DualPoseNet: Category-level 6D Object Pose and Size Estimation Using Dual Pose Network with Refined Learning of Pose Consistency

Jiehong Lin[1], Zewei Wei[1,2], Zhihao Li[3], Songcen Xu[3], Kui Jia[1]*, Yuanqing Li[1]
[1]South China University of Technology
[2]DexForce Technology Co. Ltd. [3]Noah's Ark Lab, Huawei Technologies Co. Ltd.
{lin.jiehong, eeweizewei}@mail.scut.edu.cn, {zhihao.li, xusongcen}@huawei.com,
{kuijia, auyqli}@scut.edu.cn

Abstract

Category-level 6D object pose and size estimation is to predict full pose configurations of rotation, translation, and size for object instances observed in single, arbitrary views of cluttered scenes. In this paper, we propose a new method of Dual Pose Network with refined learning of pose consistency *for this task, shortened as* DualPoseNet. *DualPoseNet stacks two parallel pose decoders on top of a shared pose encoder, where the implicit decoder predicts object poses with a working mechanism different from that of the explicit one; they thus impose complementary supervision on the training of pose encoder. We construct the encoder based on spherical convolutions, and design a module of Spherical Fusion wherein for a better embedding of pose-sensitive features from the appearance and shape observations. Given no testing CAD models, it is the novel introduction of the implicit decoder that enables the refined pose prediction during testing, by enforcing the predicted pose consistency between the two decoders using a self-adaptive loss term. Thorough experiments on benchmarks of both category- and instance-level object pose datasets confirm efficacy of our designs. DualPoseNet outperforms existing methods with a large margin in the regime of high precision. Our code is released publicly at* https://github.com/Gorilla-Lab-SCUT/DualPoseNet.

1. Introduction

Object detection in a 3D Euclidean space is demanded in many practical applications, such as augmented reality, robotic manipulation, and self-driving car. The field has been developing rapidly with the availability of benchmark datasets (e.g., KITTI [10] and SUN RGB-D [25]), where carefully annotated 3D bounding boxes enclosing object instances of interest are prepared, which specify 7 degrees of freedom (7DoF) for the objects, including translation, size, and yaw angle around the gravity axis. This 7DoF setting of 3D object detection is aligned with common scenarios where the majority of object instances stand upright in the 3D space. However, 7DoF detection cannot precisely locate objects when the objects lean in the 3D space, where the most compact bounding boxes can only be determined given full pose configurations, *i.e.*, with the additional two angles of rotation. Pose predictions of full configurations are important in safety-critical scenarios, e.g., autonomous driving, where the most precise and compact localization of objects enables better perception and decision making.

This task of pose prediction of full configurations (*i.e.*, 6D pose and size) is formally introduced in [30] as *category-level 6D object pose and size estimation* of novel instances from single, arbitrary views of RGB-D observations. It is closely related to *category-level* amodal 3D object detection [22, 31, 36, 24, 35, 21] (i.e., the above 7DoF setting) and *instance-level* 6D object pose estimation [12, 8, 14, 16, 32, 26, 20, 17, 29, 18]. Compared with them, the focused task in the present paper is more challenging due to learning and prediction in the full rotation space of $SO(3)$; more specifically, (1) the task is more involved in terms of both defining the category-level canonical poses (cf. Section 3 for definition of canonical poses) and aligning object instances with large intra-category shape variations [15, 4], (2) deep learning precise rotations arguably requires learning rotation-equivariant shape features, which is less studied compared with the 2D counterpart of learning translation-invariant image features, and (3) compared with instance-level 6D pose estimation, due to the lack of testing CAD models, the focused task cannot leverage the privileged 3D shapes to directly refine pose predictions, as done in [2, 32, 29, 18].

In this work, we propose a novel method for category-level 6D object pose and size estimation, which can partially address the second and third challenges mentioned above.

*Corresponding author

Our method constructs two parallel pose decoders on top of a shared pose encoder; the two decoders predict poses with different working mechanisms, and the encoder is designed to learn pose-sensitive shape features. A refined learning that enforces the predicted pose consistency between the two decoders is activated during testing to further improve the prediction. We term our method as *Dual Pose Network with refined learning of pose consistency*, shortened as *DualPoseNet*. Fig. 1 gives an illustration.

For an observed RGB-D scene, DualPoseNet first employs an off-the-shelf model of instance segmentation (*e.g.*, MaskRCNN [11]) in images to segment out the objects of interest. It then feeds each masked RGB-D region into the encoder. To learn pose-sensitive shape features, we construct our encoder based on spherical convolutions [9, 7], which provably learn deep features of object surface shapes with the property of rotation equivariance on $SO(3)$. In this work, we design a novel module of *Spherical Fusion* to support a better embedding from the appearance and shape features of the input RGB-D region. With the learned pose-sensitive features, the two parallel decoders either make a pose prediction *explicitly*, or *implicitly* do so by reconstructing the input (partial) point cloud in its canonical pose; while the first pose prediction can be directly used as the result of DualPoseNet, the result is further refined during testing by fine-tuning the encoder using a self-adaptive loss term that enforces the pose consistency. The use of implicit decoder in DualPoseNet has two benefits that potentially improve the pose prediction: (1) it provides an auxiliary supervision on the training of pose encoder, and (2) it is the key to enable the refinement given no testing CAD models. We conduct thorough experiments on the benchmark category-level object pose datasets of CAMERA25 and REAL275 [30], and also apply our DualPoseNet to the instance-level ones of YCB-Video [3] and LineMOD [13]. Ablation studies confirm the efficacy of our novel designs. DualPoseNet outperforms existing methods in terms of more precise pose. Our technical contributions are summarized as follows:

- We propose a new method of *Dual Pose Network* for category-level 6D object pose and size estimation. *DualPoseNet* stacks two parallel pose decoders on top of a shared pose encoder, where the implicit one predicts poses with a working mechanism different from that of the explicit one; the two decoders thus impose complementary supervision on training of the pose encoder.

- In spite of the lack of testing CAD models, the use of implicit decoder in DualPoseNet enables a refined pose prediction during testing, by enforcing the predicted pose consistency between the two decoders using a self-adaptive loss term. This further improves the results of DualPoseNet.

- We construct the encoder of DualPoseNet based on spherical convolutions to learn pose-sensitive shape features, and design a module of *Spherical Fusion* wherein, which is empirically shown to learn a better embedding from the appearance and shape features from the input RGB-D regions.

2. Related Work

Instance-level 6D Object Pose Estimation Traditional methods for instance-level 6D pose estimation include those based on template matching [12], and those by voting the matching results of point-pair features [8, 14]. More recent solutions build on the power of deep networks and can directly estimate object poses from RGB images alone [16, 32, 26, 20] or RGB-D ones [17, 29]. This task assumes the availability of object CAD models during both the training and test phases, and thus enables a common practice to refine the predicted pose by matching the CAD model with the (RGB and/or point cloud) observations [2, 32, 29, 18].

Category-level 3D Object Detection Methods for category-level 3D object detection are mainly compared on benchmarks such as KITTI [10] and SUN RGB-D [25]. Earlier approach [22, 31] leverages the mature 2D detectors to first detect objects in RGB images, and learning of 3D detection is facilitated by focusing on point sets inside object frustums. Subsequent research proposes solutions [36, 24, 35, 21] to predict the 7DoF object bounding boxes directly from the observed scene points. However, the 7DoF configurations impose inherent constrains on the precise rotation prediction, with only one yaw angle predicted around the gravity direction.

Category-level 6D Object Pose and Size Estimation More recently, category-level 6D pose and size estimation is formally introduced in [30]. Notably, Wang *et al.* [30] propose a canonical shape representation called normalized object coordinate space (NOCS), and inference is made by first predicting NOCS maps for objects detected in RGB images, and then aligning them with the observed object depths to produce results of 6D pose and size; later, Tian *et al.* [27] improve the predictions of canonical object models by deforming categorical shape priors. Instead, Chen *et al.* [5] trained a variational auto-encoder (VAE) to capture pose-independent features, along with pose-dependent ones to directly predict the 6D poses. Besides, monocular methods are also explored in recent works [6, 19].

3. Problem Statement

Studies on category-level 6D object pose and size estimation start from NOCS [30]. The problem can be formally stated as follows. Assume a training set of cluttered scenes captured in RGB-D images, where ground-truth annotations of 6D pose and size for object instances of certain

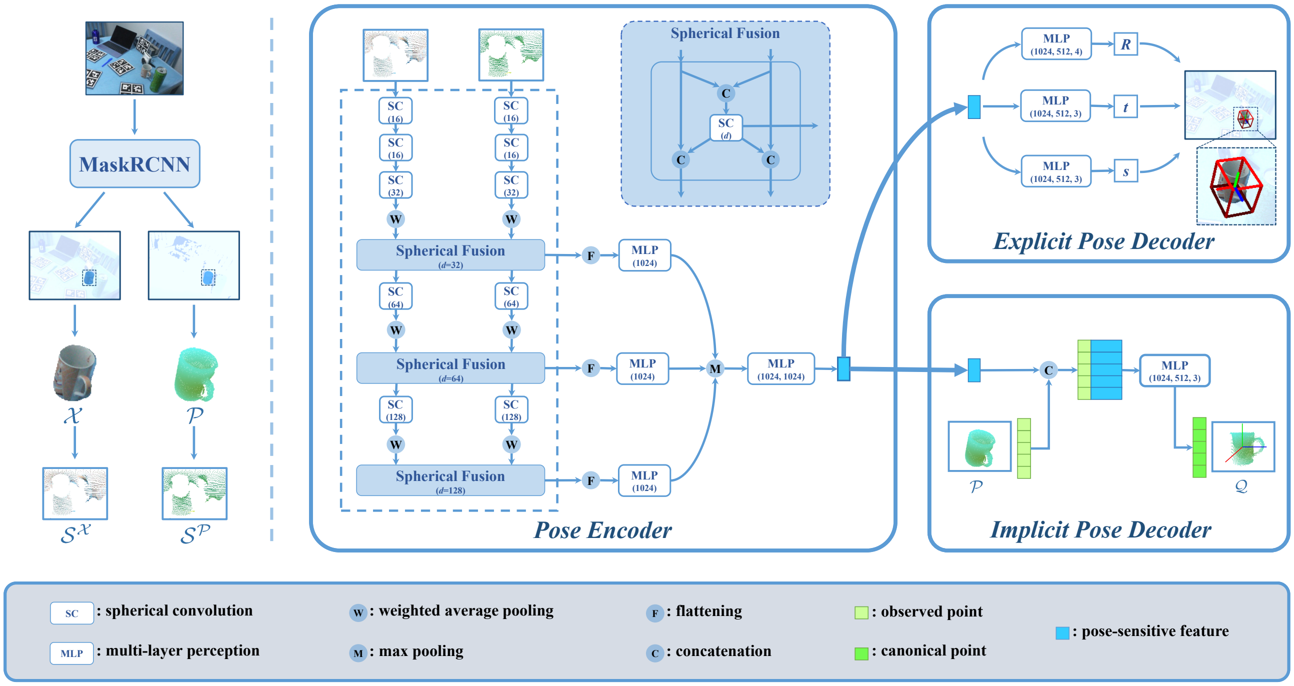

Figure 1. An illustration of our proposed DualPoseNet. For an observed RGB-D scene, DualPoseNet employs MaskRCNN [27] to segment out the object of interest, e.g., a mug, giving the observed points in $\mathcal{P}$ and the corresponding RGB values in $\mathcal{X}$, and feeds $(\mathcal{X}, \mathcal{P})$ into a **Pose Encoder** Φ to learn a pose-sensitive feature representation $\boldsymbol{f}$. Specifically, Φ is designed to have two parallel streams of spherical convolution layers to process the spherical signals $\mathcal{S}^{\mathcal{X}}$ and $\mathcal{S}^{\mathcal{P}}$ separately, which are respectively converted from $\mathcal{X}$ and $\mathcal{P}$; the resulting features are intertwined in the intermediate layers via a proposed module of *Spherical Fusion*; finally $\boldsymbol{f}$ is enriched and obtained by aggregation of multi-scale spherical features. On top of Φ, an **Explicit Pose Decoder** Ψ_{exp} is constructed to directly predict the pose, while an additional **Implicit Pose Decoder** Ψ_{im} is employed in parallel with Ψ_{exp} to generate a canonical version $\mathcal{Q}$ of $\mathcal{P}$.

categories are provided. For each of the contained object instances, the annotation is in the form of full pose configuration of rotation $\boldsymbol{R} \in SO(3)$, translation $\boldsymbol{t} \in \mathbb{R}^3$, and size $\boldsymbol{s} \in \mathbb{R}^3$, which can also be translated as a compact, oriented 3D bounding box enclosing the object (cf. Fig. 1). Note that in a 3D Euclidean space, the 6D pose of $\boldsymbol{R}$ and $\boldsymbol{t}$ is defined relatively with respect to a *canonical pose centered at the origin*. Category-level learning thus relies on the underlying assumption that all training object instances of a same category are aligned at a pre-defined canonical pose (*e.g.*, handles of instances of a *mug* category are all pointing towards a same direction); otherwise it makes no sense for any learned models to predict poses during testing. For existing datasets [30], additional annotations of object masks in RGB images are usually provided, which eases the problem and enables learning to segment out regions of interest from RGB-D images of cluttered scenes.

4. The Proposed Dual Pose Network with Refined Learning of Pose Consistency

4.1. Overview

We first present an overview of our proposed *Dual Pose Network with refined Learning of Pose Consistency*. The whole pipeline is depicted in Fig. 1. For an observed RGB-D scene, DualPoseNet first employs an off-the-shelf model of instance segmentation in images (*e.g.*, MaskRCNN [11]) to segment out the objects of interest. This produces a pair $(\mathcal{X}, \mathcal{P})$ for each segmented object, where we use $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^N$ to represent the N observed points in the masked RGB-D region and $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^3\}_{i=1}^N$ for the corresponding RGB values. DualPoseNet feeds $(\mathcal{X}, \mathcal{P})$ into a **Pose Encoder** Φ (cf. Section 4.2) to learn a pose-sensitive feature representation $\boldsymbol{f}$, followed by an **Explicit Pose Decoder** Ψ_{exp} (cf. Section 4.3) to predict the pose; an additional **Implicit Pose Decoder** Ψ_{im} (cf. Section 4.4) is employed in parallel with Ψ_{exp}, which generates a canonical version $\mathcal{Q}$ of the observed point cloud $\mathcal{P}$. The use of Ψ_{im} is the key in DualPoseNet to both improve the pose prediction from Ψ_{exp}, and enable a refined learning of pose consistency that further improves the precision of prediction, as verified in our experiments in Section 5.1.1. Given cropped RGB-D regions of interest and the ground-truth pose annotations, training of DualPoseNet can be conducted in an end-to-end manner. During testing, there exist (at least) three ways to obtain the pose predictions from DualPoseNet: (1) the direct prediction from DualPoseNet via a forward pass of $\Psi_{exp} \circ \Phi$, (2) computing $\mathcal{Q} = \Psi_{im} \circ \Phi(\mathcal{X}, \mathcal{P})$ in its canonical pose and obtaining a pose prediction by solving Umeyama algorithm [28] together with the observed $\mathcal{P}$, similar to existing methods [30, 27], and (3) using the refined learning to update the parameters of the encoder Φ, and then computing the prediction via a forward pass of $\Psi_{exp} \circ \Phi$. In this work, we use the first and third ways to obtain results of DualPoseNet. We illustrate the training and refinement processes in Fig. 2. Individual components of the network are explained as follows.

4.2. The Pose Encoder Φ

Precise prediction of object pose requires that the features learned by $\boldsymbol{f} = \Phi(\mathcal{X}, \mathcal{P})$ are sensitive to the observed pose of the input $\mathcal{P}$, especially to rotation, since translation and size are easier to infer from $\mathcal{P}$ (*e.g.*, even simple localization of center point and calculation of 3D extensions give a good prediction of translation and scale). To this end, we implement our Φ based on spherical convolutions [9, 7], which provably learn deep features of object surface shapes with the property of rotation equivariance on $SO(3)$. More specifically, we design Φ to have two parallel streams of spherical convolution layers that process the inputs $\mathcal{X}$ and $\mathcal{P}$ separately; the resulting features are intertwined in the intermediate layers via a proposed module of *Spherical Fusion*. We also use *aggregation of multi-scale spherical features* to enrich the pose information in $\boldsymbol{f}$. Fig. 1 gives the illustration.

Conversion as Spherical Signals Following [9], we aim to convert $\mathcal{X}$ and $\mathcal{P}$ separately as discrete samplings $\mathcal{S}^{\mathcal{X}} \in \mathbb{R}^{W \times H \times 3}$ and $\mathcal{S}^{\mathcal{P}} \in \mathbb{R}^{W \times H \times 1}$ of spherical signals, where $W \times H$ represents the sampling resolution on the sphere. To do so, we first compute the geometric center $\boldsymbol{c} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{p}_i$ of $\mathcal{P}$, and subtract its individual points from $\boldsymbol{c}$; this moves $\mathcal{P}$ into a space with the origin at $\boldsymbol{c}$. We then cast $W \times H$ equiangular rays from $\boldsymbol{c}$, which divide the space into $W \times H$ regions. Consider a region indexed by (w, h), with $w \in \{1, \dots, W\}$ and $h \in \{1, \dots, H\}$; when it contains points of $\mathcal{P}$, we find the one with the largest distance to $\boldsymbol{c}$, denoted as $\boldsymbol{p}_{h,w}^{max}$, and define the spherical signal at the present region as $\mathcal{S}^{\mathcal{X}}(w, h) = \boldsymbol{x}_{h,w}^{max}$ and $\mathcal{S}^{\mathcal{P}}(w, h) = \|\boldsymbol{p}_{h,w}^{max} - \boldsymbol{c}\|$, where $\boldsymbol{x}_{h,w}^{max}$ denotes the RGB values corresponding to $\boldsymbol{p}_{h,w}^{max}$; otherwise, we define $\mathcal{S}^{\mathcal{X}}(w, h) = \boldsymbol{0}$ and $\mathcal{S}^{\mathcal{P}}(w, h) = 0$ when the region contains no points of $\mathcal{P}$.

Learning with Spherical Fusion As shown in Fig. 1, our encoder Φ is constructed based on two parallel streams that process the converted spherical signals $\mathcal{S}^{\mathcal{X}}$ and $\mathcal{S}^{\mathcal{P}}$ separately. We term them as $\mathcal{X}$-stream and $\mathcal{P}$-stream for ease of presentation. The two streams share a same network structure (except the channels of the first layers), each of which stacks multiple layers of spherical convolution and weighted average pooling, whose specifics are given in Fig. 1. To enable information communication and feature mixing between the two streams, we design a module of *Spherical Fusion* that works as follows. Let $\mathcal{S}_l^{\mathcal{X}} \in \mathbb{R}^{W \times H \times d_l}$ and $\mathcal{S}_l^{\mathcal{P}} \in \mathbb{R}^{W \times H \times d_l}$ denote the learned spherical feature maps at the respective l^{th} layers of the two streams (i.e., $\mathcal{S}_0^{\mathcal{X}} = \mathcal{S}^{\mathcal{X}}$ and $\mathcal{S}_0^{\mathcal{P}} = \mathcal{S}^{\mathcal{P}}$), we compute the input feature maps of layer $l + 1$ for $\mathcal{P}$-stream as

$$\widetilde{\mathcal{S}}_l^{\mathcal{P}} = \left[\mathcal{S}_l^{\mathcal{P}}, \widetilde{\mathcal{S}}_l^{\mathcal{X},\mathcal{P}} \right] \in \mathbb{R}^{W \times H \times 2d_l} \tag{1}$$

$$\text{with } \widetilde{\mathcal{S}}_l^{\mathcal{X},\mathcal{P}} = \texttt{SCONV}\left(\left[\mathcal{S}_l^{\mathcal{X}}, \mathcal{S}_l^{\mathcal{P}} \right] \right) \in \mathbb{R}^{W \times H \times d_l}, \tag{2}$$

where SCONV denotes a trainable layer of spherical convolution [9], and $[\cdot, \cdot]$ concatenates spherical maps along the feature dimension. The same applies to $\mathcal{X}$-stream, and we have $\widetilde{\mathcal{S}}_l^{\mathcal{X}} \in \mathbb{R}^{W \times H \times 2d_l}$ as its input of layer $l+1$. The module of spherical fusion (1) can be used in a plug-and-play manner at any intermediate layers of the two streams; we use three such modules between $\mathcal{X}$-stream and $\mathcal{P}$-stream of 5 spherical convolution layers for all experiments reported in this paper. Empirical analysis in Section 5.1.1 verifies the efficacy of the proposed spherical fusion. Note that a simple alternative exists that fuses the RGB and point features at the very beginning, i.e., a spherical signal $\mathcal{S} = [\mathcal{S}^{\mathcal{X}}, \mathcal{S}^{\mathcal{P}}] \in \mathbb{R}^{W \times H \times 4}$ converted from $(\mathcal{X}, \mathcal{P})$. Features in $\mathcal{S}$ would be directly fused in subsequent layers. Empirical results in Section 5.1.1 also verify that given the same numbers of spherical convolution layers and feature maps, this alternative is greatly outperformed by our proposed spherical fusion.

Aggregation of Multi-scale Spherical Features It is intuitive to enhance pose encoding by using spherical features at multiple scales. Since the representation $\widetilde{\mathcal{S}}_l^{\mathcal{X},\mathcal{P}}$ computed by (2) fuses the appearance and geometry features at an intermediate layer l, we technically aggregate multiple of them from spherical fusion modules respectively inserted at lower, middle, and higher layers of the two parallel streams, as illustrated in Fig. 1. In practice, we aggregate three of such feature representations as follows

$$\begin{aligned} \boldsymbol{f} &= \texttt{MLP}\left(\texttt{MaxPool}\left(\boldsymbol{f}_l, \boldsymbol{f}_{l'}, \boldsymbol{f}_{l''} \right) \right) \\ \text{s.t. } \boldsymbol{f}_l &= \texttt{MLP}\left(\texttt{Flatten}\left(\widetilde{\mathcal{S}}_l^{\mathcal{X},\mathcal{P}} \right) \right), \end{aligned} \tag{3}$$

where $\texttt{Flatten}(\cdot)$ denotes a flattening operation that reforms the feature tensor $\widetilde{\mathcal{S}}_l^{\mathcal{X},\mathcal{P}}$ of dimension $W \times H \times d_l$ as a feature vector, MLP denotes a subnetwork of Multi-Layer Perceptron (MLP), and $\texttt{MaxPool}\left(\boldsymbol{f}_l, \boldsymbol{f}_{l'}, \boldsymbol{f}_{l''} \right)$ aggregates the three feature vectors by max-pooling over three entries for each feature channel; layer specifics of the two MLPs used in (3) are given in Fig. 1. We use $\boldsymbol{f}$ computed from (3) as the final output of the pose encoder Φ, i.e., $\boldsymbol{f} = \Phi(\mathcal{X}, \mathcal{P})$.

4.3. The Explicit Pose Decoder Ψ_{exp}

Given $\boldsymbol{f}$ from the encoder Φ, we implement the explicit decoder Ψ_{exp} simply as three parallel MLPs that are trained to directly regress the rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, and size $\boldsymbol{s}$. Fig. 1 gives the illustration, where layer specifics of the three MLPs are also given. This gives a direct way of pose prediction from a cropped RGB-D region as $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{s}) = \Psi_{exp} \circ \Phi(\mathcal{X}, \mathcal{P})$.

4.4. The Implicit Pose Decoder Ψ_{im}

For the observed point cloud $\mathcal{P}$, assume that its counterpart $\mathcal{Q}$ in the canonical pose is available. An affine transformation $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{s})$ between $\mathcal{P}$ and $\mathcal{Q}$ can be established, which computes $\boldsymbol{q} = \frac{1}{||\boldsymbol{s}||}\boldsymbol{R}^T(\boldsymbol{p} - \boldsymbol{t})$ for any corresponding pair of $\boldsymbol{p} \in \mathcal{P}$ and $\boldsymbol{q} \in \mathcal{Q}$. This implies an implicit way of obtaining predicted pose by learning to predict a canonical $\mathcal{Q}$ from the observed $\mathcal{P}$; upon prediction of $\mathcal{Q}$, the pose $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{s})$ can be obtained by solving the alignment problem via Umeyama algorithm [28]. Since $\boldsymbol{f} = \Phi(\mathcal{X}, \mathcal{P})$ has learned the pose-sensitive features, we expect the corresponding $\boldsymbol{q}$ can be estimated from $\boldsymbol{p}$ by learning a mapping from the concatenation of $\boldsymbol{f}$ and $\boldsymbol{p}$. In DualPoseNet, we simply implement the learnable mapping as

$$\Psi_{im}(\boldsymbol{p}, \boldsymbol{f}) = \mathtt{MLP}\left([\boldsymbol{p}; \boldsymbol{f}]\right). \tag{4}$$

Ψ_{im} applies to individual points of $\mathcal{P}$ in a point-wise manner. We write collectively as $\mathcal{Q} = \Psi_{im}(\mathcal{P}, \boldsymbol{f})$.

We note that an equivalent representation of normalized object coordinate space (NOCS) is learned in [30] for a subsequent computation of pose prediction. Different from NOCS, we use Ψ_{im} in an implicit way; it has two benefits that potentially improve the pose prediction: (1) it provides an auxiliary supervision on the training of pose encoder Ψ (note that the training ground truth of $\mathcal{Q}$ can be transformed from $\mathcal{P}$ using the annotated pose and size), and (2) it enables a refined pose prediction by enforcing the consistency between the outputs of Ψ_{exp} and Ψ_{im}, as explained shortly in Section 4.6. We empirically verify both the benefits in Section 5.1.1, and show that the use of Ψ_{im} improves pose predictions of $\Psi_{exp} \circ \Phi(\mathcal{X}, \mathcal{P})$ in DualPoseNet.

4.5. Training of Dual Pose Network

Given the ground-truth pose annotation $(\boldsymbol{R}^*, \boldsymbol{t}^*, \boldsymbol{s}^*)$ [1] for a cropped $(\mathcal{X}, \mathcal{P})$, we use the following training objective on top of the explicit decoder Ψ_{exp}:

$$\mathcal{L}_{\Phi,\Psi_{exp}} = ||\rho(\boldsymbol{R})-\rho(\boldsymbol{R}^*)||_2+||\boldsymbol{t}-\boldsymbol{t}^*||_2+||\boldsymbol{s}-\boldsymbol{s}^*||_2, \tag{5}$$

where $\rho(\boldsymbol{R})$ is the quaternion representation of rotation $\boldsymbol{R}$.

Since individual points in the predicted $\mathcal{Q} = \{\boldsymbol{q}_i\}_{i=1}^N$ from Ψ_{im} are respectively corresponded to those in the observed $\mathcal{P} = \{\boldsymbol{p}_i\}_{i=1}^N$, we simply use the following loss on top of the implicit decoder

$$\mathcal{L}_{\Phi,\Psi_{im}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{q}_i - \frac{1}{||\boldsymbol{s}^*||}\boldsymbol{R}^{*\top}(\boldsymbol{p}_i - \boldsymbol{t}^*)\right\|_2. \tag{6}$$

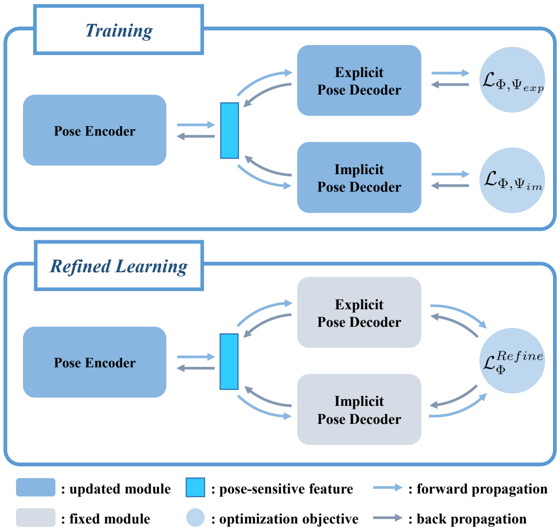

Figure 2. Illustrations on the training and refined learning of DualPoseNet. During training, we optimize the objective (7), which is a combination of $\mathcal{L}_{\Phi,\Psi_{exp}}$ and $\mathcal{L}_{\Phi,\Psi_{im}}$, in an end-to-end manner. During testing, we freeze the parameters of Ψ_{exp} and Ψ_{im}, and fine-tune those of Φ to minimize $\mathcal{L}_{\Phi}^{Refine}$ for pose consistency.

The overall training objective combines (5) and (6), resulting in the optimization problem

$$\min_{\Phi,\Psi_{exp},\Psi_{im}} \mathcal{L}_{\Phi,\Psi_{exp}} + \lambda\mathcal{L}_{\Phi,\Psi_{im}}, \tag{7}$$

where λ is a penalty parameter.

4.6. The Refined Learning of Pose Consistency

For instance-level 6D pose estimation, it is a common practice to refine an initial or predicted pose by a post-registration [2] or post-optimization [18]; such a practice is possible since CAD model of the instance is available, which can guide the refinement by matching the CAD model with the (RGB and/or point cloud) observations. For our focused category-level problem, however, CAD models of testing instances are not provided. This creates a challenge in case that more precise predictions on certain testing instances are demanded.

Thanks to the dual pose predictions from Ψ_{exp} and Ψ_{im}, we are able to make a pose refinement by learning to enforce their pose consistency. More specifically, we freeze the parameters of Ψ_{exp} and Ψ_{im}, while fine-tuning those of the encoder Φ, by optimizing the following problem

$$\min_{\Phi} \mathcal{L}_{\Phi}^{Refine} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{q}_i - \frac{1}{||\boldsymbol{s}||}\boldsymbol{R}^{\top}(\boldsymbol{p}_i - \boldsymbol{t})\right\|_2, \tag{8}$$

[1] Following [27], we use canonical $\boldsymbol{R}^*$ for symmetic objects to handle ambiguities of symmetry.

where $\mathcal{Q} = \{\boldsymbol{q}_i\}_{i=1}^N = \Psi_{im} \circ \Phi(\mathcal{X}, \mathcal{P})$ and $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{s}) = \Psi_{exp} \circ \Phi(\mathcal{X}, \mathcal{P})$ are the outputs of the two decoders. Note that during training, the two decoders are consistent in terms of pose prediction, since both of them are trained to match their outputs with the ground truths. During testing, due to an inevitable generalization gap, inconsistency between outputs of the two decoders always exists, and our proposed refinement (8) is expected to close the gap. An improved prediction relies on a better pose-sensitive encoding $\boldsymbol{f} = \Phi(\mathcal{X}, \mathcal{P})$; the refinement (8) thus updates parameters of Φ to achieve the goal. Empirical results in Section 5.1.1 verify that the refined poses are indeed towards more precise ones. In practice, we set a loss tolerance ϵ as the stopping criterion when fine-tuning $\mathcal{L}_{\Phi}^{Refine}$ (*i.e.*, the refinement stops when $\mathcal{L}_{\Phi}^{Refine} \leq \epsilon$), with fast convergence and negligible cost.

5. Experiments

Datasets We conduct experiments using the benchmark CAMERA25 and REAL275 datasets [30] for category-level 6D object pose and size estimation. CAMERA25 is a synthetic dataset generated by a context-aware mixed reality approach from 6 object categories; it includes $300,000$ composite images of $1,085$ object instances, among which $25,000$ images of 184 instances are used for evaluation. REAL275 is a more challenging real-world dataset captured with clutter, occlusion and various lighting conditions; its training set contains $4,300$ images of 7 scenes, and the test set contains $2,750$ images of 6 scenes. Note that CAMERA25 and REAL275 share the same object categories, which enables a combined use of the two datasets for model training, as done in [30, 27].

We also evaluate the advantages of DualPoseNet on the benchmark instance-level object pose datasets of YCB-Video [3] and LineMOD [13], which consist of 21 and 13 different object instances respectively.

Implementation Details We employ a MaskRCNN [11] implemented by [1] to segment out the objects of interest from input scenes. For each segmented object, its RGB-D crop is converted as spherical signals with a sampling resolution 64×64, and is then fed into our DualPoseNet. Configurations of DualPoseNet, including channel numbers of spherical convolutions and MLPs, have been specified in Fig. 1. We use ADAM to train DualPoseNet, with an initial learning rate of 0.0001. The learning rate is halved every $50,000$ iterations until a total number of $300,000$ ones. We set the batch size as 64, and the penalty parameter in Eq. (7) as $\lambda = 10$. For refined learning of pose consistency, we use a learning rate 1×10^{-6} and a loss tolerance $\epsilon = 5 \times 10^{-5}$. For the instance-level task, we additionally adopt a similar 2nd-stage iterative refinement of residual pose as [29, 34] did; more details are shown in the supplementary material.

Evaluation Metrics For category-level pose estimation, we follow [30] to report mean Average Precision (mAP) at different thresholds of intersection over union (IoU) for object detection, and mAP at n° m cm for pose estimation. However, those metrics are not precise enough to simultaneously evaluate 6D pose and object size estimation, since IoU alone may fail to characterize precise object poses (a rotated bounding box may give a similar IoU value). To evaluate the problem nature of simultaneous predictions of pose and size, in this work, we also propose a new and more strict metric based on a combination of *IoU*, *error of rotation*, and *error of relative translation*, where for the last one, we use the relative version since absolute translations make less sense for objects of varying sizes. For the three errors, we consider respective thresholds of $\{50\%, 75\%\}$ (*i.e.*, IoU$_{50}$ and IoU$_{75}$), $\{5^\circ, 10^\circ\}$, and $\{5\%, 10\%, 20\%\}$, whose combinations can evaluate the predictions across a range of precisions. For instance-level pose estimation, we follow [29] and evaluate the results of YCB-Video and LineMOD datasets by ADD-S and ADD(S) metrics, respectively.

5.1. Category-level 6D Pose and Size Estimation

5.1.1 Ablation Studies and Analyses

We first conduct ablation studies to evaluate the efficacy of individual components proposed in DualPoseNet. These studies are conducted on the REAL275 dataset [30].

We use both Ψ_{exp} and Ψ_{im} for pose decoding from DualPoseNet; Ψ_{exp} produces the pose predictions directly, which are also used as the results of DualPoseNet both with and without the refined learning, while Ψ_{im} is an implicit one whose outputs can translate as the results by solving an alignment problem. To verify the usefulness of Ψ_{im}, we report the results of DualPoseNet with or without the use of Ψ_{im} in Table 1, in terms of the pose precision from Ψ_{exp} *before the refined learning*. We observe that the use of Ψ_{im} improves the performance of Ψ_{exp} by large margins under all the metrics; for example, the mAP improvement of (IoU$_{50}, 10^\circ, 10\%$) reaches 5.8%, and that of (IoU$_{75}, 5^\circ, 10\%$) reaches 4.1%. These performance gains suggest that Ψ_{im} not only enables the subsequent refined learning of pose consistency, but also provides an auxiliary supervision on the training of pose encoder Φ and results in a better pose-sensitive embedding, implying the key role of Ψ_{im} in DualPoseNet.

To evaluate the efficacy of our proposed spherical fusion based encoder Φ, we compare with three alternative encoders: (1) a baseline of **Densefusion** [29], a pose encoder that fuses the learned RGB features from CNNs and point features from PointNet [23] in a point-wise manner; (2) **SCNN-EarlyFusion**, which takes as input the concatenation of $\mathcal{S}^{\mathcal{X}}$ and $\mathcal{S}^{\mathcal{P}}$ and feeds it into a multi-scale spherical CNN, followed by an MLP; (3) **SCNN-LateFusion**, which first feeds $\mathcal{S}^{\mathcal{X}}$ and $\mathcal{S}^{\mathcal{P}}$ into two separate multi-scale spherical CNNs and applies an MLP to the concatenation

Encoder	Ψ_{im}	Refining	mAP											
			IoU$_{75}$ 5°, 5%	IoU$_{75}$ 10°, 5%	IoU$_{75}$ 5°, 10%	IoU$_{50}$ 5°, 20%	IoU$_{50}$ 10°, 10%	IoU$_{50}$ 10°, 20%	IoU$_{50}$	IoU$_{75}$	5° 2cm	5° 5cm	10° 2cm	10° 5cm
Densefusion [29]	✓	×	1.5	3.0	7.9	11.4	17.4	26.1	64.9	35.0	9.1	15.6	19.3	36.2
SCNN-EarlyFusion	✓	×	7.7	14.4	15.8	20.3	35.5	45.8	76.1	51.9	17.3	24.5	36.2	56.8
SCNN-LateFusion	✓	×	8.4	14.7	23.8	28.5	41.7	51.4	77.0	56.6	25.7	34.3	43.5	62.8
Φ	×	×	8.2	13.6	19.7	26.1	37.3	49.1	76.1	55.2	21.3	31.3	38.5	60.4
Φ	✓	×	10.4	16.1	23.8	28.5	43.1	52.6	79.7	60.1	28.0	34.3	47.8	64.2
Φ	✓	✓	**11.2**	**17.2**	**24.8**	**29.8**	**44.5**	**55.0**	**79.8**	**62.2**	**29.3**	**35.9**	**50.0**	**66.8**

Table 1. Ablation studies on variants of our proposed DualPoseNet on REAL275. Evaluations are based on both our proposed metrics (left) and the metrics (right) proposed in [30].

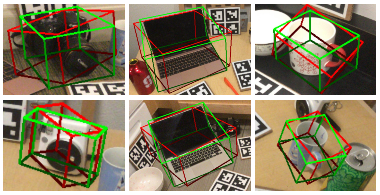

Figure 3. Qualitative results of DualPoseNet without (red) and with (green) the refined learning of pose consistency on REAL275.

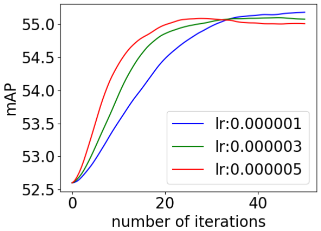

Figure 4. Plottings of prediction accuracy (mAP of (IoU$_{50}$, 10°, 20%)) versus the number of iterations when using different learning rates to fine-tune the loss (8) for the refined learning of pose consistency. Experiments are conducted on REAL275 [30].

of the two output features. The used multi-scale spherical CNN is constructed by 8 spherical convolution layers, with aggregation of multi-scale spherical features similar to Φ. We conduct ablation experiments by replacing Φ with the above encoders, while keeping Ψ_{exp} and Ψ_{im} as remained. Results (without the refined learning of pose consistency) in Table 1 show that the three alternative encoders perform worse than our proposed Φ with spherical fusion. Compared with the densefusion baseline, those based on spherical convolutions enjoy the property of rotation equivariance on $SO(3)$, and thus achieve higher mAPs. With spherical fusion, our proposed pose encoder Φ enables information communication progressively along the hierarchy, outperforming either SCNN-EarlyFusion with feature fusion at the very beginning or SCNN-LateFusion at the end.

We finally investigate the benefit from the proposed refined learning of pose consistency. Results in Table 1 show that with the refined learning, pose precisions improve stably across the full spectrum of evaluation metrics, and the improvements increase when coarser metrics are used; this suggests that the refinement indeed attracts the learning towards more accurate regions in the solution space of pose prediction. Examples in Fig. 3 give corroborative evidence of the efficacy of the refined learning. In fact, the refinement process is a trade-off between the improved precision and refining efficiency. As mentioned in Section 4.6, the refining efficiency depends on the learning rate and number of iterations for fine-tuning the encoder with the objective (8). In Fig. 4, we plot curves of mAP of (IoU$_{50}$, 10°, 20%) against the number of iterations when using different learning rates. It shows faster convergence when using larger learning rates, which, however, may end with less mature final results, even with overfitting. In practice, one may set a proper tolerance ϵ for a balanced efficiency and accuracy. We set the learning rate as 1×10^{-6} and $\epsilon = 5 \times 10^{-5}$ for results of DualPoseNet reported in the present paper. Under this setting, it costs negligible 0.2 seconds per instance on a server with Intel E5-2683 CPU and GTX 1080ti GPU.

5.1.2 Comparisons with Existing Methods

We compare our proposed DualPoseNet with the existing methods, including NOCS [30], SPD [27], and CASS [5], on the CAMERA25 and REAL275 [30] datasets. Note that NOCS and SPD are designed to first predict canonical versions of the observed point clouds, and the poses are obtained from post-alignment by solving a Umeyama algorithm [28]. Quantitative results in Table 2 show the superiority of our proposed DualPoseNet on both datasets, especially for the metrics of high precisions. For completeness, we also present in Table 2 the comparative results under the original evaluation metrics proposed in [30]; our results are better than existing ones at all but one rather coarse metric of IoU$_{50}$, which is in fact a metric less sensitive to object pose. Qualitative results of different methods are shown in Fig. 5. Comparative advantages of our method over the existing ones are consistent with those observed in Table 2. For example, the bounding boxes of laptops in the figure generated by NOCS and SPD are obviously bigger than the exact extensions of laptops, while our method predicts more compact bounding boxes with precise poses and sizes. More comparative results are shown in the supplementary material.

5.2. Instance-level 6D Pose Estimation

We apply DualPoseNet to YCB-Video [3] and LineMOD [13] datasets for the instance-level task. Results in Ta-

Dataset	Method	mAP											
		IoU$_{75}$ 5°, 5%	IoU$_{75}$ 10°, 5%	IoU$_{75}$ 5°, 10%	IoU$_{50}$ 5°, 20%	IoU$_{50}$ 10°, 10%	IoU$_{50}$ 10°, 20%	IoU$_{50}$	IoU$_{75}$	5° 2cm	5° 5cm	10° 2cm	10° 5cm
CAMERA25	NOCS [30]	22.6	29.5	31.5	34.5	54.5	56.8	83.9	69.5	32.3	40.9	48.2	64.6
	SPD [27]	47.5	61.5	52.2	56.6	75.3	78.5	**93.2**	83.1	54.3	59.0	73.3	81.5
	DualPoseNet	**56.2**	**65.1**	**65.1**	**68.0**	**78.6**	**81.5**	92.4	**86.4**	**64.7**	**70.7**	**77.2**	**84.7**
REAL275	NOCS [30]	2.4	3.5	7.1	9.3	19.7	22.3	78.0	30.1	7.2	10.0	13.8	25.2
	SPD [27]	8.6	17.2	15.0	17.4	38.5	42.5	77.3	53.2	19.3	21.4	43.2	54.1
	CASS [5]	–	–	–	–	–	–	77.7	–	–	23.5	–	58.0
	DualPoseNet	**11.2**	**17.2**	**24.8**	**29.8**	**44.5**	**55.0**	**79.8**	**62.2**	**29.3**	**35.9**	**50.0**	**66.8**

Table 2. Quantitative comparisons of different methods on CAMERA25 and REAL275. Evaluations are based on both our proposed metrics (left) and the metrics (right) proposed in [30].

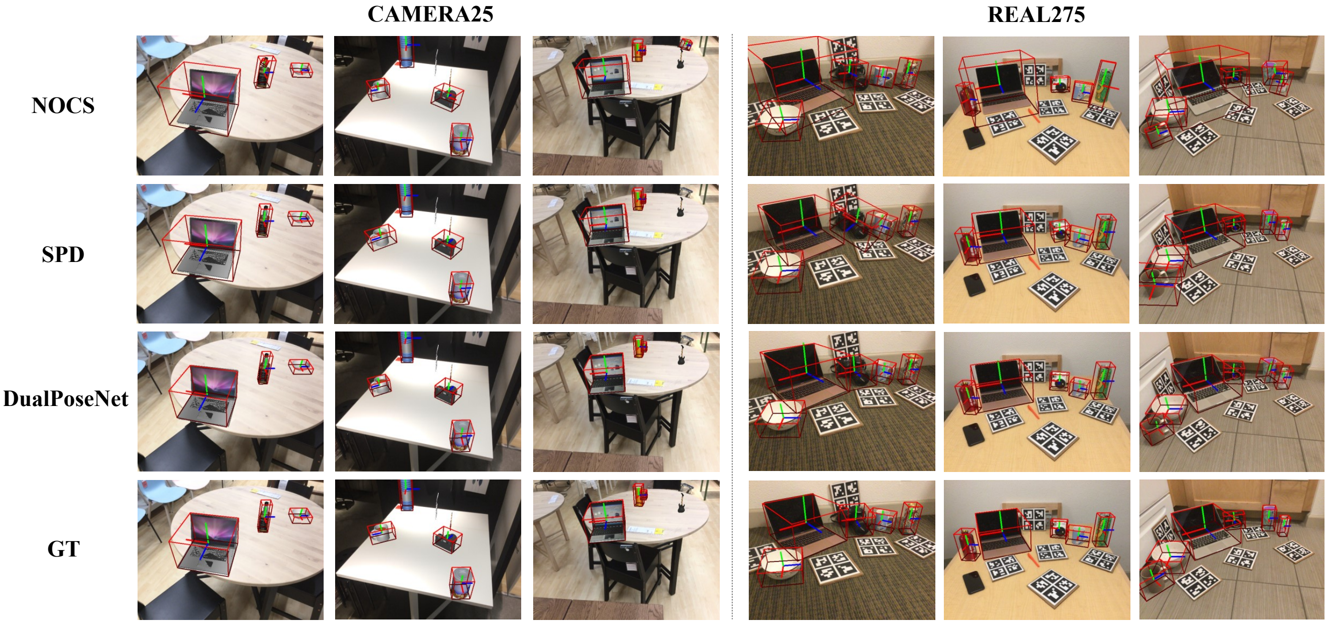

Figure 5. Qualitative results of different methods on CAMERA25 and REAL275 [30].

Encoder	Ψ_{im}	Refining	Iterative	YCB-Video	LineMOD
Densefusion [29]	×	×	×	88.2	78.7
Φ	×	×	×	90.5	88.6
Φ	✓	×	×	91.2	92.7
Φ	✓	✓	×	93.3	94.6
Φ	✓	×	✓	95.0	96.3
Φ	✓	✓	✓	**96.5**	**98.2**

Table 3. Ablation studies on variants of DualPoseNet on YCB-Video [3] and LineMOD [13] datasets for instance-level 6D pose estimation. The evaluation metrics are mean ADD-S AUC and mean ADD(S) AUC, respectively.

ble 3 confirm the efficacy of our individual components (the encoder Φ, the implicit decoder Ψ_{im}, and the refined learning of pose consistency); following [29, 34], we also augment our DualPoseNet with a 2nd-stage module for iterative refinement of residual pose, denoted as DualPoseNet(Iterative), to further improve the performance. As shown in Table 4, DualPoseNet(Iterative) achieves comparable results against other methods, showing its potential for use in instance-level tasks. More quantitative and qualitative results are shown in the supplementary material.

Method	YCB-Video	LineMOD
Pointfusion [33]	83.9	73.7
PoseCNN + ICP [32]	93.0	–
Densefusion (Per-pixel) [29]	91.2	86.2
Densefusion (Iterative) [29]	93.1	94.3
$\mathcal{W}$-PoseNet [34]	93.0	97.2
$\mathcal{W}$-PoseNet (Iterative)[34]	94.0	98.1
DualPoseNet	93.3	94.6
DualPoseNet (Iterative)	**96.5**	**98.2**

Table 4. Quantitative comparisons of different methods on YCB-Video [3] and LineMOD [13] datasets for instance-level 6D pose estimation. The evaluation metrics are mean ADD-S AUC and mean ADD(S) AUC, respectively.

6. Acknowledgement

This work was partially supported by the Guangdong R&D key project of China (No.: 2019B010155001), the National Natural Science Foundation of China (No.: 61771201), and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.: 2017ZT07X183).

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 6

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 1, 2, 5

[3] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 2, 6, 7, 8

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[5] Dengsheng Chen, Jun Li, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 7, 8

[6] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *European Conference on Computer Vision*, pages 139–156. Springer, 2020. 2

[7] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical cnns. In *ICLR*, 2018. 2, 4

[8] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010. 1, 2

[9] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018. 2, 4

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 3, 6

[12] Stefan Hinterstoisser, Cédric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient Response Maps for Real-Time Detection of Texture-Less Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, 2012. 1, 2

[13] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011. 2, 6, 7, 8

[14] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *ECCV*, 2016. 1, 2

[15] Qi-Xing Huang, Hao Su, and Leonidas Guibas. Fine-grained semi-supervised labeling of large shape collections. *ACM Transactions on Graphics (TOG)*, 2013. 1

[16] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 1, 2

[17] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 1, 2

[18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 1, 2, 5

[19] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*, 2020. 2

[20] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 2

[21] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 2

[22] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 1, 2

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 6

[24] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1, 2

[25] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 1, 2

[26] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 1, 2

[27] Meng Tian, Marcelo H Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 2, 3, 4, 5, 6, 7, 8

[28] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):376–380, 1991. 4, 5, 7

[29] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1, 2, 6, 7, 8

[30] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[31] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 1, 2

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 1, 2, 8

[33] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 8

[34] Zelin Xu, Ke Chen, and Kui Jia. W-posenet: Dense correspondence regularized pixel pair pose regression. *arXiv preprint arXiv:1912.11888*, 2019. 6, 8

[35] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019. 1, 2

[36] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 1, 2