

# Ambiguity-Aware Multi-Object Pose Optimization for Visually-Assisted Robot Manipulation

Myung-Hwan Jeon<sup>1</sup>, Jeongyun Kim<sup>2</sup>, Jee-Hwan Ryu<sup>3</sup>, and Ayoung Kim<sup>2\*</sup>

**Abstract**—6D object pose estimation aims to infer the relative pose between the object and the camera using a single image or multiple images. Most works have focused on predicting the object pose without associated uncertainty under occlusion and structural ambiguity (symmetricity). However, these works demand prior information about shape attributes, and this condition is hardly satisfied in reality; even asymmetric objects may be symmetric under the viewpoint change. In addition, acquiring and fusing diverse sensor data is challenging when extending them to robotics applications. Tackling these limitations, we present an ambiguity-aware 6D object pose estimation network, PrimA6D++, as a generic uncertainty prediction method. The major challenges in pose estimation, such as occlusion and symmetry, can be handled in a generic manner based on the measured ambiguity of the prediction. Specifically, we devise a network to reconstruct the three rotation axis primitive images of a target object and predict the underlying uncertainty along each primitive axis. Leveraging the estimated uncertainty, we then optimize multi-object poses using visual measurements and camera poses by treating it as an object SLAM problem. The proposed method shows a significant performance improvement in T-LESS and YCB-Video datasets. We further demonstrate real-time scene recognition capability for visually-assisted robot manipulation. Our code and supplementary materials are available at <https://github.com/rpmsnu/PrimA6D>.

## I. INTRODUCTION

6D object pose estimation is involved in recovering the orientation and translation of the object relative to the camera, playing a crucial role in various localization problems. In this line of study, deep learning-based methods [1, 2, 3, 4] presented remarkable performance on benchmark datasets, yet mainly focused on improving pose accuracy based on a single image. Many robotics applications often leverage multiple static cameras placed around the workspace or a single camera attached to an end-effector. In these configurations, predicting uncertainty becomes crucial to measure the prediction reliability in order to balance between inferences among multiple views or robot kinematics.

Unfortunately, in previous studies, researchers have focused on improving the prediction accuracy without associated uncertainty under occlusion, structural ambiguity (symmetricity), and object size. More critically, these studies require prior information on symmetricity for the prediction phase, which is arduous to obtain in reality; therefore, uncertainty-estimation-based methods could be more viable

<sup>†</sup>This work was supported by a grant (22TSRD-C151228-04) MOLIT.

<sup>1</sup>M. Jeon is with Inst. of Advanced Machines and Design, SNU, Seoul, S. Korea myunghwan.jeon@snu.ac.kr

<sup>2</sup>J. Kim and A. Kim with the Dept. of Mechanical Engineering, SNU, Seoul, S. Korea [ayoungk, jeongyunkim@snu.ac.kr

<sup>3</sup>J. Ryu is with Dept. of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea jhryu@kaist.ac.kr

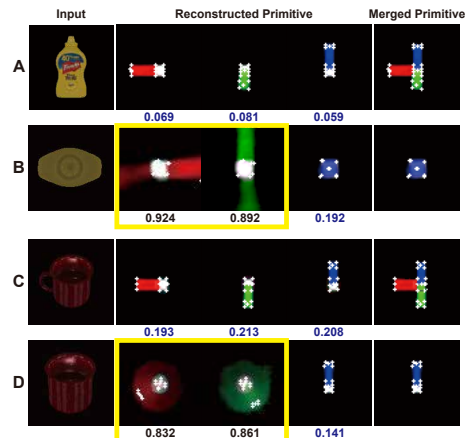


Fig. 1: Two sample cases showing varying ambiguity. An asymmetric object (A and C) may seem symmetric (B and D) depending on the viewpoint change. Unlike clearly reconstructed three rotation axes, viewpoint-induced ambiguity is captured in the reconstructed primitive axes (yellow box in B and D). We learn this induced uncertainty in an unsupervised manner to handle object ambiguity without prior information on object shape in a generic manner.

in field of robotics when such priors information is unavailable. Nevertheless, estimating the uncertainty for the estimated pose has been overlooked in the literature.

Only recently, capturing uncertainty [5, 6] and including ambiguity [7, 8] have been reported in the literature. PoseRBPF [5] employed a Rao-Blackwellized particle filter [9] representing the distribution of the predicted pose. Okorn et al. [6] examined uncertainties for asymmetric and symmetric objects using Bingham distribution and histogram distribution. In [7, 8], the authors indicated potential ambiguity in pose estimation and proposed leveraging multiple hypotheses but not directly estimating the uncertainty.

Unlike in previous studies, we introduce an object ambiguity-aware object pose estimation network, PrimA6D++, as a generic uncertainty estimation. Our previous work, PrimA6D [10], introduced the rotation primitive reconstruction which predicts color-coded orientation axes for robust object pose estimation, yet this method cannot handle the object ambiguity and uncertainty of the predicted pose. Extending from [10], the proposed network predicts three rotation axis primitive images, each corresponding to the orientation axis of the object (Fig. 1). In addition, the uncertainty for each rotation axis primitive image is estimated via unsupervised learning. Based on these uncertainties, we discern object ambiguity caused by shape symmetricity and occlusion by rejecting unreliable rotation axis primitive images. We further utilize the norm value of these uncertainties in the multi-object pose optimization.

In this work, a robot manipulator is equipped with an

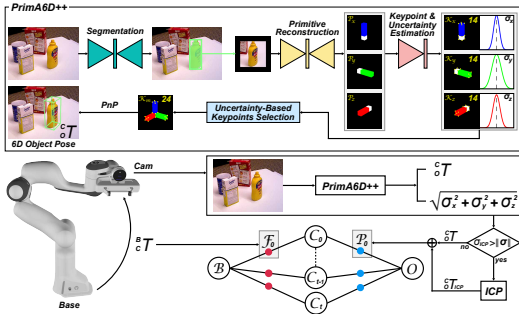


Fig. 2: The proposed network, Prima6D++, is composed of four modules including: a segmentation network, a primitive reconstruction network, a keypoint and uncertainty extraction network, and an uncertainty-based keypoints selection module. The cropped RGB image using the object segmentation mask transforms into the three rotation axis primitive images. In each rotation axis primitive image, 14 keypoints and their associated uncertainties are estimated. To reject unreliable keypoints extracted from rotation axis primitive images, we execute the uncertainty-based keypoints selection. Lastly, the 6D object poses are refined using the factor graph optimization.

RGB-D sensor attached to the end-effector (Fig. 2). Given sequential observed sensor data from our configuration, we present a system that optimizes the multi-object poses in the workspace. Specifically, leveraging the estimated uncertainty, we formulate the problem as an object SLAM by introducing the camera pose factor and object pose factor to refine the multi-object poses with camera poses to enhance 3D scene recognition in visually-assisted robot manipulation tasks. Differing from previous methods, our method presents the following contributions.

- We present a novel ambiguity-aware 6D object pose estimation network, Prima6D++, which predicts three rotation axis primitive images and the associated uncertainties. Most existing methods require symmetricity information of an object, yet in reality, this information may not be available in advance; additionally, some asymmetric objects may be symmetric depending on the viewpoint change. The proposed uncertainty estimation is generic and does not require shape information.
- Leveraging the estimated uncertainty, we formulate the problem as an object SLAM by introducing the camera pose factor and object pose factor to refine the multi-object poses with camera poses. Specifically, we verify that the proposed optimization process removes the undesired jittering of the object pose estimation.
- We verify the meaningful performance improvement of 6D object pose estimation compared to baseline methods. The proposed method improved estimation over YCB-Video to 94.4% in terms of ADD(-S), T-LESS to 87.8% in terms of  $e_{VSD}$ , and T-LESS to 0.916 in terms of  $AR_{mssd}$ .

## II. RELATED WORKS

### A. Single-View-Based Pose Refinement

The 6D object pose estimation based on deep learning has achieved substantial progress, however, the accuracy is still bounded. Researchers have tried to overcome this problem with a refinement step. DPOD [11] predicted the relative transformation between the input image observed via the RGB sensor and the rendered image using the

initial pose proposal, which then were refined iteratively. RePOSE [12] introduced the intermediate representation with a neural network to calculate the relative transformation. In addition to new representation, RNNPose [13] exploited the recurrent neural network in the iteration process. Lipson et al. [14] iteratively refined pose and correspondence in a tightly coupled manner to be more robust to outliers. DenseFusion [4] extracted the global features in the point cloud, and then the object pose was refined as iteratively reducing the residual between these global features. Although these methods generally improved the pose accuracy, they still suffer from occlusion between objects as the single-image approach.

### B. Multiple-View-Based Pose Refinement

Researchers have utilized sequential multi-view information to achieve more accurate 6D poses and deal with the inherent problem of single-view-based approaches. se(3)-TrackNet [15] estimated the object poses by performing pose residual estimation between the current observation and the synthetic rendered image for 6D object tracking. Deng et al. [16] enhanced their previous method, PoseRBPF, in a self-supervised manner along with their additional data sequences. These methods did not guarantee global consistency as a tracking approach that utilizes the information from the previous frame. MoreFusion [17] generated the occupancy map using sequential RGBD views, and then performed joint optimization to estimate collision-free multi-object poses. CosyPose [3] removed the non-consistent object pose proposal through inlier checking. Then, this 6D object pose was optimized through consistent matching between the initial pose proposals estimated in the multi-view input images. Shugurov et al. [18] exploited the differentiable rendering to make geometrical constraints using the relative camera poses. BundleTrack [19] estimated the object pose proposal using keypoint detection, then performed pose-graph optimization for selected keyframes. Merrill et al. [20] proposed an object-level simultaneous localization and mapping (SLAM) based on keypoints extracted from the network. These methods required prior information about shape attributes and may encounter the best-view selection problem. Unlike these methods, our work maintains global consistency through graph optimization with all available data, and does not need prior information about object shape.

### C. Object Pose Uncertainty Estimation

To integrate deep-learning methods with other modules in robotic tasks, uncertainty awareness for a deep-learning module becomes essential. For the object pose uncertainty, PoseRBPF sampled the particles through bounding-box variation for translation distribution. The orientation distribution was obtained by comparing the discretized rotation codebook and the sampled particles. The estimated distribution was propagated with a particle filter. Shi et al. [21] proposed the ensemble-based uncertainty quantification. In [6], the authors regressed the object orientation and its uncertainty based on

the Bingham distribution. Deng et al. [22] regressed Bingham mixture models through multi-hypothesis prediction. Manhardt et al. [7] estimated specific pose distribution from multiple object pose hypotheses. Merrill et al. [20] predicted the Gaussian covariance of object keypoints to utilize to keypoints-based SLAM. Instead, we regress the uncertainties of reconstructed primitive images, and then combine these uncertainties for the initial pose proposal and multi-object pose optimization.

### III. METHOD

#### A. Notation

We define the notation to describe our proposed method. The transformation from coordinate system  $A$  to coordinate system  $B$  is denoted as  ${}^A_B T$ , which comprises of rotation  $\mathbf{R}_{AB} \in SO(3)$  and translation  $\mathbf{t}_{AB} \in \mathbb{R}^3$ .  $W$  represents the world coordinate. The subscript  $(\cdot)_i$  indicates the specific index of the corresponding variable. The subscript  $(\cdot)_{0:n}$  indicates the group comprising the initial value to the  $n^{\text{th}}$  value.

#### B. Overview

Given a manipulator equipped with an RGB-D camera that moves over a set of objects in the workspace, the robotic system collects RGB-D images captured from various viewpoints and the associated joint encoder angles of the robot manipulator. Taking all of this sequential data as input, the proposed method aims to optimize the multi-object pose by fusing the initial object poses from the RGB camera and other heterogeneous sensor data. As shown in Fig. 2, the proposed method is two-staged. In the first stage, we use RGB images to estimate simultaneously generic primitive uncertainty and 6D object pose  ${}^C_O T$  expressed in terms of its relative transform between camera coordinate frame  $C$  and object coordinate frame  $O$ . In the second stage, we refine object poses in an object SLAM manner using camera poses via factor graph optimization.

#### C. 6D Object Pose Initialization

1) *Object Segmentation*: First, we perform object segmentation. We construct a segmentation network adopting the encoder-decoder architecture based on ResNet34. [23]. Overall, the network generates the  $N + 1$  dimension segmentation map, where  $N$  indicates the number of objects.

2) *Rotational Primitive Axis Reconstruction*: Two major challenges in object pose estimation are occlusion and symmetry. To overcome these challenges, we leverage ideas from our previous work [10] on primitive reconstruction for objects followed by keypoint extraction from these primitives. By doing so, we can reliably estimate the pose from even small and texture-less objects.

Similarly, as in [10], we crop the input RGB image and concentrate on the target object using the segmentation map. We then transform the cropped RGB image into the primitive image using the Auto Encoder (AE)-based primitive reconstruction network. Differing from [10], our network separately predicts the three rotation axis primitive images

per target object by appending three additional layers to the decoder. Each primitive image represents a respective rotation axis of the 6D object pose and each will be used to predict the uncertainty in the next stage.

To train the primitive reconstruction network, we use a primitive reconstruction loss  $\mathbf{L}_R$  [10] for each a rotation axis primitive image,  $\mathcal{P}_x$ ,  $\mathcal{P}_y$ , and  $\mathcal{P}_z$ .

$$\mathcal{L}_{\mathcal{P}} = \mathbf{L}_R(\mathcal{P}_x) + \mathbf{L}_R(\mathcal{P}_y) + \mathbf{L}_R(\mathcal{P}_z) \quad (1)$$

To improve the quality of the primitive reconstruction, we adopt the adversarial loss where the generator and the discriminator are each focused on minimizing  $\mathcal{L}_G = \mathbb{E}_{\hat{X}}[\log \mathcal{D}(\hat{X})]$  and maximizing  $\mathcal{L}_D = \mathbb{E}_X[\log(1 - \mathcal{D}(\mathcal{G}(X)))]$  over target primitive image  $X$  and estimated primitive image  $\hat{X}$ .

3) *Keypoints and Uncertainty Estimation*: Given the reconstructed primitive images, we extract keypoints and the associated uncertainty. Even an asymmetric target object might reveal ambiguity depending on the viewpoint yielding obscurity in the reconstructed primitive image as shown from two samples in Fig. 1. We exploit this degeneracy when simultaneously regressing the keypoints of the rotation axis primitive images and their associated uncertainties. Our intention is to estimate the uncertainty by measuring the quality of the reconstructed rotation axis primitive image.

To train the keypoints and uncertainty simultaneously, we assume the uncertainty as a zero-mean normal distribution  $\mathcal{N}(0, \Sigma)$  and estimate  $\Sigma$  by minimizing the negative log-likelihood of distribution over the keypoint errors  $e_k = |x_k - \hat{x}_k|$  over  $n$  keypoints in the training set.

$$\mathbf{L}(e) = \underset{\Sigma_{i:n}}{\operatorname{argmin}} \sum_{i=1}^n -\log p(e_i | \Sigma_i) \quad (2)$$

$$= \underset{\Sigma_{i:n}}{\operatorname{argmin}} \sum_{i=1}^n \log |\Sigma_i| + e_i^\top \Sigma_i^{-1} e_i \quad (3)$$

The keypoints are independent, having a positive definite covariance matrix. We assume the diagonal terms of covariance matrix  $\Sigma$  would have the same value by defining the covariance matrix as  $\Sigma = \operatorname{diag}(\sigma) = \operatorname{diag}(\exp(\xi))$ . Using [24], (3) can be re-written as:

$$\mathbf{L}(e) = \underset{\sigma}{\operatorname{argmin}} \sum_{i=1}^n \operatorname{tr}(\log \Sigma) + e_i^\top \Sigma^{-1} e_i \quad (4)$$

As shown in Fig. 2, each rotation axis of the primitive image contains 14 keypoints. each axis has colored (red, blue, or green) and white regions. Although we estimate the axis separately at the uncertainty estimation stage, we will combine the three axes later, sharing the white region. The keypoints consist of one centered point and four cornered points in the colored region and as well as centered point and eight cornered points in the white region. One centered point in the white region coincides with the center of the object. We build the keypoint and uncertainty estimation network based on ResNet18 [23]. To train this network, we use (4) per rotation axis primitive image, resulting in (5), allowing

### Algorithm 1 Uncertainty-Based Keypoints Selection

---

**Input:**  $i \in \{x, y, z\}$   
 Estimated Keypoints:  $\mathcal{K}_i = \{\mathcal{K}_{i1}, \dots, \mathcal{K}_{i14}\}$ .  
 Estimated Uncertainties:  $\sigma_i$   
 Threshold:  $\sigma_o$   
**Output:** Merged Keypoints:  $\mathcal{K}_m$

```

1:  $N = 0, x_{1:9} = 0$ 
2: for  $i$  in  $\{x, y, z\}$  do
3:   if  $\sigma_i > \sigma_o$  then
4:     continue
5:   else
6:      $N = N + 1$ 
7:     for  $j = 1 : 9$  do
8:        $x_j = \frac{N-1}{N} x_j + \frac{1}{N} \mathcal{K}_{ij}$ 
9:     end for
10:     $\mathcal{K}_m.append(\mathcal{K}_{i10:i14})$ 
11:  end if
12: end for
13: if  $size(\mathcal{K}_m) == 0$  then
14:   Return False
15: end if
16:  $\mathcal{K}_m.append(x_{1:9})$ 
17: Return  $\mathcal{K}_m$ 

```

---

us to train the keypoints in a supervised manner and the correlated uncertainty in an unsupervised manner, respectively. Consequently, we obtain three sets  $\{\mathcal{K}_x, \mathcal{K}_y, \mathcal{K}_z\}$  of 14 keypoints and three uncertainties  $\sigma = \{\sigma_x, \sigma_y, \sigma_z\}$  per input image.

$$\mathcal{L}_{\mathcal{K}} = \mathbf{L}(e_{\mathcal{K}_x}) + \mathbf{L}(e_{\mathcal{K}_y}) + \mathbf{L}(e_{\mathcal{K}_z}) \quad (5)$$

, while  $e_{\mathcal{K}}$  is the error measured over a set  $\mathcal{K}$ .

4) *Uncertainty-Based Keypoints Selection:* As the final step in the pose initialization, we reject ambiguous primitive axes using the predicted uncertainty. The rotation axis in the primitive image may not be impeccably reconstructed due to occlusion, and symmetry. In such degenerate cases, inferring the 6D object pose using all keypoints rather deteriorates the accuracy, therefore, we need to reject the ambiguous rotation axis based on the estimated uncertainty. Among three rotation axis primitive images  $\{\mathcal{P}_x, \mathcal{P}_y, \mathcal{P}_z\}$  of an object, we examine the uncertainty of each primitive image to check validity (smaller than the threshold  $\sigma_o$ ).

We discard the invalid axis and merge the keypoints only from the valid rotation axes. Fig. 5 shows exemplar primitives from objects. Keypoints from the colored region of the primitive can be immediately confirmed as the final keypoints, whereas the keypoints in the shared white region are averaged over the valid primitives. By averaging, we can ensure a more stable object center. The uncertainty of each object pose factor is the norm of the uncertainties for valid rotation axis primitive images. The algorithm for completing keypoints is Algorithm. 1.

Last, we transform the keypoints back to the whole image plane using the cropped bounding box information, then we calculate the rotation and translation of the object through Perspective-n-Point (PnP) using the confirmed keypoints. As a further enhancement before adding the estimated object pose to the factor graph, for the cases where the norm  $\|\sigma\|$  of valid uncertainties is lower than a threshold  $\sigma_{icp}$ , we perform Iterated Closest Point (ICP) refinement using depth measurement.

#### D. Object Pose Refinement via Graph SLAM

The computed initial relative pose between an object and the camera can be viewed as a factor in the object graph SLAM. We assume that the base of the robot manipulator,

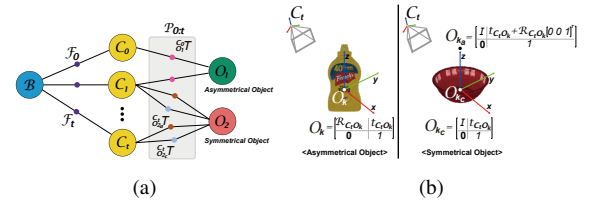


Fig. 3: (a) The factor graph illustration. The colored circles depict the state variables. Two independent measurements are the factors represented by lines that constrain the state at time  $t$ . On every occurrence of the 6D object pose estimation, two factors ( $\mathcal{F}, \mathcal{P}$ ) are added to the factor graph. The object pose factor  $\mathcal{P}$  is added in two ways depending on the type of object. (b) The constraint points definition for the object pose factor.

and the set of objects are stationary and that the relative transformation between the end-effector of the robot manipulator and the camera is given. Based on these assumptions, we constitute the graph SLAM by incorporating three states and two measurement types (Fig. 3(a)).

The state variables consist of the camera pose attached to the end-effector of the robot manipulator  $\mathcal{C}_{0:t} \triangleq \{\mathcal{C}_i^W T \in SE(3) : \forall i \in [0, t]\}$  and the  $k$  number of object pose  $\mathcal{O}_{1:k} \triangleq \{\mathcal{O}_i^W T \in SE(3) : \forall i \in [1, k]\}$  expressed in the base pose of robot manipulator  $\mathcal{B} \triangleq \{\mathcal{B}^W T \in SE(3)\}$  upto current time  $t$ .

$$\mathcal{X}_t = \{\mathcal{B}, \mathcal{C}_{0:t}, \mathcal{O}_{1:k}\} \quad (6)$$

The two types of measurement are comprised of camera pose measurement  $\mathcal{F}_t \triangleq \{\mathcal{B}_t^C T \in SE(3)\}$  and object pose measurement  $\mathcal{P}_t \triangleq \{\mathcal{O}_i^C T \in SE(3) : \forall i \in [1, k]\}$  at time  $t$ .

$$\mathcal{Z}_t = \{\mathcal{F}_t, \mathcal{P}_t\} \quad (7)$$

Since the measurement  $\mathcal{Z}_t$  only depends on the state  $\mathcal{X}_t$ , the posterior probability of our state estimation problem can be written as  $P(\mathcal{X}_t | \mathcal{Z}_t)$ .

$$p(\mathcal{X}_t | \mathcal{Z}_t) \propto p(\mathcal{F}_t, \mathcal{P}_t | \mathcal{B}, \mathcal{C}_{0:t}, \mathcal{O}_{1:k}) \quad (8)$$

$$= \prod_{i=0}^t p(\mathcal{F}_i | \mathcal{B}, \mathcal{C}_i) \prod_{i=0}^t \prod_{j=1}^k p(\mathcal{P}_i | \mathcal{C}_i, \mathcal{O}_j) \quad (9)$$

1) *Camera Pose Factor:* The first term of (9) presents the relative transform between the base  $\mathcal{B}$  and the attached camera  $\mathcal{C}_t$ , which is obtained from the robot forward kinematics using the joint angles. We utilize a standard residual error model for the measurement  $\mathcal{F}_t$  of the camera pose.

$$\Delta \mathbf{t}_{\mathcal{B}\mathcal{C}_t} = \mathbf{R}_{\mathcal{B}}^T (\mathbf{t}_{\mathcal{B}} - \mathbf{t}_{\mathcal{C}_t}) + \delta \mathbf{t}_{\mathcal{B}\mathcal{C}_t} \quad (10)$$

$$\Delta \mathbf{R}_{\mathcal{B}\mathcal{C}_t} = \mathbf{R}_{\mathcal{B}}^T \mathbf{R}_{\mathcal{C}_t} \text{Exp}(\delta \phi_{\mathcal{B}\mathcal{C}_t}) \quad (11)$$

$$\|\mathbf{r}_{\mathcal{B}\mathcal{C}_t}\|^2 = \|\delta \mathbf{t}_{\mathcal{B}\mathcal{C}_t}\|_{\Sigma_{\mathcal{C}_t}}^2 + \|\delta \phi_{\mathcal{B}\mathcal{C}_t}\|_{\Sigma_{\mathcal{C}_t}}^2 \quad (12)$$

2) *Object Pose Factor:* The second term of (9), the object pose factor, describes the relative transform between the camera  $\mathcal{C}_t$  and target object  $\mathcal{O}_k$ . We obtain two types of object states from *Prima6D++* as shown below.

(i) **Asymmetric case:** In the asymmetric cases, we can simply express the residual error in §III-D.1.

$$\|\mathbf{r}_{\mathcal{C}_t \mathcal{O}_k}\|^2 = \|\delta \mathbf{t}_{\mathcal{C}_t \mathcal{O}_k}\|_{\Sigma_{\mathcal{C}_t}}^2 + \|\delta \phi_{\mathcal{C}_t \mathcal{O}_k}\|_{\Sigma_{\mathcal{C}_t}}^2 \quad (13)$$

(ii) **Symmetric case:** To cope with the potential ambiguity in the symmetric cases, we define a factor utilizing the dominant rotation axis of the object. The dominant axis of the object is predefined as the main symmetrical axis of the object based on the 3D CAD model. We begin by defining the dominant axis using two points on the dominant vector, where the object center  $\mathcal{O}_{k_c}$  and an arbitrary point  $\mathcal{O}_{k_a}$  lie on the dominant axis ((b) of Fig. 3).

$$\begin{aligned} \mathbf{t}_{\mathcal{C}_t \mathcal{O}_{k_c}} &= \mathbf{t}_{\mathcal{C}_t \mathcal{O}_k}, \mathbf{R}_{\mathcal{C}_t \mathcal{O}_{k_c}} = \mathbf{R}_{\mathcal{C}_t \mathcal{O}_{k_a}} = \mathbf{I} \\ \mathbf{t}_{\mathcal{C}_t \mathcal{O}_{k_a}} &= \mathbf{t}_{\mathcal{C}_t \mathcal{O}_k} + \mathbf{R}_{\mathcal{C}_t \mathcal{O}_k} [0 \ 0 \ 1]^\top \end{aligned}$$

The residual error is then computed using the transform between these two points  $\mathcal{O}_{k_c}$ ,  $\mathcal{O}_{k_a}$  and the camera  $\mathcal{C}_t$ .

$$\|\mathbf{r}_{\mathcal{C}_t \mathcal{O}_{k_c}}\|^2 = \|\delta \mathbf{t}_{\mathcal{C}_t \mathcal{O}_{k_c}}\|_{\Sigma_{\mathcal{C}_t}}^2 + \|\delta \phi_{\mathcal{C}_t \mathcal{O}_{k_c}}\|_{\Sigma_{\mathcal{C}_t}}^2 \quad (14)$$

$$\|\mathbf{r}_{\mathcal{C}_t \mathcal{O}_{k_a}}\|^2 = \|\delta \mathbf{t}_{\mathcal{C}_t \mathcal{O}_{k_a}}\|_{\Sigma_{\mathcal{C}_t}}^2 + \|\delta \phi_{\mathcal{C}_t \mathcal{O}_{k_a}}\|_{\Sigma_{\mathcal{C}_t}}^2 \quad (15)$$

The initial covariance of the object pose is defined by  $\Sigma_{\mathcal{C}_t} = \text{diag}(\|\boldsymbol{\sigma}_t\|)$ .

3) *Pose Graph Optimization:* In our approach, we optimize the multi-object poses using pose graph optimization. Assuming that measurement noises are zero-mean normal distributions  $\mathcal{N}(0, \Sigma)$ , the maximum *a posteriori* (MAP) estimate of state  $\mathcal{X}_t$  is obtained by minimizing the negative log of probability, which is the same as minimizing the sum of squared residual errors.

$$\hat{\mathcal{X}}_t = -\underset{\mathcal{X}_t}{\text{argmin}} \log p(\mathcal{X}_t | \mathcal{Z}_t) \quad (16)$$

$$= \underset{\mathcal{X}_t}{\text{argmin}} \sum_{i=0}^t \|\mathbf{r}_{\mathcal{B} \mathcal{C}_i}\|_{\Sigma_{\mathcal{C}_i}}^2 + \alpha \left( \sum_{i=0}^t \sum_{j=1}^k \|\mathbf{r}_{\mathcal{C}_i \mathcal{O}_j}\|_{\Sigma_{\mathcal{C}_i}}^2 \right) +$$

$$(1 - \alpha) \left( \sum_{i=0}^t \sum_{j=1}^k \|\mathbf{r}_{\mathcal{C}_i \mathcal{O}_{j_c}}\|_{\Sigma_{\mathcal{C}_i}}^2 + \sum_{i=0}^t \sum_{j=1}^k \|\mathbf{r}_{\mathcal{C}_i \mathcal{O}_{j_a}}\|_{\Sigma_{\mathcal{C}_i}}^2 \right)$$

$$\alpha = \begin{cases} 1, & \text{if } \mathcal{O}_j \text{ is asymmetrical case} \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

#### IV. EXPERIMENT

##### A. Dataset and Implementation Detail

In total, two datasets were used for the evaluation: T-LESS dataset [25] and YCB-Video [1]. The two datasets provide the relative transform between the world and camera coordinate per scene. The YCB-Video dataset is composed of 21 high-quality 3D models, offering 92 annotated video sequences. These sequences include various lighting conditions, noise in the capture, and occlusion. The T-LESS dataset is composed of 30 3D models and offers 20 annotated video sequences. This dataset is challenging given that most objects are symmetrical and texture-less.

For training, we synthesized 100K images per the object together with their associated three rotation axis primitive images. The synthetic images were further augmented using the “imgaug” library [26]. We also utilized 50K photorealistic synthetic images from [27] and other available real image training set from public datasets. The proposed network was

implemented using Pytorch and was trained with NVIDIA RTX 2080Ti. All of the networks were trained using Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and a learning rate of  $10^{-4}$ . The segmentation network was trained for 50 epochs with a batch size of 10. Input images with a resolution of  $640 \times 480$  were used, and domain randomization was used as the data augmentation. Other networks were trained together for 220 epochs with a batch size of 100. Here, we used input images with a resolution of  $64 \times 64$  for training. For single-view inference, we set a threshold  $\sigma_o = 0.4$ . In the optimization phase, we built and solved the factor graph using gtsam [28] with the threshold  $\sigma_{icp} = 0.4$ .

##### B. Evaluation Metrics

(i) **Average distance metric:** Similar to [1], average distance metric ADD(-S) is selected for evaluation. ADD(-S) computes the distance of the transformed 3D model points between the ground truth pose and predicted pose. We also compute the area under the curve varying the distance threshold to evaluate performance of pose estimation.

(ii) **Standard visual surface discrepancy:** We measure the distance between the estimated and ground truth of visible surface on object depth. As in [29], we consider the sample as a true positive case if the error is less than 0.3 on the tolerance  $\tau = 20$  mm and visibility  $v > 0.1$ .

(ii) **BOP challenge metrics:** We use the metrics utilized in the BOP challenge [30]. It evaluates the average recall of errors measured by Visible Surface discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD). The overall performance is calculated as average recall  $\text{AR} = (\text{AR}_{\text{vsd}} + \text{AR}_{\text{mssd}} + \text{AR}_{\text{mspd}})/3$ .

##### C. Evaluation of Uncertainty

We evaluate the correlation between the estimated uncertainties and the associated 6D object pose. We further discuss the uncertainty variation under object symmetry and occlusions and the effect of ambiguity rejection.

1) *Uncertainty for Keypoint Error Measure:* To validate capacity of uncertainty estimation quantitatively, we select two representative objects, one of each asymmetrical and symmetrical objects from the YCB-Video dataset. By performing image augmentation, we generate 1,000 image samples from each object having illumination and occlusion variation. Using these images, we examine whether the estimated uncertainty captures the true keypoint error.

The result is as shown in Fig. 4(a), where we present the relation between the estimated keypoints errors and uncertainties per axis. As for the asymmetrical object, as the keypoint errors of all axes increase, the uncertainties also increase. This means the estimated uncertainties reflect the tendency of the keypoint errors. For the case of the symmetrical object, the relation between the  $z$ -axis keypoint errors  $e_{\mathcal{K}_z}$  and uncertainties  $\sigma_z$  in the third column shows the monotonically increasing form. As for the other graphs, they hardly represent any correlation, since the primitive images of a symmetrical object are rarely reconstructed in

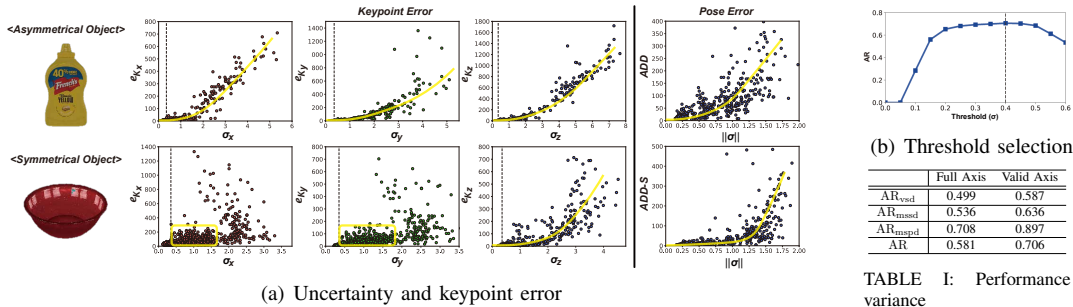


Fig. 4: (a) The scatter plots represent the relation between the estimated uncertainties and keypoint errors and the relation between the norm of estimated uncertainties and ADD(-S) error. The first row illustrates the mustard bottle of the YCB-Video dataset as the typical asymmetrical object. The second row illustrates the bowl of the YCB-Video dataset as the representative symmetrical object. The black vertical dashed line is a threshold  $\sigma_o = 0.4$ . It can be observed that higher uncertainties lead to higher keypoint errors and object pose errors. (b) The graph depicts the performance variation with the threshold  $\sigma_o$ . The black vertical dashed line is a threshold  $\sigma_o = 0.4$  used in our system. The table represents the performance variance according to uncertainty-based keypoints selection.

its unique form except for the dominant axis. In particular, since multiple poses can correspond to a single state of the symmetrical object, we observe that in unstable regions, the keypoint errors fluctuate with increasing uncertainties. By doing so, the proposed approach estimates the 6D object pose in an identical way that can reject the ambiguous axes as the advantage of uncertainty estimation.

2) *Threshold ( $\sigma_o$ ) Selection:* Using these attributes, we can find the dominant axis and ambiguous axes of the symmetrical object based on the threshold  $\sigma_o = 0.4$  and reject the ambiguous axes when inferring the 6D object pose. For this uncertainty-based selection, we have to set the proper threshold  $\sigma_o$ . In Fig. 4(b), we describe the performance variation over the T-LESS dataset in terms of AR by increasing the threshold from 0 to 0.6 by 0.05 steps. When the threshold  $\sigma_o = 0.4$ , it achieves the highest performance, and further threshold increase causes a gradual performance drop. In addition, by setting the threshold  $\sigma_o = \infty$ , the performance of using all axis primitives is shown in Table. I. The case of inferring with all axis primitives cannot handle object ambiguity, thus the performance drops by 0.125 in terms of AR. As a result, we select the threshold of  $\sigma_o = 0.4$ .

3) *Uncertainty for Ambiguity Awareness:* Next, we evaluate uncertainty as a measure of ambiguity awareness in a qualitative manner. We infer per-axis uncertainty over the T-LESS dataset and our own dataset using YCB objects. As mentioned earlier, even asymmetric objects may look symmetric depending on the viewpoint, and we validate whether this view-induced ambiguity is well detected from the reconstructed primitive images.

Exemplar cases of the object ambiguity awareness are presented in (Fig. 5). The merged keypoints are built using three valid axes for typical asymmetrical objects in (a) and (b). For the symmetrical object in (f), (h), (i), and (k) as well as the case of partial symmetry in (g) and (l), the merged keypoints are built with the keypoints extracted from the valid primitives. Moreover, we can observe that the direction of the estimated dominant axis coincides with the objects as in (f) and (h). It is noteworthy that if an asymmetric object is seen as a symmetric object depending on the camera viewpoint, as shown in (d) and (e), our method estimates the 6D object pose exploiting valid primitive images. In addition, we observe that for the occlusion, if patches that determine

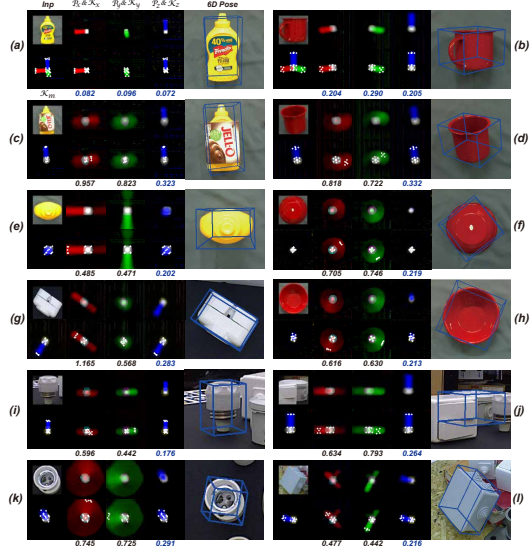


Fig. 5: The qualitative evaluation of ambiguity-aware 6D object pose estimation, PrimA6D++. The first column shows the cropped input image and merged keypoints obtained through uncertainty-based keypoints selection. Here, we set a threshold  $\sigma_o = 0.4$ . The second to fourth columns show the estimated rotation axis primitive images and their keypoints. The fifth column represents the 6D object pose estimated using merged keypoints. The below numbers indicate the estimated uncertainties corresponding to the primitive images.

the directionality of an object are visible, then our method reconstructs primitive images as shown in (c) and (j). As can be seen from these examples, PrimA6D++ can discern the case of object ambiguity through uncertainty estimation. Thanks to this superiority, the proposed method performs better than other RGB-based single view methods on the YCB-VIDEO dataset with several cases of occlusions and the T-LESS dataset with multiple symmetrical objects, as shown in Table. II and Table. III.

4) *Uncertainty for Object Pose Factor:* In our approach, the uncertainties of three primitive images are combined with the uncertainty for the final object pose factor. This section evaluates the relation between the object pose errors and the estimated uncertainties.

As shown in the last column of Fig. 4(a), we present the relation between the average distance errors of 6D object pose estimation and the norm  $\|\sigma\|$  of valid uncertainties. In the asymmetrical object, an increase in average distance errors yields an increase in the uncertainties. In the symmet-

| methods   | Single-View  |              |           |                 |                      | Multi-View    |                           |                           |                        |                           |
|-----------|--------------|--------------|-----------|-----------------|----------------------|---------------|---------------------------|---------------------------|------------------------|---------------------------|
|           | GDR-Net [31] | CosyPose [3] | PrimA6D++ | MoreFusion [17] | FFB6D [32]<br>w/ ICP | SUO-SLAM [20] | CosyPose [3]<br>w/ 8-View | PrimA6D++<br>w/ Graph Opt | PoseRBPF [5]<br>w/ SDF | PrimA6D++<br>w/ Graph Opt |
| #. models | N            | 1            | N         | 1               | N                    | 1             | 1                         | N                         | N                      | N                         |
| data type | RGB          | RGB          | RGB       | RGB-D           | RGB-D                | RGB           | RGB                       | RGB                       | RGB-D                  | RGB-D                     |
| ADD(-S)   | 84.4         | 84.5         | 91.1      | 91.0            | 93.1                 | 84.7          | -                         | 91.5                      | 87.5                   | <b>94.4</b>               |
| ADD-S     | 91.6         | 89.8         | 94.3      | 95.7            | <b>97.0</b>          | 90.3          | 93.4                      | 94.8                      | 95.2                   | 96.9                      |
| AR        | 0.755        | 0.821        | 0.845     | -               | 0.855                | -             | 0.853                     | 0.861                     | -                      | 0.936                     |

TABLE II: The result of YCB-Video dataset on the ADD(-S) and ADD-S metrics. PrimA6D++ with graph optimization is divided into RGB and RGB-D depending on the presence of ICP refinement. N refers to the total number of objects in the dataset.

| method             | Single View |             |                |              |           |       |              | Multi-View                |                           |                           |
|--------------------|-------------|-------------|----------------|--------------|-----------|-------|--------------|---------------------------|---------------------------|---------------------------|
|                    | CDPN [33]   | WS-OPE [34] | ZebraPose [35] | CosyPose [3] | PrimA6D++ | [14]  | SurfEmb [36] | CosyPose [3]<br>w/ 8-View | PrimA6D++<br>w/ Graph Opt | PrimA6D++<br>w/ Graph Opt |
| #. models          | 1           | 1           | N              | 1            | N         | N     | 1            | 1                         | N                         | N                         |
| data type          | RGB         | RGB         | RGB            | RGB          | RGB       | RGB-D | RGB-D        | RGB                       | RGB                       | RGB-D                     |
| AR <sub>vsd</sub>  | 0.377       | 0.476       | 0.459          | 0.669        | 0.587     | 0.760 | <b>0.797</b> | 0.773                     | 0.692                     | 0.752                     |
| AR <sub>mssd</sub> | 0.418       | 0.587       | 0.618          | 0.695        | 0.636     | 0.773 | 0.829        | 0.836                     | 0.920                     | <b>0.953</b>              |
| AR <sub>mspd</sub> | 0.674       | 0.652       | 0.732          | 0.821        | 0.897     | 0.795 | 0.859        | 0.907                     | 0.935                     | <b>0.963</b>              |
| AR                 | 0.490       | 0.572       | 0.603          | 0.728        | 0.706     | 0.776 | 0.828        | 0.839                     | 0.849                     | <b>0.889</b>              |

TABLE III: The result of the T-LESS dataset on the average recall metrics of three metrics used in the BOP challenge. PrimA6D++ with graph optimization is divided into RGB and RGB-D depending on the presence of ICP refinement. N refers to the total number of objects in the dataset.

| method    | Multi-View    |                           |                        |                           |
|-----------|---------------|---------------------------|------------------------|---------------------------|
|           | SUO-SLAM [20] | CosyPose [3]<br>w/ 8-View | PoseRBPF [5]<br>w/ SDF | PrimA6D++<br>w/ Graph Opt |
| data type | RGB           | RGB                       | RGB-D                  | RGB-D                     |
| $e_{vsd}$ | 63.7          | 71.6                      | 82.6                   | <b>87.8</b>               |

TABLE IV: The result of the T-LESS dataset on the  $e_{vsd}$  metric.

rical object, we find a slight stagnation in some areas due to the ambiguity of the symmetrical object; afterwards, the trend line increases monotonically. This aspect is similar to the  $\sigma_z$  graph of Fig. 4(a). We can conclude that the norm  $\|\sigma\|$  of valid uncertainties positively correlates with the 6D object pose errors.

#### D. Results on the public datasets

1) *YCB-Video dataset*: We evaluate the performance of our method with other state-of-the-art (SOTA) methods on the YCB-Video dataset. The qualitative results for the YCB-Video dataset are presented in Fig. 6. As shown in Table. II, PrimA6D++ shows the best performance in RGB-based single-view methods as well as the multi-view method compared to other methods. Since the proposed method deals with the ambiguity of object pose through uncertainty estimation, we obtain superior performance on the YCB-V dataset with various challenging points, including occlusion and clutter. Through graph optimization, our method outperforms all other methods in terms of the ADD(-S) metric.

2) *T-LESS dataset*: PrimA6D++ outperforms others with graph optimization in terms of  $e_{vsd}$  (Table. IV). Furthermore, we compare our method with those reported in the BOP challenge, as shown in Table. III. PrimA6D++ achieves the second highest performance in terms of AR. However, it is worth noting that PrimA6D++ outperforms other methods in terms of AR<sub>mspd</sub> which is relevant for AR applications and suitable for evaluating RGB-only methods. This is because



Fig. 6: The qualitative result of multi-object pose estimation. The first row represents the estimated poses and the second row is the reconstructed scenes using 3D object models on the T-LESS and YCB-Video dataset.

PrimA6D++ estimates the 6D object pose by extracting the primitive axes coincident with the dominant axes of the objects visible in the images. In addition, with the graph optimization, PrimA6D++ shows a significant performance improvement over all of the metrics, especially AR<sub>mssd</sub> which is deeply related to the chance of a successful robotic grasping. We conjecture that with the graph optimization of sequential data, the comprehension of the target workspace in 3D spaces gradually increases, which leads to improved accuracy. As a result, we achieve object pose accuracy sufficient for robot manipulation. The qualitative results for the T-LESS dataset are presented in Fig. 6.

3) *Effect of ICP refinement*: Before adding the estimated object pose proposals, we perform ICP refinement using depth measurement for the cases where the norm  $\|\sigma\|$  of valid uncertainties is lower than a threshold  $\sigma_{icp}$ . In this section, we evaluate the effect of ICP refinement on the performance of graph optimization. Table. II and Table. III list the performance variation depending on the presence of ICP refinement of PrimA6D++ with graph optimization. Even without ICP refinement, PrimA6D++ with graph optimization creates an exorbitant performance improvement since it guarantees global consistency of multi-view. ICP refinement on the object pose proposals further improves accuracy.

#### E. Real-Time Full System Demo

For the real-time full system demonstration, we use the robot system as shown in Fig. 2. Since one network is trained to handle one object, the inference is performed in parallel using four GPUs to account for multiple objects. To recognize the objects in the workspace, the robot manipulator performs a scan around the objects; concurrently, the multi-object pose estimation is executed. In the end, with our proposed method, we demonstrate real-time scene recognition capability for visually-assisted robot manipulation. The experimental results are available in the supplementary video materials `prima6d.mp4`.

## V. CONCLUSION

In this paper, we introduced a pipeline for multi-object pose estimation, which incorporates our novel ambiguity-

aware 6D object pose estimation network. We devised a network to reconstruct the three rotation axis primitive images separately and infer the underlying uncertainty along each axis. Afterwards, leveraging the estimated uncertainty, we optimized these visual measurements and camera poses through a factor graph optimization. In the experiment, we validated the effect of ambiguity awareness using estimated uncertainties on the 6D object pose estimation, as well as the correlation between the estimated uncertainties and object pose errors. We demonstrated that the accuracy of multi-object pose estimation was improved through graph optimization. We further laid the groundwork for visually-assisted robot manipulation through real-time scene recognition.

#### REFERENCES

- [1] Y. Xiang *et al.*, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” in *Proc. Robot.: Science & Sys. Conf.*, June 2018.
- [2] S. Peng *et al.*, “Pvnet: Pixel-wise voting network for 6dof pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 4561–4570.
- [3] Y. a. Labbé, “Cosypose: Consistent multi-view multi-object 6d pose estimation,” in *Proc. European Conf. on Comput. Vision*, 2020, pp. 574–591.
- [4] C. Wang *et al.*, “Densefusion: 6d object pose estimation by iterative dense fusion,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 3343–3352.
- [5] X. Deng *et al.*, “Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking,” *IEEE Trans. Robot.*, 2021.
- [6] B. Okorn *et al.*, “Learning orientation distributions for object pose estimation,” in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2020, pp. 10 580–10 587.
- [7] F. Manhardt *et al.*, “Explaining the ambiguity of object detection and 6d pose from visual data,” in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2019, pp. 6841–6850.
- [8] J. Fu, Q. Huang *et al.*, “A multi-hypothesis approach to pose ambiguity in object-based SLAM,” in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2021.
- [9] K. Murphy *et al.*, “Rao-blackwellised particle filtering for dynamic bayesian networks,” in *Sequential Monte Carlo methods in practice*, 2001, pp. 499–515.
- [10] M.-H. Jeon *et al.*, “Prima6d: Rotational primitive reconstruction for enhanced and robust 6d pose estimation,” *IEEE Robot. and Automat. Lett.*, vol. 5, no. 3, pp. 4955–4962, 2020.
- [11] S. Zakharov *et al.*, “Dpod: 6d pose object detector and refiner,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 1941–1950.
- [12] S. Iwase *et al.*, “Repose: Fast 6d object pose refinement via deep texture rendering,” in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2021, pp. 3303–3312.
- [13] Y. Xu *et al.*, “Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2022.
- [14] L. Lipson *et al.*, “Coupled iterative refinement for 6d multi-object pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2022.
- [15] B. Wen *et al.*, “se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains,” in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2020, pp. 10 367–10 373.
- [16] X. Deng *et al.*, “Self-supervised 6d object pose estimation for robot manipulation,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2020, pp. 3665–3671.
- [17] K. Wada *et al.*, “Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2020, pp. 14 540–14 549.
- [18] I. Shugurov *et al.*, “Multi-view object pose refinement with differentiable renderer,” *IEEE Robot. and Automat. Lett.*, vol. 6, no. 2, pp. 2579–2586, 2021.
- [19] B. Wen *et al.*, “Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models,” in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2021, pp. 8067–8074.
- [20] N. Merrill *et al.*, “Symmetry and uncertainty-aware object slam for 6dof object pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2022.
- [21] G. Shi *et al.*, “Fast uncertainty quantification for deep object pose estimation,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2021.
- [22] H. Deng *et al.*, “Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation,” *Intl. J. of Comput. Vision*, pp. 1–28, 2022.
- [23] K. He *et al.*, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2016, pp. 770–778.
- [24] K. Liu *et al.*, “Deep inference for covariance estimation: Learning gaussian noise models for state estimation,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2018, pp. 1436–1443.
- [25] T. Hodaň *et al.*, “T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects,” *Proc. IEEE Winter Conf. on Appl. Comput. Vision.*, 2017.
- [26] A. B. Jung *et al.*, “imgaug,” <https://github.com/aleju/imgaug>, 2020.
- [27] T. Hodaň *et al.*, “Photorealistic image synthesis for object instance detection,” in *Proc. Intl. Conf. on Image Processing*, 2019, pp. 66–70.
- [28] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep., 2012.
- [29] M. Sundermeyer *et al.*, “Implicit 3d orientation learning for 6d object detection from rgb images,” in *Proc. European Conf. on Comput. Vision*, 2018, pp. 699–715.
- [30] T. Hodan *et al.*, “Bop: Benchmark for 6d object pose estimation,” in *Proc. European Conf. on Comput. Vision*, 2018, pp. 19–34.
- [31] G. Wang *et al.*, “Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2021, pp. 16 611–16 621.
- [32] Y. He *et al.*, “Ffb6d: A full flow bidirectional fusion network for 6d pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, June 2021.
- [33] Z. Li *et al.*, “Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation,” in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2019, pp. 7678–7687.
- [34] F. Li *et al.*, “Ws-ope: Weakly supervised 6-d object pose regression using relative multi-camera pose constraints,” *IEEE Robot. and Automat. Lett.*, vol. 7, 2022.
- [35] Y. Su *et al.*, “ZebraPose: Coarse to fine surface encoding for 6dof object pose estimation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2022, pp. 6738–6748.
- [36] R. L. Haugaard *et al.*, “Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2022, pp. 6749–6758.