\rho_V(p)\} = \frac{1}{N \cdot (N(A) - 1)} \sum_{i}^{N} \sum_{a \in A \setminus I_{v_i}} T_{a,d}(\rho_{v_i}(p)) \tag{4.6}$$

where $N(A)$ is the number of sub-aperture images extracted from raw light field image. For a point p that has variance larger than a threshold τ, we check whether it has the largest likelihood value among all other points that lie on the light rays it emits out. Specifically, we first find light rays emitted from p and received by pixel (i, j) with depth d that has

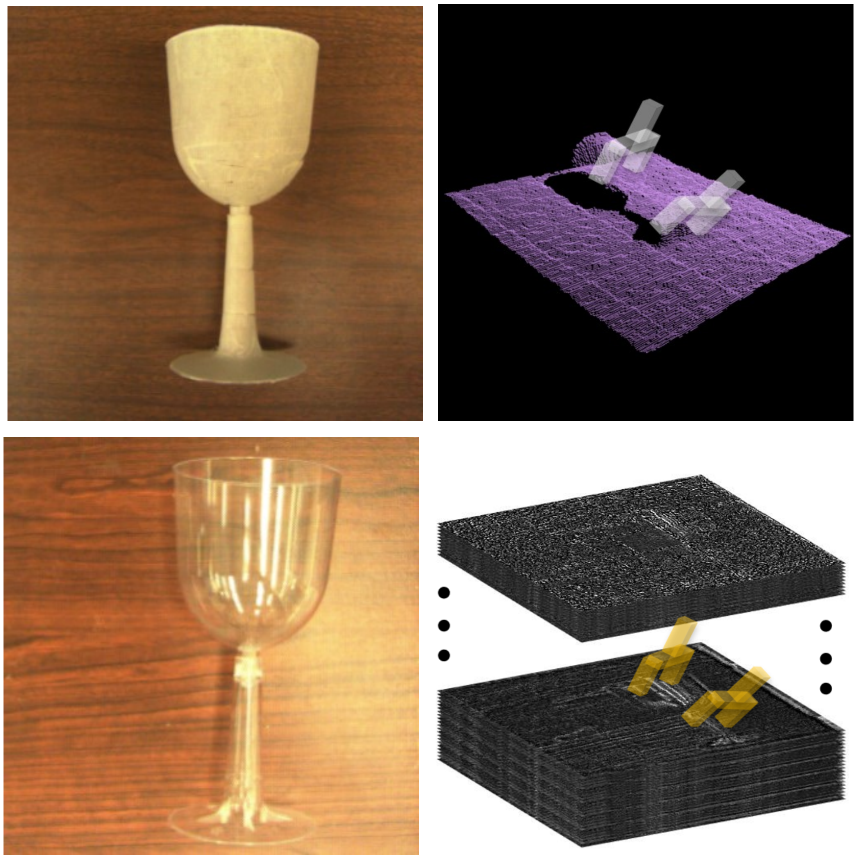


Figure 4.5: Training data generation procedure. (a) The glass cup is wrapped with opaque tape for depth sensor to get point cloud. (b) Grasp candidates are generated based on point-cloud-based method and local-to-world transform. (c) The glass cup is placed at the same pose to take multiple light field observations. (d) Grasp candidates generated from point cloud are mapped to DLV.

large variance over different view points. Then we locate all light rays received by (i, j) with depth less than d, and check whether the following equation holds:

$$\max_k \sum_{a \in A \setminus I_{v_i}} T_{a,k}(\rho_{v_i}(p)) = \sum_{a \in A \setminus I_{v_i}} T_{a,d}(\rho_{v_i}(p)) \tag{4.7}$$

If Equation 4.7 holds, it indicates this light ray has high possibility of coming from strong reflection area and will be excluded from the calculation of DLV. Figure 4.4 (left) is the sliced feature from DLV before reflective suppression which we can observe incorrect large values caused by specular highlight. Figure 4.4 (right) shows the result after processing and the previous high value area is suppressed.

4.4.3 Grasp Representation and Classification

We represent a graspable area as a 3D cuboid with length, width, and height as L, W, H respectively. The width and height of the cuboid is equal to the width and height of the volume when the robot finger close while the length is extended for capturing more feature spaces. The cuboid is voxelized into $l \times w \times h$ grid, and for each grid we interpolate the likelihood value by finding the nearest eight points in DLV. Rather than feeding into classifier with a large amount of points, we extract 2D features from the volume by projection and slicing.

We first define the three axes of the graspable volume. The x axis of the volume is defined as the approach direction of the gripper. The z axis is defined along the direction the gripper fingers close along. The y axis is the cross product of the previous two axes. We then calculate three types of features and project them to the three axes: a center slice of likelihood volume, I_c, an average likelihood map over all points, I_a, a sliced difference likelihood map, I_d, which is calculated by recursively comparing the difference between current slice of the graspable volume with the previous slice. More specifically, we can express the three types of feature as follows (take projection to x axis as example):

$$I_c(x, y) = L(x, y, z = \frac{h}{2}) \tag{4.8}$$

$$I_a(x, y) = \frac{\sum_{z=0}^{h} L(x, y, z)}{h} \tag{4.9}$$

$$I_d(x, y) = \frac{\sum_{z=0}^{h-1} |L(x, y, z) - L(x, y, z+1)|}{h} \tag{4.10}$$

We resize the images to the same size and concatenate them into different channels. Since we have three types of features and three axes to project, we have nine channels in total.

For classifier, we use the LeNet [106] structure which is a common structure for grasp pose classification and ranking [26, 107]. The output of the classifier is the binary label $\{graspable, not\ graspable\}$ associated with the confidence scores.

4.4.4 Training Data Generation

For depth-based grasp pose detection algorithms, the training data generation process relies on grasp pose sampling and labeling on point cloud. Unfortunately, depth sensors cannot provide correct point cloud for transparent objects. Instead, we wrap the object with opaque material and generate training samples by mapping grasp poses from point cloud to DLV. The detailed steps are illustrated in Figure 4.5 (a) - (d).

We have two sources to produce training samples from point cloud. One is depth-based grasp pose detection algorithms. We input those algorithms with our depth observations and label the result grasp candidates as $\{graspable\}$. In the meantime, we restore the grasp poses filtered out in those algorithms and label them as $\{not\ graspable\}$. The other is transforming pre-defined grasp pose in the local frame to the observation. By checking the gripper collision with the environment, we label the collision free grasp poses as $\{graspable\}$ and the others as $\{not\ graspable\}$.

4.4.5 Grasp Search

After we perform classification of our samples, we try to find a graspable region with relatively high classification confidence score.

Our grasp optimization builds on the particle filtering work proposed by Dellaert et al. [97], which is based on sequential Bayesian filter:

$$Bel(q_t) \propto p(z_t|q_t) \sum_j p(q_t^{(j)}|q_{t-1}^{(j)}) Bel(q_{t-1}^{(j)}) \tag{4.11}$$

where the weighted particles $\{q_t^{(j)}, w_t^{(j)}\}_{j=1}^n$ represent the sampled six-DoF grasp poses with confidence score given by classifier. The initial hypothesis of particles $q_t^{(j)}$ are uniformly generated in the 3D workspace with the identical weights. For each hypothesis, we extract the grasp features and compute the weight $w_t^{(j)}$ by normalizing the confidence score output by classifier. Importance sampling is then performed with resampling process to

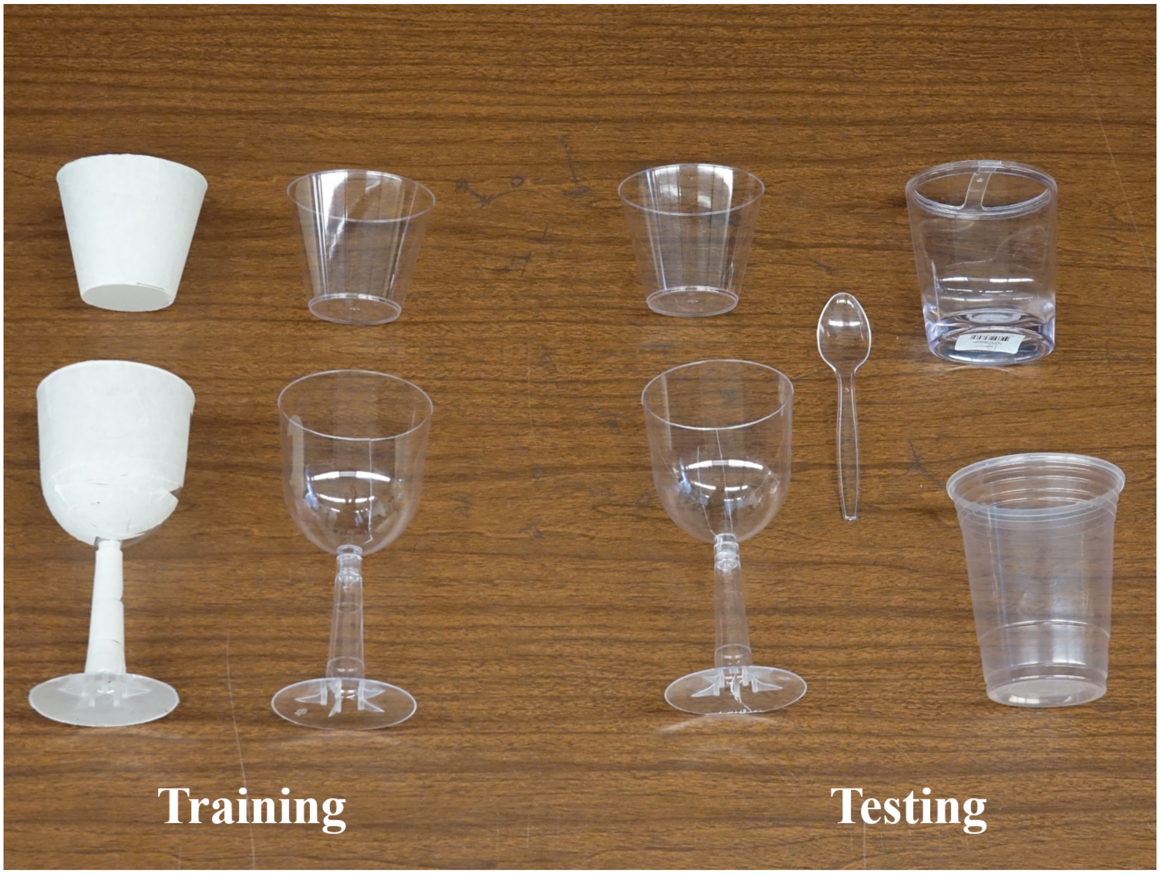


Figure 4.6: Training and testing objects for evaluating our *GlassLoc* algorithm. Two objects are used in training: wine cup and short cup (wrapped object for generating point cloud). Five objects are used in testing: wine cup, toothbrush holder, spoon, short cup, and tall cup.

concatenate grasp hypothesis to high weights region. In our case, we don't have actual action between two states, instead, we model the state transition in action model as zero-mean Gaussian noise over $SE(3)$. In other words, after we obtain resampled grasp poses (particles), we diffuse the particles by adding Gaussian noise over $(x, y, z, roll, pitch, yaw)$ to generate the new set of particles. Our convergence criterion is a fixed number of iterations.

Object	Trials	Success Rate
Toothbrush Holder	60	0.92
Wine Cup	50	0.82
Short Cup	40	0.65
Tall Cup	40	0.88
Spoon	30	0.70
Overall	220	0.81

Table 4.1: Object-wise grasp performance

	scene (a)	scene (b)	scene (c)	scene (d)	scene (e)	scene (f)	scene (g)	scene (h)
Number of Total Objects	2	2	2	2	3	3	4	4
Number of Manipulation Runs	10	10	10	10	10	10	10	10
Object Grasp Percentage	0.70	0.80	1.0	0.75	0.87	0.43	1.0	0.85

Table 4.2: Results of manipulation experiments for eight scenes. The first row shows the number of object in the scene. Number of manipulation runs shown in row two refers to the task runs for the scene. The object grasp percentage refers to successful picking ratio over all trials for each scene.

4.5 Results

4.5.1 Experimental Setup

To evaluate *GlassLoc* , we ran a series of experiments with a first generation Lytro camera and a Michigan Progress Fetch robot. The Lytro camera is mounted on the wrist of the robot and triggered by on-chip Wi-Fi to take images. In the meantime, the robot will record the camera view pose based on the current transformation from robot base to the camera. The Lytro camera intrinsic calibration and distortion correction is conducted using the toolbox created by Bok et al. [98]. The raw light field image is then decomposed into 9×9 sub-aperture images with resolution of 328×328 pixels. The boundary sub-aperture images usually have strong color noise because of the lens edge affect. In our implementation, we only keep 7×7 sub-aperture images and for each image. For each image, we crop 4 pixels at the margin.

We use two objects to construct our training samples: wine cup and short cup (Fig-

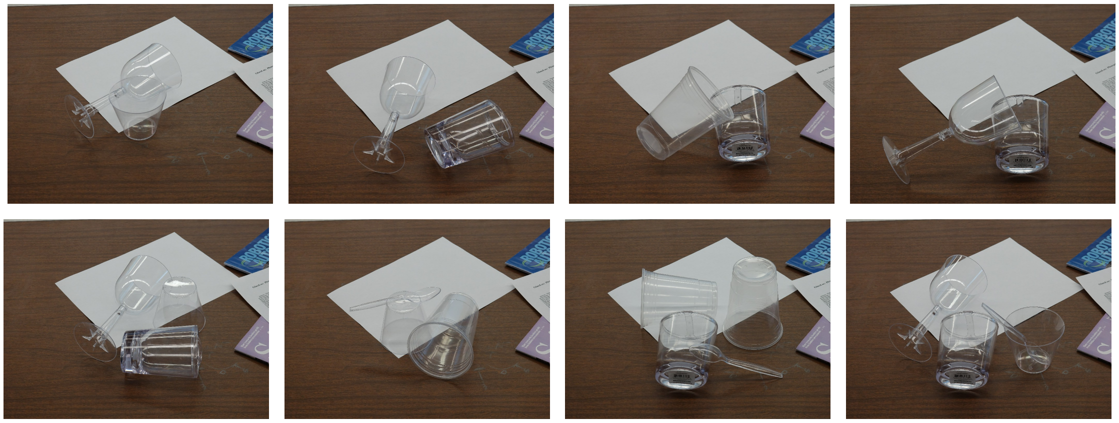


Figure 4.7: Eight scenes for evaluating *GlassLoc* pipeline. We randomly choose a number of transparent objects from the test set and put them on the table for the robot to perform manipulation on.

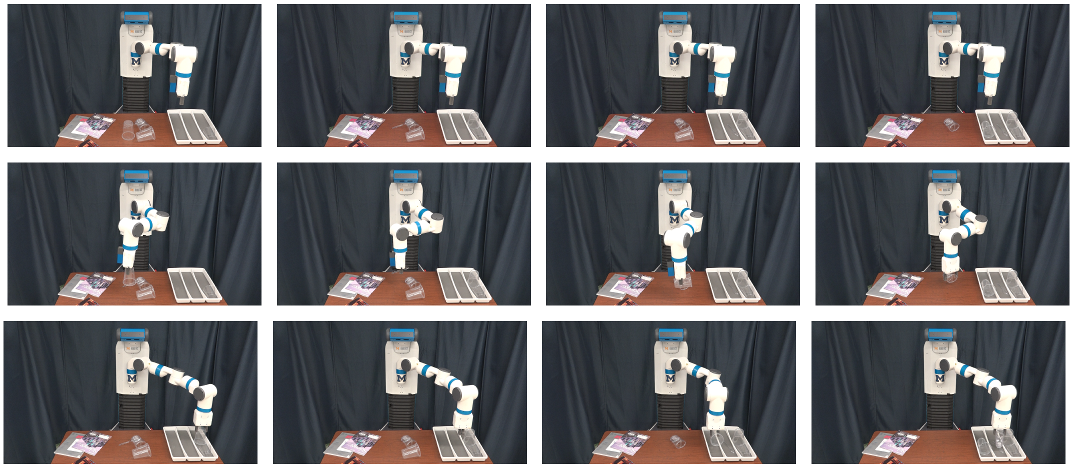


Figure 4.8: The robot successfully picks and places all transparent objects in scene (g). Each column shows the pick and place action over one object in the scene.

ure 4.6). We generate approximate 10k positive grasp samples and 15k negative grasp samples from 50 scenes containing one or more object instances. For each grasp sample, we extract corresponding graspable volume from DLV with actual size $0.10 \times 0.10 \times 0.06$ $(meters)$ and grid density $100 \times 100 \times 60$ $(points)$. We further extract gray-scale image features and resize them into 100×100. Features are concatenated into nine channels and trained on LeNet structure. We keep the default structure and parameter settings of LeNet implementation in Tensorflow except the number of nodes in the output layer (2 in our case).

The DLV construction algorithm is implemented in MATLAB with parallel computing. A DLV is sampled in a $1.0 \times 1.0 \times 1.0$ $(meters)$ box with grid density at $1000 \times 1000 \times 1000$ $(points)$.

In grasp search step, we use 100 particles with 100 iterations in our experiment. The covariance for diffusing grasp pose after each filtering iteration is set to 10^{-4} $(meter^2)$ and 0.03 (rad^2) for translation and rotation respectively.

Our implementation takes 2 minutes per view to extract sub-aperture images and 10 minutes to construct DLV on an unoptimized MATLAB code. The light field image decoding and ray corresponding are the current bottlenecks.

4.5.2 Evaluation

We evaluate our *GlassLoc* manipulation pipeline on eight transparent clutter scenes as shown in Figure 4.7. In each scene, the number of objects ranges from two to four with different pose configurations. For each manipulation run, light field images are taken from two camera poses to construct DLV. After particle filtering reaches the convergence criterion, we randomly select one grasp pose and send it to the execution module. Our robot motion planning and execution module is built on TRAC-IK [99] and MoveIt! [100]. For each scene, we perform 10 manipulation runs. We will terminate one run whenever all objects are successfully picked or the number of manipulation trials exceed the number of objects.

The manipulation results of each scene are established in Table 4.2. Object grasp percentage is calculated based on how many objects have been successfully picked over the total number of objects that should be picked in all runs of a scene. We also show the pick success rate for each object in Table 4.1.

Table 4.2 shows that the object grasp percentage is over 75% in most of the scenes. Our *GlassLoc* algorithm can generate enough reliable grasp poses based on our DLV constructed from light field observations in complex scenes where four transparent objects are randomly cluttered. The grasp percentages of these two scenes are 100% and 85% respectively.

Notably, our overall grasp success rate is 81% for the transparent cluttered environments in 220 grasps. During our experiment, we find that the short cup has the lowest grasp success rate. In most cases, it was squeezed and then slipped out from the gripper. The reason is two fold: one is that the surface of the short cup is sharply tilted, which prevents the robot from performing force closure grasping, the other is that the parallel jaw gripper hasn't been equipped with force sensors and is likely to squeeze the cup.

4.6 Summary

In this chapter, we have contributed the *GlassLoc* algorithm for robot manipulation in transparent clutter. We use multi-view light field observations to construct the Depth Likelihood Volume as a plenoptic descriptor to characterize the environments with multiple transparent objects. We show that by our algorithm, the robot is able to perform accurate grasping in tabletop transparent cluttered environments.

CHAPTER 5

LIT: Light-field Inference of Transparency for Refractive Object Localization

5.1 Motivation

Recognizing and localizing objects has a wide range of applications in robotics, and remains a very challenging problem. The challenge comes from the variety of objects in the real world and the continuous high dimension spaces of object poses. The diversity of object materials also induces strong uncertainty and noise for sensor observations. Existing works and datasets [28, 63, 58] cover a variety of texture-rich objects with distinguishable features between different types of objects. Several other works [108, 62] cover textureless objects with Lambertian surfaces, where robot sensors can still perceive rich depth information. However, many of these assumptions for objects with Lambertian surface properties are ill-posed for transparent objects.

The challenges imposed by transparency are multidimensional. First, non-Lambertian surface texture is highly reliant on the environment lighting and background appearance. Specifically, transparent surfaces will produce specularity from environmental lighting and project distorted background texture on their surfaces due to refraction. Second, transparent object depth information cannot be correctly captured by RGB-D sensors, which are commonly used by current object recognition and localization methods. This limitation imposes difficulties in collecting transparent object pose data using current labeling

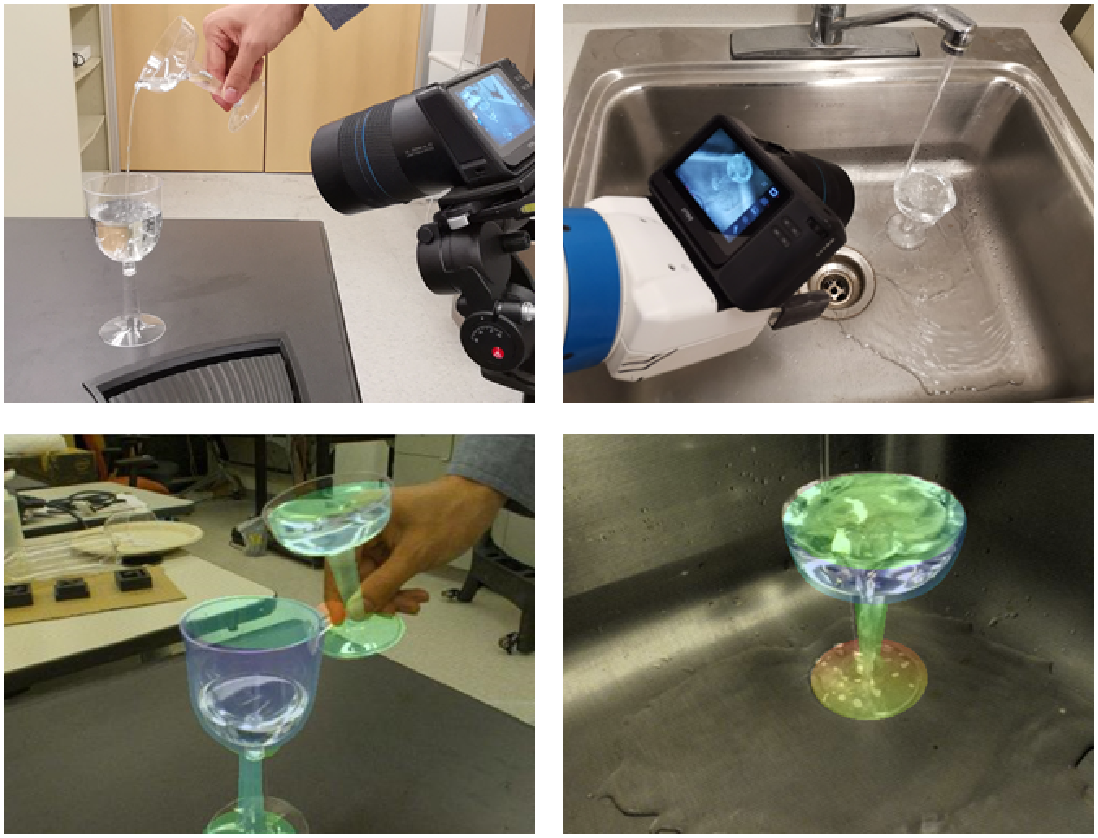


Figure 5.1: Demonstration of our *LIT* pipeline. (Top row) Lytro Illum camera is mounted on the tripod and robot arm to capture the transparent objects in challenging environments. (Bottom row) final estimated poses are overlapped to the center view of the observed light-field image.

tools [109]. As a result, transparent objects remain effectively invisible to robots using the sensors.

Recently, several works [103, 38] showed promising results using light-field (or plenoptic) photography in perceiving transparent objects. For example, Zhou *et al.* [110] generated grasp poses for transparent objects by classifying local patch features in a *Depth Likelihood Volume (DLV)* plenoptic descriptor. However, capturing and labeling over light-field images is time-consuming and computationally costly. Synthetic data is an alternative for image generation and has shown encouraging results in object recognition and localization. Georgakis *et al.* [111] rendered photorealistic images by projecting the object texture model on the real background for training object detectors. Tremblay *et al.* [58] proposed DOPE as an end-to-end pose estimator using domain randomization and photorealistic rendering [112]. We similarly address the problem of transparency using photorealistic rendering and light-field perception.

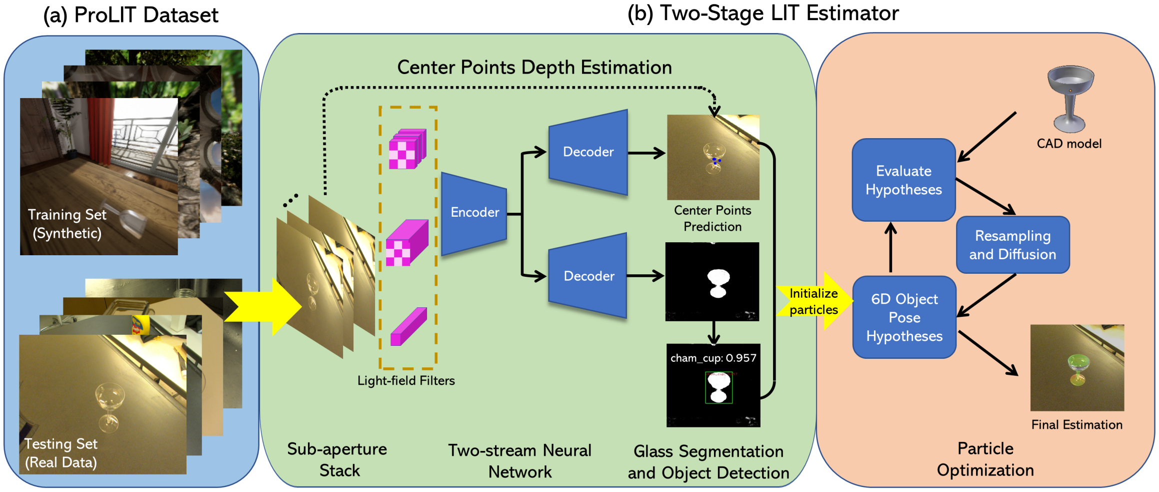


Figure 5.2: An overview of the *LIT* framework with the *ProLIT* dataset. (a) *ProLIT* contains 75,000 synthetic light-field images in training set and 300 real images with 442 object instances in testing set. (b) *LIT* estimator is a two-stage pipeline. The first stage takes light-field images as input and outputs transparent material segmentation and object center point prediction. The segmentation results are passed through a detection network to obtain object labels. In the second stage, for each predicted center point, we predict point depth likelihood by local depth estimation using Depth Likelihood Volume. The particle optimization samples over center points and converge to the pose that best matches the segmentation results.

In this chapter, we propose *LIT* [113] as a generative-discriminative method for recognition and pose estimation for transparent objects. Within *LIT*, we introduce 3D convolutional light-field filters as the first layer of our neural network. This neural network is trained purely with synthetic data from a customized light-field rendering system for virtual environments. At run time, the output of this trained neural network is used as input to a generative inference. The pose estimates resulting from this inference are then used to perform grasping and manipulation tasks. We introduce the ProgressLIT light-field dataset (*ProLIT*) for the task of transparent objects recognition, segmentation, and pose estimation. The *ProLIT* dataset contains 75,000 synthetic light-field images and 300 real images from

Lytro Illum light-field camera labeled with segmentation and 6D object poses. We show the efficacy of *LIT* with respect to state-of-the-art end-to-end methods and a generative method on our proposed *ProLIT* transparent object dataset. We additionally present a demonstration of using *LIT* for a purposeful manipulation task of building a champagne tower in a sparsely textured environment.

5.2 Related Work

5.2.1 Pose Estimation for Robot Manipulation

6D pose estimation remains a central problem in robot perception for manipulation in recent years. Deep learning methods have been a prevalent approach to perform accurate and fast inference for this problem. Xiang *et al.* [57] proposed PoseCNN to recognize and estimate objects and their 6D poses by decoupling translation and rotation separately in a neural network structure. Other end-to-end method methods have explored using synthetic data in training [58, 59], pixel-wise voting over keypoints [60, 61], and residual networks to iteratively refine object poses [62, 63]. Hybrid (or generative-discriminative) methods can achieve better performance by using deep networks to give hypotheses of object poses followed by a second stage of refinement. To get the final pose estimates, a variety of methods have been proposed for the second stage, including probabilistic generative inference [28, 114], template matching [115], and point cloud registration [108, 116].

Most deep learning methods for pose estimation are focused on texture-rich objects or those with texture-less but Lambertian surfaces [115, 108]. Transparent objects bring challenges in two main aspects, where there is: 1) no reliable depth information, and 2) no distinguishable environment-independent color textures. Prior works [35, 34] have used invalid readings from depth camera to extract object contours for pose estimation. However, these methods rely on the Lambertain reflections of the background surface to establish reliable contour of transparent objects. We take inspiration from these ideas for perception from light-field observations in two ways. First, a decent detection or segmentation intermediate result plays an important role in restricting the search area of the 6D object pose. Further, a deep network trained on a large, elaborately designed synthetic dataset can reach similar performance with those trained on real world data.

5.2.2 Light-field Perception for Transparency

The foundation of light-field image rendering was first introduced by Levoy and Hanrahan [74] for the purpose of sampling new views from existing images. Since the seminal work, light-field cameras have shown advancement in performing visual tasks in challenging environments with transparency and translucency. Maeno *et al.* [91] proposed the light-field distortion features from epipolar images for recognizing transparent objects. Recent work by Tsai *et al.* [105] further explored the light-field features to distinguish transparent and Lambertian materials. The result showed that the distortion features in the epipolar images can be used to distinguish materials with different refraction properties. Apart from refraction, specular reflection is another unique property carried by transparent materials. Tao *et al.* [117] investigated the line consistency in the light-field images with a dichromatic reflection model that removes the specularity from the images. Alperovich *et al.* [118] proposed fully convolutional networks to separate specularity in light-field images. In robotics, Zhou *et al.* [103, 110] created a plenoptic descriptor called DLV to model the depth uncertainty in a layered translucent environment. Based on this DLV, the object poses and grasp poses for robot manipulation are estimated using generative inference. Our proposed *LIT* method is built on these ideas above and leverages the power of discriminative and generative methods with data generation using photorealistic rendering.

5.3 LIT Estimator

Given an input light-field image L, the objective of *LIT* estimator is to infer the objects label l and their poses q in $SE(3)$. The pose q represents the transformation from object local coordinate frame to the camera coordinate frame. For a light-field image L with spatial resolution $H_s \times W_s$ and angular resolution $H_a \times W_a$, we assume the camera coordinate frame overlaps with the center view image's coordinate frame. The object pose q is defined in center view and parameterized into 3D translation and 3D orientation in quaternion.

5.3.1 LIT Pipeline

The two-stage *LIT* pipeline is shown in Figure 5.2. The first stage consists of a two-stream neural network that outputs pixel-wise image segmentation and 2D object center point locations. This output is followed by a detection network that classifies object labels l and clusters the corresponding center points. For each estimated center point, we construct its local DLV to generate depth estimates. The second-stage is a particle optimization initialized based on network and depth estimates, that converges to the final 6D poses.

There are several insights incorporated in the pipeline design. First, the segmentation decoder branch in the first neural network performs transparent material segmentation rather than object-class or instance segmentation. This distinction means it only decides whether a pixel belongs to a transparent material or not. The rationale for this classification is that pixel values within transparent object areas highly depend on the background and material property, rather than object types. Thus, it is difficult for a single network to distinguish different objects from raw pixel values. In addition, the center point estimation branch does not regress multiple keypoints which is common in texture-rich object pose estimation networks [60, 61]. The further rationale is that transparent objects lack features that are independent to object poses and environmental changes, such as background and lighting. In our work, we only predict the 2D object center point location.

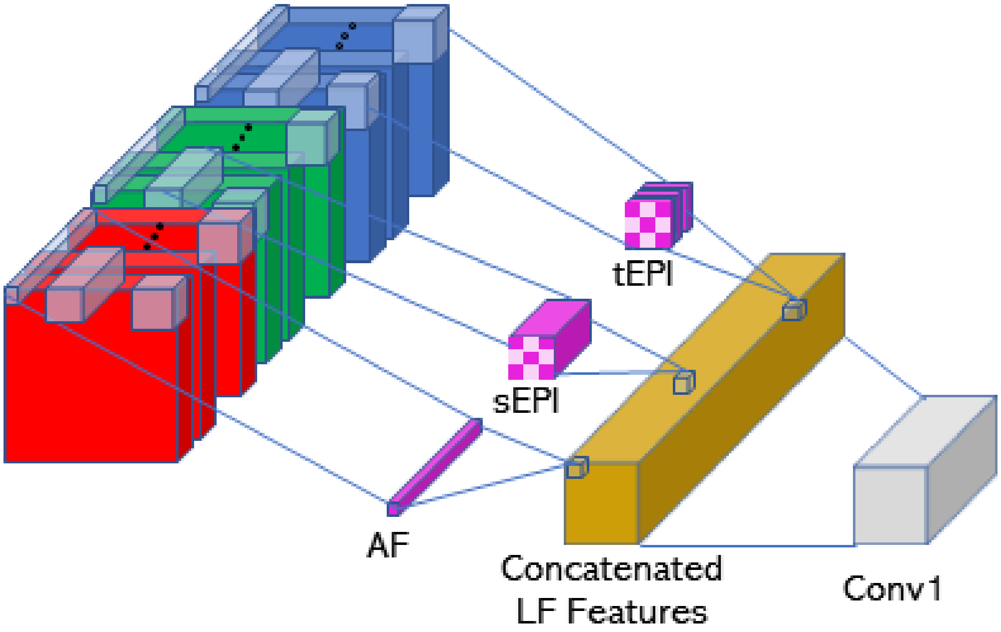


Figure 5.3: Illustration of three light-field filters. Angular filter (AF) has dimension $1 \times 1 \times (H_a \times W_a)$ to capture features in angular pixels. sEPI and tEPI filters have sizes of $n \times n \times W_a$ and $n \times n \times H_a$ respectively, here n refers to kernel size. tEPI also has a dilation W_a. All features will be concatenated together after passing filters.

5.3.2 Network Architecture

As shown in Figure 5.2, the input light-field image is first decomposed into sub-aperture image stacks. This structure gives a 3D matrix with size $H_s \times W_s \times (H_a \times W_a)$ replicated for each of the R, G, B channels. The stacks are then going through three light-field filters: angular filter [119], 3D sEPI filter, and 3D tEPI filter.

- **Angular Filter**. The angular filter aims to capture the reflection property of 3D surface points in the direction space of light ray. For instance, a non-Lambertian surface will establish different colors in a single angular patch while it will be nearly identical for a Lambertian surface. The angular filter can be expressed as an operation over each pixel (x, y) in spatial space (for the jth filter):

$$g(\sum_{s,t} w_i^j(s,t)L_i(x,y,(s,t))) \tag{5.1}$$

where $g(\cdot)$ is the activation function, s and t are the angular indices, w_i^j is the weight in the angular filter, $i \in \{r, g, b\}$ is the color channel, and $L_i(x, y, (s, t))$ is the 4D light-field function.

- **3D EPI Filters**. Transparent surfaces will produce distortion features because of refraction. In the epipolar image plane, it will produce polynomial curve patterns [105] which can be distinguished from the background texture without distortion. To capture distortion features, we propose the epipolar filters using 3D convolution layers along the two angular dimensions s and t respectively. The 3D EPI filters can be expressed as:

$$\begin{gathered} g(\sum_{u,v,s} \tilde{w}_i^j(u,v,s)L_i(x+u,y+v,(s,t))) \\ g(\sum_{u,v,t} \hat{w}_i^j(u,v,t)L_i(x+u,y+v,(s,t))) \end{gathered} \tag{5.2}$$

  where (u, v) is the index of convolution kernel in spatial space, $\tilde{w}$, $\hat{w}$ are weights in sEPI and tEPI filters, and we assume the input and output have the same dimension in spatial space by proper paddings.

Passing through the three customized filters, the embedded features of light-field images are concatenated. The result goes into an encoder-decoder structure with two branches for image segmentation and object center point regression. The output of the segmentation branch is a pixel-wise segmentation of the center view image. Each center view pixel is then predicted to be on a transparent surface, in the background, or on the boundary between a transparent object and background in the image. The output of the center point branch are the 2D pixel offsets from each pixel to their estimated center position on the image, as well as a pixel-wise confidence values.

The loss in segmentation branch $\mathcal{L}_{seg}$ is defined as the cross-entropy loss normalized by class pixel probabilities [10]. The loss of center point regression is mainly following design in [60], although we only regress the center point positions. The learning goal for each pixel p inside the segmentation area $\mathcal{M}$ is to regress the offset h_p from its location c_p to the object center g_p on 2D image. In this way, the loss $\mathcal{L}_{pos}$ is expressed as:

$$\mathcal{L}_{pos} = \sum_{p \in \mathcal{M}} \|g_p - (c_p + h_p)\|_1 \tag{5.3}$$

where $\|\cdot\|_1$ denotes L^1 loss. Each pixel's estimation is associated with a confidence value b_p, and the confidence loss $\mathcal{L}_{conf}$ is defined as:

$$\mathcal{L}_{conf} = \sum_{p \in \mathcal{M}} \left\|b_p - \exp\left(-\tau \|g_p - (c_p + h_p)\|_2\right)\right\|_1 \tag{5.4}$$

where τ is a modulating factor and $\|\cdot\|_2$ denotes L^2 loss. The overall loss $\mathcal{L}$ is calculated as:

$$\mathcal{L} = \alpha \mathcal{L}_{seg} + \beta \mathcal{L}_{pos} + \gamma \mathcal{L}_{conf} \tag{5.5}$$

where α, β, γ modulates the importance of segmentation, regression and regression confidence respectively. In practice, we select $\alpha = 1, \beta = 8, \gamma = 2$ from initial experimentation.

An object detection network is appended to differentiate object types based on geometry shapes from segmentation results. Specifically, the network takes the result of segmentation decoder branch as input and gives bounding boxes with object labels. Detected bounding boxes also play the role of clustering object center points. The overall output of the first stage is a set of bounding boxes, each with an object label and a set of object center points, which serves as the initial distribution of object center locations for the next stage.

Directly regressing the depth of center points without depth observation is difficult for neural networks. Instead, we deploy a DLV plenoptic descriptor [103] to describe the depth of a single pixel as a likelihood function rather than a deterministic value. The advantage of using a DLV is that depth likelihood can be naturally leveraged into generative inference framework in a sample initialization step. The likelihood $D(x_c, y_c, d)$ of a given center point located at (x_c, y_c) in center view image plane I_c can be calculated as:

$$D(x_c, y_c, d) = \frac{1}{N} \sum_{a \in A \setminus I_c} T_{a,d}(x_c, y_c) \tag{5.6}$$

where A is a set of sub-aperture views, $T_{a,d}(x_c, y_c)$ is the function to calculate the color intensity and gradient cost of pixel (x_c, y_c) on a specific depth d. $\frac{1}{N}$ is a normalization term that maps cost to likelihood. Detailed implementation can be referred in [103, 110].

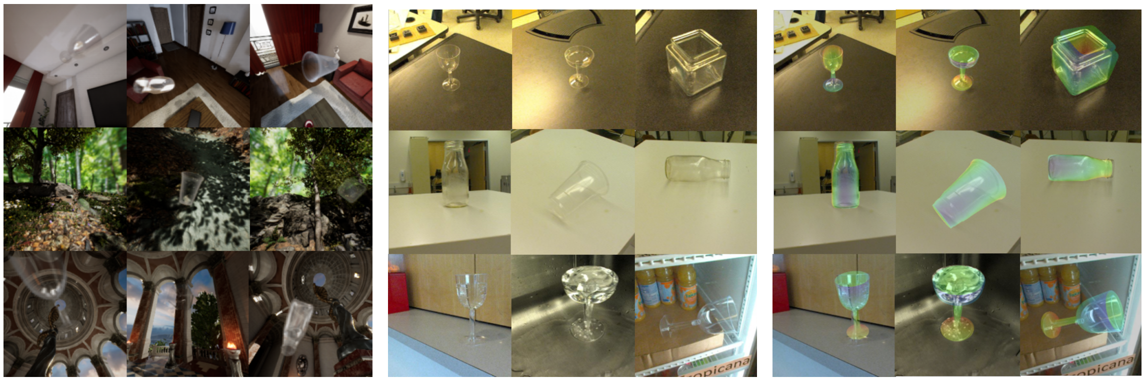
5.3.3 Particle Optimization

The second stage of pipeline estimates the 6D pose of transparent objects in a sampling-based iterative likelihood reweighting process [120]. Object pose samples are initialized based on the center point locations from the first stage. During the iterations, rendered samples are projected to 2D image and their likelihoods are calculated as the similarity between the projected rendered samples and segmentation results.

5.3.3.1 Sample Initialization

Each sample is a hypothesis of object 6D pose. Its 3D location can be derived from 2D image coordinate (u, v), depth d and camera parameters. In this way, the probability distribution of 3D center point locations is formed by leveraging center point candidates and depth likelihood volume results:

$$\begin{aligned} &u = c_x + f_x \frac{x}{z}, \quad v = c_y + f_y \frac{y}{z}, \quad d = z \\ &p(X = x, Y = y, Z = z) = b(u, v) D(u, v, d) \end{aligned} \tag{5.7}$$



(a) Training Set (b) Testing Set (c) Result

Figure 5.4: (Left) example synthetic light-field images rendered in three different environments. (Middle) example test images in different backgrounds and different pose configurations. (Right) results visualization by overlaying estimated poses to the original test images.

where b is the confidence value of object center point estimation from neural network, f_x, f_y, c_x, c_y are camera intrinsic parameters, and D is likelihood from DLV in Equation (5.6). Notably, here we only perform DLV construction in a small region near the predicted centers. We perform importance sampling over this distribution to initialize the pose sample locations. The initial orientations of samples are randomly selected in $SO(3)$ space.

5.3.3.2 Likelihood Function

The probability of each sample during iterations is calculated using the likelihood function, represented as the similarity between the projected rendered object point cloud and segmentation results from neural network. Specifically, the object points in its local frame are transformed by the sample pose and then projected to 2D image plane. The likelihood function is composed of intersection over union scores of projected rendered point clouds and segmentation masks on transparent material and its boundary:

$$weight = \eta \frac{|S_{pcd} \cap S_{seg}|}{|S_{pcd} \cup S_{seg}|} + (1-\eta) \frac{|\partial S_{pcd} \cap \partial S_{seg}|}{|\partial S_{pcd} \cup \partial S_{seg}|} \tag{5.8}$$

where S_{pcd} is the silhouette of projected rendered point cloud, S_{seg} is the pixels segmented as transparent materials, ∂S_{pcd} and ∂S_{seg} are the sets of boundary pixels of S_{pcd} and S_{seg} respectively. η is set to modulate importance of boundaries.

5.3.3.3 Update Process

We follow the procedure of iterative likelihood reweighting to produce pose estimations. The initialized samples are assigned the same weights. Then the process of calculating likelihood values, resampling based on weights, and sample diffusion is repeated in every iteration. During diffusion step, each pose sample is randomly diffused in $SE(3)$ space in translation and rotation with Gaussian noise. The algorithm terminates when the maximum sample weight reaches a threshold, or the iteration number reaches the limit.

5.4 ProLIT Light-field Dataset

We propose the *ProLIT* light-field image dataset for the task of transparent object recognition, segmentation, and 6D pose estimation. This dataset contains a total of 75,000 synthetic images and 300 real-world images with 442 object instances, each labeled with pixel-wise semantic segmentation and 6D object poses. Figure 5.4 shows examples of synthetic images, real-world images and estimation results from *LIT*. There are 5 instances of objects included in the dataset: wine cup, tall cup, glass jar, champagne cup, starbucks bottle with different geometric shapes. The images are captured using a Lytro Illum camera which is calibrated by the toolbox described in [121]. The spatial resolution of the calibrated image is 383×552, and the angular resolution is 5×5 (extracted from 9×9 sub-aperature images with stride 2). The object poses in testing data are labeled by reprojecting objects directly into the center view image and matching with observations.

The light-field rendering pipeline is built on NDDS [112] synthetic data generation plugin in Unreal Engine 4 (UE4). The created virtual light-field capturer has an angular resolution 5×5 and spatial resolution 224×224. The baseline between the adjacent virtual camera is set to 0.1cm. We generate data in three UE4 world environments: room, temple, and forest. The target objects are rendered using the transparent material. Objects move in two ways in the environment: flying in the air with random translation and rotation, or falling freely with collision and gravity enabled. When the objects move, the virtual light-field capturer will track and look at them with arbitrary azimuths and elevations. Ray tracing is enabled when capturing images.

Notably, lighting condition is critical for non-Lambertian surface and cannot be perfectly reproduced in simulated environments as in the real world. To overcome this issue, we use the domain randomization approach to highly randomized the lighting conditions including color, direction, and intensity in each simulated environment. For example, in the simulated room environment, we set an ambient light with randomized intensity to mimic the change of sunlight during the daytime. For indoor light sources, we add a collections of point lights to mimic fluorescent lamps with randomized intensity and randomized on and off. Together with fluorescent lamps, we also add separated point lights with randomized intensity, color, and direction to mimic the lights from small household electrical devices. The randomization from all these light sources will help the neural network learn the underlying invariant features, for instance, object shape and non-Lambertian surface property.

5.5 Experiments

We choose 64 light-field filters as the first feature extraction layer. The *LIT* network uses VGG16 [122] as backbone architecture and initialized with pre-trained model on ImageNet [123]. The segmentation branch outputs pixel-wise labels from over three classes: background, transparent, boundary. The center points prediction branch outputs pixel-wise offset for each segmented pixels. The detection network is a Faster R-CNN network [8] with VGG16 backbone. The input to the network is the binary masks of transparent object

Method	gAcc	mAcc	mIoU	wIoU	mBFS
2D	0.871	0.500	0.228	0.397	0.140
AF only	0.917	0.501	0.318	0.582	0.197
LIT	**0.954**	**0.520**	**0.455**	**0.854**	**0.390**

Table 5.1: Comparison of *LIT* and baseline methods on transparent material segmentation. The performance is quantified through global accuracy (gAcc), mean of class accuracy (mAcc), mean of Intersection over Union (mIoU), weighted IoU (wIoU), and mean BF (Boundary F1) contour matching score (mBFS). The definitions are detailed in [5]. 'AF only' here refers to the baseline method with only angular filters.

segmentation and the output is bounding boxes with object labels.

5.5.1 Evaluation of Light-field Filters on Image Segmentation

Segmentation is taken as the optimization target in our second stage which is critical to *LIT* pipeline. We first compare with two baseline methods to show the advantage of using light-field images with three light-field specific filters. One baseline takes input of 2D center view image, which passes through the same neural network structure as *LIT* except for light-field filters, the other is an ablation study with only the angular filter. All three networks are trained on the synthetic dataset containing 75,000 images. Table 5.1 shows segmentation accuracy results, where *LIT* achieves better performance than baseline methods in all metrics. Through the comparison with single RGB input, we show that lighting direction information captured inside light-field images helps distinguish transparent pixels from the background. Through the comparison with only an angular filter, *LIT* also achieves higher accuracy, showing that both angular features and EPI features are important in contributing to segmenting transparent objects.

5.5.2 Evaluation of Pose Estimation

We compare the 6D pose estimation results of *LIT* against a state-of-the-art RGBD-based transparent object depth completion pipeline, ClearGrasp [124], a state-of-the-art RGB-based general-purpose object pose estimator, DOPE [58], a state-of-the-art RGB-based

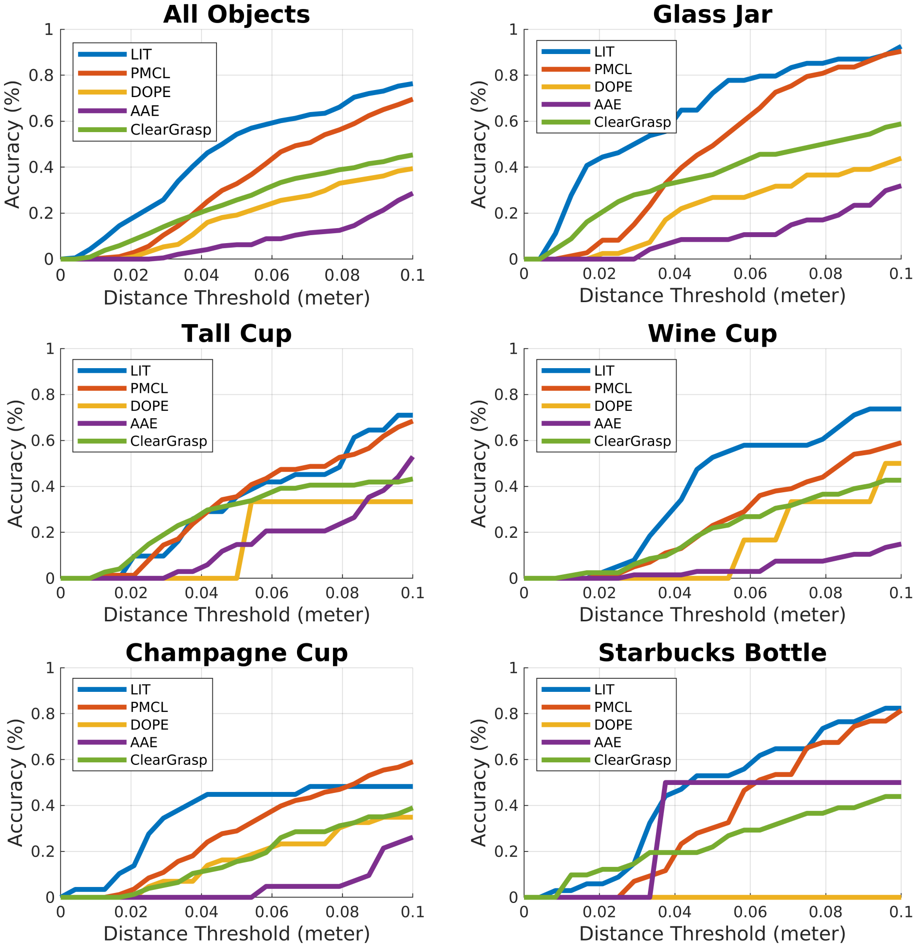


Figure 5.5: Comparison of 6D pose estimation results with respect to ADD-S and Accuracy Under Curve metric.

textureless object pose estimator, Augmented Autoencoder (AAE) [108], and a generative light-field based transparent object pose estimation method, PMCL [103].

For the fair comparison with ClearGrasp, we add the second stage generative inference as 6-DoF pose estimator. Noticeably, the generative inference pipeline we used for ClearGrasp and *LIT* is identical. For the fair comparison with DOPE and AAE, we make both methods compatible with light-field inputs. We add the three light-field filters in Section 5.3 before the first encoder layer of DOPE network as well as AAE encoder network. We adopt Faster R-CNN network as the first stage object detector for AAE. All of the learning-based methods are trained with 75,000 synthetic images for 5 objects. In the second stage of *LIT* pipeline, we diffuse the particles with Gaussian noise $\mathcal{N}(0, 0.08)$ in translation and

$\mathcal{N}$(0, 0.4) in orientation. PMCL is a generative method which requires object labels and 3D search space. We initialize PMCL with ground truth object labels and a search volume with size $40 \times 40 \times 40$ cm^3 around the ground truth object locations. The convergence threshold of particle weights is set to 0.7. We use ADD-S metric [57] to evaluate the pose results of symmetric objects. We then show the accuracy curves in Figure 5.5 with a distance threshold of 0.1m. The Area Under accuracy-threshold Curve (AUC) and algorithm computation time per object are shown in Table 5.2.

AUC	wc	tc	gj	cc	sb	all	time(s)/obj
ClearGrasp	0.20	0.27	0.45	0.17	0.24	0.24	5
DOPE	0.14	0.16	0.21	0.16	0.00	0.18	< 1
AAE	0.04	0.15	0.10	0.05	0.32	0.08	< 1
PMCL	0.24	**0.32**	0.46	0.28	0.34	0.32	300
LIT	**0.38**	**0.32**	**0.62**	**0.35**	**0.44**	**0.45**	10

Table 5.2: Comparison of *LIT*, DOPE, AAE, PMCL, and ClearGrasp on transparent object pose estimation. The column headings wc, tc, gj, cc, and sb refer to the wine cup, tall cup, glass jar, champagne cup, and starbucks bottle objects, respectively. All columns, except for the last, refers to the area under the curve (AUC) for accuracy-threshold values for the symmetric objects metric (ADD-S), shown in Figure 5.5.

From the result plots, we find that *LIT* performs much better than DOPE and AAE, and better than PMCL and ClearGrasp. For DOPE, we conjecture directly regress the eight 3D bounding box vertices and their relations is not an optimal strategy for transparent objects. First, DOPE's object recognition is embedded in the network but the transparent object's texture is not informative to distinguish different objects. Secondly, the eight vertices of 3D bounding boxes are ambiguous for networks to learn the features because of the object symmetry and lack of distinguishable features for transparent objects. For AAE, it is possible that it is difficult for the latent variable to learn the embedded features to distinguish different orientations of transparent objects. Also, it is difficult for the first stage detector to provide accurate location of the transparent objects, which heavily influences the second stage translation and orientation estimation. Since PMCL is provided with ground truth labels and search space, it performs comparatively well in the testset. However, PMCL uses single-view DLV as matching target which includes noise from specularity and distortion from transparent surfaces. Furthermore, DLV construction is computationally expensive, which takes an average 300 seconds for one object. As for ClearGrasp, like other RGBD-based methods, makes an assumption that background provides valid and smooth depth points that can infer the accurate segmentation or contour information of transparent objects. This assumption holds most of the time when transparent objects are separated and background objects have opaque surfaces. But lots of real world scenarios will break the assumption, for example, kitchen sinks with running water and wooden table with polished and reflective top surface. On the contrary, light-field based methods are well-posed for those scenario because of its capability to describe different surfaces by its reflection property. To show that, we further separate the testset into two categories– opaque background and challenging background – to evaluate five methods' performance under different backgrounds. Overall, *LIT* pipeline provides better accuracy than all three baseline methods on the testing dataset with a relatively small computationally cost.

5.5.2.1 Opaque Background Results

To evaluate all methods' performance under opaque background, we select test cases whose background is fully visible from depth camera. Examples are illustrated in Figure 5.6. The corresponding results are shown in Figure 5.7 and Table 5.3.

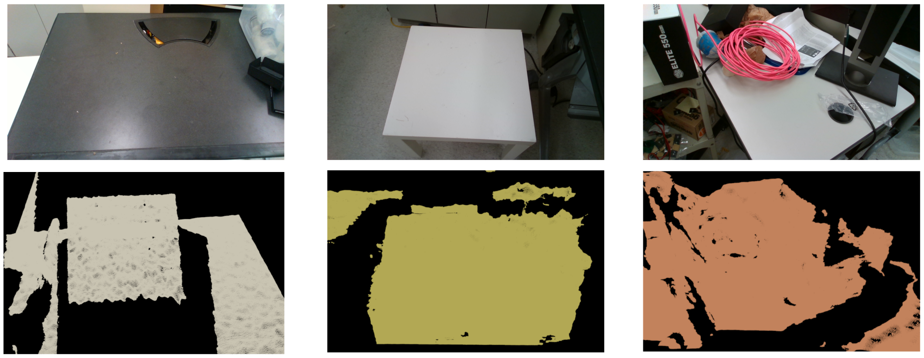


Figure 5.6: Example of RGB-D image pairs for opaque background. (Top row) color image of the background objects. (Bottom row) corresponding point cloud captured by the depth camera.

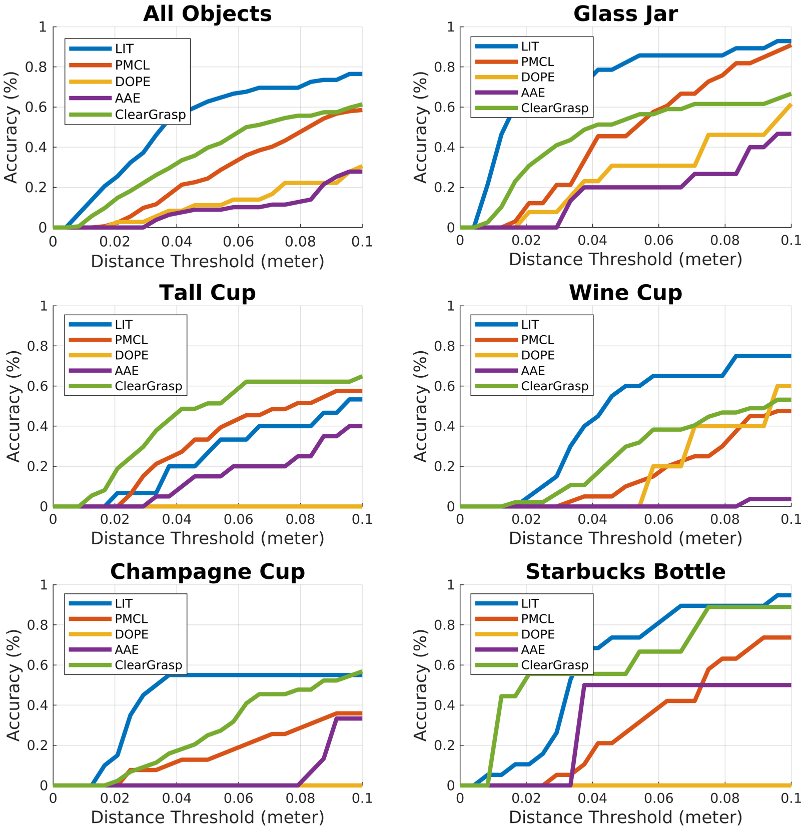


Figure 5.7: Comparison of 6D pose estimation results under opaque background with respect to ADD-S and AUC metric.

When a transparent object in an opaque background, its invalid depth readings will depict a distinguishable contour to indicate its shape and location information. With this insight, ClearGrasp further leverages the predicted normals, contact edges, and segmentation from color images to reconstruct the scene. Different from ClearGrasp, the light-field-based method directly distinguish different materials by their reflection and refraction features over light-field sub-aperture space. But in an opaque background, an RGB-D camera can obtain the same information by looking into the invalid readings in depth images. Results have shown that ClearGrasp can perform comparably well or even better for some objects compared with light-field-based methods. However, we see *LIT* still performs better in overall AUC, especially for wine cup and champagne cup. These two objects have thin handles which is difficult to capture if using invalid points as shape descriptor because

AUC	wc	tc	gj	cc	sb	all
ClearGrasp	0.25	**0.42**	0.45	0.26	**0.60**	0.35
DOPE	0.17	0.00	0.26	0.00	0.00	0.12
AAE	0.01	0.15	0.18	0.05	0.32	0.09
PMCL	0.16	0.30	0.45	0.16	0.30	0.26
LIT	**0.43**	0.25	**0.71**	**0.41**	0.59	**0.51**

Table 5.3: Pose estimation results of *LIT*, DOPE, AAE, PMCL, and ClearGrasp under opaque background.

it is always corrupted by noisy points from other parts. But for *LIT* , it will directly segment the pixels that belong to the target transparent objects from light-field image which won't be affected by part size and background noise.

5.5.2.2 Reflective Background Results

Unlike opaque background, reflective background poses more challenges for RGB-D cameras for object pose estimation. Figure 5.8 shows three examples of reflective background that are prevalent in our daily lives. The large area of invalid readings in the point cloud makes the assumption that the invalid depths belong to target transparent objects no longer holds. The results shown in Figure 5.9 and Table 5.4 establishes that light-field-based methods performs much better than the RGB-D based methods.

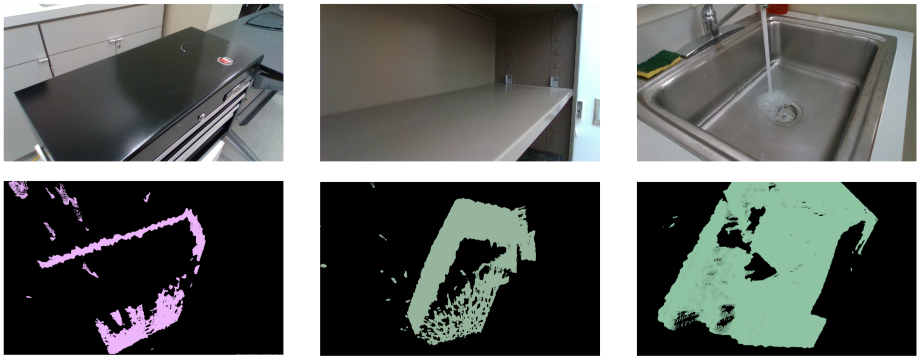


Figure 5.8: Example of RGB-D image pairs for reflective backgrounds. (Top row) color image of the background objects. (Bottom row) corresponding point cloud captured by the depth camera.

If we further calculate the AUC difference between Table 5.3 and Table 5.4, we can find that *LIT* is less sensitive to the background changes while ClearGrasp highly relies on the

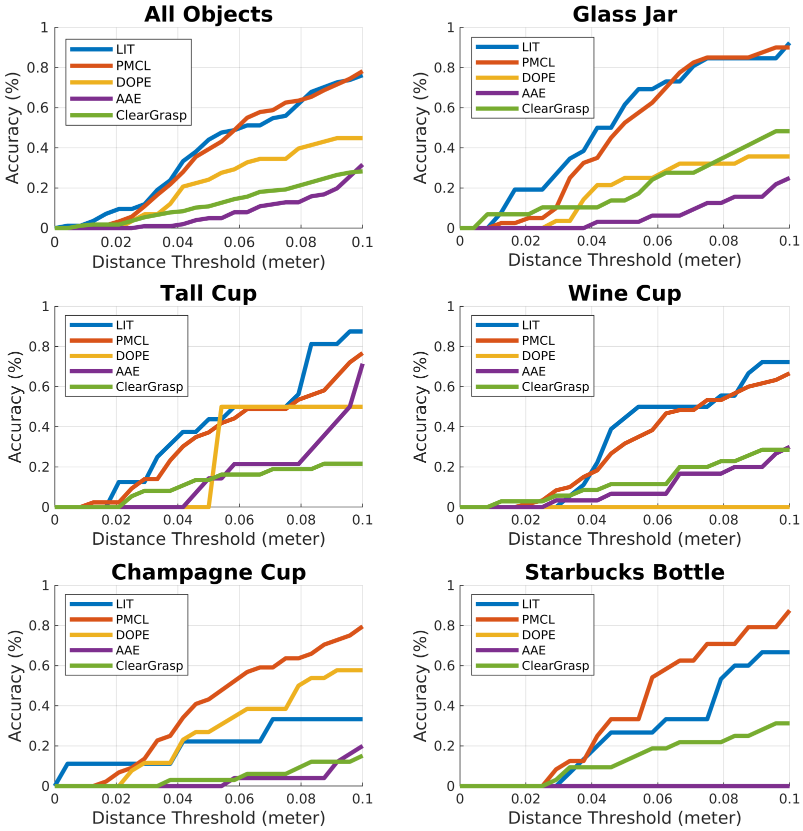


Figure 5.9: Comparison of 6D pose estimation results under reflective background with respect to ADD-S and AUC metric.

opaque background to provide clear boundary and contour of the target transparent objects.

5.5.3 Champagne Tower Demonstration

LIT is also integrated into a robotic manipulation pipeline for a purposeful manipulation task of building a champagne tower in a sparsely textured environment, as shown in Figure 5.10. In the initial setup, the champagne cups are randomly placed on a textureless white table. The Lytro Illum camera takes a light-field image and transfer the image with on-chip wifi. The Lytro camera's extrinsic matrix is calibrated with robot world frame. *LIT* then performs pose estimation over the scene followed by transforming pre-defined grasp poses to the observation. With the accurate pose estimates, the robot is able to pick up all champagne cups from the table and arrange them into a champagne tower.

AUC	wc	tc	gj	cc	sb	all
ClearGrasp	0.13	0.12	0.21	0.046	0.13	0.12
DOPE	0.00	0.24	0.18	0.27	0.00	0.22
AAE	0.09	0.16	0.06	0.03	0.00	0.08
PMCL	0.30	0.33	0.47	**0.39**	**0.37**	0.36
LIT	**0.43**	**0.39**	**0.52**	0.21	0.26	**0.37**

Table 5.4: Pose estimation results of *LIT*, DOPE, AAE, PMCL, and ClearGrasp under reflective background.

5.6 Summary

We introduce *LIT*, a two-stage generative-discriminative object and pose recognition method for transparent objects using light-field observations. *LIT* employs the learning power of deep networks to distinguish transparent objects across light-field sub-aperture images. We show that the network trained only on synthetic data can deliver a good segmentation on transparent materials, which is served as matching target for second stage pose estimation. Along with the method, we propose the light-field transparent object dataset including synthetic and real data for the tasks of object recognition, segmentation, and 6D pose estimation. We demonstrate the use of *LIT* for a purposeful robot manipulation task over transparent cups. However, our method still has limitations in cluttered environments where the first stage segmentation results cannot provide distinguishable object shapes for second stage refinement. Also, our *ProLIT* testing set is majorly collected in indoor environments representative the lab space used by the Laboratory for Progress. More specifically, our lab space has an array of fluorescent lamps, which can be treated as spot light, and a collection of other electric devices like monitors and TV screens which can be treated as point light. In the day time, there will be extra sun lights transmitted through windows but it won't be major light sources for our dataset. We aim to further extend to different light conditions in our future work. Other possible future works built on *LIT* could be instance-level segmentation based on transparent objects and single-view light-field depth estimation directly predicted by neural network.

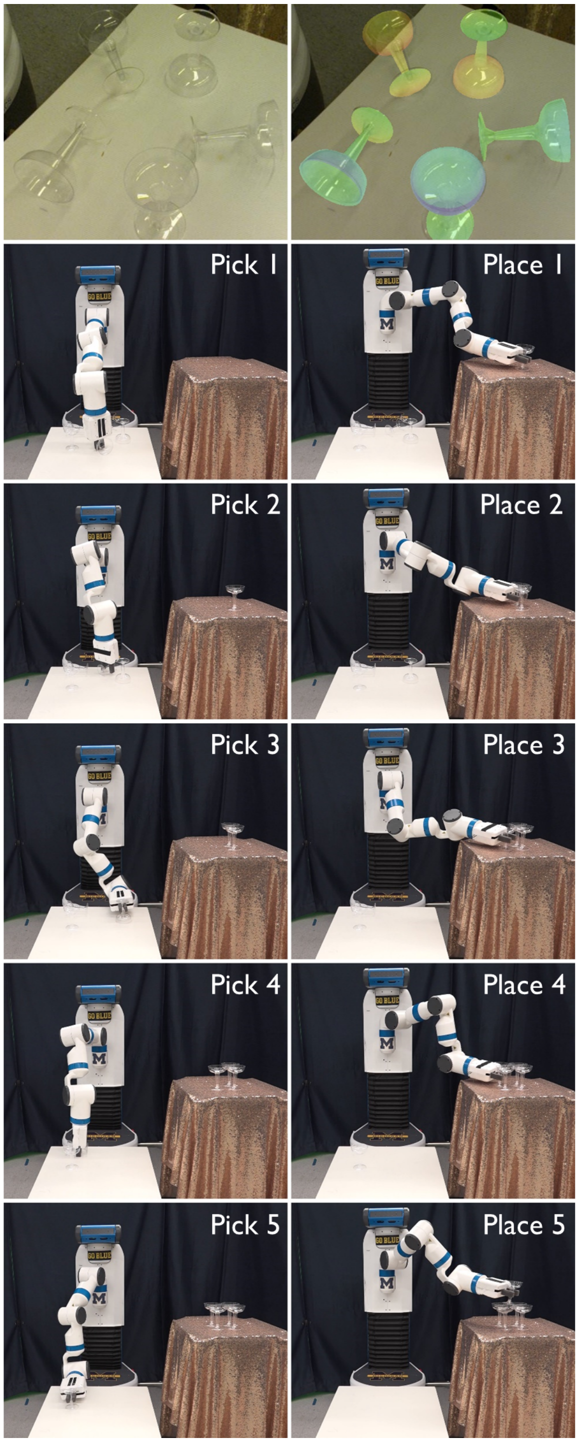


Figure 5.10: The robot is building a champagne tower by successfully picking and placing champagne cups on the table. The first row shows light-field observation (left) and pose estimation result from *LIT* (right). The following five rows show pick and place actions to finish the champagne tower.

CHAPTER 6

Conclusion and Discussion

6.1 Conclusion

A general-purpose robot can perceive and interact with all kinds of objects in the real-world, can be immensely powerful, and is an ultimate goal for the robotic community. To reach the goal, transparency and translucency is one of the biggest challenges of robotics perception. As such, roboticists must draw greater attention to the study of different sensor modalities and corresponding approaches for robotic manipulation under transparency and translucency.

Therefore, this dissertation focuses on the two major robotic perception challenges brought about by transparency and transparency: 1) invalid measurement in the depth domain and 2) highly variant textures in the RGB domain.

In the depth domain, we present a novel plenoptic descriptor *Depth Likelihood Volume (DLV)* as an alternative depth representation. DLV represents depth as a likelihood function by investigating light ray consistency in light-field sub-aperture images. Benefiting from this representation, *PMCL* (Chapter 3) and *GlassLoc* (Chapter 4) can perform generative or discriminative inferences under transparency and translucency. In an uncluttered environment, *PMCL* can estimate a single object's 6-DoF pose using particle optimization. With *PMCL*, we have shown a Fetch robot that can execute pick-and-place actions towards objects. *GlassLoc* further extends the ability of robots to perform manipulation under transparent clutter by learning local geometric features from a multi-view DLV. The learning process is carried out by a convolutional neural network and outputs feasible grasp poses with a confidence score. *GlassLoc* enables the robot to perform table cleaning tasks for household transparent objects.

In the RGB domain, we observe that a transparent object's reflection and refraction produces uneven color distribution and distorted epipolar lines in light-field sub-aperture images. By leveraging this insight with DLV, we present *LIT* (Chapter 5) as a two-stage generative-discriminative method for fast transparent object segmentation and localization. By learning reflection and refraction features using light-field filters, *LIT* is able to segment transparent objects and predict their object centers. Given segmentation and object centers, *LIT* performs generative inference over object locations. We demonstrate *LIT* outperforming four state-of-the-art object pose estimators and enables our robot to build a champagne tower from scratch.

In sum, this dissertation presents plenoptic-sensing-based perception for robotic manipulation approaches that leverages both the generative and discriminative methods for estimating transparent and translucent object pose and grasp pose under different scene clutterness, task complexity, and computation efficiency.

6.2 Strengths and Limitations

In this section, we first establish the strengths and limitations of using light-field perception in robotic manipulation over other mainstream sensors such as RGB and RGB-D camera. We then establish the strengths and limitations for each method proposed in this dissertation.

6.2.1 Comparison between Different Sensor Modalities

In Chapter 5, our experiments between RGB and light-field images on object segmentation indicate that light-field features largely increase the method's ability to distinguish different object surfaces. Further ablation studies between different light-field filters in the *LIT* network suggest that the different reflective and refractive features in the light-field sub-aperture space work jointly in improving the overall object segmentation results. More importantly, the following pose estimation results between three different sensor modalities have shown that light-field-based methods are also more robust in handling challenging real-world environments such as a sink with water or a steel shelf. Compared with RGB-D methods, which rely on the binary classification of object surfaces (valid/invalid depth readings), light-field-based methods can perform a more accurate and reliable classification over light direction space. Another benefit of using a light-field camera over an RGB camera is that it requires less information gain actions to sample light rays in the space. For instance, if we want to sample 100 different light ray directions for a scene, we need only perform one capture action by a light-field camera with 100 angular resolution while an RGB camera would need 100 actions to achieve the same results. Moreover, the light-field camera samples the light rays uniformly regardless of robot movement, while the RGB camera relies heavily on robot movement to guarantee its sampling quality.

Nevertheless, the limitations of light-field perception are also obvious. Because of the extra light direction information, a light-field image can easily reach 100 times or more in size compared with a conventional RGB camera. The larger image size also requires a sophisticated optical device, which results in greater difficulties in calibration and the need for more computational power and space to process the data. For instance, the *LIT* model used in Chapter 5 is twice as large as the RGB-input model, and the primary portion of the network weights are in the 3D light-field filter part. The size of the model further limits the application of the light-field in a time-sensitive task, such as real-time tracking. In the meantime, the large size of the light-field image limits its application in light-field video capturing. Most off-the-shelf commercial light-field cameras are capable of only capturing single frames of light-field images. Apart from its large size, a raw light-field image also requires a more complicated calibration step. Take a microlens array light-field camera, for example. The lenslet introduces an extra level of difficulty because of both its location and intrinsic need to be calibrated separately using special patterns [98]. Fortunately, researchers are increasingly trying to address these issues on the hardware end; see [125], for example.

6.2.2 Comparison between Proposed Methods

In this dissertation, the proposed light-field descriptor DLV can capture multiple layers of depth by investigating the ray consistency over sampled light directions. To capture the target object in a scene while sampling a light ray's direction in a wider cone of vision, we need to balance the distance between the light-field camera and the object. When performing DLV construction, the first generation Lytro camera's working range is from 30 centimeters to 120 centimeters while a Lytro Illum can work from 30 centimeters to 200 centimeters. Another way to increase light ray sampling directions is by incorporating multiple view angles. This strategy requires an accurate camera-robot calibration to perform ray tracing over different angles. In *GlassLoc*, we have talked about using light rays' variance to perform specularity reduction. The same method can be used over opaque surfaces for DLV consistency checks in multi-view DLV construction. When capturing multi-view light-field images, we need to wait for the robot's arm to stay stable to avoid motion blur. Theoretically, if we can capture all light rays in a specific space, we can construct a DLV that includes all possible surfaces in the scene. But in real-world implementation, when there are multiple translucent surfaces (> 2) along a light ray, DLV has difficulty capturing the surfaces located behind the second surface because of the relatively low resolution of our current light-field camera as well as the complicated light ray bouncing behavior in multiple layers of the translucent surface.

Method-wise, *PMCL* is good at generating multi-depth representation for static scenes and its pose estimation results are repeatable under single object scenes. However, constructing DLV from a single view light-field observation cannot properly handle the estimation noise introduced by specularity. Furthermore, the generative inference stage requires a ground-truth object label, CAD model, and approximate 2D bounding box, all of which restrict the method's application in multiple object scenarios.

By further introducing the multi-view DLV construction and specularity suppression, *GlassLoc* enables the robot to perform pick and place under transparent clutter without priors on object label, model, and rough location in manipulation. Nevertheless, *GlassLoc* requires a complicated step in preparing training data and has difficulty dealing with scenes mixing transparent and opaque objects. Furthermore, the property of grasp detection determines that *GlassLoc* has constraints in performing more task-oriented manipulation apart from pick-and-place.

LIT leverages strengths from both *PMCL* and *GlassLoc*, which enables the robot to perform accurate transparent object localization in daily environments. *LIT* also substantially speeds up the pipeline by using network output as a prior and then calculating local DLV only on the regions of interest. Together with *LIT*, we also proposed a *ProLIT* dataset including a pure synthetic training set and a real-world testing set. Our evaluation results on the *ProLIT* dataset have shown that *LIT* outperforms four state-of-the-art object pose estimators on the testing set. However, the *ProLIT* testing set are collected primarily in an indoor Lab environment with a relatively stable lighting condition. As we mentioned in Chapter 5, the lighting condition plays an important role in establishing non-Lambertian surfaces' appearances. Even though an indoor environment includes most of the common light sources (e.g., ambient light, direction light and point light), the different background materials and environment space will also affect the light ray transmission. For some extreme cases (e.g., dark room with barely any lights, outdoor environments with strong sun light), non-Lambertian surfaces will appear very different compared with ordinary lighting conditions. Even though we perform domain randomization in our training set to try and cover as many lighting conditions as possible, some corner cases still remain to be investigated. As such, one potential future work is to evaluate *LIT*'s performance under different lighting conditions.

6.3 Future Work

This dissertation provides several insights in addressing the challenges brought about by transparency and translucency in robotic manipulation. In particular, when tackling the problem of scenes with multiple layers of depth, DLV is proposed to represent depth along a light ray as a likelihood function. This representation divides space into a collection of 3D voxel grids and hypothesizes that each grid can emit lights with a specific combination of RGB color. By comparing the hypothesized light-field with the observed light-field, DLV converges to its most likely state in which each grid has a belief as to how likely it is to belong to an object surface. Nevertheless, as mentioned in the preceding section, this construction process is computationally expensive as it requires sampling light directions and intensities over each of the 3D voxel grids.

A recent work in view synthesis using Neural Radiance Fields (NeRF) [126] shares a similar idea but uses neural networks to dramatically speed up the scene representation construction step. NeRF is a multi-layer perceptron (MLP) network that takes 5D coordinates – 3D location (x, y, z) and 2D view pose (light direction, i.e. (θ, ϕ)) – as input. The output is the RGB channel values with one volume density channel σ. Thus, we can interpret NeRF as a function approximation of the light-field, and its training process concurrently learn a DLV similar descriptor. However, NeRF needs a great deal of training data and requires an extremely long training time, thus making it impossible to perform online light-field model construction. One possible future direction to overcome this limitation is decoupling lighting from its original model. The original NeRF learns the light-field model with baked lighting conditions, which increases the learning complexity and cannot be generalized to a new environment. Apart from treating lighting as a separate network, another potential way to decrease NeRF's training cost is to input light-field images rather than multi-view conventional RGB images. The difference between these two types of images is the way they sample the light direction: a light-field image samples light rays uniformly while a multi-view RGB image performs random sampling. We believe uniform sampling will help the NeRF learn light ray distribution with less data and faster convergence behavior.

Apart from leveraging the insight from NeRF, this dissertation's explorations in light-field perception for manipulation provide many other possible directions for future investigation.

6.3.1 Grasp Pose Detection under Cluttered Scenes Mixing with Transparent and Opaque Objects

In Chapter 4, we proposed *GlassLoc* for detecting transparent objects' grasp poses under minor cluttered environments. A more generalized scenario would be to mix transparent objects with opaque objects; for example, a bin-picking application with both transparent soda bottles and opaque candy boxes. Different from pure transparent objects in *GlassLoc*, mixed scenes bring about greater challenges, both in extracting graspable features and filtering out grasp poses that are in collision. While the graspable volume defined in *GlassLoc* could be extended to a more generalized gripper, we have left the exploration of different grasp volume designs for different types of scenes and grippers as future work.

6.3.2 Dense Surface Reconstruction for Reflective and Refractive Objects

In the *LIT* framework, we have shown that light-field filters with a two-branch encoder-decoder network can segment target transparent objects from a target scene and estimate its object centers. With this insight, a possible extension to the *LIT* network is to add extra decoder structures to further estimate a transparent object's surface normals even depths. If we can collect inference results across multiple observations of a static scene, a possible next step will be to fuse those estimations across frames and reconstruct the reflective and refractive objects.

6.4 Summary

This dissertation presents plenoptic-sensing-based perception for robotic manipulation approaches that enable robots to perform manipulation actions over transparent and translucent objects in real world scenarios. More specifically, we present three plenoptic-based pipelines that can estimate object pose and grasp pose under transparency and translucency using discriminative-generative methods. We have shown that light direction clues in the light-field epipolar image space is able to capture the reflection and refraction features and even infer the surface depth of transparent and translucent objects. Our exploration over robotic plenoptic sensing lead to a collection of promising future directions that make it possible for our robot to be a powerful and reliable assistant in performing all sorts of daily house-work in the near future.

BIBLIOGRAPHY

[1] Alvaro Collet, Dmitry Berenson, Siddhartha S Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE International Conference on Robotics and Automation*, pages 48–55. IEEE, 2009.

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

[3] Jacob Varley, Jonathan Weisz, Jared Weiss, and Peter Allen. Generating multi-fingered robotic grasps via deep learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4415–4420. IEEE, 2015.

[4] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

[5] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *Proceedings of the British Machine Vision Conference*, pages 32.1–32.11, 2013.

[6] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

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

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[13] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[14] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[15] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[16] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018.

[17] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[18] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[19] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[20] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

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

[22] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[23] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

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

[25] Andreas ten Pas and Robert Platt. Using geometry to detect grasp poses in 3d point clouds. In *International Symposium on Robotics Research*, 2015.

[26] Marcus Gualtieri, Andreas Ten Pas, Kate Saenko, and Robert Platt. High precision grasp pose detection in dense clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605. IEEE, 2016.

[27] Qingkai Lu, Kautilya Chenna, Balakumar Sundaralingam, and Tucker Hermans. Planning multi-fingered grasps as probabilistic inference in a learned deep network. In *Robotics Research*, pages 455–472. Springer, 2020.

[28] Zhiqiang Sui, Zheming Zhou, Zhen Zeng, and Odest Chadwicke Jenkins. Sum: Sequential scene understanding and manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3281–3288. IEEE, 2017.

[29] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects. *Science Robotics*, 4(30), 2019.

[30] Zhen Zeng, Zheming Zhou, Zhiqiang Sui, and Odest Chadwicke Jenkins. Semantic robot programming for goal-directed manipulation in cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7462–7469. IEEE, 2018.

[31] Zhen Zeng, Adrian Röfer, and Odest Chadwicke Jenkins. Semantic linking maps for active visual object search. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1984–1990. IEEE, 2020.

[32] Johann Borenstein and Yoram Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[33] James L Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. 1989.

[34] Ilya Lysenkov and Vincent Rabaud. Pose estimation of rigid transparent objects in transparent clutter. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 162–169. IEEE, 2013.

[35] Ilya Lysenkov. Recognition and pose estimation of rigid transparent objects with a kinect sensor. *Robotics*, 273, 2013.

[36] Ole Johannsen, Antonin Sulc, Nico Marniok, and Bastian Goldluecke. Layered scene reconstruction from multiple light field camera views. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 3–18, 2016.

[37] Sven Wanner and Bastian Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition*, pages 1–10. Springer, 2013.

[38] John Oberlin and Stefanie Tellex. Time-lapse light field photography for perceiving transparent and reflective objects. 2017.

[39] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer Berlin Heidelberg, 2014.

[40] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *Int. J. Rob. Res.*, 30(10):1284–1306, September 2011.

[41] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019, 2012.

[42] Zhiqiang Sui, Lingzhu Xiang, Odest C Jenkins, and Karthik Desingh. Goal-directed robot manipulation through axiomatic scene estimation. *The International Journal of Robotics Research*, 36(1):86–104, 2017.

[43] Moritz Tenorth and Michael Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590, 2013.

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

[45] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154. IEEE, 2004.

[46] Corey Goldfeder, Matei Ciocarlie, Hao Dang, and Peter K Allen. The columbia grasp database. In *IEEE International Conference on Robotics and Automation*, pages 1710–1716. IEEE, 2009.

[47] Jacopo Serafin and Giorgio Grisetti. Nicp: Dense normal based point cloud registration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 742–749. IEEE, 2015.

[48] M Lamine Tazir, Tawsif Gokhool, Paul Checchin, Laurent Malaterre, and Laurent Trassoudaine. Cicp: Cluster iterative closest point for sparse–dense point cloud registration. *Robotics and Autonomous Systems*, 108:66–86, 2018.

[49] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[50] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010.

[51] Mark R Stevens and J Ross Beveridge. Localized scene interpretation from 3d models, range, and optical data. *Computer Vision and Image Understanding*, 80(2):111–129, 2000.

[52] Venkatraman Narayanan and Maxim Likhachev. Perch: perception via search for multi-object recognition and localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5052–5059. IEEE, 2016.

[53] Venkatraman Narayanan and Maxim Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Robotics: Science and Systems*, 2016.

[54] Zhiqiang Sui, Odest Chadwicke Jenkins, and Karthik Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4429–4436. IEEE, 2015.

[55] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687. IEEE, 2015.

[56] Max Schwarz, Hannes Schulz, and Sven Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *IEEE international conference on robotics and automation (ICRA)*, pages 1329–1335. IEEE, 2015.

[57] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems (RSS)*, 2018.

[58] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *The Conference on Robot Learning (CoRL)*, 2018.

[59] Josip Josifovski, Matthias Kerzel, Christoph Pregizer, Lukas Posniak, and Stefan Wermter. Object detection and pose estimation based on convolutional neural networks trained with synthetic data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6269–6276. IEEE, 2018.

[60] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019.

[61] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

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

[63] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019.

[64] Bernard Faverjon and Jean Ponce. On computing two-finger force-closure grasps of curved 2d objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 424–429. IEEE, 1991.

[65] B Dizioğlu and K Lakshiminarayana. Mechanics of form closure. *Acta mechanica*, 52(1-2):107–118, 1984.

[66] Zhixing Xue, Alexander Kasper, J Marius Zoellner, and Ruediger Dillmann. An automatic grasp planning system for service robots. In *International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009.

[67] Justus H Piater. Learning visual features to predict hand orientations. 2002.

[68] Antonio Morales, Pedro J Sanz, and Angel P Del Pobil. Vision-based computation of three-finger grasps on unknown planar objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1711–1716. IEEE, 2002.

[69] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, and Andrew Y Ng. Robotic grasping of novel objects. In *Advances in neural information processing systems*, pages 1209–1216, 2007.

[70] Quoc V Le, David Kamm, Arda F Kara, and Andrew Y Ng. Learning to grasp objects with multiple contact points. In *IEEE International Conference on Robotics and Automation*, pages 5062–5069. IEEE, 2010.

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

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

[73] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*, volume 2.

[74] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[75] Ren Ng. *Digital light field photography*. Stanford University, California.

[76] Wilson S Geisler. Visual perception and the statistical properties of natural scenes. *Annu. Rev. Psychol.*, 59:167–192, 2008.

[77] Donald Gilbert Dansereau. Plenoptic signal processing for robust vision in field robotics. 2013.

[78] Donald G Dansereau, Ian Mahon, Oscar Pizarro, and Stefan B Williams. Plenoptic flow: Closed-form visual odometry for light field cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4455–4462. IEEE, 2011.

[79] Fengchun Dong, Sio-Hoi Ieng, Xavier Savatier, Ralph Etienne-Cummings, and Ryad Benosman. Plenoptic cameras in real-time robotics. *The International Journal of Robotics Research*, 32(2):206–217, 2013.

[80] Ole Johannsen, Antonin Sulc, and Bastian Goldluecke. On linear structure from motion for light field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 720–728, 2015.

[81] Katherine A Skinner and Matthew Johnson-Roberson. Underwater image dehazing with a light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 62–69, 2017.

[82] Abhishek Bajpayee, Alexandra H Techet, and Hanumant Singh. Real-time light field processing for autonomous robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4218–4225. IEEE, 2018.

[83] Paul Foster, Zhenghong Sun, Jong Jin Park, and Benjamin Kuipers. Visagge: Visible angle grid for glass environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2213–2220. IEEE, 2013.

[84] Magnus Borga and Hans Knutsson. Estimating multiple depths in semi-transparent stereo images. 1999.

[85] Venkatraman Narayanan and Maxim Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[86] Kenton McHenry and Jean Ponce. A geodesic active contour framework for finding glass. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1038–1044. IEEE, 2006.

[87] Kenton McHenry, Jean Ponce, and David Forsyth. Finding glass. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 973–979. IEEE, 2005.

[88] Zhong Lei, Kazunori Ohno, Masanobu Tsubota, Eijiro Takeuchi, and Satoshi Tadokoro. Transparent object detection using color image and laser reflectance image for mobile manipulator. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–7. IEEE, 2011.

[89] Cody J Phillips, Matthieu Lecce, and Kostas Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Proceedings of Robotics: Science and Systems*, 2016.

[90] Todor Georgiev, Zhan Yu, Andrew Lumsdaine, and Sergio Goma. Lytro camera technology: theory, algorithms, performance analysis. In *Multimedia Content and Mobile Devices*, volume 8667, page 86671J. International Society for Optics and Photonics, 2013.

[91] 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.

[92] Antonin Sulc, Anna Alperovich, Nico Marniok, and Bastian Goldluecke. Reflection separation in light fields based on sparse coding and specular flow. In *Proceedings of the Conference on Vision, Modeling and Visualization*, pages 137–144. Eurographics Association, 2016.

[93] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3487–3495. IEEE, 2015.

[94] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1547–1555, 2015.

[95] Zhan Yu, Xinqing Guo, Haibing Ling, Andrew Lumsdaine, and Jingyi Yu. Line assisted light field triangulation and stereo matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2792–2799. IEEE, 2013.

[96] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2013.

[97] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 1999.

[98] Yunsu Bok, Hae-Gon Jeon, and In So Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):287–300, 2017.

[99] Patrick Beeson and Barrett Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *IEEE-RAS International Conference on Humanoid Robots*, 2015.

[100] Ioan A Sucan and Sachin Chitta. Moveit! *Online Availabl e: http://moveit. ros. org*, 2013.

[101] Andreas ten Pas and Robert Platt. Localizing handle-like grasp affordances in 3d point clouds. In *Experimental Robotics*, pages 623–638. Springer, 2016.

[102] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[103] Zheming Zhou, Zhiqiang Sui, and Odest Chadwicke Jenkins. Plenoptic monte carlo object localization for robot grasping under layered translucency. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[104] Katherine A Skinner and Matthew Johnson-Roberson. Towards real-time underwater 3d reconstruction with plenoptic cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2014–2021. IEEE, 2016.

[105] Dorian Tsai, Donald G Dansereau, Thierry Peynot, and Peter Corke. Distinguishing refracted features using light field cameras with application to structure from motion. *IEEE Robotics and Automation Letters*, 4(2):177–184, 2018.

[106] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[107] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2015.

[108] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

[109] Pat Marion, Peter R Florence, Lucas Manuelli, and Russ Tedrake. Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[110] Zheming Zhou, Tianyang Pan, Shiyu Wu, Haonan Chang, and Odest Chadwicke Jenkins. Glassloc: Plenoptic grasp pose detection in transparent clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4776–4783. IEEE, 2019.

[111] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017.

[112] 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, 2018. `https://github.com/NVIDIA/Dataset_Synthesizer`.

[113] 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.

[114] Xiaotong Chen, Rui Chen, Zhiqiang Sui, Zhefan Ye, Yanqi Liu, R Iris Bahar, and Odest Chadwicke Jenkins. Grip: Generative robust inference and perception for semantic robot manipulation in adversarial environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3988–3995. IEEE, 2019.

[115] Kiru Park, Timothy Patten, Johann Prankl, and Markus Vincze. Multi-task template matching for object detection, segmentation and pose estimation using depth images. In *International Conference on Robotics and Automation (ICRA)*, pages 7207–7213. IEEE, 2019.

[116] Chaitanya Mitash, Abdeslam Boularias, and Kostas E Bekris. Robust 6d object pose estimation with stochastic congruent sets. In *29th British Machine Vision Conference (BMVC)*, 2018.

[117] Michael W Tao, Jong-Chyi Su, Ting-Chun Wang, Jitendra Malik, and Ravi Ramamoorthi. Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1155–1169, 2015.

[118] Anna Alperovich, Ole Johannsen, Michael Strecke, and Bastian Goldluecke. Light field intrinsics with a deep encoder-decoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9145–9154, 2018.

[119] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei A Efros, and Ravi Ramamoorthi. A 4d light-field dataset and cnn architectures for material recognition. In *European Conference on Computer Vision*, pages 121–138. Springer, 2016.

[120] Stephen J McKenna and Hammadi Nait-Charif. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Image and Vision Computing*, 25(6):852–862, 2007.

[121] Yunsu Bok, Hae-Gon Jeon, and In So Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):287–300, 2016.

[122] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[123] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

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

[125] Epiimaging, llc. http://epiimaging.com/. Accessed: 2020-04-20.

[126] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.