Trans6D

Zhang, Zhongqun; Chen, Wei; Zheng, Linfang; Leonardis, Ales; Chang, Hyung Jin

DOI:
10.1007/978-3-031-25085-9_7

Trans6D: Transformer-based 6D Object Pose Estimation and Refinement

Zhongqun Zhang[1], Wei Chen[2], Linfang Zheng[1,3], Aleš Leonardis[1], and Hyung Jin Chang[1]

[1] University of Birmingham, UK
[2] Defense Innovation Institute, China
[3] SUSTech, China

Abstract. Estimating 6D object pose from a monocular RGB image remains challenging due to factors such as texture-less and occlusion. Although convolution neural network (CNN)-based methods have made remarkable progress, they are not efficient in capturing global dependencies and often suffer from information loss due to downsampling operations. To extract robust feature representation, we propose a Transformer-based 6D object pose estimation approach (Trans6D). Specifically, we first build two transformer-based strong baselines and compare their performance: pure Transformers following the ViT (Trans6D-pure) and hybrid Transformers integrating CNNs with Transformers (Trans6D-hybrid). Furthermore, two novel modules have been proposed to make the Trans6D-pure more accurate and robust: (i) a patch-aware feature fusion module. It decreases the number of tokens without information loss via shifted windows, cross-attention, and token pooling operations, which is used to predict dense 2D-3D correspondence maps; (ii) a pure Transformer-based pose refinement module (Trans6D+) which refines the estimated poses iteratively. Extensive experiments show that the proposed approach achieves state-of-the-art performances on two datasets.

Keywords: 6D Object Pose Estimation; Transformer

1 Introduction

In this paper, we are interested in estimating the 6D pose of objects from monocular RGB images. 6D object pose estimation has been gaining attention as it can be applied in many fields such as augmented reality, robotic manipulation, autonomous driving, etc. However, estimating the 6D pose from a monocular RGB image is still challenging, especially when the target object is under heavy occlusion or in changing illumination conditions.

With the explosive growth of deep learning, deep Convolutional Neural Networks (CNNs) have greatly improved monocular 6D object pose estimation [23], even at times surpassing RGB-D-based methods [23,1,32]. Recent works in this field can be roughly divided into two categories: i) approaches that use a CNN to regress the 6D poses directly [41,39,38,37,8,34,5] and ii) indirectly [31,30]. A

common drawback of the methods in the first category is their poor generalization ability due to the large search space of the rotation [41,15]. The second category overcomes this limitation by either utilizing CNNs to detect the 2D keypoints of the objects [35,31] or estimating the dense 2D-3D correspondence maps between the input image and the available 3D models [18,43]. Given the correspondences, one can recover the 6D pose parameters via Perspective-n-Point (PnP) algorithm [2]. Great success has been achieved with these approaches. However, these CNN-based methods are not efficient in capturing non-local spatial relationships. Therefore, their performance is limited in both accuracy and robustness. The codes will be publicly available on our website.

Even though Transformer [11] is originally designed for *Natural Language Processing* (NLP) tasks, recent works [14,9,42,27,4,22] show that both "Pure Transformer" and "CNN+Transformer" have the potential to become the universal models for computer vision tasks. The self-attention mechanism [42] makes them particularly effective in capturing the global dependencies. Therefore, we are interested in designing an algorithm that can leverage the advantage of Transformers in the 6D object pose estimation task.

In this work, we propose Trans6D, a simple yet effective framework that employs Transformers for 6D pose estimation. Firstly, we build two strong baseline frameworks (Trans6D-pure and Trans6D-hybrid) using Transformer. Among them, Trans6D-pure applies the ViT [12] directly, while Trans6D-hybrid uses ResNet-34 to learn local feature maps and then uses Transformer encoders to capture global dependencies. Since dividing the image into small patches will impact the accuracy of 6D object pose regression tasks, the Trans6D-Hybrid significantly outperforms the Trans6D-pure.

Secondly, we propose two novel modules to improve the performance of the Trans6D-pure: (i) a patch-aware feature fusion (PAFF) module is proposed to predict the 2D-3D correspondence map. We reshape the tokens from the last layer of Trans6D-pure into feature maps. Inspired by the pooling operation and stridden convolution in CNN, the PAFF module proposes to use token pooling and shifted windows to reduce the dimension of feature maps. To avoid information loss and alleviate the impact of image division, the PAFF module uses cross-attention to fuse the local feature of tokens in each window. (ii) The pure Transformer-based pose refinement module (Trans6D+) is introduced to use the input image and the initial pose estimation to learn an accurate transformation between the predicted object pose and the ground-truth pose.

Experimental results on two widely used benchmark datasets, LINEMOD [16] and Occlusion LINEMOD [3], demonstrate that Trans6D has state-of-the-art performances. The contributions of the paper are summarised:

- Two Transformer-based strong baselines are proposed and assessed for 6D object pose estimation, which achieve comparable performance with CNN-based frameworks.
- A patch-aware feature fusion (PAFF) module is designed to decrease the number of tokens without information loss, which is used to predict dense 2D-3D correspondence maps.

- A pure Transformer-based Refinement (Trans6D+) module is introduced to refine the estimated poses iteratively.
- For the first time, we show a simple but effective method based on Transformers achieves state-of-the-art performance, 96.9% on the LINEMOD dataset and 57.9% on the Occlusion LINEMOD dataset.

2 Related Work

2.1 6D Object Pose Estimation

Recent work in this field can be roughly divided into two categories: direct methods and correspondence-based approaches.

Direct methods use CNN-based networks to regress a 6D pose from a single RGB image directly. For Instance, PoseCNN [41] predicted 2D localization, depth information, and rotation from a CNN backbone. However, the direct methods usually exhibit poor generalization ability due to the large search space of the rotation and the lack of depth information. Therefore, Oberweger et al. [29] and BB8 [32] attempted to limit the rotation range; they discretize the pose space and seem the prediction of rotation as classification rather than regression. G2L-Net [6] harnesses the embedding vector features to regress the 6D pose. GDR [40] regress dense correspondences first and then use patch-PnP to learning 6D pose from the correspondences.

The correspondence-based methods are prevalent recently. These methods first built 2D-3D correspondence maps, then computed the 6D pose via PnP with the RANSAC algorithm. For example, Pixel2Pose [30] used an auto-encoder architecture to estimate the 3D coordinates per pixel to build dense correspondences, while PVNet [31], PVN3D [15], and PointPoseNet [7] adopted a voting net to select 2D keypoint or 3D keypoint respectively to build dense correspondences. Furthermore, HybridPose [35] suggested predicting hybrid correspondences (including keypoints, edge vectors, and symmetry) to enhance the robustness. DPOD [43] first estimateed multi-instance correspondences and then used a learning-based refiner to improve the accuracy. It is the first approach that unifies the correspondence-based rotation estimation and the direct regression-based translation estimation.

2.2 Vision Transformer

Transformer architecture was built for the NLP task, consisting of the multi-head self-attention mechanisms and feed-forward layer, to capture the long-term correlation between words. Recently, there is a growing interest in Transformer based computer-vision tasks. On the one hand, pure Transformer [12,21,24] is attracting more and more attention. ViT [12] applied pure Transformer directly to image classification tasks and attains excellent results compared to the state-of-the-art CNN-based method. TransReID [14] showed that a pure Transformer-based model can be used for the object ReID task. On the other hand, "CNN +

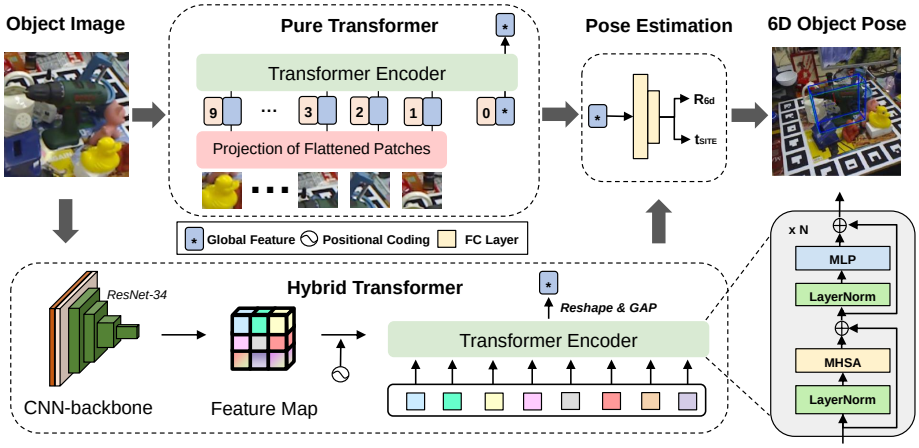

Fig. 1. An overview of the proposed Trans6D-pure and Trans6D-hybrid baselines. Given an input image of the target object, Trans6D-pure encodes the image as a sequence of patches and then models the global dependencies among each patch via ViT-like Transformer Layers. Trans6D-hybrid first extracts feature maps using a CNN-based backbone and flatten them to a sequence, then feeds the sequence into Transformer Layers. Output token marked with $*$ is served as the global feature. The global feature is then used to regress 3D rotation and 3D translation.

Transformer" [4,44] also has better performance. DETR [4] extracted features from CNN-backbone, then viewed object detection as a direct set prediction problem that was suitable for Transformer structure. TransPose [42] predicted 2D heatmaps for human pose estimation by Transformer encoder and CNN, while METRO [27] tried to model non-local interactions among body joints and mesh vertices for human mesh reconstruction.

3 Methodology

In Figure1 and Figure2, we show an overview of our proposed framework that estimates the 6D object pose from a single RGB image. The input to Trans6D is an image of size 256×256 which contains only one object with a known class. The outputs of Trans6D-pure and Trans6D-hybrid are the predicted 3D translation and 3D rotation, while the outputs of Trans6D and Trans6D+ are 2D-3D correspondence maps of size 64×64. Given the correspondences, 6D pose is calculated by PnP and RANSAC algorithm. Our method consists of three modules: Two Transformer-based strong baselines, the Patch-Aware Feature Fusion (PAFF) Module, and the Pure Transformer-based Pose Refinement Module. In the following subsections, we describe each module in detail.

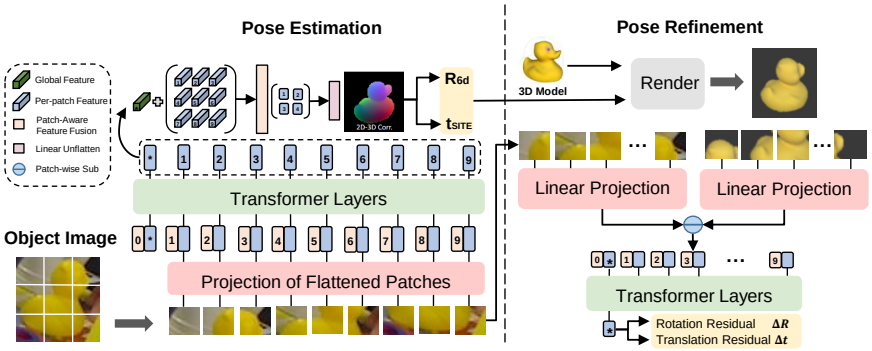

Fig. 2. An overview of the proposed Trans6D and Trans6D+. Trans6D is based on Trans6D-pure. The outputs of Trans6D-pure are reshaped as a feature map and feed it into patch-aware feature fusion (PAFF) module to downsample the tokens. Instead of directly regressing the 6D pose, Trans6D predicts the 2D-3D correspondence maps and compute the 6D pose by PnP algorithm. Trans6D+ renders the object at the estimated pose and learns to align the real image and the rendered image incrementally.

Following GDR-Net [40], the 6D pose is represented as a decoupled way, composed of a continuous 6-dimensional representation for rotation $\mathbf{R}_{6\mathrm{d}}$ in $SO(3)$ and a scale-invariant representation for translation $\mathbf{t}_{\mathrm{SITE}}$.

3.1 Transformer-based Baselines for 6D Object Pose Estimation

We build two Transformer-based strong baselines for 6D object pose estimation, which are based on pure Transformers following the ViT (Trans6D-pure) and hybrid Transformers integrating CNNs with Transformers (Trans6D-hybrid) separately.

Trans6D-hybrid Trans6D-hybrid consists of two components: a CNN backbone to extract low-level image feature maps; a Transformer Encoder to capture global dependencies between feature vectors, and each feature vector is distinguished by the position embedding, as shown in Figure 1.

We employ a Convolutional Neural Network (CNN), ResNet34 [13], as the backbone for feature extraction. This backbone is pre-trained on ImageNet classification task [10], therefore Transformer can easily benefit from large-scale pre-trained CNNs. Given an initial image $x_{\mathrm{img}} \in \mathbb{R}^{3\times H_0\times W_0}$, the CNN-based backbone generates a feature map with lower-resolution $f \in \mathbb{R}^{C\times H\times W}$. Specifically, $C = 512$ and $H, W = \frac{H_0}{32}, \frac{W_0}{32}$.

The Transformer encoder is employed to model interaction among all the pixel-level features in the image. First, we transform the feature dimension to $f \in \mathbb{R}^{d\times H\times W}$ by a 1×1 convolution. Since the Transformer encoder expects a sequence as input, we then flattened the feature into a sequence $f \in \mathbb{R}^{L\times d}$, where $L = H \times W$, and this sequence is fed into the Transformer Encoder. As

shown in Figure 1, we follow the standard Transformer architecture as closely as possible, which consists of a multi-head self-attention module and a fully connected feed-forward network (FFN).

Since the Transformer architecture is permutation-invariant, position embedding aims at giving an order to the sequence of the image feature map. Following [4], 2D Sine position embedding is used in Trans6D-hybrid. The output sequence of Transformer is reshaped to feature maps. We utilize a global average pooling operation to extract the global feature, which is fed into two FC layers to regress the 6D pose. The whole process can be formulated as:

$$\mathbf{R_{6d}}, \mathbf{t_{SITE}} = \text{FCLayers}(\text{GAP}(\text{Transformer}(\boldsymbol{f}))) \tag{1}$$

Trans6D-pure Given an image $x_{\text{img}} \in \mathbb{R}^{C \times H_0 \times W_0}$, Trans6D-pure first divides the images into non-overlapping $P \times P$ patches $\left\{x_n^i \mid i = 1, 2, \cdots, P\right\}$. These patches is then flattened into a sequence of 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$ by a linear projection $\mathcal{F}$, P^2 is dimension of each feature vector and $N = HW/P^2$. An extra learnable embedding token $(*)$ is added into the input sequences and this token is used to extract a global feature. Different from Trans6D-hybrid, position embeddings $z_{\text{pos}} \in \mathbb{R}^{\left(1+P^2\right) \times NC}$ in Tran6D-pure are learnable. Overall, The input feature matrix can be formulated as:

$$\mathcal{Z} = \left[x_*; \mathcal{F}\left(x_n^1\right); \mathcal{F}\left(x_n^2\right); \cdots; \mathcal{F}\left(x_n^P\right)\right] + z_{pos} \tag{2}$$

where $[\,]$ represented concatenation operations. Then, we feed the feature into Transformer encoder layers. The whole process can be express as:

$$\mathbf{R_{6d}}, \mathbf{t_{SITE}} = \text{FCLayers}(\text{Transformer}(\boldsymbol{\mathcal{Z}})[0, :]) \tag{3}$$

3.2 Patch-Aware Feature Fusion

Instead of directly regressing the 6D pose, Trans6D predicts 2D-3D correspondence maps. Standard ViT architecture [12] cannot be directly applied in a dense prediction task because they use a constant dimensionality of the hidden embeddings for all transformer layers. However, downsampling operations in CNNs (e.g. pooling) suffer from information loss. Furthermore, the patch division might also greatly impact the prediction because it corrupts the image structure. To solve the aforementioned problems, We design a patch-aware feature fusion (PAFF) module that can not only downsample the number of tokens without information loss but also alleviate the impact by patch division, as shown in Figure 3.

The output of Trans6D-pure consists of the global token Z^* and a sequence of patch tokens $\left\{Z^i \mid i = 1, 2, \cdots, N\right\}$. As shown in Figure 2, we first reshape the patch tokens$Z \in \mathbb{R}^{l \times c}$ as a feature map $f \in \mathbb{R}^{h \times w \times c}$ in the spatial dimension. After obtaining the re-organized feature map, we apply token pooling operation (1D convolution with 1D maxpooling) to downsample the feature map to $Z' \in \mathbb{R}^{\frac{h}{4} \times \frac{w}{4} \times c}$. Z' will serve as the learnable queries $\left\{q^i \mid i = 1, 2, \cdots, \frac{N}{4}\right\}$ in

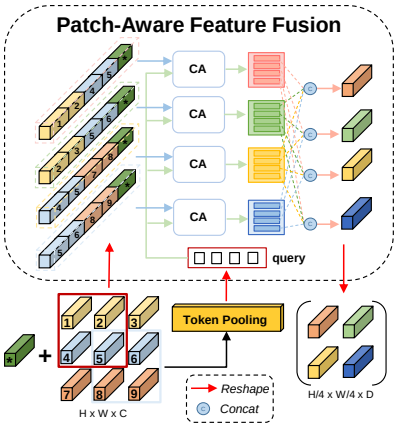

Fig. 3. Illustration of Patch-Aware Feature Fusion. We design a PAFF module which can not only downsample the number of tokens without information loss but also alleviate the impact by patch division.

PAFF module. However, simply adding a token pooling will decrease the model's representation ability. Inspired by the strided convolution in CNN, we split the feature map into overlapping patches with a sliding window. As shown in Figure 3, the patch tokens in each window are fed into PAFF module with the global token. Supposing each window contains $k \times k + 1$ patches and the sliding stride is s, the number of windows is

$$\frac{h}{4} \times \frac{w}{4} = \left\lfloor \frac{h-k}{s} + 1 \right\rfloor \times \left\lfloor \frac{w-k}{s} + 1 \right\rfloor . \tag{4}$$

The tokens of each sliding window server as query and key, and we compute their cross-attention (CA) [4] with the learnable queries. Thus the local information can be aggregated. The outputs of CA network is concatenated with each other, then they will be reshaped again to feature maps $f' \in \mathbb{R}^{\frac{h}{4} \times \frac{w}{4} \times D}$ and regress to 2D-3D correspondences map $M_{2D3D} \in \mathbb{R}^{\frac{h}{4} \times \frac{w}{4} \times 4}$ by Linear Unflatten.

3.3 Pure Transformer-based Pose Refinement

In order to further improve the performance of Trans6D, we propose a pure Transformer-based pose refinement (Trans6D+) module. Inspired by DPOD [43], our refiner aims at regress the residual of rotation and translation with the loss function:

$$\mathcal{R}_{pose} = \frac{1}{M} \sum_{j} \min_{x \in \mathcal{M}_s} \left\| (Rx_j + t) - \left(\hat{R}_i x_k + \hat{t}_i \right) \right\| \tag{5}$$

where $\mathcal{M}_s$ represents the randomly selected 3D points from the object's 3D model. R and t is the the ground truth of 6D pose, while $\hat{R}$ and $\hat{t}$ is the predicted rotation and translation.

In Figure 2, we show the pipeline of our Trans6D+. Given the 3D model of an object and the predicted 6D pose parameters, we first render the object at the initial pose and crop the image around the object. The rendered image embeds the initial pose information and our idea is to make the Trans6D+ learning to align these two images incrementally. Trans6D+ contains two parallel Transformer branches, one receives the real image as input while the other extracts the feature from rendered image. Similar to Tran6D-pure, rendered image is also divided into the same number of patches with real image branch ($\mathbf{f}_c$), and then flattened them into a sequence of 2D patches $\mathbf{f}_r \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$ by a linear projection. Then patches from two branches are subtracted and fed into the Transformer encoder layers to extract the global dependencies. An extra learnable embedding token $\mathbf{x}_*$ is added into the input sequences and this token is used to extract a global feature. Finally, the residual of rotation ΔR and translation Δt is regressed by the global feature. The whole process can be formulated as:

$$\mathbf{\Delta R}, \mathbf{\Delta t} = \text{FCLayers}(\text{Transformer}(\boldsymbol{f_c} - \boldsymbol{f_r})[0, :]) \tag{6}$$

3.4 Training

To train the transformer encoder, we apply loss functions on top of the transformer outputs, the network predicts a confidence value for each pixel to indicate whether it belongs to the object. The corresponding loss function is defined as

$$\begin{aligned}\mathcal{L}_{Corr} =& \alpha \cdot \ell_1 \left(\sum_{j=1}^{n_c=3} \left(\bar{M}_{\text{vis}} \circ \left(\hat{M}_{\text{XYZ}_j} - \bar{M}_{\text{XYZ}_j} \right) \right) \right), \\ &+ \beta \cdot \ell_1 \left(\hat{M}_{\text{vis}} - \bar{M}_{\text{vis}} \right)\end{aligned} \tag{7}$$

where $\bar{M}_{XYZ}$ represents the 3D coordinate of the available 3D model, $\bar{M}_{vis}$ represents the visible masks.

Once we get the confidence map and the correspondence map, the coordinates belong to the object can be obtained by setting a threshold for the confidence. However, the size of the object in an RGB image is different from the original image since we use a detector. To build the 2D-3D correspondences, we map the pixel from the coordinates map back to the RGB image.

4 Experiments

In this section, we first conduct ablation studies on the effectiveness of each module in Trans6D, and then evaluate Trans6D on prevalent benchmark datasets. The results show that our method achieves state-of-the-art performance.

4.1 Implementation Details

We implemented our framework using Pytorch and conducted all the experiments on an Intel i7-4930K CPU with one GTX 2080 Ti GPU. During training,

we use Adam [28] for optimization. Also, we set the initial learning rate as 1e-4 and halve it every 50 epochs. The maximum epoch is 300. For object detection part, we fine-tune the YOLO-V3 [33] architecture which is pre-trained on the ImageNet [10] to locate the 2D object and fixed its size to 256×256.

4.2 Datasets

LINEMOD: is a widely used dataset for 6D object pose estimation. It consists of 13 objects, each containing about 1.2k images with ground-truth poses for a single object. This dataset exhibits many challenges for pose estimation: textureless objects, cluttered scenes, and lighting condition variations. Following [26], we select 15% of the RGB images for training and 85% for testing. We also render 1000 images for each object as a supplement to the training set.
Occlusion LINEMOD: is a widely used dataset for 6D object pose estimation under severe occlusion. It consists of 8 objects, each containing about 1214 images with more occlusion are provided for testing. All Occlusion LINEMOD datasets are used for testing.

4.3 Evaluation Metrics

We evaluate Trans6D using average 3D distance of model points (ADD) metric [17].
ADD Metric. This metric computes the mean distance between two transformed object model using the estimated pose and the ground-truth pose. When the distance is less than 10% of the model's diameter, it is claimed that the estimated pose is correct. It can be computed by:

$$\frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{R} \cdot \mathbf{x} + \mathbf{T}) - (\widetilde{\mathbf{R}} \cdot \mathbf{x} + \widetilde{\mathbf{T}})\|, \tag{8}$$

where $|\mathcal{M}|$ is the number of points in the object model. x represents the point in object 3D model, $\mathbf{R}$ and $\mathbf{T}$ are the ground truth pose, and $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{T}}$ are the estimated pose. For symmetric objects, we use the ADD-S metric [2], where the mean distance is computed based on the closest point distance. :

$$\frac{1}{|\mathcal{M}|} \sum_{\mathbf{x}_1 \in \mathcal{M}|} \min_{\mathbf{x}_2 \in \mathcal{M}} \left\| (\mathbf{R} \cdot \mathbf{x}_1 + \mathbf{T}) - \left(\widetilde{\mathbf{R}} \cdot \mathbf{x}_2 + \widetilde{\mathbf{T}} \right) \right\|, \tag{9}$$

4.4 Ablation Studies

Compared to other methods [26], our proposed Trans6D has three novelties.

First, we build two Transformer-based baselines for 6D pose estimation: pure Transformers-based Trans6D-pure and Trans6D-hybrid combining CNNs with Transformers. As shown in Table 1, we compare Trans6D-hybrid (ResNet34 + Transformer) with CNN-based method (ResNet34 + CNN Head) using the

Table 1. Ablation studies of the effectiveness of two Transformer-based baselines on LINEMOD dataset. The metric we used to measure performance is ADD(-S) metric. 'CNN" means CNNs-based method,'Trand6D-p"and 'Trans6D-h" denote Trans6D-pure and Trans6D-hybrid, respectively.

Metric	ADD(-s)		
Method	CNN	Trans6D-p	Trans6D-h
ape	11.43%	42.33%	**49.11%**
benchvise	95.05%	94.18%	**96.46%**
camera	75.49%	88.04%	**91.84%**
can	89.57%	92.87%	**95.02%**
cat	51.50%	77.73%	**81.51%**
driller	**97.36**%	94.01%	96.79%
duck	23.19%	54.34%	**55.99%**
eggbox	**99.53**%	96.63%	98.75%
glue	**94.2**1%	89.14%	93.02%
holepuncher	68.22%	87.26%	**89.49%**
iron	93.77%	94.80%	**96.75%**
lamp	97.02%	94.86%	**98.21%**
phone	82.72%	90.15%	**92.96%**
Average	75.04%	84.34%	**87.73%**

Table 2. Ablation studies of the effectiveness of patch-aware feature fusion (PAFF) Module on Occlusion LINEMOD dataset. The metrics we used to measure performance are ADD(-S). "SW" means sliding windows operation, "TP" means token pooling operation and "CA" means cross-attention network.

Method	SW	TP	CA	ADD(-S)
EXP1	×	✓	×	40.9%
EXP2	✓	✓	×	43.1%
EXP3	×	✓	✓	45.4%
EXP4	✓	×	✓	**50.6**%

Table 3. Ablation studies of the effectiveness of pure Transformer-based refinement (Trans6D+) Module on LINEMOD dataset. The metrics we used to measure performance are ADD(-S). Compared approaches: PVNet [31], DPOD [43], DeepIm [25]

PVNet	+DPOD	+DeepIm	+Trans6D+
85.56%	92.83%	87.13%	**95.95**%

DPOD	+DPOD	+DPIM	+Trans6D+
82.98%	95.15%	88.6%	**95.86**%

Table 4. Quantitative evaluations on LINEMOD dataset. We use ADD metric to evaluate the methods. For symmetric objects *Egg Box* and *Glue*, we use the ADD-S metric. Note that, we summarize the pose estimation results reported in the original papers on LINEMOD dataset. Baseline approaches: BB8 [32], Pix2Pose [30], DPOD [43], PVNet [31], CDPN [26], Hybrid [35], GDRN [40].

Method	BB8	Pix2Pose	DPOD	PVNet	CDPN	Hybrid	GDRN	Trans6D	Trans6D+
Refinement	×	×	×(✓)	×	×	✓	×	×	✓
Ape	40.4%	58.1%	53.3% (87.7%)	43.6%	64.4%	63.1%	-	68.1%	**88.3**%
Bench Vise	91.0%	97.5%	95.3% (95.3%)	99.9%	97.8%	**99.9**%	-	99.5%	99.4%
Camera	55.7%	60.9%	90.4% (96.0%)	86.9%	91.7%	90.4%	-	93.7%	**97.8**%
Can	64.1%	84.4%	94.1% (**99.7**%)	95.5%	95.9%	98.5%	-	99.4%	99.1%
Cat	62.6%	65.0%	60.4% (**94.7**%)	79.3%	83.8%	89.4%	-	87.9%	93.2%
Driller	74.4%	76.3%	97.7% (98.8%)	96.4%	96.2%	98.5%	-	97.1%	**99.5**%
Duck	44.3%	43.8%	66.0% (86.3%)	52.6%	**66.8**%	65.0%	-	67.9%	**87.8**%
Egg Box	57.8%	96.8%	99.7% (99.9%)	99.2%	99.7%	**100.0**%	-	100%	**100**%
Glue	41.2%	79.4%	93.8% (96.8%)	95.7%	99.6%	98.8%	-	98.3%	**99.8**%
Hole Puncher	74.8%	52.8%	65.8% (86.9%)	81.9%	85.8%	89.7%	-	93.5%	**96.7**%
Iron	84.7%	83.4%	99.8% (**100**%)	98.9%	97.9%	**100.0**%	-	99.9%	99.9%
Lamp	76.5%	82.0%	88.1% (96.8%)	99.3%	97.9%	99.5%	-	99.5%	**99.7**%
Phone	54.0%	45.0%	74.2% (94.7%)	92.4%	90.8%	94.9%	-	98.7%	**99.5**%
Average	62.7%	72.4%	83.0% (95.2%)	86.3%	89.9 %	91.3%	93.7%	92.6%	**96.9%**

Table 5. Quantitative evaluations on Occlusion LINEMOD dataset. We use the ADD metric to evaluate the methods. For symmetric objects *Egg Box* and *Glue*, we use the ADD-S metric. Note that, we summarize the pose estimation results reported in the original papers on LINEMOD dataset. Baseline approaches: PoseCNN [41], Pix2Pose [30], DPOD [43], PVNet [31], Single-Stage [19], HybridPose [35], GDRN [40].

Method	PoseCNN	Pix2Pose	PVNet	Single-Stage	DPOD	Hybrid	GDRN	Trans6D	Trans6D+
Refinement	×	×	×	×	✓	✓	×	×	✓
Ape	9.6%	22.0%	15.8%	19.2%	-	20.9%	**46.8**%	31.2%	36.9%
Can	45.2%	44.7%	63.3%	65.1%	-	75.3%	90.8%	85.1%	**91.6**%
Cat	0.9%	22.7%	16.7%	18.9%	-	24.9%	40.5%	38.3%	**42.5**%
Driller	41.4%	44.7%	65.7%	69.0%	-	70.2%	**82.6**%	66.5%	70.8%
Duck	19.6%	15.0%	25.2%	25.3%	-	27.9%	**46.9**%	35.0%	41.1%
Egg Box	22.0%	25.2%	50.2%	52.0%	-	52.4%	54.2%	52.9%	**56.3**%
Glue	38.5%	32.4%	49.6%	51.4%	-	53.8%	**75.8**%	54.3%	62.0%
Hole Puncher	22.1%	49.5%	39.7%	45.6%	-	54.2%	60.1%	57.7%	**61.9**%
Average	24.9%	32.0%	40.8%	43.3%	47.3 %	47.5%	**62.2**%	52.6%	57.9%

same backbone, and also assess the performance of Trans6D-pure and Trans6D-hybrid. We observe that (i) using Transformer instead of CNN to estimate 6D parameters increased the accuracy from 75.04% to 87.73% when evaluated with ADD(-S) metric on LINEMOD dataset. (ii) The Trans6D-Hybrid (87.73%) significantly outperforms the Trans6D-pure(84.34%). The reason is that the pure Transformer-based method divides the image into small patches, which seriously impact the accuracy of regression tasks.

Table 6. Quantitative evaluations on YCB-Video Datasets. We use 2D-Proj, ADD AUC and ADD(-S) metrics. The threshold of the ADD(-S) metric is 2 cm. Note that, we summarize the pose estimation results reported in the original papers on LINEMOD dataset. State-of-the-art approaches: PoseCNN [41], DeepIM [25], Ameni et al. [36], PVNet [31], Singel-Stage [20], GDR-Net [40]

Methods	PoseCNN	DeepIM	PVNet	Single-Stage	GDR-Net	Ameni et al.	Trans6D	Trans6D+
2D-Proj.	-	-	47.4%	-	-	55.6%	53.2%	**62.4**%
ADD AUC	61.3%	81.9%	73.4%	-	84.4%	83.1%	82.5%	**85.9**%
ADD(-S)	21.3%	-	-	53.9%	60.1%	73.6%	67.7%	**75.2**%

Second, we propose the patch-aware feature fusion (PAFF) module, which is used to predict dense 2D-3D correspondence maps. PAFF module decreases the number of tokens without information loss via shifted windows, cross-attention, and token pooling operations. We compare the impact of the three operations and show the results in Table 2. As it can be seen that using the token pooling operation only exhibits the worst performance since it will decrease the model's representation ability. When combining the cross-attention with either sliding window or token pooling operations, the method has better performance than combining the sliding window and token pooling. It is because that the cross-attention can aggregate the local information. Therefore, combining those three operations can avoid information loss and alleviate the impact of image division. Moreover, cross-attention play a decisive role.

Third, we propose a pure Transformer-based pose refinement module. In table 3, we compare our Trans6D+ with DPOD and DeepIM using the same initial pose and number of iterations. Trans6D+ achieves almost 13% improvement on the PVNet while DPOD only has 7% improvement.

4.5 Comparison with State of the Arts

Object 6D pose estimation on LINEMOD: In Table 4, we compare our approach with state-of-the-art methods on LINEMOD Dataset. We use 15% of each object sequence to train and the rest of the sequence to test on LINEMOD dataset following other methods. The numbers in brackets are the results without post-refinement.

We use Trans6D-pure as our baseline. From Table1 and Table 4, we can see that Trans6D outperforms the baseline by 8.3% in ADD metric. Trans6D also outperforms Pix2Pose by 20.2% predicts the 2D-3D correspondence by CNN encoder, while we use Transformer to regress such correspondence. Trans6D+ improves the Trans6D by 4.3% and achieves the state-of-the-art performance. Comparing to the second-best method GDR-Net [40] that using hybrid presentation to estimate the 6D pose, our method outperforms it by 3.2% in ADD accuracy. DPOD has two results, one is only using correspondence to estimate the 6D pose and Trans6D outperforms it by 9.6%, the other is a result after

Fig. 4. Qualitative pose estimation results on LINEMOD and Occlusion LINEMOD dataset. Green 3D bounding boxes denote ground truth. Blue 3D bounding boxes represent our results. Our results match ground truth well.

refinement and Trans6D+ outperforms it by 1.7%. In Figure 4, we provide a visual comparison of predict pose versus ground truth pose.

Object 6D pose estimation on Occlusion LINEMOD: We use the model trained on the LINEMOD dataset for testing on the Occlusion LINEMOD dataset. Table 5 compares our method with other state-of-the-art methods [31,40] on Occlusion LINEMOD dataset in terms of ADD metric. From Table 5, we can see that Trans6D+ achieves a comparable accuracy (57.9%) and Trans6D achieves 52.6%. The improved performance demonstrates that the proposed method, enables Trans6D robust to partial occlusion.

Object 6D pose estimation on YCB-Video [41] dataset: YCB-Video dataset contains 92 real video sequences for 21 YCB object instances. This dataset is challenging due to the image noise and occlusions. By following PoseCNN, we report the results on three metrics, 2D-Proj, ADD AUC and ADD(-S) metrics. From Table6, we can see that Trans6D outperforms the baseline, PoseCNN, by 53.9% in ADD(-s) metric. Trans6D+ improves the Trans6D by 7.5% and achieves the state-of-the-art performance. Comparing to the second-best method, Ameni et al. [36] which also using pose refinement, our method outperforms it by 1.6% in ADD(-s) accuracy and 2.8% in AUC of ADD metric.

5 Conclusions

In this paper, we present a novel 6D object pose estimation framework build upon Transformers. We first construct two Transformer-based baselines and then compare their performance. One of the baselines uses pure Transformers (Trans6D-pure), and the other integrates CNNs with Transformers (Trans6D-hybrid). Then, we introduce two novel modules to improve the performance of the Trans6D-pure. The first is the patch-aware feature fusion (PAFF) module, which predicts the 2D-3D dense correspondence maps without information loss. The second is the pure Transformer-based pose refinement (Trans6D+) module, which iteratively refines the estimated pose. Our experiments demonstrate that the proposed method (Trans6D) achieves state-of-the-art performance in the LINEMOD and Occlusion LINEMOD datasets.

Furthermore, our method can be naturally extended to estimate the 6D object pose from the point cloud because the point cloud is sequence data and therefore suitable for Transformer. Despite the state-of-the-art performance, our method is not memory-friendly due to stacking a lot of self-attention modules. In future work, we plan to overcome the memory problem and extend Trans6D to more challenging scenes.

Acknowledgements

This work was supported by Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00537, Visual common sense through self-supervised learning for restoration of invisible parts in images). ZQZ was supported by China Scholarship Council (CSC) Grant No. 202208060266. AL was supported in part by the EPSRC (grant number EP/S032487/1).

References

1. Billings, G., Johnson-Roberson, M.: Silhonet: An rgb method for 6d object pose estimation. IEEE Robotics and Automation Letters **4**(4), 3727–3734 (2019). https://doi.org/10.1109/LRA.2019.2928776
2. Brachmann, E., Michel, F., Krull, A., Yang, M.Y., Gumhold, S., Rother, C.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3364–3372 (2016). https://doi.org/10.1109/CVPR.2016.366
3. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: ECCV. Springer (September 2014), `https://www.microsoft.com/en-us/research/publication/learning-6d-object-pose-estimation-using-3d-object-coordinates/`
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)

5. Chen, D., Li, J., Wang, Z., Xu, K.: Learning canonical shape space for category-level 6d object pose and size estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11970–11979 (2020). https://doi.org/10.1109/CVPR42600.2020.01199
6. Chen, W., Jia, X., Chang, H.J., Duan, J., Leonardis, A.: G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4232–4241 (2020). https://doi.org/10.1109/CVPR42600.2020.00429
7. Chen, W., Duan, J., Basevi, H., Chang, H.J., Leonardis, A.: PonitPoseNet: Point Pose Network for Robust 6D Object Pose Estimation. In: The IEEE Winter Conference on Applications of Computer Vision (WACV) (March 2020)
8. Chen, W., Jia, X., Chang, H.J., Duan, J., Shen, L., Leonardis, A.: Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism (2021)
9. Cheng, Y., Lu, F.: Gaze estimation using transformer. arXiv preprint arXiv:2105.14424 (2021)
10. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
14. He, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: Transformer-based object re-identification. arXiv preprint arXiv:2102.04378 (2021)
15. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11629–11638 (2020). https://doi.org/10.1109/CVPR42600.2020.01165
16. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. pp. 548–562. Springer (2012)
17. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) Computer Vision – ACCV 2012. pp. 548–562. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
18. Hodaň, T., Baráth, D., Matas, J.: Epos: Estimating 6d pose of objects with symmetries. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11700–11709 (2020). https://doi.org/10.1109/CVPR42600.2020.01172
19. Hu, Y., Fua, P., Wang, W., Salzmann, M.: Single-stage 6d object pose estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2927–2936 (2020). https://doi.org/10.1109/CVPR42600.2020.00300

20. Hu, Y., Fua, P., Wang, W., Salzmann, M.: Single-stage 6d object pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2930–2939 (2020)
21. Hudson, D.A., Zitnick, C.L.: Generative adversarial transformers (2021)
22. Jiang, Y., Chang, S., Wang, Z.: Transgan: Two transformers can make one strong gan. arXiv preprint arXiv:2102.07074 (2021)
23. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 1530–1538 (2017). https://doi.org/10.1109/ICCV.2017.169
24. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer (2021)
25. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: Deepim: Deep iterative matching for 6d pose estimation. In: European Conference on Computer Vision (ECCV) (2018)
26. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7677–7686 (2019). https://doi.org/10.1109/ICCV.2019.00777
27. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1954–1963 (2021)
28. Morrison, D., Tow, A.W., McTaggart, M., Smith, R., Kelly-Boxall, N., Wade-McCue, S., Erskine, J., Grinover, R., Gurman, A., Hunn, T., et al.: Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 7757–7764. IEEE (2018)
29. Oberweger, M., Rad, M., Lepetit, V.: Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 125–141. Springer International Publishing, Cham (2018)
30. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7667–7676 (2019). https://doi.org/10.1109/ICCV.2019.00776
31. Peng, S., Zhou, X., Liu, Y., Lin, H., Huang, Q., Bao, H.: Pvnet: Pixel-wise voting network for 6dof object pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3047388
32. Rad, M., Lepetit, V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 3848–3856 (2017). https://doi.org/10.1109/ICCV.2017.413
33. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
34. Shao, J., Jiang, Y., Wang, G., Li, Z., Ji, X.: Pfrl: Pose-free reinforcement learning for 6d pose estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11451–11460 (2020). https://doi.org/10.1109/CVPR42600.2020.01147
35. Song, C., Song, J., Huang, Q.: Hybridpose: 6d object pose estimation under hybrid representations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 428–437 (2020). https://doi.org/10.1109/CVPR42600.2020.00051

36. Trabelsi, A., Chaabane, M., Blanchard, N., Beveridge, R.: A pose proposal and refinement network for better 6d object pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2382–2391 (January 2021)
37. Wada, K., Sucar, E., James, S., Lenton, D., Davison, A.J.: Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14528–14537 (2020). https://doi.org/10.1109/CVPR42600.2020.01455
38. Wang, C., Martín-Martín, R., Xu, D., Lv, J., Lu, C., Fei-Fei, L., Savarese, S., Zhu, Y.: 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 10059–10066 (2020). https://doi.org/10.1109/ICRA40945.2020.9196679
39. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3338–3347 (2019). https://doi.org/10.1109/CVPR.2019.00346
40. Wang, G., Manhardt, F., Tombari, F., Ji, X.: Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16611–16621 (2021)
41. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes (2017)
42. Yang, S., Quan, Z., Nie, M., Yang, W.: Transpose: Towards explainable human pose estimation by transformer (2020)
43. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1941–1950 (2019). https://doi.org/10.1109/ICCV.2019.00203
44. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H.S., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers (2020)