DOI: 10.1111/cgf.14684
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

SO(3)-Pose: SO(3)-Equivariance Learning for 6D Object Pose Estimation

Haoran Pan[1,2], Jun Zhou[3], Yuanpeng Liu[1], Xuequan Lu[4], Weiming Wang[5†], Xuefeng Yan[1], Mingqiang Wei[1,2†]

[1]School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2]Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen, China
[3]School of Nursing, Hong Kong Polytechnic University, Hong Kong, China
[4]School of Information Technology, Deakin University, Geelong, Australia
[5]School of Science and Technology, Hong Kong Metropolitan University, Hong Kong, China

Abstract

6D pose estimation of rigid objects from RGB-D images is crucial for object grasping and manipulation in robotics. Although RGB channels and the depth (D) channel are often complementary, providing respectively the appearance and geometry information, it is still non-trivial on how to fully benefit from the two cross-modal data. From the simple yet new observation, when an object rotates, its semantic label is invariant to the pose while its keypoint offset direction is variant to the pose. To this end, we present SO(3)-Pose, a new representation learning network to explore SO(3)-equivariant and SO(3)-invariant features from the depth channel for pose estimation. The SO(3)-invariant features facilitate to learn more distinctive representations for segmenting objects with similar appearance from RGB channels. The SO(3)-equivariant features communicate with RGB features to deduce the (missed) geometry for detecting keypoints of an object with the reflective surface from the depth channel. Unlike most of existing pose estimation methods, our SO(3)-Pose not only implements the information communication between the RGB and depth channels, but also naturally absorbs the SO(3)-equivariance geometry knowledge from depth images, leading to better appearance and geometry representation learning. Comprehensive experiments show that our method achieves the state-of-the-art performance on three benchmarks. Code is available at https://github.com/phaoran9999/SO3-Pose.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models;***

1. Introduction

6D pose estimation predicts a rigid transformation (i.e., 3D rotation and translation) from the 3D coordinate system of the object to the 3D coordinate system of the camera. Accurate 6D pose enables a robot to interact with target objects in the environment effectively [CTT*12, DWLZ21]. Although recent years have witnessed a spurt of progress in 6D object pose estimation, it is still challenging and many open problems remains to solve, due to the heavy occlusions, cluttered backgrounds, and varying illuminations in the environment.

Recent methods mainly depend on learning-based techniques (e.g., CNN) [PZL*20] for 6D pose estimation. That is, at the training stage, cutting-edging models learn from 3D objects and images that contain the objects in known 6D poses; and at the test stage, given a list of object instances and an image with the objects visible in it, the trained models infer 6D poses of the listed object instances. Initially, the inputs for these learning-based models are only RGB images. When encountering the cases of 1) similar, occluded or texture-free objects, 2) poor lighting conditions, and 3)

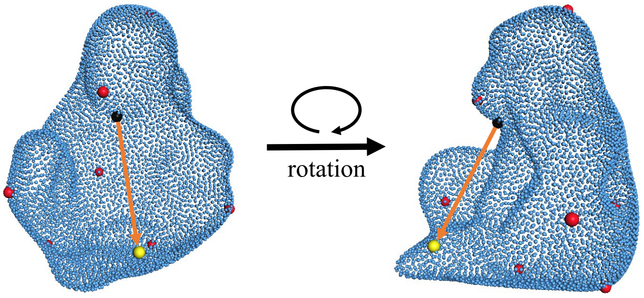
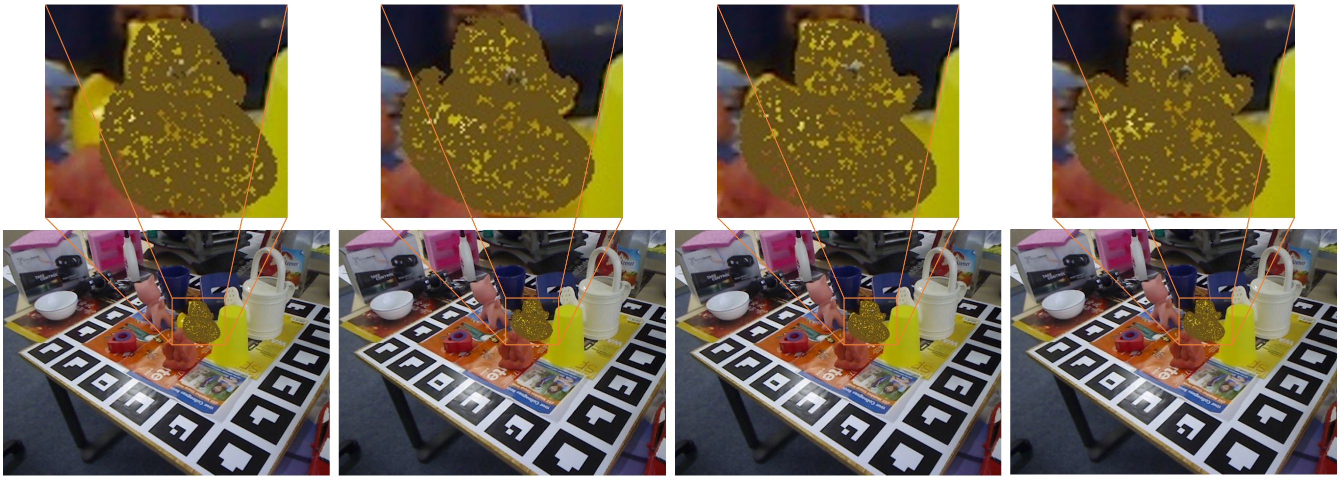


Figure 1: *SO(3) visualization. The red dots represent keypoints and the yellow dot is one of the keypoints. The black dot is an appearance point on the object. The semantic label (black dot) is invariant and the keypoint offset direction (the orange arrow between the black dot and the yellow dot) is variant when the object rotates.*

[†] Co-corresponding authors: W. Wang and M. Wei.



FFB6D SS-Conv Ours GT

Figure 2: *Comparison of our SO(3)-Pose with FFB6D [HHF*21] and SS-Conv [LLC*21] for pose estimation on the LineMOD dataset. Our SO(3)-Pose predicts a more accurate 6D object (see the Duck) pose that best fits the duck in the image than the state-of-the-arts. Please note that, the object vertices in the object coordinate system are transformed by the predicted poses from individual methods to the camera coordinate system and then projected to the image with the camera intrinsic matrix.*

low-contrast scenes, these models become difficult to learn distinctive representations from RGB images, due to lack of the geometry information. The recently popular 3D sensors, such as Microsoft Kinect, Velodyne LiDAR, Intel RealSense and LiDAR scanner of Apple iPad Pro, can capture the real world into RGB-D images. The extra depth (D) information is promising to assist the pose estimation problem.

Given an RGB-D image, how to fully benefit from the two cross-modalities for better 6D pose estimation is still an open problem. One common way of handling the cross-modalities is to extract appearance features and geometry features separately by two-stream networks, e.g., DenseFusion [WXZ*19], and PVN3D [HSH*20]. The appearance and geometry features from the two steams are fused and assigned to each pixel to realize the pose estimation. However, since the two streams seldom interact with each other to obtain the mutual gain, their performance surfers from degeneration sometimes, e.g., objects with similar appearance or with reflective surfaces. FFB6D [HHF*21] pioneers to communicate between the two streams by constructing bidirectional fusion modules. Thus, the two streams are encouraged to share the local-and-global complementary information from each stream for learning appearance and geometry representations. Although FFB6D achieves the state-of-the-art performance, the extracted features are still simply fused to learn the semantic segmentation and keypoint offset, without exploring geometry domain knowledge. Furthermore, since the full flow bidirectional fusion operation is applied on each encoding and decoding layer of the two networks, the excessive interaction of the appearance and geometry features leads to colossal time consumption, which hinders its capability on real-time applications. By contrast, our work delves into the geometric properties from the perspective of rigid transformations and turns to learn SO(3) equivalence and invariance instead of conventional geometric features, and therefore estimates the 6D object pose effectively. Herein, SO(3)-equivariance/invariance is defined as a characteristic for feature mapping that the input and output have equivalent/invariant transformation effects with a given instance in the manifold space.

In this work, we propose SO(3)-Pose, a new representation learning network to explore SO(3)-equivariant and SO(3)-invariant features from point clouds for instance-level object pose estimation which needs object CAD model. SO(3)-Pose adopts a two-stage strategy: 1) jointly segment the target objects from the RGB image with the help of SO(3)-invariant features and regress the keypoints of objects from the depth image with the help of SO(3)-equivariant features; and 2) given the keypoints, a PnP optimization problem is solved to produce the pose parameters. From our observation, the semantic label is SO(3)-invariant and the keypoint offset is SO(3)-equivariant (see Fig. 1). To this end, we design an SO(3)-equivariant encoder to extract SO(3)-equivariant features from the point cloud, and meanwhile develop an equivariant-to-invariant layer (E2Ilayer) to convert the SO(3)-equivariant features to the SO(3)-invariant features. The SO(3)-equivariant features and the RGB features are aggregated to localize keypoints, while the SO(3)-invariant features and the RGB features are aggregated to segment object instances.

Our method achieves the state-of-the-art performance (see Fig. 2). In summary, our contributions are as follows:

1) We propose a novel 6D object pose estimation network, which introduces SO(3)-equivariance for representation learning. To the best of our knowledge, this is the first work to introduce SO(3)-equivariance to pose estimation.
2) We design a new module termed E2Ilayer, which effectively converts the SO(3)-equivariant features to the SO(3)-invariant features; we also design a novel loss, dubbed as SO3loss, to guide the SO(3)-equivariance learning.

3) We show the SO(3)-equivariance benefits both tasks of semantic segmentation and keypoint detection by developing individual feature fusion modules.

2. Related Work

2.1. Traditional approaches

In the traditional setting, methods focus on holistic and local object shape representation learning. These methods can be roughly divided into two groups, template-based and descriptor-based, which are distinguished by how to utilize feature embedding and clustering for pose estimation.

The first kind mainly aims at computing holistic shape description for each target model [DWL19, HLI*12, HZL*15]. Specifically, the key technique of this type of methods is template matching. These templates are generated by projecting a 3D model onto different image planes from various viewpoints, and each template has a pose parameter. In the inference phase, the final object pose will be recovered by the correlation coefficient between the query window and the template [ZC17]. As an early attempt for this task, Hinterstoisser et al. [HLI*12] proposed a classical framework, which integrates image gradients and surface normals for robust feature embedding. This scheme achieves distinguished holistic shape representation and significant detection performance. Rios-Cabrera et al. [RCT13] devised a real-time scalable approach based on LINE2D/LINEMOD [HCI*11]. They also designed a novel strategy to distinguish templates by employing a cluster manner. This kind of methods work well with texture-less objects, but the detection performance will be severely degraded when severe occlusion exists in real scenes.

The second group of methods performs either 2D-3D or 3D-3D matching to establish correspondences in feature space, and the final pose parameters can be recovered by solving the PnP problem [LMNF09]. The core insight of this class of methods is to extract robust feature points both on the image plane and 3D geometry surface [JMX*21, MJF*21, HJX*18, LWZ21], and employs feature description on each keypoint to obtain a series of matching points with feature similarity between them. One of the most representative approaches was introduced by Mur-Artal et al. [MAMT15], which proposed a novel 2D descriptor called ORB. The ORB features are firstly extracted on the given image and the pose parameters of keyframes are calculated by the constructed correspondences from the 2D pixels to 3D points. Correspondingly, when depth data is available, the 3D descriptors [STDS14, HMZA21, DUNI10, HLRK16, VLLM18, ZLL*20] can be built on the model surfaces and the correspondence matrix will be filled with the matched 3D points. The pose estimation task turns into a rigid registration problem. Once the correspondence is established, the final pose can be solved by utilizing the SVD algorithm. In general, descriptor-based approaches have advantages in terms of computation complexity and robustness to partial occlusion, since only the local structural features of the model are highlighted and pushed into downstream calculation stages. However, due to the locality injection property of these methods, the proposed 3D descriptors are sensitive to the variation of the environment, e.g., illumination.

2.2. Learning-based approaches

Recently, with the prevailing attention of deep learning techniques, learning-based approaches have been deployed into many vision fields and reveal remarkable progress such as image classification [ATC*21], object detection [DCLC21], semantic segmentation [WZA*21] etc. Similarly, learning-based methods can be roughly divided into two categories, i.e., holistic and semi-holistic approaches.

Holistic approaches regress the 3D translation and orientation components of the target object directly from the given input data [KMT*17, SHX*21]. Xiang et al. [XSNF17] introduced a novel PoseCnn framework, which estimated the 3D translation by combining the center location of the target object in the image with the camera parameters. The 3D rotation is regressed from the neural network directly in a quaternion form by utilizing the position and the semantic label of the object. Li et al. [LWJ*18] presented a new scheme to regress the 6D pose in an iterative manner, which iteratively refines the pose from initial pose parameters by matching the test image against the rendered image. Sundermeyer et al. [SMD*18] proposed a real-time method to learn an implicit representation of the target, and leveraged an augmented auto-encoder to regress the 6D pose from the latent space. Wang et al. [WXZ*19] designed a novel multi-modal feature fusion framework, which utilizes two feature embedding branches to respectively extract the color and the geometry cues, and employs a pixel-wise feature fusion mechanism to achieve deep interaction of the multi-modality data. This line of work can achieve satisfactory performance in several benchmarks and be capable to implement end-to-end network architecture directly. However, the generalization and learning ability of these methods are restricted by the non-linearity of rotation space.

Current semi-holistic approaches first extract the keypoints on the surface of the target model, and then utilize a PnP algorithm or Least-Squares Fitting [GPH19, RL17, TSF18]. Hu et al. [HHFS19] presented a two-stream segmentation-driven framework, which utilizes each visible cell assigned by the segmentation stream to predict the 2D keypoints locations of the corresponding object. Peng et al. [PZL*20] proposed a novel pixel-wise voting method to predict 2D keypoints of the target object in a given image. The dense keypoints locations are predicted by regressing pixel-wise vectors pointing to the keypoints. He et al. [HHF*21] introduced a new bidirectional fusion network, which achieves RGB-D feature fusion effectively and employs the keypoints regression network branch proposed by [HSH*20] to obtain 3D keyponts. Keypoint-based methods perform well in occlusion scenarios. However, the detection performance will be severely degraded for texture-less objects when only the RGB images can be used.

2.3. Equivariant and Invariant Representation Learning

Recently, equivariance and invariance, as essential properties of point cloud processing in the 3D vision field, have received extensive attention.

Lin et al. [LLC*21] presented a novel convolution SS-Conv for efficient learning of SE(3)-equivariant features in the point cloud,

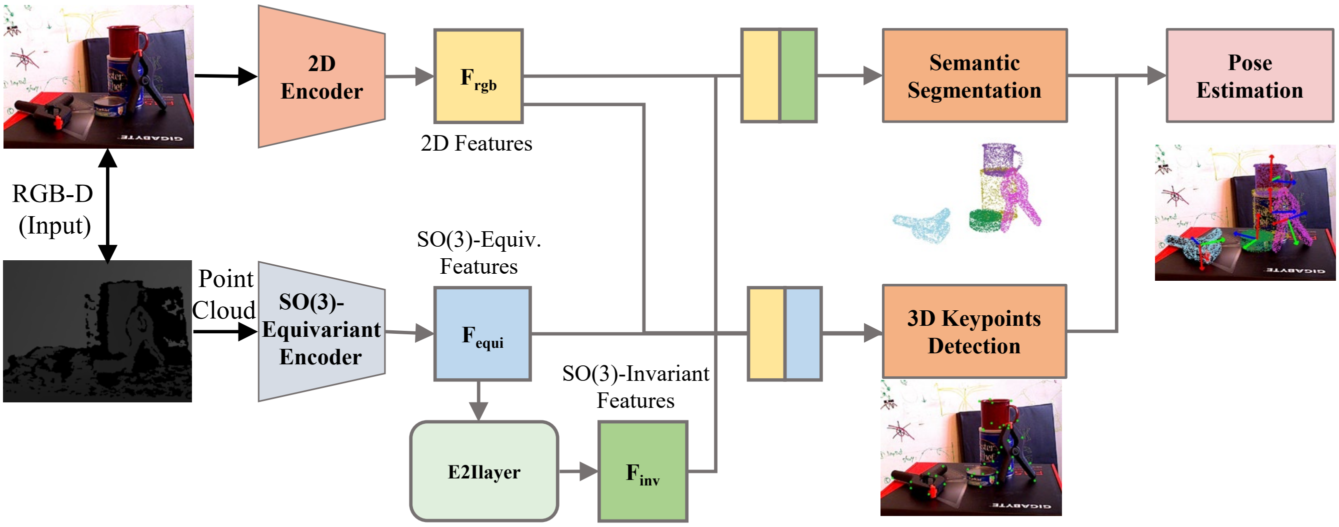


Figure 3: *Overview of our SO(3)-Pose. SO(3)-equivariance is a geometric property in that a feature rotates the same degree and direction as the object rotates. The SO(3)-invariant feature F_{inv} is obtained from the SO(3)-equivariant feature F_{equi} through the designed E2Ilayer. F_{inv} and F_{rgb} are fused and fed into a semantic segmentation module. F_{equi} and F_{rgb} are fused and fed into a 3D keypoint detection module to obtain per-object keypoints. Finally, the 6D pose parameters are fitted with the 3D-3D correspondence of keypoints.*

which designs a sparse steerable kernel based on spherical harmonics function. Li et al. [LWY*21] proposed a new SE(3)-equivariant point cloud network for category-level object pose estimation, which decouples the pose and object geometry shape by deploying an SE(3)-invariant shape reconstruction module and an SE(3)-equivariant pose estimation module. Boulevard et al. [PRPO19] introduced a novel SO(3)-invariance architecture to process point cloud directly, which employs a spherical harmonics-based kernel at different layers of the network to inject invariant features at the global and local levels. Deng et al. [DLD*21] introduced a general framework for SO(3)-equivariant feature extraction in the point cloud and designed a vector neurons representation in feature space.

3. Method

Our goal is to estimate the 6D pose parameters of a set of known objects from an RGB-D image. The 6D pose parameters can be represented by a rigid transformation $T \in SE(3)$ from the object coordinate system to the camera coordinate system, which consists of a 3D rotation matrix $R \in SO(3)$ and a 3D translation vector $t \in \mathbb{R}^3$. Given a depth image $D \in \mathbb{R}^{W\times H}$ where W and H denote the image width and height, and the camera intrinsic matrix $K \in \mathbb{R}^{3\times 3}$, we can acquire the corresponding point cloud $P \in \mathbb{R}^{(W\times H)\times 3}$ by back-projecting homogeneous pixels $I \in \mathbb{R}^{(W\times H)\times 3}$ to 3D space:

$$P = D(x,y)I(K^{-1})^T. \tag{1}$$

3.1. Overview

At the top level, we show our SO(3)-Pose in Fig. 3. SO(3)-Pose follows the high-performing keypoint-based two-stage strategy [HSH*20, HHF*21]. A 2D encoder and an SO(3)-equivariant encoder are utilized for representation learning of the RGB image and point cloud (converted from the depth image), respectively. For better appearance and geometry representation learning, SO(3)-Pose not only absorbs the geometry knowledge of SO(3)-equivariance from the point cloud, but also realizes the cross-modal information communication. By the communication, the SO(3)-invariant features facilitate to learn more distinctive representations for segmenting objects with similar appearance from RGB channels; the SO(3)-equivariant features bridge RGB features to deduce the (missed) geometry for detecting keypoints of an object with the reflective surface from the depth channel. Finally, it utilizes the 3D-3D correspondence of keypoints to fit the 6D pose parameters.

3.2. SO(3)-equivariant Layer

SO(3) can be interpreted as a 3×3 rotation matrix. A feature V is SO(3)-equivariant if and only if it satisfies:

$$f(VR) = f(V)R, \tag{2}$$

where $R \in SO(3)$ is a rotation matrix, f is a mapping function to represent a layer operation, and $V = \{v_i \in \mathbb{R}^{C\times 3}, i = 1,2,3,...\}$ (C is the feature dimension). Inspired by VNN [DLD*21], our method learns the SO(3)-equivariance properties as follows.

Learning the rotation equivariance should build a series of basic SO(3)-equivariant layers containing the linear layers, non-linear layers, pooling layers, and normalization layers. These layers all satisfy the SO(3)-equivariance property according to the definition.

Given a weight matrix $\mathbf{W} \in \mathbb{R}^{C'\times C}$, we define a linear operation $f_{\text{lin}}(\cdot;\mathbf{W})$ acting on a vector-list features $\mathbf{V} \in \mathbb{R}^{N\times C\times 3}$ as:

$$V' = f_{\text{lin}}(V;\mathbf{W}) = \mathbf{W}V \in \mathbb{R}^{C'\times 3}. \tag{3}$$

We verify that if the input rotates by $R \in SO(3)$, the output also rotates by the same matrix:

$$f_{\text{lin}}(VR;\mathbf{W}) = \mathbf{W}VR = f_{\text{lin}}(V;\mathbf{W})R = V'R, \tag{4}$$

yielding the desired equivariance property.

SO(3) has the closure property of a group. That means, the product of a rotation matrix multiplied by any rotation matrix is also a rotation matrix. If the final output is SO(3)-equivariant with the input across a neural network, any intermediate output also needs to satisfy the SO(3)-equivariance with the input. Thus, it is necessary to construct a special SO(3)-equivariance layer to achieve the same functions as the non-linear layers, pooling layers and normalization layers.

Non-linearity plays an important role in the representation learning of neural networks. The non-linear layers (e.g., ReLU, leaky-ReLU) split a feature space into two half-spaces: the positive half-space keeps its original feature and the negative half-space is muted or reduced by multiplication with a small weight. To keep SO(3)-equivariance, we dynamically predict a direction from the input vector-list feature and then truncate the portion of a vector that points into the negative half-space of the learned direction:

$$v' = \begin{cases} q & \text{if } \langle q,k\rangle \geqslant 0 \\ q - \left\langle q, \frac{k}{\|k\|}\right\rangle \frac{k}{\|k\|} & \text{otherwise} \end{cases} \tag{5}$$

where $q = \mathbf{W}V$, $k = \mathbf{U}V$, $\mathbf{W} \in \mathbb{R}^{1\times C}$, $\mathbf{U} \in \mathbb{R}^{1\times C}$ and $V \in \mathbb{R}^{C\times 3}$.

The pooling layer aims to downsample the feature. It also acts as a symmetrical function in the point cloud networks to solve the permutation problem of the point cloud. There are usually two main solutions (i.e., max-pooling and mean-pooling) used in most of the neural networks. According to the definition (Eq. 2), it is satisfied with the average pooling operation. We adopt the mean pooling to aggregate all feature information.

The normalization layer influences the convergence efficiency in the training process. Layer normalization [BKH16] and instance normalization [UVL16] only change the distributions in a sample of a batch with respect to a rotation. Batch normalization [IS15] aggregates statistics across all batch samples with respect to several rotations. Averaging across arbitrarily rotated inputs would not necessarily be useful. For example, averaging two input features rotated in opposite directions would be zero instead of producing that feature into a canonical pose. Thus, the operation of batch normalization is required to be specially designed, which cannot break the SO(3)-equivariance of features. We follow the design of VN-Batchnorm [DLD*21] that is formulated as:

$$N_b = \text{ElementwiseNorm } (V_b) \in \mathbb{R}^{N\times 1}, \tag{6}$$

$$\{N'_b\}_{b=1}^{B} = \text{BatchNorm } \left(\{N_b\}_{b=1}^{B}\right), \tag{7}$$

$$V'_b[c] = V_b[c]\frac{N'_b[c]}{N_b[c]}, \quad \forall c \in [C], \tag{8}$$

where $\{V_b\}_{b=1}^{B}$ are a batch of B vector-list features $V_b \in \mathbb{R}^{C\times 3}$, $V'_b[c]$, $V_b[c]$ are the vector channels, $V'_b[c]$, $V_b[c]$ are their scalar 2-norms, and ElementwiseNorm (V_b) computes the 2-norm of every vector channel $v_c = V_b[c] \in V_b$.

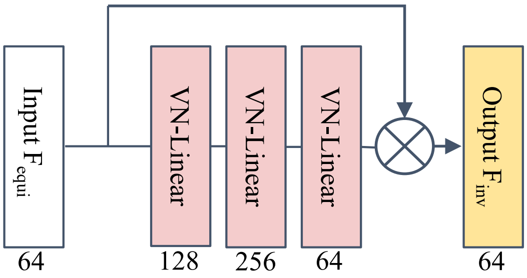


Figure 4: *The structure of E2Ilayer. Red box indicates the SO(3)-equivariant linear layer introduced in Section 3.2.*

3.3. SO(3)-invariant Layer

SO(3)-invariant property is crucial for many tasks such as classification and segmentation, where the label of an object or its parts should be invariant to the object's pose. We define an SO(3)-equivariant feature as:

$$f(VR) = f(V). \tag{9}$$

The rotation matrix is an orthonormal matrix; its inverse matrix is equal to its transpose matrix. We can compute an SO(3)-invariant feature from two SO(3)-equivariant features:

$$(VR)(TR)^\top = VRR^\top T^\top = VT^\top. \tag{10}$$

We propose an E2ILayer (an Equivariant-to-Invariant layer) to learn the SO(3)-invariant feature as illustrated in Fig. 4. We attain the SO(3)-invariant features to map the feature space onto the semantic label space. We design a series of linear layers to increase the feature expression ability.

3.4. Feature Fusion

Feature fusion is used to map the SO(3)-equivariant features to the keypoint offset space and map the SO(3)-invariant features to the semantic label space. From our observation, the semantic label is invariant when the object rotates, and the relative orientation is changed along with the rotation. From this discovery, we extract the SO(3)-invariant and SO(3)-equivariant features from the point cloud. We use a general 2D extractor from RGB images to contain local and global appearance features, then we fuse the appearance and SO(3)-equivariant features to map the feature space onto the keypoint offset space. We fuse the appearance features and SO(3)-invariant features to map the feature space onto the semantic space. Specifically, we first concatenate the geometric feature and appearance feature, and feed the fused features to MLP to generate the mixed feature space for final prediction. To avoid overfitting, we only use two linear layers.

3.5. 6D Object Pose Estimation

Once each point is assigned with a semantic label and an offset relative to the keypoints, we can start the second stage of pose estimation.

In detail, we first attain the object instance according to the per-point label and center point which is one of the predicted keypoints. Next, we cluster the keypoints to vote for certain keypoints corresponding to the keypoints selected from the object models. Finally, we utilize a least-squares fitting algorithm based on the 3D-3D corresponding relation to compute the pose parameters.

Let L_{seg} and L_{kp} be the two losses that are respectively imposed on the segmentation branch and keypoint detection branch. For the specific center point, we additionally use L_{center}. L_{seg} is a focal loss [LGG*17]. L_{kp} and L_{center} are L_1 distance losses following PVN3D [HSH*20]. Features extracted from the SO(3)-equivariant layers have the transitive dependencies property, in that each former feature rotates by R is equivalent to all latter features rotated by the same matrix R. In order to improve the expression of the SO(3)-equivariant property, we propose a new loss function, dubbed SO3loss, for the SO(3)-equivariant layers:

$$L_{\text{so3}} = \left| f(V) - f(VR)R^{\text{-1}} \right|, \tag{11}$$

where f represents a mapping through a series of SO(3)-equivariant layers, R is any rotation matrix, and V is a former feature. We jointly optimize the detection task and the auxiliary tasks by applying a gradient descent method to minimize the weighted sum of the following losses:

$$L_{\text{all}} = \lambda_1 L_{\text{seg}} + \lambda_2 L_{\text{kp}} + \lambda_3 L_{\text{center}} + \lambda_4 L_{\text{so3}}\ , \tag{12}$$

where λ_1, λ_2, λ_3 and λ_4 are the weights for different tasks.

4. Experiments

4.1. DataSets

We evaluate our method on two benchmark datasets, including the YCB-Video dataset and the LineMOD dataset.

YCB-Video dataset. YCB-Video [XSNF17] consists of 21 YCB objects [CSW*15] in 92 RGB-D videos. All the subsets of the objects that appeared in the scene are annotated with 6D pose and instance-level masks. We follow the previous works [XSNF17, WXZ*19, HSH*20] to split the training set and the testing set. We also take the synthetic images for training as per [XSNF17] and apply the hole completion algorithm to fill the depth images as per [KHW18].

LineMOD dataset. LineMOD [HHC*11] contains 13 low-textured objects in 13 videos, with annotated 6D pose and instance-semantic masks. The varying lighting, texture-less objects and cluttered scenes make this dataset challenging. We follow prior works [XSNF17, WXZ*19] to split the training set and the testing set, and we also obtain synthesis images for the training set following [PLH*19, HSH*20].

Occlusion LineMOD dataset. Occlusion LineMOD [BKM*14] is created by additionally annotating a subset of the LineMOD dataset. It contains 8 objects out of 13 LineMOD objects. Heavy occlusion in the objects makes the dataset challenging.

4.2. Metrics

We evaluate our method with the average distance metrics ADD [HHC*11] and ADD-S [XSNF17]. For non-symmetric objects, the ADD metric calculates the point-pair average distance between object model vertices transformed by the ground truth pose $[R^*, t^*]$ and the predicted pose $[R, t]$:

$$\text{ADD} = \frac{1}{m} \sum_{x \in \mathcal{O}} \left\| (R^* x + t^*) - (Rx + t) \right\|, \tag{13}$$

where x denotes a vertex in the object model $\mathcal{O}$, m is the number of vertices. For symmetric objects, the ADD-S metric calculates the mean distance based on the closest point distance:

$$\text{ADD-S} = \frac{1}{m} \sum_{x_1 \in \mathcal{O}} \min_{x_2 \in \mathcal{O}} \left\| (R^* x_1 + t^*) - (Rx_2 + t) \right\|. \tag{14}$$

In the YCB-Video dataset, we follow prior works [XSNF17, WXZ*19, HHF*21] to report the area under the ADDS and ADD(-S) [HLI*12] curve (AUC) and set the maximum threshold of AUC to be 0.1m. The ADD(-S) calculates ADD for non-symmetric objects and ADDS for symmetric objects. In the LineMOD dataset, we follow [PLH*19] to use the ADD(-S)-0.1d which indicates that the estimated pose is correct when the ADD(-S) distance is less than 10% of the model's diameter.

4.3. Implementation Details

We train our model on the original YCB-Video dataset and the additional rendering LineMOD dataset following FFB6D [HHF*21]. We select the PSPNet [ZSQ*17] as the 2D feature extractor which is widely used in the field of image segmentation. We design the point cloud feature extractor with the component presented by VNN [DLD*21]. In the training stage, we set the loss function weights $\lambda_1, \lambda_2, \lambda_3$ and λ_4 as 1.0, 1.0, 1.0, 0.5, respectively. We train 20 epochs in the YCB-Video dataset and 10 epochs in the LineMOD dataset for single objects.

4.4. Ablation Studies

	B	E	E+F	E+F+S
ADDS	93.1	93.6	94.1	94.4
ADD(-S)	87.9	88.8	90.0	90.1

Table 1: *Ablation studies on different configurations in terms of the ADDS metric and ADD(-S) metric. "B" is the baseline setting which uses the PointNet as the point cloud encoder. "E" only uses the SO(3)-equivariant layers. "E+F" adds the SO(3)-invariant layers and the feature fusion strategy, "E+F+S" further introduces the SO3loss.*

We conduct ablation studies to evaluate the effect of the feature fusion strategy and the SO(3) loss on the YCB-Video dataset. Table 1 summarizes the results of ablation studies on the YCB-video dataset.

The column "E" is based on the SO(3)-equivariance layers and uses the concatenation of SO(3)-equivariant features and appearance features to relate both the semantic label and keypoint offset. Column "B" uses the PointNet as the point cloud encoder and other settings are the same as the column "E". Column "E+F" denotes the base SO(3)-equivariance layer and an additional feature fusion

	PoseCNN		PoseCNN+ICP		DF(per-pixel)		DF(iterative)		FFB6D(trained)		Ours		Ours+ICP	
	ADDS	ADD(-S)	ADDS	ADD(-S)	ADDS	ADD(-S)	ADDS	ADD(-S)	ADDS	ADD(-S)	ADDS	ADD(-S)	ADDS	ADD(-S)
002_master_chef_can	83.9	50.2	95.8	68.1	95.3	70.7	**96.4**	73.2	95.7	78.6	95.5	82.2	95.5	**82.8**
003_cracker_box	76.9	53.1	92.7	83.4	92.5	86.9	**95.8**	**94.1**	94.4	90.0	94.3	90.6	91.1	85.1
004_sugar_box	84.2	68.4	**98.2**	**97.1**	95.1	90.8	95.8	94.1	96.3	93.4	96.5	94.2	97.3	96.3
005_tomato_soup_can	81.0	66.2	94.5	81.8	93.8	84.7	94.5	85.5	94.3	83.1	95.4	89.5	**95.7**	**89.9**
006_mustard_bottle	90.4	81.0	**98.6**	**98.0**	95.8	90.9	97.3	94.7	96.7	94.4	96.7	94.6	97.8	97.0
007_tuna_fish_can	88.0	70.7	97.1	83.9	95.7	79.6	97.1	81.9	95.5	83.1	95.8	86.6	**97.1**	**87.8**
008_pudding_box	79.1	62.7	**97.9**	**96.6**	94.3	89.3	96.0	93.3	95.5	91.6	96.8	94.2	97.5	96.1
009_gelatin_box	87.2	75.2	**98.8**	**98.1**	97.2	95.8	98.0	96.7	96.8	93.5	97.3	94.7	98.5	97.8
010_potted_meat_can	78.5	59.5	**92.7**	83.5	89.3	79.6	90.7	83.6	89.7	82.9	92.4	**84.5**	91.6	80.9
011_banana	86.0	72.3	97.1	91.9	90.0	76.7	96.2	83.3	96.8	94.0	95.8	91.3	**97.5**	**95.1**
019_pitcher_base	77.0	53.3	**97.8**	96.9	93.6	87.1	97.5	**96.9**	95.5	91.4	95.6	91.7	97.1	95.8
021_bleach_cleanser	71.6	50.3	96.9	92.5	94.4	87.5	95.9	89.9	95.4	90.6	95.1	91.4	**96.9**	**94.9**
024_bowl	69.6	69.6	81.0	81.0	86.0	86.0	**89.5**	**89.5**	86.2	86.2	88.4	88.4	87.8	87.8
025_mug	78.2	58.5	94.9	81.1	95.3	83.8	96.7	88.9	97.0	91.3	96.7	91.1	**97.7**	**94.8**
035_power_drill	72.7	55.3	**98.2**	**97.7**	92.1	83.7	96.0	92.7	95.9	93.1	95.9	93.2	96.6	94.8
036_wood_block	64.3	64.3	87.6	87.6	89.5	89.5	**92.8**	**92.8**	91.5	91.5	86.6	86.6	91.6	91.6
037_scissors	56.9	35.8	91.7	78.4	90.1	77.4	92.0	77.9	**94.1**	**83.0**	91.8	81.0	88.5	73.1
040_large_marker	71.7	58.3	97.2	85.3	95.1	89.1	97.6	93.0	94.7	85.2	94.4	86.9	**97.6**	**88.3**
051_large_clamp	50.2	50.2	75.2	75.2	71.5	71.5	72.5	72.5	91.0	91.0	94.2	94.2	**95.8**	**95.8**
052_extra_large_clamp	44.1	44.1	64.4	64.4	70.2	70.2	69.9	69.9	91.2	91.2	91.2	91.2	**91.7**	**91.7**
061_foam_brick	88.0	88.0	**97.2**	**97.2**	92.2	92.2	92.0	92.0	93.8	93.8	95.0	95.0	97.0	97.0
average	75.8	59.9	93.0	85.4	91.2	82.9	93.2	86.1	94.2	89.1	94.4	90.1	**95.1**	**91.1**

Table 2: *Quantitative evaluation of ADDS AUC and ADD(-S) AUC metrics on the YCB-Video dataset. Objects in bold are considered as the symmetric objects. FFB6D(trained) indicates the model is trained under its original settings [HHF*21].*

	RGB				RGB-D						
	PoseCNN+ DeepIM	PVNet	CDPN	DPOD	Point-Fusion	Dense-Fusion	G2LNet	PVN3D	SS-Conv	FFB6D	Ours
ape	77.0	43.6	64.4	87.7	70.4	92.3	96.8	97.3	97.4	98.4	**98.6**
benchvise	97.5	99.9	97.8	98.5	80.7	93.2	96.1	99.7	99.3	**100.0**	**100.0**
camera	93.5	86.9	91.7	96.1	60.8	94.4	98.2	99.6	99.5	**99.9**	**99.9**
can	96.5	95.5	95.9	99.7	61.1	93.1	98.0	99.5	99.6	99.8	**100.0**
cat	82.1	79.3	83.8	94.7	79.1	96.5	99.2	99.8	99.8	99.9	**100.0**
driller	95.0	96.4	96.2	98.8	47.3	87.0	99.8	99.3	99.6	100.0	**100.0**
duck	77.7	52.6	66.8	86.3	63.0	92.3	97.7	98.2	97.8	98.4	**98.5**
eggbox	97.1	99.2	99.7	99.9	99.9	99.8	**100.0**	99.8	99.9	**100.0**	**100.0**
glue	99.4	95.7	99.6	96.8	99.3	**100.0**	**100.0**	**100.0**	99.6	**100.0**	**100.0**
holepuncher	52.8	82.0	85.8	86.9	71.8	92.1	99.0	99.9	99.4	99.8	**100.0**
iron	98.3	98.9	97.9	**100.0**	83.2	97.0	99.3	99.7	99.2	99.9	**100.0**
lamp	97.5	99.3	97.9	96.8	62.3	95.3	99.5	99.8	99.7	99.9	**100.0**
phone	87.7	92.4	90.8	94.7	78.8	92.8	98.9	99.5	98.2	99.7	**99.9**
MEAN	88.6	86.3	89.9	95.2	73.7	94.3	98.7	99.4	99.2	99.7	**99.8**

Table 3: *Quantitative evaluation using the ADD(-S)-0.1d metric on the LineMOD dataset. Objects with bold name are symmetric.*

strategy. Column “E+F+S” shows the results of further adding the SO(3) loss. The 2D encoder is the same for all models.

To validate the feature fusion strategy, we compare the column “E+F” with the column “E”. The results show that the feature fusion strategy can learn a better mapping from the feature space to the semantic space and keypoint offset space, thus increasing the accuracy of pose estimation.

To analyze the SO(3)loss, we compare the pose estimation results based on the “E+F” model. The results in Column “E+F+S” demonstrate that adding the SO(3)loss further improves the accuracy of pose estimation.

4.5. Comparisons

We compare our method with the state-of-the-art methods which take RGB or RGB-D as input and output the 6D object pose.

Performance on the YCB-Video dataset. In Table 2, we compare our method with PoseCNN [XSNF17], DenseFusion [WXZ*19] and FFB6D [HHF*21] on the YCB-Video dataset in terms of the ADDS AUC metric and ADD(-S) AUC metric. Our method achieves competitive performance with its competitors. [XSNF17, WXZ*19] directly regress the rotation matrix and translation vector, while our two-stage method first predicts the keypoint localization and then fits the 6D pose parameters. The results show that our method performs well for the large clamp and the extra-large clamp which are difficult to detect because of their symmetry and similarity. Comparing with other methods with refinement pro-

Method	PoseCNN	Pix2Pose	PVNet	DPOD	Hu et al.	HybridPose	PVN3D	FFB6D	Ours
ape	9.6	22.0	15.8	-	19.2	20.9	33.9	47.2	**49.7**
can	45.2	44.7	63.3	-	65.1	75.3	88.6	85.2	**88.8**
cat	0.9	22.7	16.7	-	18.9	24.9	39.1	45.7	**50.9**
driller	41.4	44.7	65.7	-	69.0	70.2	78.4	81.4	**88.6**
duck	19.6	15.0	25.2	-	25.3	27.9	41.9	53.9	**58.1**
eggbox	22.0	25.2	50.2	-	52.0	52.4	**80.9**	70.2	60.8
glue	38.5	32.4	49.6	-	51.4	53.8	68.1	60.1	**71.0**
holepuncher	22.1	49.5	39.7	-	45.6	54.2	74.7	**85.9**	82.4
MEAN	24.9	32.0	40.8	47.3	43.3	47.5	63.2	66.2	**68.4**

Table 4: *Quantitative evaluation using the ADD(-S)-0.1d metric on the Occlusion LineMOD dataset. Hu et al. means the paper [HFWS20]. Symmetric objects' names are in bold.*

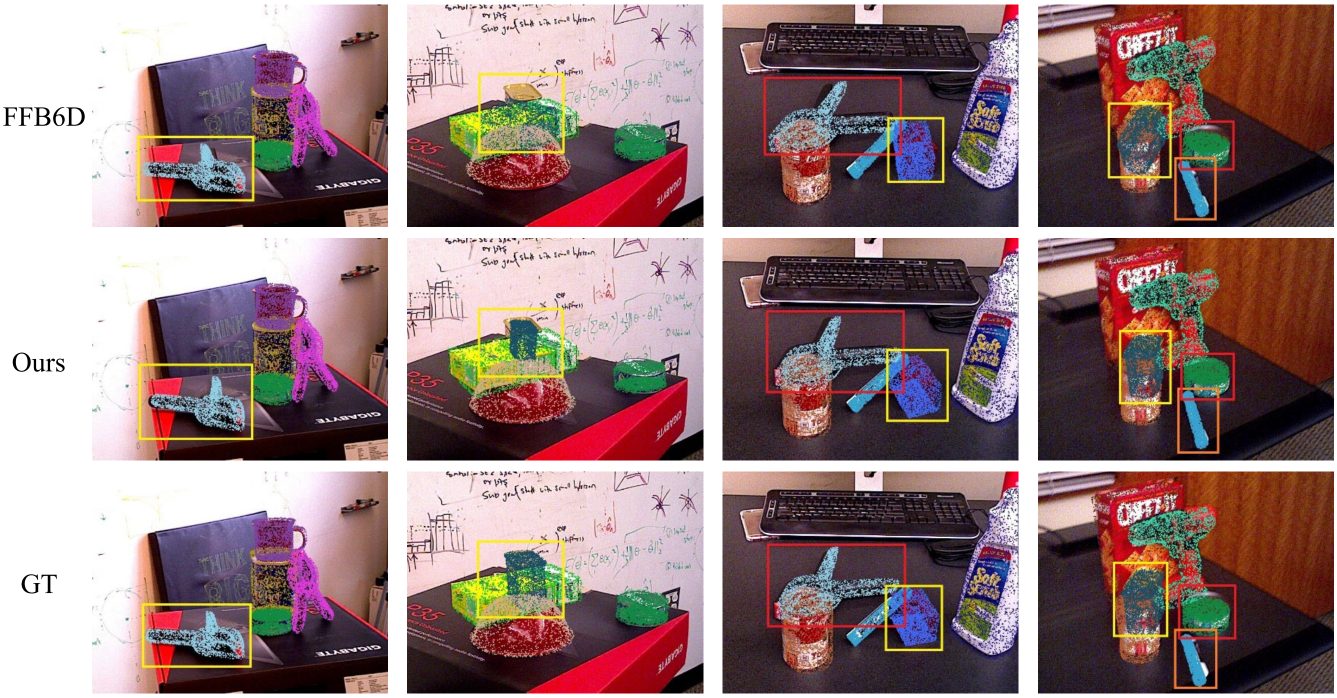


Figure 5: *Visual Comparison with FFB6D [HHF*21] on YCB-Video. Different objects in the same scene are in different colors. The points are projected back to the image after being transformed by the predicted pose.*

cedure, our model with ICP outperforms PoseCNN+ICP by 6.7% and exceeds DF (iterative) by 5.8% on the ADD(-S) metric.

Performance on the LineMOD dataset. In Table 3, we compare our method with PoseCNN [XSNF17] + DeepIM [LWJ*18], PointFusion [XAJ17], PVNet [PLH*19], DenseFusion [WXZ*19], G2LNet [CJC*20], PVN3D [HSH*20], FFB6D [HHF*21] and SS-Conv [LLC*21] on the LineMOD dataset in terms of the ADD(-S)-0.1d metrics. Our method achieves the state-of-the-art performance. [XSNF17, PLH*19] are based on the RGB images, and the other methods are based on the RGB-D images. G2LNet [CJC*20] first learns the global feature, and then learns the local feature. SS-Conv [LLC*21] learns the SE(3)-equivariant feature to relate the rotation space and the translation space. The results show that our design of learning SO(3)-equivariant features outperforms its competitors. On the ADD(-S)-0.1d metric, our model achieves the state-of-the-art at all objects.

Performance on the Occlusion LineMOD dataset. We use the model trained on the LineMOD dataset for testing on the Occlusion LineMOD dataset. Table 4 shows the comparison of our method with PoseCNN [XSNF17], Pix2Pose [PPV19], PVNet [PLH*19], DPOD [ZSI19], [HFWS20], HybridPose [SSH20], PVN3D [HSH*20] and FFB6D [HHF*21] in terms of the ADD(-

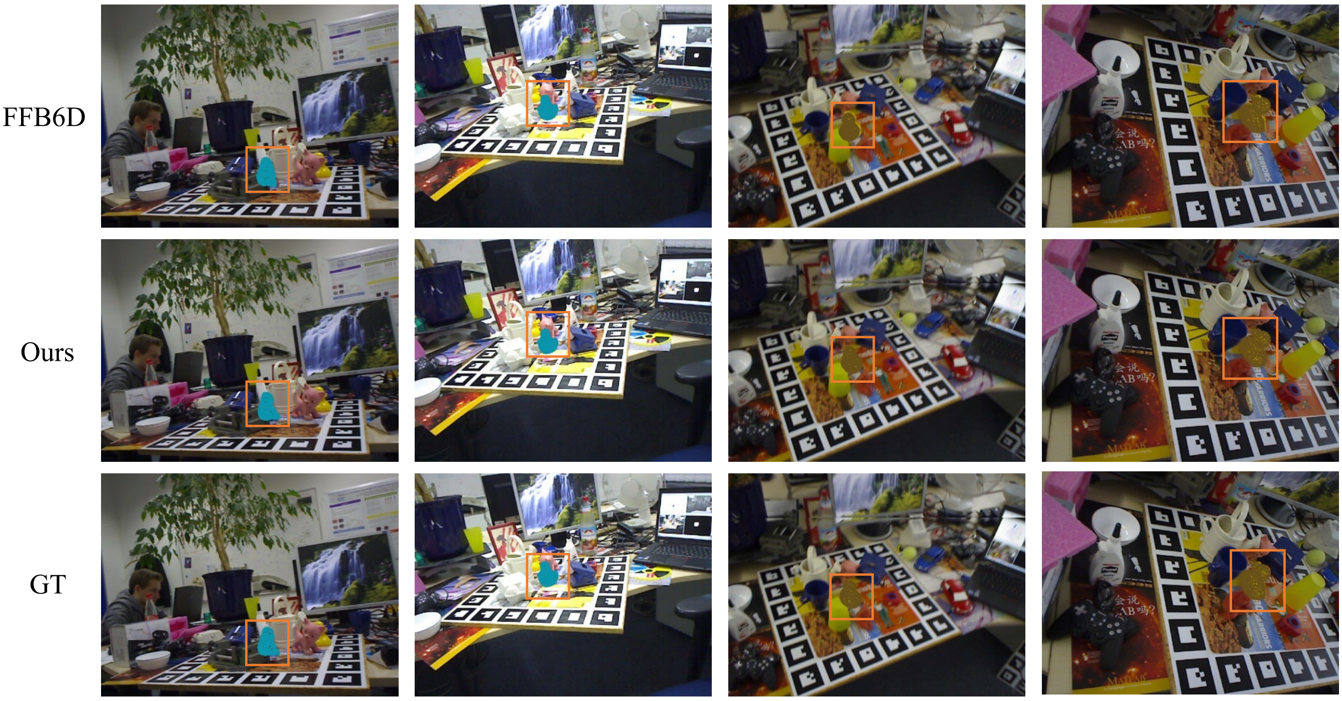


Figure 6: *Visual Comparison with FFB6D [HHF*21]. We select two challenging objects (ape boxed in blue and duck boxed in orange) from LineMOD.*

S)-0.1d metric. Our method achieves the state-of-the-art performance. Although the results of eggbox and holepuncher are lower than PVN3D and FFB6D, the results of other objects achieve the best performance among all methods. Our model exceeds FFB6D by 3.3%. And our method outperforms FFB6D by a margin of 11.4% in the ADD(-S)-0.1d metric of the cat. The improved performance demonstrates that the proposed method is highly robust to occlusion.

As for FFB6D [HHF*21], we re-train its source code under the same settings suggested by the original authors for fair comparisons. We can see that FFB6D performs very well, ranking the second place.

4.6. Visualization

We demonstrate some visualization results from the YCB-Video dataset and the LineMOD dataset in Fig. 5 and Fig. 6, respectively. We find in Fig. 5 that the ground truth of the object boxed in orange (see the right-bottom image) is not accurate, since the YCB-Video dataset is sampled from a fast-moving video, which produces some motion-blurred frames. The results show our method outperforms its competitor, even in the occlusion scenes, e.g., the object in the yellow box (see Columns 2 and 4 of Fig. 5).

5. Conclusion

In this paper, we proposed a novel method, called SO(3)-Pose, for 6D instance-level object pose estimation. We verify that leveraging SO(3)-equivariance features to learn the keypoint offset and leveraging SO(3)-invariance features to learn the semantic label are beneficial to the pose estimation task. We propose a new loss function, namely SO3loss, to facilitate the training of the proposed network. Extensive experiments on two standard pose estimation datasets demonstrate the effectiveness of our proposed method, and show that it outperforms state-of-the-art methods. However, our method is limited by the simple MLP structure. In future, we will explore the powerful structures to satisfy the SO(3)-equivariance property, such as transformer.

Acknowledgments

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (No. UGC/FDS16/E14/21), the National Natural Science Foundation of China (No. 62172218, No. 62032011), the Free Exploration of Basic Research Project, Local Science and Technology Development Fund Guided by the Central Government of China (No. 2021Szvup060), and the Natural Science Foundation of Guangdong Province (No. 2022A1515010170).

References

[ATC*21] Ali A., Touvron H., Caron M., Bojanowski P., Douze M., Joulin A., Laptev I., Neverova N., Synnaeve G., Verbeek J., et al.: Xcit: Cross-covariance image transformers. *Advances in neural information processing systems 34* (2021). 3

[BKH16] Ba J., Kiros J. R., Hinton G. E.: Layer normalization. *arXiv: Machine Learning* (2016). 5

[BKM*14] Brachmann E., Krull A., Michel F., Gumhold S., Shotton J., Rother C.: Learning 6d object pose estimation using 3d object coordinates. *european conference on computer vision* (2014). 6

[CJC*20] Chen W., Jia X., Chang H. J., Duan J., Leonardis A.: G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. *computer vision and pattern recognition* (2020). 8

[CSW*15] Calli B., Singh A., Walsman A., Srinivasa S. S., Abbeel P., Dollar A. M.: The ycb object and model set: Towards common benchmarks for manipulation research. *international conference on advanced robotics* (2015). 6

[CTT*12] Choi C., Taguchi Y., Tuzel O., Liu M.-Y., Ramalingam S.: Voting-based pose estimation for robotic assembly using a 3d sensor. In *2012 IEEE International Conference on Robotics and Automation* (2012), IEEE, pp. 1724–1731. 1

[DCLC21] Dai Z., Cai B., Lin Y., Chen J.: Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 1601–1610. 3

[DLD*21] Deng C., Litany O., Duan Y., Poulenard A., Tagliasacchi A., Guibas L. J.: Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12200–12209. 4, 5, 6

[DUNI10] Drost B., Ulrich M., Navab N., Ilic S.: Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition* (2010), Ieee, pp. 998–1005. 3

[DWL19] Du G., Wang K., Lian S.: Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review. 3

[DWLZ21] Du G., Wang K., Lian S., Zhao K.: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review 54*, 3 (2021), 1677–1734. 1

[GPH19] Gupta K., Petersson L., Hartley R.: Cullnet: Calibrated and pose aware confidence scores for object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019), pp. 0–0. 3

[HCI*11] Hinterstoisser S., Cagniart C., Ilic S., Sturm P., Navab N., Fua P., Lepetit V.: Gradient response maps for real-time detection of textureless objects. *IEEE transactions on pattern analysis and machine intelligence 34*, 5 (2011), 876–888. 3

[HFWS20] Hu Y., Fua P., Wang W., Salzmann M.: Single-stage 6d object pose estimation. *computer vision and pattern recognition* (2020). 8

[HHC*11] Hinterstoisser S., Holzer S. J. J., Cagniart C., Ilic S., Konolige K., Navab N., Lepetit V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. *international conference on computer vision* (2011). 6

[HHF*21] He Y., Huang H., Fan H., Chen Q., Sun J.: Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 3003–3013. 2, 3, 4, 6, 7, 8, 9

[HHFS19] Hu Y., Hugonot J., Fua P., Salzmann M.: Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 3385–3394. 3

[HJX*18] Hana X.-F., Jin J. S., Xie J., Wang M.-J., Jiang W.: A comprehensive review of 3d point cloud descriptors. *arXiv preprint arXiv:1802.02297 2* (2018). 3

[HLI*12] Hinterstoisser S., Lepetit V., Ilic S., Holzer S., Bradski G., Konolige K., Navab N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision* (2012), Springer, pp. 548–562. 3, 6

[HLRK16] Hinterstoisser S., Lepetit V., Rajkumar N., Konolige K.: Going further with point pair features. In *European conference on computer vision* (2016), Springer, pp. 834–848. 3

[HMZA21] Huang X., Mei G., Zhang J., Abbas R.: A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690* (2021). 3

[HSH*20] He Y., Sun W., Huang H., Liu J., Fan H., Sun J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 11632–11641. 2, 3, 4, 6, 8

[HZL*15] Hodaň T., Zabulis X., Lourakis M., Obdržálek Š., Matas J.: Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), IEEE, pp. 4421–4428. 3

[IS15] Ioffe S., Szegedy C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *international conference on machine learning* (2015). 5

[JMX*21] Jiang X., Ma J., Xiao G., Shao Z., Guo X.: A review of multimodal image matching: Methods and applications. *Information Fusion 73* (2021), 22–71. 3

[KHW18] Ku J., Harakeh A., Waslander S. L.: In defense of classical image processing: Fast depth completion on the cpu. *canadian conference on computer and robot vision* (2018). 6

[KMT*17] Kehl W., Manhardt F., Tombari F., Ilic S., Navab N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1521–1529. 3

[LGG*17] Lin T.-Y., Goyal P., Girshick R., He K., Dollár P.: Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017). 6

[LLC*21] Lin J., Li H., Chen K., Lu J., Jia K.: Sparse steerable convolutions: An efficient learning of se (3)-equivariant features for estimation and tracking of object poses in 3d space. *Advances in Neural Information Processing Systems 34* (2021). 2, 3, 8

[LMNF09] Lepetit V., Moreno-Noguer F., Fua P.: Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision 81*, 2 (2009), 155–166. 3

[LWJ*18] Li Y., Wang G., Ji X., Xiang Y., Fox D.: Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 683–698. 3, 8

[LWY*21] Li X., Weng Y., Yi L., Guibas L. J., Abbott A., Song S., Wang H.: Leveraging se (3) equivariance for self-supervised category-level object pose estimation from point clouds. *Advances in Neural Information Processing Systems 34* (2021). 4

[LWZ21] Li L., Wang R., Zhang X.: A tutorial review on point cloud registrations: Principle, classification, comparison, and technology challenges. *Mathematical Problems in Engineering 2021* (2021). 3

[MAMT15] Mur-Artal R., Montiel J. M. M., Tardos J. D.: Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics 31*, 5 (2015), 1147–1163. 3

[MJF*21] Ma J., Jiang X., Fan A., Jiang J., Yan J.: Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision 129*, 1 (2021), 23–79. 3

[PLH*19] Peng S., Liu Y., Huang Q., Zhou X., Bao H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4561–4570. 6, 8

[PPV19] Park K., Patten T., Vincze M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. *international conference on computer vision* (2019). 8

[PRPO19] Poulenard A., Rakotosaona M.-J., Ponty Y., Ovsjanikov M.: Effective rotation-invariant point cnn with spherical harmonics kernels. In *2019 International Conference on 3D Vision (3DV)* (2019), IEEE, pp. 47–56. 4

[PZL*20] Peng S., Zhou X., Liu Y., Lin H., Huang Q., Bao H.: Pvnet: pixel-wise voting network for 6dof object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). 1, 3

[RCT13] Rios-Cabrera R., Tuytelaars T.: Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the IEEE international conference on computer vision* (2013), pp. 2048–2055. 3

[RL17] Rad M., Lepetit V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 3828–3836. 3

[SHX*21] Shi Y., Huang J., Xu X., Zhang Y., Xu K.: Stablepose: Learning 6d object poses from geometrically stable patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 15222–15231. 3

[SMD*18] Sundermeyer M., Marton Z.-C., Durner M., Brucker M., Triebel R.: Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)* (2018), pp. 699–715. 3

[SSH20] Song C., Song J., Huang Q.: Hybridpose: 6d object pose estimation under hybrid representations. *computer vision and pattern recognition* (2020). 8

[STDS14] Salti S., Tombari F., Di Stefano L.: Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding 125* (2014), 251–264. 3

[TSF18] Tekin B., Sinha S. N., Fua P.: Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 292–301. 3

[UVL16] Ulyanov D., Vedaldi A., Lempitsky V.: Instance normalization: The missing ingredient for fast stylization. *arXiv: Computer Vision and Pattern Recognition* (2016). 5

[VLLM18] Vidal J., Lin C.-Y., Lladó X., Martí R.: A method for 6d pose estimation of free-form rigid objects using point pair features on range data. *Sensors 18*, 8 (2018), 2678. 3

[WXZ*19] Wang C., Xu D., Zhu Y., Martín-Martín R., Lu C., Fei-Fei L., Savarese S.: Densefusion: 6d object pose estimation by iterative dense fusion. *computer vision and pattern recognition* (2019). 2, 3, 6, 7, 8

[WZA*21] Wang H., Zhu Y., Adam H., Yuille A., Chen L.-C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 5463–5474. 3

[XAJ17] Xu D., Anguelov D., Jain A.: Pointfusion: Deep sensor fusion for 3d bounding box estimation. *computer vision and pattern recognition* (2017). 8

[XSNF17] Xiang Y., Schmidt T., Narayanan V., Fox D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *robotics science and systems* (2017). 3, 6, 7, 8

[ZC17] Zhang H., Cao Q.: Texture-less object detection and 6d pose estimation in rgb-d images. *Robotics and Autonomous Systems 95* (2017), 64–79. 3

[ZLL*20] Zhou J., Liu Y., Liu J., Xie Q., Zhang Y., Zhu X., Ding X.: Bold3d: A 3d bold descriptor for 6dof pose estimation. *Computers & Graphics 89* (2020), 94–104. 3

[ZSI19] Zakharov S., Shugurov I., Ilic S.: Dpod: 6d pose object detector and refiner. *international conference on computer vision* (2019). 8

[ZSQ*17] Zhao H., Shi J., Qi X., Wang X., Jia J.: Pyramid scene parsing network. *computer vision and pattern recognition* (2017). 6