

Article

A 6D Pose Estimation for Robotic Bin-Picking Using Point-Pair Features with Curvature (Cur-PPF)

Xining Cui, Menghui Yu, Linqigao Wu and Shiqian Wu * 

Institute of Robotics and Intelligent Systems, School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China; cuixining@wust.edu.cn (X.C.); yumenghui_hui@163.com (M.Y.); wulin_a@126.com (L.W.)

* Correspondence: shiqian.wu@wust.edu.cn; Tel.: +86-136-2711-4410

Abstract: Pose estimation is a particularly important link in the task of robotic bin-picking. Its purpose is to obtain the 6D pose (3D position and 3D posture) of the target object. In real bin-picking scenarios, noise, overlap, and occlusion affect accuracy of pose estimation and lead to failure in robot grasping. In this paper, a new point-pair feature (PPF) descriptor is proposed, in which curvature information of point-pairs is introduced to strengthen feature description, and improves the point cloud matching rate. The proposed method also introduces an effective point cloud preprocessing, which extracts candidate targets in complex scenarios, and, thus, improves the overall computational efficiency. By combining with the curvature distribution, a weighted voting scheme is presented to further improve the accuracy of pose estimation. The experimental results performed on public data set and real scenarios show that the accuracy of the proposed method is much higher than that of the existing PPF method, and it is more efficient than the PPF method. The proposed method can be used for robotic bin-picking in real industrial scenarios.

Keywords: pose estimation; robotic bin-picking; candidate targets; curvature information; weighted voting



Citation: Cui, X.; Yu, M.; Wu, L.; Wu, S. A 6D Pose Estimation for Robotic Bin-Picking Using Point-Pair Features with Curvature (Cur-PPF). *Sensors* **2022**, *22*, 1805. <https://doi.org/10.3390/s22051805>

Academic Editors: Yuansong Qiao and Seamus Gordon

Received: 20 January 2022

Accepted: 22 February 2022

Published: 24 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bin-picking is a common scene in the industry, aiming to take out objects placed in disorder by robotic arms. There are different degrees of overlap and occlusion interference with the detection and perception of objects, yielding the failure of the robotic grasping task [1]. Bin-picking is challenging, attracting many domestic and foreign scholars [2–4]. The key of bin-picking is to calculate the pose of the best picking point of the target object [5], namely, 6D pose estimation. According to the current research on pose estimation, it can be divided into correspondence method, template-based method, voting-based method, and deep learning-based method [6].

The method to find the relationship between input data and known point cloud model is called the correspondence method. According to the type of input data, the method can be divided into 2D–3D correspondence and 3D–3D correspondence [7]. The 2D–3D corresponding method is often used for objects with rich textures. The point cloud model is projected from multiple angles, and the relationship between the template image and the RGB image of the target object in a single angle, is found through feature points. Then, the Perspective-n-Point (PnP) algorithm is used to restore the pose of the current perspective. For example, Hu et al. [8] introduced a segmentation driven network framework for 6D pose estimation. This method predicts the local pose through the 2D key point position of objects in the scenario, thereby generating a set of reliable 3D to 2D correspondences, and then uses the PnP algorithm to calculate the accurate pose of each object. This method can maintain robustness in the presence of overlap among objects, but it is not suitable for untextured objects. In the 3D–3D corresponding method, the acquired depth image is converted into a 3D point cloud, and then the relationship between the two point clouds is solved

through the point cloud descriptor. The common point cloud descriptors include Fast Point Feature Histogram (FPFH) [9], Signature of Histogram of Orientation (SHOT) [10], Oriented FAST and Rotated BRIEF (ORB) [11], and so on. For example, Xue et al. [12] proposed an improved Iterative Closest Point (ICP) [13] for point cloud registration. In this method, the initial registration point-pairs are selected by FPFH, then Principal Component Analysis (PCA) algorithm is used for coarse registration, and finally, the improved ICP algorithm is used for fine registration. This method takes a long time to load the point cloud, and registration parameters need to be continuously optimized, so that it is not suitable for industrial pipeline tasks.

Template-based methods are usually used for weakly textured or untextured objects, which are difficult to extract feature points. The principle of the method is to select the most similar template with the object in the scene, and regard the template pose as one of the objects. Usually, the template is the complete point cloud of the object, and the pose calculation is formulated as a local registration problem, i.e., align the input single-view point cloud with the complete template point cloud. For example, Sarode et al. [14] proposed a new point cloud registration network (PCRNet). This method is based on the prior information of the point cloud shape, and the transformation matrix is obtained by comparing the global features of the template point cloud with the target point cloud. This method is robust to point cloud noise and the initial deviation of the pose, but is not suitable for occlusion scenarios. Hence, this method cannot be used for bin-picking.

Voting-based methods are based on each part of target objects to be able to vote on the overall output [15]. Local voting usually refers to the vote of each pixel or 3D point to obtain the final 6D poses of target objects. Such methods are often used in scenarios where there are no texture and overlapping occlusions among objects, which are suitable for robotic arms to perform bin-picking tasks. Methods based on voting strategies can be divided into indirect voting and direct voting. The former is to obtain a predefined feature point by voting for each pixel or 3D point, and obtain a 6D pose according to the 2D–3D or 3D–3D correspondence. For example, Peng et al. [16] proposed to use Pixel-wise Voting Network (PVNet) to return unit vectors to key points, then used RANdom SAmple Consensus (RANSAC) to vote for key points, and finally used PnP algorithm to derive accurate poses. This method relies on the key points of the 2D target object, and is not applicable to objects which are self-similar without texture information. Direct voting is to directly obtain a certain 6D pose by voting between each pixel or 3D point. For example, Drost et al. [17] defined a new four-dimensional point-pair feature to describe the object. Through feature matching, the relationship between scene point-pairs and model point-pairs is modeled. The generated candidate poses are voted to obtain the final result. This method performs well in scenes with noise, clutter, and partial occlusion, and, accordingly, is suitable for complex industrial environments.

In recent years, deep learning has made great breakthroughs in artificial intelligence [18]. Applying this method to robotic arms can improve the applicability of grasping. For example, Wang et al. [19] proposed a new dense fusion network to obtain pixel-level dense feature, thereby obtaining the pose of the target object. The advantage of this method is that an iterative fine-tuning process is integrated into the network architecture, which eliminates the dependence of fine pose on ICP, and is also robust to occlusion situations. But color and depth information are needed by the network, which adds complexity and cost. Braun et al. [20] designed a new method for joint target detection and pose prediction of deep convolutional networks. The disadvantage of this method is that it does not apply overlap and occlusion, and the estimation accuracy of the pose of small objects is not ideal.

As Drost's method [17] can well cope with complex scenarios, this method has been deeply studied in recent years, to make it play the best effect for different scenarios. For example, Choi et al. [21] improved the method by using color information. Thus, the four-dimensional point-pair feature is formed into a ten-dimensional point-pair feature of point cloud, which greatly improves the matching rate of colored objects. Liu et al. [22] improved the feature description of industrial parts based on the original method. In

this method, the normal vector in the original feature is changed to a tangent vector to enhance the feature representation of objects. A multi-edge appearance method of model description was proposed, to improve the efficiency by reducing useless point-pairs matching. Vidal et al. [23] presented to estimate 6D poses of free-form objects in the presence of clutter and occlusions. By considering the judgment value of surface information, a new viewpoint-dependent re-scoring process and two scene consistency verification steps were proposed to reduce false-positive cases. Ruel et al. [24] proposed a 3DLASSO system which was designed to perform real-time tracking and 6D pose estimation of target spacecraft from sparse and noisy 3D data. Different from the PPF method, instead of point-pairs larger polygons are used in a similar setup, and a faster version of the ICP algorithm is developed for pose estimation. The algorithm is quite robust to sensor noise and deviations from the reference model, but poses that do not provide enough geometric information to the algorithm showed larger errors.

In order to solve the bin-picking problem in industry, we have expanded and improved Drost's method [17]. The main contributions of the proposed method are: (1) An effective method for extracting candidate targets point cloud is adopted in preprocessing step. Specifically, the organized scene point cloud is mapped to the grayscale image, and the segmented grayscale images are mapped back to the point cloud. After threshold processing, only point cloud of unobstructed target objects in the scene are retained; (2) A new point-pair feature descriptor is proposed, which introduces curvature information based on the PPF method to effectively enhance the description of point-pair features; (3) In the pose voting link, a new weighted voting scheme is proposed by combining the curvature distribution of the model, which gives more weight to high information point-pairs, thereby further improving the accuracy of pose estimation.

The rest of this paper is organized as follows. The proposed method is presented in Section 2. Experimental results and discussions are given in Section 3. The conclusion is provided in Section 4.

2. The Proposed Method

Our work is based on the method proposed by Drost et al. [17]. Through the improvement and optimization of the PPF, 6D poses of target objects can be accurately achieved in complex industrial scenarios, enabling the robotic arm to complete the bin-picking tasks. The flow chart of the proposed method is shown in Figure 1, which comprises offline phase and online phase.

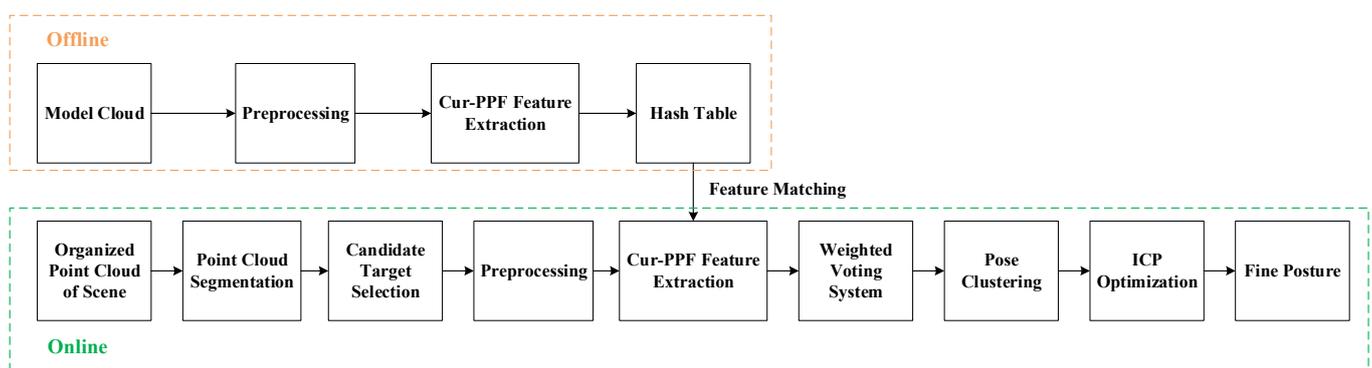


Figure 1. The full point-pair feature with curvature pipeline. The proposed method can be divided into offline stage and online stage.

In the offline stage, the CAD model of the object is used to generate point cloud, as shown in Figure 2. Firstly, the generated model point cloud is preprocessed, which mainly includes point cloud downsampling, normal calculation, and curvature calculation. Due to the mass of model point cloud, it causes calculation redundancy. In order to speed up the processing, downsampling operation is required. The normals and curvatures of

point cloud are calculated to prepare for PPF [25]. Then the high-curvature part and the low-curvature part are classified according to curvature distributions of models point cloud, and the pose weighted voting is performed on point-pairs with high information. Finally, the six-dimensional features of the model point-pairs are calculated, and features are stored in the hash table for features matching in the online phase.

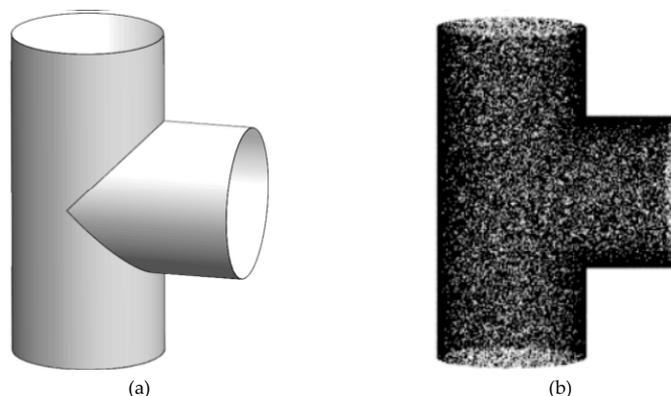


Figure 2. The three-way tube is one of models in our experiments. (a) is the CAD model of the three-way tube, and (b) is the point cloud model of the three-way tube after sampling.

In the online stage, the main work is to calculate 6D poses of target objects through PPF matching, to achieve precise grasping. We use a 3D sensor to obtain organized scene point cloud, which is outlier removed and transformed into gray images through mapping. Watershed algorithm is used [26] to segment gray images and candidate targets are extracted. For the segmented point cloud, the same preprocessing and features calculation are performed as done in the offline stage. By finding PPFs similar to target objects in the hash table, transformations among model point-pairs and scene point-pairs are derived, and the weighted votes of poses are completed in the two-dimensional accumulator. Finally, poses are clustered and the average of the highest clustered poses is used as the output result. The ICP algorithm is used to refine the pose estimation. In the next section, we will elaborate on all aspects of the proposed method, especially the differences from the PPF.

2.1. Offline Phase

2.1.1. Preprocessing

The preprocessing includes point cloud downsampling, normal calculation and curvature calculation. The point cloud downsampling and normal calculation are the same as the method by Drost et al. [17]. In the following, we focus on the point cloud curvature calculation.

Curvature can reflect the bending degree of geometry [27]. In the three-dimensional space, the curvature of the point cloud can provide special information for feature matching, which can effectively reduce matching error [28]. From the geometric description, the types of curvature can be divided into principal curvature, Gaussian curvature and average curvature. Principal curvature refers to the normal curvature in the principal direction of a point on the surface, and it is also the maximum and minimum values of the normal curvature of the surface in all directions at that point. At any point in the point cloud, there is a surface $z = r(x, y)$ approaching this point. Assuming that the principal curvature of this point is k_n , the k_n calculation formula is:

$$\begin{vmatrix} L - k_n E & M - k_n F \\ M - k_n F & N - k_n G \end{vmatrix} = 0, \quad (1)$$

$$(EG - F^2)k_n^2 - (LG - 2MF + NE)k_n + (LN - M^2) = 0. \quad (2)$$

The principal curvature k_n is obtained by solving the quadratic equation. In the formula $E = r_x r_x, F = r_x r_y, G = r_y r_y, L = r_{xx} n, M = r_{xy} n, N = r_{yy} n$; where $r_x, r_y, r_{xx}, r_{yy},$

r_{xy} is the partial differential of the surface $z = r(x, y)$, n is the value of the unit normal vector of the tangent plane of the surface $z = r(x, y)$ at the point (x_0, y_0) , that is, $n = (r_x \times r_y) / |r_x \times r_y|_{(x_0, y_0)}$. (E, F, G) is called the first basic invariant of the surface, and (L, M, N) is called the second basic invariant of the surface. The Gaussian curvature of a point on the surface is the product of the two principal curvatures, which is used to characterize the overall curvature of the local area, denoted as K , that is, $K = k_1 k_2$. The average curvature of a point on the surface is the average of the two principal curvatures, denoted as H , that is, $H = (k_1 + k_2) / 2$. Combining the principal curvature calculation Formula (2) and the Veda theorem, it can be known that the calculation formulas of Gaussian curvature and average curvature are:

$$K = \frac{LN - M^2}{EG - F^2}, \tag{3}$$

$$H = \frac{LG - 2MF + NE}{2(EG - F^2)}. \tag{4}$$

In order to better describe the change of the point cloud, we used the average curvature to represent curvature characteristics.

2.1.2. Cur-PPF Feature Extraction and Hash Table

The proposed Cur-PPF is a six-dimensional feature vector using the distance information of two points and its normal vector and average curvature. Compared with the original PPF, curvature information is introduced in the proposed method, which enhances the feature description of point-pairs. Cur-PPF is shown in Figure 3. For any point-pair (m_1, m_2) , m_1 and m_2 are two points in the model point cloud, \mathbf{n}_1 and \mathbf{n}_2 are the normal vectors of these two points, q_1 and q_2 are average curvatures of the two points, vector $\mathbf{d} = m_2 - m_1$, feature expression \mathbf{F} is:

$$\mathbf{F}_{Cur-PPF}(m_1, m_2) = (f_1, f_2, f_3, f_4, f_5, f_6) \\ = (\|\mathbf{d}\|_2, \angle(\mathbf{n}_1, \mathbf{d}), \angle(\mathbf{n}_2, \mathbf{d}), \angle(\mathbf{n}_1, \mathbf{n}_2), q_1, q_2), \tag{5}$$

where $\|\mathbf{d}\|_2$ represents the Euclidean distance between the two points, $\angle(a, b) \in [0, \pi]$ denotes the angle between two vectors. It should be noted that the feature $\mathbf{F}_{Cur-PPF}$ is asymmetric, just as $\mathbf{F}_{Cur-PPF}(m_1, m_2)$ and $\mathbf{F}_{Cur-PPF}(m_2, m_1)$ are not the same. In the offline stage, the model point cloud is represented with a set of similar features $\mathbf{F}_{Cur-PPF}$. Here we set the steps of distance, angle and curvature to d_{dist} , d_{angle} , and d_{cur} . Then point-pairs with similar characteristics are placed in the same slot of the hash table, and the keys of the hash table are characteristics of point-pairs, as shown in Figure 4. The model features $\mathbf{F}_{Cur-PPF}(m_i, m_j)$ can be searched in constant time by using $\mathbf{F}_{Cur-PPF}(s_i, s_j)$ as the key to access the hash table.

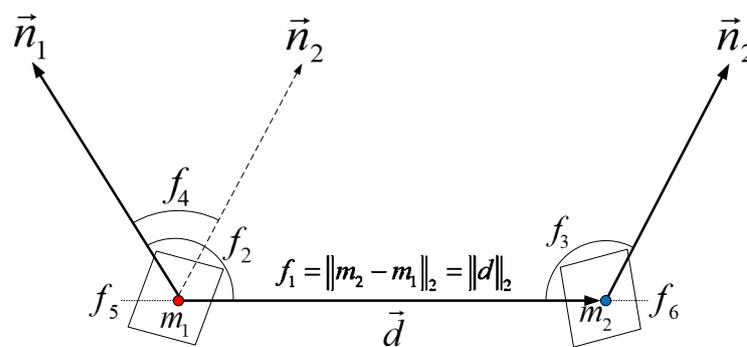


Figure 3. The model description of the Cur-PPF feature. This feature is different from the PPF because curvature information f_5 and f_6 are introduced to Cur-PPF, which strengthens the feature description.

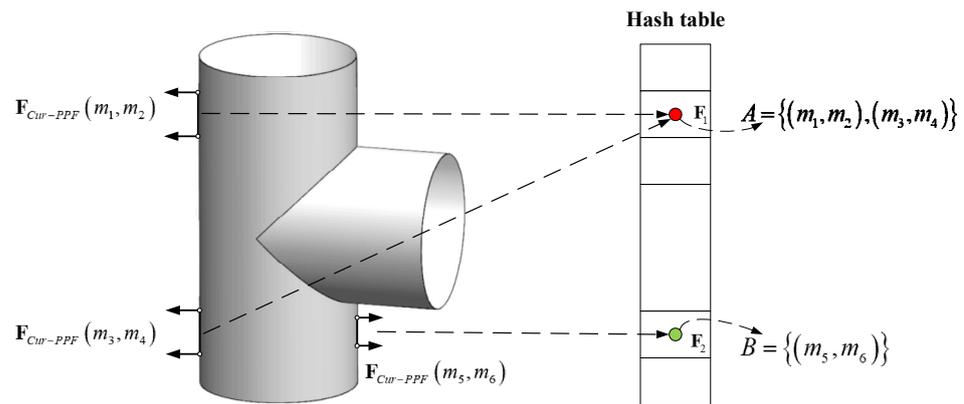


Figure 4. The hash table model. For the three point-pairs on the model, because point-pairs (m_1, m_2) and (m_3, m_4) have the same feature description, these two point-pairs are stored in the same slot of the hash table, and the key of the slot is represented by the feature F_1 of these two point-pairs; the feature description of point (m_5, m_6) is different from F_1 , so it is stored in another slot of the hash table, which is represented by feature F_2 .

2.2. Online Phase

2.2.1. Point Cloud Segmentation and Candidate Target Selection

Effectively extracting target objects in complex scenarios is very helpful for feature matching, so scene point cloud segmentation is performed. Point cloud segmentation can be divided into two categories [29]. The first type of method is the direct method, in which the point cloud is directly segmented, such as the Euclidean distance segmentation algorithm [30] integrated in the PCL library [31]. Its principle is to find a certain point in space, the n points closest to the point are found through KdTree, and the distance to the point is judged. If the distance is less than the threshold, it is considered to be of the same kind. This algorithm has to traverse all the points in the space, which is complicated and takes a long time, so it is not suitable for real-time system. The second is the indirect method. The point cloud is mapped to a two-dimensional image for segmentation, and then segmented images are mapped back to the three-dimensional space to achieve point cloud segmentation. The method is based on two-dimensional image processing, with high accuracy and less time consumed [32].

Because the point cloud is obtained by the 3D sensor in this system and the order of the point cloud is known [33], we chose the second method to achieve point cloud segmentation. Firstly, the ordered point cloud is projected onto the plane composed of x -axis and y -axis of the coordinate system, and the effective detection range of the depth value in the z -axis direction is mapped to become the gray value. Then the watershed segmentation algorithm [26] is used to segment the gray image, so an image is divided into several disjoint local areas. Finally, gray images are mapped back to the three-dimensional space to complete the point cloud segmentation. For a more detailed understanding of the segmentation process, we describe it using pseudocode, which is shown in Algorithm 1.

There are usually overlapping occlusions in the picking scenarios. The candidate objects grabbed by the robotic arm are the top priority (that is, the ones that are not occluded or have a large exposed surface), which also conforms to the logical order of grabbing. Therefore, grayscale images are thresholded after watershed segmentation. Firstly, the single-sided point cloud of a single object in the scene is obtained by a 3D sensor and mapped to a grayscale image to obtain the number of pixels of the image. Then, the number of local pixels after segmentation are compared with the number of pixels on one side of the object. If the number of surface pixels is similar to the number of surface pixels on one side of the object, and the number of contour pixels is similar to the number of contour pixels on one side of the object, we consider the object to be a candidate to be grasped by the robotic arm. Finally, each pixel is mapped to three-dimensional space to complete the effective segmentation and the selection of candidate targets. Three-way tube

is a category in the test data set of this paper, and is demonstrated as a legend, as shown in Figure 5.

Algorithm 1 Watershed Segmentation Algorithm Based on Distance Transform

```

1: Input:  $I$ , Output:  $O$ 
2: if  $I(i, j) = (255, 255, 255)$ 
    $I(i, j) = (0, 0, 0)$ 
end if
3:  $L \leftarrow$  Laplacian operator( $I$ )
4:  $S \leftarrow$  Sharp( $L$ )
5:  $G \leftarrow$  Grayscale( $S$ )
6: if  $G(i, j) > t_1$ 
    $G(i, j) = 255$ 
else
    $G(i, j) = 0$ 
end if
7:  $D \leftarrow$  Distance transform( $G$ )
8:  $N \leftarrow$  Normalized( $D$ ),  $N(i, j) \in [0, 1]$ 
9: if  $N(i, j) > t_2$ 
    $N(i, j) = 255$ 
else
    $N(i, j) = 0$ 
end if
10:  $P \leftarrow$  Erode( $N$ )
11:  $M \leftarrow$  Find and draw contours( $P$ )
12:  $O \leftarrow$  Watershed( $S, M$ )

```

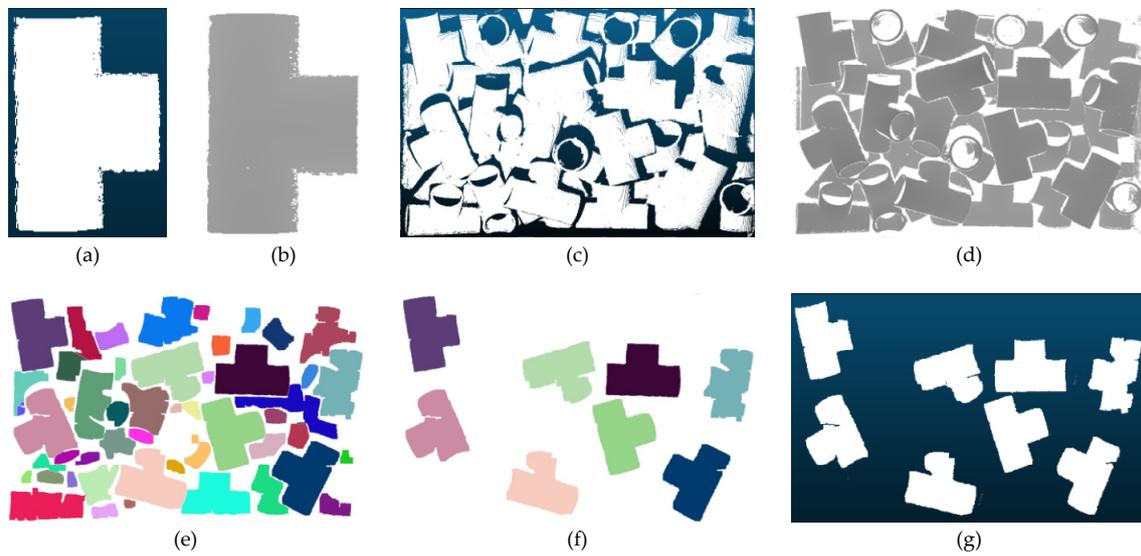


Figure 5. The process of point cloud segmentation and candidate targets selection. (a) is the point cloud of the one-sided model; (b) is the grayscale image of the one-sided model; (c) is the point cloud of the scene; (d) is the grayscale image which is mapped from the depth information of the scene; (e) is the grayscale image of the scene after segmentation; (f) is the grayscale image of the scene after target selecting; and (g) is the point cloud which is mapped by (f).

2.2.2. Feature Matching

Feature matching refers to successfully finding PPFs of the model in the hash table, so that the transformation can be calculated. In this paper, the local coordinate system is established for solving. Given a point-pair (s_r, s_j) in the scene, the Cur-PPF of the point-pair is calculated and the feature as the key value is used to find the corresponding

model point-pair (m_r, m_j) in the hash table. The two points s_r and m_r are moved to the origin of the local coordinate system, and the normals of these two points are aligned with the x – axis, so that the object can be rotated around the normal to align the model with the scene, as shown in Figure 6. The transformation from the model to the scene can be represented by a point and a rotation angle α , which is (m_r, α) . If the model point-pair (m_r, m_j) and the scene point-pair (s_r, s_j) have similar Cur-PPF, the conversion relationship between the two point-pairs can be calculated by the Formula (6).

$$s_i = \mathbf{T}_{s \rightarrow g}^{-1} \mathbf{R}_x(\alpha) \mathbf{T}_{m \rightarrow g} m_i, \quad (6)$$

where, $\mathbf{T}_{m \rightarrow g}$ is a transformation with rotation and translation, which translates the reference point m_r in the model point-pair feature (m_r, m_i) to the origin of the coordinate system, and at the same time rotates the normal vector n_r^m of the reference point m_r to the same direction as the x – axis of the coordinate system. $\mathbf{T}_{s \rightarrow g}$ is also a transformation with rotation and translation, which translates the reference point s_r in the model point-pair feature (s_r, s_i) to the origin of the coordinate system, and at the same time rotates the normal vector n_r^s of the reference point s_r to the same direction as the x – axis of the coordinate system. $\mathbf{T}_{s \rightarrow g}^{-1}$ is the inverse of $\mathbf{T}_{s \rightarrow g}$. $\mathbf{R}_x(\alpha)$ is the rotation around the x – axis with angle α .

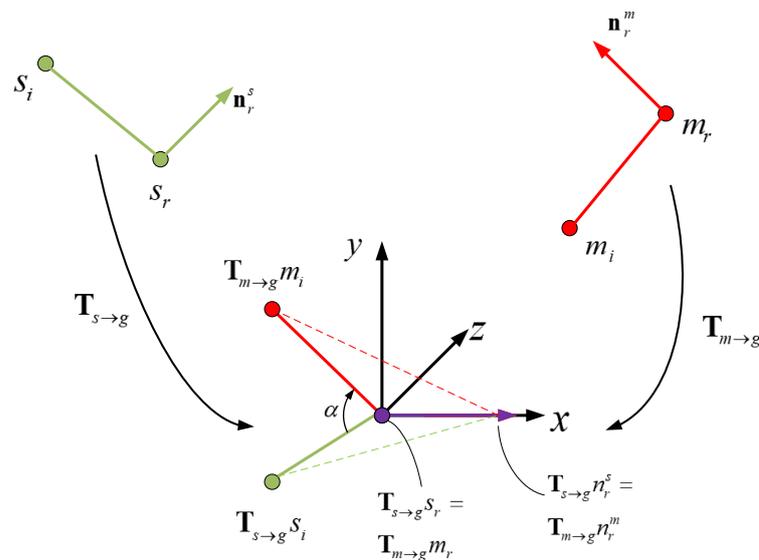


Figure 6. Transformation between model point-pairs and scene point-pairs. The transformation relationship $\mathbf{R}_x(\alpha)$ is obtained by aligning the point-pair vector and its normal vector.

In order to improve the calculation speed of α angle, α can be divided into two parts, namely $\alpha = \alpha_m - \alpha_s$. Where, α_m is the rotation angle at which the model point-pair (m_r, m_i) continues to rotate around the x – axis after the transformation of $\mathbf{T}_{m \rightarrow g}$, so that the point m_i falls on the plane composed of the x – axis and the positive half-axis of the y – axis; α_s is the rotation angle at which the scene point-pair (s_r, s_i) continues to rotate around the x – axis after the transformation of $\mathbf{T}_{s \rightarrow g}$, so that the point s_i falls on the plane composed of the x – axis and the positive half-axis of the y – axis; the direction of rotation of the two remains the same. The calculation of these two parts is independent of each other, so we can split $\mathbf{R}_x(\alpha) = \mathbf{R}_x(-\alpha_s) \mathbf{R}_x(\alpha_m)$ and use $\mathbf{R}_x^{-1}(-\alpha_s) = \mathbf{R}_x(\alpha_s)$ to get

$$\mathbf{t} = \mathbf{R}_x(\alpha_s) \mathbf{T}_{s \rightarrow g} s_i = \mathbf{R}_x(\alpha_m) \mathbf{T}_{m \rightarrow g} m_i. \quad (7)$$

i.e., \mathbf{t} lies on the half-plane defined by the x – axis and the non-negative part of the y – axis. For successfully paired point-pairs, α_m can be calculated for model point-pairs in the offline phase and store them in the hash table. In this way, only α_s needs to be calculated for scene point-pairs. The final angle α is the difference between the two angles.

2.2.3. Weighted Voting System

We search model point-pairs (m_r, m_j) with same Cur-PPF features as (s_r, s_j) from the hash table. Formula (6) is used to calculate mapping relationships α from each model point-pairs (m_r, m_j) to scene point-pairs (s_r, s_j) . Then we use a method similar to the generalized Hough transform to vote on the obtained α and select the best mapping relationship to restore the global pose of the object.

Voting process is completed through a two-dimensional accumulator. The rows N_m of the accumulator is equivalent to the number of model points M , and the columns N_{angle} is equivalent to the step length n_{angle} of the conversion relationship α . Whenever scene point-pairs (s_r, s_j) are successfully paired with model point-pairs (m_r, m_j) in the hash table, the calculated α are voted. The difference from the PPF is that our method combines the model curvature distribution in the actual voting process, and different α votes are assigned different weights. When scene point-pairs and model point-pairs are successfully paired, we will focus on the relationship between the point m_r and the point m_j in the model point-pair (m_r, m_j) . From Section 2.1.1, the average curvature value of each point in the model point cloud can be calculated. The curvature distribution of the three-way tube model is shown in Figure 7a. Different colors represent the average curvature value. It can be seen from characteristics of curvature that point cloud with similar curvature values is also similar in bending, and such point cloud is distributed in the same area in space. And point cloud with large differences in curvature values also has large differences in the degree of bending, and such point cloud is distributed far apart in space. We believe that point-pairs with the greater difference in curvature values of the two points contain more information, and the mapping relationship α is calculated by the pairing is more accurate, such that α should be given a higher weight when voting, as shown in Formula (8). For example, in the three-way tube model of this experiment, the high-curvature part and low-curvature part of the model are divided according to the curvature histogram. The curvature histogram is shown in Figure 7b. Weighted vote is performed on the calculated α that has a greater difference in curvature values between the two points in the model point-pair. The voting process is shown in Figure 8.

$$\text{Weight} = \begin{cases} +W & \text{when } m_1 \in \text{high-cur} \& m_2 \in \text{low-cur} \parallel m_1 \in \text{low-cur} \& m_2 \in \text{high-cur} \\ +1 & \text{other} \end{cases} \quad (8)$$

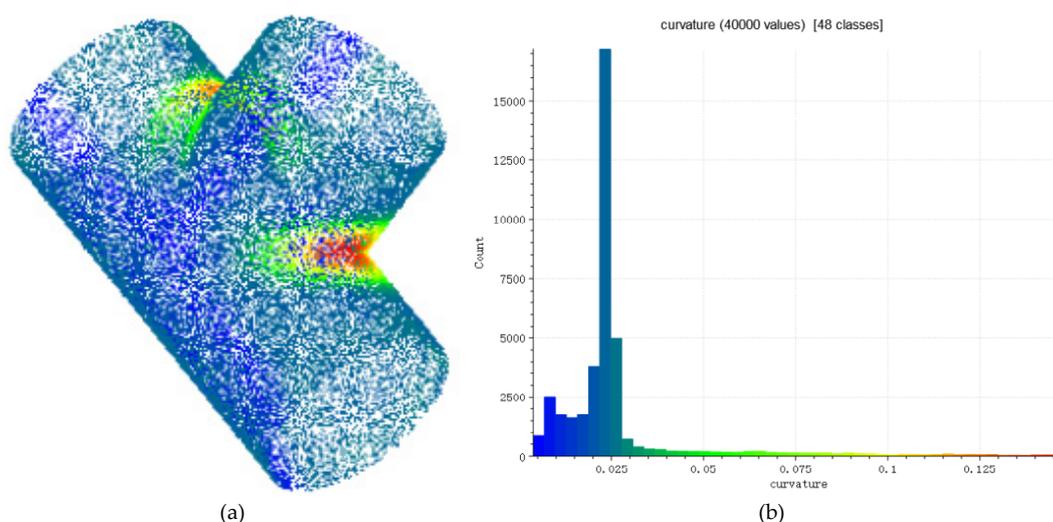


Figure 7. (a) is the curvature distribution of the three-way tube model. The color from red to blue corresponds to the average curvature of the point cloud from large to small. (b) is the curvature histogram of the three-way tube. According to the curvature histogram, we set 0–0.035 as the low-curvature range, and greater than 0.1 as the high-curvature range.

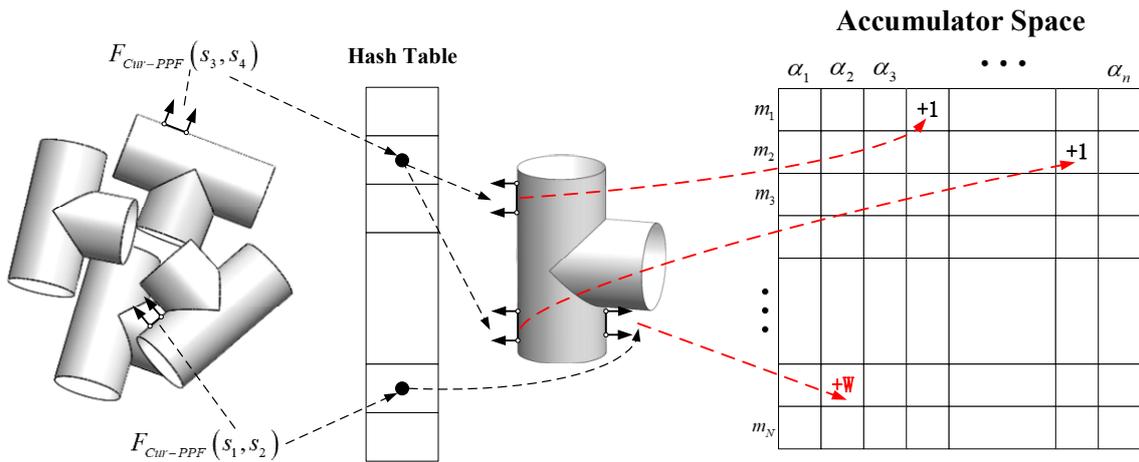


Figure 8. Because the scene point-pair (s_3, s_4) is matched with two points of the same curvature class in the model point-pair, the number of pose votes is one by default; when matching (s_1, s_2) , the two points in the model point-pair are in the set high-curvature and low-curvature ranges respectively, so the match contains more information and weighted voting is performed on the pose.

2.2.4. Pose Clustering

When reference points are located on the object surface, multiple effective point-pairs will be generated. Each point-pair will be calculated a pose result after feature matching, so an object will have a set of pose sets. The pose sets are clustered to ensure that the translation and rotation errors of all poses in each category are in the set threshold. The score of each pose is the cumulative sum of votes obtained by that pose during the voting phase. The category with the highest score is selected, and poses contained in this category are averaged to obtain the final pose results. This operation not only removes the pose data with large errors through the threshold, but also improves the accuracy of the final pose result by the average value. Since there will be multiple objects in the scene, multiple high-scoring categories will be generated, and the category with the highest number of votes is selected as the preferred pose.

2.2.5. ICP Optimization

In order to further improve the accuracy of the pose results, we used the ICP algorithm [13] for optimization after the pose obtained by the pose clustering. The clustering pose is used as the initial value of the ICP algorithm, and the error is further reduced by continuously reducing the Euclidean distance between the model point and the corresponding scene point. On basis of whether model points match scenic points successfully by setting the distance threshold. If the distance between the two points is less than the threshold, it is considered that the two points match successfully. Finally, the ratio ∂ between the number of matched points and the number of object points in the scene is taken as the matching rate, as shown in Formula (9). In real experimental scenario, when the value of the matching rate can enable the robotic arm to successfully grasp the target object, it is the minimum matching rate that we can accept.

$$\partial = \frac{\text{Number of matching success points}}{\text{Number of object points in the scene}}. \quad (9)$$

3. Experimental Results and Discussions

We used online public data set and real scene data to verify the effectiveness of the proposed method, and used a robotic arm to perform bin-picking tasks to evaluate the performance of the method in industrial applications. Our algorithm was implemented in C++ language under the Visual Studio2019 platform and was run on the NVIDIA GeForce GTX1060 processor. Through experimental comparison, the advantages of the

proposed method over the original method are verified in terms of accuracy, efficiency, and adaptability.

3.1. Public Data Set

We used the online Retrieval [34] data set to verify the advancement of the proposed method. The data set includes 6 models and 18 scenes, and the model is shown in Figure 9. Each scene has only one set of point cloud data, which prevents other factors from interfering with the experimental comparison results. For all experiments, the Leaf_size of the model point cloud and scene point cloud downsampling was set to 5 mm; the hash table distance step d_{dist} was set to 3 mm; the angle step d_{angle} was set to 12° ; and the 1/5 of the point cloud number was used as the scene reference point. The matching rate of the point cloud was calculated by Formula (8) in Section 2.2.5, where the threshold was set to 5 mm.

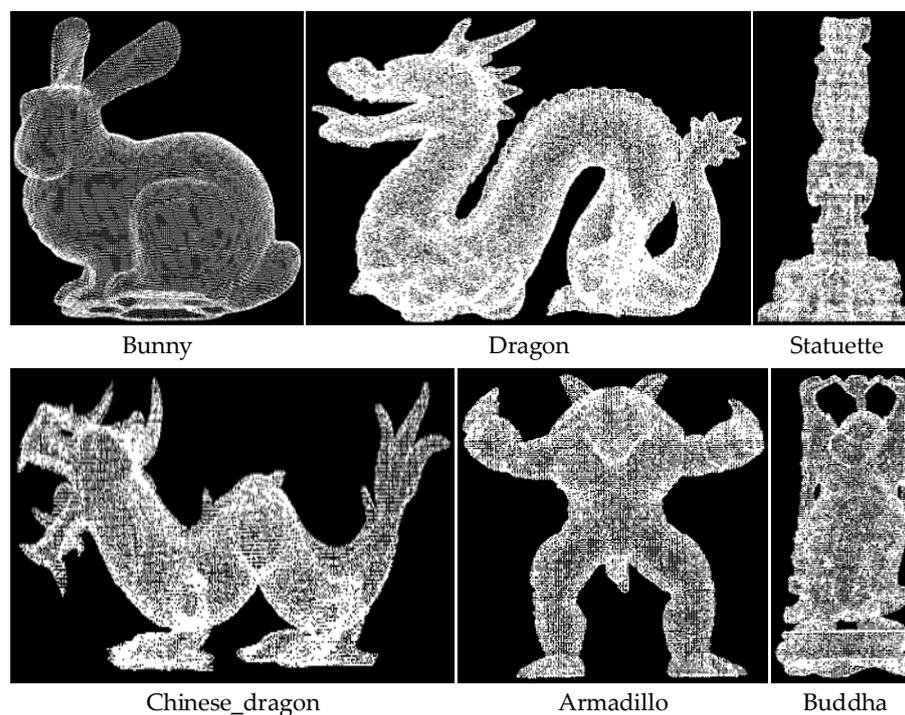


Figure 9. The model point cloud data set. The data set includes six point cloud models, namely, Bunny, Dragon, Statuette, Chinese_Dragon, Armadillo, and Buddha.

We verified the enhancement effect of curvature on the PPF description in the proposed method. Each model in the Retrieval data set corresponds to multiple scenes with different levels of noise. In order to reduce the impact of noise on the matching effect, a scene with a noise coefficient of 0.1 was selected for matching. The final matching rate is the average one between each model and multiple scenes, and the average of matching time with multiple scenes is viewed as the final time. The radius of curvature of models in the data set was set to 15 mm. Due to the different curvature distributions of each model, the curvature steps d_{cur} of Bunny, Dragon, Statuette, Chinese_Dragon, Armadillo, and Buddha were set to 0.07, 0.1, 0.13, 0.15, 0.2, and 0.11, respectively. The matching experiments of the PPF algorithm and the Cur-PPF (unweighted) algorithm were carried out respectively. A set of matching effects are shown in Figures 10 and 11. Tables 1 and 2 are the data comparison between the PPF algorithm and the Cur-PPF (unweighted) algorithm in terms of matching rate and time. The experimental results show that the introduction of curvature information can strengthen the description of the feature, and it is better than the original PPF algorithm in terms of matching rate and time.

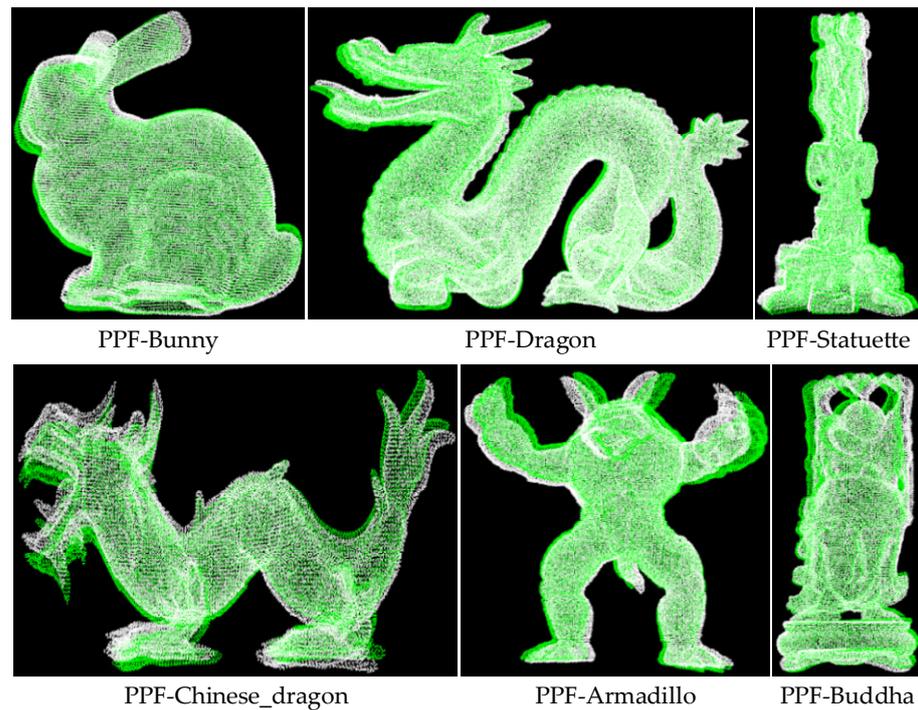


Figure 10. The PPF algorithm is used to register the six kinds of point cloud models of the data set. The pose results are used to convert the point cloud of the models into scene space, and the color is used for rendering, where white represents the point cloud of the scene, and green represents the converted model point cloud.

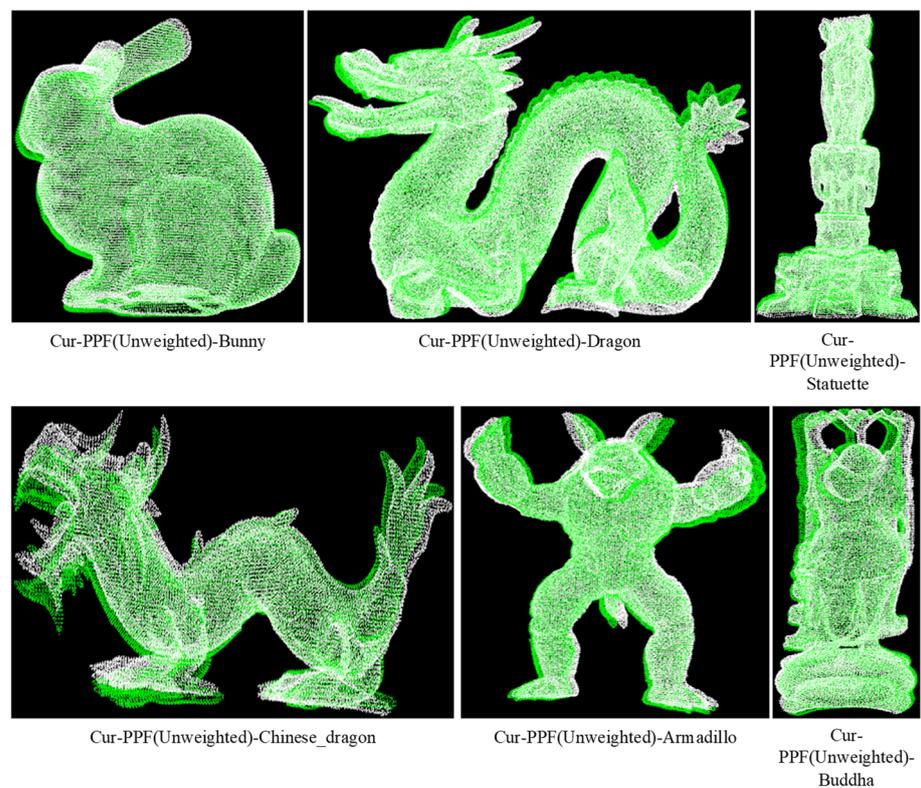


Figure 11. The Cur-PPF(Unweighted) algorithm is used to register the six kinds of point cloud models of the data set. The pose results are used to convert the point cloud of the models into scene space, and the color is used for rendering, where white represents the point cloud of the scene, and green represents the converted model point cloud.

Table 1. The matching rate of PPF and Cur-PPF(Unweight) algorithms on public data set.

Models	Bunny	Dragon	Statuette	Chinese_Dragon	Armadillo	Buddha	Average
PPF [16]	87.42%	84.71%	84.92%	94.77%	81.40%	93.25%	87.75%
Cur-PPF(Unweight)	93.12%	95.96%	89.91%	95.74%	92.94%	94.25%	93.65%

Table 2. Time using of PPF and Cur-PPF(Unweight) algorithms on public data set (ms/scene).

Models	Bunny	Dragon	Statuette	Chinese_Dragon	Armadillo	Buddha	Average
PPF [16]	145	745	1151	893	341	803	679.67
Cur-PPF(Unweight)	85	165	169	233	203	221	179.33

We also verified that the weighted voting in the proposed method has an enhanced effect on the matching effect. According to curvature histograms of point cloud models, the high-curvature part and the low-curvature part of models are divided [35]. The curvature histograms of the models point cloud are shown in Figure 12. Through multiple experiments with different models, we think that setting the weight to 2–8 is a better range. The setting of the experimental parameters is consistent with the Cur-PPF(Unweight) parameters. The matching experiments of the Cur-PPF(unweight) algorithm and the weighted Cur-PPF algorithm were carried out respectively. The matching effect of a group of the weighted Cur-PPF algorithm are shown in Figure 13. Tables 3 and 4 are the comparison of the matching rate and time between the Cur-PPF(Unweight) algorithm and the Cur-PPF algorithm. The experimental results show that the weighted operation introduced into the pose voting link further improve the point cloud matching rate, and the time is basically similar to the unweighted Cur-PPF algorithm, which proves the role of the weighted operation.

Table 3. The matching rate of Cur-PPF(Unweight) and Cur-PPF algorithms on public data set.

Models	Bunny	Dragon	Statuette	Chinese_Dragon	Armadillo	Buddha	Average
Cur-PPF(Unweight)	93.12%	95.96%	89.91%	95.74%	92.94%	94.25%	93.65%
Cur-PPF	94.40%	99.84%	95.44%	97.09%	94.20%	96.80%	96.30%

Table 4. Time using of Cur-PPF(Unweight) and Cur-PPF algorithms on public data set (ms/scene).

Models	Bunny	Dragon	Statuette	Chinese_Dragon	Armadillo	Buddha	Average
Cur-PPF(Unweight)	85	165	169	233	203	221	179.33
Cur-PPF	87	195	289	226	241	236	212.33

The method proposed by Drost et al. can recognize different objects in the same scene. In order to verify that the improved method proposed in this paper based on the original PPF can also effectively recognize different objects in the same scene, we choose the public dataset Laser Scanner as an experiment. Since the method in this paper focuses more on the scene of the same object in bin-picking, this experiment serves as a supplementary experiment to verify the ability of the proposed method to recognize different objects. We compared the matching rates of Cur-PPF and Cur-PPF+ICP. The results are shown in Figure 14, and the average matching rates are shown in Table 5. Experiments show that the improved method proposed in this paper has similar functions to the original PPF method, not only can identify different objects in the same scene, but also has a satisfactory coarse registration effect. After ICP optimization, the average matching rate of fine registration can reach 93%.

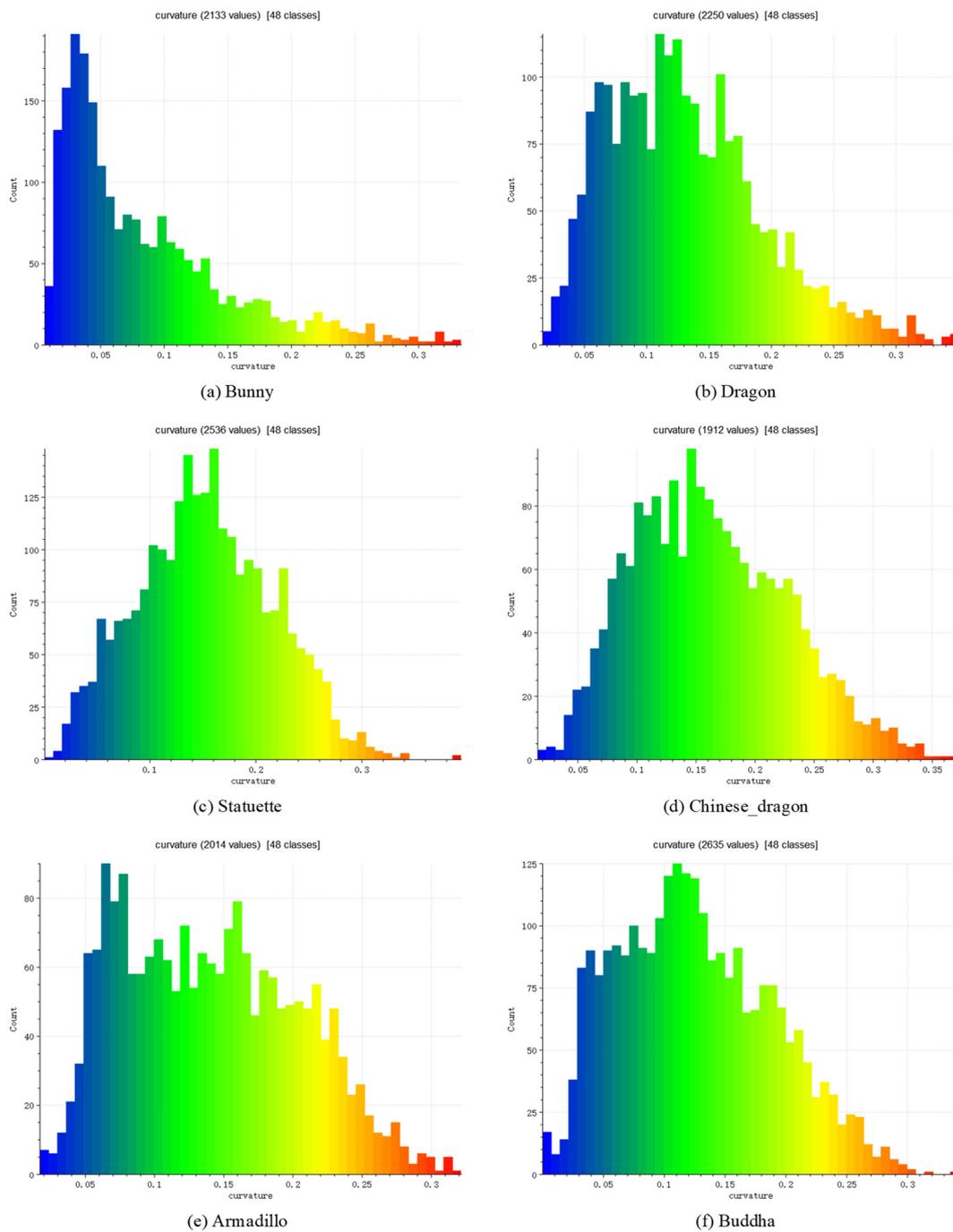


Figure 12. Curvature histograms of six models. The curvature of each model is divided into high/low part according to curvature histograms. (a) is the curvature histogram of Bunny model. Its high-curvature part is greater than 0.2 and the low-curvature part is 0–0.02; (b) is the curvature histogram of the Dragon model. Its high-curvature part is greater than 0.22 and the low-curvature part is 0–0.07; (c) is the curvature histogram of the Statuette model. Its high-curvature part is greater than 0.2 and the low-curvature part is 0–0.1; (d) is the curvature histogram of the Chinese_dragon model. Its high-curvature part is greater than 0.24 and the low-curvature part is 0–0.1; (e) is the curvature histogram of Armadillo model. Its high-curvature part is greater than 0.18 and the low-curvature part is 0–0.07; (f) is the curvature histogram of Buddha model. Its high-curvature part is greater than 0.18 and the low-curvature part is 0–0.07.

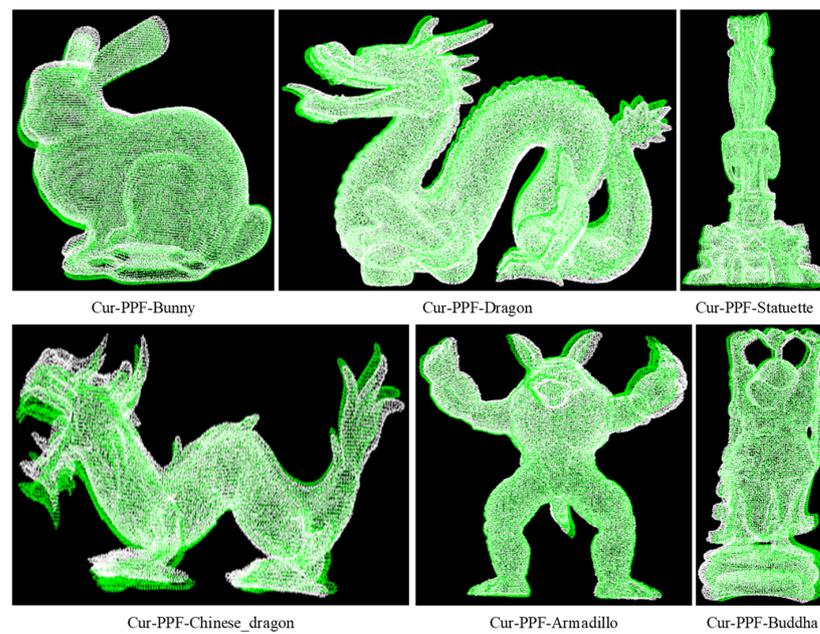
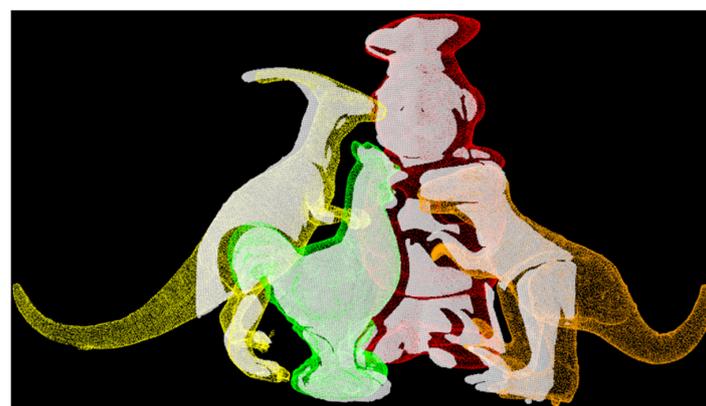
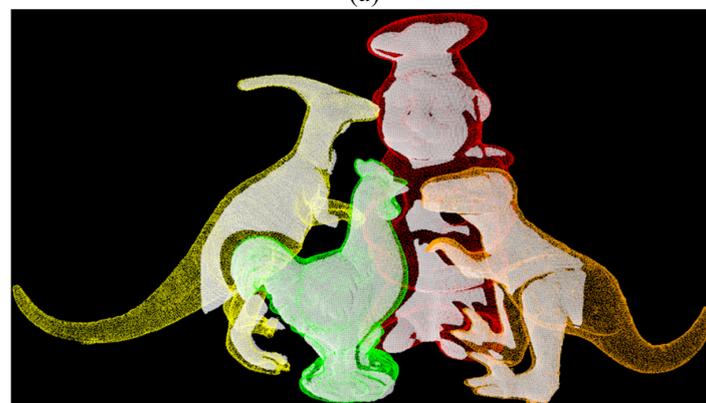


Figure 13. The weighted Cur-PPF algorithm is used to register the six kinds of point cloud models of the data set. The pose results are used to convert the point cloud of the models into scene space, and the color is used for rendering, where white represents the point cloud of the scene, and green represents the converted model point cloud.



(a)



(b)

Figure 14. (a) is the result of using our proposed Cur-PPF algorithm to recognize different objects in the same scene, and (b) is the registration result of Figure (a) after optimization by the traditional ICP algorithm.

Table 5. The matching rate of Cur-PPF and Cur-PPF+ICP algorithms on Laser Scanner.

Models	Cheff	Chicken	T-Rex	Parasaurolophus	Average
Cur-PPF	91.41%	87.60%	90.68%	86.01%	88.93%
Cur-PPF+ICP	95.15%	94.37%	92.86%	90.31%	93.17%

3.2. Real Scene Data

In the previous section, the advantages of the proposed method Cur-PPF without clutter, overlapping occlusion are verified. However, in real scenes, the environment is chaotic and noisy, and it becomes more difficult for the robot to perform grasping tasks. In order to verify that the proposed method also has advantages in complex scenes, we built a robotic arm bin-picking scene, and the system is shown in Figure 15. The bin-picking scene is also one of the common scenes in the industry. In this scene, there is overlap and occlusion among target objects, which cause interference to the matching. In order to evaluate the algorithm, we consider the point cloud matching effect and the grasping rate of the robotic arm.

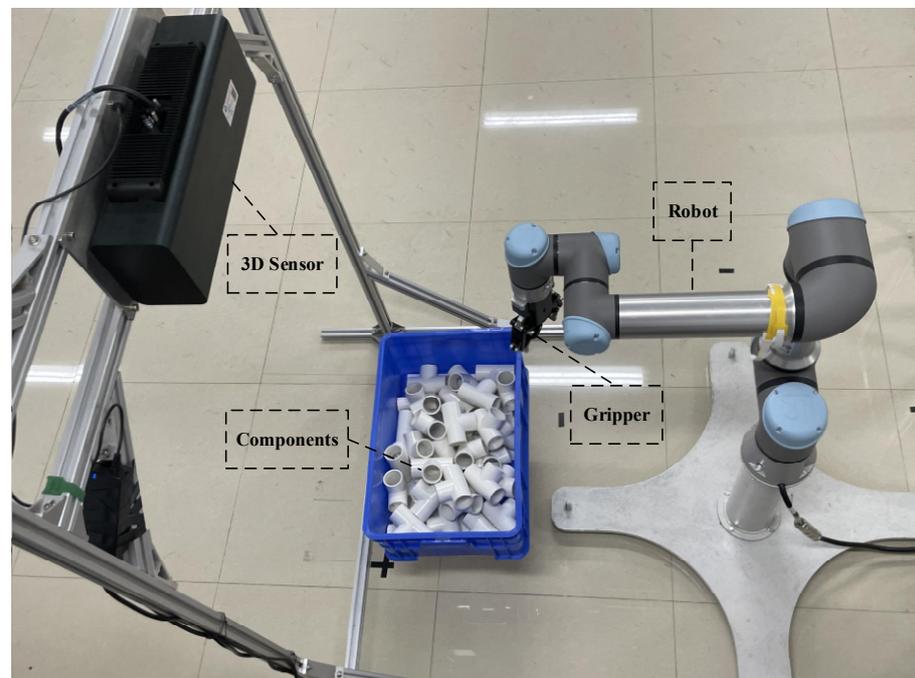


Figure 15. Bin-picking system diagram. The system is composed of robot, gripper, components, and 3D sensor.

3.2.1. Matching Effect of Real Scenario

In the real scenario matching experiment, we used common objects in the industry as test objects. The point cloud and image data were acquired by a 3D sensor (a COBOT COMATRIX-IM camera, consisting of a gray-scale camera and a projector). We randomly put test objects into the box, and collected 20 sets of test scenarios for each type of object, and used the PPF algorithm and the algorithm proposed in this paper to perform match experiments. The experimental parameters were set as follows: the Leaf_size of the model point cloud and scene point cloud downsampling were set to 3 mm; the hash table distance step d_{dist} was set to 0.5 mm; the angle step d_{angle} was set to 12° ; the 1/5 of the point cloud number was used as the scene reference point; the radius of curvature was set to 10 mm, the curvature step d_{cur} of the first type of object was set to 0.025, and the curvature step d_{cur} of the second type of object was set to 0.3; the low-curvature range of the first type of object is 0–0.015, the high-curvature range is greater than 0.06, and the voting weight was set to 3; the second type of object has low-curvature range of 0–0.015, high-curvature

range of greater than 0.065, and voting weight was set to 5. In calculating the matching rate between the model point cloud and the scene point cloud, the distance threshold was set to 5 mm.

We used the PPF algorithm [17] and the Cur-PPF algorithm proposed in this paper to perform point cloud matching respectively, and the ICP algorithm was used to correct the matching results. The point cloud matching processes are shown in Figure 16. In order to effectively compare the two algorithms, we only keep the top five matching results in the scene for the first type of object. For the second type of objects, the volume of the objects is larger, and the top layer can only be placed at most five, so only the results of the top three match rates in the scene are retained. The matching results are rendered in different colors, and the average of the matching rate is regarded as the final matching rate. Tables 6 and 7 show the comparison of the parameters of the two algorithms in terms of matching rate and time. It can be seen that the method proposed in this paper has greater advantages than the original method in the bin-picking scenario.

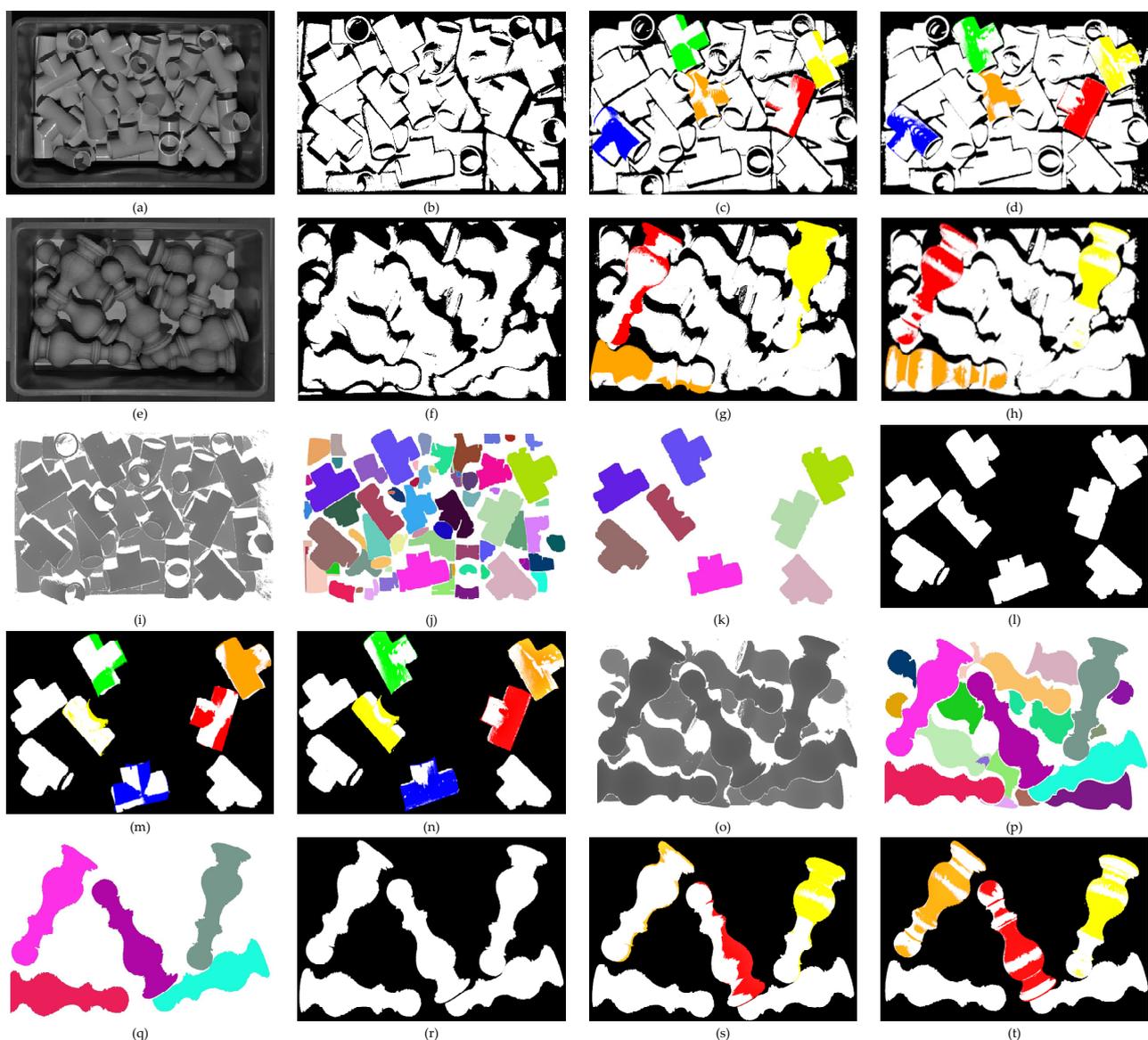


Figure 16. The matching process of the PPF algorithm and the Cur-PPF algorithm for two common objects in the industry. Among them, (a) is the three-way tube (the first type of objects); (e) is the upright column (the second type of objects); (b,f) correspond to the scene point cloud of two types of objects, the outer frame of the box is filtered out by setting thresholds on x - axis, y - axis respectively;

(c,g) are matching results of the PPF algorithm for two types of objects; (d,h) are matching effect pictures after ICP correction; (i,o) are mapped from point cloud depth information to grayscale images; (j,p) are grayscale images after segmentation; (k,q) are candidate objects that are screened out according to the number of pixels in the segmented image; (l,r) are point clouds of candidate objects; (m,s) are the point cloud matching effect diagrams of the Cur-PPF algorithm; and (n,t) are matching effect pictures after ICP correction. The matching rate from high to low is rendered in the order of red, orange, yellow, green, and blue.

Table 6. The matching rate of PPF and Cur-PPF algorithms on real data sets.

Models	Three-Way Tube	Pillar	Average
PPF	83.15%	87.84%	85.50%
Cur-PPF	95.60%	94.35%	94.98%
PPF+ICP	96.10%	95.25%	95.68%
Cur-PPF+ICP	98.90%	97.50%	98.20%

Table 7. Time using of PPF and Cur-PPF algorithms on real data sets (ms/scene).

Models	Three-Way Tube	Pillar	Average
PPF	7034	8560	7797
Cur-PPF	3256	4236	3746
PPF+ICP	8098	9362	8730
Cur-PPF+ICP	4136	5082	4609

3.2.2. Bin-Picking Performance of Robotic Arm

In order to verify the validity of the method proposed in this article, we used a six-axis robotic arm to perform bin-picking. In this system, the model of the robotic arm is UR5e (UNIVERSAL ROBOTS), the model of the gripper is AG-95 (DH ROBOTS), and the model of the 3D sensor is COMATRIX-IM (COBOT). Our experiment was carried out indoors. The light source is indoor incandescent lamp, and no specific light source is added.

We randomly placed 25 three-way tubes in the bin, and used the Cur-PPF algorithm to match the model with the scene. Each three-way tube in the scene will generate a set of pose results after weighted voting. The pose results after clustering were corrected using the ICP algorithm. According to our experience, when the matching rate is greater than 85%, the robotic arm can successfully grasp the target object. If the matching rate is less than 85%, the robotic arm will grab empty or pose error when grasping, which is considered as a wrong matching result. We carried out a total of 100 three-way tube grasping experiments, and the results showed five grasping failures, as shown in Table 8. Three of the failures were due to the close proximity of the three-way tubes, and the nearby objects were encountered before grasping, which caused the pose of the target object to change. Because of the low matching rate, which made the posture accuracy of the points captured by the robotic arm poor, and eventually led to other failure of the grasping operation.

Table 8. Capture success rate for three-way tubes.

Total Number of Experiments	Success	Failure	Success Rate
100	95	5	95%

4. Conclusions

We propose a 6D pose estimation method based on a new point-pair feature descriptor. In this method, an effective point cloud preprocessing is introduced, which can accurately extract candidate target objects and improve the matching efficiency. At the same time, the curvature information is introduced into the point-pair feature descriptor, which enhances the feature description and improves the matching accuracy. In addition, a weighted voting

method is proposed in the pose voting link, which further improves the accuracy of pose estimation. At the end of this paper, we test the proposed method and the PPF on public data set and real scenarios. The experimental results show that the average matching rate of our method on the public data set has increased by 8.55%, and the average time taken has been shortened by 467.34 ms. In real scenarios, the average matching rate of our method has increased by 12.7%, and the average time taken has been shortened by 3188 ms, and the capture rate in the bin-picking scenarios is as high as 95%. It can be seen that the method proposed in this paper has the advantages of high pose estimation accuracy and short calculation time, and can be used in actual industrial scenarios.

In the future, we will continue to study the mathematical model of high-curvature and low-curvature partitioning in the weighting strategy, which will improve the efficiency of the strategy when applied to new objects. The point cloud matching rate can also be improved by accurately dividing the model curvature; in addition, there are useless model point-pairs during matching, and it is worth exploring how to avoid useless point-pairs in the future, which will further improve the overall efficiency.

Author Contributions: X.C. and M.Y. designed and performed the experiments; X.C., M.Y. and L.W. contributed the scientific issues and the research ideas; X.C. and S.W. wrote and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grant NO.61775172) and the Hubei Key Technical Innovation Project, grant number ZDCX2019000025.

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

References

1. Inagaki, Y.; Araki, R.; Yamashita, T.; Fujiyoshi, H. Detecting layered structures of partially occluded objects for bin picking. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Macau, China, 3–8 November 2019.
2. Danielczuk, M.; Mahler, J.; Correa, C.; Goldberg, K. Linear Push Policies to Increase Grasp Access for Robot Bin Picking. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Munich, Germany, 20–24 August 2018; IEEE: Piscataway, NJ, USA, 2018.
3. Iriando, A.; Lazkano, E.; Ansuategi, A. Affordance-based grasping point detection using graph convolutional networks for industrial bin-picking applications. *Sensors* **2021**, *21*, 816. [[CrossRef](#)] [[PubMed](#)]
4. Matsumura, R.; Harada, K.; Domae, Y.; Wan, W. Learning based industrial bin-picking trained with approximate physics simulator. In Proceedings of the Advances in Intelligent Systems and Computing, Cham, Switzerland, 23 May 2018.
5. Hofer, T.; Shamsafar, F.; Benbarka, N.; Zell, A. Object Detection And Autoencoder-Based 6d Pose Estimation For Highly Cluttered Bin Picking. *arXiv* **2021**, arXiv:2106.08045.
6. Chen, J.; Zhang, L.; Liu, Y.; Xu, C. Survey on 6D Pose Estimation of Rigid Object. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020.
7. Du, G.; Wang, K.; Lian, S. Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review. *arXiv* **2019**, arXiv:1905.06658.
8. Hu, Y.; Hugonot, J.; Fua, P.; Salzmann, M. Segmentation-Driven 6D Object Pose Estimation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
9. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D Registration. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
10. Tombari, F.; Salti, S.; Di Stefano, L. Unique signatures of histograms for local surface description. In Proceedings of the European Conference on Computer Vision, Berlin, Germany, 5–11 September 2010.
11. 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, Barcelona, Spain, 6–13 November 2011.
12. Xue, S.; Zhang, Z.; Lv, Q.; Meng, X.; Tu, X. Point Cloud Registration Method for Pipeline Workpieces Based on PCA and Improved ICP Algorithms. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *612*, 032188.
13. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, San Diego, CA, USA, 19–24 July 1992.
14. Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R.A.; Lucey, S.; Choset, H. PCRNNet: Point Cloud Registration Network using PointNet Encoding. *arXiv* **2019**, arXiv:1908.07906.
15. Guo, J.; Xing, X.; Quan, W.; Yan, D.M.; Gu, Q.; Liu, Y.; Zhang, X. Efficient Center Voting for Object Detection and 6D Pose Estimation in 3D Point Cloud. *IEEE Trans. Image Process.* **2021**, *30*, 5072–5084. [[CrossRef](#)] [[PubMed](#)]
16. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNET: Pixel-Wise Voting Network for 6dof Pose Estimation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 15–21 June 2019.

17. Drost, B.; Ulrich, M.; Navab, N.; Ilic, S. Model globally, match locally: Efficient and robust 3D object recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
18. Deng, L. Artificial Intelligence in the Rising Wave of Deep Learning: The Historical Path and Future Outlook. *IEEE Signal Process. Mag.* **2018**, *35*, 177–180. [[CrossRef](#)]
19. Wang, C.; Xu, D.; Zhu, Y.; Martin-Martin, R.; Lu, C.; Fei-Fei, L.; Savarese, S. DenseFusion: 6D object pose estimation by iterative dense fusion. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
20. Braun, M.; Rao, Q.; Wang, Y.; Flohr, F. Pose-RCNN: Joint object detection and pose estimation using 3d object proposals. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, 1–4 November 2016.
21. Choi, C.; Christensen, H.I. 3D pose estimation of daily objects using an RGB-D camera. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012.
22. Liu, D.; Arai, S.; Miao, J.; Kinugawa, J.; Wang, Z.; Kosuge, K. Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic bin picking. *Sensors* **2018**, *18*, 2719. [[CrossRef](#)]
23. Vidal, J.; Lin, C.Y.; Lladó, X.; Martí, R. A method for 6D pose estimation of free-form rigid objects using point pair features on range data. *Sensors* **2018**, *18*, 2678. [[CrossRef](#)] [[PubMed](#)]
24. Ruel, S.; English, C.; Anctil, M.; Church, P. 3DLASSO: Real-time pose estimation from 3D data for autonomous satellite servicing. In Proceedings of the Proc. ISAIRAS 2005 Conference, Munich, Germany, 5–8 September 2005.
25. Mérigot, Q.; Ovsjanikov, M.; Guibas, L.J. Voronoi-based curvature and feature estimation from point clouds. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 743–756. [[CrossRef](#)]
26. Beucher, S.; Lantuejoul, C. Use of Watersheds in Contour Detection. In Proceedings of the International Workshop on Image Processing, Rennes, France, 17–21 September 1979.
27. Braeger, S.; Foroosh, H. Curvature augmented deep learning for 3D object recognition. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; IEEE: Piscataway, NJ, USA, 2018.
28. Tong, L.; Ying, X. 3D Point Cloud Initial Registration Using Surface Curvature and SURF Matching. *3D Res.* **2018**, *9*, 1–16. [[CrossRef](#)]
29. Nguyen, A.; Le, B. 3D point cloud segmentation: A survey. In Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics, RAM—Proceedings, Manila, Philippines, 12–15 November 2013.
30. Selvarasu, N.; Nachiappan, A.; Nandhitha, N.M. Euclidean Distance Based Color Image Segmentation of Abnormality Detection from Pseudo Color Thermographs. *Int. J. Comput. Theory Eng.* **2010**, *2*, 514. [[CrossRef](#)]
31. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
32. Wang, Z.; Wang, E.; Zhu, Y. Image segmentation evaluation: A survey of methods. *Artif. Intell. Rev.* **2020**, *53*, 5637–5674. [[CrossRef](#)]
33. Xiao, J.; Adler, B.; Zhang, H. 3D point cloud registration based on planar surfaces. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Hamburg, Germany, 13–15 September 2012.
34. Mian, A.; Bennamoun, M.; Owens, R. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *Int. J. Comput. Vis.* **2010**, *89*, 348–361. [[CrossRef](#)]
35. Sun, J.; Zhang, J.; Zhang, G. An automatic 3D point cloud registration method based on regional curvature maps. *Image Vis. Comput.* **2016**, *56*, 49–58. [[CrossRef](#)]