2019 IEEE/CVF International Conference on Computer Vision (ICCV)

CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation

Zhigang Li Gu Wang Xiangyang Ji
Tsinghua University
Beijing, China
{lzg15, wangg16}@mails.tsinghua.edu.cn xyji@tsinghua.edu.cn

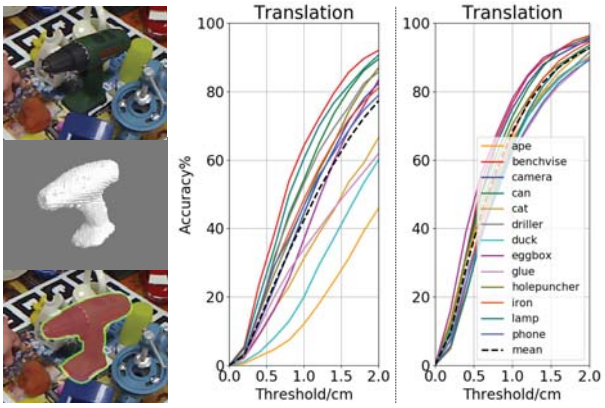
Abstract

6-DoF object pose estimation from a single RGB image is a fundamental and long-standing problem in computer vision. Current leading approaches solve it by training deep networks to either regress both rotation and translation from image directly or to construct 2D-3D correspondences and further solve them via PnP indirectly. We argue that rotation and translation should be treated differently for their significant difference. In this work, we propose a novel 6-DoF pose estimation approach: Coordinates-based Disentangled Pose Network (CDPN), which disentangles the pose to predict rotation and translation separately to achieve highly accurate and robust pose estimation. Our method is flexible, efficient, highly accurate and can deal with texture-less and occluded objects. Extensive experiments on LINEMOD and Occlusion datasets are conducted and demonstrate the superiority of our approach. Concretely, our approach significantly exceeds the state-of-the-art RGB-based methods on commonly used metrics.

1. Introduction

Object pose estimation is essential for a variety of applications in real world including robotic manipulation, augmented reality and so on. In this work, we focus on estimating 6-DoF object pose from a single RGB image, which is still a challenging problem in this area. The ideal solution should be able to deal with texture-less and occluded objects in cluttered scenes with various lighting conditions and meet the speed requirement for real-time tasks.

Traditionally, this task was considered as a geometric problem and solved by matching feature points between 2D images and 3D object models. However, they require rich textures to detect features for matching. Thus, texture-less objects cannot be handled. Benefiting from the rise of deep learning [7], plentiful data-driven approaches emerge and a large improvement has been achieved. Current leading approaches either directly regress 6-DoF object pose from image [8, 28] or predict 2D keypoints in image and indirectly solve the pose via PnP [20, 19]. However, for the direct approaches, they relies heavily on elaborate post refinement steps with 3D information to improve the accuracy of the estimated pose. For the indirect approaches, the sparse 2D-3D correspondences make them sensitive to partial occlusions. Also, they still need pose refinement to achieve a better performance. Except for these approaches, another way is coordinates-based approach, which has been confirmed to be robust to heavy occlusion [9, 18]. It predicts the 3D location in the object coordinate system for each pixel of the object to build dense 2D-3D correspondences to solve the pose. However, existing coordinates-based methods rely



I. Coordinates-based pose estimation II. Disentangled

Figure 1: I. We propose a novel coordinates-based pose estimation approach. From top to bottom (left), we show the query image, 3D object coordinates we estimate and the 2D projection of object model using the predicted 6-DoF pose. However, the translation shows unbalance performance across objects (middle). II. We further propose disentangled pose estimation approach, which is able to handle this problem and show robust and accurate translation across objects (right).

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.00777

heavily on depth information and are inefficient, failing in implementing on RGB-only cases and hard to satisfy the real-time requirement.

Existing approaches leverage deep networks either to directly estimate rotation R and translation T from image or to construct 2D-3D correspondences and indirectly solve them via PnP. However, rotation and translation have significantly different properties and are affected by different factors. For example, the size and location of the object in image have little influence to rotation but affect translation a lot. On the contrary, the appearance of the object in image affects rotation a lot while only slightly influences translation. Thus, the same treatment may bring distinct effects on R and T, and thus we need to solve R and T separately.

In this work, we aim to develop a highly accurate, robust and efficient pose estimation approach for RGB-only cases. We follow the idea of coordinates-based approaches to estimate rotation since their dense correspondences show robustness towards occlusion and clutter. However, the translation solved in this way tends to yield distinct performance across objects and even fails in some cases (see Sec. 3.4 for details). We propose *Coordinates-based Disentangled Pose Network* (CDPN) to disentangle the rotation and translation, allowing that the former is indirectly solved from coordinates via PnP while the latter is directly estimated from image.

Our contributions are summarized as follows:

- For 6D pose estimation, we propose *Coordinates-based Disentangled Pose Network* (CDPN) to efficiently characterize object's rotation and translation. To the best of our knowledge, we are the first to unify the indirect PnP-based strategy and the direct regression-based strategy to estimate the object pose.
- We propose *Dynamic Zoom In* (DZI) to make pose estimation robust to detection errors and moreover, insensitive to any specified detector.
- To afford real-time application, in terms of rotation estimation, we propose a two-stage object-level coordinates estimation, accompanied with the proposed *Masked Coordinate-Confidence Loss* (MCC Loss) to overcome the influence from the non-object region.
- For translation, to avoid the influence from coordinates, we propose *Scale-Invariant Translation Estimation* (SITE) to achieve robust and accurate translation estimation.
- Our approach is highly accurate, fast and scalable. **We achieve the state-of-the-art performance on LINEMOD dataset.** The system is fast enough ($\sim$30ms per image, which can be further shortened by utilizing faster detectors) to achieve the real-time requirement. Our approach is scalable and can work with various detectors without re-training.

2. Related Work

The approaches of solving pose from a single image can be divided into direct approaches and PnP-based ones.

Direct approaches. Direct approaches were initially proposed in some works on 6-DoF camera pose estimation and viewpoint estimation [22, 11, 10, 16, 23, 14]. Then, similar ideas appear in the domain of 6-DoF object pose estimation. [8] discretized the 3-DoF rotation space into classifiable viewpoint bins and solved it by training a classifier built on a SSD-based [15] detection framework, whereas the 3-DoF translation was obtained from the 2D bounding boxes. [28] proposed a regression-based approach PoseCNN, where the 6-DoF pose parameters, i.e., quaternion and distance were directly regressed from the input image. [6] resorted to regressing the Lie algebra for rotation, and further combined the predicted distance and instance bounding box to get the 6-DoF pose. These direct approaches often further utilize depth information and the ICP [1] (Iterative Closest Point) algorithm for pose refinement. However, they still suffer from some challenging issues such as the viewpoints ambiguity problem [14].

PnP-based approaches. Another line of works resorts to intermediate representations to solve 6-DoF object poses indirectly. [20] proposed to detect the projections of corners of 3D bounding box from image and further solved the 6-DoF pose using the PnP algorithm. They also need ICP for more accurate estimation. With similar idea, [25] developed YOLO6D based on the lightweight detector YOLOv2 [21]. Additionally, some others detected object keypoints from images to solve the pose. [19, 29] predicted semantic keypoints for pose and viewpoint estimation with an hourglass-based network [17, 5]. Except for these sparse keypoints-based methods, another type of indirect pose estimation is coordinates-based approaches. Based on RGB-D images, [2] predicted dense 3D-3D correspondences using a random forest and then optimized the 6-DoF pose hypotheses through RANSAC. [12] replaced the energy function in [2] by a CNN to handle more complicated conditions. [3] further extended this idea to RGB-only 6-DoF pose estimation. An auto-context random forest was proposed to jointly predict object labels and coordinates. However, the performance is still limited. [4] used a fully convolutional network to predict 3D coordinates of scenes for 6-DoF camera pose estimation. [18] predicted 6D object poses based on a 3D coordinates regression CNN after obtaining the object of interest via a semantic segmentation CNN, while actually they reformatted the problem as a classification problem by discretizing the coordinates into bins. More recently, [26] introduced a Normalized Object Coordinate Space (NOCS) for category-level 6D object pose estimation, whereas depth information was needed to estimate the full 6-DoF pose.

Different from these works, we treat the rotation and translation separately. For rotation, compared with exist-

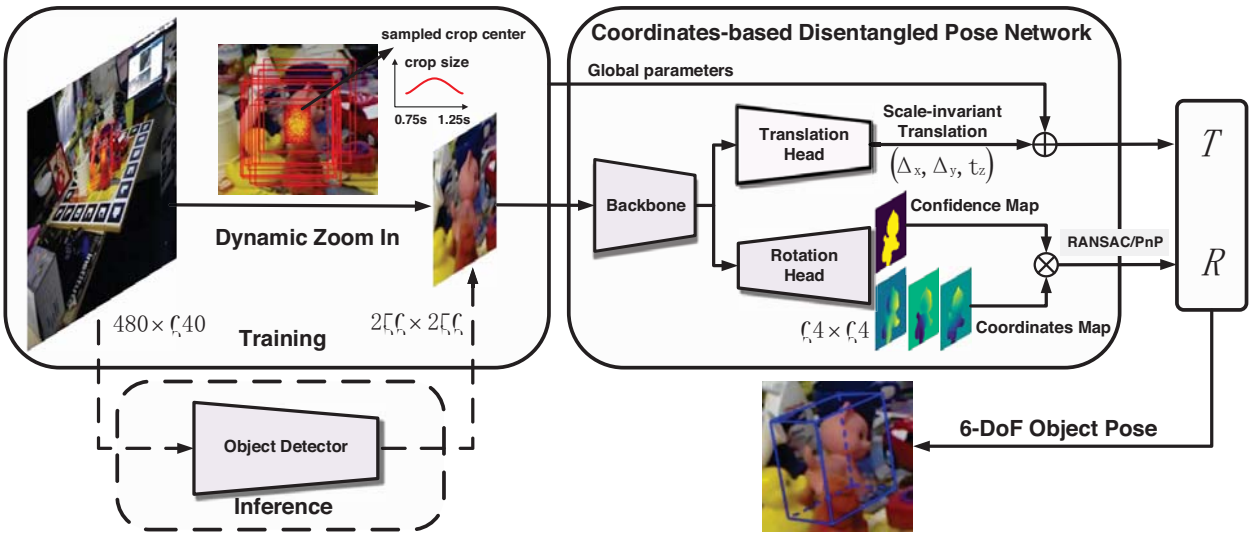


Figure 2: Overview of our approach. Given an input image, we first zoom in on target object, and then, the rotation and translation are disentangled for estimation. Concretely, the rotation is solved by PnP from predicted 3D coordinates, while the translation is estimated directly from image.

ing coordinates-based approaches, our well-designed local-region based paradigm makes the estimation more accurate and efficient. For translation, we directly estimate it from local image patches. We merge these tasks and solve them in a unified network. Our work achieves the state-of-the-art performance on common benchmarks.

3. Method

3.1. Framework

We adopt coordinates-based approach to estimate rotation for the dense correspondences show robustness towards occlusion and clutter. To build 2D-3D correspondences, we need to extract the exact object region. Some approaches [18, 28] train a semantic segmentation network for this task, however, they are inconvenient to deal with multiple identical objects in an image. Instance segmentation such as Mask-RCNN [27, 26] can handle it, however, the slow inference makes it hard to meet the real-time requirement. Contrast to them, we adopt a two-step pipeline: First, a fast, lightweight detector (e.g. tiny YOLOv3) is employed for coarse detection; Second, a fixed size segmentation is implemented to extract the object pixels. For detection, our pose estimation system can tolerate detection errors to a large extent attributing to the proposed *Dynamic Zoom In* (Sec. 3.2), so a fast but less-precise detector is enough. For segmentation, it can be merged into coordinates regression (Sec. 3.3) to make system enough light and fast. Compared with existing segmentation-based approaches, our two-step pipeline can efficiently extract exact object region in various situations.

In terms of translation, to achieve more robust and accurate estimation, we predict it from the image instead of 2D-3D correspondences to avoid the influence from the scale error in the predicted 3D coordinates. Instead of regressing translation from the whole image, we propose *Scale-Invariant Translation Estimation* to estimate it from the detected object region. In this way, the disentangled processes regarding rotation and translation are unified into a single network, namely *Coordinates-based Disentangled Pose Network* (Fig. 2).

3.2. Dynamic Zoom In (DZI)

The object size in image can change arbitrarily along with the distance to camera, which greatly increases the difficulty of regressing coordinates. It is also hard for the network to extract useful features when objects in image are small. To solve these problems, we zoom in on the object to a fixed size according to the detection.

On the other hand, our unified pose network should be robust to any detectors, which means the detection error ε must be taken into consideration. Although it is fine to directly train the pose network on a specific detector, however, the network will be closely related to the detector. We propose a better solution *Dynamic Zoom In* (DZI) for this problem. Given an image containing the target object with position $C_{x,y}$ and size $S = \max(h, w)$, we sample position $\tilde{C}_{x,y}$ and size $\tilde{S}$ from the truncated normal distribution defined in Eq. 1. The sampling range depends on the object height h, width w and coefficients $\alpha, \beta, \gamma, \rho$. Then, we extract the object using $\tilde{C}_{x,y}$ and $\tilde{S}$ and resize it to a fixed size while keeping the aspect ratio unchanged via padding

when necessary. DZI has several merits: 1) It makes the pose estimation model robust with detection errors ε. 2) it improves the system's scalability regarding detector for the training process is independent with it, and we can use a fast single-stage detector to accelerate the system during test. 3) It improves the pose estimation performance by providing more training samples. 4) It keeps the consuming time of network inference and calculation of PnP with RANSAC constant due to the fixed size output.

$$\begin{cases} \tilde{x} & \sim f_x = \frac{\phi(\frac{\tilde{x}-x}{\sigma_x})}{\sigma_x(\Phi(\frac{\alpha \cdot w}{\sigma_x}) - \Phi(\frac{-\alpha \cdot w}{\sigma_x}))} \\ \tilde{y} & \sim f_y = \frac{\phi(\frac{\tilde{y}-y}{\sigma_y})}{\sigma_y(\Phi(\frac{\beta \cdot h}{\sigma_y}) - \Phi(\frac{-\beta \cdot h}{\sigma_y}))} \\ \tilde{s} & \sim f_s = \frac{\rho\phi(\frac{\tilde{s}-s}{\sigma_s})}{\sigma_h(\Phi(\frac{\gamma \cdot s}{\sigma_s}) - \Phi(\frac{-\gamma \cdot s}{\sigma_s}))} \end{cases} \tag{1}$$

where (x, y) and (h, w) are the location of the object's center and the size of the ground-truth bounding box respectively. $s = \max(h, w)$. ϕ is standard normal distribution and Φ is its cumulative distribution function. $\alpha, \beta, \gamma, \rho$ are coefficients to limit the sampling range. $\sigma_x, \sigma_y, \sigma_s$ are used to control the distribution shape.

3.3. Continuous Coordinates Regression

Coordinates-Confidence Map Unlike those approaches [9, 18] extract image patches and predict the coordinates for the center pixel, we predict 3D coordinates for all object pixels in one-shot to achieve high efficiency. Additionally, the network predicts a confidence value for each pixel to indicate whether it belongs to object. Instead of utilizing an additional network branch, we merge this task into coordinates regression based on the fact that both of them have the same output size and their values have exact positional correspondences. In implementation, we first use a backbone network to extract features from the object region. Then, a rotation head consists of convolutional and deconvolutional layers are introduced to process and scale up the features to a four-channel Coordinates-Confidence Map ($H \times W \times 4$), including a three-channel coordinates map M_{coor} and a single-channel confidence map M_{conf}. They share all features in the network. In M_{coor}, each pixel encodes a 3D coordinate and each channel represents an axis of the object coordinate system.

Masked Coordinates-Confidence Loss The ground-truth coordinates of background pixels are unknown. Most approaches [2, 18] assign a special value for them. It works for these approaches because they predict coordinates via classification instead of regression. Since our approach directly regresses the continuous coordinates, it impels the network to predict a sharp edge on the object boundary of the coordinates map, which is challenging and tends to yield erroneous coordinates. To solve this problem, we propose *Masked Coordinates-Confidence Loss* (MCC Loss). Concretely, in terms of the coordinates map, we only compute the loss on foreground regions. While for the confidence map, we apply the loss to all areas (Eq. 2). This mechanism avoids the influence from non-object region and facilitates the network to provide more accurate coordinates. We adopt $L1$ loss in training.

$$\begin{aligned} \mathcal{L}_{CCM} =& \alpha \cdot \ell_1(\sum_{j=1}^{n_c}(M_{conf} \circ (M_{coor_j} - \tilde{M}_{coor_j}))) \\ &+ \beta \cdot \ell_1(M_{conf} - \tilde{M}_{conf}) \end{aligned} \tag{2}$$

where $n_c = 3$ is the channel number of coordinates map, M_*^i and $\tilde{M}_*^i$ represent the ground-truth map and the predicted map respectively. $\circ$ is the Hadamard product.

Building 2D-3D Correspondences The object pixels can be extracted from confidence map by setting a threshold. However, the size of object in RGB image is usually different from that in coordinates map due to the zoom in. To build the 2D-3D correspondences, we map the pixel from the coordinates map to RGB image without loss of precision. We designate the object center and size in RGB image as (c_u, c_v) and $(\tilde{S}_x, \tilde{S}_y)$, and in coordinates map as (c_i, c_j) and (S_x, S_y). For pixel (i, j) in coordinates map, the corresponding pixel $(\hat{u}, \hat{v})$ in RGB image can be computed as Eq. 3.

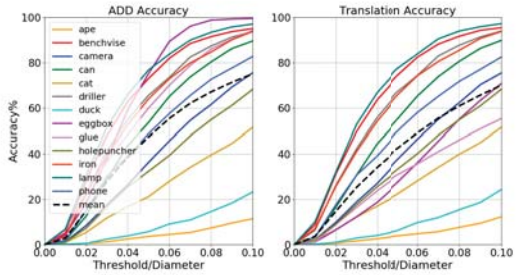
$$\begin{cases} \hat{u} & = \{c_u + S_x/\tilde{S}_x \cdot (i - c_i)\} \\ \hat{v} & = \{c_v + S_y/\tilde{S}_y \cdot (j - c_j)\} \end{cases} \tag{3}$$

where $\{\}$ represents no rounding operation. The rotation can be solved easily from the correspondences by PnP with RANSAC.

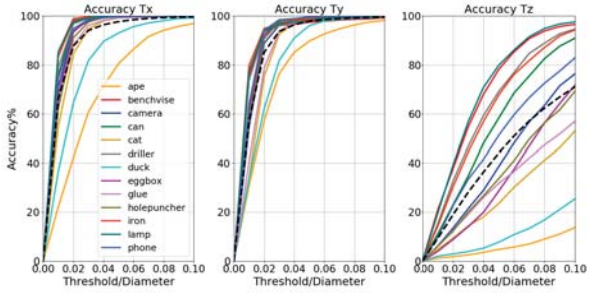
3.4. Analysis on Translation

Training the network using *Dynamic Zoom In* with *Masked Coordinates-Confidence Loss*, our approach achieves high accuracy 94.27% (*state of the art*) on metric "5cm 5°" while achieves modest accuracy 75.04% on metric "ADD" (Table 2 in Sec. 5). The former metric mainly focuses on rotation while the latter concentrates on translation in terms of LINEMOD dataset. [1] It means the approach is more suitable for rotation estimation. See Fig. 3(a), the results of "ADD" show extremely unbalanced performance across objects and are highly correlated with translation, which restricts the application a lot. In our approach, both of pixels $P_{u,v}$ and corresponding 3D coordinates $Q_{x,y,z}$ are estimated from network and they affect translation T solved by PnP (Eq. 4). We perform comprehensive analysis and find out the problem is mainly caused by the scale factor error δ_{scale} in 3D coordinates

[1] For metric "5cm 5°", 5cm is a large range for objects in LINEMOD dataset; While for metric "ADD", compared with translation, the precision requirement of rotation is lenient. Take the 'ape' for instance, the maximum acceptable rotation bias is 23° while the translation error should be smaller than 1cm.



(a) Accuracy of ADD (left) and translation (right).



(b) Accuracy of each translation component.

Figure 3: The accuracy of ADD (for 6-DoF pose) and translation. (Note: both rotation and translation are solved from coordinates via PnP.)

$Q_{x,y,z}$. δ_{scale} affects the depth component T_z of translation a lot (Fig. 3(b)). Different δ_{scale} of different objects yields the unbalanced translation performance. Detailed analysis and experiments can be found in supplementary.

To achieve more robust and accurate translation estimation, we propose to directly learn translation T from image to avoid the influence from δ_{scale} of $Q_{x,y,z}$ (Eq. 5). Estimating T from image is promising and reasonable considering the fact that the object position and size directly reveal its direction and distance to camera. This strategy has been employed in several approaches. For instance, Xiang et al. [28] train a semantic segmentation network to simultaneously learn translation and rotation from image. It achieves remarkable performance on "ADD" while the result on "5cm 5°" is poor (Table 1). It verifies that directly regressing translation from image can provide accurate translation. Starting from this point, we unify the different solving strategies into a single model, namely *Coordinates-based Disentangled Pose Network* (CDPN), in which the rotation is indirectly estimated from coordinates while the translation is directly regressed from image. Our approach is able to achieve highly accurate, robust estimation on both translation and rotation. To the best of our knowledge, we are the first to unify the indirect PnP-based strategy and the direct regression-based strategy to estimate object poses.

$$T = \mathcal{F}(\mathbb{K}, P_{u,v}, Q_{x,y,z}) \tag{4}$$

$$T = \mathcal{G}_w(I) \tag{5}$$

where $\mathbb{K}$ is camera intrinsic parameters, $\mathcal{F}$ is the PnP algorithm, I is image and $\mathcal{G}_w$ is network with parameters w.

3.5. Scale-invariant Translation Estimation

Existing approaches [10, 28, 8, 24] that directly regress translation from image are mainly based on the whole image. This strategy requires a separate network based on the whole image for translation, which is quite inefficient. Estimating the translation directly from the detected object is more efficient, but unfortunately, it is problematic. Here, we propose *Scale-Invariant Translation Estimation* (SITE) to achieve highly accurate and efficient translation estimation based on the local image patches. We first calculate the global image information T_G (including position $C_{x,y}$ and size (h, w)) of sampled local patch. Then, additional translation head net is introduced on the backbone to predict the scale-invariant translation $T_S = (\Delta_x, \Delta_y, t_z)$. Δ_x and Δ_y reveal the offset from the bounding box center to the object center. Instead of regressing the absolute offset, the network is trained to predict the relative offset (Eq. 6), which is constant (i.e. scale-invariant) to *Dynamic Zoom In* . t_z is zoomed depth. Finally, the translation $T = (T_x, T_y, T_z)$ can be solved by combining T_S with T_G (Eq. 7).

$$\begin{cases} \Delta_x &= \frac{O_x - C_x}{w} \\ \Delta_y &= \frac{O_y - C_y}{h} \\ t_z &= \frac{T_z}{r} \end{cases} \tag{6}$$

$$\begin{cases} T_x &= (\Delta_x \cdot w + C_x) \cdot \frac{T_z}{f_x} \\ T_y &= (\Delta_y \cdot h + C_y) \cdot \frac{T_z}{f_y} \\ T_z &= r \cdot t_z \end{cases} \tag{7}$$

where (O_x, O_y) and (C_x, C_y) are the projection of object center and the center of the patch in original image. (h, w) is the size of sampled object in original image. r is the resize ratio in DZI. We show the training loss of translation head net in Eq. 8.

$$\begin{aligned} \mathcal{L}_{SITE} = \ell_2(\gamma_1 \cdot (\Delta_x - \tilde{\Delta}_x) + \gamma_2 \cdot (\Delta_y - \tilde{\Delta}_y) + \\ \gamma_3 \cdot (t_z - \tilde{t}_z)) \end{aligned} \tag{8}$$

where $*$ and $\tilde{*}$ represent the predicted and ground-truth value respectively. Our SITE can deal with the case that bounding box center does not coincide with the object center and can handle occlusion situation.

3.6. Training Strategy

We find that the rotation head is more difficult to train compared with translation head. So, we adopt an alternative training strategy: First, we train rotation head with backbone to predict coordinates-confidence map. The backbone is initialized with the weights trained on ImageNet while the head is trained from scratch. Then, we train the translation head from scratch while fixing the backbone. Finally, we finetune the backbone with two heads together.

Method	w/o Refinement								w/ Refinement			
	BB8 [20]	YOLO6D [25]	PoseCNN [28]	SSD6D [8]	AAE [24]	Brachmann [3]	Nigam [18]	Ours	BB8 [20]	SSD6D [8]	Brachmann [3]	AAE [24]
5cm 5°	-	-	19.4	-	-	-	43.7†	**94.31**	69.0	-	40.6	-
ADD	43.6	55.95	62.7	2.42	31.41	32.3	-	**89.86**	62.7	79	50.2	-
2D Proj. 5px.	83.9	90.37	70.2	-	-	69.5	-	**98.10**	89.3	-	73.7	64.67

Table 1: Comparison with state-of-the-art RGB-only methods on LINEMOD using different metrics. (Note: † they only report results on 8 objects.)

4. Data Preparation

Dataset Our experiments are conducted on the LINEMOD dataset and Occlusion dataset. LINEMOD dataset is the *de facto* standard benchmark for 6-DoF object pose estimation of textureless objects in cluttered scenes. We split it into training and test sets according to [3]. The Occlusion dataset is proposed by [3], and it shares the same images with LINEMOD. 8 objects in one video sequence that are heavily occluded are annotated for testing purpose. We follow [13] to split the dataset. Concretely, the test set consists of all occluded images and objects in other video sequences without occlusion constitute the training set.

Synthetic Training Data LINEMOD dataset supplies around 200 training images per class, which is relatively small for training a deep neural network. We randomly rendered 1000 images for each class according to the pose distribution of the training set. Concretely, we calculated the angle range $\mathbb{E}_C$ in the training set for each class C and randomly generated rotation R_C in $\mathbb{E}_C$. The translation T_C was randomly generated according to the mean and variance calculated from the training set. The synthetic image can be rendered using R_C and T_C. For Occlusion dataset, the training images are highly related and lack occlusions while the test images are heavily occluded. To bridge the domain gap, we also rendered images in the similar way except that we chose 3-8 objects to render one image in order to introduce occlusions among objects. For all synthetic images, the backgrounds were randomly replaced with indoor images in the PASCAL VOC2012 dataset during training.

5. Experiments

5.1. Metrics

In our experiments, we use three common metrics for evaluation: 2D Projection, 5cm 5° and ADD. For metric 2D Projection, a pose is considered correct if the average 2D projection error of the object's vertices is smaller than 5 pixels. For 5cm 5°, a pose is considered correct if the errors of the translation and rotation are less than $5cm$ and 5° respectively. The ADD calculates the error in the 3D object space. The prediction is considered correct if the average distance ADD (Eq. 9) of the object's vertices between the predicted pose and the ground-truth pose is below $0.1d$, where d is the diameter of the object model. For symmetric objects, the closest model point is used to compute the average distance (Eq. 10).

$$ADD = \frac{1}{n}\sum_{i=1}^{n}||(\mathbf{R}x_i + \mathbf{T}) - (\tilde{\mathbf{R}}x_i + \tilde{\mathbf{T}})|| \quad (9)$$

$$ADD\text{-}S = \frac{1}{n}\sum_{i=1}^{n}\min_{x_j \in \mathbb{V}}||(\mathbf{R}x_i + \mathbf{T}) - (\tilde{\mathbf{R}}x_j + \tilde{\mathbf{T}})|| \quad (10)$$

where n is the number of object's vertices. x_i is the i-th vertice in 3D model. [**R T**] and [$\tilde{\mathbf{R}}$ $\tilde{\mathbf{T}}$] are the ground-truth pose and predicted pose respectively.

5.2. Experiments on LINEMOD dataset

Ablation Study on *Dynamic Zoom In* and *Masked Coordinates-Confidence Loss*. We conduct comprehensive evaluation for *Dynamic Zoom In* and *Masked Coordinates-Confidence Loss* on LINEMOD dataset. Table 2 shows the results obtained from the coordinates-confidence map via PnP (without *Scale-invariant Translation Estimation*). Training solely on real images, our approach significantly improves the performance from 62.89% to 84.01% on 5cm 5°. And with real+synthetic images, we also surpass the baseline with a large margin (our 94.27% vs. 78.31% on 5cm 5°). We also evaluate the contribution of each component in Table 2. For *Dynamic Zoom In* , we also evaluate the scalability of the network. See the rows 9, 10 in Table 2, utilizing YOLOv3 to provide detections during test, the performance is almost the same with Faster-RCNN. On Tiny YOLOv3, the accuracy only drops a little. We can see that DZI makes our pose network highly modularized and flexible, which can work robustly with quite a lot of detectors without re-training.

Detection VS. Segmentation. Here, we compare our detection-based framework with segmentation-based framework regarding both performance and speed. To build the segmentation-based baseline, we train a Mask-RCNN to provide object segmentation mask. Then, we use this mask to zoom in on the object during training and also use it as confidence map to build 2D-3D correspondences during test. The results are shown in Table 3. Training with DZI, our detection-based framework achieves better performance than segmentation-based one.

Ablation Study on *Scale-invariant Translation Estimation* (SITE). The evaluation results of SITE is shown

Row	Methods				5cm 5°	ADD	2D Proj.
	Syn	DZI	MCC Loss	Detector (test)			
1	✗	✗	✗	F-RCNN	62.89	50.71	87.04
2	✗	✗	✓	F-RCNN	68.51	58.87	88.94
3	✗	✓	✗	F-RCNN	69.61	57.90	89.68
4	✗	✓	✓	F-RCNN	84.01	70.24	94.94
5	✓	✗	✗	F-RCNN	78.31	67.69	92.76
6	✓	✗	✓	F-RCNN	81.75	65.92	94.08
7	✓	✓	✗	F-RCNN	88.71	73.94	95.34
8	✓	✓	✓	F-RCNN	**94.27**	**75.04**	**96.93**
9	✓	✓	✓	YOLOv3	94.02	74.99	96.76
10	✓	✓	✓	Tiny YOLOv3	93.69	74.70	96.82

Table 2: Ablation study on *Dynamic Zoom In* and *Masked Coordinates-Confidence Loss*. (Note: without using *Scale-invariant Translation Estimation*. F-RCNN is Faster-RCNN. When DZI is unemployed, the detections in training are from Faster-RCNN).

Row	Methods			5cm 5°	ADD	2D Proj.	Speed
	Arch.	DZI	MCC Loss				
1	Seg.	-	✗	81.58	66.46	94.24	$\sim 76ms$
2	Seg.	-	✓	85.69	68.42	94.81	$\sim 76ms$
3	Det.	✓	✓	**94.02**	**74.99**	**96.76**	$\sim$ **30ms**

Table 3: Detection vs. segmentation. (Note: without using *Scale-invariant Translation Estimation*. Seg. is segmentation from Mask-RCNN. Det. is detection from YOLOv3)

in Table 4. Using SITE instead of coordinates to estimate translation, the accuracy on ADD improves significantly, from 75.04 to 89.86 . The comparison of each object is shown in Table 5. SITE achieves not only better but also more balanced performance on all objects. For translation, our SITE provides highly accurate estimation on all objects. It fully demonstrates the advantages of our disentangled pose estimation strategy. More detailed and visualized comparison of translation is shown in Fig. 1(b)(c).

Comparison with the State-of-the-art Methods Table 1 summarizes the comparison results between our approach and current State-Of-The-Art (SOTA) RGB-based approaches with and without refinement. We achieve SOTA result and outperform others with a large margin. Compared with coordinates-based methods[3, 18], we surpass them with a significant margin (94.31 vs. 43.7 on 5cm 5° and 89.86 vs. 32.3 on ADD). Even without disentangled translation estimation (i.e. solving 6-DoF from coordinates), our proposed DZI and MCC loss are still able to endow the model with strong coordinates learning ability to achieve remarkable performance (Table 2). Compared with current leading approaches SSD6D[8], BB8[20], PoseCNN[28] and AAE[24], we achieve significantly better performance on all metrics. It is worth noting that even without refinement, our approach still outperforms those approaches [20, 8, 3] refined with depth and ICP.

Row	Methods		5cm 5°	ADD	2D Proj.
	SITE	Detector(Test)			
1	✗	F-RCNN	94.27	75.04	96.93
2	✓	F-RCNN	**94.31**	**89.86**	**98.10**
2	✓	YOLOv3	94.11	89.80	97.57

Table 4: Ablation study on SITE (Note: without SITE, the translation is solved from coordinates via PnP).

Metric	ADD (Rot.&Trans.)		2cm (Trans.)	
Method	w/o SITE	w/ SITE	w/o SITE	w/ SITE
ape	11.43	**64.38**	45.90	**90.29**
benchvise	95.05	**97.77**	92.05	**96.31**
camera	75.49	**91.67**	82.45	**95.49**
can	89.57	**95.87**	89.57	**95.77**
cat	51.50	**83.83**	66.57	**92.81**
driller	93.76	**96.23**	85.73	**91.48**
duck	23.19	**66.76**	60.09	**89.77**
eggbox	99.53	**99.72**	83.47	**94.84**
glue	94.21	**99.61**	61.87	**89.86**
holepuncher	68.22	**85.82**	86.96	**92.58**
iron	93.77	**97.85**	81.10	**94.48**
lamp	97.02	**97.89**	90.69	**93.57**
phone	82.72	**90.75**	78.94	**89.24**
Average	75.04	**89.86**	77.34	**92.81**

Table 5: Evaluate SITE for each object on metric ADD (for 6-DoF pose) and 2cm (for translation). (Note: w/o SITE, the translation is solved from coordinates via PnP.)

5.3. Experiments on Occlusion dataset

The comparison results on Occlusion dataset are shown in Figure 4. Here we compare with others on 2D Projection for some approaches ([25], [20]) only report results on this metric. For strict threshold (e.g. projection error should be less than 5 pixels), our approach performs pretty well on all objects and surpass the competitors by a large margin. It is worth noting that coordinates-based approach NOCS [26] utilizes Mask-RCNN to predict coordinates from RGBD image. Though they use depth, we still outperform them solidly. Despite the fact that BB8[20] leverages the ground-truth bounding box on this task, we still surpass them on most objects.

5.4. Running Time

On a desktop with an Intel 2.0 GHz CPU and a TITAN Xp GPU, given a 640×480 image and utilizing YOLOv3 as detector, our approach takes 30ms to complete pose estimation. It includes: 4ms for data loading, 15ms for detection, 7ms for a forward propagation of CDPN and 4ms for building correspondences and solving pose. Note that our system can be accelerated by utilizing faster detectors. For

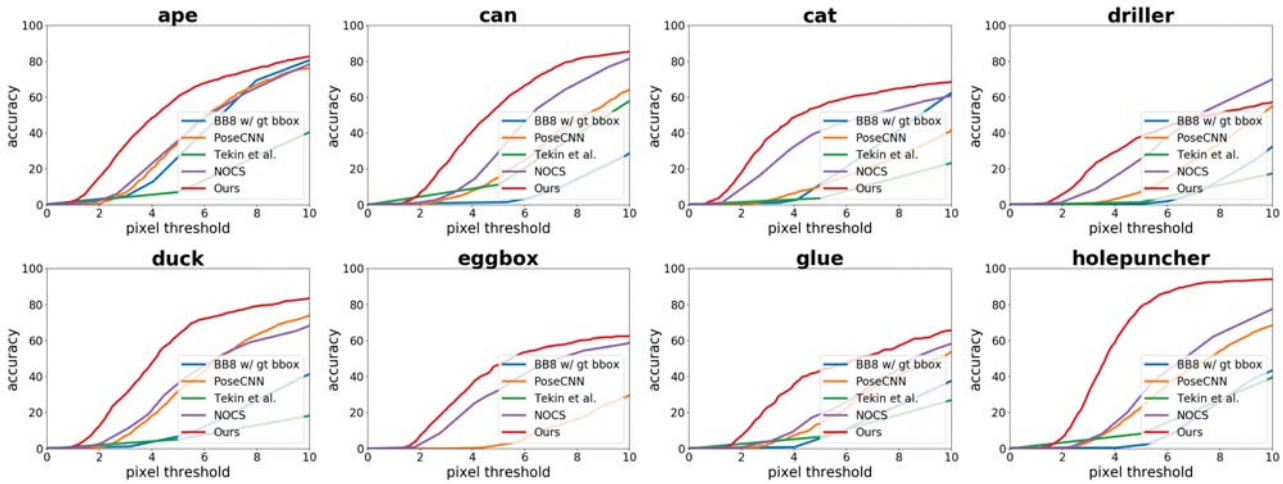


Figure 4: Comparison with state-of-the-art RGB or RGB-D-based methods on the Occlusion dataset [3]. Metric used is 2D Projection.

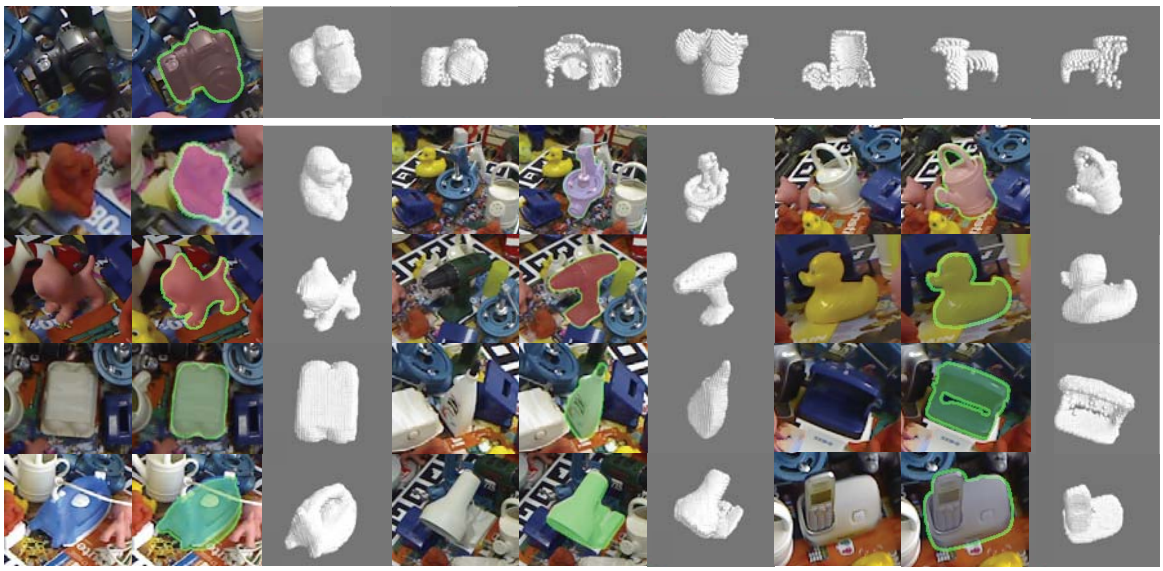


Figure 5: Qualitative results for 6-DoF pose estimation and 3D coordinates regression. For each category, we show: the query object in RGB image; the pose estimation result and the estimated 3D coordinates. The pose result is shown by projecting the model using the estimated pose. For the first row, we also show the generated 3D coordinates in various views from front, back, top, bottom, left and right respectively.

Tiny YOLOv3 detector, the detection only takes 3ms. Thus, the total time can be shortened to18ms.

6. Conclusion

In this work, we develop a coordinates-based 6-DoF pose estimation approach by disentangling the rotation and translation. We propose *Dynamic Zoom In* , *Masked Coordinates-Confidence Loss* and *Scale-invariant Translation Estimation* to achieve highly accurate, robust and efficient pose estimation. Our method is able to deal with texture-less and occluded objects and achieves state-of-the-art results on all commonly used metrics on LINEMOD. For the future, we expect our approach to be applied to other related domains (e.g. camera localization).

7. Acknowledgement

This work was supported in part by the Beijing Municipal Science and Technology Commission under Grant Z181100008918014, in part by the National Key R&D Program of China under Grant 2017YFB1002202, and in part by the Projects of International Cooperation and Exchanges NSFC under Grant 61620106005. Additionally, we are also very grateful to Jie Wang for her patience and encouragement for this work.

References

[1] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, 1992.

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. 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.

[4] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. *CoRR*, 2017.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Thanh-Toan Do, Trung Pham, Ming Cai, and Ian D. Reid. Lienet: Real-time monocular object instance 6d pose estimation. In *Proceedings of British Machine Vision Conference (BMVC)*, 2018.

[7] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 2006.

[8] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making rgb-based 3D detection and 6D pose estimation great again. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[10] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[12] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning analysis-by-synthesis for 6d pose estimation in RGB-D images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[13] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[14] Zhigang Li, Yuwang Wang, and Xiangyang Ji. Monocular viewpoints estimation for generic objects in the wild. *IEEE Access*, 2019.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[16] Siddharth Mahendran, Haider Ali, and René Vidal. 3d pose regression using convolutional neural networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[18] Apurv Nigam, Adrian Penate-Sanchez, and Lourdes Agapito. Detect globally, label locally: Learning accurate 6-dof object pose estimation by joint segmentation and coordinate regression. *IEEE Robotics and Automation Letters (RAL)*, 2018.

[19] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017.

[20] Mahdi Rad and Vincent 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.

[21] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Hao Su, Charles Ruizhongtai Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[24] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2018.

[25] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] Yaming Wang, Xiao Tan, Yi Yang, Xiao Liu, Errui Ding, Feng Zhou, and Larry S Davis. 3d pose estimation for fine-

grained object categories. In *Proceedings of The European Conference on Computer Vision Workshop (ECCVW)*, 2018.

[28] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018.

[29] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.