applied sciences

MDPI

Article

A Novel Metric-Learning-Based Method for Multi-Instance Textureless Objects' 6D Pose Estimation

Chenrui Wu *, Long Chen and Shiqing Wu

College of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; cl@usst.edu.cn (L.C.); wushiqing@usst.edu.cn (S.W.)
* Correspondence: wuchenrui@usst.edu.cn

Abstract: 6D pose estimation of objects is essential for intelligent manufacturing. Current methods mainly place emphasis on the single object's pose estimation, which limit its use in real-world applications. In this paper, we propose a multi-instance framework of 6D pose estimation for textureless objects in an industrial environment. We use a two-stage pipeline for this purpose. In the detection stage, EfficientDet is used to detect target instances from the image. In the pose estimation stage, the cropped images are first interpolated into a fixed size, then fed into a pseudo-siamese graph matching network to calculate dense point correspondences. A modified circle loss is defined to measure the differences of positive and negative correspondences. Experiments on the antenna support demonstrate the effectiveness and advantages of our proposed method.

Keywords: 6D pose estimation; metric learning; dense correspondences; antenna support

Citation: Wu, C.; Chen, L.; Wu, S. A Novel Metric-Learning-Based Method for Multi-Instance Textureless Objects' 6D Pose Estimation. *Appl. Sci.* **2021**, *11*, 10531. https://doi.org/10.3390/app112210531

Academic Editors: Xinyue Zhao, Zheng Chen and Ming Fang

Received: 21 October 2021
Accepted: 4 November 2021
Published: 9 November 2021

1. Introduction

Estimating 6D pose, i.e., 3D translation and 3D rotation of a target, is a fundamental problem in intelligent manufacturing, especially in the application fields of object grasping [1,2], assembling [3,4], bin-picking [5,6], and stacking [7] with the help of the visual sensors.

Visual sensors in an industrial environment can mainly be divided into three categories, namely, RGB, D, and RGB-D sensors. RGB sensors only achieve color information through a CMOS unit. D sensors use structured light, lidar injector–receiver, or radar injector–receiver to measure the distance from the camera to the target. RGB-D sensors combine both RGB and D sensors and leverage the calibration method to assign color information onto the depth information. However, there are limitations for D sensors in industrial environments [8]. On one hand, using depth sensors in industrial environments are not always useful, as there are plant of non-Lambert surface objects such as metal parts, glasses, and ceramics, which have uncertain reflection ratios for the light to make the depth immeasurable. On the other hand, thanks to the fast development of the deep learning technologies in recent years, the performance of 6D pose estimation methods using only RGB information is comparable with those using RGB-D information [9,10]. Therefore, we focus on the investigation of RGB-based 6D pose estimation method in this paper.

Traditional methods use different kinds of hand-crafted descriptors [11–13] to extract features surround the image points to establish the feature descriptions of the image points. The property of scale and rotation invariant is always considered to ensure the feature similarity of the same point of an object under different point-of-view in the image. These methods are sufficient for rich textured objects because of the variant color gradients of their surfaces; however, they are not capable of obtaining distinguishable point features from textureless surfaces such as metal, glasses, and ceramics. To solve this problem, geometric features such as lines [14,15], moments [16], circles [17], and gradients of edges [18,19], which can represent the geometric structures of an object, have been designed to describe the implicit features. Properties that are invariant to scale and rotation have also studied on

these geometric features [20]. However, the geometric features usually describe the overall structure of an object. When they are invariant to rotation and scale, they are only useful in object detection from an image, but lose the ability to distinguish different translation and rotation of the object.

With the fast development of deep learning technologies in recent years, many researchers have used deep neural networks to predict the 6D pose of a textureless object. SSD-6D [10] uses a direct regression strategy to predict a translation and orientation based on the popular SingleShot multibox Detector (SSD) object detection framework. DeepIM [21] proposes a CNN structure to iteratively measure the difference between the current 2D image projection of the predicted pose and the real 2D image. A deep neural network that outputs the optic flow between the two images was designed to provide pose refinement for the current pose. [22] combines semantic key-points predicted by a convolutional network with a deformable shape model to determine the 2D–3D correspondences. PVNet [9] regresses pixelwise vectors pointing to the key-points with a modified U-Net structure and proposes a voting scheme to decide the location of the key-points. HybridPose [23] extends the approach of PVNet [9] by utilizing a hybrid intermediate representation to express different geometric information in the input image, including key-points, edge vectors, and symmetry correspondences. CosyPose [24] develops a robust method for matching individual 6D object pose hypotheses across different input images in order to jointly estimate camera viewpoints and 6D poses of all the objects in a single consistent scene.

Recently, finding dense correspondences using the deep neural networks has shown advantages in 6D pose estimation [25,26]. The per-pixel matching scheme was utilized to design and train the network. In [26], a pseudo-siamese matching network was proposed to match dense correspondences in high-dimension; then, the dense correspondences were used to calculate the target pose through Perspective-n-Points (PnP) method [27]. This method achieved state-of-the-art performance in LineMod [28] and Occlusion-LineMod [29] datasets. However, both datasets contain only one instance for each object. The network is designed to directly segment the object in the image. Thus, it is not applicable for multitarget pose estimation tasks. In this paper, we improve this method in two main aspects for industrial usage.

(1) We adopt EfficientDet [30] to first detect every object in the image. Each object in the image is then cropped through the bounding box provided by the EfficientDet and resized into fixed value. All the resized images are fed into the correspondences matching network to predict dense correspondences. After obtaining the correspondences, PnP-Ransac method is used to calculate the 6D pose of the target. By adopting the two-stage network structure, we solve the problem for multi-instance 6D pose estimation.

(2) We introduce the circle loss, a well-known loss function in metric learning, to measure the similarities between pixelwise deep features from a 2D image and nodewise deep features from a 3D mesh model. We analyze the reason why the softmax cross-entropy loss [31] used in [26] is not suitable for dense correspondences matching and compare the proposed masked circle loss with the softmax cross-entropy loss though ablation studies to show the superiority of the proposed loss.

In summary, our main contribution lies in the framework of the 6D pose estimation that can deal with multiple instances in single frame and a novel metric learning loss that efficiently constrains the matching of the 2D–3D correspondences.

The remainder of this paper is organized as follows: In Section 2, we introduce the whole two-stage 6D pose estimation framework for multi-instance textureless objects. The masked circle loss for 2D–3D correspondences matching is introduced in detail. In Section 3, we test our proposed method on the pose estimation problem for the antenna support and compare it with some other state-of-the-art methods to show the effectiveness and advantages of our method. Conclusions are drawn in Section 4.

2. Methodology

Given an RGB image, the main purpose of the 6D pose estimation is to predict a rotation matrix $R \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ from the objects' coordinate system to the camera coordinate system. When the pose is accurately detected, the transformation between the industrial robot and the object can be easily inferred for further action such as object grasping or assembling. In fact, the 6D pose estimation problem can be divided into two subtasks: (1) Find out the target objects from the image. (2) Calculate the poses for all the target objects. Most of the existing works [9,25,26,32] solve the two problems in a unified framework for boosting the performance on commonly used open-evaluation datasets such as LINEMOD, Occlusion LINEMOD, and YCB Video. However, all of these datasets only contain a single target for each of the classes in one frame. When there are plenty of targets of the same type, the model cannot handle the situation well. Therefore, in this paper, we propose a two-stage framework to separately solve the 6D pose estimation problem for multi-instance environments.

2.1. Overview

In this section, we introduce the framework of the proposed multi-instance pose estimation method in detail. The framework consists of four modules, namely, the object detection module, mesh feature encoding module, image feature encoding module, and pose estimation module. The flowchart of the framework is shown in Figure 1.

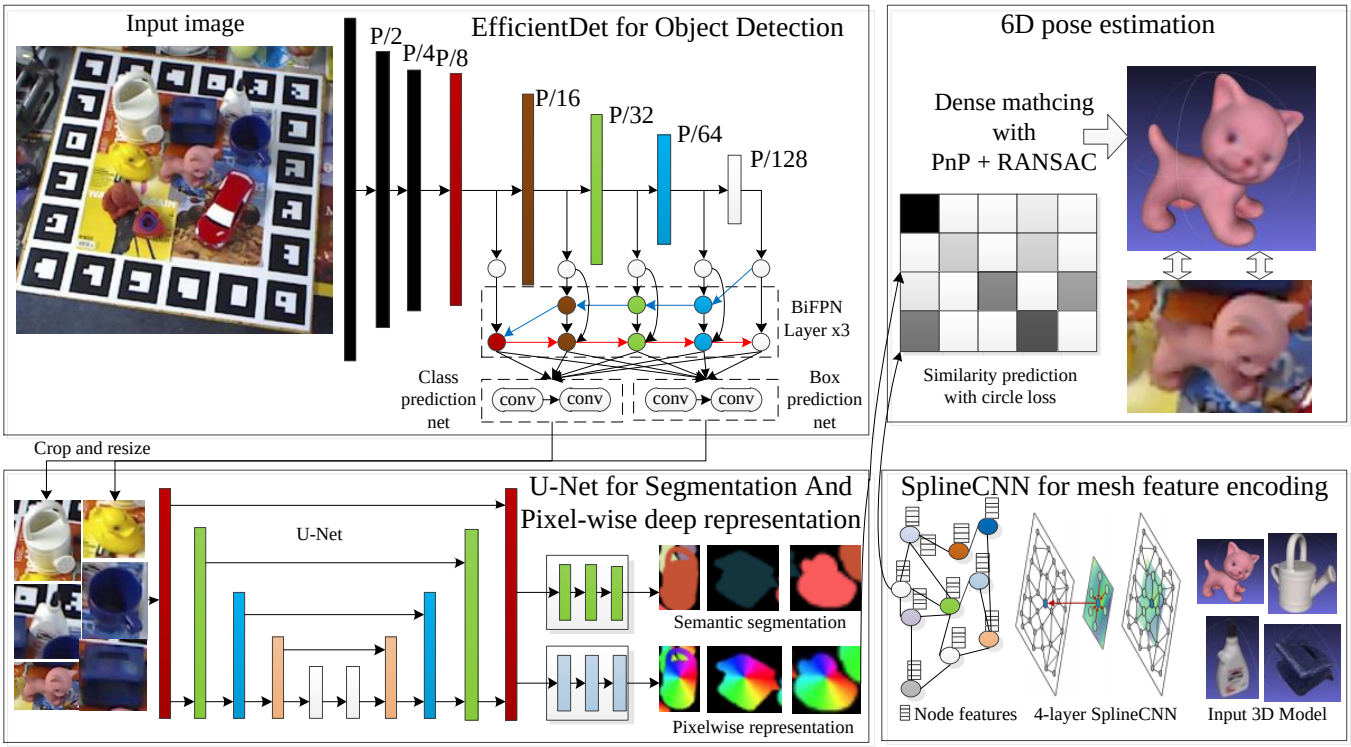


Figure 1. Flowchart of our proposed two-stage pose estimation framework.

The input of the model is an RGB image taken by an industrial camera. The image is first fed into the object detection module to find out the bounding boxes for each object in the image. We chose to use EfficientDet [30] in this module due to its light weight and high performance among current commonly used object detection modules. The EfficientDet offers seven versions with different model sizes to fit the need for variant applications. The bi-FPN used in the EfficientDet can effectively extract useful features for different kinds of objects.

After each bounding box of the objects in the image was correctly obtained, the objects were cropped out of the image through the bounding box. We expanded the bounding box by δ_w and δ_h in width and height, respectively, to ensure the object is inside the bounding boxes. The bilinear interpolation method was used to resize all the cropped images into a fixed size (H_{crop}, W_{crop}). Then, the resized images were parallely fed into the image feature encoding module to achieve the deep representation for each pixel.

The image feature encoding module utilized the U-Net structure as the backbone for feature extraction. The cropped images with the same size were fed into the U-Net to extract deep features. Two multiple-fully-connected layers were designed to predict the semantic segmentation and pixelwise deep representation, respectively. The function of the U-Net can be represented as $F_I = \Lambda_{\theta_{unet}}(I)$, where I denotes the input cropped image of an object. θ_{unet} is the parameter of the U-Net model. $F_I = (F_I^{seg}, F_I^{feat}) \in \mathbb{R}^{(C+1+D)\times H\times W}$ indicates the output tensor of the U-Net. The $F_I^{seg} \in \mathbb{R}^{(C+1)\times H\times W}$ part of the tensor is responsible for the semantic segmentation for C classes objects while the $F_I^{feat} \in \mathbb{R}^{D\times H\times W}$ part of the tensor represents the pixelwise D dimensional deep features of the object in the image.

In the mesh feature encoding module, a 4-layer SplineCNN $F_M = \Phi_{\theta_{spline}}(M) \in \mathbb{R}^{D\times L}$ is used to extract nodewise deep features from the 3D mesh model. M is the 3D mesh model of the target object. θ_{spline} denotes the parameters of the SplineCNN. F_M represents the node features calculated through the SplineCNN, where L is the number of the nodes from the 3D mesh model. The affinity submodule explicitly provides an affine transformation between the pixelwise deep features and nodewise deep features through Equation (1).

$$s_{i,j}^k = F_{M_{x_i}}^k A_k F_{I_{y_j}} \tag{1}$$

where $A_k \in \mathbb{R}^{D\times D}$ are the learnable parameters of the affinity submodule for the k-th object. $F_M^{x_i} \in \mathbb{R}^D$ and $F_I^{y_j} \in \mathbb{R}^D$ are the x_i pixel and y_j node in the image and 3D mesh model, respectively. This submodule provides the ability for the network to learning affine-invariance features that can match with each other through feature similarity $s_{i,j} \in \mathbb{R}$.

In the pose estimation module, the deep features encoded from the image feature encoding module and the mesh feature encoding module are multiplied through dot product to calculate the similarity of the correspondences. The features with the maximum similarities were chosen as the 2D–3D correspondences. As there was one correspondence from 3D model for each pixel in the RGB image, dense correspondences were directly obtained for the RANSAC-based PnP method to calculate relative pose from the camera to the object.

2.2. Masked Circle Loss for Matching Dense Correspondences

The core operation in dense 2D–3D correspondences matching is to calculate the similarity between the pixelwise deep features from an image and the nodewise deep features from a 3D model. The cosine similarity is used to measure the distance between the features

$$\mathcal{S}_k = F_M^k A_k F_I \tag{2}$$

where $\mathcal{S}_k$ denotes the similarity matrix for object k. In [26], the softmax cross-entropy loss—which is the most generally used loss function for traditional classification problem—was chosen to select the corresponding node from 3D model for each image pixel that belongs to the target object. The lost function can be described as

$$\mathcal{L}_i = -log\frac{e^{s_{il}}}{\sum_{j=1}^{n} e^{s_{ij}}} \tag{3}$$

where s_{ij} denotes the similarity between the i-th pixel in the image and the j-th node from the 3D model. l is the correct label for the matching. The softmax step $p_{iq} = \frac{e^{s_{il}}}{\sum_{j=1}^{n} e^{s_{ij}}}$, $q = 1 \ldots n$ turns each similarity s_{ij} into a probability p_{ij}. Then, the p_{ij} is used to calculate the cross entropy with the one-hot vector, which only the true class equals to one, while all the other classes remain zero. The gradient of the j-th node in the softmax cross entropy loss is

$$\frac{\partial \mathcal{L}_i}{\partial s_{ij}} = \begin{cases} p_{ij} - 1, & i = l \\ p_{ij}, & i \neq l \end{cases} . \tag{4}$$

As shown in Equation (4), the gradient of the true class is $p_{il} - 1$, which means the network is trained to make the similarity of the true class to be one, while the similarity of the false class to be zero. However, in the case of feature matching, the divergence among the classes is not as large as that in the traditional classification problems.

As shown in Figure 2, the red point denotes the true class matching from the image pixel i to the node j in 3D model. The green circle denotes a nearby region for node j. As in the softmax cross-entropy loss, all the nodes in the green circle are trained to have zero similarities with respect to pixel i while the node j is trained to have a similarity of one. This situation is apparently not reasonable for the training. In fact, the main purpose of the correspondence matching is to find the *most similar* node from 3D model for pixel i instead of the *same* node from 3D model. Thus, it is more suitable to learn a distance metric for the 2D–3D correspondences.

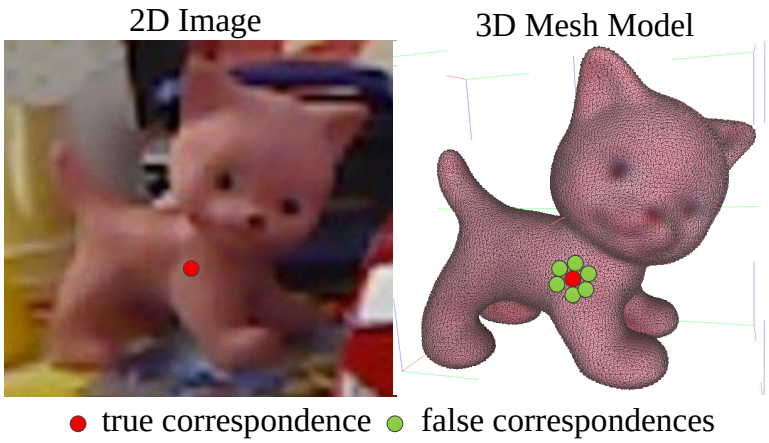


Figure 2. The illustration for the dense matching correspondences using softmax cross-entropy.

Metric learning, which is also known as similarity learning, is a conventional research area before the deep learning era. Deep metric learning introduces deep neural networks into conventional metric learning. One of the most popular metrics of learning loss is contrastive loss

$$\mathcal{L}_C = \begin{cases} \| f_i - f_j \|_2^2, & c_i = c_j \\ max(0, m - \| f_i - f_j \|_2^2), & c_i \neq c_j \end{cases} \tag{5}$$

where m is a margin among different classes and c_i denotes the i-th class. Another well-known metric loss is triplet loss

$$\mathcal{L}_T = max(0, m + \| f_i - f_j \|_2^2 - \| f_i - f_k \|_2^2), \qquad c_i = c_j, ci \neq c_k \tag{6}$$

The main difference between these two methods is that triplet loss stops the optimization of the inner class distance $\| f_i - f_j \|_2^2$ when the condition $m + \| f_i - f_j \|_2^2 - \| f_i - f_k \|_2^2 < 0$ is fulfilled, while the contrastive loss always optimizes the distance among

features that belong to the same class. Apparently, triplet loss is more suitable for the task of dense feature matching, as the similarity of the true correspondences does not have to be one, it only needs to be more similar with its correspondence compared with the others.

Circle loss [33] proposes a unified perspective of view to explain the triplet loss and the softmax cross-entropy loss. Assume there are K_{in} within-class similarities and K_{out} between-class similarities, which are denoted by $s_p^i (i = 1, 2, ..., K_{in})$ and $s_n^j (j = 1, 2, ..., K_{out})$, respectively; p and n mean the positive and negative similarity, respectively.

In order to minimize $s_n^j (\forall j \in 1, 2, ..., K_{out})$ as well as to maximize $s_p^i (\forall i \in 1, 2, ..., K_{in})$, the unified loss function can be designed as

$$\begin{aligned} \mathcal{L}_{uni} &= log[1 + \sum_{i=1}^{K_{in}} \sum_{j=1}^{K_{out}} exp(\gamma(s_n^j - s_p^i + m))] \\ &= log[1 + \sum_{j=1}^{K_{out}} exp(\gamma s_n^j) \sum_{i=1}^{K_{in}} exp(\gamma(-s_p^i + m))] \\ &= -log \frac{\sum_{i=1}^{K_{in}} exp(\gamma(s_p^i - m))}{\sum_{i=1}^{K_{in}} exp(\gamma(s_p^i - m)) + \sum_{j=1}^{K_{out}} exp(\gamma s_n^j)} \end{aligned} \tag{7}$$

where γ is a scale factor. We can find out that if we set $\gamma = 1$, $m = 0$, and $K_{in} = 1$, Equation (7) degenerates to the softmax cross-entropy loss, as shown in Equation (3). The main purpose of the function is to minimize $(s_n - s_p)$, in which reducing s_n is equivalent to increasing s_p. Circle loss introduces $(\alpha_n s_n - \alpha_p s_p)$ instead of $(s_n - s_p)$, where

$$\begin{cases} \alpha_p^i = [O_p - s_p^i]_+, \\ \alpha_n^j = [s_n^j - O_n]_+, \end{cases} \tag{8}$$

in which $[\cdot]_+$ is the ReLU function that ensures α_p^i and α_n^j are non-negative; α_p^i and α_n^j adjust the weight so the gradient of reducing s_n is equivalent to increasing s_p. When s_n approaches zero and s_p approaches one, the gradients drop to a small value according to α_p^i and α_n^j. It intuitively emphasizes the hard examples where s_n is similar to s_p.

As for the purpose of dense 2D–3D correspondence matching, we need to emphasize the hard examples and pay less attention to the easy case. Thus, the circle loss is more suitable than the softmax cross-entropy loss.

Another problem for the 2D–3D correspondence matching is that the ground-truth poses of the objects have measurement errors that lead to the mismatch of the correspondences. To overcome the problem, we assign a neighborhood area N for each pixel. If the nodes on the 3D mesh model lie in the neighborhood area, they are regarded as positive correspondences. Each pixel has its own neighborhood area to eliminate the influence of the measurement errors of the ground-truth poses.

For every neighborhood area, we set a mask on it, and name the overall loss function the masked circle loss. The masked circle loss can be formulated as

$$\mathcal{L}_{m\_circle} = \frac{1}{u} \sum_{k=1}^{u} log[1 + \sum_{i \notin N_l} exp(\gamma \alpha_n^j (s_n^j - \Delta_n)) \sum_{j \in N_k} exp(-\gamma \alpha_p^i (s_p^i - \Delta_p))] \tag{9}$$

where u denotes the number of pixels that belong to the object in the image; $\Delta_p = 1 - m$ and $\Delta_n = m$ are the margin between the positive pairs and negative pairs; N_k denotes the set of nodes from 3D mesh model that lie in the neighborhood area of pixel k.

The final loss of the network is defined as the combination of the segmentation loss and the correspondence matching loss

$$\mathcal{L}_{all} = \mathcal{L}_{seg} + \zeta \mathcal{L}_{m\_circle} \tag{10}$$

where ζ is a hyperparameter to balance the two parts of the loss; $\mathcal{L}_{seg}$ is the pixelwise softmax cross-entropy for the semantic segmentation of the objects. After the dense correspondences are obtained, PnP with RANSAC method is used to calculate the final pose of the target.

3. Results

In this section, we use our proposed method in a real industrial application to verify the effectiveness and advantages of the proposed method. The target object in the experiment is an antenna support, as shown in Figure 3a. The target is first molded through injection; then, the mounting hole is conducted using a hole puncher. Between these two steps, the antenna support needs to be collected from the conveyor belt with a correct pose, and then put on the screw for the punch. Therefore, we train a deep learning model based on our proposed method to predict the pose of the antenna support.

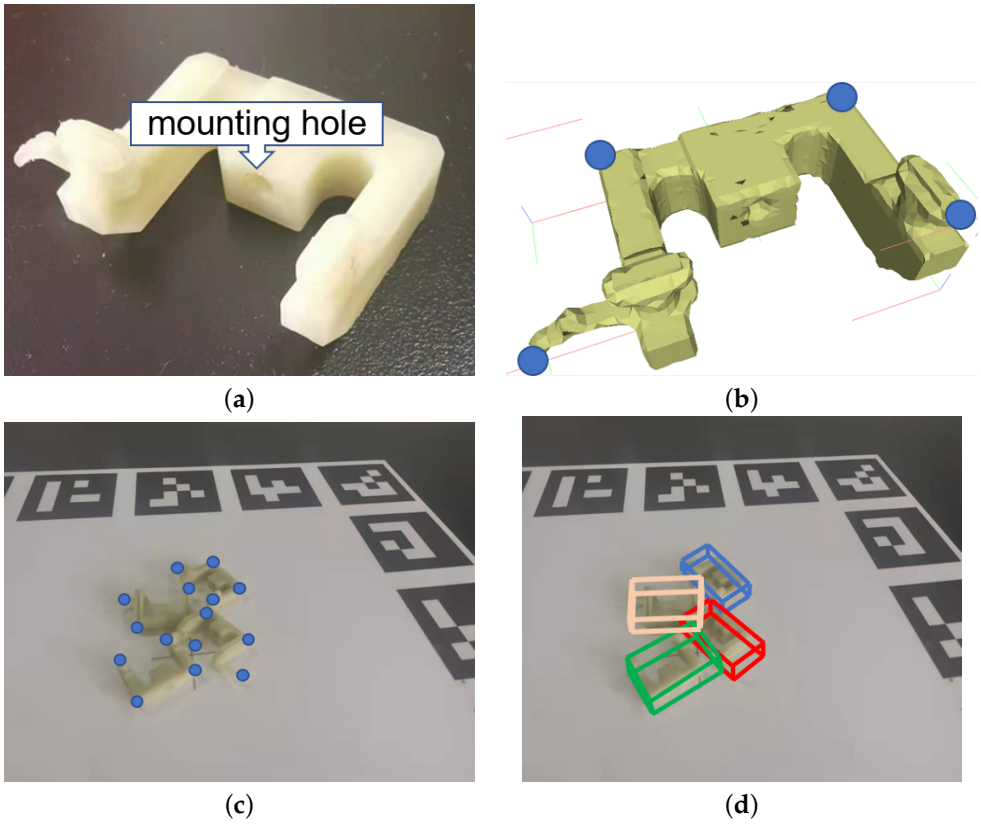


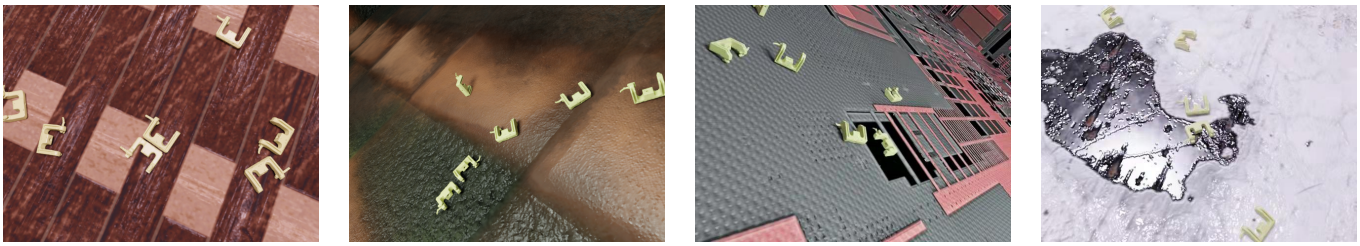
Figure 3. Preparation for the training and testing dataset of the antenna support. (**a**) The appearance of the antenna support. (**b**) Selected key points from 3D mesh model of the antenna support. (**c**) Correspondences in the 2D image. (**d**) Final ground-truth poses of the antenna supports calculated through PnP.

3.1. Implementation Details

Data collections. In order to recognize the pose of the antenna support correctly, we collected ten videos (5679 frames) of the antenna support in total as the training dataset, two videos (1096 frames) for evaluation, and another 5 videos (3105 frames) as the validation dataset. For each video, we manually selected some key points on the 3D model of the the antenna support, as shown in Figure 3b. The 2D correspondences in the first frame of the

video were then pointed out (Figure 3c) and the ground truth of the objects were calculated through PnP method, as shown in Figure 3d.

The Aruco markers were used to calculate the pose of the camera with respect to the board. The property of relevant stills among frames in the same video were used to calculate the poses of each objects with respect to the camera for the rest of the frames. To enhance the performance of the model, we further rendered 20,000 synthetic images through the BOP [34] renderer for training, as shown in Figure 4. We also added data augmentation to the original images including random cropping, resizing, 3D rotation, and color jittering during training.



(**a**) (**b**) (**c**) (**d**)

Figure 4. Examples of the rendered images. (**a**–**d**) are examples that random selected from the synthetic dataset with different view angles.

Model settings. We used EfficientDet-D2 as the object detection backbone in terms of the balance between detection accuracy and memory usage. The dimension D of the pixelwise and nodewise deep features was set to 128. The hyperparameter ζ to balance the loss of the segmentation and the loss of similarity matching was set to 0.01 through cross validation on the evaluation dataset. All the objects detected by the EfficientDet were resized to 256×256 for further calculation by the U-Net.

Training strategy. We used Pytorch [35] to implement our framework. The network was trained on two Nvidia RTX 3090 graphics cards with 24 GB RAM. The batch size was set to 16. We utilized the Adam optimizer [36] to process gradient decent of the parameters. The initial learning rate was set to 0.001 and divided by two for every twenty epochs. The model was totally trained for two hundreds of epochs and evaluated for every ten epochs. The model with the best score in the evaluation dataset was chosen as the final model for testing.

Mesh model Simplification. To reduce the memory usage of our model, we simplified the 3D mesh model of the antenna support to possess less than 8000 triangular patches and 4000 vertices through quadric edge collapse decimation in MeshLab [37]. The average of the node–pixel matching error is less than 0.5 pixel under this setting.

3.2. Evaluation Metric and Comparison

We utilized two commonly used evaluation metric to compare our proposed method with some state-of-the-art methods.

2D Projection metric. This metric computes the mean distance in the 2D image between the projections of the 3D mesh model from the estimated pose and the ground truth pose. A pose is considered correct if the distance is less than σ pixels.

ADD metric. This metric [32] computes the mean distance between two transformed model points using the estimated pose and the ground-truth pose through

$$ADD = \frac{1}{m} \sum_{x \in M} \| (Rx + \mathbf{t}) - (\tilde{R}x + \tilde{\mathbf{t}}) \| \tag{11}$$

When the distance is less than a certain percentage of the model diameter, it is claimed that the estimated pose is correct.

We compare our method with PSGMN [26], DPOD [25], and HybridPose [23]. As all of the three methods are one stage pose estimation schemes that are not able to detect multiple instances in one frame, we used the EfficientDet as the backbone for all the methods and tested these methods with the fixed size image that only contains one object per image. The results in terms of 2D Projection metric are shown in Table 1. It can be seen that our proposed method achieves better performance than the other methods, especially when the metric is stricter.

Table 1. Comparison of the proposed method with the other methods in terms of 2D Projection metric.

Methods	HybridPose	DPOD	PSGMN	Proposed Method
$\sigma = 5$	89.3	86.2	93.9	96.5
$\sigma = 4$	83.2	85.3	88.5	92.0
$\sigma = 3$	76.5	77.6	81.0	87.3
$\sigma = 2$	64.1	69.8	74.5	82.6

The results of comparison in terms of ADD metric are shown in Table 2. Our method also outperforms the other method with a large margin. As this metric focus on the measurement of the distances between the 2D–3D correspondences, our method takes advantages of the dense matching loss and shows a great improvement in the scores.

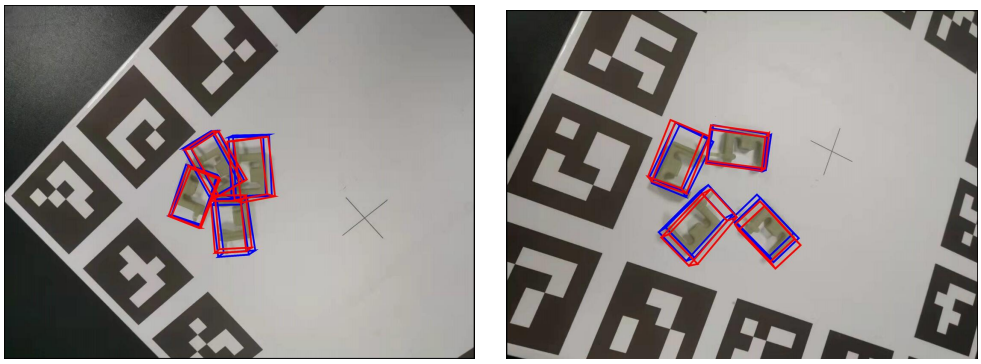
Table 2. Comparison of the proposed method with the other methods in terms of ADD metric.

Methods	HybridPose	DPOD	PSGMN	Proposed Method
0.1-ADD	72.2	71.4	76.5	84.3
0.08-ADD	64.3	65.2	72.3	79.4
0.05-ADD	51.1	53.3	57.9	74.7

Some qualitative examples of our proposed method are shown in Figure 5. It is shown that our proposed method can handle the multi-instance situation well and successfully deal with partial occlusion and light changing conditions.

(**a**) (**b**)

Figure 5. *Cont.*



(**c**) (**d**)

Figure 5. Some qualitative results of our proposed method. The bounding boxes in blue denote the ground-truth poses of the antenna support, while the bounding boxes in red denote the poses estimated using our method. (**a**–**d**) are examples that random selected from the test dataset to show the effectiveness of our proposed method. The pictures are captured from different view angles.

4. Conclusions and Future Work

In this paper, a multi-instance 6D pose estimation framework was proposed to solve the localization problem of certain objects in intelligent manufacturing. EfficientDet is used as the backbone for object detection. The detected objects in image are resized and fed into a U-Net model to further extract pixelwise deep features for 2D–3D correspondence matching. We proposed a novel, metric-based loss, named masked circle loss, for the feature matching. The results of the pose estimation of the antenna support demonstrate the effectiveness of our proposed method compared with the state-of-the-art pose estimation methods.

However, current frameworks do not consider the geometric structure and constraints among pixels; further studies will focus on the investigation of the relationships between pixels.

Author Contributions: C.W.: Conceptualization, methodology, Writing—original draft, software, and validation. L.C.: formal analysis, investigation, and supervision. S.W.: Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial supports by the National Natural Science Foundation of China (No. 52105525).

Data Availability Statement: The data in this study are available on request from the corresponding author.

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

References

1. Choi, C.; Schwarting, W.; DelPreto, J.; Rus, D. Learning object grasping for soft robot hands. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2370–2377. [CrossRef]
2. Fang, H.S.; Wang, C.; Gou, M.; Lu, C. Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11444–11453.
3. Malik, A.A.; Andersen, M.V.; Bilberg, A. Advances in machine vision for flexible feeding of assembly parts. *Procedia Manuf.* **2019**, *38*, 1228–1235. [CrossRef]
4. Yin, X.; Fan, X.; Zhu, W.; Liu, R. Synchronous AR Assembly Assistance and Monitoring System Based on Ego-Centric Vision. *Assem. Autom.* **2019**, *39*, 1–16. [CrossRef]
5. Kleeberger, K.; Landgraf, C.; Huber, M.F. Large-scale 6d object pose estimation dataset for industrial bin-picking. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 2573–2578.
6. Mahler, J.; Goldberg, K. Learning deep policies for robot bin picking by simulating robust grasping sequences. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 515–524.
7. Ouyang, Z.; Sun, X.; Chen, J.; Yue, D.; Zhang, T. Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. *IEEE Access* **2018**, *6*, 9623–9631. [CrossRef]
8. Zhang, H.; Cao, Q. Detect in RGB, optimize in edge: Accurate 6D pose estimation for texture-less industrial parts. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3486–3492.

9. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. Pvnet: Pixel-wise voting network for 6dof pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4561–4570.
10. 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 (ICCV), Venice, Italy, 22–29 October 2017; pp. 1521–1529.
11. Ke, Y.; Sukthankar, R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 27 June–2 July 2004; pp. 506–513.
12. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
13. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
14. He, Z.; Jiang, Z.; Zhao, X.; Zhang, S.; Wu, C. Sparse template-based 6-D pose estimation of metal parts using a monocular camera. *IEEE Trans. Ind. Electron.* **2019**, *67*, 390–401. [CrossRef]
15. Chen, L.; Huang, P.; Cai, J. Extracting and Matching Lines of Low-Textured Region in Close-Range Navigation for Tethered Space Robot. *IEEE Trans. Ind. Electron.* **2018**, *66*, 7131–7140. [CrossRef]
16. Tahri, O.; Chaumette, F. Complex objects pose estimation based on image moment invariants. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 18–22 April 2005; pp. 436–441.
17. Meng, C.; Li, Z.; Sun, H.; Yuan, D.; Bai, X.; Zhou, F. Satellite pose estimation via single perspective circle and line. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 3084–3095. [CrossRef]
18. Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 876–888. [CrossRef] [PubMed]
19. Muñoz, E.; Konishi, Y.; Murino, V.; Del Bue, A. Fast 6D pose estimation for texture-less objects from a single RGB image. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5623–5630.
20. He, Z.; Wu, C.; Zhang, S.; Zhao, X. Moment-Based 2.5-D Visual Servoing for Textureless Planar Part Grasping. *IEEE Trans. Ind. Electron.* **2018**, *66*, 7821–7830. [CrossRef]
21. Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; Fox, D. Deepim: Deep iterative matching for 6d pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 683–698.
22. Pavlakos, G.; Zhou, X.; Chan, A.; Derpanis, K.G.; Daniilidis, K. 6-dof object pose from semantic keypoints. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2011–2018.
23. 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 (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 431–440.
24. Labbé, Y.; Carpentier, J.; Aubry, M.; Sivic, J. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 574–591.
25. Zakharov, S.; Shugurov, I.; Ilic, S. Dpod: 6d pose object detector and refiner. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 1941–1950.
26. Wu, C.; Chen, L.; He, Z.; Jiang, J. Pseudo-Siamese Graph Matching Network for Textureless Objects' 6D Pose Estimation. *IEEE Trans. Ind. Electron.* **2021**, 1. [CrossRef]
27. Lepetit, V.; Moreno-Noguer, F.; Fua, P. Epnp: An accurate o (n) solution to the pnp problem. *Int. J. Comput. Vis.* **2009**, *81*, 155. [CrossRef]
28. 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, Korea, 5–9 November 2012; pp. 548–562.
29. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D object pose estimation using 3D object coordinates. In *Lecture Notes in Computer Science (Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2014. [CrossRef]
30. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
31. Zhang, Z.; Sabuncu, M.R. Generalized cross entropy loss for training deep neural networks with noisy labels. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 3–8 December 2018.
32. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In Proceedings of the Robotics: Science and Systems (RSS) XIV, Pittsburgh, PA, USA, 26–30 June 2018; pp. 129–136.
33. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6398–6407.
34. Hodaň, T.; Sundermeyer, M.; Drost, B.; Labbé, Y.; Brachmann, E.; Michel, F.; Rother, C.; Matas, J. BOP challenge 2020 on 6D object localization. In *European Conference on Computer Vision*; Springer: Cham, Switzerland 2020; pp. 577–594.

35. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
36. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
37. Cignoni, P.; Callieri, M.; Corsini, M.; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G. Meshlab: An open-source mesh processing tool. In Proceedings of the Eurographics Italian Chapter Conference, Salerno, Italy, 12–13 November 2008; Volume 2008, pp. 129–136.