IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

# SyMFM6D: Symmetry-aware Multi-directional Fusion for Multi-View 6D Object Pose Estimation

Fabian Duffhauss[1,2], Sebastian Koch[1,3], Hanna Ziesche[1], Ngo Anh Vien[1], and Gerhard Neumann[4]

***Abstract*—Detecting objects and estimating their 6D poses is essential for automated systems to interact safely with the environment. Most 6D pose estimators, however, rely on a single camera frame and suffer from occlusions and ambiguities due to object symmetries. We overcome this issue by presenting a novel symmetry-aware multi-view 6D pose estimator called SyMFM6D. Our approach fuses the RGB-D frames from multiple perspectives in a deep multi-directional fusion network and predicts predefined keypoints for all objects in the scene simultaneously. Based on the keypoints and an instance semantic segmentation, we efficiently compute the 6D poses by least-squares fitting. To address the ambiguity issues for symmetric objects, we propose a novel training procedure for symmetry-aware keypoint detection including a new objective function. Our SyMFM6D network significantly outperforms the state-of-the-art in both single-view and multi-view 6D pose estimation. We furthermore show the effectiveness of our symmetry-aware training procedure and demonstrate that our approach is robust towards inaccurate camera calibration and dynamic camera setups.**

***Index Terms*—Deep Learning for Visual Perception, RGB-D Perception, Data Sets for Robotic Vision, 6D Pose Estimation.**

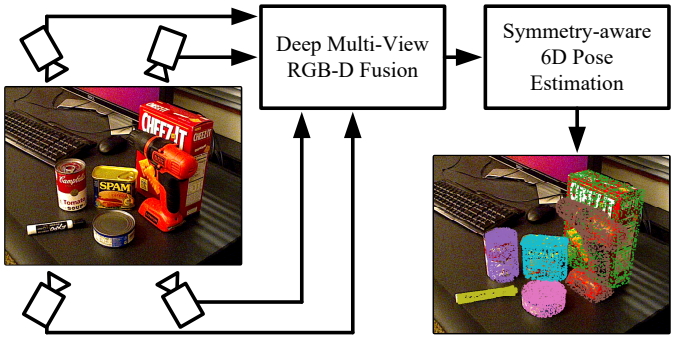



Fig. 1: Overview of our proposed SyMFM6D approach. We present a novel deep multi-directional fusion network which merges RGB-D data from multiple cameras. SyMFM6D predicts the 6D poses of all objects in the scene while considering object symmetries. It copes with very cluttered scenes and outperforms the state-of-the-art in single-view and multi-view 6D pose estimation.

## I. Introduction

ESTIMATING the 6D poses of objects is an essential computer vision task which is widely used in robotics [1]–[3], automated driving [4], [5], and augmented reality [6], [7]. In recent years, 6D pose estimators have made significant progress based on deep neural network architectures which rely on a single RGB image [1], [8], [9], on a single point cloud [10], [11], or fuse both [2], [3], [12]. Single-view methods, however, have problems detecting objects which are occluded by other objects. These problems can be overcome by considering data from multiple perspectives. Fusing multi-view data can significantly improve the accuracy and robustness of environmental understanding in complex scenarios, which can enable more flexible production and assembly processes, among other applications. There are already a few methods that consider multi-view data [13]–[15] which are, however, computationally expensive and not designed for scenes with strong occlusions. Moreover, most methods suffer from symmetric objects as they have multiple 6D poses with same visual and geometric appearance, causing most learning-based estimators to average over these multiple solutions.

We present a novel **Sy**mmetry-aware **M**ulti-directional **F**usion approach for **M**ulti-view **6D** pose estimation called SyMFM6D which overcomes the previously mentioned issues. Fig. 1 shows an overview of our system. SyMFM6D exploits the visual and geometric information from an arbitrary number of RGB-D images depicting a scene from multiple perspectives. We propose a deep multi-directional fusion network which fuses the multi-view RGB-D data efficiently and learns a compact representation of the entire scene. Our approach predicts the 6D poses of all objects in the scene simultaneously based on keypoint detection, instance semantic segmentation, and least-squares fitting. Furthermore, we present a novel symmetry-aware training procedure including a novel objective function which significantly improves the keypoint detection.

Our experiments demonstrate a large benefit of the proposed symmetry-aware training procedure, improving the accuracy of both symmetric and non-symmetric objects due to synergy effects. Thus, our approach outperforms the state-of-the-art in single-view 6D pose estimation. SyMFM6D also outperforms the state-of-the-art multi-view approach while being computationally more efficient. We furthermore show that our approach works accurately in both fixed and dynamic camera settings. Moreover, our method is robust towards inaccurate camera calibration by compensating imprecise camera pose information when using multiple views.

Manuscript received: February 8, 2023; Revised May 13, 2023; Accepted June 22, 2023.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments.)

[1]Fabian Duffhauss, Sebastian Koch, Hanna Ziesche, and Ngo Anh Vien are with the Bosch Center for Artificial Intelligence, Renningen, Germany Fabian.Duffhauss@bosch.com

[2]Fabian Duffhauss is with the University of Tübingen, Tübingen, Germany

[3]Sebastian Koch is with the Ulm University, Ulm, Germany

[4]Gerhard Neumann is with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany Gerhard.Neumann@kit.edu

Source code, datasets, and implementation details are available at https://github.com/boschresearch/SyMFM6D.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Our main contributions are:

**(i)** We propose a novel multi-directional multi-view fusion network for efficient representation learning of multiple RGB-D frames and present a novel multi-view 6D pose estimation method based on it. **(ii)** We present a novel symmetry-aware training procedure for 3D keypoint detection based on a symmetry-aware objective function. **(iii)** We present a novel synthetic dataset with photorealistic multi-view RGB-D data and labels for 6D pose estimation as well as instance semantic segmentation. **(iv)** We demonstrate significant improvements and synergy effects due to our symmetry-aware training procedure on challenging datasets including symmetric and non-symmetric objects. **(v)** Our method outperforms the state-of-the-art in single-view and multi-view 6D object pose estimation. We further demonstrate the robustness of our approach towards inaccurate camera calibration and dynamic camera setups.

## II. Related Work

Over the last few years, there has been significant progress in the area of 6D pose estimation. We now discuss the most important milestones subdivided into single-view methods, multi-view methods, and symmetry-aware methods.

### A. Single-View 6D Pose Estimation

The methods in this family require only a single input modality, which can be RGB, point cloud, or RGB-D. Traditional pose estimators using a *single RGB image* are mostly feature-based [16]–[21] or based on template matching [22]–[25]. Especially the former group of methods are often multi-staged and first extract local features from the given RGB image before matching the 2D-3D-correspondences to estimate the object's pose using a Perspective-n-Point (PnP) algorithm [26]. End-to-end trainable neural networks directly predict object poses without requiring multiple stages [1], [8], [9], [27]–[32]. These methods share similar ideas to exploit differentiable PnP or differentiable rendering techniques.

The recent advance of LiDAR and depth sensors promoted the proposal of methods based on a *single point cloud* [33], [34]. These methods apply either 3D convolutions [35], [36], or variations of PointNet [37] as backbone [38]–[40]. The authors of [41] and [42] introduce and further improve voting techniques for 3D object detection. However, since point cloud based methods cannot extract texture information, their application range is limited.

In contrast, *RGB-D based approaches* can combine the advantages of both modalities. For instance, [4] and [43] fuse an RGB image with a LiDAR point cloud by applying networks for convolutional feature extraction and for generating 3D object proposals. The approaches proposed in [44], [12], and [2] separately process the RGB image by a CNN and the point cloud by a PointNet-based network before fusing the appearance features and the geometric features with a dense fusion network. In [2] and [3] the authors employ a deep Hough voting network for 3D keypoint detection before estimating 6D poses by least-squares fitting [45]. However, most previous methods do not consider object symmetries and suffer from strong occlusions.

### B. Multi-View 6D Pose Estimation

Multi-view pose estimators consider multiple RGB(-D) frames showing the same scene from different perspectives in order to reduce the effect of occlusions and to improve the 6D pose estimation accuracy. The approach proposed in [13] first segments all frames with a CNN before aligning the known 3D models with the segmented object point cloud to estimate their poses. The authors of [14] present an end-to-end trainable CNN-based architecture based on a single RGB or RGB-D image. They perform the single-view pose estimation multiple times with images from different viewpoints before selecting the best hypothesis using a voting score that suppresses outliers. In [15] the authors propose a three-stage approach which first employs a CNN for generating object candidate proposals for each view independently. Secondly, they conduct a candidate matching considering the predictions of all views before finally performing a refinement procedure based on object-level bundle adjustment [46]. The approach of [47] directly fuses the features from multiple RGB-D views before predicting the poses based on keypoints and least-squares fitting [45]. However, the method uses a computationally expensive feature extraction and fusion network which does not consider object symmetries and it is evaluated on only synthetic datasets.

### C. Symmetry-aware 6D Pose Estimation

Symmetric objects are known to be a challenge for 6D pose estimation approaches due to ambiguities [48]. Different techniques have been proposed to address this issue. The authors of [48] and [49] propose to utilize an additional output channel to classify the type of symmetry and its domain range. In [50], a loss is introduced that is the smallest error among symmetric pose proposals in a finite pool of symmetric poses. In [51] the authors propose to use compact surface fragments as a compositional way to represent objects. As a result, this representation can easily allow handling of symmetries. The authors of [52] employ an additional symmetry prediction as output, and an extra refining step of predicted symmetry via an optimization function. A novel output space representation for CNNs is presented in [53] where symmetrical equivalent poses are mapped to the same values. In [54] the authors introduce a compact shape representation based on grouped primitives to handle symmetries. However, non of these methods outperforms the keypoint-based methods [2] and [3], even though they do not consider object symmetries. In contrast, our method extends current keypoint based methods to consider object symmetries, and consequently outperforms all previous methods on single and multi-view scenes.

## III. Proposed Method: SyMFM6D

We propose a deep multi-directional fusion approach called SyMFM6D that estimates the 6D object poses of all objects in a cluttered scene based on multiple RGB-D images while considering object symmetries. In this section, we define the task of multi-view 6D object pose estimation and present our multi-view deep fusion architecture.

6D object pose estimation describes the task of predicting a rigid transformation $\boldsymbol{p} = [\boldsymbol{R}|\boldsymbol{t}] \in SE(3)$ which transforms the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

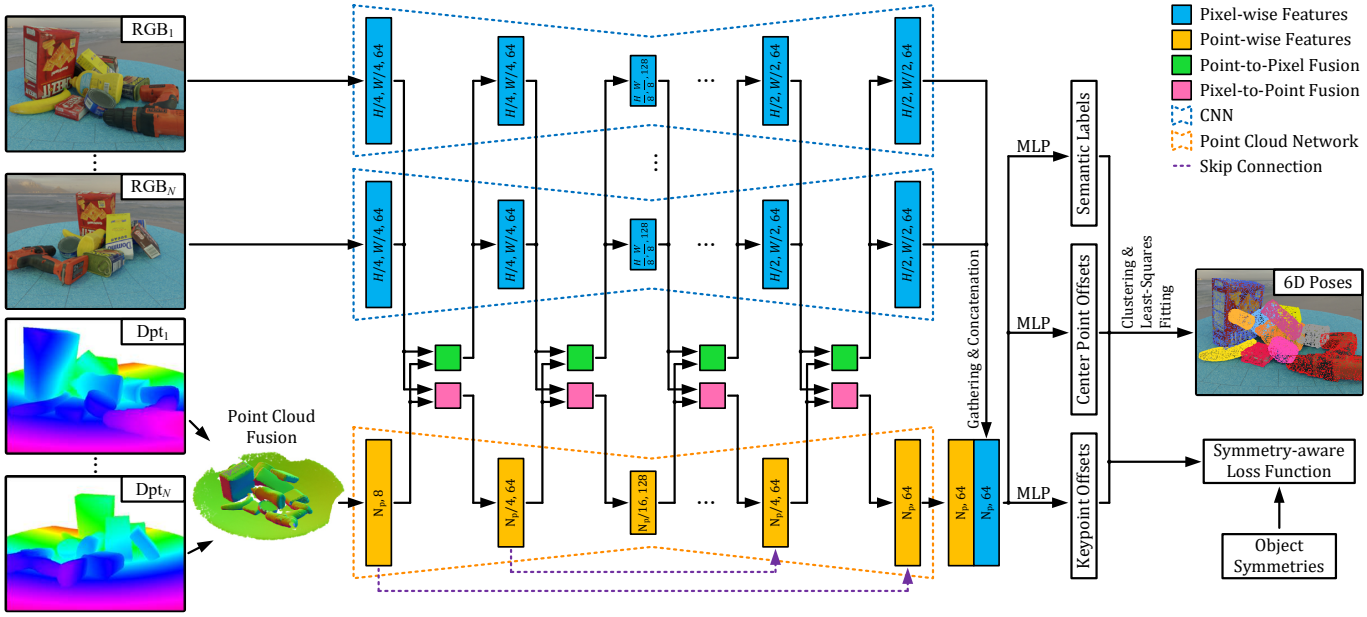



Fig. 2: Network architecture of SyMFM6D which fuses $N$ RGB-D input images. Our method converts the $N$ depth images to a single point cloud which is processed by an encoder-decoder point cloud network. The $N$ RGB images are processed by an encoder-decoder CNN. Every hierarchy contains a point-to-pixel fusion module and a pixel-to-point fusion module for deep multi-directional multi-view fusion. We utilize three MLPs with four layers each to regress 3D keypoint offsets, center point offsets, and semantic labels based on the final features. The 6D object poses are computed as in [2] based on mean shift clustering and least-squares fitting. We train our network by minimizing our proposed symmetry-aware multi-task loss function using precomputed object symmetries. $N_p$ is the number of points in the point cloud. $H$ and $W$ are height and width of the RGB images.

coordinates of an observed object from the object coordinate system into the camera coordinate system. This transformation is called 6D object pose because it is composed of a 3D rotation $\boldsymbol{R} \in SO(3)$ and a 3D translation $\boldsymbol{t} \in \mathbb{R}^3$. The designated aim of our approach is to jointly estimate the 6D poses of all objects in a given cluttered scene using multiple RGB-D images which depict the scene from multiple perspectives. We assume the 3D models of the objects and the camera poses to be known as proposed by [47].

### A. Network Overview

Our symmetry-aware multi-view network consists of three stages which are visualized in Fig. 2. The first stage receives one or multiple RGB-D images and extracts visual features as well as geometric features which are fused to a joint representation of the scene. The second stage performs a detection of predefined 3D keypoints and an instance semantic segmentation. Based on the keypoints and the information to which object the keypoints belong, we compute the 6D object poses with a least-squares fitting algorithm [45] in the third stage.

### B. Multi-View Feature Extraction

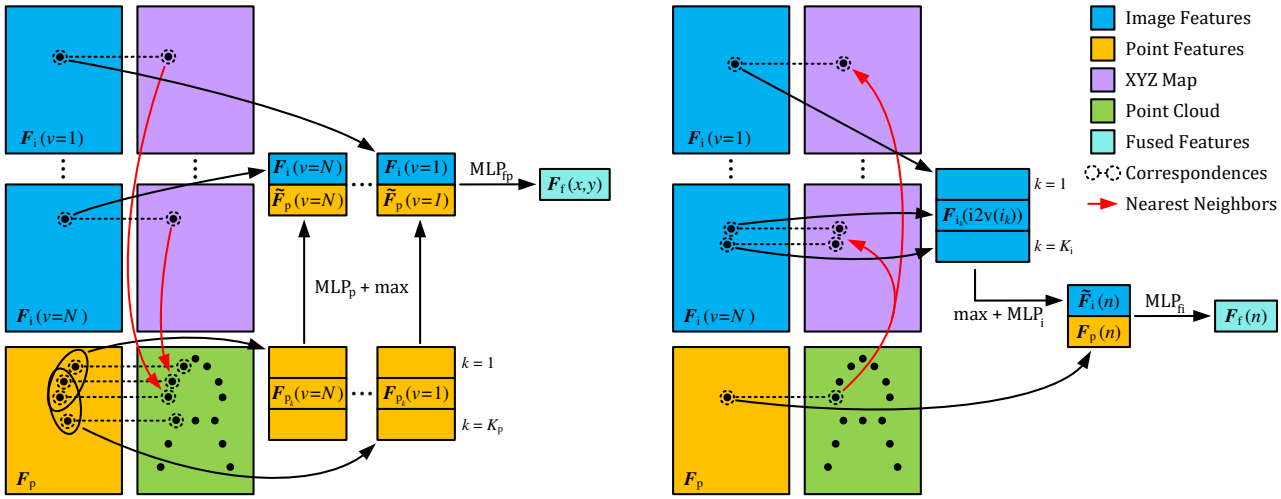
To efficiently predict keypoints and semantic labels, the first stage of our approach learns a compact representation of the given scene by extracting and merging features from all available RGB-D images in a deep multi-directional fusion manner. For that, we first separate the set of RGB images $\text{RGB}_1, ..., \text{RGB}_N$ from their corresponding depth images $\text{Dpt}_1$, ..., $\text{Dpt}_N$. The $N$ depth images are converted into point clouds, transformed into the coordinate system of the first camera, and merged to a single point cloud using the known camera poses as in [47]. Unlike [47], we employ a point cloud network based on RandLA-Net [55] with an encoder-decoder architecture using skip connections. The point cloud network learns geometric features from the fused point cloud and considers visual features from the multi-directional point-to-pixel fusion modules as described in Sec. III-C.

The $N$ RGB images are independently processed by a CNN with encoder-decoder architecture using the same weights for all $N$ views. The CNN learns visual features while considering geometric features from the multi-directional pixel-to-point fusion modules. We followed [3] and build the encoder upon a ResNet-34 [56] pretrained on ImageNet [57] and the decoder upon a PSPNet [58].

After the encoding and decoding procedures including several multi-view feature fusions, we collect the visual features from each view corresponding to the final geometric feature map and concatenate them. The output is a compact feature tensor containing the relevant information about the entire scene which is used for keypoint detection and instance semantic segmentation as described in Sec. III-D.

### C. Multi-View Feature Fusion

In order to efficiently fuse the visual and geometric features from multiple views, we extend the fusion modules of FFB6D [3] from bi-directional fusion to *multi-directional fusion*. We present two types of multi-directional fusion modules which are illustrated in Fig. 3. Both types of fusion modules take the pixel-wise visual feature maps and the point-wise geometric feature maps from each view, combine them, and compute a new feature map. This process requires a correspondence




(a) Point-to-pixel fusion module.

(b) Pixel-to-point fusion module.

Fig. 3: Overview of our proposed multi-directional multi-view fusion modules. They combine pixel-wise visual features and point-wise geometric features by exploiting the correspondence between pixels and points using the nearest neighbor algorithm. We compute the resulting features using multiple shared MLPs with a single layer and max-pooling. For simplification, we depict an example with $N = 2$ views and $K_\mathrm{i} = K_\mathrm{p} = 3$ nearest neighbors. The points of ellipsis (...) illustrate the generalization for an arbitrary number of views $N$. Please refer to [3] for better understanding the basic operations.

between pixel-wise and point-wise features which we obtain by computing an XYZ map for each RGB feature map based on the depth data of each pixel using the camera intrinsic matrix as in [3]. To deal with the changing dimensions at different layers, we use the centers of the convolutional kernels as new coordinates of the feature maps and resize the XYZ map to the same size using nearest interpolation as proposed in [3].

The *point-to-pixel* fusion module in Fig. 3a computes a fused feature map $\boldsymbol{F}_\mathrm{f}$ based on the image features $\boldsymbol{F}_\mathrm{i}(v)$ of all views $v \in \{1, \dots, N\}$. We collect the $K_\mathrm{p}$ nearest point features $\boldsymbol{F}_{\mathrm{p}_k}(v)$ with $k \in \{1, \dots, K_\mathrm{p}\}$ from the point cloud for each pixel-wise feature and each view independently by computing the nearest neighbors according to the Euclidean distance in the XYZ map. Subsequently, we process them by a shared MLP before aggregating them by max-pooling, i.e.,

$$\widetilde{\boldsymbol{F}}_\mathrm{p}(v) = \max_{k \in \{1,\dots,K_\mathrm{p}\}} \Big( \mathrm{MLP}_\mathrm{p}(\boldsymbol{F}_{\mathrm{p}_k}(v)) \Big). \tag{1}$$

Finally, we apply a second shared MLP to fuse all features $\boldsymbol{F}_\mathrm{i}$ and $\widetilde{\boldsymbol{F}}_\mathrm{p}$ as $\boldsymbol{F}_\mathrm{f} = \mathrm{MLP}_\mathrm{fp}(\widetilde{\boldsymbol{F}}_\mathrm{p} \oplus \boldsymbol{F}_\mathrm{i})$ where $\oplus$ denotes the concatenate operation.

The *pixel-to-point* fusion module in Fig. 3b collects the $K_\mathrm{i}$ nearest image features $\boldsymbol{F}_{\mathrm{i}_k}(\mathrm{i2v}(i_k))$ with $k \in \{1, ..., K_\mathrm{i}\}$. $\mathrm{i2v}(i_k)$ is a mapping that maps the index of an image feature to its corresponding view. This procedure is performed for each point feature vector $\boldsymbol{F}_\mathrm{p}(n)$. We aggregate the collected image features by max-pooling and apply a shared MLP, i.e.,

$$\widetilde{\boldsymbol{F}}_\mathrm{i} = \mathrm{MLP}_\mathrm{i} \left( \max_{k \in \{1,\dots,K_\mathrm{i}\}} \Big( \boldsymbol{F}_{\mathrm{i}_k}(\mathrm{i2v}(i_k)) \Big) \right). \tag{2}$$

One more shared MLP fuses the resulting image features $\widetilde{\boldsymbol{F}}_\mathrm{i}$ with the point features $\boldsymbol{F}_\mathrm{p}$ as $\boldsymbol{F}_\mathrm{f} = \mathrm{MLP}_\mathrm{fi}(\widetilde{\boldsymbol{F}}_\mathrm{i} \oplus \boldsymbol{F}_\mathrm{p})$.

### D. Keypoint Detection and Segmentation

The second stage of our SyMFM6D network contains modules for 3D keypoint detection and instance semantic segmentation following [47]. However, unlike [47], we use the SIFT-FPS algorithm [16] as proposed by FFB6D [3] to define eight target keypoints for each object class. SIFT-FPS yields keypoints with salient features which are easier to detect. Based on the extracted features, we apply two shared MLPs to estimate the translation offsets from each point of the fused point cloud to each target keypoint and to each object center. We obtain the actual point proposals by adding the translation offsets to the respective points of the fused point cloud. Applying the mean shift clustering algorithm [59] results in predictions for the keypoints and the object centers. We employ one more shared MLP for estimating the object class of each point in the fused point cloud as in [2].

### E. 6D Pose Computation via Least-Squares Fitting

Following [2], we use the least-squares fitting algorithm [45] to compute the 6D poses of all objects based on the estimated keypoints. As the $M$ estimated keypoints $\widehat{\boldsymbol{k}}_1, ..., \widehat{\boldsymbol{k}}_M$ are in the coordinate system of the first camera and the target keypoints $\boldsymbol{k}_1, ..., \boldsymbol{k}_M$ are in the object coordinate system, least-squares fitting calculates the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ of the 6D pose by minimizing the squared loss

$$L_\text{Least-squares} = \sum_{i=1}^{M} \left\| \widehat{\boldsymbol{k}}_i - (\boldsymbol{R}\boldsymbol{k}_i + \boldsymbol{t}) \right\|^2. \tag{3}$$

### F. Symmetry-aware Keypoint Detection

Most related work, including [2], [3], and [47] does not specifically consider object symmetries. However, symmetries lead to ambiguities in the predicted keypoints as multiple 6D poses can have the same visual and geometric appearance. Therefore, we introduce a novel symmetry-aware training procedure for the 3D keypoint detection including a novel symmetry-aware objective function to make the network

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

predicting either the original set of target keypoints for an object or a rotated version of the set corresponding to one object symmetry. Either way, we can still apply the least-squares fitting which efficiently computes an estimate of the target 6D pose or a rotated version corresponding to an object symmetry. To do so, we precompute the set $\boldsymbol{S}_I$ of all rotational symmetric transformations for the given object instance $I$ with a stochastic gradient descent algorithm [60]. Given the known mesh of an object and an initial estimate for the symmetry axis, we transform the object mesh along the symmetry axis estimate and optimize the symmetry axis iteratively by minimizing the ADD-S metric [61]. Reflectional symmetries which can be represented as rotational symmetries are handled as rotational symmetries. Other reflectional symmetries are ignored, since the reflection cannot be expressed as an Euclidean transformation. To consider continuous rotational symmetries, we discretize them into 16 discrete rotational symmetry transformations.

We extend the keypoints loss function of [2] to become symmetry-aware such that it predicts the keypoints of the closest symmetric transformation, i.e.

$$L_{\text{kp}}(\mathcal{I}) = \frac{1}{N_I} \min_{\boldsymbol{S} \in \boldsymbol{S}_I} \sum_{i \in \mathcal{I}} \sum_{j=1}^{M} \| \boldsymbol{x}_{ij} - \boldsymbol{S} \widehat{\boldsymbol{x}}_{ij} \| , \tag{4}$$

where $N_I$ is the number of points in the point cloud for object instance $I$, $M$ is the number of target keypoints per object, and $\mathcal{I}$ is the set of all point indices that belong to object instance $I$. The vector $\widehat{\boldsymbol{x}}_{ij}$ is the predicted keypoint offset for the $i$-th point and the $j$-th keypoint while $\boldsymbol{x}_{ij}$ is the corresponding ground truth.

### G. Objective Function

We train our network by minimizing the multi-task loss function

$$L_{\text{multi-task}} = \lambda_1 L_{\text{kp}} + \lambda_2 L_{\text{semantic}} + \lambda_3 L_{\text{cp}}, \tag{5}$$

where $L_{\text{kp}}$ is our symmetry-aware keypoint loss from Eq. (4). $L_{\text{cp}}$ is an L1 loss for the center point prediction, $L_{\text{semantic}}$ is a Focal loss [62] for the instance semantic segmentation, and $\lambda_1 = 2$, $\lambda_2 = 1$, and $\lambda_3 = 1$ are the weights for the individual loss functions as in [3].

## IV. Experiments

To demonstrate the performance of our method in comparison to related approaches, we perform extensive experiments on four very challenging datasets.

### A. Datasets

The **YCB-Video dataset** [1] contains a total of 133,827 RGB-D images showing 92 scenes composed of three to nine objects from the 21 Yale-CMU-Berkeley (YCB) objects [63]. Additionally, there are 80,000 synthetic non-sequential RGB-D frames showing a random subset of the YCB objects placed at random positions.

However, most frames from YCB-Video are very similar because they originate from videos with 30 frames per second recorded by a handheld camera that was moved slowly. The videos also do not show the scene from all sides but just from similar perspectives. Furthermore, the scenes do not include strong occlusions, and hence, most object poses are simple to estimate from a single perspective. Therefore, we additionally consider the recently proposed photorealistic synthetic datasets **MV-YCB FixCam** and **MV-YCB WiggleCam** [47] as they contain much more difficult scenes with strong occlusions and diverse camera perspectives. Both datasets depict 8,333 cluttered scenes composed of eleven non-symmetric YCB objects which are randomly arranged so that strong occlusions occur. Each scene is photorealistically rendered from three very different perspectives providing 24,999 RGB-D images with accurate ground truth annotations. Unlike FixCam which uses fixed camera positions while providing accurate camera poses, WiggleCam has varying camera poses which are inaccurately annotated on purpose.

Since FixCam and WiggleCam contain only non-symmetric objects, we created an additional photorealistic synthetic dataset with symmetric and non-symmetric objects called **MV-YCB SymMovCam** using Blender with physically based rendering and domain randomization as in [47]. It also depicts 8,333 cluttered scenes, but they are composed of 8 – 16 objects randomly chosen from the 21 YCB objects which results in very strong occlusions. For each scene, we created four cameras at changing positions around the scene with the restriction that in each quadrant there is only one camera so that the perspectives are very distinct. This results in a total of 33,332 annotated RGB-D images.

### B. Training Procedure

For training our model in single-view mode on YCB-Video, we randomly use the synthetic and real images of YCB-Video with a ratio of 4:1. Since consecutive real frames are very similar, we consider only every seventh real frame. For training a multi-view model, we start from the corresponding single-view checkpoint and continue training with batches of real YCB-Video frames. For training on FixCam and WiggleCam we follow [47] and use random permutations of the three available camera views. For SymMovCam, we take a random subset of three views from the available four views.

### C. Evaluation Metrics

We evaluated our method using the area-under-curve (AUC) metrics for ADD-S and ADD(-S) and the precision metrics ADD-S $< 2\,\text{cm}$ and ADD(-S) $< 2\,\text{cm}$ as these metrics are most commonly used in related work [3], [15], [47].

### D. Baseline Methods

We compare our methods with many established and some very recent methods namely DenseFusion [12], CosyPose [15], PVN3D [2], FFB6D [3], ES6D [54], and MV6D [47].

### E. Results on YCB-Video

Tab. I compares the single-view performance of our SyMFM6D network with all baseline methods using the AUC of ADD-S and ADD(-S) on YCB-Video. Please note that MV6D

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

corresponds to PVN6D in the single-view scenario. The results show that our approach copes very well with the dynamic camera setup of YCB-Video while outperforming all methods significantly. On the symmetry-aware ADD(-S) AUC metric, SyMFM6D outperforms the current state-of-the-art FFB6D by even 1.5 %. Please note that unlike DenseFusion (iterative) and CosyPose, our approach does not perform computationally expensive post processing or iterative refinement procedures.

To examine the effect of our symmetry-aware training procedure, we provide an object-wise evaluation of the three best single-view methods on YCB-Video in Tab. II. Please note that in single-view mode, our model architecture is the same as FFB6D except for our novel symmetry-aware loss function. The results show that not only most symmetric objects (highlighted in bold) are estimated more accurate but also most non-symmetric objects. This indicates that there is a synergy effect which improves the keypoint detection for non-symmetric objects due to an improvement of the keypoint detection for symmetric objects.

Fig. 4 shows a visualization of three scenes of YCB-Video with 6D pose ground truth, predictions of FFB6D, and predictions of our SyMFM6D network using only the depicted view. It can be seen that both FFB6D and SyMFM6D estimate very accurate poses as the scenes of YCB-Video contain only a few objects and not many occlusions. However, SyMFM6D predicts even more accurate poses than FFB6D due to our proposed symmetry-aware training procedure.

Tab. III compares our multi-view results with all multi-view baseline methods on YCB-Video using three and five input views. We see that our approach with disabled symmetry training procedure already outperforms all previous multi-view methods significantly. Enabling the symmetry awareness further improves the results slightly. However, using more views does not improve the accuracy as most views of YCB-Video are very similar in which case additional views do not provide beneficial information while the learning problem of fusing different views becomes slightly harder.

### *F. Results on MV-YCB FixCam, WiggleCam and SymMovCam*

We show the quantitative results on the datasets MV-YCB FixCam, MV-YCB WiggleCam, and MV-YCB SymMovCam in Tab. IV. It includes a comparison with two modified CosyPose (CP) versions with and without known camera poses as presented by [47]. Our SyMFM6D network yields the best results on all metrics on all three datasets. This shows that SyMFM6D copes very well with the strong occlusions in the datasets. The results on WiggleCam are just slightly worse than on FixCam which demonstrates that our approach is robust towards inaccurately known camera poses.

On the novel SymMovCam dataset, our method outperforms the baselines by a much larger margin than on FixCam and WiggleCam. This is due to the symmetric objects in the datasets on which the keypoint estimation of the baseline methods is inaccurate. The results also prove that our approach is robust to very dynamic camera setups where the cameras are mounted at varying positions.

### *G. Keypoint Visualization*

Fig. 5 shows predicted keypoints of FFB6D and SyMFM6D in a YCB-Video scene. We additionally visualize the keypoint proposals of each object in individual colors. The resulting predicted keypoints are white, the target keypoints are black. You can see that both FFB6D and SyMFM6D predict very accurate keypoints on all non-symmetric objects. However, FFB6D fails to predict accurate keypoints on the large clamp which has one discrete rotational symmetry. This shortcoming of FFB6D is also apparent on other symmetric objects. We believe that this is caused by the ambiguities of the object poses resulting in ambiguous target keypoints which results in averaging over the multiple solutions given by the symmetry. Therefore, the training loss is minimized when predicting keypoints on the symmetric axis rather than predicting them on the desired target locations. SyMFM6D in contrast overcomes

| | ADD-S | ADD(-S) |
|---|---|---|
| DenseFusion (per-pixel) | 91.2 | 82.9 |
| DenseFusion (iterative) | 93.2 | 86.1 |
| CosyPose | 89.8 | 84.5 |
| PVN3D | 95.5 | 91.8 |
| FFB6D | 96.6 | 92.7 |
| ES6D | 93.6 | 89.0 |
| SyMFM6D | **96.8** | **94.1** |

TABLE I: Single-view results on YCB-Video using the AUC metrics for ADD-S and ADD(-S). The best results are printed in bold.

| Object class | PVN3D | FFB6D | SyMFM6D |
|---|---|---|---|
| Master chef can | 80.5 | 80.6 | **80.7** |
| Cracker box | 94.8 | 94.6 | **94.9** |
| Sugar box | 96.3 | **96.6** | **96.6** |
| Tomato soup can | 88.5 | **89.6** | 87.9 |
| Mustard bottle | 96.2 | 97.0 | **97.8** |
| Tuna fish can | 89.3 | 88.9 | **92.3** |
| Pudding box | **95.7** | 94.6 | 93.3 |
| Gelatin box | 96.1 | **96.9** | 96.1 |
| Potted meat can | 88.6 | 88.1 | **90.0** |
| Banana | 93.7 | 94.9 | **95.2** |
| Pitcher base | 96.5 | 96.9 | **97.5** |
| Bleach cleanser | 93.2 | **94.8** | 93.9 |
| **Bowl** | 90.2 | 96.3 | **96.4** |
| Mug | 95.4 | 94.2 | **95.7** |
| Power drill | 95.1 | 95.9 | **96.4** |
| **Wood block** | 90.4 | 92.6 | **95.2** |
| Scissors | 92.7 | 95.7 | **95.8** |
| Large marker | **91.8** | 89.1 | 90.0 |
| **Large clamp** | 93.6 | 96.8 | **96.9** |
| **Extra large clamp** | 88.4 | **96.0** | 95.3 |
| **Foam brick** | 96.8 | 97.3 | **97.6** |
| ALL | 91.8 | 92.7 | **94.1** |

TABLE II: Single-view results on YCB-Video evaluated for each object class individually using the ADD(-S) AUC metric. Symmetric objects and the best results are printed in bold.

| | ADD-S | | ADD(-S) | |
|---|---|---|---|---|
| | 3 views | 5 views | 3 views | 5 views |
| CosyPose | 92.3 | 93.4 | 87.7 | 88.8 |
| MV6D | 91.2 | 91.1 | 85.6 | 84.0 |
| SyMFM6D (no sym) | 95.2 | 95.2 | 91.5 | 91.4 |
| SyMFM6D | **95.4** | **95.4** | **91.7** | **91.6** |

TABLE III: Quantitative multi-view results on YCB-Video. The best results are printed in bold.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

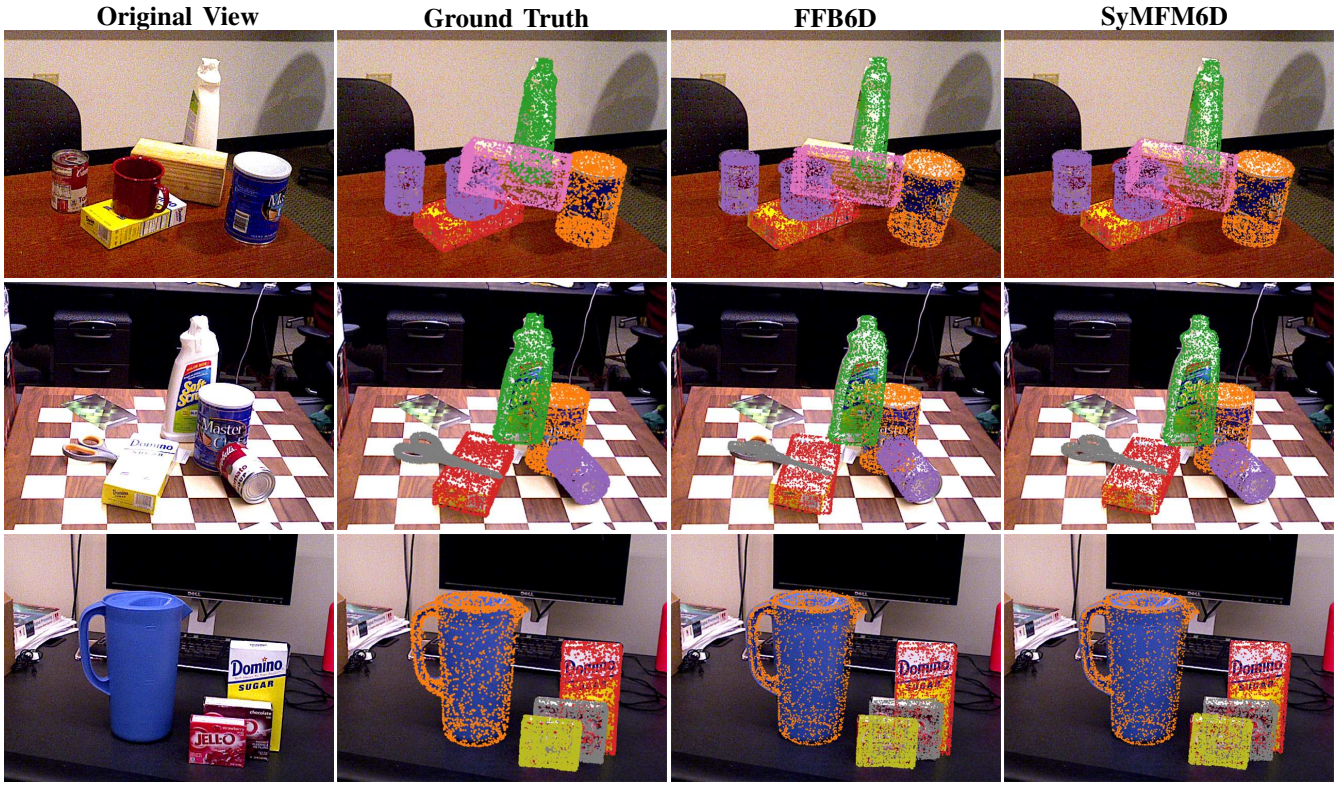



Fig. 4: Comparison of 6D pose predictions on single frames of the YCB-Video dataset.

| | MV-YCB FixCam | | | | | | MV-YCB WiggleCam | | | | | | MV-YCB SymMovCam | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PVN3D | FFB6D | CP | CP | MV6D | Ours | PVN3D | FFB6D | CP | CP | MV6D | Ours | PVN3D | FFB6D | Ours | MV6D | Ours |
| Number of views | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 3 |
| Known cam poses | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADD-S AUC | 81.3 | 82.3 | 90.8 | 91.9 | 96.9 | **97.3** | 80.8 | 81.9 | 90.0 | 91.3 | 96.2 | **96.7** | 75.0 | 79.9 | 80.6 | 92.8 | **94.2** |
| ADD(-S) AUC | 74.9 | 76.3 | 82.4 | 84.6 | 94.8 | **95.6** | 74.0 | 75.5 | 81.0 | 83.4 | 93.0 | **94.2** | 68.5 | 75.6 | 76.7 | 88.7 | **91.6** |
| ADD-S $< 2$ cm | 82.1 | 83.6 | 92.9 | 93.0 | 98.8 | **98.9** | 82.0 | 83.4 | 92.3 | 92.6 | 98.7 | **98.8** | 77.2 | 81.1 | 81.9 | 96.3 | **96.6** |
| ADD(-S) $< 2$ cm | 73.0 | 74.8 | 80.6 | 82.4 | 96.5 | **96.8** | 72.4 | 74.0 | 78.9 | 81.6 | **96.0** | **96.0** | 64.5 | 74.5 | 76.3 | 91.6 | **93.6** |

TABLE IV: Quantitative results on the datasets MV-YCB FixCam (left), MV-YCB WiggleCam (middle), and MV-YCB SymMovCam (right). The baseline CosyPose (CP) uses PVN3D as backend network as described in [47]. The best results per dataset are printed in bold.

this problem by our novel symmetry-aware training procedure as it can be seen in Fig. 5b.

(a) FFB6D (b) SyMFM6D

Fig. 5: Visualization of the predicted keypoints on single frames of the YCB-Video dataset.

### H. Implementation Details and Runtime

We trained our network up to seven days on four NVIDIA Tesla V100 GPUs with 32 GB of memory. The network architecture of our SyMFM6D approach has 3.5 million trainable parameters and requires about 46 ms for processing a single RGB-D image on a single GPU. Mean shift clustering and least-squares fitting for computing a 6D pose require additional 14 ms per object. Please visit our previously mentioned GitHub repository for code, datasets, and further details.

## V. Conclusion

In this work, we present SyMFM6D, a novel approach for symmetry-aware multi-view 6D object pose estimation based on a deep multi-directional fusion network for RGB-D data. We additionally propose a novel method for predicting predefined 3D keypoints of symmetric objects based on a symmetry-aware objective function. Using the 3D keypoint predictions and an instance semantic segmentation, we compute the 6D poses of all objects in the scene simultaneously with least-squares fitting. Our experiments show that our symmetry-aware training

procedure significantly improves the 6D pose estimation accuracy of both symmetric and non-symmetric objects due to synergy effects. Our method outperforms the state-of-the-art in single-view and multi-view 6D pose estimation on four very challenging datasets. We furthermore demonstrate the robustness of our approach towards inaccurately known camera poses and dynamic camera setups.

## REFERENCES


[1] Y. Xiang *et al.*, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *RSS*, 2018.

[2] Y. He *et al.*, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *CVPR*, 2020, pp. 11 632–11 641.

[3] ——, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *CVPR*, 2021, pp. 3003–3013.

[4] J. Ku *et al.*, "Joint 3D proposal generation and object detection from view aggregation," in *IROS*. IEEE, 2018, pp. 1–8.

[5] B. Gu *et al.*, "ECPC-ICP: A 6D vehicle pose estimation method by fusing the roadside lidar point cloud and road feature," *Sensors*, vol. 21, no. 10, p. 3489, 2021.

[6] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. Visual. Comput. Graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.

[7] Y. Su *et al.*, "Deep multi-state object pose estimation for augmented reality assembly," in *ISMAR-Adjunct*. IEEE, 2019, pp. 222–227.

[8] Y. Di *et al.*, "SO-Pose: Exploiting self-occlusion for direct 6D pose estimation," in *CVPR*, 2021, pp. 12 396–12 405.

[9] Y. Su *et al.*, "ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation," in *CVPR*, 2022, pp. 6738–6748.

[10] F. Hagelskjær and A. G. Buch, "PointVoteNet: Accurate object detection and 6 DoF pose estimation in point clouds," in *ICIP*. IEEE, 2020, pp. 2641–2645.

[11] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Voting and attention-based pose relation learning for object pose estimation from 3D point clouds," *RA-L*, vol. 7, no. 4, pp. 8980–8987, 2022.

[12] C. Wang *et al.*, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *CVPR*, 2019, pp. 3343–3352.

[13] A. Zeng *et al.*, "Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge," in *ICRA*. IEEE, 2017, pp. 1386–1383.

[14] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *ECCV*, 2018, pp. 254–269.

[15] Y. Labbé *et al.*, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *ECCV*, 2020, pp. 574–591.

[16] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2, 1999, pp. 1150–1157.

[17] ——, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*, 2006, pp. 430–443.

[19] A. Collet, M. Martinez, and S. S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *IJRR*, vol. 30, no. 10, pp. 1284–1306, 2011.

[20] A. Collet *et al.*, "Object recognition and full pose registration from a single image for robotic manipulation," in *ICRA*. IEEE, 2009.

[21] S. Peng *et al.*, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *CVPR*, 2019, pp. 4561–4570.

[22] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[23] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," in *ECCV*, 2010, pp. 408–421.

[24] S. Hinterstoisser *et al.*, "Gradient response maps for real-time detection of textureless objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 5, pp. 876–888, 2011.

[25] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *ICRA*. IEEE, 2016, pp. 2441–2448.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *CVPR*, 2015, pp. 1510–1519.

[28] Y. Li *et al.*, "DeepIM: Deep iterative matching for 6D pose estimation," in *ECCV*, 2018, pp. 683–698.

[29] K. Gupta, L. Petersson, and R. Hartley, "CullNet: Calibrated and pose aware confidence scores for object pose estimation," in *ICCVW*, 2019.

[30] W. Kehl *et al.*, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *ICCV*, 2017, pp. 1521–1529.

[31] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *CVPR*, 2018, pp. 292–301.

[32] G. Wang *et al.*, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *CVPR*, 2021.

[33] J. Chen *et al.*, "Survey on 6D pose estimation of rigid object," in *CCC*. IEEE, 2020, pp. 7440–7445.

[34] D. Fernandes *et al.*, "Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy," *Information Fusion*, vol. 68, pp. 161–191, 2021.

[35] S. Song and J. Xiao, "Sliding shapes for 3D object detection in depth images," in *ECCV*, 2014, pp. 634–651.

[36] B. Li, "3D fully convolutional network for vehicle detection in point cloud," in *IROS*. IEEE, 2017, pp. 1513–1518.

[37] C. R. Qi *et al.*, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *CVPR*, Jul. 2017, pp. 652–660.

[38] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *CVPR*, 2018, pp. 4490–4499.

[39] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[40] A. H. Lang *et al.*, "PointPillars: Fast encoders for object detection from point clouds," in *CVPR*, 2019, pp. 12 697–12 705.

[41] C. R. Qi *et al.*, "Deep hough voting for 3D object detection in point clouds," in *CVPR*, 2019, pp. 9277–9286.

[42] Q. Xie *et al.*, "VENet: Voting enhancement network for 3D object detection," in *CVPR*, 2021, pp. 3712–3721.

[43] X. Chen *et al.*, "Multi-view 3D object detection network for autonomous driving," in *CVPR*, 2017, pp. 1907–1915.

[44] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep sensor fusion for 3D bounding box estimation," in *CVPR*, 2018, pp. 244–253.

[45] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, no. 5, pp. 698–700, 1987.

[46] B. Triggs *et al.*, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*. Springer, 2000, pp. 298–372.

[47] F. Duffhauss, T. Demmler, and G. Neumann, "MV6D: Multi-view 6D pose estimation on RGB-D frames using a deep point-wise voting network," in *IROS*. IEEE, 2022.

[48] G. Pitteri *et al.*, "On object symmetries and 6D pose estimation from images," in *3DV*. IEEE, 2019, pp. 614–622.

[49] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *ICCV*, 2017, pp. 3828–3836.

[50] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *ICCV*, 2019.

[51] T. Hodan, D. Barath, and J. Matas, "EPOS: Estimating 6D pose of objects with symmetries," in *CVPR*, 2020, pp. 11 703–11 712.

[52] H. Zhang *et al.*, "Symmetry-aware 6D object pose estimation via multitask learning," *Complexity*, 2020.

[53] J. Richter-Klug and U. Frese, "Handling object symmetries in CNN-based pose estimation," in *ICRA*. IEEE, 2021, pp. 13 850–13 856.

[54] N. Mo *et al.*, "ES6D: A computation efficient and symmetry-aware 6D pose regression framework," in *CVPR*, 2022, pp. 6718–6727.

[55] Q. Hu *et al.*, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *CVPR*, 2020, pp. 11 108–11 117.

[56] K. He *et al.*, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[57] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[58] H. Zhao *et al.*, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[59] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 790–799, 1995.

[60] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *ICLR*, 2017.

[61] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *ACCV*. Springer, 2012, pp. 548–562.

[62] T. Lin *et al.*, "Focal loss for dense object detection," in *ICCV*, Oct. 2017, pp. 2999–3007.

[63] B. Calli *et al.*, "The YCB object and model set: Towards common benchmarks for manipulation research," in *ICAR*. IEEE, 2015.