PAPER

Attention Voting Network with Prior Distance Augmented Loss for 6DoF Pose Estimation*

Yong HE[†a)], ***Student Member***, **Ji LI**[†b)], **Xuanhong ZHOU**[†], **Zewei CHEN**[†], ***and*** **Xin LIU**[†], ***Nonmembers***

SUMMARY 6DoF pose estimation from a monocular RGB image is a challenging but fundamental task. The methods based on unit direction vector-field representation and Hough voting strategy achieved state-of-the-art performance. Nevertheless, they apply the smooth $\ell 1$ loss to learn the two elements of the unit vector separately, resulting in which is not taken into account that the prior distance between the pixel and the keypoint. While the positioning error is significantly affected by the prior distance. In this work, we propose a Prior Distance Augmented Loss (PDAL) to exploit the prior distance for more accurate vector-field representation. Furthermore, we propose a lightweight channel-level attention module for adaptive feature fusion. Embedding this Adaptive Fusion Attention Module (AFAM) into the U-Net, we build an Attention Voting Network to further improve the performance of our method. We conduct extensive experiments to demonstrate the effectiveness and performance improvement of our methods on the LINEMOD, OCCLUSION and YCB-Video datasets. Our experiments show that the proposed methods bring significant performance gains and outperform state-of-the-art RGB-based methods without any post-refinement.

key words: *6DoF pose estimation, semantic segmentation, keypoint localization, deep learning, attention mechanism*

1. Introduction

6DoF pose estimation is to detect the known objects and estimate the 6DoF pose, i.e., the 3D rotation and location, from an RGB(-D) image with a cluttered scene. It is a crucial and challenging task for a variety of applications including autonomous driving [1], [2], robotic manipulation [3], [4], augmented reality [5], etc. With the progress of deep learning based methods for 6DoF pose estimation, many RGB based methods [6]–[10] and RGB-D based methods [1], [11]–[14] have been proposed to overcome the limitations of traditional methods. Benefiting from the geometric information in the depth image, the latter methods can usually achieve higher accuracies than the former. However, these works rely heavily on depth images, which cannot be readily applied in certain scenes, including outdoor, underwater, etc. In contrast, 6DoF pose estimation from a single RGB image is a more challenging and extensive task.

In this work, we focus on estimating the 6DoF pose from a monocular RGB image without any post-refinement. A few recent deep learning-based methods [6], [15]–[17] build end-to-end Convolutional Neural Networks (CNNs) to directly estimate the object's initial 6DoF pose. However, some time-consuming post processes, such as Iterative Closest Point (ICP) [18], are implemented by those methods to refine the rough initial pose. Most recent state-of-the-art methods, such as [7]–[10], [19], adopt a two-stage pipeline based on dense prediction. One significant difference between these two-stage approaches is the intermediate representation. [9], [19], [20] locate keypoints by regressing relative offsets or heatmaps to establish 2D-3D correspondences. [8], [21]–[23] predict the 3D coordinates of each pixel to establish 3D-3D correspondences. These intermediate representation methods are limited by the complexity of the unlimited 2D or 3D continuous search space. In particular, in order to reduce the dimension of search space, Zakharov et al. [8] adopt texture mapping to construct a quantified 2D correspondence map. Because of the quantization, the pose estimator can only estimate a coarse initial pose. A post-refinement is exceedingly required to acquire advanced performance.

Unit direction vector-field representation and Hough voting scheme were proposed for robust 2D keypoint localization and demonstrated its superiority in [6], [7], [17]. This manner can address the common occlusion naturally. Furthermore, this representation limits the search space to the normalized 2D space, which can dramatically reduce the output space complexity without quantization. However, the Hough voting scheme is non-differentiable so that it cannot be integrated into the learning of vector-field representation for joint training, and [6], [7], [17] just use smooth $\ell 1$ loss [24] to learn it. This may cause the learned vector-field is not accurate enough for the Hough keypoint localization. More specifically, assuming that the unit vectors of different points have the same angular error, those points that are further away from the keypoint may generate greater locating deviation. But the distance is not taken into account during training. In this work, exploiting the context prior information of the distance between pixels and keypoints, we propose a Prior Distance Augmented Loss (PDAL) to enable distance to be taken into account during training. It augments the weight of those pixels that may generate larger positioning errors, thereby further improving the accuracy and robustness of keypoint localization.

A U-structure fully convolutional network is adopted by [6], [7], [14], [17] for 6DoF pose estimation. U-Net [25],

Manuscript received November 11, 2020.
Manuscript revised January 27, 2021.
Manuscript publicized March 26, 2021.

[†]The authors are with the School of Computer Science, Chongqing University, Chongqing, China.

*This work was supported by the Chongqing Technology Innovation and Application Development Special Key Project under Grant cstc2019jscx-fxydX0054.

a) E-mail: heyong18@cqu.edu.cn
b) E-mail: leedge@cqu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2020EDP7235

[26] is widely used in medical image segmentation. Due to the stable structure of human organs and concise semantic information in the medical images, it has highly efficient performance on a small dataset. However, its representation ability is limited for estimating the object's pose on a large dataset with cluttered scenes. Some of its variants [27]–[29] have achieved higher accuracy. Nevertheless, they suffer from higher model complexity and heavier computational burden. Attention mechanism has attracted a series of researches on computer vision in recent years, which shows its great potential. Inspired by [29]–[33], we design a lightweight channel-level attention module (Adaptive Fusion Attention Module, AFAM) for adaptive feature fusion, so as to maintain the advantages of the long connection while enhancing the model's feature representation ability. xEmbedding the AFAM into the PVNet [7] (a standard U-Net), we build an Attention Voting Network to further improve the performance of our method.

In summary, we first propose the PDAL to learn more accurate vector field representation for 2D keypoint localization. Then, we propose an Attention Voting Network, which further improves the performance of our method. To evaluate our approaches, we conduct experiments on three widely-used benchmarks for 6DoF pose estimation: LINEMOD [34], OCCLUSION [35] and YCB-Video [6] datasets. The experiments demonstrate that proposed methods significantly improve the performance of PVNet and outperform state-of-the-art RGB-based methods without any post-refinement.

2. Related Work

In this section, we introduce the related work of the 6DoF pose estimation and attention mechanism. The methods of 6DoF pose estimation can be divided into holistic methods and correspondence based methods.

Holistic methods. Holistic methods estimate the rotation and location parameters of known objects from a given image in a single shot. Traditionally, template-based approaches [36], [37] render synthetic image patches by comprehensive poses to construct a template database and then compute the best-matched template pose. The accuracies of these approaches depend on the completeness of the template database and they are not robust to occlusion and lighting condition variations. Some early deep learning based methods [6], [15], [38] are proposed to directly regress the 6DoF pose. Nevertheless, due to the lack of depth information and nonlinear rotation space, directly estimating the pose does not work well. To solve this problem, Kehl et al. [16] utilize depth information for post-refinement. The state-of-the-art methods based on dense prediction, including [1], [13], [14], etc., regard the depth image as a point cloud to extract geometric information and then combine it with the appearance features learned by CNNs to predict dense poses. While the above methods rely on depth images and require post-refinement to obtain state-of-the-art performance.

Correspondence-based methods. Detecting keypoint is easier than regressing the rotation and location directly, therefore correspondence-based methods design a two-stage pipeline to solve the 6DoF pose estimation problem: they first estimate the 2D-3D or 3D-3D correspondences by keypoint detection and then solve the 6DoF pose estimation through PnP [39] or ICP algorithm [18]. Traditional approaches [4], [34], [40] can establish the correspondence by detecting keypoints for textured objects. However, these methods are not robust to lighting variation and background clutters, rely on handcrafted features, and cannot resolve texture-less objects. Recent methods take advantage of CNNs for detecting keypoints to deal with these problems. Tekin et al. [19] employ the YOLO framework [41] to directly regress the object's keypoints by a sparse output grid. When disturbed by occlusion, it does not work well.

As mentioned in Sect. 1, the most recent state-of-the-art methods exploit the dense prediction method to solve occlusion. In detail, [8], [10], [12], [21] establish 3D-3D correspondences by predicting the 3D coordinates of pixels relative to the camera coordinate system. To reduce the complexity of the 3D output space, Wang et al. [12] normalize the object model coordinates. But it requires additional scale parameters. Zakharov et al. [8] exploit a quantized texture mapping and refine the pose through a deep refinement module. Li et al. [10] use an additional object detector, e.g., [41], [42], to extract object image patches and scales them to a fixed size for the keypoint detector. The real-time performance of this method depends on the object detector. In contrast, 2D keypoint detection is easier than 3D from an RGB image. Some methods, e.g., [1], [9], [19], are based on the 2D-3D correspondences. They define the corners of the 3D model's bounding box as keypoints and locate their mapping position in the 2D image. However, the virtual corner cannot be explicitly displayed on the image and its mapping location is far from the object's pixels, which will lead to a large localization deviation. [7], [22] demonstrate that selecting the surface point of the 3D model as the keypoint can acquire more accurate poses. Furthermore, Peng et al. [7] achieve state-of-the-art performance via their unit direction vector-field representation and Hough voting scheme.

Attention mechanisms. In recent years, the incorporation of attention mechanism into CNN has attracted a lot of researches, showing great potential for improving performance. There are two prevalent ways to implement the attention mechanism: channel attention [30], [32], [43]–[45] and point-wise spatial attention [46]–[49]. Channel attention is usually designed as a lightweight block and embedded into the standard segmentation architectures to implement the communication between channels. Spatial attention is a non-local operation so that each pixel can fully capture global information. In contrast, the channel attention mechanism is relatively lightweight, so that it can be flexibly integrated into various CNNs to enhance representation capabilities. Instead of directly concatenating the low-level features with the high-level features by skip connection, Li et al. [31] and Ni et al. [33] design a channel attention mod-

ule for more effective feature fusion. Their attention modules first learn an attention vector to recalibrate the low-level feature and then integrate it into the high-level feature.

3. Method

Given an RGB image, the goal of the 6DoF pose estimation is to detect objects and estimate their rigid transformation that transforms the object from the object coordinate system to the camera coordinate system. The transformation can be formulated as a rigid transformation matrix $[\mathbf{R}, \mathbf{t}] \in SE(3)$, a general representation, where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ represent the 3D rotation and location respectively.

To tackle this task, we construct a two-stage pipeline that first establishes 2D-3D correspondences and then estimates the 6DoF pose by a PnP algorithm [39]. As visualized Fig. 1, the Attention Voting Network predicts pixel-wise semantic labels and vector-field representation, then the Hough voting layer [7] uses the intermediate representation to locate the 2D keypoint as described in 3.1. Figure 1 illustrates that learning the vector-field representation and positioning the keypoint are separated as mentioned above. We propose the PDAL to solve this problem in 3.2 and introduce the AFAM to further improve the performance of our method in 3.3. The implementation details of the Attention Voting Network will be shown in the next section.

3.1 Vector-Field Representation for Keypoint Localization

The purpose of the first stage is to locate the 2D projection points of predefined 3D keypoints associated with the 3D object models, where the keypoint localization is implemented through the Hough voting scheme [7] based on the semantic mask and vector-field representation. More specifically, we represent the 2D keypoint set as $\boldsymbol{K} = \{\boldsymbol{k}_i | i = 1, 2, 3, \ldots, K\}$, where K is set to 8 in all experiments. Given a keypoint $\boldsymbol{k}_i$ of object $\boldsymbol{O}$ and the predicted vector-field (i.e., pixel-wise direction vectors) for $\boldsymbol{k}_i$, we generate a candidate point set $\boldsymbol{C}_i = \left\{\boldsymbol{c}_{i,j} | j = 1, 2, 3, \ldots, C\right\}$ for the keypoint, where a candidate point can be the intersection of any two direction vectors. Then all the direction vectors vote for the candidate points if the deviation angle relative to the direction from the pixel to the candidate point less than a certain threshold. The candidate point with the most votes as the keypoint hypothesis, i.e., the positioning result of $\boldsymbol{k}_i$. Before training, we need to predefine 3D keypoints for each 3D object model and compute the ground truth vector-field for each image.

Keypoint selection. Some typical methods [1], [15], [19] select the eight corners of the object 3D bounding box as the keypoints. However, these virtual points are far away from the object, which may bring larger localization errors. Instead, we follow [7], [14] to define the 3D point set for each object by the Farthest Point Sampling (FPS) algorithm [50]. More specifically, FPS initializes the point set by the object center. Then, it repeatedly searches for the farthest point from the current set, and adds the farthest point to the set until K points are obtained. Naturally, the selected 3D points are spread out on the surface of the object, which makes the model more robust to occlusion. For an object $\boldsymbol{O}$,

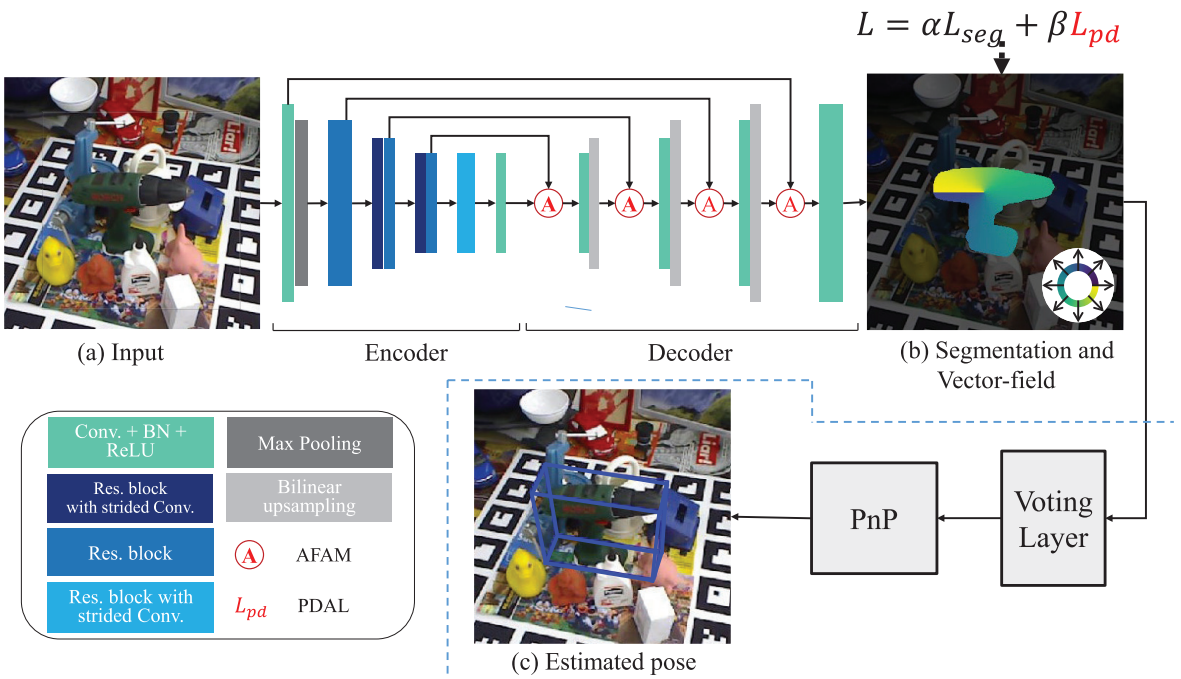

Fig. 1 **Overview of the attention voting network**: (a) an input image from the LINEMOD. (b) The segmentation and vector-field representation predicted by the network, where the dark areas are the background, the colored areas represent the object and the unit direction vector-field. (c) The 3D bounding box corresponds to the predicted pose.

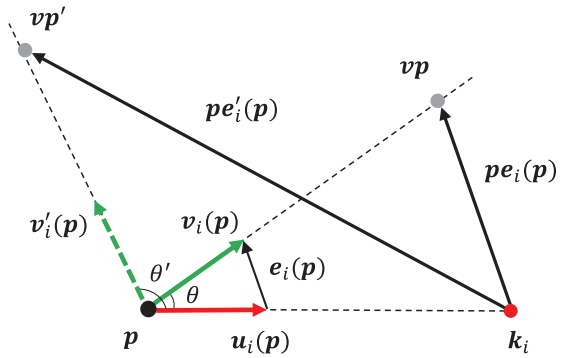

Fig. 2 **Visualization of prior distance augmented error.** $\boldsymbol{p}$ and $\boldsymbol{k}_i$ are a pixel and a keypoint respectively, $\boldsymbol{vp}$ is the hypothesis point of $\boldsymbol{k}_i$, $\boldsymbol{u}_i(\boldsymbol{p})$ is the ground truth unit vector, $\boldsymbol{v}_i(\boldsymbol{p})$ represents the predicted vector, $\boldsymbol{v}_i'(\boldsymbol{p})$ indicates a prediction with greater error, $\boldsymbol{e}_i(\boldsymbol{p})$ represents original error and $\boldsymbol{pe}_i(\boldsymbol{p})$ is prior distance augmented error.

its 3D keypoints are mapped onto the 2D image according to the ground truth pose to obtain 2D keypoints.

Vector-field representation. Vector-field representation consists of pixel-wise unit direction vectors for each 2D keypoint [6], [7], [17]. As visualized in Fig. 1 (b), different colors overlaid on the object indicate the specific directions from pixels to a 2D keypoint as shown in the indicator plate. For a pixel $\boldsymbol{p}$ and a 2D keypoint $\boldsymbol{k}_i$ of an object $\boldsymbol{O}$, the unit direction vector is defined as

$$\boldsymbol{u}_i(\boldsymbol{p}) = \frac{(\boldsymbol{k}_i - \boldsymbol{p})}{\|\boldsymbol{k}_i - \boldsymbol{p}\|_2} \tag{1}$$

and we denote the predicted vector corresponding to $\boldsymbol{u}_i(\boldsymbol{p})$ as $\boldsymbol{v}_i(\boldsymbol{p})$. Its error is expressed by $\boldsymbol{e}_i(\boldsymbol{p}) = \boldsymbol{v}_i(\boldsymbol{p}) - \boldsymbol{u}_i(\boldsymbol{p})$, as shown in Fig. 2. Peng et al. [7], Capellen et al. [17] and Xiang et al. [6] apply smooth $\ell 1$ loss [24] to learn the vector-field representation:

$$L_v = \frac{1}{KO} \sum_{\boldsymbol{k}_i \in \boldsymbol{K}} \sum_{\boldsymbol{p} \in \boldsymbol{O}} \ell 1(\boldsymbol{e}_i(\boldsymbol{p})|_x) + \ell 1(\boldsymbol{e}_i(\boldsymbol{p})|_y), \tag{2}$$

where O represents the number of the pixels belonging to object $\boldsymbol{O}$, $\boldsymbol{u}_i(\boldsymbol{p})$ represents the ground truth unit vector, $\boldsymbol{v}_i(\boldsymbol{p})$ is the predicted vector, $\boldsymbol{e}_i(\boldsymbol{p})|_x$ and $\boldsymbol{e}_i(\boldsymbol{p})|_y$ represent the two elements of $\boldsymbol{e}_i(\boldsymbol{p})$.

3.2 Prior Distance Augmented Loss

Due to the candidate points are the intersection of two random direction vectors, small direction errors can produce huge positioning errors if the pixels are far from the keypoint. Equation (2) indicates that learning vector-field representation does not take the distance $d(\boldsymbol{k}_i, \boldsymbol{p})$ into account, where $d(\boldsymbol{k}_i, \boldsymbol{p}) = \|\boldsymbol{k}_i - \boldsymbol{p}\|_2$ denotes the Euclidean distance between $\boldsymbol{k}_i$ and $\boldsymbol{p}$. Because the positioning result is unknown during training and the keypoint localization scheme is non-differentiable, the positioning error cannot be directly computed for learning vector-field representation.

To avoid this problem, we expand the vector $\boldsymbol{v}_i(\boldsymbol{p})$ by a factor of $d(\boldsymbol{k}_i, \boldsymbol{p})$ to generate a virtual point $\boldsymbol{vp}$ so that $d(\boldsymbol{vp}, \boldsymbol{p}) = d(\boldsymbol{k}_i, \boldsymbol{p})\boldsymbol{v}_i(\boldsymbol{p})$ as shown in Fig. 2. $\boldsymbol{vp}$ can be regarded as the hypothesis (or proxy) of keypoint $\boldsymbol{k}_i$ for pixel $\boldsymbol{p}$ during training. In this way, the final positioning error can be approximated by the proxy error, i.e., the error between $\boldsymbol{vp}$ and $\boldsymbol{k}_i$, and it can be described by

$$\boldsymbol{pe}_i(\boldsymbol{p}) = d(\boldsymbol{k}_i, \boldsymbol{p})(\boldsymbol{v}_i(\boldsymbol{p}) - \boldsymbol{u}_i(\boldsymbol{p})), \tag{3}$$

where $\boldsymbol{pe}_i(\boldsymbol{p})$ also can be interpreted as the prior distance augmented error of $\boldsymbol{v}_i(\boldsymbol{p})$, prior information $d(\boldsymbol{k}_i, \boldsymbol{p})$ can be easily computed before training. For a certain pixel, the prior distance augmented error $\boldsymbol{pe}_i(\boldsymbol{p})$ is positively related to the deviation angle θ between the predicted and ground truth vector, even if the deviation exceeds the right angle as the prediction $\boldsymbol{v}_i'(\boldsymbol{p})$ in Fig. 2. This is a crucial reason why the proposed loss function can be used as a stand-alone loss for vector-field representation. For different pixels, the error they generate will be augmented according to the distance from the keypoint.

Exploiting the prior distance augmented error, we can augment L_v loss by directly replacing $\boldsymbol{e}_i(\boldsymbol{p})$ with $\boldsymbol{pe}_i(\boldsymbol{p})$. So the L_v with prior distance (PD-L_v) can be defined as

$$\text{PD-}L_v = \frac{1}{KO} \sum_{\boldsymbol{k}_i \in \boldsymbol{K}} \sum_{\boldsymbol{p} \in \boldsymbol{O}} \ell 1(\boldsymbol{pe}_i(\boldsymbol{p})|_x) + \ell 1(\boldsymbol{pe}_i(\boldsymbol{p})|_y). \tag{4}$$

However, we note that the two elements of the direction vector are regarded as independent outputs by Eq. (3). This does not correspond to the definition of the unit vector well. Rather than calculate the error of the two elements separately, we use the distance between $\boldsymbol{vp}$ and $\boldsymbol{k}_i$ as the proxy positioning error. The ablation studies also show that it is a better implementation of PDAL. The final PDAL is defined as

$$L_{pd} = \frac{1}{KO} \sum_{\boldsymbol{k}_i \in \boldsymbol{K}} \sum_{\boldsymbol{p} \in \boldsymbol{O}} d(\boldsymbol{k}_i, \boldsymbol{p}) \|\boldsymbol{v}_i(\boldsymbol{p}) - \boldsymbol{u}_i(\boldsymbol{p})\|_2 . \tag{5}$$

Essentially, according to the prior distance information, PDAL augments the weight of those pixels that are further away from keypoints. In other words, it enables those pixels that may generate larger positioning errors to be allocated greater weight. PDAL is numerically close to those methods that directly predict the coordinate offset of the keypoints, but it still forces the output to fit the unit vector, which retains the advantages of unit vector representation as mentioned in Sect. 1.

3.3 Adaptive Fusion Attention Module

U-Net uses the shallow feature map with shape features, a crucial component to segmentation, to supplement object edge details for deep semantic features by long connection in the encoder. The long connection is an extension of the residual connection which can further reduce the vanishing gradient and accelerate the training convergence. Nevertheless, low-level features contain plenty of disturbing background information and each channel of different level feature maps embeds specific features for segmentation and

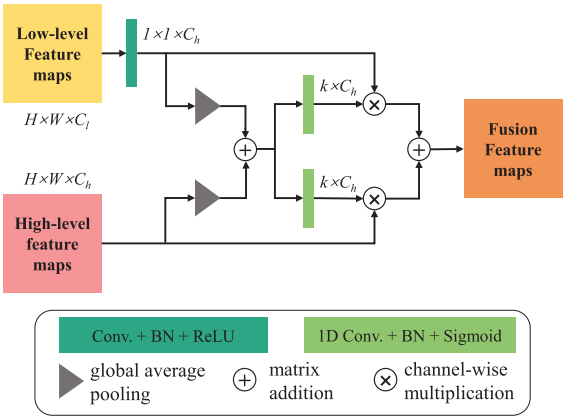

Fig. 3 **The architecture of the adaptive fusion attention module.** The inputs of the attention module are the low-level and high-level feature maps. The output is the fusion feature maps.

pose estimation. Roughly fusing multi-level feature maps (e.g., channel concatenation) may interfere with the model to mine effective information. We propose the AFAM for effective multi-level feature fusion to improve the model's representation ability and embed it into PVNet to build our Attention Voting Network.

The Global Attention Upsample (GAU) and Augmented Attention Module (AAM) proposed by Li et al. [31] and Ni et al. [33] respectively only learn attention for low-level features. However, each channel of high-level feature maps contains semantic features with different importance and also requires to be calibrated. Our AFAM simultaneously recalibrates multiple levels of feature maps and adaptively fuses them. Moreover, GAU [31] does not fully utilize multi-level global features (only high-level) to learn the attention tensor. We follow AAM [33] to fully capture cross-level and cross-channel interaction.

As illustrated in Fig. 3, we integrate two levels of context information obtained through global average pooling to achieve cross-level interaction. Before performing the global average pooling, the channel dimension of the low-level feature map needs to be increased to C_h by a 1×1 convolution layer. Then attention tensors can be learned directly through a fully connected layer with sigmoid activation. But this will significantly increase the model complexity. A low-dimensional intermediate layer can be used to reduce parameters as in [30]. But this dimensionality reduction destroys the direct correspondence between the channels and attention weights. Therefore, we follow [32] and apply 1D convolution with a kernel size of k to achieve local cross-channel interaction and obtain attention vectors, which guarantees both effectiveness and efficiency. Finally, we achieve multi-level feature fusion by adding the recalibrated feature maps.

The whole calculation process of our attention module can be formulated as

$$G(\boldsymbol{x},\boldsymbol{y}) = Avg(\delta(\boldsymbol{x})) + Avg(\boldsymbol{y}), \tag{6}$$

$$F(\boldsymbol{x},\boldsymbol{y}) = \sigma_x(G(\boldsymbol{x},\boldsymbol{y})) \circ \delta(\boldsymbol{x}) + \sigma_y(G(\boldsymbol{x},\boldsymbol{y})) \circ \boldsymbol{y}, \tag{7}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the low-level and high-level features respectively, $Avg()$ denotes the global average pooling, $\delta()$ means to perform 1×1 convolution, batch normalization (BN) and ReLU activation, in turn, $\sigma()$ is similar to $\delta()$ but the activation function and the 2D convolution is replaced by Sigmoid and 1D convolution respectively, $\circ$ is channel-wise multiplication, G refers to the fused global feature, F represents the final fusion feature maps.

Our attention module only contains a 2D convolution with a kernel size of 1×1 and two efficient 1D convolutions. It introduces negligible parameters and computational burden while bringing significant performance gains. Moreover, the output channel dimension C_h of our attention module is less than the dimension $C_l + C_h$ generated by concatenation, resulting in lower parameter complexity of our attention voting network than PVNet.

4. Experiments

In this section, we discuss the implementation details of our pipeline, datasets, metrics and evaluation results.

4.1 Implementation Details

We will introduce the implementation details of our pipeline in three aspects: network architecture, training, and inference.

Network architecture. As shown in Fig. 1, we build the Attention Voting Network by replacing the channel concatenation of U-Net with the proposed attention module. Assuming the dataset contains N categories of objects. And we predefine K keypoints for each object. Taking a $H\times W\times3$ image as input, the network outputs a $H\times W\times((N+1)+(N\times K\times2))$ tensor, where $(N+1)$ channels represent the semantic labels including object and background labels, the remaining $(N \times K \times 2)$ channels represent unit vector-field. Specifically, the encoder is a pre-trained ResNet-18 [51] backbone without the terminal pooling layer, and its fully connected layers are replaced by a convolution layer. When the size of the feature map is reduced to $H/8\times W/8$, The residual block with dilated convolution is applied to expand the receptive fields while maintaining the size of the feature maps. To learn pixel-wise prediction, the low-resolution feature maps be repeatedly performed feature fusion, convolution, and bilinear upsampling until the size is restored to $H \times W$, where the feature fusion can be implemented via channel connection as in [7], [25] or our attention module. Then a 3×3 convolution and a 1×1 convolution are used to acquire the semantic labels and the direction vector-field representation. Finally, voting layer is used to estimate the 2D-3D correspondences (described in Sect. 3.1) and the pose parameters computed by the PnP algorithm.

Training. We exploit the proposed PDAL for learning vector-field representation. For supervising the segmentation, the multi-class cross-entropy is adopted. Thus loss function for this multi-task network is defined as

$$L = \alpha L_{seg} + \beta L_{pd}, \tag{8}$$

where L_{seg} represents segmentation loss, L_{pd} is PDAL, α and β are the balance factor to balance the two tasks, L_{pd} is a large value relative to L_{seg} so that the values of α and β are of vital importance. We dynamically adjusted their values to keep the two types of losses at the same order of magnitude. Adam optimizer is employed to train the network and set the learning rate as 1e-3 and decay it to 1e-5 by a factor of 0.75 every 10 epochs. The number of images in each sequence is not enough to train the deep network, thus we use the code provided by [7] to render 10,000 synthetic images for each sequence.

Inference. During testing, an image is entered into the trained voting network to predict the segmentation label and vector-field, then the predictions are used to locate keypoints by the voting layer as elaborated in Sect. 3.1. Finally, the 6DoF pose parameters are estimated by a PnP algorithm [39] that estimates the external parameters of the camera given 2D-3D correspondences and its intrinsic parameters. For multiple instances scenes, we generate voting centers through clustering and assign masks to the nearest voting center as [7], [14].

4.2 Datasets

We evaluate our methods on three standard benchmarks: the LINEMOD, OCCLUSION and YCB-Video datasets.

LINEMOD Dataset [34], a standard benchmark for object detection and 6DoF pose estimation, consists of 13 image sequences with cluttered scenes and occlusion, each containing CAD model and around 1200 images with instance mask and 6DoF pose for a low-textured object. We follow previous works [7] to divide the training and testing set.

OCCLUSION Dataset [35] was generated via additionally annotating masks and 6DoF poses for each object of a subset of the LINEMOD. This dataset depicts 8 different objects in 1214 images. Because of significant occlusion between objects, it is a more challenging dataset. And it is barely used for testing in this work.

YCB-Video Dataset [6], a large benchmark, contains 92 videos, each of which shows a subset of 21 YCB objects in various indoor scenes and annotates them with 6DoF poses and masks. The varying lighting conditions, image noise and occlusions bring great challenges.

4.3 Metrics

In our experiments, we use 2D projection [22] and average 3D distance ADD(-S) [6], [22] metric to evaluate our methods.

2D Projection metric [22] projects the 3D model points onto the image by the predicted and ground truth pose separately, and then computes the average 2D projection distance. Normally, it is considered as correct if the average 2D projection distance of a pose does not exceed 5 pixels.

ADD(-S) metric [34] measures the average pairwise distance of the 3D model points transformed by estimated poses and ground truth for non-symmetric objects, i.e., ADD [34] metric. For symmetric objects, the ADD metric is replaced by ADD-S metric [6], i.e., computes closest point distance. A pose is considered correct if the mean 3D transform distance is less than a certain threshold. For the LINEMOD and OCCLUSION datasets, the threshold is set to 10% of the model's diameter. For YCB-Video dataset, we follow [6], [7] and compute the area under the accuracy-threshold curve, i.e., the ADD(-S) AUC metric.

4.4 Evaluation

In this section, we explore the effectiveness of our PDAL and AFAM, discuss the influence of the segmentation accuracy on the 6DoF pose estimation, and compare the proposed methods with state-of-the-art methods. Our work focus on estimating accurate initial pose from a monocular RGB image, so we merely compare our methods with those state-of-the-art methods that are based on RGB images and do not perform any post-refinement. Besides, we keep other experimental settings, e.g., K and C (both described in 3.1), consistent with PVNet. Naturally, PVNet is a baseline for our methods.

Ablation study. We conduct comparative experiments on different vector-field representation loss (VFR-Loss) functions, attention modules and capacities of the network. As shown in Table 1, both PD-L_v and PDAL bring considerable accuracy improvements, which illustrates that the prior distance information is critical and the final PDAL is a better implementation. We re-implement GAU [31] and AAM [33] to compare with our AFAM. Comparing PVNet+GAU and

Table 1 The ablation studies on the LINEMOD dataset in terms of the ADD(-S) metric. These results show the influence of vector-field representation loss (VFR-Loss), attention module and capacity of the network.

Model	Backbone	VFR-Loss	Average
PVNet	Res-18	L_v	86.27
PVNet	Res-18	PD-L_v	89.04
PVNet	Res-18	PDAL	90.17
PVNet	Res-34	L_v	86.93
PVNet+GAU	Res-18	L_v	87.66
PVNet+AAM	Res-18	L_v	88.05
PVNet+AFAM	Res-18	L_v	89.19
PVNet+AFAM	Res-18	PDAL	91.91

Table 2 The example results of the influences of segmentation accuracy (AP, AR, IoU) on 6-DoF pose estimation on the LINEMOD dataset. The original results are bold.

cat						
	PVNet	PDAL+AFAM				
AP	**95.59**	29.99	67.05	83.25	94.63	**96.96**
AR	**95.49**	87.15	68.53	42.10	77.29	**97.17**
IoU	**91.50**	28.71	51.25	39.11	74.05	**93.00**
ADD(-S)	**78.94**	20.96	48.60	79.84	86.24	**87.72**
driller						
	PVNet	PDAL+AFAM				
AP	**96.63**	45.96	67.29	72.08	91.62	**96.73**
AR	**97.14**	87.92	68.95	73.70	49.99	**97.60**
IoU	**93.98**	43.24	51.64	57.33	47.79	**94.23**
ADD(-S)	**96.63**	57.58	77.70	82.66	96.72	**99.01**

Table 3 The accuracies of our method and the state-of-the-art methods on the LINEMOD dataset in terms of the ADD(-S) metric. The name of symmetric object are bold.

Methods	BBS [15]	Pix2pose [23]	DPOD [8]	CDPN [10]	PVNet [7]	Ours		
						PDAL	AFAM	PDAL+AFAM
ape	27.9	58.1	53.28	64.38	43.62	63.14	57.04	**69.43**
benchwise	62.0	91.0	95.34	97.77	99.90	99.90	99.52	**100.00**
cam	40.1	60.9	90.36	91.67	86.86	90.88	88.04	**92.45**
can	48.1	84.4	94.10	95.87	95.47	97.93	98.13	**99.21**
cat	45.2	65.0	60.38	83.83	79.34	84.83	85.33	**87.72**
driller	58.6	73.6	97.72	96.23	96.43	98.12	97.52	**99.01**
duck	32.8	43.8	66.01	66.76	52.58	63.75	62.53	**67.79**
eggbox	40.0	96.8	99.72	99.72	99.15	99.91	99.53	**100.00**
glue	27.0	79.4	93.83	**99.61**	95.66	97.39	96.33	98.94
holepuncher	42.4	74.8	65.83	85.82	81.92	82.87	83.92	**86.01**
iron	67.0	83.4	99.80	97.85	98.88	**99.90**	99.08	99.38
lamp	39.9	82.0	88.11	97.89	99.33	99.71	98.94	**99.81**
phone	35.2	45.0	74.24	90.75	92.41	93.94	93.56	**95.10**
Average	43.6	72.4	82.98	89.86	86.27	90.17	89.19	**91.91**

Table 4 The average accuracies of our methods and the state-of-the-art methods on the LINEMOD and OCCLUSION dataset in terms of the 2D projection metric. '-' indicates that it is not provided in the original paper.

Methods	BBS [15]	CDPN [10]	Oberweger [20]	PVNet [7]	Ours		
					PDAL	AFAM	PDAL+AFAM
LINEOMD	83.9	98.10	-	99.00	99.35	99.23	**99.43**
OCCLUSION	17.1	-	60.9	61.06	62.15	62.88	**63.67**

Table 5 The accuracies of our methods and the state-of-the-art methods on the OCCLUSION dataset in terms of the ADD(-S) metric. The name of symmetric object are bold. '-' indicates that it was not provided in the original paper.

Methods	PoseCNN [6]	Pix2pose [23]	DPOD [8]	PVNet [7]	Ours		
					PDAL	AFAM	PDAL+AFAM
ape	9.6	22.0	-	15.81	23.50	22.73	**25.47**
can	45.2	44.7	-	63.30	**68.93**	67.85	68.20
cat	0.93	**22.7**	-	16.68	21.48	18.46	22.26
duck	19.6	15.0	-	25.24	28.50	26.11	**32.61**
driller	41.4	44.7	-	65.65	66.37	67.89	**68.33**
eggbox	22.0	25.2	-	**50.17**	40.60	50.04	45.28
glue	38.5	32.4	-	49.62	47.73	**50.39**	49.28
holepuncher	22.1	**49.5**	-	39.67	44.19	45.75	47.51
Average	24.9	32.0	32.79	40.77	42.66	43.65	**44.87**

PVNet+AAM with PVNet+AFAM in Table 1 shows that the proposed AFAM results in more obvious performance gains. Moreover, the parameters of PVNet with Res-34 backbone are nearly doubled, but its improvement is very limited, which shows that simply increasing the capacity of the network does not bring significant improvement and the performance gain is indeed due to the layout of AFAM.

We also evaluate the influence of the accuracy of semantic segmentation on 6DoF pose estimation. We randomly change the predicted semantic labels of the foreground and background pixels according to a certain probability to obtain different segmentation results with different accuracy. This will not affect the accuracy of the vector-field representation. We show the evaluation results of three semantic segmentation metrics, i.e., Average Precision (AP), Average Recall (AR) and Intersection over Union (IoU), on the cat and driller objects respectively to evaluate the influence of the segmentation accuracy. As shown in Table 2, the accuracy of 6Dof pose estimation is positively correlated with AP but has no obvious correlation with AR and IoU. This is because only the direction vectors belonging to the foreground pixels are valid (the VFR-Loss only penalizes the errors of the ground truth foreground pixels), and the voting layer just uses a part of the foreground pixels for pose estimation during inference. Moreover, the original segmentation accuracy of the proposed method is slightly higher than that of the bassline, but this cannot support the pose estimation to obtain large performance gains, which shows that the proposed method does improve the quality of the vector-field representation.

Evaluation on the LINEMOD dataset. Table 3 summarizes our experimental results for all 13 sequences of the LINEMOD dataset w.r.t. the ADD(-S) metric. Comparing our methods with PVNet [7], both PDAL and AFAM have significant improvements on most objects, especially on ape, cat, duck, etc., which illustrates their effectiveness. Table 3 also shows the comparison results between our methods with BB8 [15], Pix2pose [23], DPOD [8],

Table 6 The average accuracies of our methods and the state-of-the-art methods on the YCB-Video dataset in terms of the 2D projection metric and ADD(-S) AUC. '-' indicates that it is not provided in the original paper.

Methods	PoseCNN [6]	Oberweger [20]	PVNet [7]	Ours		
				PDAL	AFAM	PDAL+AFAM
2D Projection	-	39.4	47.4	51.42	50.65	**53.27**
ADD(-S) AUC	61.30	72.8	73.4	75.05	74.81	**76.12**

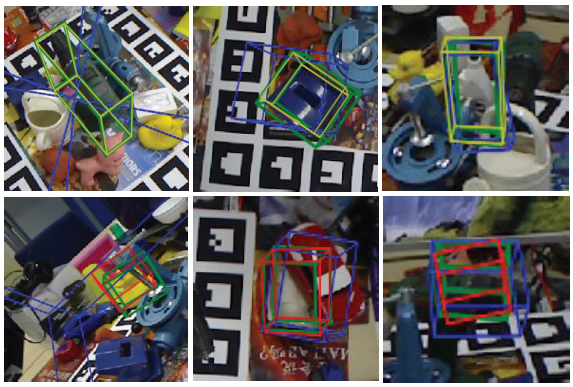

Fig. 4 **Visual comparison examples on the LINEMOD and OCCLUSION datasets.** Green 3D bounding boxes represent the ground truth poses, blue 3D bounding boxes correspond to the poses predicted by the baseline (PVNet [7]), yellow and red 3D bounding boxes respectively represent the predicted poses of PDAL and AFAM.

CDPN [10] and the PVNet [7]. It shows that our methods outperform most methods with a large margin. Noted that CDPN is based on an additional object detector [41], [42]. Nevertheless, the performances of our PDAL and PDAL+AFAM still exceed it. As shown in Table 4, our methods have certain improvements compared to other advanced methods on the 2D Projection metric.

Evaluation on the OCCLUSION dataset. The OCCLUSION dataset is used to evaluate the robustness to occlusion. In Table 5, we show the evaluation results on the OCCLUSION dataset w.r.t. the ADD(-S) metric. Because the accuracy on eggbox (a symmetric object) is severely reduced, the gain of average accuracy on the OCCLUSION dataset brought by PDAL is not as significant as on the LINEMOD dataset. PDAL may not have enough stability for the symmetric object with severe occlusion, but AFAM can get more balanced performance improvements. PDAL+AFAM further improves the accuracy of pose estimation for most objects to show its advantages in dealing with occlusion. Although their performances do not have obvious advantages on symmetric objects (eggbox and glue), they have significant superiority on most non-symmetric objects.

Comparing with other state-of-the-art methods [8], [15], [23], as reported in Table 5, our methods significantly outperform them on most objects, even on symmetric objects. The final result surpasses them with a large margin of at least 12.08%. Table 4 compares the results of our methods and PVNet in terms of the 2D projection metric on the OCCLUSION dataset, where the quality of our methods outperform PVNet in keypoint positioning.

Visual comparison examples on the LINEMOD and OCCLUSION datasets. We also show the visual comparison examples of the poses predicted by the baseline and the proposed methods (PDAL and AFAM). As shown in Fig. 4, many predefined keypoints of the object are occluded, and most of the visible pixels of the object are far away from these keypoints. The accuracy of the direction vectors predicted by the baseline on these pixels is insufficient, resulting in a pose that is completely out of the image range. PDAL uses the prior distance and AFAM learns more effective features through the attention mechanism to generate more accurate direction vectors, which corrects these erroneous poses. In other examples, the proposed methods can further improve the accuracy of 6Dof pose estimation.

Evaluation on the YCB-Video dataset. The proposed methods may benefit from some special settings of the LINEMOD and OCCLUSION datasets, such as markers around the objects. We evaluate our methods on the YCB-Video dataset to further verify that they are not limited to the LINEMOD and OCCLUSION datasets. Table 6 illustrates that the proposed methods still have considerable improvement and surpass PoseCNN [6] and Oberweger [20], which further demonstrates the effectiveness of our methods.

5. Conclusion

In this work, we proposed the PDAL to learn more accurate vector-field representation for 2D keypoint localization and the AFAM for the voting network to improve its representation ability. Extensive experiments demonstrated that the proposed method significantly improved the performance of PVNet for 6DoF pose estimation on common evaluation metrics. Compared with other state-of-the-art RGB-only methods, our method outperforms them on the LINEMOD dataset. We also showed the advantages of our methods to deal with occluded and texture-less objects, which obtained the best accuracy on the OCCLUSION and YCB-Video datasets. In the case of symmetrical objects with severe occlusion, our methods did not obtain significant performance gains or even negative gains, which requires further research and improvement.

References

[1] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.244–253, 2018.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," 2012 IEEE Conference

on Computer Vision and Pattern Recognition, pp.3354–3361, IEEE, 2012.
[3] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," arXiv preprint arXiv:1809.10790, 2018.
[4] M. Zhu, K.G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," 2014 IEEE International Conference on Robotics and Automation (ICRA), pp.3936–3943, IEEE, 2014.
[5] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," IEEE Trans. Vis. Comput. Graphics, vol.22, no.12, pp.2633–2651, 2015.
[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," arXiv preprint arXiv:1711.00199, 2017.
[7] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4561–4570, 2019.
[8] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," Proc. IEEE International Conference on Computer Vision, pp.1941–1950, 2019.
[9] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6d object pose estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3385–3394, 2019.
[10] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," Proc. IEEE International Conference on Computer Vision, pp.7678–7687, 2019.
[11] C. Li, J. Bai, and G.D. Hager, "A unified framework for multi-view multi-class object pose estimation," Proc. European Conference on Computer Vision (ECCV), pp.254–269, 2018.
[12] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L.J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.2642–2651, 2019.
[13] 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," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3343–3352, 2019.
[14] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11632–11641, 2020.
[15] 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," Proc. IEEE International Conference on Computer Vision, pp.3828–3836, 2017.
[16] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," Proc. IEEE International Conference on Computer Vision, pp.1521–1529, 2017.
[17] C. Capellen, M. Schwarz, and S. Behnke, "Convposecnn: Dense convolutional 6d object pose estimation," arXiv preprint arXiv:1912.07333, 2019.
[18] P. Besl and H. McKay, "A method for registration of 3-d shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol.14, pp.239–256, 03 1992.
[19] B. Tekin, S.N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.292–301, 2018.
[20] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," Proc. European Conference on Computer Vision (ECCV), pp.119–134, 2018.
[21] O.H. Jafari, S.K. Mustikovela, K. Pertsch, E. Brachmann, and C. Rother, "ipose: instance-aware 6d pose estimation of partly occluded objects," Asian Conference on Computer Vision, pp.477–492, Springer, 2018.
[22] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," Proc. IEEE conference on computer vision and pattern recognition, pp.3364–3372, 2016.
[23] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," Proc. IEEE International Conference on Computer Vision, pp.7668–7677, 2019.
[24] R. Girshick, "Fast r-cnn," Proc. IEEE international conference on computer vision, pp.1440–1448, 2015.
[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," International Conference on Medical image computing and computer-assisted intervention, pp.234–241, Springer, 2015.
[26] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O.R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," Pattern Recognition, vol.106, p.107404, 2020.
[27] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp.3–11, Springer, 2018.
[28] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu, "Ce-net: Context encoder network for 2d medical image segmentation," IEEE Trans. Med. Imag., vol.38, no.10, pp.2281–2292, 2019.
[29] O. Oktay, J. Schlemper, L.L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N.Y. Hammerla, B. Kainz, et al., "Attention u-net: Learning where to look for the pancreas," arXiv preprint arXiv:1804.03999, 2018.
[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," Proc. IEEE conference on computer vision and pattern recognition, pp.7132–7141, 2018.
[31] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," arXiv preprint arXiv:1805.10180, 2018.
[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11534–11542, 2020.
[33] Z.-L. Ni, G.-B. Bian, X.-H. Zhou, Z.-G. Hou, X.-L. Xie, C. Wang, Y.-J. Zhou, R.-Q. Li, and Z. Li, "Raunet: Residual attention u-net for semantic segmentation of cataract surgical instruments," International Conference on Neural Information Processing, pp.139–149, Springer, 2019.
[34] 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," Asian conference on computer vision, pp.548–562, Springer, 2012.
[35] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," European conference on computer vision, pp.536–551, Springer, 2014.
[36] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," European Conference on Computer Vision, pp.408–421, Springer, 2010.
[37] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.5, pp.876–888, 2011.
[38] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," Proc. IEEE international conference on computer vision, pp.2938–2946, 2015.
[39] 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, p.155, 2009.
[40] M. Aubry, D. Maturana, A.A. Efros, B.C. Russell, and J. Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," Proc. IEEE conference on computer vision

and pattern recognition, pp.3762–3769, 2014.
[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE conference on computer vision and pattern recognition, pp.779–788, 2016.
[42] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," Proc. IEEE international conference on computer vision, pp.2961–2969, 2017.
[43] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," Proc. IEEE conference on Computer Vision and Pattern Recognition, pp.7151–7160, 2018.
[44] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," Proc. European conference on computer vision (ECCV), pp.325–341, 2018.
[45] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," Proc. IEEE conference on computer vision and pattern recognition, pp.1857–1866, 2018.
[46] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A.L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," Proc. IEEE conference on computer vision and pattern recognition, pp.3640–3649, 2016.
[47] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," Proc. European Conference on Computer Vision (ECCV), pp.267–283, 2018.
[48] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.12416–12425, 2020.
[49] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," Proc. IEEE International Conference on Computer Vision, pp.9167–9176, 2019.
[50] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in neural information processing systems, pp.5099–5108, 2017.
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

Yong He received the B.E. degree in Intelligent Science and Technology from Chongqing University of Posts and Telecommunications in 2018. He is currently a postgraduate student at the School of Computer Science of Chongqing University. His research interests include 6DoF pose estimation, object detection and semantic segmentation.

Ji Li received the B.E., M.E., and Ph.D. degree in Computer Science from Chongqing University in 1995, 2001 and 2005, respectively. He is currently an associate professor at Chongqing University. His research interests include cloud computing and computer version.

Xuanhong Zhou received the B.E. degree in Software Engineering from Xihua University in 2018. He is currently studying for a master degree in Computer Science at Chongqing University. His research interests include object detection and semantic segmentation.

Zewei Chen received the B.E. degree in Electronic and Information Engineering from Huaqiao University in 2019. He is currently studying for a master degree in Computer Science at Chongqing University. His research interests include edge computing and image processing.

Xin Liu received the B.E. degree in Computer Science from Yangtze University in 2019. She is currently studying for a master degree in Computer Science at Chongqing University. Her research interests include edge computing and image processing.