IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Edge Enhanced Implicit Orientation Learning with Geometric Prior for 6D Pose Estimation

Yilin Wen[1], Hao Pan[2], Lei Yang[1] and Wenping Wang[1]

***Abstract*— Estimating 6D poses of rigid objects from RGB images is an important but challenging task. This is especially true for textureless objects with strong symmetry, since they have only sparse visual features to be leveraged for the task and their symmetry leads to pose ambiguity. The implicit encoding of orientations learned by autoencoders [31], [32] has demonstrated its effectiveness in handling such objects without requiring explicit pose labeling. In this paper, we further improve this methodology with two key technical contributions. First, we use edge cues to complement the color images with more discriminative features and reduce the domain gap between the real images for testing and the synthetic ones for training. Second, we enhance the regularity of the implicitly learned pose representations by a self-supervision scheme to enforce the geometric prior that the latent representations of two images presenting nearby rotations should be close too. Our approach achieves the state-of-the-art performance on the T-LESS benchmark in the RGB domain; its evaluation on the LINEMOD dataset also outperforms other synthetically trained approaches. Extensive ablation tests demonstrate the improvements enabled by our technical designs. Our code is publicly available for research use*.**

I. Introduction

Detecting rigid objects and estimating their 6D poses from images is fundamental in robotics and computer vision and critical for applications like robotic grasping and augmented reality. While object detection has seen great advancements due to the emergence of deep neural networks that recognize and locate objects robustly from diverse surroundings, the object pose estimation problem remains challenging due to the complexity introduced by rotational symmetries of the objects. It is further complicated by the lack of visual salient textures to distinguish different rotations, as can be seen in many common objects, *e.g.* water bottles in daily life [14] or bolts and nuts at manufacturing sites [15].

To handle the textureless inputs with rotational ambiguities, a common approach taken by previous works is to predefine the symmetries manually and solve a perspective-*n*-point (P*n*P) problem [21] for detected 2D/3D keypoint pairs while modulating the symmetry-induced ambiguities [26]–[28], [33].

A drastically different approach proposed by [31] learns a latent space to encode rotations by an autoencoder such that the symmetric poses are implicitly aligned in the encoding space, thus avoiding the prohibitive manual labeling of object symmetries. However, learning a regular and robust encoding space requires a large amount of training data to cover diverse real environments, which is impractical to capture. To solve this problem, [31] instead synthesizes training images by rendering the objects in diverse augmented environments to reduce the gap between the rendered images and those captured from real scenarios, thus naming their approach the augmented autoencoder (AAE).

While AAE shows impressive robustness against textureless and symmetric objects, we propose two key designs to further reduce the domain gap between real and synthetic data and improve the implicit orientation learning, thereby establishing new state-of-the-art performances. First, we observe that the augmented synthetic training images exhibit significant domain gap from real test images, due to the diverse conditions of real lighting, material, occlusion, etc. that are hard to simulate by synthetic images. On the other hand, sharp features (*i.e.* edges) of the images of textureless objects are generally invariant across different conditions, thus providing a robust cue for pose estimation. Therefore, by combining the edge cues with the color images we achieve enhanced discriminative learning of different poses (Sec. V-C). Second, the latent space learned by AAE that encodes orientations generally lacks regularity, in the sense that changes of the pose are not mapped to corresponding changes in latent code (Fig.3), which is a common problem with autoencoders [1]. To address this issue, we propose a geometric prior for the self-supervised learning of latent codes to impose a regularity constraint: we sample a sparse set of reference rotations, and enforce that for any rotation of the object its latent code should be close to the code of its nearest reference rotation. The geometric prior applied leads to further performance improvements as shown in Sec. V-C.

Our network is trained solely on synthetic data, and combined with 2D detection backbones for evaluation on the real benchmarks of T-LESS [15] and LINEMOD [14] containing textureless objects with various symmetries.

To summarize, our main contributions are:

- We combine the color cue with the edge cue to reduce the domain gap between real images for inference and synthetic ones for training.
- We introduce a self-supervision scheme with the geometric prior imposed on the implicit orientation learning that maps input images into latent representations.
- As a result of the two technical contributions, by training on synthetic data only, our method achieves the state-of-the-art performance for 6D pose estimation on

This paper was supported in part by ITF Grant ITS/457/17FP and in part by AIR@InnoHK. *(Corresponding author: Hao Pan.)*

[1]Y. Wen, L. Yang, and W. Wang are with the Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong {ylwen,lyang,wenping}@cs.hku.hk

[2]H. Pan is with Microsoft Research Asia, 5 Danling Street, Haidian, Beijing, China haopan@microsoft.com

*The code is available at https://github.com/fylwen/EEGP-AAE

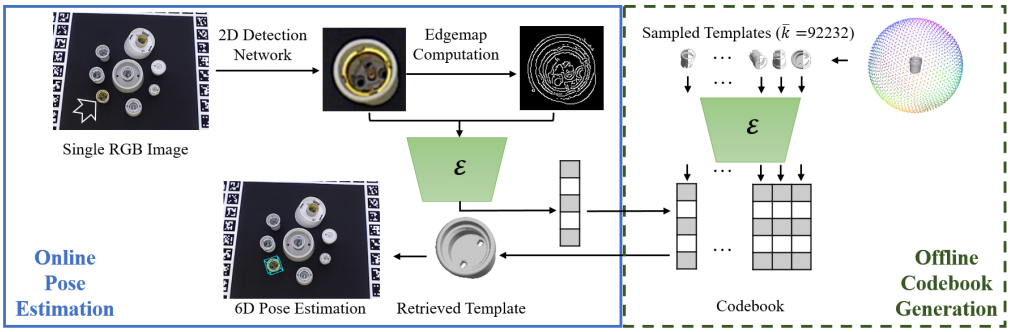


Fig. 1. The pose estimation process at test time. Given a query RGB image and the object of interest as input, the object in the image is detected by a 2D detector. The detected object region is cropped and resized to feed to the trained encoder. Using the offline generated codebook consisting of encodings of the representative rotations, we estimate the rotation of the query instance via nearest code retrieval, and infer the translation based on the bounding box scale ratio between the retrieved pose template and the detected 2D bounding box.

T-LESS [15] in the RGB domain, and outperforms other synthetically trained approaches on LINEMOD [14].

II. Related Work

In this section, we briefly review the most related works on 6D pose estimation and self-supervised learning methods that share similarities with our geometric prior regularization.

A. 6D Pose Estimation Based on RGB Images

6D pose estimation is an active research area with a large body of literature [16]. As our network solely consumes RGB data, making it essentially different from the RGBD approaches that fuse RGB and depth data via networks for pose estimation (e.g. PointFusion [40] and DenseFusion [37]), we briefly review the RGB-based methods from the aspect of the general paradigm taken, and focus on the works most relevant to our approach.

Keypoints matching. This approach detects for a sparse set of 3D keypoints their corresponding 2D image points, and use the perspective-*n*-point(P*n*P) algorithm [21] to estimate the rigid transformation from these 2D-3D correspondences. While [28] and [33] use bounding box corners as keypoints, a recent work [27] explores using designated surface keypoints for more robust 2D keypoint localization.

Dense matching. Methods in this category regress the 3D object points for each 2D image pixel within the object mask, and then estimate the object pose with the dense 2D-3D correspondences using e.g. P*n*P with RANSAC [8] to obtain robustness against noisy correspondences. For example, [3] uses an auto-context network to regress the pixel-wise distribution of 3D coordinates. Pix2Pose [26] is a very recent work that trains an autoencoder and outputs the dense object coordinates with a GAN loss to hallucinate the occluded parts, as well as a predicted confidence map to filter unreliable correspondences.

Direct pose regression. PoseCNN [39] uses a CNN to directly regress the rotations represented as quaternions as well as depths for translation. Similarly, SilhoNet [2] regresses the rotation quaternion by first predicting a silhouette representation that is invariant across synthetic and real images, which shares similarities with our edge cue. However, [42] shows that the quaternion representation of rotations has a non-Euclidean topology and is challenging to learn directly. SSD-6D [18] instead represents rotation as the combination of sampled viewpoints on the bounding sphere and in-plane rotation, and directly regresses orientation by classification into the samples.

We note that for all these different approaches, the prior labeling of symmetry is required for training against pose ambiguity, which can be tedious and impractical for objects with complex symmetries.

Template matching. These methods discretize the rotation space into sampled templates, and retrieve the closest template for a given object image, thus bypassing the explicit labeling of pose ambiguity. E.g., [41] extracts edgelets from the object image, utilizes the directional chamfer distance [17] to retrieve pose templates, and conducts further refinement. But the handcrafted feature computation is time-consuming.

AAE [31] uses an autoencoder trained with synthetically augmented data to map the images to a latent space, where pose ambiguity is implicitly handled through similar latent codes for symmetric poses. However, limitations still exist such as the domain gap between real and synthetic data and regularity of the latent embedding space (Sec. I).

Our approach inherits the merit of the implicit rotation representation learned by an autoencoder and avoids the labeling of pose ambiguity. Meanwhile, we introduce edge cues to narrow down the domain gap between real and synthetic data, and a self-supervision scheme of geometric prior to regularize the implicit rotation representation.

B. Unsupervised/Self-supervised Representation Learning

Autoencoders [10, Chapter 14] are standard for unsupervised representation learning, which suits the need for orientation representation without labeling pose ambiguity for our 6D pose estimation. The naive autoencoders are known to lack regularity for the latent encoding [1], and numerous improvements have been made to address this problem in general. For example, the denoising autoencoder [36] forces the recovery of clean data from noisy input to better capture the inherent low-dimensional structure of the training data. Variational autoencoder [20] encourages the latent embedding space to follow a regular Gaussian distribution. Vector-quantized autoencoders [29], [35] quantize the latent code against a codebook to enhance the regularity of the learned embedding space while avoiding space degeneracy associated with variational autoencoders. Different from these general enhancements, our geometric prior is a regularization of the latent encoding that is tailored

for the rotation representation, and utilizes the contrastive loss to enforce its geometric structure.

Contrastive losses have shown great promise for self-supervised representation learning [5], [11], [34], [38]. Our geometric prior uses the contrastive loss to relate the input rotations to nearby reference poses, which is shown to be an effective self-supervision for rotation representation learning.

III. Method

A. The Autoencoder Framework

As shown in Fig. 2, the autoencoder framework [31], [32] has a pair of encoder and decoder convolutional neural networks, denoted as $\mathscr{E}$ and $\mathscr{D}$ respectively. The encoder $\mathscr{E}$ takes as input an RGB image $I_x \in \mathbb{R}^{W\times H\times 3}$ at pose x, and maps it to a low dimensional code $\mathscr{E}(I_x) \in \mathbb{R}^d$ in the latent space ($d \ll W\times H\times 3$). The decoder recovers an image $\mathscr{D}(\mathscr{E}(I_x)) \in \mathbb{R}^{W\times H\times 3}$ via an inverse mapping. By minimizing the reconstruction loss

$$\sum_{x\in\mathscr{R}} ||\mathscr{D}(\mathscr{E}(I_x)) - I_x||^2 \tag{1}$$

with respect to the parameters of $\mathscr{E},\mathscr{D}$ on the corresponding images of a set of rotations $\mathscr{R}$, we expect the latent code $\mathscr{E}(I_x)$ to capture the pose information that solely distinguishes x from the other poses in $\mathscr{R}$. One can immediately see that for symmetric rotations x and x', since their corresponding images are similar $I_x \approx I_{x'}$, so are their latent codes $\mathscr{E}(I_x) \approx \mathscr{E}(I_{x'})$. This automatically handles the pose ambiguity problem without manual labeling (c.f. [31, Fig. 3-(3)] and Fig. 3-(c)).

To apply the trained networks to pose estimation, as shown in Fig. 1, a dense set of template poses $\overline{\mathscr{R}}$ are sampled and a corresponding codebook $\overline{\mathscr{C}} = \{\mathscr{E}(I_x)|x\in\overline{\mathscr{R}}\}$ is built. Given an input real image I bounding the detected object, we then search for $x^* = \arg\min_{x\in\overline{\mathscr{R}}} d(\mathscr{E}(I),\mathscr{E}(I_x))$ as the closest matching pose, where $d(\cdot,\cdot)$ is the angle between two code vectors. With the retrieved rotation, one can estimate the translation by comparing the relative scaling of the image and the template [18], [31], [32], or by sampling the depth image, if available, at the corresponding region.

Several problems arise in the process that affects its effectiveness. First, a large set of training images $\{I_x|x\in\mathscr{R}\}$ are needed to train the autoencoder. While the training images can be synthesized by rendering the object model in arbitrary poses, they may be visually different from the real input images, since conditions such as lighting and occlusion are hard to simulate. AAE [31], [32] tries to resolve this issue by augmenting the training images, e.g. by changing the lighting conditions, random scaling and cropping, overlaying to a random background, etc. Denoting the random augmentation operator as $G(\cdot)$, the reconstruction loss function in (1) becomes

$$\sum_{x\in\mathscr{R}} ||\mathscr{D}(\mathscr{E}(G(I_x))) - I_x||^2. \tag{2}$$

In Sec. III-B, we propose to reduce this domain gap by using edge cues which are well known to be discriminative features and consistent across the synthesized and real images.

Second, the latent space learned by a regular autoencoder is known to lack regularity even with a large amount of training data [1]. However, we expect the latent space to be regular and represent rotations with a strong geometric structure. We enforce this structure through a self-supervised learning scheme detailed in Sec. III-C.

B. Combining Color and Edge Maps

We use the Canny operator [4] $C(\cdot)$ to compute the binary edge map of an image, but other edge detectors may well be applied. As shown in Fig. 2, given an input image I, we compute its edge map $C(I) \in \mathbb{R}^{W\times H\times 1}$, concatenate the two images as $\bar{I} = [I;C(I)]$, and feed $\bar{I}$ to our encoder. Meanwhile, our decoder has two branches, with $\mathscr{D}^c$ recovering the color image and $\mathscr{D}^e$ the edge map. We adopt a single encoder architecture instead of using two separate encoders for encoding color and edge images, respectively. In this way, we can achieve efficient computation at test time and circumvent the additional design of a combination scheme to fuse the outputs from two encoders. As a result, the autoencoder loss function for color image reconstruction becomes

$$L_{color} = \sum_{x\in\mathscr{R}} ||\mathscr{D}^c(\mathscr{E}(\overline{G(I_x)})) - I_x||^2. \tag{3}$$

The loss function for the edge map reconstruction is

$$L_{edge} = \sum_{x\in\mathscr{R}} \mathrm{BCE}\Big(\mathscr{D}^e(\mathscr{E}(\overline{G(I_x)})), C(I_x)\Big), \tag{4}$$

where $\mathrm{BCE}(\cdot)$ computes a weighted binary cross-entropy between the reconstructed edge map and the input one. Denoting $\mathscr{D}^e(\mathscr{E}(\overline{G(I_x)}))$ as E_x, we define

$$\begin{aligned}\mathrm{BCE}(E_x, C(I_x)) = &-\beta \sum_{C(I_x)(i,j)=1} \log E_x(i,j) \\ &-(1-\beta) \sum_{C(I_x)(i,j)=0} \log(1-E_x(i,j)),\end{aligned} \tag{5}$$

where $C(I_x)(i,j) = 1$ when the pixel (i,j) is an edge pixel, and β is the fraction of the number of non-edge pixels over the total number of pixels in $C(I_x)$.

The total reconstruction loss for training the autoencoder therefore is

$$L_{recon} = L_{color} + L_{edge}. \tag{6}$$

By providing the autoencoder with the edge map as input and forcing it to recover the edge map, we expect the latent pose encoding to be more aware of the discriminative edge cues that are robust across synthesized and real images, thus minimizing the domain gap.

C. Regularization via Geometric Prior

The geometric prior aims to impose the structure of the rotation space $SO(3)$ on the latent encoding space by requiring images presenting nearby rotations to be mapped to nearby latent codes, while repelling the codes for images presenting rotations that are far away.

To implement the geometric prior during network training, we evenly sample a set of reference rotations $\mathscr{R}_c = \{x_q \in SO(3)|q = 1,2,\cdots,k\}$ to serve as anchors spanning the rotation space. Meanwhile, we maintain a corresponding

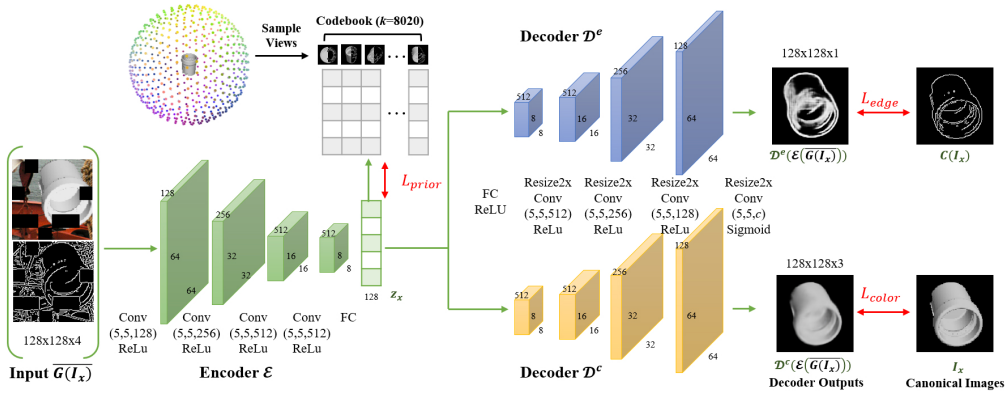


Fig. 2. Overview of the training process. Given a pair of the augmented color image and its edge map, the encoder maps the concatenated image pair to a code in the latent space. The code is compared against a set of reference codes to impose the geometry prior of the rotation space (Sec. III-C). Meanwhile, the code is passed through the color and edge decoders to reconstruct the canonical color image (lower branch) and edge map (upper branch), respectively (Sec. III-B). The reconstruction loss and the geometric prior loss together help the autoencoder to learn an implicit orientation encoding that is more aware of the discriminative edge cues and closer to the rotation space geometry.

latent codebook $\mathscr{C} = \{c_q \in \mathbb{R}^d | q = 1, 2, \cdots, k\}$, where c_q is the latent code for the reference rotation x_q.

For any given rotation x and its latent code $z_x = \mathscr{E}(\overline{G(I_x)})$, we expect z_x to approximate c_q if x and x_q are close, or be different from c_q if x and x_q are far away, thus fulfilling the geometric prior. To this end, we use a contrastive loss to achieve the geometric prior. In particular, we define a probability distribution for z_x to measure its proximity to c_q as

$$p(c_q|z_x) = \frac{\exp(\dot{c_q}^T \dot{z_x}/t)}{\sum_j \exp(\dot{c_j}^T \dot{z_x}/t)}, \tag{7}$$

where t controls the sharpness of proximity (usually called temperature in a contrastive loss), and $\dot{a} = a/||a||_2$ denotes vector normalization. Meanwhile, we define a target probability distribution over the reference rotations as $w^x = [w^x_1, \cdots, w^x_k] \in \mathbb{R}^k$, where $w^x_{q^*} = 1$ for the closest rotation $q^* = \arg\min \angle(x_q, x)$, and $w^x_q = 0$ otherwise. The target distribution represents the closeness between x and the reference rotations. Finally, the contrastive loss is defined as the cross-entropy between the two distributions:

$$L_{prior} = -\sum_{x \in \mathscr{R}} \sum_q w^x_q \log p(c_q|z_x). \tag{8}$$

While the network parameters are trained by a stochastic gradient descent (SGD) solver, the reference codebook $\mathscr{C}$ is updated by exponential moving average to stabilize training. Specifically, for each c_q, there are two accumulated variables $n_q \geq 0$ and $m_q \in \mathbb{R}^d$; they are initialized as 0 and a random unit vector, respectively, and later updated in each SGD iteration following the rules:

$$\begin{aligned} n_q &:= \gamma n_q + (1-\gamma) \sum_x w^x_q, \\ m_q &:= \gamma m_q + (1-\gamma) \sum_x w^x_q z_x, \\ c_q &:= m_q / n_q, \end{aligned} \tag{9}$$

where x iterates over the training samples in a mini-batch. Here $\gamma = 0.99$ is the exponential decay weight.

To summarize, the final loss for training our autoencoder combines the geometric prior and reconstruction losses:

$$L = L_{recon} + \lambda L_{prior}, \tag{10}$$

where λ is a hyper-parameter weighing the two terms.

IV. Implementation

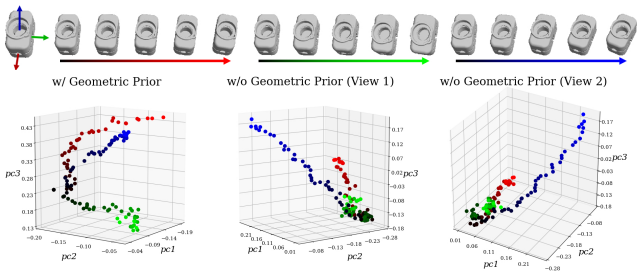
A. Data Generation

To prepare training data, we randomly sample 20,000 rotations as $\mathscr{R}$ for an object. The reference codebook $\mathscr{C}$ has $k = 8020$ rotations formed by combining 20 in-plane rotations with 401 quasi-equidistant views sampled from the Fibonacci lattice on a unit sphere [9]. For pose query at the test stage, we prepare a larger codebook $\overline{\mathscr{C}}$ with $\overline{k} = 92232$ evenly sampled rotations. They are formed by 2562 equidistant spherical views based on refining the icosahedron [13] and further multiplied by 36 in-plane rotations for each view.

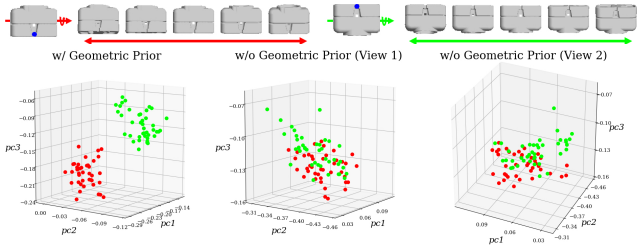
With the sampled rotations, for generating the non-augmented I_x used as the training ground-truth or in computing the codebook $\overline{\mathscr{C}}$, we center and rotate the object and render it under a fixed lighting with a black background. For generating the input training images, the image augmentation operator $G(\cdot)$ follows [31] and consists of 1) randomizing lighting conditions, 2) applying random 2D translation and scaling to the rendered mesh model, 3) combining rendered images with random background images from [7], 4) varying the color values, and 5) adding partial occlusion. In addition, the edge map operator with Canny $C(\cdot)$ uses a fixed threshold parameters $t_1 = 50$, $t_2 = 150$, but we randomize (t_1, t_2) with $t_1 \sim \mathrm{U}(30, 100)$, $t_2 = rt_1$, and $r \sim \mathrm{U}(1.2, 2)$ to augment edge maps extracted from the augmented training color images, where $\mathrm{U}(a, b)$ is the uniform distribution in range $[a, b]$.

B. Network Details

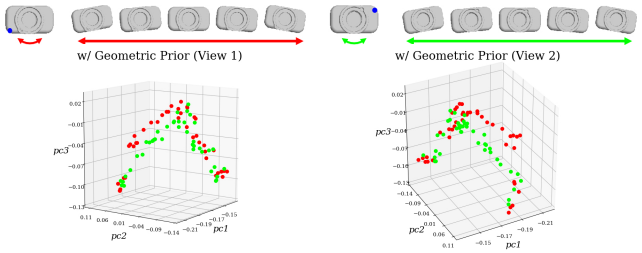
Fig. 2 illustrates the structure of the convolutional neural networks. We empirically set the dimension of the latent space $d = 128$, $\lambda = 0.004$ and $t = 0.07$ for the network training, and also introduce a bootstrap factor of 4 for L_{color} [31]. The Adam optimizer [19] is adopted to train the autoencoder with a fixed learning rate of 0.0002. The batch size is set to 64 and the maximum number of iterations is 30k. During testing, given the 2D bounding box of an object detected by a backbone detector, the input image is cropped and resized to 128×128 and fed to the encoder. We use different detectors in various experiments for a fair comparison, as detailed in Sec. V.



(a) Projected latent codes for three local orientation changing trajectories by rotating an arbitrary pose around three axes.



(b) Projected latent codes for two local orientation changing trajectories by rotating two opposite poses with similar appearances around the same axis.



(c) Projected latent codes for two local orientation changing trajectories by rotating two different but symmetric poses around the symmetry axis.

Fig. 3. Plotting the top three principal component projections (pc_1, pc_2, pc_3) of latent codes for different orientation transition trajectories. PCA bases are computed from the codes of reference poses $\mathscr{R}_c$. The trajectories are obtained by rotating around given axes for 20 degrees with step size 0.5°. Views 1 and 2 are two different views of a same plot in each sub-figure respectively, to better visualize the 3D embedding. (a) and (b) show that with geometric prior, similar orientations are better distinguished by their latent codes than without the geometric prior. (c) shows that for highly symmetric poses, the geometric prior does not prevent the latent codes from getting nearly identical.

V. Experiments

A. Dataset

We evaluate our approach and compare with previous methods on two most widely used datasets, T-LESS and LINEMOD, for 6D pose estimation. The T-LESS dataset [15] contains 30 CAD objects. These objects are highly symmetric and have similar shapes, but have very limited texture information. Moreover, most test images have significant occlusions and/or clutters, which presents further difficulty. Therefore, the dataset is a challenging test for 6D pose estimation. For all the experiments presented, we use the textureless CAD meshes provided by the dataset to prepare the synthetic training images, and leave the real images only for testing.

The LINEMOD dataset [14] contains 15 objects that are more common in daily scenarios. These objects also lack detailed and discriminative textures. For each object, we use its reconstructed mesh provided by the dataset to prepare the synthetic images for network training, and use the real images for testing only.

Compared with T-LESS, most objects in LINEMOD are free from pose ambiguity. Moreover, due to the quality of the 3D models, the inaccurate intrinsics, and sensor registration errors between the RGB and depth images of LINEMOD noted in [25], the pinhole camera model is deeply affected and thus cannot provide an accurate depth estimation, as noticed in [18], [31], [32]. In comparison, networks trained with real data can take advantage of the strong correlation between the real training and testing sets. Taking together these factors, we consider the T-LESS dataset to be more indicative for evaluating our method, as we focus on handling rotation estimation for textureless objects with strong symmetry. In addition, to eliminate the large biases caused by inaccurate depth estimation on LINEMOD and better evaluate our rotation estimation, we refer to the depth image for post-process refinement, using either mean depth or point-to-plane ICP [6]. During the refinement, as commonly done in registration, we eliminate the outlier pixels from consideration. To find the outlier pixels in the depth images, we first measure the maximum distance ε between pixel depths and the average depth for the synthetic depth map rendered under the estimated pose, and consider a pixel of real depth map as an outlier if its depth from the average exceeds 2ε.

B. Evaluation Metrics

Visible Surface Discrepancy [16], denoted as e_{VSD}, computes the difference of the visible depth values between models transformed by the estimated 6D pose and by the ground-truth pose:

$$e_{VSD} = \underset{p \in V_{est} \cup V_{gt}}{\text{avg}} \begin{cases} 0, & p \in V_{est} \cap V_{gt} \wedge |V_{est}(p) - V_{gt}(p)| < \tau \\ 1, & otherwise \end{cases} \tag{11}$$

where V_{est}, V_{gt} are the visible depth maps for the estimated and ground-truth poses, respectively. Therefore e_{VSD} is not sensitive to pose ambiguity because of object symmetry or partial occlusion. We adopt the criterion proposed in [16] that an estimated pose is correct when its $e_{VSD} < 0.3$ with threshold $\tau = 20mm$. We follow [26] and use the reconstructed meshes for error computation.

Average Distance [14], $e_{AD\{D|I\}}$, computes the mean mesh vertex distances as the model M is transformed by the ground-truth pose (R_{gt}, T_{gt}) and by the estimated pose (R_{est}, T_{est}), respectively:

$$e_{ADD} = \underset{v \in M}{\text{avg}} ||(R_{gt} v + T_{gt}) - (R_{est} v + T_{est})||. \tag{12}$$

For symmetric objects, the distance to the nearest vertex is calculated instead:

$$e_{ADI} = \underset{v \in M}{\text{avg}} \min_{v' \in M} ||(R_{gt} v + T_{gt}) - (R_{est} v' + T_{est})||. \tag{13}$$

Following [14], an estimated pose is considered correct if the error $e_{AD\{D|I\}}$ is less than $0.1 d_M$, where d_M is the diameter of the given model.

C. Ablation Tests

We evaluate the effectiveness of the different components proposed in our method. We compare the four alternatives: 1) the original autoencoder approach proposed by AAE, 2) the edge enhanced autoencoder, 3) the original autoencoder with geometric prior, where λ is halved to 0.002 due to the missing edge term (Eq. 6), and 4) the edge enhanced autoencoder with geometric prior. We test for all 30 objects on all Primesense test images provided by the TLESS dataset. To control the inaccuracies introduced by the backbone detection network, we use the ground-truth bounding boxes of each object instead, and report on all instances whose visible portions are larger than 10%.

Tab. I reports the average recall rate with respect to e_{VSD} for all instances in the testing set. Compared with the baseline network using only color images as input and reconstruction target, the introduction of either edge cue for domain gap reduction or geometric prior to latent encoding regularization improves the recall rate by a large margin. On top of that, the combination of them brings the most benefits.

TABLE I

ABLATION STUDY ON DIFFERENT COMPONENTS. AVERAGE RECALL RATE OF $e_{VSD} < 0.3$ FOR ALL INSTANCES OF TLESS OBJECTS WITH VISIBLE PORTION OVER 10% IS REPORTED. OUR TWO NOVEL COMPONENTS BRING SIGNIFICANT IMPROVEMENTS.

Color cue	Edge cue	Geometric prior	Ave.
✓	×	×	64.19
✓	✓	×	67.59
✓	×	✓	68.13
✓	✓	✓	**70.77**

Fig. 3 further visualizes the benefits of introducing geometric prior to the edge enhanced autoencoder, where we use *principal component analysis* (PCA) to project the latent space into $\mathbb{R}^3$ with the top three principal components. Specifically, we use the code set $\{\mathscr{E}(\overline{I_x})|x \in \mathscr{R}_c\}$ for the reference rotations $\mathscr{R}_c$ to compute the PCA bases, and inspect the latent code transitions for three representative cases:

(a) Three different orientation changing transition trajectories around an arbitrary view.
(b) Two local transition trajectories around two opposing orientations that have similar views.
(c) Two local transition trajectories around two different orientations that have nearly perfect symmetry.

As shown in Fig. 3, in the first two cases, without geometric prior, the trajectories have codes that are mixed up. In contrast, with geometric prior, the trajectories are well distinguishable. In the last case, even with geometric prior the codes of two trajectories are very close due to the very negligible differences of the two views, although they are still slightly distinguishable. The three cases indicate that the geometric prior induces regularity of the latent orientation encoding space, in the sense that subtle pose differences are well distinguished while strong symmetries are preserved.

D. Comparison

In this part we compare our method with the state-of-the-art methods on both T-LESS and LINEMOD datasets. Unlike in Sec. V-C where all instances for an object are considered, here we follow the single instance for one object protocol specified in the SIXD challenge [16] to compare fairly with existing works, and use the detected 2D bounding boxes instead of the ground truth ones. Some qualitative results are shown in Fig. 4. Our pipeline was also applied in a grasping task, with setting and a sample result shown in Fig. 4; the demo video is provided in the supplemental material.

TABLE II

COMPARISON WITH STATE-OF-THE-ART METHODS OF THE FULL DETECTION+POSE ESTIMATION PIPELINE. REPORTED ARE THE RECALL RATES OF $e_{VSD} < 0.3$ WITH $\tau = 20mm$ USING ALL PRIMESENSE TEST IMAGES IN THE T-LESS DATASET [15].

obj *id*	**AAE** [31], [32]	**Zhang** [41]	**Pix2Pose** [26]	**Ours** +RetinaNet	**Ours** +GT 2D
01	12.67	7.32	**38.4**	37.01	65.22
02	16.01	12.31	**35.3**	29.78	73.44
03	22.84	14.55	40.9	**44.42**	87.34
04	6.70	5.94	26.3	**26.71**	65.50
05	38.93	38.43	55.2	**56.22**	72.07
06	28.26	18.35	31.5	**47.49**	65.73
07	26.56	19.44	1.1	**26.88**	53.19
08	18.01	21.34	13.1	**22.98**	56.49
09	33.36	**39.46**	33.9	33.84	74.87
10	33.15	9.54	**45.8**	35.79	78.80
11	17.94	10.34	**30.7**	23.27	73.17
12	18.38	9.59	**30.4**	26.25	76.46
13	16.20	6.83	**31.0**	27.70	64.31
14	10.58	5.63	**19.5**	16.76	69.81
15	40.50	35.59	**56.1**	35.81	75.03
16	35.67	29.32	**66.5**	59.31	74.83
17	50.47	**58.82**	37.9	55.20	89.34
18	33.63	50.15	45.3	**60.11**	85.77
19	23.03	**27.45**	21.7	7.49	73.62
20	5.35	4.39	1.9	**9.83**	57.31
21	**19.82**	14.35	19.4	13.77	78.94
22	20.25	**20.57**	9.5	12.4	77.11
23	19.15	15.98	**30.7**	24.19	70.79
24	27.94	8.34	18.3	**37.37**	77.73
25	**51.01**	23.30	9.5	33.98	73.43
26	33.00	10.23	13.9	**42.54**	76.54
27	**33.61**	18.94	24.4	28.14	66.70
28	30.88	19.45	43.0	**56.06**	81.36
29	35.57	35.54	25.8	**49.30**	73.35
30	44.33	37.45	28.8	**59.43**	92.21
Mean	26.79	20.96	29.5	**34.67**	73.35

T-LESS. Following the previous Pix2Pose [26], we use a fine-tuned RetinaNet [23] as the backbone object detector which was pretrained on the MS-COCO dataset [24]. Tab. II presents the recall rate with respect to e_{VSD} and compares our method to other approaches with corresponding detectors: AAE [31] and Pix2Pose [26] with RetinaNet [23], and Zhang et al. [41] with YOLO [30]. Objects with visible portion over 10% on all Primesense scenes are considered. Note that the results of AAE are derived from the latest version [32]. The results show that our method not only brings significant improvements to AAE [31] (by 7% in recall rate) and [41] which uses edge cues in a hand-crafted manner, but also outperforms the state-of-the-art Pix2Pose by more than 5%.

In addition, we argue that should a more accurate 2D detection be provided, the recall rate can be further improved. This is demonstrated by the results produced with ground-truth bounding boxes in Tab. II, where we only process the instance of the highest visible portion for each object in

TABLE III

COMPARISON WITH STATE-OF-THE-ART METHODS OF THE FULL DETECTION+POSE ESTIMATION PIPELINE. REPORTED ARE THE RECALL RATES OF $e_{AD\{D|I\}}$ WITH REGARD TO 10% OF OBJECT DIAMETER ON LINEMOD DATASET [14]. OBJECTS WITH SYMMETRY ARE IN BOLD NAME.

	Synthetic RGB + Depth Refinement				Real RGB					Real RGBD	
	AAE [31] [32]+ICP	**SSD-6D** [18]+ICP	**Ours** +Mean Depth (+MaskR-CNN)	**Ours** +ICP (+MaskR-CNN)	**Brachmann** [3] w/ Ref.	**BB8** [28] w/ Ref.	**Tekin** [33]	**Pix2Pose** [26]	**PoseCNN** [39] +DeepIM [22]	**PointFusion** [40]	**DenseFusion** [37] w/ Ref.
Ape	24.35	65	72.90	**87.38**	33.2	40.4	21.62	58.1	77.0	70.4	92.3
B.Vise	89.13	80	92.83	**96.13**	64.8	91.8	81.80	91.0	97.5	80.7	93.2
Cam	82.10	78	69.28	**91.01**	38.4	55.7	36.57	60.9	93.5	60.8	94.4
Can	70.82	86	85.28	**89.46**	62.9	64.1	68.80	84.4	96.5	61.1	93.1
Cat	72.18	70	91.52	**96.61**	42.7	62.6	41.82	65.0	82.1	79.1	96.5
Driller	44.87	73	70.29	**77.95**	61.9	74.4	63.51	76.3	95.0	47.3	87.0
Duck	54.63	66	52.95	**69.38**	30.2	44.3	27.23	43.8	77.7	63.0	92.3
E.box	96.62	**100**	**100.00**	**100.00**	49.9	57.8	69.58	96.8	97.1	99.9	99.8
Glue	94.18	**100**	99.02	99.02	31.2	41.2	80.02	79.4	99.4	99.3	100.0
HoleP.	51.25	49	55.86	**66.45**	52.8	67.2	42.63	74.8	52.8	71.8	92.1
Iron	77.86	78	96.09	**98.78**	80.0	84.7	74.97	83.4	98.3	83.2	97.0
Lamp	86.31	73	91.44	**94.38**	67.0	76.5	71.11	82.0	97.5	62.3	95.3
Phone	86.24	79	83.51	**93.48**	38.1	54.0	47.74	45.0	87.7	78.8	92.8
Mean	71.58	79	81.61	**89.23**	50.2	62.7	55.95	72.4	88.6	73.7	**94.3**

each image. This serves as idealized upper-bounds on the performances of our approach under the single object single instance protocol, although significant occlusions still exist.

LINEMOD. Following Pix2Pose, we use a fine-tuned Mask R-CNN [12] as the detector. We report the recall rate with respect to $e_{AD\{D|I\}}$ on 13 of 15 objects in Tab. III, where comparing methods are divided into three domains by considering whether real data are used to train the pose estimation network and how depth data are used. We mainly focus on the comparison with SSD-6D [18] and AAE [31], [32], which are similar to ours by training the rotation estimation network solely on synthetic images and estimating the translation by the pinhole model. Both AAE and SSD-6D use depth images for full 6D pose refinement by ICP at the inference stage.

First, we coarsely refine the translation of our results by calculating the mean depth, i.e. "Ours+Mean Depth". Under this setting our results are already comparable to SSD-6D, although SSD-6D samples only a limited range of poses from $SO(3)$ which eases rotation estimation, and refines both rotation and translation by ICP. Meanwhile, our method outperforms the baseline AAE and even an RGBD-based method trained on real data, i.e. PointFusion [40], by over 10% and nearly 8%, respectively. Compared with most of the RGB-based methods which are trained on the real data, our method can also achieve a comparable recall rate with the translation refinement by mean depth only. We further conduct point-to-plane ICP to refine the full 6D pose of our results, shown as "Ours+ICP". This achieves a recall rate that exceeds SSD-6D by a significant margin and is comparable to PoseCNN [39] refined by DeepIM [22].

E. Runtime

The inference time of our method is measured with T-LESS images of size 720×540 as input, on a machine with i7-6700K 4GHz CPU and Nvidia GTX 1080 GPU.

While the RetinaNet [23] takes around 105*ms* to detect objects, our pose estimation for a single instance takes about 11*ms*. In comparison, Pix2Pose [26] uses 25-45*ms* for a single instance, more than twice of ours.

VI. CONCLUSION

In this paper, we have introduced a new method to perform 6D pose estimation from RGB images, which handles tex-

Fig. 4. Visualization of estimated poses of several testing images from T-LESS and LINEMOD (*Row 1&2*), and a grasping task (*Row 3*). The grasping setting is shown on the left. The *green* boxes and *blue* boxes are ground truth poses and our estimation, respectively. Our network works robustly in these diverse environments.

tureless objects with strong symmetry. Based on the implicit orientation encoding framework, we propose two key designs for improvement. Specifically, we show that combining the color images and edge maps can help bridge the domain gap between the synthetic training images and the real testing data. In addition, the geometric prior designed to impose the rotation space geometry onto the latent space enhances the regularity of the learned orientation encoding, thus further improving the performance. Extensive evaluations on the challenging T-LESS and LINEMOD datasets demonstrate the effectiveness of our method.

Limitations and future work. While our approach works well with symmetry and pose ambiguity, it does not explicitly address the issues caused by occlusion or cluttered backgrounds. In the future, we would like to take these factors into consideration and make our approach more robust for complex 6D pose estimation scenarios.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] G. Billings and M. Johnson-Roberson, "Silhonet: An rgb method for 3d object pose estimation and grasp planning," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3727–3734, 2019.

[3] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3364–3372.

[4] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[6] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images." *Image Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

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

[9] Á. González, "Measurement of areas on a sphere using fibonacci and latitude–longitude lattices," in *Mathematical Geosciences*, vol. 42, no. 1. Springer, 2010, p. 49, doi:10.1007/s11004-009-9257-x.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[13] S. Hinterstoisser, S. Benhimane, V. Lepetit, P. Fua, and N. Navab, "Simultaneous recognition and homography extraction of local patches with a simple linear classifier." in *Proceedings of the British Machine Vision Conference (BMVC)*, 2008, pp. 1–10.

[14] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision (ACCV)*. Springer, 2012, pp. 548–562.

[15] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[16] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T. Kim, J. Matas, and C. Rother, "Bop: Benchmark for 6d object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[17] M. Imperoli and A. Pretto, "D^{2}CO: fast and robust registration of 3d textureless objects using the directional chamfer distance," in *International Conference on Computer Vision Systems*. Springer, 2015, pp. 316–328.

[18] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1521–1529.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[21] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate $O(n)$ solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

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

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.

[25] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6d pose refinement in rgb," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 800–815.

[26] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

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

[28] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3828–3836.

[29] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in Neural Information Processing Systems (NIPS)*, pp. 14 866–14 876, 2019.

[30] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[31] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 699–715.

[32] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *International Journal of Computer Vision*, pp. 1–16, 2019.

[33] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 292–301.

[34] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[35] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6306–6315.

[36] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine learning (ICML)*. ACM, 2008, pp. 1096–1103.

[37] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3343–3352.

[38] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3733–3742.

[39] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems*, 2018, doi:10.15607/RSS.2018.XIV.019.

[40] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 244–253.

[41] H. Zhang and Q. Cao, "Detect in rgb, optimize in edge: Accurate 6d pose estimation for texture-less industrial parts," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3486–3492.

[42] Y. Zhou, C. Barnes, L. Jingwan, Y. Jimei, and L. Hao, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.