IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 25, 2020, accepted October 23, 2020, date of publication October 28, 2020,
date of current version December 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034386

Detecting 6D Poses of Target Objects From Cluttered Scenes by Learning to Align the Point Cloud Patches With the CAD Models

XUZHAN CHEN[1], YOUPING CHEN[1], BANG YOU[1], JINGMING XIE[1], AND HOMAYOUN NAJJARAN[2], (Member, IEEE)
[1]School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2]School of Engineering, The University of British Columbia, Kelowna, BC V1V1V7, Canada

Corresponding author: Jingming Xie (xjmhust@hust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant No. 51775215.

ABSTRACT 6D target object detection is of great importance to many applications such as robotics, industrial automation, and unmanned vehicles and is increasingly influencing broad industries including manufacturing, transportation, and retail industries, to name a few. This paper focuses on detecting the 6D poses of the target objects from the point cloud of a cluttered scene. However, conventional point cloud-based 6D object detection methods rely on the robustness of key-point detection results that are not straightforward for humans to understand. The drawback makes conventional point cloud-based methods require expert knowledge to tune. In this paper, we introduced a 6D target object detection method that uses segmented object point cloud patches instead of key points to predict object 6D poses and identity. Our main contributions are an end-to-end data-driven pose correction model that is enhanced with a novel simple yet efficient basis spanning layer booster. Experiments show that although the proposed model is trained only using object CAD models, its 6D detection performance matches that of the models using view data. Thus, the proposed method is suitable for 6D detection applications that have object CAD models instead of labeled scene data.

INDEX TERMS Deep learning, object 6D detection, point cloud, point cloud segmentation.

I. INTRODUCTION

The 6D target object detection refers to the process of identifying an object and estimating its 6-DOF pose in the scene of interest. This process is of great importance to many real-world computer vision applications such as robotics, industrial automation, and unmanned vehicles. In this paper, we focus on the problem of 6D target object detection on cluttered scenes where the object is represented by its shape or geometry e.g., a point cloud. Figure 1 portrays the steps involved in this process using a real scene. Unlike the category level detection problem that learns to detect unseen objects using numerous labeled scene data [1], *a priori* clues about the target objects are provided in the 6D target object detection.

Research on 6D target object detection can be roughly classified into two categories based on the input used for object detection: i) view-based methods and ii) geometry-based methods. The view-based methods take the 2D view of the scene as the input and generate 6D detection results [2]–[6].

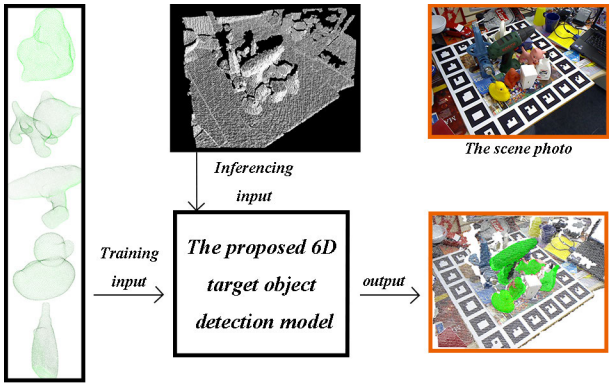

FIGURE 1. 6D detection using geometry models of the target object. The scene is shown in the image at the upper-right corner. It is noted that the color on the output result is only added for better visualization. The scene is represented by the point cloud without the color and texture.

The associate editor coordinating the review of this manuscript and approving it for publication was Mingjun Dai.

Thanks to the well-developed deep learning-based object detection methods, the view-based 6D target object detection methods such as [5], [6] and the view-geometry fusion method such as [7] yielded promising results on a public dataset. Similar to the RGB images, researchers represented the object geometry with the depth image and achieved satisfactory results [8].

Instead of using RGB or depth images as the input for the 6D target object detection problem, we are interested in solving this problem with the point cloud data directly for the following reasons: i) the point cloud is a general representation of geometric information, which can be used to represent not only regularized 2.5D depth image but also 3D CAD models and irregular Lidar generated point-cloud data, and ii) compared to the depth image, the additional dimension of the point cloud makes it possible to segment the object out of the background without using complex methods such as a segmentation network. The permutation invariant property and sparsity of point clouds make the architectures of the well-developed image data-based 6D detection methods such as those in [5], [8] obsolete for the point cloud data. Thus, detecting object 6D pose from the point cloud data in an end-to-end fashion is an interesting problem that we have addressed in this paper.

The conventional point cloud-based 6D detection methods first detect repeatable key-points, then the geometry information around the key-point is encoded into numerical features using feature descriptors [9], [10]. In this way, the object can be detected based on the feature matching result. Thus, the geometry descriptor-based methods greatly rely on the robustness of the key-point detection results that are not straightforward for humans to recognize. That is only exceptionally trained persons can properly tune a conventional descriptor-based method.

In this paper, we propose a new machine learning-enhanced pipeline for point cloud-based 6D target object detection. We firstly segment the scene point cloud into patches, and then the patches are processed by the proposed pose correction model (PCM) to predict object 6D poses and identify the object in an end-to-end fashion. In this way, the key-point detection procedure is substituted by point cloud segmentation. Compared to the key-point detection procedure that generates hundreds of key-points, the scene segmentation results are easier to visualize and more understandable for humans, i.e., the proposed pipeline is easy to set-up. A basis spanning layer (BSL) booster is proposed to reduce the regression error of the transformation matrix and accelerate the convergence of the PCM model. Both theoretical analysis and experiments show that the BSL booster can effectively improve the performance of the proposed method. Besides, a practical loss function that can calculate the point cloud patch pose estimation error parallelly is proposed to train the PCM more efficiently.

Compared to descriptor-based object detection pipelines, the experiment results show that the proposed method robust and adaptive to objects with various geometry features. Furthermore, experiments on the public dataset show that the proposed method has comparable performance against view-based methods trained using labeled real-world data. The novelties of this paper can be summarized as follows:

- A point cloud patch-based 6D target object detection pipeline that is easy to set-up for real-world applications.
- A pose correction model that can be trained using 3D CAD models and applied to the real-world 3D perception data without the need for finetuning.
- A simple yet efficient basis spanning layer (BSL) booster that can accelerate the learning process and improve the pose estimation precision.
- A practical loss function that is suitable for aligning a point cloud patch to the CAD model.

The paper is organized as follows. Section 2 reviews the previous work related to 6D detection. Section 3 presents a detailed description of the proposed model. In Section 4, the proposed model is verified using benchmarks. Section 5 summarizes the conclusion of this work.

II. RELATED WORKS

In this section, we will review the literature related to the 6D target object detection problems. The methods can be roughly divided into two categories: view-based methods and geometry-based methods.

A. VIEW-BASED METHODS

Inspired by the recent development of machine learning techniques, many researchers have proposed view-based target object detection methods. Estimating the object 6D pose based on the 2D-3D matching has shown promising results. Peng *et al.* proposed a pixel-wise voting method that solves the object occlusion problem by predicting the key-point position at every pixel. In this way, the object 6D pose can be estimated by key-point matching [5]. Park *et al.* proposed a novel Pix2Pose model that predicts the depth image of the object area and then calculates the object pose by the PnP method based on their implementation [6].

The 6D pose information-encoded global descriptor can be used to retrieve the object 6D pose efficiently. Wohlhart *et al.* applied machine learning to generate a global descriptor containing the pose and category information [11]. Zakharov *et al.* generated a discriminating descriptor using a triplet loss so the object can be detected by descriptor matching [12]. Balntas *et al.* improved the feature quality using poses to guide the learning process [13]. Recently, Sundermeyer *et al.* proposed a view patch-based object 6D pose estimation method, they used a domain randomization autoencoder to extract features invariant to backgrounds, and the nearest neighborhood method is used to determine object poses [3]. The view-based methods mentioned above estimate object 6D pose based on the global descriptor matching results. However, estimating the object 6D pose using a global descriptor has an inevitable error depending on the discretization resolution of the object pose. In the experiment part, we compared the pose estimation precision of

the view patch-based method and the proposed point cloud patch-based method.

End-to-end learning-based methods directly predict 6D detection results including object position and 3D rigid-body pose from raw input data [15]. Xiang *et al.* proposed a CNN network that has 3 branches including segmentation, translation predicting, and pose estimation for end-to-end 6D detection [14]. Rad *et al.* proposed the BB8 method that regresses the corner of the bounding box instead of the coordinates of the corners [4]. Kehl *et al.* proposed the SSD6D method that uses a single-shot model to detect objects and estimate object poses by classification [16]. Tekin *et al.* proposed a single-shot end-to-end CNN model for real-time 6D detection [17]. Wu *et al.* proposed a machine learning model to predict object poses based on instance segmentation masks [18]. The end-to-end learning-based methods yielded promising results on the 6D target object detection task. In the experiment part, we compared the object detection results against the end-to-end learning-based methods and the proposed method showed comparable performance.

B. GEOMETRY-BASED METHODS

Some researchers developed successful geometry-based methods using a depth image as the input to detect the 6D pose of the object from the scene. Wang *et al.* proposed an iterative dense fusion-based 6D object pose estimation method that extracts pixel-wise dense feature embedding to estimate object 6D pose [7]. Park *et al.* proposed an MTTM method for object detection, segmentation, and pose estimation. In contrast to other methods that rely on the color and texture information, the MTTM method uses depth images as input and estimates the object poses using the nearest neighborhood matching [8].

Detecting objects and estimating the 6D pose from the geometry information represented by the point cloud is getting more and more attention. Extracting robust local features from the scene and objects is an interesting current research topic [19]. Recently, Kehl *et al.* proposed an auto-encoder-based local feature learning method that can improve the quality of local features [20]. Srivastava *et al.* proposed the DeepPoint3D model that learns local features from a point cloud [21]. Point pair feature (PPF), signatures of histograms feature (SHOT) are two widely used descriptors for local feature extraction and 6D target object detection since the usability. The SHOT descriptor is compact and generalizes well on different shapes [9]. PPF descriptor-based 6D target object detection method predicts the 6D pose of the object by hough voting so the PPF method also does not require key-point detection procedure [10]. However, using local features to detect the target objects requires robust key-point detection results, i.e., once failed in detecting repeatable key-points, the object can not be detected. In the experiment part, we compared the proposed method with both PPF and SHOT descriptors on the challenging shapes that have lots of similar local features.

In contrast to the local feature-based methods mentioned above, the proposed method uses the segmented patches instead of detecting key-points. A similar idea is proposed by Dube *et al.* [22]. They use a SegMap feature description method that firstly segments the scene point cloud and then convert the point cloud patches into a feature vector. However, our method is intrinsically different from the method proposed in [22]. The differences include i) the method in [22] aims to localize the robot in the global map while our method aims to detect 6D pose of the object from the scene; ii) the method in [22] generates a rotation-invariant feature vector from the point cloud patches while our method preserves the object pose information using the proposed pose correction model.

III. METHODOLOGY

A. PATCH-BASED 6D DETECTION PIPELINE

We use the patch-based pipeline for 6D target object detection, shown in Figure 2. Firstly, the scene is segmented by the point cloud segmentation algorithms that are off-shelf and can be adjusted to fit the application scenarios. Then, the proposed pose correction model (PCM) takes the segmented patches as the input and yields the object 6D poses and identifications.

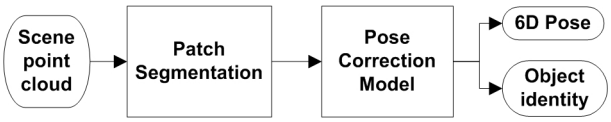

FIGURE 2. The proposed patch-based 6D detection pipeline.

B. POSE CORRECTION MODEL

Pose correction model (PCM) takes the point cloud of the object patch as the input and generates an action that can align the observed object patch with the reference CAD model. In our configuration, the object patch is represented by the point cloud. Figure 3 shows the architecture of the PCM and the color bars in the figure are the learned features. Pose correction model (PCM) consists of basis spanning (marked in green), global feature learning (marked in red), affine matrix learning (marked in blue), and object identification (marked in brown) modules. In the model, $conv(m, n)$ is $1d$ convolution that maps features or points from an m dimensional space to an n dimensional space. *BatchNorm* is the batch normalization function. *MaxPool* function fuses all features generated from each point by maximum pooling.

The basis spanning layer module (marked in green) in Figure 3 is not activated by any non-linear function so the point cloud is up-sampled to 128 dimensions, linearly. The global feature learning module (marked in red) is designed to extract a high dimensional global feature vector h that represents the input point cloud. In the global feature learning module, the ReLU function is used to activate the features processed by all of the convolution functions.

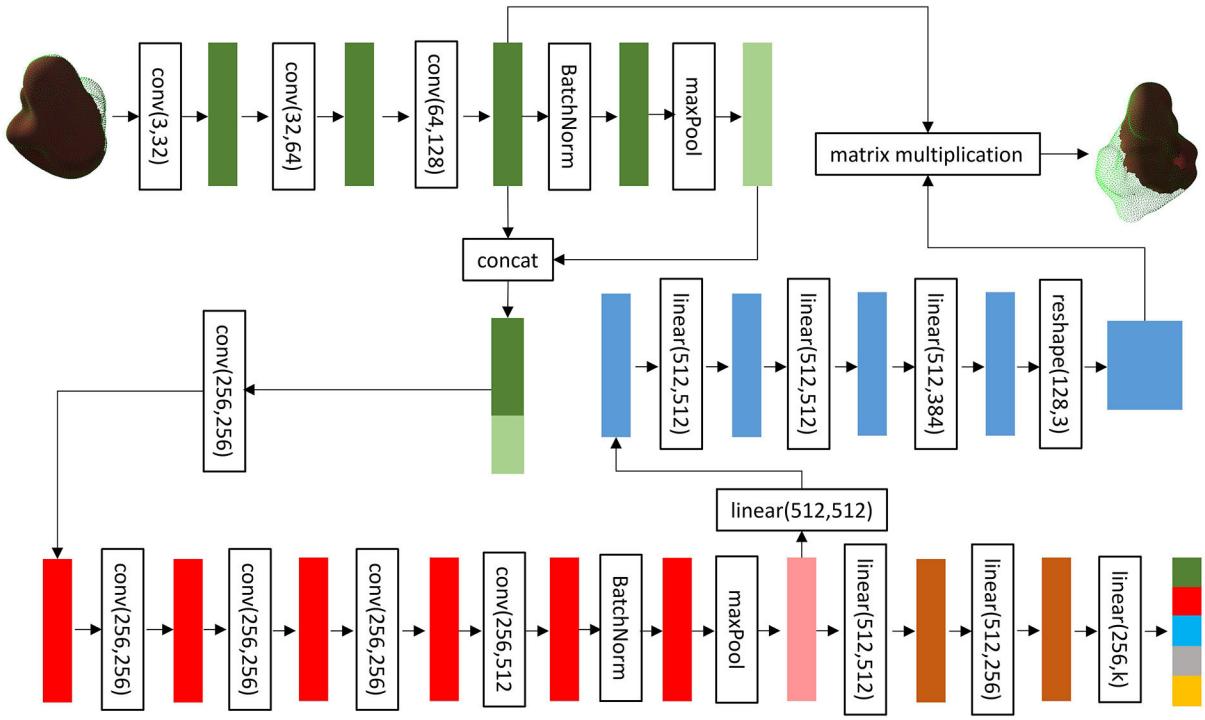

FIGURE 3. The architecture of the network.

The affine matrix learning module (marked in blue) takes the learned feature h as the input and learns a transformation matrix a using the fully connected layers, the ReLU layer is used to activate the feature processed by $linear(512, 512)$ layers.

The reshape function in the affine matrix learning module transforms the learned feature vector into a 128×3 dimension transformation matrix a. In this way, the rotation and translation parameters of the observed patch are encoded in the matrix a.

The object identification module takes the global feature vector h as input and yields the identification result. The ReLU activation function is used to process the $linear(512, 512)$ and $linear(512, 256)$ layers, and the Softmax function is used to process the $linear(256, k)$ layer.

C. THE BASIS SPANNING LAYER (BSL) BOOSTER

The basis spanning layer (BSL) booster is proposed to improve the precision of transformation matrix regression and accelerate the learning process. In this section, the BSL booster is elaborated in detail and its effectiveness is theoretically proven. Eq. 1 describes the process of mapping the point cloud from a 3D to *n-dimensional* space through linear up-sampling by,

$$q = A \cdot p \tag{1}$$

where the 3×1 vector p is the coordinates of a point in the 3D point cloud P. The $n \times 3$ matrix A can be viewed as a set of linear transformation applied to the basis of p, shown in Eq. 2 and Eq. 3.

$$[y_1, y_2, y_3, \ldots, y_n] \cdot q = [x_1, x_2, x_3] \cdot p \tag{2}$$

$$[x_1, x_2, x_3] = [y_1, y_2, y_3, \ldots, y_n] \cdot [a_1, a_2, .., a_i, .., a_n]^T \tag{3}$$

where $[x_1, x_2, x_3]$ is the original orthogonal basis of the observed point cloud patch. q is the coordinate of the point in the spanned n-dimensional space. a_i is the row vector of the matrix A. $[y_1, y_2, y_3, \ldots, y_n]$ is a set of linearly dependent basis and can be further clustered into several groups of linearly independent bases noted as $[y_i, y_j, y_k]$. In this way, we augment the original point cloud by transforming P to several different linear spaces.

The advantages of the BSL booster are twofold,

- In the training process, it is easier for the model to fit the data since the solutions of the pose are not unique for the linearly dependent basis. As a result, the rate of convergence will increase significantly because of the BSL booster.
- The precision of the pose estimation can be improved since the input point cloud data is augmented with a learn-able linearly transformation matrix.

Experiments are conducted to demonstrate the role of the BSL booster and its advantages. The results show that the rate of convergence and pose estimation precision are both improved, considerably.

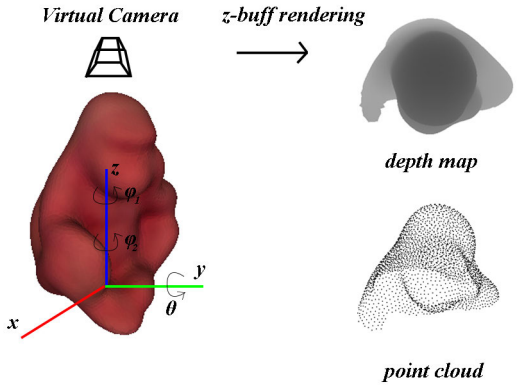

FIGURE 4. Generating the point cloud via rendering the CAD model. The pose of the object is adjusted with the following operations: rotating ϕ_1 degree around the z-axis, rotating θ degree around the y-axis and rotating ϕ_2 degree around the z-axis.

D. THE MODEL TRAINING METHOD

The point cloud used for model training is generated by rendering the CAD model of the target object. The process is shown in Figure 4. First, the pose of the model is adjusted by z-y-z rotation so the ground-truth value of the target object 6D pose is known. Second, a depth image is rendered from the CAD model using the z-buff algorithm. In this way, the self-occlusion of the target object is taken into account. Third, the point cloud is generated based on the known parameters of the virtual 3D camera. Besides, to distinguish the background and the objects, we randomly sample point cloud patches from the data of the cluttered scene to generate the background patches.

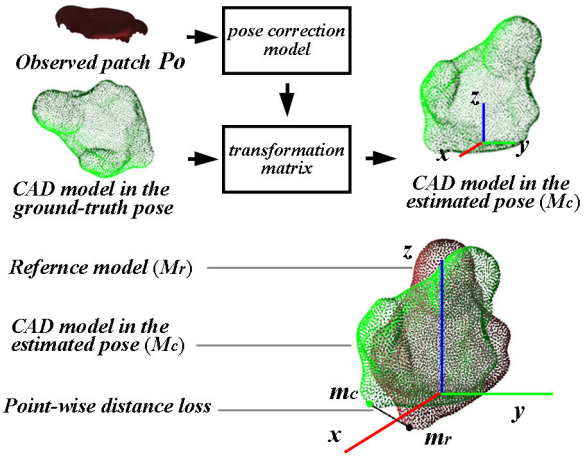

FIGURE 5. Calculating the distance loss using the M_c and M_r.

Inspired by the successful and widely applied Iterative Closest Point (ICP) method and the advantage of the point cloud data, i.e., the Euclidean information is explicitly recorded, we use the alignment error of two point clouds as the loss function of our optimization process. However, finding the closest point of two point clouds is computationally intensive and inefficient for the modern deep learning architecture. Here, we propose a practical strategy to address this problem. As shown in Figure 5, instead of aligning the observed patch P_o to the reference CAD model M_r, we first translate the CAD model to the ground truth pose of P_o. Then the vertices of the CAD model are multiplied by a transformation matrix generated by the pose correction model. In this way, the loss function can be calculated efficiently because the vertices of M_c and M_r are automatically paired.

The input and output data of the proposed Pose Correction Model (PCM) are point cloud data, and the average Euclidean distance (calculated by Eq. 4) between points is used to represent the discrepancy between the actual and the reference point clouds. Although Eq. 4 has the same form as L2 loss function, Eq. 4 is the representation of error as it calculates the Euclidean distance between the corresponding points of two point clouds. The constraint function can be noted as,

$$l_p = \frac{1}{M} \sum_{M} |m_r - m_c|_2 \tag{4}$$

In Eq. 4, m_r and m_c are vertices of M_r and M_c, respectively. With the loss function construction using Eq. 4, PCM learns to align the CAD model in estimated pose (M_c) with the CAD model in the defined reference coordinate (M_r). As a result, PCM can align P_o to M_r.

The configuration of model training is listed as below: batch size 64, SGDM optimizer, learning rate 0.005, momentum 0.9.The object is identified using the output of the Softmax function of the object identification module. The output function predicts k values as the confidence of $k-1$ different objects and the background. Cross Entropy Loss l_i is used to train the identification module. In this way, the loss function for model training can be noted as,

$$l = (1 - \lambda) \cdot l_p + \lambda \cdot l_i \tag{5}$$

In Eq. 5, a dynamic parameter λ is introduced to adjust the ratio of the pose correction constraint and the identification constraint. For the object patches, λ is set as 0.5 and for the background patches, λ is set as 1.0.

IV. EXPERIMENTS

Experiments on CAD models and the real-world dataset are conducted to validate the proposed method. The object identification and pose estimation experiment on CAD models aims to prove that the proposed pose correction model (PCM) has a competitive performance against both descriptor- and view-based methods on 6D pose estimation of objects that have a large number of similar local features. The 6D detection experiment on the real-world dataset aims to prove that the proposed method, trained only using CAD models, is comparable with the method that uses labeled real-world data for 6D detection. The advantages and drawbacks of the proposed method are discussed in this section. It must be emphasized that the consideration of the performance of the proposed method is on top of its main advantage of a 6D detection method that detects objects based on geometry data.

TABLE 1. The object identification and pose estimation recall rates with ADD metric.

Objects	PPF [10] %	SHOT [9] %	Sundermeyer et al. [3]%	Ours %
Bed	62.1	53.4	97.3	98.1
Bookshelf	65.2	72.9	92.5	96.5
Chair	64.7	61.0	98.9	96.2
Dresser	75.4	77.6	97.2	95.9
Desk	66.8	62.8	96.7	98.5
Monitor	78.2	58.1	91.6	97.6
Average	68.7	64.3	95.7	97.1

A. THE OBJECT IDENTIFICATION AND POSE ESTIMATION EXPERIMENTS ON CAD MODELS

The proposed method is compared with the descriptor-based and view-based methods, such as point pair feature (PPF) [10], signatures of histograms feature (SHOT) [9], and view-based method [3] in terms of pose estimation precision and object identification accuracy. The data for comparison are CAD models that have similar local features (Figure 6) including a bed, a chair, a desk, a bookshelf, a dresser, and a monitor. These CAD models have several similar local geometry features, which makes it difficult for conventional 3D descriptors to extract geometry information efficiently.

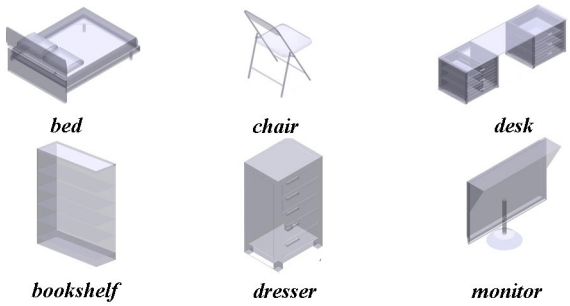

FIGURE 6. CAD models with similar local features.

The CAD models sequentially rotate about z, y, and z axes to generate the training point cloud. The ranges of the rotation angles are 0° to 360° about the z axis, 180° about the y axis and another 360° about the z axis, all with an interval of 9°. For the testing data, the ranges of the rotation angles are 4.5° to 355.5° with an interval of 9°, 175.5° and 355.5°, respectively, with an interval of 9°. In this way, the minimum pose difference between the training and testing data is 4.5°.

The point pair feature-based (PPF) [10] method built a PPF between every two points of the object patch point cloud. For the PPF method, the clustering threshold was 10° for the rotation and 10% of the object diameter for the translation. For the SHOT method [9], the key-points were generated using the 3D Harris key-point. The descriptor radius was 5% of the object diameter. The matching process was accelerated by the K-nearest-neighborhood searching method. The view-based method was trained using rendered depth images and the configuration of training was similar to [3]. The experiment uses the ADD metric ($k_m = 0.05$), i.e., the object is correctly recognized and aligned with the standard model when the average alignment error is less than 5% of the corresponding object diameter. For the view-based method [3], we use the ADD metric ($k_m = 0.1$) because the object pose estimation precision of the view-based method [3] depends on the resolution of pose discretization of the training data while the proposed method uses the global feature to estimate continuous object poses.

The object identification and pose estimation performances of the methods is shown in Table 1. The results show that the average recall rate of the proposed method is 97.1% and it is around 30% higher than the recall rates of PPF and SHOT methods. The proposed method shows advantages against the descriptor-based method because the proposed method learned global features from the training data and the feature extraction ability generalized well to unseen poses. Besides, the average running time for the proposed method is 94 milliseconds while PPF and SHOT methods took more than 2 seconds. The state-of-the-art view-based method [3] showed a competitive recall rate (95.7%) in the experiment, which is 1.4% lower compared to the proposed method. However, the proposed method is evaluated under the criteria of the ADD metric ($k_m = 0.05$), i.e., the proposed method is tested under a more restrictive standard. More interestingly, comparing the standard deviations in the last row of the table, it is conceivable that the proposed method has a more consistent recall and maybe more reliable for various object classes.

B. THE 6D DETECTION EXPERIMENTS ON LINEMOD DATASET

The proposed method is compared to the state-of-the-art view-based methods including the BB8 [4], DenseFusion [7] and the methods introduced in [17] and [3] on the public LINEMOD dataset. The LINEMOD dataset has thirteen different objects. For each object, the CAD model is provided, more than 1000 RGB-D images of cluttered scenes captured from different angles are provided. The 6D pose of the objects is labeled. Figure 7 visualizes eight frames of images from the LINEMOD dataset. Besides the CAD model experiments in Section IV. A., the LINEMOD dataset is collected using the RGB-D sensor. These objects are highly self-occluded and have been subjected to severe noise. Detecting object 6D poses from the LINEMOD dataset needs to deal with the influence of sensor noise and target object segmentation error. In this way, the robustness of the methods can be tested. The detection results are generated by the proposed method and denoted by green shading. The proposed method

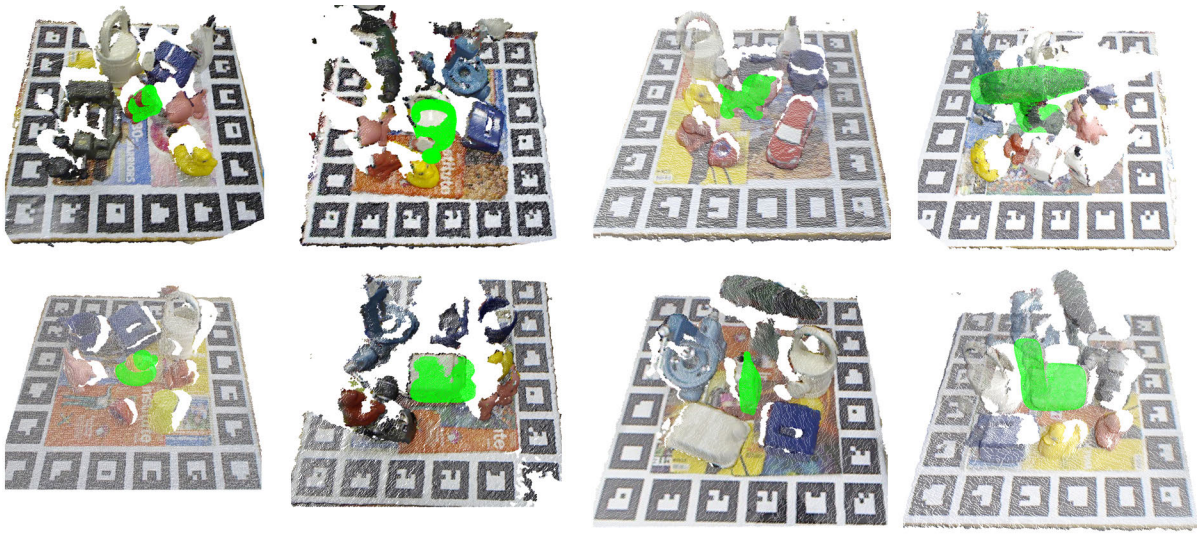

FIGURE 7. 6D detection result of the proposed method on LINEMOD dataset.

is tested in terms of 6D detection accuracy and computational efficiency.

Unlike the other state-of-the-art 6D detection methods such as [3] that train a deep learning detection model to extract the cropped image of the object from the scene, we used a conventional hand-crafted segmentation approach to process the scene. In the 6D detection experiment, the scene is segmented into point cloud patches using a smooth normal plane extraction and color-enhanced euclidean clustering. The clustering algorithm is carefully tuned for different objects to achieve a satisfying performance.

In the experiment on the LINEMOD dataset, we train the model using the synthetic data. All data in the LINEMOD dataset is used as the test dataset to test the proposed method. Thirteen different object classes are used for testing i.e., similar to [3]. The CAD models sequentially rotate about z, y, and z axes to generate the training point cloud. The ranges of the rotation angles are 0° to 360° about the z axis, 180° about the y axis and another 360° about the z axis, all with an interval of 9°. The CAD models in different poses are converted into depth image using Blender. Then, the point cloud for training is generated from the depth image. The configuration of model training is described in Section 2. The model was trained for 300 epochs and the training time is around 23 hours with the Nvidia RTX 2070 GPU. All of the labeled data in the LINEMOD dataset is used for testing.

1) THE ACCURACY OF THE PROPOSED 6D DETECTION METHOD

The comparison of performances of object identification and pose estimation is shown in Table 2. The criterion is ADD ($k_m = 0.1$) as in [3]. The results of BB8 [4], Tekin *et al.* [17], Sundermeyer *et al.* [3] and DenseFusion [7] are taken from the original papers. Shown in Table 2, the average recall rate of 6D object detection on the LINEMOD dataset of the proposed method is 66.8%.

Although the performance of the proposed method is inferior to the state-of-the-art RGB-D-based method such as the DenseFusion method [7], our method is comparable with the view-based methods such as [3], [4], [17] that are proposed before 2018. The main reason for the inferior performance can be summarized as i) the loss of color and texture-information makes object pose estimation a challenging task, ii) the geometry of the object segmented from the real-world scene is not the same as the provided CAD model because of the error of intrinsic parameters of the Kinect sensor, and iii) the real-world noise and segmentation error affects the pose estimation and object identification results. The further analysis of the influence of the sensor noise and segmentation error is discussed in Section V.

It is worth mentioning that the proposed method takes point cloud data as input while the most well-established methods generally focus on 6D detection from RGB or RGB-D data. Detecting target objects from the point cloud remains an open problem for researchers to achieve better performance in terms of detection accuracy and pose estimation precision. With more attention and contributions from the researchers, the point cloud-based 6D target object detection performance can be further improved and be more competitive in comparison to the methods using RGB-D data.

2) THE EFFICIENCY OF THE PROPOSED 6D DETECTION METHOD

We downloaded and ran the official implementation of the method proposed in [3] using a computer with a 4 core CPU, 12 GB memory, and an Nvidia RTX 2070 GPU. The proposed method runs slightly slower (34 ms) compared to [3]. The main reason is that the proposed method spends more time on patch segmentation since the patch segmentation algorithm

TABLE 2. The object identification and pose estimation recall rates on LINEMOD dataset.

Data Type	RGB	RGB	RGB	RGB-D	Point Cloud
Objects	BB8 [4] %	Tekin et al. [17] %	Sundermeyer et al. [3] %	DenseFusion [7]%	Ours %
Ape	40.4	21.62	20.55	80	73.6
Bench vise	91.8	81.80	64.25	84	59.9
Cam	55.7	36.57	63.20	77	71.5
Can	64.1	68.80	76.0	87	65.7
Cat	62.6	41.82	72.01	89	72.3
Driller	74.4	63.51	41.58	78	54.9
Duck	44.3	27.23	32.38	76	73.3
Eggbox	57.8	69.58	98.64	100	61.9
Glue	41.2	80.02	96.39	99	74.5
Hole puncher	67.2	42.63	49.88	79	63.7
Iron	84.7	74.97	63.11	92	58.5
Lamp	75.6	71.11	91.69	92	60.8
Phone	54.0	47.74	70.96	88	77.4
Average	62.7	55.95	64.67	86	66.8

TABLE 3. The object identification and pose estimation recall rates with ADD metric.

Steps	Sundermeyer et al. [3]/ms	Ours/ms
Patch Segmentation	~ 17	~ 162
6D detection	~ 6	~ 125
Pose refinement	~ 230	-
Total	~ 253	~ 287

runs on one core of the CPU. The method proposed in [3] used a GPU-accelerated machine-learning model to crop the object. Although a machine learning-based object cropping step is parallelable and faster, the proposed clustering-based patch segmentation method is fully unsupervised. It is reasonable to believe that the proposed method could be accelerated by training a machine learning model with labeled data to generate object patches similar to [3]. The inference time is shown in Table 3.

In summary, our experiments show that the average 6D detection recall rate of the proposed method is 66.8% so the PCM has a competitive performance on 6D detection in terms of accuracy and efficiency against the view-based methods such as BB8 [4], Tekin *et al.* [17] and Sundermeyer *et al.* [3]. The object identification accuracy and pose estimation precision PCM are more than 30% higher than the descriptor-based methods such as PPF [10] and SHOT [9] according to our implementation. Although the performance of the proposed method is inferior to the state-of-the-art RGB-D-based method such as the DenseFusion method [7], it is worth to mention that the proposed method takes object point cloud as the input and Detection object from the point cloud is still an open problem, i.e., the performance of the point cloud-based 6D detection pipeline will be improved in the future.

C. THE EFFECTIVENESS OF THE BASIS SPANNING LAYER (BSL) BOOSTER

The experiments are conducted to quantify the impact and effectiveness of the proposed basis spanning layer (BSL) booster on the rate of convergence and pose estimation precision. In the experiments, we used the CAD models provided in the LINEMOD dataset to generate the training and testing data. The CAD models sequentially rotate about z, y, and z axes to generate the training point cloud. The ranges of the rotation angles are 0° to 360° about the z axis, 180° about the y axis and another 360° about the z axis, all with an interval of 9°. For the testing data, the ranges of the rotation angles are 4.5° to 355.5° with an interval of 9°, 175.5° and 355.5°, respectively, with an interval of 9°. The model is trained for 30 epochs using the configuration described in Section III.D. The training process is illustrated in Figure 8.

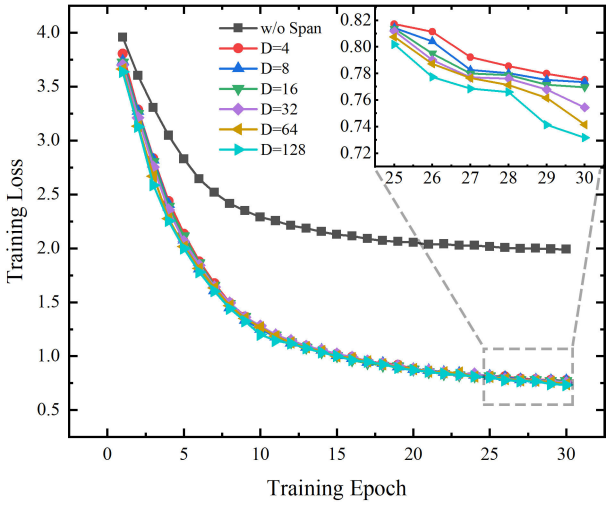

FIGURE 8. The training process of model with different BSL dimensions.

In Figure 8, *w/o Span* denotes the pose correction model without the BSL booster enhancement, i.e., the heterogeneous coordinates of the points are multiplied by a 3×4 matrix. D is the dimension of the BSL i.e., the dimension of the up-sampling. As shown in Figure 8, the training loss is around 0.7 at the 30^{th} epoch for the model enhanced with the BSL booster while the training loss is greater than 2.0 for the model without the proposed BSL booster enhancement. It is clear that the BSL booster can significantly improve the rate of convergence.

TABLE 4. Comparison of model performance on the synthetic dataset and segmented real-world data.

Data type	Ape/%	Cam/%	Cat/%	Duck/%	Driller/%	Glue/%	Phone/%
Synthetic data	97.5	96.9	96.8	95.3	100	99.2	100
Segmented data	73.6	71.5	72.3	73.3	54.9	74.5	77.4

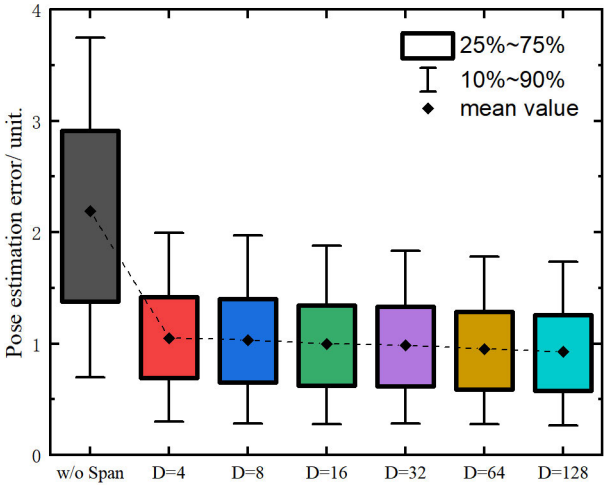

FIGURE 9. The pose estimation error of the model with different BSL dimensions.

Furthermore, we tested the effectiveness of the BSL booster on the pose estimation precision. In the experiments, the model is trained for thirty epochs, and then the trained model is evaluated on the test dataset. The pose estimation error is calculated as the average distance between the point cloud and the corresponding points on the reference CAD model. A comparison between the pose estimation results with and without the BSL booster shown in Figure 9 demonstrates the improved pose estimation precision because of the proposed BSL booster. The results of the experiment show that the proposed BSL booster can significantly reduce the pose estimation error (from around 2.2 units to less than 1 unit). Furthermore, the deviation of the pose estimation error is also reduced as the BSL dimension increases. We attribute the reduction in deviation to the augmentation of the input data.

The results of the experiment show that the proposed basis spanning layer (BSL) booster can significantly reduce the pose estimation error (from around 2.2 units to less than 1 unit). Furthermore, the deviation of the pose estimation error is also reduced with the increase of the BSL dimension. We attribute the reduction in deviation to the augmentation of the input data.

D. DISCUSSION

1) PATCH SEGMENTATION

The proposed method uses segmented object patches as input so that the effect of the segmentation algorithm influences the final result of 6D detection. In this section, an experiment based on a synthetic LINEMOD point cloud is conducted to quantitatively analyze the effect of the segmentation algorithm. In the experiment, the CAD models of the ape, cam, cat, duck, drill, glue, and phone are selected from the LINEMOD dataset. A testing set is generated by rotating each object around z, y, and z axes. The ranges of the rotation angles are 0° to 360°, 90° and 360°, respectively. The rotation angle around each axis is uniformly sampled and the sampling times are 16, 4, 16, respectively. 1024 uniformly sampled poses are generated for each object, and a synthetic LINEMOD dataset with 7168 point cloud patches is generated by rendering LINEMOD objects into a point cloud.

The pose correction model (PCM), trained with the same configuration described in Section 4.2, is used to recognize the object and estimate its 6D pose in the synthetic dataset. The selected objects have less concave surfaces so that the sensor noise has less influence compared to other objects. In this way, the PCM is mainly affected by the object segmentation performance. Thus, comparing object identification and 6D pose estimation performance on the synthetic LINEMOD dataset and real-world data is sufficient to evaluate the influence of patch segmentation. The ADD ($k_m = 0.1$) criterion is used for this comparison (Table 4).

The result shows that the average recall rate of the proposed method is 98.0% when using the synthetic data as the input. However, the average recall rate is 71.1% when using segmented data. The main reason is that the smooth normal segmentation occasionally ignores small objects, especially where the normal estimation radius is set too large. On the other hand, over-segmentation can happen for relatively large objects such as the drill. Because the handle of the drill is sometimes partially self-occluded by the body part, the depth sensor cannot capture a continuous geometry shape and the Euclidean clustering fails in this case. In this way, an improved segmentation method can elevate the performance of the proposed patch-based method.

2) SENSOR NOISE

The sensor noise leads to the difference between the training data and the real application scenarios. The effect of the sensor noise is alleviated when the model uses labeled real-world data to train. It is necessary to analyze the effect of sensor noise by experiments. The objects including the bench vise, egg box, hole puncher, and iron are selected as the testing objects. The common feature of these selected objects is that they all have complex concave surfaces.

A synthetic dataset composed of the above objects is generated in the same way as the patch segmentation experiment. Furthermore, the object patches in the real-world data are segmented out using the ground truth poses so the influence of segmentation is omitted. The result is shown in Table 5.

TABLE 5. Comparison of model performance on the synthetic dataset and corresponding real-world data.

Data type	Bench vise/%	Egg box/%	Hole puncher/%	Iron/%
Synthetic data	96.2	98.8	97.4	95.0
Real-world data	81.4	86.5	82.3	84.2

The model performance is evaluated using the object identification recall rates, and the criterion is ADD ($k_m = 0.1$).

The results show that the sensor noise results in approximately 20% loss of recall rates for the selected objects. The main reason is that the complex non-convex structures are hard to be scanned for actual sensors with a fixed resolution. Specifically, the bench vise has a metal part which is highly sensitive to the sensor noise.

3) LIMITATIONS

The experiments show that the proposed method has a competitive performance for 6D target object detection, but it suffers from the following limitations. First, the patch segmentation influences the 6D target object detection by under- or over-segmentation of the scene. Because the proposed method assumes that the *a priori* information about the scene is unknown and cannot be synthesized, an unsupervised algorithm is used to generate patches. However, it is reasonable to infer that supervised machine learning-based scene segmentation can improve the 6D detection performance. Second, the sensor noise is not negligible when the object has concave surfaces or complex local details such as holes or slots. The proposed method is less competitive for objects with such features, e.g., objects in the T-Less dataset [23]. One of the possible solutions is to substitute the geometry feature learning module of PCM with image feature extraction models. In this way, image features such as edges and corners can be helpful for 6D detection.

V. CONCLUSION

In this work, we presented a point cloud patch-based pipeline for the 6D target object detection problem. A pose correction model (PCM) that is enhanced with a simple yet efficient basis spanning layer (BSL) booster is proposed to predict the 6D pose and identity of the segmented point cloud patches in an end-to-end fashion. Experiments on synthetic data show that the average 6D detection recall rate of the proposed method is 97.1%, which is approximately 30% higher than that of the widely used local feature descriptor-based methods including SHOT and PPF. Besides, experiments on the public LINEMOD dataset show that the 6D detection accuracy is 66.8%, i.e., the performance of the proposed method is comparable to the end-to-end trained view-based methods which are enhanced by deep learning. To sum up, the proposed method has the potential for real-world 6D detection applications where only the point cloud data of an object is available.

REFERENCES

[1] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[2] S. Hinterstoisser, V. Lepetit, S. Ilic, and S. Holzer, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Daejeon, South Korea, 2012, pp. 548–562.

[3] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6d object detection from rgb images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 699–715.

[4] 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 *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 3828–3836.

[5] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4561–4570.

[6] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Long Beach, CA, USA, Oct. 2019, pp. 7668–7677.

[7] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3343–3352.

[8] K. Park, T. Patten, J. Prankl, and M. Vincze, "Multi-task template matching for object detection, segmentation and pose estimation using depth images," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 7207–7213.

[9] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understand.*, vol. 125, pp. 251–264, Aug. 2014.

[10] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 100–998.

[11] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3109–3118.

[12] S. Zakharov, W. Kehl, B. Planche, A. Hutter, and S. Ilic, "3D object instance recognition and pose estimation using triplet loss with dynamic margin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 552–559.

[13] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose guided RGBD feature learning for 3D object pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)* Venice, Italy, Oct. 2017, pp. 3856–3864.

[14] Y. Xiang *et al.*, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*. [Online]. Available: https://arxiv.org/abs/1711.00199

[15] R. Wang, J. Xu, and T. X. Han, "Object instance detection with pruned alexnet and extended training data," *Signal Process., Image Commun.*, vol. 70, pp. 145–156, Feb. 2019.

[16] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1521–1529.

[17] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 292–301.

[18] J. Wu, B. Zhou, R. Russell, V. Kee, S. Wagner, M. Hebert, A. Torralba, and D. M. S. Johnson, "Real-time object pose estimation with pose interpreter networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 6798–6805.

[19] W. Guo, W. Hu, C. Liu, and T. Lu, "3D object recognition from cluttered and occluded scenes with a compact local feature," *Mach. Vis. Appl.*, vol. 30, no. 4, pp. 763–783, Apr. 2019.

[20] K. Wadim, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 205–220.
[21] S. Srivastava and B. Lall, "DeepPoint3D: Learning discriminative local descriptors using deep metric learning on 3D point clouds," *Pattern Recognit. Lett.*, vol. 127, pp. 27–36, Nov. 2019.
[22] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *Int. J. Robot. Res.*, vol. 39, nos. 2–3, pp. 339–355, Mar. 2020.
[23] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Santa Rosa, CA, USA, Mar. 2017, pp. 880–888.

XUZHAN CHEN was born in Shangrao, Jiangxi, China, in 1992. He received the B.S. degree in mechanical engineering from the University of Science and Technology Beijing, in 2014, and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology, in 2019. Since 2019, he has been a Postdoctoral Researcher with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. He is also the Co-Supervisor of the graduate students in ACIS lab, The University of British Columbia. He is the author of two articles and two conference papers. His research interests include 3D robotics vision and deep learning on 3D data.

YOUPING CHEN was born in Harbin, Heilongjiang, China, in 1957. He received the B.S. and M.S. degrees in mechanical engineering from Shanghai Jiao Tong University, in 1982 and 1984, respectively, and the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, in 1990. He is currently a Professor and Ph.D. Supervisor with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. He published more than 160 literatures on scopes including mechatronics, machine tool, and robotics.

BANG YOU received the B.S. degree in mechanical engineering from the Shenyang University of Technology, Shenyang, China in 2018. He is currently pursuing the Ph.D. degree in mechanical engineering with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His research interests include machine learning, reinforcement learning, and robot learning.

JINGMING XIE received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, China, in 2003. He is currently an Associate Professor with the School of Mechanical and Electronic Engineering, Huazhong University of Science and Technology, China. His research interests include network control, machine vision, and intelligent control.

HOMAYOUN NAJJARAN (Member, IEEE) received the Ph.D. degree from the Department of Mechanical and Industrial Engineering, University of Toronto, in 2002. He is currently a Professor with the School of Engineering, The University of British Columbia (UBC). He worked as a Research Officer with the National Research Council Canada, where his research focused on the development of sensor and robotic systems. He joined UBC and founded the UBC Advanced Control and Intelligent Systems (ACIS) Laboratory, in 2006. His research focuses on the analysis and design of mechatronics and control systems with broad applications including unmanned ground and aerial vehicles, industrial automation, and microelectromechanical systems. Over the past decade, he and his students have contributed to multiple aspects of the safe and reliable operation of robots through computer vision, artificial intelligence, and machine learning techniques. He is a Professional Engineer, Fellow of CSME, and also the President of Advanced Engineering Solutions Inc. providing design and technical consulting services to the automation industry.