

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

Zheming Zhou Xiaotong Chen Odest Chadwicke Jenkins

**Abstract**—Translucency is prevalent in everyday scenes. As such, perception of transparent objects is essential for robots to perform manipulation. Compared with texture-rich or textureless Lambertian objects, transparency induces significant uncertainty on object appearances. Ambiguity can be due to changes in lighting, viewpoint, and backgrounds, each of which brings challenges to existing object pose estimation algorithms. In this work, we propose *LIT*, a two-stage method for transparent object pose estimation using light-field sensing and photorealistic rendering. *LIT* employs multiple filters specific to light-field imagery in deep networks to capture transparent material properties, with robust depth and pose estimators based on generative sampling. Along with the *LIT* algorithm, we introduce the light-field transparent object dataset *ProLIT* for the tasks of recognition, localization and pose estimation. With respect to this *ProLIT* dataset, we demonstrate that *LIT* can outperform both state-of-the-art end-to-end pose estimation methods and a generative pose estimator on transparent objects. The link of supplementary material can be found at: <https://sites.google.com/umich.edu/prolit>

## I. INTRODUCTION

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 [1], [2], [3] cover a variety of texture-rich objects with distinguishable features between different types of objects. Several other works [4], [5] 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 specularities 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 tools [6]. As a result, transparent objects remain effectively invisible to robots using the sensors.

The authors are with the Department of Electrical Engineering and Computer Science, Robotics Institute, University of Michigan, Ann Arbor, MI, USA, 48109-2121 [zhezhou|cxt|ocj]@umich.edu



Fig. 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.

Recently, several works [7], [8] showed promising results using light-field (or plenoptic) photography in perceiving transparent objects. For example, Zhou *et al.* [9] 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.* [10] rendered photorealistic images by projecting the object texture model on the real background for training object detectors. Tremblay *et al.* [3] proposed DOPE as an end-to-end pose estimator using domain randomization and photorealistic rendering [11]. We similarly address the problem of transparency using photorealistic rendering and light-field perception.

In this paper, we propose *LIT* 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

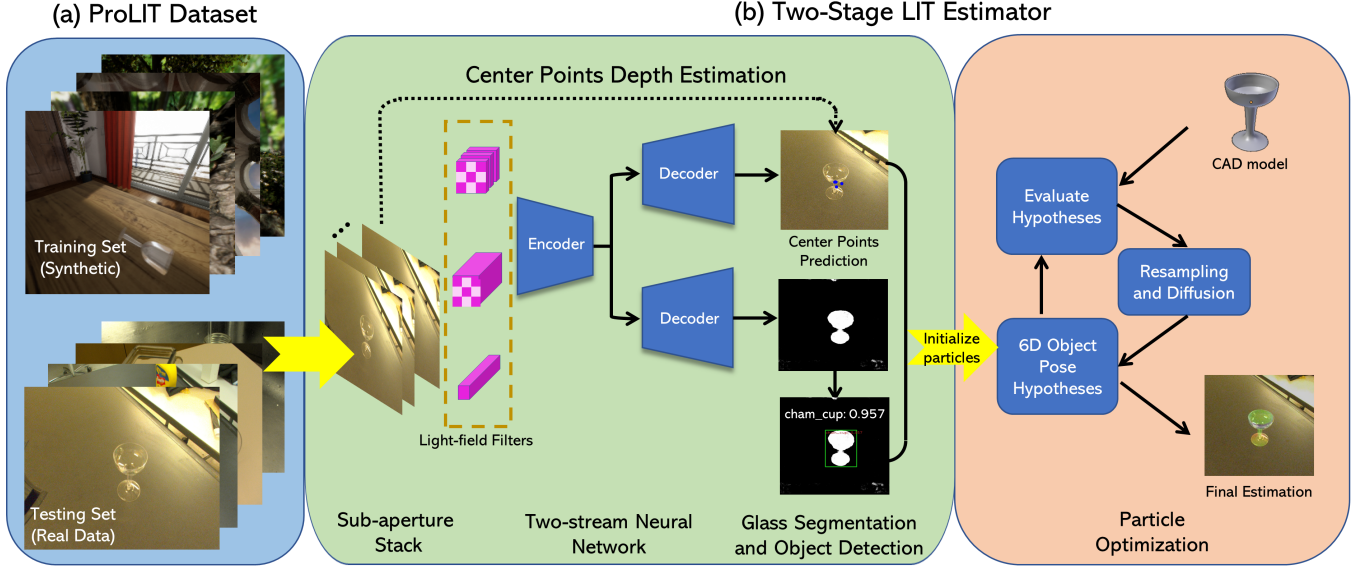


Fig. 2: An overview of the *LIT* framework with the *ProLIT* dataset. (a) *ProLIT* contains 75,000 synthetic light-field images in training set and 300 real images with 442 object instances in testing set. (b) *LIT* estimator is a two-stage pipeline. The first stage takes light-field images as input 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.

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.

## II. RELATED WORK

### A. 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.* [12] 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 [3], [13], pixel-wise voting over keypoints [14], [15], and residual networks to iteratively refine object poses [5], [2]. 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 [1], [16], template matching [17], and point cloud registration [4], [18].

Most deep learning methods for pose estimation are focused on texture-rich objects or those with texture-less but Lambertian surfaces [17], [4]. 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 [19], [20] have used invalid readings from depth camera to extract object contours for pose estimation. However, these methods rely on the Lambertian 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.

### B. Light-field Perception for Transparency

The foundation of light-field image rendering was first introduced by Levoy and Hanrahan [21] 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.* [22] proposed the light-field distortion features from epipolar images for recognizing transparent objects. Recent work by Tsai *et al.* [23] 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.* [24] investigated the line consistency in the light-field images with a dichromatic reflection model that removes the specularity from the images. Alperovich *et al.* [25] proposed fully convolutional networks to separate specularity in light-field images. In robotics, Zhou *et al.* [7], [9] 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.

### III. 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.

#### A. LIT Pipeline

The two-stage *LIT* pipeline is shown in Figure 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. A light-field based object depth estimator gives object center depth distributions. 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 [14], [15]. 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.

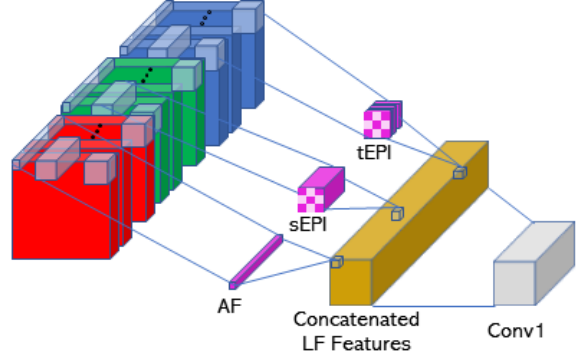


Fig. 3: Illustration of three light-field filters. Angular filter (AF) has dimension  $1 \times 1 \times (H_a \times W_a)$  to capture features in angular pixels. sEPI and tEPI filters have sizes of  $n \times n \times W_a$  and  $n \times n \times H_a$  respectively, here  $n$  refers to kernel size. tEPI also has a dilation  $W_a$ . All features will be concatenated together after passing filters.

#### B. Network Architecture

As shown in Figure 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 [26], 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  $j$ th filter):

$$g\left(\sum_{s,t} w_i^j(s,t) L_i(x,y,(s,t))\right) \quad (1)$$

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 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:

$$g\left(\sum_{u,v,s} \tilde{w}_i^j(u,v,s) L_i(x+u,y+v,(s,t))\right) \quad (2)$$

$$g\left(\sum_{u,v,t} \hat{w}_i^j(u,v,t) L_i(x+u,y+v,(s,t))\right)$$

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 [27]. The loss of center point regression is mainly following design in [14], 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:

$$\mathcal{L}_{pos} = \sum_{p \in \mathcal{M}} \|g_p - (c_p + h_p)\|_1 \quad (3)$$

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:

$$\mathcal{L}_{conf} = \sum_{p \in \mathcal{M}} \|b_p - \exp(-\tau \|g_p - (c_p + h_p)\|_2)\|_1 \quad (4)$$

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

$$\mathcal{L} = \alpha \mathcal{L}_{seg} + \beta \mathcal{L}_{pos} + \gamma \mathcal{L}_{conf} \quad (5)$$

where  $\alpha, \beta, \gamma$  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 [7] 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:

$$D(x_c, y_c, d) = \frac{1}{N} \sum_{a \in A \setminus I_c} T_{a,d}(x_c, y_c) \quad (6)$$

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 [7], [9].

### C. Particle Optimization

The second stage of pipeline estimates the 6D pose of transparent objects in a sampling-based iterative likelihood reweighting process [28]. 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.

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:

$$\begin{aligned} u &= c_x + f_x \frac{x}{z}, & v &= c_y + f_y \frac{y}{z}, & d &= z \\ p(X=x, Y=y, Z=z) &= b(u, v) D(u, v, d) \end{aligned} \quad (7)$$

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 (6). 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.

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:

$$weight = \eta \frac{|S_{pcd} \cap S_{seg}|}{|S_{pcd} \cup S_{seg}|} + (1 - \eta) \frac{|\partial S_{pcd} \cap \partial S_{seg}|}{|\partial S_{pcd} \cup \partial S_{seg}|} \quad (8)$$

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



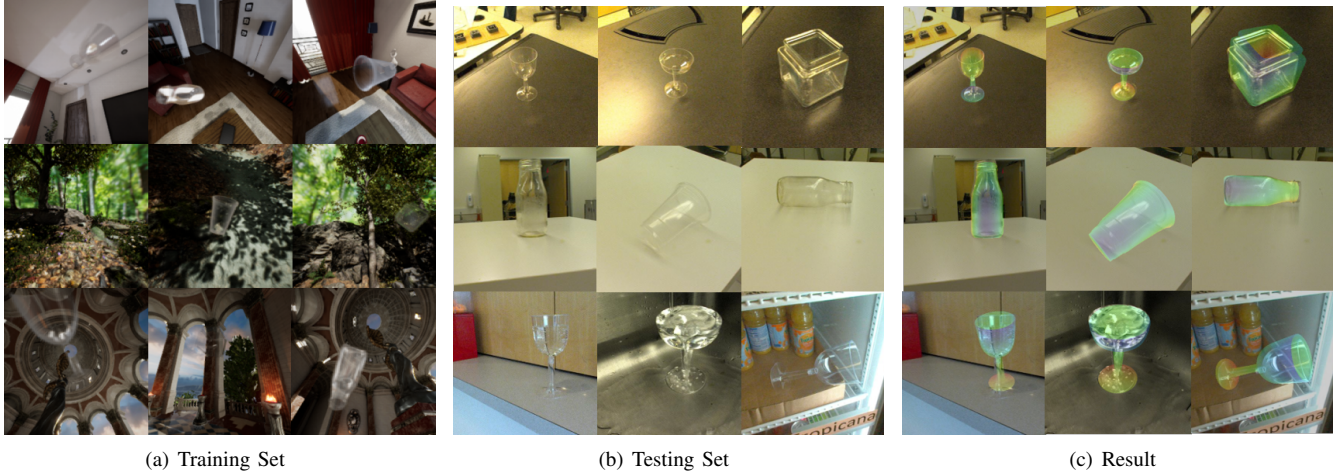


Fig. 4: (Left) example synthetic light-field images rendered in three different environments. (Middle) example test images in different backgrounds and different pose configurations. (Right) results visualization by overlaying estimated poses to the original test images.

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.

#### IV. 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 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 [29]. The spatial resolution of the calibrated image is  $383 \times 552$ , and the angular resolution is  $5 \times 5$  (extracted from  $9 \times 9$  sub-aperture 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 [11] synthetic data generation plugin in Unreal Engine 4 (UE4). The created virtual light-field capturer has an angular resolution  $5 \times 5$  and spatial resolution  $224 \times 224$ . The baseline between the adjacent virtual camera is set to 0.1cm. We generate data in three UE4 world environments: room, temple, and forest. In each environment, we highly randomized the lighting conditions including color, direction, and intensity.

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.

#### V. EXPERIMENTS

We choose 64 light-field filters as the first feature extraction layer. The *LIT* network uses VGG16 [30] as backbone architecture and initialized with pre-trained model on ImageNet [31]. 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 [32] with VGG16 backbone. The input to the network is the binary masks of transparent object segmentation and the output is bounding boxes with object labels.

##### A. 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 I 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

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
<b>LIT</b>	<b>0.954</b>	<b>0.520</b>	<b>0.455</b>	<b>0.854</b>	<b>0.390</b>

TABLE I: 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 [33]. ‘AF only’ here refers to the baseline method with only angular filters.

higher accuracy, showing that both angular features and EPI features are important in contributing to segmenting transparent objects.

### B. Evaluation of Pose Estimation

We compare the 6D pose estimation results of *LIT* against a state-of-the-art general-purpose end-to-end object pose estimator, DOPE [3], a state-of-the-art textureless object pose estimator, Augmented Autoencoder (AAE) [4], and a generative light-field based transparent object pose estimation method, PMCL [7].

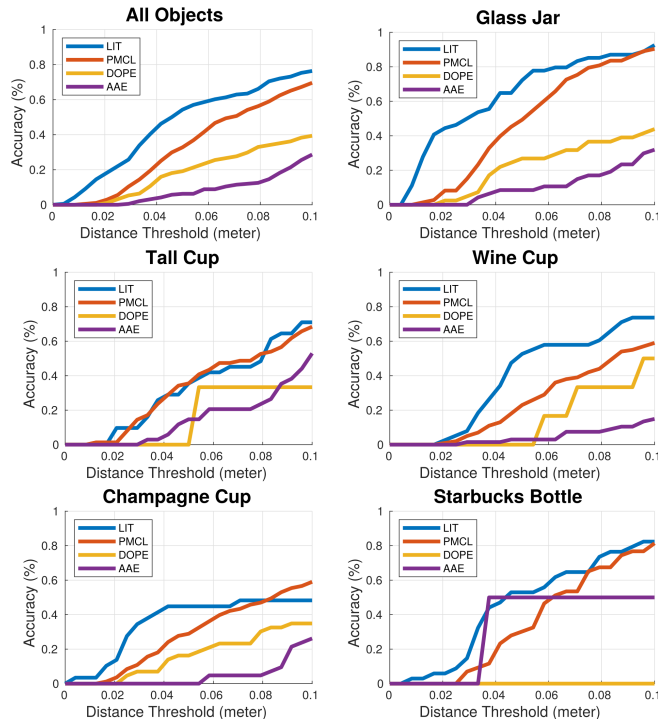


Fig. 5: Comparison of 6D pose estimation results with respect to ADD-S and Accuracy Under Curve metric.

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 III 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 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  $40 \times 40 \times 40 \text{ cm}^3$  around the ground truth object locations. The convergence threshold of particle weights is set to 0.7. We use ADD-S metric [12] to evaluate the pose results of symmetric objects. We then show the accuracy curves in Figure 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 II.

From the result plots, we find that *LIT* performs much better than DOPE and AAE, and better than PMCL. For DOPE, we believe 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 specularly and distortion from transparent surfaces. Furthermore, DLV construction is computationally expensive, which takes an average 300 seconds for one object. In conclusion, *LIT* pipeline provides better accuracy than all three baseline methods on the testing dataset with a relatively small computationally cost.

AUC	wc	tc	gj	cc	sb	all	time(s)/obj
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	<b>0.32</b>	0.46	0.28	0.34	0.32	300
<b>LIT</b>	<b>0.38</b>	<b>0.32</b>	<b>0.62</b>	<b>0.35</b>	<b>0.44</b>	<b>0.45</b>	< 10

TABLE II: Comparison of *LIT*, DOPE, AAE, and PMCL 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.

### C. 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 6. 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 wif. The Lytro camera’s extrinsic matrix is calibrated with



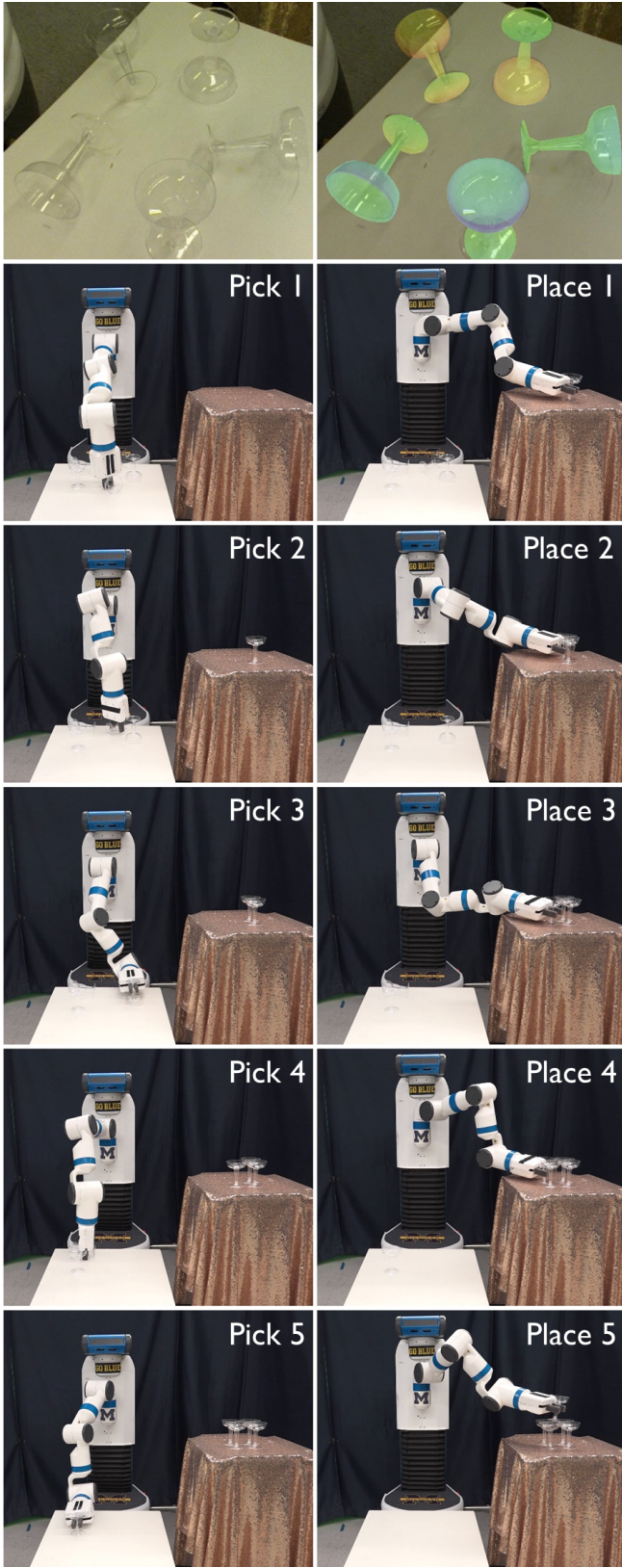


Fig. 6: The robot is building a champagne tower by successfully picking and placing champagne cups on the table. The first row shows light-field observation (left) and pose estimation result from *LIT* (right). The following five rows show pick and place actions to finish the champagne tower.

robot world frame. *LIT* then performs pose estimation over the scene, and the results are then adopted to transform the pre-defined grasp poses from the object's local coordinate frame to the robot world frame. 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.

## VI. CONCLUSIONS

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. 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.

## REFERENCES

- [1] 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)*, 2017, pages 3281–3288. IEEE, 2017.
- [2] 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.
- [3] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [4] 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.
- [5] 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.
- [6] 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)*, 2018, pages 1–8. IEEE, 2018.
- [7] 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)*, 2018, pages 1–8. IEEE, 2018.
- [8] John Oberlin and Stefanie Tellex. Time-lapse light field photography for perceiving transparent and reflective objects. 2017.
- [9] Zheming Zhou, Tianyang Pan, Shiyu Wu, Haonan Chang, and Odest Chadwicke Jenkins. Glassloc: Plenoptic grasp pose detection in transparent clutter. *arXiv preprint arXiv:1909.04269*, 2019.
- [10] 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.

- [11] Thang To, Jonathan Tremblay, Duncan McKay, Yukie Yamaguchi, Kirby Leung, Adrian Balanón, Jia Cheng, William Hodge, and Stan Birchfield. NDDS: NVIDIA deep learning dataset synthesizer, 2018.
- [12] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.
- [13] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6269–6276. IEEE, 2018.
- [14] 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.
- [15] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.
- [16] Xiaotong Chen, Rui Chen, Zhiqiang Sui, Zhifan Ye, Yanqi Liu, R Bahar, and Odest Chadwicke Jenkins. Grip: Generative robust inference and perception for semantic robot manipulation in adversarial environments. *arXiv preprint arXiv:1903.08352*, 2019.
- [17] Kiru Park, Timothy Patten, Johann Prankl, and Markus Vincze. Multi-task template matching for object detection, segmentation and pose estimation using depth images. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7207–7213. IEEE, 2019.
- [18] Chaitanya Mitash, Abdeslam Boularias, and Kostas Bekris. Robust 6D object pose estimation with stochastic congruent sets. *arXiv preprint arXiv:1805.06324*, 2018.
- [19] Ilya Lysenkov. Recognition and pose estimation of rigid transparent objects with a kinect sensor. *Robotics*, 273, 2013.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] Anna Alperovich, Ole Johannsen, Michael Strecke, and Bastian Gold-luecke. 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [32] 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.
- [33] 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.