IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 October 2023, accepted 17 October 2023, date of publication 26 October 2023, date of current version 7 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327787

RESEARCH ARTICLE

6D Pose Estimation Method Using Curvature-Enhanced Point-Pair Features

YUFAN LIU[1] **AND SHOUTING FENG**[2]
[1]School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006, China
[2]School of Electronic and Information Engineering, South China Normal University, Foshan 528225, China

Corresponding author: Shouting Feng (fst@scnu.edu.cn)

ABSTRACT Pose estimation has garnered significant attention in recent years and has found extensive application in fields such as autonomous driving, robotics, and augmented reality. In the current research, point cloud recognition algorithms based on point-pair-features have been shown to be effective in recognizing objects and pose estimation, but redundant points included in the characterization of model features degrade the recognition performance and computational efficiency of the algorithms. To address this issue, this paper introduces curvature features to filter out unnecessary points and enhance the expression of model features. The resulting global model description is stored in a hash table, and the estimated pose is obtained through the combination of curvature-weighted voting and the Iterative Closest Point (ICP) algorithm for optimization. Additionally, a background removal technique is proposed for fixed usage scenarios, which significantly improves operational efficiency in real-world situations. Experimental results using various datasets and real environments demonstrate that the proposed approach reduces redundancy, improves point-pair feature (PPF) expression, and enhances recognition rate and matching speed by 4.7% and 46.7%, respectively.

INDEX TERMS Pose estimation, point pair features, curvature, point cloud.

I. INTRODUCTION

3D object recognition is a prominent research area in the field of computer vision, with applications in robotics, virtual reality, medical diagnosis, and other fields. With the advancement of depth measurement sensors, the accuracy of 3D data acquisition has significantly improved, leading to increased attention towards 3D object recognition and 6D pose estimation. The primary objective of 3D object recognition and 6D pose estimation is to accurately identify target objects in a scene and determine their poses using rotations and translations in three-dimensional coordinate systems. These technologies play a crucial role in intelligent manufacturing and industrial robot picking operations. Over the past two decades, extensive research has been conducted on algorithms for 3D object recognition and 6D pose estimation, resulting in the proposal of various pose estimation methods, including correspondence methods, template matching-based methods, feature descriptor-based methods, voting-based methods, and deep learning-based methods. However, despite these advancements, 3D object recognition and 6D pose estimation still face challenges in complex scenes characterized by occlusion, noise, and other disturbances.

There are two implementation methods available for establishing correspondence between input data and the complete 3D object model. The first method utilizes 2D images to establish correspondence between 2D points and 3D points, while the second method is based on 3D point clouds [1]. The 2D-3D method typically requires objects with rich textures and the projection of the model from multiple angles to establish the connection between the template image and the RGB image in a single-angle image. The Perspective-n-Point (PnP) algorithm is employed to determine the pose of the current perspective view. This approach is robust for objects with overlapping features but is not suitable for non-textured objects. Feature descriptors can be generated by extracting feature points from 3D point clouds [2]. These descriptors utilize point coordinates, normal vectors, and curvature features to describe the geometric characteristics of the

The associate editor coordinating the review of this manuscript and approving it for publication was Gongbo Zhou.

points. However, determining the correspondence between 3D feature points and 3D models, as well as computing their feature expressions, requires significant computational resources.

The template-based method has emerged as a result of advancements in 2D computer vision. It leverages prior knowledge from templates to evaluate the posture of corresponding scenes and target objects. The template matching algorithm [3], [4], [5] combines the gradient information of the color image with the normal vector of the object surface to establish a template. One of the most well-known methods in this category is linemod [6]. It utilizes the gradient information of the RGB image, which provides strong expressiveness and resilience against backgrounds, noise, and lighting variations [7]. However, its accuracy is moderate, and the effectiveness of its templates and matching scenes is influenced by the quality of the data. Additionally, it is not suitable for application in obstructed scenes.

Deep learning algorithms, distinguished by their potent feature extraction capabilities, have recently achieved state-of-the-art performance in the realm of computer vision [8]. Numerous deep learning-based pose estimation methods have been proposed [9], [10], [11], [12]. For instance, Xiang et al. [13] introduced PoseCNN, an end-to-end 6D pose estimation network. It incorporates convolutional and pooling layers for feature extraction and embeds high-dimensional features in low-dimensional semantic segmentation, 3D translation estimation, and 3D rotation estimation. The team also introduced the YCB-Video dataset, a comprehensive RGB-D dataset, particularly valuable for deep learning frameworks requiring substantial training data. Thanh-Toan et al. [14] extended the Mask R-CNN instance segmentation network by incorporating a novel pose estimation branch that represents rotational regression using Lie algebra. This approach yields a differentiable regression loss and leads to faster inference speeds compared to other multi-stage pose estimation methods. Sock et al. [15] proposed an innovative strategy to enhance environmental comprehension and pose stability. They accomplish this by capturing new scenes from alternative viewpoints through camera movement when the target pose cannot be adequately determined from a single viewpoint. To plan camera movement, they employ perspective entropy and assess various benchmark methods, including LCHF [16], sparse auto-encoder [17], among others. These experiments consistently demonstrate significant accuracy improvements when transitioning from a single viewpoint to six viewpoints. While deep learning methods have progressively outperformed traditional techniques in terms of pose estimation accuracy, it is important to note that they demand a substantial amount of data and computational resources, which can limit their applicability compared to traditional methods.

Voting methods in 6D pose estimation can be categorized into two methodologies: direct and indirect voting. The core objective of the voting technique is to select effective features for precise pose estimation. Indirect voting entails casting votes for feature points, aligning each pixel or 3D point with either 2D-3D or 3D-3D correspondences. For instance, Peng et al. [10] introduced the Pixel-wise Voting Network (PVNet), which employs Random Sampling Consensus (RANSAC) to identify key points. This method subsequently achieves accurate pose estimation through Perspective-n-Point algorithms. However, it is important to note that PVNet relies on 2D target objects and is less suitable for objects lacking self-similarity and texture information. On the other hand, direct voting involves directly casting votes for specific 6D object coordinates or poses at each pixel or 3D point. Drost et al. [18] introduced a novel four-dimensional point pair feature to represent objects. The final result is obtained through modeling object point pair features, computing scene point pair matches, and generating candidate votes. This point pair feature effectively combines the advantages of global and local feature representations, facilitating efficient object feature description.

The PPF algorithm [18] has showcased exceptional object recognition capabilities across various 3D datasets [19]. Due to its outstanding performance, several authors have sought to enhance and expand upon the PPF algorithm [20], [21], [22], [23], [24], [25], [26], [27], [28]. However, a common challenge shared by both the PPF algorithm and its variants is the high computational complexity inherent in the algorithm. This complexity arises from the necessity to extract and store all point pair features of the model in a hash table. Consequently, when dealing with a large number of input data points, this approach can result in a significant surplus of redundant features, leading to a notable decline in recognition performance. Additionally, the presence of complex scenes and background noise can further impact the algorithm's effectiveness. Moreover, the process of reducing redundancy carries the risk of inadvertently eliminating critical information from the model. To mitigate these limitations, we propose leveraging the curvature information of the point cloud as a feature for the selection of key points within the model. Importantly, this information can be retained during the redundancy reduction process. During recognition, we assign greater weight to the characteristic information of these selected points, influencing the estimation outcome. Furthermore, we incorporate background removal techniques in real-world scenarios to enhance both the accuracy and processing speed of the algorithm.Our main contributions are described as follows. 1)Based on PPF, we introduce a novel method that enhances the accuracy and efficiency of the algorithm by incorporating curvature-based six-dimensional point-pair features, downsampling, and weighted voting. 2)Optimizations have been implemented to improve object recognition and position estimation in static scenes. These optimizations take advantage of background similarity to significantly reduce the number of background points, thereby reducing the processing of irrelevant data and accelerating the matching process. 3)Experimental comparisons

conducted across multiple datasets demonstrate the superior performance and robustness of our proposed algorithm. The remaining sections of this paper are structured as follows. Section II introduces related work. Section III presents the details of the proposed method. Experimental results and discussions are provided in Section IV. Section V concludes the paper.

II. RELATED WORK

In this section, we provide an overview of Point Pair Features (PPF) and its various modifications.

The PPF algorithm initially emerged as a groundbreaking solution for recognizing free-form 3D objects within point clouds [18]. Since its inception, it has garnered significant attention, owing to its exceptional computational speed and its demonstrated capability to perform precise pose estimation, even in challenging and cluttered industrial manufacturing environments.A pivotal development in the evolution of PPF occurred in a seminal work [19], where a standardized approach for rigid 6D pose estimation, leveraging a single RGB-D input image while accommodating different lighting conditions, was introduced. A comprehensive evaluation encompassing 15 distinct methods revealed that the PPF-based approach consistently outperformed competing techniques, including template matching, learning-based methods, and traditional 3D feature-based approaches.Recognizing the potential for enhanced accuracy in the voting process through the inclusion of color information, Choi et al. [20] extended PPF by incorporating RGB features alongside the existing four-dimensional features, resulting in a ten-dimensional color point-pair feature. This augmentation yielded notably improved results, particularly when applied to RGB datasets.Subsequently, another significant enhancement was proposed by [3], which introduced the incorporation of edge points for constructing point pair features. This innovation substantially reduced the number of points required in both the training and matching phases. Impressively, this adapted method exhibited excellent performance, particularly when dealing with objects characterized by multiple planes.Building upon these advancements, Choi [21] further refined the methodology by selectively encoding geometric surface shape and photometric color features, contributing to its continual improvement in the field.

In addition to point-pair feature representation, [4] introduced the Maximum Margin Learning framework. The algorithm selects and ranks features based on task importance, thereby enhancing method performance. Hinterstoisser et al. [22] introduced Hs-PPF, which significantly improved Drost-PPF's performance in handling sensor noise and background clutter with a new PPF descriptor propagation strategy. Wang et al. [23] introduced a novel voting strategy based on Hs-PPF to reduce computational costs, albeit at the expense of a lower recognition rate. Vidal et al. [24] introduced a novel preprocessing step based on PPF, which considers the discriminative value of surface information. Liu et al. [25] introduced PPF-MEAM based on B2B-TL descriptors, employing multiple models to describe the target object. They proposed an effective algorithm to enhance recognition rates and reduce computation time, particularly suitable for objects lacking details, though less applicable to general objects. Birdal and Ilic [26] proposed a multi-view rendering strategy for eliminating hidden or invisible geometry and introduced a sparse voxel-based algorithm utilizing Poisson disk sampling and normal-space sampling to handle global vertex and normal distributions. Their approach has similarities with ours, with the main difference being that our goal is to utilize the feature distribution to reduce redundant point-pair features. However, their method aims to create an unbiased, sparser model. In contrast, Cui et al. [27] proposed Cur-PPF, which employs curvature information of point pairs to enhance feature representation and recognition accuracy. However, redundant point pairs persist, and their saved curvature information values are sensitive to model and scene data accuracy. Ge et al. [28] introduced an effective method for sampling key point pairs to reduce redundancy. In contrast, our approach proposes an adaptable method for screening redundant point pairs through curvature feature distribution calculations, simultaneously enhancing point-pair feature representation and optimizing the voting strategy.

III. PROPOSED METHOD

Previous research [18] has established that the Point Pair Feature (PPF) method stands out as a highly effective approach for estimating object positions. The PPF method operates as a voting algorithm, employing four-dimensional point pair features stored in a hash table to describe and match model features. This method comprises two distinct stages: global modeling and local matching. During the global modeling stage, feature information is extracted from all points and compiled into four-dimensional point pair features stored within the hash table. In the subsequent local matching stage, a reference point is selected from the scene, and other scene points are combined into feature point pairs. These feature point pairs are then matched with the hash table, and potential matching poses are identified through voting. Finally, high-scoring poses are output following pose clustering. Downsampling is typically utilized for preprocessing large input point clouds. However, the uniform downsampling method, which is used in PPF, has its shortcomings. According to [29], contour features are not apparent, and the matching effect in feature prominent parts is unsatisfactory. To address this issue, we propose a new downsampling method, based on curvature, that highlights geometric features during the downsampling process. Section 2.1.2 details this process. Furthermore, [30] has established that distance emerges as the most influential feature in the four-dimensional point pair feature descriptor. In an effort to enhance the representation of point pairs concerning angle variations, we introduce curvature information in Section 2.1.3 to construct a six-dimensional point

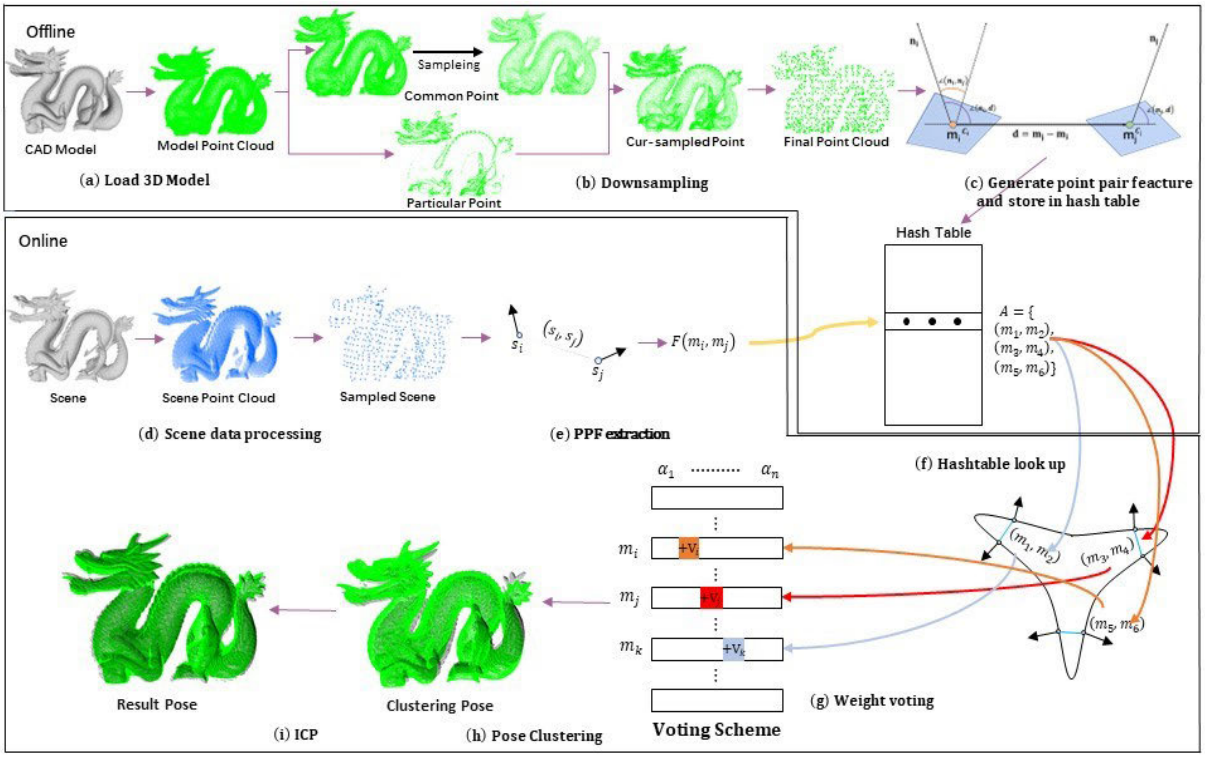


FIGURE 1. The algorithm flow. In the offline stage, the sampled 3D model produces point-pair features, which are subsequently saved within a hash table(a-c). In the online stage, after pre-processing the input scene (d), PPF features are extracted (e). These PPF features are then matched against the hash table to produce attitudes through weight voting and pose clustering (f-h), and finally ICP optimization (i) is performed.

pair feature, thereby augmenting the feature representation capability. Moreover, several studies grounded in the PPF framework have proposed enhancements. For example, [31] has pointed out that model features tend to concentrate around edges and points exhibiting substantial changes. They identify normal vectors and curvature as direct parameters for expressing these features. Consequently, a weighted voting strategy has been implemented to emphasize the role of point pairs with more extensive feature richness in the pose-matching process. Fig. 1 provides an overview of the overall process of our proposed method.

A. OFFLINE STAGE

During the offline stage, a comprehensive representation of the model is constructed and stored within a hash table. The PPF algorithm relies on the characterization of point pairs, acquired through preprocessing the 3D model point cloud of the object, as depicted in Fig. 2. The model point cloud typically contains a substantial volume of data to capture the object's intricate details, resulting in the generation of a considerable number of point pairs. In such instances, numerous nearby points may exhibit identical features to the point pairs generated from a given point, leading to redundant information. To mitigate redundancy, PPF implements spatial downsampling of the model point cloud to expedite processing. However, downsampling also

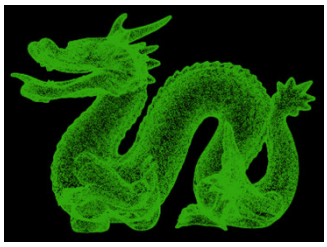


FIGURE 2. 3D model point cloud.

has the potential to eliminate points that carry local model-specific information, thereby significantly impacting the model's expressive capacity. Striking a balance between redundancy reduction and the preservation of crucial feature expressions poses a formidable challenge.

Our objective is to uphold the model's capability to express features while simultaneously reducing the number of the point cloud. To achieve this objective, we introduce a downsampling method based on curvature information. This proposed approach streamlines computational complexity by eliminating redundant points based on curvature distribution, all the while retaining most feature points, even with a substantial downsampling parameter. Notably, this method exhibits favorable results when applied to point cloud models characterized by numerous redundant points. Nonetheless, this technique does lead to a reduction in the count

of generated model description point pairs. To address this limitation and ensure the preservation of the model's features, especially when portions of the object are obscured, we incorporate curvature into the point pair features. This expansion enhances the feature dimension from four to six and fortifies the overall capacity for feature expression. These augmented features are subsequently stored within the hash table, thereby creating a more robust global representation of the model.

1) PREPROCESSING

The preprocessing phase encompasses several essential steps, including point cloud downsampling, normal vector computation, and curvature calculation. Downsampling plays a critical role in handling substantial datasets. This study employs a combination of uniform and curvature-based sampling techniques to address this essential step. Uniform sampling involves partitioning the point cloud into uniformly sized spherical regions and selecting the nearest point to the center within each region. This process effectively reduces the overall number of points while preserving the fundamental shape of the point cloud. Importantly, it retains spatial structural information and ensures that feature expression points remain unaltered, even when combined with ordinary points during curvature sampling.

Curvature serves as a measure of the degree of curvature of geometric objects [32]. We give a 3D point cloud data set $P = \{p_i\}_{i=1}^{N}$, and denote the k-nearest neighbor points of p_i as $N^k(p_i) = p_{ij}\ ,\ 1 \le j \le k$. We also introduce the tangent plane $\mathbf{E}$ to represent point p_i with n_i being the corresponding normal vector.Assuming that the k-nearest neighbor data points p_{ij} define the tangent plane $\mathbf{E}$, we can proceed to fit the plane equation as follows:

$$x_i \cdot cos\alpha + y_i \cdot cos\beta + z_i \cdot cos\gamma = L \tag{1}$$

The values of $cos\alpha, cos\beta, cos\gamma$ represent the cosine of the angles between the normal vector n_i at point p_i. Meanwhile, L represents the distance from the origin to the plane. Equation (1) can be expressed as follows:

$$a \cdot x_i + b \cdot y_i + c \cdot z_i = d\ (d \ge 0)\,, \quad a^2 + b^2 + c^2 = 1 \tag{2}$$

The equation of the plane $\mathbf{E}$ is estimated using the least squares method: $\mathbf{E} : r_i \cdot n_i - d = 0$, where r_i represents the curvature radius of point p_i, and the fitting error of the plane is calculated as:

$$min(\sum_{j=1}^{k} (p_{ij} \cdot n_i - d)^2) \tag{3}$$

To calculate the minimum value of the least squares, $d = |e^T\ (p_i - \bar{p_i})|$. Eigenvalue analysis of the local neighborhood covariance matrix has been demonstrated to be an effective method for estimating local surface properties, as evidenced by previous studies (e.g. [33], [34]). This problem can be resolved by the following.

$$C_i = \begin{bmatrix} p_{i1} - \overline{p_i} \\ \cdots \\ p_{i2} - \overline{p_i} \\ p_{ik} - \overline{p_i} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} p_{i1} - \overline{p_i} \\ \cdots \\ p_{i2} - \overline{p_i} \\ p_{ik} - \overline{p_i} \end{bmatrix}, i_j \in N^k(p_i) \tag{4}$$

The matrix C_i is a symmetric positive semi-definite one, where $\overline{p_i}$ represents the barycenter of the point set $N^k\ (p_i)$, and the eigenvalues $\lambda_1,\ \lambda_2,\ \lambda_3$ are calculated. The curvature of a point can be determined using the following equation [35]:

$$\sigma_n(p) = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3) \tag{5}$$

2) CURVATURE SAMPLING

Objects that require 3D information for pose estimation usually consist of complex surfaces.. In the case of PPF pose estimation, the distribution of normal vectors plays a critical role, and regions exhibiting high curvatures are associated with large changes in normal vectors [30]. Curvature denotes the degree of concavity and convexity within a 3D environment, with elevated curvature values signifying heightened surface variability, and diminished values indicating a predisposition toward planarity. Within 3D models, regions exhibiting higher curvature values are considered to offer a more precise representation of object features. Consequently, we introduce a novel approach to sample feature points, wherein points characterized by high curvature values are designated as feature points, while those with lower curvature values are classified as ordinary points.

The frequently employed voxel down-sampling technique entails dividing the point cloud into spatially uniform grids of equal dimensions and subsequently determining the centroid of the points contained within each grid. While this method serves as an effective low-pass filter, it has the tendency to eliminate edge information within voxel grids. To address the issue of filtering out high curvature points due to this sampling method, we implement a strategy that entails classifying the points into two categories: feature points and ordinary points.

$$p_i = \begin{cases} Particular,\ cur_i < Cur_{thre} \\ Common,\ cur_i \ge Cur_{thre} \end{cases} \tag{6}$$

In our approach, we initially preserve the feature points after the point cloud segmentation process. We then proceed to uniformly sample the ordinary points separately. This distinct sampling of ordinary points serves to minimize redundancy while retaining the fundamental structure of the model. Subsequently, we merge these ordinary points with the previously segmented feature points. As a result, the resulting 3D point cloud dataset comprises a profusion of feature points that effectively convey the model's overarching framework, alongside a subset of ordinary points that collectively form continuous surfaces. This comprehensive dataset description ensures a thorough representation of the

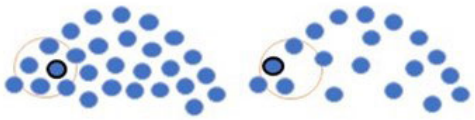
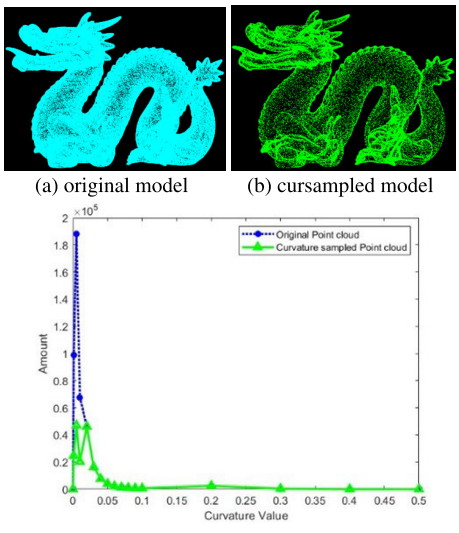


FIGURE 3. Curvature-sampling comparison.

model. As demonstrated in Fig. 3, the uniformly sampled ordinary point cloud exhibits lower density compared to the feature point cloud. Consequently, we execute a secondary step of uniform sampling targeting the ordinary points. This additional point sampling strategy is designed to ensure that the sampling space housing the feature points contains a greater concentration of feature points relative to ordinary points. This approach prevents the inadvertent loss of feature-related information during the sampling process.

In our research, a significant aspect involves the establishment of both the sampling threshold and the sampling ratio for the ordinary point cloud. As depicted in Fig. 4, this figure showcases the curvature distribution of the feature points that exceed the defined threshold. Furthermore, it illustrates the merged model point cloud after the ordinary point sampling, which is compared with the curvature histogram of the original unsampled model. This sampling approach serves a dual purpose: it eliminates redundant points that do not significantly affect the overall model description, while simultaneously enhancing the importance of feature points within the model representation. This augmentation contributes to improved recognition precision and efficacy. Detailed insights into the two hyperparameters governing this sampling method will be provided in the experimental section.



(a) original model (b) cursampled model

(c) histogram

FIGURE 4. The original model point cloud, the sampled point cloud, and their corresponding curvature histograms.

3) SIX-DIMENSIONAL POINT PAIR FEATURES WITH CURVATURE

The standard PPF comprises four dimensions, with three of them conveying angle information but lacking surface variation details. To address this limitation, we have incorporated curvature data, resulting in the construction of six-dimensional point pair features. This augmentation significantly enhances the capacity for expressing information and provides additional matching data during the identification process, consequently stability and robustness. Consider two points m_i and m_j within 3D point cloud data, we generate a point pair. Here $\mathbf{n_i}$ and $\mathbf{n_j}$ represent the normal vectors associated with each point while d represents the directional vector between m_i,m_j. The angles $\angle(\mathbf{n_i}, d)$, $\angle(\mathbf{n_j}, d)$ and $\angle(\mathbf{n_i}, \mathbf{n_j})$ represent the vector angles between these entities, with angles falling within the range $(0, \pi)$. We denote the Euclidean distance between the two points as $||D||$. Additionally, C_i, C_j represent the curvature of each point. Fig. 5 provides a visual representation of the schematic diagram of the point pair feature descriptor with curvature, which is defined as follows:

$$F(m_i, m_j) = (\angle(\mathbf{n_i}, d)\ \angle(\mathbf{n_j}, d)\ \angle(\mathbf{n_i}, \mathbf{n_j})\ ||D||\ C_i\ C_j) \tag{7}$$

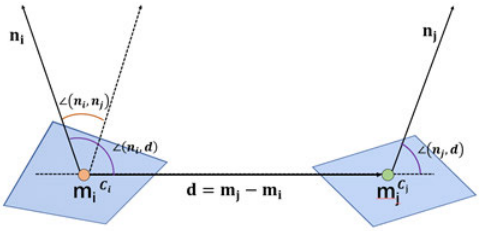


FIGURE 5. Schematic diagram of point pair feature.

In the Cur-PPF [27] approach, point pairs are extended to six-dimensional characteristics by incorporating curvature data. However, their hash table feature storage method introduces a curvature step size. Unlike stable model data, the quality of acquired scene point clouds is susceptible to fluctuations due to equipment variations and environmental factors, leading to errors in curvature calculations. Furthermore, the curvature values of different points may span several orders of magnitude, complicating matching efforts. Therefore, our proposed technique for constructing six-dimensional point pair features avoids the sub-division and classification of curvature information during feature description storage. This approach enhances the stability of curvature-based features, making them less sensitive to variations in curvature values and better suited for real-world scene point cloud data.

4) HASH TABLE

Following the sampling of the model's point cloud, we compute individual point pair features denoted as F before storing the paired-point data that exhibits similar characteristics within the hash table space, as depicted in Fig. 6. The hash table serves as a comprehensive representation of the 3D model. During the online phase, the computation of the

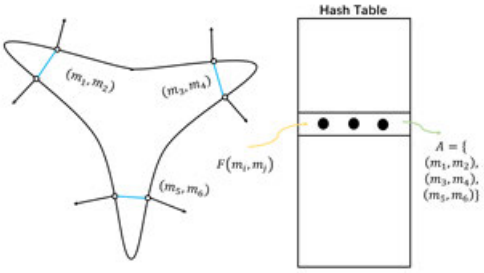


FIGURE 6. Hash table storage of point pair features.

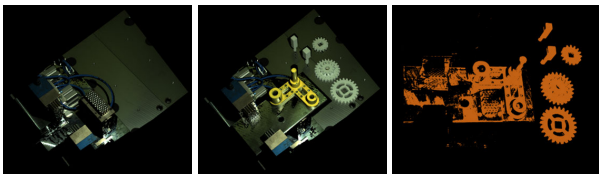


FIGURE 7. Background point cloud, scene point cloud, and point cloud obtained after removing the background.

key value for the scene's point pair feature $F(s_i, s_j)$ in the hash table leads to the identification of the corresponding model point pair (m_i, m_j). To mitigate complexity arising from the quality of generated point clouds affecting curvature, we determine the key value solely based on distance and angle features. Subsequently, in the feature-matching process, point pairs that share the same key value in the hash table undergo additional scrutiny for matching curvature information. The asymmetry in the expression of point pair features is primarily attributed to the introduction of curvature data. A point pair match is deemed successful if the discrepancy in point curvature is less than 10% of the value, thereby eliminating redundant point pairs and enhancing overall accuracy.

B. ONLINE STAGE

In the online stage, we apply similar preprocessing operations as those used for the model point cloud after acquiring the scene point cloud. This involves segmenting and downsampling the stationary scene to remove any invalid background point clouds. Subsequently, we calculate point pair features. Following this preprocessing, we conduct weighted voting matching against the hash table generated during the offline stage. This allows us to ascertain the transformation matrix for point pairs between the scene and model. To refine the final output, we integrate the ICP algorithm [37].

1) SCENE PROCESSING

Scene preprocessing in the online stage mirrors that of the offline phase. After acquiring the scene point cloud data, we calculate the PPF with curvature information and utilize voting matching to determine posture. In industrial scenarios involving 6D pose estimation, depth sensors are usually affixed to stationary scenes, responsible for detecting and estimating newly introduced objects within the scene. The processing of extensive background information demands additional computational resources. Consequently, we employ background point cloud segmentation methods during this stage.

Initially, we acquire the point cloud E, which does not contain objects, using a 3D sensor. For E we define $e_i \in E$. Next, we obtain the point cloud S of the scene with the object that requiring recognition, where $s_j \in S$, as shown in Fig. 7(a) and 7(b). We subsequently search for the K nearest neighbors of s_j in E, with the point distance adhering to:

$$d_j = \sqrt{\left(s_j.x - e_i.x\right)^2 + \left(s_j.y - e_i.y\right)^2 + \left(s_j.z - e_i.z\right)^2} \quad (8)$$

In this point cloud segmentation method, we incorporate the points from the scene into the background point cloud. We utilize a KD-tree structure to accelerate the search for points that are close to s_j within a distance of d_{thre} from the background point cloud E. If the number of points within this distance exceeds a predefined threshold n_{thre}, these points are identified as the background points B_j.

$$B_j = \sum_{j=1}^{N} \left\{ s_j,\ num\left\{KNN\left(s_j\right)\right\} < n_{thre},\ d_j < d_{thre} \right\},\ B_j \in S \quad (9)$$

Upon completing the computation for the scene, we remove the background from the scene point cloud, retaining only the object, as illustrated in Fig. 7(c). Simultaneously, we employ noise filtering to efficiently extract the target within the scene while minimizing unnecessary computations.

2) WEIGHTED VOTING STRATEGY

Within the scene point cloud, we calculate the Key value associated with point pair features (s_i, s_j) and cross-reference it with the hash table. When the values match, we can consider point s_i as a matching model point m_i. Afterward, we relocate points s_i and m_i the origin of the local coordinate system, aligning the normal vector with the positive X-axis. Subsequently, a rotation around the X-axis facilitates the alignment of matching model and scene point pairs. The transformation relationship between corresponding point pairs is represented by the following matrix:

$$s_i = T_{s\to g}^{-1} R_x\left(\alpha\right) T_{m\to g} m_i \quad (10)$$

$T_{s\to g}$ denotes the coordinate system of the reference point in the scene after translation and rotation. $T_{m\to g}$ represents the coordinate system of the model reference point after translation and rotation. The rotation transformation matrix for rotation around the x-axis is depicted as $R_x\left(\alpha\right)$, whereby α signifies the angle of rotation in the positive half-axis direction. The next step involves the selection of the optimal transformation matrix to determine the object's pose using the rotation angle α, obtained through generalized Hough voting. During this voting process, a two-dimensional accumulator

is created, with rows and columns representing the number of model points and N_{angle} rotation angle steps, respectively. If point pair (m_i, m_j) matches a scene pair, a vote is cast for α. Unlike conventional PPF, the point pair characteristics in this paper incorporate curvature information, and the number of pose votes is adjusted based on curvature and normal vectors. Specifically, we reduce the weight of point pairs with narrow angles between their normal vectors and low curvature. Such point pairs typically comprise two points within a similar zone, having a flat surface having many points, or perhaps even a plane. These pairs provide minimal information, contributing little to key features for pose calculation and potentially leading to inaccuracies in pose estimation. Consequently, we decrease the voting weight for such point pairs, as expressed below:

$$V = \delta \cdot \left(1 - \angle\left(s_i, s_j\right)/\pi\right), \ \angle\left(s_i, s_j\right) < \theta, \ C_i, C_j < \eta \quad (11)$$

δ represents the weight coefficient, while the angle and curvature threshold values are denoted by θ and η respectively. The voting process is designed to reduce the weight of point pairs providing weak information, leading to a decrease in the total number of votes. Conversely, point pairs with prominent features require an increase in voting weight. When describing model point pairs, those containing feature-expressing points inherently carry more information. Therefore, we assign relative weights to point pairs based on their curvature information, as the curvature values for each point can vary significantly. We divide these values into several interval sets, each with a specific coefficient. Weighted voting is performed based on the sum of the coefficients of the point pairs, and the formula can be expressed as follows:

$$P = \sum_{i=1}^{k} set_i \quad (12)$$

$$V = V + 0.1 \cdot \left(w\left(s_i\right) + w\left(s_j\right)\right), \ s_i \in p_i, \ s_j \in p_j \quad (13)$$

Within the algorithm, we divide the curvature values into K intervals, ranging from small to large. The coefficient for each interval of point curvature is denoted as $w\left(s_i\right)$. By assigning these weights, point pairs with more informative curvature values receive a higher number of votes, which ultimately enhances the accuracy of pose estimation. After weighting all the scene points through the voting process, we collect multiple pose votes.

3) POSE CLUSTERING

In situations where the reference point is located on the surface of the object, multiple valid point pairs are generated, triggering feature matching and resulting in vote counts. Once the pose with the highest number of votes is determined, the process of creating a cluster set commences, where similar poses are grouped together into the same class. Within each class, translational and rotational disparities among all poses fall within a specified threshold range. Upon completing the clustering, the total votes for each class are calculated, and the classes are re-ranked based on the accumulated votes. The final pose is generated by averaging the poses within the class that received the highest number of votes. In cases where multiple objects coexist within a scene, the possibility of producing multiple high-vote classes arises. However, the preference remains for the class that has gathered the most votes.

4) ICP OPTIMIZATION

The algorithm of choice for point cloud registration is the Iterative Closest Point Algorithm (ICP) [37]. This algorithm identifies corresponding points in the scene point cloud that are close to each point in the model point cloud. Only those points falling within a certain threshold are considered in the iterative point set. To begin, the clustered pose serves as the initial pose, and the distance between the model point cloud and the transformed scene point cloud is calculated iteratively. A comparison is made to determine whether the distance between the model point cloud and the scene point cloud is smaller than the specified threshold. The iterative process concludes when the percentage of matched points within the point set reaches a predetermined level.

IV. EXPERIMENTAL RESULTS

To evaluate the validity of our method, we conducted experiments using a diverse range of datasets, which included public datasets, synthetic datasets, and datasets acquired from real environments. To assess performance, we employed the Average Point Distance (ADD) as the metric for pose error [38]. Specifically, we compared the recognition effectiveness of our proposed algorithm with the commonly used Point Pair Features (PPF) method using the Stanford dataset [39]. Additionally, we evaluated the efficiency of our background segmentation technique and the overall performance of the algorithm by conducting tests on a synthetic dataset designed to emulate real-world environments. Ultimately, we demonstrated the practical reliability and robustness of our method by using real scene data obtained from the Intel D435i sensor. For implementing our algorithm, we utilized the C++ programming language and algorithm libraries such as the Point Cloud Library (PCL) [40] and Eigen on the Linux platform.

A. DATASETS

1) STANFORD DATASET

For our experiment, we employed three point cloud models, namely the Stanford bunny, dragon, and armadillo, all of which were sourced from the publicly accessible 3D model database known as the Stanford 3D Scanning Repository [39]. These models are depicted in Fig. 8. Each model was scanned using a Cyber-ware 3030 MS scanner. Subsequently, each model was scaled to match the scene information from the database. This scaling process ensured that both the models and scene point clouds were at the same scale, rendering them compatible with our algorithm.

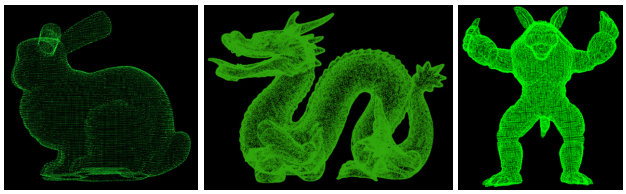


FIGURE 8. Stanford 3D Model.

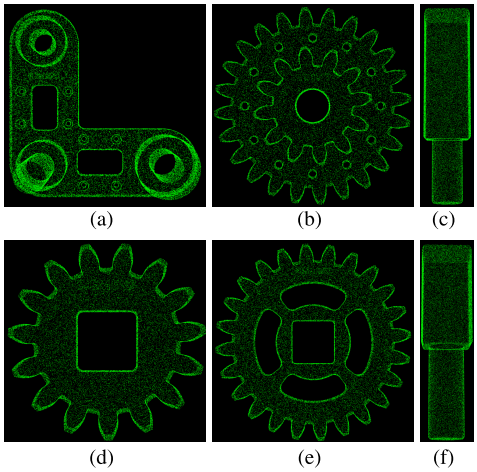


FIGURE 9. 3D Workpiece Model. (a) is the baseplate; (b) (d) and (e) is the gear; (c) and (f) is the shaft.

2) SYNTHETIC DATASET

The Synthetic Dataset includes six common industrial processing models, as illustrated in Fig. 9. These models have sizes ranging from several tens of centimeters. Data for each model were generated across ten different scene settings. In each scene, we defined the initial position and orientation of the object, and initial pose matrix values were randomly sampled, resulting in a new pose and a corresponding 4×4 transformation matrix. This process also generated new scene point cloud data. The scene data and transformation matrix were saved separately, creating synthetic 3D scene data and the ground truth for the pose.

3) REAL SCENE DATASET

The real scene data was collected using an Intel D435i stereo 3D camera, capturing four object models as depicted in Fig. 10. For each model, the object was placed in multiple scenes with varying attitudes, collecting more than five scene data for each setting. Fig. 11 shows the actual data collection.

B. ASSESSMENT CRITERIA

This paper primarily employs ADD [13] as the primary error metric. However, due to the symmetry of some objects in the synthetic dataset, the exact pose determination outcome may remain uncertain. Consequently, we utilize ADD-S as the evaluation standard. In the context of the true pose value, represented as $\bar{P} = (\bar{\mathbf{R}}, \bar{t})$, and the pose estimation value

FIGURE 10. Real objects and 3D models. Including Mario, Luigi, Cole and the box.

FIGURE 11. Real Scene point cloud.

$\hat{P} = (\hat{\mathbf{R}}, \hat{t})$, we obtain the 3×3 rotation matrix $\mathbf{R}$ and the 3×1 translation matrix t. The calculations for ADD and ADD-S are as follows:

$$ADD = \frac{1}{n} \sum_{x \in M} \left\| (\bar{\mathbf{R}}x + \bar{t}) - \left(\hat{\mathbf{R}}x + \hat{t}\right) \right\|_2 \tag{14}$$

$$ADD - S = \frac{1}{n} \sum_{x_1 \in M} \min_{x_2 \in M} \left\| (\bar{\mathbf{R}}x_1 + \bar{t}) - \left(\hat{\mathbf{R}}x_2 + \hat{t}\right) \right\|_2 \tag{15}$$

We consider an estimate as successful when the ADD falls below 10% of the model's diameter, Otherwise, it is a failure. The final recognition rate is expressed as follows:

$$Accuracy = \frac{Number\ of\ matching\ success}{Number\ of\ matching\ data} \tag{16}$$

C. PARAMETRIC ANALYSIS

Our experiments reveal that the manipulation of parameters across different models and scenes significantly impacts the performance of the algorithm. Nevertheless, the parameters employed in this paper adhere to specific rules, with the majority maintaining constant values that yield favorable results across various scenarios. For instance, we have set the downsampling values for point cloud models and scene data to 0.01 times the scale unit of the model. Consequently, for a model with an approximate scale of 10 cm, the downsampling parameter equates to 1 mm. Subsequently, we have adjusted other pertinent parameters based on this rule. The nearest neighbor distance utilized for calculating curvature radius is established at 0.05 times the scale unit of the model. Additionally, we sample the threshold for curvature sampling from the top 20% of curvature values, set the normal point sampling ratio at 25%, and employ the random sampling method. Our experimental results corroborate that this configuration effectively balances the distribution of feature points and normal points across most models. We have stored the distance and angular step size

in the hash table as 0.5 mm and 10°, respectively, primarily classifying the data based on the distance between point pairs. For the number of weighted votes, we consider it solely as a supplementary tool for capturing partial feature information, expressing the number of votes as the coefficient multiplied by 0.1.

D. EXPERIMENT RESULT

1) STANFORD DATASET

We present comprehensive estimation results for three objects in Table 1, wherein we conducted a comparative analysis between the original PPF method and our proposed methods: curvature sampling, curvature-based point pair feature, and a combination of curvature sampling and point pair feature. As depicted in Table 1 and 2, our curvature sampling method, as proposed, notably reduces the requisite number of point pairs for feature description, consequently leading to a significant reduction in matching time. Furthermore, the higher proportion of feature representation point pairs within the overall point pairs results in enhanced recognition rates. The incorporation of curvature into the point pair feature enriches the model's point pair representation, thereby outperforming the conventional PPF method. The inclusion of curvature voting weight further enhances matching accuracy. Likewise, following model curvature sampling, we observe improvements in matching speed and accuracy, as evidenced in Fig. 12. By incorporating all the improvement strategies, we were able to attain a significant enhancement in both the recognition rate and processing speed, achieving a 6.5% increase in recognition rate and a substantial 44% boost in processing speed.

TABLE 1. Recognition rate of the algorithms for the Standford dataset.

Methods	Bunny	Armadillo	Dragon	Average
PPF	87.98%	90.90%	89.68%	89.52%
sampled	90.01%	92.35%	92.00%	91.45%
Cur	91.50%	3.19 %	92.13%	92.27%
Cur(w)	91.74%	95.46%	94.02%	93.74%
Cur(w)+sampled	**94.54%**	**96.90%**	**96.67%**	**96.03%**

TABLE 2. Recognition speed of the algorithms for the Standford dataset.

Methods	Bunny	Armadillo	Dragon	Average
PPF	3.20s	2.29s	2.57s	2.69s
sampled	2.00s	1.79s	1.97s	1.92s
Cur	1.60s	1.74s	1.76s	1.70s
Cur(w)	1.57s	1.66s	1.75s	1.66s
Cur(w)+sampled	**1.49s**	**1.35s**	**1.62s**	**1.49s**

2) SYNTHETIC DATASET

In this section, we present our utilization of synthetic datasets for simulating pose recognition scenarios. Here, we conducted a comparative analysis involving the original PPF method, the background removal method, and our proposed method in combination with background removal to assess their performance. Table 3 and 4 provide a clear demonstration of these results. The original method attained an accuracy of 89.22%, but it required a significant amount of time, approximately 60.4 seconds, to process the dataset. This extended processing time proved cumbersome, particularly when dealing with high-resolution background scenes. Upon implementing background removal, as illustrated in Fig. 13, a substantial reduction in redundant information was achieved, resulting in an accuracy of 95.37% and a processing time of only 2.44 seconds. Building upon the foundation of background removal, our proposed method further improved the performance, achieving an impressive accuracy of 98.23% with a significantly reduced processing time of just 1.27 seconds. These enhancements correspond to an improvement of 2.9% in recognition rate and an impressive 48% increase in processing speed, respectively. This not only enhances efficiency but also accuracy when working with synthetic datasets.

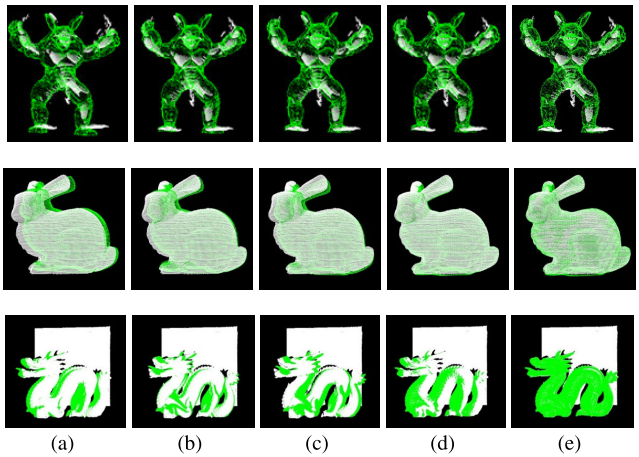


FIGURE 12. Each column corresponds to the actual effect of the method used in Table 1 and 2. White represents the data set and green represents the pose estimation results.

TABLE 3. Recognition rate of the algorithms for the Synthetic dataset.

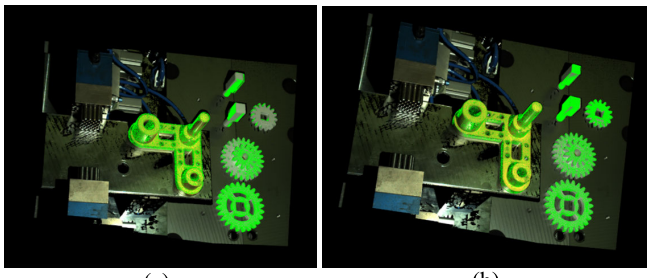
Methods	Baseplate	Gear	Shaft	Average
PPF	85.99%	91.22%	87.82%	89.22%
PPF+bgrm	95.76%	95.91%	94.38%	95.37%
OurPPF+bgrm	**99.71%**	**98.23%**	**97.48%**	**98.23%**

TABLE 4. Recognition speed of the algorithms for the Synthetic dataset.

Methods	Baseplate	Gear	Shaft	Average
PPF	116.0s	50.27	47.8s	60.4s
PPF+bgrm	4.69s	2.33s	1.48s	2.44s
OurPPF+bgrm	**1.97s**	**1.4s**	**0.8s**	**1.27s**

3) REAL SCENE

To assess the performance of our proposed method in real-world scenarios, we employed the Intel D435i RealSense stereo camera to capture point cloud data. Background data was recorded in the absence of any recognition objects, facilitating subsequent background removal processes. Since



(a) (b)

FIGURE 13. Background-removed PPF method (a) and the effect of our proposed method (b) on synthetic datasets.

real pose data was not readily available, we quantified the average distance between each model and the nearest environmental point, which we denoted as ADD-R. The verification of ADD-R values was conducted manually.

$$ADD-R=\frac{1}{n}\sum_{x_1\in M}\min_{x_2\in S}\left\|\left(\hat{\mathbf{R}}x_1+\hat{t}\right)-x_2\right\|_2 \tag{17}$$

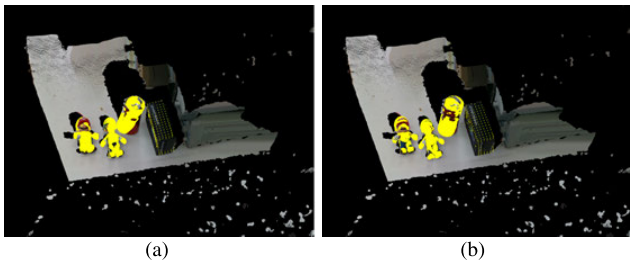
Table 5 and 6 provide an overview of the recognition performance and matching time in real-world scenarios. Notably, our proposed method exhibited superior performance compared to the original PPF method, achieving a recognition rate of 92.08% with a matching time of 1.30 seconds, surpassing the original PPF method's recognition rate of 87.48% and matching time of 2.51 seconds. The recognition rate experienced a notable increase of 4.6%, while the processing speed improved significantly by 48%. Fig. 14 visually present the identification outcomes of both the original method and our proposed method.

TABLE 5. Recognition rate of the algorithms for the Real scene.

Methods	Mario	Luigi	Cola	Box	Average
PPF	91.54%	93.34%	93.02%	92.53%	92.61%
OurPPF	**96.59%**	**96.18%**	**97.56%**	**96.27%**	**96.65%**

TABLE 6. Recognition speed of the algorithms for the Real scene.

Methods	Mario	Luigi	Cola	Box	Average
PPF	1.48s	1.70s	1.83s	1.60s	1.65s
OurPPF	**1.17s**	**1.38s**	**1.39s**	**0.81s**	**1.19s**



(a) (b)

FIGURE 14. PPF method (a) and our proposed method (b) in real scene recognition effect.

V. CONCLUSION

We present a 6D pose estimation method founded on curvature-enhanced point pair features. This method introduces an efficient point cloud preprocessing sampling technique aimed at representing the model through curvature features, effectively addressing the redundancy inherent in the original PPF method. Our proposed approach substantially enhances model representation capabilities, matching accuracy, efficiency, and overall robustness compared to conventional techniques. Furthermore, the incorporation of curvature features into the point-to-point feature-enhanced description not only bolsters overall accuracy and efficiency but also augments these attributes. Weighted voting and background removal further contribute to the precision and efficiency of pose estimation. Through comprehensive evaluations using both publicly available datasets and real-world data, our method surpasses the original PPF method, boasting a 4.7% increase in accuracy and a remarkable 46.7% reduction in matching time. These results highlight the advantages of our method and underscore the pivotal role played by curvature information in the representation of 3D model features. It's worth noting that computational performance is contingent on the quality and accuracy of point cloud data acquisition. Therefore, future research endeavors will explore the potential to achieve commendable performance with data of varying quality. Additionally, further investigations are warranted to ensure stability in real-time applications across more intricate scenarios.

REFERENCES

[1] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1677–1734, Mar. 2021.

[2] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.

[3] C. Choi, A. J. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Jul. 2013, pp. 1568–1575.

[4] O. Tuzel, M.-Y. Liu, Y. Taguchi, and A. Raghunathan, "Learning to rank 3D features," in *Computer Vision–ECCV*. Zurich, Switzerland: Springer, Sep. 2014, pp. 520–535.

[5] T. Birdal and S. Ilic, "Point pair features based object detection and pose estimation revisited," in *Proc. Int. Conf. 3D Vis.*, Oct. 2015, pp. 527–535.

[6] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 80–91, Sep. 2012.

[7] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypotheses verification method for 3D object recognition," in *Computer Vision–ECCV*, Florence, Italy: Springer, Oct. 2012, pp. 511–524.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2016, pp. 770–778.

[9] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3343–3352.

[10] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.

[11] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7668–7677.
[12] 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 *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.
[13] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*.
[14] T.-T. Do, M. Cai, T. Pham, and I. Reid, "Deep-6DPose: Recovering 6D object pose from a single RGB image," 2018, *arXiv:1802.10367*.
[15] J. Sock, S. H. Kasaei, L. S. Lopes, and T.-K. Kim, "Multi-view 6D object pose estimation and camera motion planning using RGBD images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jul. 2017, pp. 2228–2235.
[16] A. Tejani, R. Kouskouridas, A. Doumanoglou, D. Tang, and T.-K. Kim, "Latent-class Hough forests for 6 DoF object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 119–132, Jan. 2018.
[17] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6D object pose and predicting next-best-view in the crowd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3583–3592.
[18] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.
[19] T. Hodan, "BOP: Benchmark for 6D object pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.
[20] C. Choi and H. I. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3342–3349.
[21] C. Choi and H. I. Christensen, "RGB-D object pose estimation in unstructured environments," *Robot. Auto. Syst.*, vol. 75, pp. 595–613, Jan. 2016.
[22] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *Computer Vision–ECCV*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 834–848.
[23] G. Wang, L. Yang, and Y. Liu, "An improved 6D pose estimation method based on point pair feature," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 455–460.
[24] J. Vidal, C.-Y. Lin, and R. Martí, "6D pose estimation using an improved method based on point pair features," in *Proc. 4th Int. Conf. Control, Autom. Robot. (ICCAR)*, 2018, pp. 405–409.
[25] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, "Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic bin picking," *Sensors*, vol. 18, no. 8, p. 2719, Aug. 2018.
[26] T. Birdal and S. Ilic, "A point sampling algorithm for 3D matching of irregular geometries," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6871–6878.
[27] X. Cui, M. Yu, L. Wu, and S. Wu, "A 6D pose estimation for robotic bin-picking using point-pair features with curvature (Cur-PPF)," *Sensors*, vol. 22, no. 5, p. 1805, Feb. 2022.
[28] Z. Ge, X. Shen, Q. Gao, H. Sun, X. Tang, and Q. Cai, "A fast point cloud recognition algorithm based on keypoint pair feature," *Sensors*, vol. 22, no. 16, p. 6289, Aug. 2022.
[29] L. Zhu, W. Chen, X. Lin, L. He, and Y. Guan, "Curvature-variation-inspired sampling for point cloud classification and segmentation," *IEEE Signal Process. Lett.*, vol. 29, pp. 1868–1872, 2022.
[30] L. Kiforenko, B. Drost, F. Tombari, N. Krüger, and A. G. Buch, "A performance evaluation of point pair features," *Comput. Vis. Image Understand.*, vol. 166, pp. 66–80, Jan. 2018.
[31] C. Liu, F. Chen, L. Deng, R. Yi, L. Zheng, and C. Zhu, "6D pose estimation based on edge-enhanced point pair features for surgical navigation," in *Proc. 8th Int. Conf. Virtual Reality (ICVR)*, May 2022, pp. 392–400.
[32] S. Braeger and H. Foroosh, "Curvature augmented deep learning for 3D object recognition," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3648–3652.
[33] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. 19th Annu. Conf. Comput. Graph. Interact. Techn.*, 1992, pp. 71–78.
[34] Y. Asao and Y. Ike, "Curvature of point clouds through principal component analysis," 2021, *arXiv:2106.09972*.
[35] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proc. IEEE Vis.*, Oct. 2002, pp. 163–170.
[36] J. Vidal, C.-Y. Lin, X. Lladó, and R. Martí, "A method for 6D pose estimation of free-form rigid objects using point pair features on range data," *Sensors*, vol. 18, no. 8, p. 2678, Aug. 2018.
[37] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.
[38] T. Hodaň, J. Matas, and Š. Obdržálek, "On evaluation of 6D object pose estimation," in *Computer Vision–ECCV*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 606–619.
[39] (Apr. 6, 2023). *The Stanford 3D Scanning Repository*. [Online]. Available: https://graphics.stanford.edu/data/3Dscanrep/
[40] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

YUFAN LIU received the B.S. degree in electronic information science and technology from South China Normal University, Guangzhou, China, in 2021, where he is currently pursuing the degree in circuits and systems with the School of Physics and Telecommunication Engineering. His main research interests include machine vision, 3D vision, point cloud processing, and target pose estimation.

SHOUTING FENG received the B.S. and Ph.D. degrees from the South China University of Technology (SCUT), in 2002 and 2007, respectively. He is currently a Lecturer with the School of Electronic and Information Engineering, South China Normal University (SCNU). His current research interests include machine vision and image processing.