IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Simulation of Self-Occlusion Virtual Dataset Method for Robust Point Matching Algorithm, With Applications to Positioning of Guide Vanes

FENGLIN HAN[1,2], HANG PENG[2], XIAN WU[2], HAONAN REN[2], YIWEI SUN[3], AND BIN SU[2]

[1]State Key Laboratory of Precision Manufacturing for Extreme Service Performance, Central South University, Changsha, Hunan 410083 China
[2]College of Mechanical and Electrical Engineering, Central South University, Changsha, Hunan 410083 China
[3]Dundee International Institute of Central South University, Changsha, Hunan 410083 China

Corresponding author: Bin Su (e-mail: 223711075@csu.edu.cn).

This work was supported by the National Natural Science Foundation of China (Grant No.52227806), the Science and Technology Innovation Program of Hunan Province (Grant No.2022GK4027), Science and Technology Plan Project of Changsha City (Grant No.kh2301024), and Central South University Research Programme of Advanced Interdisciplinary Studies (Grant No.2023QYJC027).

ABSTRACT The application of point cloud registration technology for workpiece positioning compensation using optical three-dimensional measurement methods has attracted widespread attention in the manufacturing industry, particularly point cloud registration methods integrated with deep learning are booming. Since the training of current deep learning registration methods is often based on public datasets, however, the performance of point cloud registration of guide vanes depends on the relevance, quality, and quantity of the training dataset, if the training is directly based on the current public dataset directly used for the guide vanes, the accuracy of the registration criteria cannot meet the requirements, and secondly, in real industrial scenarios, manually obtaining the real dataset is time-consuming, labor-intensive and error-prone. To address these issues, this paper proposes a virtual simulation method based on the CAD model of the workpiece to set up a virtual camera so that a large number of near-real datasets can be generated quickly. The method can simulate the incomplete vanes point cloud obtained by real shooting due to self-occlusion by setting multi-angle virtual cameras on the hemispherical surface wrapped in the CAD model. The experimental results show that the combination of the deep learning registration method and the virtual dataset method in this paper can improve the accuracy, efficiency, and stability of the deep learning registration for workpiece positioning compensation, which has a good prospect for practical application.

INDEX TERMS Guide vanes registration, Virtual dataset, Deep learning, Guide vanes Positioning.

I. INTRODUCTION

THE guide vanes are the vital component that determines the energy conversion efficiency, reliability, and service life of an aero-engine [1]–[3]. In order to ensure the performance of the guide vanes and long-term stability under extreme working conditions, it is indispensable to strictly control the geometric and dimensional accuracy of the guide vanes, especially for the more complex and precise multi-connected guide vanes, whose machining and manufacturing errors may directly determine the overall performance of the aero-engine.

Blanks of guide vanes before machining are usually an accurate investment casting, and the vanes profile will be used as a datum to machine a new datum before cooling hole machining. In the event of a significant error in the positioning of the coarsest datum, this error will be progressively amplified with each subsequent datum conversion. Therefore, the rapid and precise positioning of the vanes prior to the initial rough machining is of paramount importance to the subsequent quality of the machining process. In the contemporary guide vane manufacturing industry, vane positioning usually requires manual alignment to fine-tune the machining G-code generated from the CAD model. Consequently, the accuracy of rough machining operations is contingent upon the experience of the operator, which does not guarantee the consistency and efficiency of machining [4]. In order to ensure the rough machining accuracy of the vanes, a novel compensation method for the positioning error is urgently needed.

It is well known that point cloud registration technology has been widely employed in the positioning compensation of machining processes and the measurement problems of

workpieces, especially, deep learning-based point cloud registration methods have demonstrated clear advantages in registration accuracy, stability, and speed. These methods have been widely recognized and adopted by scholars due to their high robustness to noise, low overlap, and the ability to perform local-to-whole registration. Aoki et al. [5] proposed the PointNetLK algorithm, which modifies the Lucas and Kanade (LK) algorithm and integrates PointNet [6] and LK algorithms into a trainable recurrent deep neural network, which minimizes the difference in the global features of the two pieces of the point cloud. Sarode et al. [7] introduced a point cloud registration network (PCRNet), which uses PointNet to encode the point cloud and input it into a multilayer perceptron layer (MLP) to solve the change matrix, improving the efficiency and robustness of the algorithm. Wang et al. [8] proposed the deep closest point (DCP), which is based on the Transformer network structure to get the soft correspondence between two pieces of point clouds, and then find the optimal registration relationship. Yew and Lee [9] proposed robust point matching (RPM), which uses a feature extraction network to learn a mixture of features for each point containing spatial coordinates and local geometric information to obtain a soft match of the corresponding point, which enhances the robustness of rigid point cloud registration.

However, due to the specificity of the shape and machining of the multi-connected guide vanes, the direct selection of one of the above deep learning registration methods for vanes positioning compensation also has the following two challenges: firstly, the network training of these deep learning point cloud registration methods is often based on commonly used public datasets such as ModelNet40, etc. [10], [11] Since these public datasets do not contain a specific guide vane model required for industrial machining if they are directly used for vane point cloud registration for positioning compensation the effect is not good, so there is a need to seek a method that can collect large amounts of point cloud datasets of guide vanes for many times. Secondly, if the camera is used to collect vane point cloud data several times to create a dataset, it will inevitably face the problem of collecting insufficient training datasets and taking a long time to collect them, so it is necessary to have a high efficiency of this method of collecting datasets. Overall, there is an urgent need to find a method to quickly generate a large number of guide vane point clouds close to the real shooting scene in order to build up the dataset and improve the performance of the algorithm.

In recent years, the application of virtual simulation technology to obtain simulated datasets has shown great promise in network training to improve algorithm performance. Bochinski et al. [12] proposed to utilize a method based on the automatic generation of virtual world datasets to train CNN classifiers using virtual world datasets, which achieves results comparable to classifiers trained on real-world datasets. Yang et al. [13] proposed a method to build a virtual dataset to address the difficulty of collecting industrial datasets by constructing a 3D model of the sprayed workpiece and the factory environment in a virtual environment automatically generating the labeled data, and finally using real datasets, self-constructed virtual datasets, and hybrid datasets for model training, which confirms that the virtual dataset can significantly improve the accuracy of the trained model. Xue et al. [14] proposed a virtual and real fusion dataset for ViRFD, which improves the segmentation accuracy of the rock segmentation algorithm model by fusing the dataset generated by the virtual engine Unity with the real dataset. Therefore, in response to our aforementioned need to quickly generate a large amount of guided vane point cloud data close to the real shooting scene, virtual simulation technology can provide inspiring ideas for our work, so as to obtain a near-real virtual vane dataset to train deep learning algorithms to improve the algorithm performance.

However, the special structure of the multi-connected guide vanes using virtual simulation technology to obtain the virtual dataset in this paper also has the following challenges: firstly, the multi-connected guide vanes have the phenomenon of self-occlusion in the process of acquiring the vane dataset by actual shooting, and it is not entirely applicable to this paper to directly use the aforementioned virtual simulation methods to generate virtual datasets. Meanwhile, in order to quickly generate a guide vane dataset close to the real shooting, creating a dataset by just manually clipping the CAD-generated point cloud does not reflect the real situation where the point cloud is partially visible due to self-occlusion.

Therefore, In order to solve the problem of poor accuracy of the selected deep learning registration algorithm for guide vanes positioning compensation, this paper will propose a method based on virtual simulation technology to quickly generate a near-real guide vanes point cloud dataset, which can realistically simulate the guide vanes self-occlusion caused by the partially visible point cloud of the real situation, and the generated dataset will be fed into the learned registration algorithm for training to improve the performance of the network to achieve more accurate compensation of workpiece positioning before machining. The main layout of this paper is as follows: (1) take the given guide vanes as the research object, select the high-performance deep learning registration algorithm adapted to the object, (2) propose a virtual simulation method based on the CAD model to set up a virtual camera so that a large number of datasets can be generated quickly to improve the accuracy of the network registration after training effectively.

In summary, the contributions of this article are as follows: We propose a method to quickly generate a simulation dataset based on a workpiece CAD model setup with a virtual camera. By using the dataset generated by this method to train a deep learning registration algorithm significantly improves the accuracy and stability of the algorithm in guide vanes registration, thus reducing the positioning compensation error before machining of the workpiece.

II. SELECTION OF HIGHLY ROBUST ALGORITHM

In this section, in order to select the deep learning registration algorithm with high robustness and stability to adapt the

partial point cloud with noise points to the complete smooth point cloud, we will comprehensively compare the registration effect of current traditional registration algorithms and learning-based registration algorithms, and before the comparison, we will propose face-to-face mean distance error (FFMDE) to more accurately evaluate the registration results and the vane positioning effect.

In order to evaluate the performance advantages of each algorithm, the evaluation metrics used include 1) average registration time is used to evaluate the network registration speed and characterize the algorithm efficiency; 2) root mean square error (RMSE), which represents the square root of the squared mean value of the difference between the actual value and the predicted value, and is used as a measure of the distance or error between the two-point clouds as shown in (1):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \tag{1}$$

where N is the total number of points, and x_i is the point in the source point cloud, and y_i represents the corresponding point in the reference point cloud.

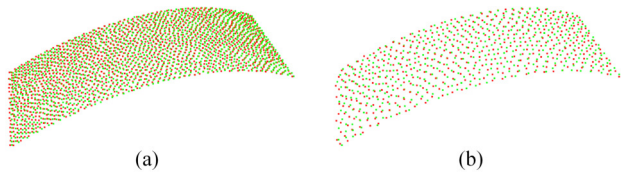


FIGURE 1. Relationship between RMSE and density.

The smaller RMSE values mean that the two pieces of the point cloud are closer to each other, but in fact, as shown in Fig. 1, under the same degree of registration, the RMSE is closely related to the density of the point cloud, the higher the density of the two pieces of the point cloud, the closer the distance between the points, the smaller the calculated RMSE is, which means that the density of the point cloud is also a variable that affects the result of the registration. However, it is difficult to ensure the consistency of this variable in the pre-processing of the photographed point clouds, which makes it unreliable to judge the registration effect only by the RMSE value after the registration. Therefore, for the aviation vane surface point cloud, this paper proposes a more applicable evaluation metrics, the face-to-face mean distance error (FFMDE), which can well solve the problem of the failure of the discriminative registration effect caused by the difference in the density of the point cloud after the preprocessing, and more reasonably reflect the degree of fit of the two vane profiles.

As shown in Fig. 2, for each point x_i in the source point cloud, search for its nearest point in the reference point cloud as the corresponding point y_i, and according to the unit normal vector of its corresponding point, construct FFMDE calculated as in (2):

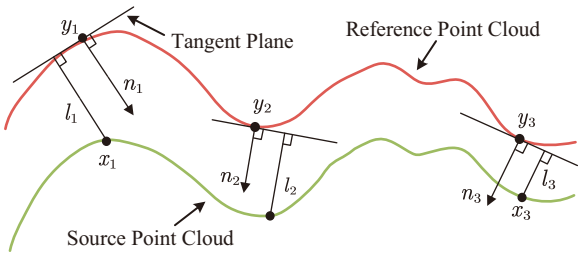


FIGURE 2. Schematic of calculating the face-to-face mean distance error(FFMDE).

$$FFMDE = \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{n_i}\cdot\overrightarrow{x_iy_i} \tag{2}$$

where $\overrightarrow{x_iy_i}$ denotes a set of connecting line vectors of the corresponding points.

Due to the difference in geometric structure between the source point cloud and the reference point cloud, the source point cloud data will be missing due to occlusion and the presence of noise data will fluctuate, however, the reference point cloud converted from the CAD model is complete, smooth and homogeneous, which increases the difficulty of the algorithm's registration, so it is necessary to choose deep learning registration algorithms that have high robustness and stability in order to reduce these negative effects.

To comprehensively analyze the performance of the registration algorithms, a comparison test of the current common registration algorithms will be conducted, including the traditional point cloud registration algorithms such as SAC-IA, ICP, and FGR, and the deep learning-based point cloud registration algorithms represented by PointNetLK, DCP, RPM, and PCRNet, and finally, the algorithms that are most suitable for this paper will be selected as the beneficiary of the virtual simulation technology. All the above algorithms were subjected to several parameter adjustments, and the best registration results are shown in Fig. 3, where the alignment effect of the learning-based algorithm is after training on the public dataset ModelNet40.

TABLE 1. Experimental results on test data

Methods	RMSE(mm)	FFMDE(mm)	Time(s)
ICP	8.8914	3.7936	0.0344
SAC-IA+ICP	2.2983	1.1315	9.7802
FGR	12.3026	7.5014	2.9570
DCP	11.3498	4.9108	8.1425
PointNetLK	6.2657	2.7085	0.8754
PCRNet	8.4870	3.8836	0.4218
RPM	0.8982	0.4568	3.9138

The results of the algorithm comparison are shown in Table I, and it can be seen that the registration error of the RPM algorithm is much lower than that of other registration algorithms, with the RMSE of 0.8982mm, the FFMDE of 0.4568mm, and the speed of registration is also faster compared to other, with an average registration time of 3.9138s.

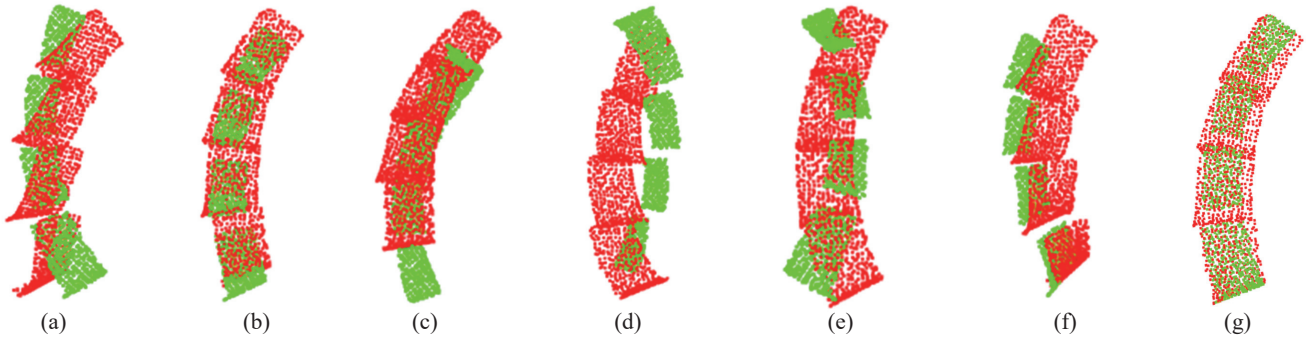


FIGURE 3. **Comparison of algorithmic registration visualizations. (a) ICP; (b) SAC-IA+ICP; (c) FGR; (d) DCP; (e) PointNetLK; (f) PCRNet; (g) RPM.**

Although the RPM algorithm has a greater advantage in accuracy compared to other algorithms, the direct application of the guiding vane registration still does not meet the current needs, so the RPM algorithm is chosen to be the beneficiary of the method of obtaining the dataset by the virtual simulation technology. In the following sections will be expressed in detail how to use the method to improve the RPM algorithm on the guide vanes registration accuracy.

III. VIRTUAL DATASET ACQUISITION METHOD

A. OVERALL OPERATION PROCESS

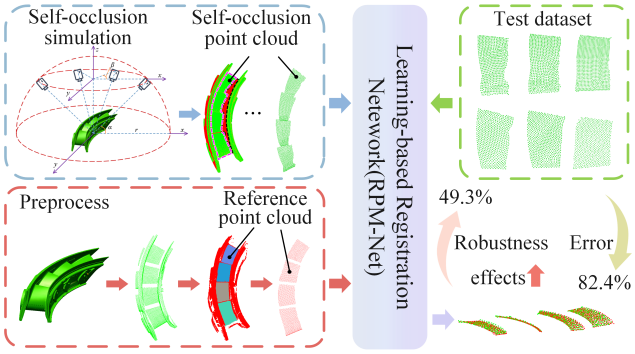


FIGURE 4. **Schematic of the overall operational flow of the proposed methodology.**

We intend to propose a method to set up a virtual camera around the CAD model of the workpiece and generate a large number of high-quality simulation datasets that meet the application requirements, as shown in Fig. 4. We first simulate the real shooting scene of the structured-light depth camera used in the experiment by setting up the internal parameters of the virtual camera and the capture attitudes, so as to obtain several sets of point cloud data of guide vanes with different degrees of self-occlusion under different capture attitudes. The point cloud data of guide vanes with varying degrees of self-occlusion under different capture postures are obtained. Then the label value corresponding to each set of data is obtained by setting up a virtual camera coordinate system. Then the preprocessed virtual vane point cloud data are inputted into the algorithmic network for training. Then several sets of vane data with the largest degree of missingness are selected as the test dataset to validate the effectiveness of the method, which is greatly improved in terms of both accuracy and robustness.

B. VIRTUAL CAMERA SETTINGS

The virtual camera setup consists of two main parts: internal parameter settings and capture pose simulation, the former to adjust the data characteristics of the point cloud to be as consistent as possible with the data acquired by the depth camera, and the latter to control the relative position of the virtual camera to the workpiece to obtain the point cloud data under different viewpoints.

1) Internal parameter settings

For the virtual camera setup, Visual C++ combined with PCL (Point Cloud Library version 1.13.1) is used to set up the virtual camera, and it is known that the parameters of the currently adopted structured light camera are shown in Table II, and according to the camera parameters, the internal parameters of the virtual camera are set up using the functions of the PCL as follows:

TABLE 2. **Virtual camera internal parameters**

Parameters	Value
Working Distances(mm)	300–1300
Near Field of View(mm)	230×140@0.3m
Far Field of View(mm)	1000×600@1.3m
Measurement Accuracy(mm)	±0.1–1
Resolution	1280×1024
3D Capture Time(s)	0.8–1.2

(1) The virtual camera working distance is set using the *setCameraClipDistances*() function, and its value is 300-1300mm;

(2) Virtual camera field of view, using the *setCameraFieldOfView*() function to set, the value should be based on the depth of the camera proximal field of view or the distal field of view calculated by the formula as in (3), here the choice of the proximal field of view parameter value is brought into the set to 0.73 rad, which is 41.9°.

$$\tan(\frac{\omega}{2}) = \frac{\overline{ab}}{2 \times \overline{oc}} \tag{3}$$

Where $\overline{ab}$ is the diagonal of the field of view, $\overline{oc}$ is the working distance, and ω represents the camera's field of view.

(3) The virtual camera resolution, which determines the size of the point cloud density, is set through the *renderView*() function, and here it is set to 1280×1024.

2) Camera capture pose simulation

Algorithm 1: Camera capture pose simulation algorithm based on CAD model.

Input: CAD model in STL format;
Output: N sets of point cloud data in different viewpoints (Vane virtual dataset), where $N = I \times J = 216$;
Initialization: Working distance $r = 400mm$, horizontal angle $\beta = 0°$, vertical angle $\alpha = 60°$, iteration $I = (85-\alpha)/5+1$, iteration $J = (350-0)/10+1$;
for $i = 0 : I$ **do**
  • *Change of vertical angle* α
  for $j = 0 : J$ **do**
    • *Change of vertical angle* β
    $\alpha = 60 + 5i, \beta = 10j$;
    • *Calculate the current viewpoints* α *and* β
    $z_{ij} = r \times \sin\alpha$;
    • *Current viewpoint of the camera relative to the z-coordinate of the workpiece*
    $x_{ij} = r \times \cos\beta \sin\beta$;
    • *Current viewpoint of the camera relative to the x-coordinate of the workpiece*
    $y_{ij} = r \times \cos\alpha \sin\beta$
    • *Current viewpoint of the camera relative to the y-coordinate of the workpiece*
  end
end

The simulation of the capture pose is based on the actual scene to construct all possible viewpoints. As shown in Fig. 4, the CAD model of the guide vanes is taken as the center, and the working distance of the camera is taken as the radius to make a hemisphere. Since the optimal shooting distance of the depth camera in the actual application is about 300–500mm, and the point cloud data will be inaccurate or incomplete if it is too far or too close. Therefore, the working distance of the virtual camera takes its middle value, that is, 400mm, so that the whole hemisphere contains all the viewpoints of the virtual camera. In the meantime, to ensure that the virtual camera can collect the point cloud containing the vane profile, which means that the camera setup needs to have a certain height, so it is necessary to set the vertical angle α between the connection line (between the virtual camera and the center of the sphere o) and the ground plane, and this paper initially set the angle α to 60°. Finally, in order to ensure that the collected dataset has different degrees of self-occlusion, the viewpoints should be uniformly taken out from the hemisphere, i.e., keep the vertical angle α constant, change the horizontal angle β between the line connecting the virtual camera and the center o' of the circle and the x-axis, and allow the virtual camera to rotate around the axis of rotation to generate a series of viewpoints one by one and the specific operation pseudo-code is shown in Algorithm 1.

As can be seen from Algorithm 1, the vertical angle α increases from 60° to 85°, each time increasing by 5°, the number of changes is 6. In contrast, with each change in the angle α, the horizontal angle β will increase from 0° to 350°, in turn increasing by 10°, the number of changes is 36, and a total of 216 viewpoints can be generated, so far, all the views of the virtual camera settings are complete. due to the coordinate system of the workpiece and the world coordinate system overlapping in this paper, and the center of the workpiece is located at the origin, so the three coordinate values in each loop are brought into the function *setCameraPosition*() to complete the virtual camera capture pose simulation.

C. ACQUISITION OF VIRTUAL DATASET AND CORRESPONDING LABELS

After the virtual camera setup, the CAD model is input in STL format. The workpiece's point cloud model under different view angles is output by the *renderView*() function in PCL. Figure 5 shows a part of the virtual dataset capture, where the green part is the reserved incomplete point cloud captured by the virtual camera, i.e., the guide vanes point cloud data existed in the virtual dataset, and the red part is a series of self-occlusion point clouds due to the different viewpoints of the virtual camera, which is excluded from the virtual dataset, so as to simulate the incomplete point cloud data due to the self-occlusion in a natural way. Here, to reflect the changes of the vane profile in different viewpoints, the two-point clouds are converted to the same coordinate system, and it can be seen from Fig. 5 that this method is more in line with the real shooting scene compared with the manually cropped point cloud.

However, in reality, the generated dataset is in the virtual camera coordinate system, and the rigidity transformation matrix between it and the world coordinate system (workpiece coordinate system) is the label of the dataset, which can be used to provide the correct direction for the subsequent training of the iterative parameters of the registration network. The rigidity transformation matrix is essentially a transformation of the world coordinates where the workpiece is located into the observation coordinates relative to the position and orientation of the virtual camera, so it is first necessary to define the virtual camera, as shown in Fig. 6, that is, by the spatial position P of the virtual camera in the world coordinate system, the direction vector D, a vector U pointing up to it, and a vector R pointing to the right of it, to create a virtual camera with the virtual camera's location as its origin and the three unit axes are perpendicular to each other.

The spatial position P of the virtual camera can be obtained by the multi-view sampling code. Here, to make the direction

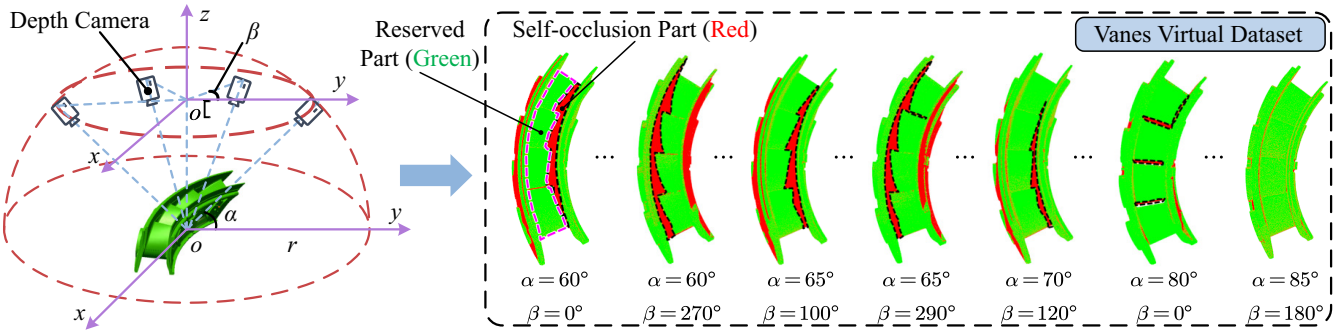


FIGURE 5. Schematic of multi-view virtual point cloud data acquisition based on vane CAD model setup with a virtual camera.

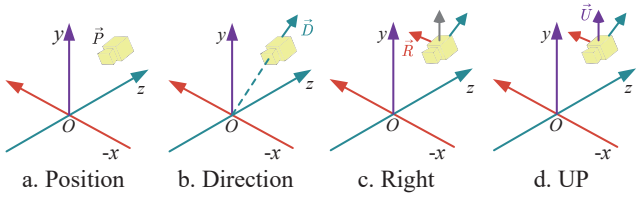


FIGURE 6. Schematic of the virtual camera coordinate system for acquisition of labels.

vector D point to the positive direction of the z-axis of the virtual camera, it is defined as pointing from the center of the workpiece (the origin of the world coordinate system) to the location of the virtual camera, and then the vector D is:

$$D = P - \{0, 0, 0\} \tag{4}$$

The right vector R represents the positive direction of the x-axis of the virtual camera coordinate system. To obtain the R vector, first define an upper vector $u_0 = \{0, 1, 0\}$, and then solve for R by cross-multiplying u_0 with the direction vector D:

$$R = u_0 \times D \tag{5}$$

The up vector U represents the positive direction of the y-axis of the virtual camera coordinate system, and according to the obtained direction vector D and the right vector R, the final up vector U is equal to the result of the cross-multiplication of the two:

$$U = D \times R \tag{6}$$

Thus, the three mutually perpendicular axes define a coordinate space, and the rigid transformation matrix M can be solved by transforming these three axis vectors plus the positional coordinate P:

$$M = \begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

D. VIRTUAL DATASET TRAINING EXPERIMENTS AND ANALYSIS

After obtaining the virtual dataset and its corresponding labels, we will first complete the pre-processing of the virtual dataset before the registration, and then both are inputted into the registration network to act as the source point cloud for training, while the reference point cloud is the CAD model point cloud (also pre-processed).

In order to evenly and reasonably divide the virtual dataset into a training dataset and test dataset, this paper takes the horizontal angle β as the distinction, starting from 0°, and increasing to 40° each time, the dataset obtained from the viewpoints are all set to be the test dataset, which generates a total of 54 groups of test dataset, and the remaining are all set to be the training dataset. It is also worth noting that, in order to increase the complexity of the training dataset, the point cloud is preprocessed and then randomly sampled, the reference point cloud is reduced from the number of points γ to 1024, and the source point cloud is reduced from the number of points X to $1024 \times X/Y$, and a Gaussian noise that satisfies the normal distribution $N(0, 0.01)$ is applied to both pieces of the point cloud to simulate the interference of the noise, to ensure that the algorithm is not the same for the training dataset in each round.

The environment used for the experiment includes Visual C++ as well as PCL (version 1.13.1) in addition to Python 3.6+Pytorch 1.10.0, which is carried out under the Windows operating system, in addition, both the training *batch_size* and the testing *batch_size* need to be set to 1 because different datasets have different size dimensions and cannot be processed in parallel. As for the aviation vane point cloud, according to the training effect, the number of neighboring points k required by the feature extraction module is set to 32 to achieve the best training effect.

1) Virtual dataset testing

Firstly, the test dataset is used to check the registration accuracy of the network at this time, in order to increase the difficulty of registration, the six viewpoints with the lowest vane profile integrity are selected from the test dataset with different vertical angles, and the results of the registration are shown in Fig. 7, with the green part as the source point

cloud obtained from the multi-view sampling conversion by the virtual camera, and the red part as the reference point cloud obtained from the conversion by the CAD model of the workpiece.

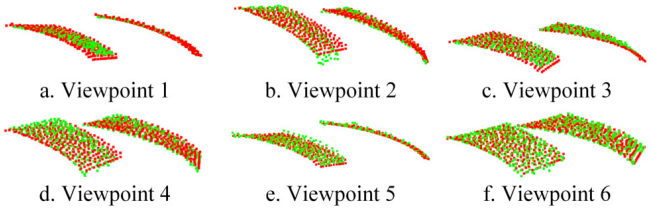


FIGURE 7. Algorithm (trained on the virtual dataset) partial viewpoint registration results (only two pieces of the profile point cloud are shown).

The above six groups of views in the two pieces of point cloud input to the algorithm network trained by the virtual dataset can complete the registration without obvious deviation, and compare it with the algorithm registration results trained by the public dataset (ModelNet40 dataset), to more accurately describe the two registration errors, the real rigidity change matrix and the algorithm (virtual dataset and ModelNet40 public dataset) estimated change matrix are converted into the rotation angle and translation distance along the three coordinate axes direction, which are used as the judgment metrics.

At the same time, we take the corresponding label values of the test dataset as the real values, take the estimated value 1 as the result of the RPM algorithm registration trained on the virtual dataset, and take the estimated value 2 as the result of the algorithm registration trained on the public dataset, and from this, we calculate the average error of averaging 6 groups of views along the three coordinate axis directions, and the comparison is displayed in Fig. 8. The angle and translation registration error along each axis is shown in (8) and (9):

$$MRE_k = \frac{\sum_{i=1}^{N} \left| {R^{(k)}}_i - {Rt^{(k)}}_i \right|}{N}, k = \{x, y, z\} \tag{8}$$

$$MTE_k = \frac{\sum_{i=1}^{N} \left| {T^{(k)}}_i - {Tt^{(k)}}_i \right|}{N}, k = \{x, y, z\} \tag{9}$$

Where MRE_k and MTE_k represent the angle and translation error in one of the x, y, z axes respectively. N is the number of viewpoints selected for the experiment. $Rt_i^{(k)}$ and $Tt_i^{(k)}$ are respectively the true rotation and translation values of the test dataset in each axis. $R_i^{(k)}$ and $T_i^{(k)}$ are respectively the estimated rotations and translations in each axis direction after the algorithm registration.

In order to compare the stability of the virtual dataset and the public dataset after the training of the registration, we calculate the rotation angle error and translation error after the average of the three coordinate axes direction of the six groups of viewpoints after the registration with the formula shown in (10) and (11), and the comparison results are shown in Fig. 9.

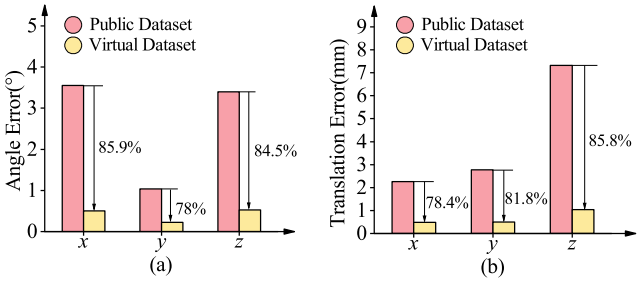


FIGURE 8. Comparison of algorithms (trained on virtual vs. ModelNet40 public dataset) for multi-view registration error.

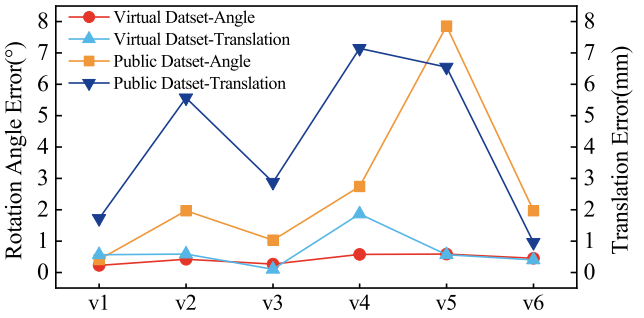


FIGURE 9. Comparison of algorithms (trained on virtual vs. ModelNet40 public dataset) for registration stability.

$$MRE_j = \frac{\sum_{i=x}^{k} \left| {R^{(i)}}_j - {Rt^{(i)}}_j \right|}{3}, j = \{1, 2, \cdots, 6\} \tag{10}$$

$$MTE_j = \frac{\sum_{i=x}^{k} \left| {T^{(i)}}_j - {Tt^{(i)}}_j \right|}{3}, j = \{1, 2, \cdots, 6\} \tag{11}$$

Where j is the number of viewpoints selected for the experiment. $k = \{x, y, z\}$ is axis label. $Rt_j^{(i)}$ and $Tt_j^{(i)}$ are respectively the true rotation and translation values of the test dataset in each viewpoint. $R_j^{(i)}$ and $T_j^{(i)}$ are respectively the estimated rotations and translations in each viewpoint after the algorithm registration.

Figure 8 illustrates that, in comparison to the RPM algorithm trained on the ModelNet40 dataset, algorithms based on the virtual dataset approach exhibit a combined reduction of approximately 80% in registration error. As illustrated in Figure 9, while the discrepancy between the training outcomes of the RPM-based algorithm on the public dataset and the virtual dataset is minimal, the overall variability is considerable, and the stability is inferior. In conclusion, for the guided vane model, the virtual dataset enhances the performance of the RPM algorithm to a certain extent in comparison to the public dataset (ModelNet40).

2) Real dataset testing

However, the depth camera acquisition data is not the same as the test dataset. Therefore, in order to verify the registration effect of the RPM algorithm after training on the virtual dataset, the following three tests are carried out: strong light

environment registration test, multi-angle shooting registration test, and multi-distance shooting registration test.

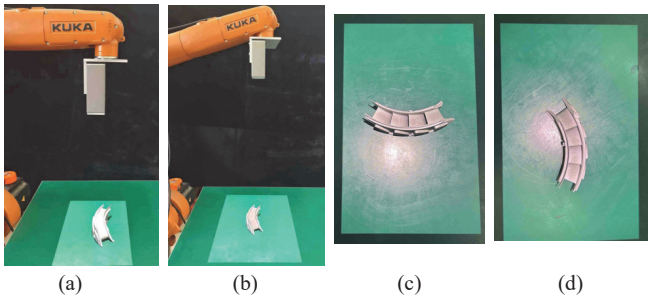


FIGURE 10. Test experiments on the real dataset. (a) Distance 30; (b) Distance 60; (c) Angle 0°; (d) Angle 270°.

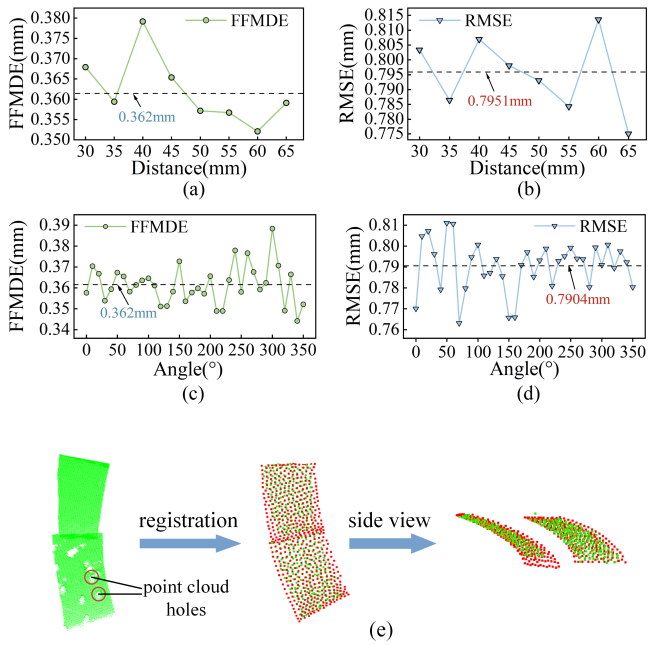


FIGURE 11. Real dataset test results. (a) Multi-distance testing (FFMDE); (b) Multi-distance testing (RMSE); (c) Multi-angle (FFMDE); (d) Multi-angle (RMSE); (e) Strong light environment registration test.

As shown in Fig. 10, we keep the guide vanes stationary by placing them horizontally on the platform, the structured light camera is connected to the sixth axis of the KUKA robotic arm and fixed on top of the workpiece, and the shooting distance and shooting angle are single-controlled by adjusting the axes of the robotic arm to validate the registration effect of the real dataset, and the strong lighting environment validation aspect individually adjusts the exposure time of the camera to create holes in the acquired point cloud to validate the robustness of the registration.

As shown in Fig. 11, in the multi-distance test, the FFMDE and the RMSE are kept at a low level with small fluctuations, with average values of 0.3620mm and 0.7951mm respectively; in the multi-angle test, the average values of the two types of registration errors are 0.3620mm and 0.7904mm respectively, which also show strong robustness; in the strong light environment test, the intuitive view of the registration does not show large deviation with a high degree of overlap. Overall, the RPM algorithm trained on the virtual dataset has a high registration performance in the real dataset, and the stability is strong.

IV. POSITION RECOGNITION ACCURACY VERIFICATION EXPERIMENT

A. POSITION RECOGNITION ACCURACY VERIFICATION SYSTEM

In order to assess the feasibility of the position recognition method in this paper, this paper develops an experimental platform for verifying the position recognition accuracy of guide vanes as shown in Fig. 12, where the main equipment includes: (1) KUKA-KR5arc 6-axis robotic arm; (2) UTECH 3D Scanner L model structured light camera, the acquisition is only 0.8-1.2s, the measurement accuracy of ±0.1-1mm, the resolution of 1280 × 1024; (3) high-precision motion platforms and general-purpose fixtures, can drive the guide vanes to realize the translational movement in the direction of X, Y, and Z coordinate axes as well as the rotation around the direction of R axis; (4) Handyprobe coordinate measuring machine (CMM), consisting of a C-Track binocular positioning sensor and a handheld probe, can achieve positioning accuracies of up to 0.015mm.

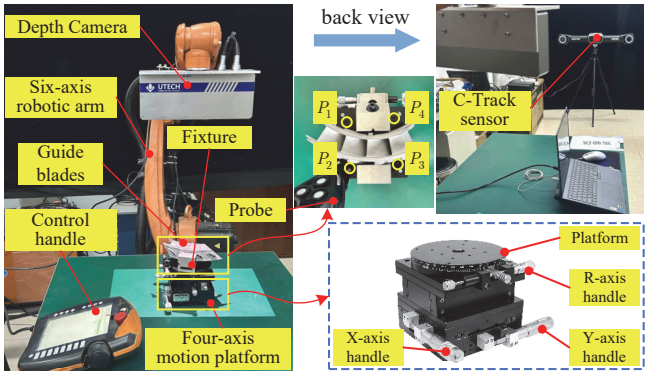


FIGURE 12. Position recognition accuracy verification system.

B. ROTATION ANGLE ACCURACY AND TRANSLATION DISTANCE VERIFICATION

1) Principle of rotation accuracy verification

The rotation angle accuracy verification refers to verifying whether the difference between the real rotation angle of the guide vane around a certain direction and the rotation angle estimated by the algorithm meets the requirements, here the rotation angle of the high-precision motion platform around the R-axis direction is taken as the real value, and the rigidity transformation matrix obtained by the algorithm estimation is taken as the estimation, which is finally transformed into the rotational motion around the R-axis direction to compare

with the real value based on the theorem of the equivalent rotational axis and the equivalent rotational angle.

In order to verify the stability of the position identification method in this paper, the guiding vanes are allowed to rotate around the R-axis direction of the high-precision motion platform 11 times, the corresponding angular value is recorded, and the initial position angle is subtracted as the true value. Then the camera is used to scan the aerial vanes in these 12 groups of positions (including the initial position) because the practical application is to align the vane acquisition point cloud and the model point cloud to carry out the position estimation, to ensure that the error source is the same, so it is necessary to use this paper's optimization algorithm to align these 12 groups of point cloud data with the vane CAD point cloud, respectively and then converted to the rigidity change matrix relative to the initial position under the rotational as shown in Fig. 13. The rigidity change matrix between the rotated point cloud and the initial position point cloud is $T = T_2 T_1^{-1}$, where T_1 and T_2 are obtained by the algorithm in this paper.

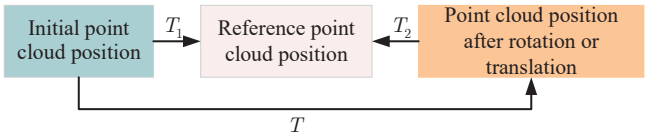


FIGURE 13. Schematic of the calculation principle of the transform matrix after translation or rotation

2) Principle of translation accuracy verification

In order to verify that the difference between the true distance between the two relative positions of the vane under translational motion and the distance estimated by the algorithm meets the requirements. The Handyprobe CCM is used to measure the true translation matrix and complete the relative position calibration with the depth camera so that all of them are converted to the light pen meter coordinate system for comparison.

In this paper, a 60mm×60mm×60mm cube is used for calibration, and its flatness and perpendicularity are both 0.02mm. The main principle is to use the three orthogonal faces of the cube to determine its accurate position under the coordinate system of the depth camera and the light-pen surveyor, respectively, and then the three faces of the point cloud data under the two positions are used for the registration, so as to get the rigidity transformation matrix between the two coordinate systems.

The Handyprobe CMM fixed on a tripod, the structured light camera fixed at the end of the robotic arm, and the calibration block will be fixed using the fixture tilt, so that the three orthogonal surfaces are simultaneously located in the depth camera and the shooting area of the C-Track, the use of the depth camera acquisition and the handheld probe to obtain orthogonal surfaces, respectively, the effect shown in Fig. 14.

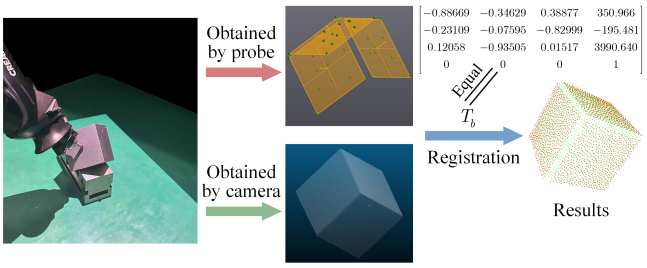


FIGURE 14. Schematic of the calibration process (obtaining the conversion matrix between the structured light camera and C-Track).

Using CloudCompare to convert the plane model obtained by the probe to point cloud data, the commonly used SAC-IA coarse registration and ICP fine registration, the orthogonal plane registration results are shown in Fig. 14, to reduce the calibration error, the calibration block is placed at different positions for a total of 8 groups of calibration, and the 8 groups of calibration matrix is converted to Euler angles and translation distance along the three coordinate axes. The final calibration result is T_b.

To avoid involving rotation motion, a single translation motion is provided by the manual control of the high-precision motion platform, and at the same time, as shown in Fig. 12, to avoid introducing human error, the coordinates of the four concave points with high repeatability on the platform are measured with the probe for each translation motion, and the average of the translation value of each point relative to the initial position is taken as the real translation value of the vanes, and the real translation value is calculated as in (8), then compare it with the translation value of the algorithm registration matrix T_{CCM} under the CCM coordinate system, the conversion relationship is as in (9):

$$ {}^{j}_{1}P = \frac{1}{4}\sum_{i=1}^{4}({}_{1}P_i - {}_{j}P_i) \tag{12} $$

$$ T_{\mathrm{CCM}} = T_b T_{\mathrm{camera}} {T_b}^{-1} \tag{13} $$

Where the relative position under the registration matrix T_{camera} is also converted by the principle of Figure 13 to get. j is the number of groups, i is the number of points, ${}^{j}_{1}P$ is the translation value for groups j to 1.

3) Experimental results

The results of the translation distance verification and rotation angle verification experiments are shown in Fig. 15(a) and Fig. 15(b), from which it can be seen that the maximum angular error of the RPM algorithm optimized by the method in this paper is no more than 0.32°, which is lower than the required 0.5°, and the average error is only 0.142°; the maximum translation error is within 0.85 mm, and the average error is 0.692mm, which also meets the actual processing requirements. Since each group of experiments was carried out two times of the vane point cloud and reference point cloud registration for conversion, and the calibration introduced

errors in the translation accuracy verification, the actual error is smaller compared with the experimental results.

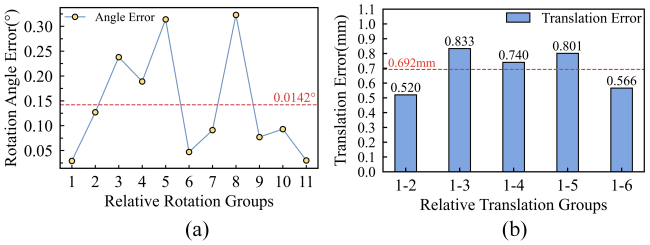


FIGURE 15. Experimental results: (a)Rotation accuracy verification; (b) Translation accuracy verification.

V. CONCLUSION

In this article, we combined the virtual simulation technology and the RPM deep learning registration algorithm to provide the RPM network with a large number of high-quality virtual datasets of specific guide vanes, which improves the network's registration performance for our desired guide vanes and improves the accuracy of the RPM algorithm applied to the positioning compensation of guide vanes prior to their machining. Firstly, in order to more accurately reflect the positioning compensation error, the face-to-face mean distance error (FFMDE) is proposed, and then we evaluated and selected the RPM algorithm to be applied to the positioning compensation of guide vanes. In addition, we designed an algorithm to set up a multi-angle virtual camera to generate a simulation dataset based on the workpiece model and trained with the dataset generated by this algorithm to improve the alignment performance of the RPM algorithm. Finally, by carrying out multi-scene registration experiments and comparisons, we verified that the optimized registration effect of this paper based on the virtual simulation technology for the RPM algorithm dataset meets the process requirements.

Nevertheless, the proposed method still has some deficiencies and some future work should be done. In the validation experiment, limited by the accuracy of the instrument, the coordinate conversion process between the instruments introduced multiple systematic errors, overestimating the error of the algorithm in this paper; secondly, the environment of the experiment is still different from the real machining environment, for example, the cutting fluid, cutting chips and different light sources in the machine tool machining environment are not taken into account, in the future work, we investigate how to solve the above problems.

REFERENCES

[1] L. Yang and Z. Zhang, "A conditional convolutional autoencoder-based method for monitoring wind turbine blade breakages," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6390–6398, Sep. 2021.
[2] Z. Liu, X. Tang, X. Wang, J. E. Mugica, and L. Zhang, "Wind turbine blade bearing fault diagnosis under fluctuating speed operations via bayesian augmented lagrangian analysis," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4613–4623, Jul. 2021.
[3] J. H. Perepezko, "The hotter the engine, the better," *Science*, vol. 326, no. 5956, pp. 1068–1069, Nov. 2009.
[4] X.-C. Xi, J. Wang, S.-M. Zhu, J.-Y. Ma, and W.-S. Zhao, "Adaptive drilling of film cooling holes of turbine vanes based on registration of point clouds," *IEEE Trans. Ind. Informat.*, vol. 19, no. 12, pp. 11 920–11 928, Dec. 2023.
[5] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 7156–7165.
[6] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 77–85.
[7] V. Sarode, X. Li, H. Goforth, Y. Aoki, R. A. Srivatsan, S. Lucey, and H. Choset, "Pcrnet: Point cloud registration network using pointnet encoding," 2019, *arXiv:1908.07906*, DOI:10.48550/arXiv.1908.07906.
[8] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Seoul, Korea (South), Oct. 2019, pp. 3522–3531.
[9] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 11 821–11 830.
[10] Y. Wang and J. Solomon, "Prnet: self-supervised learning for partial-to-partial registration," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, Vancouver BC, Canada, Dec. 2019, pp. 8814–8826.
[11] L. Yang and Z. Zhang, "A conditional convolutional autoencoder-based method for monitoring wind turbine blade breakages," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6390–6398, Jul. 2021.
[12] E. Bochinski, V. Eiselein, and T. Sikora, "Training a convolutional neural network for multi-class object detection using solely virtual world data," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance*, Colorado Springs, CO, USA, Aug. 2016, pp. 278–285.
[13] K. Yang, L. Zhao, and C. Wang, "Workpiece tracking based on improved siamfc++ and virtual dataset," *Multimedia Syst.*, vol. 29, no. 6, pp. 3639–3653, Oct. 2023.
[14] Z. Xue, L. Chen, Z. Liu, Y. Liu, and W. Mao, "Virfd: A virtual-realistic fused dataset for rock size analysis in tbm construction," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13 485–13 498, Mar. 2022.

FENGLIN HAN received the B.S. degree in mechanical engineering in 2004 from Chongqing University, Chongqing, China, the Ph.D. Degree in mechanical engineering in 2010 from Huazhong University of Science and Technology, Wuhan, China.

Now he is an associate professor in Central South University, Changsha, China. He is working on the field of soft robotics design and control technologies.

HANG PENG received the B.S. degree in mechanical design and manufacturing and automation from School of Mechanical and Electrical Engineering, Southwest Petroleum University, Chengdu, China, in 2022, and is currently pursuing the M.S. degree in mechanical engineering from College of Mechanical and Electrical Engineering, Central South University, Changsha, China.

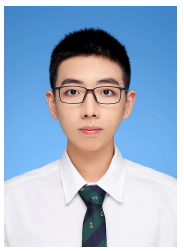
His current research interests include intelligent detection and point cloud registration.

XIAN WU is now a student from the College of Mechanical and Electrical Engineering in Central South University, majoring in Mechanical Design, Manufacturing and Automation.

He is current interested in the researches of industrial robot with its applications and point cloud data processing, including point cloud registration and segmentation, as well as point cloud classification.

HAONAN REN received the B.S. degree in Mechanical design, manufacturing and automation from College of Mechanical and Electrical Engineering, Central South University, Changsha, China, in 2023, and is currently pursuing the M.S. degree in mechanical engineering from College of Mechanical and Electrical Engineering, Central South University, Changsha, China.

His current research interests include intelligent detection and point cloud registration.



YIWEI SUN is now a student from Dundee International Institute of Central South University, majoring in Mechanical Design and Manufacturing and Automation.

His current research interests are in 3D optical measurement and point cloud registration.

BIN SU received the B.S. degree in mechanical design and manufacturing and automation from School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China, in 2021, the M.S. degree in mechanical engineering from College of Mechanical and Electrical Engineering, Central South University, Changsha, China, in 2024.

His current research interests include intelligent detection and point cloud registration.