electronics

MDPI

Article

A Robust CoS-PVNet Pose Estimation Network in Complex Scenarios

Jiu Yong [1,*], Xiaomei Lei [2], Jianwu Dang [1] and Yangping Wang [1]

[1] The School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China; dangjw@mail.lzjtu.cn (J.D.); 13519311970@163.com (Y.W.)
[2] College of Intelligence and Computing, Tianjin University, Tianjin 300072, China; leixiaomei@tju.edu.cn
* Correspondence: yongjiu@mail.lzjtu.cn

Abstract: Object 6D pose estimation, as a key technology in applications such as augmented reality (AR), virtual reality (VR), robotics, and autonomous driving, requires the prediction of the 3D position and 3D pose of objects robustly from complex scene images. However, complex environmental factors such as occlusion, noise, weak texture, and lighting changes may affect the accuracy and robustness of object 6D pose estimation. We propose a robust CoS-PVNet (complex scenarios pixel-wise voting network) pose estimation network for complex scenes. By adding a pixel-weight layer based on the PVNet network, more accurate pixel point vectors are selected, and dilated convolution and adaptive weighting strategies are used to capture local and global contextual information of the input feature map. At the same time, the perspective-n-point localization algorithm is used to accurately locate 2D key points to solve the pose of 6D objects, and then, the transformation relationship matrix of 6D pose projection is solved. The research results indicate that on the LineMod and Occlusion LineMod datasets, CoS-PVNet has high accuracy and can achieve stable and robust 6D pose estimation even in complex scenes.

Keywords: 6D pose estimation; complex scenes; CoS-PVNet; weight self-learning; global attention mechanism

check for updates

Citation: Yong, J.; Lei, X.; Dang, J.; Wang, Y. A Robust CoS-PVNet Pose Estimation Network in Complex Scenarios. *Electronics* **2024**, *13*, 2089. https://doi.org/10.3390/electronics13112089

Academic Editors: Agnieszka Pregowska and Klaudia Proniewska

Received: 30 April 2024
Revised: 23 May 2024
Accepted: 27 May 2024
Published: 27 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Object 6D pose estimation, as an important task in the field of computer vision, has many applications in fields such as augmented reality (AR), virtual reality (VR), robotics, and autonomous driving. As shown in Figure 1, by estimating the 6D pose of an object in the camera coordinate system, namely the 3D position and 3D pose, virtual and real objects can be combined in the real environment to enhance people's perception of the real world [1]. In addition, in industrial manufacturing, robots can perform precise part positioning and assembly operations through pose estimation. In autonomous driving navigation, cars need to understand their location in the environment in order to plan the optimal path. However, due to the influence of complex conditions such as background clutter and target occlusion in the real environment [2], the 6D object pose estimation is inaccurate, and the robustness is poor. Therefore, accurately and robustly estimating the 6D pose of the target object from complex scenes is crucial for improving the performance of AR, VR, robotics, and autonomous driving [3].

The 6D pose estimation of target objects aims to detect targets and estimate their direction and translation relative to the standard framework [4]. The main challenge of traditional 6D pose estimation is to establish a correspondence between the input image and available 3D models and then use the perspective-n-point (PnP) to calculate pose parameters. However, the quality of the correspondence is sensitive to factors such as lighting changes, weak textures, and cluttered backgrounds [5], making it difficult for traditional methods to handle textureless objects and exhibiting poor robustness to severe occlusion and background changes [6]. In recent years, deep learning-based methods

have shown strong capabilities in handling 6D pose estimation, which can generally be divided into two categories: end-to-end methods based on direct regression and two-stage methods based on object class priors. In end-to-end methods, training a neural network and directly regressing the 6D pose from the input image using the neural network is not as accurate as traditional geometry-based PnP algorithms, although this type of method is highly efficient [7]. In two-stage methods, CNN is first used to regress the intermediate representation, establishing 2D and 3D correspondence, and then the PnP algorithm is executed based on this correspondence. However, this type of method usually uses regression and multiple representations to estimate the pose, requiring accurate acquisition of key point information of the target object. It can be seen that the existing mainstream 6D object pose estimation methods model this problem as a regression task, requiring special designs to deal with multiple solution problems when dealing with symmetric and partially visible objects.

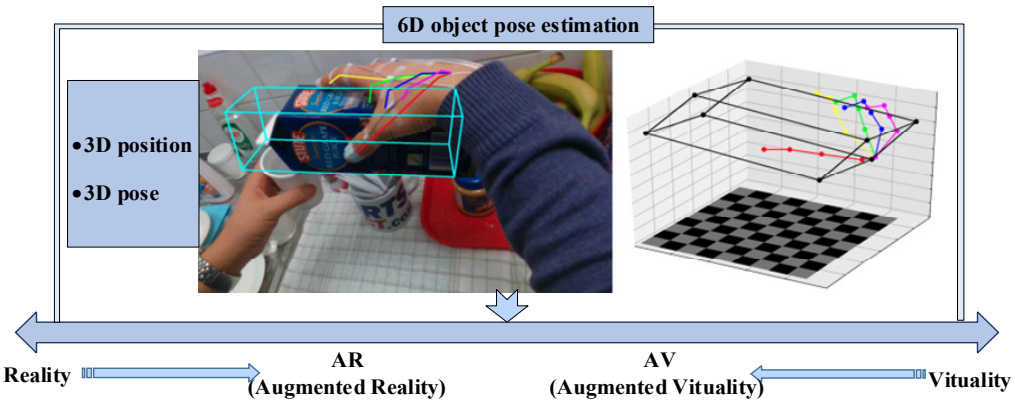


Figure 1. Relationship between real world and virtual information in an AR system.

We propose a deep learning-based CoS-PVNet (complex scenarios pixel-wise voting network) for 6D object pose estimation in complex scenes, which achieves accurate and robust 6D pose estimation of the target object. CoS-PVNet provides support for achieving stable and robust 6D pose estimation in virtual real fusion interactive applications. The main work of this article is as follows: (1) In complex scenes such as cluttered environments and severe occlusion, a CoS-PVNet object pose estimation network framework is proposed, which can enhance the key point feature processing ability of RGB images, accurately filter and predict pixel vectors, and effectively improve the accuracy and robustness of 6D pose estimation in complex scenes. (2) Inaccurate vector field prediction will affect the quality of generating key point assumptions. By adding a pixel-weight self-learning module between the encoder and decoder of the PVNet to predict pixel confidence, it can adapt to more complex image features and changes through learnability, prevent key feature information loss, and make semantic segmentation results more accurate. (3) To improve the quality of key point feature extraction in complex scenes, a pixel-weight layer is added to PVNet to filter out more accurate pixel vectors, and a global attention mechanism is proposed to enhance the extraction of useful key point features while adding contextual information to enhance the performance of CoS-PVNet in extracting weak texture scene features.

2. Related Work

A deep learning network model is used to calculate the 6D pose [*R*,*T*] of the target object from the image given an RGB/RGB-D image containing the target object and a 3D model of the target object [8], as shown in Formula (1).

$$[R, T] = F\{[I, Model|\theta]\} \quad (1)$$

where F is the deep learning model, I is the input image, *Model* is the 3D model of the object, and θ is the model parameter.

Deep learning-based 6D object pose estimation typically uses object detection networks or semantic segmentation networks as feature extraction networks to annotate target regions from images and encode pose semantic features [9]. However, unlike pixel-level classification based on semantic segmentation, object detection has a faster inference speed and is more in line with the real-time requirements of AR, VR, robotics, and autonomous driving. Therefore, early 6D pose estimation often used object detection networks as feature extraction networks [10]. Algorithms such as SSD-6D [11], YOLO-6D [12], and CDPN [13] first calculate the 2D bounding box of the object based on SSD [11], YOLO V2 [14], and Faster R-CNN [15] target detection network, respectively, and then send the 2D bounding box area into the pose calculation branch to estimate the 6D pose of the target object. The inference speeds of YOLO-6D and CDPN are 20 ms and 33 ms, which are much higher than the pose estimation models based on semantic segmentation during the same period. However, due to the fact that the 2D bounding boxes output by the object detection network contain some background or occlusion areas, the features input to the pose calculation module inevitably contain interference features, thereby reducing the accuracy of 6D pose estimation and the robustness of model occlusion. Semantic segmentation is a pixel-level object detection method that accurately segments objects along their contours, eliminating occlusion and irrelevant background regions. Therefore, it is more suitable as a feature extraction network for complex scenes [16]. For example, the average accuracy of PoseCNN in occluded scenes in 2017 [17] was 24.9%, much higher than the 6.42% of YOLO-6D [12] in 2018, but the frame rate of PoseCNN was 10 FPS, only 1/5 of YOLO-6D. The semantic segmentation architecture of PoseCNN is similar to FCN [18], with an encoder of VGG [19], gradually encoding semantic features of different dimensions. However, the output high-resolution semantic features severely lack detailed information, and transposed convolution is inefficient. U-Net [20] is a classic semantic segmentation network that adopts a symmetric encoding and decoding structure and fuses detailed features with deep semantic features through skip connections to improve the network's understanding of images. Inspired by the U-Net network, the PVNet [21] uses residual blocks and bilinear interpolation to reconstruct a lightweight U-Net as a feature extraction network with an inference speed of 40 ms. However, inaccurate vector field predictions in complex scenes will affect the quality of generated key point assumptions, and PVNet can lead to difficult and insufficient feature extraction for target objects in complex scenes, affecting the accuracy and robustness of 6D pose estimation.

In summary, semantic segmentation-based pose estimation methods are more suitable for 6D object pose estimation in complex scenes, and pose estimation based on different architecture feature extraction networks belongs to multitask learning [22], which not only annotates the target object from the input image but also calculates the 6D pose of target object [23]. Therefore, appropriate multitask, self-learning weights help to explore the correlation between object detection tasks, semantic segmentation tasks, and pose estimation tasks and extract sufficient semantic features to distinguish target objects from occluded objects to reduce the impact of occluded areas to improve the capability of semantic feature expression and accurate estimation of 6D pose, and thus improve the performance of 6D object pose estimation network.

3. CoS-PVNet Pose Estimation Network

3.1. Overall Framework Structure of CoS-PVNet

In response to the difficulty in accurately estimating the 6D pose of objects in complex scenes [24], this paper proposes a two-stage CoS-PVNet pose estimation network based on a single RGB image for PVNet pose estimation. By integrating key point localization into a deep learning architecture, CNN is used to establish the correspondence between the 2D and 3D of the target object, accurately locate 2D key points, and then use the global attention mechanism and voting mechanism to execute the PnP algorithm to solve the 6D pose information of objects, accurately estimating the 6D pose of the target object without any pose refinement.

The overall framework structure of CoS-PVNet is shown in Figure 2. Given a single RGB image containing the target object, a weight self-learning module is added between the skip connections of PVNet, and three tasks are performed: constructing semantic labels for predicting pixel directions, constructing unit vectors, and predicting pixel weights. Then, a new global attention mechanism (GAM) is proposed to enhance the extraction of useful features and increase contextual information. Furthermore, the ASPP-DF-PVNet algorithm [25] is used to optimize RANSAC voting for locating 2D key points, filtering out biased votes, and further optimizing the voting results to obtain more accurate 2D key points. Finally, the PnP algorithm is used to solve the 6D pose of the target, and a homogeneous coordinate transformation matrix composed of translation and rotation transformations of the target object coordinate system relative to the camera input coordinates is solved, achieving the transformation of the CoS-PVNet coordinate system.

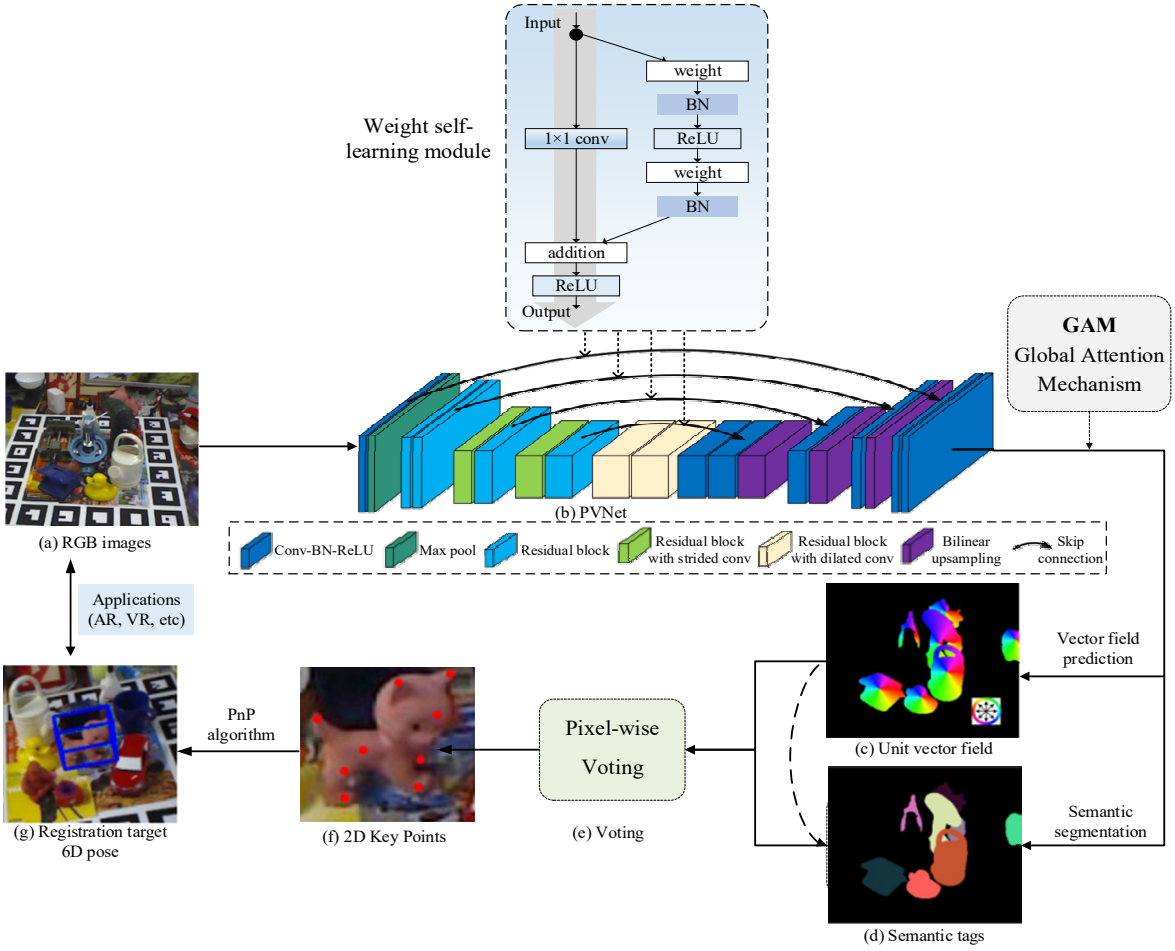


Figure 2. Overall structure of CoS-PVNet.

3.2. CoS-PVNet Weight Self-Learning Module Structure

The weight self-learning module structure consists of a series of residual units. As shown in Figure 2, the overall backbone of the network is a pretrained ResNet-18 [26], followed by a weight self-learning module and several convolutional and upsampling layers, described as a weight self-learning structure. In the network structure, a weight self-learning module is added to the skip connections, and through the weight self-learning module, larger weight information is added to prevent the loss of key information, thereby making the semantic segmentation results more accurate.

The weight self-learning module has added conv5-conv10x to the network structure of ResNet-18. Take the image of $H \times W \times 3$ as input for downsampling until the feature map reaches $H/8 \times W/8$, and then replace the convolution in the last two blocks of ResNet-18

with rate = 2 and rate = 4. Subsequently, the feature maps output by the encoder are input into the weight self-learning module to extract dense features. Finally, connect the result feature maps from all branches and feed them to another 1×1 convolution to obtain the desired spatial dimension. In the weight self-learning module, the number of output channels is set to 256. After obtaining the feature map processed by the weight self-learning module, upsampling is performed until the size reaches $H \times W$. Assuming there are C object classes and each object has K key points, 1×1 convolution is applied on the feature map to output tensors for vector field representations of $H \times W \times (K \times 2 \times C)$ key points and $H \times W \times (C + 1)$ semantic labels.

The semantic labels of the segmented image and the predicted unit vector $v_k(p)$ of each pixel for K key points are outputted with the same size by inputting an RGB image. $v_k(p)$ represents the direction of each pixel voted pointing to a key point X_k, and $v_k(p)$ is calculated as the distance difference between the current pixel P and the K-th key point divided by the binomial of the distance difference between the two:

$$v_k(p) = \frac{x_k - p}{\|x_k - p\|_2} \tag{2}$$

The pixel-weight outputs by CoS-PVNet represent the confidence score obtained by each pixel, which is used to filter out outliers and internal pixels for voting before calculating the two-dimensional position of the key points. I_e represents pixel weight, which estimates the cosine value between the predicted vector and the target vector $v_k(p)$, the I_e is:

$$I_{\mathrm{e}} = \cos(\hat{v}_k(p), v_k(p)) \tag{3}$$

The larger the pixel-weight value, the closer the predicted vector is to the true value. In the process of calculating the key points later, the pixels to be voted on are selected based on the predicted pixel-weight values from the previous ones to ensure the accuracy of pose estimation. The total loss function is:

$$L = \lambda_{\mathrm{vec}} L_{vec} + \lambda_{sem} L_{sem} + \lambda_e L_e \tag{4}$$

where L_{vec} is the vector field prediction loss function, L_{sem} is the semantic segmentation loss function, and L_e is the weight prediction loss function. λ_{vec}, λ_{sem}, λ_e represents the corresponding coefficient. The loss function for vector field prediction is defined as follows:

$$L_{\mathrm{vec}} = \frac{1}{m} \sum_{k=1}^{9} \sum_{p \in O} l_1(\Delta v_k(p)|_x + l_1(\Delta v_k(p)|_y)) \tag{5}$$

$$\Delta v_k(p) = \hat{v}_k(p) - v_k(p) \tag{6}$$

3.3. CoS-PVNet Global Attention Mechanism

To cope with complex scene feature extraction or lack of features and scenes without features, a global attention mechanism is proposed in the CoS-PVNet algorithm to enhance the extraction of useful features and increase contextual information for more effective extraction of input feature maps. As shown in Figure 3, this mechanism adopts dilated convolution and adaptive weighting strategies to capture local and global contextual information of the input feature map.

Firstly, a dilated convolutional layer is applied to the input feature map X to capture the local contextual information of the input feature map. The dilated convolutional layer outputs a feature map D, which contains spatial information of the original feature map and contextual information captured through dilated convolution. Next, a global average pooling is performed on feature map D to extract global contextual information. Global average pooling transforms feature map D into a feature vector G that represents global information. In order to achieve an adaptive attention mechanism, the global information feature vector G is utilized and passed to a shared fully connected layer MLP. The fully

connected layer MLP outputs a weight matrix W with the same dimension as the input feature map X. Subsequently, the weight matrix W is used to perform weighted fusion on the input feature map X. The weighted fusion operation can be expressed as $A = W \otimes X$, where $\otimes$ represents element-wise multiplication. In this way, a weighted feature map A containing adaptive attention information is obtained.

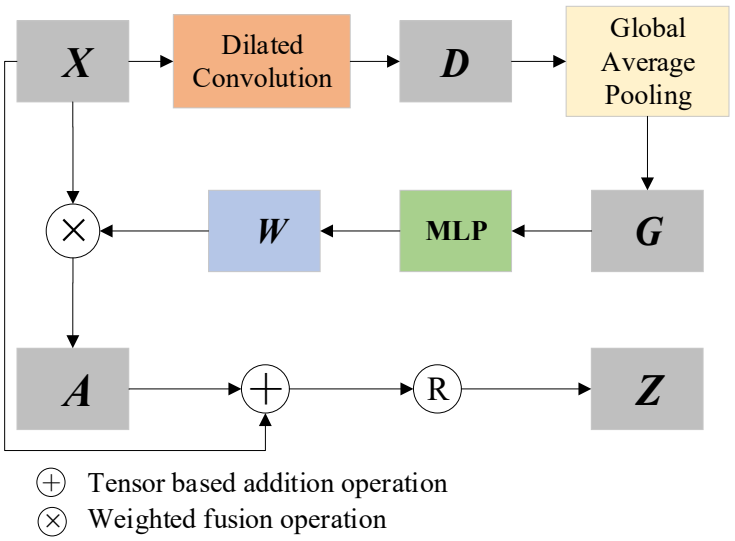


Figure 3. Global attention mechanism structure of CoS-PVNet.

Then, an element-wise addition operation is performed on the weighted feature map A and X to obtain the added feature map. Finally, the final feature map Z is generated through the ReLU activation function. For the given input X, GAM is expressed as:

$$D = C_d(X) \tag{7}$$

$$G = GAP(D) \tag{8}$$

$$W = MLP(G) \tag{9}$$

$$A = X \cdot W \tag{10}$$

$$Z = \mathrm{ReLU}(A + X) \tag{11}$$

where D is the feature map obtained by dilated convolution, G is the feature vector generated by average pooling, W is the output weight matrix of the fully connected layer, and Z is the feature map generated by the ReLU activation function after addition. Therefore, by integrating local and global contextual information, this adaptive attention mechanism can more effectively extract information from input feature maps. The adaptive weighting strategy enables the model to automatically adjust attention weights based on input feature maps, thereby improving the performance of CoS-PVNet. In addition, since PVNet uses cosine similarity between two vectors to determine voting, the method is more reliable when the key point assumption is consistent with more predicted directions [25]. However, when pixels are far from the key point assumption, the small angle between two directional vectors may cause significant voting bias. When the two assumptions are close, it will lead to inaccurate voting. Therefore, this paper uses the ASPP-DF-PVNet algorithm to optimize RANSAC voting for locating 2D key points, in order to obtain more accurate 2D key points and provide support for subsequent accurate target object pose estimation.

3.4. CoS-PVNet Target Object Pose Estimation

After determining the key 2D positions of the target object, CoS-PVNet achieves pose estimation through the PnP algorithm. By calculating the mean μ_k and covariance matrix $\sum_k (k = 1, \cdots, K)$ of the estimated target object and using the minimum Mahalanobis distance, the 6D pose (R, t) is calculated:

$$\underset{R,T}{\text{minimize}} \sum_{k=1}^{K} (\overline{X}_k - \mu_k)^T \sum_k^{-1} (\overline{X}_k - \mu_k) \tag{12}$$

$$\overline{X}_k = \pi(RX_k + t) \tag{13}$$

where X_k represents the 3D coordinates of the key points, $\overline{X}_k$ represents the 2D mapping of 3D coordinates X_k, and π is the perspective mapping function. The rotation and translation parameters R and t are initialized using the EPnP (efficient perspective-n-point) algorithm. Due to the uncertainty of the features, the Levenberg–Marquardt (nonlinear least squares algorithm) is used to minimize the remapping error and solve Formula (12). Therefore, based on the voting results, PnP can accurately locate and utilize 2D key points, allowing distance filtering voting schemes to improve the performance of pose estimation further. In addition, in the subsequent experiments of this article, to explore the impact of the number of key points on pose estimation, different numbers of key points are used to compare the results, and K = 8 is taken into account for efficiency and accuracy.

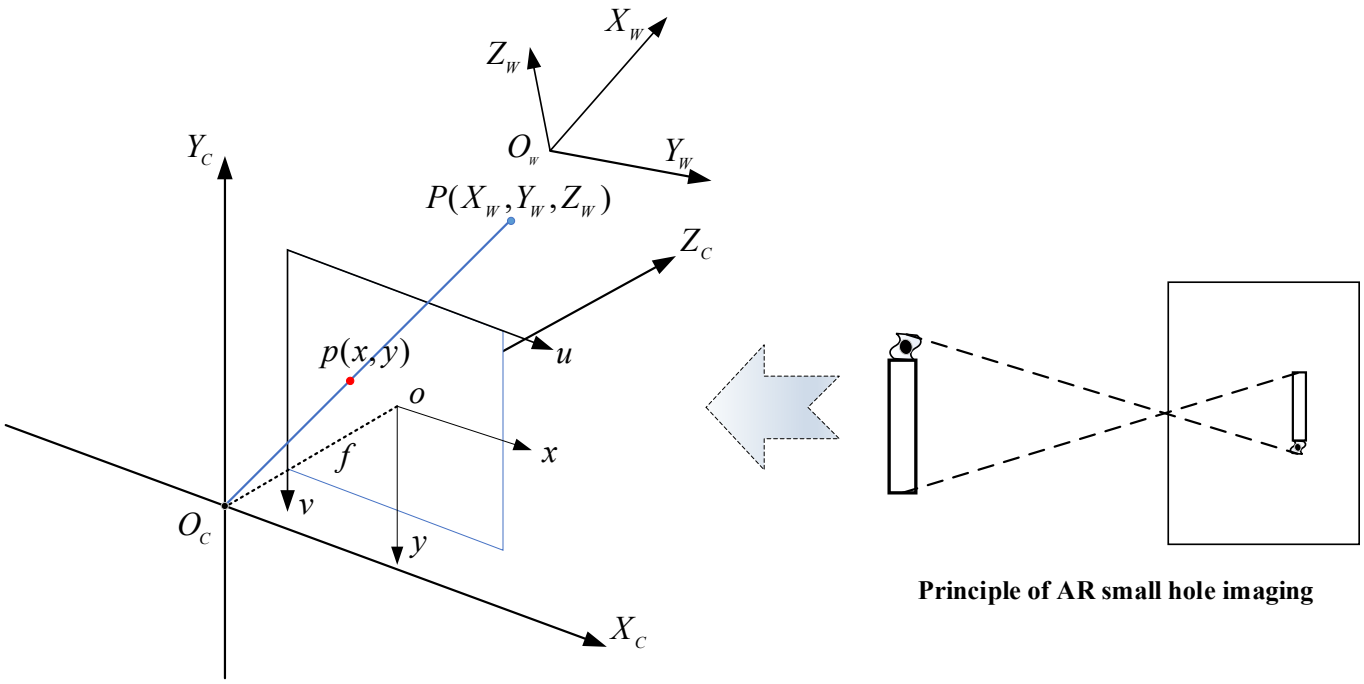
3.5. CoS-PVNet Coordinate System Conversion Relationship

The 6D pose estimation refers to estimating the 3D position and 3D pose of an object in the camera coordinate system. At this time, the coordinate system of the original object itself can be regarded as the world coordinate system, that is, obtaining a homogeneous transformation matrix composed of translation and rotation transformations from the world coordinate system of the original object to the camera coordinate system. As shown in Figure 4, CoS-PVNet registration mainly establishes the transformation relationship between the world coordinate system, camera coordinate system, image coordinate system, and pixel coordinate system.

Transforming 3D points from O_W (world coordinate system) to O_C (camera coordinate system) involves R, t (camera extrinsic). Transforming from a 3D point in the O_C (camera coordinate system) to a 2D point in the $O - xy$ (image coordinate system) involves K (camera reference). Rotation and translation transformations will occur during the camera shooting process. According to the principle of small-hole imaging, the target image is inverted, and the transformation is represented by a homography matrix. The mapping relationship between the homography matrix and the rotation translation matrix R, t is used to calculate and solve various parameters. By inferring the relationship between coordinate systems, it can be inferred that the relationship between point (u, v) in the pixel coordinate system of CoS-PVNet and point (X_w, Y_w, Z_w) in the world coordinate system is:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{14}$$

where $\begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ represents the internal reference matrix of the camera, which can be obtained by calibrating the camera. When the camera is actually shooting, the camera pose can be solved according to the above Formula (14). Therefore, after completing the spatial projection coordinate transformation based on CoS-PVNet, the camera pose parameters can be solved, achieving stable and robust system applications in AR, VR, robotics, and autonomous driving.



Principle of AR small hole imaging

The following four coordinate systems are involved in image processing:

$O_W - X_W Y_W Z_W$: World coordinate system, describing camera position

$O_C - X_C Y_C Z_C$: Camera coordinate system with optical center as origin

$o - xy$: Image coordinate system, with optical center as the midpoint of the image

uv : Pixel coordinate system, with the origin in the upper left corner of the image

P : A point in the world coordinate system is a real point in life

p : Point p is an imaging point in the image, with coordinates (x, y) in the image coordinate system and (u, v) in the pixel coordinate system

f : Camera focal length, equal to the distance between o and O_C , $f = \|o - O_C\|$

Figure 4. CoS-PVNet coordinate system relationship transformation.

3.6. CoS-PVNet Pose Estimation and Application Process

The pose estimation and application process of CoS-PVNet are shown in Figure 5. By extracting feature information from the input RGB image and using a weight self-learning module, the model can automatically adjust and optimize weights during the training process, improving the flexibility and adaptability of the model. Then, the CoS-PVNet predicts the key point position of each target object on the feature map and combines the global attention mechanism to enhance the extraction of useful features and increase contextual information in order to extract information more effectively from the input feature map. Subsequently, CoS-PVNet generates a voting vector for each detected key point, using the voting results to estimate the 6D pose of the target object. Finally, the CoS-PVNet coordinate system transformation relationship is solved to further realize applications of AR, VR, robotics, and autonomous driving.

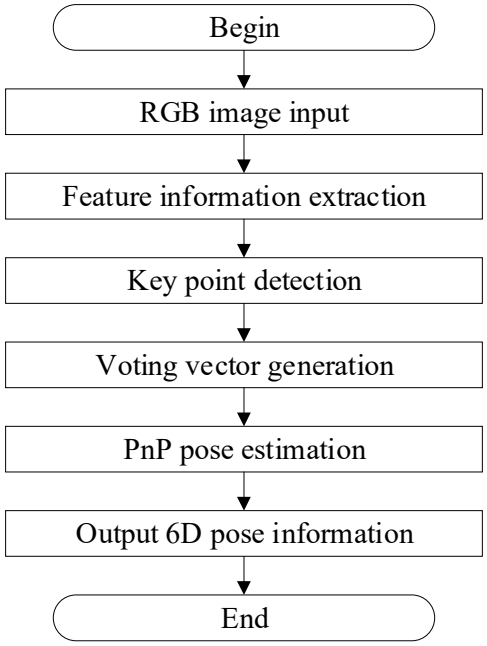


Figure 5. CoS-PVNet pose estimation and application process.

The specific steps for CoS-PVNet pose estimation and application are:

Step 1. Use the camera to input an RGB image containing the target object.
Step 2. Feed the input image into a pre-trained ResNet18 convolutional neural network to accurately extract feature information such as the shape, texture, and color of objects in the input image. For different RGB image data, the CoS-PVNet weight self-learning module can balance the focus of the model by adjusting the weights of different categories.
Step 3. During the training process, CoS-PVNet updates its weights through a weight self-learning module and backpropagation algorithm to minimize the value of the loss function and generate accurate key point feature maps.
Step 4. CoS-PVNet predicts the key point positions of each target object on the feature map, usually the corners, centers, or other prominent feature points of the target object.
Step 5. A set of key points is defined on the 3D model of an object with fixed coordinates (X, Y, Z) in 3D space. When the object is placed in a certain posture in the real world and captured as an image, these 3D key points will be projected onto the image plane to form 2D key points, which involve the internal parameters (such as focal length, principal point, etc.) and external parameters (such as rotation matrix and translation vector) of the phase machine. These parameters can map points in 3D space to the 2D image plane.
Step 6. Before predicting the feature map, the global attention mechanism is used to enhance the extraction of useful features and increase contextual information, which is used to extract input feature map information more effectively and better correspond to the 2D−3D relationship of the target object.
Step 7. CoS-PVNet generates a voting vector for each detected key point, uses the Gaussian kernel function to balance the importance of different votes, aggregates all voting vectors in the image space, and can form a voting density map or voting cloud, which reflects the 3D position and 3D pose of the target object in the image.
Step 8. In CoS-PVNet, PnP is used to calculate the 3D position of an object from the centroid position of the vote, and the relative position relationship between key points is used to estimate the rotation of an object.
Step 9. CoS-PVNet can estimate the pose parameter matrix of the camera, including rotation matrix, translation vector, or quaternion, based on a set of known 3D points and their projections in the image and apply it to AR, VR, robotics, and autonomous driving.

4. Experimental Results and Analysis

4.1. Experimental Environment and Dataset

4.1.1. Experimental Environment Configuration

CoS-PVNet provides accurate initial pose estimation based on RGB images, aiming to accurately locate and estimate the 3D direction and 3D translation relationship of objects. This article conducts experimental analysis on PVNet, CoS-PVNet, and the latest 6D pose estimation algorithm and uses ablation experiments to analyze the performance of each module of CoS-PVNet. The configuration of the experimental environment in this article is shown in Table 1.

Table 1. Experimental environment configuration.

Experimental Platform	Configuration
Operating System	Ubuntu 18.04
Graphics Card	Intel(R) Xeon(R) Platinum 8358P@2.60 GHz (15 core)
Memory/hard disk capacity	48 G/2 T
Deep learning frameworks	Python3.8, Pytorch1.6.0

4.1.2. Dataset and Model Training

This article conducts experiments on two benchmark datasets, LineMod [27] and Occupation LineMod [28], which are widely used in 6D pose estimation experiments to evaluate the performance of CoS-PVNet. The LineMod dataset exhibits significant clutter, diversity, multiview, and true pose annotation but only slight occlusion. The Occlusion LineMod dataset introduces interference of different occlusion levels based on LineMod, which is characterized by the complex relationship between the target object and the background. This provides more information for the performance evaluation of the 6D object pose estimation network.

(1) LineMod is a benchmark dataset used for 6D object pose estimation, as shown in Figure 6, consisting of 15 objects, each consisting of over 1200 images, with a total of 15,783 images. It not only annotates the central object in each RGB image but also provides the inherent characteristics of 3D CAD models and cameras for each object. The complex factors of LineMod include background clutter, textureless objects, and lighting changes.

(2) The Occupation LineMod dataset, as a subset of LineMod, contains 1214 images of 8 objects and provides additional pose annotations on non-central objects. Compared with LineMod, the images in the Occlusion LineMod dataset contain multiple objects under severe occlusion, making 6D pose estimation extremely challenging.

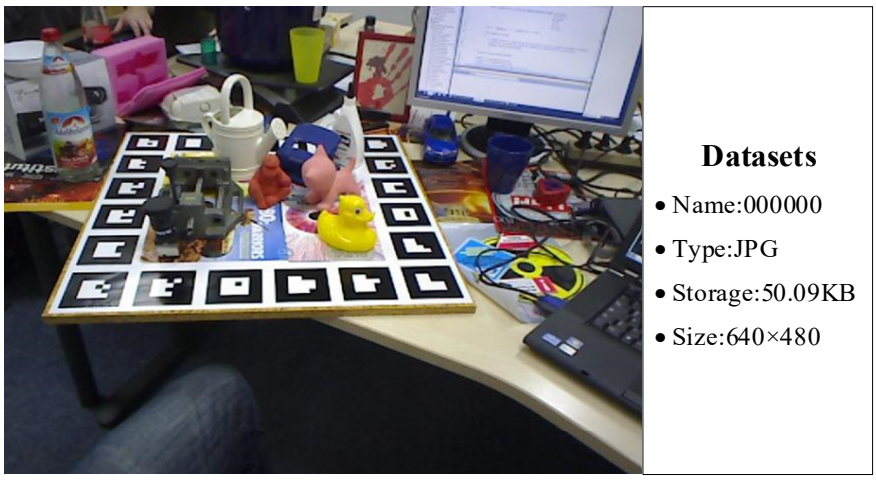


Figure 6. Image information of LineMod dataset.

In order to ensure fairness in conducting comparative experiments with PVNet and related algorithms, the same training test segmentation is used on the LineMod dataset (15% for training and 85% for testing), while the Occlusion LineMod dataset is only used for testing. In addition to the training images provided by LineMod, synthetic images are used to enhance training data. Moreover, this article adopts data augmentation techniques to prevent overfitting, including rotating images between certain angles $(-30°, 30°)$, randomly blurring and cropping with a 50% probability, and randomly changing the original brightness and contrast of each image from 0.9 to 1.1 times. In addition, during the training dataset stage, an Adam optimizer with an initial learning rate of 0.001 is used, and the batch size is set to 20. The network is trained with 100 epochs.

4.1.3. Evaluation Indicators

The performance of CoS-PVNet can be evaluated using two metrics: 2D projection metric and a model point average 3D distance (ADD) metric [29], which can measure pose errors in 2D−3D space. The 2D projection metric mainly measures the average distance between the estimated pose and the 3D model point projection under the real pose, specifically:

$$projection - 2d = \frac{1}{m} \sum_{x \in M} \| K(Rx + T) - K(\hat{R}x + \hat{T}) \| \tag{15}$$

where M represents the set of 3D model points, m is the number of points. K is the inherent matrix of the camera. R and T are estimated rotation and translation matrices, while $\hat{R}$ and $\hat{T}$ are real poses. When the average distance after 2D projection measurement is within 5 pixels, it is considered that the estimated 6D pose is correct.

Two common metrics—ADD (average distance) metric and ADD-S metric—are used to estimate attitude in the ADD metric, which is represented uniformly with ADD (-S) in this paper.

(1) ADD metric: Convert model points based on estimated and ground true attitudes and calculate the average distance between the two conversion point sets. When the distance is less than 10% of the model diameter, the estimated attitude is correct, as shown in Formula (16).

$$V_{ADD} = \frac{1}{\mathrm{m}} \sum_{\mathrm{y} \in \mathrm{W}} \| \left(R_{\mathrm{t}} \mathrm{y} + T_1 \right) - \left(R_{\mathrm{v}} \mathrm{y} + T_2 \right) \| \tag{16}$$

where W represents the set of sampling points for the target 3D model, y represents the point in W, and m represents the total number of sampling point sets. R_{t} and T_1 represent the actual rotation and translation, while R_{v} and T_2 represent the predicted rotation and translation respectively.

(2) ADD-S metric: For symmetric objects, use ADD-S metric, where the average distance is calculated based on the distance to the nearest point. Evaluate the target using ADD-S accuracy and AUC (Area Under Curve) area, where AUC is the area under the accuracy threshold curve, obtained by changing the distance threshold in the evaluation. ADD-S metric is represented by ADD-S, as shown in Formula (17).

$$V_{ADD-S} = \frac{1}{\mathrm{m}} \sum_{\mathrm{y} \in \mathrm{W}} \min_{\mathrm{W}} \| \left(R_{\mathrm{t}} \mathrm{y} + T_1 \right) - \left(R_{\mathrm{v}} \mathrm{y} + T_2 \right) \| \tag{17}$$

4.2. CoS-PVNet Experiment Results

4.2.1. LineMod Dataset Experimental Results

The visualization results of CoS-PVNet on the LineMod dataset for pose estimation are shown in Figure 7. The green 3D border represents the true pose, and the blue border repre-

sents the estimated pose. It can be seen from the figure that CoS-PVNet has high accuracy, with the estimated target object almost overlapping with the estimated bounding box.



Figure 7. Pose estimation results of LineMod dataset.

4.2.2. Occlusion LineMod Dataset Experimental Results

The Occlusion LineMod dataset is only used as a testing set, and the previously trained model can be used for experimental testing. The pose estimation results of the Occupation LineMod dataset are shown in Figure 8. The green 3D border also represents the true pose, and the blue border also represents the estimated pose. Compared with the baseline method PVNet, it can be seen that CoS-PVNet can produce accurate results even in severe occlusion. However, the last column also shows that CoS-PVNet cannot provide sufficient information for 6D pose estimation when the target area is too small.

Therefore, testing the renderings on the LineMod dataset and Occlusion LineMod dataset shows that CoS-PVNet has a good overlap effect, indicating that CoS-PVNet still has high accuracy in complex backgrounds. However, when the object target is too small, it cannot accurately estimate its 6D pose, which is related to overfitting caused by the weight self-learning module in CoS-PVNet. Correspondingly, this article uses data augmentation to prevent this from occurring.

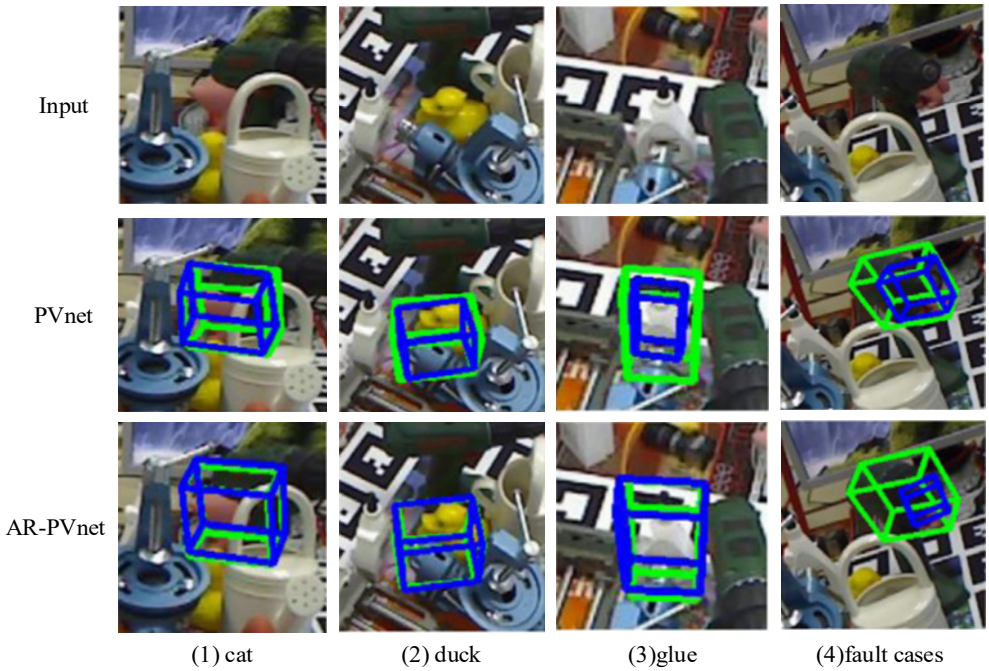


Figure 8. Pose estimation results of Occlusion LineMod dataset.

4.3. CoS-PVNet Comparative Experiment

4.3.1. Comparison Experiment of 2D Projection Metrics

CoS-PVNet is compared with relevant RGB image-based pose estimation methods. On the LineMod dataset and Occlusion LineMod dataset, CoS-PVNet is compared quantitatively with BB8 [30], YOLO-6D [12], and PVNet [21] in 2D projection metrics. The experimental results of 2D projection metric comparisons are shown in Table 2.

Table 2. Comparison results of 2D projection metrics (unit: %).

Algorithms		BB8	YOLO-6D	PVNet	CoS-PVNet
LineMod	ape	95.30	92.10	99.20	**99.60**
	can	84.10	97.40	99.56	**99.60**
	cat	97.00	97.30	99.30	**99.76**
	eggbox	87.90	90.33	99.24	**99.45**
	duck	81.20	94.60	98.00	**98.10**
	glue	89.00	93.24	98.40	**98.44**
	Mean	89.08	94.16	98.95	**99.16**
Occlusion LineMod	cat	3.62	10.40	64.52	**65.59**
	duck	5.07	31.80	61.44	**63.47**
	glue	4.86	29.63	54.28	**55.13**
	Mean	4.52	23.94	60.08	**61.40**
Mean		46.8	59.05	79.52	**80.28**

Note: Bold represents the maximum value of each row, and the meaning expressed in subsequent tables is the same.

As shown in Table 2, BB8 and YOLO-6D use the eight corners of a 3D bounding box plus an object center as the key point and directly regress its coordinates, while PVNet and CoS-PVNet apply a voting strategy to locate eight surface key points and one object center from the predicted vector field. When using the same loss as PVNet, CoS-PVNet achieves better performance on most objects, increasing the average accuracy of BB8 by 10.08% on the LineMod dataset, especially increasing the accuracy of target objects can and cat by more than 15%. This indicates that CoS-PVNet is also more accurate for small-scale object pose estimation. In the case of target occlusion, CoS-PVNet is improved by 37.46%

on the 2D projection metric measurement of target object categories compared to YOLO-6D. CoS-PVNet also shows better performance compared to the PVNet, with an improvement of 1.32% in the 2D Projection metric evaluation. Therefore, compared with the indicator evaluation results on the LineMod dataset mentioned above, CoS-PVNet performs better than PVNet in complex occlusion scenes on the Occlusion LineMod dataset, which also proves the correctness of the CoS-PVNet pose estimation in this paper.

4.3.2. Comparative Experiment of CoS-PVNet Algorithm ADD (-S)

(1) LineMod Dataset ADD (-S) Comparative Experiment

Experiments are conducted on the LineMod dataset, comparing CoS-PVNet with algorithms such as YOLO-6D [12], PoseCNN [17], DenseFusion [31], Dual Stream [32], and PVNet [21]. Two symmetrical objects, egg-box and glue, are evaluated using the ADD-S metric, while other objects are evaluated using the ADD metric. The comparative experimental results are shown in Table 3.

Table 3. Comparison of precision results for LineMod dataset (unit: %).

LineMod	Algorithms					
	YOLO-6D	**PoseCNN**	**DenseFusion**	**DualStream**	**PVNet**	**CoS-PVNet**
ape	21.6	77.0	**92.3**	91.3	43.6	88.1
benchwise	81.8	97.5	93.2	93.5	99.9	**100.0**
cam	36.6	93.5	**94.4**	94.0	86.9	94.3
can	68.8	96.5	93.1	94.3	95.5	**97.1**
cat	41.8	82.1	**96.5**	95.8	79.3	92.3
driller	63.5	95.0	87.0	92.9	96.4	**99.2**
duck	27.2	77.7	92.3	**94.7**	52.6	92.1
egg-box	69.9	97.1	99.8	99.9	99.2	**100.0**
glue	80.0	99.4	**100.0**	99.9	95.7	97.3
hole puncher	42.6	52.8	92.1	**92.8**	81.9	86.9
iron	74.9	98.3	97.0	95.1	98.9	**99.3**
lamp	71.1	97.5	95.3	94.6	99.3	**99.7**
phone	47.7	87.7	92.8	94.0	92.4	**94.5**
Mean	55.9	88.6	94.3	94.8	86.3	**95.4**

As shown in Table 3, CoS-PVNet has improved the average values of YOLO-6D, PoseCNN, DenseFusion, DualStream, and PVNet algorithms by 39.5%, 6.8%, 1.1%, 0.6%, and 9.1% respectively. For four types of objects: ape, cat, duck, and hole puncher, the accuracy improvement of pose estimation is relatively small. The reason for this is that CoS-PVNet has certain advantages in extracting features for large-scale target objects, and when the ADD metric is less than 10% of the maximum diameter of the target during testing, the pose estimation is considered to be correct. The maximum diameter of the above four types of targets is small, resulting in relatively small improvement. CoS-PVNet performs better in estimating the pose of other target objects, indicating that CoS-PVNet fully extracts features of the target object and can effectively improve the accuracy of 6D pose estimation for objects in complex scenes.

(2) Comparison Experiment of the Occupation LineMod Dataset ADD (-S)

Experiments are conducted on the Occlusion LineMod dataset to compare CoS-PVNet with HybridPose [33], SSPE [34], RePOSE [35], SegDriven [36], PoseCNN [17] and PVNet [21]. Using the same indicators as the test LineMod dataset, the comparative experimental results are shown in Table 4.

Table 4. Comparison of precision results of occlusion LineMod dataset (unit: %).

Occlusion LineMod	Algorithms						
	HybridPose	**SSPE**	**RePOSE**	**SegDriven**	**PoseCNN**	**PVNet**	**CoS-PVNet**
ape	20.9	19.2	31.1	12.1	9.6	15.8	**22.9**
can	75.3	65.1	**80.0**	39.9	45.2	63.3	74.6
cat	24.9	18.9	**25.6**	8.2	0.9	16.7	25.4
driller	70.2	69.0	73.1	45.2	41.4	65.7	**74.2**
duck	27.9	25.3	**43.0**	17.2	19.6	25.2	35.7
egg-box	52.4	52.0	51.7	22.1	22.0	50.2	**52.9**
glue	53.8	51.4	54.3	35.8	38.5	49.6	**56.5**
hole puncher	**54.2**	45.6	53.6	36.0	22.1	39.7	51.7
Mean	47.5	43.3	51.6	27.0	24.9	40.8	**49.2**

As shown in Table 4, CoS-PVNet outperforms HybridPose, SSPE, SegDriven, PoseCNN, and PVNet on the average mean values, with improvements of 1.7%, 5.9%, 22.2%, 24.3%, and 8.4%, respectively. However, CoS-PVNet is 2.4% lower than RePOSE on the average mean value, mainly focusing on three aspects: can, cat, and duck. This indicates that RePOSE can quickly and accurately refine the pose by minimizing the feature measurement error between input and rendered image representations. However, when small targets are severely occluded, or the extracted features are insufficient to recognize the target object well, the performance of CoS-PVNet is even better.

4.4. CoS-PVNet Ablation Experiment

To verify the effectiveness of each module, ablation experiments are conducted to analyze each module of CoS-PVNet. Table 5 shows the results of gradually adding the CoS-PVNet algorithm modules separately for comparison. Due to the significant improvement in some categories on the LineMod dataset, ablation experiments have a certain representativeness. Therefore, this paper uses the LineMod dataset for accuracy and speed testing.

Table 5. CoS-PVNet pose estimation ablation experiment.

Pose Estimation Method	Accuracy (%)	Speed (FPS)
CoS-PVNet weight self-learning + PnP algorithm	46.3	14
CoS-PVNet weight self-learning + Global attention mechanism + PnP algorithm	63.6	21

As shown in Table 5, the accuracy and velocity of pose estimation for different modules of CoS-PVNet are shown. If the predicted translation and rotation errors with the actual pose are less than 5 cm and 5°, respectively, it is considered that the predicted object pose is correct. If the CoS-PvNet weight self-learning module is directly added, the accuracy and speed of using the PnP algorithm to solve pose are 46.3% and 14 FPS. If the CoS-PvNet global attention mechanism is added to infer a pose, the accuracy is improved by 17.3%, and the FPS is improved by 7 FPS. Therefore, if the CoS-PVNet weight self-learning module is directly added, the PnP algorithm is prone to incomplete simulation or overfitting, resulting in lower accuracy of CoS-PVNet pose estimation. However, CoS-PVNet directly utilizes the local and global context information of the global attention mechanism feature map, further improving the robustness of CoS-PVNet pose estimation.

5. Discussion

Object 6D pose estimation is a core technology for applications such as AR, VR, robotics, and autonomous driving. However, due to complex scene factors, such as background clutter, target occlusion, and weak texture features, it can easily lead to inaccurate 6D pose estimation. This article proposes a robust CoS-PVNet pose estimation network

for complex scenes. Firstly, by adding a pixel-weight self-learning layer on the basis of the PVNet network structure, the pixel-weight values are predicted to be selected for voting. Then, stable and robust useful features are extracted using the global attention mechanism of local and global contextual information in the input feature map. Finally, the PnP algorithm is used to solve the 6D pose, which improves the accuracy and robustness of 6D object pose estimation in complex scenes.

6D object pose estimation is an important research topic in the field of computer vision, which determines the 3D position and direction of an object in the camera center coordinate system. In the field of AR, virtual elements can be superimposed on objects to maintain their relative pose as they move. With the maturity of technologies such as SLAM, robots have been able to perform good positioning in 3D space, but 6D pose estimation technology is still needed for object grasping interaction. In the field of autonomous driving, the 6D pose estimation assistance mode can achieve dynamic 360° panoramic driving. In this paper, by adding pixel-weight layers on the basis of the PVNet network, more accurate pixel point vectors are selected, and the pose of the object is estimated based on local and global contextual information of the feature map, and then the coordinate system transformation matrix is solved. CoS-PVNet for virtual real fusion interactive application framework is shown in Figure 9. By using feature detection operators to extract key feature points and descriptors from real-world scene images and matching them with the corresponding natural feature templates constructed offline, CoS-PVNet is used to solve the pose of AR cameras and assembly objects through geometric visual transformation [37], and 3D virtual real interaction technology is used to empower stable and robust virtual real fusion interactive applications of AR, VR, robotics, and autonomous driving.

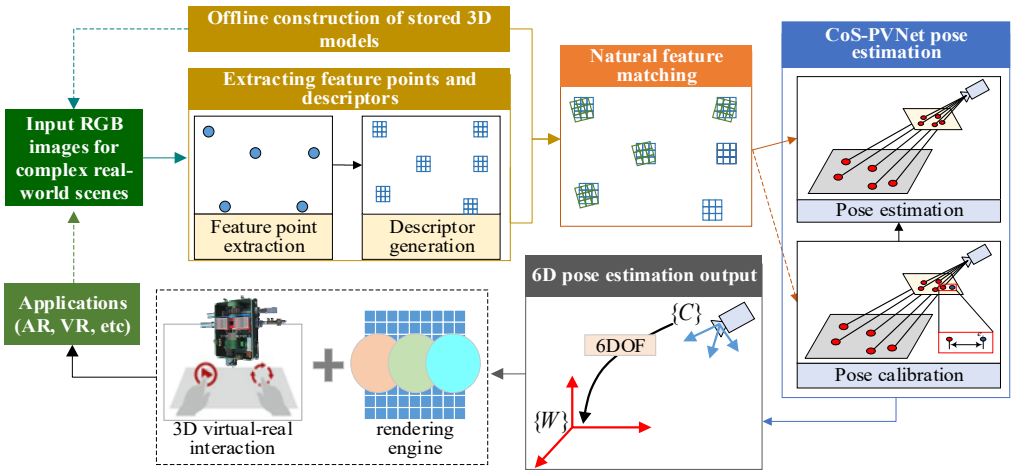


Figure 9. CoS-PVNet algorithm virtual real fusion interactive application framework.

In recent years, 6D pose estimation methods have made significant progress in fields such as AR registration, robot grasping, and autonomous driving navigation. However, the lack of higher dimensional semantic modeling and understanding of specific complex interactive application scenarios has made it difficult to meet the accuracy and robustness of 6D pose estimation in different job scenarios [38]. On the other hand, with the optimization of deep learning models and the development of new architectures, a 6D pose estimation algorithm will be able to process object recognition and pose estimation in complex scenes more quickly. Although the CoS-PVNet pose estimation algorithm proposed in this article has achieved good results on the LineMod and Occlusion LineMod dataset, the dynamic uncertainty of "human-machine-object" in AR, VR, robotics, and autonomous driving [39] makes pose estimation for severely occluded and truncated target objects still difficult and important in the field of 6D object pose estimation [40]. Therefore, there is still a lot of room for improvement in the accuracy of 6D pose estimation in complex scenes. Future work will utilize the latest advances in target area semantic segmentation models to

accelerate the inference process and consider combining reinforcement learning to achieve active 6D object pose estimation [3]. This will also provide support for improving the system performance of AR, VR, robotics, and autonomous driving, effectively promoting the digital and intelligent transformation and upgrading of manufacturing, transportation, and other industries.

6. Conclusions

We propose a robust CoS-PVNet pose estimation network for complex scenes to address the low accuracy in object 6D pose estimation. By adding pixel-weight self-learning layers on the basis of PVNet, more accurate pixel point vectors are selected, and a global attention mechanism is proposed to improve the performance of feature extraction by adding contextual information, thereby estimating the pose of CoS-PVNet target objects and solving the CoS-PVNet coordinate system transformation matrix, providing support for the implementation of AR, VR, robotics, and autonomous driving. The performance of CoS-PVNet is evaluated on the LineMod and Occlusion LineMod datasets. The experimental results show that CoS-PVNet can accurately estimate the 6D pose of target objects and effectively estimate the 6D pose of occluded objects with higher accuracy and robustness. However, this study also has limitations in not fully integrating geometric, normal, and other multivariate features. The next step is to deeply integrate industry application context feature information to adapt to more complex industry application scenarios.

Author Contributions: Conceptualization, J.Y.; methodology, J.Y.; software, J.Y. and X.L.; validation, X.L.; formal analysis, J.D. and Y.W.; data curation, X.L.; writing—original draft preparation, J.Y.; writing—review and editing, J.Y., X.L., J.D. and Y.W.; visualization, J.Y. and X.L.; supervision, J.D. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China, grant numbers 62367005 and 62067006; in part by the Research Projects of the Humanities and Social Sciences Foundation of the Ministry of Education of China, grant numbers 21YJC880085; in part by the Natural Science Foundation of Gansu Province, grant numbers 23JRRA845; and in part by the Youth Science and Technology Talent Innovation Project of Lanzhou, grant numbers 2023-QN-117.

Data Availability Statement: The data are contained within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Baroroh, D.K.; Chu, C.H.; Wang, L. Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *J. Manuf. Syst.* **2021**, *61*, 696–711. [CrossRef]
2. Parger, M.; Tang, C.; Xu, Y.; Twigg, C.D.; Tao, L.; Li, Y.; Wang, R. UNOC: Understanding occlusion for embodied presence in virtual reality. *IEEE Trans. Vis. Comput. Graph.* **2021**, *28*, 4240–4251. [CrossRef] [PubMed]
3. Li, W.; Wang, J.; Liu, M.; Zhao, S.; Ding, X. Integrated registration and occlusion handling based on deep learning for augmented reality assisted assembly instruction. *IEEE Trans. Ind. Inform.* **2022**, *19*, 6825–6835. [CrossRef]
4. Gonzalez, M.; Kacete, A.; Murienne, A.; Marchand, E. L6dnet: Light 6 DoF network for robust and precise object pose estimation with small dataset. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2914–2921. [CrossRef]
5. Hansen, L.H.; Fleck, P.; Stranner, M.; Schmalstieg, D.; Arth, C. Augmented reality for subsurface utility engineering, revisited. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 4119–4128. [CrossRef] [PubMed]
6. Haouchine, N.; Juvekar, P.; Nercessian, M.; Wells, W.; Golby, A.; Frisken, S. Pose estimation and non-rigid registration for augmented reality during neurosurgery. *IEEE Trans. Biomed. Eng.* **2021**, *69*, 1310–1317. [CrossRef] [PubMed]
7. Lee, T.; Lee, B.-U.; Kim, M.; Kweon, I.S. Category-level metric scale object shape and pose estimation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8575–8582. [CrossRef]
8. Kirch, S.; Olyunina, V.; Ondřej, J.; Pagés, R.; Martin, S.; Pérez-Molina, C. RGB-D-Fusion: Image Conditioned Depth Diffusion of Humanoid Subjects. *IEEE Access* **2023**, *11*, 99111–99129. [CrossRef]
9. Romero-Ramire, F.J.; Munoz-Salinas, R.; Medina-Carnicer, R. Fractal Markers: A new approach for long-range marker pose estimation under occlusion. *IEEE Access* **2019**, *7*, 169908–169919. [CrossRef]
10. Sarmadi, H.; Munoz-Salinas, R.; Berbis, M.A.; Medina-Carnicer, R. Simultaneous multi-view camera pose estimation and object tracking with squared planar markers. *IEEE Access* **2019**, *7*, 22927–22940. [CrossRef]

11. Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1521–1529.
12. Tekin, B.; Sinha, S.N.; Fua, P. Real-time seamless single shot 6d object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 292–301.
13. Li, Z.; Wang, G.; Ji, X. CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-Dof object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7678–7687.
14. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. Yu, S.; Zhai, D.-H.; Xia, Y. Robotic Grasp Detection Based on Category-Level Object Pose Estimation with Self-Supervised Learning. *IEEE/ASME Trans. Mechatron.* **2024**, *29*, 625–635. [CrossRef]
17. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
18. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 3431–3440.
19. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
20. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer International Publishing: New York, NY, USA, 2015; pp. 234–241.
21. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNet: Pixel-wise voting network for 6D of pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4561–4570.
22. Wang, L.; Wu, X.; Zhang, Y.; Zhang, X.; Xu, L.; Wu, Z.; Fei, A. DeepAdaIn-Net: Deep Adaptive Device-Edge Collaborative Inference for Augmented Reality. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 1052–1063. [CrossRef]
23. Tang, F.; Wu, Y.; Hou, X.; Ling, H. 3D map and 6D pose computation for real time augmented reality on cylindrical objects. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2887–2899. [CrossRef]
24. Yu, G.; Hu, Y.; Dai, J. TopoTag: A robust and scalable topological fiducial marker system. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 3769–3780. [CrossRef]
25. Zhu, Y.; Wan, L.; Xu, W.; Wang, S. ASPP-DF-PVNet: Atrous Spatial Pyramid Pooling and Distance-Filtered PVNet for occlusion resistant 6D estimation. *Signal Process. Image Commun.* **2021**, *95*, 116268. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. 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 Proceedings of the Asian Conference on Computer Vision (ACCV), Daejeon, Republic of Korea, 5–9 November 2012; pp. 548–562.
28. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 536–551.
29. Brachmann, E.; Michel, F.; Krull, A.; Yang, M.Y.; Gumhold, S. Uncertainty-driven 6d pose estimation of objects and scenes from a single RGB image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3364–3372.
30. 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 Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
31. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3343–3352.
32. Li, Q.; Hu, R.; Xiao, J.; Wang, Z.; Chen, Y. Learning latent geometric consistency for 6D object pose estimation in heavily cluttered scenes. *J. Vis. Commun. Image Represent.* **2020**, *70*, 102790. [CrossRef]
33. Song, C.; Song, J.; Huang, Q. Hybridpose: 6D object pose estimation under hybrid representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 431–440.
34. 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, Seattle, WA, USA, 14–19 June 2020; pp. 2930–2939.
35. Iwase, S.; Liu, X.; Khirodkar, R.; Yokota, R.; Kitani, K.M. Repose: Fast 6D object pose refinement via deep texture rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 3303–3312.
36. Hu, Y.; Hugonot, J.; Fua, P.; Salzmann, M. Segmentation-driven 6D object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3385–3394.

37. Assa, A.; Janabi-Sharifi, F. A robust vision-based sensor fusion approach for real-time pose estimation. *IEEE Trans. Cybern.* **2013**, *44*, 217–227. [CrossRef] [PubMed]
38. Abhiraj, D.; Inki, K. The effects of augmented reality on improving spatial problem solving for object assembly. *Adv. Eng. Inform.* **2018**, *38*, 760–775.
39. Pang, J.; Zheng, P.; Li, S.; Liu, S. A verification-oriented and part-focused assembly monitoring system based on multi-layered digital twin. *J. Manuf. Syst.* **2023**, *68*, 477–492. [CrossRef]
40. Tao, W.; Lai, Z.-H.; Leu, M.C.; Yin, Z.; Qin, R. A self-aware and active-guiding training & assistant system for worker-centered intelligent manufacturing. *Manuf. Lett.* **2019**, *21*, 45–49.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.