BiCo-Net: Regress Globally, Match Locally for Robust 6D Pose Estimation

Zelin Xu\(^1\), Yichen Zhang\(^1\), Ke Chen\(^{1,2,\ast}\) and Kui Jia\(^{1,2,\ast}\)

\(^1\)South China University of Technology
\(^2\)Peng Cheng Laboratory
{eexuzelin, eezyc}@mail.scut.edu.cn, {chenk, kuijia}@scut.edu.cn

Abstract

The challenges of learning a robust 6D pose function lie in 1) severe occlusion and 2) systematic noises in depth images. Inspired by the success of point-pair features, the goal of this paper is to recover the 6D pose of an object instance segmented from RGB-D images by locally matching pairs of oriented points between the model and camera space. To this end, we propose a novel Bi-directional Correspondence Mapping Network (BiCo-Net) to first generate point clouds guided by a typical pose regression, which can thus incorporate pose-sensitive information to optimize generation of local coordinates and their normal vectors. As pose predictions via geometric computation only rely on one single pair of local oriented points, our BiCo-Net can achieve robustness against sparse and occluded point clouds. An ensemble of redundant pose predictions from locally matching and direct pose regression further refines final pose output against noisy observations. Experimental results on three popularly benchmarking datasets can verify that our method can achieve state-of-the-art performance, especially for the more challenging severe occluded scenes. Source codes are available at https://github.com/Gorilla-Lab-SCUT/BiCo-Net.

1 Introduction

The problem of 6 Degree-of-Freedom pose estimation aims to predict the orientation and location of one detected object instance in 3D space from a canonical model via recovering a rigid transformation from the object space to the camera space. Such problem has been widely encountered in fields of engineering such as robotic grasping and autonomous driving. A large number of deep methods including regression based [Xiang \textit{et al.}, 2018; Park \textit{et al.}, 2019] and keypoint-based [Tekin \textit{et al.}, 2018; Peng \textit{et al.}, 2019] only rely on extracting texture information from RGB images, which are sensitive to objects with poor textures. Alternatively, with the development and wide application of depth sensors, exploring 6D pose estimation on RGB-D images becomes popular in recent years, where depth images can provide complementary geometry information to RGB images.

The pioneering works [Xiang \textit{et al.}, 2018; Li \textit{et al.}, 2018] on RGB-D images are in a two-stage structure: deep pose estimation and post-processing refinement with iterative closest point (ICP), which leads to less efficiency during inference. Residual learning based refinement in [Wang \textit{et al.}, 2019a] is proposed to improve efficiency with orders of magnitude faster than the ICP, which can ensure real-time inference. From a practical perspective, there remain two compounded challenges for learning a robust 6D pose function to address the problem: 1) severe occlusion and 2) unavoidable systematic noises during depth imaging [Barron and Malik, 2013].

Most of the existing methods [Wang \textit{et al.}, 2019a; He \textit{et al.}, 2021] concern on coping with the former by improving feature encoding on integrating textural and shape features into discriminative representation, while very few work pays attention to the latter. In [Zhou \textit{et al.}, 2021], a point refinement network is the first attempt to explicitly polish point clouds via completion and denoising, whose features are combined with those encoded from raw point clouds and RGB images for better multi-modal feature fusion to regress final poses. However, performance gain of PR-GCN in [Zhou \textit{et al.}, 2021]
over the pose regression baseline can be sensitive to point cloud generation, which itself remains an active and challenging task especially under incomplete and noisy observations.

We argue that encoding pose-sensitive local features and modeling a statistical distribution of pose inliers are two key factors for accurate and robust performance in 6D pose estimation. On one hand, pose estimation dependent on local texture and geometry can perform stably when missing a part of object regions, which is thus robust against occlusion, but those local features are sensitive to the quality (e.g. noises and resolution) of the acquired data. On the other hand, the distribution of pose predictions on local features can be explored to alleviate negative effects of systematic noises in depth imaging. In this paper, we propose a novel deep model for 6D pose estimation on RGB-D images, i.e. a Bi-directional Correspondence Mapping Network (BiCo-Net) as shown in Fig. 1, by simultaneously addressing both challenges in a unified and implicit manner.

Given the point cloud (generated from the depth image) and the RGB image of an object under the observation pose, the feature output of DenseFusion feature encoder [Wang et al., 2019a] is decoded to generate the corresponding oriented points in the canonical space. To effectively exploit object model priors as a reference, a clean and complete model point cloud under the observed pose is similarly produced in an encoder-decoder learning style from the corresponding oriented points under the canonical pose, which is in an opposite direction of the aforementioned point cloud generation. Inspired by the success of point-pair features [Drost et al., 2010], a set of oriented point pairs from the input (e.g. the camera space) can be randomly sampled to produce accurate pose predictions by matching with those corresponding ones in the output space (e.g. the model space). As pose prediction relies only on one pair of local oriented points, such a characteristic can encourage robustness against sparse and occluded point clouds. Moreover, owing to revealing a global distribution of rigid pose transformation favored by multiple point pairs, selecting and combining pose predictions of the bi-directional correspondence mapping can alleviate negative effects of outliers in noisy point clouds. For imposing pose-sensitive information, features of the bi-directional point cloud generation can be regularized by a typical pose regression as [Wang et al., 2019a].

In general, the whole network consists of an ensemble of two parts: direct pose regression on a concatenation of global features and point-wise features as [Wang et al., 2019a] and pose computation via locally matching of oriented point pairs in-between canonical and observed poses, which can further refine pose predictions. Extensive experiments on three popular benchmarks, i.e. YCB-Video [Xiang et al., 2018], LineMOD [Hinterstoisser et al., 2011] and Occlusion LineMOD [Brachmann et al., 2014], can verify superior performance of the proposed BiCo-Net to the state-of-the-art methods, especially for severe occluded scenes. Main contributions of this paper lie as follows:

- The proposed BiCo-Net is implicitly robust against occlusion and sparse point distribution owing to exploiting the pose-sensitive characteristic of each single pair of oriented points under different poses.
- Negative effects of outliers in noisy depth images can be mitigated via selection and an ensemble of redundant pose predictions.
- Our BiCo-Net can gain state-of-the-art performance on three benchmarks, especially on the more challenging Occlusion LineMOD dataset.

2 Related Works

Keypoint-based 6D Pose Estimation. A typical keypoint-based pose estimation algorithm is designed in a two-stage pipeline: first localizing 2D projection of pre-defined key points in 3D space, and pose predictions can be generated via 2D-to-3D key point correspondence with a PnP [Lepetit et al., 2008]. Existing methods can be categorized into two groups: object detection based [Rad and Lepetit, 2017; Tekin et al., 2018] and dense heatmap based [Oberweger et al., 2018; Pavlakos et al., 2017]. The former performs well on localizing sparse key points of object foreground, but are sensitive to occlusion [Oberweger et al., 2018]. The latter group of methods are more robust to inter-object occlusion in a dense correspondence mapping style. PVNet [Peng et al., 2019] is proposed to detect 2D keypoints via voting on pixel-wise predictions of the directional vector that points to keypoints and is robust to truncation and occlusion. PVN3D [He et al., 2020] extends the 2D keypoints into 3D space by building 3D-3D correspondence and then uses the least-squares fitting [Arun et al., 1987] to generate the pose prediction. Learning of directional vectors pointing to 3D keypoints under camera space suffers from the variation of object pose and the vast 3D search space. In contrast, the bi-directional correspondence mapping in our method makes point-wise predictions on 3D oriented points between the model and camera space directly to build up dense correspondence, instead of their projection in 2D images or the directional vector pointing to 3D keys, which enhances local feature discrimination via direct regression based regularization in a pose-sensitive manner.

Dense Regression-based 6D Pose Estimation. An alternative group of algorithms to cope with occlusion is to produce dense pose predictions for each pixel or local patches with hand-crafted features [Liebelt et al., 2008; Sun et al., 2010], CNN patch-based feature encoding [Doumanoglou et al., 2016; Kehl et al., 2016] and CNN pixel-based feature encoding [Wang et al., 2019a; Zhou et al., 2021], whose final pose output is selected via a voting scheme. DenseFusion [Wang et al., 2019a] fuses RGB and Depth information for each point and uses point-wise features to regress dense poses. Zhou et al. proposes PR-GCN [Zhou et al., 2021] to handle incomplete and noisy point clouds in practice via designing a PRN to complete and denoise observed point clouds and use graph convolution to better integrated the RGB and Depth information. In [Wang et al., 2019b], normalized object coordinate space is proposed to build up 3D-3D correspondences for each pixel in category level then recover the
6D pose and size by the least-squares fitting. The proposed BiCo-Net is designed to both regress dense poses from point-wise features and locally match pose-sensitive oriented points in a unified framework. The complementary characteristics of dense pose regression and dense correspondence mapping can be fully utilized to gain robust pose predictions. Moreover, compared with the least-squares fitting, our point-pair pose computation which uses the correspondence of two oriented points can perform more robust under heavy occlusion.

3 Methodology

The problem of 6D pose estimation of an object given the RGB-D image and its canonical CAD model is to estimate their rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$, which can be defined as a rigid transformation $T = [R; t]$ from the model space with respect to the camera space. The whole pipeline of our BiCo-Net is illustrated in Figure 2, which can be divided into three modules: feature encoding on segmented object instances (see Sec. 3.1) in a blue block, generation of oriented points via Bi-directional Correspondence Mapping (see Sec. 3.2) in a green block, and an ensemble of pose predictions by point pair matching and point-wise regression (see Sec. 3.3) in a yellow block.

3.1 Instance Segmentation and Feature Encoding

Following existing methods [Wang et al., 2019a; Zhou et al., 2021], BiCo-Net first employs an off-the-shelf instance segmentation method for RGB images (e.g. Mask RCNN [He et al., 2017] in our experiments) to segment the object of interests, which produces a cropped image patch $I$ and one scene point cloud under the camera space $S = \{(s_i, n_i^s) \in \mathbb{R}^6\}_{i=1}^N$, where $s_i \in \mathbb{R}^3$ denotes 3D coordinates converted from the masked depth region and $n_i^s \in \mathbb{R}^3$ denotes its normal vectors physically computed by PCA. Both $I$ and $S$ are fed into CNN-based and MLP-based feature encoders respectively to extract texture and geometric features from heterogeneous data sources, which are further fused by the DenseFusion module introduced in [Wang et al., 2019a] to obtain point-wise features $F^S = \{f_i \in \mathbb{R}^{1024}\}_{i=1}^N$. Similarly, for exploiting the model priors, we randomly sample one clean model point cloud $\hat{m}$ from the canonical CAD model and compute point-wise normal vectors $\hat{n}^m$ to form $\hat{M} = \{(\hat{m}_i, \hat{n}^m_i) \in \mathbb{R}^{6} \}_{i=1}^M$ as input of an MLP-based feature encoder to generate point-wise features $F^{\hat{M}} = \{f_i^m \in \mathbb{R}^{512}\}_{i=1}^M$.

3.2 Bi-directional Correspondence Mapping

Given features $F^S$ from visual observation $I$ and $S$, the BCM-scene (BCM-S) in the top row of Figure 2 aims to regress the corresponding oriented point cloud under the model space $\hat{M} = \{(m_i, n^m_i) \in \mathbb{R}^{6}\}_{j=1}^N$, where $m_i = R^{-1}(s_i - t)$ and $n^m_i = R^{-1}n^s_i$. With the point-wise features $F^S$ as input, we employ a MLP with $[512, 1024]$ neurons and an average pooling (AvgPool) to generate a global feature $g^S = \text{AvgPool}1\text{MLP}_{S}(F^S) \in \mathbb{R}^{1024}$. Finally, $\hat{M} = \text{BCM}_S(F^S, g^S)$ for superior robustness to only using $F^S$, where generated points $\hat{M} = \{(\hat{m}_i, \hat{n}^m_i)\}_{i=1}^N$.

Similarly, the BCM-model (BCM-M) in the bottom row of Figure 2 aims to reconstruct a clean point cloud under the camera space $\hat{S} = \{(\hat{s}_j, \hat{n}^s_j) \in \mathbb{R}^{6}\}_{j=1}^M$ from the features under the model space $F^{\hat{M}}$, where $\hat{s}_j = R\hat{m}_j + t$ and $\hat{n}^s_j = R\hat{n}^m_j$. To this end, a global feature $g^M = \text{AvgPool}1\text{MLP}_{M}(F^{\hat{M}})$ is generated from points aggregated from $F^S$ together with $F^{\hat{M}}$. Learning to regress $\hat{S}$, while $\hat{M} = \text{BCM}_M(F^{\hat{M}}, g^M)$ is the generated point cloud under the camera space by the BCM-M branch. Moreover, to impose pose sensitive information for generation of point clouds in the BCM-S and BCM-M, a direct point-wise pose regression (PR) in the middle row of Figure 2 on $F^S$, $g^M$, and $p^S$ to predict $\hat{r}^{PR} = \text{PoseReg}(F^S, g^M, p^S)$, where $\hat{r}^{PR} = \{(\hat{r}_i, \hat{p}_i)\}_{i=1}^N$ and $g^M = \text{AvgPool}1\text{MLP}_{M}(F^{\hat{M}})$.

Loss Functions. We use the Euclidean distance to supervise both BCM branches as follows:

$$L_{\text{BCM-S}} = \frac{1}{N} \sum_{i} (\|m_i - \hat{m}_i\|^2 + \lambda \|n^m_i - \hat{n}^m_i\|^2)$$

$$L_{\text{BCM-M}} = \frac{1}{M} \sum_{j} (\|\hat{s}_j - \hat{s}_j\|^2 + \lambda \|\hat{n}^s_j - \hat{n}^s_j\|^2)$$

where $\lambda$ is a trade-off parameter, $(m_i, n^m_i)$ and $(\hat{s}_j, \hat{n}^s_j)$ are ground truth oriented points of the BCM-S and BCM-M.
branches, while \((\hat{m}_i, \hat{n}_i), (\hat{s}_j, \hat{z}_j)\) are the generated points. To ensure the pose consistency between BCM branches and direct regression branch, we replace the ground truth pose in \((m_i, n_i), (s_j, z_j)\) with the mean of point-wise predicted pose \([\hat{R}_i, \hat{t}_i]\) for symmetric objects. For supervising \(T^{PR}\) with \(T = [R_t, t_t]\) in the pose regression branch, we use the ADD Loss [Xiang et al., 2018] for asymmetric objects and ADD-S Loss for symmetric objects:

\[
L_{PR} = \left\{ \begin{array}{ll}
\frac{1}{K} \sum_k \| [(R_x k + t) - (\hat{R}_i x_k + \hat{t}_i)] \| & \text{if asym.}
\frac{1}{K} \sum_{i=0}^{K} \min_{0<i<K} \| [(R_x k + t) - (\hat{R}_i x_i + \hat{t}_i)] \| & \text{if sym.}
\end{array} \right.
\]

where \(K\) is the number of points sampled from the surface of the CAD model, \([R_t, t_t]\) and \([\hat{R}_i, \hat{t}_i]\) are ground truth and point-wise predicted poses respectively. The total loss of our BiCo-Net can thus be written as:

\[
L_{Total} = \frac{1}{N} \sum_i L_{PR} + L_{BCM-S} + L_{BCM-M}.
\]

3.3 Point Pair Matching and Prediction Ensemble

With generated point clouds \(\mathcal{M}\) and \(\mathcal{S}\) under the model and camera space respectively, our goal is to learn a rigid transformation \(T\) or \(T^{-1}\) between camera and model space. Encouraged by Point-pair feature (PPF) [Drost et al., 2010] to describe object poses by matching local features generated from oriented point pairs, any pair of oriented points in \(\mathcal{M}\) and \(\mathcal{S}\) can be written as a transformation \(\hat{m}_i\) to \(\hat{s}_i\) is defined as the following [Drost et al., 2010]:

\[
s_i = T_{s_i \rightarrow z}^{-1} R_x (\alpha) T_{m_i \rightarrow x} \hat{m}_i,
\]

where \(T_{m_i \rightarrow x}\) denotes a transformation that translates \(m_i\) into the origin and rotates \(n_i\) on to the \(x\)-axis, and the same definition of \(T_{s_i \rightarrow z}\) for transforming \((s_i, n_i)\). When \(m_i\) and \(s_i\) are aligned to the \(x\)-axis, there is a \(\alpha\) angle difference about the \(x\)-axis between \(T_{s_i \rightarrow x} s_i\) and \(T_{m_i \rightarrow x} \hat{m}_i\), which encourages to use \(R_x (\alpha)\), a rotation with respect to the \(x\)-axis, to align \(T_{s_i \rightarrow x} s_i\) and \(T_{m_i \rightarrow x} \hat{m}_i\). As a result, the rigid transformation \(\Phi: (s_i, s_i, \hat{m}_i, \hat{m}_i) \rightarrow T\) can be written as:

\[
\Phi(s_i, s_i, \hat{m}_i, \hat{m}_i) = T_{s_i \rightarrow z}^{-1} R_x (\alpha) T_{m_i \rightarrow x}.
\]

Note that, 6D object pose by locally matching a pair of oriented points under the camera and model space can be readily computed for desirable real-time inference.

As our method only relies on each single point pair to estimate 6D object pose, it allows sparse and imbalanced point distributions, which is thus able to achieve good performance for severe occlusion. For alleviating noises in point clouds and increasing inference speed, we use the FPS algorithm to downsample \(\mathcal{S}\) and \(\mathcal{M}\) to a subset of \(Z\) points, which are then to generate \(Z^2\) point pairs to compute pair-wise pose candidates \(T_z = [R_z, t_z]\) for the BCM-S and BCM-M respectively. For avoiding unreliable pose candidates from point-pairs constructed by neighboring points, pose predictions will be filtered out with the following error measure:

\[
E(T_z) = \frac{1}{N} \sum_i \| (R_z^{-1}(s_i - t_z)) - \hat{m}_i \|,
\]

and preserving the top 10% of candidates as the pose prediction sets \(T_{BCM-S}\) and \(T_{BCM-M}\) of these two branches.

An Ensemble of Pose Predictions. As mentioned in Sec. 1, we obtain three sets of pose predictions \(i.e. T^{PR}\) from direct pose regression; \(T^{BCM-S}\) and \(T^{BCM-M}\) via locally matching with point pairs), from three branches of the BiCo-Net. For achieving superior robustness via using the complementary information of three sets, we consider applying the average pose of \(T^{PR} \cup T^{BCM-S} \cup T^{BCM-M}\) as the final pose output.

4 Experiments

4.1 Datasets and Settings

Datasets. To evaluate our BiCo-Net comprehensively, experiments are conducted on three popular benchmarks – the YCB-Video dataset [Xiang et al., 2018], the LineMOD [Hinterstoisser et al., 2011], and the more challenging Occlusion LineMOD [Brachmann et al., 2014]. The YCB-Video dataset has 92 videos in total, each of which shows a subset of 21 objects placed under different cluttered environments. We use the standard training/testing split as [Wang et al., 2019a; Zhou et al., 2021], we adopt 16,189 frames from 80 videos with an additional 80,000 synthetic images provided by [Xiang et al., 2018] for training and extract 2949 key frames from the remaining 12 videos for testing. The LineMOD contains 15,783 images belonging to 13 low-textural objects placed under different cluttered environments. We use the standard training/testing split as [Xiang et al., 2018; Wang et al., 2019a]. The Occlusion LineMOD provides 6D pose labels of 8 objects selected from the LineMOD and includes 1214 images with multiple heavily occluded objects, which is made more challenging.

Performance Metrics. Following [Wang et al., 2019a; Zhou et al., 2021], we adopt the average distance (ADD) [Xiang et al., 2018] and ADD-Symmetric (ADD-S) as performance metrics. 6D pose predictions are considered to be correct if the ADD/ADD-S is smaller than a predefined threshold. For the YCB-Video dataset, we vary from 0 to 10cm to plot an accuracy-threshold curve and report the area under the curve (AUC). We also report the result of ADD-S < 2cm as [Wang et al., 2019a; Zhou et al., 2021]. For the LineMOD and the Occlusion LineMOD, we use ADD-S for symmetric objects \(i.e.\) eggbox and glue and ADD for the remaining objects having an asymmetric geometry while taking 10% of the diameter as the threshold.

4.2 Implementation Details

The numbers of scene/model points, \(i.e., N/M\), are set to 1000/1000. In point-pair pose computation, we downsample the scene points and model points to \(100\) points by the
Table 1: Comparison of AUC (%) and ADD-S < 2cm ("<"2cm) for short) on the YCB-Video dataset. Symmetric objects are highlighted in bold. Comparative methods with the proposed BiCo-Net are PoseCNN+ICP [Xiang et al., 2018], DenseFusion [Wang et al., 2019a], G2L-Net [Chen et al., 2020], PVN3D [He et al., 2020] and PR-GCN [Zhou et al., 2021].

<table>
<thead>
<tr>
<th>Method</th>
<th>PoseCNN+ICP</th>
<th>DenseFusion</th>
<th>G2L-Net</th>
<th>PVN3D</th>
<th>PR-GCN</th>
<th>BiCo-Net (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC &lt;2cm</td>
<td>AUC &lt;2cm</td>
<td>AUC &lt;2cm</td>
<td>AUC &lt;2cm</td>
<td>AUC &lt;2cm</td>
<td>AUC &lt;2cm</td>
</tr>
<tr>
<td>002_master_chef_can</td>
<td>95.8</td>
<td>100.0</td>
<td>96.4</td>
<td>100.0</td>
<td>94.0</td>
<td>96.0</td>
</tr>
<tr>
<td>003_cracker_box</td>
<td>92.7</td>
<td>91.6</td>
<td>95.5</td>
<td>99.5</td>
<td>88.7</td>
<td>96.1</td>
</tr>
<tr>
<td>004_sugar_box</td>
<td>98.2</td>
<td>100.0</td>
<td>97.5</td>
<td>100.0</td>
<td>96.0</td>
<td>97.4</td>
</tr>
<tr>
<td>005_tomato_soup_can</td>
<td>94.5</td>
<td>99.9</td>
<td>94.6</td>
<td>96.9</td>
<td>86.4</td>
<td>96.2</td>
</tr>
<tr>
<td>006_mustard_bottle</td>
<td>98.6</td>
<td>100.0</td>
<td>97.2</td>
<td>100.0</td>
<td>95.9</td>
<td>97.5</td>
</tr>
<tr>
<td>007_tuna_fish_can</td>
<td>97.1</td>
<td>100.0</td>
<td>96.6</td>
<td>100.0</td>
<td>84.1</td>
<td>96.0</td>
</tr>
<tr>
<td>008_pudding_box</td>
<td>97.9</td>
<td>100.0</td>
<td>96.5</td>
<td>100.0</td>
<td>93.5</td>
<td>97.1</td>
</tr>
<tr>
<td>009_geleatn_box</td>
<td>98.8</td>
<td>100.0</td>
<td>98.1</td>
<td>100.0</td>
<td>96.8</td>
<td>97.7</td>
</tr>
<tr>
<td>010_potted_meat_can</td>
<td>92.7</td>
<td>93.6</td>
<td>91.3</td>
<td>93.1</td>
<td>86.2</td>
<td>93.3</td>
</tr>
<tr>
<td>011_banana</td>
<td>97.1</td>
<td>99.7</td>
<td>96.6</td>
<td>100.0</td>
<td>96.3</td>
<td>96.6</td>
</tr>
<tr>
<td>019_pitcher_base</td>
<td>97.8</td>
<td>100.0</td>
<td>97.1</td>
<td>100.0</td>
<td>97.1</td>
<td>97.9</td>
</tr>
<tr>
<td>021_bleach_cleaner</td>
<td>96.9</td>
<td>99.4</td>
<td>95.8</td>
<td>100.0</td>
<td>92.0</td>
<td>96.0</td>
</tr>
<tr>
<td>024_bowl</td>
<td>81.0</td>
<td>54.9</td>
<td>88.2</td>
<td>98.8</td>
<td>86.7</td>
<td>90.2</td>
</tr>
<tr>
<td>025_mug</td>
<td>95.0</td>
<td>99.8</td>
<td>97.1</td>
<td>100.0</td>
<td>95.4</td>
<td>97.6</td>
</tr>
<tr>
<td>035_power_drill</td>
<td>98.2</td>
<td>99.6</td>
<td>96.0</td>
<td>98.7</td>
<td>95.2</td>
<td>96.7</td>
</tr>
<tr>
<td>036_wood_block</td>
<td>87.6</td>
<td>80.2</td>
<td>89.7</td>
<td>94.6</td>
<td>86.2</td>
<td>90.4</td>
</tr>
<tr>
<td>037_actors</td>
<td>91.7</td>
<td>95.6</td>
<td>95.2</td>
<td>100.0</td>
<td>83.8</td>
<td>96.7</td>
</tr>
<tr>
<td>040_large_marker</td>
<td>97.2</td>
<td>99.7</td>
<td>97.5</td>
<td>100.0</td>
<td>96.8</td>
<td>96.7</td>
</tr>
<tr>
<td>051_large_clamp</td>
<td>75.2</td>
<td>74.9</td>
<td>72.9</td>
<td>79.2</td>
<td>94.4</td>
<td>93.6</td>
</tr>
<tr>
<td>052_extra_large_clamp</td>
<td>64.4</td>
<td>48.8</td>
<td>69.8</td>
<td>76.3</td>
<td>92.3</td>
<td>88.4</td>
</tr>
<tr>
<td>061_foam_blank</td>
<td>97.2</td>
<td>100.0</td>
<td>92.5</td>
<td>100.0</td>
<td>94.7</td>
<td>96.8</td>
</tr>
<tr>
<td>ALL</td>
<td>93.0</td>
<td>93.2</td>
<td>93.1</td>
<td>96.8</td>
<td>92.4</td>
<td>95.3</td>
</tr>
</tbody>
</table>

Table 2: Comparative evaluation of 6D pose estimation in terms of ADD-S (%) on the LineMOD dataset. Objects in bold are symmetric.

<table>
<thead>
<tr>
<th>Method</th>
<th>PoseCNN+ICP</th>
<th>DenseFusion</th>
<th>G2L-Net</th>
<th>PVN3D</th>
<th>PR-GCN</th>
<th>BiCo-Net (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ape</td>
<td>98.6</td>
<td>86.4</td>
<td>63.2</td>
<td>76.1</td>
<td>72.0</td>
<td>41.6</td>
</tr>
<tr>
<td>can</td>
<td>92.0</td>
<td>44.7</td>
<td>63.3</td>
<td>57.3</td>
<td>41.6</td>
<td>32.4</td>
</tr>
<tr>
<td>cat</td>
<td>96.7</td>
<td>22.0</td>
<td>15.8</td>
<td>20.9</td>
<td>33.9</td>
<td>40.2</td>
</tr>
<tr>
<td>drill</td>
<td>41.4</td>
<td>44.7</td>
<td>65.7</td>
<td>70.2</td>
<td>78.4</td>
<td>423</td>
</tr>
<tr>
<td>duck</td>
<td>19.6</td>
<td>15.0</td>
<td>25.2</td>
<td>27.9</td>
<td>41.9</td>
<td>30.0</td>
</tr>
<tr>
<td>egg</td>
<td>25.9</td>
<td>28.2</td>
<td>50.2</td>
<td>52.4</td>
<td>90.0</td>
<td>68.8</td>
</tr>
<tr>
<td>glue</td>
<td>39.6</td>
<td>32.4</td>
<td>49.6</td>
<td>53.8</td>
<td>68.1</td>
<td>67.0</td>
</tr>
<tr>
<td>holp</td>
<td>22.1</td>
<td>49.5</td>
<td>37.9</td>
<td>54.2</td>
<td>74.7</td>
<td>97.2</td>
</tr>
<tr>
<td>MEAN</td>
<td>24.9</td>
<td>29.0</td>
<td>40.8</td>
<td>47.5</td>
<td>63.2</td>
<td>70.3</td>
</tr>
</tbody>
</table>

Table 3: Comparison of ADD-S (%) on the Occlusion LineMOD. Symmetric objects are marked in bold. Competing methods with our BiCo-Net are PoseCNN [Xiang et al., 2018], Pix2pose [Park et al., 2019], PVNet [Peng et al., 2019], HybridPose [Song et al., 2020], PVN3D [He et al., 2020] and PR-GCN [Zhou et al., 2021].

FPS which thus generates $Z^2 = 10,000$ pose candidates from point pairs. The hyper-parameter $\lambda$ in the losses of BCM-S and BCM-M branches is empirically set to 0.05. We use the Adam optimizer with a $10^{-4}$ learning rate to train our model for 50 epochs, and the learning rate decays 0.3 per 10 epochs.

4.3 Comparison with State-of-the-art Methods

Comparative evaluation of the proposed BiCo-Net and state-of-the-art methods on the YCB-Video, LineMOD and Occlusion LineMOD datasets are showed in Tables 1, 2, and 3. In general, our method can consistently achieve state-of-the-art performance in all benchmarks. Specifically, on the YCB-Video dataset, our method achieves the best performance on both metrics in Table 1, and similar results on the LineMOD in Table 2 can also be observed. Compared to moderate improvement on the YCB-Video and LineMOD, the proposed BiCo-Net can gain accuracy of 69.5% on the more challenging Occlusion LineMOD, which is significantly superior to the state-of-the-art methods as illustrated in Table 3. Such results can verify the effectiveness of our BiCo-Net for 6D pose estimation on RGB-D images. In addition, we measure the inference time on average of the proposed method: the forward time of BiCo-Net is 16ms; the point-pair pose computation time is 29ms; the segmentation network takes 30ms. As a result, the average time for processing a frame for inference is 75ms with a GTX 1080 Ti GPU, which is comparable to existing methods (e.g., 60ms for the DenseFusion and 68ms for the PR-GCN) to meet desirable real-time inference in practical applications.

4.4 Ablation Studies

Robustness against Inter-Object Occlusion. To evaluate the robustness of our method against occlusion, we follow
Effects of An Ensemble of Filtered Pose Predictions. To learn a correspondence mapping using multiple instances, a typical robust scheme is least-square fitting based RANSAC to optimize the hypothesis with the maximum inliers. We conduct one experiment in terms of the AUC metrics of the YCB-Video dataset to obtain 79.1%/89.1% on the output of the BCM-S/BCM-M branch, which is significantly inferior to the results using point-pair matching in our BiCo-Net, reaching 94.9%/95.6% only with the method introducing the BCM-S branch can gain comparable performance to the baseline by 0.8% and 4.0% on two datasets respectively. This can be credited to exploiting shape priors of the CAD model provides an ideal reference for the pose regression network, which alleviates the partiality of scene point clouds due to (self-)occlusion. The combination of BCM-S and BCM-M further gains an improvement of 1.1% and 4.5% on both datasets, indicating that these two branches provide complementary information, which further supports our claim about an ensemble of pose predictions.

Evaluation on Size of Point-pairs. We evaluate pose predictions via point-pair matching by taking an average of the BCM-S and BCM-M branches of our method, we conduct experiments on the YCB-Video dataset and the Occlusion LineMOD. The baseline method takes scene points S and cropped image patch I as input and only performs point-wise pose regression by \( T^{PR} = \text{PoseReg}(F^S, p^S) \). The average of \( T^{PR} \) of each object instance is utilized as its pose prediction. As shown in Table 4, the method introducing the BCM-S branch can improve 0.7% and 1.6% on two datasets respectively, indicating that BCM-S effectively improved the discrimination of local feature coding owing to introducing point-wise pose sensitive regularization. The BCM-M branch can outperform the baseline by 0.8% and 4.0% on two datasets respectively. This can be credited to exploiting shape priors of the CAD model provides an ideal reference for the pose regression network, which alleviates the partiality of scene point clouds due to (self-)occlusion. The combination of BCM-S and BCM-M further gains an improvement of 1.1% and 4.5% on both datasets, indicating that these two branches provide complementary information, which further supports our claim about an ensemble of pose predictions.

Acknowledgments
This work is supported in part by the National Natural Science Foundation of China (Grant No.: 61771201, 61902131),
the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No.: 2017ZT07X183).

References


