Recognition and position estimation method for stacked untextured parts

Zhao Zhang*, *Jifang* Wang, *Kaixuan* Guo, and *Kai* Wang

College of Mechanical Electrical Engineering, Beijing Information Science and Technology University, Beijing, China

Keywords: Parts identification, Posture estimation, Deep learning, ICP.

Abstract. To address the challenge of recognizing and estimating the position of untextured stacked parts, which are common in industrial environments, this study proposes an integrated approach that incorporates the YOLOv7 target detection algorithm and point cloud alignment techniques. First, the YOLOv7 algorithm is utilized to quickly identify and locate the 2D position of the part, followed by a mapping technique to transform the 2D region of interest (ROI) into the corresponding 3D point cloud region. In the point cloud processing stage, depth threshold segmentation and Euclidean clustering segmentation methods are used to separate the target part from the background and other interfering objects. The pose estimation stage uses the SAC-IA algorithm for coarse alignment, followed by an improved ICP algorithm that introduces an adaptive weighting mechanism and a global optimization strategy for fine alignment to obtain the final 6D pose of the part. The improved strategy significantly optimizes the point-pair selection and alignment process and enhances the robustness and accuracy of the algorithm. Through experimental validation on publicly available part piece datasets, the results show that the part identification and pose estimation method proposed in this study can realize fast and accurate identification and pose estimation of different shapes, non-textured, and scattered stacked parts, where the position error can reach up to 1mm and the angular error within 1°, which meets the requirements of practical applications.

1 Introduction

With the rapid development of industrial automation and intelligent manufacturing, part recognition and position estimation are increasingly widely used in manufacturing, especially in the fields of automated assembly, quality inspection, and robot vision. Traditional part recognition and position estimation techniques mainly rely on the texture features of the parts, however, in practical applications, many parts lack obvious texture features on the surface, or the texture features on the surface of the parts are not easy to be captured under specific working environments, which poses a great challenge to the

* Corresponding author: zhangzhaoskr@163.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

recognition and position estimation of weakly textured parts. Currently, with the rapid development of computer vision, machine learning and other technologies, it has become a research hotspot to solve the problem of weak texture part recognition and position estimation through these advanced technologies.

2 Related work

The main methods for object recognition and position estimation are template matching method, point cloud alignment method, and machine learning methods that have emerged in recent years. For the template matching method, Hinterstoisser of the Technical University of Munich [1] firstly proposed the LINE2D algorithm based on RGB images, and then proposed the LINEMOD algorithm supporting RGB-D data in 2012 [2, 3]. These algorithms still follow the idea of template matching, only the composition of the template is different from the traditional template. Traditional point cloud alignment algorithms such as ICP and its variants are widely used because of their simplicity and effectiveness. Iteration Closest Point (ICP, Iteration Closest Point) is a classical alignment method proposed by Zhang et al [4], which searches for the closest points of the target point cloud and the model point cloud to form a correspondence each time, and solves for the relative positional attitude by iterative means. In recent years, thanks to the unique advantages of deep learning in 2D image classification, detection and segmentation tasks, using only color image information, Papers [5-7] migrated excellent target detection networks, such as Faster R-CNN and region based fully convolutional network(R-FCN), to the pose estimation problem to achieve the following results The problem of estimating and grasping the position of a simply-placed object in a plane is difficult to adapt to the situation of cluttered background and multiple target objects stacked in a cluttered manner; Su H et al. [8] utilizes CNNs to classify the rendered 3D models by viewpoint angles, thus converting 3D position estimation into a classification task. Despite the significant progress, part recognition and pose estimation still face many challenges, such as occlusion between parts, recognition accuracy in complex backgrounds, and stability under different lighting conditions.

3 Part recognition and position estimation

3.1 Part recognition based on YOLOv7

YOLOv7 [9] was released in 2022 by Chienyao Wang and Alexey Bochkovskiy et al. Compared to previous versions of the YOLO series, YOLOv7 offers a better balance between detection efficiency and accuracy. The YOLO family of algorithms is also the most popular single-stage target detection algorithm, which is widely used because it is fast and suitable for real-time detection tasks. The biggest improvement of YOLOv7 is that the structural reparameterization method in RepVGG network is used in the model testing phase, which fuses the three branches of 1x1 convolution, BN (batch normalization) and 3x3 convolution during training into a single line model, which greatly saves the memory overhead and improves the model speed.In addition, the improvement points of YOLOv7 are the adoption of a more efficient positive sample allocation strategy to obtain more prior frames, and the introduction of the auxiliary head from coarse to fine bootstrap allocation strategy in the Head structure.

Based on the above background, this paper proposes a part identification and position estimation model based on the YOLOv7 algorithm, and through the advantages of the deep learning model, we are able to effectively deal with the problems of stacking, sticking, and

occlusion between the parts, and improve the adaptability and flexibility of the automation system.

3.2 3D point cloud mapping and point cloud segmentation of localized ROI regions

The YOLOv7 target recognition and detection algorithm identifies the part and locates the part in the 2D image, and the detection results are marked with a bounding box sense to indicate the position of the part, which is the local ROI region of the part in the 2D image. In order to accurately estimate the 3D position of the part, it is also necessary to map the 2D ROI to the corresponding 3D point cloud region. This process involves the application of RGB-D cameras, where the RGB camera is responsible for capturing high-quality color images, while a pair of infrared cameras is used to generate depth information and point cloud data, the latter providing the necessary structural information for the 3D space.

In order to realize the accurate mapping from the 2D image space to the 3D point cloud space, it is necessary to construct a transformation model between the RGB camera reference coordinate system and the depth camera reference coordinate system. This transformation relationship between coordinate systems is determined by applying the Zhang Zhengyou calibration method [10], as shown in figure 1. O_{rgb} is the RGB camera reference coordinate system, O_{ing} is the infrared camera coordinate system, according to the calibration method to solve the relationship between the two camera coordinate systems and the calibration plate coordinate system respectively, and get the transformation matrices $\mathrm{rgb}T_{rgb}$ and $\mathrm{cal}T_{inf}$. the transformation relationship between the RGB camera and the infrared camera reference coordinate system is:

$$^{\mathrm{rgb}}T_{inf} = {}^{\mathrm{rgb}}T_{cal} * {}^{\mathrm{cal}}T_{inf} \tag{1}$$

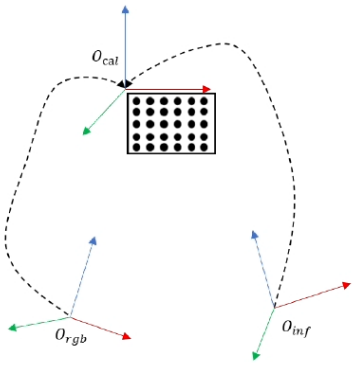


Fig. 1. Schematic diagram of the calibration between the RGB camera and the reference coordinate system of the infrared camera.

After aligning the detection bounding box of YOLOv7 to the depth image, the point cloud data of the localized ROI region can be obtained according to the conversion relationship introduced above, but the point cloud data contains not only the information of the object itself, but also the information of the environmental background and other interfering objects. Therefore, background point cloud segmentation becomes an indispensable step. The specific method is to first customize a threshold value, and constantly adjust to select the appropriate depth threshold, which is used to distinguish the background from the foreground point cloud. During the segmentation process, all points

with depth values less than this threshold are categorized as background points, and the remaining points are regarded as foreground points, i.e., parts to be processed, through which parts can be effectively separated from the complex background. After the successful application of depth thresholding based background point cloud segmentation, a new challenge faced is the separation between parts in an unorganized stacked parts scene. Although the background and parts have been effectively distinguished, the point cloud data obtained from the depth map transformation still contains small portions of point clouds of other parts. These stray point clouds may interfere with the subsequent part identification and position estimation processes. Therefore, a method is needed to further refine the point cloud data to ensure that each part is recognized and processed individually. Object point cloud segmentation methods based on Euclidean clustering are particularly important in this context. The essence of the Euclidean clustering segmentation method lies in utilizing the Euclidean distances between points in the point cloud to classify different objects. The principle of the cluster segmentation method can be briefly described as follows: first, for each point in the point cloud, the Euclidean distance between it and all other points is calculated. Then, if this distance is less than a predetermined threshold, these two points are considered to belong to the same object. This process can be represented by the following distance calculation formula:

$$D(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \tag{2}$$

where D (p,q) denotes the Euclidean distance between points p and q, p_x, p_y, p_z and q_x, q_y, q_z are their coordinates in 3D space, respectively.

3.3 Position estimation based on point cloud alignment

The purpose of point cloud alignment is to find the best correspondence between two sets of point clouds, so as to determine the position of one object relative to another. Among the many point cloud alignment schemes, the use of a sequential alignment scheme, i.e., coarse alignment followed by fine alignment, is a common and effective strategy. The purpose of coarse alignment is to quickly obtain an approximate correspondence between two sets of point clouds, while the fine alignment criterion is dedicated to further refine this correspondence on this basis to achieve higher alignment accuracy. In this study, we choose to adopt Sample Consensus Initial Alignment (SAC-IA) as the coarse alignment step, followed by the refined alignment using the improved Iterative Closest Point (ICP) algorithm.

SAC-IA is an algorithm for coarse alignment of point cloud data, which is based on the idea of Sample Consensus and the initial alignment strategy. The algorithm mainly realizes coarse alignment by iteratively selecting pairs of points in a point cloud, estimating the transformation matrix, and using this transformation matrix to align two sets of point clouds.

3.3.1 Fine alignment based on improved ICP algorithm

Aiming at the limitations of the original iterative closest point (ICP) algorithm in the point cloud alignment process, such as being highly sensitive to the initial estimation and easy to fall into the local optimal solution, this paper proposes an improved ICP algorithm scheme aiming at improving the accuracy and robustness of the point cloud alignment. The improved scheme is based on the introduction of an adaptive weighting mechanism and a global optimization strategy to optimize the point pair selection and alignment process.

The original ICP algorithm achieves alignment by iteratively finding the nearest point pairs between two sets of point clouds and minimizing the distance between these pairs. However, this process is often susceptible to outliers and is highly sensitive to the initial position estimation. To overcome these limitations, we propose the following improvement strategy:

1. Adaptive weighting mechanism: for each pair of point pairs, different weights are assigned according to the size of their distances, and the smaller the distance, the greater the weight of the point pair, and vice versa. This can reduce the influence of anomalies on the alignment results and improve the robustness of the algorithm.

2. Global optimization strategy: the global optimization algorithm Particle Swarm Optimization (PSO) algorithm is introduced during each iteration of ICP to avoid the algorithm from falling into the local optimal solution and to find a more global optimal solution.

Suppose two sets of point clouds are P and Q, where P is the source point cloud and Q is the target point cloud. We define a weight function w(d), with d being the distance between pairs of points. The weight function can be defined as:

$$w(d) = \exp(-\frac{d^2}{2\sigma^2}) \tag{10}$$

where σ is a parameter that controls the width of the function, which can be adaptively adjusted according to the distribution of the point cloud.

In each iteration, for each point p_i in the source point cloud P, find the nearest point q_i in the target point cloud Q, calculate the weighted distance sum between them, and then optimize the following objective function using the global optimization algorithm:

$$\min \sum_{i=1}^{n} w(d_i) \cdot \|(R \cdot p_i + t) - q_i\|^2 \tag{11}$$

where R and t are the rotation matrix and translation vector, respectively, and n is the number of point pairs.

The above objective function is optimized by iterating until the termination condition is satisfied and the maximum number of iterations is reached, so as to obtain the final rotation matrix R and translation vector t, and achieve the improved point cloud fine alignment.

4 Experimental results and analysis

4.1 Data set

The dataset in this paper is derived from the publicly available dataset T-Less [11], which contains 30 industrial electrical parts that do not have a distinct texture, they have only minor differences in color and reflective properties, and they are extremely similar in shape and size. Similar industrial parts are common in industrial environments, from which we selected three representative parts in four different complex scenarios, totaling 3,500 images, as recognition objects.

The point cloud dataset for part position estimation contains a source point cloud dataset and a template point cloud dataset. The source point cloud dataset is the point cloud dataset containing the target part obtained by the method in Section 3.2 by aligning the inspection result map of YOLOv7 to the depth map provided by the T-Less dataset, and then

converted by the camera matrix. The template point cloud dataset is a point cloud dataset that converts the 3D model of the part into .ply format.

4.2 Experimental parameters and evaluation indicators

The experimental environment used was Windows 10 operating system, NVIDIA GeForce RTX3090 GPU, CUDA 11.1, Python 3.9 and PyTorch 1.13.1.

The evaluation metrics of the part recognition experiments are mainly the detection average precision mAP (average precision) and the time performance of the algorithm FPS (frames per second). Among them, mAP is defined as follows:

$$\mathrm{mAP} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}-1}^{\mathrm{n}}\int_{0}^{1}\mathrm{P}(\mathrm{R})\mathrm{dR} \tag{12}$$

Important metrics for evaluating the performance of algorithms in point cloud alignment and position estimation experiments include root mean square error (RMSE) and rotational translation error. The RMSE is a commonly used metric to measure the accuracy of point cloud alignment, which is calculated as the square root of the mean of the sum of the squares of the differences in the corresponding positions of the points between the point clouds after alignment. The rotation translation error, on the other hand, more directly reflects the rotation and translation differences between the source and target point clouds after point cloud alignment. Compared with the single root mean square error, the rotational translation error provides more information about the alignment effect, which makes the performance evaluation more comprehensive and detailed, so the rotational translation error is chosen as the evaluation index in this experiment. The root mean square error is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{N}}\|\mathrm{p_i} - \mathrm{q_i}\|^2} \tag{13}$$

where N is the overview of the corresponding point pairs after alignment, and are the corresponding points in the source and target point clouds, respectively, after alignment. The smaller the RMSE value, the higher the alignment accuracy.

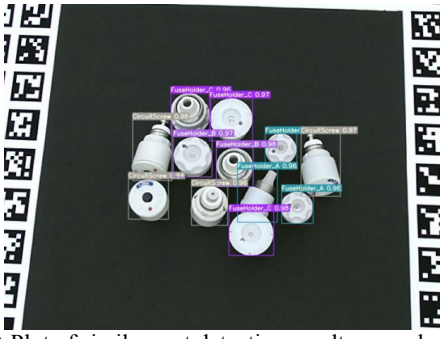
4.3 Parts recognition experiment

In order to verify the superiority of YOLOv7's part recognition detection model, we comparatively trained five popular object detection algorithms including Faster R-CNN, SSD, YOLOv3, YOLOv5, and YOLOv7, which have the same parameters and datasets in the training phase.We randomly divide the dataset into training, validation and test data with 8:1:1, the batch size is set to 16, the initial learning rate is 0.01, the training cycle is 300, and the size of the input image is 640×640. The results of the experiment are shown in table 1.

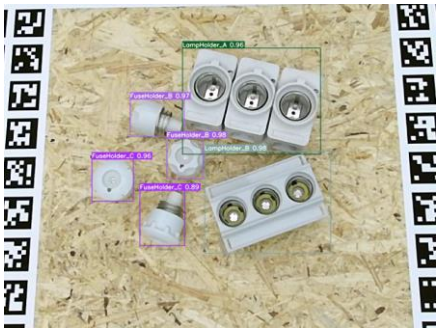
Table 1. Comparison experiment of detection algorithms.

Method	Model size(M)	mAP	FPS
Faster R-CNN	108	0.695	11
SSD	92.6	0.707	46
YOLOv3	117	0.771	62
YOLOv5	51	0.893	66
YOLOv7	46.5	0.908	73

First, it can be seen that the model parameter size of our proposed YOLOv7 model is 46.5M, which is easier to deploy on industrial vision inspection terminals compared to algorithms such as Faster R-CNN, SSD, YOLOv3, and so on, and thus can be used for real-time inspection of parts. Secondly, the mAP of YOLOv7 reaches 0.908, which is significantly higher than other models. Finally, comparing the FPS of the model, which is used as an index to evaluate the target detection speed, the FPS of YOLOv7's model reaches 73, which can satisfy the real-time requirements of detection, and the detection results of YOLOv7's model are shown in figure 2, from which it can be seen that when the parts are located in the complex background, or the parts are in the stacked situation, etc., the model can successfully recognize the different parts, and has a high accuracy rate, with almost no missed detections and false alarms.



(a) Plot of similar part detection results on a clean background



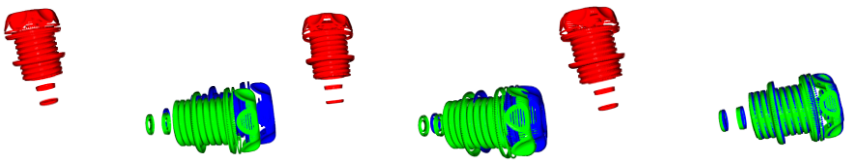
(b) Plot of part detection results in the stacked scenario

Fig. 2. Plot of part detection results.

4.4 Posture estimation experiment

In order to verify the effectiveness of the improved ICP algorithm, experiments are conducted on previously prepared point cloud datasets to compare with the original ICP algorithm and other existing point cloud alignment algorithms, and the main evaluation metrics include alignment accuracy as well as computation time.

Fifty images are selected from the test dataset, and the RGB images are inputted into the YOLOv7 network to obtain multiple ROI regions, and then through the mapping of the 2D ROI regions to the 3D point cloud space, the local point cloud data containing the target part is calculated, and then it is aligned with the corresponding template files in the template point cloud dataset, and the results of the point cloud alignment are visualized as in the figure 3. The original ICP algorithm, SAC-IA+ICP method and SAC-IA + improved ICP algorithm were used to align the extracted point cloud respectively, the RMSE value of each alignment was recorded, the alignment results of all the test samples were statistically analyzed, and the average RMSE value was calculated, the experimental results are shown in the table 2.



(a) Using the ICP algorithm. (b) Using the SAC-IA+ICP algorithm.
(c) Using the SAC-IA+Improved ICP Algorithm.

Fig. 3. Plot of point cloud alignment results (Red is the template point cloud, green is the source point cloud, and blue is the result after alignment).

Table 2. Statistics of alignment results for different parts using different methods.

Part Type	ICP			SAC-IA+ICP			SAC-IA+ Improved ICP Algorithm		
	E_r(°)	E_t(mm)	Time(s)	E_r(°)	E_t(mm)	Time(s)	E_r (°)	E_t(mm)	Time(s)
Part A	3.46	4.496	40.7	2.58	2.586	21.1	0.38	0.739	22.0
Part B	7.95	9.527	33.5	3.43	7.198	29.3	0.92	1.005	28.8
Part C	4.02	4.06	62.5	2.92	1.6	45.1	0.32	0.278	45.7

5 Summarize

In this study, the YOLOv7 algorithm effectively realizes the fast identification and localization of texture-free stacked parts, and separates the target parts from the complex background by 2D to 3D mapping and depth thresholding and Euclidean clustering techniques. Using SAC-IA and the improved ICP algorithm for point cloud alignment, the introduced adaptive weights and global optimization strategies substantially improve the accuracy and robustness of the position estimation, ensuring accurate 6D position estimation of the parts to support robot gripping and automated assembly. Tests show that this method can quickly and accurately identify parts with different shapes and acquire 6D position to meet the requirements of industrial applications, marking a significant progress in this field, and algorithm optimization and application extensions will be explored in the future to enhance system versatility.

References

1. Hinterstoisser S., Holzer S., Cagniart C., et al. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes[C]// IEEE International Conference on Computer Vision, ICCV, Barcelona, Spain, November 6-13, 2011: 858-865.
2. Hinterstoisser S., Cagniart C., Ilic S., et al. Gradient response maps for real-time detection of textureless objects[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2012, 34(5):876-888.
3. Hinterstoisser S., Lepetit V., Ilic S., et al. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes[C]// Proceedings of the 12th international conference on Computer Vision - Volume Part III. Springer-Verlag, 2012: 548-562.
4. Zhang Z. Iterative point matching for registration of free-form curves and surfaces[J]. International Journal of Computer Vision, 1994, 13(2): 119-152.
5. Wu X R,Huang G M,Sun L N.Fast visual identification and location algorithm for industrial sorting robots based on deep learning[J].Robot,2016,38(6):711-719.
6. Du X D,Cai Y H,Lu T,et al.A robotic grasping method based on deep learning[J].Robot,2017,39(6):820-828,837.
7. Xia J,Qian K,Ma X D,et al.Fast planar grasp pose detection for robot based on cascaded deep convolutional neural networks [J].Robot,2018,40(6):794-802.
8. Su H,Qi C R,Li Y,et al.Render for CNN:Viewpoint esti mation in images using CNNs trained with rendered 3D model views[C]//IEEE International Conference on Computer Vision. Piscataway,USA:IEEE,2015:2686-2694.
9. C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint, 2022, doi:10.48550/arXiv.2207.02696.

10. Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations; proceedings of the Seventh IEEE International Conference on Computer Vision, F, 1999 [C].
11. T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis and X. Zabulis, "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 2017, pp. 880-888, doi: 10.1109/WACV.2017.103