ContourPose: Monocular 6D pose estimation method for reflective texture-less metal parts

Zaixing He, *Senior Member*, *IEEE*, Quanzhi Li, Xinyue Zhao, Jin Wang, Huarong Shen, Shuyou Zhang, and Jianrong Tan

***Abstract*—Pose estimation is an essential technology for industrial robots to perform precise gripping and assembly. The state-of-the-art deep learning-based approach uses an indirect strategy, i.e., first finding local correspondence between the 2D image and 3D model, and then using the PnP and RANSAC methods to calculate the poses of ordinary objects. However, the metal parts in industry are reflective and texture-less, making it difficult to identify distinguishable point features to establish 2D-3D correspondences. To address this problem, in this paper, we propose a novel deep learning based two-stage method for pose estimation of reflective texture-less metal parts, which accurately estimates the target pose using monocular RGB images. Since contours play an important role in both keypoints prediction and pose estimation stages, our method is named ContourPose. First, an additional contour decoder is adopted to implicitly constrain the keypoints prediction in the former stage, which improves the accuracy of the keypoints prediction. Then, the predicted contour of the previous stage is taken as geometric prior that is used to iteratively solve for the optimal pose. Experiments indicate that the proposed approach for reflective texture-less metal parts has a significant improvement over the state-of-the-art approaches.**

***Index Terms*— 6D pose estimation, reflective texture-less metal parts, deep learning.**

I. Introduction

SIX degrees of freedom (6D) pose estimation, i.e., recovering the rotation and translation of an object in 3D Euclidean space, is a key task for robotic vision. With the fast development of intelligent manufacturing, pose estimation of industrial objects has become a crucial technology for tasks such as part gripping [1], unit assembly [2], and human-machine collaboration [3].

Metal parts are the main components of machines and industrial products. Although many pose estimation methods have achieved promising results in recent years, the reflective and texture-less nature of metal parts poses a great challenge to them, due to their heavy reliance on surface features.

The existing methods to handle this task can be classified as RGB image-based or RGB-D image-based methods. In terms of ordinary textured and non-reflective objects, the RGB-D based method achieves higher accuracy and robustness because they use additional depth information to estimate the pose. However, RGB-D cameras have limited ability to accurately capture depth information for non-Lambertian materials, such as metal parts, which often produce fragmented depth images [4]. In addition, it is expensive to acquire high precision point clouds of objects with industrial-grade depth camera cameras or 3D scanners, while the cost of consumer-grade depth cameras is considerably low. Therefore, it is more appropriate to use RGB monocular cameras for 6D pose estimation of such objects.

The RGB-based methods also have limitations. Traditional pose estimation methods use various types of effective image feature descriptors, such as SIFT [5], SURF [6] and ORB [7], to establish 2D-3D correspondences based on the similarity of these descriptors. Then, the target pose can be obtained by solving the Perspective-n-Point (PnP) [8] problem. However, these methods are only applicable to objects with rich textures. Reflective texture-less metal parts barely show apparent gradient variations in images and these descriptors will not be able to find an exact correspondence. To address this issue, some methods use geometric features such as lines [9], [10], moments [11] or edges [12] as templates to retrieve the best matching result. However, these methods cannot achieve high accuracy because the matched result is chosen from a limited number of samples. Increasing the number of templates to enhance accuracy requires more computational resources and may result in matching errors when too many templates are used, consequently leading to pose estimation errors.

In recent years, deep learning has been widely used in pose estimation, for its significant improvements compared to traditional methods. For example, some methods use CNNs to regress 2D keypoints, establish 2D-3D correspondences, and then use the PnP algorithm to calculate the pose. If only sparse keypoints, e.g., several or tens of keypoints, are selected to predict, like BB8 [13], PVNet [14], and Yolo-6D [15], these

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported by the National Natural Science Foundation of China under Grants 52275514 and 52275547, and 52175032, and Zhejiang Provincial Natural Science Foundation of China under Grant LY21E050021. (Corresponding author: Xinyue Zhao)

Zaixing He, Quanzhi Li, Xinyue Zhao, Jin Wang, Shuyou Zhang, and Jianrong Tan are with the School of Mechanical Engineering, the State Key Lab of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310058, China (Emails: zaixinghe@zju.edu.cn; liquanzhi@zju.edu.cn; zhaoxinyue@zju.edu.cn; zsy@zju.edu.cn, and egi@zju.edu.cn).

Huarong Shen is with with Zhejiang Feihang Intelligent Technology Co., LTD, Huzhou 313200, China (Email:noodleshr@163.com).

methods will be referred to as pose estimation methods using sparse keypoint predictions. On the contrary, if it needs to establish pixel-wise 2D-3D correspondences of the objects in the image, like CDPN [16], DPOD [17], these methods will be instead indicated as pose estimation methods using dense keypoint predictions. Although these methods work well on public benchmarks (e.g., LINEMOD [18] and YCB-Video [19]) of ordinary objects, they have limitations in measuring metal parts. For example, PVNet predicts the location of 2D keypoints by pixel-wise voting, but the texture-less surface of metal parts does not provide effective semantic information. In addition, due to the reflective nature of metal parts, slight changes in shooting angles can lead to large color differences in the images, which can cause incorrect correspondences with dense methods such as DPOD and CDPN.

Although the current method of dense keypoints prediction achieves superior performance for pose estimation of ordinary objects, the surface of reflective and texture-less metal parts can provide little semantic information. To address this issue, we propose a novel method, called ContourPose. The proposed method can be divided into two stages.

In the first stage, the proposed method uses the object contours to implicitly constrain the prediction of keypoints. We call this stage ContourNet. In the second stage, the proposed method utilizes the contours predicted in the previous stage as geometric priors to achieve higher accuracy in solving the pose. Experiments show that it improves the accuracy of the final pose estimation compared to the widely used RANSAC + PnP [20] method.

In summary, this work has the following contributions:

- A novel pose estimation approach is proposed for reflective texture-less metal parts, which achieves superior performance on industrial metal parts.
- A new deep learning framework is proposed for the pose estimation of reflective texture-less metal parts. This framework adopts an additional contour decoder to implicitly constrain the prediction of keypoints. The proposed method does not need a pre-detector like YOLO [21] or RetinaNet [22] to locate the target object.
- A new method is proposed to iteratively solve the optimal pose using the contour as a geometric prior. The proposed method improves the accuracy of final pose estimation compared to widely used RANSAC-based methods.

II. Related Work

This section introduces recent related deep learning-based 6D pose estimation methods dealing with RGB images, which can be divided into the following four categories.

A. Direct pose regression methods

Given an input image, these methods output the 6D pose of the target object directly using CNNs, these methods are also referred to as single-stage methods in some studies. PoseNet [22] introduces a CNN architecture that regresses the camera pose directly from an RGB image, which is similar to object pose estimation. SSD-6D [24] extends the popular single-shot multibox detector (SSD) [25] object detection framework to cover the full 6D pose space by adding a translation and orientation regression module. However, SSD-6D predicts rotation by scoring the discrete views, so it needs to add the pose refinement process [26] to get better results. PoseCNN [19] decouples translation and rotation prediction to predict 6D object pose. The 3D rotation of the object is represented using unitary quaternions. However, predicting 3D rotations is difficult, because the non-linearity of the rotation space makes CNNs less generalizable. The methods mentioned above all rely on the feature extraction ability of CNNs, which cannot achieve good results on textureless industrial objects and are difficult to apply to more challenging reflective metal parts.

DenseFusion [27] designs a dense pixel-level fusion method that integrates features of RGB data and point clouds in an appropriate way to obtain good results. But these point cloud based methods are not applicable to metal parts, because it is difficult to obtain reliable point clouds for this kind of objects due to their reflective nature. EfficientPose [28] achieves end-to-end multi-object pose estimation by extending an efficient and accurate 2D object detection method EfficientDet [29]. However, pose estimation methods based on object detection usually depend on the results of prior detection, but the features of reflective metal parts are different under different lighting and viewing angles, making it difficult to train a detector for these reflective parts.

B. Template matching-based methods

These methods discretize the rotation space into sampled templates. Given an input image, they match the closest template from the database, thus bypassing the explicit labeling of pose ambiguity. Some traditional methods [18], [30] create features by hand, which is very time-consuming. STB (Spares Templated-Based method) [9] uses high-level geometric features and the correlation of straight contours to estimate the 6D pose of metal parts. But this method is sensitive to background noise. AAE [31] trains an autoencoder with synthetic data, which encodes the rotation of the object into a latent space. There, pose ambiguity is implicitly handled through similar latent codes for symmetric poses. Multipath-AAE [32] extends the AAE [31] to multiple objects. However, these methods have some limitations, such as the domain gap between real and synthetic data, and the regularity of the latent space. Although [33] introduces edges to supervise the regularity of the latent space, addressing the domain gap problem remains challenging, particularly for metal parts, due to the difficulty in rendering realistic reflections of these parts. The method [34] proposes a approach to estimate pose by using YOLO [21] and LINEMOD [18]. But the accuracy of this method is not high and can only be used for the bin picking task. GFI [35] proposes a Generative Feature-to-Image network to generate edge templates for pose estimation by matching the most consistent edges. This method achieves good results for pose estimation of metal parts but is computationally slow.

C. Render and compare methods

Another category of methods is based on rendering synthetic images of the object model under different poses and comparing

them with the observed image. These methods rely on iterative procedures that consider previous estimate of a pose p_k and predict an update Δp_{k+1} that will make current internal mesh state to better fit the observed object. For example, BB8 [13] designs an iterative refinement process step to compare the input image and the rendering of the initial pose object using another CNN, thereby improving the prediction of 2D projections. Given an initial pose estimation, DeepIM [36] designs a network to iteratively refine the pose by matching the rendered image against the observed image, and gradually improves the matching result by iteratively updating the rendered image. CosyPose [37] optimizes the pose hypotheses using a multi-view consistency module that resolves conflicts and inconsistencies and estimates the camera viewpoints. However, these methods are not suitable for reflective metal parts because they rely on renderers, which have difficulty simulating the surface color and texture changes of reflective parts under different lighting conditions.[38] This variation results in significant differences between rendered and real images, making it difficult to make effective comparisons.

D. Keypoints localization-based methods

These methods follow a two-stage pipeline for pose estimation. They establish 2D-3D correspondence using a neural network in the first stage, and then calculate the target pose using the PnP and RANSAC [39] methods in the second stage. The methods for keypoints localization can be divided into two categories, one is to select only several keypoints, e.g., several or tens of keypoint for prediction, and this type of method is classified as the sparse methods. The other category is to establish pixel-wise 2D-3D correspondences of object in an image, and this type of method is classified as the dense methods.

BB8 [13] uses CNN as a keypoints detector to output 2D coordinates of eight corner points of the object's 3D bounding box. YOLO-6D [15] uses a YOLO [21] architecture to regress keypoints. These methods output the keypoints directly through a fully connected layer, which depend on the distribution of the training images, and lead to problems like poor spatial generalization and proneness to overfitting. Inspired by the success of 2D human pose estimation [40], another category of methods output pixel-wise heatmaps of keypoints to improve accuracy. The method [41] predicts heatmaps of multiple small blocks independently and then overlays them to obtain accurate and robust keypoints. PVNet [14] regresses pixel-wise vectors point at the keypoints and uses these vectors to vote for the location of the keypoints. HybridPose [42] extends the approach of PVNet [14] by utilizing a hybrid intermediate representation to express different geometric information in the input image, including keypoints, edge vectors, and symmetry correspondences. All the above methods pre-select only a few keypoints, classified as sparse methods. These methods are designed for ordinary objects, and the selected keypoints usually do not have semantic information. It is not appropriate for metal parts with many semantic points, such as corner points and circle centers.

The methods where each pixel is used as a keypoint to generate the corresponding prediction, we will refer to this as a dense method. On some irregularly curved objects, the accuracy and robustness of dense prediction are usually higher than that of sparse prediction. Pix2Pose [40] outputs a 3D coordinate for each pixel of the object in the picture, thus establishing dense 2D-3D correspondences. DPOD [17] considers that it is difficult to directly predict 3D coordinates, so it predicts UV maps instead, and then establishes 2D-3D correspondence through UV maps. CDPN [16] decouples the prediction of rotation and translation; rotation is obtained by PnP calculation and translation is determined by 2D bounding box and depth estimation, but the method requires a pre-detector. PSGMN [44] utilizes the information from the 3D model. First extracting the features of the 3D model using a graph convolutional neural network, and then establishing the 2D-3D correspondences through a pseudo-siamese neural network[45].

However, due to the reflective nature of metal parts, the colors of the object's pixels in the image varies significantly with the lighting conditions and the shooting angle. Moreover, the surfaces of metal parts are less distinguishable than that of textured objects. For this reason, methods regressing dense keypoints might be less effective with this category of objects. To handle this problem, we first predict sparse semantic points, and then we use dense contour points for validation in the pose estimation stage to iteratively solve for the optimal pose. Our method combines the advantages of sparse and dense prediction.

III. PROPOSED APPROACH

In this paper, we propose a novel 6D pose estimation method for reflective texture-less metal parts in industry. The task of 6D pose estimation is to detect the object and estimate its 3D rotation and translation in the given image. The 6D pose represents a rigid transformation of the object coordinate system to the camera coordinate system, where R denotes the 3D rotation and t denotes the 3D translation.

Once at least three pairs of 2D-3D correspondences are determined, $[R|t]$ can be computed by solving the PnP problem. Therefore, the key step of pose estimation is to obtain the precise 2D-3D correspondences. We estimate the object pose using a two-stage pipeline: The first stage is a neural network that predicts a heatmap representing the 2D keypoints that we call "ContourNet". The second stage is an iterative pose optimization algorithm that uses the object contours as prior information. We call this stage PECP (Pose Evaluation using Contour as a Priori). We observe that the contour of objects with sharp edges is invariant to different viewing angles and lighting conditions. In the first stage, with the constraint of contour, the keypoints are constrained to be within the region of the target object in the image, which allows our method to achieve high accuracy without a pre-detector. In the second stage, the contour of the previous stage is used as a geometric prior to iteratively solve the optimal pose.

The architecture of ContourPose which consists of two stages, is shown in Figure 1.

A. *Keypoint prediction with implicit constraints on contour*

1) *Network Framework:*

The basic framework of the ContourNet is shown in Figure 1. In this stage, the input is an RGB image of an object and a modified FCN model is proposed to encode the image. To predict keypoints, the first step is to locate the position of the target object in the image. Some methods [16], [17] use the result of the pre-detector as input so that the input image contains only the target object. Other methods [14], [42], [44] use masks and then locate keypoints within the mask. We observe that for metal parts, the contour has richer semantic information compared to mask. The contour can act as a mask by restricting keypoints within the contour and also establish implicit constraints with the predicted semantic points. Therefore, we add a decoder for predicting contour to implement implicit constraints on keypoints prediction. The predictions of contour and keypoints heatmaps co-train the parameters of the encoder of this network, so the prediction of contour actually affects the prediction of keypoints as well.

Many methods such as PVNet [14] and PSGMN [44] will output both the keypoints information and the mask using only one decoder. In other words, one part of the output tensor represents the keypoints and the other part represents the mask. However, our experiments in Section Ⅳ demonstrate that this approach leads to the prediction of keypoints being completely dependent on the contour. This means that the contours have a very strong constraint on the keypoints. For example, in some extreme cases, imprecise predictions of contours can result in completely erroneous predictions of keypoints. Therefore, to decouple the contour and the keypoints, we use one decoder to output the contour and another decoder to output the heatmap of the keypoints. This structure ensures that the contour has an implicit constraint on the keypoints, and prevents the keypoints from being overly dependent on the contour.

Figure 2 provides more detailed information about ContourNet. Residual blocks [46] are utilized to convolute the input features. To down-sample the inputs and extract features at different scales, we employ max pooling and strided convolutional layers. Dilated convolution [47] can expand the receptive field and capture multi-scale contextual information. The bilinear upsampling layer is employed to increase the resolution of the low-resolution feature map to the same size as the original input image. The skip connection concatenates the feature maps of the residual blocks before down-sampling and after up-sampling to preserve the raw information of the image.

Given C classes of objects and K keypoints for each class, our network takes the $H \times W \times 3$ RGB image as input, then processes it with a fully convolutional architecture with

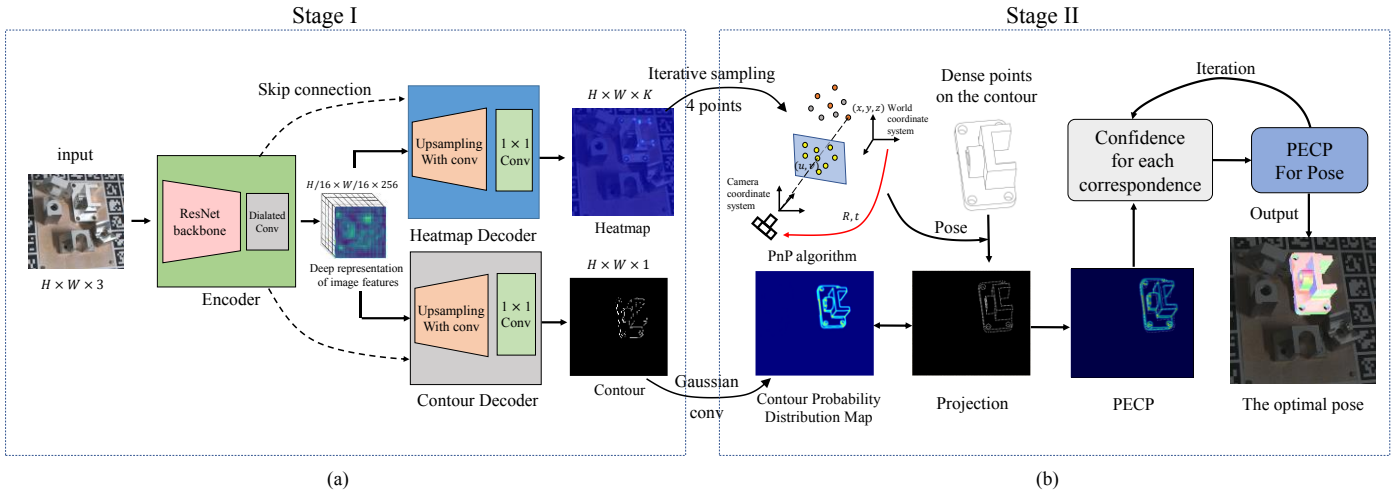


Figure. 1. Overview of ContourPose: (a) Keypoints prediction with implicit constraints on contour, named ContourNet. In this stage, we predict the keypoints and contours of the target object. (b) Pose estimation using contour as a priori. In this stage, the contour predicted in the previous stage is used as geometric priors to eliminate outlier poses, and finally output the optimal pose in the result set.

ResNet-18 [46] as the backbone. When the resolution of the feature map is downsampled to $H/16 \times W/16$, we discard the subsequent downsampling and convolution steps of ResNet-18 [46]. Instead, we add dilated convolutions to improve the receptive field of the network. After that, we repeatedly apply upsampling and convolution the feature map until its size is the same as that of the input image, i.e., $H \times W$. Finally, we apply 1×1 convolution on the feature map to obtain the contour or keypoints heatmap. We implement a skip connection between the decoder and the encoder at the same resolution of the feature map. The predictions of keypoints and contours are obtained by two independent decoders.

ContourNet can be seen as a regression model for two tasks, one is the regression of the keypoints heatmap, which we can consider as a sub-model of ContourNet, denoted by Φ, and the other is the regression of the contour of the target object, which is denoted by Λ. The heatmap follows a Gaussian distribution, and each pixel of the heatmap represents the probability of the keypoint being present at that location. The pixel with the highest probability is considered as the keypoint. To learn the heatmap, we use ℓ_2 loss. The corresponding loss function is defined as

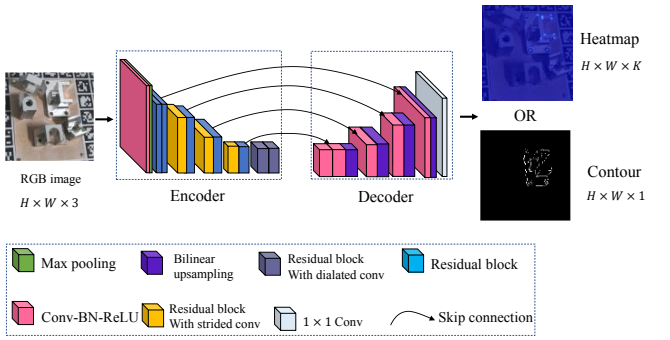


Figure. 2 ContourNet: keypoints prediction network with implicit constraints on contours. Input a picture containing the target object, after the encoding-decoding process, output the heatmap and contour of the keypoints of the object. The design of decoder for heat map and contour is identical, only the output channels are different.

$$l(w_E, w_{HD}) = \sum_{k=1}^{K} \ell_2(\tilde{\mathrm{H}}_k(w_E, w_{HD}) - \mathrm{H}_k), \quad (1)$$

$$\tilde{\mathrm{H}}_k = \Phi_{w_E, w_{HD}}(I), \quad (2)$$

where w_E represents the parameters of the encoder, w_{HD} represents the parameters of the heat map decoder, K is the number of keypoints for each object, H represents the groundtruth of the heatmap, $\tilde{\mathrm{H}}$ represents the predicted heat map, and I represents the input image.

In the contour regression task, there is a problem of positive and negative category imbalance in learning because the contour of the object accounts for a small part. For this reason, we use the weighted cross entropy [48] as the loss function. The corresponding loss function is defined as

$$l(w_E, w_{CD}) = -\beta \sum_{p \in Y^+} \log(\tilde{Y}_p = 1; w_E, w_{CD}) - (1-\beta) \sum_{p \in Y^-} \log(\tilde{Y}_p = 0; w_E, w_{CD}) \quad (3)$$

$$\tilde{Y} = \Lambda_{w_E, w_{CD}}(I) \quad (4)$$

where w_{CD} represents the parameters of the contour decoder, $\tilde{Y}$ is the predicted contour, and p denotes each pixel in the contour map. $\beta = |Y^-| / |Y^+ + Y^-|$ and $1-\beta = |Y^+| / |Y^+ + Y^-|$, where $|Y^+|$ and $|Y^-|$ denote the edge and non-edge in the contour ground truth. The final loss function of the ContourNet is defined as:

$$l(w) = l(w_E, w_{HD}) + \rho l(w_E, w_{CD}) \quad (5)$$

w is the parameter of the whole network, which contains the encoder w_E, the heatmap decoder w_{HD} and the contour decoder w_{CD}. This shows that the contour and keypoints heatmap regression tasks jointly train the network encoder, while their respective decoders do not interfere with each other. ρ is a hyper-parameter to balance the two parts of the loss.

2) Keypoints Generation:

Keypoints need to be defined based on the model of the 3D object. One simple approach is to directly select the eight corner points of the 3D bounding box as keypoints. However, most of these points are not on the objects and as such are less suited as prediction targets [13],[15]. Common methods use the Farthest Point Sampling (FPS) [49] algorithm to select several points at the farthest Euclidean distance on the model , such as PVNet [14]. However, these methods are designed for ordinary object datasets like LINEMOD [18], YCB-Video [19], because most of these objects are curved and it is difficult to manually select keypoints. In contrast, G. Pavlakos [50] proposes to select semantic points as keypoints, which is very suitable for metal parts, but the paper does not mention the specific details of how to select these points.

Algorithm 1 Keypoints generation process

Input: S_{cand}, TS
Output: S_{pred}

```
if Len(S_cand) ≤ 8 then
    S_pred ← Use_FPS_Algorithm
    Return S_pred
end if
for i in n do
    h_i ← Create_Hashtable
    S_2d ← Projection(S_cand)
    for j in Len(S_cand) do
        h_i[S_2d_j] ← S_cand_j
    end for
end for
arr ← Create_Array(Len(S_cand))
for ts_i in TS do
    S^s_2d ← Semantic_Point_Detection(ts_i)
    for j in Len(S^s_2d) do
        if h_i[S^s_2dj] ≠ null then
            C_3d ← h_i[S^s_2dj]
            idx ← Get_Index(C_3d)
            arr[idx]++
        end if
    end for
end for
S_pred ← Get_Top_K_Points(arr)
Return S_pred;
```

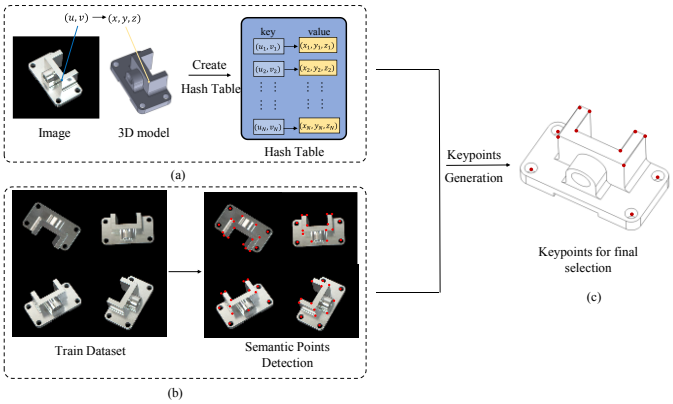


Figure. 3 Keypoints generation process. (a) Establish 2D-3D correspondence of candidate semantic points and store them in a hash table, N represents the number of candidate semantic points. (b) Iterate through the training images and record the frequency of candidate points with the semantic point detection algorithm. (c)The K points with the highest confidence are chosen as keypoints.

In fact, selecting appropriate keypoints is crucial for accurate

pose estimation. Unlike ordinary curved objects, metal parts have more semantic points such as corner points of contours, the center of circles, midpoints of lines and arcs, etc. Figure 3 illustrates an intuitive and effective algorithm proposed for generating keypoints for metal parts. For each class of parts, we traverse all its training images $TS = \{ts_i \mid i = 1,2,...,n\}$ and then build a 2D-3D hash table for each image $H = \{h_i \mid i = 1,2,...,n\}$, where n denotes the number of all training images for each part. This hash table contains the 2D-3D correspondence of candidate semantic points S_{cand}. After that, semantic point detection algorithms are applied to each image to detect semantic points, such as circle centers, corner points and midpoints. Specifically, Hough transform is used to detect circle centers and midpoints, while Shi-Tomasi corner detection [51] is used to detect corner points. The semantic points that appear most frequently in the perspective of the training images are selected. Using the established hash table, we locate the 3D points corresponding to the 2D semantic points, and finally select K points into S_{pred} as keypoints. The keypoints selected by our method are evenly dispersed on the object surface, which makes the PnP algorithm more stable. It is worth noting that some previous methods have selected a fixed number of keypoints, but since each metal part has a different shape and its semantic points are not the same, K is determined according to the part shape in our method. If the object has less than 8 semantic points, our method uses the FPS algorithm to define the keypoints. Algorithm 1 shows the flow of the keypoints generation algorithm.

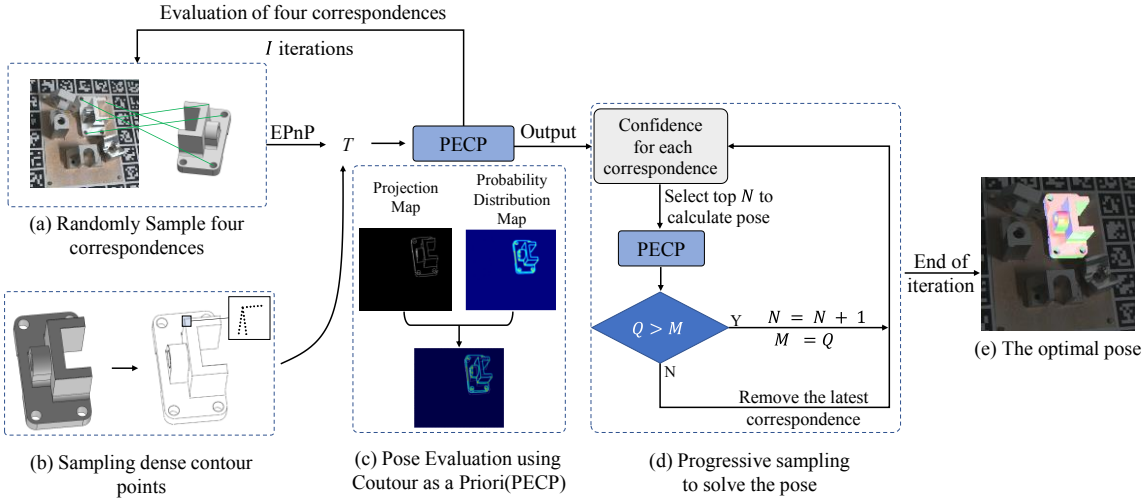


Figure. 4. Pose estimation using contours as geometric priors. (a) Randomly sample four correspondences. (b) Sampling dense contour points from 3D model. (c) Pose Evaluation using Contour as a Priori (PECP) (d) Progressive sampling to solve the pose (e) Output the optimal pose

B. *Pose estimation using contours as geometric priors*

After obtaining the keypoints in the first stage, a common approach is to iteratively solve for the optimal pose by eliminating the incorrect correspondence based on the RANSAC and PnP methods. However, since RANSAC only uses the distribution of points to distinguish between inner and outer points, the RANSAC-based method works well for dense points but may not be applicable when there are few points. To solve this issue, we propose a method that utilizes the contour as geometric priors to iteratively solve for the optimal pose.

Our approach is shown in Figure 4. First, the proposed method randomly samples four points from K points and uses the EPnP [52] algorithm to calculate a temporary pose T in each iteration.

To validate our pose estimation, we sample dense points on the contour of the 3D model, which are referred to as contour validation points S, as shown in Figure 4 (b). Specifically, S i represents the set of all 3D contour points. $S = \{s_i \mid i = 1,2,...,m\}, s_i = [x_i, y_i, z_i, 1]^T$, m represents the number of points. We calculate the image coordinate system coordinates P of these points using the image projection model.

$$P = \kappa[R \mid t]S \tag{6}$$

$$\kappa = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where P denotes the projection of the contour validation points S on the 2D image, which is referred to as the projection map. $P = \{p_i \mid i = 1,2,...,m\}$, $p_i = [u_i, v_i, 1]^T$, u, v denotes its position in the pixel coordinate system. κ denotes the intrinsic parameters of the camera, which represents the transformation between the camera coordinate system to the image coordinate system. R and t denote the 3D rotation and translation of the object coordinates, and $[R|t]$ represents the object's pose.

The accuracy of the pose is positively associated with the overlap of the projection map and the contour image from ContourNet. We refer to this algorithmic process as Pose Evaluation using Contours as a Prior (PECP). Specifically, we convolve the contours predicted in the previous stage. The convolution function is defined as

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{8}$$

$$H(u,v) = (G * \tilde{Y})(u,v) = \sum_x \sum_y G(x,y)\tilde{Y}(u-x, v-y) \tag{9}$$

We use a Gaussian convolution kernel, where the size of the convolution kernel size is 3, $\delta = 9$. The probability distribution

map is modeled using a Gaussian function, where the closer a pixel is to the contour, the higher its probability. With a given pose, the confidence score can be calculated using the PECP method:

$$\mu_T = \sum_{p \in P(T)} H(u_i(p), v_i(p)) \tag{10}$$

where μ represents the confidence score of the pose, H is the probability distribution map, and P is the projection of the contour validation points. If the confidence score is higher than the threshold that we indicate with θ, we assign this score to the four correspondences which compute the pose.

$$\lambda_i = \begin{cases} \lambda_i + \mu_T - \theta & \mu_T > \theta \\ \lambda_i & \mu_T \le \theta \end{cases}, \tag{11}$$

TABLE I Symbols and description in the pose estimation stage

Name	Description
T	Temporary pose
S	Set of 3D contour points
P	Projection of 3D contour points in 2D plane
m	Number of contour points
μ	Confidence of temporary pose ρ
θ	Threshold of correct pose
I	Iterations
M	Maximum confidence of the pose obtained by progressive sampling
Q	Confidence of current pose
λ	A set of correspondence which is ordered by PECP

where $\lambda = \{\lambda_i \mid i = 1, 2, ..., K\}$ represents each corresponding confidence. The value of θ is determined based on the number of contour validation points, and in our experiments, it was set to 1/3 m. We select four points from the correspondence set obtained in the previous stage to calculate a temporary pose T. Next, we calculate the confidence score μ_T of pose T using PECP and Eq. (11), and assigned this score to the four correspondences used to compute the pose. After I iterations, the correspondence set sorted by confidence is obtained. The number of iterations I can be calculated as follows:

$$I = \frac{\log(1 - \Pr)}{\log(1 - r^4)} \tag{12}$$

where r denotes the probability that the sampled correspondences are the correct ones. It is assumed that only four correspondences are correct in the extreme cases, $r = 4/K$. Pr is the probability of being correct at least once after I iterations. After the iterations are completed, an initial pose is calculated with the four highest confidence correspondences. PECP is used to calculate the confidence of this pose, denoted as M. Then the next highest confidence point is selected to solve for the pose, and the Eq. (11) is used to calculate its confidence, denoted as Q. If $Q > M$, the point is kept and the value of M is updated. Otherwise, this point is discarded. When all the points in the set λ are sampled, the optimal pose is obtained. All symbols in this stage can be found in Table I. Our method improves the robustness and accuracy of pose estimation compared to the RANSAC-based method.

IV. Experimental Results

In this section, we present the performance of ContourPose compared to other models for 6D pose estimation methods of metal parts and describe the ablation experiments.

A. Experimental Setup

Dataset. We demonstrate our method on two datasets: a dataset of reflective, texture-less metal parts created by our team, and the T-LESS [53] dataset. The reflective metal dataset includes ten metal parts with varying shapes, as shown in Figure 5. Each part comprises 660 training images, and each image contains only one object part. The test dataset comprises images with multiple parts stacked up randomly, including some unrelated parts, which increases the difficulty of pose estimation for the target object. The scenes in the test dataset are based on real industrial scenes and can be divided into four cases: black background, black textured background, rust background, and reflective metal background, as shown in Figure 6. Each type of background is shot under four lighting cases, including bright artificial lighting, bright natural lighting, dim lighting, and artificial lighting. Each category contains four scenes with different lighting and different parts distribution, so we tested our method on a total of 16 different scenes. The T-LESS dataset [53] contains 30 objects that lack color and texture information. Moreover, most test images exhibit obvious occlusion or stacking.

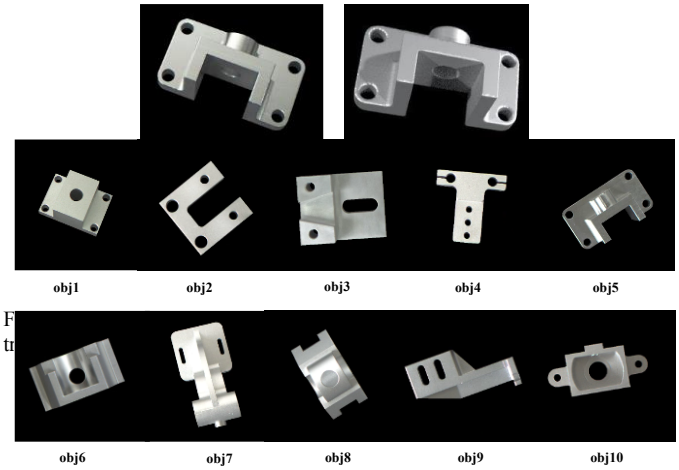


Figure. 5. Training dataset. Ten parts with different shapes are included.

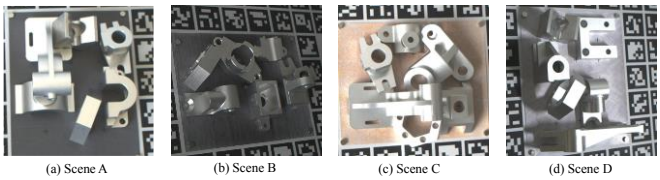


Figure. 6.Test dataset with four background scenes. (a) black background (b) black background with texture (c) simulated rust background (d) reflective metal background.

The dataset used for evaluation resembles the actual environment of an industrial site and presents a challenging task for pose estimation. Obtaining good results on these datasets would demonstrate the feasibility of our approach in real industrial settings.

Data augmentation. Since there were only 660 training images in reflective metal dataset, we rendered 6600 realistic reflective and texture-less synthetic images of the metal parts with Blender [14]. Our network was trained at a ratio of 1:10 between real and synthetic images. To create these synthetic images, we first rendered the 3D model surface to the glossy metal part in the Blender GUI, and then distributed the sampled cameras with viewpoints in a Fibonacci arrangement, with the camera pitch and azimuth angles covering the range of possible poses. After that, we arrange multiple light sources randomly and vary the light intensity randomly to simulate various scenarios of possible reflections of metal parts. The rendered image is shown in Figure 7 (b), and we render it as similar as possible to the real image in Figure 7 (a). To prevent overfitting, we employed a cut-and-paste strategy during training [54], i.e., the target object was extracted and pasted onto a random background, which was randomly sampled from SUN397 [55]. We also added more data augmentation to the original image including random cropping during training, random scaling, color dithering, and 3D rotation.

Generating Contour Groundtruth. Since our network is required to regress the contours, we need the groundtruth of the contours to calculate the loss. However, the groundtruth of the contours is not provided within the dataset. Therefore, we use OpenGL [35] to render the synthetic image corresponding to each training image, and then apply the Canny edge detection algorithm to extract the contours and obtain the ground truth.

Training strategy. We set the initial learning rate as 0.1 and halve it every 20 epochs. All models are trained for 150 epochs using the AdamW optimizer with 2 Nvidia RTX 3090 GPUs. The weight decay of the optimizer is set to 0.1. The hyper-parameter ρ in the loss function is set to 100 to balance the magnitudes of the two parts in the loss function. We train a specific model for each metal part the same as in [14], [44].

B. Metric

We evaluate our method using three standard metrics: the point-wise mean 2D projection metric [56], the average 3D distance of model points (ADD) metric [19], and the average R/t error of the valid poses. The T-LESS dataset specifies that the dataset is evaluated with the e_{VSD} metric [57]. Therefore, our experiments on the T-LESS dataset are evaluated with e_{VSD}.

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. It is generally accepted that if this distance is less than five pixels, the pose is considered correct. However, this assessment criterion is not rigorous enough for accurate attitude estimation in industrial environments. Thus, we gradually reduced the constraint from 5 pixels to 0 pixels and compared the performance variations of each method.

ADD(-S) metric. This metric computes the mean distance between two transformed model points using the estimated pose and the ground-truth pose. It is claimed that the estimated pose is correct if the distance is less than 10% of the model diameter. For symmetric objects, we use the ADD-S metric, where the mean distance is computed based on the closest point distance.

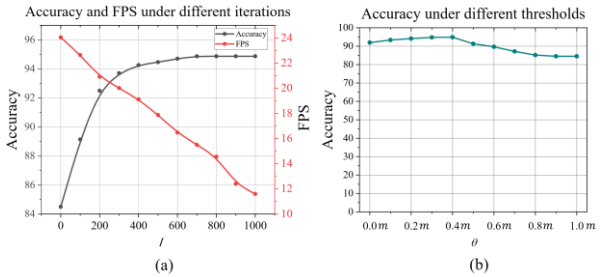


Figure. 8. Ablation studies with iterations and threshold. (a) Accuracy and computational speed under different iterations, where the value of θ is set to $1/3\ m$. (b) Accuracy under different threshold θ, where the number of iterations T is calculated by equation (12).

R/t error metric. We specify that if the pose is correct under the ADD metric, it is a valid pose, and the R/t error metric is intended to quantify the actual error in the valid pose. where R error denotes the error α, β, γ in 3D rotation, while the t error denotes the error x, y, z in 3D translation.

e_{VSD} metric. This metric is proposed by [57], and it evaluates the Visible Surface Discrepancy (VSD) between the target pose and the predicted pose. This metric was used in most experiments evaluating the T-LESS dataset. In [57] a estimated pose with e_{VSD} <0.3 is defined as a valid pose.

C. Parameter analysis

We conduct the parameter analysis experiment to investigate the effect of two parameters: the number of iterations I and the correct pose threshold θ. We gradually increase the number of iterations I from 0 to 1000 while keeping θ set to 1/3 m, and calculate the accuracy and computational speed at different iterations. The accuracy is defined by the ADD(-S) metric, and the computational speed is defined by FPS, as shown in Figure 8 (a). When the number of iterations is 0, it corresponds to no confidence ranking of the correspondence obtained in the previous stage, which may lead to errors in the initial four correspondences used to solve the pose, resulting in decreased accuracy. As the number of iterations increases, the accuracy rate gradually increases, but the computational speed of the method decreases. To balance accuracy and computational speed, we set the optimal number of iterations between 300 and 500. In fact, the number of iterations calculated by Eq. (12) falls within this range.

θ is the threshold value of the correct pose, as shown in Figure 8 (b). We evaluate the impact of different θ on accuracy, where accuracy is defined by the ADD(-S) metric. Generally, the value of θ is not sensitive to the accuracy of the final pose estimation. As θ increases, accuracy and precision first slightly increase and then decrease. This is because when θ is small, it follows from Eq. (11) that the pose with higher accuracy contributes to higher confidence. As the threshold value increases, the influence of some incorrect poses on the confidence can be avoided, thus improving accuracy and precision. However, if θ is too large, many correct poses may be judged as incorrect, ultimately leading to a decrease in accuracy and precision of the final pose estimation. As shown in Figure 8 (b), when the value of θ is less than 0.4 m, the proposed method performs well.

D. Ablation studies

1) Network analysis:

We conduct ablation studies in network design schemes to demonstrate the effectiveness of our introduction of contour decoders, as shown in Figure 9. Table II shows the performance of different networks for different inputs in terms of ADD(-S) metrics, using obj5 as an example. In our initial design, as shown in Figure 9 (a), only the semantic points of the object are regressed by the heatmap. However, since the test image contains multiple metal parts, each of which also has many semantic points, it needs a pre-detector to achieve good results.

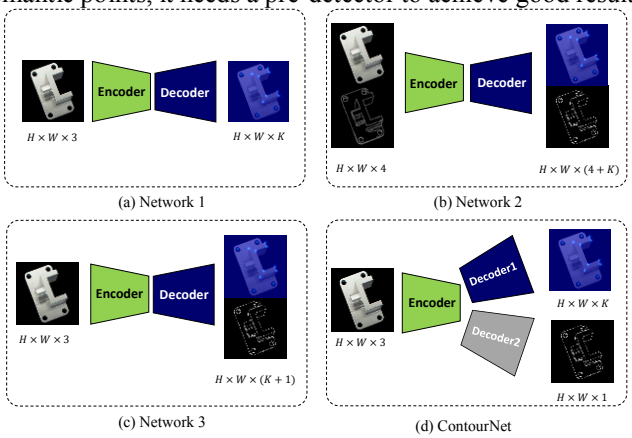


Figure. 9. Ablation study of network design. (a) Network1. Input an image and output the heatmap of semantic points. (b)Network2. Input a four-channel image, the first three channels is RGB channels, and the last channel is the contour detected with Canny. Output heat map and noise reduced contour. (c)Network3. Input an RGB image. Output a feature map with $K+1$ channels, K represents the number of keypoints, and the last channel is the predicted contour. (d) Proposed method. Input an RGB image, output the heatmap by one decoder and the contour by another decoder.

A common practice is to localize the target object by introducing the mask. However, for metal parts, the contour provides more advanced mask with more semantic information, so we chose to use the contour to constrain the prediction region of keypoints. As shown in Figure 9 (b), the input is a four-channel image, where the first three channels are RGB channels, and the fourth channel is the contour obtained by Canny [58] detection. Since Canny does not detect contours well, we want the network to implement a denoising autoencoder function, i.e., input a mutilated noisy contour and output a complete contour. Through experiments, we found that this approach leads to a "lazy" network because the contour and semantic points are strongly associated, causing the network to only focus on the input contour and ignore the RGB images. When the accuracy of the input contour is high, the output keypoints can achieve high accuracy, and once the input contour is less effective, the performance of the method is degraded significantly.

The third scheme is to implement the prediction of contours with a network, as shown in Figure 9 (c). In this scheme, the contour and keypoints co-train the model, and we expect to establish the relationship between the keypoint prediction and the contour in this way, thus achieving a constraint of the contour on the keypoints prediction. However, experiments show that this scheme creates a strong constraint between contours and keypoints. If the contour prediction is accurate, the keypoints are also accurate. But in extreme cases where the contour prediction accuracy decreases, it results in decreased accuracy of keypoints prediction.

To decouple the contour and keypoints, the proposed network architecture uses a separate decoder to predict the contour. The predictions of contours and heatmaps train the encoder together, but each trains its corresponding decoder. This scheme makes the contour implicitly constrain the prediction of keypoints instead of guiding it directly. When the contours are entirely incorrect, the prediction of keypoints degrades to the network1. As shown in Table II, the ContourNet architecture significantly improves the accuracy and robustness of pose estimation.

TABLE II COMPARING THE PERFORMANCE OF DIFFERENT NETWORK SCHEMES FOR INPUT IMAGES IN TERMS OF ADD (-S) METRIC

Method	GT Mask	GT BBOX	Yolov3	w/o pre-detector
Network1	91.35	80.53	77.16	64.50
Network2	86.79	76.49	54.96	24.04
Network3	98.55	88.55	72.11	48.31
ContourNet	**99.59**	**94.62**	**92.22**	**90.79**

2) Comparison with the RANSAC-based pose calculation:

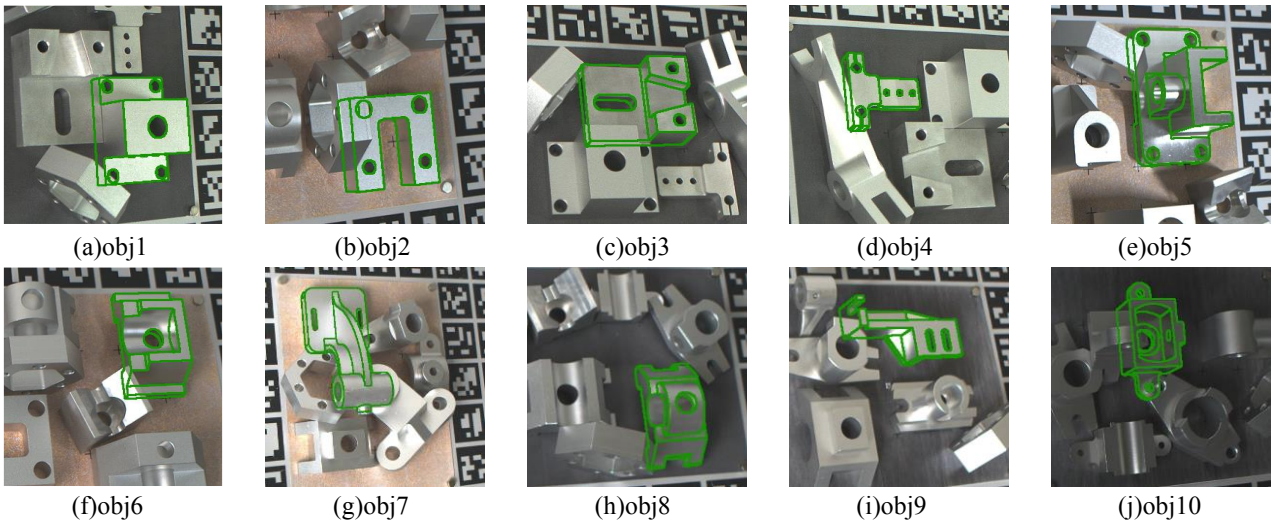
To demonstrate the effectiveness of PECP in solving the optimal pose, we compare it with RANSAC-based methods. Table Ⅲ presents the results of the comparison in terms of different metrics. The proposed method outperforms the RANSAC-based method in both accuracy and precision.

While RANSAC relies only on the point distribution to eliminate outlier correspondences, our method uses the contour as a geometric prior to compute the confidence score of each correspondence, and then iteratively solve for the optimal pose with progressive sampling. To investigate the robustness of our method against outlier correspondences, we conducted an experiment where we systematically introduced outlier correspondences and computed the pose. we altered the location of the predicted keypoint so that it became an outlier, and then solved the pose using the RANSAC-based and PECP-based methods, respectively. We evaluated the results using the ADD metric.

As shown in Figure 10, we used obj5 as an example and sequentially increased the number of error points. As the proportion of errors gradually increased, PECP demonstrated higher accuracy than RANSAC-based methods. Particularly, when the number of error points exceeded half of all points, the performance of RANSAC dropped sharply, whereas our method still achieved good results.

E. *Comparison with the state-of-the-art methods*

We compared the 2D projection metric with the other methods as shown in Figure 11 Our method is significantly better than other methods. In addition to that, our method shows more performance when the constraints are more rigorous. With a pixel threshold of only 2 pixels, the proposed method achieves 87.46% in the 2D projection metric. GFI [35] achieves better results when the pixel threshold is less than 2 pixels. The method achieves high accuracy by matching the most similar templates.



(a)obj1 (b)obj2 (c)obj3 (d)obj4 (e)obj5

(f)obj6 (g)obj7 (h)obj8 (i)obj9 (j)obj10

Figure. 12. Qualitative results for reflective metal parts dataset, where the green edge line is a visualization of our predicted pose. We place each part in 2-3 scenes, and each scene also contains 1-3 target parts.

TERMS OF DIFFERENT METRICS.

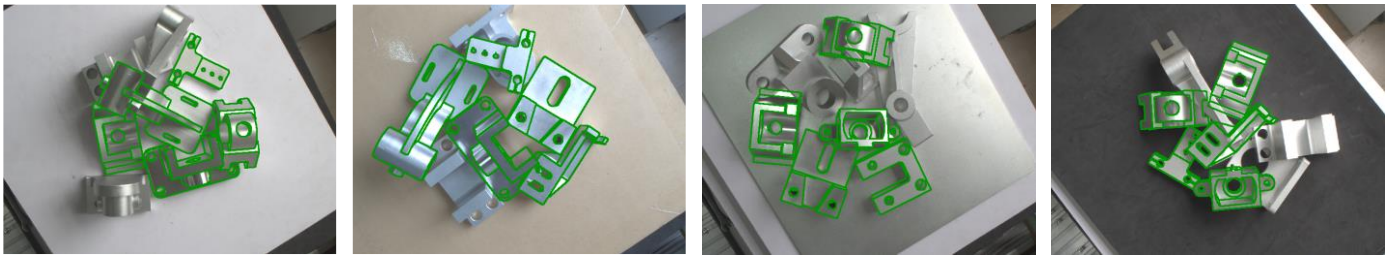
Method	ContourNet + RANSAC	ContourNet+ PECP
2D Projection	95.81%	**97.06%**
ADD	90.15%	**94.1%**
R	2.16 °	**0.84 °**
t	5.92 mm	**4.44 mm**

We compared our method with BB8 [13], AAE [31], STB [9], PSGMN [44], and GFI [35] on the reflective metal parts dataset, while STB, PSGMN, and GFI were designed specifically for pose estimation on metal parts. BB8 improves the end branch of VGG [59] by directly outputting the 2D coordinates of the eight corner points of the 3D bounding box. We adopted the optimization method of BB8, which regresses the keypoints by heatmap. AAE encodes object rotation into the latent space using an augmented autoencoder, which only requires synthetic data for training. This method can effectively learn the 6D pose of objects under different environmental backgrounds and occlusion conditions. Given an image, AAE only predicts the object's rotation, while the object's translation is estimated using a 2D bounding box. PSGMN is a dense matching method that establishes the pixel-wise correspondence between 3D models and 2D images using graph neural networks, and then calculates the pose by RANSAC and PnP algorithms. STB and GFI were specifically designed for pose estimation of metal parts. STB uses high-level geometric features and linear contours to represent metal part templates requiring only sparse templates to obtain highly accurate poses. GFI proposed a feature-image generation model. Given a feature representing a pose, it can generate an image of the object in the exact same pose. This method avoids extracting features from reflective metal parts.

TABLE IV COMPARISON WITH DIFFERENT METHODS ON REFLECTIVE METAL PARTS DATASET USING THE ADD(-S) METRIC

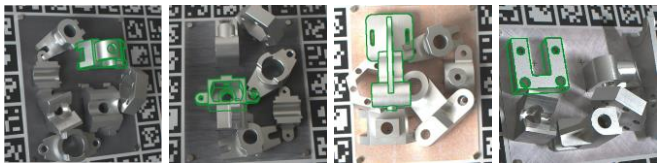
Methods	BB8	AAE	STB	PSGMN	GFI	Ours
Obj 1	71.93	76.96	64.21	94.23	95.32	**100.00**
Obj 2	43.49	76.43	66.49	70.86	96.77	**97.54**
Obj 3	43.44	84.32	54.65	82.45	92.16	**95.35**
Obj 4	26.72	32.42	48.90	74.95	**91.49**	88.14
Obj 5	48.80	64.77	36.46	79.57	87.85	**90.70**
Obj 6	68.42	54.32	62.36	84.34	85.03	**96.71**
Obj 7	18.73	49.33	29.45	74.11	76.31	**91.82**
Obj 8	19.21	72.12	45.49	75.89	84.22	**95.31**
Obj 9	44.95	67.09	61.26	79.94	89.92	**93.50**
Obj 10	39.18	71.32	59.23	87.30	85.11	**92.30**
Average	42.60	64.91	52.85	80.36	88.42	**94.14**

Table IV summarizes the comparison with other methods using the ADD(-S) metric. BB8 [13] estimates the pose by regressing the eight corner points of the object bbox, which can easily result in incorrect keypoints due to complex backgrounds or interference from other parts. AAE [31] only encodes rotation information of the object and will suffer from decreased accuracy when dealing with complex parts. STB[9] performs well for individual parts but can encounter errors due to template matching issues when there are interferences from other parts in the dataset, as is the case with similar geometric features across all parts. PSGMN [44] matches the object pixels in the image with the nodes of the 3D model However, in the test dataset, there are multiple metal parts with similar surface



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

Figure.15. Pose estimation of target parts in complex scenes. Each scene contains five target parts and several unrelated parts. (a) is a white background, (b) is a wooden background (c) is a metal background (d) is a black background. Our method can accurately estimate the pose of each target part. For the convenience of display, we combine the pose estimation results of multiple parts in one figure



(a)Scene A (b)Scene B (c)Scene C (d)Scene D

Figure. 13. Qualitative results for different scenes. The scenes are divided into A, B, C, and D according to the different backgrounds. Each scene has a different lighting situation

features, which can cause semantic segmentation errors and incorrect matching results. Therefore, PSGMN requires a pre-detector to achieve accurate pose estimation. GFI [35] has to do edge detection on the target object in the image, then generate the image that is most similar to the edge detection result. After that, this method can get the pose by the feature corresponding to the image. The method relies on the result of edge detection, and the performance of the method will be degraded when the edges are inaccurate. STB[9] and GFI [35] rely on contours to estimate the poses, and the accuracy of the pose estimation decreases when the contours are not clear or accurate. In contrast, our method only takes the contour as an implicit constraint, establishes 2D-3D correspondence by predicting semantic points, and then solves the poses with PnP. The contour only serves to improve the accuracy of the pose estimation, not to determine it. The comparison shows that our approach is superior to the state-of-the-art method. Some qualitative results of pose estimation are shown in Figure 12.

Table Ⅴ shows the performance of each method in different scenes in terms of ADD(-S) metric. Scene A is a black background, Scene B is a black textured background, Scene C is a rusted background, and Scene D is a reflective metallic background. Our method demonstrates robustness in various scenes, including those with different lighting conditions. Figure 13 provides some qualitative results of pose estimation in different scenes.

Table Ⅵ summarizes the comparison of other methods in terms of the *R/t* error metric. We only considered valid poses, i.e., those that were correct under the ADD(-S) metric. The R error represents the 3D rotational error of the object, while the t error represents the 3D translation error. Our method achieved an average error of less than 1mm in 2D translation and only about 1° for Euler angles α, β, and γ. This shows that our method can achieve high accuracy for reflective texture-less

TABLE Ⅵ COMPARISON OF DIFFERENT METHODS USING THE *R/T* ERROR METRIC IN VALID POSES

Method	x/mm	y/mm	z/mm	α/°	β/°	γ/°
BB8	1.87	1.51	7.61	3.10	2.89	0.85
AAE	1.48	1.1	7.92	5.25	4.99	2.23
STB	1.47	0.94	7.49	1.11	1.44	0.85
PSGMN	2.50	1.97	8.45	4.00	3.74	1.52
GFI	2.47	1.94	6.59	1.85	1.91	0.97
Ours	**0.71**	**0.79**	**4.31**	**1.00**	**1.10**	**0.42**

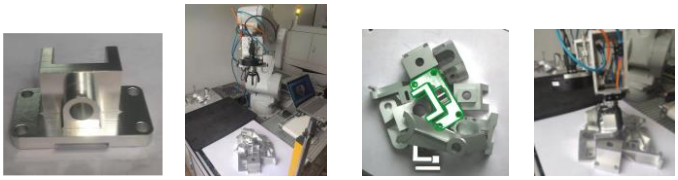
metal parts, which can be implemented for practical applications in industrial scenes.

To evaluate the effectiveness of our proposed method on general industrial objects, we conducted experiments on the T-LESS [53] dataset. We used the pre-detector provided by Pix2Pose [43] to identify the target object and trained our model using real and synthetic images, as was done in CosyPose [37], SurfEmb [60] and CDPN [16]. We then estimated the 6D pose

Figure. 14. Qualitative results for T-LESS dataset, where the green edge line is a visualization of our predicted pose

TABLE Ⅶ COMPARISON WITH DIFFERENT METHODS ON THE T-LESS DATASET

Method	e_{VSD} < 0.3
AAE	18.2
Pix2Pose	34.4
CDPN	47.8
CosyPose	72.8
SurfEmb	**77.0**
Ours	72.4



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

Figure.16. Illustration of grasping task of reflective metal parts in complex scene. (a) The metal part to grasp. (b) Initial position of the industrial robot. (c) Image taken by the camera at the initial position. The proposed method successfully estimates the 6D pose of the target part. (d) Successful grasping of the metal part.

of the target object using our method. Figure 14 shows the qualitative results of our method on the T-LESS dataset, demonstrating its applicability even for objects without sharp edges. Table Ⅶ provides a summary comparison of our method with other methods on the T-LESS dataset, where all methods used RGB images as input. Our method is designed for reflective metal parts with sharp contours that can significantly improve the accuracy of the proposed method. While our method did not achieve the best performance on the T-LESS dataset, the experimental results confirm that it can handle general industrial objects.

F. Pose Estimation for metal part grasping

To further demonstrate the effectiveness of our proposed method, we design some more challenging scenes. These scenes are prevalent with parts stacked and placed in different backgrounds, as well as interference from other similar parts. As shown in Figure 15, our method can accurately estimate the pose of the target part in such complex scenes.

Moreover, we implemented our method in a real grasping task for reflective texture-less metal parts, as illustrated in Figure 16. The grasping task was performed using a 6-axis industrial robot with a CCD camera mounted on its end-effector, and the intrinsic parameters of the camera and hand-eye matrix are well-calibrated. We place several parts in a plane with some stacking as the target. The robot's end-effector is first moved to the initial position, where the picture is taken and the pose of the target part is calculated, and then it is moved to the corresponding position for grasping. Our proposed pose estimation method achieves a real-time performance of 20 FPS on a single Nvidia RTX 3090 GPU.

V. Conclusions and Future Work

In this paper, we proposed a 6D pose estimation method of reflective texture-less metal parts with contour constraints and geometric prior. The ContourPose consists of two stages, namely the keypoints prediction stage and the pose estimation stage. We achieve an implicit constraint on the keypoints prediction by adding a decoder to predict the contour in the first stage. In the pose estimation stage, we utilized the predicted contours from the previous stage as a priori to iteratively solve the optimal pose, which greatly enhances the accuracy and precision of the pose. Our proposed method for reflective texture-less metal parts outperforms current state-of-the-art pose estimation methods. Furthermore, we demonstrated the effectiveness of our method in an industrial application for grasping tasks. The dataset in this paper can be found on https://github.com/lqz123/Reflective-Metal-Dataset.

Our proposed method trains a specific network to estimate a single object similar to PVNet[14] , CDPN [16] and PSGMN [44]. However, unlike these methods, our method does not require a pre-detector if there is only one target object in the scene. Currently, for multiple identical objects in one scene, our method still relies on a pre-detector to locate each instance. In future work, we plan to leverage topological constraints to group the keypoints of each instance and enable detection of multiple objects of the same class.

References

[1] Z. He, C. Wu, S. Zhang, and X. Zhao, "Moment-based 2.5-D visual servoing for textureless planar part grasping," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7821–7830, 2018.

[2] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, 2018.

[3] Z. He, W. Feng, X. Zhao, and Y. Lv, "6D pose estimation of objects: Recent technologies and challenges," *Appl. Sci.*, vol. 11, no. 1, p. 228, 2020.

[4] H. Zhang and Q. Cao, "Detect in RGB, Optimize in Edge: Accurate 6D Pose Estimation for Texture-less Industrial Parts," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, May 2019, pp. 3486–3492. doi: 10.1109/ICRA.2019.8794330.

[5] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, 2004, p. II–II.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[8] S. Li, C. Xu, and M. Xie, "A robust O (n) solution to the perspective-n-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1444–1450, 2012.

[9] Z. He, Z. Jiang, X. Zhao, S. Zhang, and C. Wu, "Sparse template-based 6-D pose estimation of metal parts using a monocular camera," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 390–401, 2019.

[10] L. Chen, P. Huang, and J. Cai, "Extracting and matching lines of low-textured region in close-range navigation for tethered space robot," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7131–7140, 2018.

[11] O. Tahri and F. Chaumette, "Complex objects pose estimation based on image moment invariants," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 436–441.

[12] S. Hinterstoisser *et al.*, "Gradient response maps for real-time detection of textureless objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 876–888, 2011.

[13] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3828–3836.

[14] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[15] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 292–301. doi: 10.1109/CVPR.2018.00038.

[16] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 7677–7686. doi: 10.1109/ICCV.2019.00777.

[17] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D Pose Object Detector and Refiner," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 1941–1950. doi: 10.1109/ICCV.2019.00203.

[18] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*, Springer, 2012, pp. 548–562.

[19] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *ArXiv171100199 Cs*, May 2018, Accessed: Feb. 19, 2022. [Online]. Available: http://arxiv.org/abs/1711.00199

[20] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PnP problem with algebraic outlier rejection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 501–508.

[21] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv Prepr. ArXiv180402767*, 2018.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[23] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[24] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[25] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[26] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings third international conference on 3-D digital imaging and modeling*, IEEE, 2001, pp. 145–152.

[27] C. Wang *et al.*, "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 3338–3347. doi: 10.1109/CVPR.2019.00346.

[28] Y. Bukschat and M. Vetter, "Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach," *ArXiv Prepr. ArXiv201104307*, 2020.

[29] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 10778–10787. doi: 10.1109/CVPR42600.2020.01079.

[30] M. Imperoli and A. Pretto, "D $$^${$2$}$ $$ CO: Fast and Robust Registration of 3D Textureless Objects Using the Directional Chamfer Distance," in *International conference on computer vision systems*, Springer, 2015, pp. 316–328.

[31] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 699–715.

[32] M. Sundermeyer *et al.*, "Multi-path learning for object pose estimation across domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13916–13925.

[33] Y. Wen, H. Pan, L. Yang, and W. Wang, "Edge enhanced implicit orientation learning with geometric prior for 6D pose estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4931–4938, 2020.

[34] A. Blank *et al.*, "6DoF Pose-Estimation Pipeline for Texture-less Industrial Components in Bin Picking Applications," in *2019 European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic: IEEE, Sep. 2019, pp. 1–7. doi: 10.1109/ECMR.2019.8870920.

[35] Z. He, M. Wu, X. Zhao, S. Zhang, and J. Tan, "A Generative Feature-to-Image Robotic Vision Framework for 6D Pose Measurement of Metal Parts," *IEEEASME Trans. Mechatron.*, pp. 1–12, 2021, doi: 10.1109/TMECH.2021.3109344.

[36] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep Iterative Matching for 6D Pose Estimation," p. 16.

[37] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision*, Springer, 2020, pp. 574–591.

[38] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–40, 2022.

[39] M. Zuliani, "RANSAC for Dummies," *Vis. Res. Lab Univ. Calif. St. Barbara*, 2009.

[40] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*, Springer, 2016, pp. 483–499.
[41] M. Oberweger, M. Rad, and V. Lepetit, "Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation," *ArXiv180403959 Cs*, Jul. 2018, Accessed: Aug. 11, 2021. [Online]. Available: http://arxiv.org/abs/1804.03959
[42] C. Song, J. Song, and Q. Huang, "Hybridpose: 6d object pose estimation under hybrid representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 431–440.
[43] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation," *2019 IEEECVF Int. Conf. Comput. Vis. ICCV*, pp. 7667–7676, Oct. 2019, doi: 10.1109/ICCV.2019.00776.
[44] C. Wu, L. Chen, Z. He, and J. Jiang, "Pseudo-Siamese Graph Matching Network for Textureless Objects' 6D Pose Estimation," *IEEE Trans. Ind. Electron.*, pp. 1–1, 2021, doi: 10.1109/TIE.2021.3070501.
[45] D. Chicco, "Siamese neural networks: An overview," *Artif. Neural Netw.*, pp. 73–94, 2021.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[47] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *ArXiv151107122 Cs*, Apr. 2016, Accessed: Aug. 11, 2021. [Online]. Available: http://arxiv.org/abs/1511.07122
[48] X. Soria, E. Riba, and A. D. Sappa, "Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection," *ArXiv190901955 Cs*, Feb. 2020, Accessed: Jan. 17, 2022. [Online]. Available: http://arxiv.org/abs/1909.01955
[49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
[50] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 2011–2018.
[51] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 1994, pp. 593–600. doi: 10.1109/CVPR.1994.323794.
[52] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.
[53] T. Hodan, P. Haluza, Š. Obdrzálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects".
[54] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1301–1310.
[55] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 3485–3492.
[56] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, and S. Gumhold, "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*, 2016, pp. 3364–3372.
[57] T. Hodan *et al.*, "Bop: Benchmark for 6d object pose estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.
[58] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved CANNY edge detection algorithm," in *2014 IEEE international conference on mechatronics and automation*, IEEE, 2014, pp. 577–582.
[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv Prepr. ArXiv14091556*, 2014.
[60] R. L. Haugaard and A. G. Buch, "SurfEmb: Dense and Continuous Correspondence Distributions for Object Pose Estimation with Learnt Surface Embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6749–6758.

Zaixing He received his B.Sc. and M.Sc. degrees in Mechanical Engineering from Zhejiang University, China in 2006 and 2008, respectively. He received his Ph. D. degree in 2012 from the Graduate School of Information Science and Technology, Hokkaido University, Japan.
He is currently an associate professor in the School of Mechanical Engineering, Zhejiang University. His research interests include robotic vision, Visual intelligence of manufacturing equipment, and optical-based measurement. He has published over 40 peer reviewed papers in prestigious journals such as IEEE TRO, TIE, TII, TIM, IEEE/ASME TMech, Pattern Recognition, Neurocumputing. He served as technical committee members of IEEE Consumer Technology Society and Intelligent Transportation Systems Society, Lead Guest Editor or Guest Editor of several journals including IEEE TCE and Mathematics, Program Chair or TPC of more than 10 international conferences. He is a senior member of IEEE.

Yue Chao received the B.S. degree in mechanical engineering from Shandong University, Shandong, China, in 2020. He is currently pursuing the M.S. degree in mechanical engineering with Zhejiang University, Hangzhou, China. His research interests include structured robotic vision, deep learning.

Mengtian Wu was born in Ningbo, Zhejiang province, China. He received the B.Eng. degree from Sichuan University, Sichuan, China, IN 2014. He is pursuing the Ph.D. in the School of Mechanical Engineering, Zhejiang University. His research interests include Computer Vision, deep learning.

Yilong Hu was born in Shaoxing, Zhejiang province, China. He received the B.S. degree in mechanical engineering from Zhejiang University, Zhejiang, China, in 2022. He is currently pursuing the M.S. degree in mechanical engineering with Zhejiang University, Hangzhou, China. His research interests include Computer Vision, deep learning.

Xinyue Zhao received her M.S. degree in Mechanical Engineering from Zhejiang University, China in 2008, and her Ph.D degree in Graduate School of Information Science and Technology from Hokkaido University, Japan in 2012. She is currently an associate professor in the School of Mechanical Engineering, Zhejiang University, China. Her research interests include machine vision and image processing. She has published nearly 50 peer reviewed journal papers.

Shuyou Zhang received his M.S. degree in Mechanical Engineering and the Ph.D. degree in State Key Lab. Of CAD&CG from Zhejiang University, China, in 1991 and 1999, respectively.

He is currently a professor in the Department of Mechanical Engineering, Zhejiang University, China. His research interests include product digital design, design and stimulation for complex equipments, and engineering and computer graphics.

Jianrong Tan received the B.S. degree in mechanical engineer and electronic engineering from The Open University of China, Beijing, China, in 1982, the M.S. degree in engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985, and the Ph.D. degree in mathematics from Zhejiang University, Hangzhou, China, in 1987.

He is an academician of China Engineering Academy, and is currently a professor at State Key Laboratory of CAD&CG, Zhejiang University. His main research interests include virtual-reality-based simulation, machine learning, CAX and robotics.